

Senturus Analytics Connector

User Guide

Cognos to Tableau

Contents

Overview	3
ODBC vs SQL Server Connection	4
Connecting to Cognos from Tableau	5
Use Microsoft SQL Server Connection	6
Use ODBC Connection	8
Create Data Source	10
Link Columns and Dummy Joins	11
Custom Joins	11
Relationships	12
Cognos Calculations	13
Data Source Best Practices	14
Warning When Switching from Data Source Tab	14
Parameterized Tables and Reports	16
Working with Parameterized Tables	16
Working with Reports	19
Function Syntax	22
FAQs	25
CONTACT US	26

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.

ODBC vs SQL Server Connection

Starting from version 5.0.8, you can connect from Tableau Desktop, Tableau Server and Tableau Bridge to Analytics Connector 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 client.
2. Use Windows authentication from Tableau Desktop/Server to Analytics Connector server to Cognos.¹

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](#).²

Senturus Analytics Connector ODBC client is a customized PostgreSQL ODBC driver³, that provides several new features, for example you can select database name from a drop-down list.

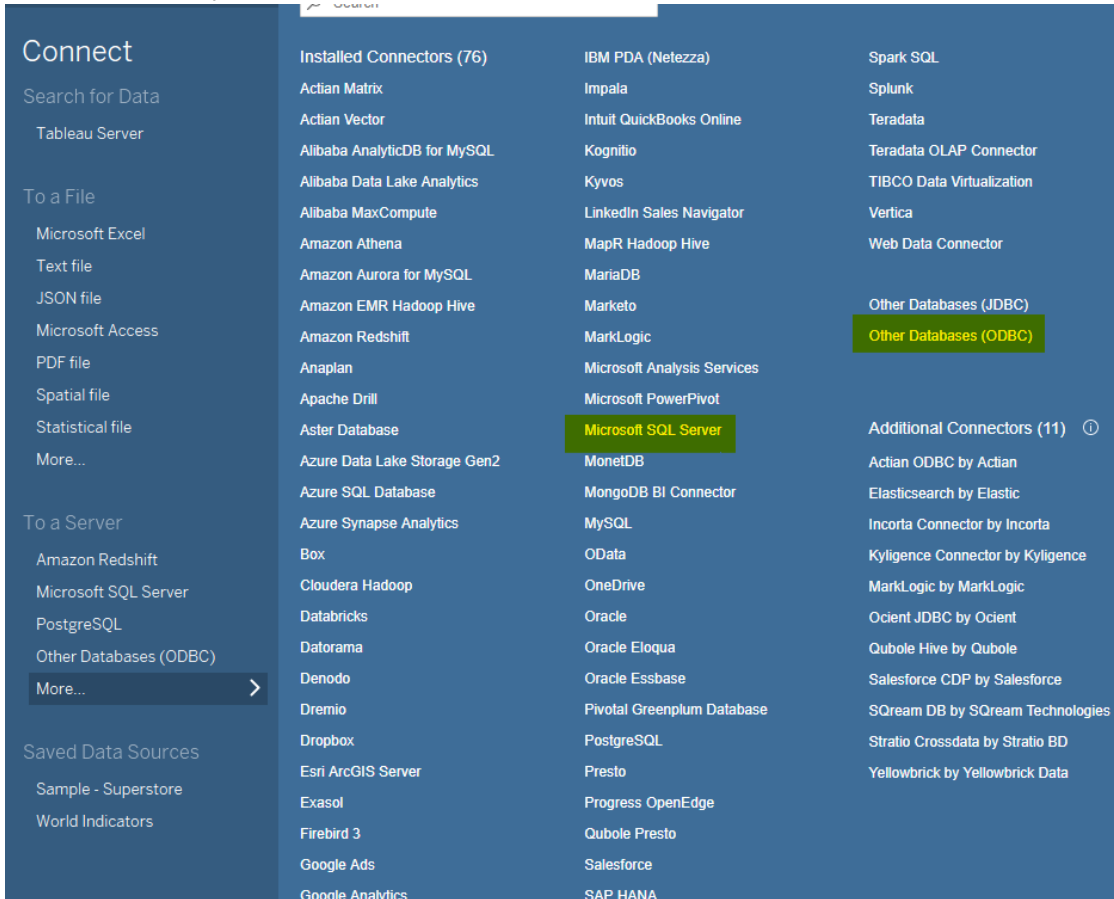
¹ Both Tableau Desktop/Server and Cognos server must be secured by the same Active Directory service.

² 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.

³ Tableau installer comes with its own PostgreSQL ODBC driver, but that driver is an older version that is not compatible Senturus Analytics Connector server.

