23.0

Visma Severa API Data Model

for use with Invantive SQL

invantive
the SQL company

Copyright

Important Safety and Usage Information

Intended Use and Limitations: This software, developed by Invantive, is designed to support a variety of business and information technology data processing functions, such as accounting, financial reporting an sales reporting. It is important to note that this software is not designed, tested, or approved for use in environments where malfunction or failure could lead to life-threatening situations or severe physical or environmental damage. This includes, but is not limited to:

- Nuclear facilities: The software should not be used for operations or functions related to the control, maintenance, or operation of nuclear facilities.
- Defense and Military Applications: This software is not suitable for use in defense-related applications, including but not limited to weaponry control, military strategy planning, or any other aspects of national defense.
- Aviation: The software is not intended for use in the operation, navigation, or communication systems of any aircraft or air traffic control environments.
- Healthcare and Medicine Production: This software should not be utilized for medical device operation, patient data analysis for critical health decisions, pharmaceutical production, or medical research where its failure or malfunction could impact patient health.
- Chemical and Hazardous Material Handling: This software is not intended for the management, control, or operational aspects of chemical plants or hazardous material handling facilities. Any malfunction in software used in these settings could result in dangerous chemical spills, explosions, or environmental disasters.
- Transportation and Traffic Control Systems: The software should not be used for the control, operation, or management of transportation systems, including railway signal controls, subway systems, or traffic light management. Malfunctions in such critical systems could lead to severe accidents and endanger public safety.
- Energy Grid and Utility Control Systems: This software is not designed for the control or operation of energy grid systems, including electrical substations, renewable energy control systems, or water utility control systems. The failure of software in these areas could lead to significant power outages, water supply disruptions, or other public utility failures, potentially endangering communities and causing extensive damage.
- Other High-Risk Environments: Any other critical infrastructure and environments where a failure of the software could result in significant harm to individuals or the environment.

User Responsibility: Users must ensure that they understand the intended use of the software and refrain from deploying it in any setting that falls outside of its designed purpose. It is the responsibility of the user to assess the suitability of the software for their intended application, especially in any scenarios that might pose a risk to life, health, or the environment.
Disclaimer of Liability: Invantive disclaims any responsibility for damage, injury, or legal consequences resulting from the use or misuse of this software in prohibited or unintended applications.

Contents

1 SQL Driver for Visma Severa API

Invantive UniversalSQL is the fastest, easiest and most reliable way to exchange data with the Visma Severa API.

Use the "Search" option in the left menu to search for a specific term such as the table or column description. When you already know the term, please use the "Index" option. When you can't find the information needed, please click on the Chat button at the bottom or place your question in the user community. Invantive Support or other users will try to help you.

Visma Severa is cloud-based project management software. Visma Severa is available in the Nordics and the Netherlands.

The Visma Severa driver covers 264 tables and 5299 columns.

Visma Severa API Clients

Invantive UniversalSQL is available on many user interfaces ("clients" in traditional server-client paradigma). All Invantive UniversalSQL statements can be exchanged with a close to 100% compatibility across all clients and operating systems (Windows, MacOS, Linux, iOS, Android).

The clients include Microsoft Excel, Microsoft Power BI, Microsoft Power Query, Microsoft Word and Microsoft Outlook. Web-based clients include Invantive Cloud, Invantive Bridge Online as OData proxy, Invantive App Online for interactive apps, Online SQL Editor for query execution and Invantive Data Access Point as extended proxy.

The Visma Severa Power BI connector is based on the Invantive UniversalSQL driver for Visma Severa, completed by a high-performance OData connector which works straight on Power BI without any add-on. The OData protocol is always version 4, independent whether the backing platform uses OData, SOAP or another protocol.

For technical users there are command-line editions of Invantive Data Hub running on iOS, Android, Windows, MacOS and Linux. Invantive Data Hub is also often used for enterprise server applications such as ETL. High-volume replication of data taken from the Visma Severa API into traditional databases such as SQL Server (on-premises and Azure), MySQL, PostgreSQL and Oracle is possible using Invantive Data Replicator. Invantive Data Replicator automatically creates and maintains Visma Severa datawarehouses, possibly in combination with data from over 75 other (cloud) platforms. Invantive Data Replicator supports data volumes up to over 1 TB and over 5.000 companies. The on-premise edition of Invantive Bridge offers an Visma Severa ADO.net provider.

Finally, online web apps can be build for Visma Severa using App Online of Invantive Cloud.

Monitor API Calls

When a query or DML-statement has been executed on Invantive UniversalSQL a developer can evaluate the actual calls made to the Visma Severa API using a query on sessionios@DataDictionary. As an alternative, extensive request and response logging can be enabled by setting log-native-calls-to-disk to true. In the %USERPROFILE% \Invantive\NativeLog folder Invantive UniversalSQL will create log files per Visma Severa API request and response.

Specifications

The SQL driver for Visma Severa does not support partitioning. Define one data container in a database for each company in Visma Severa to enable parallel access for data from multiple companies.

An introduction into the concepts of Invantive UniversalSQL such as databases, data containers and partitioning can be found in the Invantive UniversalSQL grammar.

The configuration can be changed using various attributes from the database definition, on log on and during use. A full list of configuration options is listed in the driver attributes 2.

The catalog name is used to compose the full qualified name of an object like a table or view. The schema name is used to compose the full qualified name of an object like a table or view. On Visma Severa the comparison of two texts is case sensitive by default.

Changes and bug fixes on the Visma Severa SQL driver can be found in the release notes. Get access to the community through the Visma Severa section of the Invantive forums.

Driver code for use in settings.xml: `VismaSevera`

Alias: `severa`

Recommended alias: `sva`

Driver code for use in settings.xml

General documentation on Visma Severa is available at http://severa.visma.com.

Updated 10-06-2024 19:00 using Invantive UniversalSQL version 24.1.3-BETA+4689.

2 SQL Driver Attributes for Visma Severa API

The SQL driver for Visma Severa has many attributes that can be finetuned to improve handling in scenarios with unreliable network connections to the API server of Visma Severa or high volumes of data. Also, many drivers have driver-specific attributes to finetune actual behaviour or handle data not matching specifications.

The Visma Severa driver attributes are assigned a default value which seldom requires change. However, changes can be applied when needed on four levels, which are reflected in the table below by separate checkmarks:

- Connection string: the connection string from the settings*.xml file and applied during log on.
- Set SQL statement: a set SQL-statement to be executed once connection has been established.
- Log on: value to be specified interactively by user during log on in a user interface.

The connection string for Visma Severa can be found in the settings*.xml file used for the database. The reference manuals contain instructions how to relocate the settings*.xml files. Settings*.xml files are typically located in the `%USERPROFILE%\invantive` folder in most deployment scenarios. Each data container of a database in the connection string can have a `connectionString` element specifying the name and values of attributes. Both name and value must be properly escaped according to XML-semantics. Actual application of the value is solely done during log on. A new connection must be established to change the value of a driver attribute using a connection string.

The set SQL statement can be executed after log on. The syntax is: `set NAME VALUE`, or for a distributed database: `set NAME@ALIAS VALUE`. In some scenarios you may need to enclose the driver attribute name in square brackets to escape it from parsing, for instance when a reserved SQL keyword is part of the name. The new value takes effect straight after execution of the set-statement. The set-statement can be executed as often as needed during a session.

Driver attributes that can be interactively set to a value are typically presented in the log on window. Depending on the platform and design decisions of the user interface designer, some or all of the available driver attributes can have been made available.

The Visma Severa driver can be configured using the following attributes:

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
analysis-enforce-row-uniqueness	Enforce rows to be unique for software analysis. A fingerprint is calculated from the whole row of data when the primary key column is unknown.	Shared	False	✓	✓	✓	
api-url	URL of web service.	Visma Severa		✓		✓	
bulk-delete-page-size-rows	Number of rows to delete per batch when bulk deleting.	Shared	10000	✓	✓	✓	
bulk-insert-page-size-bytes	Approximate maximum size in bytes of batch when bulk inserting.	Shared	10000000	✓	✓	✓	
bulk-insert-page-size-rows	Number of rows to insert per batch when bulk inserting.	Shared	10000	✓	✓	✓	
download-error-other-exception-max-tries	Maximum number of tries when an unqualified error occurs during retrieval of data.		3	✓	✓	✓	
download-error-other-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an unqualified error ocurrs during retrieval of data.		10000	✓	✓	✓	
download-error-other-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an unqualified error ocurrs during retrieval of data.		300000	✓	✓	✓	
download-error-other-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an unqualified error ocurrs during retrieval of data.		2	✓	✓	✓	
force-case-sensitive-identifiers	Consider identifiers as case-sensitive independent of the platform capabilities.	Shared	False	✓	✓	✓	
forced-casing-identifiers	Forced casing of identifiers. Choose from: Unset, Lower, Upper and Mixed.	Shared		✓	✓	✓	
http-disk-cache-compression-level	Compression level for the HTTP disk cache, ranging from 1 (little) to 9 (intense). Default is 5.	Shared	5	✓	✓	✓	
http-disk-cache-directory	Directory where HTTP cache is stored.	Shared	C:\Users\guido\Invantive\Cache\http\guido\shared	✓	✓	✓	
http-disk-cache-ignore-write-errors	Whether to ignore write errors to disk cache.	Shared	False	✓	✓	✓	
http-disk-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP disk cache.	Shared	2592000	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
http-get-timeout-max-ms	HTTP GET timeout maximum on retry (ms)		24000	✓	✓	✓	
http-get-timeout-ms	HTTP GET timeout (ms)		56000	✓	✓	✓	
http-memory-cache-compression-level	Compression level for the HTTP memory cache, ranging from 1 (little) to 9 (intense). Default is 5.		5	✓	✓	✓	
http-memory-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP memory cache.		14400	✓	✓	✓	
http-post-timeout-max-ms	HTTP POST timeout maximum on retry (ms)		58000	✓	✓	✓	
http-post-timeout-ms	HTTP POST timeout (ms)		57000	✓	✓	✓	
invantive-sql-compress-sparse-arrays	Whether to compress sparse arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-correct-invalid-date	Whether to correct dates considered invalid since they are before 01-01-1753. When nullable, they are removed. Otherwise they are replaced by 01-01-1753.	SQL Engine V1	False	✓	✓	✓	
invantive-sql-execution-profile-disk-path	itgen_pae_invantive_sql_execution_profile_disk_path	SQL Engine V1	c:\temp\profiles	✓	✓	✓	
invantive-sql-execution-profile-to-disk	itgen_pae_invantive_sql_execution_profile_to_disk	SQL Engine V1	True	✓	✓	✓	
invantive-sql-forward-filters-to-data-containers	Whether to forward filters to data containers.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-share-byte-arrays	Whether to share the memory used by identical byte arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-share-strings	Whether to share the memory used by identical strings in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-shuffle-fetch-results-data-containers	Whether to shuffle results fetched from data containers.	SQL Engine V1	False	✓	✓	✓	
invantive-use-cache	Whether to cache the results of a query.	SQL Engine V1	True	✓	✓	✓	
log-native-calls-to-disk-max-events	Maximum number of call events to register from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-max-seconds	Maximum number of seconds to register calls from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-on-error	Registers native calls to data container backend as disk files when the call raised an error.	Shared	False	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
log-native-calls-to-disk-on-success	Registers native calls to data container backend as disk files when the call raised no error.	Shared	False	✓	✓	✓	
log-native-calls-to-trace	Log native calls to data container backend on the trace.	Shared	False	✓	✓	✓	
max-url-length-accepted	The maximum accepted URL length before raising an error.	Shared	8000	✓	✓	✓	
max-url-length-desired	The maximum desired URL length.	Shared	8000	✓	✓	✓	
maximum-length-identifiers	Non-default maximum length in characters of identifier names.	Shared		✓	✓	✓	
minimum-length-text	Extend all text columns to this length to allow processing of XML that uses longer text values than the XSD specifies.	XML		✓			✓
partition-slot-based-rate-limit-length-ms	Total length in milliseconds across all slots of a partition-based rate limit.	Shared	60000	✓		✓	
partition-slot-based-rate-limit-slots	Number of slots per partition-based rate limit. Null means no slot-based rate limit.	Shared		✓		✓	
pre-request-delay-ms	Pre-request delay in milliseconds per request.	Shared	0	✓	✓	✓	
requested-page-size	Preferred number of rows to exchange per round trip; only effective on limited platforms such as AFAS Online.	Shared		✓	✓	✓	
requests-parallel-max	Maximum number of parallel data requests from individual partitions on the data container.	Shared	32	✓	✓	✓	
result-set-memory-cache	Action: set to 'empty' to empty.	XML			✓		
slot-based-rate-limit-length-ms	Total length in milliseconds across all slots of a slot-based rate limit.	Shared	60000	✓		✓	
slot-based-rate-limit-slots	Number of slots of a slot-based rate limit. Null means no slot-based rate limit.	Shared		✓		✓	
standardize-identifiers-casing	Rewrite all identifiers to the recommended standard platform-specific casing when changing a data model on a case-dependent platform.	Shared	True	✓	✓	✓	
standardize-identifiers	Rewrite all identifiers to the preferred standards as configured by standardize-identifiers-casing and maximum-length-identifiers.	Shared	True	✓	✓	✓	
use-http-disk-cache-read	Whether to use HTTP responses from previous queries stored on disk to answer the current query.	Shared	False	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
use-http-disk-cache-write	Whether to memorize HTTP responses on disk.	Shared	False	✓	✓	✓	
use-http-memory-cache-read	Whether to use HTTP responses from previous queries stored in memory that can answer the current query.		True	✓	✓	✓	
use-http-memory-cache-write	Whether to memorize HTTP responses from previous queries for use by future queries.		True	✓	✓	✓	
use-metadata-memory-cache	Whether to use the metadata in memory calculated previously.	XML	True	✓	✓	✓	
use-result-memory-cache	Whether to use result sets cached in memory from previous queries that can answer the current query.	XML	False	✓	✓	✓	

3 Schema: API

3.1 Tables

3.1.1 AccessRightsProfileByGUID: Visma Severa Access Rights Profile by GUID

Catalog: Severa

Schema: API

Label: Access Rights Profile by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: guid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccessRightsProfileByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
guid	string	☐		

Columns of Table Function

The columns of the table function AccessRightsProfileByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		
RIGHTS_ACCOUNT	string			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	string			
RIGHTS_CASE	string			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOURAPPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	string			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	string			
RIGHTS_USERS	string			
RIGHTS_WORKHOURAPPROVAL	string			

3.1.2 AccessRightsProfiles: Visma Severa Access Rights Profiles

Catalog: Severa

Schema: API

Label: Access Rights Profiles

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table AccessRightsProfiles are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		

Name	Data Type	Label	Required	Documentation
NAME	string	Name		
RIGHTS_ACCOUNT	string			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	string			
RIGHTS_CASE	string			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOURAPPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	string			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	string			
RIGHTS_USERS	string			
RIGHTS_WORKHOURAPPROVAL	string			

3.1.3 AccountByGUID: Visma Severa Account by GUID

Catalog: Severa

Schema: API

Label: Account by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accountGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountGuid	string	☐		

Columns of Table Function

The columns of the table function AccountByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.4 AccountByName: Visma Severa Account by Name

Catalog: Severa

Schema: API

Label: Account by Name

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accountName

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountByName. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountName	string	☐		

Columns of Table Function

The columns of the table function AccountByName are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		

Name	Data Type	Label	Required	Documentation
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.5 AccountByNumber: Visma Severa Account by Number

Catalog: Severa

Schema: API

Label: Account by Number

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accountNumber

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountByNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountNumber	int64	☐		

Columns of Table Function

The columns of the table function AccountByNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		

Name	Data Type	Label	Required	Documentation
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.6 AccountByVatNumber: Visma Severa Account by VAT Number

Catalog: Severa

Schema: API

Label: Account by VAT Number

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: vatNumber

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountByVatNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
vatNumber	string	☐		

Columns of Table Function

The columns of the table function AccountByVatNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.7 AccountGroups: Visma Severa Account Groups

Catalog: Severa

Schema: API

Label: Account Groups

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountGroups. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
options	int32	□		

Columns of Table Function

The columns of the table function AccountGroups are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		
PARENTGUID	string	Parent GUID		

3.1.8 AccountGroupsByAccountGUID: Visma Severa Account Group by Account GUID

Catalog: Severa

Schema: API

Label: Account Group by Account GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accountGuid- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountGroupsByAccountGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountGuid	string	☐		
options	int32	☐		

Columns of Table Function

The columns of the table function AccountGroupsByAccountGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		
PARENTGUID	string	Parent GUID		

3.1.9 Accounts: Visma Severa Accounts

Catalog: Severa

Schema: API

Label: Accounts

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: options- startFromIndex- count

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Accounts. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
count	int32	☐		
options	string	☐		
startFromIndex	int32	☐		

Columns of Table Function

The columns of the table function Accounts are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
NAME	string	Name		
NUMBER	int64	Number		
VATNUMBER	string	VAT Number		

3.1.10 AccountsByCompanyGUID: Visma Severa Accounts by Company GUID

Catalog: Severa

Schema: API

Label: Accounts by Company GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: companyGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountsByCompanyGUID. A value must be provided at all times for required parameters,

but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
companyGuid	string	☐		

Columns of Table Function

The columns of the table function AccountsByCompanyGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		

Name	Data Type	Label	Required	Documentation
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.11 AccountsByInsertDate: Visma Severa Accounts by Insert Date

Catalog: Severa

Schema: API

Label: Accounts by Insert Date

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: insertStartDate- insertEndDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountsByInsertDate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
insertEndDate	datetime	☐		
insertStartDate	datetime	☐		

Columns of Table Function

The columns of the table function AccountsByInsertDate are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		

Name	Data Type	Label	Required	Documentation
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.12 AccountsChangedSince: Visma Severa Accounts Changed since

Catalog: Severa

Schema: API

Label: Accounts Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountsChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
options	int32	☐		
startDateTime	datetime	☐		

Columns of Table Function

The columns of the table function AccountsChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.13 AccountsChangedSinceOpts: Visma Severa Accounts Changed since Opts.

Catalog: Severa

Schema: API

Label: Accounts Changed since Opts.

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountsChangedSinceOpts. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
options	string	☐		
startDateTime	datetime	☐		

Columns of Table Function

The columns of the table function AccountsChangedSinceOpts are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.14 ActiveLeadSources: Visma Severa Active Lead Sources

Catalog: Severa

Schema: API

Label: Active Lead Sources

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table ActiveLeadSources are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.15 ActivitiesByDate: Visma Severa Activities by Date

Catalog: Severa

Schema: API

Label: Activities by Date

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: startDate- endDate- activityCategory- businessUnitGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivitiesByDate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityCategory	string	☐		
businessUnitGuid	string	☐		
endDate	datetime	☐		
startDate	datetime	☐		

Columns of Table Function

The columns of the table function ActivitiesByDate are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACTIVITYTYPECODE	string	Activity Type Code		
ACTIVITYTYPEGUID	string	Activity Type GUID		
CASEGUID	string	Project GUID		
ENDDATETIME	datetime	End Date Time		
GUID	string	GUID		
ISALLDAY	boolean	Is All Day		
ISCLOSED	boolean	Is Closed		
ISDURATION	boolean	Is Duration		
ISPRIVATE	boolean	Is Private		
ISREADONLY	boolean	Is Read-only		
ISUNASSIGNED	boolean	Is Unassigned		
LOCATION	string	Location		

Name	Data Type	Label	Required	Documentation
NAME	string	Name		
NOTES	string	Notes		
OWNERUSERBUSINESSUNITCODE	string	Owner User Business Unit Code		
OWNERUSERCODE	string	Owner User Code		
OWNERUSERFIRSTNAME	string	Owner User First Name		
OWNERUSERGUID	string	Owner User GUID		
OWNERUSERLASTNAME	string	Owner User Last Name		
STARTDATETIME	datetime	Start Date Time		
TASKGUID	string	Task GUID		

3.1.16 ActivityByGUID: Visma Severa Activity by GUID

Catalog: Severa

Schema: API

Label: Activity by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivityByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityGUID	string	☐		

Columns of Table Function

The columns of the table function ActivityByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACTIVITYTYPECODE	string	Activity Type Code		
ACTIVITYTYPEGUID	string	Activity Type GUID		
CASEGUID	string	Project GUID		
ENDDATETIME	datetime	End Date Time		
GUID	string	GUID		
ISALLDAY	boolean	Is All Day		
ISCLOSED	boolean	Is Closed		
ISDURATION	boolean	Is Duration		
ISPRIVATE	boolean	Is Private		
ISREADONLY	boolean	Is Read-only		
ISUNASSIGNED	boolean	Is Unassigned		
LOCATION	string	Location		
NAME	string	Name		
NOTES	string	Notes		
OWNERUSERBUSINESSUNITCODE	string	Owner User Business Unit Code		
OWNERUSERCODE	string	Owner User Code		
OWNERUSERFIRSTNAME	string	Owner User First Name		
OWNERUSERGUID	string	Owner User GUID		
OWNERUSERLASTNAME	string	Owner User Last Name		
STARTDATETIME	datetime	Start Date Time		
TASKGUID	string	Task GUID		

3.1.17 ActivityContactParticipants: Visma Severa Activity Contact Participants

Catalog: Severa

Schema: API

Label: Activity Contact Participants

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivityContactParticipants. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default

to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityGUID	string	☐		

Columns of Table Function

The columns of the table function ActivityContactParticipants are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTACTGUID	string	Contact GUID		
NAME	string	Name		
RESOURCEGUID	string	Resource GUID		
STATUS	int16	Status		
USERGUID	string	User GUID		

3.1.18 ActivityInstances: Visma Severa Activity Instances

Catalog: Severa

Schema: API

Label: Activity Instances

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: userGUID- startsAfter- startsBefore- endsAfter- endsBefore- activityTypeGUID- accountGUID- caseGUID- firstRow- maxRows

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivityInstances. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountGUID	string	☐		
activityTypeGUID	string	☐		
caseGUID	string	☐		
endsAfter	datetime	☐		
endsBefore	datetime	☐		
firstRow	int32	☐		
maxRows	int32	☐		
startsAfter	datetime	☐		
startsBefore	datetime	☐		
userGUID	string	☐		

Columns of Table Function

The columns of the table function ActivityInstances are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACTIVITYTYPECODE	string	Activity Type Code		
ACTIVITYTYPEGUID	string	Activity Type GUID		
CASEGUID	string	Project GUID		
ENDDATETIME	datetime	End Date Time		
GUID	string	GUID		
ISALLDAY	boolean	Is All Day		
ISCLOSED	boolean	Is Closed		
ISDURATION	boolean	Is Duration		
ISPRIVATE	boolean	Is Private		
ISREADONLY	boolean	Is Read-only		
ISUNASSIGNED	boolean	Is Unassigned		
LOCATION	string	Location		
NAME	string	Name		
NOTES	string	Notes		
OWNERUSERBUSINESSUNITCODE	string	Owner User Business Unit Code		
OWNERUSERCODE	string	Owner User Code		

Name	Data Type	Label	Required	Documentation
OWNERUSERFIRSTNAME	string	Owner User First Name		
OWNERUSERGUID	string	Owner User GUID		
OWNERUSERLASTNAME	string	Owner User Last Name		
STARTDATETIME	datetime	Start Date Time		
TASKGUID	string	Task GUID		

3.1.19 ActivityParticipants: Visma Severa Activity Participants

Catalog: Severa

Schema: API

Label: Activity Participants

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivityParticipants. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityGUID	string	☐		

Columns of Table Function

The columns of the table function ActivityParticipants are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTACTGUID	string	Contact GUID		
NAME	string	Name		

Name	Data Type	Label	Required	Documentation
RESOURCEGUID	string	Resource GUID		
STATUS	int16	Status		
USERGUID	string	User GUID		

3.1.20 ActivityResourceParticipants: Visma Severa Activity Resource Participants

Catalog: Severa

Schema: API

Label: Activity Resource Participants

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivityResourceParticipants. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityGUID	string	☐		

Columns of Table Function

The columns of the table function ActivityResourceParticipants are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTACTGUID	string	Contact GUID		
NAME	string	Name		
RESOURCEGUID	string	Resource GUID		
STATUS	int16	Status		
USERGUID	string	User GUID		

3.1.21 ActivityTypeByGUID: Visma Severa Activity Type by GUID

Catalog: Severa

Schema: API

Label: Activity Type by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityTypeGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivityTypeByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityTypeGUID	string	☐		

Columns of Table Function

The columns of the table function ActivityTypeByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CATEGORY	string	Category		
CODE	string	Code		
GUID	string	GUID		
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPAIDLEAVE	boolean	Is Paid Leave		
NAME	string	Name		

3.1.22 ActivityTypes: Visma Severa Activity Types

Catalog: Severa

Schema: API

Label: Activity Types

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table ActivityTypes are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CATEGORY	string	Category		
CODE	string	Code		
GUID	string	GUID		
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPAIDLEAVE	boolean	Is Paid Leave		
NAME	string	Name		

3.1.23 ActivityUserParticipants: Visma Severa Activity User Participants

Catalog: Severa

Schema: API

Label: Activity User Participants

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivityUserParticipants. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityGUID	string	☐		

Columns of Table Function

The columns of the table function ActivityUserParticipants are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTACTGUID	string	Contact GUID		
NAME	string	Name		
RESOURCEGUID	string	Resource GUID		
STATUS	int16	Status		
USERGUID	string	User GUID		

3.1.24 AddCaseMemberUser: Visma Severa Add Project Member User

Catalog: Severa

Schema: API

Label: Add Project Member User

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGuid- userGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddCaseMemberUser. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGuid	string	☐		

Name	Data Type	Required	Default Value	Documentation
userGuid	string	☐		

Columns of Table Function

The columns of the table function AddCaseMemberUser are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DUMMY	string	Dummy	☑	Dummy column

3.1.25 AddNewAccessRightsProfile: Visma Severa Add New Access Rights Profile

Catalog: Severa

Schema: API

Label: Add New Access Rights Profile

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accessRightsProfile_GUID- accessRightsProfile_IsActive- accessRightsProfile_IsDefault- accessRightsProfile_Name- accessRightsProfile_Rights_Account- accessRightsProfile_Rights_AccountDelete- accessRightsProfile_Rights_Administrator- accessRightsProfile_Rights_Case- accessRightsProfile_Rights_CaseDelete- accessRightsProfile_Rights_CaseOwnerWorkHourApproval- accessRightsProfile_Rights_ScheduleJobs- accessRightsProfile_Rights_Sharing- accessRightsProfile_Rights_TravelReimbursement- accessRightsProfile_Rights_Users- accessRightsProfile_Rights_WorkHourApproval

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewAccessRightsProfile. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accessRightsProfile_GUID	string	☐		

Name	Data Type	Required	Default Value	Documentation
accessRightsProfile_IsActive	boolean	☐		
accessRightsProfile_IsDefault	boolean	☐		
accessRightsProfile_Name	string	☐		
accessRightsProfile_Rights_Account	string	☐		
accessRightsProfile_Rights_AccountDelete	boolean	☐		
accessRightsProfile_Rights_Administrator	string	☐		
accessRightsProfile_Rights_Case	string	☐		
accessRightsProfile_Rights_CaseDelete	boolean	☐		
accessRightsProfile_Rights_CaseOwnerWorkHourApproval	boolean	☐		
accessRightsProfile_Rights_ScheduleJobs	string	☐		
accessRightsProfile_Rights_Sharing	boolean	☐		
accessRightsProfile_Rights_TravelReimbursement	string	☐		
accessRightsProfile_Rights_Users	string	☐		
accessRightsProfile_Rights_WorkHourApproval	string	☐		

Columns of Table Function

The columns of the table function AddNewAccessRightsProfile are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		
RIGHTS_ACCOUNT	string			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	string			
RIGHTS_CASE	string			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOURAPPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	string			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	string			
RIGHTS_USERS	string			

Name	Data Type	Label	Required	Documentation
RIGHTS_WORKHOURAPPROVAL	string			

3.1.26 AddNewAccount: Visma Severa Add New Account

Catalog: Severa

Schema: API

Label: Add New Account

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accountInfo_AccountGroupGUIDs- accountInfo_AccountOwnerUserGUID- accountInfo_AccountRating- accountInfo_CompanyGUID- accountInfo_CurrencyCode- accountInfo_CurrencyGUID- accountInfo_GUID- accountInfo_InsertTS- accountInfo_InvoicingVat- accountInfo_IsActive- accountInfo_IsInternal- accountInfo_LanguageCode- accountInfo_LanguageGUID- accountInfo_Name- accountInfo_Notes- accountInfo_Number- accountInfo_OverdueInterest- accountInfo_PaymentTerm- accountInfo_PricelistGUID- accountInfo_ReverseCharge- accountInfo_ReverseChargeDescription- accountInfo_UpdatedTS- accountInfo_UrlToAccount- accountInfo_eInvoiceAddress- accountInfo_eInvoiceOperator- companyInfo_AnnualRevenue- companyInfo_AnnualRevenue2- companyInfo_Email- companyInfo_Employees- companyInfo_GUID- companyInfo_HeadOfficeAddressGUID- companyInfo_HierarchyGUID- companyInfo_IndustryGUID- companyInfo_LanguageGUID- companyInfo_Name- companyInfo_TimezoneGUID- companyInfo_VatNumber- companyInfo_Website- addressInfo_Addressline- addressInfo_Addressline2- addressInfo_Addressline3- addressInfo_City- addressInfo_CompanyGUID- addressInfo_CountryCode- addressInfo_CountryGUID- addressInfo_CountryName- addressInfo_CountryRegionGUID- addressInfo_CountryRegionName- addressInfo_Fax- addressInfo_GUID- addressInfo_IsBillingAddress- addressInfo_IsPostalAddress- addressInfo_IsVisitAddress- addressInfo_Phone- addressInfo_PostalCode- contactInfo_AccountGUID- contactInfo_AddressGUID- contactInfo_CommunicationMethods- contactInfo_ContactRoleGUID- contactInfo_DateOfBirth- contactInfo_Description- contactInfo_FirstName- contactInfo_GUID- contactInfo_IsActive- contactInfo_JobTitle- contactInfo_Keywords- contactInfo_LanguageGUID- contactInfo_LastName- contactInfo_Salutation- contactInfo_SatisfactionLevel- contactInfo_TimeZoneGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewAccount. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountInfo_AccountGroupGUIDs	string	☐		
accountInfo_AccountOwnerUserGUID	string	☐		
accountInfo_AccountRating	int32	☐		
accountInfo_CompanyGUID	string	☐		
accountInfo_CurrencyCode	string	☐		
accountInfo_CurrencyGUID	string	☐		
accountInfo_eInvoiceAddress	string	☐		
accountInfo_eInvoiceOperator	string	☐		
accountInfo_GUID	string	☐		
accountInfo_InsertTS	datetime	☐		
accountInfo_InvoicingVat	decimal	☐		
accountInfo_IsActive	boolean	☐		
accountInfo_IsInternal	boolean	☐		
accountInfo_LanguageCode	string	☐		
accountInfo_LanguageGUID	string	☐		
accountInfo_Name	string	☐		
accountInfo_Notes	string	☐		
accountInfo_Number	int64	☐		
accountInfo_OverdueInterest	decimal	☐		
accountInfo_PaymentTerm	int32	☐		
accountInfo_PricelistGUID	string	☐		
accountInfo_ReverseCharge	boolean	☐		
accountInfo_ReverseChargeDescription	string	☐		
accountInfo_UpdatedTS	datetime	☐		
accountInfo_UrlToAccount	string	☐		
addressInfo_Addressline	string	☐		
addressInfo_Addressline2	string	☐		
addressInfo_Addressline3	string	☐		
addressInfo_City	string	☐		
addressInfo_CompanyGUID	string	☐		
addressInfo_CountryCode	string	☐		
addressInfo_CountryGUID	string	☐		
addressInfo_CountryName	string	☐		
addressInfo_CountryRegionGUID	string	☐		
addressInfo_CountryRegionName	string	☐		
addressInfo_Fax	string	☐		
addressInfo_GUID	string	☐		

