

# Senturus Analytics Connector

## User Guide

Cognos to Tableau

## Overview

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

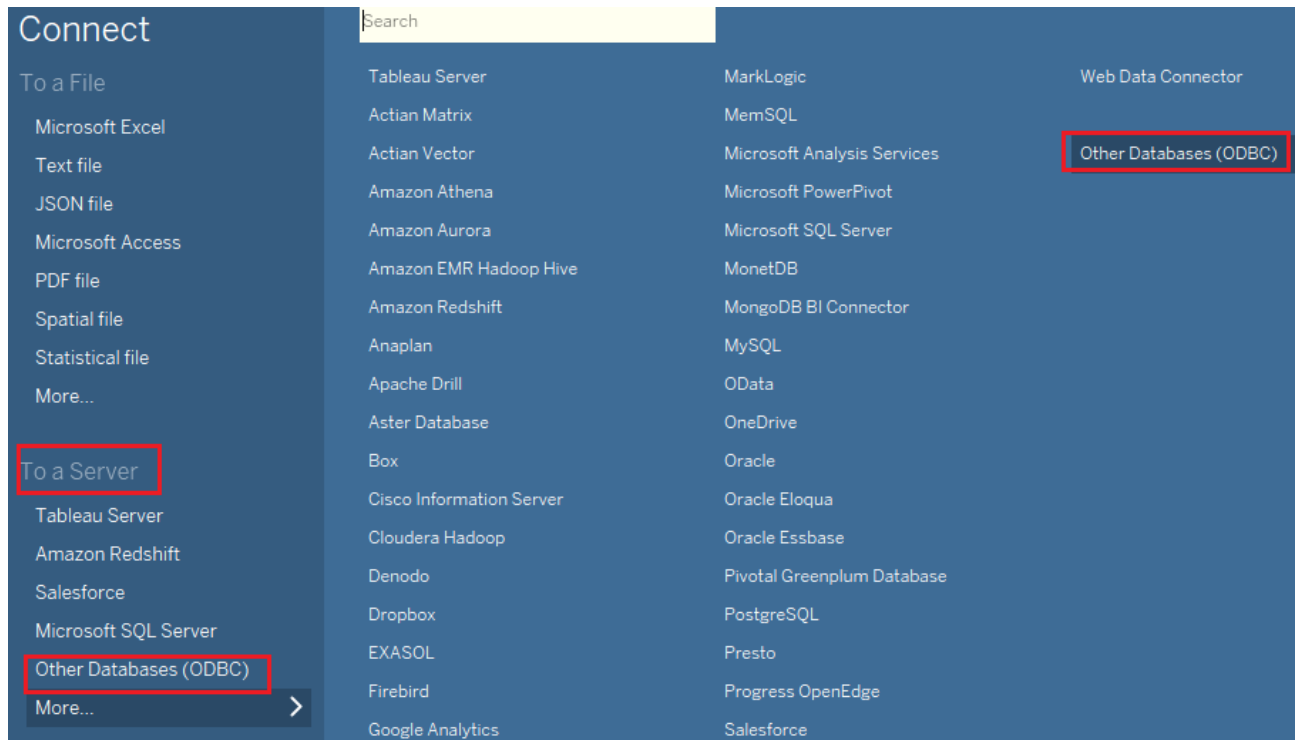
Please refer to the Installation Guide for instructions on installing and configuring the Analytics Connector.

The Analytics Connector has been tested against different versions of Tableau, including both Tableau Desktop and Tableau Server on Windows from version 9.3 to the latest 2018.3.

## Connecting to Cognos from Tableau

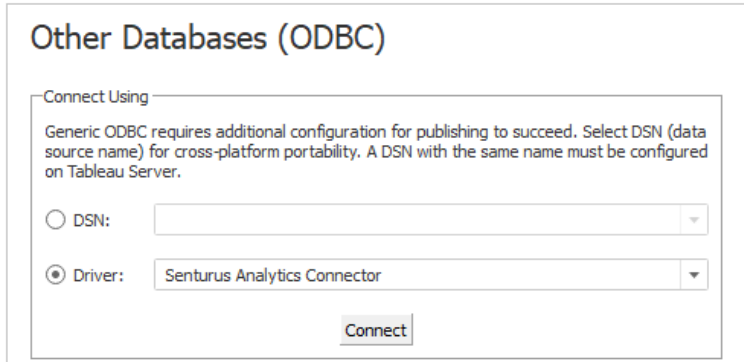
You can use a pre-configured ODBC DSN or the Analytics Connector ODBC driver directly to connect Tableau to a Cognos data source.

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



## Driver

To use the driver directly, select the radio button next to **Driver**, select *Senturus Analytics Connector* from the list, then click **Connect**.