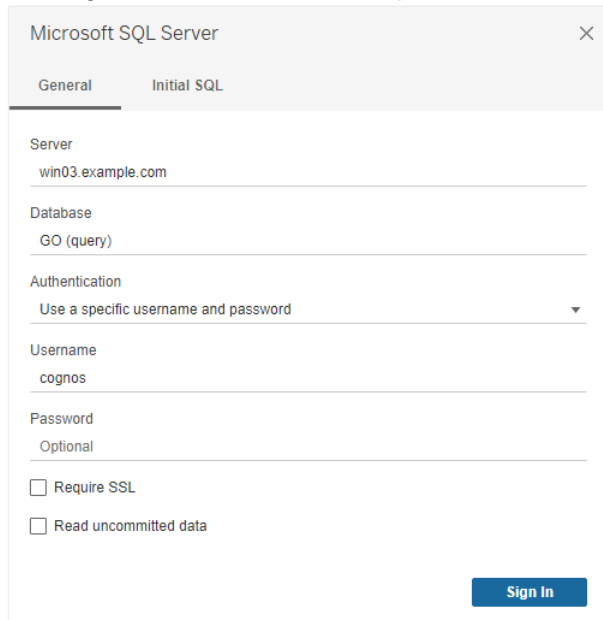
Connecting to Cognos from Tableau

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



Use Microsoft SQL Server Connection

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. The dialog has two tabs: "General" (selected) and "Initial SQL". The "General" tab contains the following fields and options:

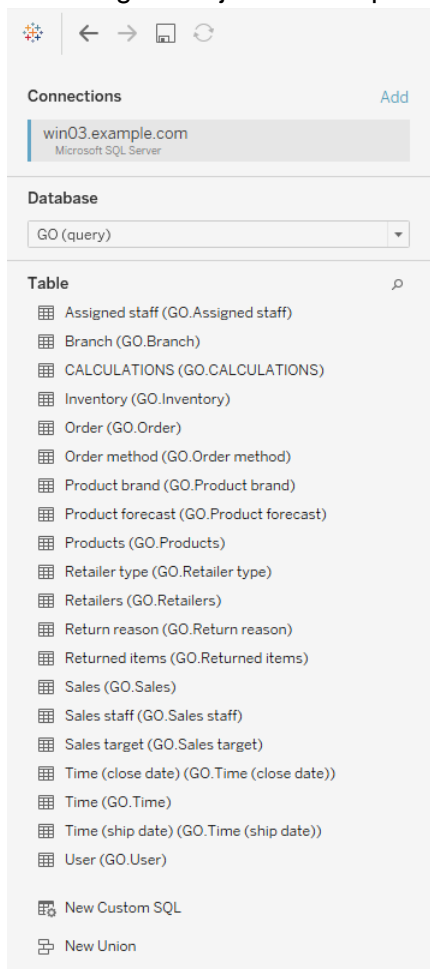
- Server:** win03.example.com
- Database:** GO (query)
- Authentication:** Use a specific username and password (dropdown menu)
- Username:** cognos
- Password:** Optional
- Require SSL
- Read uncommitted data

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

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.

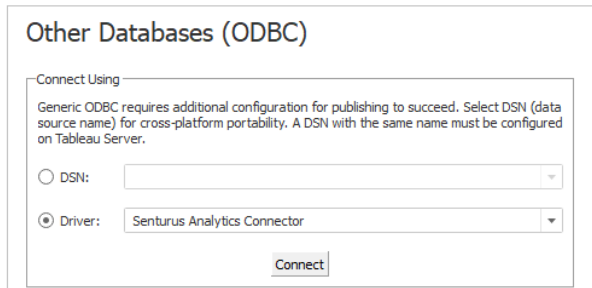
After clicking on “Sign in” button, Tableau will connect to Analytics Connector Server and bring back Cognos objects and represent them as tables, as below.



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.

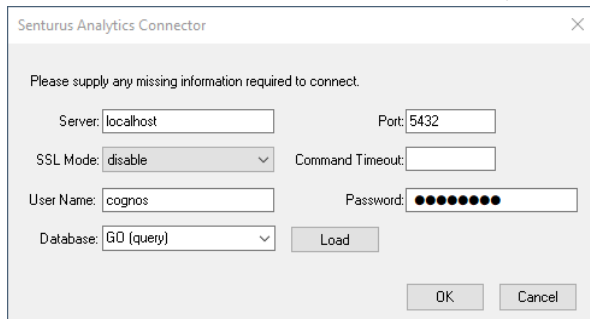
DSN:

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.