Name	Data Type	Required	Default Value	Documentation
addressInfo_IsBillingAddress	boolean	☐		
addressInfo_IsPostalAddress	boolean	☐		
addressInfo_IsVisitAddress	boolean	☐		
addressInfo_Phone	string	☐		
addressInfo_PostalCode	string	☐		
companyInfo_AnnualRevenue	int32	☐		
companyInfo_AnnualRevenue2	int64	☐		
companyInfo_Email	string	☐		
companyInfo_Employees	int32	☐		
companyInfo_GUID	string	☐		
companyInfo_HeadOfficeAddressGUID	string	☐		
companyInfo_HierarchyGUID	string	☐		
companyInfo_IndustryGUID	string	☐		
companyInfo_LanguageGUID	string	☐		
companyInfo_Name	string	☐		
companyInfo_TimezoneGUID	string	☐		
companyInfo_VatNumber	string	☐		
companyInfo_Website	string	☐		
contactInfo_AccountGUID	string	☐		
contactInfo_AddressGUID	string	☐		
contactInfo_CommunicationMethods	string	☐		
contactInfo_ContactRoleGUID	string	☐		
contactInfo_DateOfBirth	datetime	☐		
contactInfo_Description	string	☐		
contactInfo_FirstName	string	☐		
contactInfo_GUID	string	☐		
contactInfo_IsActive	boolean	☐		
contactInfo_JobTitle	string	☐		
contactInfo_Keywords	string	☐		
contactInfo_LanguageGUID	string	☐		
contactInfo_LastName	string	☐		
contactInfo_Salutation	string	☐		
contactInfo_SatisfactionLevel	int32	☐		
contactInfo_TimeZoneGUID	string	☐		

Columns of Table Function

The columns of the table function AddNewAccount are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.27 AddNewAccountGroup: Visma Severa Add New Account Group

Catalog: Severa

Schema: API

Label: Add New Account Group

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accountGroupInfo_GUID- accountGroupInfo_IsActive- accountGroupInfo_Name- accountGroupInfo_ParentGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewAccountGroup. A value must be provided at all times for required parameters, but

optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountGroupInfo_GUID	string	☐		
accountGroupInfo_IsActive	boolean	☐		
accountGroupInfo_Name	string	☐		
accountGroupInfo_ParentGUID	string	☐		

Columns of Table Function

The columns of the table function AddNewAccountGroup are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		
PARENTGUID	string	Parent GUID		

3.1.28 AddNewAccountGroupMembers: Visma Severa Add New Account Group Members

Catalog: Severa

Schema: API

Label: Add New Account Group Members

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accountGUID- accountGroupGUIDs

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewAccountGroupMembers. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountGroupGUIDs	string	☐		
accountGUID	string	☐		

Columns of Table Function

The columns of the table function AddNewAccountGroupMembers are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DUMMY	string	Dummy	☑	Dummy column

3.1.29 AddNewActivity: Visma Severa Add New Activity

Catalog: Severa

Schema: API

Label: Add New Activity

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityInfo_AccountGUID- activityInfo_ActivityTypeCode- activityInfo_ActivityTypeGUID- activityInfo_CaseGUID- activityInfo_EndDateTime- activityInfo_GUID- activityInfo_IsAllDay- activityInfo_IsClosed- activityInfo_IsDuration- activityInfo_IsPrivate- activityInfo_IsReadOnly- activityInfo_IsUnassigned- activityInfo_Location- activityInfo_Name- activityInfo_Notes- activityInfo_OwnerUserBusinessUnitCode- activityInfo_OwnerUserCode- activityInfo_OwnerUserFirstName- activityInfo_OwnerUserGUID- activityInfo_OwnerUserLastName- activityInfo_StartDateTime- activityInfo_TaskGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewActivity. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityInfo_AccountGUID	string	☐		
activityInfo_ActivityTypeCode	string	☐		
activityInfo_ActivityTypeGUID	string	☐		
activityInfo_CaseGUID	string	☐		
activityInfo_EndDateTime	datetime	☐		
activityInfo_GUID	string	☐		
activityInfo_IsAllDay	boolean	☐		
activityInfo_IsClosed	boolean	☐		
activityInfo_IsDuration	boolean	☐		
activityInfo_IsPrivate	boolean	☐		
activityInfo_IsReadOnly	boolean	☐		
activityInfo_IsUnassigned	boolean	☐		
activityInfo_Location	string	☐		
activityInfo_Name	string	☐		
activityInfo_Notes	string	☐		
activityInfo_OwnerUserBusinessUnitCode	string	☐		
activityInfo_OwnerUserCode	string	☐		
activityInfo_OwnerUserFirstName	string	☐		
activityInfo_OwnerUserGUID	string	☐		
activityInfo_OwnerUserLastName	string	☐		
activityInfo_StartDateTime	datetime	☐		
activityInfo_TaskGUID	string	☐		

Columns of Table Function

The columns of the table function AddNewActivity are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACTIVITYTYPECODE	string	Activity Type Code		
ACTIVITYTYPEGUID	string	Activity Type GUID		
CASEGUID	string	Project GUID		
ENDDATETIME	datetime	End Date Time		
GUID	string	GUID		

Name	Data Type	Label	Required	Documentation
ISALLDAY	boolean	Is All Day		
ISCLOSED	boolean	Is Closed		
ISDURATION	boolean	Is Duration		
ISPRIVATE	boolean	Is Private		
ISREADONLY	boolean	Is Read-only		
ISUNASSIGNED	boolean	Is Unassigned		
LOCATION	string	Location		
NAME	string	Name		
NOTES	string	Notes		
OWNERUSERBUSINESSUNITCODE	string	Owner User Business Unit Code		
OWNERUSERCODE	string	Owner User Code		
OWNERUSERFIRSTNAME	string	Owner User First Name		
OWNERUSERGUID	string	Owner User GUID		
OWNERUSERLASTNAME	string	Owner User Last Name		
STARTDATETIME	datetime	Start Date Time		
TASKGUID	string	Task GUID		

3.1.30 AddNewActivityContactParticipant: Visma Severa Add New Activity Contact Participant

Catalog: Severa

Schema: API

Label: Add New Activity Contact Participant

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityGUID- contactGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewActivityContactParticipant. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityGUID	string	☐		
contactGUID	string	☐		

Columns of Table Function

The columns of the table function AddNewActivityContactParticipant are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDNEWACTIVITYCONTACTPARTICIPANTRESULT	boolean			

3.1.31 AddNewActivityResourceParticipant: Visma Severa Add New Activity Resource Participant

Catalog: Severa

Schema: API

Label: Add New Activity Resource Participant

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityGUID- resourceGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewActivityResourceParticipant. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityGUID	string	☐		
resourceGUID	string	☐		

Columns of Table Function

The columns of the table function AddNewActivityResourceParticipant are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDNEWACTIVITYRESOURCEPARTICIPANTRESULT	boolean			

3.1.32 AddNewActivityType: Visma Severa Add New Activity Type

Catalog: Severa

Schema: API

Label: Add New Activity Type

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityTypeInfo_Category- activityTypeInfo_Code- activityTypeInfo_GUID- activityTypeInfo_Icon- activityTypeInfo_IsActive- activityTypeInfo_IsDefault- activityTypeInfo_IsPaidLeave- activityTypeInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewActivityType. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityTypeInfo_Category	string	☐		
activityTypeInfo_Code	string	☐		
activityTypeInfo_GUID	string	☐		
activityTypeInfo_Icon	string	☐		
activityTypeInfo_IsActive	boolean	☐		
activityTypeInfo_IsDefault	boolean	☐		
activityTypeInfo_IsPaidLeave	boolean	☐		
activityTypeInfo_Name	string	☐		

Columns of Table Function

The columns of the table function AddNewActivityType are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CATEGORY	string	Category		
CODE	string	Code		
GUID	string	GUID		
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPAIDLEAVE	boolean	Is Paid Leave		
NAME	string	Name		

3.1.33 AddNewActivityUserParticipant: Visma Severa Add New Activity User Participant

Catalog: Severa

Schema: API

Label: Add New Activity User Participant

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityGUID- userGUID- memberStatus

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewActivityUserParticipant. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityGUID	string	☐		
memberStatus	int16	☐		
userGUID	string	☐		

Columns of Table Function

The columns of the table function AddNewActivityUserParticipant are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDNEWACTIVITYUSERPARTICIPANTRESULT	boolean			

3.1.34 AddNewAddress: Visma Severa Add New Address

Catalog: Severa

Schema: API

Label: Add New Address

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: addressInfo_Addressline- addressInfo_Addressline2- addressInfo_Addressline3- addressInfo_City- addressInfo_CompanyGUID- addressInfo_CountryCode- addressInfo_CountryGUID- addressInfo_CountryName- addressInfo_CountryRegionGUID- addressInfo_CountryRegionName- addressInfo_Fax- addressInfo_GUID- addressInfo_IsBillingAddress- addressInfo_IsPostalAddress- addressInfo_IsVisitAddress- addressInfo_Phone- addressInfo_PostalCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewAddress. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
addressInfo_Addressline	string	☐		
addressInfo_Addressline2	string	☐		
addressInfo_Addressline3	string	☐		
addressInfo_City	string	☐		
addressInfo_CompanyGUID	string	☐		
addressInfo_CountryCode	string	☐		
addressInfo_CountryGUID	string	☐		

Name	Data Type	Required	Default Value	Documentation
addressInfo_CountryName	string	☐		
addressInfo_CountryRegionGUID	string	☐		
addressInfo_CountryRegionName	string	☐		
addressInfo_Fax	string	☐		
addressInfo_GUID	string	☐		
addressInfo_IsBillingAddress	boolean	☐		
addressInfo_IsPostalAddress	boolean	☐		
addressInfo_IsVisitAddress	boolean	☐		
addressInfo_Phone	string	☐		
addressInfo_PostalCode	string	☐		

Columns of Table Function

The columns of the table function AddNewAddress are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
ADDRESSLINE2	string	Addressline 2		
ADDRESSLINE3	string	Addressline 3		
CITY	string	City		
COMPANYGUID	string	Company GUID		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYNAME	string	Country Name		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
FAX	string	Fax		
GUID	string	GUID		
ISBILLINGADDRESS	boolean	Is Billing Address		
ISPOSTALADDRESS	boolean	Is Postal Address		
ISVISITADDRESS	boolean	Is Visit Address		
PHONE	string	Phone		
POSTALCODE	string	Postal Code		

3.1.35 AddNewBillingAndExpenseForecastToCase: Visma Severa Add New Billing and Expense Forecast to Project

Catalog: Severa

Schema: API

Label: Add New Billing and Expense Forecast to Project

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGuid- billingAndExpenseForecastInfo_BillingForecast- billingAndExpenseForecastInfo_ExpenseForecast- billingAndExpenseForecastInfo_GUID- billingAndExpenseForecastInfo_InhouseRevenueForecast- billingAndExpenseForecastInfo_Month- billingAndExpenseForecastInfo_Notes- billingAndExpenseForecastInfo_OutsourcingRevenueForecast- billingAndExpenseForecastInfo_Year

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewBillingAndExpenseForecastToCase. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
billingAndExpenseForecastInfo_BillingForecast	decimal	☐		
billingAndExpenseForecastInfo_ExpenseForecast	decimal	☐		
billingAndExpenseForecastInfo_GUID	string	☐		
billingAndExpenseForecastInfo_InhouseRevenueForecast	decimal	☐		
billingAndExpenseForecastInfo_Month	int32	☐		
billingAndExpenseForecastInfo_Notes	string	☐		
billingAndExpenseForecastInfo_OutsourcingRevenueForecast	decimal	☐		
billingAndExpenseForecastInfo_Year	int32	☐		
caseGuid	string	☐		

Columns of Table Function

The columns of the table function AddNewBillingAndExpenseForecastToCase are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
BILLINGFORECAST	decimal	Billing Forecast		
EXPENSEFORECAST	decimal	Expense Forecast		
GUID	string	GUID		
INHOUSEREVENUEFORECAST	decimal	Inhouse Revenue Forecast		
MONTH	int32	Month		
NOTES	string	Notes		
OUTSOURCINGREVENUEFORECAST	decimal	Outsourcing Revenue Forecast		
YEAR	int32	Year		

3.1.36 AddNewBusinessUnit: Visma Severa Add New Business Unit

Catalog: Severa

Schema: API

Label: Add New Business Unit

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitInfo_Code- businessUnitInfo_CostCenterGUID- businessUnitInfo_CurrencyCode- businessUnitInfo_CurrencyGUID- businessUnitInfo_FormattingCulture- businessUnitInfo_GUID- businessUnitInfo_InsertTS- businessUnitInfo_IsActive- businessUnitInfo_LanguageCode- businessUnitInfo_LanguageGUID- businessUnitInfo_Name- businessUnitInfo_NextInvoiceNumber- businessUnitInfo_OverDueInterest- businessUnitInfo_ParentBusinessUnitGUID- businessUnitInfo_PaymentTerm- businessUnitInfo_ShowLogoOnInvoices- businessUnitInfo_ShowLogoOnPdfReports- businessUnitInfo_TimeZoneGUID- businessUnitInfo_UpdatedTS- businessUnitInfo_VatNumber

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewBusinessUnit. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

businessUnitInfo_Code	string	☐		
businessUnitInfo_CostCenterGUID	string	☐		
businessUnitInfo_CurrencyCode	string	☐		
businessUnitInfo_CurrencyGUID	string	☐		
businessUnitInfo_FormattingCulture	string	☐		
businessUnitInfo_GUID	string	☐		
businessUnitInfo_InsertTS	datetime	☐		
businessUnitInfo_IsActive	boolean	☐		
businessUnitInfo_LanguageCode	string	☐		
businessUnitInfo_LanguageGUID	string	☐		
businessUnitInfo_Name	string	☐		
businessUnitInfo_NextInvoiceNumber	int32	☐		
businessUnitInfo_OverDueInterest	decimal	☐		
businessUnitInfo_ParentBusinessUnitGUID	string	☐		
businessUnitInfo_PaymentTerm	int32	☐		
businessUnitInfo_Show LogoOnInvoices	boolean	☐		
businessUnitInfo_Show LogoOnPdfReports	boolean	☐		
businessUnitInfo_TimeZoneGUID	string	☐		
businessUnitInfo_UpdatedTS	datetime	☐		
businessUnitInfo_VatNumber	string	☐		

Columns of Table Function

The columns of the table function AddNewBusinessUnit are shown below. Each column has an SQL data type.

CODE	string	Code		
COSTCENTERGUID	string	Cost Center GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
FORMATTINGCULTURE	string	Formatting Culture		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NEXTINVOICENUMBER	int32	Next Invoice Number		

Name	Data Type	Label	Required	Documentation
OVERDUEINTEREST	decimal	Overdue Interest		
PARENTBUSINESSUNITGUID	string	Parent Business Unit GUID		
PAYMENTTERM	int32	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	string	VAT Number		

3.1.37 AddNewCase: Visma Severa Add New Project

Catalog: Severa

Schema: API

Label: Add New Project

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseInfo_AccountGUID- caseInfo_AccountName- caseInfo_AccountNumber- caseInfo_BillingAddress- caseInfo_BillingAddressGUID- caseInfo_BillingInternalNotes- caseInfo_BillingNotesBeforeGrid- caseInfo_BusinessUnitGUID- caseInfo_BusinessUnitName- caseInfo_BusinessUnitNumber- caseInfo_CaseCompanyCurrencyGUID- caseInfo_CaseNumber- caseInfo_CaseOwnerUserCode- caseInfo_CaseOwnerUserGUID- caseInfo_ClosedDate- caseInfo_CompletionEstimate- caseInfo_ContactGUID- caseInfo_CostCenterGUID- caseInfo_CostCenterNumber- caseInfo_CurrencyGUID- caseInfo_CurrencyShortform- caseInfo_CurrencyShortformOfCaseCompanyCurrency- caseInfo_CurrentCaseStatusDescription- caseInfo_CurrentCaseStatusGUID- caseInfo_CurrentCaseStatusInsertTs- caseInfo_DeadlineDate- caseInfo_Description- caseInfo_ExpectedValue- caseInfo_GUID- caseInfo_IncludeExceptions- caseInfo_IncludeOverTime- caseInfo_InsertTs- caseInfo_InternalName- caseInfo_InvoiceTemplateGUID- caseInfo_IsClosed- caseInfo_IsDailyAllowanceBillable- caseInfo_IsInternal- caseInfo_IsMileageBillable- caseInfo_IsOtherTravelExpensesBillable- caseInfo_LatestEstimationDate- caseInfo_LeadRating- caseInfo_LeadSourceGUID- caseInfo_Name- caseInfo_OrderNumber- caseInfo_OurReference- caseInfo_OverdueInterest- caseInfo_PaymentTerm- caseInfo_PricelistGUID- caseInfo_PricingRule- caseInfo_PricingType- caseInfo_Priority- caseInfo_Probability- caseInfo_RootTaskGUID- caseInfo_SalesCloseDate- caseInfo_SalesPersonUserCode- caseInfo_SalesPersonUserGUID- caseInfo_SalesProcessGUID- caseInfo_SalesStatusGUID- caseInfo_SharedToExtranet- caseInfo_StartDate- caseInfo_UpdatedTs- caseInfo_UrlToCase- caseInfo_UseDefaultProductsInWorkTimeEntry- caseInfo_UseDefaultWorkTypes- caseInfo_YourReference

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewCase. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseInfo_AccountGUID	string	☐		
caseInfo_AccountName	string	☐		
caseInfo_AccountNumber	int64	☐		
caseInfo_BillingAddress	string	☐		
caseInfo_BillingAddressGUID	string	☐		
caseInfo_BillingInternalNotes	string	☐		
caseInfo_BillingNotesBeforeGrid	string	☐		
caseInfo_BusinessUnitGUID	string	☐		
caseInfo_BusinessUnitName	string	☐		
caseInfo_BusinessUnitNumber	string	☐		
caseInfo_CaseCompanyCurrency GUID	string	☐		
caseInfo_CaseNumber	int64	☐		
caseInfo_CaseOw nerUserCode	string	☐		
caseInfo_CaseOw nerUserGUID	string	☐		
caseInfo_ClosedDate	datetime	☐		
caseInfo_CompletionEstimate	int32	☐		
caseInfo_ContactGUID	string	☐		
caseInfo_CostCenterGUID	string	☐		
caseInfo_CostCenterNumber	string	☐		
caseInfo_CurrencyGUID	string	☐		
caseInfo_CurrencyShortform	string	☐		
caseInfo_CurrencyShortformOfCa seCompanyCurrency	string	☐		
caseInfo_CurrentCaseStatusDescr iption	string	☐		
caseInfo_CurrentCaseStatusGUID	string	☐		
caseInfo_CurrentCaseStatusInsert Ts	datetime	☐		
caseInfo_DeadlineDate	datetime	☐		
caseInfo_Description	string	☐		

Name	Data Type	Required	Default Value	Documentation
caseInfo_ExpectedValue	decimal	☐		
caseInfo_GUID	string	☐		
caseInfo_IncludeExceptions	boolean	☐		
caseInfo_IncludeOverTime	boolean	☐		
caseInfo_InsertTs	datetime	☐		
caseInfo_InternalName	string	☐		
caseInfo_InvoiceTemplateGUID	string	☐		
caseInfo_IsClosed	boolean	☐		
caseInfo_IsDailyAllow anceBillable	boolean	☐		
caseInfo_IsInternal	boolean	☐		
caseInfo_IsMileageBillable	boolean	☐		
caseInfo_IsOtherTravelExpensesBillable	boolean	☐		
caseInfo_LatestEstimationDate	datetime	☐		
caseInfo_LeadRating	int32	☐		
caseInfo_LeadSourceGUID	string	☐		
caseInfo_Name	string	☐		
caseInfo_OrderNumber	string	☐		
caseInfo_OurReference	string	☐		
caseInfo_OverdueInterest	decimal	☐		
caseInfo_PaymentTerm	int32	☐		
caseInfo_PricelistGUID	string	☐		
caseInfo_PricingRule	string	☐		
caseInfo_PricingType	string	☐		
caseInfo_Priority	int32	☐		
caseInfo_Probability	int32	☐		
caseInfo_RootTaskGUID	string	☐		
caseInfo_SalesCloseDate	datetime	☐		
caseInfo_SalesPersonUserCode	string	☐		
caseInfo_SalesPersonUserGUID	string	☐		
caseInfo_SalesProcessGUID	string	☐		
caseInfo_SalesStatusGUID	string	☐		
caseInfo_SharedToExtranet	boolean	☐		
caseInfo_StartDate	datetime	☐		
caseInfo_UpdatedTs	datetime	☐		
caseInfo_UrlToCase	string	☐		
caseInfo_UseDefaultProductsInWorkTimeEntry	boolean	☐		
caseInfo_UseDefaultWorkTypes	boolean	☐		
caseInfo_YourReference	string	☐		

Columns of Table Function

The columns of the table function AddNewCase are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		
CASEOWNERUSERGUID	string	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	string	Ddescription		
EXPECTEDVALUE	decimal	Expected Value		
GUID	string	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		

Name	Data Type	Label	Required	Documentation
INSERTTS	datetime	Inserted		
INTERNALNAME	string	Internal Name		
INVOICETEMPLATEGUID	string	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	int32	Lead Rating		
LEADSOURCEGUID	string	Lead Source GUID		
NAME	string	Name		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
PRICINGRULE	string	Pricing Rule		
PRICINGTYPE	string	Pricing Type		
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOTTASKGUID	string	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	string	Sales Person User Code		
SALESPERSONUSERGUID	string	Sales Person User GUID		
SALESPROCESSGUID	string	Sales Process GUID		
SALESSTATUSGUID	string	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	string	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	string	Your Reference		

3.1.38 AddNewCaseStatus: Visma Severa Add New Project Status

Catalog: Severa

Schema: API

Label: Add New Project Status

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseStatusInfo_GUID- caseStatusInfo_Icon- caseStatusInfo_IsActive- caseStatusInfo_Name- insertPosition

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewCaseStatus. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseStatusInfo_GUID	string	☐		
caseStatusInfo_Icon	string	☐		
caseStatusInfo_IsActive	boolean	☐		
caseStatusInfo_Name	string	☐		
insertPosition	int32	☐		

Columns of Table Function

The columns of the table function AddNewCaseStatus are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.39 AddNewCaseTag: Visma Severa Add New Project Tag

Catalog: Severa

Schema: API

Label: Add New Project Tag

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGUID- tagInfo_Context- tagInfo_GUID- tagInfo_IsActive- tagInfo_Keyword- tagInfo_Weight

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewCaseTag. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGUID	string	☐		
tagInfo_Context	string	☐		
tagInfo_GUID	string	☐		
tagInfo_IsActive	boolean	☐		
tagInfo_Keyword	string	☐		
tagInfo_Weight	int32	☐		

Columns of Table Function

The columns of the table function AddNewCaseTag are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTEXT	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyword		
WEIGHT	int32	Weight		

3.1.40 AddNewContact: Visma Severa Add New Contact

Catalog: Severa

Schema: API

Label: Add New Contact

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: contactInfo_AccountGUID- contactInfo_AddressGUID- contactInfo_CommunicationMethods- contactInfo_ContactRoleGUID- contactInfo_DateOfBirth- contactInfo_Description- contactInfo_FirstName- contactInfo_GUID- contactInfo_IsActive- contactInfo_JobTitle- contactInfo_Keywords- contactInfo_LanguageGUID- contactInfo_LastName- contactInfo_Salutation- contactInfo_SatisfactionLevel- contactInfo_TimeZoneGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewContact. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
contactInfo_AccountGUID	string	☐		
contactInfo_AddressGUID	string	☐		
contactInfo_CommunicationMethods	string	☐		
contactInfo_ContactRoleGUID	string	☐		
contactInfo_DateOfBirth	datetime	☐		
contactInfo_Description	string	☐		
contactInfo_FirstName	string	☐		
contactInfo_GUID	string	☐		
contactInfo_IsActive	boolean	☐		
contactInfo_JobTitle	string	☐		
contactInfo_Keywords	string	☐		
contactInfo_LanguageGUID	string	☐		
contactInfo_LastName	string	☐		
contactInfo_Salutation	string	☐		
contactInfo_SatisfactionLevel	int32	☐		

Name	Data Type	Required	Default Value	Documentation
contactInfo_TimeZoneGUID	string	☐		

Columns of Table Function

The columns of the table function AddNewContact are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ADDRESSGUID	string	Address GUID		
COMMUNICATIONMETHODS	string			
CONTACTROLEGUID	string	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	string	Ddescription		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	string	Job Title		
KEYWORDS	string			
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
SALUTATION	string	Salutation		
SATISFACTIONLEVEL	int32	Satisfaction Level		
TIMEZONEGUID	string	Timezone GUID		

3.1.41 AddNewContactTag: Visma Severa Add New Contact Tag

Catalog: Severa

Schema: API

Label: Add New Contact Tag

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: contactGUID- tagInfo_Context- tagInfo_GUID- tagInfo_IsActive- tagInfo_Keyword- tagInfo_Weight

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewContactTag. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
contactGUID	string	☐		
tagInfo_Context	string	☐		
tagInfo_GUID	string	☐		
tagInfo_IsActive	boolean	☐		
tagInfo_Keyword	string	☐		
tagInfo_Weight	int32	☐		

Columns of Table Function

The columns of the table function AddNewContactTag are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTEXT	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyword		
WEIGHT	int32	Weight		

3.1.42 AddNewCustomer: Visma Severa Add New Customer

Catalog: Severa

Schema: API

Label: Add New Customer

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: customer_AccountGroupGUIDs- customer_AccountOwnerUserGUID- customer_AccountRating- customer_AnnualRevenue- customer_AnnualRevenue2- customer_CompanyGUID- customer_CurrencyCode- customer_CurrencyGUID- customer_Email- customer_Employees- customer_GUID- customer_HeadOfficeAddressGUID- customer_IndustryGUID- customer_InsertTS- customer_InvoicingVat- customer_IsActive- customer_IsInternal- customer_LanguageCode- customer_LanguageGUID- customer_Name- customer_Notes- customer_Number- customer_OverdueInterest- customer_PaymentTerm- customer_PricelistGUID- customer_ReverseCharge- customer_ReverseChargeDescription-

customer_TimezoneGUID- customer_UpdatedTS- customer_UrlToAccount- customer_VatNumber- customer_Website- customer_eInvoiceAddress- customer_eInvoiceOperator

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewCustomer. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
customer_AccountGroupGUIDs	string	☐		
customer_AccountOwnerUserGUID	string	☐		
customer_AccountRating	int32	☐		
customer_AnnualRevenue	int32	☐		
customer_AnnualRevenue2	int64	☐		
customer_CompanyGUID	string	☐		
customer_CurrencyCode	string	☐		
customer_CurrencyGUID	string	☐		
customer_eInvoiceAddress	string	☐		
customer_eInvoiceOperator	string	☐		
customer_Email	string	☐		
customer_Employees	int32	☐		
customer_GUID	string	☐		
customer_HeadOfficeAddressGUID	string	☐		
customer_IndustryGUID	string	☐		
customer_InsertTS	datetime	☐		
customer_InvoicingVat	decimal	☐		
customer_IsActive	boolean	☐		
customer_IsInternal	boolean	☐		
customer_LanguageCode	string	☐		
customer_LanguageGUID	string	☐		
customer_Name	string	☐		
customer_Notes	string	☐		
customer_Number	int64	☐		

Name	Data Type	Required	Default Value	Documentation
customer_OverdueInterest	decimal	☐		
customer_PaymentTerm	int32	☐		
customer_PricelistGUID	string	☐		
customer_ReverseCharge	boolean	☐		
customer_ReverseChargeDescription	string	☐		
customer_TimezoneGUID	string	☐		
customer_UpdatedTS	datetime	☐		
customer_UrlToAccount	string	☐		
customer_VatNumber	string	☐		
customer_Website	string	☐		

Columns of Table Function

The columns of the table function AddNewCustomer are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
ANNUALREVENUE	int32	Annual Revenue		
ANNUALREVENUE2	int64			
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
EMAIL	string	Email		
EMPLOYEES	int32	Employees		
GUID	string	GUID		
HEADOFFICEADDRESSGUID	string	Head Office Address GUID		
INDUSTRYGUID	string	Industry GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		

Name	Data Type	Label	Required	Documentation
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		
VATNUMBER	string	VAT Number		
WEBSITE	string	Website		

3.1.43 AddNewEmployment: Visma Severa Add New Employment

Catalog: Severa

Schema: API

Label: Add New Employment

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: userGUID- employmentInfo_CompanyBusinessUnitCode- employmentInfo_CompanyBusinessUnitGUID- employmentInfo_CompanyBusinessUnitName- employmentInfo_DailyHours- employmentInfo_EndDate- employmentInfo_GUID- employmentInfo_HourCost- employmentInfo_IsOvertimeAllowed- employmentInfo_StartDate- employmentInfo_Title

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewEmployment. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
employmentInfo_CompanyBusinessUnitCode	string	☐		
employmentInfo_CompanyBusinessUnitGUID	string	☐		
employmentInfo_CompanyBusinessUnitName	string	☐		
employmentInfo_DailyHours	decimal	☐		
employmentInfo_EndDate	datetime	☐		
employmentInfo_GUID	string	☐		
employmentInfo_HourCost	decimal	☐		
employmentInfo_IsOvertimeAllowed	boolean	☐		
employmentInfo_StartDate	datetime	☐		
employmentInfo_Title	string	☐		
userGUID	string	☐		

Columns of Table Function

The columns of the table function AddNewEmployment are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
COMPANYBUSINESSUNITCODE	string	Company Business Unit Code		
COMPANYBUSINESSUNITGUID	string	Company Business Unit GUID		
COMPANYBUSINESSUNITNAME	string	Company Business Unit Name		
DAILYHOURS	decimal	Daily Hours		
ENDDATE	datetime	End Date		
GUID	string	GUID		
HOURCOST	decimal	Hour Cost		
ISOVERTIMEALLOWED	boolean	Is Overtime Allowed		
STARTDATE	datetime	Start Date		
TITLE	string	Title		

3.1.44 AddNewExpenseToProductByPoNumber: Visma Severa Add New Expense to Product by PO Number

Catalog: Severa

Schema: API

Label: Add New Expense to Product by PO Number

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: poNumber- itemInfo_BillingDependencyTaskGUID- itemInfo_BillingSchedule- itemInfo_CaseGUID- itemInfo_CaseNumber- itemInfo_CostCenterGUID- itemInfo_CostCenterIdentifier- itemInfo_CostCurrencyShortform- itemInfo_CostCurrencyShortformOfInvoicingCurrency- itemInfo_CostCurrencyShortformOfOrganizationCurrency- itemInfo_CostCurrencyShortformOfUserCompanyCurrency- itemInfo_Description- itemInfo_DisplayPeriodStart- itemInfo_EventDate- itemInfo_GUID- itemInfo_InvoiceDescription- itemInfo_InvoiceGUID- itemInfo_InvoiceQuantity- itemInfo_InvoiceRowGUID- itemInfo_IsApproved- itemInfo_IsBillable- itemInfo_IsOutSourcingCost- itemInfo_IsReadOnly- itemInfo_IsRealised- itemInfo_MeasurementUnit- itemInfo_Name- itemInfo_OrderDate- itemInfo_PlannedBillingDate- itemInfo_PriceCurrencyShortformOfCaseCompanyCurrency- itemInfo_PriceCurrencyShortformOfInvoicingCurrency- itemInfo_PriceCurrencyShortformOfOrganizationCurrency- itemInfo_PriceCurrencyShortformOfUserCompanyCurrency- itemInfo_ProductCode- itemInfo_ProductGUID- itemInfo_ProductType- itemInfo_PurchaseOrderNumber- itemInfo_PurchaseVAT- itemInfo_Quantity- itemInfo_RecurrenceEndDate- itemInfo_RecurrenceEndType- itemInfo_RecurrenceFrequency- itemInfo_RecurrenceStartDate- itemInfo_RecurrenceTimes- itemInfo_RecurringItemGUID- itemInfo_SalesAccountGUID- itemInfo_SalesStatusGUID- itemInfo_TaskGUID- itemInfo_TravelEndTime- itemInfo_TravelStartTime- itemInfo_UnitCost- itemInfo_UnitCostInInvoicingCurrency- itemInfo_UnitCostInOrganizationCurrency- itemInfo_UnitCostInUserCompanyCurrency- itemInfo_UnitCostVatIncludedAmount- itemInfo_UnitPrice- itemInfo_UnitPriceInBaseCurrency- itemInfo_UnitPriceInCaseCompanyCurrency- itemInfo_UnitPriceInUserCompanyCurrency- itemInfo_UserCode- itemInfo_UserGUID- itemInfo_VAT