Other Databases (ODBC)

Connect Using

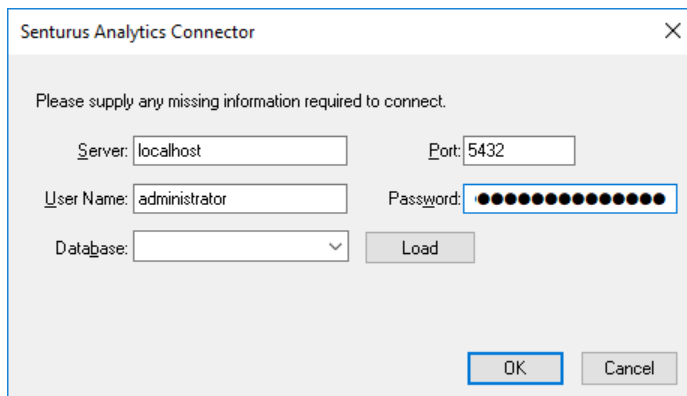
Generic ODBC requires additional configuration for publishing to succeed. Select DSN (data source name) for cross-platform portability. A DSN with the same name must be configured on Tableau Server.

DSN:

Driver:

**Enter** your 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.



Senturus Analytics Connector

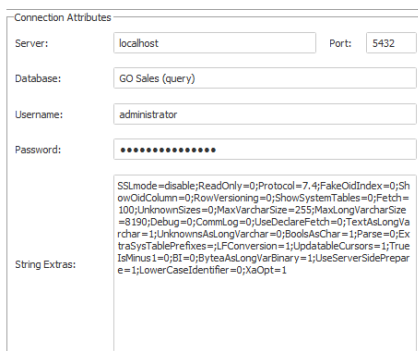
Please supply any missing information required to connect.

Server:  Port:

User Name:  Password:

Database:

Select a Database from the loaded list, then click **OK**.



Connection Attributes

Server:  Port:

Database:

Username:

Password:

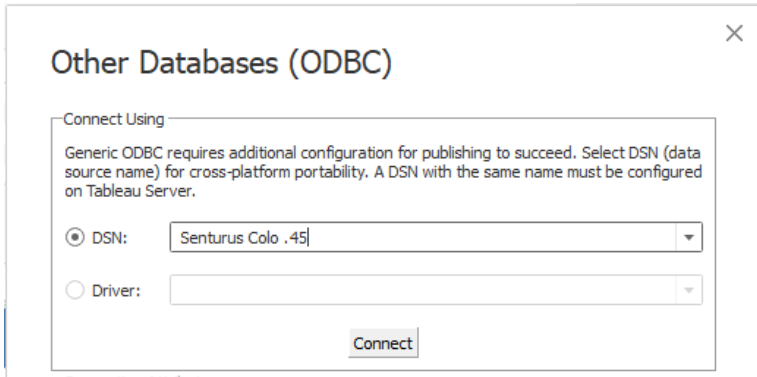
String Extras: 

```
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;UnknownAsLongVarchar=0;BlobAsChar=1;Parse=0;ExtraSysTablePrefixes=;LCConversion=1;UpdatableCursors=1;TrueIsMinus1=0;BI=0;ByteAsLongVarBinary=1;UseServerSidePrepare=1;LowerCaseIdentifier=0;XaOpt=1
```

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

## DSN

If you prefer to configure and use a DSN, select the radio button next to **DSN**, select the name of the configured DSN from the list, then click **Connect**.



Other Databases (ODBC)

Connect Using

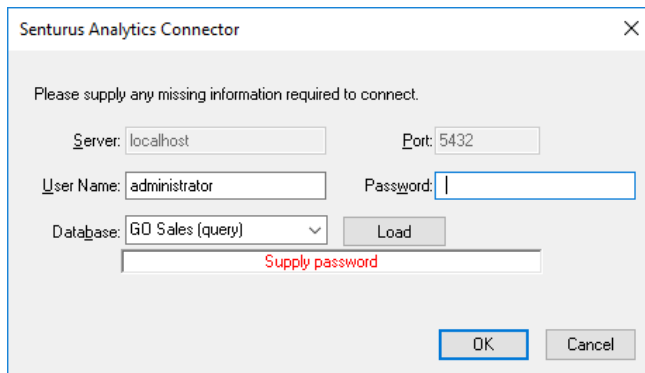
Generic ODBC requires additional configuration for publishing to succeed. Select DSN (data source name) for cross-platform portability. A DSN with the same name must be configured on Tableau Server.

DSN: Senturus Colo .45

Driver:

Connect

If you did not save the username and/or password in the DSN, you will be prompted for it. The Server and Port will be read only.



Senturus Analytics Connector

Please supply any missing information required to connect.