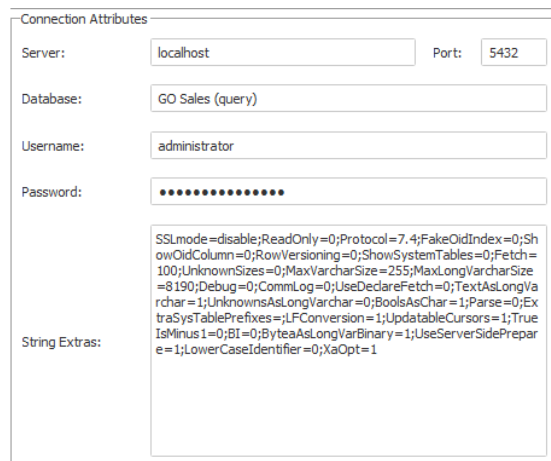
Server: Port:

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:

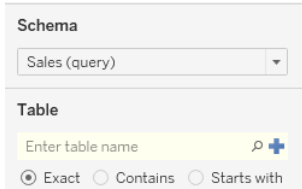
Tableau will connect to Analytics Connector Server and bring back Cognos objects and represent them as tables, as below.

The screenshot shows the Tableau interface for connecting to a data source. At the top, there are navigation icons (home, back, forward, refresh). Below that is the 'Connections' section with an 'Add' button and a list containing 'localhost' with the subtext 'Other Databases (ODBC)'. The 'Database' dropdown is set to 'GO Sales (query)'. The 'Schema' dropdown is set to 'Sales (query)'. The 'Table' section has a search bar with the placeholder 'Enter table name' and a search icon. Below the search bar are three radio buttons: 'Exact' (selected), 'Contains', and 'Starts with'. A list of tables is displayed, each with a grid icon and a plus sign. The tables are: Branch (Sales (query).Branch), CALCULATIONS (Sales (query).CALCULATIONS), Order (Sales (query).Order), Order method (Sales (query).Order method), Parameterized Products (Sales (query).Parameterized Products), Parameterized Products 2 (Sal...ry).Parameterized Products 2), Products (Sales (query).Products), Retailer type (Sales (query).Retailer type), Retailers (Sales (query).Retailers), Sales (Sales (query).Sales), Sales staff (Sales (query).Sales staff), Time (close date) (Sales (query).Time (close date)), Time (Sales (query).Time), Time (ship date) (Sales (query).Time (ship date)), User (Sales (query).User), and New Custom SQL.

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.



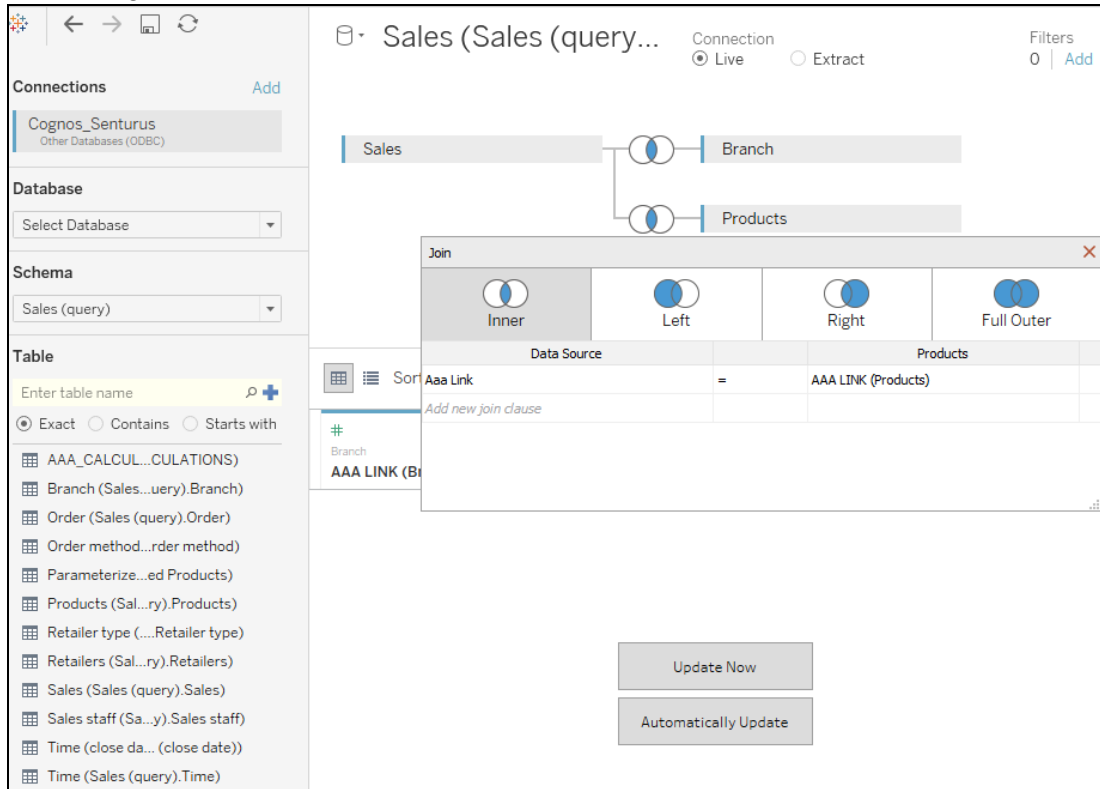
The screenshot shows the Tableau Data Source configuration window. It is divided into two main sections: 'Schema' and 'Table'. The 'Schema' section has a search field containing 'Sales (query)'. The 'Table' section has a search field containing 'Enter table name' and a search icon. Below the search field are three radio buttons: 'Exact' (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.