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewExpenseToProductByPoNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
itemInfo_BillingDependencyTaskGUID	string	☐		
itemInfo_BillingSchedule	string	☐		
itemInfo_CaseGUID	string	☐		
itemInfo_CaseNumber	int64	☐		

Name	Data Type	Required	Default Value	Documentation
itemInfo_CostCenterGUID	string	☐		
itemInfo_CostCenterIdentifier	string	☐		
itemInfo_CostCurrencyShortform	string	☐		
itemInfo_CostCurrencyShortformOfInvoicingCurrency	string	☐		
itemInfo_CostCurrencyShortformOfOrganizationCurrency	string	☐		
itemInfo_CostCurrencyShortformOfUserCompanyCurrency	string	☐		
itemInfo_Description	string	☐		
itemInfo_DisplayPeriodStart	datetime	☐		
itemInfo_EventDate	datetime	☐		
itemInfo_GUID	string	☐		
itemInfo_InvoiceDescription	string	☐		
itemInfo_InvoiceGUID	string	☐		
itemInfo_InvoiceQuantity	decimal	☐		
itemInfo_InvoiceRow GUID	string	☐		
itemInfo_IsApproved	boolean	☐		
itemInfo_IsBillable	boolean	☐		
itemInfo_IsOutSourcingCost	boolean	☐		
itemInfo_IsReadOnly	boolean	☐		
itemInfo_IsRealised	boolean	☐		
itemInfo_MeasurementUnit	string	☐		
itemInfo_Name	string	☐		
itemInfo_OrderDate	datetime	☐		
itemInfo_PlannedBillingDate	datetime	☐		
itemInfo_PriceCurrencyShortformOfCaseCompanyCurrency	string	☐		
itemInfo_PriceCurrencyShortformOfInvoicingCurrency	string	☐		
itemInfo_PriceCurrencyShortformOfOrganizationCurrency	string	☐		
itemInfo_PriceCurrencyShortformOfUserCompanyCurrency	string	☐		
itemInfo_ProductCode	string	☐		
itemInfo_ProductGUID	string	☐		
itemInfo_ProductType	string	☐		
itemInfo_PurchaseOrderNumber	string	☐		
itemInfo_PurchaseVAT	decimal	☐		
itemInfo_Quantity	decimal	☐		
itemInfo_RecurrenceEndDate	datetime	☐		
itemInfo_RecurrenceEndType	int32	☐		
itemInfo_RecurrenceFrequency	int32	☐		

Name	Data Type	Required	Default Value	Documentation
itemInfo_RecurrenceStartDate	datetime	☐		
itemInfo_RecurrenceTimes	int32	☐		
itemInfo_RecurringItemGUID	string	☐		
itemInfo_SalesAccountGUID	string	☐		
itemInfo_SalesStatusGUID	string	☐		
itemInfo_TaskGUID	string	☐		
itemInfo_TravelEndTime	datetime	☐		
itemInfo_TravelStartTime	datetime	☐		
itemInfo_UnitCost	decimal	☐		
itemInfo_UnitCostInInvoicingCurrency	decimal	☐		
itemInfo_UnitCostInOrganizationCurrency	decimal	☐		
itemInfo_UnitCostInUserCompanyCurrency	decimal	☐		
itemInfo_UnitCostVatIncludedAmount	decimal	☐		
itemInfo_UnitPrice	decimal	☐		
itemInfo_UnitPriceInBaseCurrency	decimal	☐		
itemInfo_UnitPriceInCaseCompanyCurrency	decimal	☐		
itemInfo_UnitPriceInUserCompanyCurrency	decimal	☐		
itemInfo_UserCode	string	☐		
itemInfo_UserGUID	string	☐		
itemInfo_VAT	decimal	☐		
poNumber	string	☐		

Columns of Table Function

The columns of the table function AddNewExpenseToProductByPoNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing		

Name	Data Type	Label	Required	Documentation
		Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		

Name	Data Type	Label	Required	Documentation
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.45 AddNewFileTag: Visma Severa Add New File Tag

Catalog: Severa

Schema: API

Label: Add New File Tag

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: fileGUID- tagInfo_Context- tagInfo_GUID- tagInfo_IsActive- tagInfo_Keyword- tagInfo_Weight

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewFileTag. A value must be provided at all times for required parameters, but optional

parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
fileGUID	string	☐		
tagInfo_Context	string	☐		
tagInfo_GUID	string	☐		
tagInfo_IsActive	boolean	☐		
tagInfo_Keyword	string	☐		
tagInfo_Weight	int32	☐		

Columns of Table Function

The columns of the table function AddNewFileTag are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTEXT	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyword		
WEIGHT	int32	Weight		

3.1.46 AddNewHourEntry: Visma Severa Add New Hour Entry

Catalog: Severa

Schema: API

Label: Add New Hour Entry

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: hourEntryInfo_AccountGUID- hourEntryInfo_CaseBusinessUnitCode- hourEntryInfo_CaseCostCenterIdentifier- hourEntryInfo_CaseGUID- hourEntryInfo_CaseName- hourEntryInfo_CaseNumber- hourEntryInfo_CostCurrencyShortform- hourEntryInfo_CostCurrencyShortformOfInvoicingCurrency-

hourEntryInfo_CostCurrencyShortformOfOrganizationCurrency-
hourEntryInfo_CostCurrencyShortformOfUserCompanyCurrency-
hourEntryInfo_Description- hourEntryInfo_EndTime- hourEntryInfo_EventDate-
hourEntryInfo_GUID- hourEntryInfo_InvoiceDescription- hourEntryInfo_InvoiceGUID-
hourEntryInfo_InvoiceQuantity- hourEntryInfo_InvoiceRowGUID- hourEntryInfo_IsApproved-
hourEntryInfo_IsBillable- hourEntryInfo_IsProductive- hourEntryInfo_IsReadOnly-
hourEntryInfo_OverTimeGUID- hourEntryInfo_OvertimeCode- hourEntryInfo_PhaseGUID-
hourEntryInfo_PriceCurrencyShortformOfCaseCompanyCurrency-
hourEntryInfo_PriceCurrencyShortformOfInvoicingCurrency-
hourEntryInfo_PriceCurrencyShortformOfOrganizationCurrency-
hourEntryInfo_PriceCurrencyShortformOfUserCompanyCurrency- hourEntryInfo_Quantity-
hourEntryInfo_StartTime- hourEntryInfo_TaskGUID- hourEntryInfo_UnitCost-
hourEntryInfo_UnitCostInInvoicingCurrency- hourEntryInfo_UnitCostInOrganizationCurrency-
hourEntryInfo_UnitCostInUserCompanyCurrency- hourEntryInfo_UnitPrice-
hourEntryInfo_UnitPriceInBaseCurrency- hourEntryInfo_UnitPriceInCaseCompanyCurrency-
hourEntryInfo_UnitPriceInUserCompanyCurrency- hourEntryInfo_UseInvoiceQuantity-
hourEntryInfo_UserBusinessUnitCode- hourEntryInfo_UserCode-
hourEntryInfo_UserFirstName- hourEntryInfo_UserGUID- hourEntryInfo_UserLastName-
hourEntryInfo_WorkTypeCode- hourEntryInfo_WorkTypeGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewHourEntry. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
hourEntryInfo_AccountGUID	string	☐		
hourEntryInfo_CaseBusinessUnitCode	string	☐		
hourEntryInfo_CaseCostCenterIdentifier	string	☐		
hourEntryInfo_CaseGUID	string	☐		
hourEntryInfo_CaseName	string	☐		
hourEntryInfo_CaseNumber	int64	☐		
hourEntryInfo_CostCurrencyShortform	string	☐		
hourEntryInfo_CostCurrencyShortformOfInvoicingCurrency	string	☐		
hourEntryInfo_CostCurrencyShortformOfOrganizationCurrency	string	☐		

Name	Data Type	Required	Default Value	Documentation
hourEntryInfo_CostCurrencyShortformOfUserCompanyCurrency	string	☐		
hourEntryInfo_Description	string	☐		
hourEntryInfo_EndTime	datetime	☐		
hourEntryInfo_EventDate	datetime	☐		
hourEntryInfo_GUID	string	☐		
hourEntryInfo_InvoiceDescription	string	☐		
hourEntryInfo_InvoiceGUID	string	☐		
hourEntryInfo_InvoiceQuantity	decimal	☐		
hourEntryInfo_InvoiceRow GUID	string	☐		
hourEntryInfo_IsApproved	boolean	☐		
hourEntryInfo_IsBillable	boolean	☐		
hourEntryInfo_IsProductive	boolean	☐		
hourEntryInfo_IsReadOnly	boolean	☐		
hourEntryInfo_OvertimeCode	string	☐		
hourEntryInfo_OverTimeGUID	string	☐		
hourEntryInfo_PhaseGUID	string	☐		
hourEntryInfo_PriceCurrencyShortformOfCaseCompanyCurrency	string	☐		
hourEntryInfo_PriceCurrencyShortformOfInvoicingCurrency	string	☐		
hourEntryInfo_PriceCurrencyShortformOfOrganizationCurrency	string	☐		
hourEntryInfo_PriceCurrencyShortformOfUserCompanyCurrency	string	☐		
hourEntryInfo_Quantity	decimal	☐		
hourEntryInfo_StartTime	datetime	☐		
hourEntryInfo_TaskGUID	string	☐		
hourEntryInfo_UnitCost	decimal	☐		
hourEntryInfo_UnitCostInInvoicingCurrency	decimal	☐		
hourEntryInfo_UnitCostInOrganizationCurrency	decimal	☐		
hourEntryInfo_UnitCostInUserCompanyCurrency	decimal	☐		
hourEntryInfo_UnitPrice	decimal	☐		
hourEntryInfo_UnitPriceInBaseCurrency	decimal	☐		
hourEntryInfo_UnitPriceInCaseCompanyCurrency	decimal	☐		
hourEntryInfo_UnitPriceInUserCompanyCurrency	decimal	☐		
hourEntryInfo_UseInvoiceQuantity	boolean	☐		
hourEntryInfo_UserBusinessUnitCode	string	☐		

Name	Data Type	Required	Default Value	Documentation
hourEntryInfo_UserCode	string	☐		
hourEntryInfo_UserFirstName	string	☐		
hourEntryInfo_UserGUID	string	☐		
hourEntryInfo_UserLastName	string	☐		
hourEntryInfo_WorkTypeCode	string	☐		
hourEntryInfo_WorkTypeGUID	string	☐		

Columns of Table Function

The columns of the table function AddNewHourEntry are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		

Name	Data Type	Label	Required	Documentation
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.47 AddNewIndustry: Visma Severa Add New Industry

Catalog: Severa

Schema: API

Label: Add New Industry

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: industryInfo_Code- industryInfo_GUID- industryInfo_IsActive- industryInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewIndustry. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
industryInfo_Code	string	☐		
industryInfo_GUID	string	☐		
industryInfo_IsActive	boolean	☐		
industryInfo_Name	string	☐		

Columns of Table Function

The columns of the table function AddNewIndustry are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.48 AddNewInvoiceLogEntry: Visma Severa Add New Invoice Log Entry

Catalog: Severa

Schema: API

Label: Add New Invoice Log Entry

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: invoiceGuid- context- text- source- eventTimeUtc- ignoreDuplicates

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewInvoiceLogEntry. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
context	string	☐		
eventTimeUtc	datetime	☐		
ignoreDuplicates	boolean	☐		
invoiceGuid	string	☐		
source	string	☐		
text	string	☐		

Columns of Table Function

The columns of the table function AddNewInvoiceLogEntry are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DUMMY	string	Dummy	☑	Dummy column

3.1.49 AddNewItem: Visma Severa Add New Item

Catalog: Severa

Schema: API

Label: Add New Item

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: itemInfo_BillingDependencyTaskGUID- itemInfo_BillingSchedule- itemInfo_CaseGUID- itemInfo_CaseNumber- itemInfo_CostCenterGUID- itemInfo_CostCenterIdentifier- itemInfo_CostCurrencyShortform- itemInfo_CostCurrencyShortformOfInvoicingCurrency- itemInfo_CostCurrencyShortformOfOrganizationCurrency- itemInfo_CostCurrencyShortformOfUserCompanyCurrency- itemInfo_Description- itemInfo_DisplayPeriodStart- itemInfo_EventDate- itemInfo_GUID- itemInfo_InvoiceDescription- itemInfo_InvoiceGUID- itemInfo_InvoiceQuantity- itemInfo_InvoiceRowGUID- itemInfo_IsApproved- itemInfo_IsBillable- itemInfo_IsOutSourcingCost- itemInfo_IsReadOnly- itemInfo_IsRealised- itemInfo_MeasurementUnit- itemInfo_Name- itemInfo_OrderDate- itemInfo_PlannedBillingDate- itemInfo_PriceCurrencyShortformOfCaseCompanyCurrency- itemInfo_PriceCurrencyShortformOfInvoicingCurrency- itemInfo_PriceCurrencyShortformOfOrganizationCurrency- itemInfo_PriceCurrencyShortformOfUserCompanyCurrency- itemInfo_ProductCode- itemInfo_ProductGUID- itemInfo_ProductType- itemInfo_PurchaseOrderNumber- itemInfo_PurchaseVAT- itemInfo_Quantity- itemInfo_RecurrenceEndDate- itemInfo_RecurrenceEndType- itemInfo_RecurrenceFrequency- itemInfo_RecurrenceStartDate- itemInfo_RecurrenceTimes- itemInfo_RecurringItemGUID- itemInfo_SalesAccountGUID- itemInfo_SalesStatusGUID- itemInfo_TaskGUID- itemInfo_TravelEndTime- itemInfo_TravelStartTime- itemInfo_UnitCost- itemInfo_UnitCostInInvoicingCurrency- itemInfo_UnitCostInOrganizationCurrency- itemInfo_UnitCostInUserCompanyCurrency- itemInfo_UnitCostVatIncludedAmount- itemInfo_UnitPrice- itemInfo_UnitPriceInBaseCurrency- itemInfo_UnitPriceInCaseCompanyCurrency- itemInfo_UnitPriceInUserCompanyCurrency- itemInfo_UserCode- itemInfo_UserGUID- itemInfo_VAT- currencyCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewItem. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
currencyCode	string	☐		
itemInfo_BillingDependencyTaskGUID	string	☐		
itemInfo_BillingSchedule	string	☐		
itemInfo_CaseGUID	string	☐		
itemInfo_CaseNumber	int64	☐		
itemInfo_CostCenterGUID	string	☐		

Name	Data Type	Required	Default Value	Documentation
itemInfo_CostCenterIdentifier	string	☐		
itemInfo_CostCurrencyShortform	string	☐		
itemInfo_CostCurrencyShortformOfInvoicingCurrency	string	☐		
itemInfo_CostCurrencyShortformOfOrganizationCurrency	string	☐		
itemInfo_CostCurrencyShortformOfUserCompanyCurrency	string	☐		
itemInfo_Description	string	☐		
itemInfo_DisplayPeriodStart	datetime	☐		
itemInfo_EventDate	datetime	☐		
itemInfo_GUID	string	☐		
itemInfo_InvoiceDescription	string	☐		
itemInfo_InvoiceGUID	string	☐		
itemInfo_InvoiceQuantity	decimal	☐		
itemInfo_InvoiceRow GUID	string	☐		
itemInfo_IsApproved	boolean	☐		
itemInfo_IsBillable	boolean	☐		
itemInfo_IsOutSourcingCost	boolean	☐		
itemInfo_IsReadOnly	boolean	☐		
itemInfo_IsRealised	boolean	☐		
itemInfo_MeasurementUnit	string	☐		
itemInfo_Name	string	☐		
itemInfo_OrderDate	datetime	☐		
itemInfo_PlannedBillingDate	datetime	☐		
itemInfo_PriceCurrencyShortformOfCaseCompanyCurrency	string	☐		
itemInfo_PriceCurrencyShortformOfInvoicingCurrency	string	☐		
itemInfo_PriceCurrencyShortformOfOrganizationCurrency	string	☐		
itemInfo_PriceCurrencyShortformOfUserCompanyCurrency	string	☐		
itemInfo_ProductCode	string	☐		
itemInfo_ProductGUID	string	☐		
itemInfo_ProductType	string	☐		
itemInfo_PurchaseOrderNumber	string	☐		
itemInfo_PurchaseVAT	decimal	☐		
itemInfo_Quantity	decimal	☐		
itemInfo_RecurrenceEndDate	datetime	☐		
itemInfo_RecurrenceEndType	int32	☐		
itemInfo_RecurrenceFrequency	int32	☐		
itemInfo_RecurrenceStartDate	datetime	☐		

Name	Data Type	Required	Default Value	Documentation
itemInfo_RecurrenceTimes	int32	☐		
itemInfo_RecurringItemGUID	string	☐		
itemInfo_SalesAccountGUID	string	☐		
itemInfo_SalesStatusGUID	string	☐		
itemInfo_TaskGUID	string	☐		
itemInfo_TravelEndTime	datetime	☐		
itemInfo_TravelStartTime	datetime	☐		
itemInfo_UnitCost	decimal	☐		
itemInfo_UnitCostInInvoicingCurrency	decimal	☐		
itemInfo_UnitCostInOrganizationCurrency	decimal	☐		
itemInfo_UnitCostInUserCompanyCurrency	decimal	☐		
itemInfo_UnitCostVatIncludedAmount	decimal	☐		
itemInfo_UnitPrice	decimal	☐		
itemInfo_UnitPriceInBaseCurrency	decimal	☐		
itemInfo_UnitPriceInCaseCompanyCurrency	decimal	☐		
itemInfo_UnitPriceInUserCompanyCurrency	decimal	☐		
itemInfo_UserCode	string	☐		
itemInfo_UserGUID	string	☐		
itemInfo_VAT	decimal	☐		

Columns of Table Function

The columns of the table function AddNewItem are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		

Name	Data Type	Label	Required	Documentation
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		

Name	Data Type	Label	Required	Documentation
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.50 AddNewItems: Visma Severa Add New Items

Catalog: Severa

Schema: API

Label: Add New Items

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: items

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewItems. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
items	string	☐		

Columns of Table Function

The columns of the table function AddNewItems are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		

Name	Data Type	Label	Required	Documentation
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		

Name	Data Type	Label	Required	Documentation
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.51 AddNewPhase: Visma Severa Add New Phase

Catalog: Severa

Schema: API

Label: Add New Phase

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: phaseInfo_CaseGUID- phaseInfo_Code- phaseInfo_Deadline- phaseInfo_Description- phaseInfo_FlatRate_AdditionalHoursUnitPrice- phaseInfo_FlatRate_AreAdditionalHoursBillable- phaseInfo_FlatRate_Hours- phaseInfo_FlatRate_Price- phaseInfo_GUID- phaseInfo_IsCompleted- phaseInfo_IsLocked- phaseInfo_Name- phaseInfo_OriginalDeadline- phaseInfo_OriginalPlannedStartDate- phaseInfo_OriginalWorkEstimate- phaseInfo_OwnerUserGUID- phaseInfo_ParentPhaseGUID- phaseInfo_PlannedStartDate- phaseInfo_PricePerHour- phaseInfo_SortOrder- phaseInfo_WorkEstimate- phaseInfo_WorkTypeGUID- members

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewPhase. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
members	string	☐		
phaseInfo_CaseGUID	string	☐		

Name	Data Type	Required	Default Value	Documentation
phaseInfo_Code	string	☐		
phaseInfo_Deadline	datetime	☐		
phaseInfo_Description	string	☐		
phaseInfo_FlatRate_AdditionalHoursUnitPrice	decimal	☐		
phaseInfo_FlatRate_AreAdditionalHoursBillable	boolean	☐		
phaseInfo_FlatRate_Hours	decimal	☐		
phaseInfo_FlatRate_Price	decimal	☐		
phaseInfo_GUID	string	☐		
phaseInfo_IsCompleted	boolean	☐		
phaseInfo_IsLocked	boolean	☐		
phaseInfo_Name	string	☐		
phaseInfo_OriginalDeadline	datetime	☐		
phaseInfo_OriginalPlannedStartDate	datetime	☐		
phaseInfo_OriginalWorkEstimate	decimal	☐		
phaseInfo_OwnerUserGUID	string	☐		
phaseInfo_ParentPhaseGUID	string	☐		
phaseInfo_PlannedStartDate	datetime	☐		
phaseInfo_PricePerHour	decimal	☐		
phaseInfo_SortOrder	int32	☐		
phaseInfo_WorkEstimate	decimal	☐		
phaseInfo_WorkTypeGUID	string	☐		

Columns of Table Function

The columns of the table function AddNewPhase are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CASEGUID	string	Project GUID		
CODE	string	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	string	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	decimal	Flat Rate Additional Hours Unit Price		
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	decimal	Flat Rate Hours		
FLATRATE_PRICE	decimal	Flat Rate Price		
GUID	string	GUID		

Name	Data Type	Label	Required	Documentation
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	decimal	Original Work Estimate		
OWNERUSERGUID	string	Owner User GUID		
PARENTPHASEGUID	string	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	decimal	Price per Hour		
SORTORDER	int32	Sort Order		
WORKESTIMATE	decimal	Work Estimate		
WORKTYPEGUID	string	Work Type GUID		

3.1.52 AddNewProduct: Visma Severa Add New Product

Catalog: Severa

Schema: API

Label: Add New Product

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: productInfo_Code- productInfo_Description- productInfo_GUID- productInfo_IsActive- productInfo_MeasurementUnit- productInfo_Name- productInfo_Type- productInfo_UnitCost- productInfo_UnitPrice- productInfo_UseInWorkTimeEntry- productInfo_VAT- productInfo_ProductCategoryGUID- productInfo_SalesAccountGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewProduct. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
productInfo_Code	string	☐		
productInfo_Description	string	☐		
productInfo_GUID	string	☐		
productInfo_IsActive	boolean	☐		
productInfo_MeasurementUnit	string	☐		
productInfo_Name	string	☐		
productInfo_ProductCategoryGUID	string	☐		
productInfo_SalesAccountGUID	string	☐		
productInfo_Type	string	☐		
productInfo_UnitCost	decimal	☐		
productInfo_UnitPrice	decimal	☐		
productInfo_UseInWorkTimeEntry	boolean	☐		
productInfo_VAT	decimal	☐		

Columns of Table Function

The columns of the table function AddNewProduct are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
PRODUCTCATEGORYGUID	string	Product Category GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.53 AddNewProductCategory: Visma Severa Add New Product Category

Catalog: Severa

Schema: API

Label: Add New Product Category

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: productCategoryInfo_Code- productCategoryInfo_GUID- productCategoryInfo_IsActive- productCategoryInfo_IsDefault- productCategoryInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewProductCategory. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
productCategoryInfo_Code	string	☐		
productCategoryInfo_GUID	string	☐		
productCategoryInfo_IsActive	boolean	☐		
productCategoryInfo_IsDefault	boolean	☐		
productCategoryInfo_Name	string	☐		

Columns of Table Function

The columns of the table function AddNewProductCategory are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		

3.1.54 AddNewResource: Visma Severa Add New Resource

Catalog: Severa

Schema: API

Label: Add New Resource

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: resourceInfo_GUID- resourceInfo_IsActive- resourceInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewResource. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
resourceInfo_GUID	string	☐		
resourceInfo_IsActive	boolean	☐		
resourceInfo_Name	string	☐		

Columns of Table Function

The columns of the table function AddNewResource are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.55 AddNewResourceAllocation: Visma Severa Add New Resource Allocation

Catalog: Severa

Schema: API

Label: Add New Resource Allocation

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: resourceAllocationInfo_AllocatedHours- resourceAllocationInfo_AllocatedPercentage- resourceAllocationInfo_CaseGUID-

resourceAllocationInfo_EndDate- resourceAllocationInfo_GUID- resourceAllocationInfo_PhaseGUID- resourceAllocationInfo_StartDate- resourceAllocationInfo_UserGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewResourceAllocation. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
resourceAllocationInfo_AllocatedHours	decimal	☐		
resourceAllocationInfo_AllocatedPercentage	int32	☐		
resourceAllocationInfo_CaseGUID	string	☐		
resourceAllocationInfo_EndDate	datetime	☐		
resourceAllocationInfo_GUID	string	☐		
resourceAllocationInfo_PhaseGUID	string	☐		
resourceAllocationInfo_StartDate	datetime	☐		
resourceAllocationInfo_UserGUID	string	☐		

Columns of Table Function

The columns of the table function AddNewResourceAllocation are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.56 AddNewResourceAllocations: Visma Severa Add New Resource Allocations

Catalog: Severa

Schema: API

Label: Add New Resource Allocations

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: resourceAllocations

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewResourceAllocations. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
resourceAllocations	string	☐		

Columns of Table Function

The columns of the table function AddNewResourceAllocations are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.57 AddNewSalesProcess: Visma Severa Add New Sales Process

Catalog: Severa

Schema: API

Label: Add New Sales Process

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: salesProcessInfo_DefaultProbability- salesProcessInfo_GUID- salesProcessInfo_Icon- salesProcessInfo_IsActive- salesProcessInfo_IsAlsoItemStatus- salesProcessInfo_IsInProgress- salesProcessInfo_IsLost- salesProcessInfo_IsOffer- salesProcessInfo_IsWon- salesProcessInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewSalesProcess. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
salesProcessInfo_DefaultProbability	int32	☐		
salesProcessInfo_GUID	string	☐		
salesProcessInfo_Icon	string	☐		
salesProcessInfo_IsActive	boolean	☐		
salesProcessInfo_IsAlsoItemStatus	boolean	☐		
salesProcessInfo_IsInProgress	boolean	☐		
salesProcessInfo_IsLost	boolean	☐		
salesProcessInfo_IsOffer	boolean	☐		
salesProcessInfo_IsWon	boolean	☐		
salesProcessInfo_Name	string	☐		

Columns of Table Function

The columns of the table function AddNewSalesProcess are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DEFAULTPROBABILITY	int32	Default Probability		
GUID	string	GUID		
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
ISALSOITEMSTATUS	boolean	Is also Item Status		
ISINPROGRESS	boolean	Is in Progress		
ISLOST	boolean	Is Lost		
ISOFFER	boolean	Is Offer		
ISWON	boolean	Is Won		
NAME	string	Name		

3.1.58 AddNewSalesStatus: Visma Severa Add New Sales Status

Catalog: Severa

Schema: API

Label: Add New Sales Status

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: salesStatusInfo_CaseGUID- salesStatusInfo_GUID- salesStatusInfo_SalesProcessGUID- salesStatusInfo_TimeStamp

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewSalesStatus. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
salesStatusInfo_CaseGUID	string	☐		
salesStatusInfo_GUID	string	☐		
salesStatusInfo_SalesProcessGUID	string	☐		
salesStatusInfo_TimeStamp	datetime	☐		

Columns of Table Function

The columns of the table function AddNewSalesStatus are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CASEGUID	string	Project GUID		
GUID	string	GUID		
SALESPROCESSGUID	string	Sales Process GUID		
TIMESTAMP	datetime	Time Stamp		

3.1.59 AddNewTravelExpense: Visma Severa Add New Travel Expense

Catalog: Severa

Schema: API

Label: Add New Travel Expense

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: travelExpenseInfo_Code- travelExpenseInfo_Description- travelExpenseInfo_GUID- travelExpenseInfo_IsActive- travelExpenseInfo_MeasurementUnit- travelExpenseInfo_Name- travelExpenseInfo_Type- travelExpenseInfo_UnitCost- travelExpenseInfo_UnitPrice- travelExpenseInfo_UseInWorkTimeEntry- travelExpenseInfo_VAT- travelExpenseInfo_CostCurrencyGUID- travelExpenseInfo_ExpenseClass

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewTravelExpense. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
travelExpenseInfo_Code	string	☐		
travelExpenseInfo_CostCurrencyGUID	string	☐		

Name	Data Type	Required	Default Value	Documentation
travelExpenseInfo_Description	string	☐		
travelExpenseInfo_ExpenseClass	string	☐		
travelExpenseInfo_GUID	string	☐		
travelExpenseInfo_IsActive	boolean	☐		
travelExpenseInfo_MeasurementUnit	string	☐		
travelExpenseInfo_Name	string	☐		
travelExpenseInfo_Type	string	☐		
travelExpenseInfo_UnitCost	decimal	☐		
travelExpenseInfo_UnitPrice	decimal	☐		
travelExpenseInfo_UseInWorkTimeEntry	boolean	☐		
travelExpenseInfo_VAT	decimal	☐		

Columns of Table Function

The columns of the table function AddNewTravelExpense are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
COSTCURRENCYGUID	string	Cost Currency GUID		
DESCRIPTION	string	Ddescription		
EXPENSECLASS	string	Expense Class		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.60 AddNewUser: Visma Severa Add New User

Catalog: Severa

Schema: API

Label: Add New User

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: userInfo_Addressline- userInfo_BankAccountNumber- userInfo_BusinessUnitCode- userInfo_BusinessUnitGUID- userInfo_BusinessUnitName- userInfo_Code- userInfo_CountryCode- userInfo_CountryGUID- userInfo_CountryRegionGUID- userInfo_CountryRegionName- userInfo_Culture- userInfo_DefaultActivityTypeGUID- userInfo_Email- userInfo_FirstName- userInfo_GUID- userInfo_IsActive- userInfo_LanguageCode- userInfo_LanguageGUID- userInfo_LastName- userInfo_LicenseType- userInfo_Notes- userInfo_OrganizationGUID- userInfo_Phone- userInfo_PhotoFileGUID- userInfo_PostOffice- userInfo_PostalCode- userInfo_ProfileGUID- userInfo_Salutation- userInfo_SendDailyMail- userInfo_SendWeeklyMail- userInfo_SocialSecurityNumber- userInfo_SuperiorUserGUID- userInfo_TimezoneGUID- userInfo_Title- userInfo_WorktypeGUID- sendWelcomeEmail

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewUser. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
sendWelcomeEmail	boolean	☐		
userInfo_Addressline	string	☐		
userInfo_BankAccountNumber	string	☐		
userInfo_BusinessUnitCode	string	☐		
userInfo_BusinessUnitGUID	string	☐		
userInfo_BusinessUnitName	string	☐		
userInfo_Code	string	☐		
userInfo_CountryCode	string	☐		
userInfo_CountryGUID	string	☐		
userInfo_CountryRegionGUID	string	☐		
userInfo_CountryRegionName	string	☐		
userInfo_Culture	string	☐		
userInfo_DefaultActivityTypeGUID	string	☐		
userInfo_Email	string	☐		
userInfo_FirstName	string	☐		
userInfo_GUID	string	☐		

Name	Data Type	Required	Default Value	Documentation
userInfo_IsActive	boolean	☐		
userInfo_LanguageCode	string	☐		
userInfo_LanguageGUID	string	☐		
userInfo_LastName	string	☐		
userInfo_LicenseType	string	☐		
userInfo_Notes	string	☐		
userInfo_OrganizationGUID	string	☐		
userInfo_Phone	string	☐		
userInfo_PhotoFileGUID	string	☐		
userInfo_PostalCode	string	☐		
userInfo_PostOffice	string	☐		
userInfo_ProfileGUID	string	☐		
userInfo_Salutation	string	☐		
userInfo_SendDailyMail	boolean	☐		
userInfo_SendWeeklyMail	boolean	☐		
userInfo_SocialSecurityNumber	string	☐		
userInfo_SuperiorUserGUID	string	☐		
userInfo_TimezoneGUID	string	☐		
userInfo_Title	string	☐		
userInfo_WorktypeGUID	string	☐		