Server: localhost Port: 5432

User Name: administrator Password:

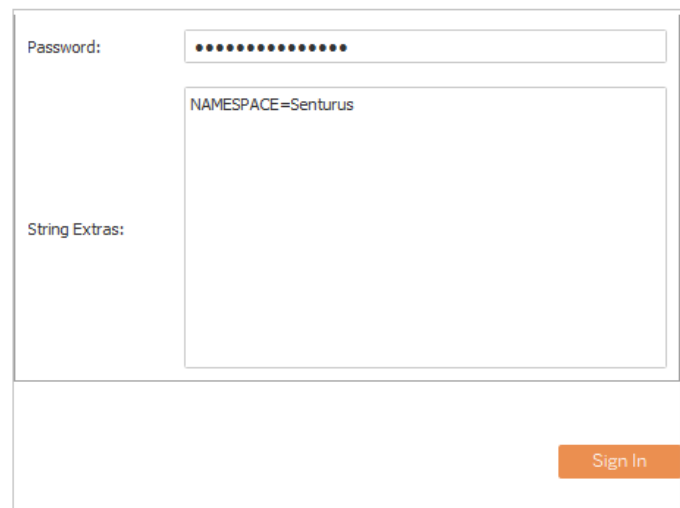
Database: GD Sales (query) Load

Supply password

OK Cancel

Click **Sign In** to complete the connection.

You will then see the Tableau Data Source tab for this connection.



Password: .....

String Extras:



NAMESPACE=Senturus

Sign In

## Working with Tables

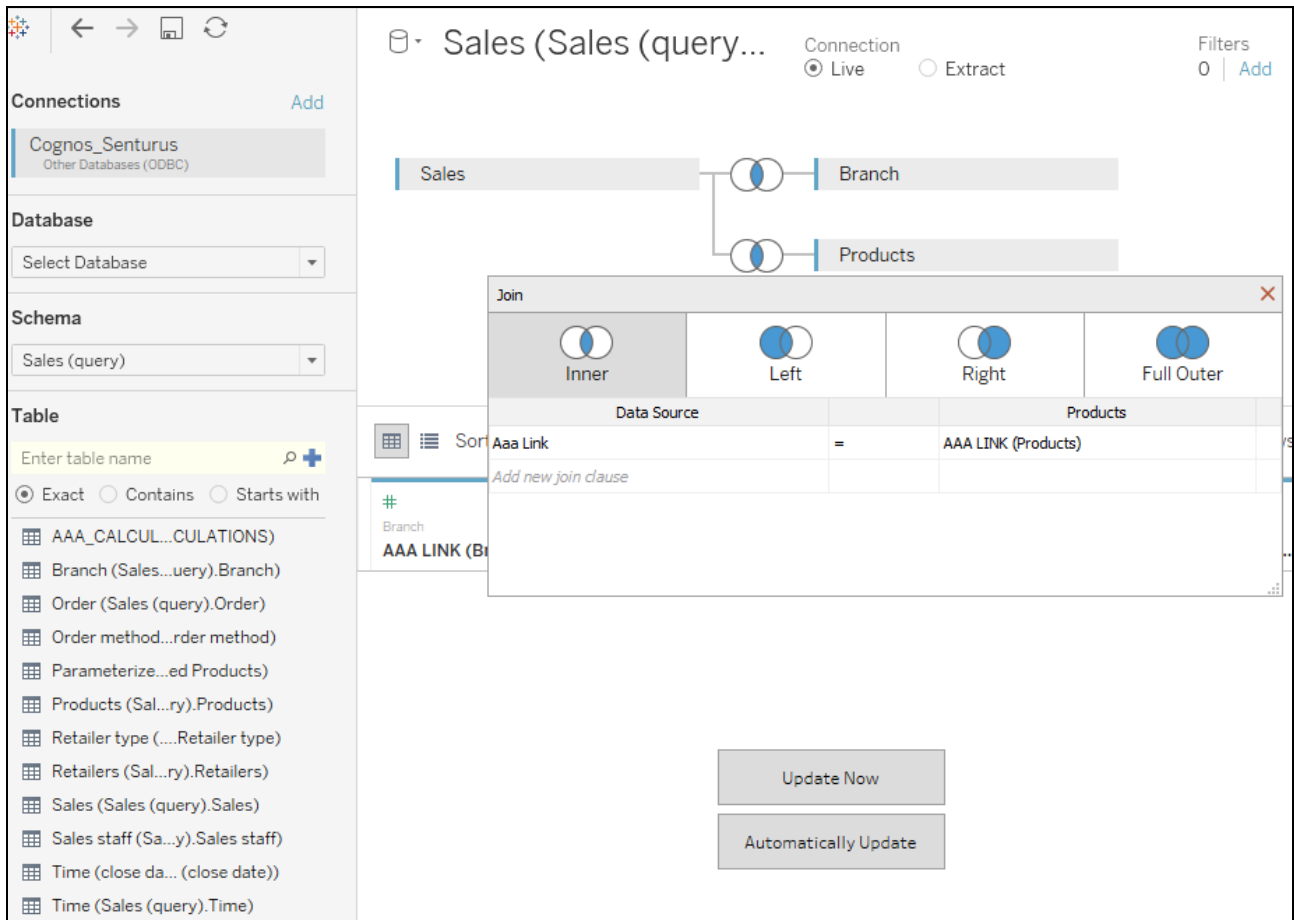
After connecting successfully, the database, schema and tables will show on the **Data Source** tab.

