

Senturus Analytics Connector

User Guide

Cognos to Tableau

Contents

Overview	3
Connecting to Cognos from Tableau	4
SQL Server Connection vs ODBC	4
Use Microsoft SQL Server Connection	6
Use ODBC Connection	8
Create Data Source	10
Mapping from Cognos to Database Schema	11
Link Columns and Dummy Joins	12
Custom Joins	12
Relationships	13
Custom SQL	13
Live vs. Extract	14
Data Source Best Practices	14
Warning When Switching from Data Source Tab	16
Parameterized Tables and Reports	17
Working with Parameterized Tables	17
Working with Reports	20
Function Syntax	23
FAQs	26
CONTACT US	27

Overview

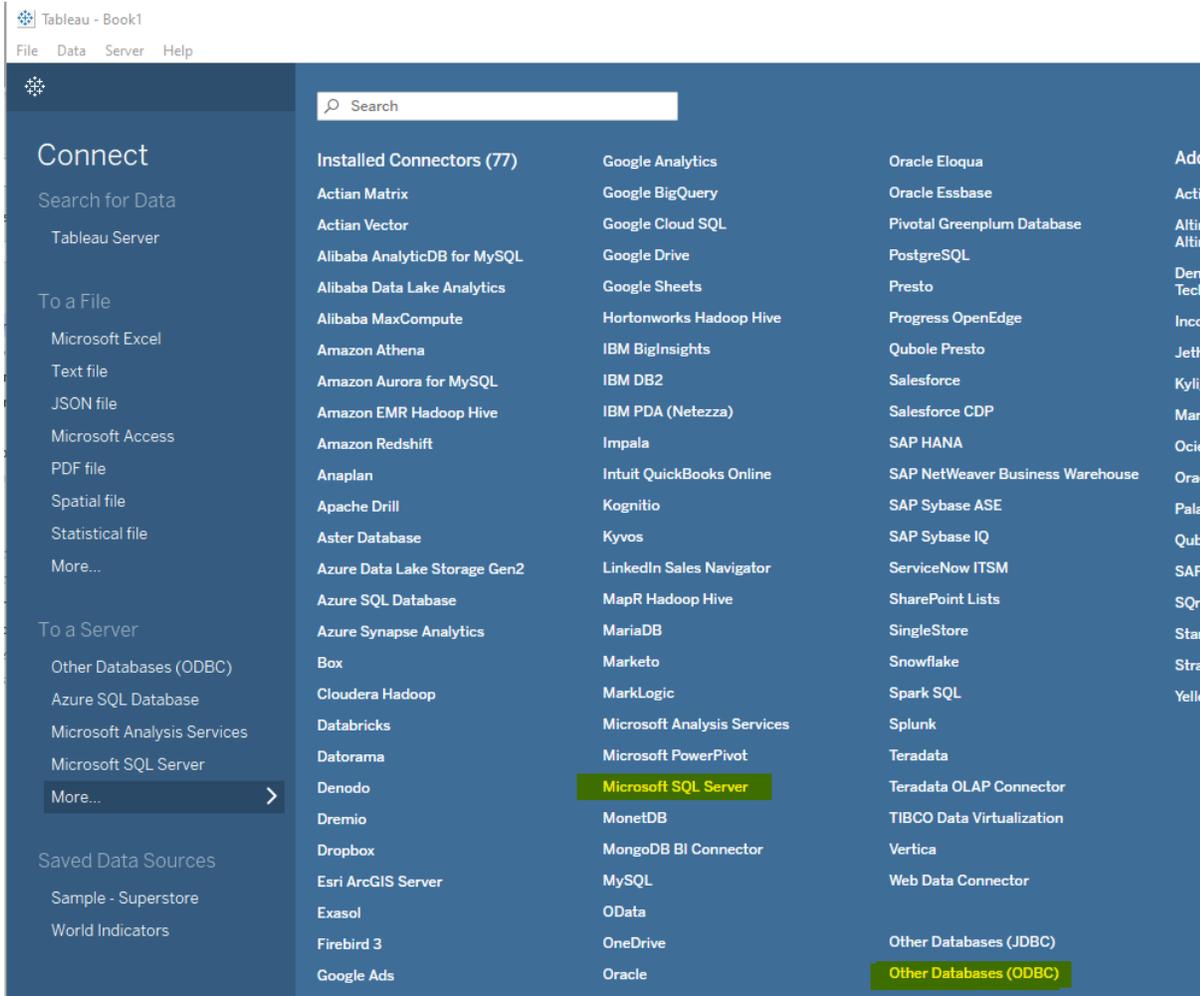
This guide describes how the Senturus Analytics Connector is used from Tableau after it has been installed and configured.

Please refer to the Installation and Configuration Guide for instructions on installing and configuring the Analytics Connector.

The Analytics Connector has been tested against different versions of Tableau, including both Tableau Desktop and Tableau Server on Windows from version 9.3.

Connecting to Cognos from Tableau

You can use either “Microsoft SQL Server” or “Other Database (ODBC)” (via Senturus Analytics Connector ODBC client) connection to connect Tableau to Analytics Connector server.



SQL Server Connection vs ODBC

Starting from version 5.0.8, you can connect from Tableau Desktop, Tableau Server and Tableau Bridge to Analytics Connector server via SQL Server connection.

If you have old workbooks and/or server-side data sources, you may still use ODBC connections, all your old workbooks and data sources will continue working. And you can still use ODBC connection in new workbooks if you wish.

Comparing to ODBC connection, using SQL Server connection gives you following advantages:

1. No need to install Analytics Connector ODBC client¹.
2. Use Windows authentication from Tableau Desktop/Server to Analytics Connector server to Cognos.²

To use custom Data Connector, you need to install Senturus Analytics Connector ODBC client on the computer where Tableau Desktop and/or Tableau Server will be making connection from. The latest ODBC client can be found at [Senturus Analytics Connector - Customer Resources](#).³

Senturus Analytics Connector ODBC client is a customized PostgreSQL ODBC driver⁴, that provides several new features, for example you can select database name from a drop-down list.

¹ You still need to install SQL Server Native Client 11 or above on your client computer.