Columns of Table Function

The columns of the table function AddNewUser are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.61 AddNewUserTag: Visma Severa Add New User Tag

Catalog: Severa

Schema: API

Label: Add New User Tag

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: userGUID- tagInfo_Context- tagInfo_GUID- tagInfo_IsActive- tagInfo_Keyword- tagInfo_Weight

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewUserTag. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
tagInfo_Context	string	☐		
tagInfo_GUID	string	☐		
tagInfo_IsActive	boolean	☐		
tagInfo_Keyword	string	☐		
tagInfo_Weight	int32	☐		
userGUID	string	☐		

Columns of Table Function

The columns of the table function AddNewUserTag are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTEXT	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyword		
WEIGHT	int32	Weight		

3.1.62 AddNewWorkType: Visma Severa Add New Work Type

Catalog: Severa

Schema: API

Label: Add New Work Type

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: workTypeInfo_Code- workTypeInfo_GUID- workTypeInfo_IsActive- workTypeInfo_IsDefault- workTypeInfo_IsProductive- workTypeInfo_Name- workTypeInfo_SalesAccountGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewWorkType. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
workTypeInfo_Code	string	☐		
workTypeInfo_GUID	string	☐		
workTypeInfo_IsActive	boolean	☐		
workTypeInfo_IsDefault	boolean	☐		
workTypeInfo_IsProductive	boolean	☐		
workTypeInfo_Name	string	☐		
workTypeInfo_SalesAccountGUID	string	☐		

Columns of Table Function

The columns of the table function AddNewWorkType are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	string	Name		
SALESACCOUNTGUID	string	Sales Account GUID		

3.1.63 AddressByGUID: Visma Severa Address by GUID

Catalog: Severa

Schema: API

Label: Address by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: addressGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddressByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
addressGUID	string	☐		

Columns of Table Function

The columns of the table function AddressByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
ADDRESSLINE2	string	Addressline 2		
ADDRESSLINE3	string	Addressline 3		
CITY	string	City		
COMPANYGUID	string	Company GUID		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYNAME	string	Country Name		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
FAX	string	Fax		
GUID	string	GUID		
ISBILLINGADDRESS	boolean	Is Billing Address		
ISPOSTALADDRESS	boolean	Is Postal Address		
ISVISITADDRESS	boolean	Is Visit Address		
PHONE	string	Phone		
POSTALCODE	string	Postal Code		

3.1.64 AddressesChangedSince: Visma Severa Addresses Changed since

Catalog: Severa

Schema: API

Label: Addresses Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accountGUID- startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddressesChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountGUID	string	☐		
options	int32	☐		
startDateTime	datetime	☐		

Columns of Table Function

The columns of the table function AddressesChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
ADDRESSLINE2	string	Addressline 2		
ADDRESSLINE3	string	Addressline 3		
CITY	string	City		
COMPANYGUID	string	Company GUID		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYNAME	string	Country Name		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
FAX	string	Fax		
GUID	string	GUID		
ISBILLINGADDRESS	boolean	Is Billing Address		

Name	Data Type	Label	Required	Documentation
ISPOSTALADDRESS	boolean	Is Postal Address		
ISVISITADDRESS	boolean	Is Visit Address		
PHONE	string	Phone		
POSTALCODE	string	Postal Code		

3.1.65 BillingAddress: Visma Severa Billing Address

Catalog: Severa

Schema: API

Label: Billing Address

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table BillingAddress are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
ADDRESSLINE2	string	Addressline 2		
ADDRESSLINE3	string	Addressline 3		
CITY	string	City		
COMPANYGUID	string	Company GUID		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYNAME	string	Country Name		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
FAX	string	Fax		
GUID	string	GUID		
ISBILLINGADDRESS	boolean	Is Billing Address		
ISPOSTALADDRESS	boolean	Is Postal Address		
ISVISITADDRESS	boolean	Is Visit Address		
PHONE	string	Phone		
POSTALCODE	string	Postal Code		

3.1.66 BillingAndExpenseForecastByCaseGUID: Visma Severa Billing and Expense Forecast by Project GUID

Catalog: Severa

Schema: API

Label: Billing and Expense Forecast by Project GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function BillingAndExpenseForecastByCaseGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGuid	string	☐		

Columns of Table Function

The columns of the table function BillingAndExpenseForecastByCaseGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
BILLINGFORECAST	decimal	Billing Forecast		
EXPENSEFORECAST	decimal	Expense Forecast		
GUID	string	GUID		
INHOUSEREVENUEFORECAST	decimal	Inhouse Revenue Forecast		
MONTH	int32	Month		
NOTES	string	Notes		
OUTSOURCINGREVENUEFORECAST	decimal	Outsourcing Revenue Forecast		
YEAR	int32	Year		

3.1.67 BusinessUnitByCode: Visma Severa Business Unit by Code

Catalog: Severa

Schema: API

Label: Business Unit by Code

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function BusinessUnitByCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitCode	string	☐		

Columns of Table Function

The columns of the table function BusinessUnitByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
COSTCENTERGUID	string	Cost Center GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
FORMATTINGCULTURE	string	Formatting Culture		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NEXTINVOICENUMBER	int32	Next Invoice Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PARENTBUSINESSUNITGUID	string	Parent Business Unit GUID		

Name	Data Type	Label	Required	Documentation
PAYMENTTERM	int32	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	string	VAT Number		

3.1.68 BusinessUnitByGUID: Visma Severa Business Unit by GUID

Catalog: Severa

Schema: API

Label: Business Unit by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function BusinessUnitByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGuid	string	☐		

Columns of Table Function

The columns of the table function BusinessUnitByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		

Name	Data Type	Label	Required	Documentation
COSTCENTERGUID	string	Cost Center GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
FORMATTINGCULTURE	string	Formatting Culture		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NEXTINVOICENUMBER	int32	Next Invoice Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PARENTBUSINESSUNITGUID	string	Parent Business Unit GUID		
PAYMENTTERM	int32	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	string	VAT Number		

3.1.69 BusinessUnitHierarchy: Visma Severa Business Unit Hierarchy

Catalog: Severa

Schema: API

Label: Business Unit Hierarchy

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: rootBusinessUnitGuid- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function BusinessUnitHierarchy. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
options	int32	☐		
rootBusinessUnitGuid	string	☐		

Columns of Table Function

The columns of the table function BusinessUnitHierarchy are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
COSTCENTERGUID	string	Cost Center GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
FORMATTINGCULTURE	string	Formatting Culture		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NEXTINVOICENUMBER	int32	Next Invoice Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PARENTBUSINESSUNITGUID	string	Parent Business Unit GUID		
PAYMENTTERM	int32	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	string	VAT Number		

3.1.70 BusinessUnitsChangedSince: Visma Severa Business Units Changed since

Catalog: Severa

Schema: API

Label: Business Units Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function BusinessUnitsChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
options	int32	☐		
startDateTime	datetime	☐		

Columns of Table Function

The columns of the table function BusinessUnitsChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
COSTCENTERGUID	string	Cost Center GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
FORMATTINGCULTURE	string	Formatting Culture		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NEXTINVOICENUMBER	int32	Next Invoice Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PARENTBUSINESSUNITGUID	string	Parent Business Unit GUID		

Name	Data Type	Label	Required	Documentation
PAYMENTTERM	int32	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	string	VAT Number		

3.1.71 CaseByGUID: Visma Severa Project by GUID

Catalog: Severa

Schema: API

Label: Project by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CaseByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGuid	string	☐		

Columns of Table Function

The columns of the table function CaseByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		

Name	Data Type	Label	Required	Documentation
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		
CASEOWNERUSERGUID	string	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	string	Ddescription		
EXPECTEDVALUE	decimal	Expected Value		
GUID	string	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	string	Internal Name		
INVOICETEMPLATEGUID	string	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		

Name	Data Type	Label	Required	Documentation
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	int32	Lead Rating		
LEADSOURCEGUID	string	Lead Source GUID		
NAME	string	Name		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
PRICINGRULE	string	Pricing Rule		
PRICINGTYPE	string	Pricing Type		
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOTTASKGUID	string	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	string	Sales Person User Code		
SALESPERSONUSERGUID	string	Sales Person User GUID		
SALESPROCESSGUID	string	Sales Process GUID		
SALESSTATUSGUID	string	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	string	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	string	Your Reference		

3.1.72 CaseByNumber: Visma Severa Project by Number

Catalog: Severa

Schema: API

Label: Project by Number

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseNumber

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CaseByNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseNumber	int64	☐		

Columns of Table Function

The columns of the table function CaseByNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		
CASEOWNERUSERGUID	string	Project Owner User GUID		

Name	Data Type	Label	Required	Documentation
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	string	Ddescription		
EXPECTEDVALUE	decimal	Expected Value		
GUID	string	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	string	Internal Name		
INVOICETEMPLATEGUID	string	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	int32	Lead Rating		
LEADSOURCEGUID	string	Lead Source GUID		
NAME	string	Name		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
PRICINGRULE	string	Pricing Rule		
PRICINGTYPE	string	Pricing Type		

Name	Data Type	Label	Required	Documentation
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOTTASKGUID	string	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	string	Sales Person User Code		
SALESPERSONUSERGUID	string	Sales Person User GUID		
SALESPROCESSGUID	string	Sales Process GUID		
SALESSTATUSGUID	string	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	string	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	string	Your Reference		

3.1.73 CaseByTaskGUID: Visma Severa Project by Task GUID

Catalog: Severa

Schema: API

Label: Project by Task GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: taskGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CaseByTaskGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
taskGuid	string	☐		

Columns of Table Function

The columns of the table function CaseByTaskGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		
CASEOWNERUSERGUID	string	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	string	Ddescription		
EXPECTEDVALUE	decimal	Expected Value		

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	string	Internal Name		
INVOICETEMPLATEGUID	string	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	int32	Lead Rating		
LEADSOURCEGUID	string	Lead Source GUID		
NAME	string	Name		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
PRICINGRULE	string	Pricing Rule		
PRICINGTYPE	string	Pricing Type		
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOTTASKGUID	string	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	string	Sales Person User Code		
SALESPERSONUSERGUID	string	Sales Person User GUID		
SALESPROCESSGUID	string	Sales Process GUID		
SALESSTATUSGUID	string	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	string	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	string	Your Reference		

3.1.74 CaseMemberUsersByCaseGUID: Visma Severa Project Member Users by Project GUID

Catalog: Severa

Schema: API

Label: Project Member Users by Project GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CaseMemberUsersByCaseGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGuid	string	☐		

Columns of Table Function

The columns of the table function CaseMemberUsersByCaseGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		

Name	Data Type	Label	Required	Documentation
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.75 Cases: Visma Severa Projects

Catalog: Severa

Schema: API

Label: Projects

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- criteria_InternalName- criteria_OnlyActiveCases- criteria_OnlyCustomerCases- criteria_OnlyInactiveCases- criteria_OnlyInternalCases- criteria_SalesProcess- criteria_Tags

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Cases. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
criteria_InternalName	string	☐		
criteria_OnlyActiveCases	boolean	☐		
criteria_OnlyCustomerCases	boolean	☐		
criteria_OnlyInactiveCases	boolean	☐		
criteria_OnlyInternalCases	boolean	☐		
criteria_SalesProcess	string	☐		
criteria_Tags	string	☐		

Columns of Table Function

The columns of the table function Cases are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		

Name	Data Type	Label	Required	Documentation
CASEOWNERUSERGUID	string	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	string	Ddescription		
EXPECTEDVALUE	decimal	Expected Value		
GUID	string	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	string	Internal Name		
INVOICETEMPLATEGUID	string	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	int32	Lead Rating		
LEADSOURCEGUID	string	Lead Source GUID		
NAME	string	Name		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
PRICINGRULE	string	Pricing Rule		

Name	Data Type	Label	Required	Documentation
PRICINGTYPE	string	Pricing Type		
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOTTASKGUID	string	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	string	Sales Person User Code		
SALESPERSONUSERGUID	string	Sales Person User GUID		
SALESPROCESSGUID	string	Sales Process GUID		
SALESSTATUSGUID	string	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	string	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	string	Your Reference		

3.1.76 CasesByAccountGUID: Visma Severa Projects by Account GUID

Catalog: Severa

Schema: API

Label: Projects by Account GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- accountGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CasesByAccountGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountGuid	string	□		
businessUnitGUID	string	□		

Columns of Table Function

The columns of the table function CasesByAccountGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		
CASEOWNERUSERGUID	string	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		

Name	Data Type	Label	Required	Documentation
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	string	Ddescription		
EXPECTEDVALUE	decimal	Expected Value		
GUID	string	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	string	Internal Name		
INVOICETEMPLATEGUID	string	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	int32	Lead Rating		
LEADSOURCEGUID	string	Lead Source GUID		
NAME	string	Name		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
PRICINGRULE	string	Pricing Rule		
PRICINGTYPE	string	Pricing Type		
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOTTASKGUID	string	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	string	Sales Person User Code		
SALESPERSONUSERGUID	string	Sales Person User GUID		
SALESPROCESSGUID	string	Sales Process GUID		
SALESSTATUSGUID	string	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	string	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		

Name	Data Type	Label	Required	Documentation
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	string	Your Reference		

3.1.77 CasesChangedSince: Visma Severa Projects Changed since

Catalog: Severa

Schema: API

Label: Projects Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CasesChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
options	int32	☐		
startDateTime	datetime	☐		

Columns of Table Function

The columns of the table function CasesChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		

Name	Data Type	Label	Required	Documentation
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		
CASEOWNERUSERGUID	string	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	string	Ddescription		
EXPECTEDVALUE	decimal	Expected Value		
GUID	string	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	string	Internal Name		
INVOICETEMPLATEGUID	string	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		

Name	Data Type	Label	Required	Documentation
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	int32	Lead Rating		
LEADSOURCEGUID	string	Lead Source GUID		
NAME	string	Name		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
PRICINGRULE	string	Pricing Rule		
PRICINGTYPE	string	Pricing Type		
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOTTASKGUID	string	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	string	Sales Person User Code		
SALESPERSONUSERGUID	string	Sales Person User GUID		
SALESPROCESSGUID	string	Sales Process GUID		
SALESSTATUSGUID	string	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	string	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	string	Your Reference		

3.1.78 CaseStatusByGUID: Visma Severa Project Status by GUID

Catalog: Severa

Schema: API

Label: Project Status by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseStatusGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CaseStatusByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseStatusGUID	string	☐		

Columns of Table Function

The columns of the table function CaseStatusByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.79 CaseStatuses: Visma Severa Project Statuses

Catalog: Severa

Schema: API

Label: Project Statuses

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table CaseStatuses are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ICON	string	Icon		

Name	Data Type	Label	Required	Documentation
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.80 CaseTags: Visma Severa Project Tags

Catalog: Severa

Schema: API

Label: Project Tags

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table CaseTags are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTEXT	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyword		
WEIGHT	int32	Weight		

3.1.81 CompanyByGUID: Visma Severa Company by GUID

Catalog: Severa

Schema: API

Label: Company by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: companyGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CompanyByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
companyGuid	string	☐		

Columns of Table Function

The columns of the table function CompanyByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ANNUALREVENUE	int32	Annual Revenue		
ANNUALREVENUE2	int64			
EMAIL	string	Email		
EMPLOYEES	int32	Employees		
GUID	string	GUID		
HEADOFFICEADDRESSGUID	string	Head Office Address GUID		
HIERARCHYGUID	string	Hierarchy GUID		
INDUSTRYGUID	string	Industry GUID		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
TIMEZONEGUID	string	Timezone GUID		
VATNUMBER	string	VAT Number		
WEBSITE	string	Website		

3.1.82 ContactByGUID: Visma Severa Contact by GUID

Catalog: Severa

Schema: API

Label: Contact by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: contactGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContactByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
contactGUID	string	☐		

Columns of Table Function

The columns of the table function ContactByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ADDRESSGUID	string	Address GUID		
COMMUNICATIONMETHODS	string			
CONTACTROLEGUID	string	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	string	Ddescription		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	string	Job Title		
KEYWORDS	string			
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
SALUTATION	string	Salutation		
SATISFACTIONLEVEL	int32	Satisfaction Level		
TIMEZONEGUID	string	Timezone GUID		

3.1.83 ContactsByAccountGUID: Visma Severa Contacts by Account GUID

Catalog: Severa

Schema: API

Label: Contacts by Account GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accountGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContactsByAccountGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountGUID	string	☐		

Columns of Table Function

The columns of the table function ContactsByAccountGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ADDRESSGUID	string	Address GUID		
COMMUNICATIONMETHODS	string			
CONTACTROLEGUID	string	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	string	Ddescription		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	string	Job Title		
KEYWORDS	string			
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
SALUTATION	string	Salutation		
SATISFACTIONLEVEL	int32	Satisfaction Level		
TIMEZONEGUID	string	Timezone GUID		

3.1.84 ContactsByCommunicationMethodType: Visma Severa Contacts by Communication Method Type

Catalog: Severa

Schema: API

Label: Contacts by Communication Method Type

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: type- value

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContactsByCommunicationMethodType. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
type	string	☐		
value	string	☐		

Columns of Table Function

The columns of the table function ContactsByCommunicationMethodType are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ADDRESSGUID	string	Address GUID		
COMMUNICATIONMETHODS	string			
CONTACTROLEGUID	string	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	string	Ddescription		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	string	Job Title		

Name	Data Type	Label	Required	Documentation
KEYWORDS	string			
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
SALUTATION	string	Salutation		
SATISFACTIONLEVEL	int32	Satisfaction Level		
TIMEZONEGUID	string	Timezone GUID		

3.1.85 ContactsChangedSince: Visma Severa Contacts Changed since

Catalog: Severa

Schema: API

Label: Contacts Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accountGUID- startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContactsChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountGUID	string	☐		
options	int32	☐		
startDateTime	datetime	☐		

Columns of Table Function

The columns of the table function ContactsChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		

Name	Data Type	Label	Required	Documentation
ADDRESSGUID	string	Address GUID		
COMMUNICATIONMETHODS	string			
CONTACTROLEGUID	string	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	string	Ddescription		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	string	Job Title		
KEYWORDS	string			
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
SALUTATION	string	Salutation		
SATISFACTIONLEVEL	int32	Satisfaction Level		
TIMEZONEGUID	string	Timezone GUID		

3.1.86 CostCenterByGUID: Visma Severa Cost Center by GUID

Catalog: Severa

Schema: API

Label: Cost Center by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: costCenterGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CostCenterByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
costCenterGuid	string	☐		

Columns of Table Function

The columns of the table function CostCenterByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
IDENTIFIER	string	Identifier		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		

3.1.87 CostCenters: Visma Severa Cost Centers

Catalog: Severa

Schema: API

Label: Cost Centers

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table CostCenters are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
IDENTIFIER	string	Identifier		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		

3.1.88 CountryByGUID: Visma Severa Country by GUID

Catalog: Severa

Schema: API

Label: Country by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: countryGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CountryByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
countryGUID	string	☐		

Columns of Table Function

The columns of the table function CountryByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DEFAULTLANGUAGECODE	string	Default Language Code		
DEFAULTLANGUAGEGUID	string	Default Language GUID		
GUID	string	GUID		
ISEU	boolean	Is EU		
ISOCODEA2	string	ISO Code A2		
ISOCODEA3	string	ISO Code A3		
NAME	string	Name		
TIMEZONEGUID	string	Timezone GUID		
TRANSLATEDNAME	string	Translated Name		

3.1.89 CountryByISO: Visma Severa Country by ISO

Catalog: Severa

Schema: API

Label: Country by ISO

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: isoName

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CountryByISO. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
isoName	string	☐		

Columns of Table Function

The columns of the table function CountryByISO are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DEFAULTLANGUAGECODE	string	Default Language Code		
DEFAULTLANGUAGEGUID	string	Default Language GUID		
GUID	string	GUID		
ISEU	boolean	Is EU		
ISOCODEA2	string	ISO Code A2		
ISOCODEA3	string	ISO Code A3		
NAME	string	Name		
TIMEZONEGUID	string	Timezone GUID		
TRANSLATEDNAME	string	Translated Name		

3.1.90 CountryByName: Visma Severa Country by Name

Catalog: Severa

Schema: API

Label: Country by Name

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: countryName

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CountryByName. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
countryName	string	☐		

Columns of Table Function

The columns of the table function CountryByName are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DEFAULTLANGUAGECODE	string	Default Language Code		
DEFAULTLANGUAGEGUID	string	Default Language GUID		
GUID	string	GUID		
ISEU	boolean	Is EU		
ISOCODEA2	string	ISO Code A2		
ISOCODEA3	string	ISO Code A3		
NAME	string	Name		
TIMEZONEGUID	string	Timezone GUID		
TRANSLATEDNAME	string	Translated Name		

3.1.91 CreateInvoices: Visma Severa Create Invoices

Catalog: Severa

Schema: API

Label: Create Invoices

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: invoiceInformation_Date- invoiceInformation_DueDate- invoiceInformation_InvoiceStatus- invoiceInformation_ItemGUIDs- invoiceInformation_LanguageGUID- invoiceInformation_Notes-

invoiceInformation_OurReference- invoiceInformation_OverDueInterest-
invoiceInformation_PaymentTerm- invoiceInformation_ReceiverAddressline-
invoiceInformation_ReceiverCity- invoiceInformation_ReceiverContactEmail-
invoiceInformation_ReceiverContactName- invoiceInformation_ReceiverCountry-
invoiceInformation_ReceiverCustomerName- invoiceInformation_ReceiverPostalCode-
invoiceInformation_ReceiverState- invoiceInformation_ReceiverVatNumber-
invoiceInformation_SenderAddressline- invoiceInformation_SenderCity-
invoiceInformation_SenderContactName- invoiceInformation_SenderCountry-
invoiceInformation_SenderName- invoiceInformation_SenderPostalCode-
invoiceInformation_SenderState- invoiceInformation_SenderVatNumber-
invoiceInformation_YourReference

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CreateInvoices. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
invoiceInformation_Date	datetime	☐		
invoiceInformation_DueDate	datetime	☐		
invoiceInformation_InvoiceStatus	string	☐		
invoiceInformation_ItemGUIDs	string	☐		
invoiceInformation_LanguageGUID	string	☐		
invoiceInformation_Notes	string	☐		
invoiceInformation_OurReference	string	☐		
invoiceInformation_OverDueInterest	decimal	☐		
invoiceInformation_PaymentTerm	int32	☐		
invoiceInformation_ReceiverAddressline	string	☐		
invoiceInformation_ReceiverCity	string	☐		
invoiceInformation_ReceiverContactEmail	string	☐		
invoiceInformation_ReceiverContactName	string	☐		
invoiceInformation_ReceiverCountry	string	☐		
invoiceInformation_ReceiverCustomerName	string	☐		

Name	Data Type	Required	Default Value	Documentation
invoiceInformation_ReceiverPostalCode	string	☐		
invoiceInformation_ReceiverState	string	☐		
invoiceInformation_ReceiverVatNumber	string	☐		
invoiceInformation_SenderAddressline	string	☐		
invoiceInformation_SenderCity	string	☐		
invoiceInformation_SenderContactName	string	☐		
invoiceInformation_SenderCountry	string	☐		
invoiceInformation_SenderName	string	☐		
invoiceInformation_SenderPostalCode	string	☐		
invoiceInformation_SenderState	string	☐		
invoiceInformation_SenderVatNumber	string	☐		
invoiceInformation_YourReference	string	☐		

Columns of Table Function

The columns of the table function CreateInvoices are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANISATIONNAME	string			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			

Name	Data Type	Label	Required	Documentation
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	string			
CULTURE	string	Culture		
CURRENCYGUID	string	Currency GUID		
CURRENCYNAME	string	Currency Name		
CURRENCYRATE	decimal	Currency Rate		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	string	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	string	GUID		
INVOICEALLOWEDCASEGUIDS	string			
INVOICECONFIG_BILLEXPENSE	string			
INVOICECONFIG_BILLHOUR	string			
INVOICECONFIG_BILLPRODUCT	string			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURPRIMARYGROUPBY	string			
INVOICECONFIG_HOURSECONDARYGROUPBY	string			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	string			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	string			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	int32			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETYPETEXT	string			
INVOICEDETAILS_ORDERIDENTIFIER	string			
INVOICEDETAILS_ORIGINCODE	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	decimal			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	string			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	string			
INVOICEDETAILS_VATBASEAMOUNT	decimal			
INVOICEDETAILS_VATRATEAMOUNT	decimal			

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_VATRATEPERCENT	int32			
INVOICENUMBERINGBUSINESSUNITGUID	string	Invoice Numbering Business Unit GUID		
INVOICEROWS	string			
INVOICESTATUS	string	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
MAINCASEGUID	string	Main Case GUID		
NOTES	string	Notes		
NOTESAFTERINVOICEROWS	string	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	string	Notes before Invoice Lines		
NUMBER	int32	Number		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	string	Payment Status		
PAYMENTTERM	int32	Payment Term		
PONUMBER	string	PO Number		
RECEIVERADDRESSLINE	string	Receiver Addressline		
RECEIVERCITY	string	Receiver City		
RECEIVERCONTACTEMAIL	string	Receiver Contact Email		
RECEIVERCONTACTNAME	string	Receiver Contact Name		
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			

Name	Data Type	Label	Required	Documentation
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREETEXT	string			
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONENUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLEREINVOICEADDRESS	string			
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		

Name	Data Type	Label	Required	Documentation
YOURREFERENCE	string	Your Reference		

3.1.92 Currencies: Visma Severa Currencies

Catalog: Severa

Schema: API

Label: Currencies

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table Currencies are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISOCODE	string	ISO Code		
RATE	decimal	Rate		
SYMBOL	string	Symbol		

3.1.93 CurrencyByGUID: Visma Severa Currency by GUID

Catalog: Severa

Schema: API

Label: Currency by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: currencyGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CurrencyByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
currencyGUID	string	☐		

Columns of Table Function

The columns of the table function CurrencyByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISOCODE	string	ISO Code		
RATE	decimal	Rate		
SYMBOL	string	Symbol		

3.1.94 CurrencyByIsoCode: Visma Severa Currency by ISO Code

Catalog: Severa

Schema: API

Label: Currency by ISO Code

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: currencyIsoCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CurrencyByIsoCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
currencyIsoCode	string	☐		

Columns of Table Function

The columns of the table function CurrencyByIsoCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISOCODE	string	ISO Code		
RATE	decimal	Rate		
SYMBOL	string	Symbol		

3.1.95 CustomerByGUID: Visma Severa Customer by GUID

Catalog: Severa

Schema: API

Label: Customer by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accountGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomerByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountGuid	string	☐		

Columns of Table Function

The columns of the table function CustomerByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
ANNUALREVENUE	int32	Annual Revenue		
ANNUALREVENUE2	int64			
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
EMAIL	string	Email		
EMPLOYEES	int32	Employees		
GUID	string	GUID		
HEADOFFICEADDRESSGUID	string	Head Office Address GUID		
INDUSTRYGUID	string	Industry GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		
VATNUMBER	string	VAT Number		
WEBSITE	string	Website		

3.1.96 Customers: Visma Severa Customers

Catalog: Severa

Schema: API

Label: Customers

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: options- startFromIndex- count- criteria_AccountGroupGuids

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Customers. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
count	int32	☐		
criteria_AccountGroupGuids	string	☐		
options	int32	☐		
startFromIndex	int32	☐		

Columns of Table Function

The columns of the table function Customers are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
ANNUALREVENUE	int32	Annual Revenue		
ANNUALREVENUE2	int64			
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
EMAIL	string	Email		

Name	Data Type	Label	Required	Documentation
EMPLOYEES	int32	Employees		
GUID	string	GUID		
HEADOFFICEADDRESSGUID	string	Head Office Address GUID		
INDUSTRYGUID	string	Industry GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		
VATNUMBER	string	VAT Number		
WEBSITE	string	Website		

3.1.97 CustomersChangedSince: Visma Severa Customers Changed since

Catalog: Severa

Schema: API

Label: Customers Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomersChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
options	int32	☐		
startDateTime	datetime	☐		

Columns of Table Function

The columns of the table function CustomersChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
ANNUALREVENUE	int32	Annual Revenue		
ANNUALREVENUE2	int64			
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
EMAIL	string	Email		
EMPLOYEES	int32	Employees		
GUID	string	GUID		
HEADOFFICEADDRESSGUID	string	Head Office Address GUID		
INDUSTRYGUID	string	Industry GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		

Name	Data Type	Label	Required	Documentation
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		
VATNUMBER	string	VAT Number		
WEBSITE	string	Website		

3.1.98 Database: Visma Severa Database

Catalog: Severa

Schema: API

Label: Database

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table Database are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DATABASERESULT	string			

3.1.99 DeleteAccessRightsProfileByGUID: Visma Severa Delete Access Rights Profile by GUID

Catalog: Severa

Schema: API

Label: Delete Access Rights Profile by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: guid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteAccessRightsProfileByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the

execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
guid	string	☐		

Columns of Table Function

The columns of the table function DeleteAccessRightsProfileByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DUMMY	string	Dummy	☑	Dummy column

3.1.100 DeleteActivity: Visma Severa Delete Activity

Catalog: Severa

Schema: API

Label: Delete Activity

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteActivity. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityGUID	string	☐		

Columns of Table Function

The columns of the table function DeleteActivity are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DELETEACTIVITYRESULT	boolean			

3.1.101 DeleteActivityType: Visma Severa Delete Activity Type

Catalog: Severa

Schema: API

Label: Delete Activity Type

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityTypeGUID- replacementActivityTypeGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteActivityType. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityTypeGUID	string	☐		
replacementActivityTypeGUID	string	☐		

Columns of Table Function

The columns of the table function DeleteActivityType are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DELETEACTIVITYTYPERESULT	boolean			

3.1.102 DeleteCase: Visma Severa Delete Project

Catalog: Severa

Schema: API

Label: Delete Project

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteCase. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGuid	string	☐		

Columns of Table Function

The columns of the table function DeleteCase are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DELETECASERESULT	boolean			

3.1.103 DeleteCaseStatus: Visma Severa Delete Project Status

Catalog: Severa

Schema: API

Label: Delete Project Status

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseStatusGUID- replacementCaseStatusGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteCaseStatus. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseStatusGUID	string	☐		
replacementCaseStatusGUID	string	☐		

Columns of Table Function

The columns of the table function DeleteCaseStatus are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DELETECASESTATUSRESULT	boolean			

3.1.104 DeleteContact: Visma Severa Delete Contact

Catalog: Severa

Schema: API

Label: Delete Contact

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: contactGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteContact. A value must be provided at all times for required parameters, but optional

parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
contactGUID	string	☐		

Columns of Table Function

The columns of the table function DeleteContact are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DELETECONTACTRESULT	boolean			

3.1.105 DeleteEmployment: Visma Severa Delete Employment

Catalog: Severa

Schema: API

Label: Delete Employment

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: employmentGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteEmployment. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
employmentGUID	string	□		

Columns of Table Function

The columns of the table function DeleteEmployment are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DELETEEMPLOYMENTRESULT	boolean			

3.1.106 DeleteHourEntry: Visma Severa Delete Hour Entry

Catalog: Severa

Schema: API

Label: Delete Hour Entry

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: hourEntryGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteHourEntry. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
hourEntryGUID	string	□		

Columns of Table Function

The columns of the table function DeleteHourEntry are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DELETEHOURENTRYRESULT	boolean			

3.1.107 DeleteItem: Visma Severa Delete Item

Catalog: Severa

Schema: API

Label: Delete Item

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: itemGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteItem. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
itemGUID	string	☐		

Columns of Table Function

The columns of the table function DeleteItem are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DELETEITEMRESULT	boolean			

3.1.108 DeletePhase: Visma Severa Delete Phase

Catalog: Severa

Schema: API

Label: Delete Phase

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: phaseGUID- phaseGUIDToTransferItemsTo

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeletePhase. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
phaseGUID	string	☐		
phaseGUIDToTransferItemsTo	string	☐		

Columns of Table Function

The columns of the table function DeletePhase are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DELETEPHASERESULT	boolean			