**Table**

Enter table name   

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.

Drag or double click the desired tables into the data source.



The screenshot shows the Tableau interface with a data source named "Sales (Sales (query...))". The interface includes a left sidebar with sections for "Connections", "Database", "Schema", and "Table". The "Table" section lists various tables, including "AAA\_CALCUL...CULATIONS)", "Branch (Sales...query).Branch", "Order (Sales (query).Order)", "Order method...rder method)", "Parameterize...ed Products)", "Products (Sal...ry).Products)", "Retailer type (...Retailer type)", "Retailers (Sal...ry).Retailers)", "Sales (Sales (query).Sales)", "Sales staff (Sa...y).Sales staff)", "Time (close da... (close date))", and "Time (Sales (query).Time)".

The main area shows a diagram of the data source with three tables: "Sales", "Branch", and "Products". The "Sales" table is connected to "Branch" and "Products" via join symbols. A "Join" dialog box is open, showing the "Inner" join type selected between "Aaa Link" and "AAA LINK (Products)". The dialog also shows "Update Now" and "Automatically Update" buttons.

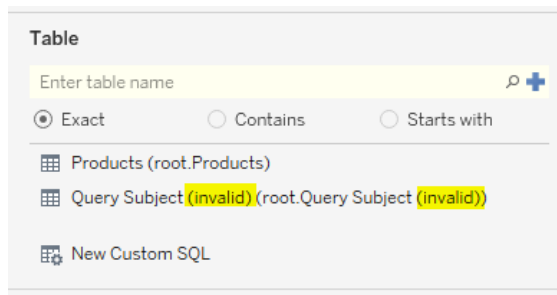
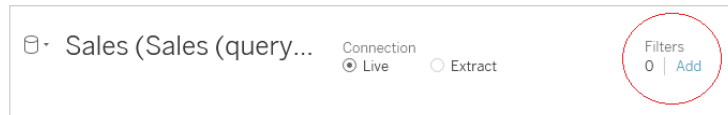
## Link Columns

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* or *\_LINK\_COLUMN\_*. 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.

## Data Source Best Practices

- 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 window. Appropriate filters can improve query performance when designing the visualizations.
- If you have duplicate columns configured in a package, the table containing the duplicated column 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.



## Design Visualizations

To begin designing your visualization, change to the **Sheet** tab (typically named “Sheet 1”)

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



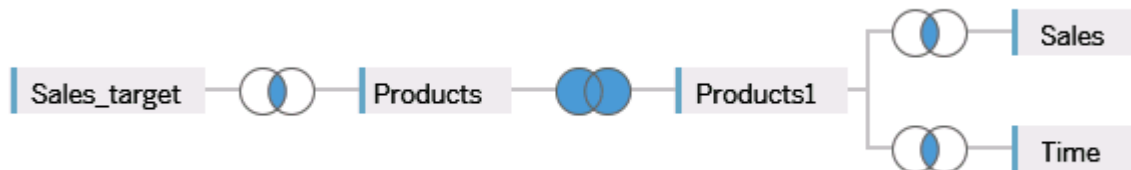
Tableau uses several special features to optimize queries, but the Analytics Connector and Cognos do not support all of them. This dialog box is only a warning and your workbook will function correctly.

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

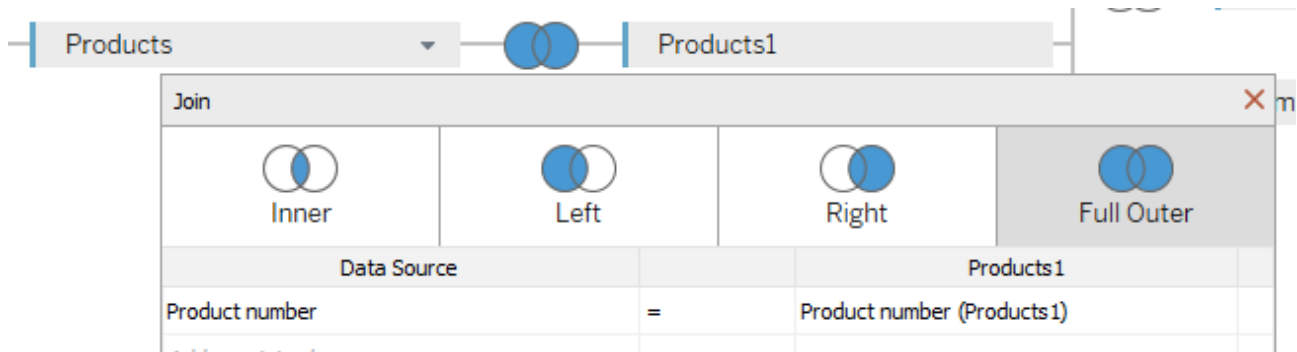
You are now able to design your visualization using the Cognos Dimensions and Measures in Tableau.