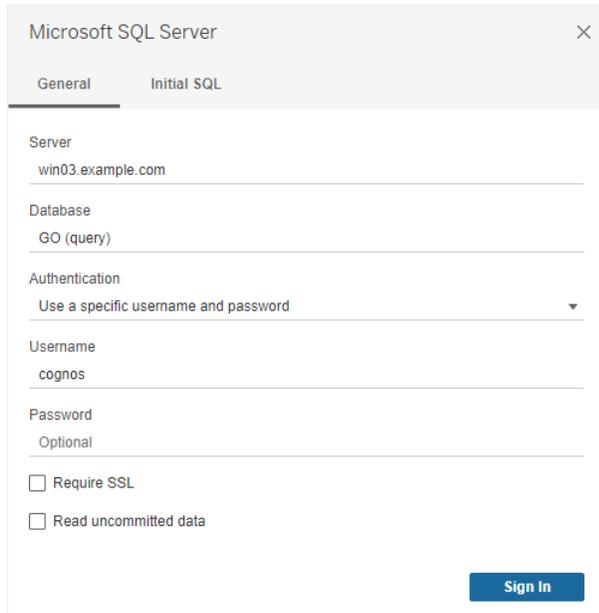
² Both Tableau Desktop/Server and Cognos server must be secured by the same Active Directory service and additional configuration is needed to enable Windows authentication. Please refer to Senturus Analytics Connector Server Installation and Configuration guide for more information.

³ ODBC client is automatically installed when you are installing Senturus Analytics Connector server. So, you don't have to install ODBC client separately on the same computer.

⁴ Tableau installer comes with its own PostgreSQL ODBC driver, but that driver is an older version that is not compatible Senturus Analytics Connector server.

Use Microsoft SQL Server Connection⁵

To begin, select the **More...** option under **To a Server**, then select **Microsoft SQL Server**.



The screenshot shows a dialog box titled "Microsoft SQL Server" with a close button (X) in the top right corner. It has two tabs: "General" (selected) and "Initial SQL". The "General" tab contains the following fields and options:

- Server:** win03.example.com
- Database:** GO (query)
- Authentication:** Use a specific username and password (dropdown menu)
- Username:** cognos
- Password:** Optional
- Require SSL
- Read uncommitted data

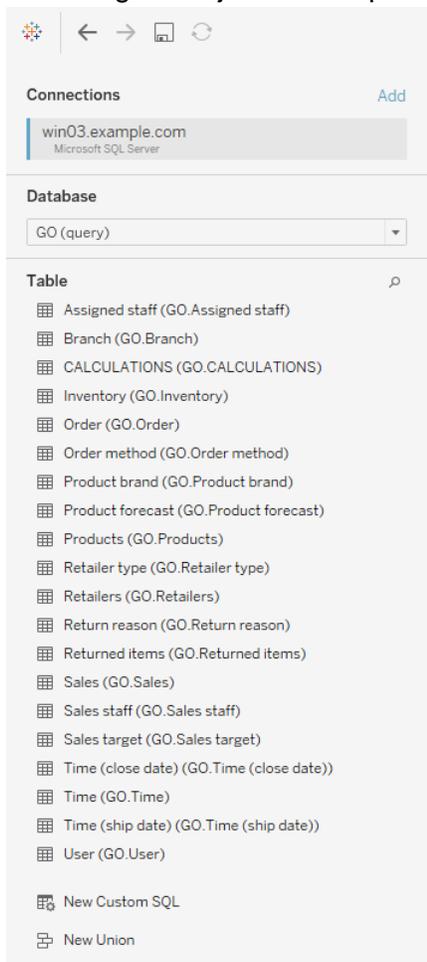
A blue "Sign In" button is located at the bottom right of the dialog box.

Above is a Microsoft SQL Server connection dialog. You need to provide following information:

1. The Analytics Connector server (not Cognos server) name or IP.
2. Optionally, type in the user-friendly database name that maps a Cognos package or data module.
3. Select either windows or user/password authentication and provide username (Cognos username) and password as needed.
4. If your Analytics Connector Server is SSL secured, you can optionally check "Require SSL" box.

⁵ Make sure you have installed SQL Server Native Client 11 or above in order to use SQL database connection. The download can be found here: <https://www.microsoft.com/en-us/download/details.aspx?id=50402>

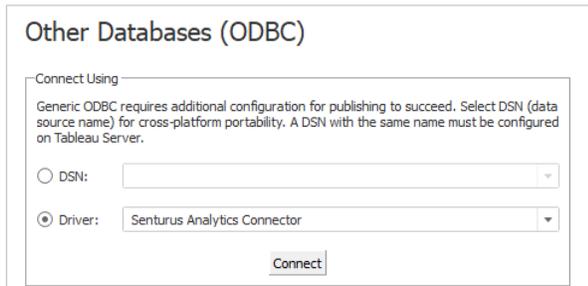
After clicking on “Sign in” button, Tableau will connect to Analytics Connector Server and bring back Cognos objects and represent them as tables, as below.



Use ODBC Connection

To begin, select the **More...** option under **To a Server**, then select **Other Databases (ODBC)**.

Select the radio button next to **Driver**, select *Senturus Analytics Connector* from the list, then click **Connect**.



Other Databases (ODBC)

Connect Using

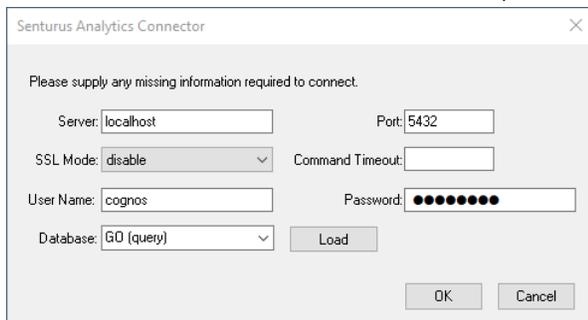
Generic ODBC requires additional configuration for publishing to succeed. Select DSN (data source name) for cross-platform portability. A DSN with the same name must be configured on Tableau Server.

DSN:

Driver:

Enter Analytics Connector server name or IP, Port number, User Name and Password.

Click **Load** to load a list of configured databases the user has access to on the specified server. Select a Database from the loaded list, then click **OK**.



Senturus Analytics Connector

Please supply any missing information required to connect.

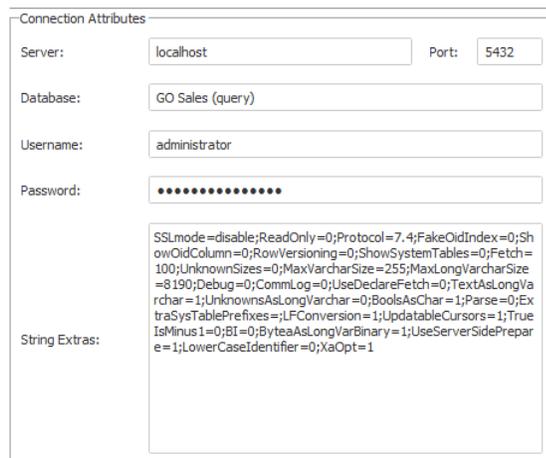
Server: Port:

SSL Mode: Command Timeout:

User Name: Password:

Database:

Click **Sign In** to complete the connection. You will then see the Tableau Data Source tab for this connection.



Connection Attributes

Server: Port:

Database:

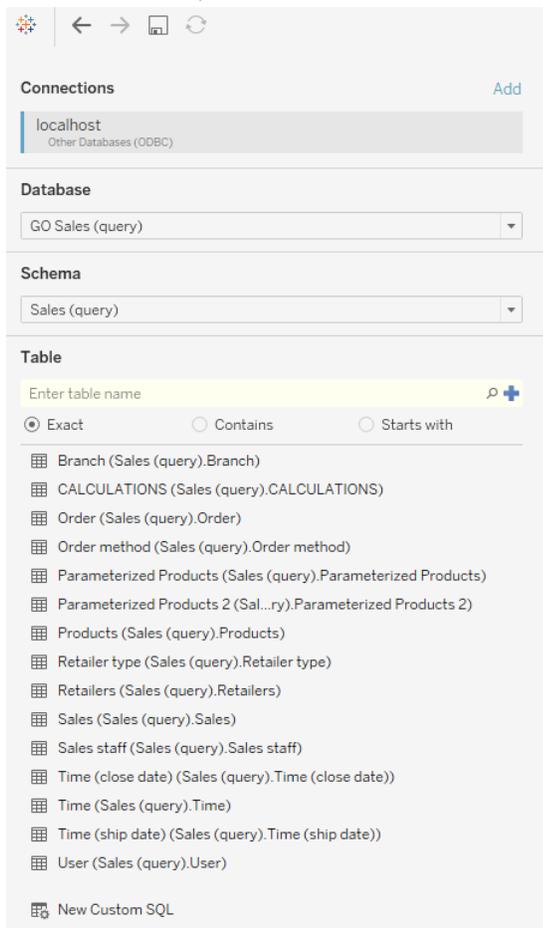
Username:

Password:

String Extras:

```
SSLmode=disable;ReadOnly=0;Protocol=7.4;FakeOIDIndex=0;ShowOldColumn=0;RowVersioning=0;ShowSystemTables=0;Fetch=100;UnknownSizes=0;MaxVarcharSize=255;MaxLongVarcharSize=8190;Debug=0;CommLog=0;UseDeclareFetch=0;TextAsLongVarchar=1;UnknownsAsLongVarchar=0;BoolsAsChar=1;Parse=0;ExtraSysTablePrefixes=;LFCConversion=1;UpdatableCursors=1;TrueIsMinus1=0;BI=0;ByteAsLongVarBinary=1;UseServerSidePrepare=1;LowerCaseIdentifier=0;XaOpt=1
```

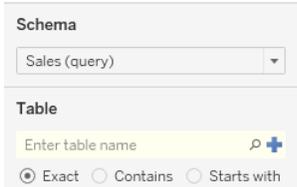
Tableau will connect to Analytics Connector Server and bring back Cognos objects and represent them as tables, as below.



Create Data Source

After connecting successfully, the database, schema and tables will show on the **Data Source** tab.

The full list of schemas and tables appears in the **Table** section. To filter the list, type letters of your desired schemas/tables into the search field.



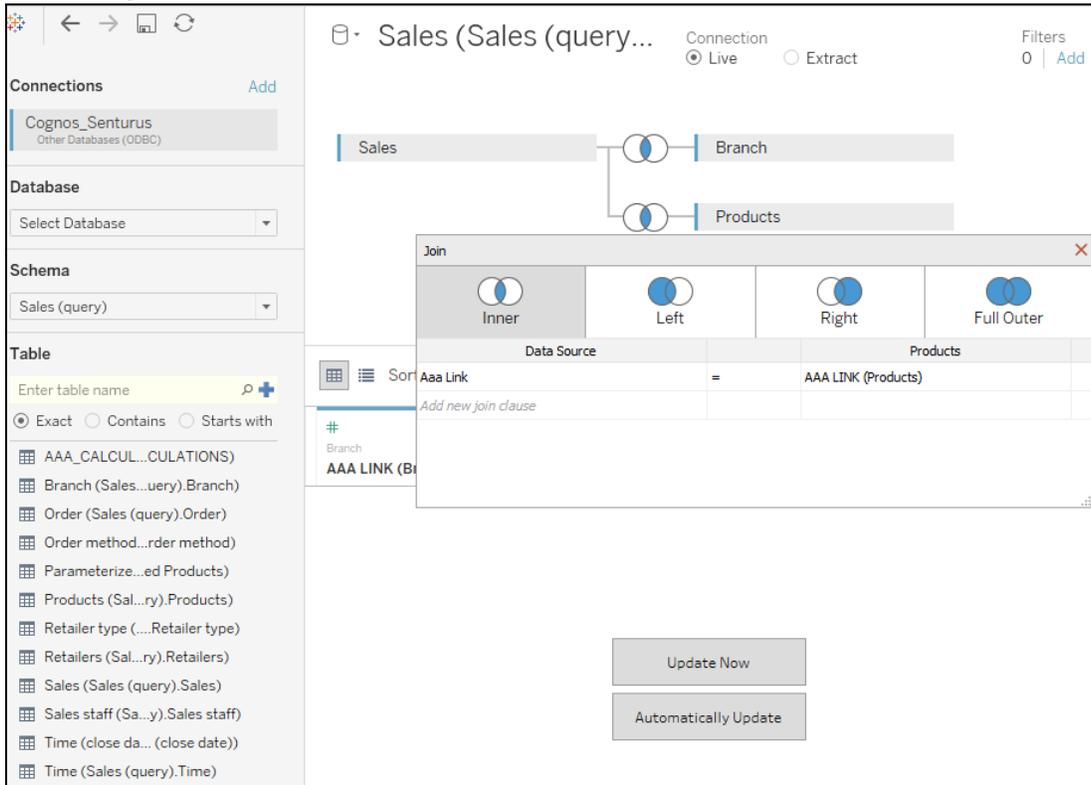
The screenshot shows the Tableau Data Source configuration window. It is divided into two main sections: 'Schema' and 'Table'. The 'Schema' section has a dropdown menu currently set to 'Sales (query)'. The 'Table' section features a search input field with the placeholder text 'Enter table name' and a magnifying glass icon. Below the search field are three radio button options: 'Exact' (which is selected), 'Contains', and 'Starts with'.

If you are using version 2020.2 or newer, Tableau Desktop will create a logical table when you add a database table to the canvas and will let you define relationships between logical tables. But in most cases, you want to define dummy joins in Tableau so that you can use Cognos built-in relationships. Click on the down down array and select “Open...”

☞ Sales (Sales (query).Sales) (G)



Then drag or double click the more tables into the canvas.



Mapping from Cognos to Database Schema

Each FM namespace is mapped to a database schema. For data module or cube data sources without namespace, a single schema with same name as database will be used.

Each FM query subject/measure dimension/regular dimension hierarchy, data module table/navigation path is mapped to a database table.

Query items, columns under those Cognos objects are mapped to columns under corresponding tables.

If there is any stand-alone FM calculation under a namespace, a special table named CALCULATIONS⁶ will appear under the corresponding database schema and all calculations will be columns for this special table.

Each report is mapped to a table.

Analytics Connector does not support FM filters.

⁶ Configurable via Data Source Configuration tool.

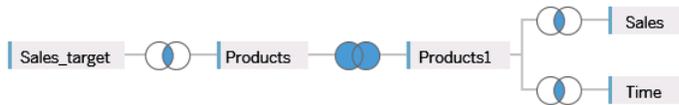
Link Columns and Dummy Joins

Tableau requires joins between the tables in a data source, but link columns are usually not available in the schemas converted from Cognos packages. To resolve this limitation, the Analytics Connector creates a link column on each table named *AAA_LINK*⁷. The link column name can be changed using the Configuration Utility.

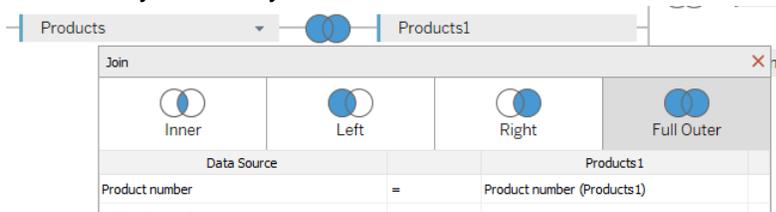
Tableau will automatically join the tables using the link columns, but the ODBC driver will ignore them when passing the queries to Cognos.

Custom Joins

As mentioned previously, Tableau will automatically join on the link columns the Analytics Connector creates. Usually, you will not want to change these joins because you want to leverage the underlying Cognos relationships. However, you can create custom joins to build Tableau models that your Cognos implementation does not include.



Using the example shown above, you can change the automatic join between *Products* and *Products1* to a full outer join on the “Product Number” column. This type of custom join will be honored by the Analytics Connector.



Additionally, the Analytics Connector will split the five tables into two sub queries. The first sub query contains *Sales_target* and *Products* and the second sub query has *Products1*, *Sales* and *Time*. The results of the two sub queries will be joined together by the Product Number column.

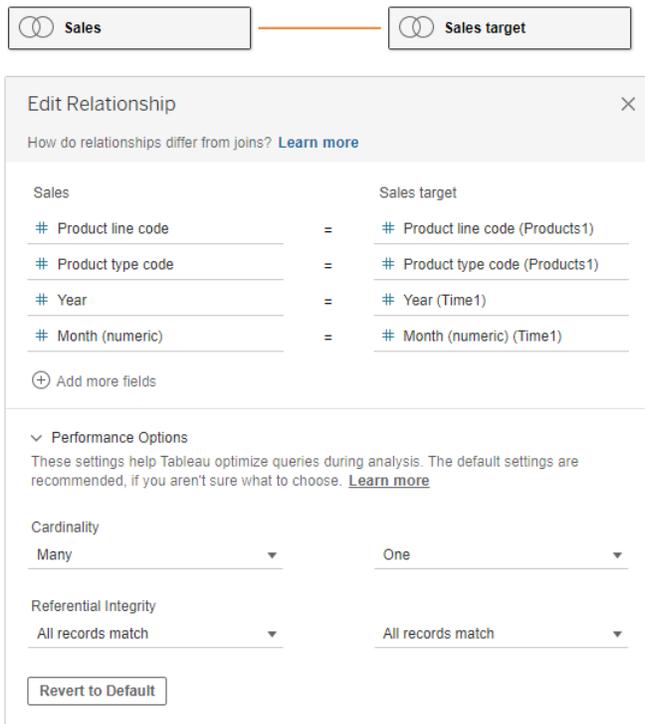
⁷ AAA_LINK is the default dummy column name, but your Senturus Analytics Connector server administrators can change it via Data Source Configuration tool.

Relationships

One of the benefits of using Tableau relationships is that you can avoid writing Level of Detail (LOD) expressions to handle multi-fact, multi-grain data.

Taking the “Sales vs. Sales Target” data for example, you can define two logical tables first, then set up a relationship between them as below.

☐ Sales vs. Sales Target



Custom SQL

In addition to tables, you can drag a “New Custom SQL” to Tableau data source. Within this Custom SQL, you can write a single select query using standard SQL syntax. Analytics Connector support most SQL features and functions, e.g. where clause, group clause, order by clause, union operation, aggregate, sub queries, etc. However, it does not support with clause and other (e.g. update, select ... into statements).

As mentioned in section [Link Columns and Dummy Joins](#), you should join tables using dummy/link columns.

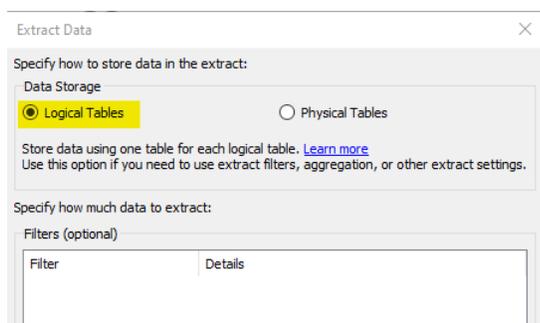
To execute a Cognos report or invoke parameterized tables, Custom SQL is used too. Please refer to section [Parameterized Tables and Reports](#) for more information.

Live vs. Extract

Senturus Analytics Connector support both Live and Extract queries. The bottom line is, Senturus Analytics Connector sever has no clue if a query is sent for a Live or a Extract purpose.

- When choosing live vs. extract, you need to consider if you need to preseve Cognos securities. When connecting Analytics Connector server via live queries, Analytics Connector server keeps a live connection to Cognos suer on behalf of the Tableau user, so all object security and data security (row/column levels) are preserved. For extract, Cognos login is only used when creating/refreshing the extract. Afterwards, data is stored witin Tableau and open to all users who have access to this Tableau workbook/data source.
- Queries and datast size are very different too. For live queries, they are highly aggreated, and in most cases they are filtered, so the dataset are usually very small – handful columns by dozens rows. But extract queries are plan select without groups/aggregates, they may have filters, but the dataset are usually very big – dozens of columns by hundreds of thousands rows or more.
- Performance wise, extrat usually provide better performance, but it's not always true. If your extract is huge while live queries are small and fact, live queries may overperformance extract.

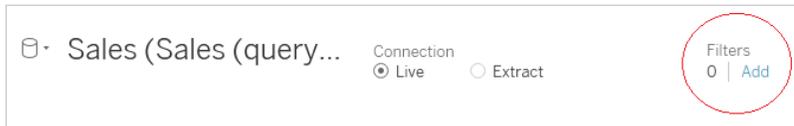
An important note, make sure you checked “**Logical Tables**” (default) for “**Data Storage**” when creating the extract. Otherwise Tableau will store tables separately and join them using AAA_LINK column when query extract – thus generating incorrect result.



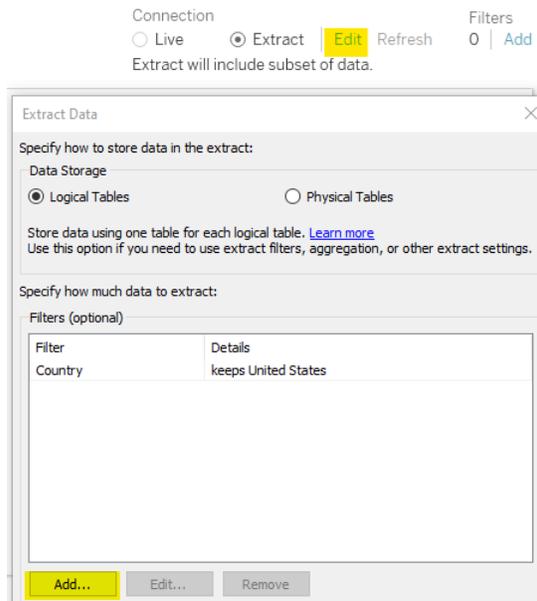
Data Source Best Practices

- Unless absolutely needed, always use joins (joined by dummy link columns) in a single logical table. If your Cognos model missed a relationship, you may define a join using columns other than link columns. Or if you have a multi-fact, multi-grain data, you may add two logical tables and create relationship using columns other than link columns.

- Always add fact table(s) first in logical tables.
- Choose Live or Extract wisely based on your own use cases. Please refer to section [Live vs. Extract](#) for more information.
- You can add Filters using the **Filters Add** feature in the top right corner of the widow. Appropriate filters can improve query performance when designing the visualizations.



- If you want to define filters in Extract, make sure you use the **Edit** button right to **Extract** checkbox, then click **Add...** button to add filters. Filters defined in other places won't be send via the extract queries to data sources.



- Limit the number of rows and columns in extracts. You can remove tables you don't need and hide columns you don't need, to limit the number of rows in Tableau extracts. To filter our unwanted rows, add filters using the Edit button right to the Extract radio checkbox.
- Write a custom SQL to further reduce the size of your extract. Tableau extract queries won't aggregate your data. If all your measures are additive, you can write a custom SQL to aggregate your data, in addition to filtering columns/rows.
- If you something goes wrong, the first place you want to check is "C:\Program Files\Senturus\Analytics Connector\Log\java.log".

Warning When Switching from Data Source Tab

When you switch from the **Data Source** tab to the **Sheet** tab, the below warning may appear.



Tableau uses several special features to optimize queries, but the Analytics Connector and Cognos do not support all of them. This dialog box is only a warning, and your workbook will function correctly.

To prevent this pop up from appearing again, check *Do not show again for this data source*, then click **OK**.

You are now able to design your visualization using the Cognos Dimensions and Measures in Tableau.

Parameterized Tables and Reports

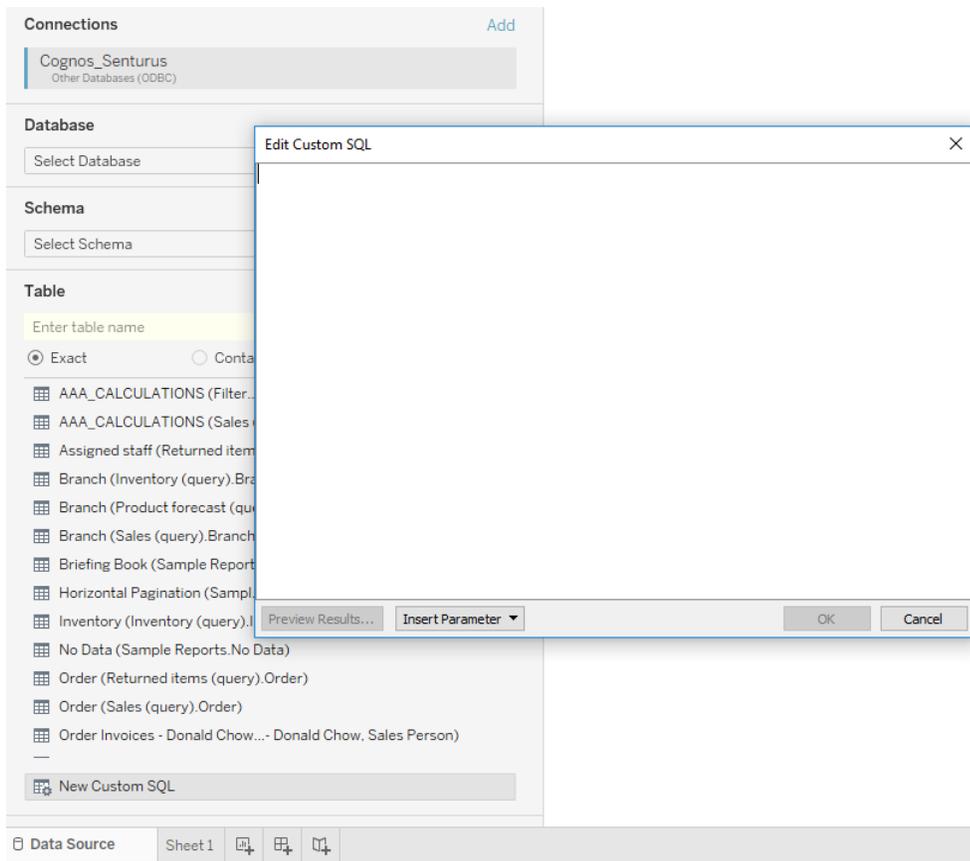
Regular database tables do not have or support parameters. But when working with a Cognos parameterized query subject or a report, users will be prompt to answer parameter values before it can be executed.

Analytics Connector supports parameterized tables and reports by utilizing Tableau “Custom SQL” feature and two special functions.

Working with Parameterized Tables

To use a parameterized table in a Tableau workbook and pass parameter values, use the Tableau **New Custom SQL** feature.

To begin, select a Database then double click **New Custom SQL** to launch the **Edit Custom SQL** window. Enter the SQL to execute.



Using the *OpenTable* function, you can pass parameters to a parameterized Cognos table.

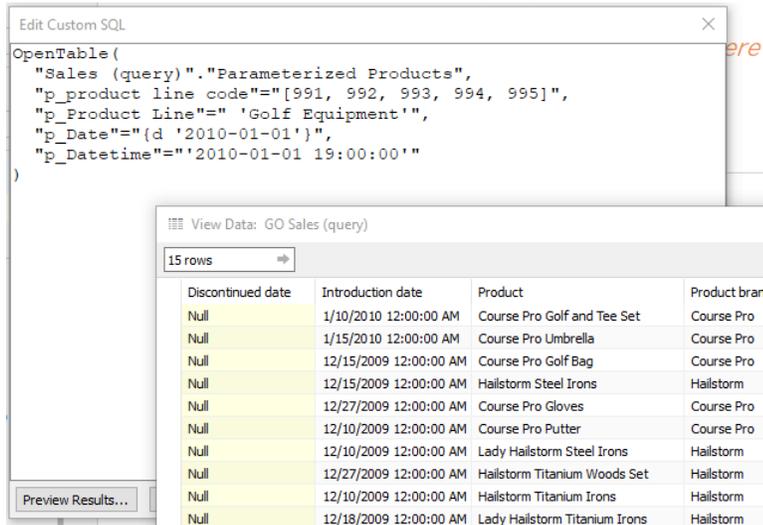
Example passing static valued parameters:

```

OpenTable(
  "Sales (query)"."Parameterized Products",
  "p_product line code"="[991, 992, 993, 994, 995]",
  "p_Product Line"=" 'Golf Equipment'",
  "p_Date"="{d '2010-01-01'}",
  "p_Datetime"="2010-01-01 19:00:00"
)

```

Click **Preview Results...** to run the query. The results will display in a new window.



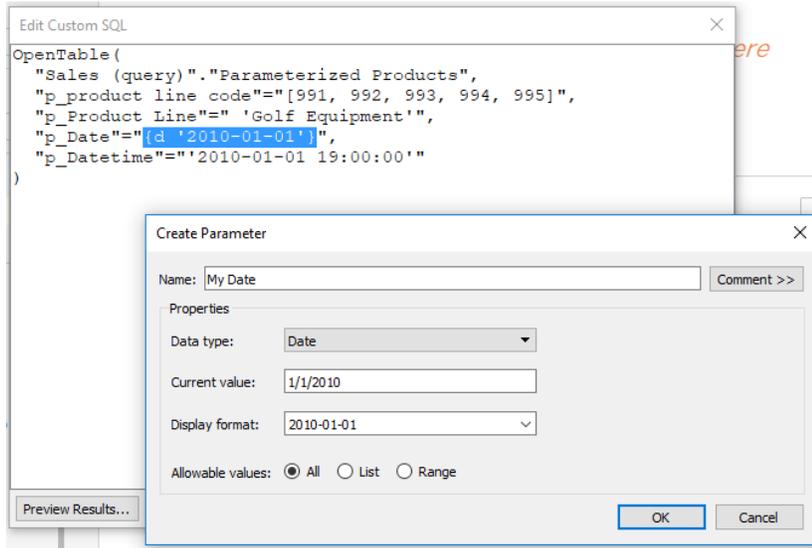
The first parameter is “[schema name]”. “[table name]” of the Cognos parameterized table. This is followed by zero or more Cognos parameter name-value pairs separated by commas.

Optionally, you can replace the static values with Tableau parameters to create dynamic values.

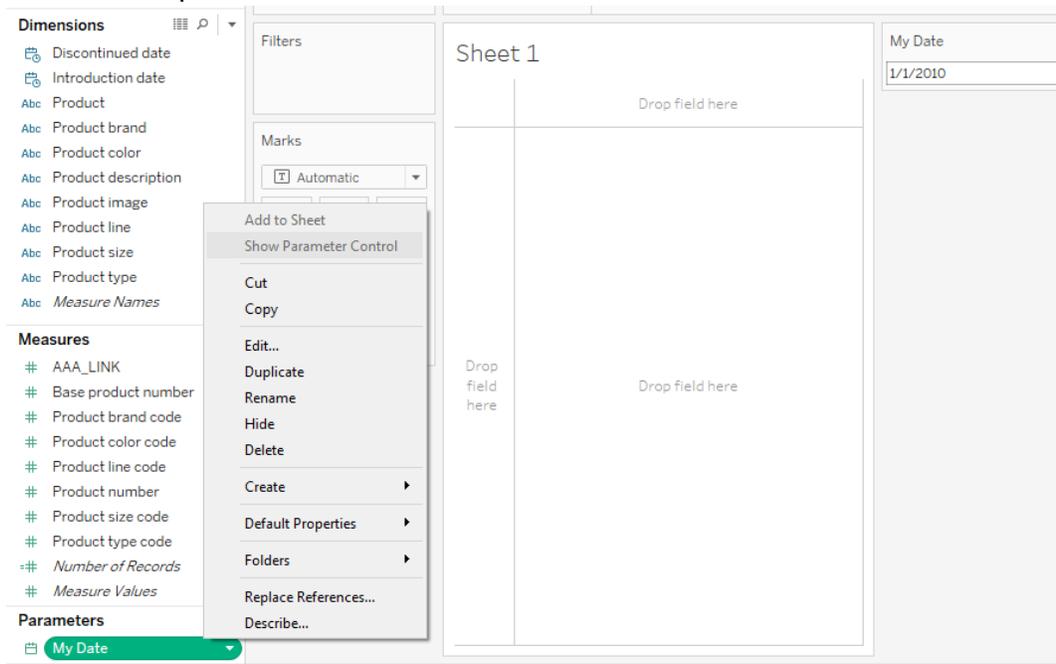
For example:

- Highlight the date static value.
- Select the Insert Parameter button, then Create a New Parameter.
- Define the Tableau date parameter and click **OK**.

It is important that you select the correct Data type for the parameter or the SQL may error.



The Tableau parameter can then be shown on the visualization or dashboard for user interaction.



You can use more than one OpenTable function in a Tableau query, as long as each one is contained in individual Custom SQL object. Like other tables, they are joined using the generated Link Columns which are then ignored at execution time.

You can also mix parameterized tables with other tables in one Tableau data source. These tables will also join using the generated Link Columns. Example advanced query:

```

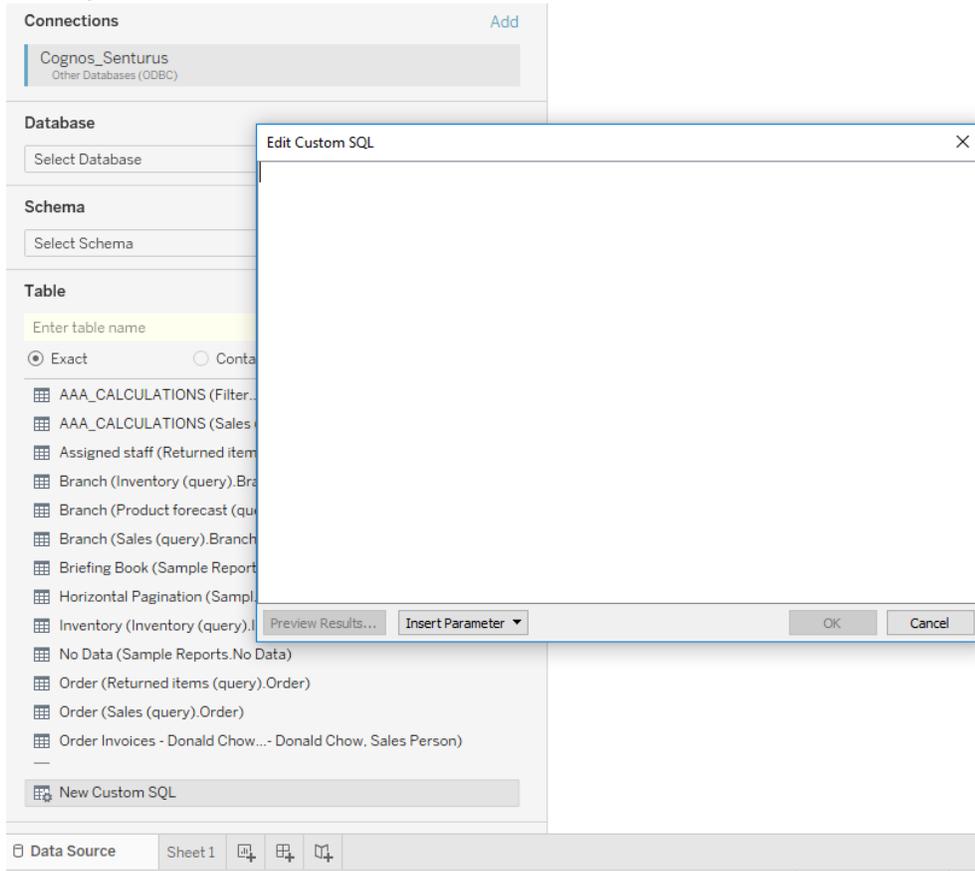
Select
  "Products"."Product line",
  "Products"."Product type",
  Sum("Sales"."Revenue") as "Revenue"
From
  OpenTable(
    "Sales (query)". "Parameterized Products",
    "p_product line code"="[991, 992, 993, 994, 995]",
    "p_Product Line"=" 'Golf Equipment'",
    "p_Date"="{d '2010-01-01'}",
    "p_Datetime"="'2010-01-01 19:00:00'"
  ) as "Products"
join "Sales (query)". "Sales" as "Sales"
  on ("Products"."AAA_LINK" = "Sales"."AAA_LINK")
Group by "Products"."Product line"

```

Working with Reports

To use a Report in a Tableau workbook and pass parameter values, use the Tableau **New Custom SQL** feature.

To begin, select a Database then double click **New Custom SQL**. Enter the SQL to execute.

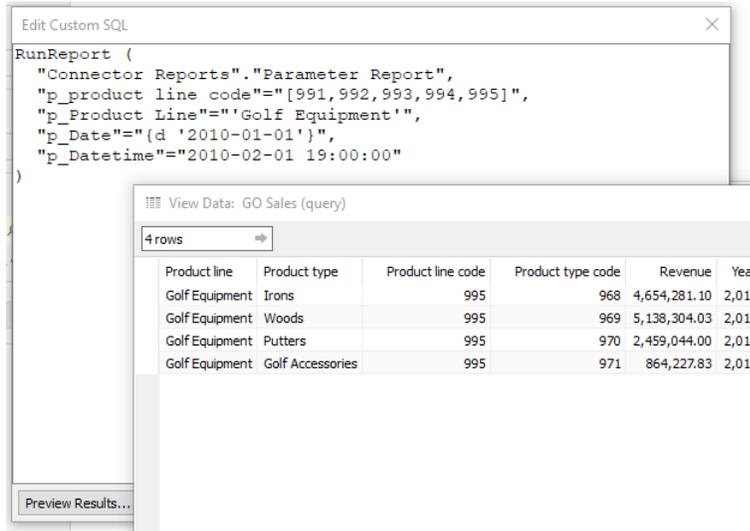


Using the *RunReport* function, you can pass parameters to a Cognos Report.

Example passing static valued parameters:

```
RunReport (  
    "Connector Reports"."Parameter Report",  
    "p_product line code"="[991,992,993,994,995]",  
    "p_Product Line"="Golf Equipment",  
    "p_Category"=["Books', 'Electronics', 'Movies']",  
    "p_Date"="{d '2010-01-01'}",  
    "p_Datetime"="2010-02-01 19:00:00"  
)
```

Clicking the **Preview Results...** button will run the query and display the results in a new window.



The screenshot shows two overlapping windows. The top window, titled "Edit Custom SQL", contains the following SQL query:

```
RunReport (
  "Connector Reports"."Parameter Report",
  "p_product line code"="[991,992,993,994,995]",
  "p_Product Line"="'Golf Equipment'",
  "p_Date"="{d '2010-01-01'}",
  "p_Datetime"="2010-02-01 19:00:00"
)
```

The bottom window, titled "View Data: GO Sales (query)", displays a table with 4 rows. The table has the following columns: Product line, Product type, Product line code, Product type code, Revenue, and Year.

Product line	Product type	Product line code	Product type code	Revenue	Year
Golf Equipment	Irons	995	968	4,654,281.10	2,010
Golf Equipment	Woods	995	969	5,138,304.03	2,010
Golf Equipment	Putters	995	970	2,459,044.00	2,010
Golf Equipment	Golf Accessories	995	971	864,227.83	2,010

At the bottom of the SQL editor window, there is a button labeled "Preview Results..."

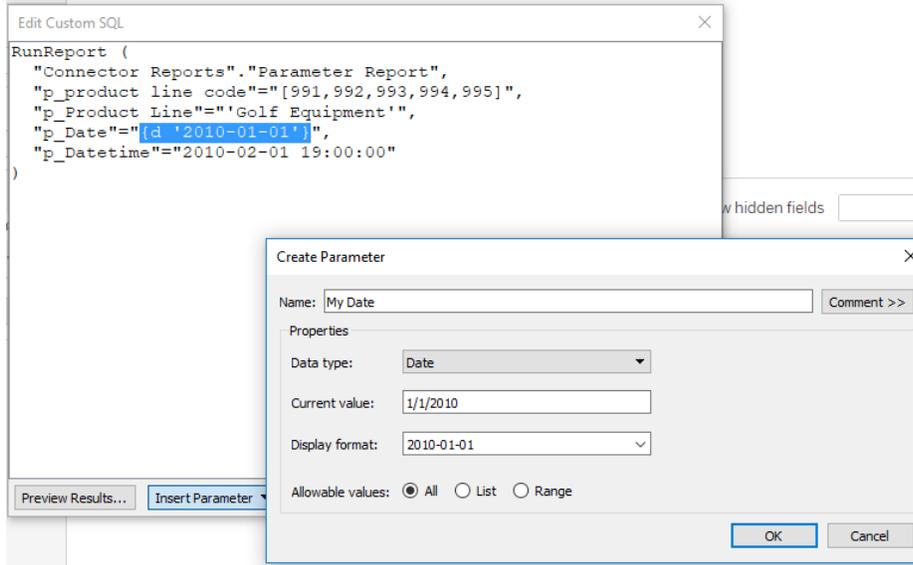
The first parameter is "[schema name]".[table name]" of the Cognos Report. This is followed by zero or more Cognos parameter name-value pairs separated by commas.

Optionally, replace the static values with Tableau parameters to create dynamic values.

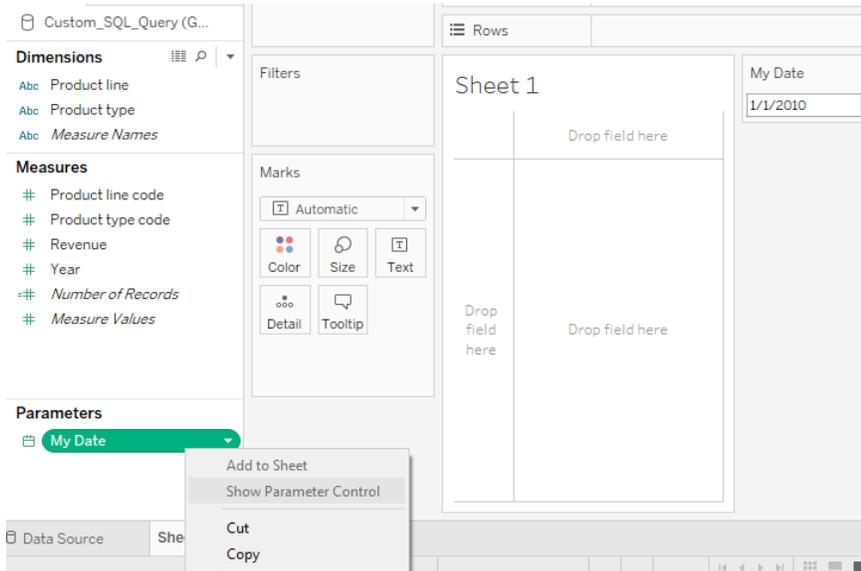
Example:

- Highlight the date static value.
- Click the **Insert Parameter** button, then Create a New Parameter.
- Define the Tableau date parameter and click **OK**.

It is important that you select the correct Data type for the parameter or the SQL may error.



The Tableau parameter can then be shown on the visualization or dashboard for user interaction.



Function Syntax

Syntax

Both `OpenTable` and `RunReport` function adopt following syntax:

```
OpenTable ( "Schema Name"."Table Name" (, "Parameter"="value", ...) )
```

The first parameter is a table identifier, using standard SQL syntax. All identifiers are case insensitive and the schema name can be omitted if the table name is unique cross schemas.

The second, if any, and following parameters are name-value pairs, separated by commas. Name and values should be surrounded by double quotation marks.

Parameters

Use parameter name in place of "Parameter" for the Cognos connector.

Values

All values should follow SQL/ODBC standard.

Only plain string representation is allowed for numeric values. Formatted representations such as **1,200.00** with thousand separators, **(15)** for -15, or **50%** for .5 are not supported.

For string values, single quotation marks are used around the value. For example, 'Golf Equipment' or 'Sam's Club'.

For date/time/timestamp values, use the ODBC standard format. For example {d '2010-01-01'} or ISO 8601 standard format, 'YYYY-MM-DD' for date, 'hh:mm:ss' for time and 'YYYY-MM-DD hh:mm:ss' for timestamp.

An array of values is passed using brackets. For example:

- "product line code"="[991,992,993,994,995]"

The syntax for a Range parameters is represented by a 2 element array. For example:

- "YearRange"="[,2019]" – All years less than or equal to 2019
- "YearRange"="[2015,]" – All years greater than or equal to 2015
- "YearRange"="[2015,2019]" – Years between 2015 and 2019

Character escaping

Certain characters must be escaped.

For single quotes within a string value, use two single quotes to escape a single quote inside the string.

Double quotes in parameter name and values need to be escaped with two double quotes.

Passing multiple values to a parameter

A Cognos parameter may accept multiple values, for example [Product line code] in (?p_Product line code?). To specify multiple values in your function, put multiple values in a pair of square brackets ([]) and separate each value by a comma (,) just like CSV format.

Using Tableau parameters

You can use Tableau parameters in place of literal values but take the following notes:

- Tableau will replace the parameter name with a formatted string when executing a string. For example, Tableau will put single quotes around string values, escape single quotes with two single quotes in string values, and format date value using ODBC format. You do not need to “format” them in a function call.
- Tableau will not escape double quotes, so you need to escape them when you provide values to Tableau parameter.
- Tableau does not support multiple value parameters. To pass multiple values to one Cognos parameter, define multiple parameters in Tableau and put them in your function. For example "param"="[<Parameters.p1>, <Parameters.p2>, <Parameters.p3>]".

FAQs

Q: What happens when the underlying Cognos model or metadata changes?

A: If a column or table is removed or renamed in Cognos, Tableau will mark the old measure(s) and dimension(s) as invalid when a saved workbook is opened. The user can then decide how to correct the workbook:

- The user can use Tableau's **Replace References** feature to reassign all occurrences with a new, valid column.
- The user can remove references to the old column from the workbook.

If table(s) or column(s) are added, Tableau will display the new column the next time it connects to the data source. This typically happens when a workbook is opened or when a data source is refreshed.

There is nothing specific to the Analytic Connector that changes this behavior of Tableau. For more information on Tableau Data Sources, see this Tableau article:

https://onlinehelp.tableau.com/current/pro/desktop/en-us/refreshing_data.htm

CONTACT US

The latest version of the Analytics Connector along with related documentation and contact information can be found at <https://www.senturus.net/connector-download/>.

If you have any additional questions, please contact us at CustomerSupport@senturus.com.