3.1.109 DeleteProduct: Visma Severa Delete Product

Catalog: Severa

Schema: API

Label: Delete Product

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: productGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteProduct. A value must be provided at all times for required parameters, but optional

parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
productGUID	string	☐		

Columns of Table Function

The columns of the table function DeleteProduct are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DELETEPRODUCTRESULT	boolean			

3.1.110 DeleteProductCategory: Visma Severa Delete Product Category

Catalog: Severa

Schema: API

Label: Delete Product Category

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: productCategoryGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteProductCategory. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
productCategoryGUID	string	☐		

Columns of Table Function

The columns of the table function DeleteProductCategory are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DELETEPRODUCTCATEGORYRESULT	boolean			

3.1.111 DeleteResource: Visma Severa Delete Resource

Catalog: Severa

Schema: API

Label: Delete Resource

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: resourceGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteResource. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
resourceGUID	string	☐		

Columns of Table Function

The columns of the table function DeleteResource are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DELETERESOURCERESULT	boolean			

3.1.112 DeleteResourceAllocation: Visma Severa Delete Resource Allocation

Catalog: Severa

Schema: API

Label: Delete Resource Allocation

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: resourceAllocationGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteResourceAllocation. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
resourceAllocationGUID	string	☐		

Columns of Table Function

The columns of the table function DeleteResourceAllocation are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DELETERESOURCEALLOCATIONRESULT	boolean			

3.1.113 DeleteTag: Visma Severa Delete Tag

Catalog: Severa

Schema: API

Label: Delete Tag

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: tagGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteTag. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
tagGUID	string	☐		

Columns of Table Function

The columns of the table function DeleteTag are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DELETETAGRESULT	boolean			

3.1.114 DeleteTravelExpense: Visma Severa Delete Travel Expense

Catalog: Severa

Schema: API

Label: Delete Travel Expense

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: travelExpenseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteTravelExpense. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
travelExpenseGUID	string	☐		

Columns of Table Function

The columns of the table function DeleteTravelExpense are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DELETETRAVELEXPENSERESULT	boolean			

3.1.115 DisableUser: Visma Severa Disable User

Catalog: Severa

Schema: API

Label: Disable User

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DisableUser. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userGUID	string	☐		

Columns of Table Function

The columns of the table function DisableUser are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DISABLEUSERRESULT	boolean			

3.1.116 EmploymentByGUID: Visma Severa Employment by GUID

Catalog: Severa

Schema: API

Label: Employment by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: employmentGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function EmploymentByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
employmentGUID	string	☐		

Columns of Table Function

The columns of the table function EmploymentByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
COMPANYBUSINESSUNITCODE	string	Company Business Unit Code		
COMPANYBUSINESSUNITGUID	string	Company Business Unit GUID		
COMPANYBUSINESSUNITNAME	string	Company Business Unit Name		
DAILYHOURS	decimal	Daily Hours		
ENDDATE	datetime	End Date		
GUID	string	GUID		
HOURCOST	decimal	Hour Cost		
ISOVERTIMEALLOWED	boolean	Is Overtime Allowed		
STARTDATE	datetime	Start Date		
TITLE	string	Title		

3.1.117 EmploymentsByUserGUID: Visma Severa Employments by User GUID

Catalog: Severa

Schema: API

Label: Employments by User GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function EmploymentsByUserGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userGUID	string	☐		

Columns of Table Function

The columns of the table function EmploymentsByUserGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
COMPANYBUSINESSUNITCODE	string	Company Business Unit Code		
COMPANYBUSINESSUNITGUID	string	Company Business Unit GUID		
COMPANYBUSINESSUNITNAME	string	Company Business Unit Name		
DAILYHOURS	decimal	Daily Hours		
ENDDATE	datetime	End Date		
GUID	string	GUID		
HOURCOST	decimal	Hour Cost		
ISOVERTIMEALLOWED	boolean	Is Overtime Allowed		
STARTDATE	datetime	Start Date		
TITLE	string	Title		

3.1.118 EnableUser: Visma Severa Enable User

Catalog: Severa

Schema: API

Label: Enable User

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function EnableUser. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userGUID	string	□		

Columns of Table Function

The columns of the table function EnableUser are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ENABLEUSERRESULT	boolean			

3.1.119 EntitySchemaVersion: Visma Severa Entity Schema Version

Catalog: Severa

Schema: API

Label: Entity Schema Version

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table EntitySchemaVersion are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GETENTITYSCHEMAVERSIONRESULT	string			

3.1.120 ExtranetCaseByCaseGUID: Visma Severa Extranet Project by Project GUID

Catalog: Severa

Schema: API

Label: Extranet Project by Project GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ExtranetCaseByCaseGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default

to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGuid	string	☐		

Columns of Table Function

The columns of the table function ExtranetCaseByCaseGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTACTS	string			
GUID	string	GUID		
RIGHTS_EDITCASESTATUS	boolean			
RIGHTS_SHOWBILLING	boolean			
RIGHTS_SHOWCASESTATUS	boolean			
RIGHTS_SHOWFILESANDLINKS	boolean			
RIGHTS_SHOWGANTTCHART	boolean			
RIGHTS_SHOWPHASETREE	boolean			
RIGHTS_SHOWREADYTOBILL	boolean			
RIGHTS_SHOWWORKHOURESTIMATES	boolean			
RIGHTS_SHOWWORKHOURS	boolean			
RIGHTS_SHOWWORKHOURSTARTANDENDTIME	boolean			

3.1.121 ExtranetContactCaseRightsByContactGUID: Visma Severa Extranet Contact Project Rights by Contact GUID

Catalog: Severa

Schema: API

Label: Extranet Contact Project Rights by Contact GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: contactGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ExtranetContactCaseRightsByContactGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
contactGuid	string	☐		

Columns of Table Function

The columns of the table function ExtranetContactCaseRightsByContactGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CASEGUID	string	Project GUID		
CASERIGHTS_EDITCASESTATUS	boolean			
CASERIGHTS_SHOWBILLING	boolean			
CASERIGHTS_SHOWCASESTATUS	boolean			
CASERIGHTS_SHOWFILESANDLINKS	boolean			
CASERIGHTS_SHOWGANTTCHART	boolean			
CASERIGHTS_SHOWPHASETREE	boolean			
CASERIGHTS_SHOWREADYTOBILL	boolean			
CASERIGHTS_SHOWWORKHOURESTIMATES	boolean			
CASERIGHTS_SHOWWORKHOURS	boolean			
CASERIGHTS_SHOWWORKHOURSTARTANDENDTIME	boolean			
CONTACTRIGHTS_APPROVAL	boolean			
CONTACTRIGHTS_COMMENTS	boolean			
CONTACTRIGHTS_FILEUPLOAD	boolean			

3.1.122 FileByGUID: Visma Severa File by GUID

Catalog: Severa

Schema: API

Label: File by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: guid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function FileByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
guid	string	☐		

Columns of Table Function

The columns of the table function FileByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTENTTYPE	string	Content Type		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
NAME	string	Name		
SIZE	int32	Size		

3.1.123 FileBytesByGUID: Visma Severa File Contents by GUID

Catalog: Severa

Schema: API

Label: File Contents by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: guid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function FileBytesByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
guid	string	☐		

Columns of Table Function

The columns of the table function FileBytesByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GETFILEBYTESBYGUIDRESULT	byte[]			

3.1.124 Files: Visma Severa Files

Catalog: Severa

Schema: API

Label: Files

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table Files are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTENTTYPE	string	Content Type		
DESCRIPTION	string	Ddescription		

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
NAME	string	Name		
SIZE	int32	Size		

3.1.125 FinvoicesByStatus: Visma Severa Financial Invoices by Status

Catalog: Severa

Schema: API

Label: Financial Invoices by Status

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- status- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function FinvoicesByStatus. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
options	int32	☐		
status	string	☐		

Columns of Table Function

The columns of the table function FinvoicesByStatus are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
FINVOICE	string	Financial Invoice		
GUID	string	GUID		

3.1.126 FinvoicesByStatusAndDate: Visma Severa Financial Invoices by Status and Date

Catalog: Severa

Schema: API

Label: Financial Invoices by Status and Date

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: finvoiceVersion- businessUnitGUID- invoiceStatus- invoiceDateStart- invoiceDateEnd- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function FinvoicesByStatusAndDate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
finvoiceVersion	string	☐		
invoiceDateEnd	datetime	☐		
invoiceDateStart	datetime	☐		
invoiceStatus	string	☐		
options	int32	☐		

Columns of Table Function

The columns of the table function FinvoicesByStatusAndDate are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
FINVOICE	string	Financial Invoice		
GUID	string	GUID		

3.1.127 FirstAccountsBillingAddress: Visma Severa First Account Billing Addresses

Catalog: Severa

Schema: API

Label: First Account Billing Addresses

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accountGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function FirstAccountsBillingAddress. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountGUID	string	☐		

Columns of Table Function

The columns of the table function FirstAccountsBillingAddress are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
ADDRESSLINE2	string	Addressline 2		
ADDRESSLINE3	string	Addressline 3		
CITY	string	City		
COMPANYGUID	string	Company GUID		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYNAME	string	Country Name		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
FAX	string	Fax		
GUID	string	GUID		

Name	Data Type	Label	Required	Documentation
ISBILLINGADDRESS	boolean	Is Billing Address		
ISPOSTALADDRESS	boolean	Is Postal Address		
ISVISITADDRESS	boolean	Is Visit Address		
PHONE	string	Phone		
POSTALCODE	string	Postal Code		

3.1.128 HeadOfficeAddressByAccountGUID: Visma Severa Head Office Address by Account GUID

Catalog: Severa

Schema: API

Label: Head Office Address by Account GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accountGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HeadOfficeAddressByAccountGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountGUID	string	☐		

Columns of Table Function

The columns of the table function HeadOfficeAddressByAccountGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
ADDRESSLINE2	string	Addressline 2		
ADDRESSLINE3	string	Addressline 3		

Name	Data Type	Label	Required	Documentation
CITY	string	City		
COMPANYGUID	string	Company GUID		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYNAME	string	Country Name		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
FAX	string	Fax		
GUID	string	GUID		
ISBILLINGADDRESS	boolean	Is Billing Address		
ISPOSTALADDRESS	boolean	Is Postal Address		
ISVISITADDRESS	boolean	Is Visit Address		
PHONE	string	Phone		
POSTALCODE	string	Postal Code		

3.1.129 HourEntriesByCaseGUID: Visma Severa Hour Entries by Project GUID

Catalog: Severa

Schema: API

Label: Hour Entries by Project GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGuid- startDate- endDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HourEntriesByCaseGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGuid	string	☐		
endDate	datetime	☐		

Name	Data Type	Required	Default Value	Documentation
startDate	datetime	☐		

Columns of Table Function

The columns of the table function HourEntriesByCaseGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		

Name	Data Type	Label	Required	Documentation
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.130 HourEntriesByDate: Visma Severa Hour Entries by Date

Catalog: Severa

Schema: API

Label: Hour Entries by Date

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- startdate- enddate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HourEntriesByDate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
enddate	datetime	☐		
startdate	datetime	☐		

Columns of Table Function

The columns of the table function HourEntriesByDate are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		

Name	Data Type	Label	Required	Documentation
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		

Name	Data Type	Label	Required	Documentation
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.131 HourEntriesByDateAndUser: Visma Severa Hour Entries by Date and User

Catalog: Severa

Schema: API

Label: Hour Entries by Date and User

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- startdate- enddate- userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HourEntriesByDateAndUser. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
enddate	datetime	☐		
startdate	datetime	☐		
userGUID	string	☐		

Columns of Table Function

The columns of the table function HourEntriesByDateAndUser are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		

Name	Data Type	Label	Required	Documentation
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		

Name	Data Type	Label	Required	Documentation
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.132 HourEntriesByDateAndUserGUID: Visma Severa Hour Entry by Date and User GUID

Catalog: Severa

Schema: API

Label: Hour Entry by Date and User GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- startdate- enddate- userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HourEntriesByDateAndUserGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
enddate	datetime	☐		
startdate	datetime	☐		
userGUID	string	☐		

Columns of Table Function

The columns of the table function HourEntriesByDateAndUserGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		

Name	Data Type	Label	Required	Documentation
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.133 HourEntriesChangedSince: Visma Severa Hour Entries Changed since

Catalog: Severa

Schema: API

Label: Hour Entries Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: startDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HourEntriesChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
startDate	datetime	☐		

Columns of Table Function

The columns of the table function HourEntriesChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		

Name	Data Type	Label	Required	Documentation
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		

Name	Data Type	Label	Required	Documentation
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.134 HourEntryByGUID: Visma Severa Hour Entry by GUID

Catalog: Severa

Schema: API

Label: Hour Entry by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: hourEntryGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HourEntryByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
hourEntryGUID	string	☐		

Columns of Table Function

The columns of the table function HourEntryByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User		

Name	Data Type	Label	Required	Documentation
		Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.135 HourEntryPhasesByUserGUID: Visma Severa Hour Entry Phases by User GUID

Catalog: Severa

Schema: API

Label: Hour Entry Phases by User GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HourEntryPhasesByUserGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userGUID	string	☐		

Columns of Table Function

The columns of the table function HourEntryPhasesByUserGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
CASEGUID	string	Project GUID		
CHILDPHASES	string	Child Phases		
GUID	string	GUID		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
PARENTPHASEGUID	string	Parent Phase GUID		
WORKTYPEGUID	string	Work Type GUID		

3.1.136 Industries: Visma Severa Industries

Catalog: Severa

Schema: API

Label: Industries

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table Industries are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.137 InvoiceAttachmentFile: Visma Severa Invoice Attachment File

Catalog: Severa

Schema: API

Label: Invoice Attachment File

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: attachmentGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceAttachmentFile. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
attachmentGuid	string	☐		

Columns of Table Function

The columns of the table function InvoiceAttachmentFile are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GETINVOICEATTACHMENTFILERESULT	byte[]			

3.1.138 InvoiceAttachmentFileGUIDs: Visma Severa Invoice Attachment File GUIDs

Catalog: Severa

Schema: API

Label: Invoice Attachment File GUIDs

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: invoiceGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceAttachmentFileGUIDs. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
invoiceGuid	string	☐		

Columns of Table Function

The columns of the table function InvoiceAttachmentFileGUIDs are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
MIMETYPE	string	MIME-type		
NAME	string	Name		
PRINTORDER	int32	Print Order		

3.1.139 InvoiceByGUID: Visma Severa Invoice by GUID

Catalog: Severa

Schema: API

Label: Invoice by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: invoiceGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
invoiceGUID	string	□		

Columns of Table Function

The columns of the table function InvoiceByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANISATIONNAME	string			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	string			

Name	Data Type	Label	Required	Documentation
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	string			
CULTURE	string	Culture		
CURRENCYGUID	string	Currency GUID		
CURRENCYNAME	string	Currency Name		
CURRENCYRATE	decimal	Currency Rate		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	string	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	string	GUID		
INVOICEALLOWEDCASEGUIDS	string			
INVOICECONFIG_BILLEXPENSE	string			
INVOICECONFIG_BILLHOUR	string			
INVOICECONFIG_BILLPRODUCT	string			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURPRIMARYGROUPBY	string			
INVOICECONFIG_HOURSECONDARYGROUPBY	string			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	string			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	string			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	int32			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETYPETEXT	string			
INVOICEDETAILS_ORDERIDENTIFIER	string			
INVOICEDETAILS_ORIGINCODE	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	decimal			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	string			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	string			
INVOICEDETAILS_VATBASEAMOUNT	decimal			
INVOICEDETAILS_VATRATEAMOUNT	decimal			
INVOICEDETAILS_VATRATEPERCENT	int32			
INVOICENUMBERINGBUSINESSUNITGUID	string	Invoice Numbering Business Unit GUID		
INVOICEROWS	string			

Name	Data Type	Label	Required	Documentation
INVOICESTATUS	string	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
MAINCASEGUID	string	Main Case GUID		
NOTES	string	Notes		
NOTESAFTERINVOICEROWS	string	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	string	Notes before Invoice Lines		
NUMBER	int32	Number		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	string	Payment Status		
PAYMENTTERM	int32	Payment Term		
PONUMBER	string	PO Number		
RECEIVERADDRESSLINE	string	Receiver Addressline		
RECEIVERCITY	string	Receiver City		
RECEIVERCONTACTEMAIL	string	Receiver Contact Email		
RECEIVERCONTACTNAME	string	Receiver Contact Name		
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREETEXT	string			

Name	Data Type	Label	Required	Documentation
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONENUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLEREINVOICEADDRESS	string			
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		
YOURREFERENCE	string	Your Reference		

3.1.140 InvoiceByNumber: Visma Severa Invoice by Number

Catalog: Severa

Schema: API

Label: Invoice by Number

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: invoiceNumber- businessUnitGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceByNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
invoiceNumber	int32	☐		

Columns of Table Function

The columns of the table function InvoiceByNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			

Name	Data Type	Label	Required	Documentation
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANISATIONNAME	string			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	string			
CULTURE	string	Culture		
CURRENCYGUID	string	Currency GUID		
CURRENCYNAME	string	Currency Name		
CURRENCYRATE	decimal	Currency Rate		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	string	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	string	GUID		
INVOICEALLOWEDCASEGUIDS	string			
INVOICECONFIG_BILLEXPENSE	string			
INVOICECONFIG_BILLHOUR	string			
INVOICECONFIG_BILLPRODUCT	string			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURPRIMARYGROUPBY	string			
INVOICECONFIG_HOURSECONDARYGROUPBY	string			
INVOICECONFIG_ISTAXFREE	boolean			

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	string			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	string			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	int32			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETYPETEXT	string			
INVOICEDETAILS_ORDERIDENTIFIER	string			
INVOICEDETAILS_ORIGINCODE	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	decimal			

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_PAYMENTTERMSFREETEXT	string			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	string			
INVOICEDETAILS_VATBASEAMOUNT	decimal			
INVOICEDETAILS_VATRATEAMOUNT	decimal			
INVOICEDETAILS_VATRATEPERCENT	int32			
INVOICENUMBERINGBUSINESSUNITGUID	string	Invoice Numbering Business Unit GUID		
INVOICEROWS	string			
INVOICESTATUS	string	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
MAINCASEGUID	string	Main Case GUID		
NOTES	string	Notes		
NOTESAFTERINVOICEROWS	string	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	string	Notes before Invoice Lines		
NUMBER	int32	Number		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	string	Payment Status		
PAYMENTTERM	int32	Payment Term		
PONUMBER	string	PO Number		
RECEIVERADDRESSLINE	string	Receiver Addressline		
RECEIVERCITY	string	Receiver City		
RECEIVERCONTACTEMAIL	string	Receiver Contact Email		
RECEIVERCONTACTNAME	string	Receiver Contact Name		
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		

Name	Data Type	Label	Required	Documentation
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREETEXT	string			
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONENUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLEREINVOICEADDRESS	string			
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		

Name	Data Type	Label	Required	Documentation
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		
YOURREFERENCE	string	Your Reference		

3.1.141 InvoicedHourEntriesByCaseGUID: Visma Severa Invoiced Hour Entries by Project GUID

Catalog: Severa

Schema: API

Label: Invoiced Hour Entries by Project GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicedHourEntriesByCaseGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGuid	string	☐		

Columns of Table Function

The columns of the table function InvoicedHourEntriesByCaseGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		

Name	Data Type	Label	Required	Documentation
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		

Name	Data Type	Label	Required	Documentation
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.142 InvoiceOriginDetailsByInvoiceGUID: Visma Severa Invoice Origin Details by Invoice GUID

Catalog: Severa

Schema: API

Label: Invoice Origin Details by Invoice GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: invoiceGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceOriginDetailsByInvoiceGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
invoiceGUID	string	☐		

Columns of Table Function

The columns of the table function InvoiceOriginDetailsByInvoiceGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CASES	string	Projects		
INVOICEROWORIGINS	string			

3.1.143 InvoicePdf: Visma Severa Invoice PDF

Catalog: Severa

Schema: API

Label: Invoice PDF

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: invoiceNumber- invoiceNumberingBusinessUnitGUID- getAttachmentOnly

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicePdf. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
getAttachmentOnly	string	☐		
invoiceNumber	int32	☐		
invoiceNumberingBusinessUnitGUID	string	☐		

Columns of Table Function

The columns of the table function InvoicePdf are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GETINVOICEPDFRESULT	byte[]			

3.1.144 InvoicePdfByInvoiceGUID: Visma Severa Invoice PDF by Invoice GUID

Catalog: Severa

Schema: API

Label: Invoice PDF by Invoice GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: invoiceGuid- getAttachmentOnly

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicePdfByInvoiceGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
getAttachmentOnly	string	☐		
invoiceGuid	string	☐		

Columns of Table Function

The columns of the table function InvoicePdfByInvoiceGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GETINVOICEPDFBYINVOICEGUIDRESULT	byte[]			

3.1.145 InvoicesByDate: Visma Severa Invoices by Date

Catalog: Severa

Schema: API

Label: Invoices by Date

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- startDate- endDate- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicesByDate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
endDate	datetime	☐		
options	int32	☐		
startDate	datetime	☐		

Columns of Table Function

The columns of the table function InvoicesByDate are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		

Name	Data Type	Label	Required	Documentation
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANISATIONNAME	string			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	string			
CULTURE	string	Culture		
CURRENCYGUID	string	Currency GUID		
CURRENCYNAME	string	Currency Name		
CURRENCYRATE	decimal	Currency Rate		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	string	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	string	GUID		
INVOICEALLOWEDCASEGUIDS	string			
INVOICECONFIG_BILLEXPENSE	string			

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_BILLHOUR	string			
INVOICECONFIG_BILLPRODUCT	string			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURPRIMARYGROUPBY	string			
INVOICECONFIG_HOURSECONDARYGROUPBY	string			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	string			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	string			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	int32			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETYPETEXT	string			
INVOICEDETAILS_ORDERIDENTIFIER	string			
INVOICEDETAILS_ORIGINCODE	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	decimal			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	string			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	string			
INVOICEDETAILS_VATBASEAMOUNT	decimal			
INVOICEDETAILS_VATRATEAMOUNT	decimal			
INVOICEDETAILS_VATRATEPERCENT	int32			
INVOICENUMBERINGBUSINESSUNITGUID	string	Invoice Numbering Business Unit GUID		
INVOICEROWS	string			
INVOICESTATUS	string	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
MAINCASEGUID	string	Main Case GUID		
NOTES	string	Notes		
NOTESAFTERINVOICEROWS	string	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	string	Notes before Invoice Lines		
NUMBER	int32	Number		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	string	Payment Status		
PAYMENTTERM	int32	Payment Term		
PONUMBER	string	PO Number		
RECEIVERADDRESSLINE	string	Receiver Addressline		

Name	Data Type	Label	Required	Documentation
RECEIVERCITY	string	Receiver City		
RECEIVERCONTACTEMAIL	string	Receiver Contact Email		
RECEIVERCONTACTNAME	string	Receiver Contact Name		
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREETEXT	string			
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONENUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLEREINVOICEADDRESS	string			
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			

Name	Data Type	Label	Required	Documentation
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		
YOURREFERENCE	string	Your Reference		

3.1.146 InvoicesByDateAndStatus: Visma Severa Invoices by Date and Status

Catalog: Severa

Schema: API

Label: Invoices by Date and Status

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- startDate- endDate- status- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicesByDateAndStatus. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
endDate	datetime	☐		
options	int32	☐		
startDate	datetime	☐		
status	string	☐		

Columns of Table Function

The columns of the table function InvoicesByDateAndStatus are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANISATIONNAME	string			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	string			

Name	Data Type	Label	Required	Documentation
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	string			
CULTURE	string	Culture		
CURRENCYGUID	string	Currency GUID		
CURRENCYNAME	string	Currency Name		
CURRENCYRATE	decimal	Currency Rate		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	string	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	string	GUID		
INVOICEALLOWEDCASEGUIDS	string			
INVOICECONFIG_BILLEXPENSE	string			
INVOICECONFIG_BILLHOUR	string			
INVOICECONFIG_BILLPRODUCT	string			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURPRIMARYGROUPBY	string			
INVOICECONFIG_HOURSECONDARYGROUPBY	string			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	string			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	string			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	int32			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETYPETEXT	string			
INVOICEDETAILS_ORDERIDENTIFIER	string			
INVOICEDETAILS_ORIGINCODE	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	decimal			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	string			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	string			
INVOICEDETAILS_VATBASEAMOUNT	decimal			
INVOICEDETAILS_VATRATEAMOUNT	decimal			
INVOICEDETAILS_VATRATEPERCENT	int32			
INVOICENUMBERINGBUSINESSUNITGUID	string	Invoice Numbering Business Unit GUID		
INVOICEROWS	string			
INVOICESTATUS	string	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
MAINCASEGUID	string	Main Case GUID		

Name	Data Type	Label	Required	Documentation
NOTES	string	Notes		
NOTESAFTERINVOICEROWS	string	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	string	Notes before Invoice Lines		
NUMBER	int32	Number		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	string	Payment Status		
PAYMENTTERM	int32	Payment Term		
PONUMBER	string	PO Number		
RECEIVERADDRESSLINE	string	Receiver Addressline		
RECEIVERCITY	string	Receiver City		
RECEIVERCONTACTEMAIL	string	Receiver Contact Email		
RECEIVERCONTACTNAME	string	Receiver Contact Name		
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREETEXT	string			
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONENUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLEREINVOICEADDRESS	string			

Name	Data Type	Label	Required	Documentation
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		
YOURREFERENCE	string	Your Reference		

3.1.147 InvoicesByStatus: Visma Severa Invoices by Status

Catalog: Severa

Schema: API

Label: Invoices by Status

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- status- options- maxInvoices

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicesByStatus. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
maxInvoices	int32	☐		
options	int32	☐		
status	string	☐		

Columns of Table Function

The columns of the table function InvoicesByStatus are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANISATIONNAME	string			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			

Name	Data Type	Label	Required	Documentation
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	string			
CULTURE	string	Culture		
CURRENCYGUID	string	Currency GUID		
CURRENCYNAME	string	Currency Name		
CURRENCYRATE	decimal	Currency Rate		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	string	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	string	GUID		
INVOICEALLOWEDCASEGUIDS	string			
INVOICECONFIG_BILLEXPENSE	string			
INVOICECONFIG_BILLHOUR	string			
INVOICECONFIG_BILLPRODUCT	string			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURPRIMARYGROUPBY	string			
INVOICECONFIG_HOURSECONDARYGROUPBY	string			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	string			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	string			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	int32			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETYPETEXT	string			
INVOICEDETAILS_ORDERIDENTIFIER	string			
INVOICEDETAILS_ORIGINCODE	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	decimal			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	string			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	string			
INVOICEDETAILS_VATBASEAMOUNT	decimal			
INVOICEDETAILS_VATRATEAMOUNT	decimal			

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_VATRATEPERCENT	int32			
INVOICENUMBERINGBUSINESSUNITGUID	string	Invoice Numbering Business Unit GUID		
INVOICEROWS	string			
INVOICESTATUS	string	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
MAINCASEGUID	string	Main Case GUID		
NOTES	string	Notes		
NOTESAFTERINVOICEROWS	string	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	string	Notes before Invoice Lines		
NUMBER	int32	Number		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	string	Payment Status		
PAYMENTTERM	int32	Payment Term		
PONUMBER	string	PO Number		
RECEIVERADDRESSLINE	string	Receiver Addressline		
RECEIVERCITY	string	Receiver City		
RECEIVERCONTACTEMAIL	string	Receiver Contact Email		
RECEIVERCONTACTNAME	string	Receiver Contact Name		
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			

Name	Data Type	Label	Required	Documentation
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREETEXT	string			
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONENUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLEREINVOICEADDRESS	string			
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		

Name	Data Type	Label	Required	Documentation
YOURREFERENCE	string	Your Reference		

3.1.148 InvoicesByStatusAndBusinessUnit: Visma Severa Invoices by Status and Business Unit

Catalog: Severa

Schema: API

Label: Invoices by Status and Business Unit

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: status- businessUnitGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicesByStatusAndBusinessUnit. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
status	string	☐		

Columns of Table Function

The columns of the table function InvoicesByStatusAndBusinessUnit are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		

Name	Data Type	Label	Required	Documentation
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANISATIONNAME	string			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	string			
CULTURE	string	Culture		
CURRENCYGUID	string	Currency GUID		
CURRENCYNAME	string	Currency Name		
CURRENCYRATE	decimal	Currency Rate		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	string	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	string	GUID		
INVOICEALLOWEDCASEGUIDS	string			
INVOICECONFIG_BILLEXPENSE	string			
INVOICECONFIG_BILLHOUR	string			
INVOICECONFIG_BILLPRODUCT	string			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	string			

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURPRIMARYGROUPBY	string			
INVOICECONFIG_HOURSECONDARYGROUPBY	string			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	string			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	string			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	int32			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETYPETEXT	string			

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_ORDERIDENTIFIER	string			
INVOICEDETAILS_ORIGINCODE	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	decimal			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	string			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	string			
INVOICEDETAILS_VATBASEAMOUNT	decimal			
INVOICEDETAILS_VATRATEAMOUNT	decimal			
INVOICEDETAILS_VATRATEPERCENT	int32			
INVOICENUMBERINGBUSINESSUNITGUID	string	Invoice Numbering Business Unit GUID		
INVOICEROWS	string			
INVOICESTATUS	string	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
MAINCASEGUID	string	Main Case GUID		
NOTES	string	Notes		
NOTESAFTERINVOICEROWS	string	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	string	Notes before Invoice Lines		
NUMBER	int32	Number		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	string	Payment Status		
PAYMENTTERM	int32	Payment Term		
PONUMBER	string	PO Number		
RECEIVERADDRESSLINE	string	Receiver Addressline		
RECEIVERCITY	string	Receiver City		
RECEIVERCONTACTEMAIL	string	Receiver Contact Email		
RECEIVERCONTACTNAME	string	Receiver Contact Name		

Name	Data Type	Label	Required	Documentation
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREETEXT	string			
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONENUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLEREINVOICEADDRESS	string			
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			

Name	Data Type	Label	Required	Documentation
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		
YOURREFERENCE	string	Your Reference		

3.1.149 InvoiceStatuses: Visma Severa Invoice Statuses

Catalog: Severa

Schema: API

Label: Invoice Statuses

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table InvoiceStatuses are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DESCRIPTION	string	Ddescription		
HASINVOICENUMBER	boolean	Has Invoice Number		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPAID	boolean	Is Paid		
ISREADONLY	boolean	Is Read-only		
ISSENT	boolean	Is Sent		
ISWAITINGPAYMENT	boolean	Is Waiting Payment		
NAME	string	Name		
SORTORDER	int32	Sort Order		

3.1.150 ItemByGUID: Visma Severa Item by GUID

Catalog: Severa

Schema: API

Label: Item by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: itemGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ItemByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
itemGUID	string	☐		

Columns of Table Function

The columns of the table function ItemByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		

Name	Data Type	Label	Required	Documentation
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		

Name	Data Type	Label	Required	Documentation
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.151 ItemByPurchaseOrderNumber: Visma Severa Item by Purchase Order Number

Catalog: Severa

Schema: API

Label: Item by Purchase Order Number

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: poNumber

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ItemByPurchaseOrderNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
poNumber	string	☐		

Columns of Table Function

The columns of the table function ItemByPurchaseOrderNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		

Name	Data Type	Label	Required	Documentation
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		

Name	Data Type	Label	Required	Documentation
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.152 ItemsByCaseGUID: Visma Severa Items by Project GUID

Catalog: Severa

Schema: API

Label: Items by Project GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ItemsByCaseGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGUID	string	☐		

Columns of Table Function

The columns of the table function ItemsByCaseGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		

Name	Data Type	Label	Required	Documentation
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		

Name	Data Type	Label	Required	Documentation
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.153 ItemsByInsertDate: Visma Severa Items by Insert Date

Catalog: Severa

Schema: API

Label: Items by Insert Date

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: insertStartDate- insertEndDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ItemsByInsertDate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
insertEndDate	datetime	☐		
insertStartDate	datetime	☐		