## Working with 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.



## Working with 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**.

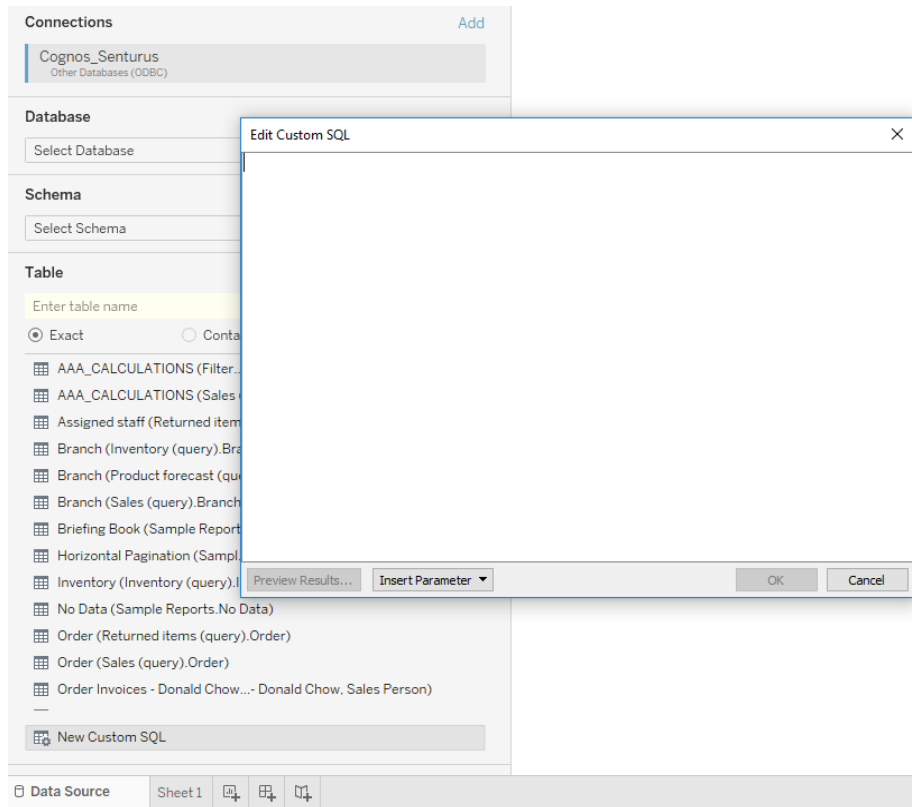
The table will also contain a link column. The special calculation table is joined to other tables using the link column like any other table in the data source.



## Working with Parameterized Tables

The Analytics Connector supports working with Cognos parameterized tables in Tableau. 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 code:

```
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 and times. The "Product" column contains various golf equipment items, and the "Product brand" column contains brand names like "Course Pro" and "Hailstorm".

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

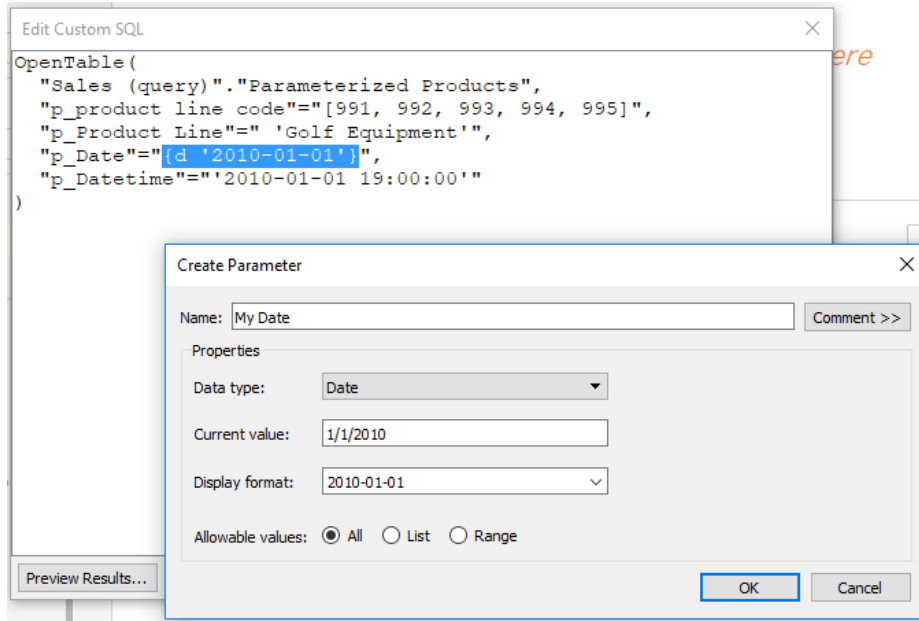
The first parameter is "[schema name]".[table name]" of the Cognos parameterized table. This is followed by zero or more Cognos parameter name-value pairs separated by commas.

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

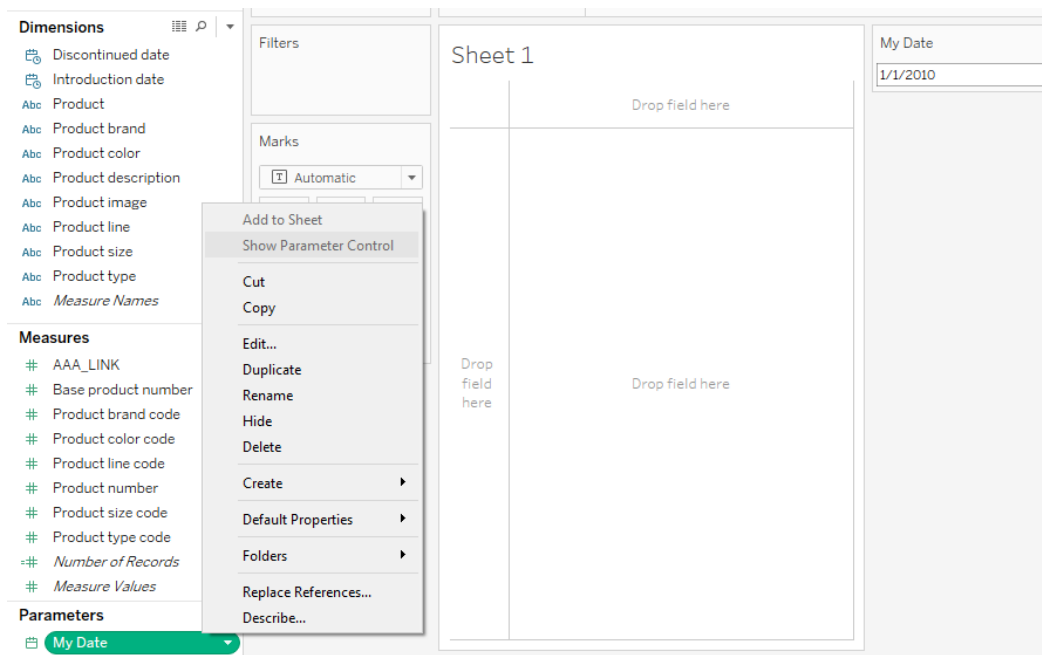
For example:

- 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:

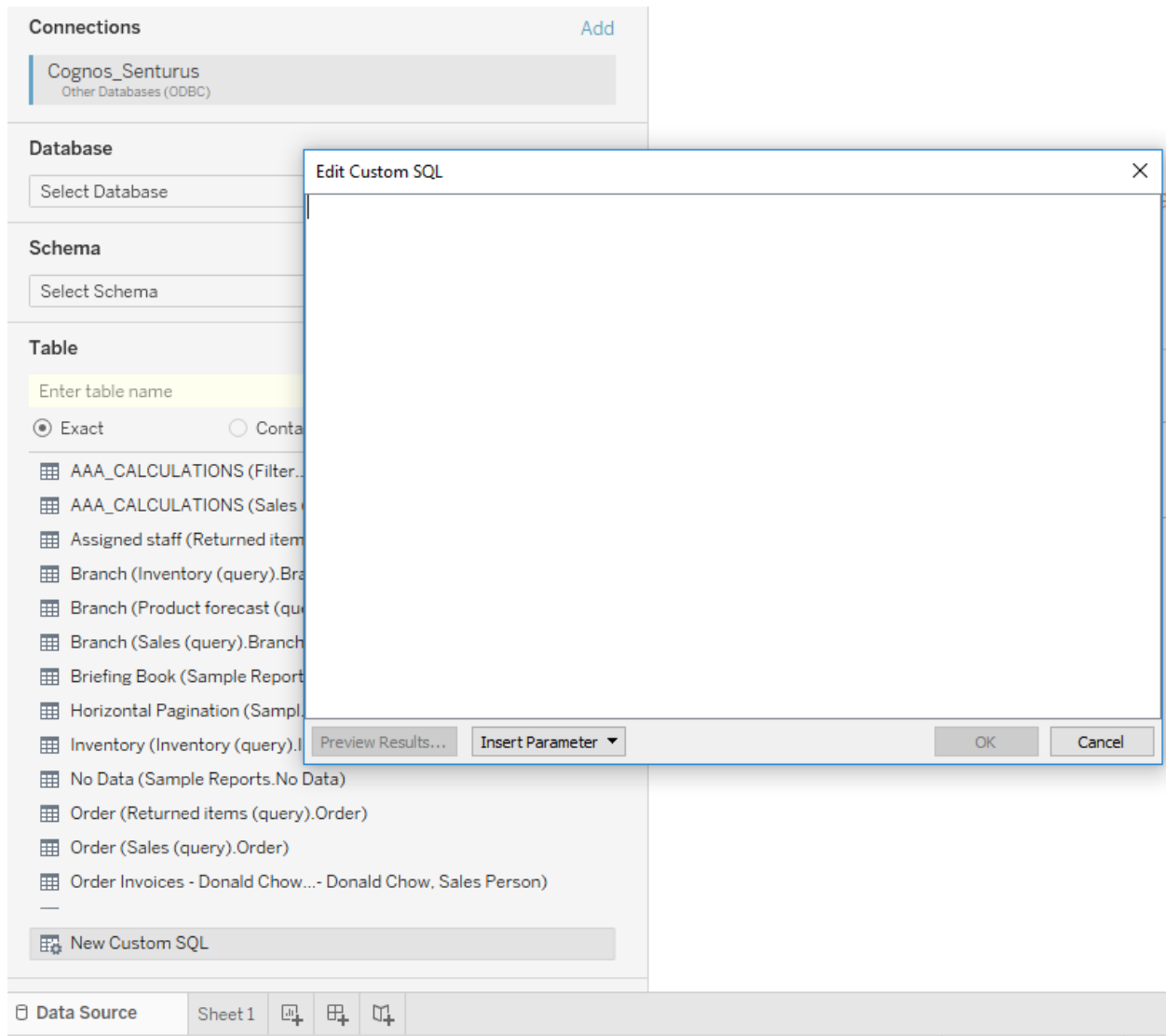
```
Select
  "Products"."Product line",
  "Products"."Product type",
  Sum("Sales"."Revenue") as "Revenue"
From
  OpenTable(
    "Sales (query)"."Parameterized Products",
    "p_product line code"="[991, 992, 993, 994, 995]",
    "p_Product Line"=" 'Golf Equipment'",
    "p_Date"="{d '2010-01-01'}",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

The Analytics Connector supports working with Cognos Reports in Tableau.

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_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" containing the following SQL code:

```
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"  
)
```

Below the SQL editor, a "View Data: GO Sales (query)" window is open, displaying 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 window, there is a "Preview Results..." button.

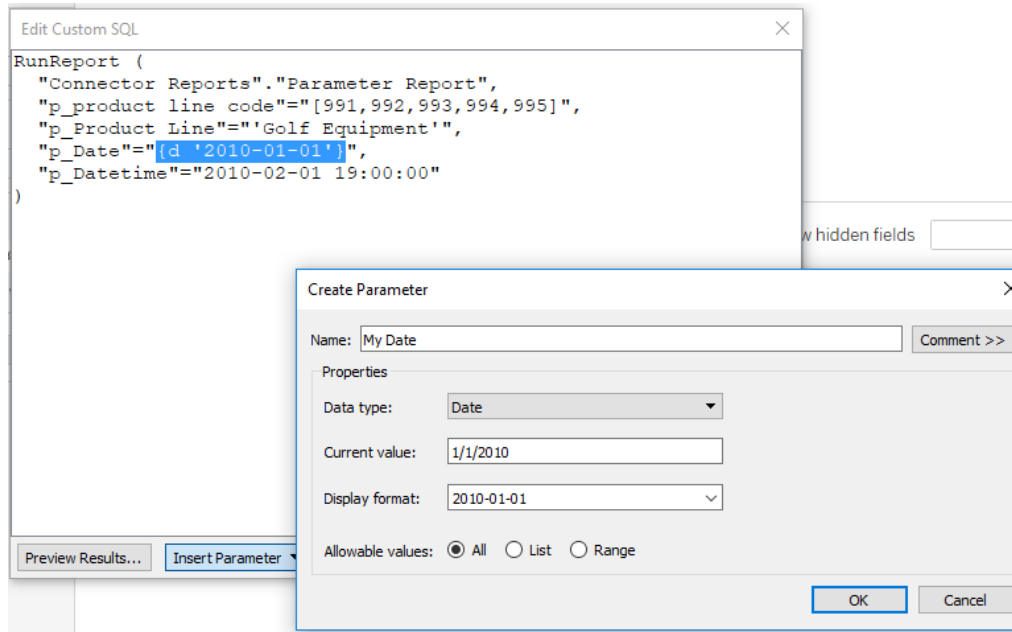
The first parameter is "[schema name]".[table name]" of the Cognos Report. This is followed by zero or more Cognos parameter name-value pairs separated by commas.

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

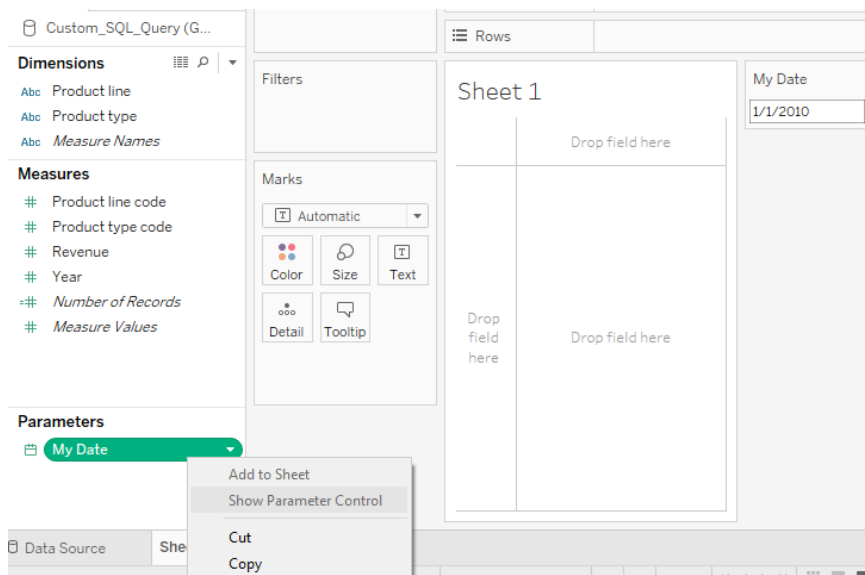
Example:

- 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.



## Technical Reference

### 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.

#### **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>]".

## Supported ODBC Features

All identifiers (catalog, schema, table and column names) are case insensitive. This ODBC driver supports 15 Cognos data types. Refer to the data type mapping in following table.

<b>Cognos Data Type</b>	<b>ODBC Data Type</b>	<b>Max Length/Precision</b>	<b>Note</b>
int16	smallInt	5	
int32	integer	10	
int64	bigInt	19	
float32	real	7	
float64	double	15	
decimal	decimal	38	
character	char	8000	
characterLength16	varchar	8000	
nChar	nchar	4000	
nVarChar	nvarchar	4000	
date	date	10	
time	time	8	no milliseconds
datetime	timestamp	19	no milliseconds
textBlob	varchar	8000	mapped to varchar
unknown	varchar	8000	mapped to varchar

This driver has been tested against SQL Server, DB2 and Oracle databases (relational and DMR model). It may not support all data types/functions for other databases.

This driver only supports a) packages with one data source or b) packages with multiple data sources. However, all database connections are of the same type (e.g. SQL Server native connection). It may work for other packages, but some functions may fail.

### ODBC features supported by this driver

Numeric, string, date/time, null literals

Cast/convert

Simple calculation (+, -, \*, /, and %)

String concatenate\And, or, not logical operators

Comparison (<, >, =, <=, >=, !=, between, not between, in, not in, like, not like)

Is null, is not null

Case (if else) statement

Parenthesis

Sub queries

### Functions supported by this driver

**Aggregate function:** AVERAGE, AVG, COUNT, MAX, MAXIMUM, MIN, MINIMUM, SUM, TOTAL.

**Numeric functions:** ABS, ACOS, ASIN, ATAN, CEILING, COS, COT, DEGREES, EXP, FLOOR, LOG, LOG10, MOD, POWER, RADIANS, ROUND, SIGN, SIN, SQRT, TAN, TRUNCATE.

**Date and timestamp functions:** CURRENT\_DATE, CURRENT\_TIMESTAMP, DAY, DAYNAME, DAYOFMONTH, DAYOFWEEK, DAYOFYEAR, HOUR, MINUTE, MONTH, MONTHNAME, QUARTER, SECOND, TIMESTAMPADD, TIMESTAMPDIFF, WEEK, YEAR.

**Character functions:** ASCII, CHAR, CHAR\_LENGTH, CHARACTER\_LENGTH, CONCAT, LCASE, LCASE, LEFT, LOCATE, LTRIM, LTRIM, REPLACE, RIGHT, RTRIM, RTRIM, SPACE, SUBSTR, SUBSTRING, TRIM, TRIM, UCASE, UCASE.

Others: CONVERT, IFNULL

## 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](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](mailto:CustomerSupport@senturus.com).