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*⁴. The link column name can be changed using the Configuration Utility.

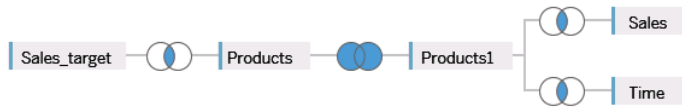
Tableau will automatically join the tables using the link columns, but the ODBC driver will ignore them when passing the queries to Cognos.

Custom Joins

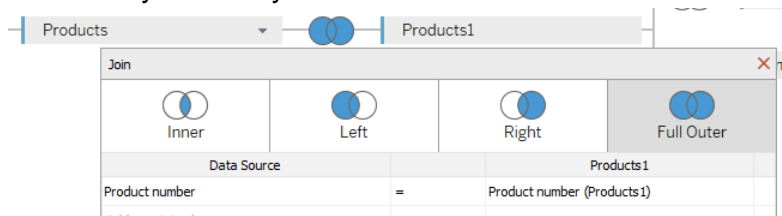
As mentioned previously, Tableau will automatically join on the link columns the Analytics Connector creates. Usually, you will not want to change these joins because you want to leverage the underlying Cognos relationships. However, you can create custom joins to build Tableau

⁴ AAA_LINK is the default dummy column name, but your Senturus Analytics Connector server administrators can change it via Data Source Configuration tool.

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.



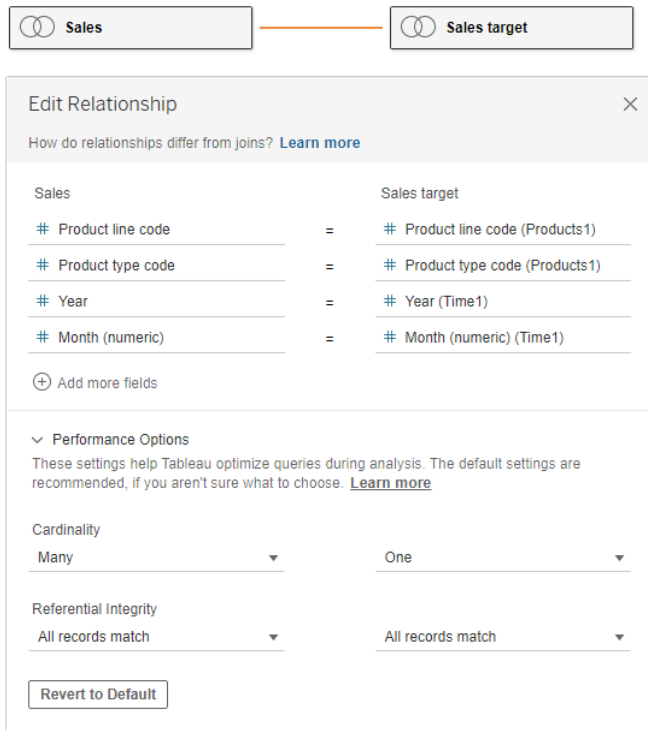
Additionally, the Analytics Connector will split the five tables into two sub queries. The first sub query contains *Sales_target* and *Products* and the second sub query has *Products1*, *Sales* and *Time*. The results of the two sub queries will be joined together by the Product Number column.

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



Cognos Calculations

The Analytics Connector supports working with Cognos Calculations in Tableau.

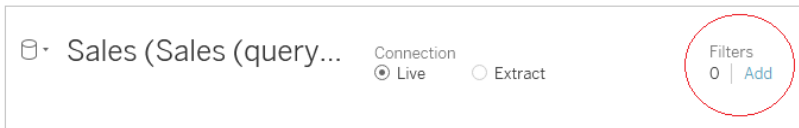
Calculations in Cognos come across in a special table named *AAA_CALCULATIONS* or *_CALCULATIONS_* under the appropriate schema. The calculations table name can be changed using the **Configuration Utility**.

The table will also contain a link column. The special calculation table is joined to other tables using the link column like any other table in the data source.

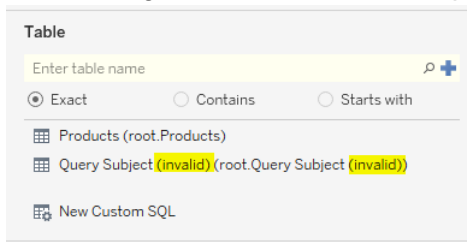


Data Source Best Practices

- 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.
- Do NOT click **Update Now** on the Data Source tab or you will trigger a large query that will take a very long time to run.
- Leave the connection set to **Live** so that queries are passed to Cognos for execution under the authenticated user.
- 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 have duplicate columns in a table, the table will be marked as “(invalid)” and will be unusable in your data source. You will need to correct your Cognos model to resolve a duplicate column issue. Review the java.log file found at {Analytics Connector Install Folder}\Log for the name of the duplicated column.



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

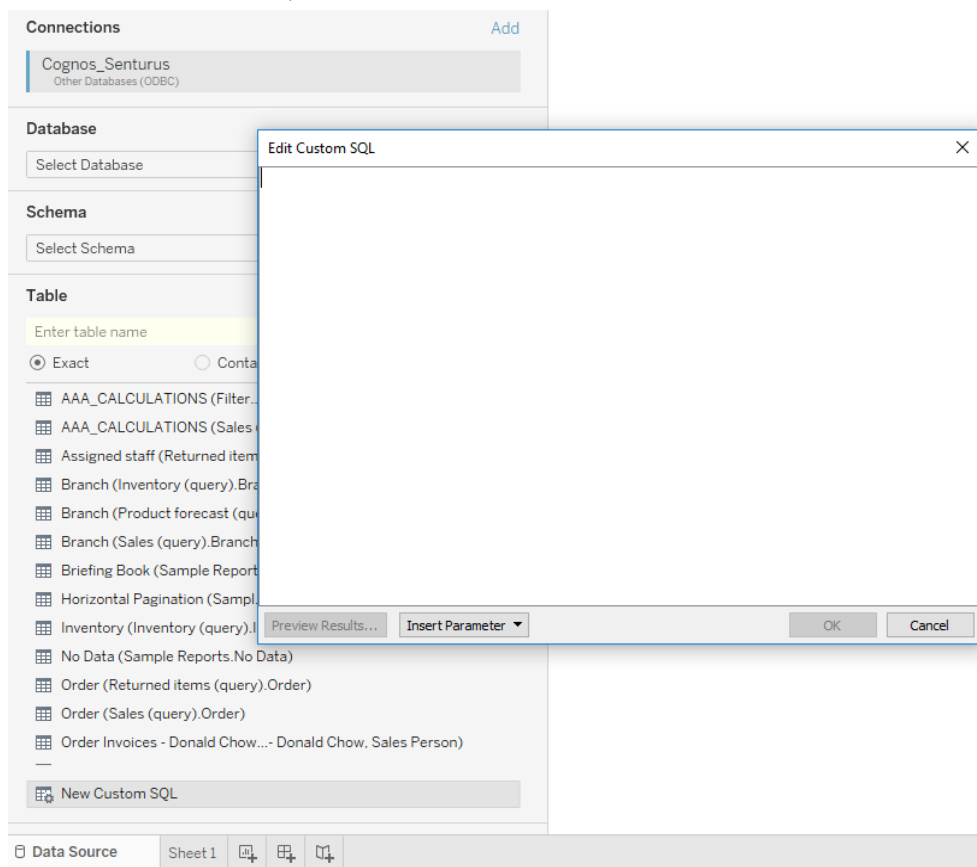
Regular database tables do not have or support parameters. But when working with a Cognos parameterized query subject or a report, users will be prompt to answer parameter values before it can be executed.

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

```

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""
)

```

The bottom window, titled 'View Data: GO Sales (query)', displays a table with 15 rows. The table has four columns: 'Discontinued date', 'Introduction date', 'Product', and 'Product brand'. The 'Discontinued date' column contains 'Null' for all rows. The 'Introduction date' column contains dates in the format 'MM/DD/YYYY HH:MM:SS AM'. The 'Product' and 'Product brand' columns contain the names of the products and their respective brands.

Discontinued date	Introduction date	Product	Product brand
Null	1/10/2010 12:00:00 AM	Course Pro Golf and Tee Set	Course Pro
Null	1/15/2010 12:00:00 AM	Course Pro Umbrella	Course Pro
Null	12/15/2009 12:00:00 AM	Course Pro Golf Bag	Course Pro
Null	12/15/2009 12:00:00 AM	Hailstorm Steel Irons	Hailstorm
Null	12/27/2009 12:00:00 AM	Course Pro Gloves	Course Pro
Null	12/10/2009 12:00:00 AM	Course Pro Putter	Course Pro
Null	12/10/2009 12:00:00 AM	Lady Hailstorm Steel Irons	Hailstorm
Null	12/27/2009 12:00:00 AM	Hailstorm Titanium Woods Set	Hailstorm
Null	12/10/2009 12:00:00 AM	Hailstorm Titanium Irons	Hailstorm
Null	12/18/2009 12:00:00 AM	Lady Hailstorm Titanium Irons	Hailstorm

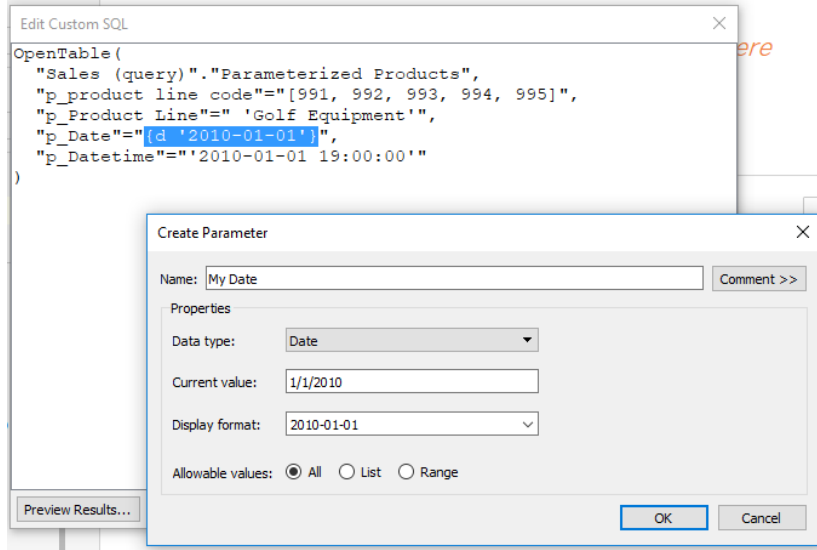
The first parameter is “[*schema name*].”[*table name*]” of the Cognos parameterized table. This is followed by zero or more Cognos parameter name-value pairs separated by commas.

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

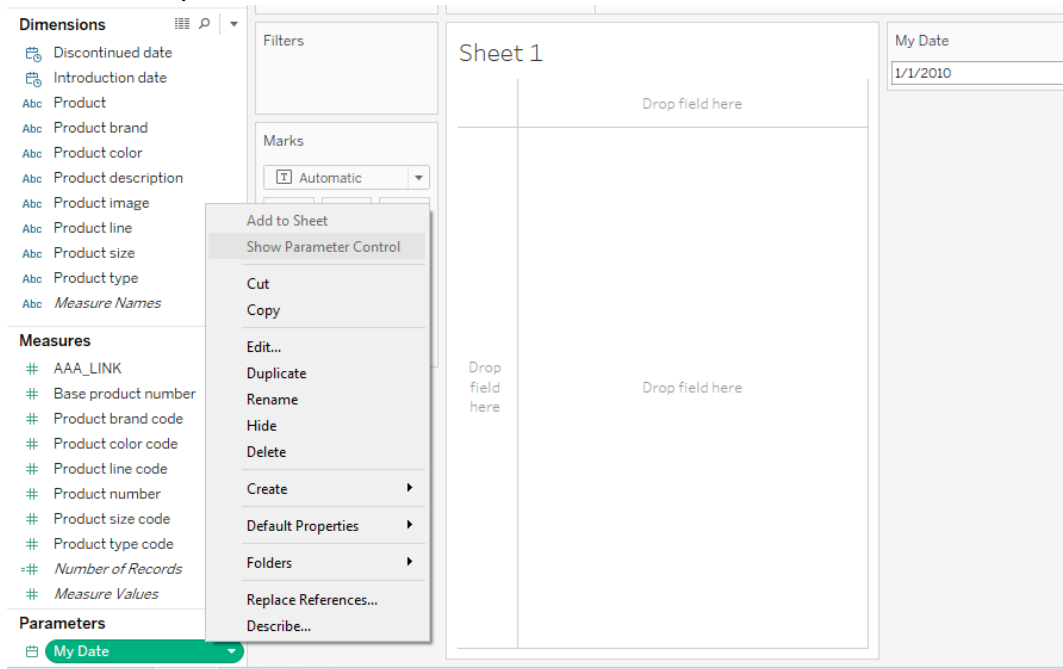
For example:

- 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.

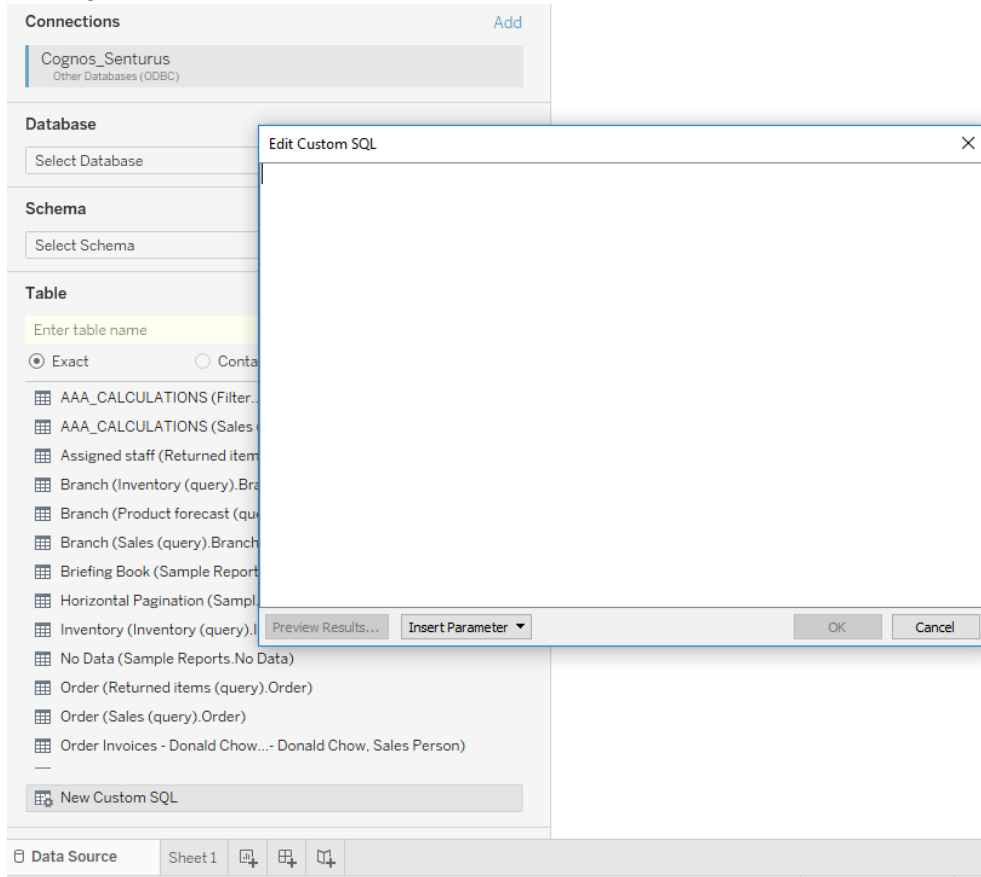
You can also mix parameterized tables with other tables in one Tableau data source. These tables will also join using the generated Link Columns. Example advanced query:

```
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.

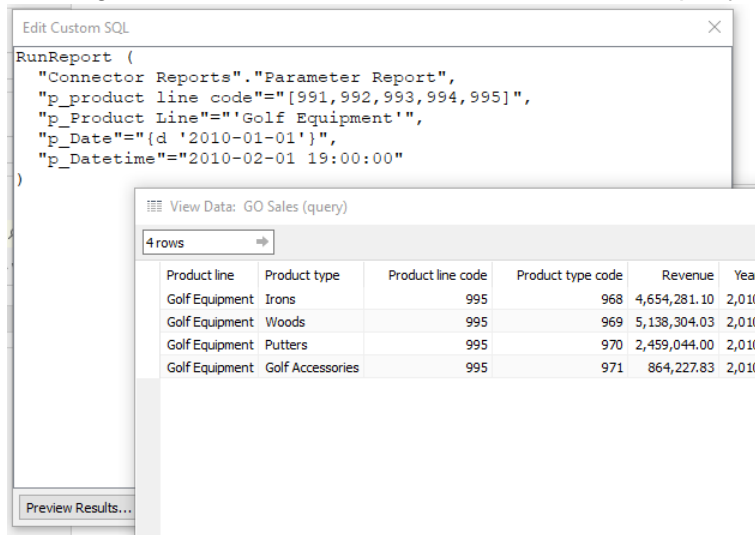


Using the *RunReport* function, you can pass parameters to a Cognos Report.

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 a window titled "Edit Custom SQL" with a SQL query. Below the query, a "Preview Results..." button is visible. A separate window titled "View Data: GO Sales (query)" is overlaid, showing a table with 4 rows and 6 columns: Product line, Product type, Product line code, Product type code, Revenue, and Year.

```
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"
)
```

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

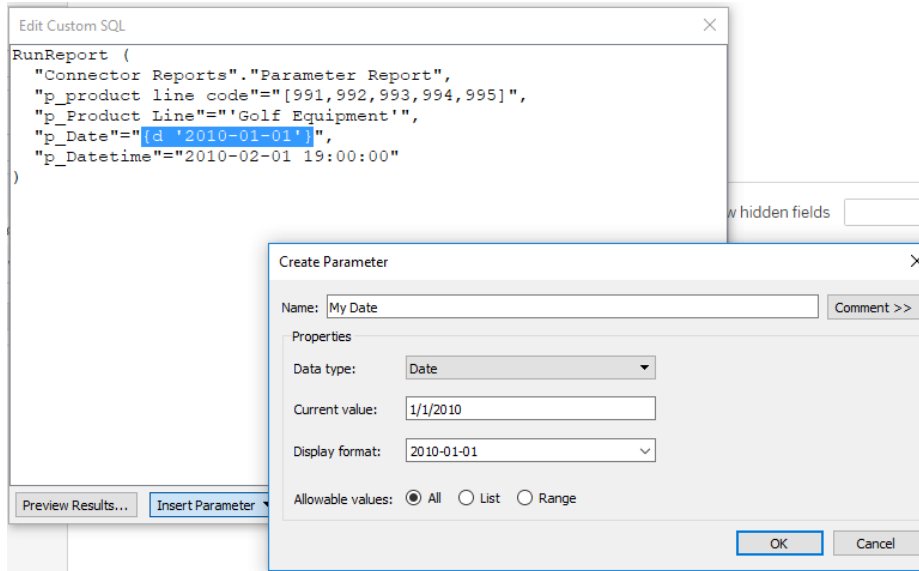
The first parameter is "[schema name]".[table name]" of the Cognos Report. This is followed by zero or more Cognos parameter name-value pairs separated by commas.

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

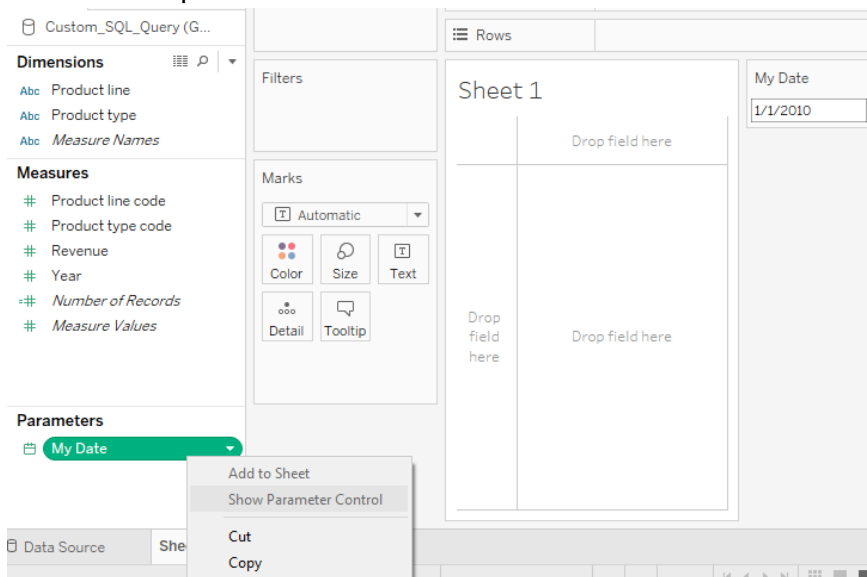
Example:

- 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

Q: How does a SQL Server nullable uniqueidentifier column work with Cognos and the Analytics Connector?

A uniqueidentifier is typically mapped to a varchar(50) or nvarchar(50) in Cognos models. An issue arises when a nullable uniqueidentifier has a null value. When the output is sent to CSV, the null becomes a '\0' which isn't a valid uniqueidentifier and will cause an error.

This can be worked around by adjusting the model to "CAST([uniqueidentifier_column] as nvarchar(50))" instead of just "[uniqueidentifier_column]".

This can also be worked around at the Tableau level using the Custom SQL feature and applying the same "CAST([uniqueidentifier_column] as nvarchar(50))" instead of just "[uniqueidentifier_column]" as part of the SELECT statement.

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.