Columns of Table Function

The columns of the table function ItemsByInsertDate are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		

Name	Data Type	Label	Required	Documentation
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.154 ItemsChangedSince: Visma Severa Items Changed since

Catalog: Severa

Schema: API

Label: Items Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: startDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ItemsChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
startDate	datetime	☐		

Columns of Table Function

The columns of the table function ItemsChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project		

Name	Data Type	Label	Required	Documentation
		Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		

Name	Data Type	Label	Required	Documentation
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.155 LanguageByGUID: Visma Severa Language by GUID

Catalog: Severa

Schema: API

Label: Language by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: languageGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function LanguageByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
languageGUID	string	☐		

Columns of Table Function

The columns of the table function LanguageByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
IETFLANGUAGETAG	string	IETF Language Tag		
NAME	string	Name		

3.1.156 LanguageByIetfTag: Visma Severa Language by IETF Tag

Catalog: Severa

Schema: API

Label: Language by IETF Tag

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: ietfLanguageTag

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function LanguageByIetfTag. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
ietfLanguageTag	string	☐		

Columns of Table Function

The columns of the table function LanguageByIetfTag are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
IETFLANGUAGETAG	string	IETF Language Tag		
NAME	string	Name		

3.1.157 LeadSourceByGUID: Visma Severa Lead Source by GUID

Catalog: Severa

Schema: API

Label: Lead Source by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: leadSourceGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function LeadSourceByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
leadSourceGUID	string	□		

Columns of Table Function

The columns of the table function LeadSourceByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.158 LeadSources: Visma Severa Lead Sources

Catalog: Severa

Schema: API

Label: Lead Sources

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table LeadSources are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		

Name	Data Type	Label	Required	Documentation
NAME	string	Name		

3.1.159 OverTimeByGUID: Visma Severa Overtime by GUID

Catalog: Severa

Schema: API

Label: Overtime by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: overTimeGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function OverTimeByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
overTimeGUID	string	☐		

Columns of Table Function

The columns of the table function OverTimeByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
INCLUDEINFLEXTIME	boolean	Include in Flextime		
ISACTIVE	boolean	Is Active		
ISWORKTIME	boolean	Is Work Time		
NAME	string	Name		
PERCENTAGE	decimal	Percentage		

3.1.160 Overtimes: Visma Severa Overtimes

Catalog: Severa

Schema: API

Label: Overtimes

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table Overtimes are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
INCLUDEINFLEXTIME	boolean	Include in Flextime		
ISACTIVE	boolean	Is Active		
ISWORKTIME	boolean	Is Work Time		
NAME	string	Name		
PERCENTAGE	decimal	Percentage		

3.1.161 PhaseByGUID: Visma Severa Phase by GUID

Catalog: Severa

Schema: API

Label: Phase by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: phaseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PhaseByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
phaseGUID	string	☐		

Columns of Table Function

The columns of the table function PhaseByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CASEGUID	string	Project GUID		
CODE	string	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	string	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	decimal	Flat Rate Additional Hours Unit Price		
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	decimal	Flat Rate Hours		
FLATRATE_PRICE	decimal	Flat Rate Price		
GUID	string	GUID		
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	decimal	Original Work Estimate		
OWNERUSERGUID	string	Owner User GUID		
PARENTPHASEGUID	string	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	decimal	Price per Hour		
SORTORDER	int32	Sort Order		
WORKESTIMATE	decimal	Work Estimate		
WORKTYPEGUID	string	Work Type GUID		

3.1.162 PhaseMembersByPhaseGUID: Visma Severa Phase Members by Phase GUID

Catalog: Severa

Schema: API

Label: Phase Members by Phase GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: phaseGUID- onlyActive

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PhaseMembersByPhaseGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
onlyActive	boolean	☐		
phaseGUID	string	☐		

Columns of Table Function

The columns of the table function PhaseMembersByPhaseGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
STRING	string			

3.1.163 PhasesByCaseGUID: Visma Severa Phases by Project GUID

Catalog: Severa

Schema: API

Label: Phases by Project GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PhasesByCaseGUID. A value must be provided at all times for required parameters, but

optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGUID	string	☐		

Columns of Table Function

The columns of the table function PhasesByCaseGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CASEGUID	string	Project GUID		
CODE	string	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	string	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	decimal	Flat Rate Additional Hours Unit Price		
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	decimal	Flat Rate Hours		
FLATRATE_PRICE	decimal	Flat Rate Price		
GUID	string	GUID		
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	decimal	Original Work Estimate		
OWNERUSERGUID	string	Owner User GUID		
PARENTPHASEGUID	string	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	decimal	Price per Hour		
SORTORDER	int32	Sort Order		
WORKESTIMATE	decimal	Work Estimate		

Name	Data Type	Label	Required	Documentation
WORKTYPEGUID	string	Work Type GUID		

3.1.164 PhasesByCode: Visma Severa Phases by Code

Catalog: Severa

Schema: API

Label: Phases by Code

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: phaseCode- caseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PhasesByCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGUID	string	☐		
phaseCode	string	☐		

Columns of Table Function

The columns of the table function PhasesByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CASEGUID	string	Project GUID		
CODE	string	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	string	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	decimal	Flat Rate Additional Hours Unit Price		

Name	Data Type	Label	Required	Documentation
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	decimal	Flat Rate Hours		
FLATRATE_PRICE	decimal	Flat Rate Price		
GUID	string	GUID		
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	decimal	Original Work Estimate		
OWNERUSERGUID	string	Owner User GUID		
PARENTPHASEGUID	string	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	decimal	Price per Hour		
SORTORDER	int32	Sort Order		
WORKESTIMATE	decimal	Work Estimate		
WORKTYPEGUID	string	Work Type GUID		

3.1.165 PhasesChangedSince: Visma Severa Phases Changed since

Catalog: Severa

Schema: API

Label: Phases Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PhasesChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
options	int32	☐		
startDateTime	datetime	☐		

Columns of Table Function

The columns of the table function PhasesChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CASEGUID	string	Project GUID		
CODE	string	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	string	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	decimal	Flat Rate Additional Hours Unit Price		
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	decimal	Flat Rate Hours		
FLATRATE_PRICE	decimal	Flat Rate Price		
GUID	string	GUID		
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	decimal	Original Work Estimate		
OWNERUSERGUID	string	Owner User GUID		
PARENTPHASEGUID	string	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	decimal	Price per Hour		
SORTORDER	int32	Sort Order		
WORKESTIMATE	decimal	Work Estimate		
WORKTYPEGUID	string	Work Type GUID		

3.1.166 PricelistByGUID: Visma Severa Price List by GUID

Catalog: Severa

Schema: API

Label: Price List by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: pricelistGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PricelistByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
pricelistGuid	string	☐		

Columns of Table Function

The columns of the table function PricelistByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISINTERNAL	boolean	Is Internal		
ISVOLUMEPRICING	boolean	Is Volume Pricing		
NAME	string	Name		

3.1.167 Pricelists: Visma Severa Price Lists

Catalog: Severa

Schema: API

Label: Price Lists

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table Pricelists are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISINTERNAL	boolean	Is Internal		
ISVOLUMEPRICING	boolean	Is Volume Pricing		
NAME	string	Name		

3.1.168 ProductByCode: Visma Severa Product by Code

Catalog: Severa

Schema: API

Label: Product by Code

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: productCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductByCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
productCode	string	☐		

Columns of Table Function

The columns of the table function ProductByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
PRODUCTCATEGORYGUID	string	Product Category GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.169 ProductByCodeIncludeInactive: Visma Severa Product by Code Include Inactive

Catalog: Severa

Schema: API

Label: Product by Code Include Inactive

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: productCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductByCodeIncludeInactive. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
productCode	string	☐		

Columns of Table Function

The columns of the table function ProductByCodeIncludeInactive are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
PRODUCTCATEGORYGUID	string	Product Category GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.170 ProductByGUID: Visma Severa Product by GUID

Catalog: Severa

Schema: API

Label: Product by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: productGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
productGUID	string	☐		

Columns of Table Function

The columns of the table function ProductByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
PRODUCTCATEGORYGUID	string	Product Category GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.171 ProductCategoriesChangedSince: Visma Severa Product Categories Changed since

Catalog: Severa

Schema: API

Label: Product Categories Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductCategoriesChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
options	int32	☐		
startDateTime	datetime	☐		

Columns of Table Function

The columns of the table function ProductCategoriesChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		

3.1.172 ProductCategoryByCode: Visma Severa Product Category by Code

Catalog: Severa

Schema: API

Label: Product Category by Code

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: productCategoryCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductCategoryByCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
productCategoryCode	string	☐		

Columns of Table Function

The columns of the table function ProductCategoryByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		

3.1.173 ProductCategoryByGUID: Visma Severa Product Category by GUID

Catalog: Severa

Schema: API

Label: Product Category by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: productCategoryGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductCategoryByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
productCategoryGUID	string	□		

Columns of Table Function

The columns of the table function ProductCategoryByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		

3.1.174 ProductCostAccountAssociationsByGUID: Visma Severa Product Cost Account Associations by GUID

Catalog: Severa

Schema: API

Label: Product Cost Account Associations by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: productGUID- includeDefault

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductCostAccountAssociationsByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
includeDefault	boolean	☐		
productGUID	string	☐		

Columns of Table Function

The columns of the table function ProductCostAccountAssociationsByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISTRAVELTYPEDEFAULT	boolean	Is Travel Type Default		
NAME	string	Name		
NUMBER	string	Number		
TAXPERCENTAGE	decimal	Tax (%)		

3.1.175 ProductsChangedSince: Visma Severa Products Changed since

Catalog: Severa

Schema: API

Label: Products Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductsChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
options	int32	☐		

Name	Data Type	Required	Default Value	Documentation
startDateTime	datetime	☐		

Columns of Table Function

The columns of the table function ProductsChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
PRODUCTCATEGORYGUID	string	Product Category GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.176 ReimbursedHourEntriesByDate: Visma Severa Reimbursed Hour Entries by Date

Catalog: Severa

Schema: API

Label: Reimbursed Hour Entries by Date

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- startdate- enddate- inCreditNote

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ReimbursedHourEntriesByDate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
enddate	datetime	☐		
inCreditNote	boolean	☐		
startdate	datetime	☐		

Columns of Table Function

The columns of the table function ReimbursedHourEntriesByDate are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		

Name	Data Type	Label	Required	Documentation
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.177 ReimbursedHourEntriesChangedSince: Visma Severa Reimbursed Hour Entries Changed since

Catalog: Severa

Schema: API

Label: Reimbursed Hour Entries Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: startDate- inCreditNote

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ReimbursedHourEntriesChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
inCreditNote	boolean	☐		
startDate	datetime	☐		

Columns of Table Function

The columns of the table function ReimbursedHourEntriesChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		

Name	Data Type	Label	Required	Documentation
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		

Name	Data Type	Label	Required	Documentation
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.178 ReimbursedHourEntryByGUID: Visma Severa Reimbursed Hour Entry by GUID

Catalog: Severa

Schema: API

Label: Reimbursed Hour Entry by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: hourEntryGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ReimbursedHourEntryByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
hourEntryGUID	string	☐		

Columns of Table Function

The columns of the table function ReimbursedHourEntryByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization		

Name	Data Type	Label	Required	Documentation
		Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.179 ReimbursedItemByGUID: Visma Severa Reimbursed Item by GUID

Catalog: Severa

Schema: API

Label: Reimbursed Item by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: itemGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ReimbursedItemByGUID. A value must be provided at all times for required parameters, but

optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
itemGUID	string	□		

Columns of Table Function

The columns of the table function ReimbursedItemByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		

Name	Data Type	Label	Required	Documentation
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		

Name	Data Type	Label	Required	Documentation
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.180 ReimbursedItemsChangedSince: Visma Severa Reimbursed Items Changed since

Catalog: Severa

Schema: API

Label: Reimbursed Items Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: startDate- inCreditNote

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ReimbursedItemsChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
inCreditNote	boolean	☐		
startDate	datetime	☐		

Columns of Table Function

The columns of the table function ReimbursedItemsChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		

Name	Data Type	Label	Required	Documentation
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.181 RemoveAccountGroupMembers: Visma Severa Remove Account Group Members

Catalog: Severa

Schema: API

Label: Remove Account Group Members

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accountGUID- accountGroupGUIDs

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function RemoveAccountGroupMembers. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountGroupGUIDs	string	☐		
accountGUID	string	☐		

Columns of Table Function

The columns of the table function RemoveAccountGroupMembers are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DUMMY	string	Dummy	☑	Dummy column

3.1.182 RemoveActivityContactParticipant: Visma Severa Remove Activity Contact Participant

Catalog: Severa

Schema: API

Label: Remove Activity Contact Participant

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityGUID- contactGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function RemoveActivityContactParticipant. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityGUID	string	☐		
contactGUID	string	☐		

Columns of Table Function

The columns of the table function RemoveActivityContactParticipant are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
REMOVEACTIVITYCONTACTPARTICIPANTRESULT	boolean			

3.1.183 RemoveActivityResourceParticipant: Visma Severa Remove Activity Resource Participant

Catalog: Severa

Schema: API

Label: Remove Activity Resource Participant

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityGUID- resourceGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function RemoveActivityResourceParticipant. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityGUID	string	☐		
resourceGUID	string	☐		

Columns of Table Function

The columns of the table function RemoveActivityResourceParticipant are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
REMOVEACTIVITYRESOURCEPARTICIPANTRESULT	boolean			

3.1.184 RemoveActivityUserParticipant: Visma Severa Remove Activity User Participant

Catalog: Severa

Schema: API

Label: Remove Activity User Participant

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityGUID- userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function RemoveActivityUserParticipant. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityGUID	string	☐		
userGUID	string	☐		

Columns of Table Function

The columns of the table function RemoveActivityUserParticipant are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
REMOVEACTIVITYUSERPARTICIPANTRESULT	boolean			

3.1.185 ResourceAllocationByGUID: Visma Severa Resource Allocations by GUID

Catalog: Severa

Schema: API

Label: Resource Allocations by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: resourceAllocationGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ResourceAllocationByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
resourceAllocationGUID	string	☐		

Columns of Table Function

The columns of the table function ResourceAllocationByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.186 ResourceAllocations: Visma Severa Resource Allocations

Catalog: Severa

Schema: API

Label: Resource Allocations

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGUIDs- userGUIDs- businessUnitGUIDs- supervisorUserGUIDs- caseStatusGUIDs- salesProcessGUIDs- caseTagGUIDs- userTagGUIDs- startDate- endDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ResourceAllocations. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUIDs	string	☐		
caseGUIDs	string	☐		
caseStatusGUIDs	string	☐		
caseTagGUIDs	string	☐		
endDate	datetime	☐		
salesProcessGUIDs	string	☐		

Name	Data Type	Required	Default Value	Documentation
startDate	datetime	☐		
supervisorUserGUIDs	string	☐		
userGUIDs	string	☐		
userTagGUIDs	string	☐		

Columns of Table Function

The columns of the table function ResourceAllocations are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.187 ResourceAllocationsByCaseGUID: Visma Severa Resource Allocations by Project GUID

Catalog: Severa

Schema: API

Label: Resource Allocations by Project GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ResourceAllocationsByCaseGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGUID	string	☐		

Columns of Table Function

The columns of the table function ResourceAllocationsByCaseGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.188 ResourceAllocationsByCaseGUIDUserGUID: Visma Severa Resource Allocations by Project GUID and User GUID

Catalog: Severa

Schema: API

Label: Resource Allocations by Project GUID and User GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGUID- userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ResourceAllocationsByCaseGUIDUserGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGUID	string	☐		
userGUID	string	☐		

Columns of Table Function

The columns of the table function ResourceAllocationsByCaseGUIDUserGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.189 ResourceAllocationsByUserGUID: Visma Severa Resource Allocations by User GUID

Catalog: Severa

Schema: API

Label: Resource Allocations by User GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ResourceAllocationsByUserGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userGUID	string	□		

Columns of Table Function

The columns of the table function ResourceAllocationsByUserGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.190 ResourceAllocationsChangedSince: Visma Severa Resource Allocations Changed since

Catalog: Severa

Schema: API

Label: Resource Allocations Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: startDateTime

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ResourceAllocationsChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
startDateTime	datetime	☐		

Columns of Table Function

The columns of the table function ResourceAllocationsChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.191 ResourceByGUID: Visma Severa Resource by GUID

Catalog: Severa

Schema: API

Label: Resource by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: resourceGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ResourceByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
resourceGUID	string	☐		

Columns of Table Function

The columns of the table function ResourceByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.192 Resources: Visma Severa Resources

Catalog: Severa

Schema: API

Label: Resources

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table Resources are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.193 SalesProcessByGUID: Visma Severa Sales Process by GUID

Catalog: Severa

Schema: API

Label: Sales Process by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: salesProcessGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SalesProcessByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
salesProcessGUID	string	☐		

Columns of Table Function

The columns of the table function SalesProcessByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DEFAULTPROBABILITY	int32	Default Probability		
GUID	string	GUID		
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
ISALSOITEMSTATUS	boolean	Is also Item Status		
ISINPROGRESS	boolean	Is in Progress		
ISLOST	boolean	Is Lost		
ISOFFER	boolean	Is Offer		
ISWON	boolean	Is Won		
NAME	string	Name		

3.1.194 SalesProcesses: Visma Severa Sales Processes

Catalog: Severa

Schema: API

Label: Sales Processes

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table SalesProcesses are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DEFAULTPROBABILITY	int32	Default Probability		
GUID	string	GUID		

Name	Data Type	Label	Required	Documentation
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
ISALSOITEMSTATUS	boolean	Is also Item Status		
ISINPROGRESS	boolean	Is in Progress		
ISLOST	boolean	Is Lost		
ISOFFER	boolean	Is Offer		
ISWON	boolean	Is Won		
NAME	string	Name		

3.1.195 SalesStatusByGUID: Visma Severa Sales Status by GUID

Catalog: Severa

Schema: API

Label: Sales Status by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: salesStatusGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SalesStatusByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
salesStatusGUID	string	☐		

Columns of Table Function

The columns of the table function SalesStatusByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CASEGUID	string	Project GUID		

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
SALESPROCESSGUID	string	Sales Process GUID		
TIMESTAMP	datetime	Time Stamp		

3.1.196 SalesStatusesForCase: Visma Severa Sales Statuses for Project

Catalog: Severa

Schema: API

Label: Sales Statuses for Project

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SalesStatusesForCase. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGuid	string	☐		

Columns of Table Function

The columns of the table function SalesStatusesForCase are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CASEGUID	string	Project GUID		
GUID	string	GUID		
SALESPROCESSGUID	string	Sales Process GUID		
TIMESTAMP	datetime	Time Stamp		

3.1.197 Server: Visma Severa Server

Catalog: Severa

Schema: API

Label: Server

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table Server are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
SERVERRESULT	string			

3.1.198 SetInvoiceNumber: Visma Severa Set Invoice Number

Catalog: Severa

Schema: API

Label: Set Invoice Number

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: invoiceGUID- invoiceNumber- referenceNumber- status

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SetInvoiceNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
invoiceGUID	string	☐		
invoiceNumber	int32	☐		
referenceNumber	string	☐		
status	string	☐		

Columns of Table Function

The columns of the table function SetInvoiceNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DUMMY	string	Dummy	☑	Dummy column

3.1.199 SetInvoicePaymentDate: Visma Severa Set Invoice Payment Date

Catalog: Severa

Schema: API

Label: Set Invoice Payment Date

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: invoiceGUID- paymentDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SetInvoicePaymentDate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
invoiceGUID	string	☐		
paymentDate	datetime	☐		

Columns of Table Function

The columns of the table function SetInvoicePaymentDate are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DUMMY	string	Dummy	☑	Dummy column

3.1.200 SetInvoiceStatus: Visma Severa Set Invoice Status

Catalog: Severa

Schema: API

Label: Set Invoice Status

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: invoiceGUID- status

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SetInvoiceStatus. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
invoiceGUID	string	☐		
status	string	☐		

Columns of Table Function

The columns of the table function SetInvoiceStatus are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DUMMY	string	Dummy	☑	Dummy column

3.1.201 SetPhaseMemberListForPhase: Visma Severa Set Phase Member List for Phase

Catalog: Severa

Schema: API

Label: Set Phase Member List for Phase

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: phaseGUID- userGUIDs

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SetPhaseMemberListForPhase. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
phaseGUID	string	☐		
userGUIDs	string	☐		

Columns of Table Function

The columns of the table function SetPhaseMemberListForPhase are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
SETPHASEMEMBERLISTFORPHASERESULT	boolean			

3.1.202 SetTravelReimbursementStatus: Visma Severa Set Travel Reimbursement Status

Catalog: Severa

Schema: API

Label: Set Travel Reimbursement Status

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: travelReimbursementGUID- status

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SetTravelReimbursementStatus. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
status	string	☐		
travelReimbursementGUID	string	☐		

Columns of Table Function

The columns of the table function SetTravelReimbursementStatus are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DUMMY	string	Dummy	☑	Dummy column

3.1.203 TagByGUID: Visma Severa Tag by GUID

Catalog: Severa

Schema: API

Label: Tag by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: tagGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TagByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
tagGUID	string	☐		

Columns of Table Function

The columns of the table function TagByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTEXT	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyword		
WEIGHT	int32	Weight		

3.1.204 TagsByCaseGUID: Visma Severa Tags by Project GUID

Catalog: Severa

Schema: API

Label: Tags by Project GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TagsByCaseGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseGUID	string	☐		

Columns of Table Function

The columns of the table function TagsByCaseGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTEXT	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyword		
WEIGHT	int32	Weight		

3.1.205 TimeEntriesByDate: Visma Severa Time Entries by Date

Catalog: Severa

Schema: API

Label: Time Entries by Date

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: eventDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TimeEntriesByDate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
eventDate	datetime	☐		

Columns of Table Function

The columns of the table function TimeEntriesByDate are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DESCRIPTION	string	Ddescription		

Name	Data Type	Label	Required	Documentation
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
USERGUID	string	User GUID		

3.1.206 TimeEntriesByDateAndUserGUID: Visma Severa Time Entries by Date and User GUID

Catalog: Severa

Schema: API

Label: Time Entries by Date and User GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: eventDate- userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TimeEntriesByDateAndUserGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
eventDate	datetime	☐		
userGUID	string	☐		

Columns of Table Function

The columns of the table function TimeEntriesByDateAndUserGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DESCRIPTION	string	Ddescription		

Name	Data Type	Label	Required	Documentation
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
USERGUID	string	User GUID		

3.1.207 TimeEntriesChangedSince: Visma Severa Time Entries Changed since

Catalog: Severa

Schema: API

Label: Time Entries Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: startDateTime

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TimeEntriesChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
startDateTime	datetime	☐		

Columns of Table Function

The columns of the table function TimeEntriesChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DESCRIPTION	string	Ddescription		

Name	Data Type	Label	Required	Documentation
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
USERGUID	string	User GUID		

3.1.208 TimezoneByGUID: Visma Severa Timezone by GUID

Catalog: Severa

Schema: API

Label: Timezone by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: timezoneGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TimezoneByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
timezoneGuid	string	☐		

Columns of Table Function

The columns of the table function TimezoneByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DIFFERENCETOGMT	decimal	Difference to GMT		

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
NAME	string	Name		
SYSTEMNAME	string	System Name		

3.1.209 Timezones: Visma Severa Timezones

Catalog: Severa

Schema: API

Label: Timezones

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table Timezones are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DIFFERENCETOGMT	decimal	Difference to GMT		
GUID	string	GUID		
NAME	string	Name		
SYSTEMNAME	string	System Name		

3.1.210 TravelExpenseByCode: Visma Severa Travel Expense by Code

Catalog: Severa

Schema: API

Label: Travel Expense by Code

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: travelExpenseCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelExpenseByCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
travelExpenseCode	string	□		

Columns of Table Function

The columns of the table function TravelExpenseByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
COSTCURRENCYGUID	string	Cost Currency GUID		
DESCRIPTION	string	Ddescription		
EXPENSECLASS	string	Expense Class		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.211 TravelExpenseByCodeIncludeInactive: Visma Severa Travel Expense by Code Include Inactive

Catalog: Severa

Schema: API

Label: Travel Expense by Code Include Inactive

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: travelExpenseCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelExpenseByCodeIncludeInactive. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the

execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
travelExpenseCode	string	☐		

Columns of Table Function

The columns of the table function TravelExpenseByCodeIncludeInactive are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
COSTCURRENCYGUID	string	Cost Currency GUID		
DESCRIPTION	string	Ddescription		
EXPENSECLASS	string	Expense Class		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.212 TravelExpenseByGUID: Visma Severa Travel Expense by GUID

Catalog: Severa

Schema: API

Label: Travel Expense by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: travelExpenseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelExpenseByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
travelExpenseGUID	string	☐		

Columns of Table Function

The columns of the table function TravelExpenseByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
COSTCURRENCYGUID	string	Cost Currency GUID		
DESCRIPTION	string	Ddescription		
EXPENSECLASS	string	Expense Class		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.213 TravelExpensesByDate: Visma Severa Travel Expenses by Date

Catalog: Severa

Schema: API

Label: Travel Expenses by Date

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- startDate- endDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelExpensesByDate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
endDate	datetime	☐		
startDate	datetime	☐		

Columns of Table Function

The columns of the table function TravelExpensesByDate are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User		

Name	Data Type	Label	Required	Documentation
		Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		

Name	Data Type	Label	Required	Documentation
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.214 TravelExpensesChangedSince: Visma Severa Travel Expenses Changed since

Catalog: Severa

Schema: API

Label: Travel Expenses Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelExpensesChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
options	int32	☐		
startDateTime	datetime	☐		

Columns of Table Function

The columns of the table function TravelExpensesChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
COSTCURRENCYGUID	string	Cost Currency GUID		
DESCRIPTION	string	Ddescription		
EXPENSECLASS	string	Expense Class		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.215 TravelReimbursementAttachments: Visma Severa Travel Reimbursement Attachments

Catalog: Severa

Schema: API

Label: Travel Reimbursement Attachments

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: guid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelReimbursementAttachments. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
guid	string	☐		

Columns of Table Function

The columns of the table function TravelReimbursementAttachments are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTENTTYPE	string	Content Type		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
NAME	string	Name		
SIZE	int32	Size		

3.1.216 TravelReimbursementByGUID: Visma Severa Travel Reimbursement by GUID

Catalog: Severa

Schema: API

Label: Travel Reimbursement by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: travelReimbursementGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelReimbursementByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
travelReimbursementGUID	string	☐		

Columns of Table Function

The columns of the table function TravelReimbursementByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADVANCEPAYMENT	decimal	Advance Payment		
APPROVALDATE	datetime	Approval Date		
APPROVEDBYUSERGUID	string	Approved by User GUID		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
GUID	string	GUID		
NOTES	string	Notes		
NUMBER	int32	Number		
TRAVELREIMBURSEMENTROWS	string			
TRAVELREIMBURSEMENTSTATUS	string	Travel Reimbursement Status		
USERGUID	string	User GUID		

3.1.217 TravelReimbursementByNumber: Visma Severa Travel Reimbursement by Number

Catalog: Severa

Schema: API

Label: Travel Reimbursement by Number

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: travelReimbursementNumber-travelReimbursementNumberingBusinessUnitGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelReimbursementByNumber. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
travelReimbursementNumber	int32	☐		
travelReimbursementNumberingBusinessUnitGUID	string	☐		

Columns of Table Function

The columns of the table function TravelReimbursementByNumber are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADVANCEPAYMENT	decimal	Advance Payment		
APPROVALDATE	datetime	Approval Date		
APPROVEDBYUSERGUID	string	Approved by User GUID		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
GUID	string	GUID		
NOTES	string	Notes		
NUMBER	int32	Number		
TRAVELREIMBURSEMENTROWS	string			
TRAVELREIMBURSEMENTSTATUS	string	Travel Reimbursement Status		
USERGUID	string	User GUID		

3.1.218 TravelReimbursementPdf: Visma Severa Travel Reimbursement PDF

Catalog: Severa

Schema: API

Label: Travel Reimbursement PDF

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: guid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelReimbursementPdf. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
guid	string	☐		

Columns of Table Function

The columns of the table function TravelReimbursementPdf are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GETTRAVELREIMBURSEMENTPDF RESULT	byte[]			

3.1.219 TravelReimbursementsByDate: Visma Severa Travel Reimbursement by Date

Catalog: Severa

Schema: API

Label: Travel Reimbursement by Date

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: startDate- endDate- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelReimbursementsByDate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the

execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
endDate	datetime	☐		
options	int32	☐		
startDate	datetime	☐		

Columns of Table Function

The columns of the table function TravelReimbursementsByDate are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADVANCEPAYMENT	decimal	Advance Payment		
APPROVALDATE	datetime	Approval Date		
APPROVEDBYUSERGUID	string	Approved by User GUID		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
GUID	string	GUID		
NOTES	string	Notes		
NUMBER	int32	Number		
TRAVELREIMBURSEMENTROWS	string			
TRAVELREIMBURSEMENTSTATUS	string	Travel Reimbursement Status		
USERGUID	string	User GUID		

3.1.220 TravelReimbursementsByStatus: Visma Severa Travel Reimbursement by Status

Catalog: Severa

Schema: API

Label: Travel Reimbursement by Status

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: status- options- businessUnitGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelReimbursementsByStatus. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
options	int32	☐		
status	string	☐		

Columns of Table Function

The columns of the table function TravelReimbursementsByStatus are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADVANCEPAYMENT	decimal	Advance Payment		
APPROVALDATE	datetime	Approval Date		
APPROVEDBYUSERGUID	string	Approved by User GUID		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
GUID	string	GUID		
NOTES	string	Notes		
NUMBER	int32	Number		
TRAVELREIMBURSEMENTROWS	string			
TRAVELREIMBURSEMENTSTATUS	string	Travel Reimbursement Status		
USERGUID	string	User GUID		

3.1.221 TravelReimbursementsByStatusGUID: Visma Severa Travel Reimbursement by Status GUID

Catalog: Severa

Schema: API

Label: Travel Reimbursement by Status GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: travelReimbursementStatusGUID- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelReimbursementsByStatusGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
options	int32	☐		
travelReimbursementStatusGUID	string	☐		

Columns of Table Function

The columns of the table function TravelReimbursementsByStatusGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADVANCEPAYMENT	decimal	Advance Payment		
APPROVALDATE	datetime	Approval Date		
APPROVEDBYUSERGUID	string	Approved by User GUID		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
GUID	string	GUID		
NOTES	string	Notes		
NUMBER	int32	Number		

Name	Data Type	Label	Required	Documentation
TRAVELREIMBURSEMENTROWS	string			
TRAVELREIMBURSEMENTSTATUS	string	Travel Reimbursement Status		
USERGUID	string	User GUID		

3.1.222 TravelReimbursementStatuses: Visma Severa Travel Reimbursement Statuses

Catalog: Severa

Schema: API

Label: Travel Reimbursement Statuses

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table TravelReimbursementStatuses are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISAPPROVED	boolean	Is Approved		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
SORTORDER	int32	Sort Order		

3.1.223 UpdateAccessRightsProfile: Visma Severa Update Access Rights Profile

Catalog: Severa

Schema: API

Label: Update Access Rights Profile

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accessRightsProfile_GUID- accessRightsProfile_IsActive- accessRightsProfile_IsDefault- accessRightsProfile_Name- accessRightsProfile_Rights_Account- accessRightsProfile_Rights_AccountDelete- accessRightsProfile_Rights_Administrator- accessRightsProfile_Rights_Case- accessRightsProfile_Rights_CaseDelete- accessRightsProfile_Rights_CaseOwnerWorkHourApproval- accessRightsProfile_Rights_ScheduleJobs- accessRightsProfile_Rights_Sharing- accessRightsProfile_Rights_TravelReimbursement- accessRightsProfile_Rights_Users- accessRightsProfile_Rights_WorkHourApproval

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateAccessRightsProfile. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accessRightsProfile_GUID	string	☐		
accessRightsProfile_IsActive	boolean	☐		
accessRightsProfile_IsDefault	boolean	☐		
accessRightsProfile_Name	string	☐		
accessRightsProfile_Rights_Account	string	☐		
accessRightsProfile_Rights_AccountDelete	boolean	☐		
accessRightsProfile_Rights_Administrator	string	☐		
accessRightsProfile_Rights_Case	string	☐		
accessRightsProfile_Rights_CaseDelete	boolean	☐		
accessRightsProfile_Rights_CaseOwnerWorkHourApproval	boolean	☐		
accessRightsProfile_Rights_ScheduleJobs	string	☐		
accessRightsProfile_Rights_Sharing	boolean	☐		
accessRightsProfile_Rights_TravelReimbursement	string	☐		
accessRightsProfile_Rights_Users	string	☐		
accessRightsProfile_Rights_WorkHourApproval	string	☐		

Columns of Table Function

The columns of the table function UpdateAccessRightsProfile are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		

Name	Data Type	Label	Required	Documentation
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		
RIGHTS_ACCOUNT	string			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	string			
RIGHTS_CASE	string			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOURAPPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	string			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	string			
RIGHTS_USERS	string			
RIGHTS_WORKHOURAPPROVAL	string			

3.1.224 UpdateAccount: Visma Severa Update Account

Catalog: Severa

Schema: API

Label: Update Account

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: accountInfo_AccountGroupGUIDs- accountInfo_AccountOwnerUserGUID- accountInfo_AccountRating- accountInfo_CompanyGUID- accountInfo_CurrencyCode- accountInfo_CurrencyGUID- accountInfo_GUID- accountInfo_InsertTS- accountInfo_InvoicingVat- accountInfo_IsActive- accountInfo_IsInternal- accountInfo_LanguageCode- accountInfo_LanguageGUID- accountInfo_Name- accountInfo_Notes- accountInfo_Number- accountInfo_OverdueInterest- accountInfo_PaymentTerm- accountInfo_PricelistGUID- accountInfo_ReverseCharge- accountInfo_ReverseChargeDescription- accountInfo_UpdatedTS- accountInfo_UrlToAccount- accountInfo_eInvoiceAddress- accountInfo_eInvoiceOperator- companyInfo_AnnualRevenue- companyInfo_AnnualRevenue2- companyInfo_Email- companyInfo_Employees- companyInfo_GUID- companyInfo_HeadOfficeAddressGUID- companyInfo_HierarchyGUID- companyInfo_IndustryGUID- companyInfo_LanguageGUID- companyInfo_Name- companyInfo_TimezoneGUID- companyInfo_VatNumber- companyInfo_Website

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateAccount. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountInfo_AccountGroupGUIDs	string	☐		
accountInfo_AccountOwnerUserGUID	string	☐		
accountInfo_AccountRating	int32	☐		
accountInfo_CompanyGUID	string	☐		
accountInfo_CurrencyCode	string	☐		
accountInfo_CurrencyGUID	string	☐		
accountInfo_eInvoiceAddress	string	☐		
accountInfo_eInvoiceOperator	string	☐		
accountInfo_GUID	string	☐		
accountInfo_InsertTS	datetime	☐		
accountInfo_InvoicingVat	decimal	☐		
accountInfo_IsActive	boolean	☐		
accountInfo_IsInternal	boolean	☐		
accountInfo_LanguageCode	string	☐		
accountInfo_LanguageGUID	string	☐		
accountInfo_Name	string	☐		
accountInfo_Notes	string	☐		
accountInfo_Number	int64	☐		
accountInfo_OverdueInterest	decimal	☐		
accountInfo_PaymentTerm	int32	☐		
accountInfo_PricelistGUID	string	☐		
accountInfo_ReverseCharge	boolean	☐		
accountInfo_ReverseChargeDescription	string	☐		
accountInfo_UpdatedTS	datetime	☐		
accountInfo_UrlToAccount	string	☐		
companyInfo_AnnualRevenue	int32	☐		
companyInfo_AnnualRevenue2	int64	☐		
companyInfo_Email	string	☐		
companyInfo_Employees	int32	☐		
companyInfo_GUID	string	☐		
companyInfo_HeadOfficeAddressGUID	string	☐		
companyInfo_HierarchyGUID	string	☐		

Name	Data Type	Required	Default Value	Documentation
companyInfo_IndustryGUID	string	☐		
companyInfo_LanguageGUID	string	☐		
companyInfo_Name	string	☐		
companyInfo_TimezoneGUID	string	☐		
companyInfo_VatNumber	string	☐		
companyInfo_Website	string	☐		

Columns of Table Function

The columns of the table function UpdateAccount are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.225 UpdateActivity: Visma Severa Update Activity

Catalog: Severa

Schema: API

Label: Update Activity

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityInfo_AccountGUID- activityInfo_ActivityTypeCode- activityInfo_ActivityTypeGUID- activityInfo_CaseGUID- activityInfo_EndDateTime- activityInfo_GUID- activityInfo_IsAllDay- activityInfo_IsClosed- activityInfo_IsDuration- activityInfo_IsPrivate- activityInfo_IsReadOnly- activityInfo_IsUnassigned- activityInfo_Location- activityInfo_Name- activityInfo_Notes- activityInfo_OwnerUserBusinessUnitCode- activityInfo_OwnerUserCode- activityInfo_OwnerUserFirstName- activityInfo_OwnerUserGUID- activityInfo_OwnerUserLastName- activityInfo_StartDateTime- activityInfo_TaskGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateActivity. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityInfo_AccountGUID	string	☐		
activityInfo_ActivityTypeCode	string	☐		
activityInfo_ActivityTypeGUID	string	☐		
activityInfo_CaseGUID	string	☐		
activityInfo_EndDateTime	datetime	☐		
activityInfo_GUID	string	☐		
activityInfo_IsAllDay	boolean	☐		
activityInfo_IsClosed	boolean	☐		
activityInfo_IsDuration	boolean	☐		
activityInfo_IsPrivate	boolean	☐		
activityInfo_IsReadOnly	boolean	☐		
activityInfo_IsUnassigned	boolean	☐		
activityInfo_Location	string	☐		
activityInfo_Name	string	☐		

Name	Data Type	Required	Default Value	Documentation
activityInfo_Notes	string	☐		
activityInfo_OwnerUserBusinessUnitCode	string	☐		
activityInfo_OwnerUserCode	string	☐		
activityInfo_OwnerUserFirstName	string	☐		
activityInfo_OwnerUserGUID	string	☐		
activityInfo_OwnerUserLastName	string	☐		
activityInfo_StartDateTime	datetime	☐		
activityInfo_TaskGUID	string	☐		

Columns of Table Function

The columns of the table function UpdateActivity are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACTIVITYTYPECODE	string	Activity Type Code		
ACTIVITYTYPEGUID	string	Activity Type GUID		
CASEGUID	string	Project GUID		
ENDDATETIME	datetime	End Date Time		
GUID	string	GUID		
ISALLDAY	boolean	Is All Day		
ISCLOSED	boolean	Is Closed		
ISDURATION	boolean	Is Duration		
ISPRIVATE	boolean	Is Private		
ISREADONLY	boolean	Is Read-only		
ISUNASSIGNED	boolean	Is Unassigned		
LOCATION	string	Location		
NAME	string	Name		
NOTES	string	Notes		
OWNERUSERBUSINESSUNITCODE	string	Owner User Business Unit Code		
OWNERUSERCODE	string	Owner User Code		
OWNERUSERFIRSTNAME	string	Owner User First Name		
OWNERUSERGUID	string	Owner User GUID		
OWNERUSERLASTNAME	string	Owner User Last Name		
STARTDATETIME	datetime	Start Date Time		
TASKGUID	string	Task GUID		

3.1.226 UpdateActivityType: Visma Severa Update Activity Type

Catalog: Severa

Schema: API

Label: Update Activity Type

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: activityTypeInfo_Category- activityTypeInfo_Code- activityTypeInfo_GUID- activityTypeInfo_Icon- activityTypeInfo_IsActive- activityTypeInfo_IsDefault- activityTypeInfo_IsPaidLeave- activityTypeInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateActivityType. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityTypeInfo_Category	string	☐		
activityTypeInfo_Code	string	☐		
activityTypeInfo_GUID	string	☐		
activityTypeInfo_Icon	string	☐		
activityTypeInfo_IsActive	boolean	☐		
activityTypeInfo_IsDefault	boolean	☐		
activityTypeInfo_IsPaidLeave	boolean	☐		
activityTypeInfo_Name	string	☐		

Columns of Table Function

The columns of the table function UpdateActivityType are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CATEGORY	string	Category		
CODE	string	Code		
GUID	string	GUID		

Name	Data Type	Label	Required	Documentation
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPAIDLEAVE	boolean	Is Paid Leave		
NAME	string	Name		

3.1.227 UpdateAddress: Visma Severa Update Address

Catalog: Severa

Schema: API

Label: Update Address

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: addressInfo_Addressline- addressInfo_Addressline2- addressInfo_Addressline3- addressInfo_City- addressInfo_CompanyGUID- addressInfo_CountryCode- addressInfo_CountryGUID- addressInfo_CountryName- addressInfo_CountryRegionGUID- addressInfo_CountryRegionName- addressInfo_Fax- addressInfo_GUID- addressInfo_IsBillingAddress- addressInfo_IsPostalAddress- addressInfo_IsVisitAddress- addressInfo_Phone- addressInfo_PostalCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateAddress. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
addressInfo_Addressline	string	☐		
addressInfo_Addressline2	string	☐		
addressInfo_Addressline3	string	☐		
addressInfo_City	string	☐		
addressInfo_CompanyGUID	string	☐		
addressInfo_CountryCode	string	☐		
addressInfo_CountryGUID	string	☐		

Name	Data Type	Required	Default Value	Documentation
addressInfo_CountryName	string	☐		
addressInfo_CountryRegionGUID	string	☐		
addressInfo_CountryRegionName	string	☐		
addressInfo_Fax	string	☐		
addressInfo_GUID	string	☐		
addressInfo_IsBillingAddress	boolean	☐		
addressInfo_IsPostalAddress	boolean	☐		
addressInfo_IsVisitAddress	boolean	☐		
addressInfo_Phone	string	☐		
addressInfo_PostalCode	string	☐		

Columns of Table Function

The columns of the table function UpdateAddress are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
ADDRESSLINE2	string	Addressline 2		
ADDRESSLINE3	string	Addressline 3		
CITY	string	City		
COMPANYGUID	string	Company GUID		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYNAME	string	Country Name		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
FAX	string	Fax		
GUID	string	GUID		
ISBILLINGADDRESS	boolean	Is Billing Address		
ISPOSTALADDRESS	boolean	Is Postal Address		
ISVISITADDRESS	boolean	Is Visit Address		
PHONE	string	Phone		
POSTALCODE	string	Postal Code		

3.1.228 UpdateBillingAndExpenseForecast: Visma Severa Update Billing and Expense Forecast

Catalog: Severa

Schema: API

Label: Update Billing and Expense Forecast

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: billingAndExpenseForecastInfo_BillingForecast- billingAndExpenseForecastInfo_ExpenseForecast- billingAndExpenseForecastInfo_GUID- billingAndExpenseForecastInfo_InhouseRevenueForecast- billingAndExpenseForecastInfo_Month- billingAndExpenseForecastInfo_Notes- billingAndExpenseForecastInfo_OutsourcingRevenueForecast- billingAndExpenseForecastInfo_Year

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateBillingAndExpenseForecast. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
billingAndExpenseForecastInfo_BillingForecast	decimal	☐		
billingAndExpenseForecastInfo_ExpenseForecast	decimal	☐		
billingAndExpenseForecastInfo_GUID	string	☐		
billingAndExpenseForecastInfo_InhouseRevenueForecast	decimal	☐		
billingAndExpenseForecastInfo_Month	int32	☐		
billingAndExpenseForecastInfo_Notes	string	☐		
billingAndExpenseForecastInfo_OutsourcingRevenueForecast	decimal	☐		
billingAndExpenseForecastInfo_Year	int32	☐		

Columns of Table Function

The columns of the table function UpdateBillingAndExpenseForecast are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
BILLINGFORECAST	decimal	Billing Forecast		
EXPENSEFORECAST	decimal	Expense Forecast		
GUID	string	GUID		
INHOUSEREVENUEFORECAST	decimal	Inhouse Revenue Forecast		
MONTH	int32	Month		
NOTES	string	Notes		
OUTSOURCINGREVENUEFORECAST	decimal	Outsourcing Revenue Forecast		
YEAR	int32	Year		

3.1.229 UpdateBusinessUnit: Visma Severa Update Business Unit

Catalog: Severa

Schema: API

Label: Update Business Unit

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitInfo_Code- businessUnitInfo_CostCenterGUID- businessUnitInfo_CurrencyCode- businessUnitInfo_CurrencyGUID- businessUnitInfo_FormattingCulture- businessUnitInfo_GUID- businessUnitInfo_InsertTS- businessUnitInfo_IsActive- businessUnitInfo_LanguageCode- businessUnitInfo_LanguageGUID- businessUnitInfo_Name- businessUnitInfo_NextInvoiceNumber- businessUnitInfo_OverDueInterest- businessUnitInfo_ParentBusinessUnitGUID- businessUnitInfo_PaymentTerm- businessUnitInfo_ShowLogoOnInvoices- businessUnitInfo_ShowLogoOnPdfReports- businessUnitInfo_TimeZoneGUID- businessUnitInfo_UpdatedTS- businessUnitInfo_VatNumber

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateBusinessUnit. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitInfo_Code	string	☐		
businessUnitInfo_CostCenterGUID	string	☐		
businessUnitInfo_CurrencyCode	string	☐		
businessUnitInfo_CurrencyGUID	string	☐		
businessUnitInfo_FormattingCulture	string	☐		
businessUnitInfo_GUID	string	☐		
businessUnitInfo_InsertTS	datetime	☐		
businessUnitInfo_IsActive	boolean	☐		
businessUnitInfo_LanguageCode	string	☐		
businessUnitInfo_LanguageGUID	string	☐		
businessUnitInfo_Name	string	☐		
businessUnitInfo_NextInvoiceNumber	int32	☐		
businessUnitInfo_OverDueInterest	decimal	☐		
businessUnitInfo_ParentBusinessUnitGUID	string	☐		
businessUnitInfo_PaymentTerm	int32	☐		
businessUnitInfo_Show LogoOnInvoices	boolean	☐		
businessUnitInfo_Show LogoOnPdf Reports	boolean	☐		
businessUnitInfo_TimeZoneGUID	string	☐		
businessUnitInfo_UpdatedTS	datetime	☐		
businessUnitInfo_VatNumber	string	☐		

Columns of Table Function

The columns of the table function UpdateBusinessUnit are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
COSTCENTERGUID	string	Cost Center GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
FORMATTINGCULTURE	string	Formatting Culture		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NEXTINVOICENUMBER	int32	Next Invoice Number		

Name	Data Type	Label	Required	Documentation
OVERDUEINTEREST	decimal	Overdue Interest		
PARENTBUSINESSUNITGUID	string	Parent Business Unit GUID		
PAYMENTTERM	int32	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	string	VAT Number		

3.1.230 UpdateCase: Visma Severa Update Project

Catalog: Severa

Schema: API

Label: Update Project

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseInfo_AccountGUID- caseInfo_AccountName- caseInfo_AccountNumber- caseInfo_BillingAddress- caseInfo_BillingAddressGUID- caseInfo_BillingInternalNotes- caseInfo_BillingNotesBeforeGrid- caseInfo_BusinessUnitGUID- caseInfo_BusinessUnitName- caseInfo_BusinessUnitNumber- caseInfo_CaseCompanyCurrencyGUID- caseInfo_CaseNumber- caseInfo_CaseOwnerUserCode- caseInfo_CaseOwnerUserGUID- caseInfo_ClosedDate- caseInfo_CompletionEstimate- caseInfo_ContactGUID- caseInfo_CostCenterGUID- caseInfo_CostCenterNumber- caseInfo_CurrencyGUID- caseInfo_CurrencyShortform- caseInfo_CurrencyShortformOfCaseCompanyCurrency- caseInfo_CurrentCaseStatusDescription- caseInfo_CurrentCaseStatusGUID- caseInfo_CurrentCaseStatusInsertTs- caseInfo_DeadlineDate- caseInfo_Description- caseInfo_ExpectedValue- caseInfo_GUID- caseInfo_IncludeExceptions- caseInfo_IncludeOverTime- caseInfo_InsertTs- caseInfo_InternalName- caseInfo_InvoiceTemplateGUID- caseInfo_IsClosed- caseInfo_IsDailyAllowanceBillable- caseInfo_IsInternal- caseInfo_IsMileageBillable- caseInfo_IsOtherTravelExpensesBillable- caseInfo_LatestEstimationDate- caseInfo_LeadRating- caseInfo_LeadSourceGUID- caseInfo_Name- caseInfo_OrderNumber- caseInfo_OurReference- caseInfo_OverdueInterest- caseInfo_PaymentTerm- caseInfo_PricelistGUID- caseInfo_PricingRule- caseInfo_PricingType- caseInfo_Priority- caseInfo_Probability- caseInfo_RootTaskGUID- caseInfo_SalesCloseDate- caseInfo_SalesPersonUserCode- caseInfo_SalesPersonUserGUID- caseInfo_SalesProcessGUID- caseInfo_SalesStatusGUID- caseInfo_SharedToExtranet- caseInfo_StartDate- caseInfo_UpdatedTs- caseInfo_UrlToCase- caseInfo_UseDefaultProductsInWorkTimeEntry- caseInfo_UseDefaultWorkTypes- caseInfo_YourReference

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateCase. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseInfo_AccountGUID	string	☐		
caseInfo_AccountName	string	☐		
caseInfo_AccountNumber	int64	☐		
caseInfo_BillingAddress	string	☐		
caseInfo_BillingAddressGUID	string	☐		
caseInfo_BillingInternalNotes	string	☐		
caseInfo_BillingNotesBeforeGrid	string	☐		
caseInfo_BusinessUnitGUID	string	☐		
caseInfo_BusinessUnitName	string	☐		
caseInfo_BusinessUnitNumber	string	☐		
caseInfo_CaseCompanyCurrencyGUID	string	☐		
caseInfo_CaseNumber	int64	☐		
caseInfo_CaseOwnerUserCode	string	☐		
caseInfo_CaseOwnerUserGUID	string	☐		
caseInfo_ClosedDate	datetime	☐		
caseInfo_CompletionEstimate	int32	☐		
caseInfo_ContactGUID	string	☐		
caseInfo_CostCenterGUID	string	☐		
caseInfo_CostCenterNumber	string	☐		
caseInfo_CurrencyGUID	string	☐		
caseInfo_CurrencyShortform	string	☐		
caseInfo_CurrencyShortformOfCaseCompanyCurrency	string	☐		
caseInfo_CurrentCaseStatusDescription	string	☐		
caseInfo_CurrentCaseStatusGUID	string	☐		
caseInfo_CurrentCaseStatusInsertTs	datetime	☐		
caseInfo_DeadlineDate	datetime	☐		
caseInfo_Description	string	☐		

Name	Data Type	Required	Default Value	Documentation
caseInfo_ExpectedValue	decimal	☐		
caseInfo_GUID	string	☐		
caseInfo_IncludeExceptions	boolean	☐		
caseInfo_IncludeOverTime	boolean	☐		
caseInfo_InsertTs	datetime	☐		
caseInfo_InternalName	string	☐		
caseInfo_InvoiceTemplateGUID	string	☐		
caseInfo_IsClosed	boolean	☐		
caseInfo_IsDailyAllow anceBillable	boolean	☐		
caseInfo_IsInternal	boolean	☐		
caseInfo_IsMileageBillable	boolean	☐		
caseInfo_IsOtherTravelExpensesBillable	boolean	☐		
caseInfo_LatestEstimationDate	datetime	☐		
caseInfo_LeadRating	int32	☐		
caseInfo_LeadSourceGUID	string	☐		
caseInfo_Name	string	☐		
caseInfo_OrderNumber	string	☐		
caseInfo_OurReference	string	☐		
caseInfo_OverdueInterest	decimal	☐		
caseInfo_PaymentTerm	int32	☐		
caseInfo_PricelistGUID	string	☐		
caseInfo_PricingRule	string	☐		
caseInfo_PricingType	string	☐		
caseInfo_Priority	int32	☐		
caseInfo_Probability	int32	☐		
caseInfo_RootTaskGUID	string	☐		
caseInfo_SalesCloseDate	datetime	☐		
caseInfo_SalesPersonUserCode	string	☐		
caseInfo_SalesPersonUserGUID	string	☐		
caseInfo_SalesProcessGUID	string	☐		
caseInfo_SalesStatusGUID	string	☐		
caseInfo_SharedToExtranet	boolean	☐		
caseInfo_StartDate	datetime	☐		
caseInfo_UpdatedTs	datetime	☐		
caseInfo_UrlToCase	string	☐		
caseInfo_UseDefaultProductsInWorkTimeEntry	boolean	☐		
caseInfo_UseDefaultWorkTypes	boolean	☐		
caseInfo_YourReference	string	☐		

Columns of Table Function

The columns of the table function UpdateCase are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		
CASEOWNERUSERGUID	string	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	string	Ddescription		
EXPECTEDVALUE	decimal	Expected Value		
GUID	string	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		

Name	Data Type	Label	Required	Documentation
INSERTTS	datetime	Inserted		
INTERNALNAME	string	Internal Name		
INVOICETEMPLATEGUID	string	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	int32	Lead Rating		
LEADSOURCEGUID	string	Lead Source GUID		
NAME	string	Name		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
PRICINGRULE	string	Pricing Rule		
PRICINGTYPE	string	Pricing Type		
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOTTASKGUID	string	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	string	Sales Person User Code		
SALESPERSONUSERGUID	string	Sales Person User GUID		
SALESPROCESSGUID	string	Sales Process GUID		
SALESSTATUSGUID	string	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	string	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	string	Your Reference		

3.1.231 UpdateCaseStatus: Visma Severa Update Project Status

Catalog: Severa

Schema: API

Label: Update Project Status

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: caseStatusInfo_GUID- caseStatusInfo_Icon- caseStatusInfo_IsActive- caseStatusInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateCaseStatus. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
caseStatusInfo_GUID	string	☐		
caseStatusInfo_Icon	string	☐		
caseStatusInfo_IsActive	boolean	☐		
caseStatusInfo_Name	string	☐		

Columns of Table Function

The columns of the table function UpdateCaseStatus are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.232 UpdateContact: Visma Severa Update Contact

Catalog: Severa

Schema: API

Label: Update Contact

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: contactInfo_AccountGUID- contactInfo_AddressGUID- contactInfo_CommunicationMethods- contactInfo_ContactRoleGUID- contactInfo_DateOfBirth- contactInfo_Description- contactInfo_FirstName- contactInfo_GUID- contactInfo_IsActive- contactInfo_JobTitle- contactInfo_Keywords- contactInfo_LanguageGUID- contactInfo_LastName- contactInfo_Salutation- contactInfo_SatisfactionLevel- contactInfo_TimeZoneGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateContact. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
contactInfo_AccountGUID	string	☐		
contactInfo_AddressGUID	string	☐		
contactInfo_CommunicationMethods	string	☐		
contactInfo_ContactRoleGUID	string	☐		
contactInfo_DateOfBirth	datetime	☐		
contactInfo_Description	string	☐		
contactInfo_FirstName	string	☐		
contactInfo_GUID	string	☐		
contactInfo_IsActive	boolean	☐		
contactInfo_JobTitle	string	☐		
contactInfo_Keywords	string	☐		
contactInfo_LanguageGUID	string	☐		
contactInfo_LastName	string	☐		
contactInfo_Salutation	string	☐		
contactInfo_SatisfactionLevel	int32	☐		

Name	Data Type	Required	Default Value	Documentation
contactInfo_TimeZoneGUID	string	☐		

Columns of Table Function

The columns of the table function UpdateContact are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ADDRESSGUID	string	Address GUID		
COMMUNICATIONMETHODS	string			
CONTACTROLEGUID	string	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	string	Ddescription		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	string	Job Title		
KEYWORDS	string			
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
SALUTATION	string	Salutation		
SATISFACTIONLEVEL	int32	Satisfaction Level		
TIMEZONEGUID	string	Timezone GUID		

3.1.233 UpdateCustomer: Visma Severa Update Customer

Catalog: Severa

Schema: API

Label: Update Customer

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: customer_AccountGroupGUIDs- customer_AccountOwnerUserGUID- customer_AccountRating- customer_AnnualRevenue- customer_AnnualRevenue2- customer_CompanyGUID- customer_CurrencyCode- customer_CurrencyGUID- customer_Email- customer_Employees- customer_GUID- customer_HeadOfficeAddressGUID- customer_IndustryGUID- customer_InsertTS- customer_InvoicingVat- customer_IsActive- customer_IsInternal- customer_LanguageCode- customer_LanguageGUID- customer_Name- customer_Notes- customer_Number- customer_OverdueInterest- customer_PaymentTerm- customer_PricelistGUID- customer_ReverseCharge- customer_ReverseChargeDescription- customer_TimezoneGUID- customer_UpdatedTS- customer_UrlToAccount-

customer_VatNumber- customer_Website- customer_eInvoiceAddress- customer_eInvoiceOperator

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateCustomer. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
customer_AccountGroupGUIDs	string	☐		
customer_AccountOwnerUserGUID	string	☐		
customer_AccountRating	int32	☐		
customer_AnnualRevenue	int32	☐		
customer_AnnualRevenue2	int64	☐		
customer_CompanyGUID	string	☐		
customer_CurrencyCode	string	☐		
customer_CurrencyGUID	string	☐		
customer_eInvoiceAddress	string	☐		
customer_eInvoiceOperator	string	☐		
customer_Email	string	☐		
customer_Employees	int32	☐		
customer_GUID	string	☐		
customer_HeadOfficeAddressGUID	string	☐		
customer_IndustryGUID	string	☐		
customer_InsertTS	datetime	☐		
customer_InvoicingVat	decimal	☐		
customer_IsActive	boolean	☐		
customer_IsInternal	boolean	☐		
customer_LanguageCode	string	☐		
customer_LanguageGUID	string	☐		
customer_Name	string	☐		
customer_Notes	string	☐		
customer_Number	int64	☐		

Name	Data Type	Required	Default Value	Documentation
customer_OverdueInterest	decimal	☐		
customer_PaymentTerm	int32	☐		
customer_PricelistGUID	string	☐		
customer_ReverseCharge	boolean	☐		
customer_ReverseChargeDescription	string	☐		
customer_TimezoneGUID	string	☐		
customer_UpdatedTS	datetime	☐		
customer_UrlToAccount	string	☐		
customer_VatNumber	string	☐		
customer_Website	string	☐		

Columns of Table Function

The columns of the table function UpdateCustomer are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
ANNUALREVENUE	int32	Annual Revenue		
ANNUALREVENUE2	int64			
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
EMAIL	string	Email		
EMPLOYEES	int32	Employees		
GUID	string	GUID		
HEADOFFICEADDRESSGUID	string	Head Office Address GUID		
INDUSTRYGUID	string	Industry GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		

Name	Data Type	Label	Required	Documentation
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		
VATNUMBER	string	VAT Number		
WEBSITE	string	Website		

3.1.234 UpdateEmployment: Visma Severa Update Employment

Catalog: Severa

Schema: API

Label: Update Employment

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: employmentInfo_CompanyBusinessUnitCode- employmentInfo_CompanyBusinessUnitGUID- employmentInfo_CompanyBusinessUnitName- employmentInfo_DailyHours- employmentInfo_EndDate- employmentInfo_GUID- employmentInfo_HourCost- employmentInfo_IsOvertimeAllowed- employmentInfo_StartDate- employmentInfo_Title

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateEmployment. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
employmentInfo_CompanyBusinessUnitCode	string	☐		
employmentInfo_CompanyBusinessUnitGUID	string	☐		
employmentInfo_CompanyBusinessUnitName	string	☐		
employmentInfo_DailyHours	decimal	☐		
employmentInfo_EndDate	datetime	☐		
employmentInfo_GUID	string	☐		
employmentInfo_HourCost	decimal	☐		
employmentInfo_IsOvertimeAllowed	boolean	☐		
employmentInfo_StartDate	datetime	☐		
employmentInfo_Title	string	☐		

Columns of Table Function

The columns of the table function UpdateEmployment are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
COMPANYBUSINESSUNITCODE	string	Company Business Unit Code		
COMPANYBUSINESSUNITGUID	string	Company Business Unit GUID		
COMPANYBUSINESSUNITNAME	string	Company Business Unit Name		
DAILYHOURS	decimal	Daily Hours		
ENDDATE	datetime	End Date		
GUID	string	GUID		
HOURCOST	decimal	Hour Cost		
ISOVERTIMEALLOWED	boolean	Is Overtime Allowed		
STARTDATE	datetime	Start Date		
TITLE	string	Title		

3.1.235 UpdateHourEntry: Visma Severa Update Hour Entry

Catalog: Severa

Schema: API

Label: Update Hour Entry

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: hourEntryInfo_AccountGUID- hourEntryInfo_CaseBusinessUnitCode- hourEntryInfo_CaseCostCenterIdentifier- hourEntryInfo_CaseGUID- hourEntryInfo_CaseName- hourEntryInfo_CaseNumber- hourEntryInfo_CostCurrencyShortform- hourEntryInfo_CostCurrencyShortformOfInvoicingCurrency- hourEntryInfo_CostCurrencyShortformOfOrganizationCurrency- hourEntryInfo_CostCurrencyShortformOfUserCompanyCurrency- hourEntryInfo_Description- hourEntryInfo_EndTime- hourEntryInfo_EventDate- hourEntryInfo_GUID- hourEntryInfo_InvoiceDescription- hourEntryInfo_InvoiceGUID- hourEntryInfo_InvoiceQuantity- hourEntryInfo_InvoiceRowGUID- hourEntryInfo_IsApproved- hourEntryInfo_IsBillable- hourEntryInfo_IsProductive- hourEntryInfo_IsReadOnly- hourEntryInfo_OverTimeGUID- hourEntryInfo_OvertimeCode- hourEntryInfo_PhaseGUID- hourEntryInfo_PriceCurrencyShortformOfCaseCompanyCurrency- hourEntryInfo_PriceCurrencyShortformOfInvoicingCurrency- hourEntryInfo_PriceCurrencyShortformOfOrganizationCurrency- hourEntryInfo_PriceCurrencyShortformOfUserCompanyCurrency- hourEntryInfo_Quantity- hourEntryInfo_StartTime- hourEntryInfo_TaskGUID- hourEntryInfo_UnitCost- hourEntryInfo_UnitCostInInvoicingCurrency- hourEntryInfo_UnitCostInOrganizationCurrency- hourEntryInfo_UnitCostInUserCompanyCurrency- hourEntryInfo_UnitPrice- hourEntryInfo_UnitPriceInBaseCurrency- hourEntryInfo_UnitPriceInCaseCompanyCurrency- hourEntryInfo_UnitPriceInUserCompanyCurrency- hourEntryInfo_UseInvoiceQuantity- hourEntryInfo_UserBusinessUnitCode- hourEntryInfo_UserCode- hourEntryInfo_UserFirstName- hourEntryInfo_UserGUID- hourEntryInfo_UserLastName- hourEntryInfo_WorkTypeCode- hourEntryInfo_WorkTypeGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateHourEntry. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
hourEntryInfo_AccountGUID	string	☐		
hourEntryInfo_CaseBusinessUnitCode	string	☐		
hourEntryInfo_CaseCostCenterIdentifier	string	☐		
hourEntryInfo_CaseGUID	string	☐		
hourEntryInfo_CaseName	string	☐		
hourEntryInfo_CaseNumber	int64	☐		
hourEntryInfo_CostCurrencyShortform	string	☐		

Name	Data Type	Required	Default Value	Documentation
hourEntryInfo_CostCurrencyShortformOfInvoicingCurrency	string	☐		
hourEntryInfo_CostCurrencyShortformOfOrganizationCurrency	string	☐		
hourEntryInfo_CostCurrencyShortformOfUserCompanyCurrency	string	☐		
hourEntryInfo_Description	string	☐		
hourEntryInfo_EndTime	datetime	☐		
hourEntryInfo_EventDate	datetime	☐		
hourEntryInfo_GUID	string	☐		
hourEntryInfo_InvoiceDescription	string	☐		
hourEntryInfo_InvoiceGUID	string	☐		
hourEntryInfo_InvoiceQuantity	decimal	☐		
hourEntryInfo_InvoiceRow GUID	string	☐		
hourEntryInfo_IsApproved	boolean	☐		
hourEntryInfo_IsBillable	boolean	☐		
hourEntryInfo_IsProductive	boolean	☐		
hourEntryInfo_IsReadOnly	boolean	☐		
hourEntryInfo_OvertimeCode	string	☐		
hourEntryInfo_OverTimeGUID	string	☐		
hourEntryInfo_PhaseGUID	string	☐		
hourEntryInfo_PriceCurrencyShortformOfCaseCompanyCurrency	string	☐		
hourEntryInfo_PriceCurrencyShortformOfInvoicingCurrency	string	☐		
hourEntryInfo_PriceCurrencyShortformOfOrganizationCurrency	string	☐		
hourEntryInfo_PriceCurrencyShortformOfUserCompanyCurrency	string	☐		
hourEntryInfo_Quantity	decimal	☐		
hourEntryInfo_StartTime	datetime	☐		
hourEntryInfo_TaskGUID	string	☐		
hourEntryInfo_UnitCost	decimal	☐		
hourEntryInfo_UnitCostInInvoicingCurrency	decimal	☐		
hourEntryInfo_UnitCostInOrganizationCurrency	decimal	☐		
hourEntryInfo_UnitCostInUserCompanyCurrency	decimal	☐		
hourEntryInfo_UnitPrice	decimal	☐		
hourEntryInfo_UnitPriceInBaseCurrency	decimal	☐		
hourEntryInfo_UnitPriceInCaseCompanyCurrency	decimal	☐		

Name	Data Type	Required	Default Value	Documentation
hourEntryInfo_UnitPriceInUserCompanyCurrency	decimal	☐		
hourEntryInfo_UseInvoiceQuantity	boolean	☐		
hourEntryInfo_UserBusinessUnitCode	string	☐		
hourEntryInfo_UserCode	string	☐		
hourEntryInfo_UserFirstName	string	☐		
hourEntryInfo_UserGUID	string	☐		
hourEntryInfo_UserLastName	string	☐		
hourEntryInfo_WorkTypeCode	string	☐		
hourEntryInfo_WorkTypeGUID	string	☐		

Columns of Table Function

The columns of the table function UpdateHourEntry are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		

Name	Data Type	Label	Required	Documentation
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.236 UpdatePhase: Visma Severa Update Phase

Catalog: Severa

Schema: API

Label: Update Phase

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: phaseInfo_CaseGUID- phaseInfo_Code- phaseInfo_Deadline- phaseInfo_Description- phaseInfo_FlatRate_AdditionalHoursUnitPrice- phaseInfo_FlatRate_AreAdditionalHoursBillable- phaseInfo_FlatRate_Hours- phaseInfo_FlatRate_Price- phaseInfo_GUID- phaseInfo_IsCompleted- phaseInfo_IsLocked- phaseInfo_Name- phaseInfo_OriginalDeadline- phaseInfo_OriginalPlannedStartDate- phaseInfo_OriginalWorkEstimate- phaseInfo_OwnerUserGUID- phaseInfo_ParentPhaseGUID- phaseInfo_PlannedStartDate- phaseInfo_PricePerHour- phaseInfo_SortOrder- phaseInfo_WorkEstimate- phaseInfo_WorkTypeGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdatePhase. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
phaseInfo_CaseGUID	string	☐		
phaseInfo_Code	string	☐		
phaseInfo_Deadline	datetime	☐		
phaseInfo_Description	string	☐		
phaseInfo_FlatRate_AdditionalHoursUnitPrice	decimal	☐		
phaseInfo_FlatRate_AreAdditionalHoursBillable	boolean	☐		
phaseInfo_FlatRate_Hours	decimal	☐		
phaseInfo_FlatRate_Price	decimal	☐		
phaseInfo_GUID	string	☐		
phaseInfo_IsCompleted	boolean	☐		
phaseInfo_IsLocked	boolean	☐		
phaseInfo_Name	string	☐		

Name	Data Type	Required	Default Value	Documentation
phaseInfo_OriginalDeadline	datetime	☐		
phaseInfo_OriginalPlannedStartDate	datetime	☐		
phaseInfo_OriginalWorkEstimate	decimal	☐		
phaseInfo_OwnerUserGUID	string	☐		
phaseInfo_ParentPhaseGUID	string	☐		
phaseInfo_PlannedStartDate	datetime	☐		
phaseInfo_PricePerHour	decimal	☐		
phaseInfo_SortOrder	int32	☐		
phaseInfo_WorkEstimate	decimal	☐		
phaseInfo_WorkTypeGUID	string	☐		

Columns of Table Function

The columns of the table function UpdatePhase are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CASEGUID	string	Project GUID		
CODE	string	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	string	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	decimal	Flat Rate Additional Hours Unit Price		
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	decimal	Flat Rate Hours		
FLATRATE_PRICE	decimal	Flat Rate Price		
GUID	string	GUID		
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	decimal	Original Work Estimate		
OWNERUSERGUID	string	Owner User GUID		
PARENTPHASEGUID	string	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	decimal	Price per Hour		
SORTORDER	int32	Sort Order		
WORKESTIMATE	decimal	Work Estimate		

Name	Data Type	Label	Required	Documentation
WORKTYPEGUID	string	Work Type GUID		

3.1.237 UpdateProduct: Visma Severa Update Product

Catalog: Severa

Schema: API

Label: Update Product

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: productInfo_Code- productInfo_Description- productInfo_GUID- productInfo_IsActive- productInfo_MeasurementUnit- productInfo_Name- productInfo_Type- productInfo_UnitCost- productInfo_UnitPrice- productInfo_UseInWorkTimeEntry- productInfo_VAT- productInfo_ProductCategoryGUID- productInfo_SalesAccountGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateProduct. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
productInfo_Code	string	☐		
productInfo_Description	string	☐		
productInfo_GUID	string	☐		
productInfo_IsActive	boolean	☐		
productInfo_MeasurementUnit	string	☐		
productInfo_Name	string	☐		
productInfo_ProductCategoryGUID	string	☐		
productInfo_SalesAccountGUID	string	☐		
productInfo_Type	string	☐		
productInfo_UnitCost	decimal	☐		
productInfo_UnitPrice	decimal	☐		
productInfo_UseInWorkTimeEntry	boolean	☐		
productInfo_VAT	decimal	☐		

Columns of Table Function

The columns of the table function UpdateProduct are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
PRODUCTCATEGORYGUID	string	Product Category GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.238 UpdateProductCategory: Visma Severa Update Product Category

Catalog: Severa

Schema: API

Label: Update Product Category

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: productCategoryInfo_Code- productCategoryInfo_GUID- productCategoryInfo_IsActive- productCategoryInfo_IsDefault- productCategoryInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateProductCategory. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
productCategoryInfo_Code	string	☐		
productCategoryInfo_GUID	string	☐		
productCategoryInfo_IsActive	boolean	☐		
productCategoryInfo_IsDefault	boolean	☐		
productCategoryInfo_Name	string	☐		

Columns of Table Function

The columns of the table function UpdateProductCategory are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		

3.1.239 UpdateResource: Visma Severa Update Resource

Catalog: Severa

Schema: API

Label: Update Resource

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: resourceInfo_GUID- resourceInfo_IsActive- resourceInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateResource. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
resourceInfo_GUID	string	☐		
resourceInfo_IsActive	boolean	☐		
resourceInfo_Name	string	☐		

Columns of Table Function

The columns of the table function UpdateResource are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.240 UpdateResourceAllocation: Visma Severa Update Resource Allocation

Catalog: Severa

Schema: API

Label: Update Resource Allocation

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: resourceAllocationInfo_AllocatedHours- resourceAllocationInfo_AllocatedPercentage- resourceAllocationInfo_CaseGUID- resourceAllocationInfo_EndDate- resourceAllocationInfo_GUID- resourceAllocationInfo_PhaseGUID- resourceAllocationInfo_StartDate- resourceAllocationInfo_UserGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateResourceAllocation. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
resourceAllocationInfo_AllocatedHours	decimal	☐		
resourceAllocationInfo_AllocatedPercentage	int32	☐		
resourceAllocationInfo_CaseGUID	string	☐		
resourceAllocationInfo_EndDate	datetime	☐		
resourceAllocationInfo_GUID	string	☐		
resourceAllocationInfo_PhaseGUID	string	☐		
resourceAllocationInfo_StartDate	datetime	☐		
resourceAllocationInfo_UserGUID	string	☐		

Columns of Table Function

The columns of the table function UpdateResourceAllocation are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.241 UpdateTravelExpense: Visma Severa Update Travel Expense

Catalog: Severa

Schema: API

Label: Update Travel Expense

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: travelExpenseInfo_Code- travelExpenseInfo_Description- travelExpenseInfo_GUID- travelExpenseInfo_IsActive- travelExpenseInfo_MeasurementUnit- travelExpenseInfo_Name- travelExpenseInfo_Type- travelExpenseInfo_UnitCost- travelExpenseInfo_UnitPrice- travelExpenseInfo_UseInWorkTimeEntry- travelExpenseInfo_VAT- travelExpenseInfo_CostCurrencyGUID- travelExpenseInfo_ExpenseClass

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateTravelExpense. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
travelExpenseInfo_Code	string	☐		
travelExpenseInfo_CostCurrencyGUID	string	☐		
travelExpenseInfo_Description	string	☐		
travelExpenseInfo_ExpenseClass	string	☐		
travelExpenseInfo_GUID	string	☐		
travelExpenseInfo_IsActive	boolean	☐		
travelExpenseInfo_MeasurementUnit	string	☐		
travelExpenseInfo_Name	string	☐		
travelExpenseInfo_Type	string	☐		
travelExpenseInfo_UnitCost	decimal	☐		
travelExpenseInfo_UnitPrice	decimal	☐		
travelExpenseInfo_UseInWorkTimeEntry	boolean	☐		
travelExpenseInfo_VAT	decimal	☐		

Columns of Table Function

The columns of the table function UpdateTravelExpense are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
COSTCURRENCYGUID	string	Cost Currency GUID		
DESCRIPTION	string	Ddescription		
EXPENSECLASS	string	Expense Class		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		

Name	Data Type	Label	Required	Documentation
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.242 UpdateUser: Visma Severa Update User

Catalog: Severa

Schema: API

Label: Update User

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: userInfo_Addressline- userInfo_BankAccountNumber- userInfo_BusinessUnitCode- userInfo_BusinessUnitGUID- userInfo_BusinessUnitName- userInfo_Code- userInfo_CountryCode- userInfo_CountryGUID- userInfo_CountryRegionGUID- userInfo_CountryRegionName- userInfo_Culture- userInfo_DefaultActivityTypeGUID- userInfo_Email- userInfo_FirstName- userInfo_GUID- userInfo_IsActive- userInfo_LanguageCode- userInfo_LanguageGUID- userInfo_LastName- userInfo_LicenseType- userInfo_Notes- userInfo_OrganizationGUID- userInfo_Phone- userInfo_PhotoFileGUID- userInfo_PostOffice- userInfo_PostalCode- userInfo_ProfileGUID- userInfo_Salutation- userInfo_SendDailyMail- userInfo_SendWeeklyMail- userInfo_SocialSecurityNumber- userInfo_SuperiorUserGUID- userInfo_TimezoneGUID- userInfo_Title- userInfo_WorktypeGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateUser. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userInfo_Addressline	string	☐		

Name	Data Type	Required	Default Value	Documentation
userInfo_BankAccountNumber	string	☐		
userInfo_BusinessUnitCode	string	☐		
userInfo_BusinessUnitGUID	string	☐		
userInfo_BusinessUnitName	string	☐		
userInfo_Code	string	☐		
userInfo_CountryCode	string	☐		
userInfo_CountryGUID	string	☐		
userInfo_CountryRegionGUID	string	☐		
userInfo_CountryRegionName	string	☐		
userInfo_Culture	string	☐		
userInfo_DefaultActivityTypeGUID	string	☐		
userInfo_Email	string	☐		
userInfo_FirstName	string	☐		
userInfo_GUID	string	☐		
userInfo_IsActive	boolean	☐		
userInfo_LanguageCode	string	☐		
userInfo_LanguageGUID	string	☐		
userInfo_LastName	string	☐		
userInfo_LicenseType	string	☐		
userInfo_Notes	string	☐		
userInfo_OrganizationGUID	string	☐		
userInfo_Phone	string	☐		
userInfo_PhotoFileGUID	string	☐		
userInfo_PostalCode	string	☐		
userInfo_PostOffice	string	☐		
userInfo_ProfileGUID	string	☐		
userInfo_Salutation	string	☐		
userInfo_SendDailyMail	boolean	☐		
userInfo_SendWeeklyMail	boolean	☐		
userInfo_SocialSecurityNumber	string	☐		
userInfo_SuperiorUserGUID	string	☐		
userInfo_TimezoneGUID	string	☐		
userInfo_Title	string	☐		
userInfo_WorktypeGUID	string	☐		

Columns of Table Function

The columns of the table function UpdateUser are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.243 UpdateUserRightsByUserGUID: Visma Severa Update User Rights by User GUID

Catalog: Severa

Schema: API

Label: Update User Rights by User GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: guid- rights_API- rights_Rights_Account- rights_Rights_AccountDelete- rights_Rights_Administrator- rights_Rights_Case- rights_Rights_CaseDelete- rights_Rights_CaseOwnerWorkHourApproval- rights_Rights_ScheduleJobs- rights_Rights_Sharing- rights_Rights_TravelReimbursement- rights_Rights_Users- rights_Rights_WorkHourApproval

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateUserRightsByUserGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
guid	string	☐		
rights_API	boolean	☐		
rights_Rights_Account	string	☐		
rights_Rights_AccountDelete	boolean	☐		
rights_Rights_Administrator	string	☐		
rights_Rights_Case	string	☐		
rights_Rights_CaseDelete	boolean	☐		
rights_Rights_CaseOwnerWorkHourApproval	boolean	☐		
rights_Rights_ScheduleJobs	string	☐		
rights_Rights_Sharing	boolean	☐		
rights_Rights_TravelReimbursement	string	☐		
rights_Rights_Users	string	☐		
rights_Rights_WorkHourApproval	string	☐		

Columns of Table Function

The columns of the table function UpdateUserRightsByUserGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
API	boolean	API		
RIGHTS_ACCOUNT	string			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	string			
RIGHTS_CASE	string			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOURAPPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	string			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	string			
RIGHTS_USERS	string			
RIGHTS_WORKHOURAPPROVAL	string			

3.1.244 UpdateWorkType: Visma Severa Update Work Type

Catalog: Severa

Schema: API

Label: Update Work Type

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: workTypeInfo_Code- workTypeInfo_GUID- workTypeInfo_IsActive- workTypeInfo_IsDefault- workTypeInfo_IsProductive- workTypeInfo_Name- workTypeInfo_SalesAccountGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateWorkType. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
workTypeInfo_Code	string	☐		
workTypeInfo_GUID	string	☐		
workTypeInfo_IsActive	boolean	☐		
workTypeInfo_IsDefault	boolean	☐		
workTypeInfo_IsProductive	boolean	☐		
workTypeInfo_Name	string	☐		
workTypeInfo_SalesAccountGUID	string	☐		

Columns of Table Function

The columns of the table function UpdateWorkType are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	string	Name		
SALESACCOUNTGUID	string	Sales Account GUID		

3.1.245 UserByCode: Visma Severa User by Code

Catalog: Severa

Schema: API

Label: User by Code

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: userCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UserByCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userCode	string	☐		

Columns of Table Function

The columns of the table function UserByCode are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		

Name	Data Type	Label	Required	Documentation
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.246 UserByGUID: Visma Severa User by GUID

Catalog: Severa

Schema: API

Label: User by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UserByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userGUID	string	☐		

Columns of Table Function

The columns of the table function UserByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.247 UserByName: Visma Severa User by Name

Catalog: Severa

Schema: API

Label: User by Name

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: firstName- lastName

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UserByName. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
firstName	string	☐		
lastName	string	☐		

Columns of Table Function

The columns of the table function UserByName are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		

Name	Data Type	Label	Required	Documentation
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.248 UserRightsByUserGUID: Visma Severa User Rights by User GUID

Catalog: Severa

Schema: API

Label: User Rights by User GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: guid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UserRightsByUserGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a

pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
guid	string	☐		

Columns of Table Function

The columns of the table function UserRightsByUserGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
API	boolean	API		
RIGHTS_ACCOUNT	string			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	string			
RIGHTS_CASE	string			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOURAPPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	string			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	string			
RIGHTS_USERS	string			
RIGHTS_WORKHOURAPPROVAL	string			

3.1.249 Users: Visma Severa Users

Catalog: Severa

Schema: API

Label: Users

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- includeInactive

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Users. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
includeInactive	boolean	☐		

Columns of Table Function

The columns of the table function Users are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			

Name	Data Type	Label	Required	Documentation
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.250 UsersByAccessRightsProfileGUID: Visma Severa Users by Access Rights Profile GUID

Catalog: Severa

Schema: API

Label: Users by Access Rights Profile GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: guid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersByAccessRightsProfileGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
guid	string	☐		

Columns of Table Function

The columns of the table function UsersByAccessRightsProfileGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		

Name	Data Type	Label	Required	Documentation
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.251 UsersByGUIDs: Visma Severa Users by GUIDs

Catalog: Severa

Schema: API

Label: Users by GUIDs

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: userGUIDs

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersByGUIDs. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userGUIDs	string	☐		

Columns of Table Function

The columns of the table function UsersByGUIDs are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.252 UsersByUserKeywords: Visma Severa Users by User Keywords

Catalog: Severa

Schema: API

Label: Users by User Keywords

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- userKeywords

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersByUserKeywords. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
userKeywords	string	☐		

Columns of Table Function

The columns of the table function UsersByUserKeywords are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		

Name	Data Type	Label	Required	Documentation
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.253 UsersChangedSince: Visma Severa Users Changed since

Catalog: Severa

Schema: API

Label: Users Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
options	int32	☐		
startDateTime	datetime	☐		

Columns of Table Function

The columns of the table function UsersChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		

Name	Data Type	Label	Required	Documentation
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.254 UserTags: Visma Severa User Tags

Catalog: Severa

Schema: API

Label: User Tags

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table UserTags are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CONTEXT	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyword		
WEIGHT	int32	Weight		

3.1.255 Version: Visma Severa Version

Catalog: Severa

Schema: API

Label: Version

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table Version are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
GETVERSIONRESULT	string			

3.1.256 WorkDayInfoByDateRangeAndUserGUID: Visma Severa Workday Information by Date Range and User GUID

Catalog: Severa

Schema: API

Label: Workday Information by Date Range and User GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: businessUnitGUID- startdate- enddate- userGUID- includeUnPaidAbsenses- includeInfoForInactiveUser

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WorkDayInfoByDateRangeAndUserGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID	string	☐		
enddate	datetime	☐		
includeInfoForInactiveUser	boolean	☐		
includeUnPaidAbsenses	boolean	☐		
startdate	datetime	☐		
userGUID	string	☐		

Columns of Table Function

The columns of the table function WorkDayInfoByDateRangeAndUserGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACTIVITIES	string			
DATE	datetime	Date		
ENTRIES	string			
EXPECTEDHOURS	decimal	Expected Hours		
OVERTIMEALLOWED	boolean	Overtime Allowed		

Name	Data Type	Label	Required	Documentation
READONLY	boolean	Read-only		
READY	boolean	Ready		
TOTALHOURS	decimal	Total Hours		

3.1.257 WorkTypeByGUID: Visma Severa Work Type by GUID

Catalog: Severa

Schema: API

Label: Work Type by GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: workTypeGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WorkTypeByGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
workTypeGUID	string	☐		

Columns of Table Function

The columns of the table function WorkTypeByGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		

Name	Data Type	Label	Required	Documentation
NAME	string	Name		
SALESACCOUNTGUID	string	Sales Account GUID		

3.1.258 WorkTypeByHourGUID: Visma Severa Work Type by Hour GUID

Catalog: Severa

Schema: API

Label: Work Type by Hour GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: hourGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WorkTypeByHourGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
hourGUID	string	☐		

Columns of Table Function

The columns of the table function WorkTypeByHourGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	string	Name		

Name	Data Type	Label	Required	Documentation
SALESACCOUNTGUID	string	Sales Account GUID		

3.1.259 WorkTypes: Visma Severa Work Types

Catalog: Severa

Schema: API

Label: Work Types

This is a read-only table. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Table Columns

The columns of the table WorkTypes are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	string	Name		
SALESACCOUNTGUID	string	Sales Account GUID		

3.1.260 WorkTypesByPhaseGUID: Visma Severa Work Types by Phase GUID

Catalog: Severa

Schema: API

Label: Work Types by Phase GUID

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: phaseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WorkTypesByPhaseGUID. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
phaseGUID	string	☐		

Columns of Table Function

The columns of the table function WorkTypesByPhaseGUID are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	string	Name		
SALESACCOUNTGUID	string	Sales Account GUID		

3.1.261 WorkTypesChangedSince: Visma Severa Work Types Changed since

Catalog: Severa

Schema: API

Label: Work Types Changed since

This is a read-only table function. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

Filters on API Server: startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WorkTypesChangedSince. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
options	int32	☐		
startDateTime	datetime	☐		

Columns of Table Function

The columns of the table function WorkTypesChangedSince are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CODE	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	string	Name		
SALESACCOUNTGUID	string	Sales Account GUID		

4 Schema: Native

4.1 Tables

4.1.1 NATIVEPLATFORMSCALARREQUESTS: Visma Severa Native Platform Scalar Requests

{res:itgen_native_platform_scalar_requests_desc}

Catalog: Severa

Schema: Native

Alias: `npt`

Label: Native Platform Scalar Requests

Documentation:

The NativePlatformScalarRequests table provides direct access to the native API protocol over an established connection to the Visma Severa API server. It will contain a new row for every row inserted with a native API request in PAYLOAD_TEXT with the results of unaltered forwarding of the payload to the Visma Severa API server.

Retrieve: true

Insert: true

Update: false

Delete: false

View Columns

The columns of the view NATIVEPLATFORMSCALARREQUESTS are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
BLOB_PREFERRED	boolean	BLOB Preferred	☑	Indicator w hether a BLOB result is preferred over text.
BOL_RESPONSE_CACHE_MAX_AGE_SEC	int32	Response Cache Maximum Age (sec)	☐	Maximum age in seconds of Bridge Online response cache entries to be used.
CONTENT_TYPE	string(240)	Content Type	☐	
DATE_ENDED	datetime	End Date	☑	
DATE_STARTED	datetime	Start Date	☑	
DRY_RUN	boolean	Run w ithout Actions	☑	
DURATION_MS	int64	Duration (ms)	☑	
ERROR_MESSAGE_CODE	string(30)	Error Message Code	☐	
ERROR_MESSAGE_TEXT	string(32000)	Error Message Text	☐	
FAIL_ON_ERROR	boolean	Fail on Error	☑	Whether to raise an exception w hen processing the native request triggered an error from the provider.
HTTP_DISK_CACHE_MAX_AGE_SEC	int32	HTTP Disk Cache Maximum Age (sec)	☐	Maximum age in seconds of HTTP disk cache entries to be used.
HTTP_DISK_CACHE_SAVE	boolean	Save HTTP Disk Cache	☐	Whether results can be stored in HTTP disk cache.
HTTP_DISK_CACHE_USE	boolean	Use HTTP Disk Cache	☐	Whether results can be fetched from HTTP disk cache.
HTTP_MEMORY_CACHE_MAX_AGE_SEC	int32	HTTP Memory Cache Maximum Age (sec)	☐	Maximum age in seconds of HTTP memory cache entries to be used.
HTTP_MEMORY_CACHE_SAVE	boolean	Save HTTP Memory Cache	☐	Whether results can be stored in HTTP memory cache.
HTTP_MEMORY_CACHE_USE	boolean	Use HTTP Memory Cache	☐	Whether results can be fetched from HTTP memory cache.
HTTP_METHOD	string(30)	HTTP Method	☐	
HTTP_STATUS_CODE	int32	HTTP Status Code	☐	
ORIG_SYSTEM_GROUP	string(4000)	Original System Group	☐	
ORIG_SYSTEM_REFERENCE	string(4000)	Original System Reference	☐	
PAYLOAD_TEXT	string	Payload	☐	
RESULT_BLOB	byte[]	Result BLOB	☐	
RESULT_DATE_TIME_UTC	datetime	Result Date Time	☐	
RESULT_NUMBER	decimal	Result Number	☐	
RESULT_TEXT	string	Result Text	☐	
SUCCESSFUL	boolean	Succesful	☑	
TIMEOUT_SEC	int32	Timeout (sec)	☐	Timeout in seconds.
TRANSACTION_ID	int64	Transaction ID	☑	Incrementing ID of the transaction.

Name	Data Type	Label	Required	Documentation
URL	string(4000)	URL	☐	

5 Schema: Views

5.1 Views

5.1.1 HourEntries2Years: Visma Severa Hour Entries Last 2 Years

Catalog: Severa

Schema: Views

Label: Hour Entries Last 2 Years

This is a read-only view. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

View Columns

The columns of the view HourEntries2Years are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
cse_ACCOUNTGUID	string	Account GUID		
cse_ACCOUNTNAME	string	Account Name		
cse_ACCOUNTNUMBER	int64	Account Number		
cse_BILLINGADDRESS	string	Billing Address		
cse_BILLINGADDRESSGUID	string	Billing Address GUID		
cse_BILLINGINTERNALNOTES	string	Billing Internal Notes		
cse_BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		

Name	Data Type	Label	Required	Documentation
cse_BUSINESSUNITGUID	string	Business Unit GUID		
cse_BUSINESSUNITNAME	string	Business Unit Name		
cse_BUSINESSUNITNUMBER	string	Business Unit Number		
cse_CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
cse_CASENUMBER	int64	Project Number		
cse_CASEOWNERUSERCODE	string	Project Owner User Code		
cse_CASEOWNERUSERGUID	string	Project Owner User GUID		
cse_CLOSEDDATE	datetime	Closed Date		
cse_COMPLETIONESTIMATE	int32	Completion Estimate		
cse_CONTACTGUID	string	Contact GUID		
cse_COSTCENTERGUID	string	Cost Center GUID		
cse_COSTCENTERNUMBER	string	Cost Center Number		
cse_CURRENCYGUID	string	Currency GUID		
cse_CURRENCYSHORTFORM	string	Currency Short Form		
cse_CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
cse_CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
cse_CURRENTCASESTATUSGUID	string	Current Project Status GUID		
cse_CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
cse_DEADLINEDATE	datetime	Deadline Date		
cse_DESCRIPTION	string	Ddescription		
cse_EXPECTEDVALUE	decimal	Expected Value		
cse_GUID	string	GUID		
cse_INCLUDEEXCEPTIONS	boolean	Include Exceptions		
cse_INCLUDEOVERTIME	boolean	Include Overtime		
cse_INSERTTS	datetime	Inserted		
cse_INTERNALNAME	string	Internal Name		
cse_INVOICETEMPLATEGUID	string	Invoice Template GUID		
cse_ISCLOSED	boolean	Is Closed		
cse_ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
cse_ISINTERNAL	boolean	Is Internal		
cse_ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
cse_ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
cse_LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
cse_LEADRATING	int32	Lead Rating		

Name	Data Type	Label	Required	Documentation
cse_LEADSOURCEGUID	string	Lead Source GUID		
cse_NAME	string	Name		
cse_ORDERNUMBER	string	Order Number		
cse_OURREFERENCE	string	Our Reference		
cse_OVERDUEINTEREST	decimal	Overdue Interest		
cse_PAYMENTTERM	int32	Payment Term		
cse_PRICELISTGUID	string	Pricelist GUID		
cse_PRICINGRULE	string	Pricing Rule		
cse_PRICINGTYPE	string	Pricing Type		
cse_PRIORITY	int32	Priority		
cse_PROBABILITY	int32	Probability		
cse_ROOTTASKGUID	string	Root Task GUID		
cse_SALESCLOSEDATE	datetime	Sales Close Date		
cse_SALESPERSONUSERCODE	string	Sales Person User Code		
cse_SALESPERSONUSERGUID	string	Sales Person User GUID		
cse_SALESPROCESSGUID	string	Sales Process GUID		
cse_SALESSTATUSGUID	string	Sales Status GUID		
cse_SHAREDTOEXTRANET	boolean	Shared to Extranet		
cse_STARTDATE	datetime	Start Date		
cse_UPDATEDTS	datetime	Updated		
cse_URLTOCASE	string	URL to Project		
cse_USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
cse_USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
cse_YOURREFERENCE	string	Your Reference		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		

Name	Data Type	Label	Required	Documentation
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

5.1.2 Invoices: Visma Severa Invoices

Catalog: Severa

Schema: Views

Label: Invoices

This is a read-only view. The Visma Severa API may not support changing the data or the Invantive UniversalSQL driver for Visma Severa does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Visma Severa API.

View Columns

The columns of the view Invoices are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANISATIONNAME	string			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	string			
CULTURE	string	Culture		
CURRENCYGUID	string	Currency GUID		
CURRENCYNAME	string	Currency Name		
CURRENCYRATE	decimal	Currency Rate		
CURRENCYSHORTFORM	string	Currency Short Form		

Name	Data Type	Label	Required	Documentation
DATE	datetime	Date		
DESCRIPTION	string	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	string	GUID		
INVOICEALLOWEDCASEGUIDS	string			
INVOICECONFIG_BILLEXPENSE	string			
INVOICECONFIG_BILLHOUR	string			
INVOICECONFIG_BILLPRODUCT	string			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURPRIMARYGROUPBY	string			
INVOICECONFIG_HOURSECONDARYGROUPBY	string			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	string			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	string			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDETAILS_INVOICEDATE	datetime			

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	int32			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETYPETEXT	string			
INVOICEDETAILS_ORDERIDENTIFIER	string			
INVOICEDETAILS_ORIGINCODE	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	string			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	decimal			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	string			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	string			
INVOICEDETAILS_VATBASEAMOUNT	decimal			
INVOICEDETAILS_VATRATEAMOUNT	decimal			
INVOICEDETAILS_VATRATEPERCENT	int32			
INVOICENUMBERINGBUSINESSUNITGUID	string	Invoice Numbering Business Unit GUID		
INVOICEROWS	string			
INVOICESTATUS	string	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
MAINCASEGUID	string	Main Case GUID		
NOTES	string	Notes		
NOTESAFTERINVOICEROWS	string	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	string	Notes before Invoice Lines		
NUMBER	int32	Number		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		

Name	Data Type	Label	Required	Documentation
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	string	Payment Status		
PAYMENTTERM	int32	Payment Term		
PONUMBER	string	PO Number		
RECEIVERADDRESSLINE	string	Receiver Addressline		
RECEIVERCITY	string	Receiver City		
RECEIVERCONTACTEMAIL	string	Receiver Contact Email		
RECEIVERCONTACTNAME	string	Receiver Contact Name		
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREETEXT	string			
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONENUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLEREINVOICEADDRESS	string			
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			

Name	Data Type	Label	Required	Documentation
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		
YOURREFERENCE	string	Your Reference		

Index

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- J -

- K -

- L -

- M -

- N -

- O -

- P -

- Q -

- R -

- S -

- T -

- U -

- V -

- W -

- Y -

Invantive B.V.
Biesteweg 11
3849 RD Hierden
the Netherlands

Tel: +31 88 00 26 500
Fax: +31 84 22 58 178
info@invantive.com
invantive.com

IBAN NL25 BUNQ 2098 2586 07
Chamber of Industry and Commerce
13031406
VAT NL812602377B01
RSIN 8122602377
Managing Director: Guido Leenders
Registered office: Roermond

23.0.92
24/08/2024, 20:39
ISBN 978-94-6302-000-8: