Oracle® Analytics

User’s Guide for Oracle Analytics Desktop

E70158-38
October 2021

Oracle Analytics User’s Guide for Oracle Analytics Desktop,

E70158-38

Primary Author: Stefanie Rhone

Contributing Authors: Nick Fry, Pete Brownbridge

Contributors: Oracle Analytics development, product management, and quality assurance teams

Contents

Preface

1 Get Started with Oracle Analytics Desktop

2 Connect to Data Sources

4 Enrich and Transform Data

5 Curate Your Data Using Data Flows

E Data Preparation Reference

F Expression Editor Reference

G Oracle Analytics Desktop SDK Reference

Preface

Learn how to explore data using Oracle Analytics Desktop.

Topics

- Audience
- Documentation Accessibility
- Related Resources
- Conventions

Audience

User's Guide for Oracle Analytics Desktop is intended for business users who upload and query data sources, create visualizations to analyze data, build and work with workbooks, and import and export workbooks.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Resources

These related Oracle resources provide more information.

- Oracle Analytics Product Information
- Oracle Community Forum
- Oracle Analytics Desktop Installation Download
- Oracle Analytics Library

Conventions

Conventions used in this document are described in this topic.

Text Conventions

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Videos and Images

Your company can use skins and styles to customize the look of the application, dashboards, reports, and other objects. It is possible that the videos and images included in the product documentation look different than the skins and styles your company uses.

Even if your skins and styles are different than those shown in the videos and images, the product behavior and techniques shown and demonstrated are the same.

1 Get Started with Oracle Analytics Desktop

This topic describes the benefits of using Oracle Analytics Desktop and how to get started using the samples provided.

▶ Video

Topics:

- About Oracle Analytics Desktop
- Get Started with Sample Workbook

About Oracle Analytics Desktop

Oracle Analytics Desktop provides standalone data exploration and visualization in a per-user desktop download. Oracle Analytics Desktop is the tool for quick exploration of sample data from multiple sources or for analyses and investigation of your own local datasets.

Oracle Analytics Desktop enables you to visualize your data so you can focus on exploring data patterns. Just upload data files or connect to Oracle Applications or a database, select the elements that you're interested in, and let Oracle Analytics Desktop find the best way to visualize it. Choose from a variety of visualizations to look at data in a specific way.

Oracle Analytics Desktop also gives you a preview of the self-service visualization capabilities included in Oracle Analytics Cloud, Oracle's cloud analytics platform. Oracle Analytics Cloud extends the data exploration and visualization experience by offering secure sharing and collaboration across the enterprise, additional data sources, and a full mobile experience including proactive self-learning analytics delivered to your device.

Get Started with Sample Workbook

Use the sample workbook provided to explore and discover Oracle Analytics Desktop's capabilities. Sample Workbook is designed to teach you best practices, which you can then use to create your own workbooks.

Sample Workbook uses the Sample Order Lines and Sample States spreadsheets as its dataset. These spreadsheets contain meaningful dimensions, distributions, examples of data wrangling, calculated columns, and more. The visualizations included in Sample Workbook illustrate how to create different views of the data and arrange them on multiple canvases to provide a comprehensive look at the data.

You can choose to download Sample Workbook during installation. If you didn't download Sample Workbook during installation, then you can still get it by uninstalling and then reinstalling Oracle Analytics Desktop. Your personal data isn't deleted when you uninstall and reinstall.

Oracle also provided examples designed to help you enhance your data visualizations. The examples are .dva files that you can re-use with your own data. See Oracle Analytics Examples Library.

2

Connect to Data Sources

Create data source connections to access the data that you want to analyze and explore. Examples of data sources are: a database, Oracle applications, or a snowflake data warehouse.

Topics:

About Data Sources

A data source is any tabular structure. You get to see data source values after you load a file or send a query to a service that returns results.

A data source can contain any of the following:

- **Match columns** - These contain values that are found in the match column of another source, which relates this source to the other, for example, Customer ID or Product ID.
- **Attribute columns** - These contain text, dates, or numbers that are required individually and aren’t aggregated, for example, Year, Category Country, Type, or Name.
- **Measure columns** - These contain values that should be aggregated, for example, Revenue or Miles driven.

You can analyze a data source on its own, or you can analyze two or more data sources together, depending on what the data source contains. If you use multiple sources together, then at least one match column must exist in each source. The requirements for matching are:

- The sources contain common values, for example, Customer ID or Product ID.
- The match must be of the same data type, for example, number with number, date with date, or text with text.

Manage Connections to Data Sources

Manage connections to your data sources.

Topics:

- Create a Connection to a Data Source
- Edit a Data Source Connection
- Delete a Data Source Connection
- Share a Data Source Connection
- Database Connection Options
- Database Connection Limits

Create a Connection to a Data Source

You can create a connection to enable you to analyze data in that data source.

1. On the Home page, click **Create**, then click **Connection**.
2. In the Create Connection dialog, click the icon for the connection type that you want to create a connection for (for example, **Oracle Database**).
3. Enter the required connection information, such as host, port, username, password, and service name.
4. Optional: When you connect to some database types, you might have to specify **Authentication Options**.
   - **Always use these credentials** - Oracle Analytics always uses the login name and password you provide for the connection. Users aren’t prompted to log in.
   - **Require users to enter their own credentials** - Oracle Analytics prompts users to enter their own user name and password for the data source. Users can only access the data for which they have the permissions, privileges, and role assignments.
   - (Displayed if Oracle Analytics supports impersonation for this database type) **Use the active user’s credentials** - Oracle Analytics doesn't prompt users to sign in to access the data. The same credentials they used to sign in to Oracle Analytics are also used to access this data source.
5. Save the details.

   You can now begin creating workbooks or datasets using this connection.

Edit a Data Source Connection

You can update a data source's connection details.

1. On the Home page click the Navigator, select **Data**, and then click **Connections**.

2. Hover over the connection that you want to edit. To the right of the highlighted connection, click **Actions menu**, and select **Inspect**.
3. In the Inspect dialog, edit the connection details.
4. Click **Save**.

If you're editing an SSL connection to an Oracle Database and you need to use a new `cwallet.sso` file, in the **Client Wallet** field, click **Select** to browse for the `cwallet.sso` file. Ask your administrator for the location of the `cwallet.sso` file.

You must provide a unique **Connection Name**. If a connection with the same name already exists in your system, an error message is displayed. You can't see or edit the current password for your connection. If you need to change it, you must create a connection that uses the same password.

Delete a Data Source Connection

You can remove a data source connection from Oracle Analytics Cloud. For example, you must delete a database connection and create a new connection when the database's password has changed.

If the connection contains any datasets, then you must delete the datasets before you can delete the connection.

1. Go to the Data page and select **Connections**.
2. Hover over the connection that you want to delete. To the right of the highlighted connection, click **Actions menu**, and select **Delete**.
3. Click **Yes**.

Database Connection Options

When you specify connection details using the Create Connection dialog or Inspect dialog, some database types have extra options.

General Options

- When you create connections to Oracle Databases, you can connect in two ways using the **Connection Type** option:
  - **Basic** - Specify the **Host**, **Port**, and **Service Name** of the database.
  - **Advanced** - In the **Connection String** field specify the Single Client Access Name (SCAN) ID of databases running in a RAC cluster. For example:
    ```
    sales.example.com =(DESCRIPTION= (ADDRESS_LIST= (LOAD_BALANCE=on)
    (FAILOVER=ON) (ADDRESS=(PROTOCOL=tcp)(HOST=123.45.67.111)(PORT=1521))
    (ADDRESS=(PROTOCOL=tcp)(HOST=123.45.67.222)(PORT=1521))
    (ADDRESS=(PROTOCOL=tcp)(HOST=123.45.67.333)(PORT=1521)))
    (CONNECT_DATA=(SERVICE_NAME= salesservice.example.com)))
    ```
- **Enable Bulk Replication** - If you're loading a dataset for a workbook, then this option should be turned off and you can ignore it. This option is reserved for data analysts and advanced users for replicating data from one database to another database.

Authentication Options

- **Always use these credentials** - Oracle Analytics always uses the login name and password you provide for the connection. Users aren't prompted to log in.

- **Require users to enter their own credentials** - Oracle Analytics prompts users to enter their own user name and password for the data source. Users can only access the data for which they have the permissions, privileges, and role assignments.
- (Displayed if Oracle Analytics supports impersonation for this database type) **Use the active user's credentials** - Oracle Analytics doesn't prompt users to sign in to access the data. The same credentials they used to sign in to Oracle Analytics are also used to access this data source.

Share a Data Source Connection

You can assign access permissions to the data source connections that you create or administer.

1. On the Home page click the **Navigator**. Click **Data**, and then click **Connections**.
2. Hover over the connection that you want to share. Select the connection, click the **Actions menu**, and then select **Inspect**.
3. Click **Access**, and use the tabs to grant access:
    - **All** - Share the connection with individual users or roles.
    - **Users** - Share the connection with individual users.
    - **Roles** - Share the connection with application roles (for example, BI Consumer), so that all users with those roles can use the connection.

    When users next log in, they can use connections that you've shared to visualize data from this database.

Database Connection Limits

Understand your database connection requirements before your create the database connection.

The maximum number of database tables displayed in Oracle Analytics is 10,000. If you need additional tables, Oracle recommends that you ask your database administrator to create a database user with access to the specific objects you want to analyze and specify that user's credentials when you create the database connection.

Create a Connection for an Oracle Database

You can create a connection to a database and use the connection to access data, build a dataset, and visualize data.

▶ Video

You can't use remote connections to save a dataset from a Data Flow.

1. On the Home page, click **Create**, then click **Connection**.
2. In Create Connection dialog, click the connection type, for example, **Oracle Database**.
3. Enter a unique name for the new connection, and database connection details.

- If you’re not creating an SSL connection, specify the connection information for the database, such as the hostname, port, credentials, service name and so on.
- If you’re creating an SSL connection, in the **Client Wallet** field, click **Select** to browse for the `cwallet.sso` file. Ask your administrator for the location of the `cwallet.sso` file.

4. Use the **Connection Type** option to specify how you'd like to connect.
   - When you create connections to Oracle Databases, you can connect in two ways using the **Connection Type** option:
     - **Basic** - Specify the **Host**, **Port**, and **Service Name** of the database.
     - **Advanced** - In the **Connection String** field specify the Single Client Access Name (SCAN) ID of databases running in a RAC cluster. For example:
       ```
       sales.example.com =(DESCRIPTION= (ADDRESS_LIST= (LOAD_BALANCE=on)
       (FAILOVER=ON) (ADDRESS=(PROTOCOL=tcp)(HOST=123.45.67.111)(PORT=1521))
       (ADDRESS=(PROTOCOL=tcp)(HOST=123.45.67.222)(PORT=1521))
       (ADDRESS=(PROTOCOL=tcp)(HOST=123.45.67.333)(PORT=1521)))
       (CONNECT_DATA=(SERVICE_NAME= salesservice.example.com)))
       ```
   - **Enable Bulk Replication** - If you’re loading a dataset for a workbook, then this option should be turned off and you can ignore it. This option is reserved for data analysts and advanced users for replicating data from one database to another database.
5. Click **Save**.

Connect to Oracle Applications

You can create connections to Oracle Applications to help you visualize, explore, and understand the data in your Oracle Fusion Applications with Oracle Transactional Business Intelligence and Oracle BI EE subject areas and analyses.

Topics:

- Create an Oracle Applications Connection
- Edit an Oracle Applications Connection
- Delete an Oracle Applications Connection

Create an Oracle Applications Connection

You can create a connection to Oracle Applications and use the connection to access your data. For example, you can connect to Oracle Fusion Applications with Oracle Transactional Business Intelligence or Oracle BI Enterprise Edition.

1. On the Data page or Home page, click **Create**, then click **Connection**.
2. Click the **Oracle Applications** icon.
3. Enter the connection details.
4. For **URL**, enter the URL for Oracle Fusion Applications with Oracle Transactional Business Intelligence or Oracle BI EE.
5. Under **Authentication**, specify how you'd like to authenticate the connection:

- **Always use these credentials** - Oracle Analytics always uses the login name and password you provide for the connection. Users aren’t prompted to log in.
- **Require users to enter their own credentials** - Oracle Analytics prompts users to enter their own user name and password for the data source. Users can only access the data for which they have the permissions, privileges, and role assignments.
- (Displayed if Oracle Analytics supports impersonation for this database type) **Use the active user’s credentials** - Oracle Analytics doesn't prompt users to sign in to access the data. The same credentials they used to sign in to Oracle Analytics are also used to access this data source.

6. Save the details.

   You can now create datasets from the connection.

The connection is visible only to you (the creator), but you can create and share datasets for it.

Edit an Oracle Applications Connection

You can edit an Oracle Applications connection. For example, you must edit a connection if your system administrator changed the Oracle Applications login credentials.

1. In the Data page, click **Connections**.
2. In the Edit Connection dialog, edit the connection details. Note that you can’t see or edit the password that you entered when you created the connection. If you need to change the connection’s password, then you must create a new connection.
3. Click **Save**.

Delete an Oracle Applications Connection

You can delete an Oracle Applications connection.

1. Delete any datasets that use the connection you need to delete.
2. In the Data page, click **Connections**.
3. Click **Yes**.

Create a Connection to Oracle Essbase

You can create a connection to Oracle Essbase 11g and use the connection to access source data.

1. On the Home page, click **Create**, and then click **Connection**.
2. Click **Oracle Essbase**.
3. Enter the connection details.
4. For **DSN** (data source name), enter the agent URL for your data source.

   If you want to connect to an Oracle Essbase 11g database, enter the hostname and agent port number on which Oracle Essbase is running. Use the format: `hostname:port`

For example: `essbase.example.com:1423`

The default port is 1423.

Your Essbase administrator must open agent port 1423 and server ports in the range 30000-34000 to allow the connection.

5. For **Username** and **Password**, enter user credentials with access to the Essbase data source.
6. Under **Authentication**, specify how you'd like to authenticate the connection:
   - **Always use these credentials** - Oracle Analytics always uses the login name and password you provide for the connection. Users aren’t prompted to log in.
   - **Require users to enter their own credentials** - Oracle Analytics prompts users to enter their own user name and password for the data source. Users can only access the data for which they have the permissions, privileges, and role assignments.
   - (Displayed if Oracle Analytics supports impersonation for this database type) **Use the active user’s credentials** - Oracle Analytics doesn't prompt users to sign in to access the data. The same credentials they used to sign in to Oracle Analytics are also used to access this data source.
7. Save the details.

   You can now create datasets from the connection.

Connect to Google BigQuery

You can create a connection to a Google BigQuery database and use the connection to visualize data.

1. In Google BigQuery, create a Service Account.
   a. Add a role (for example, BigQuery User) with the permission `bigquery.jobs.create` to the Service Account.
   b. Add Users to the role.
   c. Add a JSON key.
2. On the Oracle Analytics Home page, click **Create**, and then click **Connection**.
3. Click **BigQuery**.
4. Enter the connection details.
   - In **Workbook**, specify the workbook name exactly as defined in Google BigQuery.
   - In **Service Account Private Key**, upload the Service Account Private Key (JSON format).
5. Save the details.

   You can now create analytics dataset workbooks based on your Google BigQuery data. When you create a dataset, you navigate to BigQuery tables and select the data facts and measures to analyze. Alternatively, you can use a SQL Query to get the data directly.

Connect to Dropbox

You can create a connection to Dropbox and use the connection to access data.

Before you start, set up a Dropbox application. See the Dropbox documentation.

1. Ask your Oracle Analytics administrator to allow connections to Dropbox.

   Your Oracle Analytics administrator must register the following domains as safe:

   `api.dropboxapi.com`

   `*.dropbox.com`

   See Register Safe Domains.

2. On the Oracle Analytics Home page, click **Create**, and then click **Connection**.
3. Click **Dropbox**.
4. Configure the connection details.
   a. Enter a **Connection Name** and **Description**.
   b. Open and sign into the Dropbox application. Copy and paste the **Redirect URL** into the Dropbox **OAuth 2 Redirect URIs** field, and then click **Add**.
   c. In Dropbox, locate the **App Key** field, copy the key value and paste it in the Oracle Analytics **Client ID** field.
   d. In Dropbox, locate the **App Secret** field, and then click **Show**.
   e. In Oracle Analytics, paste the App Secret value in the **Client Secret** field.
   f. Click **Authorize**.
   g. In the Dropbox prompt to authorize the connection, click **Allow**.

      The Create Connection dialog refreshes and displays the name of the Dropbox account and associated email account.
5. Save the connection.

Connect to Google Drive or Google Analytics

You can create a connection to Google Drive or Google Analytics and use the connection to access data.

Before you start, set up a Google Drive or Google Analytics application. See the Google documentation.

The latest Google Analytics connector supports column-listing. If you can’t access column lists using a Google Analytics connection, create another Google Analytics connection and use that one to get the column-listing capability.

1. Ask your Oracle Analytics administrator to allow connections to Google.

   Your Oracle Analytics administrator must register the following domains as safe:

   `api.google.com`

   `*.google.com`

2. On the Oracle Analytics Home page, click **Create**, and then click **Connection**.
3. Click **Google Drive** or **Google Analytics**.
4. Configure the connection details.
   a. Enter a **Connection Name** and **Description**.
   b. Copy the URL displayed in the **Redirect URL** field.

   c. Open and sign into the Google application. Copy and paste the **Redirect URL** into the Google **Authorized redirect URIs** field and then click **Add**.
   d. In Google, locate the **Client ID** field, copy the key value and paste it into the Oracle Analytics **Client ID** field.
   e. In Google, locate the **Client Secret** field, and then click **Show**.
   f. In Oracle Analytics, enter the Google Client Secret value into the **Client Secret** field.
   g. In **View Name**, specify the name of the default reporting view in Google Analytics that you'd like to analyze.
   h. Click **Authorize**.
   i. In the Google prompt to authorize the connection, click **Allow**.

      The Create Connection dialog refreshes and displays the name of the Google account, and its associated email account.
5. Save the connection.

Create JDBC Connections

You can create JDBC connections to databases and use those connections to access data sources, for example, to connect to databases that aren’t listed with the default connection types.

You can use drivers in a JDBC JAR file to connect to databases.

Check with your administrator that the JDBC driver version must match the database version, even when using an Oracle database. A version mismatch can result in data issues.

For the connection criteria:

- Avoid using instance-specific connection names such as host names, because the same connection can be configured against different databases in different environments, for example, development and production.
- Check the driver documentation and the JAR file for specifying the URL of your JDBC data sources.
- Find the driver class name in the JAR file or from wherever you downloaded the JAR file.
- For the **URL** value, specify the JDBC URL, for example, `jdbc:oracle:thin:@host:port/database`.
- For the **Driver Class Name** value, use the name specified in the JDBC driver download pack that you're using, for example, `oracle.jdbc.driver.OracleDriver`.

1. Copy the required JDBC driver’s JAR file into the `Oracle Analytics Desktop\lib` directory.

   For example, `C:\Program Files\Oracle Analytics Desktop\lib`.
2. On the Data or Home page, click **Create**, then click **Connection**.
3. In the Create Connection dialog, locate and click the **JDBC** icon.
4. Enter the connection criteria.
5. Click **Save**.

   You can now create datasets from the connection.

If you import a workbook containing a JDBC connection into an Oracle Analytics installation where the JDBC driver isn’t installed, the import still works. However, the connection doesn’t work when you try to run the workbook or Data Flow. You must recreate the JDBC connection and JDBC driver to a suitable data source.

Create Generic ODBC Connections

You can create generic ODBC connections to databases and use the connections to access data sources. For example, to connect to databases and database versions that aren’t listed with the default connection types.

You can only use generic ODBC connections to connect on Windows systems.

1. Confirm that the appropriate database driver is installed on your computer.

   You must have the required database driver installed on your computer to create an ODBC Data Source Name (DSN). If you need to install a database driver, use installation instructions provided by the organization that supplies the database driver.

2. Create the new ODBC data source in Windows.

   a. In Windows, locate and open the ODBC Data Source Administrator dialog.

   b. Click the **System DSN** tab, and then click **Add** to display the Create New Data Source dialog.

   c. Select the driver appropriate for your data source, and then click **Finish**.

   d. The remaining configuration steps are specific to the data source you want to configure.

      Refer to the documentation for your data source.

3. Create the generic ODBC data source.

   a. On the Data or Home page, click **Create**, then click **Connection**.

   b. In the Create Connection dialog, locate and click the **ODBC** icon.

   c. Enter the connection criteria. Note the following:

      - **Name** - Any name that uniquely identifies the connection.
      - **DSN** - The name of the system DSN that you set up on your computer.

   d. Click **Save**.

      You can now create datasets from the connection.

If you import a workbook containing an ODBC connection into an Oracle Analytics Desktop installation where the ODBC DSN doesn’t exist, and the ODBC driver isn’t installed, the import still works. However, the connection doesn’t work when you try to run the workbook or Data Flow. You must recreate the ODBC connection, and recreate the ODBC DSN, and ODBC driver to a suitable data source.

Connect to Oracle Autonomous Data Warehouse

You can create a connection to Oracle Autonomous Data Warehouse and use the connection to access data.

Tutorial

1. Before you create connections to Oracle Autonomous Data Warehouse, you must obtain the client credentials zip file containing the trusted certificates that enable Oracle Analytics to connect to Oracle Autonomous Data Warehouse.

   See Download Client Credentials (Wallets) in *Using Oracle Autonomous Database on Shared Exadata Infrastructure*.

   The credentials wallet file secures communication between Oracle Analytics and Oracle Autonomous Data Warehouse. The wallet file (for example, `wallet_ADWC1.zip`) that you upload must contain SSL certificates, to enable SSL on your Oracle Database Cloud connections.

2. To create a connection to Oracle Autonomous Data Warehouse:

   a. On the Home page, click **Create** then click **Connection**.

   b. Click **Oracle Autonomous Data Warehouse** to display the fields for the connection.

   c. Enter the connection details.

   d. For **Client Credentials** field, click **Select** to browse for the Client Credentials wallet file (for example, `wallet_ADWC1.zip`).

      The **Client Credentials** field displays the `cwallet.sso` file.

   e. Enter the **Username** and **Password**, and select a **Service Name** from the list.

   f. Save the details.

      You can now create datasets from the connection.

Connect to Oracle Autonomous Transaction Processing

You can create a connection to Oracle Autonomous Transaction Processing and use the connection to access data.

Before you create connections to Oracle Autonomous Transaction Processing, you must obtain the client credentials zip file containing the trusted certificates that enable Oracle Analytics to connect to Oracle Autonomous Transaction Processing.

See Download Client Credentials (Wallets) in *Using Oracle Autonomous Database on Shared Exadata Infrastructure*.

The credentials wallet file secures communication between Oracle Analytics and Oracle Autonomous Transaction Processing. The wallet file (for example, `wallet_SALESATP.zip`) that you upload must contain SSL certificates, to enable SSL on your Oracle Database Cloud connections.

1. On the Home page, click **Create** then click **Connection**.
2. Click **Oracle Autonomous Transaction Processing**.
3. Enter the connection details.
4. For **Client Credentials**, click **Select** to browse for the Client Credentials wallet file (for example, `wallet_SALESATP.zip`).
5. Enter the **Username**, and **Password**, and select a **Service Name** from the list.
6. Click **Save**.

Connect to Oracle Talent Acquisition Cloud

You can create a connection to Oracle Talent Acquisition Cloud and use the connection to access data.

1. On the Home page, click **Create** and then click **Connection**.
2. Click **Oracle Talent Acquisition** to display the fields for the connection.
3. Enter the connection details.
4. For **Host**, enter the URL for the Oracle Talent Acquisition data source.

   For example, if the Oracle Talent Acquisition URL is `https://example.taleo.net`, then the connection URL that you must enter is `https://example.taleo.net/smartorg/Bics.jss`.
5. Select an **Authentication** option.
   - Select **Always use these credentials**, so that the login name and password you provide for the connection are always used and users aren’t prompted to log in.
   - Select **Require users to enter their own credentials** when you want to prompt users to enter their user name and password to use the data from the Oracle Talent Acquisition Cloud data source. Users are required to log in see only the data that they have the permissions, privileges, and role assignments to see.
6. Click **Save**.

Connect to Snowflake Data Warehouse

You can create a connection to Snowflake Data Warehouse and use the connection to access data.

See format guidelines, `https://docs.snowflake.net/manuals/user-guide/connecting.html`.

1. On the Home page, click **Create**, and then click **Connection**.
2. Click **Snowflake Data Warehouse**.
3. Enter a Connection Name.
4. In **Hostname**, enter the host account name using the following format.

   `<account>.snowflakecomputing.com`

   Where `account` is the Snowflake account name that you want to use to access the data, for example: `exampleaccountname.snowflakecomputing.com`.
5. For **Username** and **Password**, enter user credentials with access to the Snowflake data source.
6. For **Database Name**, enter the name of the database containing the schema tables and columns that you want to connect to.
7. For **Warehouse**, enter the name of the warehouse containing the database, schema tables and columns that you want to connect to. For example, `Example-WH`.

8. Click **Save**.

Connect to NetSuite

Connect to a NetSuite (NetSuite2.com) data source to visualize ERP and CRM data.

When you analyze data from a NetSuite2.com data source, you create a connection to store connection details. To obtain the connection details for your NetSuite application, go to the NetSuite Portal home page, and navigate to **Settings** then **Set up SuiteAnalytics Connect**.

3 Connect to Your Data Using Datasets

This topic describes how to use a data source connection to create a dataset.

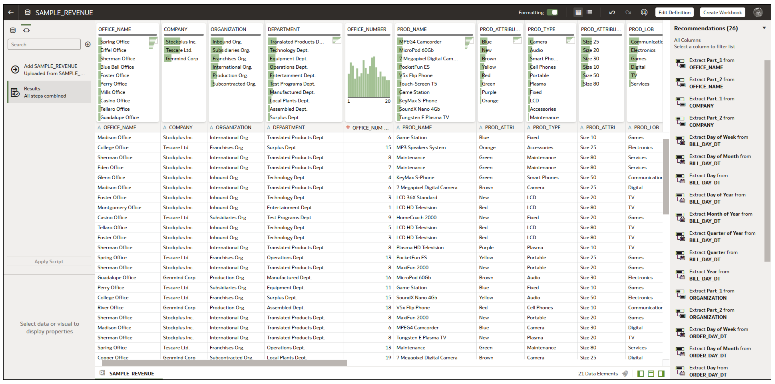
Topics

- Create a Dataset from an Oracle Application's Analysis
- Create a Dataset from an Essbase Connection

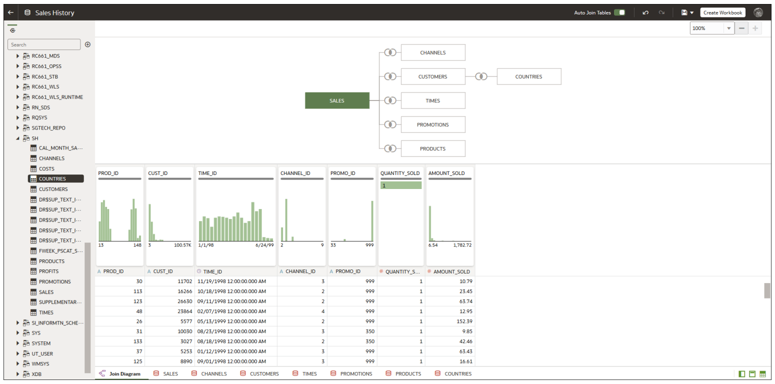
About Opening Datasets

You can perform self-service data modeling to create datasets with multiple tables. The type of data source the dataset uses determines how you navigate the user interface when you create a new dataset, and when you open an existing dataset by itself or from a workbook.

When you create or open a dataset that is based on Google Analytics, Dropbox, Essbase, a spreadsheet, or a file, the dataset is displayed in the Transform editor.



When you create or open a dataset with multiple tables, the dataset is displayed in the Dataset editor.



When working in a workbook, use the **Data** tab to view and open the workbook's datasets. The dataset determines which editor Oracle Analytics opens.

- If the dataset is based on Google Analytics, Dropbox, Essbase, a spreadsheet, or a file, then the Transform editor is opened in a new window.
- If the dataset contains multiple tables, then the Dataset editor is opened in a new window.

What Are Datasets?

Datasets are self-service data models that you build specifically for your data visualization and analysis requirements.

A dataset can be based on one table, spreadsheet, or file. Or a dataset can be a self-service data model that contains multiple tables with relationships defined between the tables.

You can create datasets from data from files, SaaS applications, Oracle Analytics reports, and many relational and big data sources. See Data Sources Available for Use in Datasets Containing Multiple Tables.

A dataset contains data source connection information, tables, the columns you specify, and the data enrichments and transformations that you apply.

You can use a dataset in multiple workbooks and in data flows. If you change a dataset, changes affect all workbooks and data flows that use the dataset.

When you create and build a dataset, you can:

- Choose between many types of connections or spreadsheets.
- Create datasets based on data from multiple tables in a database connection, Oracle data source, or a local subject area. And create datasets based on data from tables in different connections and subject areas. For example, you can build a dataset that includes tables from an Autonomous Data Warehouse connection, tables from a Spark connection, and tables from a local subject area.
- Specify joins between tables.
- Transform and enrich the columns in the dataset.

Typical Workflow to Create a Dataset from a Connection

Here are the common tasks for connecting to a data source and creating a dataset.

Task	Description	More Information
Understand the Dataset editor	Use the Dataset editor to create datasets from connections that support datasets with multiple tables.	About the Dataset Editor Data Sources Available for Use in Datasets Containing Multiple Tables
Find or create a connection	Review a list of data source connections that you can use to create a dataset. Or create the data source connection that you need.	View Available Connections
Create the dataset and choose a connection	Start a new dataset and connect it to your data source.	Create a Dataset From a Connection
Add more connections	Add other connections to access more tables to add to the dataset. Some connections such as Google Analytics, Dropbox, and Essbase aren't available when you create a dataset with multiple tables. This option is available for datasets with multiple tables.	Add Multiple Connections to a Dataset

Task	Description	More Information
Add tables to the dataset	Drag and drop tables to the dataset's Join Diagram. This option is available for datasets with multiple tables.	Create a Dataset From a Connection
Inspect or add joins	Review and update the table joins that were automatically added. Create additional joins. This option is available for datasets with multiple tables.	Add Joins Between Dataset Tables
Prepare data	Perform data transformations and enrichments.	Enrich and Transform Your Data
Edit the table's definition	Select columns to include or remove from the data source. Specify how you want to get data updates from the source.	Remove or Restore a Dataset Table's Columns Apply a Filter to a Dataset Table Specify Whether a Dataset Table Is Cached or Live

Typical Workflow to Create a Dataset from a File

Here are the common tasks for creating a dataset from a file.

Task	Description	More Information
Upload a file	Browse for and select a spreadsheet to use as a dataset.	Create a Dataset from a File Uploaded From Your Computer Create a Dataset From a File Uploaded from Dropbox, Google Analytics, or Google Drive
Prepare data	Perform data transformations and enrichments.	Enrich and Transform Your Data

View Available Connections

A dataset uses one or more connections to data sources to access and supply data for analysis and visualization.

> **Note:**
>
> You can create datasets with multiple tables from most data sources. Exceptions include files, Google Analytics, Dropbox, and Essbase.

If you need to create a connection, see Create a Connection to a Data Source.

1. In the Home Page, click **Navigator** and then click **Data**.
2. Click the **Connections** tab to view your connections list.

Data Sources Available for Use in Datasets Containing Multiple Tables

Learn which data sources you can use to create datasets with multiple tables.

Oracle Data Sources

You can use one of more of these Oracle data source connections to create a dataset with multiple tables.

- Oracle Applications
- Oracle Autonomous Data Warehouse
- Oracle Autonomous Transaction Processing
- Oracle Database
- Oracle NetSuite
- Oracle Service Cloud

Non-Oracle Data Sources

You can use one or more of these data source connections to create a dataset containing multiple tables.

- Amazon EMR
- Amazon Redshift
- Apache Hive
- DB2
- GreenPlum
- HortonWorks Hive
- IBM BigInsights Hive
- Impala (Cloudera)
- Informix
- MapR Hive
- Microsoft Azure SQL Database
- MongoDB
- Pivotal HD Hive
- PostgreSQL
- Salesforce
- Snowflake
- Spark
- SQL Server
- Sybase ASE
- Sybase IQ

- MySQL

Data Sources Not Available for Datasets Containing Multiple Tables

These data source connections aren't available to create or include in a dataset containing multiple tables.

- Dropbox
- Essbase
- Files (XLSX, XLS, CSV, and TXT)
- Google Analytics

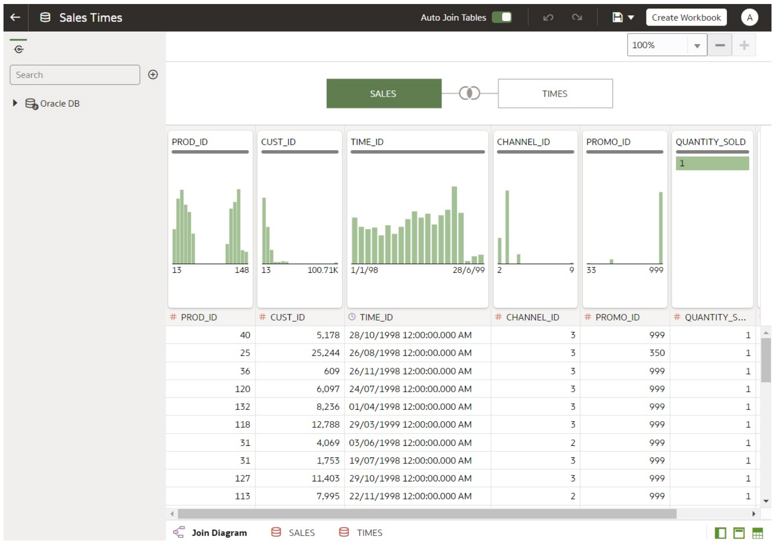
About the Dataset Editor

Use the Dataset editor to create or edit a dataset that includes tables, joins, and enriched data.

For information about how to create a dataset, see Typical Workflow to Create a Dataset from a Connection.

When you create a dataset from a connection that supports datasets with multiple tables, Oracle Analytics displays the Dataset editor. See Data Sources Available for Use in Datasets Containing Multiple Tables.

You can't create datasets with multiple tables from Google Analytics, Dropbox, and Essbase connections. Use the Transform editor with spreadsheets, CSV and text files, and the connections that don't support datasets with multiple tables.



# PROD_ID	# CUST_ID	TIME_ID	# CHANNEL_ID	# PROMO_ID	# QUANTITY_S...
40	5,178	28/10/1998 12:00:00.000 AM	3	999	1
25	25,244	26/08/1998 12:00:00.000 AM	3	350	1
36	609	26/11/1998 12:00:00.000 AM	3	999	1
120	6,097	24/07/1998 12:00:00.000 AM	3	999	1
132	8,236	01/04/1998 12:00:00.000 AM	3	999	1
118	12,788	29/03/1999 12:00:00.000 AM	3	999	1
31	4,069	03/06/1998 12:00:00.000 AM	2	999	1
31	1,753	19/07/1998 12:00:00.000 AM	3	999	1
127	11,403	29/10/1998 12:00:00.000 AM	3	999	1
113	7,995	22/11/1998 12:00:00.000 AM	2	999	1

Each Dataset editor feature and what you can do with it is explained here:

- **Connections Pane**

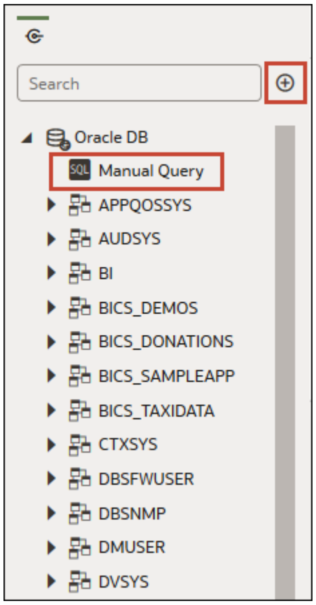
  This pane displays the data source connection's schemas and tables. To add a table from the Connections Pane to the Join Diagram, drag and drop it from the list to the diagram or double-click it. You can add more than one table.

Click **Add** to open the Add Connection dialog where you can select or create another connection to add to the dataset.

Use **Search** to find a schema or table displayed in the Connections pane.

To write a SQL statement to create a table, drag and drop the Manual Query item to the diagram. Then go to the diagram and double-click the Manual Query table to access the SQL editor.

> **Note:**
>
> Use Manual Query when a JDBC connection doesn't provide a schema and tables listing.

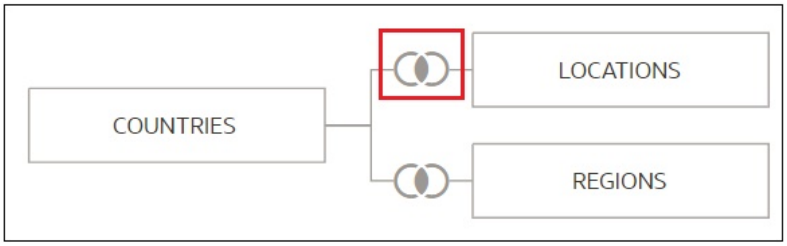


- **Join Diagram**

  The Join Diagram displays all of the tables and joins in the dataset. Drag and drop tables to reorder them in the diagram.

  When you select a table, the Preview Data area displays a sample of the table's data. If you double-click a table, then the Transform editor is displayed where you'll transform or enrich the table's data.

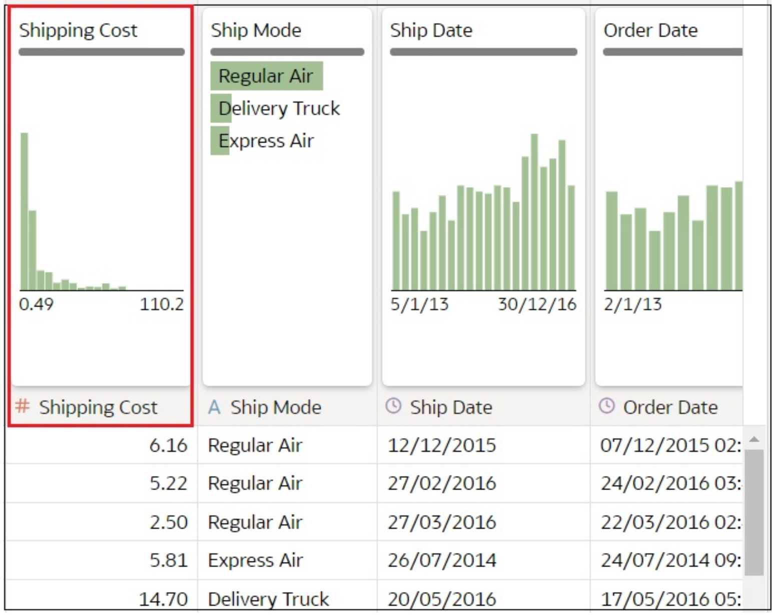
  The diagram's join links and icons indicate which tables are joined and the type of join used. A join is created when you drag and drop one table onto another. If you click on a join, the Join editor is displayed where you can view, update, or fix it.



- **Data Preview**

  The data preview area displays a data sample for the table selected in the **Join Diagram**. From the data preview area you can click a column's type attribute to change its type. For example, change the CALENDAR_YEAR column from a measure to an attribute.

  Be default, a quality insight tile is displayed for each column. Quality Insights uses a representative sample of the column's data to determine information about the column's data quality. See What are Quality Insights?



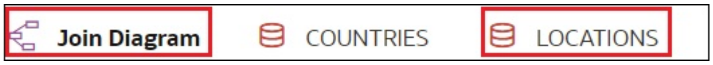
# Shipping Cost	A Ship Mode	Ship Date	Order Date
6.16	Regular Air	12/12/2015	07/12/2015 02:
5.22	Regular Air	27/02/2016	24/02/2016 03:
2.50	Regular Air	27/03/2016	22/03/2016 02:
5.81	Express Air	26/07/2014	24/07/2014 09:
14.70	Delivery Truck	20/05/2016	17/05/2016 05:

- **Auto Join Tables Toggle Button**

  The **Auto Join Tables** toggle button determines if joins are automatically created when you drag and drop tables onto the Join Diagram. By default Oracle Analytics creates joins, but in some cases you might want to turn off automatic join creation and manually create table joins.

- **Dataset Table Page Tabs**

  The Dataset editor displays a tab for each table in the dataset. Click a table tab to go to the Transform editor to transform or enrich the table's data. The tabs also display at the bottom of the Transform editor. Use them to navigate to other tables or back to the Join Diagram.



- **Dataset Editor Toggle Buttons**

  These buttons control which parts of the Dataset editor are displayed.

  This button hides or reveals the connections panel:

  This button hides or reveals the quality tiles:

  This button hides or reveals the preview data area:

Create a Dataset From a Connection

When you create a dataset, you can add tables from one or more data source connections, add joins, and enrich data.

Video

Tutorial

Before you create the dataset, you can check to see if the data source connections you need already exist. See View Available Connections.

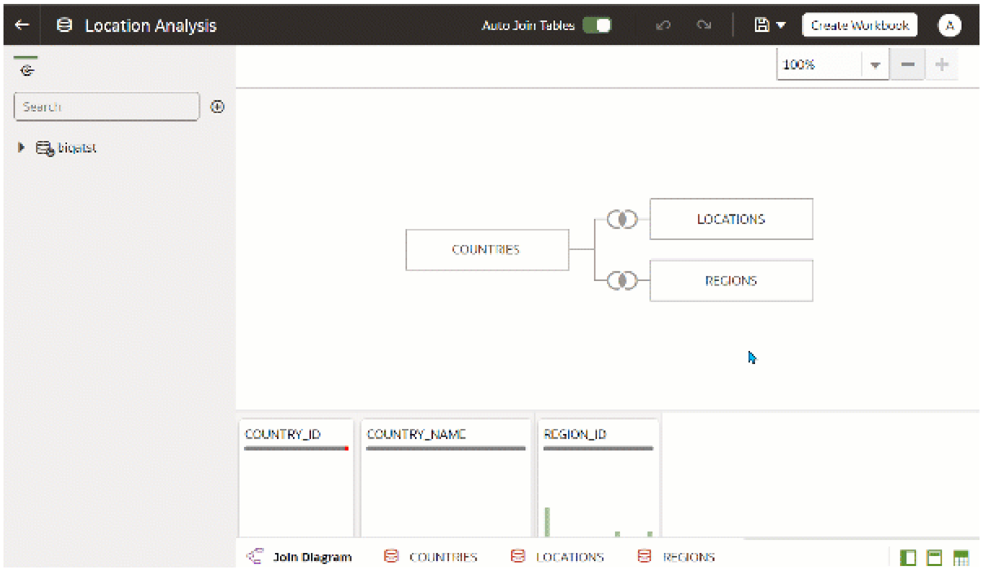
> **Note:**
>
> You can create datasets with multiple tables from most data sources. Exceptions include files, Google Analytics, Dropbox, and Essbase.

1. On the Home page, click **Create** and then click **Dataset**.
2. In the Create Dataset dialog, select a connection.
3. In the Dataset editor, go to the **Connections** pane and browse or search for a schema.
4. Optional: If you want to build the dataset's table joins manually rather than have Oracle Analytics automatically create them, then deselect the **Auto Join Tables** toggle to turn off automatic join creation.

5. Click the schema to view a list of its tables. Drag and drop tables to the **Join Diagram**. If you know the fact table, then add it first before you add dimension tables. The fact table must be the left-most table in the **Join Diagram**.

   By default, the dataset you're building includes any table joins defined in the data source. If no joins exist in the data source then Oracle Analytics identifies and adds joins.

6. Optional: To add data from additional data sources, go to the **Connections** pane, click **Add**, and then click **Add Connection** to select and add another connection to the dataset.
7. Optional: If you added another connection, then from the **Connections** pane, click the new connection's schema to open it, and drag and drop tables to the **Join Diagram**.
8. Optional: When you drag and drop a table to the **Join Diagram** from some data sources such as Oracle Applications subject areas, then no columns are displayed. Go to the **Table Page Tabs**, click the subject area table, and use the **Transform Editor** to specify which columns to include in the table. Click **OK**.
9. In the **Join Diagram**, inspect, modify, add, or delete joins.
   - To inspect or modify a join, click the join to open the Join editor and inspect or modify the join type and join conditions.
   - To add a join, locate the table that you want to join, hover over it to select it, and then click and drag and drop it on to the table that you want to join it to. Open the Join editor to inspect or update the join type and conditions.
   - To delete a join, hover over the join icon and click **Delete Join**.
10. Click **Save**.
11. Enter a name and click **OK**.
12. To prepare a table's data, go to the tab row at the bottom of the Dataset editor and click the table's tab. Alternatively, from the Join Diagram, double-click the table, or right-click the table and select **Open**.



13. Use the Transform editor to perform any data transformations or enrichments. Click **Save Dataset**.
14. To edit a table's definition such as columns to include or exclude and data access, in the Transform editor, click **Edit Definition** and make any changes. Click **OK** and then click **Save Dataset**.

Add Multiple Connections to a Dataset

A dataset can include more than one connection. Adding more connections allows you to access and join all of the tables and data that you need to build your dataset.

You can add more connections to datasets that support multiple tables. Most connections are available for you to add. Exceptions include Google Analytics, Dropbox, and Essbase.

Before you create the dataset, you can check to see if the data source connections you need already exist. See View Available Connections.

If you drag a subject area or analysis to the **Join Diagram**, then no columns are displayed. You must use the **Transform Editor** to specify which columns to include in the subject area's dataset table.

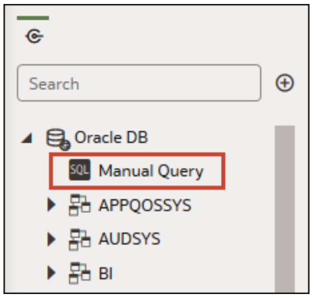
1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.
4. In the Connections pane, click **Add**, and click **Add Connection**.
5. In the Add Connection dialog, locate and click a connection to add it to the Connections pane.
6. In the Connections pane, confirm that the connection you selected was added.
7. Click **Save**.

Use a SQL Statement to Create a Dataset Table

You can write a source specific SQL SELECT statement to query a connection's data and create a table in a dataset. This manual query option is available for most connection types.

Use the manual query option to create tables when you don't want to use a connection's tables in the dataset but instead want to target specific information. Or use the manual query option when the data source connection doesn't provide a schema listing. The SQL statement you use is determined by the data source connection type.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.
4. In the Connections pane, locate and expand the connection that you want to query, and locate the **Manual Query** option.



5. Drag and drop **Manual Query** to the Join Diagram to create a table shell.
6. Double-click the Manual Query table.
7. In the Add Dataset page, rename the table and confirm that **Enter SQL** is selected.
8. In the **Statement** field, enter the SQL statement.
9. Click **OK**.
10. Optional: To modify the SQL statement, go to the **Table Page Tabs** and confirm that the tab for the table you created is selected. Click **Edit Definition** to access the Add Dataset page and modify the SQL statement.

Understand Dataset Table Joins

Joins indicate relationships between a dataset's tables.

By default, when you drag and drop tables to the Join diagram to create a dataset, any joins that are defined in the data source are automatically created between tables in the dataset. Oracle Analytics also automatically creates joins base on the column name matches it finds between the tables. When defining your dataset, be sure to add the most detailed table first. This is usually the primary table for your analysis. For example, the Sales table. Then add the remaining tables like Products, Customers, Channels, and so on that provide context for your analysis. If you're adding tables from a dataset that is dimensionally modeled or a transactional table, then the most detailed table is possibly a fact table with measures.

In some cases you might want to build a dataset's joins manually. For example, if your dataset contains too many matching columns and it's time consuming for you to delete the unnecessary matches. In such cases you can turn off automatic joins for your dataset. See Disable Auto Joins in Datasets.

If you're creating a dataset based on facts and dimensions and if joins already exist in the source tables, then joins are automatically created in the dataset. If you're creating a dataset from multiple connections and schemas, then you manually define the joins between tables. To do this, drag and drop a dimension table onto a fact table and specify the type of join to use and the join columns. You can create a join on more than one column. The columns that you're joining must be attribute columns and have the same data format.

Not all data source types are available for you to add and join to in a dataset that contains multiple tables. For example, files and Essbase aren't available. When you're working in a workbook and include data from different datasets, for example a dataset containing multiple tables from Oracle Autonomous Data Warehouse and a dataset containing data from a spreadsheet, then you'll use blending to match the columns. See Blend Datasets.

> **Note:**
>
> To analyze data based on two star schemas, create a dataset for each star schema, add the datasets to a workbook, and then blend the datasets.

The Join Diagram displays join icons representing the type of joins used between tables and if any joins are incomplete. Join icons are Venn diagram thumbnails that represent the join type.

Icon	Description
⚭	Indicates an inner join. When this icon is selected, it's displayed in blue. An inner join returns rows from the left table that match with the rows from the right table.
⚭	Indicates a left join. When this icon is selected, it's displayed in blue. The left join returns all rows from the left table with the matching rows if available from the right table. If there is no matching row found from the right table, then the left join contains null values for the columns of the right table.
⚭	Indicates a right join. When this icon is selected, it's displayed in blue. The right join returns all rows from the right table with the matching rows if available from the left table. If there is no matching row found from the left table, then the right join contains null values for the columns of the left table.
⚭	Indicates a full outer join. When this icon is selected, it's displayed in blue. The full outer join returns a result set that contains all rows from both left and right tables, with the matching rows from both sides where available. If there is no match, then the missing side has nulls.
- - -⚭- - -	The dashed line indicates an incomplete join. When this icon is selected, it's displayed in blue. When this icon isn't selected, it's displayed in yellow. Click the icon to open the Join editor and assign or fix the join.

Add Joins Between Dataset Tables

Add joins to indicate relationships between a fact table and the dimension tables.

By default, when you drag and drop tables to a dataset containing multiple tables, any joins defined in the data source are automatically created between the dataset's tables. Oracle Analytics also automatically creates joins base on the column name matches it finds between the tables. However Oracle Analytics can't always figure out how to automatically build joins. In such cases you must manually add joins before you can save the dataset.

If your dataset table contains deleted columns, then those columns aren't available to use in joins. If your dataset table contains hidden columns, then those columns are available to use in joins. See Hide or Delete a Column.

For information about using joins, see Understand Dataset Table Joins.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.

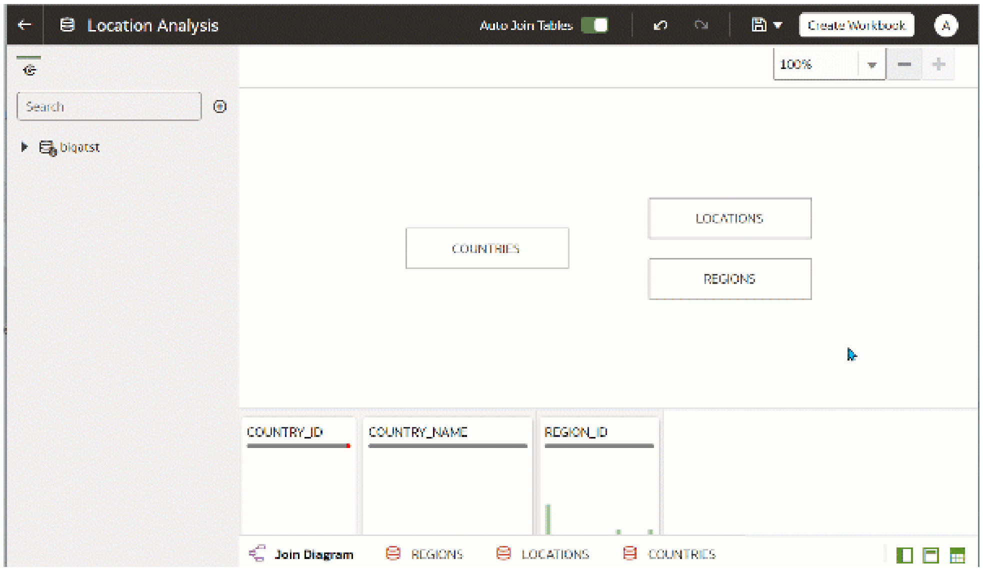
4. In the **Join Diagram** locate the table that you want to join to another table.

   Make sure that the fact table is the left-most table in the **Join Diagram**.

5. Click the table and then drag and drop the selected table onto the target table that you want to join it to.

   The join icon is displayed in blue.

6. Optional: Click the join to open the Join dialog and inspect the join.



Disable Auto Joins in Datasets

You can use the **Auto Join Tables** toggle button to prevent Oracle Analytics from creating joins when you add tables to your dataset.

By default, when you drag and drop tables to a dataset containing multiple tables, any joins defined in the data source are automatically created between the dataset's tables. Oracle Analytics also automatically creates joins based on the column name matches it finds between the tables.

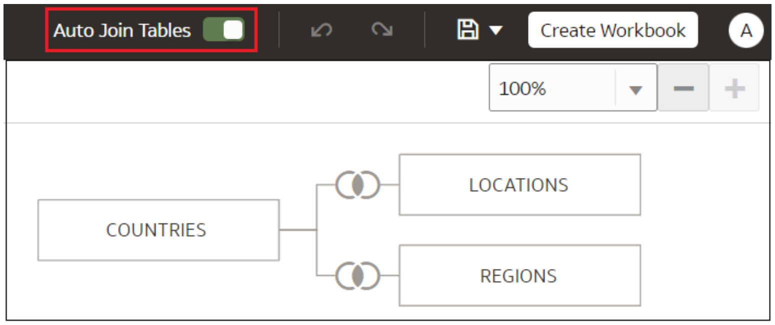
In some cases you might want to build the dataset's joins rather than have Oracle Analytics build joins for you. For example, if your datasets contains too many matching columns and it's time consuming for you to delete the unnecessary matches. In such cases you deselect the **Auto Join Tables** toggle button to turn off automatic joins.

Deselecting **Auto Join Tables** doesn't remove the dataset's existing joins, so you must remove those joins manually.

After you turn off automatic joins and add tables to the dataset, you must manually add joins before you can save the dataset. For information about creating dataset table joins, see Add Joins Between Dataset Tables.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.
4. Click the **Join Diagram** tab.

5. Click **Auto Join Tables** to prevent Oracle Analytics from automatically creating joins for tables you add to the dataset.

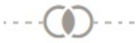


Modify or Fix Joins in a Dataset

You can modify any joins that were automatically added to the dataset and fix any incomplete or broken joins.

> **Note:**
>
> Each table in the dataset must be joined to one or more tables.

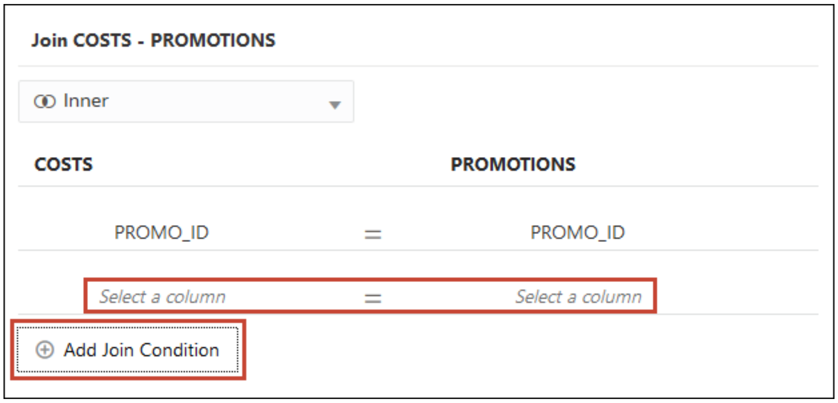
This icon indicates an incomplete join:



For information about using joins, see Understand Dataset Table Joins.

If the columns that you need to create joins aren't listed in the Join dialog, then you can update the columns to make them available for joins. See Add Joins When Table Columns Don't Match.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.
4. In the **Join Diagram** locate the join, hover over it, and click it to open the Join dialog.
5. Optional: Click the join type and select a new type. The join type that you select determines the join icon in the Join Diagram.
6. Optional: To modify a join condition, click a table's join column name and pick a different column to join on.
7. Optional: To add another join condition, click **Add Join Condition** and select a join column for each table.



Join COSTS - PROMOTIONS

Inner

COSTS		PROMOTIONS
PROMO_ID	=	PROMO_ID
Select a column	=	Select a column

8. Click **Save**.

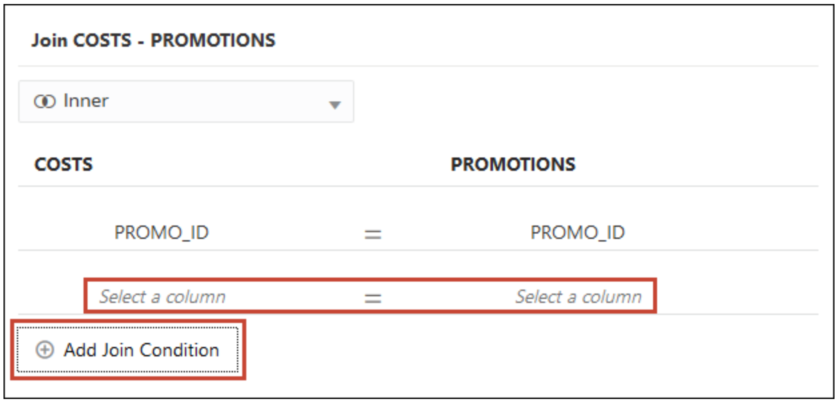
Add Joins When Table Columns Don't Match

If your dataset contains multiple tables and the column data doesn't match between the tables, then you can transform the column's data into the format required to create joins.

Video

Examples of transformations include changing the data type, splitting or concatenating data in the column, or enriching the column data. The list of available transformation menu options for a column depends on the type of data in that column. See Transform Recommendation Reference and Column Menu Options for Quick Data Transformations.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.
4. In the **Join Diagram** locate the table with the column that you want to transform.
5. Right-click the table and select **Open**.
6. In the Transform editor, locate the column that you want to transform and click **Options**.
7. From the menu, select a transformation and complete any required fields.
8. Click **Save Dataset** and then click the **Join Diagram** tab.
9. In the Join Diagram, click and hold the table to select it and then drag and drop the selected table onto the target table that you want to join it to.
10. In the **Join Diagram** locate the join that you just added, hover over it, and click it to open the Join dialog.
11. Optional: Click the join type and select a new type. The join type that you select determines the join icon in the Join Diagram.
12. Optional: To add a join condition, click **Add Join Condition** and select a join column for each table.

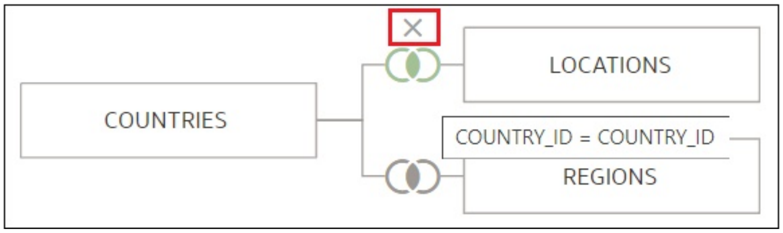

Join COSTS - PROMOTIONS

13. Click **Save**.

Delete Joins from a Dataset

You can delete any table join from a dataset.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.
4. In the **Join Diagram** locate the join, hover over it, and click **X**.



5. Click **Save**.

Include a Dataset Table in Data Source Queries

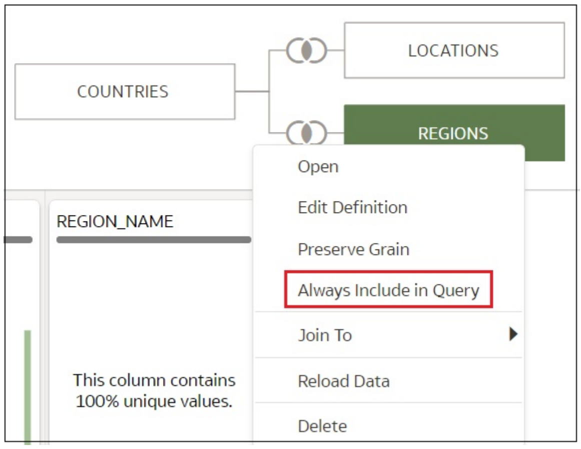
Oracle Analytics generates source-specific optimized queries for each visualization. During this process, a dataset is treated as a data model in that only the tables needed to satisfy a visualization are used in the query.

However, there are scenarios where you may want to use a table in the query even if the table isn't queried in the visualization. In these scenarios, you can set the table to **Always Include in Query**. For example, you may want to see only Sales where a product is associated but the visualization is based on only Customer and Sales. Or you may want to apply a date filter to all visualizations in a workbook.

Oracle Analytics prunes any of the dataset's tables not used in the visualization or not specified to be included in the query.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.

3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.
4. In the **Join Diagram** locate the table, right-click, and select **Always Include in Query**.



5. Click **Save Dataset**.

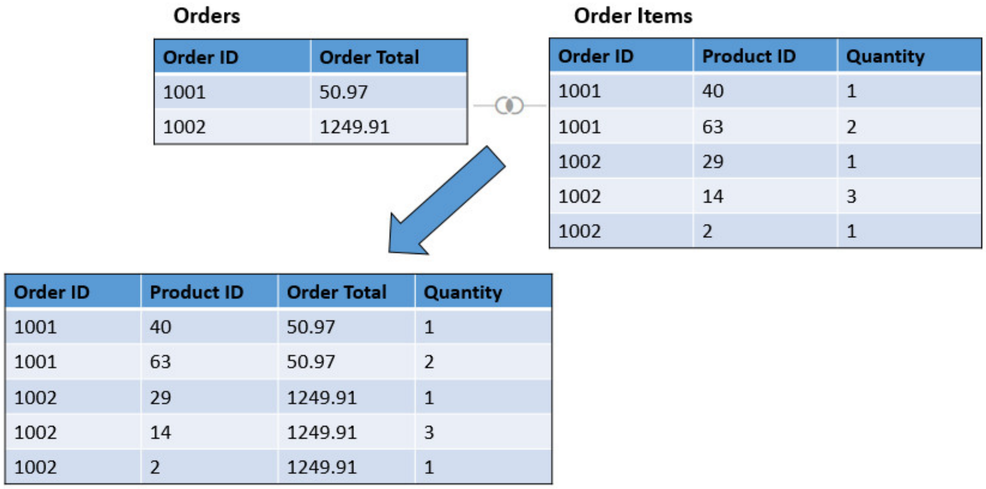
What is Preserve Grain?

By default a dataset's grain is determined by the table with the lowest grain, but you can use the preserve grain setting to specify which table determines the dataset's grain.

See Specify Which Table Determines Grain.

You can create a measure in any table in a dataset. However, this can cause the measure on one side of a one-to-many or many-to-many relationship to be duplicated. In cases like this you can set the table on one side of the cardinality to **Preserve Grain** to keep its level of detail.

For example, if you have an Orders table and an Order Items table, join them on the Order ID column, and query them, then the Order Total is duplicated for each order item. This is because the Order Items table has the lower grain.



Orders

Order ID	Order Total
1001	50.97
1002	1249.91

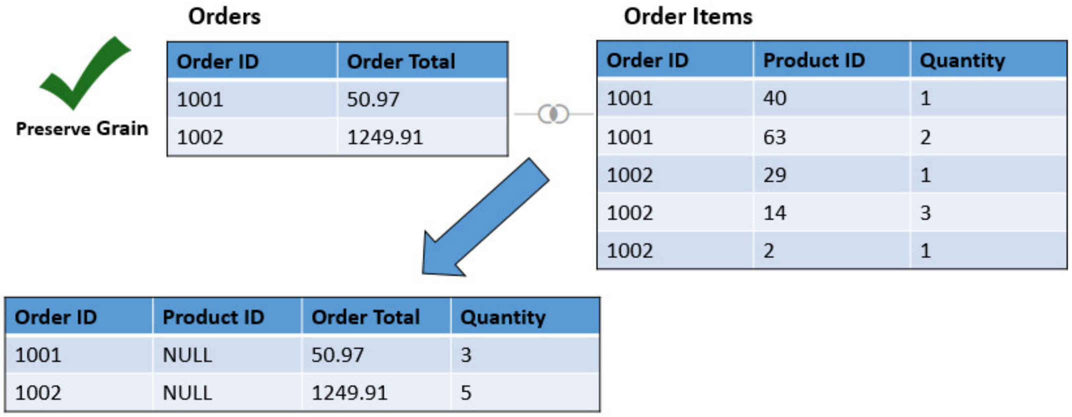
Order Items

Order ID	Product ID	Quantity
1001	40	1
1001	63	2
1002	29	1
1002	14	3
1002	2	1

Order ID	Product ID	Order Total	Quantity
1001	40	50.97	1
1001	63	50.97	2
1002	29	1249.91	1
1002	14	1249.91	3
1002	2	1249.91	1

However, if you want the query results to show data at the Orders table's level, then in the Dataset Editor's Data Diagram, you set the Orders table to **Preserve Grain**.

> **Note:**
>
> In the following example, the Product ID column is NULL because there are multiple values for PRODUCT for each Order. The Values are set to NULL to preserve the grain at the Order level.



Orders

Order ID	Order Total
1001	50.97
1002	1249.91

Order Items

Order ID	Product ID	Quantity
1001	40	1
1001	63	2
1002	29	1
1002	14	3
1002	2	1

Order ID	Product ID	Order Total	Quantity
1001	NULL	50.97	3
1002	NULL	1249.91	5

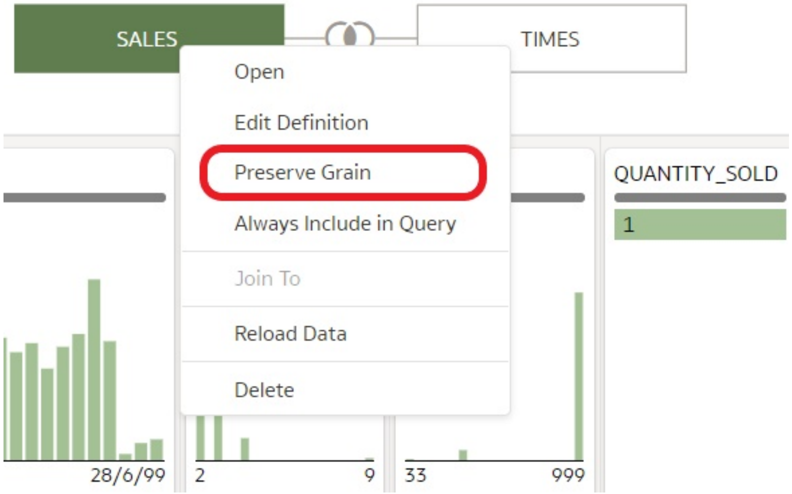
Specify Which Table Determines Grain

You can change which dataset table determines grain.

For an example of why you'd change a table's grain, see What is Preserve Grain?

When you set a table to **Preserve Grain**, a green bar is displayed at the top of table in the Join Diagram. This green bar indicates which table's grain the dataset uses.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.
4. In the **Join Diagram** locate the table, right-click, and select **Preserve Grain**.



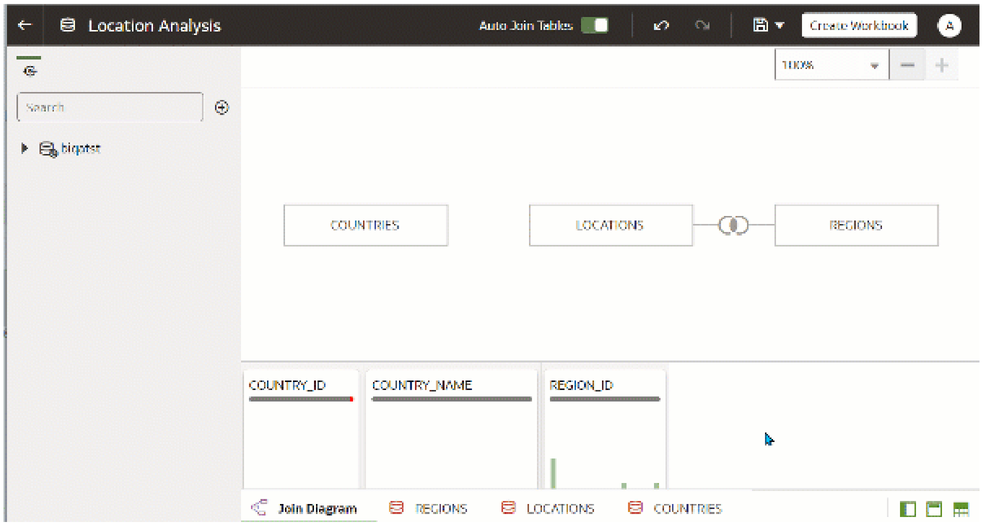
5. Click **Save Dataset**.

Rearrange the Order of a Dataset's Tables

When you include a dataset in a workbook, the Workbook Editor's **Data Panel** displays the dataset's tables as folders and in the same order that you added them to the dataset.

In the Dataset editor's **Table Page Tabs** you can drag and drop tables into an order that makes it easy for you to locate the folders and columns that you'll use most often in visualizations.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.
4. In the **Table Page Tabs** locate the table that you want to reposition.
5. Click and hold, and then drag and drop the table to a different position on the tabs bar.

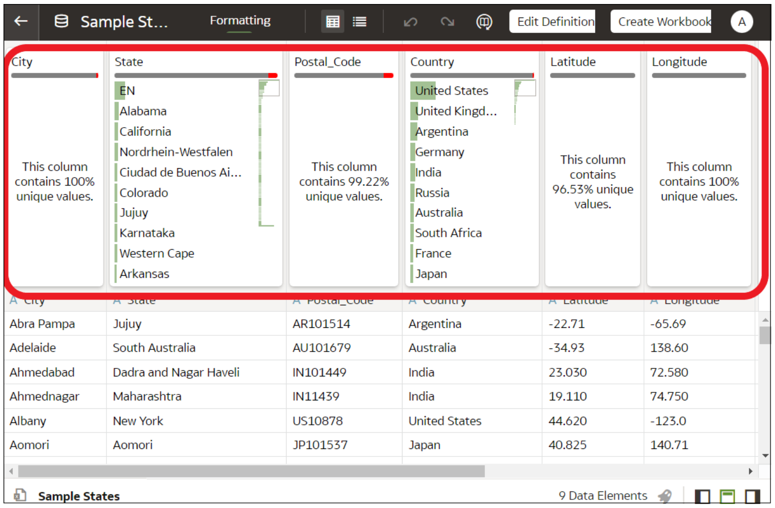


6. Click **Save Dataset**.

What are Quality Insights?

Oracle Analytics automatically analyzes data quality to help you cleanse and enrich your data.

When you edit a table in the join diagram or transform editor, Oracle Analytics analyzes the quality of the data and provides a visual summary known as a quality insight in a tile above each column. Quality insights enable you to explore your data, and use a visual overview of the content to assess and improve data quality. The quality summary is based on a sample of the data, but changes that you make are applied to all of your data. Quality insights display a frequency tile for text or a histogram for dates and numbers.



This column contains 100% unique values.

This column contains 99.22% unique values.

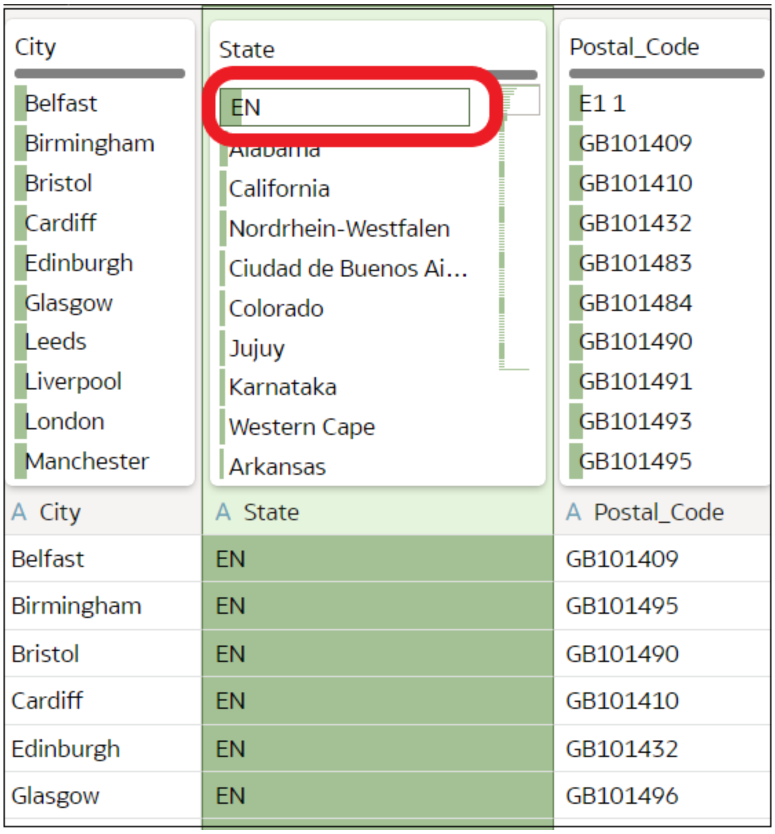
This column contains 96.53% unique values.

This column contains 100% unique values.

City	State	Postal_Code	Country	Latitude	Longitude
Abra Pampa	Jujuy	AR101514	Argentina	-22.71	-65.69
Adelaide	South Australia	AU101679	Australia	-34.93	138.60
Ahmedabad	Dadra and Nagar Haveli	IN101449	India	23.030	72.580
Ahmednagar	Maharashtra	IN11439	India	19.110	74.750
Albany	New York	US10878	United States	44.620	-123.0
Aomori	Aomori	JP101537	Japan	40.825	140.71

Quality tiles enable you to:

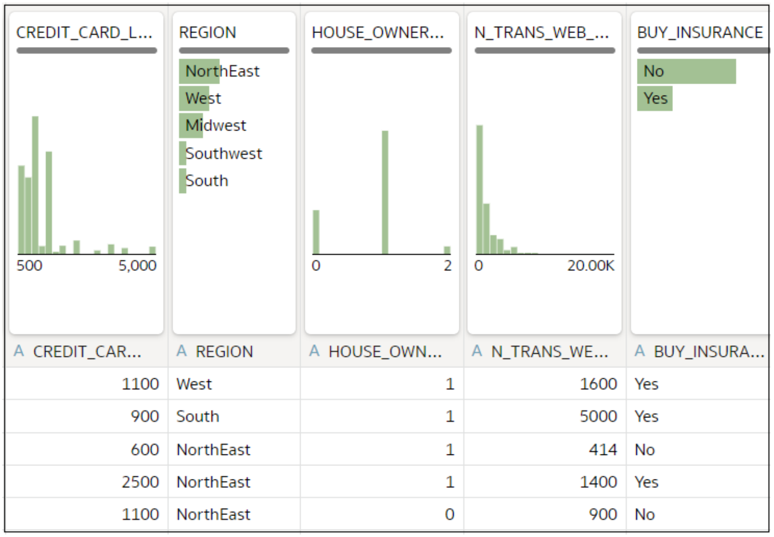
- **Explore** - Explore your data in real time using instant filtering, which enables you to temporarily filter data across all column in the table. You can filter on multiple values at the same time.



A City	A State	A Postal_Code
Belfast	EN	GB101409
Birmingham	EN	GB101495
Bristol	EN	GB101490
Cardiff	EN	GB101410
Edinburgh	EN	GB101432
Glasgow	EN	GB101496

The value being filtered is marked with a green box. Filters aren't added to the data preparation script.

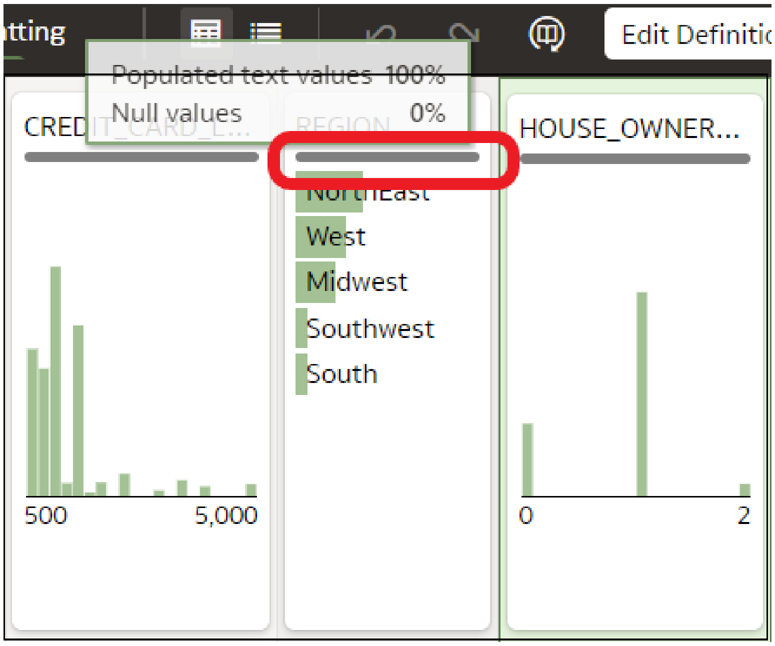
- **Review** - Use the interactive visualizations such as bar charts and histograms to evaluate your data and identify anomalies and outliers.



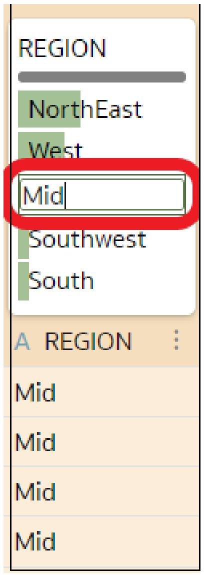
A CREDIT_CAR...	A REGION	A HOUSE_OWN...	A N_TRANS_WE...	A BUY_INSURA...
1100	West	1	1600	Yes
900	South	1	5000	Yes
600	NorthEast	1	414	No
2500	NorthEast	1	1400	Yes
1100	NorthEast	0	900	No

The quality tiles at the top of each column provide an instant assessment of the quality of content in that column based on a deep semantic understanding of the data.

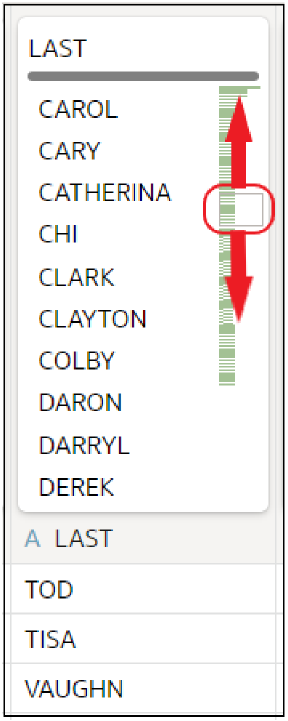
- **Assess** - Hover over the quality bar to see a popup summary showing the percentage of valid and invalid values.



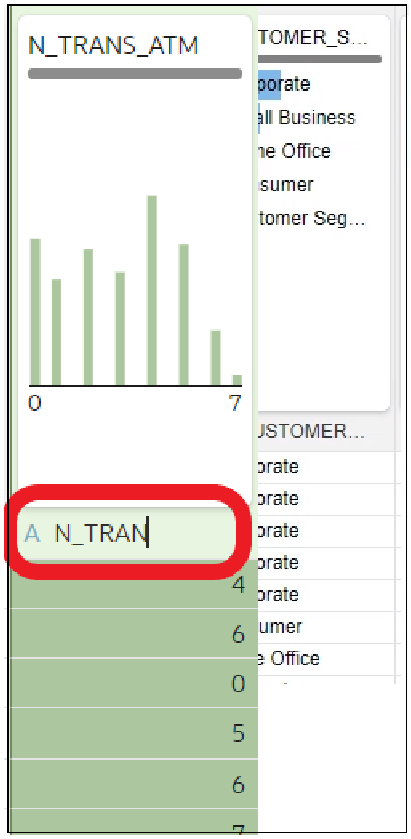
- **Replace or correct** - When you've identified anomalies and outliers, use inline replace to correct values. You get immediate feedback on the improvements to the data quality on the quality bar.



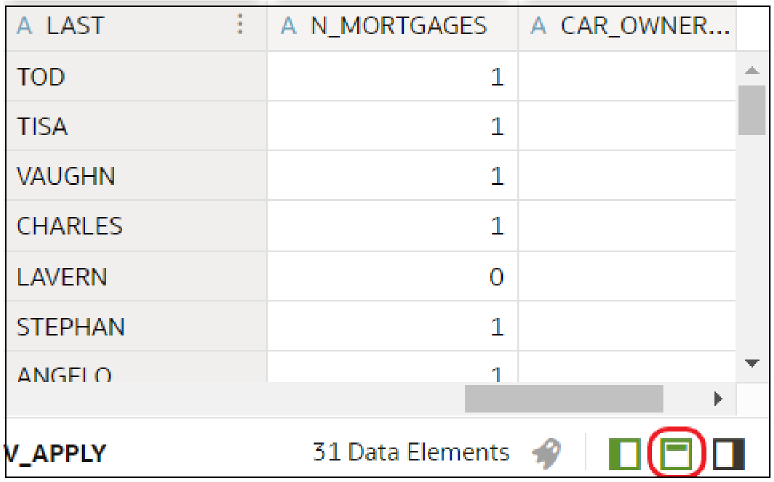
- **Scroll** - Navigate large datasets using the scrollable mini-map.



- **Rename columns** - Create more readable column names easily.



Make sure that you toggle the **Quality Tiles** option bottom right to display quality tiles.



A LAST	A N_MORTGAGES	A CAR_OWNER...
TOD	1	
TISA	1	
VAUGHN	1	
CHARLES	1	
LAVERN	0	
STEPHAN	1	
ANGELO	1	

Explore Your Data Using Quality Insights

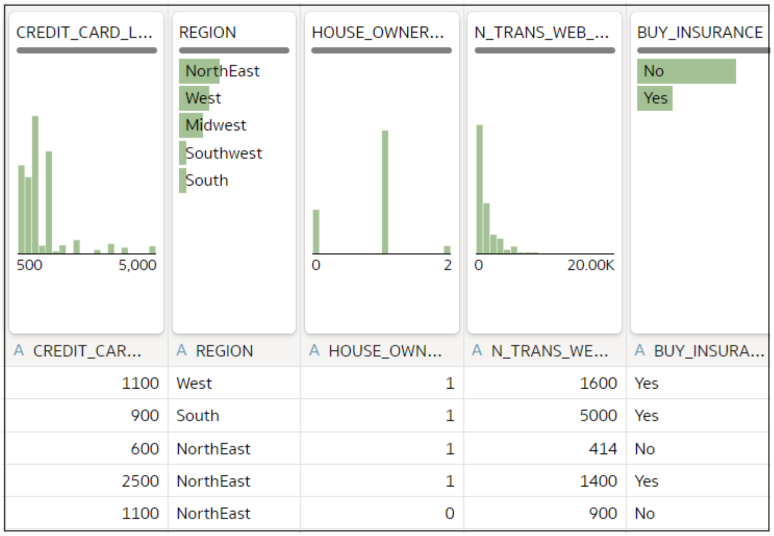
Explore your data using Oracle Analytic's quality insights. When you edit a table in the transform editor, Oracle Analytics provides a quality insight tile above each column.

1. On the Home page, open a dataset or workbook.

   - Select the dataset, click the **Actions menu**, and then select **Open**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.
   - Select the workbook that contains the dataset, click the **Actions menu**, select **Open**, then click **Data**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.

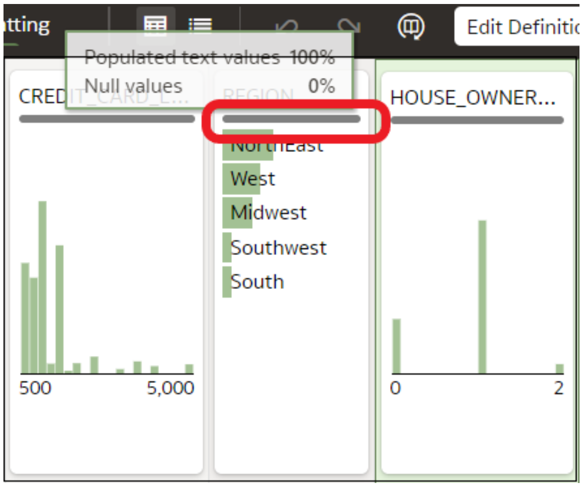
   > **Note:**
   >
   > Oracle Analytics displays a Data Diagram for single-table datasets or a Join Diagram for multi-table datasets. For file-based data sources, Oracle Analytics displays the transform editor directly.

2. Use the quality insights to get an instant assessment of the quality of content in that column based on a deep semantic understanding of the data.



A CREDIT_CAR...	A REGION	A HOUSE_OWN...	A N_TRANS_WE...	A BUY_INSURA...
1100	West	1	1600	Yes
900	South	1	5000	Yes
600	NorthEast	1	414	No
2500	NorthEast	1	1400	Yes
1100	NorthEast	0	900	No

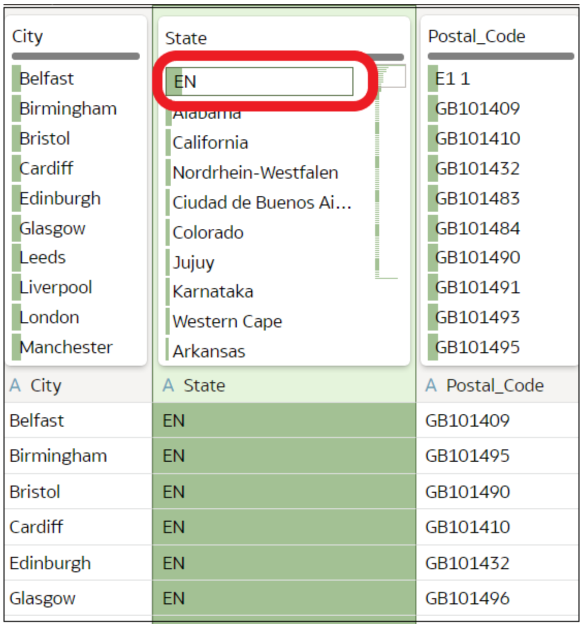
For example, hover over a quality bar to see how many valid and invalid values are in the column.



Populated text values 100%
Null values 0%

3. To explore your data by applying a temporary filter, click one or more values to use as a filter.

   Oracle Analytics highlights filter values with a green box. To unfilter a value, click the value again.



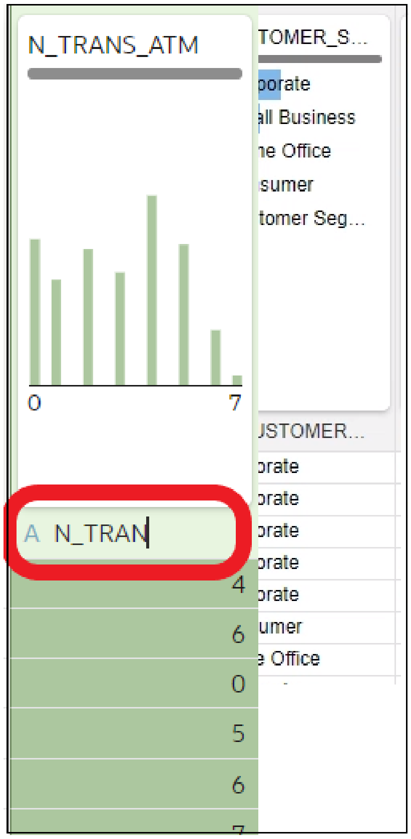
City	State	Postal_Code
Belfast	EN	E1 1
Birmingham	Alabama	GB101409
Bristol	California	GB101410
Cardiff	Nordrhein-Westfalen	GB101432
Edinburgh	Ciudad de Buenos Ai...	GB101483
Glasgow	Colorado	GB101484
Leeds	Jujuy	GB101490
Liverpool	Karnataka	GB101491
London	Western Cape	GB101493
Manchester	Arkansas	GB101495

A City	A State	A Postal_Code
Belfast	EN	GB101409
Birmingham	EN	GB101495
Bristol	EN	GB101490
Cardiff	EN	GB101410
Edinburgh	EN	GB101432
Glasgow	EN	GB101496

When you filter on values, Oracle Analytics updates all data columns to instantly display only the rows relating to the filter values you selected. If you filter a value nearer the bottom of a long list, you might have to scroll down the list to locate it so that you can deselect the value.

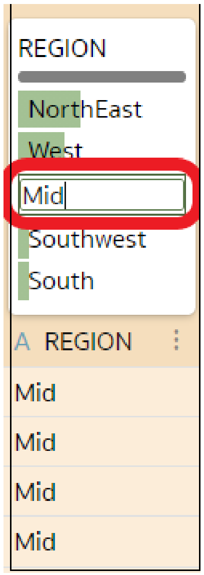
Note: Temporary filters that you apply in quality tiles aren't saved with your data (that is, aren't added to the data preparation script).

4. To rename a column, click the column name to edit the value.



5. To edit values.

   a. In the quality tile, double-click the value you want to change.

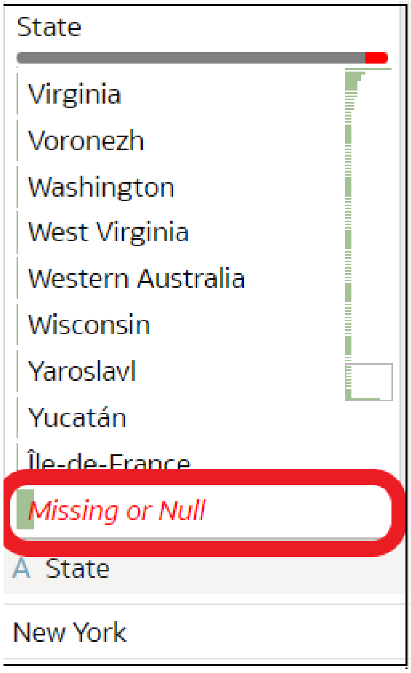


   b. Oracle Analytics inserts the value that you specify into every row that contained the original value.

6. To fix null or missing values:

a. In the quality tile, scroll to the last value in the list.

If your data has missing values, you'll see **Missing or Null** highlighted in red.



State

Virginia

Voronezh

Washington

West Virginia

Western Australia

Wisconsin

Yaroslavl

Yucatán

Île-de-France

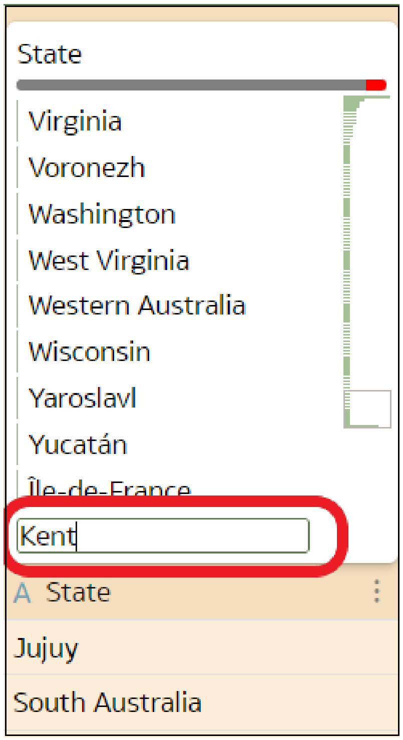
Missing or Null

A State

New York

b. Double-click **Missing or Null**, then enter the value you want to use.

Oracle Analytics inserts the value that you specify into every Missing or Null row.



State

Virginia

Voronezh

Washington

West Virginia

Western Australia

Wisconsin

Yaroslavl

Yucatán

Île-de-France

Kent

A State

Jujuy

South Australia

Changes that you apply in the quality insight tiles are added to the data preparation script (except for filters).

About Removing or Restoring Dataset Columns

Removing columns doesn't permanently delete them from the table, and you can re-add the removed columns as needed for enrichments and transformations, or for inclusion in the visualizations that you'll create from the dataset.

It's important to understand that removing a column is different than deleting a column from a dataset:

- Use Edit Definition to *remove* a column.
- The Transform editor doesn't indicate which columns were removed, and removing a column doesn't add a step to the Preparation Script Panel. To confirm which columns were deleted or to re-add them, go to Edit Definition.
- Use the Transform editor to *delete* a column.
- Deleting a column creates a step in the Preparation Script Panel. Remove the step to restore the column.

You can create dataset filters on removed columns.

In some cases profiling and generating quality insights for a table that contains many columns can be time and resource-intensive. If you're working with a table with many columns and want to increase system performance, Oracle recommends that you remove any columns that you don't need before you perform any enrichments or transformations.

See Hide or Delete a Column and Remove or Restore a Dataset Table's Columns.

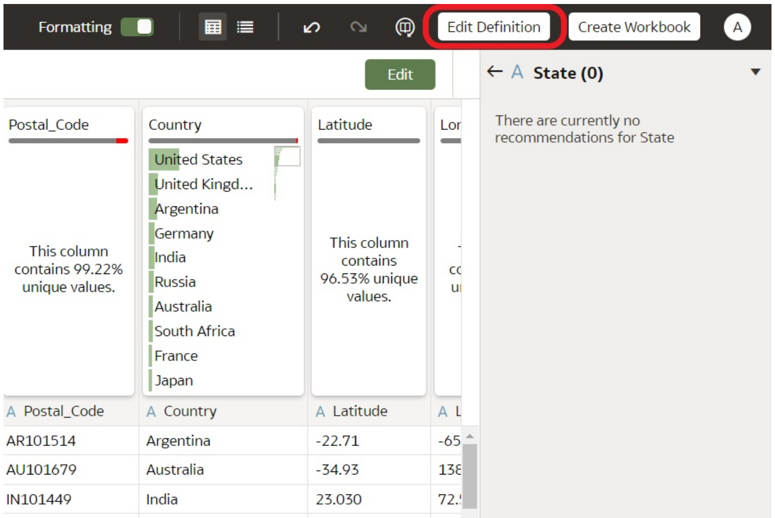
Remove or Restore a Dataset Table's Columns

You can remove columns from a dataset table when the dataset doesn't need the columns, and you can restore any removed columns.

See About Removing or Restoring Dataset Columns.

When you remove or restore a column, those changes are displayed in the table's view in the Transform editor. Oracle Analytics warns you if the column you've chosen to remove is used in the Transform editor's Preparation Script pane. But Oracle Analytics won't warn you if the column you've chosen to remove is used in a workbook or visualization.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.
4. In the **Join Diagram** go to the **Table Page Tabs** and click the table that you want to work with.
5. In the Transform editor, click **Edit Definition**.



A Postal_Code	A Country	A Latitude	A L
AR101514	Argentina	-22.71	-65
AU101679	Australia	-34.93	138
IN101449	India	23.030	72.!

6. (Optional) To remove columns, either click **Remove All** to remove all columns, or select the columns that you want to remove and click **Remove Selected**.
7. (Optional) To re-add columns, either click **Add All** to add all columns not included in the table, or select the columns that you want to add and click **Add Selected**.
8. Click **OK**.

Apply a Filter to a Dataset Table

In most cases when you add a table to the dataset, all of the table's column values are included. You can add filters so that the table contains only the values needed in the dataset.

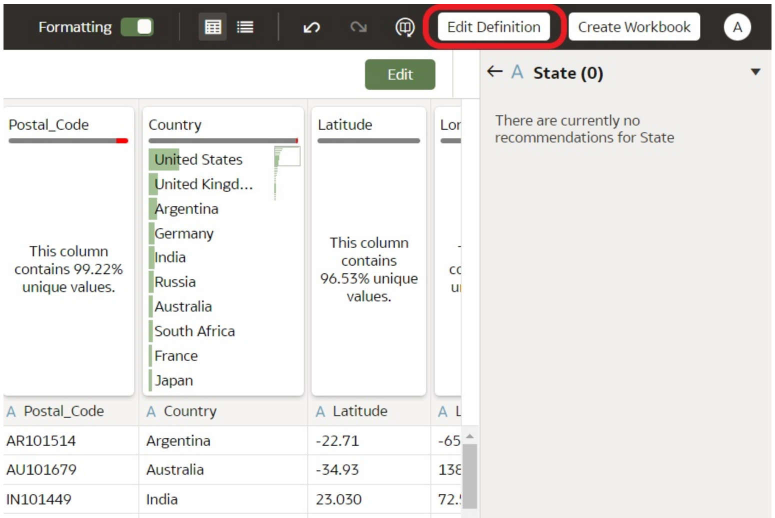
Applying a filter to a column limits everything in the table. For example, if a table contains data for every world region, you can create a filter on the COUNTRY_REGION column and set its value to Americas so that your table's columns contain data for North, Central, and South America.

Any filters that you apply limit what is displayed in the table's quality tiles, the Transform editor, and the table's data preview.

You can create filters on columns that have been removed from the table. See Remove or Restore a Dataset Table's Columns.

In some cases you might want to use the table's filter to query other visualizations in a workbook. See Include a Dataset Table in Data Source Queries.

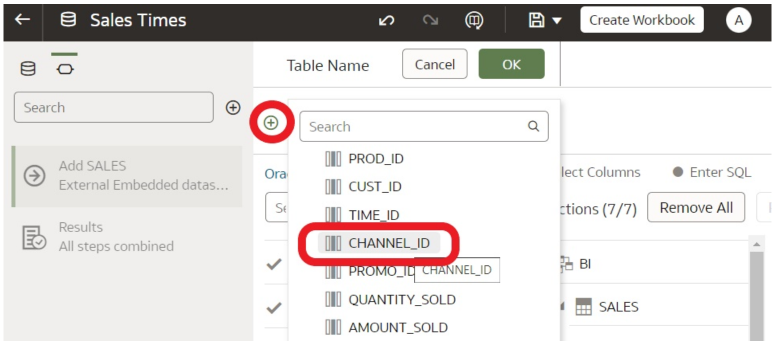
1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.
4. In the **Join Diagram** go to the **Table Page Tabs** and click the table that you want to work with.
5. In the Transform editor, click the **Edit Definition** button.



Postal_Code	Country	Latitude	L
AR101514	Argentina	-22.71	-65
AU101679	Australia	-34.93	138
IN101449	India	23.030	72.!

6. Click **Add Filter**.

7. Select a column and limit the values in the column, or click **Add Expression Filter** and write and apply a filter expression.



8. Click **OK**.

Specify Whether a Dataset Table Is Cached or Live

A dataset table's data access setting determines if the table's data is loaded into cache, or if the table gets its data directly from the data source.

You can set a table to **Automatic Caching** or **Live**.

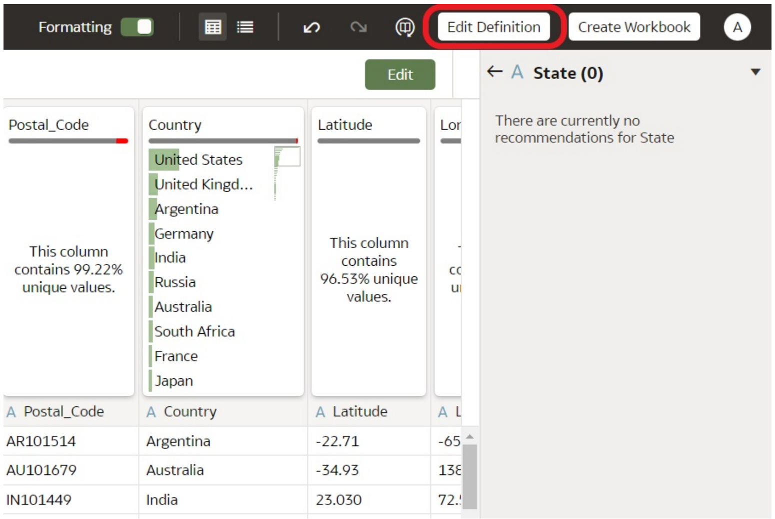
- **Automatic Caching** - If you select this option, then the table loads or reloads its data into cache. This option can provide faster performance when you refresh a table's data from the **Transform** editor or from a workbook. Selecting this option causes the **Reload** menu option to display at the table and dataset level.
  This option is limited to 2GB of data after compression. If the data is larger than 2GB or if the data takes too long to load, the data access mode uses a live query if the connection type supports it. If you want to use automatic caching and the data is larger than 2GB, then add column filters to the table, for example limit a date range to reduce the size of the data.
- **Live** - If you select this option, then the table gets its data directly from the data source. When a table is set to **Live**, the source system manages the table's data

source queries. This option is useful when the data is stored in a high-performance system such as Oracle Autonomous Data Warehouse. It also ensures that the latest data is used.

In a dataset with multiple tables, some tables can use automatic caching and some can include live data. If you reload multiple tables using the same connection and the reloading of data on one table fails, then any tables set to use automatic caching are switched to use live data.

For best performance, set all tables in a dataset to access data in the same way. When the tables in one dataset are mixed between automatic caching and live data, the system must resolve joins and performance varies based on the amount of data needed from each table to complete the queries.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.
4. In the **Join Diagram** go to the **Table Page Tabs** and click the table that you want to work with.
5. In the Transform editor, click **Edit Definition**.



There are currently no recommendations for State

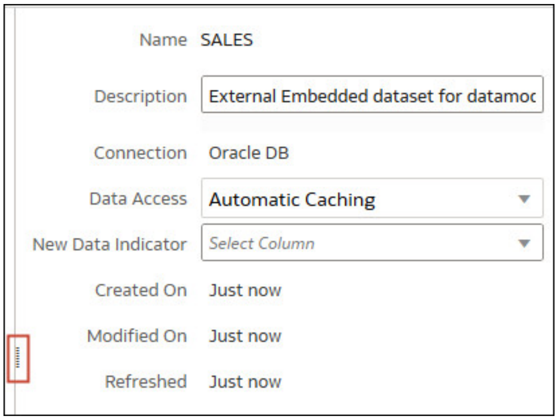
This column contains 99.22% unique values.

This column contains 96.53% unique values.

A Postal_Code	A Country	A Latitude	A L
AR101514	Argentina	-22.71	-65
AU101679	Australia	-34.93	138
IN101449	India	23.030	72.!

6. Confirm that the data access panel is displayed.

   If the data access panel isn't displayed, go to the right center edge of the window to locate and drag the handle to open the panel.



Name SALES

Description External Embedded dataset for datamoc

Connection Oracle DB

Data Access Automatic Caching

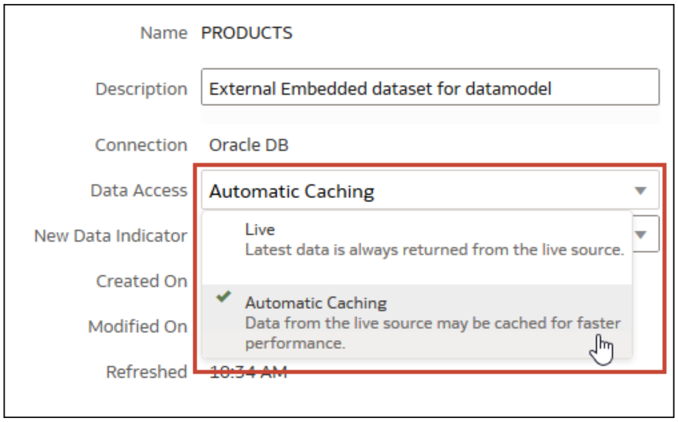
New Data Indicator *Select Column*

Created On Just now

Modified On Just now

Refreshed Just now

7. In the **Data Access** field, specify how you want the table to access its data.



Name PRODUCTS

Description External Embedded dataset for datamodel

Connection Oracle DB

Data Access Automatic Caching

New Data Indicator

Live
Latest data is always returned from the live source.

✓ Automatic Caching
Data from the live source may be cached for faster performance.

Created On

Modified On

Refreshed 10:34 AM

8. Click **OK**.

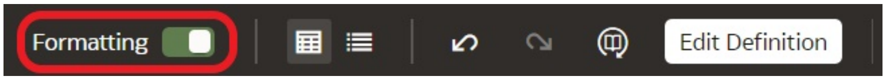
View a Dataset Table's Original Formatting

By default Oracle Analytics applies formatting to the numbers and dates in the dataset. You can toggle this default formatting off to view numbers and dates as they're formatted in the dataset's data source.

For example, when Oracle Analytics's default formatting is applied, dates are displayed as 06/20/2019. But when default formatting is turned off, then dates are displayed as 2019-06-20.

You can toggle the default formatting off and on, but you can't save the default formatting. To modify a column's formatting, see Adjust the Display Format of a Date or Number Column.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.
4. In the **Join Diagram** go to the **Table Page Tabs** and click the table that you want to work with.
5. In the Transform editor, click the **Formatting** toggle to turn formatting off or on.



Create Datasets from Files

This topic describes how to upload files and use them to create datasets.

Topics:

- About Files for Datasets
- Create a Dataset from a File Uploaded From Your Computer
- Create a Dataset From a File Uploaded from Dropbox, Google Analytics, or Google Drive

About Files for Datasets

You can upload Microsoft Excel spreadsheets (XLSX and XLS), CSV files, and TXT files and then create datasets from them. The maximum file size you can upload is 250 MB and the data column limit for a single file is 250 columns.

You must structure an Excel spreadsheet in a data-oriented way with no pivoted data. These rules apply for Excel tables:

- Tables must start in Row 1 and Column 1 of the Excel file.
- Tables must have a regular layout with no gaps or inline headings. An example of an inline heading is one that's repeated on every page of a printed report.
- Row 1 must contain the names of the columns in the table. For example, Customer Given Name, Customer Surname, Year, Product Name, Amount Purchased, and so on. In this example:
  - Column 1 has customer given names.
  - Column 2 has customer surnames.
  - Column 3 has year values.
  - Column 4 has product names.
  - Column 5 has the amount each customer purchased for the named product.
- The names in Row 1 must be unique. If two columns hold year values, then you must add a second word to one or both of the column names to make them unique. For example, if you have two columns named Year Lease, then you can rename the columns to Year Lease Starts and Year Lease Expires.
- Row 2 and greater must contain the data for the table, and those rows can't contain column names.
- Data in a column must be of the same type because it's often processed together. For example, the Amount Purchased column must contain only numbers (and possibly nulls), enabling it to be summed or averaged. The Given Name and Surname columns must contain text values because they might be concatenated, and you might need to split dates into months, quarters, or years.
- Data must be at the same granularity. A table can't contain both aggregations and details for those aggregations. For example, suppose that you have a sales table at the

granularity of Customer, Product, and Year that contains the sum of Amount Purchased for each Product by each Customer by Year. In this case, you wouldn’t include invoice level details or daily summary values in the same table, because the sum of the Amount Purchased values wouldn't be calculated correctly. If you must analyze at the invoice level, the day level, and the month level, then you can do either of the following:

- Have a table of invoice details: Invoice Number, Invoice Date, Customer, Product, and Amount Purchased. You can roll these up to the day, month, or quarter.
- Have multiple tables, one at each granular level (invoice, day, month, quarter, and year).

Create a Dataset from a File Uploaded From Your Computer

You can upload Excel spreadsheets (XLSX or XLS), CSV files, and TXT files from your computer and use them to create datasets.

> **Note:**
>
> Files aren't available to create or include in a dataset with multiple tables.

Confirm that the file that you want to upload meets these requirements:

- The file is either an Excel spreadsheet in .XLSX or .XLS format, a CSV file, or a TXT file.
- The spreadsheet contains no pivoted data.
- The spreadsheet is structured properly for import and use as a dataset. See About Files for Datasets.

1. On the Home page, click **Create** and then click **Dataset**.
2. In the Create Dataset dialog, either drag and drop a file to the dialog, or click the Upload icon to browse your computer for a file to upload.
3. In the Add Dataset page, provide a name for the dataset.
4. If you’re uploading a CSV or TXT file, then in the **Separated By**, **Thousand Separator**, and **Decimal Separator** fields, confirm or change the default delimiters.

   To specify a custom delimiter, choose Custom in the **Separated By** field and enter the character you want to use as the delimiter. In the CSV or TXT file, a custom delimiter must be one character. The following example uses a pipe (|) as a delimiter: Year|Product|Revenue|Quantity|Target Revenue| Target Quantity.
5. Click **Add** to upload the file and create the dataset.

Create a Dataset From a File Uploaded from Dropbox, Google Analytics, or Google Drive

You can upload Excel spreadsheets (XLSX or XLS), CSV files, and TXT files from Dropbox, Google Analytics, or Google Drive and use them to create datasets.

> **Note:**
>
> Files aren't available to create or include in a dataset with multiple tables.

Before you create the dataset, confirm that the connection you need exists. See View Available Connections.

Confirm that the file that you want to upload meets these requirements:

- The file is either an Excel spreadsheet in .XLSX or .XLS format, a CSV file, or a TXT file.
- The spreadsheet contains no pivoted data.
- The spreadsheet is structured properly for import and use as a dataset. See About Files for Datasets.

1. On the Home page, click **Create** and then click **Dataset**.
2. In the Create Dataset dialog, select a connection.
3. Browse for and select the file that you want to upload.
4. In the preparation canvas, provide a name for the dataset.
5. If you’re uploading a CSV or TXT file, then in the **Separated By**, **Thousand Separator**, and **Decimal Separator** fields, confirm or change the default delimiters.

   To specify a custom delimiter, choose Custom in the **Separated By** field and enter the character you want to use as the delimiter. In the CSV or TXT file, a custom delimiter must be one character. The following example uses a pipe (|) as a delimiter: Year|Product|Revenue|Quantity|Target Revenue| Target Quantity.
6. Click **Add** to upload the file and create the dataset.

Create a Dataset from an Oracle Application's Subject Area

You can create datasets with multiple tables from subject areas stored in Oracle Applications such as Oracle Fusion Applications with Oracle Transactional Business Intelligence.

When you drag and drop a subject area to the **Join Diagram**, by default Oracle Analytics doesn't include any columns in the dataset table. You must specify which columns to include in the table.

Oracle Analytics doesn't automatically join tables created from subject areas. You need to manually join these tables. See Understand Dataset Table Joins.

1. On the Home page, click **Create** and then click **Dataset**.
2. In the Create Dataset dialog, select an Oracle Applications connection.
3. In the Dataset editor, go to the **Connections** pane and browse or search for a subject area.

4. Drag and drop one or more subject areas to the **Join Diagram**.
5. To add columns to a table, go to the **Table Page Tabs**, click a subject area table, and use the **Transform Editor** to specify which columns to include in the table. Click **OK**.
6. In the **Join Diagram**, locate the table that you want to join, hover over it to select it, and then click and drag and drop it on to the table that you want to join it to. Open the Join editor to inspect or update the join type and conditions.
7. Click **Save**.
8. Enter a name and click **OK**.

Create a Dataset from an Oracle Application's Analysis

You can create datasets with multiple tables from analyses stored in Oracle applications such as Oracle Fusion Applications with Oracle Transactional Business Intelligence.

Oracle Analytics doesn't automatically join tables created from analyses. You need to manually join these tables. See Understand Dataset Table Joins.

1. On the Home page, click **Create** and then click **Dataset**.
2. In the Dataset editor, go to the **Connections** pane and browse or search for an analysis.
3. Drag and drop one or more analysis to the **Join Diagram**.
4. In the **Join Diagram**, locate the table that you want to join, hover over it to select it, and then click and drag and drop it on to the table that you want to join it to. Open the Join editor to inspect or update the join type and conditions.
5. Click **Save**.
6. Enter a name and click **OK**.

Create a Dataset from an Essbase Connection

You can use an Essbase connection to create a dataset.

> **Note:**
>
> Essbase connections aren't available to create or include in a dataset with multiple tables.

Before you create the dataset, confirm that the data source connection you need exists. See View Available Connections.

1. On the Home page, click **Create** and then click **Dataset**.
2. In the Create Dataset dialog, select the Essbase connection.
3. In the Transform editor's Add Dataset step, double-click the Essbase cube that you want to use in the dataset.
4. Optional: Select an **Alias** value.

If you select an alias value other than the default, then values from the selected alias table are displayed in visualizations that use this Essbase dataset.

5. Click **Add** to save the dataset and go to the Transform editor to transform and enrich the dataset's data.

4

Enrich and Transform Data

Data preparation involves cleansing, standardizing, and enriching your dataset before you visualize the data.

Topics:

- About Data Enrichment and Transformation in Oracle Analytics
- Enrich and Transform Your Data
- Accept Enrichment Recommendations
- Transform Data Using Column Menu Options
- Transform Data Using Replace
- Convert Text Columns to Date or Time Columns
- Adjust the Display Format of a Date or Number Column
- Create a Bin Column When You Prepare Data
- Edit Column Properties
- Hide or Delete a Column
- Restore a Hidden or Deleted Column
- Add Columns to a Dataset
- Create Calculated Data Elements in a Workbook
- Edit the Data Preparation Script
- Enrich and Transform Reference

About Data Enrichment and Transformation in Oracle Analytics

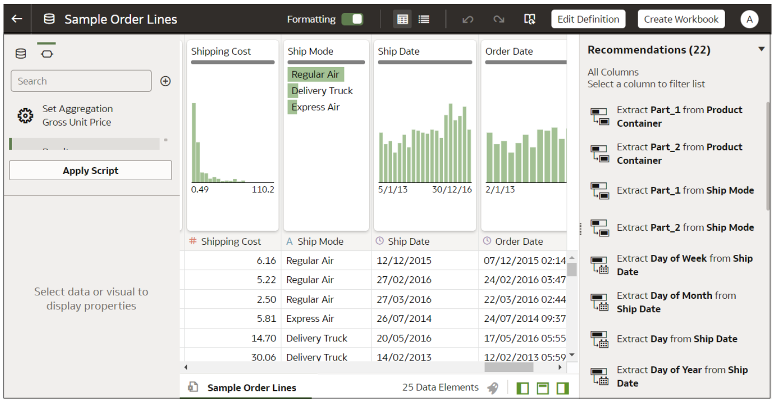
Oracle Analytics makes it easy to enrich and transform your data before you make it available for analysis.

Tutorial

Navigate to the transform editor

Workbooks - If you open a workbook, click the Data page, and in the **Data Diagram**, select the dataset you want to prepare. If the dataset contains multiple tables, you'll see a **Join Diagram**, with a tab for each table. Select a table to open it in the transform editor.

Datasets - If you open a dataset, you'll see the transform editor. If the dataset contains multiple tables, you'll see a **Join Diagram**, with a tab for each table. Select a table to open it in the transform editor.



Shipping Cost	Ship Mode	Ship Date	Order Date
6.16	Regular Air	12/12/2015	07/12/2015 02:14
5.22	Regular Air	27/02/2016	24/02/2016 03:47
2.50	Regular Air	27/03/2016	22/03/2016 02:44
5.81	Express Air	26/07/2014	24/07/2014 09:37
14.70	Delivery Truck	20/05/2016	17/05/2016 05:55
30.06	Delivery Truck	14/02/2013	12/02/2013 05:59

Recommendations (22)

All Columns
Select a column to filter list

- Extract **Part_1** from **Product Container**
- Extract **Part_2** from **Product Container**
- Extract **Part_1** from **Ship Mode**
- Extract **Part_2** from **Ship Mode**
- Extract **Day of Week** from **Ship Date**
- Extract **Day of Month** from **Ship Date**
- Extract **Day** from **Ship Date**
- Extract **Day of Year** from **Ship Date**

Applying recommended transformations

When you create a workbook and add a dataset to it, the data undergoes column level profiling that runs on a representative sample of the data. After profiling the data, you can implement transformation and enrichment recommendations provided for the recognizable columns in the dataset. The following types of recommendations are provided to perform single-click transforms and enrichments on the data:

- Global positioning system enrichments such as latitude and longitude for cities or zip codes.
- Reference-based enrichments, for example, adding gender using on the person's first name as the attribute for make the gender decision.
- Column concatenations, for example, adding a column with the person's first and last name.
- Part extractions, for example, separating out the house number from the street name in an address.
- Semantic extractions, for example, separating out information from a recognized semantic type such as domain from an email address.
- Date part extractions, for example, separating out the day of week from a date that uses a month, day, year format to make the data more useful in the visualizations.
- Full and partial obfuscation or masking of detected sensitive fields.
- Recommendations to delete columns containing detected sensitive fields.
- Custom Knowledge enrichments that your administrator has added to Oracle Analytics.

Applying your own transformations

In addition to the recommended transformations that you see in the Column Action Panel, you can create your own transformations in different ways:

- Use the **Options** menu at the top of each column to apply common transformations such as Rename, Uppercase, and Trim. See Transform Data Using Column Menu Options.
- Click **Add Preparation Step** in the Data Panel to add a column based on a custom transformation. You can build your column using a wide range of functions, for example, operators, math, aggregates, and conversion. See Add Columns to a Dataset.

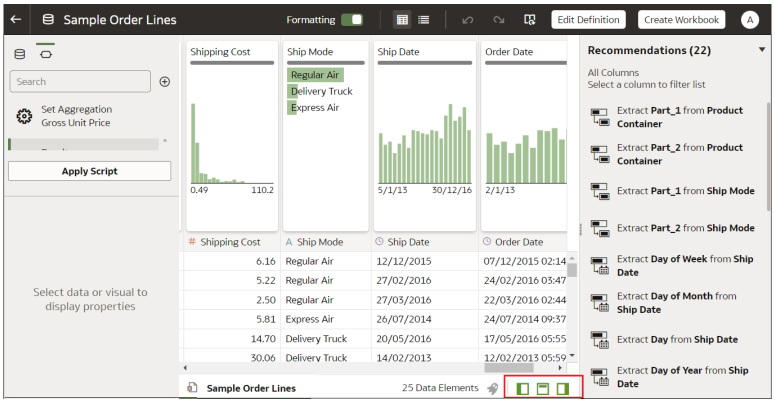
- Use the **Element** options pane in the bottom left hand corner of the Data Panel to select **Treat As** - Attribute or Measure, or change the default **Aggregation** type.

How dataset transformations affect workbooks and data flows

The data transformation and enrichment changes that you apply to a dataset affect all workbooks and data flows that use the same dataset. When you open a workbook that shares the dataset, you'll see a message indicating that the workbook uses updated data. When you refresh data in a dataset, the preparation script changes are automatically applied to the refreshed data.

Set up your transform editor

Before you start, it's best to configure your transform editor by displaying the Data Panel, Recommendations Panel, and Quality Tiles. Use these toggle options displayed bottom right: **Toggle Data Panel** , **Toggle Column Action Panel** , and **Toggle Quality Tiles**



# Shipping Cost	A Ship Mode	Ship Date	Order Date
6.16	Regular Air	12/12/2015	07/12/2015 02:14
5.22	Regular Air	27/02/2016	24/02/2016 03:47
2.50	Regular Air	27/03/2016	22/03/2016 02:44
5.81	Express Air	26/07/2014	24/07/2014 09:37
14.70	Delivery Truck	20/05/2016	17/05/2016 05:55
30.06	Delivery Truck	14/02/2013	12/02/2013 05:59

Recommendations (22)

All Columns
Select a column to filter list

- Extract **Part_1** from **Product Container**
- Extract **Part_2** from **Product Container**
- Extract **Part_1** from **Ship Mode**
- Extract **Part_2** from **Ship Mode**
- Extract **Day of Week** from **Ship Date**
- Extract **Day of Month** from **Ship Date**
- Extract **Day** from **Ship Date**
- Extract **Day of Year** from **Ship Date**

Enrich and Transform Your Data

Before you deploy your visualization workbooks, you typically enrich and transform your data. For example, you might rename data columns, fix cell phone data, or add calculations.

1. On the Home page, open a dataset or workbook.
   - Select the dataset, click the **Actions menu**, and then select **Open**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.
   - Select the workbook that contains the dataset, click the **Actions menu**, select **Open**, then click **Data**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.

   > **Note:**
   >
   > Oracle Analytics displays a Data Diagram for single-table datasets or a Join Diagram for multi-table datasets. For file-based data sources, Oracle Analytics displays the transform editor directly.

2. Select columns and use the enrich and transform tools to enhance your data.

- Use the **Recommendations** panel to apply suggested enrichments and transformations.
- Use the **Options** menu at the top of each column to apply common transformations such as Rename, Uppercase, and Trim.
- From the **Options** menu at the top of a column, click **Edit** to enhance columns with functions and expressions from the Oracle Analytics functions library. For example, aggregates, strings, expressions, and math functions.
- Click **Add Preparation Step** in the Data Panel to add a column based on a custom transformation. You can build your column using a wide range of functions and expressions from the Oracle Analytics functions library. For example, aggregates, strings, expressions, and math functions.
- Use the properties pane at the bottom of the Data Panel to review and change a column's type and aggregation. For example, you might select **Treat As** and change from attribute to measure, or change the default **Aggregation** type from Sum to Average.
- Use the Quality Insights tile above each column to explore and enrich your data.

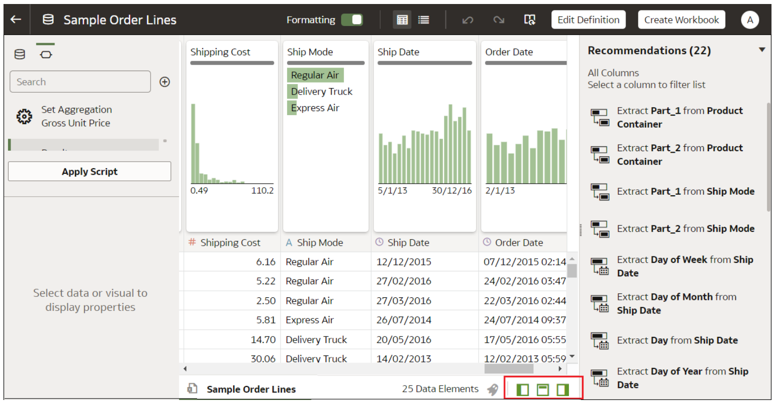
When you edit data, a step is automatically added to the Preparation Script pane. A blue dot indicates that a step hasn't been applied.

3. To save your data preparation changes and apply them to your data, click **Save**.

   If you're working with a single table dataset or a file-based dataset, click **Apply Script**.

4. If you're working in a workbook, click **Save** and click **Visualize** to review the enriched columns.

Note: It's best to configure your transform editor by displaying the Data Panel, Recommendations Panel, and Quality Insights. Use these toggle options displayed bottom right **Toggle Data Panel** , **Toggle Column Action Panel** , and **Toggle Quality Insights**.



Set Aggregation
Gross Unit Price

Select data or visual to display properties

Shipping Cost	Ship Mode	Ship Date	Order Date
6.16	Regular Air	12/12/2015	07/12/2015 02:14
5.22	Regular Air	27/02/2016	24/02/2016 03:47
2.50	Regular Air	27/03/2016	22/03/2016 02:44
5.81	Express Air	26/07/2014	24/07/2014 09:37
14.70	Delivery Truck	20/05/2016	17/05/2016 05:55
30.06	Delivery Truck	14/02/2013	12/02/2013 05:59

Recommendations (22)

All Columns
Select a column to filter list

- Extract **Part_1** from **Product Container**
- Extract **Part_2** from **Product Container**
- Extract **Part_1** from **Ship Mode**
- Extract **Part_2** from **Ship Mode**
- Extract **Day of Week** from **Ship Date**
- Extract **Day of Month** from **Ship Date**
- Extract **Day** from **Ship Date**
- Extract **Day of Year** from **Ship Date**

Accept Enrichment Recommendations

When you edit a dataset in the transform editor, Oracle Analytics suggests recommendations for enriching your data.

You can upload or open an existing dataset to modify the data using enrichment recommendations. After making the changes to the dataset, you can create a workbook.

If an enrichment recommendation adds information to data such as enhancing a zip code attribute column with the state name, a new column is added to the dataset containing the name of the states associated with the zip codes. When you select a recommendation, the change is added to the Preparation Script and is applied when you click **Apply Script** or **Save Dataset**. If you delete or undo the change, the recommendation is displayed once again as an available option in the Recommendations panel.

1. On the Home page, open a dataset or workbook.
   - Select the dataset, click the **Actions menu**, and then select **Open**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.
   - Select the workbook that contains the dataset, click the **Actions menu**, select **Open**, then click **Data**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.

   > **Note:**
   >
   > Oracle Analytics displays a Data Diagram for single-table datasets or a Join Diagram for multi-table datasets. For file-based data sources, Oracle Analytics displays the transform editor directly.

2. Select a column to enrich, then in the Recommendations Panel click recommendations to add them as steps to the Preparation Script.

   > **Note:**
   >
   > Make sure that you've toggled the **Column Action Panel** option bottom right to display recommendations.

3. Use the Preparation Script panel to review your data enrichments and make changes if required.

   For example, hover-over a step and click X to remove an enrichment.

4. To save your data preparation changes and apply them to your data, click **Save Dataset**.

   If you're working with a single table dataset or a file-based dataset, click **Apply Script**.

> **Note:**
>
> If the System Knowledge or Custom Knowledge has been updated and you have author privileges on the dataset, you can reprofile and refresh the results for the currently open dataset by clicking **Refresh Profile Results** on the toolbar (). Oracle Analytics usually manages this refresh for you, so in most cases you won't need to use this option. Use **Refresh Profile Results** to reprofile the data if the System Knowledge has been updated after an uprgrade, or the Custom Knowledge has been updated and you want to incorporate this in the currently open dataset.

Transform Data Using Column Menu Options

You can use column menu options to transform data. For example, you might convert text to uppercase or trim leading and trailing spaces from data.

If you don’t apply the transformation script and close the workbook or the dataset, you lose all the data transform changes you’ve performed.

1. On the Home page, open a dataset or workbook.
   - Select the dataset, click the **Actions menu**, and then select **Open**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.
   - Select the workbook that contains the dataset, click the **Actions menu**, select **Open**, then click **Data**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.

   > **Note:**
   >
   > Oracle Analytics displays a Data Diagram for single-table datasets or a Join Diagram for multi-table datasets. For file-based data sources, Oracle Analytics displays the transform editor directly.

2. In the Preview data panel, select a column to transform.
3. Click **Options**, and select a transformation option (for example, **Uppercase** or **Trim**).

   Oracle Analytics adds a step to the **Preparation Script** panel, ready to be applied when you click **Apply Script** or **Save Dataset**.

Transform Data Using Replace

Transform data in a column using a simple pattern match or a regular (regex) expression. For example, in a State column, you might change CA to California.

You can use any Perl Compatible Regular Expression (PCRE), which are also referred to as regex expressions. Transforms are applied to every applicable row in a dataset.

1. On the Home page, open a dataset or workbook.

- Select the dataset, click the **Actions menu**, and then select **Open**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.
- Select the workbook that contains the dataset, click the **Actions menu**, select **Open**, then click **Data**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.

> **Note:**
>
> Oracle Analytics displays a Data Diagram for single-table datasets or a Join Diagram for multi-table datasets. For file-based data sources, Oracle Analytics displays the transform editor directly.

2. Select the text column that you want to transform.
3. Click **Options**, and then click **Replace**.
4. In the Replace step editor, specify how you'd like to match values.
   - Click **Use regular expression** to match using a complex regular expression (regex).
   - Click **Match entire values only** to change "male" to "M" but not "female" to "feM".
5. In the **String to replace** field and **New string** field, specify the literal text or regular expressions to search and replace (case-sensitive).

   Examples:

   - To change "2553 Bolman Court" to "2553 #Bolman# #Court#", select **Use regular expression**, enter `([A-Za-z]+)` in the **String to replace** field, and enter `#$1#` in the **New String** field.
   - To change "male" with "M", select **Match entire values only**, enter `male` in the **String to replace** field, and enter `M` in the **New String** field. . This changes "male" to "M" but won't change "female" to "feM".

   When you complete both fields, you'll see a preview of the transformed data in the preview pane. Adjust the replace options until you get the transformation you want in the preview pane.
6. Click **Add Step**.

   Oracle Analytics adds a step to the Preparation Script panel, ready to be applied when you click **Apply Script** or **Save Dataset**.

Example Replace Transforms Using Regular Expressions

Here're just a few examples of using regular (regex) expressions to transform data.

In these examples, the Search Expression column shows what you'd enter in the **String to replace** field, and the Replace Value shows what you'd enter in the **New string** field.

Table 4-1   Example replace transforms

Search Expression	Replace Value	Original Text	Replaced Text	Notes
`@([a-z]+)(?=\.[a-z]{3})`	@example	MichelePFalk@yahoo.com	MichelePFalk@example.com	This example replaces domain details in email addresses.
`^Gray|Grey$`	Silver	Grey Gray Graystone	Silver Silver Graystone	The ^$ characters mean only look for entire string matches. The vertical bar is the regular expression for OR, so in this case the regular expression looks for either "Gray" or "Grey" and replace with "Silver". The string Graystone isn't transformed because the regular expression is looking for entire value matches only.
`\d+`	9999	8398 Park Street 123 Oracle Parkway	9999 Park Street 9999 Oracle Parkway	This regular expression looks for one" \d" or more "+" digits and replaces them with "9999". The replace also works when the original text has only three digits.
`([A-z]+)(\d+)`	$2	UA101654 US829383	101654 829383	This regular expression is looking for one "[A-z]" or more "+" consecutive letters followed by one" \d" or more "+" digits - each of the two expressions is bounded by parentheses, which captures two groups - the first one ([A-z]) and the second one (\d+) - these groups are numbered automatically and can then be used in the replacement by using the dollar sign for the second group, for example, "$2".
`([A-z]+)(\d+)`	Postal Code: $2	UA101654 US829383	Postal Code: 101654 Postal Code: 829383	This example uses the same match expression as the previous example, except that it shows how to insert your own replacement text in addition to a group. Text can be inserted before and after a recalled match group.

Convert Text Columns to Date or Time Columns

You can convert any text column to a date, time, or timestamp column.

For example, you can convert an attribute text column to a true date column.

1. On the Home page, open a dataset or workbook.
   - Select the dataset, click the **Actions menu**, and then select **Open**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.
   - Select the workbook that contains the dataset, click the **Actions menu**, select **Open**, then click **Data**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.

> **Note:**
>
> Oracle Analytics displays a Data Diagram for single-table datasets or a Join Diagram for multi-table datasets. For file-based data sources, Oracle Analytics displays the transform editor directly.

2. Hover over the column that you want to convert.
3. Click **Options** and select one of the conversion options, for example, **Convert to Number**, or **Convert to Date**.

   Oracle Analytics adds a step to the Preparation Script panel, ready to be applied when you click **Apply Script** or **Save Dataset**.

   The changes you make apply to all workbooks using that data source.

Adjust the Display Format of a Date or Number Column

You can adjust the display format and granularity level of a date column.

For example, you might want to change the format of a transaction date column from `November 1, 2017` to `2017-11-01`, or change the level of granularity from year to month.

1. On the Home page, open a dataset or workbook.
   - Select the dataset, click the **Actions menu**, and then select **Open**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.
   - Select the workbook that contains the dataset, click the **Actions menu**, select **Open**, then click **Data**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.

> **Note:**
>
> Oracle Analytics displays a Data Diagram for single-table datasets or a Join Diagram for multi-table datasets. For file-based data sources, Oracle Analytics displays the transform editor directly.

2. Hover over the column you'd like to edit and click **Options**.
3. To change the format of a date column, click **Convert to Date**, then click **Source Format** and select the format you want to use in the visualization.
4. To change the format of a number column, click **Edit**, and use functions in the functions library to change the format.

   For example, to round up a number column DISCOUNT_VALUE from two decimal places to no decimal places, specify `ROUND(DISCOUNT_VALUE, 0)`.
5. Click **Add Step**.

Create a Bin Column When You Prepare Data

Binning a measure creates a new column based on the value of the measure. For example, you might want to assign temperature values into two bin categories for less than or equal to 70 degrees (<=70) and more than 70 degrees (>70).

You can assign a value to the bin dynamically by creating the number of equal-sized bins or by explicitly specifying the range of values for each bin. You can create a bin column based on a data element.

1. On the Home page, open a dataset or workbook.
   - Select the dataset, click the **Actions menu**, and then select **Open**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.
   - Select the workbook that contains the dataset, click the **Actions menu**, select **Open**, then click **Data**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.

   > **Note:**
   >
   > Oracle Analytics displays a Data Diagram for single-table datasets or a Join Diagram for multi-table datasets. For file-based data sources, Oracle Analytics displays the transform editor directly.

2. Select the column that you want to modify.
3. Click **Options** for the selected column, and select **Bin**.
4. In the Bin step editor, specify the options for the bin column.
   - Enter a number or use the arrows to increment or decrement the number of bins.
   - Based on your selection in the **Method** field, the range and count of the bins are updated.
     - In the **Manual** method, you select the boundary (that is, minimum and maximum) of each bin. You can also change the default name of each bin.
     - In the **Equal Width** method, the boundary of each bin is the same, but the count differs. Based on your selection in the **Bin Labels** field, the bin column labels are updated.
     - In the **Equal Height** method, the height of each bin is the same or very slightly different but the range is equal.
   - If you select the **Equal Width** method, click to select a dimension (that is, an attribute data element) on which to apply the bin.
5. Click **Add Step**.

   Oracle Analytics adds a step to the Preparation Script panel, ready to be applied when you click **Apply Script** or **Save Dataset**.

Edit Column Properties

You can edit the properties of a dataset column in the transform editor. For example, you might change a column from a measure to an attribute, or change the aggregation of a number column from Sum to Average.

1. On the Home page, open a dataset or workbook.
   - Select the dataset, click the **Actions menu**, and then select **Open**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.
   - Select the workbook that contains the dataset, click the **Actions menu**, select **Open**, then click **Data**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.

   > **Note:**
   >
   > Oracle Analytics displays a Data Diagram for single-table datasets or a Join Diagram for multi-table datasets. For file-based data sources, Oracle Analytics displays the transform editor directly.

2. Select a column and use the properties pane at the bottom of the Data Panel to review and change the column's properties. For example:
   - Use the **General** tab to change the selected column properties such as Treat As and Data Type.
   - Use the **Treat As** option to toggle the column from a measure to an attribute.
   - Use the **Data Type** and **Aggregation** options to change column details.

   For each property change, Oracle Analytics adds a step to the Preparation Script panel, ready to be applied when you click **Apply Script** or **Save Dataset**.

Hide or Delete a Column

In the Transform editor, you can hide or delete a column from the dataset. Hiding or deleting columns doesn't permanently remove them, and you can restore columns when needed.

It's important to understand that deleting a column is different than removing a column from a dataset:

- Use the Transform editor to *delete* a column.
- Deleting a column creates a step in the Preparation Script Panel. Remove the step to restore the deleted column.
- Use Edit Definition to *remove* a column.
- The Transform editor doesn't indicate which columns were removed, and removing a column doesn't add a step to the Preparation Script Panel. To confirm which columns were removed or to re-add them, go to Edit Definition.

See Remove or Restore a Dataset Table's Columns.

You can also hide columns in the dataset. Hiding a column is useful when you want to use a column to create a formula but don't need the column to display in the dataset.

> **Note:**
>
> Hidden columns are available for table joins. Deleted columns are not available for table joins.

1. On the Home page, open a dataset or workbook.
   - Select the dataset, click the **Actions menu**, and then select **Open**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.
   - Select the workbook that contains the dataset, click the **Actions menu**, select **Open**, then click **Data**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.

   > **Note:**
   >
   > Oracle Analytics displays a Data Diagram for single-table datasets or a Join Diagram for multi-table datasets. For file-based data sources, Oracle Analytics displays the transform editor directly.
2. Hover over the column that you want to hide or delete.
3. Click **Options**, and select **Hide** or **Delete**.

To unhide columns, click **Hidden columns** in the bottom right hand corner of the transform editor and select a column, or click **Unhide All**.

Restore a Hidden or Deleted Column

In the Transform editor, you can restore any dataset columns that you hid or deleted.

If you don't see a column's **Delete Step** in the Preparation Script Panel, then the column might have been removed from the dataset. Use Edit Definition to confirm if the column has been removed instead of deleted. See Remove or Restore a Dataset Table's Columns.

> **Note:**
>
> Hidden columns are available for table joins. Deleted columns are not available for table joins.

1. On the Home page, open a dataset or workbook.
   - Select the dataset, click the **Actions menu**, and then select **Open**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.

- Select the workbook that contains the dataset, click the **Actions menu**, select **Open**, then click **Data**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.

> **Note:**
>
> Oracle Analytics displays a Data Diagram for single-table datasets or a Join Diagram for multi-table datasets. For file-based data sources, Oracle Analytics displays the transform editor directly.

2. Go to the Preparation Script Panel and locate the Hide Column or Delete Column step.
3. Click **Delete Step**.

   To unhide columns, click **Hidden columns** in the bottom right hand corner of the transform editor and select a column, or click **Unhide All**.

Edit the Data Preparation Script

You can edit the data transformation changes added to the Preparation Script. For example, you can edit, preview, or delete transformation steps before you apply them.

Both before and after you've applied recommended changes displayed in the Column Action Panel, you can edit the data transformation steps. The edit option isn't available for all types of transform steps.

The updates to the columns are applied only to the dataset and not to the visualization. To ensure that you see the most up-to-date data, on the Visualize canvas, click **Refresh Data**.

1. On the Home page, open a dataset or workbook.
   - Select the dataset, click the **Actions menu**, and then select **Open**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.
   - Select the workbook that contains the dataset, click the **Actions menu**, select **Open**, then click **Data**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.

> **Note:**
>
> Oracle Analytics displays a Data Diagram for single-table datasets or a Join Diagram for multi-table datasets. For file-based data sources, Oracle Analytics displays the transform editor directly.

2. Select a step in the Preparation Script pane and click **Edit Transform**.
3. Select a transform step in the Preparation Script pane and click the pencil icon or **Edit Transform**.

   If a step isn't editable, a message is displayed when you hover over the pencil icon that editing for the transform step has been disabled.

4. In the step editor, update the fields to edit the data transform changes that are applied to the columns.
5. Click **OK** to update the column and close the step editor.
6. To save your data preparation changes and apply them to your data, click **Save Dataset**.

   If you're working with a single table dataset or a file-based dataset, click **Apply Script**.

Add Columns to a Dataset

You can develop your datasets by adding columns. For example, you might create a new column that concatenates columns ADDRESS_LINE_1, ADDRESS_LINE_2, and ADDRESS_LINE_3.

1. On the Home page, select the dataset, click the **Actions menu**, and then select **Open**. In the Data Diagram or Join Diagram, right-click a data source and click **Open** to display the transform editor.
2. In the transform editor, click **Add Preparation Step**.
3. In **Create Column**, enter a **Name** for the column, and in the expression box define an expression to add values to the column. For example, to concatenate values in columns **Category** and **Sub-category**, you might specify: `CONCAT(Category, Sub Category)`.

   Use the **f(x)** function picker to build an expression using operators, mathematical, string, and conversion functions.
4. Click **Validate** and review the new column in the data preview.
5. Click **Add Step**.

   Oracle Analytics adds a step to the Preparation Script panel, ready to be applied when you click **Apply Script** or **Save Dataset**.

Enrich and Transform Reference

Use this reference information to help you enrich and transform your data.

Topics:

- Transform Recommendation Reference
- Data Profiles and Semantic Recommendations
- General Custom Format Strings
- Column Menu Options for Quick Data Transformations

Transform Recommendation Reference

Find out about the data transform options.

Option	Description
Edit	Edits the column. For example, you can change the name, select another column, or update functions.
Hide	Hides the column in the Data Panel and in the visualizations. If you want to see the hidden columns, click **Hidden columns** (ghost icon) on the page footer. You can then unhide individual columns or unhide all the hidden columns at the same time.
Group, Conditional Group	Select **Group** to create your own custom groups. For example, you can group States together with custom regions, and you can categorize dollar amounts into groups indicating small, medium, and large.
Split	Splits a specific column value into parts. For example, you can split a column called, Name, into first and last name.
Uppercase	Updates the contents of a column with the values in all uppercase letters.
Lowercase	Updates the contents of a column with the values all in lowercase letters.
Sentence Case	Updates the contents of a column to make the first letter of the first word of a sentence uppercase.
Rename	Allows you to change the name of any column.
Duplicate	Creates a column with identical content of the selected column.
Convert to Text	Changes the data type of a column to text.
Replace	Changes specific text in the selected column to any value that you specify. For example, you can change all instances of *Mister* to *Mr.* in the column.
Create	Creates a column based on a function.
Convert to Number	Changes the data type of the column to number, which deletes any values that aren't numbers from the column.
Convert to Date	Changes the data type of the column to date and deletes any values that aren’t dates from the column.
Bin	Creates your own custom groups for number ranges. For example, you can create bins for an Age column with age ranges binned into Pre-Teen, Young Adult, Adult, or Senior based on custom requirements.
Log	Calculates the natural logarithm of an expression.
Power	Raises the values of a column to the power that you specify. The default power is 2.
Square Root	Creates a column populated with the square root of the value in the column selected.

Data Profiles and Semantic Recommendations

After creating a dataset, the dataset undergoes column-level profiling to produce a set of semantic recommendations to repair or enrich your data. These recommendations are based on the system automatically detecting a specific semantic type during the profile step.

There are various categories of semantic types such as geographic locations identified by city names, a specific pattern such as a credit card number or email address, a specific data type such as a date, or a recurring pattern in the data such as a hyphenated phrase. You can also create your own custom semantic types.

Topics:

Semantic Type Categories

Profiling is applied to various semantic types.

Semantic type categories are profiled to identify:

- Geographic locations such as city names.
- Patterns such as those found with credit cards numbers or email addresses.
- Recurring patterns such as hyphenated phrase data.

Semantic Type Recommendations

Recommendations to repair, enhance, or enrich the dataset, are determined by the type of data.

Examples of semantic type recommendations:

- **Enrichments** - Adding a new column to your data that corresponds to a specific detected type, such as a geographic location. For example, adding population data for a city.
- **Column Concatenations** - When two columns are detected in the dataset, one containing first names and the other containing last names, the system recommends concatenating the names into a single column. For example, a *first_name_last_name* column.
- **Semantic Extractions** - When a semantic type is composed of subtypes, for example a *us_phone* number that includes an area code, the system recommends extracting the subtype into its own column.
- **Part Extraction** - When a generic pattern separator is detected in the data, the system recommends extracting parts of that pattern. For example if the system detects a repeating hyphenation in the data, it recommends extracting the parts into separate columns to potentially make the data more useful for analysis.
- **Date Extractions** - When dates are detected, the system recommends extracting parts of the date that might augment the analysis of the data. For example, you might extract the day of week from an invoice or purchase date.
- **Full and Partial Obfuscation/Masking/Delete** - When sensitive fields are detected such as a credit card number, the system recommends a full or partial masking of the column, or even removal.

Recognized Pattern-Based Semantic Types

Semantic types are identified based on patterns found in the data.

Recommendations are provided for these semantic types:

- Dates (in more than 30 formats)
- US Social Security Numbers (SSN)
- Credit Card Numbers
- Credit Card Attributes (CVV and Expiration Date)
- Email Addresses
- North American Plan Phone Numbers
- First Names (typical first names in the United States)
- Last Names (typical surnames in the United States)
- US Addresses

Reference-Based Semantic Types

Recognition of semantic types is determined by loaded reference knowledge provided with the service.

Reference-based recommendations are provided for these semantic types:

- Country names
- Country codes
- State names (Provinces)
- State codes
- County names (Jurisdictions)
- City names (Localized Names)
- Zip codes

Recommended Enrichments

Recommended enrichments are based on the semantic types.

Enrichments are determined based on the geographic location hierarchy:

- Country
- Province (State)
- Jurisdiction (County)
- Longitude
- Latitude
- Population
- Elevation (in Meters)
- Time zone

- ISO country codes
- Federal Information Processing Series (FIPS)
- Country name
- Capital
- Continent
- GeoNames ID
- Languages spoken
- Phone country code
- Postal code format
- Postal code pattern
- Phone country code
- Currency name
- Currency abbreviation
- Geographic top-level domain (GeoLTD)
- Square KM

Required Thresholds

The profiling process uses specific thresholds to make decisions about specific semantic types.

As a general rule, 85% of the data values in the column must meet the criteria for a single semantic type in order for the system to make the classification determination. As a result, a column that might contain 70% first names and 30% “other”, doesn't meet the threshold requirements and therefore no recommendations are made.

Custom Knowledge Recommendations

Use custom knowledge recommendations to augment the Oracle Analytics system knowledge. Custom knowledge enables the Oracle Analytics semantic profiler to identify more business-specific semantic types and make more relevant and governed enrichment recommendations. For example, you might add a custom knowledge reference that classifies prescription medication into USP drug categories Analgesics or Opioid.

Tutorial

You can use existing semantic files such as Unsupervised Semantic Parsing (USP) files, or you can create your own semantic files. Ask your administrator to upload custom knowledge files to Oracle Analytics. When content authors enrich datasets, Oracle Analytics presents enrichment recommendations based on this semantic data.

Creating Your Own Custom Knowledge Files

When you create your own semantic files, follow these guidelines:

- Create a data file in CSV or Microsoft Excel (XLSX) format.

- Populate the first column with the key, which Oracle Analytics uses to profile the data.
- Populate the other columns with the enrichment values.

Ask your administrator to upload your custom knowledge file to Oracle Analytics.

For an example semantic file, refer to the sample file provided with the tutorial.

General Custom Format Strings

You can use these strings to create custom time or date formats.

The table shows the general custom format strings and the results that they display. These allow the display of date and time fields in the user's locale.

General Format String	Result
[FMT:dateShort]	Formats the date in the locale's short date format. You can also type `[FMT:date]`.
[FMT:dateLong]	Formats the date in the locale's long date format.
[FMT:dateInput]	Formats the date in a format acceptable for input back into the system.
[FMT:time]	Formats the time in the locale's time format.
[FMT:timeHourMin]	Formats the time in the locale's time format but omits the seconds.
[FMT:timeInput]	Formats the time in a format acceptable for input back into the system.
[FMT:timeInputHourMin]	Formats the time in a format acceptable for input back into the system, but omits the seconds.
[FMT:timeStampShort]	Equivalent to typing `[FMT:dateShort] [FMT:time]`. Formats the date in the locale's short date format and the time in the locale's time format. You can also type `[FMT:timeStamp]`.
[FMT:timeStampLong]	Equivalent to typing `[FMT:dateLong] [FMT:time]`. Formats the date in the locale's long date format and the time in the locale's time format.
[FMT:timeStampInput]	Equivalent to `[FMT:dateInput] [FMT:timeInput]`. Formats the date and the time in a format acceptable for input back into the system.
[FMT:timeHour]	Formats the hour field only in the locale's format, such as 8 PM.
YY or yy	Displays the last two digits of the year, for example `11` for 2011.
YYY or yyy	Displays the last three digits of the year, for example, `011` for 2011.
YYYY or yyyy	Displays the four-digit year, for example, `2011`.
M	Displays the numeric month, for example, `2` for February.
MM	Displays the numeric month, padded to the left with zero for single-digit months, for example, `02` for February.
MMM	Displays the abbreviated name of the month in the user's locale, for example, `Feb`.
MMMM	Displays the full name of the month in the user's locale, for example, `February`.
D or d	Displays the day of the month, for example, `1`.
DD or dd	Displays the day of the month, padded to the left with zero for single-digit days, for example, `01`.
DDD or ddd	Displays the abbreviated name of the day of the week in the user's locale, for example, `Thu` for Thursday.

General Format String	Result
DDDD or dddd	Displays the full name of the day of the week in the user's locale, for example, `Thursday`.
DDDDD or ddddd	Displays the first letter of the name of the day of the week in the user's locale, for example, `T` for Thursday.
r	Displays the day of year, for example, 1.
rr	Displays the day of year, padded to the left with zero for single-digit day of year, for example, 01.
rrr	Displays the day of year, padded to the left with zero for single-digit day of year, for example, 001.
w	Displays the week of year, for example, 1.
ww	Displays the week of year, padded to the left with zero for single-digit weeks, for example, 01.
q	Displays the quarter of year, for example, 4.
h	Displays the hour in 12-hour time, for example `2`.
H	Displays the hour in 24-hour time, for example, `23`.
hh	Displays the hour in 12-hour time, padded to the left with zero for single-digit hours, for example, `01`.
HH	Displays the hour in 24-hour time, padded to the left with zero for single digit hours, for example, `23`.
m	Displays the minute, for example, `7`.
mm	Displays the minute, padded to the left with zero for single-digit minutes, for example, `07`.
s	Displays the second, for example, `2`. You can also include decimals in the string, such as s.# or s.00 (where # means an optional digit, and 0 means a required digit).
ss	Displays the second, padded to the left with zero for single-digit seconds, for example, `02`. You can also include decimals in the string, such as ss.# or ss.00 (where # means an optional digit, and 0 means a required digit).
S	Displays the millisecond, for example, 2.
SS	Displays the millisecond, padded to the left with zero for single-digit milliseconds, for example, 02.
SSS	Displays the millisecond, padded to the left with zero for single-digit milliseconds, for example, 002.
tt	Displays the abbreviation for ante meridiem or post meridiem in the user's locale, for example, `pm`.
gg	Displays the era in the user's locale.

5

Curate Your Data Using Data Flows

You can use data flows to produce curated (combined, organized, and integrated) datasets.

Video

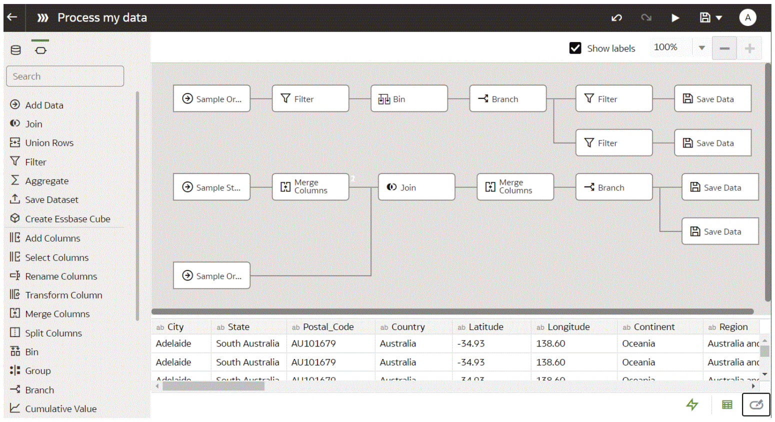
Topics:

- About Data Flows
- Prepare Data by Running Data Flows
- Create a Data Flow
- Configure Incremental Processing in a Data Flow
- Customize the Names and Descriptions of Data Flow Steps
- Run a Data Flow
- Save Output Data from a Data Flow
- Reuse a Data Flow
- Modify Parameter Prompts When You Run a Data Flow
- Create a Sequence of Data Flows
- Manage Your Data Flows
- Data Flow Steps Reference

About Data Flows

Data flows enable you to organize and integrate your data to produce a curated dataset that your users can analyze.

For example, you might merge two datasets containing order data, strip out columns you don't need, aggregate the value of orders, and save the results in a new dataset named `Order Summary`.



ab City	ab State	ab Postal_Code	ab Country	ab Latitude	ab Longitude	ab Continent	ab Region
Adelaide	South Australia	AU101679	Australia	-34.93	138.60	Oceania	Australia anc
Adelaide	South Australia	AU101679	Australia	-34.93	138.60	Oceania	Australia anc
Adelaide	South Australia	AU101679	Australia	-34.93	138.60	Oceania	Australia anc

To build a data flow, you add steps. Each step performs a specific function, for example, add data, join tables, merge columns, transform data, save your data. Use the data flow editor to add and configure your steps. Each step is validated when you add or change it. When you've configured your data flow, you execute it to produce a dataset.

When you add your own columns or transform data, you can use a wide range of SQL operators (for example, BETWEEN, LIKE, IN), conditional expressions (for example, CASE), and functions (for example, Avg, Median, Percentile).

Working in the Data Flow Editor

You build data flows in the data flow editor.

Working in the data flow editor:

Name	Icon	What it does
Data Flow Steps	⬭	Display the steps pane, where you can drag and drop steps onto the data flow editor.
Data pane		Display the data pane, where you can drag and drop data columns onto the data flow editor.
Steps editor	✎	Hide or display the Step editor pane by clicking the **Step editor** icon at the bottom right-hand corner of the data flow editor.
Preview data	▦	Hide or display the Preview data columns pane by clicking **Preview data** at the bottom right-hand corner of the data flow editor. This pane updates automatically when you make changes to the data flow. You can specify whether to automatically refresh step changes in the Preview data pane by clicking **Auto apply**.
Show available functions/hide available functions	f(x)	Display or hide the expression pick list. This icon is only displayed for steps that enable you to build your own expressions, for example, the "Add Columns" step or "Transform Column" step.

Prepare Data by Running Data Flows

Run (that is, execute) data flows to prepare data so that it's ready to visualize.

Data flows can take a long time to run, so you can perform other tasks in Oracle Analytics, and then come back to the Data Flows pane to check the status of a data flow.

You can cancel a long-running data flow. To do so, go to the Data Flows section, click the data flow’s **Action menu** and select **Cancel**.

If it’s the first time you’re running a data flow, then a new dataset is created, and you can find it in the Datasets section of the Data page. The dataset contains the name that you specify on the data flow’s **Save Data** step. If you’ve run the data flow before, then the resulting data source already exists, and its data is refreshed.

If your data flow is writing to a database, set the database’s query mode to **Live**. Setting the query mode to **Live** allows the data flow to access data from the database (versus the data cache) and pushes any expensive operations such as joins to the database.

Create a Data Flow

Create a data flow to curate data from one or more datasets. For example, you might merge two datasets, cleanse the data, and output the results to a new dataset that you can make available to analysts.

1. On the Home or Data page, click **Create** and select **Data Flow**.
2. In the Add Dataset dialog, select a dataset.

   You can add more data sources at any time by clicking **Add Step (+)**, then clicking **Add Data**.
3. In the Add Data pane, configure your data. For example, you can:
   - Replace the selected dataset (click the data source name next to **Add Data -**).
   - Include or exclude columns.
   - Rename columns.
   - Prompt for a data source when the data flow is executed (using the **When Run Prompt to select Dataset** option).
4. Build your data flow:
   - For each function that you want to perform, click **Add a step (+)**, click the step type you want, then specify the properties in the Step editor pane.
     Tip: Hover over the last step to display the **Add a step (+)** option. You can also edit your flow and add steps using **Options** in the Column header. For example, you can rename, reformat, merge, or transform columns.
   - To remove a step, hover over the step and click **X** or right-click the step and click **Delete**. If you've invalidated part of the data flow by deleting a step you'll see a red X icon over the offending step.
   - To undo or redo an edit in a data flow that you haven't saved, go to the workflow diagram panel toolbar and click **Undo Last Edit** or **Redo Last Edit**.
   - At the end of your data flow, add a **Save Data** step and specify a meaningful name.
5. Save your data flow.

   You can start the data flow now using the **Run Data Flow** option or later using the **Data Flows** panel on the Data page. If you run it now you can access the generated dataset on the **Datasets** panel on the Data page.
6. Run your data flow.
   - To start the data flow now, click **Run Data Flow**.
   - To start the data flow later, navigate to the **Data Flows** panel on the Data page, right click the data flow and click **Run**.

   When the data flow completes, navigate to the **Datasets** panel on the Data page to access the dataset generated.

Run a Data Flow

Run (that is, execute) a data flow to generate or update a dataset that you can visualize.

To run a data flow from the data flow editor:

1. On the Home page, click **Navigator**, then **Data**, then **Data Flows**.
2. Run the data flow:
    - Right-click the data flow and select **Run** (or click **New schedule** to run later).
    - Alternatively, open the data flow, and in the data flow editor click **Run Data Flow**.

If your data flow creates or updates a dataset, to locate the dataset, go to the **Data** page and click **Datasets**.
If you've arranged multiple data-flows into a sequence, on the **Data** page click **Sequence**, and run or schedule the sequence.

Run a Saved Data Flow

You can run a saved data flow to create a new dataset or to refresh existing data.

1. In the Data page, go to the Data Flows section, and locate the data flow that you want to run.
2. Right-click the data flow and select **Run** now (or click **New schedule** to run later).
    - Complex data flows take longer to run. While the data flow is running, you can go to and use other parts of the application, and then come back to the Data Flows pane to check the status of the data flow.
    - You can cancel a long-running data flow. To do so, go to the Data Flows section, click the data flow's **Action menu** and select **Cancel**.
    - If it's the first time you're running a data flow, then a new dataset is created, and you can find it in the Datasets section of the Data page. The dataset contains the name that you specify on the data flow's **Save Data** step. If you've run the data flow before, then the resulting data source already exists, and its data is refreshed.
    - When creating a new database data source, set the database's query mode to **Live**. Setting the query mode to **Live** allows the data flow to access data from the database (versus the data cache) and pushes any expensive operations such as joins to the database.

Reuse a Data Flow

In a data flow, you can add parameters so you can reuse the data flow with a different source dataset or use different criteria to process and select data.

Parameters help you identify the type of data appropriate for the data flow and if you want to select an alternative dataset when running or scheduling the data flow. You can also apply parameters to modify default values when creating an Essbase cube.

For example, using a parameter you can:

- Process a new dataset that has the same format as the default input dataset.
- Process and store different aspects of a large dataset based on date range, individual departments, or regions into alternative target datasets.

1. On the Home page, click **Navigator**, then **Data**, then **Data Flows**.
2. Open your data flow.
3. In the Step editor pane, apply parameters to your steps.
   In an **Add Data** step, select the **When Run Prompt to select Dataset** option then provide the **Name** and **Prompt** values for the parameter.

   In a **Save Data** step, select the **When Run Prompt to specify Dataset** option, then provide the **Name** and **Prompt** values for the parameter.

   In a **Create Essbase Cube** step, select the **When Run Prompt to specify Dataset** option, then provide the **Cube** name, **Application** name, and **Prompt** value for the parameter.

Configure Incremental Processing in a Data Flow

Configure incremental processing to load only new or updated records from a database. You can deploy incremental processing if your dataset is sourced from a database (using a database connection).

Before you start, create a connection to one of the supported target databases, for example Oracle, Oracle Autonomous Data Warehouse, Apache Hive, Hortonworks Hive, or Map R Hive.

1. Specify an incremental identifier for the source dataset.
   a. On the home page, click **Navigator**, then click **Data**.
   b. Select a dataset, click the **Actions menu** or right-click, then select **Open**.
   c. In the **Join Diagram**, double-click the table that includes the incremental identifier you'd like to use.
   d. Click **Edit Definition** on the toolbar.
   e. From the **New Data Indicator** list, select a column to detect when new data is added.
   f. Click **OK**.
2. Apply incremental processing to the output data connection used in the data flow.
   a. Create or open the data flow in which you want to apply incremental processing.
   b. In the Data Flow editor select the **Save Data** step to display the Step editor pane.
   c. In the **Dataset** field, specify the name of the input dataset (the dataset specified in the **Add Data** step).
   d. At the **Save data to** option select **Database Connection**.
   e. Click **Select Connection** and select a connection to one of the supported target databases.
   f. In the **Data** field, specify the name of the target table that you're writing to.
   g. in the **When run** option, select **Add new data to existing data**.
   h. Click **Save**.

   You're now ready to schedule your data flow to load new data regularly.

Modify Parameter Prompts When You Run a Data Flow

Parameter prompts are displayed before the job runs, when you run a data flow with parameter prompts. Prompts allow you to review the default values or settings and to select or define an alternate value or setting.

1. On the Home page, click **Navigator** then click **Data**, then click **Data Flows**.
2. Select the data flow with parameter prompts that you want to run.
3. Click the data flow’s **Actions menu** or right-click and select **Run**.
4. In the Data Flow Prompt dialog, either use the default values or define alternate values.
   - In the Sources section, click the default Target - existing dataset name, then select a new source dataset in the Add Dataset dialog. Click **Add**.
   - In the Targets section, do one of the following:
     - Change the default Target - existing dataset name.
     - For a data flow with **Create Essbase Cube** step, change the default Target - Application and Target - Cube names.
5. Click **OK**.

Customize the Names and Descriptions of Data Flow Steps

You can change the names of steps to make flows easier to understand, and annotate data flows by adding your own descriptions.

1. On the Home page, click **Navigator** then click **Data**, then click **Data Flows**.
2. Open a data flow and click the step you want to customize.

   Notice that the step name is displayed in the panel below the step diagram.
3. In the panel below the step diagram, click the step name (for example, **Merge Columns**).
4. Use the **Name** and **Description** fields to change the default values.

   The new name that you specify is appended to the default name.
5. To save your changes, press **Enter**.

Create a Sequence of Data Flows

A sequence is collection of data flows that you run together. They're useful when you want to run multiple data flows as a single transaction. If any flow within a sequence fails, then all the changes made in the sequence are rolled back.

You can also include other saved sequences inside a sequence.

Video

1. On the Home page click **Create** and select **Sequence**.
2. Drag and drop the data flows and sequences to the Sequence pane.

3. Click the menu icon to move an item up or down in the list, and to remove an item.
4. Click **Save**. When you save a sequence, it’s displayed in the Sequence area of the Data page.
5. Go to the Sequence area of the Data page, select the sequence, and click **Execute Sequence**.

   You can also schedule a sequence to execute at a particular date and time.
6. To schedule a sequence to execute at a particular date and time:
   1. From the Home page, click the **Navigator**, and click **Data**, then **Sequences**.
   2. Right-click a sequence, click **New Schedule**, and complete the information on the Schedule page.

      Note: To manage scheduled sequences, from the Home page, click the **Navigator**, then **Jobs**, then **Schedules**.
7. Go to the Data page and click **Datasets** to see the list of resulting datasets.

Manage Your Data Flows

Manage your data flows on the Data Flows page.

1. On the Home page, click **Navigator**, then **Data**, then **Data Flows**.
2. Hover over a data flow and use the **Actions menu** to access the options described in the table below.
   - Use **Run** to execute the data flow to prepare your data.
   - Use **Open/Open in a New Tab** to edit the data flow.
   - Use **New schedule** to execute the data flow regularly.
   - Use **Inspect** to view general information about the data flow, such as the source and target data, when the flow was last executed, scheduled executions, and execution history.
   - Use **Export** to export a data flow with its dependent data and credentials as a .DVA file to the download folder of your machine. Use the export and import options to migrate data flows from one system to another or back up your data flows. When you export data flows, you can select what to include in the export file. For example, enable **Include Permissions** to include access permissions so that shared connections continue to work when you import the data flow. To import a data flow that you've downloaded, on the Data Flows page, click **Page Menu**, then click **Import Workbook/Flow**. Follow the on-screen instructions to select a local .DVA file to import. When you import the data flow, select **Import Permissions (if available)** to include access permissions so that shared connections continue to work.
   - Use **Delete** to remove the data flow from your system.

Data Flow Steps Reference

You build data flows using steps to curate your data. Steps are functions that change your data in a specific way. For example, steps can aggregate values, perform time series analysis, or perform machine learning algorithms.

Step	Use this step to:	More Information
Add Columns	Add a new output data column to your data flow using a wide range of functions, conditional expressions, and SQL operators.	Add Columns in a Data Flow
Add Data	Add a data source to your data flow.	Add Data in a Data Flow
Aggregate	Apply aggregate functions to group data in a data flow.	Add Aggregates to a Data Flow
Analyze Sentiment	Detect sentiment for a text column by applying a sentiment analysis to the data flow.	Add a Sentiment Analysis to a Data Flow
Apply Model	Apply a machine learning model to your data (also known as scoring a data model).	Apply a Predictive or Registered Oracle Machine Learning Model to a Dataset
Bin	Assign your data values into categories, such as high, low, or medium.	Create a Bin Column in a Data Flow
Branch	Creates multiple outputs from a data flow using a branch.	Create Multiple Pipelines in a Data Flow Using a Branch
Create Essbase Cube	Create an Essbase cube from a dataset.	Create and Customize an Essbase Cube in a Data Flow
Cumulative Value	Group data by applying cumulative aggregate functions in a data flow.	Add Cumulative Values to a Data Flow
Database Analytics	Use advanced analytic functions, such as anomaly detection, unpivot, sampling, and advanced clustering (Requires Oracle Database or Oracle Autonomous Data Warehouse).	Add Database Analytics to a Data Flow
Filters	Use filters to limit the data in a data flow output.	Filter Your Data in a Data Flow
Graph Analytics	Perform geo-spatial analysis, such as calculating the distance or the number of hops between two vertices.	Add Graph Analytics to a Data Flow
Group	Create a group column of attribute values in a dataset.	Create a Group in a Data Flow
Join	Join multiple tables or datasets.	Add a Join in a Data Flow
Merge Columns	Combine two or more columns in your data flow.	Merge Columns in a Data Flow
Merge Rows	Combine two or more rows in your data flow.	Merge Rows in a Data Flow
Rename Columns	Change the name of data columns to something more meaningful.	Rename Columns in a Data Flow
Save Data	Before running a data flow, modify or select the database name, attribute or measure, and aggregation rules for each columns of the output dataset.	Save Output Data from a Data Flow
Save Model	Change the default model name (untitled) and provide a description.	Save Model
Select Columns	Specify which data columns to include in your data flow.	Select Columns to Include in a Data Flow
Split Columns	Extract useful data from within data columns.	Split Columns in a Data Flow
Time Series Forecast	Apply a time series forecast calculation to a dataset to create additional rows.	Add a Time Series Forecast to a Data Flow

Step	Use this step to:	More Information
Train Binary-Classifier	Train a machine learning model to classify your data into one of two predefined categories.	Train a Binary Classifier Model in a Data Flow
Train Clustering	Train a machine learning model to segregate groups with similar traits and assign them into clusters.	Train a Clustering Model in a Data Flow
Train Multi-Classifier	Train a machine learning model to classify your data into three or more predefined categories.	Train a Multi-Classifier Model in a Data Flow
Train Numeric Prediction	Train a machine learning model to predict a numeric value based on known data values.	Train a Numeric Prediction Model in a Data Flow
Transform Column	Modify data in a column using a wide range of functions, conditional expressions, and SQL operators.	Transform Data in a Data Flow

Add Columns in a Data Flow

You can add columns to your target data and customize the format. For example, you might calculate the value of your stock by multiplying the number of units in a `UNITS` column by the sale price in a `RETAIL_PRICE` column.

Use the **Add Columns** step in the data flow editor.

1. In an open data flow, click **Add a step (+)**, and select **Add Columns**.
2. In the Add Columns pane, use the expression builder to define your column. For example, to calculate the value of stock items you might specify `UNITS * RETAIL_PRICE`.

   Select SQL operators, functions, and conditional expressions from the expression pick list.

Add Data in a Data Flow

When you create a new data flow and select a dataset, you'll see a step with the name of your dataset. You can add additional data from multiple data sources to your data flow.

Use the **Add Data** step in the data flow editor.

Note: Oracle Essbase and EPM Cloud datasets cannot be used in the **Add Data** step as inputs to Data Flows.

1. In an open data flow, use the options on the Add Data pane to configure the dataset. For example, change the default name, or include and exclude columns.
2. To add another dataset to your flow, click **Add a step (+)**, and select **Add Data**.

   If matching columns are found in datasets, a Join step is automatically added to enable you to define the relationship between the datasets. For example, you might want to combine rows from two datasets where the `CustomerID` in the first dataset matches the `CustomerID` in the second dataset.
3. If you don't get a Join step automatically, click **Add a step (+)**, and select **Join**.

   To complete the join, on the data flow diagram click the circle on the dotted line between the data source step and the Join step. Then use the Join pane to configure the relationship between the datasets.

4. Click your dataset step again and use the options on the Add Data pane to configure the dataset.
    - Use **Add Data - <Data source name>** to edit the step name and description.
    - Use **Select...** to change the dataset or data source. Changing the dataset or data source might break other steps in your flow.
    - Use **When Run Prompt to select Dataset** to specify the name of the output dataset when the data flow is executed. For example, you might want to specify a different name for the output dataset each time the flow is executed.
    - Use **Remove Duplicates** to remove duplicate rows based on the selected column. For example, you might select this option for City if you want only one row for each city.

Add Aggregates to a Data Flow

Create group totals by applying aggregate functions such as count, sum, and average.

Use the **Aggregate** step in the data flow editor.

1. In an open data flow, click **Add a step (+)** and select **Aggregate**.

    In the Aggregate pane you'll see a suggested aggregate column for each numeric column.
2. Use the options on the Aggregate pane to configure your aggregate.
    - Use **Aggregate** to select a column you want to aggregate.
    - Use **Function** to select an aggregate function such as Sum, Average, Minimum, or Count to apply to the selected column.
    - Use **New column name** to change the default name of the aggreate column.
3. Add or remove aggregates.
    - To remove an aggregate, select the aggregate and click **X**.
    - To see the **Add Aggregate** option, scroll to the bottom of the Aggregate pane.

Add a Sentiment Analysis to a Data Flow

You can detect sentiment for a given text column by applying a sentiment analysis to your data flow. For example, you might analyze customer feedback to determine whether it's positive or negative.

Sentiment analysis evaluates text based on words and phrases that indicate a positive, neutral, or negative emotion. Based on the outcome of the analysis, a new column contains a Positive, Neutral, or Negative string type result.

Use the **Analyze Sentiment** step in the data flow editor.

1. In an open data flow, click **Add a step (+)**, and select **Analyze Sentiment**.
2. In the Analyze Sentiment pane and Output section, specify an output column for the emotion result value.
3. Optional: Change the default column name 'emotion'.
4. In the Analyze Sentiment pane and Parameters section, specify the value for **Text to Analyze**.

Select a text column with natural language content to analyze.

Create a Bin Column in a Data Flow

Use a bin to categorize your data by creating a new column based on the value of a measure. For example, you might categorize values for `RISK` into three bins for low, medium, and high.

Use the **Bin** step in the data flow editor.

1. In an open data flow, click **Add a step (+)**, and select **Bin**.

   You also create bins when you add columns using the **Add Column** step.

2. Select the column whose values you want to categorize.
3. Use the options on the Bin pane to configure your bin.
   - The **Bin** option displays the column that you selected in Step 2. To categorize values in a different column, click the column name and select a different column.
   - In **Histogram View**, based on the **Method** selected, you'll see the histogram range (width) and histogram count (height) of the bins.
   - Use **List View** to change the name of the bins and define the range for each bin (if you select the **Manual** option).
   - Use **Method** to specify how the data boundaries are calculated.
   - – In the **Manual** method, the range is divided by the number of bins.
     – In the **Equal Width** method, the histogram range is divided into intervals of the same size. For equal width binning, the column values are measured, and the range is divided into equal-sized intervals. The edge bins can accommodate very low or very high values in the column.
     – In the **Equal Height** method, the height of each bin is same or very slightly different but the histogram range is equal. For equal height or frequency binning, the intervals of each bin is based on each interval containing approximately the equal number of elements (that is, records). **Equal Height** method is preferred specifically for the skewed data.

Based on your changes, the data preview (for example, the bin column name) is updated.

Create Multiple Pipelines in a Data Flow Using a Branch

Creates multiple outputs from a data flow using a branch. For example, if you have sales transactions data based on country, you might save data for United States in the first branch and data for Canada in the second branch.

Use the **Branch** step in the data flow editor.

1. In an open data flow, click **Add a step (+)** and select **Branch**.

   You'll see a **Branch** step and two **Save Data** steps added to the data flow. Select the **Branch** step and use the **Branch into** option to add or remove branches. The minimum number of branches is two, and the maximum is five.

2. To configure each branch, click connection line between the **Branch** step and the **Save Data** step, click **Add a step (+)** and select a step type that processes your branch.

   For example, you might add a **Filter** to the first branch that saves data from United States, and add a **Filter** to the second branch that saves data from Canada. Or, you might use the **Split Columns** step to save some columns in the first branch and other columns in the second branch.

3. Click each **Save Data** step and in the Save Dataset pane specify the properties for saving the output datasets.

Create and Customize an Essbase Cube in a Data Flow

Create an Essbase cube from a spreadsheet or database.

You can create Essbase cubes only for Oracle Analytics Cloud – Essbase. When selecting an Essbase connection for creating a cube, you might see remote connections to on-premises Oracle Essbase instances. You can't create a cube from data in on-premises Oracle Essbase instances.

Use the **Create Essbase Cube** step in the data flow editor.

1. In an open data flow, click **Add a step (+)**, and select **Create Essbase Cube**.
2. In the Create Essbase Cube pane, specify the values for creating the cube such as connection and application name.
3. To configure the input columns, do the following:
   a. Move the slider to enable the **Customize Cube** option.
   b. Select the number of rows you want to analyze and click **Configure**.
   c. Decide how to process each column in the Dimensions, Measure, and Skip sections:
      - Cut
      - Paste as Sibling
      - Paste as Child
      - Skip
      - Delete
   d. Change the following column values:
      - Column name in the Data Elements column.
      - Designation type in the Treat As column.
4. Select the **When Run Prompt to specify Dataset** option to apply parameters to change the default values when creating the Essbase cube.

Cut, Paste, and Skip Rules

The cut, paste, and skip actions you perform for each column follow pre-configured rules.

- When you skip a column, it moves to the Skip section of the table. You can only paste a column as a sibling of the Skip header, or as a sibling of any skipped column.
- Any columns that are pasted as a Measure follow the rule of the paste command. Measure hierarchies are allowed, but the designation type doesn't change.
- **Paste as Child** action for Dimension columns:

  - When a column is pasted as a child of the Dimensions header, the cut column is pasted as a Dimension.
  - When a column is pasted as a child of the Dimension column:
    * The cut column is pasted as a Generation.
    * If the Dimension column already has a Generation child, the existing Generation (and its children) becomes the children of the new Generation column.
  - When a column is pasted as a child of the Generation column:
    * The cut column is pasted as a child of the Generation if the cut column is an Alias, Attribute, or UDA.
    * The cut column is pasted as a Generation if the cut column isn’t an Alias, Attribute, or UDA.
  - **Paste as Child** for any Dimension column isn't allowed if the target is an Alias, Attribute, or UDA.
- **Paste as Sibling** action for Dimension columns:
  - When a column is pasted as a sibling of a Dimension column, it's pasted as a Dimension.
  - When a column is pasted as a sibling of an Attribute, Alias, or UDA and it isn’t an Alias, Attribute, or UDA, the column is pasted as an Attribute.
  - **Paste as Sibling** for any Dimension column isn't allowed if the target is a Generation.

Designation Change Rules for Generation Columns

The Generation columns follow specific pre-configured rules when you change their designation type.

- **Generation** to **Attribute**/**Alias**/**UDA** - If the Generation column has any children, they move up a level and become children of the Generation column’s parent.
- **Attribute**/**Alias**/**UDA** to **Generation** - If the new Generation column has a sibling Generation column, the existing Generation column (and its children) become children of the new Generation column.

Add Cumulative Values to a Data Flow

You can calculate cumulative totals such as moving aggregate or running aggregate.

Use the **Cumulative Value** step in the data flow editor.

1. In an open data flow, click **Add a step (+)**, and select **Cumulative Value**.
2. Use the options on the Cumulative Value pane to configure your aggregate.

   For example, use the **Aggregate** option to select the data column to calculate, or use the **Function** option to select the cumulative function to apply.

Filter Your Data in a Data Flow

You use filters to limit the amount of data included in the data flow output. For example, you might create a filter to limit sales revenue data to the years 2020 through 2022.

Use the **Filter** step in the data flow editor.

1. In an open data flow, click **Add a step (+)**, and select **Filter**.
2. In the Filter pane, select the data element you want to filter.
   - Use **Add Filter (+)** to select the data element you want to filter, in the Available Data dialog. Alternatively, click **Data Elements** in the Data Panel, and drag and drop a data element to the Filter pane.
   - Use **Filter Fields** to change the values, data or selection of the filter (for example, maximum and minimum range). Based on the data element, specific filter fields are displayed. You can apply multiple filters to a data element.
   - Use **Filter menu icon** to select a function to clear the filter selection and disable or delete a filter.
   - Use **Filter pane menu icon** to select a function to clear all filter selections, remove all filters, and auto-apply filters. You can select to add an expression filter.
   - Use **Add Expression Filter** to select to add an Expression Filter. Click **f(x)**, select a function type, and then double-click to add a function in the **Expression** field. Then click **Apply**.
   - Use **Auto-Apply Filters** to select an auto-apply option for the filters, such as Default (On).

Create a Group in a Data Flow

You can categorize non-numeric data into groups that you define. For example, you might put orders for lines of business `Communication` and `Digital` into a group named `Technology`, and orders for `Games` and `Stream` into a group named `Entertainment`.

Use the **Group** step in the data flow editor.

1. In an open data flow, click **Add a step (+)**, and select **Group**.
2. For each group that you want to create, use the Group pane:
   a. Use the pop list of columns to select the column you'd like categorize. For example, to categorize orders by line of business, you might select LINE_OF_BUSINESS.
   b. Optional: Click the group name to change the default name `Group 1`. For example, you might change `Group 1` to `Technology`.
   c. Optional: In the **Name** field, change the default name of the new column from `new_name1` to a more meaningful name.
   d. In the center box, select one of more categories to add to the group. For example, to analyze line of business you might put `Communication` and `Digital` in a group named Technology.

In the Preview Data pane, you'll see a new column with the groups that you defined displayed as the value for each row. For example, values might be Technology or Entertainment.

3. To add more groups, click **Group (+)**.

Add a Join in a Data Flow

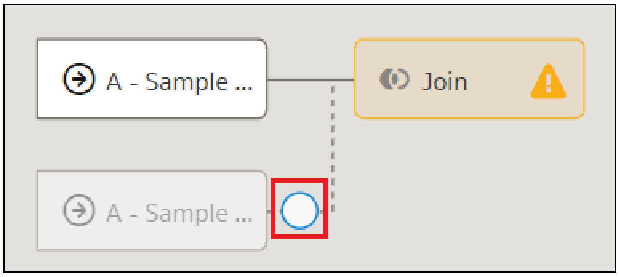
When you add data from multiple data sources to your data flow, you can join them on a common column. For example, you might join an `Orders` dataset to a `Customer_orders` dataset using a customer ID field.

When you use the **Add Data** step to add an extra data source, a **Join** step is automatically added to your data flow. But you can also manually add a **Join** step if you have more than one data source defined in your data flow.

Use the **Join** step in the data flow editor.

1. In an open data flow, add the data sources you'd like to join.
2. Select a data source, click **Add a step**, then click **Join**.

   You'll see a suggested connection with a node on the connection line.



3. Click the node on the connection line to complete the connection.
4. Use the options on the Join pane to configure your step.
   - Use **Keep rows** to specify how you want to join your data. Click an option to preview your merged data (if you're displaying the Data Preview pane).
   - Use **Match columns** to specify the common field on which you'd like to join the data sources.

Merge Columns in a Data Flow

You can combine multiple columns into a single column. For example, you might merge the street address, street name, state, and ZIP code columns so that they display as one item in visualizations.

Use the **Merge Columns** step in the data flow editor.

1. In an open data flow, click **Add a step (+)**, and select **Merge Columns**.
2. Use the options on the Merge Columns pane to configure your merge:
   - **(+) Column** field - Select more columns you want to merge.
   - **Delimiter** field - Select a delimiter to separate column names (for example, Space, Comma, Dot, or Custom Delimiter).

Rename Columns in a Data Flow

Rename columns to create more meaningful data column names in your generated datasets. For example, you might change CELL to Contact Cell Number.

Use the **Rename Columns** step in the data flow editor.

1. In an open data flow, click **Add a step (+)**, and select **Rename Columns**.
2. Use the **Rename** fields to specify a more meaningful name for columns in your generated dataset.

Save Output Data from a Data Flow

For the data created by a data flow you can change the default name and description, specify where to save the data, and specify runtime parameters.

If you're saving the output from your data flow to a database, before you start, create a connection to one of the supported database types. You can't save data to datasets that use remote connections (that is, via Data Gateway).

Use the **Save Data** step in the data flow editor.

1. In an open data flow, click **Add a step (+)** and select **Save Data**. Or, if you’ve already saved the data flow, then click the **Save Data** step.
2. In the Save Dataset pane, optionally change the default **Name** and add a **Description**.

   If you don't change the default **Name** value, you'll generate a dataset named 'untitled'. After you run this data flow, you'll see the generated dataset in the Datasets page (click **Data** from the navigator on the Home page).
3. Click **Save data to** and select a location:
   - Choose **Dataset Storage** to save the output data in a dataset in Oracle Analytics.
   - Choose **Database Connection** save the output data in one of the supported database types.
4. If you’ve selected **Database Connection**, specify details about the database connection.

   Before you start, create a connection to one of the supported database types.

   a. Click **Select connection** to display the Save Data to Database Connection dialog, and select a connection.

      You can save to a range of databases, including Oracle, Oracle Autonomous Data Warehouse, Apache Hive, Hortonworks Hive, and Map R Hive.

      To find out which databases you can write to, refer to the More Information column in Supported Data Sources.

   b. In the **Table** field, optionally change the default table name.

      The table name must conform to the naming conventions of the selected database. For example, the name of a table in an Oracle database can’t begin with numeric characters.

   c. In the **When run** field, specify whether you'd like to replace existing data or add new data to existing data.

5. Select the **When Run Prompt to specify Dataset** option if you want to specify the name of the output dataset or table at run time.
6. In the **Columns** table, change or select the database name, the attribute or measure, and the aggregation rules for each column in the output dataset:

When you run the data flow

- If you've selected dataset storage, go to the Data page and select Datasets to see your output dataset in the list.
  - Click **Actions menu** or right-click and select **Inspect**, to open the dataset dialog.
  - In the dataset dialog, click **Data Elements** and check the Treat As and Aggregation rules that you've selected for each column in the **Save Data** step.
- If you're saving output data to a database, go to the table in that database and inspect the output data.

Save Model

You can change the default name of your machine learning model and add a description.

Use the **Save Model** step in the data flow editor. You'll see a **Save Model** step added automatically in the data flow editor when you add one of the train model steps, for example, Train Numeric Prediction, or Train Binary Classifier.

1. In an open data flow, add one of the train model steps to your data flow. For example, Train Numeric Prediction, or Train Binary Classifier.

   Oracle Analytics automatically adds a **Save Model** step to the work flow.
2. Click the **Save Model** step.
3. Optional: In the Save Model pane, change the default **Model name**, and specify a **Model description** to identify the model type and script used.

   If you don't change the default **Model name** value, you'll save a model named 'New Dataset'. After you run this data flow, you'll see the new model in the Machine Learning page. On the Home page, click **Navigator**, then click **Machine Learning**.

Select Columns to Include in a Data Flow

Select which columns to include in your data flow. By default, all data columns are included in your data flow.

Use the **Select Columns** step in the data flow editor.

1. In an open data flow, click **Add a step (+)**, and select **Select Columns**.
2. Use the on-screen options to select or remove columns.

Split Columns in a Data Flow

You can strip out useful data from columns of concatenated data. For example, if a column contains `001011Black`, you might split this data into two separate columns, `001011` and `Black`.

Use the **Split Columns** step in the data flow editor.

Before you start, turn on Data Preview so that you can see the new columns as you configure the split. If your data source has many columns, use a **Select Columns** step to remove extraneous columns first to improve the preview.

1. In an open data flow, click **Add a step (+)**, and select **Split Columns**.
2. Use the options on the Split Columns panel to configure the data flow.
   - In the **Split Column** option, click **Select Column** to specify the data column you'd like to split. If a column is already chosen, click the column name to choose a different column.
   - Use **On** to specify whether to split the column by delimiter or by position. Select **Delimiter** if the column has separator characters, such as commas or spaces. Select **Position** if the column doesn't have separator characters. If you split on position, you can only create two new columns.
   - (Displayed when **On** is set to **Delimiter**) Use **Delimiter** to specify the separator used in your data column (for example, space, comma, custom).
   - (Displayed when **On** is set to **Position**) Use **Position** to specify where the second column starts. For example, if your column contains `AABBBCCCDDD`, specify 6 to put `AABBB` in the first column and `CCCDDD` in the second column.
   - Use **Number of parts to create** to specify the number of new columns to create when **On** is set to **Delimiter** (you can't change the default value 2 if **On** is set to **Position**). For example, if your source data column contains `AA BBBBB CCC DD`, you might select **4** to put each sub-string into a different column.
   - Use **Occurrence** to specify how many of the sub-strings in the source column to include in each new column. Examples based on data `AA BBBBB CCC DD` with **Delimiter** set to **Space**:
   - – If you set **Occurrence** to **1**, **Number of parts to create** to **1**, the new column contains `AA`. If you set the **Occurrence** to **2**, the new column contains `AA BBBBB`.
     - If you set **Occurrence** to **1**, **Number of parts to create** to **2**, the first new column contains `AA` and the second new column contains `BBBBB CCC DD`.
     - If you set **Occurrence** to **1**, **Number of parts to create** to **4**, the first new column contains `AA`, the second new column contains `BBBBB`, the third new column contains `CCC`, and the fourth new column contains `DD`.
   - Use **New column <number>** name to change the default name of new columns to a more meaningful name. Use the adjacent check box to display or hide new columns.

Add a Time Series Forecast to a Data Flow

You can calculate forecasted values by applying a Time Series Forecast calculation.

A forecast takes a time column and a target column from a given dataset and calculates forecasted values for the target column and puts the values in a new column. All additional columns are used to create groups. For example, if an additional column 'Department' with values 'Sales', 'Finance', and 'IT' is present, the forecasted values of the target column are based on the past values of the given group. Multiple columns with diverse values lead to a large number of groups that affect the precision of the forecast. Select only columns that are relevant to the grouping of the forecast.

Use a **Time Series Forecast** step in the data flow editor.

1. In an open data flow, click **Add a step (+)**, and select **Time Series Forecast**.
2. In the Time Series Forecast pane and Output section, specify an output column for the forecasted value.
3. In the Time Series Forecast pane, configure your forecast calculation.
   - Use **Target** to select a data column with historical values.
   - Use **Time** to select a column with date information. Forecasted values use a daily grain.
   - Use **Periods** to select the value that indicates how many periods (days) are forecasted per group.

Train a Binary Classifier Model in a Data Flow

You train a machine learning model using your existing data to evaluate how accurate the model is in predicting known outcomes.

Train a Binary Classifier model to evaluate how accurately it classifies your data into one of two predefined categories. For example, you might predict whether a product instance will pass or fail a quality control test.

Use the **Train Binary Classifier** step in the data flow editor.

1. In an open data flow, click **Add a step (+)**, and select **Train Binary Classifier**.
2. At the Select Train Two-Classification Model Script dialog, select a script type, then click **OK**. For example, you might select Naive Bayes.
3. Click **Select a column** and select the data column to analyze.
4. Use the on-screen options to configure the script parameters.

Create and Train a Predictive Model

Train a Clustering Model in a Data Flow

You train a machine learning model using your existing data to evaluate how accurate the model is in predicting known outcomes.

Train a Clustering model to evaluate how accurately it segregates groups with similar traits and assigns them into clusters. For example, you might assign your customers into clusters (such as big-spenders, regular spenders and so on) based on their purchasing habits.

Use the **Train Clustering** step in the data flow editor.

1. In an open data flow, click **Add a step (+)**, and select **Train Clustering**.
2. At the Select Train Clustering Model Script dialog, select a script type, then click **OK**. For example, you might select Hierarchical Clustering for model training.
3. Use the on-screen options to configure the script parameters.

Create and Train a Predictive Model

Train a Multi-Classifier Model in a Data Flow

You train a machine learning model using your existing data to evaluate how accurate the model is in predicting known outcomes.

Train a Multi-Classifier model to evaluate how accurately it classifies your data into three or more predefined categories. For example, you might predict whether a piece of fruit is an orange, apple, or pear.

Use the **Train Multi-Classifier** step in the data flow editor.

1. In an open data flow, click **Add a step (+)**, and select **Train Multi-Classifier**.
2. At the Select Train Two-Classification Model Script dialog, select a script type, then click **OK**. For example, you might select Naive Bayes.
3. Click **Select a column** and select the data column to analyze.
4. Use the on-screen options to configure the script parameters.

Create and Train a Predictive Model

Train a Numeric Prediction Model in a Data Flow

You train a machine learning model using your existing data to evaluate how accurate the model is in predicting known outcomes.

Train a Numeric Prediction model to evaluate how accurately it predicts a numeric value based on known data values. For example, you might predict the value of a property based on square-footage, number of rooms, zip code, and so on.

Use the **Train Numeric Prediction** step in the data flow editor.

1. In an open data flow, click **Add a step (+)**, and select **Train Numeric Prediction**.
2. At the Select Train Numeric Prediction Model Script dialog, select a script type, then click **OK**. For example, you might select Random Forest for Numeric model training.
3. Click **Select a column** and select the data column to analyze.
4. Use the on-screen options to configure the train model.

Create and Train a Predictive Model

Transform Data in a Data Flow

You can use a data flow to transform data. For example, you might convert text to uppercase or trim leading and trailing spaces from data.

You can quickly transform the data in a column by using the **Options** column menu in Data Preview. For example, you might choose **Convert to Number**, **Uppercase**, or **Trim**. The list of available menu options for a column depends on the type of data in that column. Use transform options to:

- Update or modify the data in a column.
- Group or merge multiple columns in a dataset.
- Add a column to or remove a column from a dataset.

Use the **Transform Column** step in the data flow editor.

1. In an open data flow, do one of the following:
   - Click **Add a step (+)**, select **Transform Column**, then select a column.
   - Drag and drop the **Transform Column** step from the Data Flow Steps panel to the workflow diagram panel and select a column.
   - Select a column in the Preview data panel and click **Options**, then select a transform option, for example, **Uppercase**, or **Trim**. See Column Menu Options for Quick Data Transformations.
2. In the Step editor pane, compose an expression or update the fields to configure the changes. You can review the changes in the Preview data panel.

   If you're composing an expression, do the following:
   - Click **Validate** to check if the syntax is correct.
   - If the expression is valid, click **Apply** to transform the column data.

Merge Rows in a Data Flow

You can merge the rows of two data sources (known as a UNION command in SQL terminology).

Before you merge the rows, do the following:

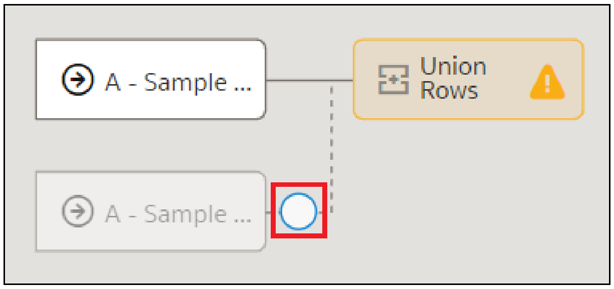
- Confirm that each dataset has the same number of columns.
- Check that the data types of the corresponding columns of the datasets match. For example, column 1 of dataset 1 must have the same data type as column 1 of dataset 2.

Use the **Union Rows** step in the data flow editor.

1. In an open data flow, add the data sources you want to merge.

   For example, you might add datasets named Order and Orders.
2. On one of the data sources, click **Add a step (+)** and select **Union Rows**.

   You'll see a suggested connection with a node on the connection line.



3. Click the node on the connection line to complete the connection.
4. Use the options on the Union Rows pane to configure your step.

Field	Description
Keep	Use these options to specify how you want to join your data. Click an option to display an explanatory diagram and preview your merged data (if you're displaying the Data Preview pane).

Add Database Analytics to a Data Flow

Database analytics enable you to perform advanced analysis and data mining analysis. For example, you can detect anomalies, cluster data, sample data, and perform affinity analysis. To use database analytics, connect to an Oracle database or Oracle Autonomous Data Warehouse (the analytics are computed in the database, not in Oracle Analytics).

Use the **Database Analytics** step in the data flow editor.

Before you start, create a connection to your Oracle database or Oracle Autonomous Data Warehouse and use it to create a dataset.

1. In an open data flow, click **Add a step (+)**, and select **Database Analytics**.

   If you aren't connected to an Oracle database or Oracle Autonomous Data Warehouse, or your administrator has disabled analytics functions, you won't see the **Database Analytics** option.

2. At the Select Database Analytics page, select the analytics operation you want to use, then click **OK**.

3. On the **Analytics Operation <type>** pane, configure the operation.

   - Use the **Inputs** or **Outputs** area to specify the data columns to analyze.
   - Use the **Parameters** area to configure options for the operation.
     To help you configure the operation, use the on-screen guidance displayed for each parameter.

Database Analytics Functions

Database analytics functions enable you to perform advanced analysis and data mining analysis, for example, detecting anomalies, clustering data, sampling data, and affinity analysis. Analytics Functions are available when you connect to an Oracle database or Oracle Autonomous Data Warehouse.

Function Types	Description
Dynamic Anomaly Detection	Detect anomalies in your input data without a pre-defined model. For example, you might want to highlight unusual financial transactions. When you deploy this function with large datasets, configure the partition columns to maximize performance.
Dynamic Clustering	Cluster your input data without a pre-defined model. For example, you might want to characterize and discover customer segments for marketing purposes. When you deploy this function with large datasets, configure the partition columns to maximize performance.

Function Types	Description
Frequent Item Set	Discover relationships in your data by identifying sets of items that often appear together. This data mining technique is also known as association rule learning, affinity analysis, or in the retail industry as market basket analysis. If you use frequent item set as a market basket analysis tool, you might find that customers who buy shampoo also buy hair conditioner. This operation is resource intensive and its performance depends on several factors, such as input dataset volume, cardinality of transaction id, and cardinality of Item value column. To avoid potential performance degradation on the database, try with a higher value of minimum support percent (default is 0.25) and gradually reduce it to accommodate more itemsets in your output.
Sampling Data	Selects a random sample percentage of data from a table. You simply specify the percentage of data you want to sample. For example, you might want to randomly sample ten percent of your data.
Text Tokenization	Analyze textual data by breaking it down into distinct words and counting the occurrences of each word. When you run your data flow, Oracle Analytics creates a table in the database named DR$*IndexName*$I, which contains the token text and the token count related details. Use the DR$*IndexName*$I table to create a dataset. • Under **Outputs**, use the **Create** option next to each field to select the columns to index. • Under **Parameters**, then **Text Column**, click **Select a column** to select the field you'd like to break down into separate words. Use the **Reference Column<number>** options to include one or more columns in the output dataset. The database connection that you use for your data flow requires special database privileges. Check with your administrator that: • Your database account has `grant EXECUTE on CTXSYS.CTX_DDL to schema name`. • You use an Oracle Analytics connection with the same username as the schema where the source table exists. This is best practice to avoid access privilege issues when the data flow runs. • The database table column you're analyzing has no existing CONTEXT index. If there's an existing CONTEXT index on the database table that you're analyzing, remove that index before you run the text tokenization data flow.
Un-pivoting Data	Transpose data that's stored in columns into row format. For example, you might want to transpose multiple columns showing a revenue metric value for each year to a single revenue column with multiple value rows for the year dimension. You simply select the metric columns to transpose and specify a name for the new column.You'll get a new dataset with fewer columns and more rows.

Note: To use analytic functions, make sure that the administrator has enabled analytics functions (via Console, System Systems, Performance and Compatibility, Enable Database Analytics Node in Data Flows).

Add Graph Analytics to a Data Flow

Graph analytics enable you to perform geo-spatial analysis. For example, you might calculate the distance or the number of hops between two vertices. To use graph analytics, connect to an Oracle database or Oracle Autonomous Data Warehouse (the analytics are computed in the database, not in Oracle Analytics).

Use the **Graph Analytics** step in the data flow editor.

Before you start, create a connection to your Oracle database or Oracle Autonomous Data Warehouse and use it to create a dataset.

1. In an open data flow, click **Add a step (+)**, and select **Graph Analytics**.

   If you aren't connected to an Oracle database or Oracle Autonomous Data Warehouse, you won't see the **Graph Analytics** option.

2. At the Select Database Analytics page, select the analytics operation you want to use, then click **OK**.

3. On the **Analytics Operation <type>** pane, configure the operation.

   - Use the **Inputs** or **Outputs** area to specify the data columns to analyze.
   - Use the **Parameters** area to configure options for the operation.
     To help you configure the operation, use the on-screen guidance displayed for each parameter.

Graph Analytics Functions

Graph analytics enable you to perform geo-spatial analysis. For example, you might calculate the distance or the number of hops between two vertices. To use graph analytics, connect to an Oracle database or Oracle Autonomous Data Warehouse (the analytics are computed in the database, not in Oracle Analytics).

Functions:

Function Types	Description
Clustering	Finds connected components or clusters in a graph.
Node Ranking	Measure the importance of the nodes in a graph.
Shortest Path	Find the shortest path between two vertices in a graph.
Sub Graph	Find all nodes within a specified number (n) of hops of a given node.

6
Manage Datasets

This topic describes the different ways you can manage your datasets.

Topics:

- Dataset Type Icons
- View a List of Datasets and Information About Them
- Reload a Dataset's Data
- Inspect a Dataset's Properties
- Rename a Dataset and Change Its Description
- Copy a Dataset's Object ID
- View a Dataset's Data Elements
- Download a Dataset
- Duplicate a Dataset
- Delete a Dataset

Dataset Type Icons

A dataset's icon identifies the connection type or file it uses to source its data.

Dataset type icons are displayed in your list of datasets. See View a List of Datasets and Information About Them.

Dataset Icon	Description
🗐	Dataset uses two or more database connections. For example, Oracle Database and Oracle Analytics Warehouse.
🛢	Dataset uses one database connection.
🗎	Dataset uses an XLXS or XLS file.
🗎	Dataset uses a CSV or TXT file.
☁	Dataset uses an Oracle Applications connection. This dataset can use a subject area or analysis.

View a List of Datasets and Information About Them

You can access, browse, sort, or search a list of datasets. You can choose any available dataset to use in a workbook.

You can use the dataset list to find out more information about each dataset, for example which data source connection it uses and what data elements it contains. See Dataset Type Icons.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Optional: Enter a term in the **Search** field to search the list for a specific dataset.
4. To check a dataset's details, click **Actions** and then click **Inspect**.
5. Optional: Click the General tab to see information about the dataset, such as the connection used to create it and the database type that the dataset gets its data from.
6. Optional: Click the Data Elements tab to find information that can help you decide if the dataset contains the data that you need for your visualizations. For example, each element's name, data type, and aggregation.

Reload a Dataset's Data

This topic explains the relationship between reloading the data used in a dataset and refreshing the data used in a workbook. It also explains the different ways that you can reload a dataset's data.

Topics:

- About Reloading a Dataset's Data
- Reload an Individual Table in a Dataset
- Reload a Dataset's File

About Reloading a Dataset's Data

You can reload a dataset that uses a file as its source, when one or more tables in the dataset are cached, or when the dataset is created from running a data flow.

Reloading data ensures that workbooks and visualizations contain current data. The most current data is displayed in workbooks and visualizations after the dataset reload is complete and you refresh the workbooks. See Refresh a Workbook's Data.

How you reload data depends on the dataset's data source type.

Data from External Connections

When you reload data for a dataset with an external connection, the dataset's SQL statement is rerun and the current data for the tables with the **Data Access** field set to **Automatic Caching** is loaded into cache.

If your dataset contains one or more tables with the **Data Access** field set to **Automatic Caching**, then you can use the **Data** page to reload the tables in the dataset. To reload individual tables, use the **Dataset** editor.

Upload Excel, CSV, or TXT File

When you reload a Microsoft Excel file (XLSX or XLS), you must ensure that the newer spreadsheet file contains a sheet with the same name as the original file. And the sheet must contain the same columns that are in the dataset. The data reload fails if the file you load is missing columns. You can reload an Excel file that contains new columns.

When you reload a CSV or TXT file, you must ensure that it contains the same columns that are in the dataset. The data reload fails when the file you load is missing columns.

Oracle Applications

You can reload data and metadata for Oracle Applications data sources. If the Oracle Applications data source uses logical SQL, reloading data reruns the dataset's SQL statement.

Reload an Individual Table in a Dataset

You can use the Dataset editor to reload data for an individual dataset table. Reloading queries the table's data source and loads the current data into cache.

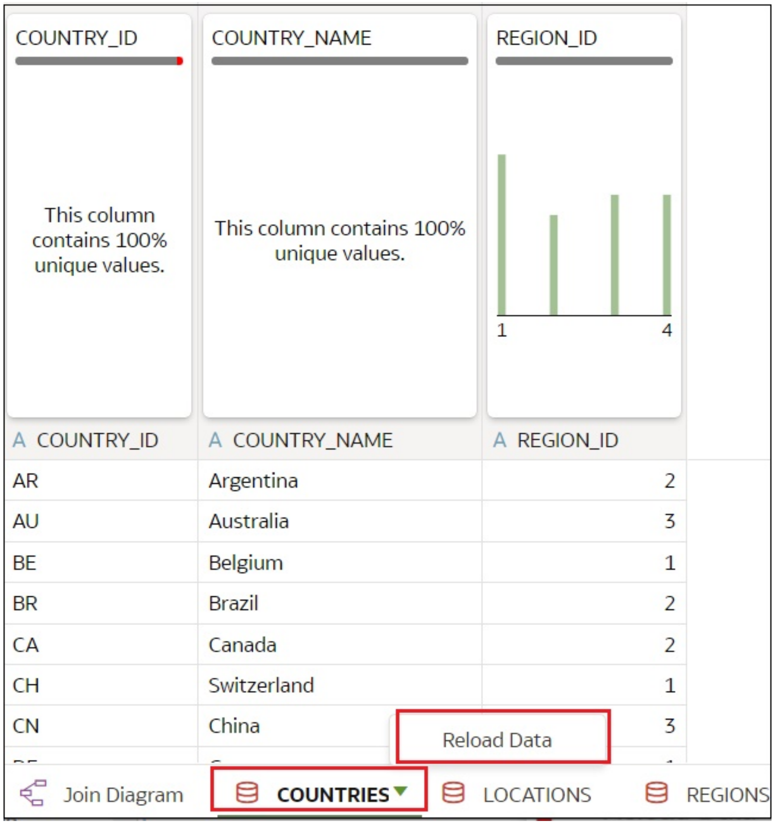
The Reload option is available for any dataset table with the **Data Access** field set to **Automatic Caching**.

Typically you reload all of the dataset's tables at the same time so that the data is consistent across all tables. But sometimes it makes sense to reload only one of the dataset's tables. For example, where you know that the data has changed in a fact table but not in any of the dimension tables.

Reloading doesn't update the table's Profile data preview information. To see the most current data in the data preview, you must re-profile the table after you reload the table.

If you're using the Dataset editor and your dataset contains more than one table, then you must reload each table individually.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.
4. In the **Table Page Tabs**, locate the table that you want to reload, and click its arrow.
5. Click **Reload Data**.



This column contains 100% unique values.

This column contains 100% unique values.

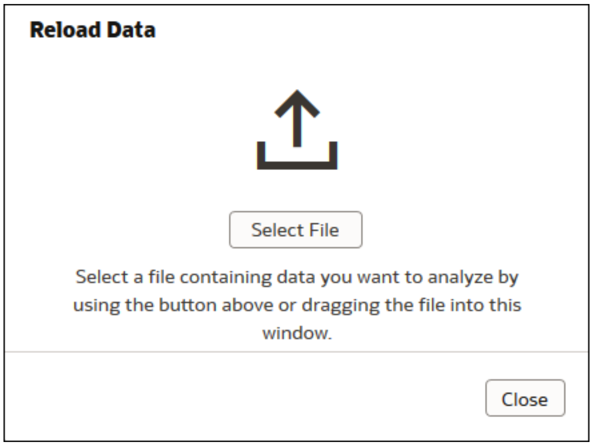
A COUNTRY_ID	A COUNTRY_NAME	A REGION_ID
AR	Argentina	2
AU	Australia	3
BE	Belgium	1
BR	Brazil	2
CA	Canada	2
CH	Switzerland	1
CN	China	3

Reload a Dataset's File

You can use the Transform editor to reload the XLSX, XLS, CSV, or TXT file that supplies a dataset's data

For information about structuring spreadsheets to ensure a successful upload, see About Files for Datasets.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to open, click **Actions**, and then click **Open**.
4. In the **Table Page Tabs**, locate the table that you want to reload, and click its arrow.
5. In the menu, click **Reload Data**.
6. In the Reload Data dialog, drag and drop the file onto the dialog or click **Select File** to browse for the file.



Reload Data

Select a file containing data you want to analyze by using the button above or dragging the file into this window.

7. Click **OK**.

Inspect a Dataset's Properties

You can view information about the dataset such as when it was created, when it was last modified, and a list of its data elements.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to inspect properties for, click **Actions**, and then click **Inspect**.
4. Click the tabs to navigate and inspect the dataset's properties.
5. Click **Close**.

Rename a Dataset and Change Its Description

You can rename and change the description of any dataset.

Renaming a dataset doesn't affect a workbook's reference to the dataset, and any workbooks using the dataset display the new name and continue to work as designed. When you add a dataset to a workbook, the workbook connects to the dataset by the dataset's fixed object ID, which was assigned when the dataset was created and saved, and not by its name. See Copy a Dataset's Object ID.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to rename, click **Actions**, and then click **Inspect**.
4. Click the **General** tab.
5. Go to the **Name** field and rename the dataset.
6. Go to the **Description** field and update the dataset's description.
7. Click **Save**.

Copy a Dataset's Object ID

Each dataset has an assigned, unique, fixed object ID. You can use this ID to help troubleshoot Oracle Analytics errors and issues, search log files, or log an Oracle Support request.

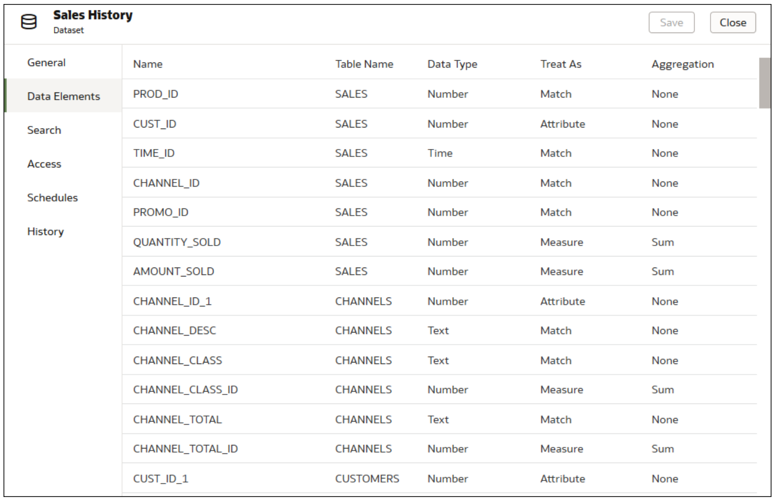
The dataset's object ID, and not the dataset's name, is used to reference, track, and manage the dataset. In most cases, the object ID is based on the name that you provide when you create the dataset. Updating a dataset's name doesn't update its object ID. See Rename a Dataset and Change Its Description.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that with the ID that you want to copy, click **Actions**, and then click **Inspect**.
4. Click the **General** tab.
5. Go to the **Object ID** field and click the **Copy** button.

View a Dataset's Data Elements

The data elements list contains information such as data element name, data type, and aggregation. You can use this information to determine if the dataset contains data that you want to visualize and analyze.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to view data elements for, click **Actions**, and then click **Inspect**.
4. Click the **Data Elements** tab.



Name	Table Name	Data Type	Treat As	Aggregation
PROD_ID	SALES	Number	Match	None
CUST_ID	SALES	Number	Attribute	None
TIME_ID	SALES	Time	Match	None
CHANNEL_ID	SALES	Number	Match	None
PROMO_ID	SALES	Number	Match	None
QUANTITY_SOLD	SALES	Number	Measure	Sum
AMOUNT_SOLD	SALES	Number	Measure	Sum
CHANNEL_ID_1	CHANNELS	Number	Attribute	None
CHANNEL_DESC	CHANNELS	Text	Match	None
CHANNEL_CLASS	CHANNELS	Text	Match	None
CHANNEL_CLASS_ID	CHANNELS	Number	Measure	Sum
CHANNEL_TOTAL	CHANNELS	Text	Match	None
CHANNEL_TOTAL_ID	CHANNELS	Number	Measure	Sum
CUST_ID_1	CUSTOMERS	Number	Attribute	None

Download a Dataset

You can download any file-based dataset.

Here are some reasons why you would download a dataset:

- You uploaded the dataset file, but deleted, lost, or misplaced the original file.
- You imported the workbook and need to access the original dataset file.

When you download a dataset to a file, by default the resulting file is the same format as the file used to create the dataset. However, you can change the file type if you choose to save the file.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to download, click **Actions**, and then click **Download File**.
4. When prompted, open or save the file.

Duplicate a Dataset

You can duplicate any dataset. Duplicating a dataset and modifying the copy can be faster than creating and building a dataset.

You're the owner of the duplicate dataset. The default name of the duplicate dataset is the copied dataset's name appended with Copy. For example, SAMPLE_REVENUE-Copy.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to duplicate, click **Actions**, and then click **Duplicate**.

Delete a Dataset

Delete a dataset when it's no longer needed or you need to free up space on your system.

> **Note:**
>
> Deleting a dataset permanently removes it and breaks any workbooks that use it. When you choose to delete a dataset, Oracle Analytics won't list workbooks that use the dataset. Before you delete a dataset you must be sure that it isn't used in any workbooks.

1. On the Home page, click **Navigator** and then click **Data**.
2. Click the **Datasets** tab.
3. Locate the dataset that you want to delete, click **Actions**, and then click **Delete**.

7

Visualize and Analyze Data

This topic describes the many ways that you can visualize and analyze your data.

Video

Topics:

- Begin to Build a Workbook and Create Visualizations
- Work with a Workbook's Datasets
- Build a Visualization by Adding Data from the Data Panel
- Enhance Data in Visualizations with Advanced Analytics Functions
- Use Spark Charts to Examine Trends
- Create Calculated Data Elements in a Workbook
- Undo and Redo Edits
- About Refreshing a Workbook's Data
- Refresh a Workbook's Data
- Pause Data Queries in a Workbook
- Work with Canvas Properties
- Copy and Paste a Visualization or Canvas
- Copy and Paste a Visualization's Data
- Work with Multiple Visualizations on a Canvas
- Change Visualization Types
- Adjust Visualization Properties
- Highlight Important Data Events with Conditional Formatting
- Apply Color to Visualizations
- Format Numeric Values of Columns
- Set Currency Symbols for Visualizations
- Add Notes to Visualizations
- Format Numeric Values of Visualizations
- Apply Map Backgrounds and Map Layers to Enhance Visualizations
- Sort, Drill, and Select Data in Visualizations
- Replace a Dataset in a Workbook
- Remove a Dataset from a Workbook
- Analyze Data with Explain
- About Warnings for Data Issues in Visualizations

Begin to Build a Workbook and Create Visualizations

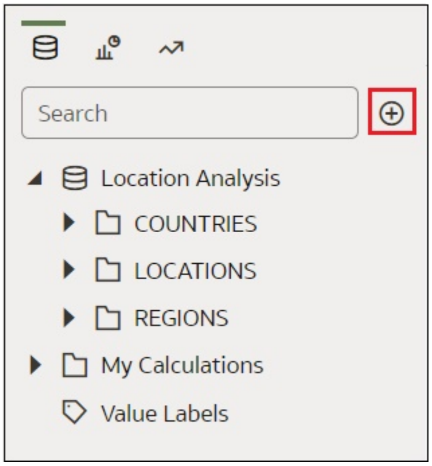
To start a workbook, choose and add a dataset and then drag and drop its columns onto the Visualize page's canvas to create visualizations.

Tutorial

If you want to learn about the datasets that are available to you, then you can view a list of datasets and find out more information about each dataset. See View a List of Datasets and Information About Them.

A workbook can contain one or more datasets. When you add more than one dataset, Oracle Analytics blends the data. You can check the default blending, or update or add data blending between datasets. See Blend Datasets.

1. On the Home page, click **Create** and click **Workbook**.
2. In the Add Dataset dialog, click a dataset and then click **Add to Workbook**.
3. Optional: To add another dataset, go to the Data Panel, click **Add** (+).



4. Optional: Select **Add Dataset**, and in the Add Dataset select a dataset and click **Add to Workbook**.
5. In the Data Panel, locate and drag and drop data columns onto the Visualize canvas to start building visualizations.
6. Click **Save**.

Work with a Workbook's Datasets

This topic describes what you need to know to work with a workbook's datasets.

Topics:

- Add Datasets to a Workbook
- Replace a Dataset in a Workbook
- Remove a Dataset from a Workbook
- Modify a Workbook's Datasets

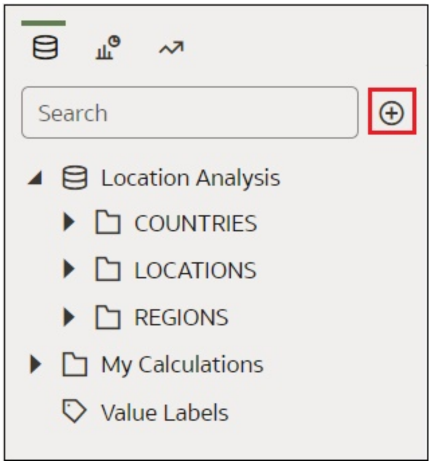
- Blend Datasets

Add Datasets to a Workbook

Your workbook can use more than one dataset.

When you add more than one dataset, Oracle Analytics blends the data. You can check the default blending, or update or add data blending between datasets. See Blend Datasets.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the Visualize canvas, go to the Data panel, click **Add** (+).



3. Select **Add Dataset**. In Add Dataset, select a dataset and click **Add to Workbook**.
4. Optional: Click **Data** to go to the **Data Diagram** and view how the datasets are blended.

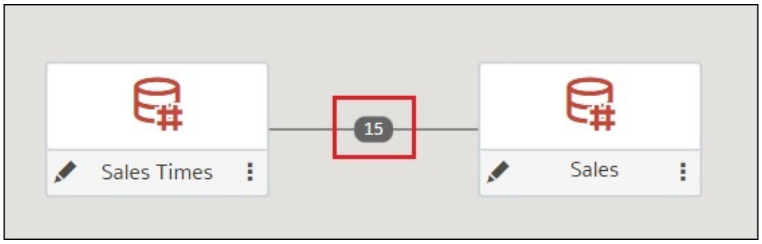
Replace a Dataset in a Workbook

When you replace a dataset, Oracle Analytics suggests how you can re-map the columns from the dataset that you replaced to the replacement dataset. You can accept or update these suggested mappings.

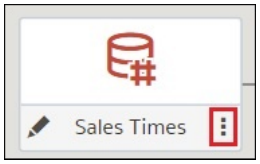
For example, you might replace a workbook's test dataset with a production dataset. Or you can copy a workbook and its visualizations, and then use the duplicate workbook as a template. After you rename the duplicated workbook, you can add a different dataset.

When you replace the dataset, any data mapping that you specify is applied to the workbook. For example, if you map a data element to **None**, the specific data is removed from the workbook's visualizations, calculations, and filters.

1. On the Home page, select a workbook to use as a template, click the **Actions menu**, and then select **Open**.
2. From the Visualize page, click **Data**.
3. In the Data page, click the **Data Diagram**.
4. Optional: If the dataset that you want to replace has been blended with another dataset, then in the **Data Diagram** click the number between the dataset icons.



5. Optional: In the Blend Data dialog, click **Delete All Matches**, specify which dataset to delete conformance from, and then click **OK**.
6. Locate the dataset to replace and click **Actions**.

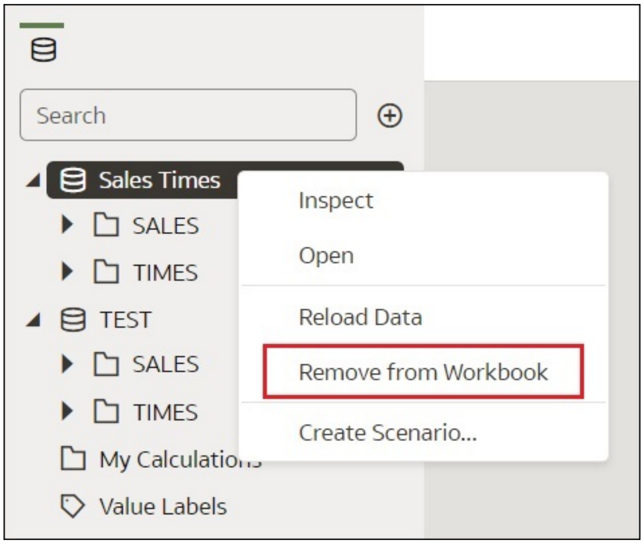


7. Select **Replace Dataset**.
8. In the Replace Dataset dialog, click the replacement dataset. Click **Select**.
9. Review, update, and specify the column mappings for the workbook's visualizations, calculations, and filters.
10. Click **Replace**.
11. Optional: If you want to add blending between datasets, then in the **Data Diagram**, click between the datasets that you want to blend, and in the Blend Data dialog add column matches.

Remove a Dataset from a Workbook

You can remove a dataset from a workbook. When you do this, you remove all of the dataset's columns from the workbook's visualizations, filters, and calculations.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the **Visualize** canvas, go to the Data Panel pane, right-click the dataset that you want to remove, and select **Remove from Workbook**.



3. Click **Save**.

Modify a Workbook's Datasets

You can use the Workbook Editor's **Data** page to navigate to the Dataset editor or Transform editor to view or modify a workbook's datasets.

When you locate a dataset's icon in the data diagram and click **Edit**, a new window opens that contains the editor that you use to view or modify the dataset. The Workbook Editor remains open in its own window and immediately reflects any saved dataset changes.

> **Note:**
>
> A dataset can be used in multiple workbooks and data flows. Modifying a dataset impacts all workbooks and data flows that use the dataset.

See About the Dataset Editor and Enrich and Transform Your Data.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the workbook editor, click **Data** to go to the Data page.
3. In the data diagram, locate the dataset and click its **Open** button.

   The window that's displayed depends upon the dataset you selected:

   - If the dataset uses a connection that doesn't support multiple tables (for example, spreadsheet or file, or Essbase), then the Transform Editor is opened in a window.
   - If the dataset uses a connection that supports multiple tables (for example, Oracle Database or Oracle Autonomous Data Warehouse), then the Dataset Editor is opened in a window.
4. Optional: If the Dataset editor is displayed, click the tab for the table that you want to view or modify and the Transform Editor is displayed.
5. Modify and save the dataset.

Blend Datasets

This topic explains blending and how to blend a workbook's datasets.

Topics:

- Understand Blending
- About Mismatched Values in Blended Data
- Blend Datasets
- Change Data Blending in a Visualization

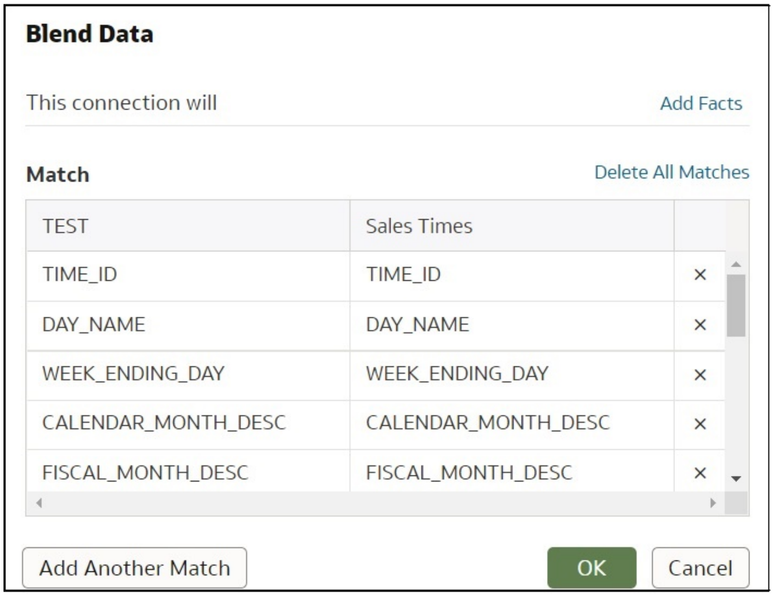
Understand Blending

When you add more than one dataset to a workbook, Oracle Analytics tries to match columns between the datasets that you added. This matching is called blending.

Video

For example, Dataset A might contain new dimensions that extend the attributes of Dataset B. Or Dataset B might contain new facts that you can use alongside the measures that already exist in Dataset A.

Blending automatically matches external dimensions where they share a common name and have a compatible data type with attributes in the existing dataset.



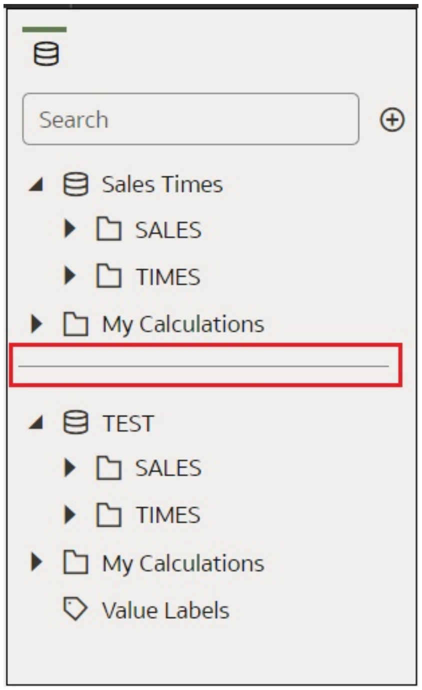
Blend Data

This connection will

Match

TEST	Sales Times	
TIME_ID	TIME_ID	×
DAY_NAME	DAY_NAME	×
WEEK_ENDING_DAY	WEEK_ENDING_DAY	×
CALENDAR_MONTH_DESC	CALENDAR_MONTH_DESC	×
FISCAL_MONTH_DESC	FISCAL_MONTH_DESC	×

Datasets that aren't blended are divided by a line in the workbook's Data Panel.



There are restrictions between data elements and visualizations when a workbook contains datasets that aren't blended. For example, if your workbook contains two datasets that aren't blended, then you can't include the data elements of one dataset in the filters, visualizations, or calculations of the other dataset.

Blending relationships are stored in the tables and not in the workbook. So any blending matches that you add or remove impact the other datasets that use the same tables.

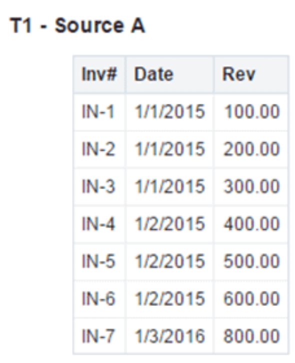
> **Note:**
>
> Your workbook might contain a dataset that includes tables from different connections and schemas. When datasets contain multiple tables, the tables most likely contain joins. See Understand Dataset Table Joins. In your workbook, you can blend datasets that contain multiple tables and joins. For example, you can add and blend datasets with multiple tables when you want your workbook to contain cross fact analysis.

About Mismatched Values in Blended Data

In some cases when the rows of data that you expect to see in a dataset are missing, then you must specify which dataset to use for data blending.

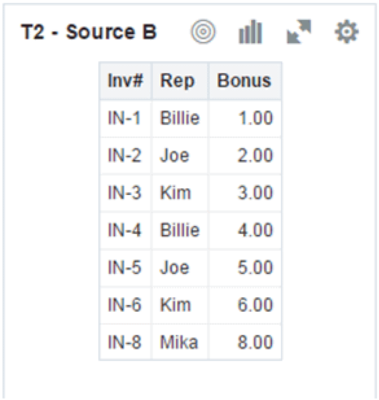
Sometimes rows of data are missing when your workbook includes data from two datasets that contain a mixture of attributes and values, and there are match values in one source that don’t exist in the other.

Suppose we have two datasets (Source A and Source B) with slightly different rows, as shown in the following image. Note that Source A doesn‘t include IN-8 and Source B doesn’t include IN-7.



T1 - Source A

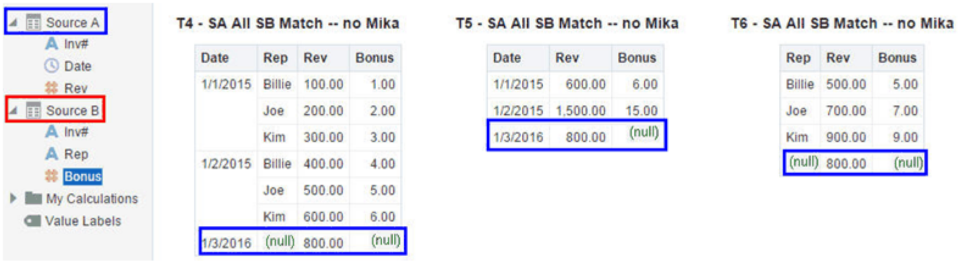
Inv#	Date	Rev
IN-1	1/1/2015	100.00
IN-2	1/1/2015	200.00
IN-3	1/1/2015	300.00
IN-4	1/2/2015	400.00
IN-5	1/2/2015	500.00
IN-6	1/2/2015	600.00
IN-7	1/3/2016	800.00



T2 - Source B

Inv#	Rep	Bonus
IN-1	Billie	1.00
IN-2	Joe	2.00
IN-3	Kim	3.00
IN-4	Billie	4.00
IN-5	Joe	5.00
IN-6	Kim	6.00
IN-8	Mika	8.00

The following results are displayed if you select the **All Rows** data blending option for Source A and select the **Matching Rows** data blending option for Source B. Because IN-7 doesn't exist in Source B, the results contain null Rep and null Bonus.



T4 - SA All SB Match -- no Mika

Date	Rep	Rev	Bonus
1/1/2015	Billie	100.00	1.00
	Joe	200.00	2.00
	Kim	300.00	3.00
1/2/2015	Billie	400.00	4.00
	Joe	500.00	5.00
	Kim	600.00	6.00
1/3/2016	(null)	800.00	(null)

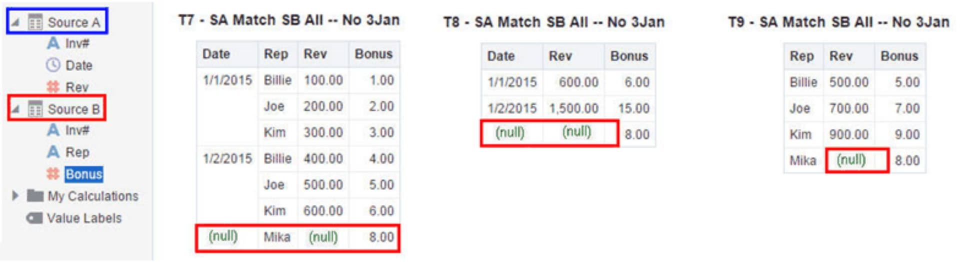
T5 - SA All SB Match -- no Mika

Date	Rev	Bonus
1/1/2015	600.00	6.00
1/2/2015	1,500.00	15.00
1/3/2016	800.00	(null)

T6 - SA All SB Match -- no Mika

Rep	Rev	Bonus
Billie	500.00	5.00
Joe	700.00	7.00
Kim	900.00	9.00
(null)	800.00	(null)

The following results are displayed if you select the **Matching Rows** data blending option for Source A and select the **All Rows** data blending option for Source B. Because IN-8 doesn't exist in Source A, the results contain null Date and null Revenue.



T7 - SA Match SB All -- No 3Jan

Date	Rep	Rev	Bonus
1/1/2015	Billie	100.00	1.00
	Joe	200.00	2.00
	Kim	300.00	3.00
1/2/2015	Billie	400.00	4.00
	Joe	500.00	5.00
	Kim	600.00	6.00
(null)	Mika	(null)	8.00

T8 - SA Match SB All -- No 3Jan

Date	Rev	Bonus
1/1/2015	600.00	6.00
1/2/2015	1,500.00	15.00
(null)	(null)	8.00

T9 - SA Match SB All -- No 3Jan

Rep	Rev	Bonus
Billie	500.00	5.00
Joe	700.00	7.00
Kim	900.00	9.00
Mika	(null)	8.00

The visualization for Source A includes Date as an attribute, and Source B includes Rep as an attribute, and the match column is Inv#. Under dimensional rules, you can't use these attributes with a measure from the opposite table unless you also use the match column.

There are two settings for blending tables that contain both attributes and measures. These are set independently in each visualization based on what columns are used in the visualization. The settings are **All Rows** and **Matching Rows** and they describe which source rows the system uses when returning data to be visualized.

The system automatically assigns data blending according to the following rules:

- If the visualization contains a match column, then the system sets sources with the match column to **All Rows**.
- If the visualization contains an attribute, then the system sets its source to **All Rows** and sets the other sources to **Matching Rows**.
- If attributes in the visualization come from the same source, then the system sets the source to **All Rows**, and sets the other sources to **Matching Rows**.
- If attributes come from multiple sources, then the system sets the source listed first in the workbook's elements panel to **All Rows** and sets the other sources to **Matching Rows**.

Blend Datasets

If your workbook contains multiple datasets, then you can blend data in one dataset with data in another dataset.

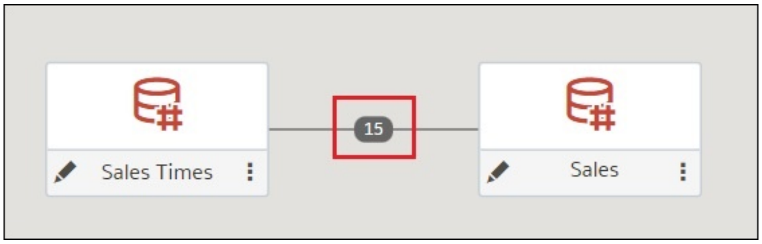
When you add more than one dataset to a workbook, the system tries to find matches for the data that's added. It automatically matches external dimensions where they share a common name and have a compatible data type with attributes in the existing dataset. Blending relationships are stored in the tables and not in the workbook. So any blending matches that you add or remove affects the other datasets that use the same tables.

See Understand Blending.

> **Note:**
>
> Datasets can include multiple tables from different connections and schemas. These datasets use joins, which are defined in the Dataset editor. To reduce or eliminate blending in workbooks, Oracle suggests that whenever possible you create datasets that contain multiple tables and use joins. See Create a Dataset From a Connection.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. To add another dataset to the workbook, then click **Data**, and in the Data page, go to the Data Panel, click **Add** (+), and then **Add Dataset**.
3. In the Add Dataset dialog, select a dataset and click Add to Workbook.
4. In the Data page, go to the **Data Diagram** and locate the datasets to blend.

   You can't blend datasets that use Oracle Essbase or Oracle Enterprise Performance Management data sources.
5. Click the number between the dataset icons.



6. Optional: To change the match for a column, click the name of the column and select a different column.

   If columns have the same name and same data type, then they're recognized as a possible match. You can customize this and specify that one column matches another by explicitly selecting it even if its name isn't the same. You can select only those columns with a matching data type.

7. Optional: Click **Add Another Match**, and in the table, select the columns that you want to join.

8. Optional: For a measure that you're uploading for the first time, specify the aggregation type such as *Sum* or *Average*.

9. Click **OK**.

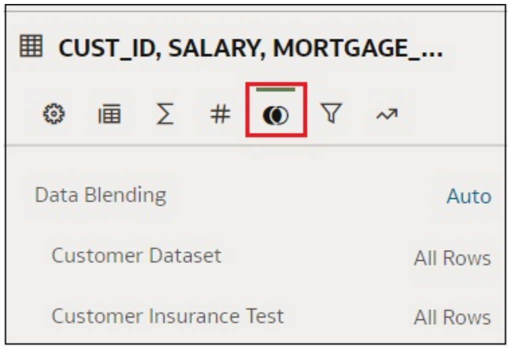
Change Data Blending in a Visualization

You can modify a visualization's properties to override the workbook's data blending settings.

See Understand Blending.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.

2. In the workbook, select a visualization, and in the Properties Panel click **Datasets**.



CUST_ID, SALARY, MORTGAGE_...

Data Blending Auto

Customer Dataset All Rows

Customer Insurance Test All Rows

3. To change the default blending, click **Data Blending**, and select either **Auto** or **Custom**.

   If you choose **Custom**, you can set the blending to either **All Rows** or **Matching Rows**.

   - You must assign at least one source to **All Rows**.
   - If both sources are **All Rows**, then the system assumes that the tables are purely dimensional.
   - You can't assign both sources to **Matching Rows**.

Build a Visualization by Adding Data from the Data Panel

This topic describes how to add data to a visualization.

Topics:

- Automatically Create the Best Visualization
- Use the Grammar Panel to Add Data to the Visualization
- Use the Assignment Panel to Add Data to the Visualization
- Modify a Visualization's Tooltips

Different Methods to Add Data

You can use different methods to add data from the Data Panel to create or update visualizations on a canvas.

Use one of the following methods to add data from the Data Panel:

- Drag data elements from the Data Panel and drop them onto the canvas.
- Select a data element or use Shift-click or Ctrl-click to select multiple data elements in the Data Panel, then right-click to select either a particular visualization type or the option to create a visualization automatically.
- Double-click a data element or use Shift-click or Ctrl-click to select multiple data elements in the Data Panel, then right-click to add them to the canvas.

When you update an existing visualization, you can add data from the Data Panel onto the Grammar Panel or Assignments Pane on the canvas. You can move the data elements from one area to another in the Grammar Panel or Assignments Pane. The visualization is updated based on your selection.

You can create a visualization by selecting a visualization type from the Visualization tab of the Data Panel and adding data elements to the canvas.

Automatically Create the Best Visualization

You can automatically create the best visualization on the canvas based on a set of data elements.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. To automatically create a visualization on the canvas, do one of the following:
   - Drag data elements from the Data Panel and drop them directly onto the canvas.
   - Right-click data elements on the Data Panel and click **Create Best Visualization**.
   - Double-click data elements to add them to the canvas.

   A visualization is automatically created on the canvas, and the best visualization type is selected based on the preconfigured logic. The selected data element is also positioned on a specific area of the Grammar Panel. For example, if you add a revenue measure to the canvas, the data element is placed in the Values area of the Grammar Panel, and **Tile** is selected as the visualization type.
3. Continue adding data elements directly to the canvas to build your visualization.

The visualization type and the position of the data elements on the Grammar Panel might change when you add more data elements to the canvas.

You turn off the mode to automatically create visualizations when you perform actions such as:

- Changing the visualization properties such as type.
- Adding or removing analyses from the Analytics tab in the Data Panel.
- Dropping a data element onto a specific area of the Grammar Panel or Assignments Pane.
- Changing the filter setting.

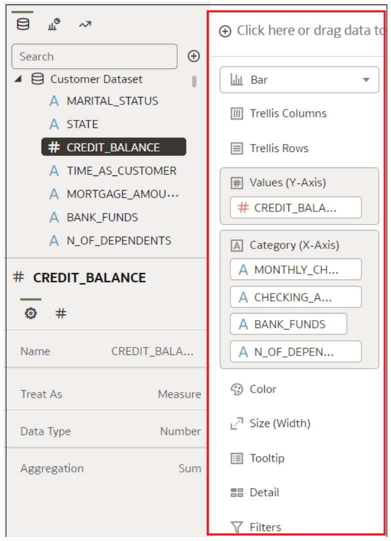
When the automatically create visualization mode is turned off, you can select the wizard icon or **Auto Visualization** in the visualization type list to turn on the automatically create mode.

Use the Grammar Panel to Add Data to the Visualization

After you've selected the datasets for your workbook, you can begin to add data elements such as measures and attributes to visualizations.

If you haven't already created a visualization, you need to create one. See Begin to Build a Workbook and Create Visualizations.

You can select compatible data elements from the datasets and drop them onto the Grammar Panel in the Visualize canvas. Based on your selections, visualizations are created on the canvas. The Grammar Panel contains sections such as Columns, Rows, Values, and Category.



Name CREDIT_BALA...
Treat As Measure
Data Type Number
Aggregation Sum

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. If you created a workbook, then add a dataset to it.
3. Use one of the following methods to add data elements to the Grammar Panel:

You can only drop data elements based on attribute and type onto a specific Grammar Panel section.

- Drag and drop one or more data elements from the Data Panel to the Grammar Panel in the Visualize canvas.
  The data elements are automatically positioned, and if necessary the visualization changes to optimize its layout.
- Replace a data element by dragging it from the Data Panel and dropping it over an existing data element.
- Swap data elements by dragging a data element already inside the Visualize canvas and dropping it over another data element.
- Reorder data elements in the Grammar Panel section (for example, Columns, Rows, Values) to optimize the visualization, if you've multiple data elements in the Grammar Panel section.
- Remove a data element by selecting a data element in the Grammar Panel, and click **X**.

Use the Assignment Panel to Add Data to the Visualization

You can use the Assignment Panel to help you position data elements in the optimal locations for exploring content.

A workbook must contain one or more datasets before you can add data elements to the Assignment Panel. The sections in the Assignment Panel are the same as in the Grammar Panel.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Select a visualization on the canvas.

   Alternatively, drag and drop a data element to the blank canvas or between visualizations on the canvas to create a new visualization.
3. Drag a data element to the visualization (but not to a specific drop target); you'll see a blue outline around the recommended assignments in the visualization.

   Hover the data element on the Assignment Panel to identify other valid assignments.
4. Drop the data element on the selected assignment.

To display the Assignment Panel, click **Show Assignments** on the visualization toolbar.

Modify a Visualization's Tooltips

When you hover over a data point in a visualization, a tooltip displays and provides specific information about the data point. You can choose to see all tooltips or only the measures included in the Tooltip section of the Grammar Panel.

For example, if you create a simple bar chart visualization that shows revenue for countries in the Americas region, the tooltip displays the region's name, the country's name, and exact revenue amount. If you add Target Revenue to the Tooltip section of the Grammar Panel, then the target revenue amount is displayed in the tooltip and the user can easily compare the actual revenues with the target revenues. Set the **Tooltip** field in the General Properties Pane to **Tooltip Grammar Only** if you want the tool tip to contain only the target revenue amount.

Note the following restrictions:

- You can drag and drop only measure columns to the Tooltip section in the Grammar Panel.
- The Tooltip section in the Grammar Panel doesn't display for all visualization types.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Select a visualization on the canvas.
3. Drag and drop one or more measure columns from the Data Panel to the Tooltip section in the Grammar Panel.

   Hover the mouse pointer over a data point on the visualization to display the tooltip. Because the **Tooltip** field is set to **All Data** by default, the tooltip contains the data point's values for all columns included in the visualization. The data values for the columns that you added to the Tooltip section are displayed at the bottom of the tooltip.
4. Optional: Use the **Tooltip** field to display only the data values that you want or to turn off the tooltip.
   - If you want the tooltip to display data values for only the columns you dragged to the Tooltip section, then set the **Tooltip** field to **Tooltip Grammar Only**.
   - If you don't want the tooltip to be displayed, then confirm that there are no columns in the Tooltip section and set the **Tooltip** field to **Tool Tip Grammar Only**.

Enhance Data in Visualizations with Advanced Analytics Functions

Advanced analytics are statistical functions that you apply to enhance the data displayed in visualizations.

The Analytics area in the Data panel contains standard analytics functions (for example, Clusters and Trend Line). You can use analytics functions as they are, or use them to create your own calculated columns that reference statistical scripts.

Topics:

- About Advanced Analytics Functions
- Add Advanced Analytics Functions to Visualizations
- Add Reference Lines to Visualizations

About Advanced Analytics Functions

You can use advanced analytics functions, like forecast, cluster, and outliers, in charts to achieve better insights into your data.

Forecast

The forecast function uses linear regression to predict future values based on existing values along a linear trend.

You can set a number of time periods in the future for which you want to predict the value, based on your existing time series data. See Add Advanced Analytics Functions to Visualizations.

Oracle supports these forecast model types:

- **Auto-Regressive Integrated Moving Average (ARIMA)** - Use if your past time series data is nonseasonal but provides enough observations (at least 50, but preferably more than 100 observations) to explain and project the future.
- **Seasonal ARIMA** - Use if your data has a regular pattern of changes that repeat over time periods. For example, seasonality in monthly data might be when high values occur during summer months and low values occur during winter months.
- **Exponential Triple Smoothing (ETS)** - Use to analyze repetitive time series data that doesn't have a clear pattern. It produces an exponential moving average that takes into account the tendency of data to repeat itself in intervals over time.

Prediction Interval - Provides an interval within which we expect the value to lie with a specified probability. For example, a 95% prediction interval suggests that you can be 95% confident that the new observation will fall within the range.

Create a custom calculation using the `FORECAST` function to have more control over settings, or if you want to use the forecast in other visualizations. See Time Series Functions.

Clusters

The cluster function groups a set of objects in such a way that objects in the same group show more coherence and proximity to each other than to objects in other groups. For example, you can use colors in a scatter chart to show clusters of different groups. See Add Advanced Analytics Functions to Visualizations.

- **K-means clustering** - Use to partition "n" observations into "k" clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.
- **Hierarchical clustering** - Use to create a hierarchy of clusters built using either an agglomerative (bottom-up) approach, or a divisive (top-down) approach.

Create a custom calculation using the `CLUSTER` function to have more control over settings, or if you want to use the cluster in other visualizations. See Analytics Functions.

Outliers

The outliers function displays data records that are located the furthest away from the average expectation of individual values. For example, extreme values that deviate the most from other observations fall into this category. Outliers can indicate variability in measurement, experimental errors, or a novelty. If you add outliers to a chart that already has clusters, then the outliers are depicted as different shapes.

Outliers can use K-means clustering or hierarchical clustering. See Add Advanced Analytics Functions to Visualizations.

Create a custom calculation using the `OUTLIER` function to have more control over settings, or if you want to use the outlier in other visualizations. See Analytics Functions.

Reference Lines

The reference lines function defines horizontal or vertical lines in a chart that correspond to the X-axis or Y-axis values. See Add Reference Lines to Visualizations.

- **Line** - You can choose to compute the line between average, minimum, or maximum. For example, in the airline industry, if passenger turnout is plotted against time, the reference line can show whether passenger turnout for a particular month is above or below average.
- **Band** - A band represents upper and lower range of data points. You can choose a custom option or a standard deviation function, and between average, maximum, and minimum. For example, if you're analyzing sales by month and you use a custom reference band from average to maximum, you can identify months where sales are above average, but below the maximum.

Trend Lines

The trend line function indicates the general course of the metric in question. A trend line is a straight line connecting a number of points on a graph. A trend line helps you analyze the specific direction of a group of value sets in a visualization. See Add Advanced Analytics Functions to Visualizations.

- **Linear** - Use with linear s. Your data is linear if the pattern in its data points resembles a line. A linear trend line shows that your metric is increasing or decreasing at a steady rate.
- **Polynomial** - Use this curved line when data fluctuates. It's useful, for example, for analyzing gains and losses over a large dataset.
- **Exponential** - Use this curved line when data values rise or fall at increasingly higher rates. You can't create an exponential trend line if your data contains zero or negative values.

Use the `TRENDLINE` function to create custom calculations, or if you want to use the trend line in other visualizations. See Analytics Functions.

Add Advanced Analytics Functions to Visualizations

You can apply advanced analytics functions to your workbook's visualizations.

Before you can use analytic functions in visualizations, you must do the following:

- Install DVML.

  On Windows go to **Start**, browse to and expand your system's Oracle folder, and click **Install DVML**.

  On Mac, go to **Applications** and click **Oracle Analytics Desktop Configure Python**.
- Create a workbook or visualization that you can apply one or more analytic functions to.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click the **Analytics** icon in the Data panel.
3. Drag and drop an advanced analytic function from the **Analytics** pane to a visualization.

Add Reference Lines to Visualizations

You can use advanced analytics reference lines to identify the range of data element values in a visualization.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the Data Panel, click the **Analytics** icon ↗.
3. Drag and drop **Reference Line** into a visualization. Alternatively, you can double-click **Reference Line** to add it to the selected visualization.
4. In the properties pane select the **Analytics** tab.
5. In the **Method** row, click the value to select **Line** or **Band**.
6. Click the current reference **Function** and select the value that you want to use.
   - When you choose the **Line** method, select a reference function, and enter appropriate values if required.
   - When you choose the **Band** method, select one or more reference functions, and enter appropriate values if required.
7. Click **Save**.

Based on the selected **Method** or reference **Function**, a line is displayed in the visualization to highlight the value.

Use Spark Charts to Examine Trends

You can add a spark chart to a tile visualization to view aggregate data trends over time.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click **Visualize**.
3. In the Data pane, drag an aggregate data element to the canvas to create a tile visualization. For example, select **Revenue**.
4. In the Data pane drag a time-based data element to **Category (Chart)** to add a spark chart to the tile visualization. For example, select and drag **Month**.

   The red dot on the spark chart line shows the lowest value, and the green dot shows the highest value. The tooltip shows the first, last, lowest, highest and average aggregate values for the selected category. Hover your cursor anywhere over the spark chart to display the tooltip.
5. If you want to add a filter, drag a data element from the Data pane to **Click here or drag data to add a filter**.

   For example, **Year**.
   The filter changes the spark line to show the trend for the filtered data.
6. If you want to change spark chart display settings, click General in the Properties pane.
   - Click **Chart** to display one of the following spark charts, Line, Bar, or Area.
   - Click **Reference Line** to display or remove a reference line. Use the **Average** option to display a reference line showing the average trend. Use the **None** option to remove the reference line.

- Click **High/Low Marks** to hide or show the high and low marks, displayed as a green and a red dot.

7. Click **Save**.

Create Calculated Data Elements in a Workbook

You can create a new data element (typically a measure) to add to your visualization. For example, you can create a new measure called Profit that uses the Revenue and Discount Amount measures.

The calculated data elements are stored in the dataset's My Calculations folder and not in the workbook. In a workbook with a single dataset only one My Calculations folder is available and the new calculated data elements are added to it. In a workbook with multiple datasets My Calculations folder is available for each set of joined and not-joined datasets. Ensure that you're creating the calculated data elements for the required dataset or joined dataset. The new calculated data elements are added to the My Calculations folder of the datasets (joined and non-joined) that you create the calculation for.

If your workbook doesn't have a visualization, create one. See Begin to Build a Workbook and Create Visualizations.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.

   You'll land in the Visualize pane,

2. Select a canvas to work in.
3. Navigate to the bottom of the Data Panel, right-click My Calculations, and click **Add Calculation** to open the New Calculation dialog.
4. Enter a name.
5. In the expression builder pane, compose and edit an expression. See About Composing Expressions.

   You can drag and drop a column into the expression builder pane only if the column is joined to the dataset.

6. Click **Validate**.
7. Click **Save**.

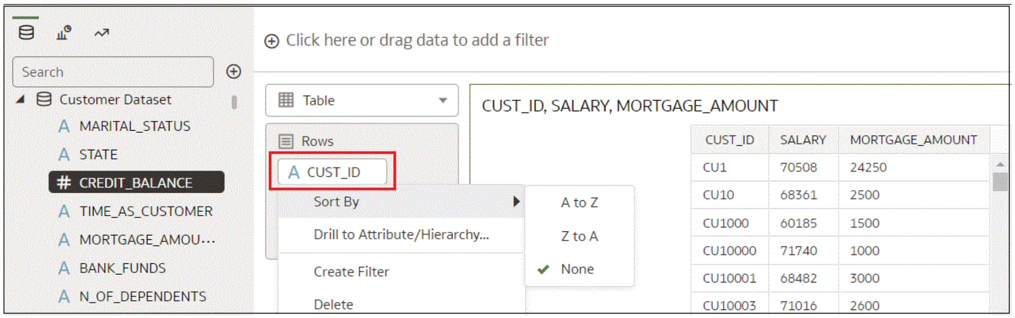
Sort Data in Visualizations

Sometimes you're working with a lot of data in visualizations. To optimize your view of that data in the workbook's visualization canvas, you need to sort it.

If your workbook doesn't have a visualization, create one. See Begin to Build a Workbook and Create Visualizations.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the Visualize canvas, do one of the following:
   - In the main editing panel, right click the data element you want to sort, hover over **Sort By** next to the element label, and select a sorting option (for example, Low to High, High to Low, None).

- In the Visualization Grammar Panel, right click the data element you want to sort, hover over **Sort By** then click a sorting option (for example, Low to High, A to Z, None).



CUST_ID, SALARY, MORTGAGE_AMOUNT

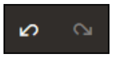
CUST_ID	SALARY	MORTGAGE_AMOUNT
CU1	70508	24250
CU10	68361	2500
CU1000	60185	1500
CU10000	71740	1000
CU10001	68482	3000
CU10003	71016	2600

Undo and Redo Edits

You can quickly undo your last action and then redo it if you change your mind. For example, you can try a different visualization type when you don't like the one you've just selected, or you can go back to where you were before you drilled into the data.

The undo and redo options are useful as you experiment with different visualizations. You can undo all the edits you've made since you last saved a workbook. However, in some cases, you can't undo and then redo an edit. For example, in the Create Dataset page, you've selected an analysis from an Oracle Application data source to use as a dataset in the workbook. In the next step, if you use the undo option to remove the dataset, you can't redo this change.

- To undo or redo an edit, go to the toolbar for the workbook or the dataset and click **Undo Last Edit** or **Redo Last Edit**. You can use these options only if you haven't saved the workbook since making the changes.



- When you're working on a workbook, click **Menu** on the workbook toolbar and select **Revert to Saved** to undo all the edits you've made since you last saved your workbook. **Revert to Saved** is enabled after you've saved the workbook for the first time. This option is automatically disabled if you select the **Auto Save** option.

About Refreshing a Workbook's Data

Refreshing the data in a workbook ensures that it uses current data.

When you click the **Refresh Data** button for a workbook, it runs the queries for the visualizations on the canvas. How the datasets are configured and the data sources they use determines where the canvases get their refreshed data.

If a dataset table's access mode is set to **Live**, then the table bypasses any presentation caches to fetch new data from the source. If a datasets table's access is set to **Automatic Caching**, then the table re-queries the cached data.

Refreshing data doesn't trigger a data cache reload. Sometimes the cached data is stale, so if you refresh the data and the data isn't the most current, then the datasets may need to be reloaded. See Reload a Dataset's Data.

Oracle recommends that you refresh a workbook's dataset rather than replace it. Replacing a dataset can be destructive. Don’t replace a dataset unless you understand what can happen:

- If the columns and data types don't match between the existing dataset and the new dataset, then replacing a dataset breaks the workbooks that use the existing dataset.
- Any modified or added columns in the existing dataset are lost and workbooks using the dataset are likely to break.

Refresh a Workbook's Data

Refresh the data in a workbook to ensure that your visualizations contain the most current data.

See About Refreshing a Workbook's Data.

You can set the **Refresh Data when Canvas Is Opened**, **Auto Refresh Data**, and **Refresh Interval** properties to automatically refresh the canvas' data. See About Canvas Data Refresh Properties.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the Visualize canvas, go to the workbook toolbar and click **Refresh Data**.

Pause Data Queries in a Workbook

You can disable the Auto Apply Data option to pause issuing new queries as you change visualization content in a workbook.

You can quickly configure a visualization without having to wait for data updates after each change.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click **Visualize**.
3. Click the **Auto Apply Data** button to pause data queries.

   Data queries are temporarily disabled.
4. Make changes to a visualization in the workbook.

   Your changes are displayed but data isn't updated, and a bubble is displayed indicating the number of data requests that have been skipped.
5. Click the bubble if you want to refresh the data now.

   Data queries are still disabled.
6. Click the **Auto Apply Data** button to re-enable data queries.

Work with Canvas Properties

You can work with canvas properties, such as layout, width, height, synchronizing visualizations on a canvas, and refreshing data.

Topics:

- Update Canvas Properties
- About Canvas Layout Properties
- About Canvas Data Refresh Properties
- About Synchronizing Visualizations in a Canvas

Update Canvas Properties

You can use the canvas properties to perform many different tasks. For example, change the layout, set brushing, or configure canvas refresh settings.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Right-click a canvas tab and click **Canvas Properties**.
3. Update the canvas properties.
4. Click **Save**.

About Canvas Data Refresh Properties

If you have administration privileges, you can configure canvas data to refresh when you open a canvas, or to refresh automatically at a specified interval.

You configure how canvas data is refreshed using canvas property settings.

Refresh Data when canvas is opened

Configures whether to refresh data when a user opens the canvas.

- **On** - Use to refresh the data when the canvas is opened.
  The client and server caches are cleared every time the canvas is opened, ensuring that you see the current data.
- **Off** - Use to not refresh the data when the canvas is opened.

Auto Refresh Data

Configures whether to automatically refresh the canvas data at specified time intervals.

- **Enabled** - Use to automatically refresh the canvas data using a **Refresh Interval** that you enter.
- **Disabled** - Use to not automatically refresh the canvas data.

Auto start for viewers

Configures whether to start the automatic data refresh when the canvas is opened.

- **On** - Use to automatically refresh canvas data when the canvas is opened, and then at the specified intervals.
- **Off** - Use to not automatically start the canvas data at the specified refresh interval when the canvas is opened, until the user clicks **Refresh Data** in the workbook.

About Canvas Layout Properties

You can configure the layout, width and height of visualizations on a canvas.

You configure how visualizations are displayed in a canvas using canvas property settings.

Layout

Configures whether visualizations are displayed as freeform or auto fit.

- **Auto Fit** - Use to automatically arrange or correctly align visualizations on a canvas with multiple visualizations. You can also resize a visualization by dragging its edges to the appropriate dimensions.
- **Freeform** - Use to rearrange a visualization on the canvas using drag and drop to a space between visualizations where you want to place it. You can also resize a visualization by dragging its edges to the appropriate dimensions.
  The **Order Visualization** option (only available when you use the **Freeform** canvas layout) enables you to use **Bring to Front**, **Bring Forward**, **Send Backward**, or **Send to Back**, to move a visualization on a canvas with multiple visualizations.

Width and Height

Use **Width**, and **Height** settings in conjunction with **Layout** settings to specify the layout size of visualizations in a canvas

- **Screen** - Layout fits in the available screen space.
  Not available when you select **Freeform**.
- **Grow** - Layout automatically grows in height or width to accommodate a visualization at its optimal size.
  For example, if you add multiple visualizations, or rows of visualizations to a canvas, the layout grows so that the added visualizations display at their optimum size.
- **Fixed** - Layout uses the specified size.

About Synchronizing Visualizations in a Canvas

You can specify whether or not to synchronize Visualizations in a canvas.

You use the **Synchronize Visualizations** setting to specify how the visualizations on your canvas interact. By default, visualizations are linked for automatic synchronization.

When **Synchronize Visualizations** is set to **On**, and you add or remove data values from a visualization, this automatically adds or removes the values in corresponding visualizations. When **Synchronize Visualizations** is set to **Off**, this unlinks your visualizations and turns off automatic synchronization.

When **Synchronize Visualizations** is **On**, then all filters on the filter bar and actions that create filters (such as Drill) apply to:

- All the visualizations in a canvas with a single dataset.
- All the visualizations of joined datasets with multiple datasets.

  If a data element from a dataset is specified as a filter but isn't matched with the joined datasets, then the filter only applies to the visualization of the dataset that it was specified for.

When you hover the mouse pointer over a visualization to see the filters applied to the visualization, any filter that isn't applied to the visualization is grayed out.

Any visualization-level filters are applied only to the visualization.

When **Synchronize Visualizations** is set to **Off**, then analytic actions such as Drill affect only the visualization to which you applied the action.

Copy and Paste a Visualization or Canvas

You can copy and paste a visualization or canvas within the same workbook, to another open workbook, or to another workbook open in a different browser tab.

When you copy a visualization or canvas from one workbook to another, Oracle Analytics does the following:

- Data - The dataset for the pasted visualization or canvas is added to the target workbook. When you open or create the target workbook that you're pasting to, it doesn't need to include the dataset used by the visualization or canvas that you'll copy and paste.
- Filters - The filters in the target workbook and in the pasted visualization or canvas are maintained. You don't need to add the visualization or canvas filters to the target workbook. If there's a conflict between the target workbook and the pasted visualization or canvas filters, then the pasted filters won't overwrite the target's filters.
- Color assignments - The color scheme of the target workbook is applied to the pasted visualization or canvas.
- Calculations - If the same calculation name exists in the target workbook, then the pasted calculation is added and renamed.

Use the following steps to copy and paste a visualization or canvas:

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the Visualize canvas, copy a visualization or canvas.
   - To copy a visualization, click its **Menu**, hover over **Edit**, and then click **Copy Visualization**.
   - To copy a canvas, right-click it and click **Copy Canvas**.
3. Navigate to a visualization or canvas and paste the object.
   - To paste a visualization into a canvas that contains visualizations, click an existing visualization's **Menu**, hover over **Edit**, and then click **Paste Visualization**.
   - To paste a visualization into a blank canvas, right-click the canvas bar and select **Add Canvas**. Right-click the new canvas, hover over **Edit**, and then click **Paste Visualization**.
   - To paste a canvas, right-click the canvas bar and then click **Paste Canvas**.

Copy and Paste a Visualization's Data

You can copy all of a visualization's data to the clipboard and then paste the data to an another application, like Word or Excel.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click a visualization's **Menu**, hover over **Edit**, and then click **Copy Data**.

   This copies the visualization data to the clipboard.
3. Open a target application and paste in the visualization's data.

Work with Multiple Visualizations on a Canvas

You can work with multiple visualizations at the same time, update common visualization properties, copy and paste multiple visualizations, and delete multiple visualizations on a canvas.

Topics:

- Update Common Properties for Multiple Visualizations on a Canvas
- Copy and Paste Multiple Visualizations on a Canvas
- Delete Multiple Visualizations on a Canvas

Update Common Properties for Multiple Visualizations on a Canvas

You can update visualization properties for multiple selected visualizations on a canvas.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the Visualize canvas, use Ctrl-click to select multiple visualizations on the canvas.
3. In **Common Properties** click the property to change and apply your changes.

   Common property values are displayed when they share the same value for each of the selected visualizations.
4. Click **Save**.

Copy and Paste Multiple Visualizations on a Canvas

You can copy and paste multiple selected visualizations on a canvas.

1. On the Home page, select a workbook that contains the visualizations you want to copy and paste, click the **Actions menu**, and then select **Open**.
2. In the Visualize canvas, use Ctrl-click to select multiple visualizations on the canvas.
3. To copy the selected visualizations hover over the selected visualizations, use right-click, click **Edit**, and then click **Copy Visualizations**.

4. To paste the copied multiple visualizations, click a canvas location where you want to paste the copied visualizations, use Right-click and click **Paste Visualizations**.
5. Click **Save**.

Delete Multiple Visualizations on a Canvas

You can delete multiple selected visualizations on a canvas.

1. On the Home page, select a workbook that contains the visualizations you want to delete, click the **Actions menu**, and then select **Open**.
2. In the Visualize canvas, use Ctrl-click to select multiple visualizations.
3. Right-click, and select **Delete Visualizations**.
4. Click **Save**.

Change Visualization Types

You can change visualization types to best suit the data you’re exploring.

When you create a workbook and add data elements to the canvas, Auto Visualization mode chooses the most appropriate visualization type based on the selected data element. The Auto Visualization mode is on (selected) by default. If you add more data elements, the visualization type is automatically updated, and the best type is selected based on the data elements.

If you want to use a different visualization type, then you need to select it from the visualization type list. When you change the visualization type, Auto Visualization mode is turned off. When the Auto Visualization mode is off (deselected), adding more data elements to the canvas won’t change the visualization type automatically.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Select a visualization on the canvas, and on the visualization toolbar, click **Change Visualization Type**.
3. Select a visualization type. For example, select Treemap to change the visualization type from Pivot to Treemap.

   When you change the visualization type, the data elements are moved to matching drop target names. If an equivalent drop target doesn’t exist for the new visualization type, then the data elements are moved to a Grammar Panel section labeled **Unused**. You can then move them to the Grammar Panel section you prefer.

Adjust Visualization Properties

You can customize how the visualizations in your workbook look and function. For example, the legend's location, number format, and font used in titles and labels.

The tabs and fields displayed depend on the type of visualization that you're working with.

If your workbook doesn't have a visualization, create one. See Begin to Build a Workbook and Create Visualizations.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the Visualize canvas, select a visualization.

The visualization's properties display in the Properties pane.

3. Use the tabs in the Properties pane to adjust the visualization's properties as needed.

Visualize Properties

Use the Visualize properties to see the functions of each tab.

Properties Pane Tab	Description
General	Change the title display settings. • Auto - Use the default title. • Custom - Enter your own title text. • None - Don't display a title. Change the title tooltip display settings. • Auto - Use the default title. • Custom - Enter your own tooltip text. • Use Title in Tooltip - Use the default title tooltip along with your custom text. • None - Don't display a tooltip. Change the title font, size, and color. Add a custom background image or fill color, and adjust its transparency. Format the visualization legend. Change the alignment of a pivot or a table. • Start - Align to the left. • Center - Align in the middle. • End - Align to the right.
Axis	Turn grid lines on or off. Set the axis labels for horizontal and vertical values and the axis values to start and end. Display or hide the axis labels. Change the axis label text. Change the label text's font, size, and color.
Edge Labels	Show or hide a table visualization's headers. Change the table's header, data, and total text's font, size, and color. Add or remove column totals. Wrap column labels. Change a column text's font, size, and color.
Action	Add URLs to Tile, Image, and Text Box visualizations.
Values	Change the display and placement of the data labels. Change the labels' font, size, and color. Change numbers to display as currency or percentage. Change the aggregation method. Display a trellis row (Y2 axis).
Analytics	Add clusters or outliers to the visualization. Add reference lines, trend lines, and bands to display at the minimum or maximum values of a measure included in the visualization.
Filters	Change and format the visualization's filter title and selection names. Change the font, size, and color.

Highlight Important Data Events with Conditional Formatting

Use conditional formatting to highlight important data events in your visualizations so that you can take action.

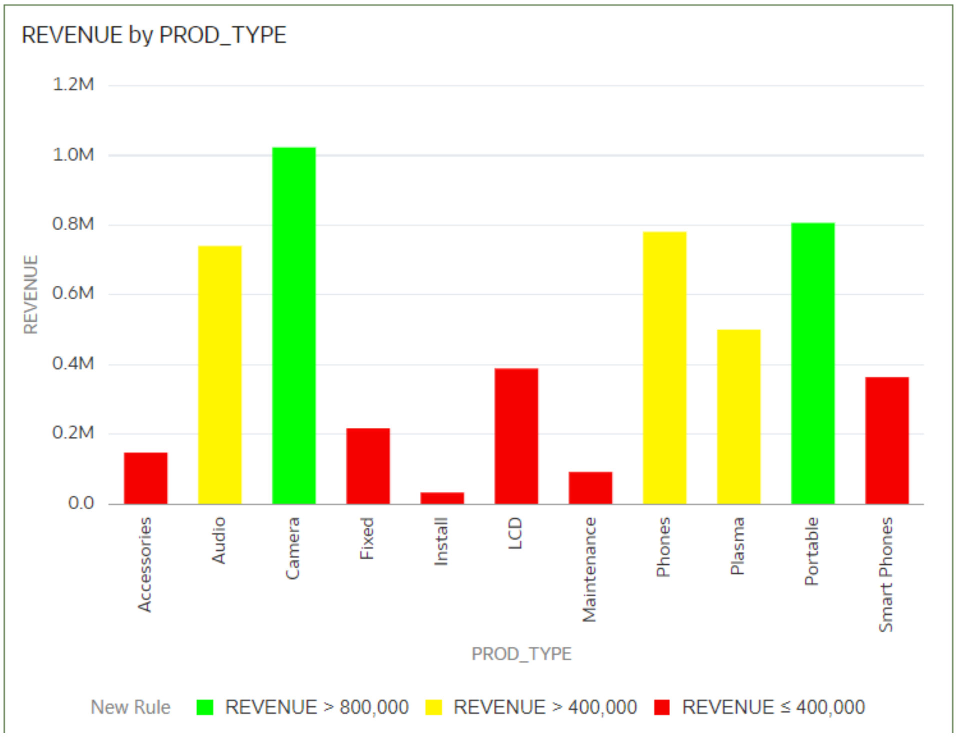
Video

Topics:

- What Can I Do With Conditional Formatting?
- Apply Conditional Formatting to a Visualization
- Example - Compare a Measure to a Set of Thresholds
- Example - Compare a Measure to a Target or Goal
- Example - Compare a Measure to a Complex Expression Value
- Example - Compare a Measure to a Percentage of a Value

What Can I Do With Conditional Formatting?

With conditional formatting, you apply rules to your data to highlight when something important happens. For example, you might want to show when revenues meet high, medium, and low thresholds.



You can make conditional formatting rules available to one or multiple visualizations in a workbook.

Conditional formatting provides a way for the user to see events or changes in their data driven rules. For example, if the user wants to see when sales fall below a threshold, the user would write a rule that says when sales are below 500K, format the data points using the color red.

In each canvas, you can:

- Choose which rules to apply to the data in the canvas.
- Apply multiple rules to a canvas at the same time.
- Apply multiple rules to a measure at the same time.
- Change the order in which rules are applied.
- Turn rules on and off for a visualization.

You can compare a measure to:

- A set of thresholds.
  For example, highlight values in red if my blood pressure is above 90 or under 70.
- A target or goal.
  For example, highlight values in red if my costs exceed my budget.
- A percentage of a target.
  For example, highlight values in green when I reach 80% of my sales goal.
- A complex expression.
  For example, highlight values in green when I achieve 5% growth in sales compared to the same period last year.

You can format:

- Fill color and color density.
- Font, font size, font color, and font style.

You can also:

- Add tooltips and legends.
- Deploy conditional formatting in the Oracle Analytics app.

Apply Conditional Formatting to a Visualization

Apply conditional formatting to a visualization to highlight important events in your data. You can apply existing conditional formatting rules or create your own rules.

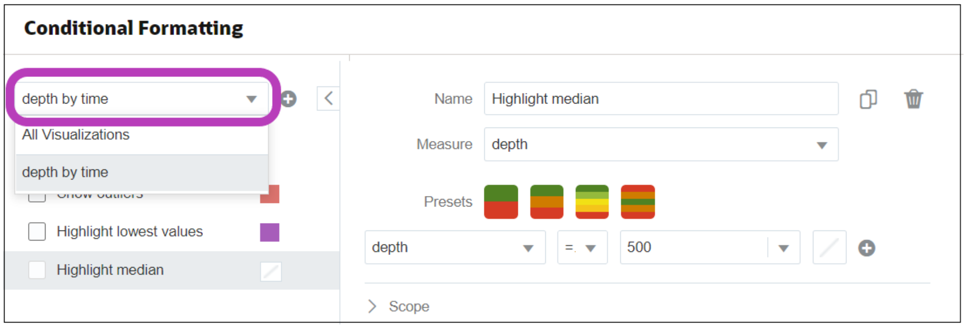
Video

Tutorial

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click **Visualize**.
3. From the visualization toolbar, click **Manage Conditional Formatting Rules**.

   Available formatting rules are displayed at the left side (not selected by default).
4. To highlight data by applying an existing rule to your visualization:

a. If you want to display existing rules available to all visualizations or just the current measure, click the drop-down list in the top left hand corner, and select either **All Visualizations** or the name of the current visualization.



Conditional Formatting

Name: Highlight median

Measure: depth

depth = 500

If you select **All Visualizations** and the workbook doesn't have any rules in that category, you won't be able to select any rules. Select the name of the current visualization to display rules that you can apply.

b. Click the check box next to a rule to apply that formatting rule to your visualization. Activated rules have a green tick.

If a rule is displayed without a check box, click the drop-down list above the rules list and select a measure to display rules that you can use.

c. If you're applying multiple rules to the same measure, use the grab handles to position the rules in the order you want (rules at the top activate first).

If you're applying multiple rules to a measure, the last rule that evaluates to true is the one that colors the item. For example, if rule 1 is revenue greater than one million and highlights in green, and rule 2 is revenue less than target and highlights in red, an item where revenue meets both criteria will highlight in red.

5. To highlight data by applying a new rule to your visualization:

a. Optional: Change the default name in the **Name** field to a more user-friendly term.

b. In **Measure**, specify the data measure that the rule evaluates.

For example, to highlight sales channels where revenue is greater than one million, you might select Revenue.

c. Use the operator field and adjacent value field to specify the threshold.

For example, to highlight sales channels where revenue is greater than one million, select the greater than symbol **>** in the operator field and enter 1,000,000 in the value field.

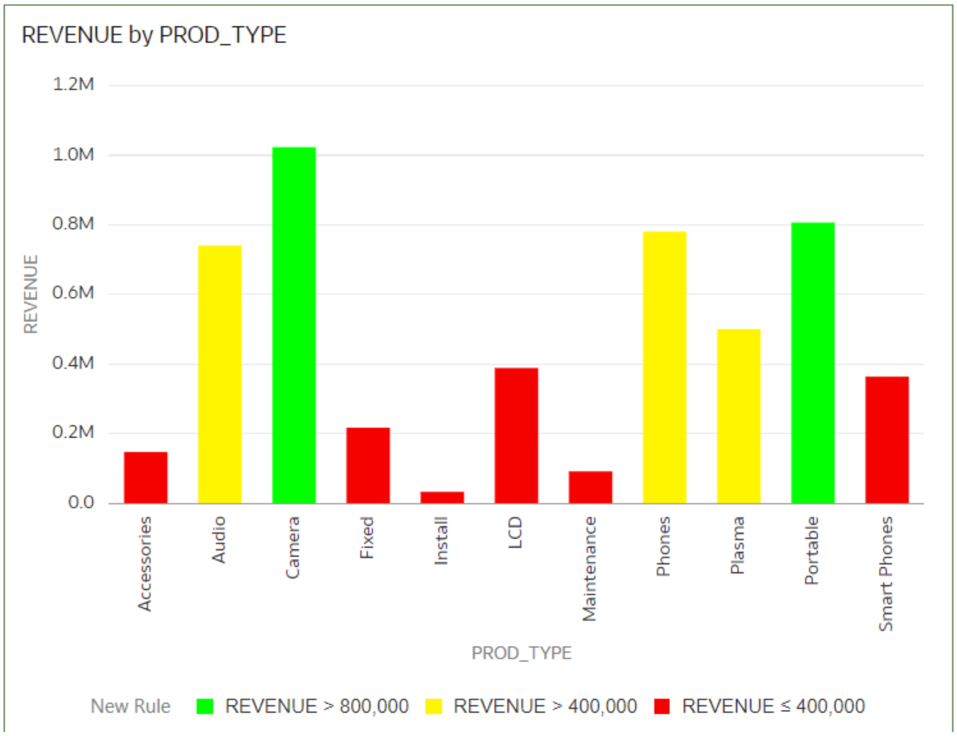
You can also create a rule by clicking one of the **Presets** and defining a value for each threshold. For example, click **3 Steps Threshold** to create a set of stoplight thresholds.

d. Click **Save**.

Example - Compare a Measure to a Set of Thresholds

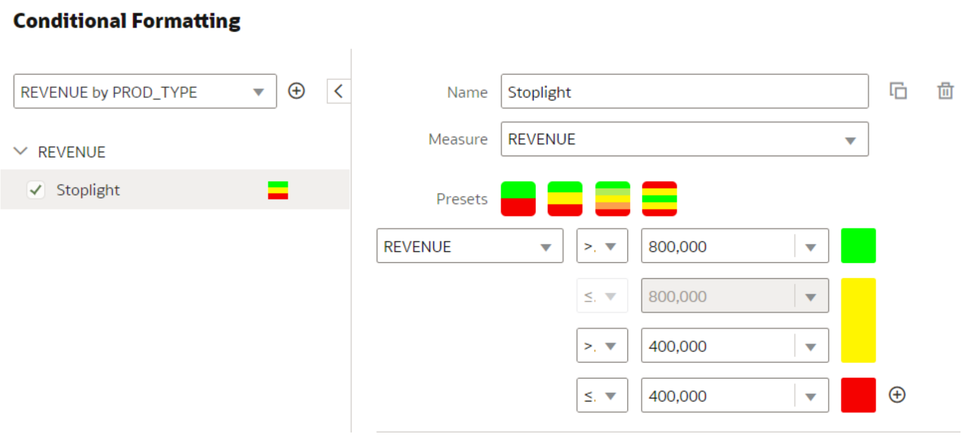
This example shows how to use conditional formatting to compare a measure to low, medium, and high thresholds, also known as stoplight formatting.

In this example you highlight the revenue level for product types such as audio, phones, and accessories. You show the revenues over 800,000 in green, revenues between 400,000 and 800,000 in amber, and revenues below 400,000 in red.



Here's how to configure this example using the Conditional Formatting dialog.

- In the **Name** field, enter Stoplight.
- In the **Measure** field, select Revenue.
- In **Presets**, click **3 Steps Threshold** to display a three-step template.
- In the first step, select the greater than symbol **>** and enter 800,000 as the value.
- In the second step, select the greater than symbol **>** and enter 400,000 as the value.



Conditional Formatting

REVENUE by PROD_TYPE

REVENUE

☑ Stoplight

Name: Stoplight

Measure: REVENUE

Presets

REVENUE > 800,000 🟩

≤ 800,000 🟨

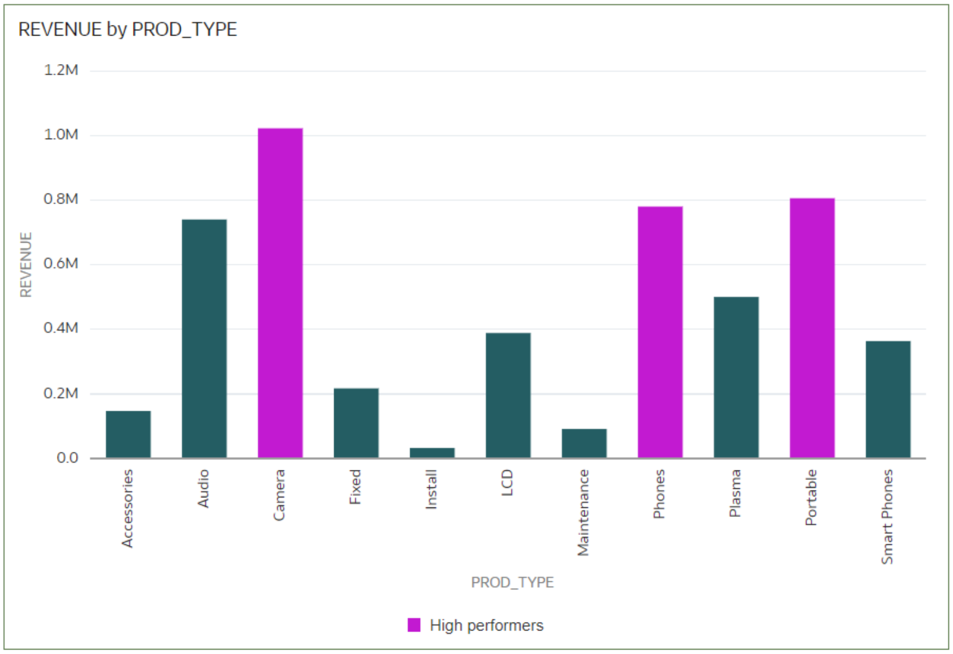
> 400,000 🟨

≤ 400,000 🟥

Example - Compare a Measure to a Target or Goal

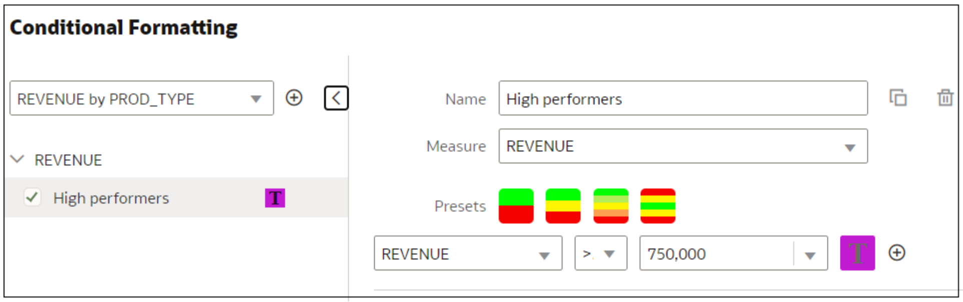
This example shows how to use conditional formatting to compare a measure to a target or goal.

In this example you highlight product types such as audio, phones, and accessories with a revenue of more than 750,000.



Here's how to configure this example using the Conditional Formatting dialog.

- In the **Name** field, enter High performers.
- In the **Measure** field, select Revenue.
- In the operator list, select the greater than symbol **>**, and in the value box, enter 750,000.
- Click **Format** to display the color picker and select a purple shade.



Conditional Formatting

REVENUE by PROD_TYPE

REVENUE

✓ High performers

Name: High performers

Measure: REVENUE

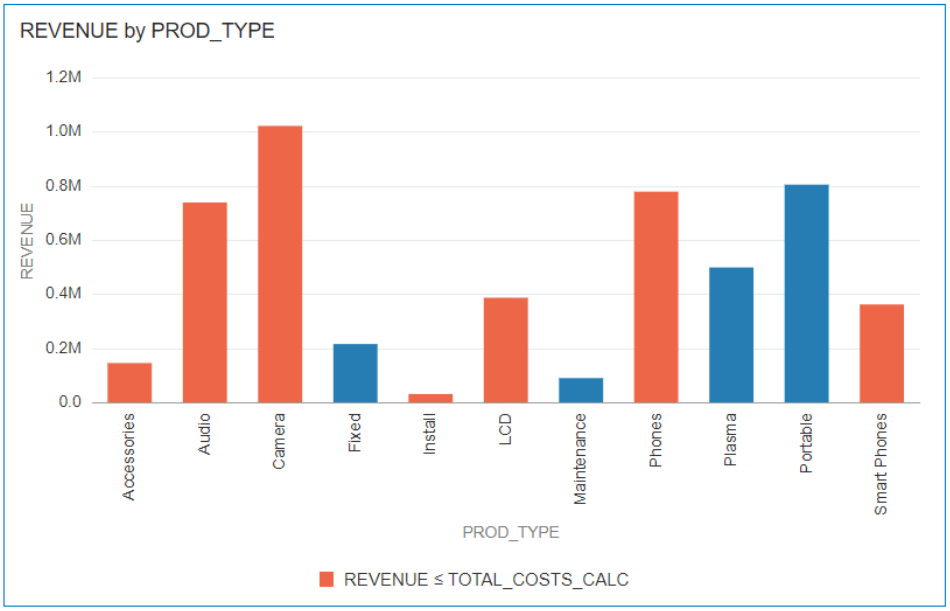
Presets

REVENUE > 750,000

Example - Compare a Measure to a Complex Expression Value

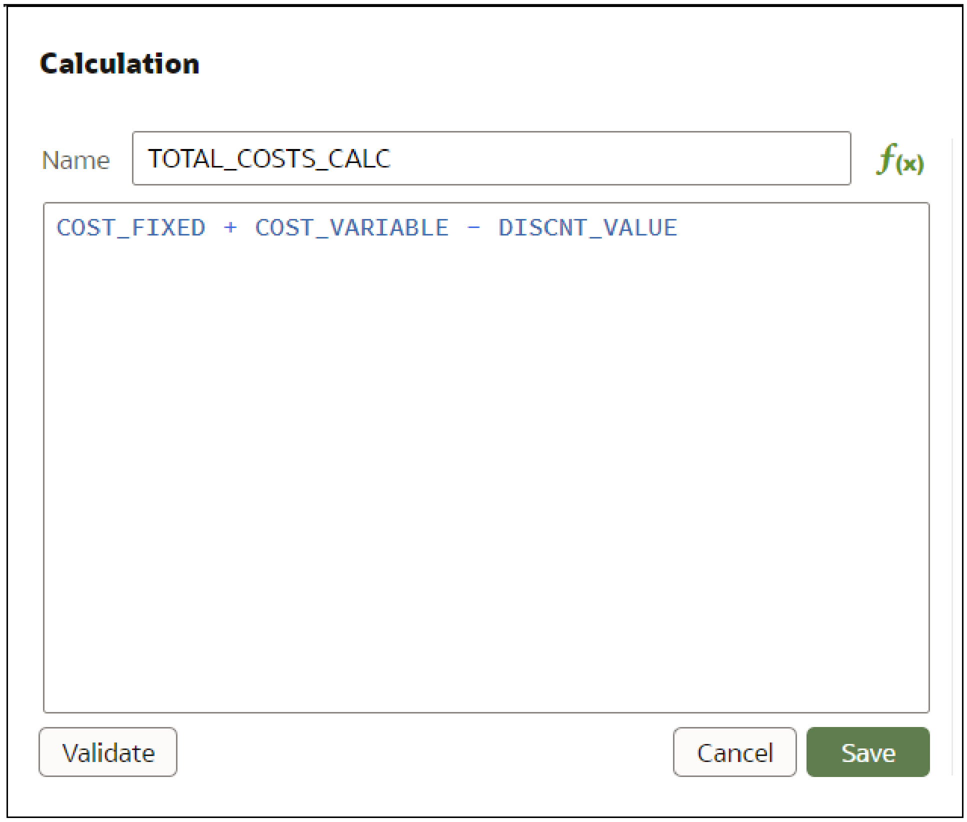
This example shows how to use conditional formatting to compare a measure to a value calculated by an expression.

In this example, you compare revenue to total costs, which you calculate using an expression that sums fixed costs, variable costs, and discount value.



Here's how to configure this example using the Conditional Formatting dialog.

- In the **Name** field, enter Compare revenue to costs.
- In the **Measure** field, select Revenue.
- In the operator list, select the equal to or less than symbol ≤.
- In the value box, click the down arrow, then click **f(x)** to display the calculation editor.
- In the **Name** field, specify TOTAL_COSTS_CALC, and in the calculation field specify `COST_FIXED + COST_VARIABLE + DISCNT_VALUE`.

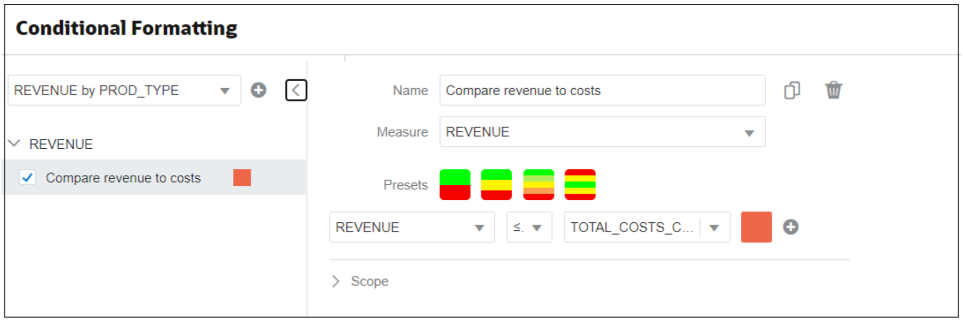


Calculation

Name: TOTAL_COSTS_CALC

```
COST_FIXED + COST_VARIABLE - DISCNT_VALUE
```

- Validate the expression and click **Save**.
- Click **Format** to display the color picker and select a red shade.



Conditional Formatting

REVENUE by PROD_TYPE

REVENUE

☑ Compare revenue to costs 🟥

Name: Compare revenue to costs

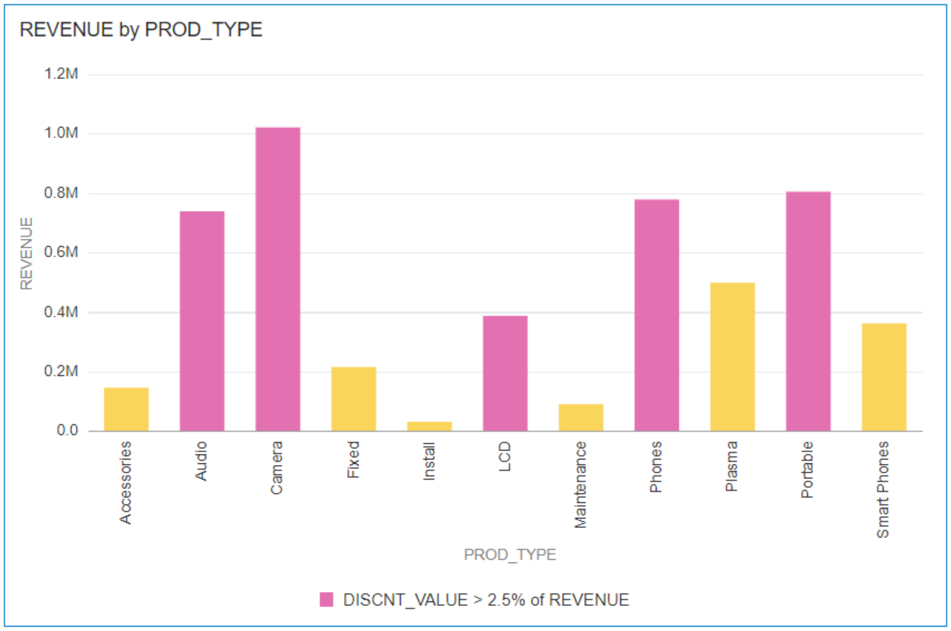
Measure: REVENUE

REVENUE ≤ TOTAL_COSTS_C... 🟥

Example - Compare a Measure to a Percentage of a Value

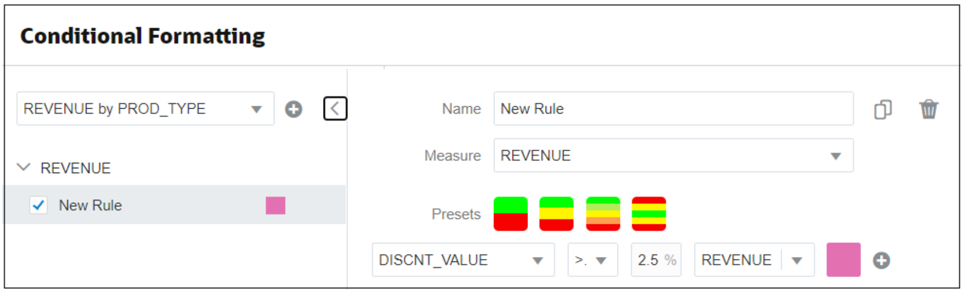
This example shows how to use conditional formatting to compare a measure to a percentage of a value.

In this example you highlight product types such as audio, phones, and accessories where the discount value (stored in DISCNT_VALUE) is greater than 2.5% of revenue.



Here's how to configure this example using the Conditional Formatting dialog.

- In the **Measure** field, select Revenue.
- In the drop down list below **Presets** select DISCNT_VALUE.
- In the operator drop down list, select the greater than symbol '>', in the value box enter '2.5', and click **%**
- Click **Format** to display the color picker and select a purple shade.


Conditional Formatting

REVENUE by PROD_TYPE

REVENUE

☑ New Rule

Name: New Rule

Measure: REVENUE

Presets

DISCNT_VALUE >. 2.5 % REVENUE

Apply Color to Visualizations

Use color to enhance visualizations.

Topics:

- About Color Assignments in Visualizations
- Access Color Options
- Change the Color Palette
- Assign Colors to Columns

About Color Assignments in Visualizations

You can work with color to make visualizations more attractive, dynamic, and informative. You can color a series of measure values (for example, Sales or Forecasted Sales) or a series of attribute values (for example, Product and Brand).

Your color choices are shared across all visualizations on the canvas, so if you change the series or data point color in one visualization, then it appears on the other visualizations.

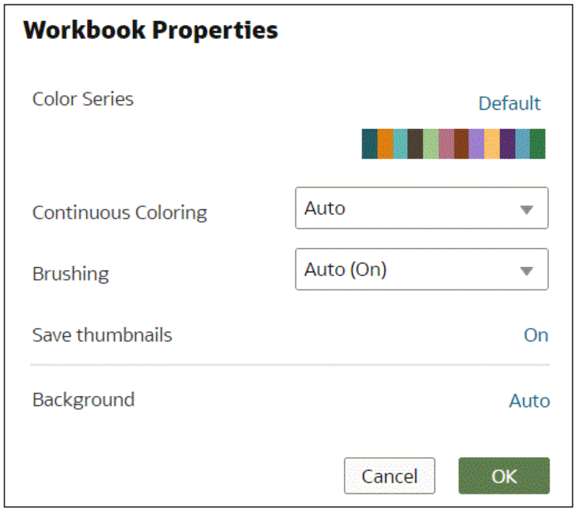
The **Visualize** canvas has a Color section in the Grammar Panel where you can put a measure column, attribute column, or set of attributes columns. The canvas assigns color to the columns that are included in the Color section:

- When a measure is in the Color section, then you can select different measure range types (for example, single color, two color, and three color) and specify advanced measure range options (for example, reverse, number of steps, and midpoint).
- When you've one attribute in the Color section, then the stretch palette is used by default. Color palettes contain a set number of colors (for example, 12 colors), and those colors repeat in the visualization. The stretch palette extends the colors in the palette so that each value has a unique color shade.
- If you've multiple attributes in the Color section, then the hierarchical palette is used by default, but you can choose to use the stretch palette, instead. The hierarchical palette assigns colors to groups of related values. For example, if the attributes in the Color section are Product and Brand and you've selected Hierarchical Palette, then in your visualization, each brand has its own color, and within that color, each product has its own shade.

Access Color Options

You can set color options for your workbook and for individual visualizations within your workbook.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. If you want to edit color options for the whole workbook:
   a. Click **Menu** on the workbook toolbar and select **Workbook Properties**.
   b. Use the **General** tab to edit the color series or continuous coloring.
3. If you want to edit color options for a visualization.
   a. Select the visualization and click **Menu** or right-click.
   b. Select **Color**. The available color options depend on how the measures and attributes are set up in your visualization.



Color Series: Default

Continuous Coloring: Auto

Brushing: Auto (On)

Save thumbnails: On

Background: Auto

c. You can experiment with visualization colors and select **Reset Visualization Colors** to revert to the original colors.

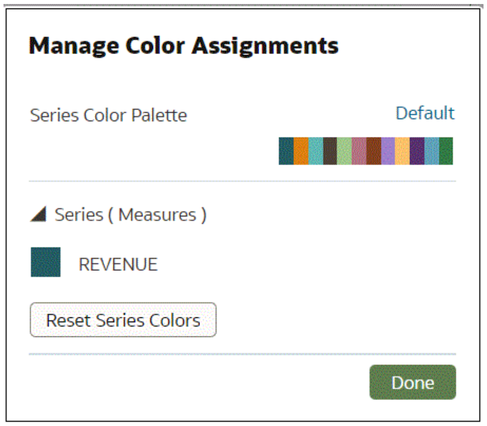
d. Select **Stretch Palette** to turn this option on or off.
Color palettes have a set number of colors, and if your visualization contains more values than the number of color values, then the palette colors are repeated. Use the Stretch Palette option to expand the number of colors in the palette. Stretch coloring adds light and dark shades of the palette colors to give each value a unique color. For some visualizations, stretch coloring is used by default.

Change the Color Palette

You can switch between the various color palettes until you find the one you want.

Each color palette contains 12 colors that you can apply to a visualization.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Select a visualization to change the color palette for.
3. Click **Menu** or right-click and select **Color**, then select **Manage Assignments**.
4. Locate the **Series Color Palette** and click the color palette that's currently used in the visualization (for example, Default or Alta).



Manage Color Assignments

Series Color Palette Default

◢ Series (Measures)

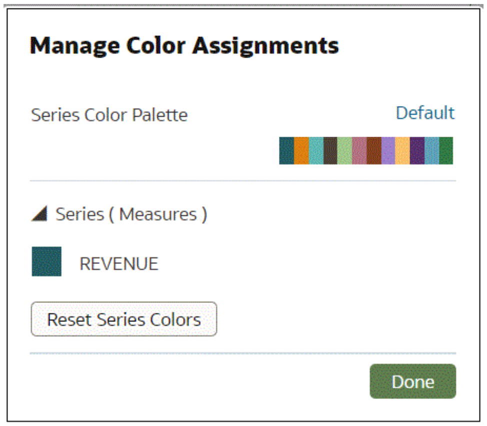
REVENUE

5. From the list, select the color palette that you want to apply to the visualization.

Assign Colors to Columns

Instead of using the palette's default colors, you can choose specific colors to fine-tune the look of your visualizations.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Select the visualization that you want to manage the colors for.
3. Click **Menu** on the visualization toolbar or right-click and select **Color**, then select **Manage Assignments**.
4. If you're working with a measure column, you can do the following:
   - Click the box containing the color assigned to the measure. From the color picker dialog, select the color that you want to assign to the measure. Click **OK**.
   - Specify how you want the color range to be displayed for the measure (for example, reverse the color range, pick a different color range, and specify how many shades you want in the color range).



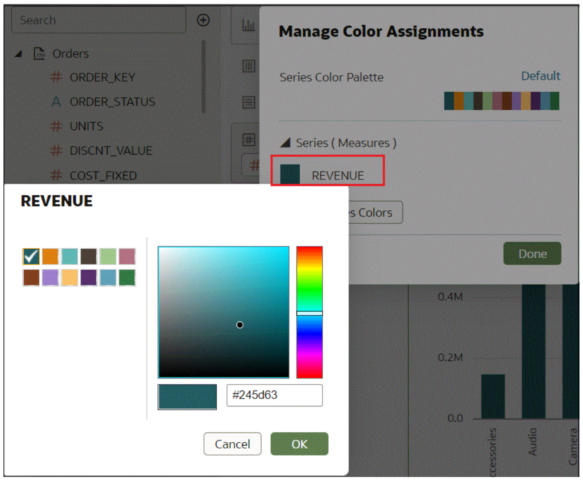
Manage Color Assignments

Series Color Palette Default

◢ Series (Measures)

REVENUE

5. If you're working with an attribute column, then click the box containing the color assignment that you want to change. From the color picker dialog, select the color that you want to assign to the value. Click **OK**.



Format Numeric Values of Columns

You can format numeric values of a column in your visualizations using a wide range of ready-to-use formats. For example, you might change the aggregation type from Sum to Average.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the Data Panel, select the column.
3. In the properties pane for the selected column, use the **General** or **Number Format** tabs to change the numeric properties.
   - **General** - Change the column name, data type, treat as (measure or attribute), and aggregation type.
     For example, to change how a number is aggregated, use the **Aggregation** option.
   - **Number Format** - Change the default format of a number column.
4. Click **Save**.

Set Currency Symbols for Visualizations

You can set measure values in a visualization to display an appropriate currency symbol.

You can configure a measure to use a custom currency to display the symbol associated with a currency. For example, if you set a canvas filter to display a European Ledger, then the Euro symbol is displayed for each measure value that's associated with the custom currency property. The workbook data must contain a currency code column, for example, a Ledger Currency column. The currency code column enables the display of an appropriate currency symbol for the measure column, for example, a Profit column.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the Properties panel for a selected measure, click the **Values** tab, click **Number Format**, and select **Currency**.

3. In the **Currency** field, click the currently displayed value and select **Custom**.
4. In the **Custom** field, add the column that determines the currency code.
5. Click **Save**.

Add Notes to Visualizations

Annotate your visualizations with notes to call out areas of interest or emphasize specific data points.

Topics:

- Add Notes to a Visualization
- Connect a Note to Data Points in a Visualization
- Show or Hide a Note's Data Point Connectors
- Delete a Note's Data Connectors
- Show or Hide a Visualization's Notes

Add Notes to a Visualization

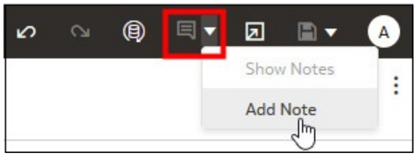
You can add notes to call out important information in a workbook's visualizations. You can use notes to annotate one or all of the visualizations in a canvas, or to emphasize specific data points in a visualization, such as columns in a table or clusters in a scatter plot.

There are many formatting options that you can use to customize a note's content. For example, you can choose font type, size, and color; add a bulleted or numbered list; and add URL links.

When you add a note, you can add data connectors to call out specific points in the visualization, or you can create the note and add or adjust the data point connectors later. See Connect a Note to Data Points in a Visualization.

By default, the notes that you add are displayed, but you can hide a visualization's notes. See Show or Hide a Visualization's Notes.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the workbook, click **Visualize**.
   - To add a note without data point connectors, click the **Notes** icon and select **Add Note**.
   - To add a note with data point connectors, go to the visualization where you want to add the note and hold down the Ctrl key and click up to ten data points that you want the note to connect to. Then click the **Notes** icon and select **Add Note**.



3. In the note box, enter the note text and use the formatting options to specify the note's font style, color, size, and so on.

4. Optional: To add a link to the note, highlight the note text that you want to make into a link. Click **Link** and in the Hyperlink dialog, enter a URL. Click **OK**.
5. Click **Save**.

Connect a Note to Data Points in a Visualization

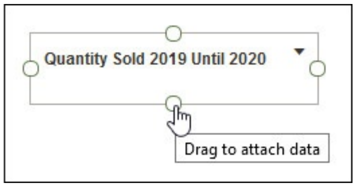
Add connectors to a note to identify specific data points in a visualization.

You can connect a note to up to ten data points on a visualization or canvas. You can’t connect a note to a data point on these visualization types:

- Chord Diagram
- Correlation Matrix
- List
- Map
- Parallel Coordinates
- Tile

By default, a note's data point connectors are displayed, but you can hide them. See Show or Hide a Note's Data Point Connectors.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Locate and hover over the note that you want to attach to data points. Click and hold a circle and drag the line to the data point that you want to connect the note to. You can connect a circle to multiple data points.

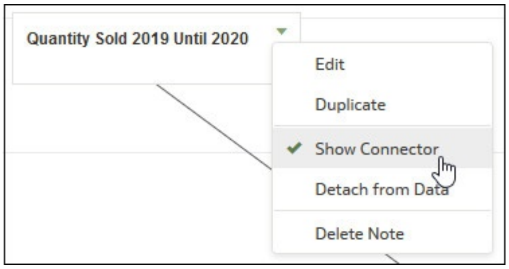


3. Click **Save**.

Show or Hide a Note's Data Point Connectors

By default a note's data point connectors are displayed, but you can hide or show them as needed.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Locate and hover over the note with the connectors to hide or show. Click **Click to edit** and then click **Show Connector**.



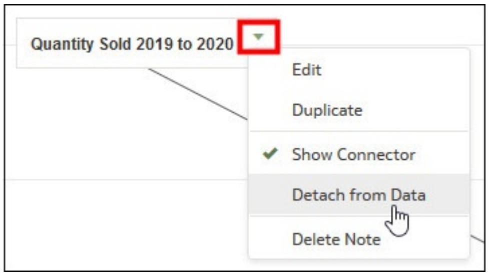
3. Click **Save**.

Delete a Note's Data Connectors

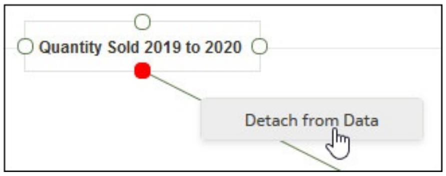
You can delete one or more of a note's data connectors.

Instead of deleting a note's data connectors, you can hide them. See Show or Hide a Note's Data Point Connectors.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Optional: To delete all connectors, locate and hover over a note with connectors. Click **Click to edit** and then click **Detach from Data**.



3. Optional: To delete an individual connector, hover over and right-click a connector, and then click **Detach from Data**.



4. Click **Save**.

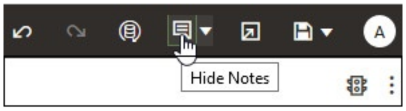
Show or Hide a Visualization's Notes

By default a workbook's visualization notes are displayed, but you can hide or show all notes.

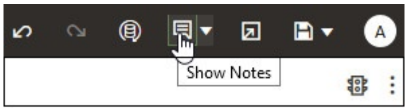
If you create a story from a workbook, then all of the included canvases' notes are displayed in the story even if you hide all notes in Visualize. In Present where you add canvases to create the story, you can use a canvas' **Notes** properties to show and hide individual notes. See Create a Story.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.

2. Optional: If the workbook's notes are displayed, then click **Hide Notes**.



3. Optional: If the workbook's notes aren't displayed, then click **Show Notes**.



4. Click **Save**.

Format Numeric Values of Visualizations

You can format numeric properties of a visualization using a wide range of ready-to-use formats. For example, you might change the aggregation type from Sum to Average.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click **Visualize** and select a visualization.
3. In the properties pane for the selected visualization, use the **Values** tab to change the numeric properties.

   For example, to change how a number is aggregated, use the **Aggregation Method** option.
4. Click **Save**.

Apply Map Backgrounds and Map Layers to Enhance Visualizations

You can use geographical information to enhance the analysis of your data.

Topics:

- About Map Backgrounds
- Enhance Visualizations with Map Backgrounds
- Use Different Map Backgrounds in a Workbook
- Interpret Data Values with Color and Size in Map Visualizations
- Add Custom Map Layers
- Update Custom Map Layers
- Apply Multiple Data Layers to a Single Map Visualization
- Use an Image as a Map Background and Draw Map Layer Shapes on the Image
- Assign a Map Layer to a Data Column

- Auto Focus on Data for a Map Visualization
- Review Location Matches for a Map Visualization
- Create Heatmap Layers on a Map Visualization
- Create Cluster Layers on a Map Visualization
- Represent Point Data With Custom Icons on a Map
- Select Points or Area on a Map
- Represent Line Data Using Size and Color on a Map
- Make Map Layers and Backgrounds Available to Users
- Use a Map Background as the Default
- Add Map Backgrounds

About Map Backgrounds

You can enhance map visualizations in workbooks by adding and maintaining map backgrounds.

You can apply the ready-to-use map backgrounds to a workbook. You can also add backgrounds from the available list of Web Map Service (WMS) providers such as Google Maps and Baidu Maps, or you can add a map background by specifying appropriate Web Map Service or Tiled Web Map details. Background maps from these providers offer details and language support (such as city or region name) that certain geographic regions (such as Asian countries) require.

You can modify backgrounds in the following ways:

- Modify the background parameters such as map type, format and API keys. The parameters are different for each WMS provider.
- Assign or change the default background in a workbook.
- Reverse the inherited default background settings in a workbook.

You can add a WMS provider and perform the following types of functions:

- Add the WMS map servers, and make them available as additional map background options.
- Select one or more map backgrounds available from the WMS provider.
- Assign an added WMS provider’s map as the default map background.

Enhance Visualizations with Map Backgrounds

You can use map backgrounds to enhance visualizations in a workbook.

Tutorial

Based on the column values, a specific set of dimensions and metrics is displayed on the map visualization. You'll see either the default map background or an existing Oracle map background if no default is set.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click **Visualize**.
3. To select a column and render it in a map view, do one of the following:

- Right-click a map-related column in the Data Element pane and click **Pick Visualization**, then select **Map**.
- Drag and drop a map-related column from the Data Element pane to the blank canvas, or between visualizations on the canvas. On the visualization toolbar, click **Change Visualization Type** and select **Map**.

4. In the properties pane, click **Map** and specify the visualization properties.
5. If you want to use a different map background, click the **Background Map** value in the properties pane and select a background. For example, select Google Maps, and the visualization displays Google Maps as the map background.
   - If you want to see the list of available map backgrounds or change the backgrounds that you can use, do one of the following:
     - Click the **Background Map** value and select **Manage Map Backgrounds** to display the Map Background tab.
     - Open the Console page, click **Maps** and select the Backgrounds tab.
   - Select another map background such as Satellite, Road, Hybrid, or Terrain.
6. Click **Save**.

Use Different Map Backgrounds in a Workbook

As an author you can use different map backgrounds in map visualizations.

Here is an example of how you might use a map background in a workbook.

1. On the Home page click **Create**, then click **Workbook**.
2. Select a dataset in the Add Dataset dialog.
3. Click **Add to Workbook**.

   The Workbook pane and list of Data Elements is displayed.
4. Select a map-related data element (for example, click **City**), and click **Pick Visualization**.
5. Select **Map** from the list of available visualizations.

   Either the default map background or an existing Oracle map background if no default is set is displayed.
6. In the visualization properties pane, select the **Map** tab.
7. Click the **Background Map** value and select a map from the drop-down list.

   For example, select Google Maps and Google Maps as the map background is displayed.
8. Optional: Click another value to change the type of map (such as Satellite, Road, Hybrid, or Terrain).
9. Optional: Click **Manage Map Backgrounds** from the **Background Map** options to display the Map Backgrounds pane.

   Use this option to maintain the map backgrounds that you want to use.

Interpret Data Values with Color and Size in Map Visualizations

You can use the color and size of a shape such as a polygon or a bubble, to interpret values in a map visualization.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Select a column and render it in a map view, doing one of the following:
   - Right-click a map-related column in the Data Element pane and click **Pick Visualization**, then select **Map**.
   - Drag and drop a map-related column from the Data Element pane to the blank canvas, or between visualizations on the canvas. On the visualization toolbar, click **Change Visualization Type** and select **Map**.
3. Drag and drop columns to the following sections on the Visualization Grammar Pane.
   - Use **Color** to change the color for geometries displayed in the corresponding map layer (for example, polygon fill color, bubble color) based on the values.
   - Use **Size (Bubble)** to change color bubble size based on the measure column values. To change the size of the color bubble you've to drag and drop measure columns only. The size shows the aggregated measure for a specific geographic location in a map visualization.
   - Use **Trellis Columns / Rows** to Compare multiple map visualizations based on the column values using filters.

In the map visualization, you can also use the following to interpret measure columns and attribute values:

- **Legend** - If a measure column or an attribute has multiple values, then the legend is displayed that shows values by size or color.
- **Tooltip** - If you hover the mouse pointer over a color bubble or data point, then the values are displayed in a tooltip.

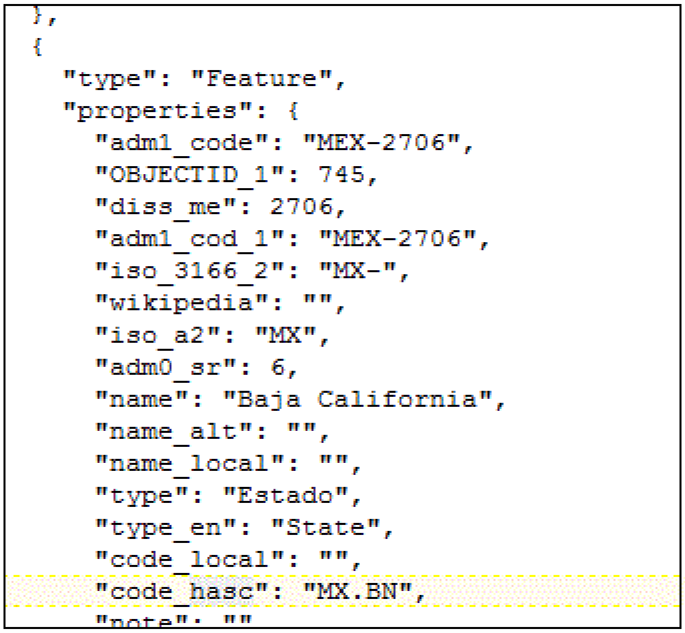
Add Custom Map Layers

You can add custom map layers to use in map visualizations.

Video

You add a custom map layer using a geometric data file with the .json extension that conforms to GeoJSON schema https://en.wikipedia.org/wiki/GeoJSON (the maximum file size allowed is 20MB).You then use the custom map layer to view geometric map data in a workbook. For example, you might add a Mexico_States.json file to enable you to visualize geometric data in a map of Mexico States.

When creating a custom map layer, you must select layer keys that correspond with data columns that you want to analyze in a map visualization. For example, if you want to analyze Mexican States data on a map visualization, you might start by adding a custom map layer for Mexican States, and select HASC code layer key from the Mexican_States.json file. Here's an extract from the Mexican_States.json file that shows some of the geometric data for the Baja California state.



```
},
{
  "type": "Feature",
  "properties": {
    "adm1_code": "MEX-2706",
    "OBJECTID_1": 745,
    "diss_me": 2706,
    "adm1_cod_1": "MEX-2706",
    "iso_3166_2": "MX-",
    "wikipedia": "",
    "iso_a2": "MX",
    "adm0_sr": 6,
    "name": "Baja California",
    "name_alt": "",
    "name_local": "",
    "type": "Estado",
    "type_en": "State",
    "code_local": "",
    "code_hasc": "MX.BN",
    "note": ""
```

If you wanted to use the Mexican_States.json file, the layer keys that you select must match columns that you want to analyze from the Mexican States Data tables. For example, if you know there's a data cell for the Mexican state Baja California then select the corresponding name field in the JSON file to display state names in the Map visualization. When you create a workbook and select column (such as State, and HASC), then Mexican states are displayed on the map. When you hover the mouse pointer over a state, the HASC code (such as MX BN) for each state is displayed on the map.

1. On the Home page, click **Navigator**, and then select **Console** to display the Console page.
2. Click **Maps** to display the Map Layers page.

   You can enable and disable System Map Layers but you can't create or delete them.
3. To add a custom map layer, click **Add Custom Layer** or drag and drop a JSON file from File Explorer to the Custom Maps area.
4. Browse the Open dialog, and select a JSON file (for example, Mexico_States.json).

   The JSON file must be a GeoJSON file that conforms to the standard specified in https://en.wikipedia.org/wiki/GeoJSON (the maximum file size allowed is 20MB).

   Custom layers that use the Line String geometry type aren't fully supported. The Color and Size section on Visualization Grammar pane doesn't apply to line geometries.
5. Click **Open** to display the Map Layer dialog.
6. Enter a **Name** and an optional **Description**.
7. Select the layer keys that you want to use from the Layer Keys list.

   The layer keys are a set of property attributes for each map feature, such as different codes for each state in Mexico. The layer keys originate from the JSON file. Where possible, select only the layer keys that correspond with your data.

8. Click **Add**. A success message is displayed when the process is complete and the layer is added.

Update Custom Map Layers

You can maintain custom map layers.

1. On the Home page click **Navigator**, and then click **Console**.
2. Click **Maps** to display the Map Layers page.
3. In the Custom Map Layers section, right-click the map layer and click **Options**, then do the following:
   - To view or make changes to the map layer settings, select **Inspect**.
     The Map Layer dialog is displayed where you can update the **Name**, **Description**, or the **Layer Keys** used in this layer.
   - To upload the JSON file again, select **Reload**.
   - To save the JSON file locally, select **Download**.
   - To delete the custom map layer, select **Delete**.
     You can disable or enable a System Map Layer and a Custom Map Layer, but you can’t add or delete a System Map Layer.
4. Click the map layer to enable or disable it. For example, if you want to exclude us_states_hexagon_geo on the map, click the layer to disable it and remove it from searches.
5. To switch from using one map layer to another, do the following:
   a. In the properties pane, select the **Map** tab to display the map properties.
   b. Click the current **Map Layer** for example Mexican States. This displays a list of available custom map layers that you can choose from.
   c. Click the map layer that you want to use to match your data points.

Apply Multiple Data Layers to a Single Map Visualization

You can use the data layer feature to display multiple data series (different sets of dimensions and metrics) on a single map visualization. The data layers are overlaid on one another in a single map visualization.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Drag and drop a measure or attribute columns containing map-related data from the Data Panel to the Category (Location) section on the Grammar Panel.

   If you’re creating a map visualization, in the Data Panel, right-click an attribute column and click **Pick Visualization** then select **Map**.
3. Click **Layer options** in the Category (Location) section of the Grammar Panel and click **Add Layer** to add a new data layer (for example, Layer 2).

   Alternatively in the Data Layers tab click **Add Layer (+)**.
4. Drag and drop a column to the Category (Location) section. Based on the column values the map visualization automatically updates with a different set of dimensions, and it overlays on the previous layer.
5. Repeat step 3 and 4 to add multiple data layers on the map visualization.

6. Click **Layer options** to use other options such as Order Layer, Hide Layer, and Manage Layers.
7. In the Data Layers tab of the properties pane, you can specify the options for a layer such as Layer Type, Map Layer, Transparency, and Show Layer.
8. To refine the data shown for the measure and attribute columns in all the data layers, you can apply a filter such as a Range Filter or List Filter, to the map visualization. For example, you can select a measure or attribute for a layer, then apply filter to reduce the amount of data shown, and add the same measure or attribute to the Color section on the Grammar Panel.

Use an Image as a Map Background and Draw Map Layer Shapes on the Image

You can upload an image, prepare the image as a map background, draw map layer shapes onto the image, and associate data with the map background layer.

▶ Video

Topics:

- Upload an Image as a Map Background
- Draw Custom Map Layer Shapes on an Uploaded Image
- Associate a Dataset with Map Layer Shapes Drawn on an Uploaded Image

Upload an Image as a Map Background

You can upload an image as a map background and then draw layers on top of the uploaded image.

1. On the Home page, click **Navigator**, and then click **Console**.
2. On the Console page, click **Maps**.
3. In Maps, click the **Backgrounds** tab, and expand **Image Backgrounds**.
4. Click **Add Image**, select your image, and click **Open**.
5. Enter a name and description for the uploaded image, and click **Save.**

Draw Custom Map Layer Shapes on an Uploaded Image

You can draw and edit custom map layer shapes on an uploaded image and associate the shapes with data in map visualizations.

1. On the Home page, click **Navigator**, and then click **Console**.
2. On the Console page, click **Maps**.
3. In Maps, click the **Backgrounds** tab, and expand **Image Backgrounds**.
4. In Image Backgrounds, select the image, click **Options**, and then select **Create Map Layer**.
5. Select **Polygon**, **Line**, or **Point**, and draw a shape onto the image.

Shape	Actions
Polygon	Click the image, drag and click to draw each edge of the polygon until the shape is complete, then click to finish.
Circle	Click the image, drag to increase the circle size until the circle reaches the required size, then click to finish.
Line	Click the image, then drag and click to draw each line edge until the line is complete.
Point	Click the image in the location where you want to draw a data point.

If you uploaded a motorcycle image, you could draw a shape outline over each visible part. For example, you might draw a polygon to represent an irregular shape like the fuel tank, or a line to represent a fork, or a circle to represent a tire, and so on.

Each new shape that you create is given a default name and is listed under Features.

6. Enter a name for each shape that corresponds to a key column value in the dataset.

   For example, if you drew a petrol tank polygon shape and the key column PartID in the dataset has the value PT for petrol tank, then you must enter `PT` as the shape name.

   You can also edit a shape name by clicking the corresponding object in the Features list.

7. Optional: If you want to reposition a shape, click the shape, then drag it to move it to a new position.
8. Optional: If you want to resize a shape, click and hold the shape, or edge, and drag it until it reaches the required size, then click again to finish.
9. Click **Save**.

Associate a Dataset with Map Layer Shapes Drawn on an Uploaded Image

You can associate a dataset with the map background layer shapes that you drew on an uploaded image, and use it in a workbook.

1. In the Home page, click **Create** to bring in a dataset file that you want to associate with your map background.

   For example, you might select `motorbike.xls` to create a motorcycle dataset with a PartID key column containing values that match the part names of the shapes that you drew.

2. In the key column of the dataset, click **Options**, select **Location Details**, choose the custom map layer, and click **OK** to assign the key column to the selected map layer.
3. Click **Apply Script** to associate the key column with the map layer.
4. Create a workbook.
5. Drag and drop the key column into the visualization. This automatically places the column into Category (Location).

   A map visualization is suggested based on the key column, and the associated map background is displayed.

6. Continue to add columns, and create visualizations as required.
7. Click **Save**.

Assign a Map Layer to a Data Column

Assign a map layer to a data column to use it consistently in any workbook.

You can assign a map layer to a column that contains text or numeric attributes, for example, columns such as Airport Name, Latitude, and Longitude. When you select a data column with a map layer assignment for a visualization, Oracle Analytics automatically creates a map visualization.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click **Data** to go to the Data page.
3. In the **Data Diagram** select the dataset that contains the column that you want to prepare and click the dataset's **Edit** icon.

   If the dataset contains multiple tables, the Dataset editor is displayed and you'll see the Join Diagram with a tab for each table. Select the table that contains the column you want to prepare to open it in the Transform editor.
4. In the Transform editor, click **Options** for any numeric or text attribute column, and then click **Location Details**.
5. In Location Details, review the associated map layer, change the map layer if you want to, and then click **OK**.

   The property change is listed as a Change Property step in the Preparation Script pane.
6. Depending on your dataset, in the Preparation Script pane click **Apply Script**, or on the toolbar click **Save Dataset**.

   The updated column displays the location icon indicating that the location preference has been set.
7. In the Properties panel for the updated column, click the **Location** icon to verify the map layer associated with the column.
8. Create a visualization with the data columns for which you have set the location details.

   The visualization type is now automatically set as Map and the map layer is available for the specific columns. You no longer need to set the location details for each visualization.

Auto Focus on Data for a Map Visualization

You can override the current viewport when you focus on new data in a map visualization.

The ability to auto focus on data enables you to view the visualization for the specific data that you displayed in a map. For example, if your map visualization displays sales by countries and you have first viewed sales for Australia, the map zooms to Australia. However, if you focus on sales for Italy, the map zooms to Italy.

1. On the Home page, select a workbook containing a map visualization, click the **Actions menu**, and then select **Open**.
2. In Visualization Properties, click **Map**, and then set **Auto Focus on Data** to **On**.

The default value of this property is **Off**.

Review Location Matches for a Map Visualization

You can review mismatch issues between data and match results in map layers, such as when there are ambiguous or partial matches between words.

If ambiguous results are present, consider adding more columns to the map visualization to get the specific match. For example, your map might have layers for City, Country, and Continent. You can also exclude rows of data.

1. On the Home page, select a workbook containing a map visualization, click the **Actions menu**, and then select **Open**.
2. Click the **Visualization** tab.
3. Right-click the map visualization and select **Location Matches** from the menu to display the Location Matches dialog.
4. Select a tab representing a map layer in the current visualization to inspect how well your data matches the map layer.

   For example, select the **Country** tab to see how well your data matches with the Country map layer.
5. Optional: Click **Map Layer** to select a different layer or click **Manage Map Layers** to display the Console page.
6. Review and resolve data mismatches use the columns.

   Use **Match** to see what data items match with the map layer information. A match can be anything from No Match, Partial Match, to a 100% match. Matches are displayed initially sorted top down from the worst to the best matches.

   - No Match - Displays a red warning triangle indicator.
   - A Match with an issue - Displays a yellow warning triangle indicator.
     The warning doesn't indicate a poor match but an imperfect match for which you might want to review the use case.
   - A Perfect Match - Displays no warning triangle indicator.
   - If you are matching Latitude and Longitude, the match values are Valid or Invalid.

   The summary section above the table displays the number of locations and any issues.
7. Click the filter icon in the **Match** column title and select a filter option.
   - Use **All Data** to display all types of matches.
   - Use **Good Matches** to display only 100% perfect matches.
   - Use **All Issues** to display partial matches, multiple matches and no match.
   - Use **Partial Matches** to indicate the percentage difference between the strings being matched. For example: Part of a string is exactly right, such as `Paulo` versus `Sao Paulo`. Most of a word is exactly right, such as `Caiyro` versus `Cairo`.
   - Use **Multiple Matches** to indicate how many matches exist for ambiguous cases. For example, you may see `Barcelona, Spain` matching with `Barcelona Argentina`. In this case you might want to revisit the data to add more detail to your GEO columns to ensure that you only match the correct columns.
8. Click in the **Exclude** column for each row of data that you want to exclude.

9. Click the **Exclude** menu:
   - Click **Select All** or **Deselect All**.
   - Click one of **Workbook Scope**, **Canvas Scope**, or **Visual Scope**.
10. Optional: Add in more columns to the Category (Location) edge on the visualization to make your match more specific. For example, add Country data to remove a mismatch like Barcelona, Spain versus Barcelona, Argentina.
11. Display the Location Matches dialog to check the summary for any remaining mismatches, and click **OK** when you are satisfied, or repeat previous steps as required.

Create Heatmap Layers on a Map Visualization

You can use a heatmap as a data layer type on a map visualization to identify the density or high concentration of point values or metric values associated with the points. For example, you can use a heatmap to identify the high profit stores in a geographic region or country.

You can create two types of heatmap layers:

- **Density heatmap** - Uses only map-related column data (such as latitude and longitude columns). Density heatmap layers show the cumulative sum of a point, where each point carries a specific weight. A point has a radius of influence around it, such that other points that fall in the same area also contribute to the total cumulative result of a point.
- **Metric heatmap** - Uses measure column data in the same layer. For example, if you add a measure column to the Color section on the Grammar Panel the heatmap is updated to show interpolated metric values.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Create an empty map visualization.
3. Drag and drop attribute columns containing map-related data from the Data Panel to the Category (Geography) section on the Grammar Panel.
   - If you're creating a workbook with a map visualization, in the Data Panel, right-click an attribute column and click **Pick Visualization** then select **Map**.
4. Go to the Data Layers tab of the properties pane.
   - Alternatively, click **Layer options** in the Category (Geography) section and click **Manage Layers**.
5. To create a density heatmap, click **Layer Type** value and select **Heatmap**.
   - Alternatively, you can add a new map layer, change the layer type to **Heatmap** and then add attribute columns to the Category (Geography) section.
6. To create a metric heatmap, drag and drop a metric column from the Data Panel to the Color section. The heatmap visualization changes from density to metric.
7. In the Data Layers tab of the properties pane, specify the options for the heatmap layer such as Radius, Interpolation, Transparency, and Color.
   - The default interpolation method is automatically selected based on the aggregation rule of the metric column or value that you've selected for the layer.

- You can select the radius value in pixels (px). The radius value is the extent of influence of a measure around a point value on a map.

The heatmap is automatically updated based on the options selected in the Data Layers tab.

Create Cluster Layers on a Map Visualization

You can use a cluster layer as a type of data layer on a map visualization. In a cluster layer, points that are positioned near each other are grouped together into a common bubble.

The number of points clustered in the group is indicated in the bubble's label. If selected points are grouped with unselected points, the circle is dotted to indicate a partial selection. Individual points are displayed as pin icons to emphasize the distinction between grouped and ungrouped points. Points are grouped based on their proximity in pixels and on the map's zoom factor.

1. On the Home page, click the **Actions menu**, and then select **Open**.
2. Create an empty map visualization by dragging the Map visualization from the Data Panel to the canvas.
3. Drag and drop attribute columns containing map-related data from the Data Panel to the Category (Geography) section on the Grammar Panel.

   If you’re creating a workbook with a map visualization, in the Data Panel, right-click an attribute column and click **Pick Visualization** then select **Map**.
4. Click the Data Layers tab of the properties pane.

   Alternatively, click **Layer options** in the Category (Geography) section and click **Manage Layers**.
5. To create a point cluster, click **Layer Type** value and select **Cluster**.

   Alternatively, you can add a new map layer, change the layer type to **Cluster** and then add attribute columns to the Category (Geography) section.

The point cluster is automatically updated based on the zoom level.

Represent Point Data With Custom Icons on a Map

You can use the Shapes edge to represent point data with custom icons in a map visualization.

You can associate a column with the Shape edge to display a custom shape for point data. For example, you can distinguish between cities by displaying them using custom shapes (for example, a square, a triangle, or a currency symbol). You can also change which custom shape you want to apply to one or more data points.

1. On the Home page, select a workbook containing a map visualization that contains point data, click the **Actions menu**, and then select **Open**.
2. Drag and drop an attribute column containing point data (for example, city) from the Data Panel to the Category (Geography) edge on the Grammar Panel.
3. Drag and drop a column from the Data Panel to the **Shapes** edge and optionally to the **Color** edge on the Grammar Panel.

   The map visualization automatically updates based on your selection and overlays the previous layer.

4. Optional: You can change how you assign custom shapes to data points and to the map legend.
    a. Highlight one or more data points on the map using one of the selection tools, or use Ctrl-click to select one or more data points.
    b. Right-click one of the multiple-selected data points, select **Custom Shapes** and then select **Series** or **Data Point**.
    c. Select a custom shape and click **Done**.

        Custom shapes are applied as follows:

        - **Series dialog** - custom data point shape not previously set
          Replaces highlighted data points and series items with the custom shape.
        - **Series dialog** - custom data point shape previously set
          Replaces only corresponding series items with the custom shape.
        - **Data Point dialog**
          Replaces only highlighted data points with the custom shape.
5. If you want to reassign the custom shape for a data point:
    a. Right-click any data point, select **Shape**, and click **Custom Shapes**.
    b. To change the custom shape assigned to a data point, click the shape corresponding to the data point that you want to change.
    c. Select a new custom shape and click **Done**, then click **Done** again.
6. If you want to reset all of the custom shapes currently applied to data points on a map visualization, right-click any data point, select **Shape**, and click **Reset Custom Shapes**.

    This resets all of the shapes applied to data points on the map to the default setting.

Select Points or Area on a Map

You can select multiple points on the map in a specific area that you define using the selection tools.

1. On the Home page, select a workbook containing a map visualization, click the **Actions menu**, and then select **Open**.
2. In the Visualize canvas, select the map visualization.
3. On the visualization toolbar, do one of the following:
    - Click the **Rectangle Selection** tool and drag a rectangle on the map to select the points or area you want.
    - Click the **Radial Selection** tool and select a point on the map, then drag outward to create a circle. The unit shows the total distance covered on the map.
    - Click the **Polygon Selection** tool and drag a freehand border around the points or area you want to select on the map.

    The selected points or area is highlighted on the map.

Represent Line Data Using Size and Color on a Map

You can represent the weight of line data through thickness and color in a map visualization.

You can associate a measure with the Size edge to indicate the relative weight of a line. For example, to compare delays in airline routes, you can display flight routes with varying line thickness, where a thicker line and a darker color correspond to a higher number of delays.

1. On the Home page, select a workbook with a map visualization that contains line data, click the **Actions menu**, and then select **Open**.
2. In the Visualize canvas, select the map visualization that displays the line data.
3. Drag and drop a column containing line data (for example airline routes) from the Data Panel to the Category (Geography) edge on the Grammar Panel.
4. Drag and drop a line measure column from the Data Panel to the **Size** edge and optionally to the **Color** edge on the Grammar Panel.

   The map visualization automatically updates based on your selection and overlays the previous layer.

Make Map Layers and Backgrounds Available to Users

For visualization workbooks, as an administrator you can hide or display map layers and backgrounds for users.

1. On the Home page, click **Navigator** and then click **Console**.
2. Click **Maps** and then click **Backgrounds** or **Map Layers**.
3. Click the **Include** blue tick option to make the selected row item available or hidden from end users.

   You can hide or display map backgrounds, image backgrounds, custom map layers, and system map layers.

Use a Map Background as the Default

For visualization workbooks, as an administrator you can make a map background the default for users.

1. On the Home page, click **Navigator**, and then click **Console**.
2. Click **Maps**, click **Backgrounds** and then click **Map Backgrounds**.
3. Click the **Default** column field in the row of a map background to make it the default.

   The map background is used by default in new visualizations.

Add Map Backgrounds

You can add Google, Baidu, and other web map service backgrounds to use in visualizations.

Topics:

- Add Google Map Backgrounds
- Add Baidu Map Backgrounds

- Add Web Map Service (WMS) Backgrounds
- Add Tiled Web Map (XYZ) Backgrounds
- Web Map Background Troubleshooting Tips

Add Google Map Backgrounds

You can add Google map backgrounds to use in map visualizations.

1. On the Home page, click **Navigator**, and click **Console**.
2. In the Console, select **Maps**, click **Backgrounds**, and then click **Map Backgrounds**.
3. Click **Add Background** and select a **Google** from the list.
4. Enter a helpful description if needed.
5. Click **Account Type** to select your account.
6. Copy and paste your Google Maps API access key.

   To use Google Maps tiles, you must obtain a Google Maps API access key from Google. Google prompts you to enter your Maps API access key and, when applicable, your Google "Client ID". Usage of the tiles must meet the terms of service specified by Google in the Google Developers Site Terms of Service.
7. Click **Default Map Type** if applicable.
8. Click **Save** to include the map in the list of available map backgrounds.

Add Baidu Map Backgrounds

You can add Baidu map backgrounds to use in map visualizations.

1. On the Home page, click **Navigator**, and click **Console**.
2. In the Console, select **Maps**, click **Backgrounds**, and then click **Map Backgrounds**.
3. Click **Add Background** and select **Baidu** from the list.
4. Enter a helpful description if needed.
5. Copy and paste your Baidu Maps API access key.

   To use the Baidu Maps tiles, you must obtain a Baidu Maps API access key from Baidu. Baidu prompts you to enter your Maps API access key. Usage of the tiles must meet the terms of service specified by Baidu in their user agreement.
6. Click **I agree to trust this external host** if applicable.
7. Click **Save** to include the map in the list of available map backgrounds.

Add Web Map Service (WMS) Backgrounds

You can add web map service backgrounds and use them in map visualizations.

Web map service backgrounds are hosted dynamically on a web server using the Web Map Service (WMS) protocol. You can use them to integrate maps containing information that you might not have in your enterprise, and easily present it spatially with your data.

1. On the Home page, click **Navigator**, and click **Console**.
2. In the Console, select **Maps**, click **Backgrounds**, and then click **Map Backgrounds**.
3. Click **Add Background** and select **Web Map Service** from the list.
4. Click the **General** tab and enter a name and description.
5. In **URL**, enter the full URL for the web map service.

   Contact your web map service provider to find out the correct URL.

   For example, `https://www.gebco.net/data_and_products/gebco_web_services/web_map_services/mapserv`.

   Version automatically displays the web map service protocol version used by the WMS provider. The default value is 1.1.1 but you can select an alternative version from the drop-down list.

   Coordinate Reference System automatically displays the reference system used by the host provider. The default value EPSG:3857 is the coordinate system used for projecting maps in 2 dimensions. Hover over the tooltip icon for details.
6. Click **I agree to trust this external host** to automatically add the host to your list of safe domains.
7. In **Layers**, enter the name of each background map layer that you want to use. Click the cross **(x)** icon to remove a layer.
8. Click **Format** if you need to change the image type.
9. Display the **Parameters** tab and click **Add Parameter**.

   Parameters that you enter are included in the URL and tell the hosting server what to display in the map background (for example, image type, layer, geographic extent of the map, size of the returned image).
10. Enter parameters that you want to pass in the URL to the host server in `key:value` format.

    Use this URL to find parameters that you can use for this web map service:

    ```
    http://<url_of_the_map_server>?request=getCapabilities&service=wms
    ```
11. Click **Save** to add the background map layer to the list of available map backgrounds.

    You must refresh a page to see any changes.
12. Click **Preview** to display a preview of the map background.

    The Preview tab only becomes available after the page is saved and refreshed or Oracle Analytics is restarted. This is because the refresh or restart enables recognition of the safe domains.

Add Tiled Web Map (XYZ) Backgrounds

You can add tiled web map (XYZ) backgrounds and use them in map visualizations.

Tiled web map (XYZ) backgrounds are displayed in a browser by seamlessly joining dozens of individually requested image or vector data files over the Internet through a web server. You can use them to integrate maps containing information that you might not have in your enterprise, and easily present them spatially with your data.

You configure tile URL strings to specify vector tiles or raster tiled map images to load. The host evaluates each string and determines which tile to load.

Contact your tiled web map service provider to find out the correct URLs. Parameters in the URLs tell the hosting server what to display in the map background. For example, the map name, version, and the number of tiles to use at the specified focus. Here are some examples of Mapbox tiled web map background URLs:

- `https://api.mapbox.com/styles/v1/mapbox/streets-v11/tiles/256/{z}/{x}/{y}`
- `https://api.mapbox.com/styles/v1/mapbox/satellite-v9/tiles/256/{z}/{x}/{y}`
- `https://api.mapbox.com/styles/v1/mapbox/light-v10/tiles/256/{z}/{x}/{y}`
- `https://api.mapbox.com/styles/v1/mapbox/dark-v10/tiles/256/{z}/{x}/{y}`
- `https://api.mapbox.com/styles/v1/mapbox/outdoors-v11/tiles/256/{z}/{x}/{y}`

1. On the Home page, click **Navigator**, and click **Console**.
2. In the Console, select **Maps**, click **Backgrounds**, and then click **Map Backgrounds**.
3. Click **Add Background** and select **Tiled Web Map** from the list.
4. Click the **General** tab and enter a name and description.
5. In **URL**, enter the full URL for the host tiled web map service.
6. Click the plus **(+)** icon to enter additional server URLs in the same domain, for load balancing.
7. Click **I agree to trust this external host** to automatically add the host to your list of safe domains.
8. Display the **Parameters** tab and click **Add Parameter**.

   Parameters that you enter are included in the URL and tell the hosting server what to display in the map background (for example, image type, layer, geographic extent of the map, size of the returned image).

   For example, you might enter `access_token` with a value of `exampleaccessTokenXyZ123456789nnnxxxZZz`.

   Contact your provider for details.
9. Enter parameters that you want to pass in the URL to the host server in `key:value` format.
10. Click **Save** to add the specified tiled web maps to the list of available map backgrounds.

    You must refresh a page to see any changes.
11. Click **Preview** to display a preview of the map background.

    The Preview tab only becomes available after the page is saved and refreshed or Oracle Analytics is restarted. This is because the refresh or restart enables recognition of the safe domains.

Web Map Background Troubleshooting Tips

Use this topic if you experience errors when you add a web map background.

When a web map background image isn't displayed in the Preview tab or in a visualization, you must take action to resolve the issue.

- If the error persists and you want to diagnose the issue:
  - Click **F12** to display the browser Developer Tools application and search for errors in the browser console tab. For example, search for `error` or `CORS`. Error messages are displayed in red text.
  - If you see an error message similar to `Access to image at http://example.com has been blocked by CORS policy...`, contact the host provider to resolve the issue. Error messages that refer to the Cross Origin Resource Sharing (CORS) policy can only be resolved by the host provider.
- Alternatively, if the error persists contact Oracle Support to help you resolve the issue.

Sort, Drill, and Select Data in Visualizations

You can narrow your focus to explore certain aspects of your data by sorting, drilling, and selecting data.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Select a visualization and click **Menu**.
3. Select one of the following:
   - Click **Sort By** and then select **Custom**, or click **Sort** in the visualization tool bar to display the Sort Order dialog where you can sort one or more attributes in a visualization. You can create and view sorts, define sort attributes, arrange the sort order for multiple sorts, and view and resolve sort conflicts. You can also sort an attribute by a measure column that isn't used in the visualization.
     If the table includes subtotals or contains a dimension, then all column sorts on columns after the subtotal or dimension, are sorted in the subtotal group.

     If you're working with a table view with multiple sorts, then the system always sorts the last column that you sorted first. In some cases where specific values display in the left column, you can't sort the center column. For example, if the left column is Product and the center column is Product Type, then you can't sort the Product Type column. To work around this issue, swap the positions of the columns and try to sort again.
   - Click **Drill** to create a drill to a data element, and to create a drill through hierarchies in data elements, for example you can create a drill to weeks within a quarter. You can also create drills using multiple data elements. For example, you can select two separate year members that are columns in a pivot table, and drill into those members to see the details.
   - Click **Drill to [*Attribute Name*]** to directly drill to a specific attribute within a visualization.
   - Click **Keep Selected** to keep only the selected members and remove all others from the visualization and its linked visualizations. For example, you can keep only the sales that are generated by a specific sales associate.

- Click **Remove Selected** to remove selected members from the visualization and its linked visualizations. For example, you can remove the Eastern and Western regions from the selection.

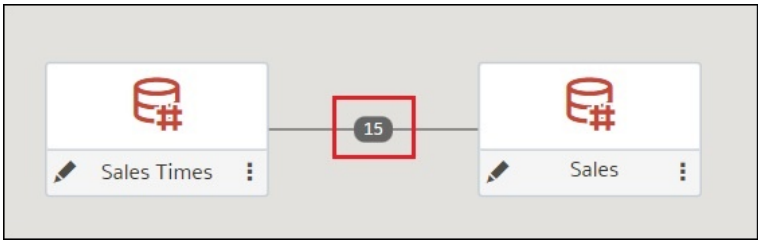
Replace a Dataset in a Workbook

When you replace a dataset, Oracle Analytics suggests how you can re-map the columns from the dataset that you replaced to the replacement dataset. You can accept or update these suggested mappings.

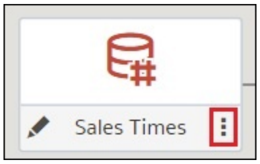
For example, you might replace a workbook's test dataset with a production dataset. Or you can copy a workbook and its visualizations, and then use the duplicate workbook as a template. After you rename the duplicated workbook, you can add a different dataset.

When you replace the dataset, any data mapping that you specify is applied to the workbook. For example, if you map a data element to **None**, the specific data is removed from the workbook's visualizations, calculations, and filters.

1. On the Home page, select a workbook to use as a template, click the **Actions menu**, and then select **Open**.
2. From the Visualize page, click **Data**.
3. In the Data page, click the **Data Diagram**.
4. Optional: If the dataset that you want to replace has been blended with another dataset, then in the **Data Diagram** click the number between the dataset icons.



5. Optional: In the Blend Data dialog, click **Delete All Matches**, specify which dataset to delete conformance from, and then click **OK**.
6. Locate the dataset to replace and click **Actions**.



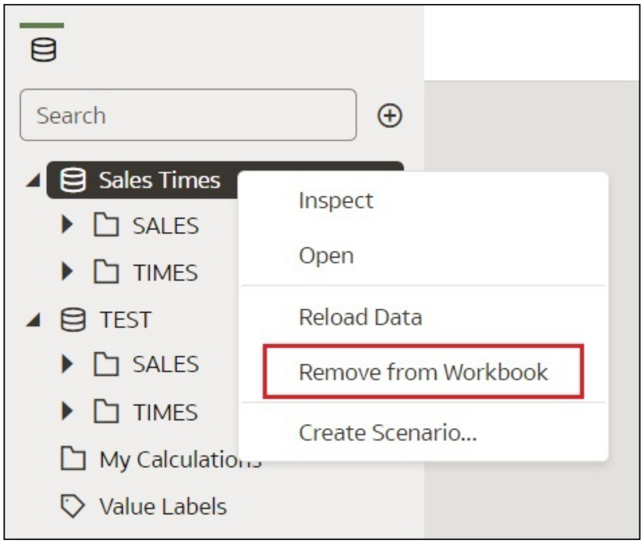
7. Select **Replace Dataset**.
8. In the Replace Dataset dialog, click the replacement dataset. Click **Select**.
9. Review, update, and specify the column mappings for the workbook's visualizations, calculations, and filters.
10. Click **Replace**.

11. Optional: If you want to add blending between datasets, then in the **Data Diagram**, click between the datasets that you want to blend, and in the Blend Data dialog add column matches.

Remove a Dataset from a Workbook

You can remove a dataset from a workbook. When you do this, you remove all of the dataset's columns from the workbook's visualizations, filters, and calculations.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the **Visualize** canvas, go to the Data Panel pane, right-click the dataset that you want to remove, and select **Remove from Workbook**.



3. Click **Save**.

Analyze Data with Explain

Explain uses machine learning to find useful insights about your data.

Topics:

- What is Explain?
- What Are Insights?
- Use Explain to Discover Data Insights

What is Explain?

Explain analyzes the selected column within the context of its dataset and generates text descriptions about the insights it finds. For example, for any column you'll find out basic facts, key drivers, segments that explain the column, and anomalies.

Video

Explain uses Oracle's machine learning to generate accurate, fast, and powerful information about your data, and creates corresponding visualizations that you can add to your workbook's canvas.

Explain is for data analysts who might not know what data trends they're looking for, and don't want to spend time experimenting by either dragging and dropping columns onto the canvas, or using data flows to train and apply predictive models.

Explain is also a useful starting point for data analysts to confirm a trend that they're looking for in their data, and then use that information to create and tune predictive models to apply to other datasets.

What Are Insights?

Insights are categories that describe the selected column within the context of its dataset.

The insights that Explain delivers are based on the column type or aggregation that you chose and will vary according to the aggregation rule set for the chosen metric. Explain generates only the insights that makes sense for the column type that you chose.

Insight Type	Description
Basic Facts	Displays the basic distribution of the column's values. Column data is broken down against each of the dataset's measures. This insight is available for all column types. • For a selected metric, this insight shows the distribution of the aggregated metric value for each member of each attribute column. • For a selected attribute, this insight shows the value of each metric in the dataset across the member values of the attribute.
Key Drivers	Shows the columns in the dataset that have the highest degree of correlation with the selected column outcome. Charts display the distribution of the selected value across each correlated attributes value. This tab displays only when explaining attribute columns, or when explaining a metric column that has an average aggregation rule.
Segments	Displays the key segments (or groups) from the column values. Explain runs a classification algorithm on the data to determine data value intersections and identifies ranges of values across all dimensions that generate the highest probability for a given outcome of the attribute. For example, a group of individuals of a certain age range, from a certain set of locations, with a certain range of years of education form a segment that has a very high probability of purchasing a given product. This tab displays only when explaining attribute columns.
Anomalies	Identifies a series of values where one of the (aggregated) values deviates substantially from what the regression algorithms expect.

Use Explain to Discover Data Insights

When you select a column and choose the Explain feature, Oracle Analytics uses machine learning to analyze the column in the context of the dataset. For example, Explain searches the selected data for key drivers and anomalies.

Explain displays its findings to you as text descriptions and visualizations, which you can add to your workbook's canvas.

Tutorial

If you perform explain on a column and the results contain too many correlated and highly ranked columns (for example, ZIP code with city and state), then excluding some columns from the dataset so that Explain can identify more meaningful drivers. See Hide or Delete a Column.

Explain is available after you install machine learning into your local Oracle Analytics Desktop directory. See How do I install Machine Learning for Desktop?

1. In the Home page, click **Create** and then **Workbook** to create a new workbook.
2. Click **Visualize** to open the Visualize page.
3. In the Data Panel, right-click a column and select **Explain <Data Element>**.

   For Explain to successfully analyze an attribute, the attribute must have three to 99 distinct values.

   The Explain dialog displays basic facts, anomalies, and other information about the selected column.
4. Optional: In the Segments view, select the segments (or groups) that predict outcomes for the column you selected.
   - Click one or more columns to see how they impacts the column's outcome.
   - Sort how the information is displayed in the Segments. For example, confidence high to low or low to high.
5. For each visualization that you want to include in your workbook's canvas, hover over it and click **Select for Canvas**.

   You'll see a green tick next to selected items.
6. Click **Add Selected**.

   You can manage the Explain (data insight) visualizations like any other visualizations you’ve manually created on the canvas.

About Warnings for Data Issues in Visualizations

You see a data warning icon when the full set of data associated with a visualization isn't rendered or retrieved properly. If the full set of data can't be rendered or retrieved properly, then the visualization displays as much data as it can as per the fixed limit, and the remaining data or values are truncated or not displayed.

8

Create and Apply Filters

This topic describes how you can use filters to find and focus on the data you want to explore.

Topics:

About Filters and Filter Types

Filters reduce the amount of data shown in visualizations, canvases, and workbooks.

The Range, List, Date, and Expression filter types are specific to either a visualization, canvas, or workbook. Filter types are automatically determined based on the data elements you choose as filters, but you can also switch to select another compatible filter type.

- **Date filters** - Use calendar controls to adjust time or date selections. You can either select a single contiguous range of dates, or you can use a date range filter to exclude dates within the specified range. See Apply Date Range Filters.
- **Expression filters** - Let you define more complex filters using SQL expressions. See Apply Expression Filters.
- **List filters** - Applied to data elements that are text data types, number data types that aren’t aggregatable, and date data types. List filters are applied to the whole canvas or to a specific visualization on the canvas. See Apply List Filters.
- **On-canvas filters** - Applied to attribute data elements. This filter type is added to and displays on the canvas and allows users to choose how to filter visualizations on the canvas. See Create On-Canvas Filters.
- **Range filters** - Generated for data elements that are number data types and that have an aggregation rule set to something other than none. Range filters are applied to data elements that are measures, and that limit data to a range of contiguous values, such as

revenue of $100,000 to $500,000. Or you can create a range filter that excludes (as opposed to includes) a contiguous range of values. Such exclusive filters limit data to noncontiguous ranges (for example, revenue less than $100,000 or greater than $500,000). See Apply Range Filters.

- **Top and bottom filters -** Applied to measure and attribute data elements. You can specify whether to filter by top or bottom, specify the number of items to display, and which measure or attribute to limit by. See Apply Top Bottom N Filters.

How Datasets Interact with Filters

There are several ways that datasets can interact with filters in a workbook.

Various factors affect the interaction of datasets and filters in workbooks:

- The number of datasets within a workbook.
- The datasets that are blended or not blended (for a workbook with multiple datasets).
- The data elements (columns) that are matched between blended datasets.

You can use the Data Diagram in the Data page of a workbook to:

- See blended and not-blended datasets.
- Blend or connect multiple datasets by matching the data elements in the datasets.
- Disconnect the datasets by removing matched data elements.

How the Number of Datasets Affects Filters

Filters can interact differently with visualizations depending on the number of datasets, whether the datasets are joined, and what the filters are applied to.

You can add filters to the filter bar or to individual visualizations in a workbook.

Single Dataset	Filter Interaction
Add a filter to the filter bar	It applies to all visualizations in the workbook.
Add a filter to a visualization	It is applied after filters on the filter bar are applied.
Add multiple filters	By default filters restrict each other based on the values that you select.

Multiple Datasets	Filter Interaction
If you add filters to the filter bar.	• The filters apply to all the visualizations using the joined datasets. For visualizations using the not-joined datasets, you must add a separate filter to each dataset. • You can't specify data elements of a dataset as a filter of other datasets, if the two datasets aren't joined. • If a data element of a dataset is specified as a filter, but doesn’t match the joined datasets, then the filter applies only to the visualization of that particular dataset, and does not apply to other visualizations of joined or not-joined datasets. • You can select **Pin to All Canvases** of a filter, to apply a filter to all canvases in the workbook.

Multiple Datasets	Filter Interaction
If you hover over a filter name to see the visualization to which the filter is applied.	Any visualizations that don't use the data element of the filter are grayed out.
If you add filters to visualizations	• If you specify a filter on an individual visualization, that filter applies to that visualization after the filters on the filter bar are applied. • If you select the **Use as Filter** option and select the data points that are used as a filter in the visualization, then filters are generated in the other visualizations of joined datasets and matched data elements.

You can use the **Limit Values By** options to remove or limit how the filters in the filter bar restrict each other.

About Automatically Applied Filters

By default, the filters in the filter bar and filter drop target are automatically applied. However, you can turn this behavior off if you want to manually apply the filters.

When the **Auto-Apply Filters** is selected in the filter bar menu, the selections you make in the filter bar or filter drop target are immediately applied to the visualizations. When **Auto-Apply Filters** is off or deselected, the selections you make in the filter bar or filter drop target aren’t applied to the canvas until you click the **Apply** button in the list filter panel.

Create Filters on a Workbook

You can add filters to limit the data that’s displayed in the visualizations on the canvases in your workbook.

If your workbook contains multiple datasets and some aren’t joined, then there are restrictions for how you can use filters. Any visualization that doesn't use the data element of the filter is grayed out.

Instead of or in addition to adding filters to the workbook or to an individual canvas, you can add filters to an individual visualization.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click **+ Add Filter**, and select a data element. Alternatively, drag and drop a data element from the Data Panel to the filter bar.

   You can't specify data elements of a dataset as a filter of other datasets, if the two datasets aren’t joined.
3. Click the filter **Menu** and hover the mouse pointer over the **Filter Type** option to select an alternative compatible filter type.
4. Set the filter values. How you set the values depends upon the data type that you’re filtering.
   - Apply a range filter to filter on columns such as Cost or Quantity Ordered.
   - Apply a list filter to filter on columns such as Product Category, Product Name, or Date.
   - Apply a date filter to filters on columns such as Ship Date or Order Date.

5. Click the filter bar menu or right-click, then select **Add Expression Filter**.
6. Click the filter **Menu** and hover the mouse pointer over the **Limit Values By** option to specify how the filter interacts with the other filters in the filter bar. Note the following:
    - By default, the **Auto** option causes the filter to limit other related filters in the filter bar.
      For example, if you've filters for Product Category and Product Name, and if you set the Product Category filter to Furniture and Office Supplies, then the options in the Product Name filter value pick list is limited to the product names of furniture and office supplies. You can select **None** to turn this limit functionality off.
    - You can specify any individual filter in the filter bar that you don't want to limit. For example, if you have filters for Product Category, Product Sub Category, and Product Name, and in the **Limit Value By** option for the Product Category filter you click Product Sub Category, then the product subcategory filter shows all values and not a list of values limited by what you select for Product Category. However, the values shown for Product Name is limited to what you select for Product Category.
7. Click the filter bar menu or right-click and select **Auto-Apply Filters**, then click **Off** to turn off the automatic apply. When you turn off the automatic apply, then each filter's selection displays an **Apply** button that you must click to apply the filter to the visualizations on the canvas.
8. Click the filter bar menu or right-click and select **Pin to All canvases** of a filter to apply a filter to all canvases in the workbook.

You can also go to the filter bar and perform the following steps:

- Select a filter and right-click, then select **Delete** to remove it from the workbook.
- Right-click and select **Clear All Filter Selections** to clear the selection list of all the filters in the filter bar.
- Right-click and select **Remove All Filters** to remove all the filters in the filter bar.

Create Filters on a Visualization

You can add filters to limit the data that's displayed in a specific visualization on the canvas.

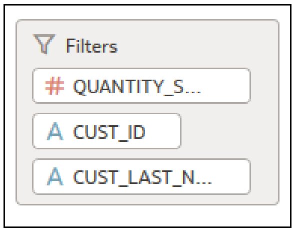
If a workbook contains multiple datasets and some aren't joined, then there are restrictions for how you can use filters. Any visualization that doesn't use the data element of the filter is grayed out.

Visualization filters can be automatically created by selecting **Drill** on the visualization's **Menu** when the **Synchronize Visualizations** option is turned off in the **Canvas Properties** dialog.

Instead of or in addition to adding filters to an individual visualization, you can add filters to the workbook or to an individual canvas. Any filters included on the canvas are applied before the filters that you add to an individual visualization.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Select the visualization that you want to add a filter to.

3. Drag and drop one or more data element from the Data Panel to the Filter drop target in the Grammar Panel.

   To use data elements of a dataset as a filter in the visualization of another dataset, you've to join both the dataset, before using the data elements as filters.

4. Set the filter values. How you set the values depends upon the data type that you're filtering.
   - To set filters on columns such as Cost or Quantity Ordered, see Apply Range Filters.
   - To set filters on columns such as Product Category or Product Name, see Apply List Filters.
   - To set filters on columns such as Ship Date or Order Date, see Apply Date Range Filters.
5. Optional: Click the filter bar menu or right-click and click **Auto-Apply Filters**, then select **Off** to turn off automatic apply for all filters on the canvas and within the visualization. When you turn off automatic apply, then each filter's selection displays an **Apply** button that you must click to apply the filter to the visualization.
6. Optional: Click **Visualization Filter Bar** in the **Filter** tab of the **Grammar Panel** to show or hide the visualization filter.

Create On-Canvas Filters

You can add a filter directly to the canvas so that users can filter the data for some or all of the visualizations on the canvas. On-canvas filters are displayed in the Visualize canvas, Present canvas, and in presentation mode, and are available for date, measure, and attribute columns.

By default the on-canvas filter's values are limited by other filters (report, canvas, on-canvas, and visualization filters). You can use the Properties pane to modify how the filters interact.

1. On the Home page, select a workbook, click the **Action Menu**, and then select **Open**.
2. Click the **Visualize** tab.
3. In the Data Elements panel, select a date, attribute, or measure to use as a filter on the canvas, and then drag it to the canvas to create a new visualization.
4. On the visualization's toolbar, click **Change Visualization Type** and select **List Box**.
5. Optional: In the on-canvas filter's list box Properties pane, select the General tab and click **Type** to specify the on-canvas filter type for:
   - **Dates** - List, Date Range (default), Relative Time
     When you select Date Range or Relative Time, then Multi Select and Default Value on-canvas filter properties aren't available.
   - **Measures** - Measure Range (default), Top Bottom N

When you select Measure Range or Top Bottom N, then Multi Select and Default Value on-canvas filter properties aren't available.

- **Attributes** - List (default), Top Bottom N
  When you select Top Bottom N, then Multi Select and Default Value on-canvas filter properties aren't available.

6. Optional: In the on-canvas filter list box Properties pane, click the General tab and adjust how you want the filter to display (for example, **Title**, **Label**, **Align**).
7. Select the on-canvas filter and set the filter values. How you set filter values depends upon the data type that you're filtering.

   See Apply Different Filter Types.
8. Optional: In the on-canvas filter's list box Properties pane, click the Filters tab, and specify how you want the on-canvas filter to interact with the canvas and visualization filters and which items on the canvas to apply the filter to.

Change the Scope of Filters Between the Main Filter Bar and Visualizations

You can change the scope of a filter by moving or copying it between the main filter bar and visualizations.

1. On the Home page, select a workbook, click the **Action Menu**, and then select **Open**.
2. Click the **Visualize** tab.
3. Change the scope of a filter by moving or copying it between the main filter bar and visualizations.
   - **Move a main filter to a visualization** - Drag and drop to move a filter from the main filter bar to the **Grammar Panel** or into the visualization.
     - When **Pin to all Canvasses** isn't selected, the filter applies to the visualization but also remains in the main filter bar because it still applies to other canvasses.
     - When **Pin to all Canvasses** is selected, the filter only applies to the visualization, and is removed from the main filter bar.
   - **Copy a main filter to a visualization** - Hold down the **Shift** key and drag and drop a filter from the main filter bar to the **Grammar Panel** or into the visualization.
   - **Move a visualization filter to the main filter bar** - Drag and drop a visualization filter from the **Grammar Panel** to the main filter bar.
     This moves the filter into the main filter bar and removes the filter from the visualization.
   - **Copy a filter between visualizations -** Drag and drop a visualization filter from the **Grammar Panel** into another visualization.
     This copies the filter into the selected visualization.
   - **Move a filter between visualizations** - Hold down the **Shift** key and drag and drop a visualization filter from the **Grammar Panel** into another visualization.
     This moves the filter into the selected visualization.

Use a Visualization as a Filter

You can configure a visualization to filter other visualizations on the canvas.

If a workbook contains multiple datasets and some aren't joined, then there are restrictions for how you can use filters. To use data elements of a dataset as a filter in the visualization of another dataset, you must join both the datasets, before using the data elements as filters. Any visualization that doesn't use the data element of the filter is dimmed.

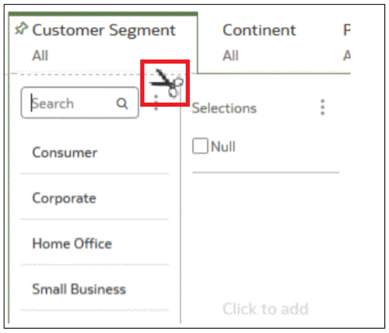
1. On the Home page, select a workbook, click the **Action Menu**, and then select **Open**.
2. Click the **Visualize** tab.
3. Select a visualization to use as a filter.
4. Click the **Use as Filter** button to activate.

Move Filter Panels

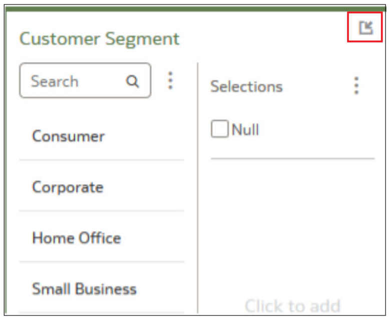
You can move filter panels from the filter bar to a different location on the canvas.

When you expand filters in the filter bar, it can block your view of the visualization that you're filtering. Moving the panels makes it easy to specify filter values without having to collapse and reopen the filter selector.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the Visualize page, place the cursor at the top of the filter panel until it changes to a scissors icon, then click it to detach the panel and drag it to another location on the canvas.



3. To reattach the panel to the filter bar, click the reattach panel icon.



Apply Different Filter Types

You apply filter types depending on the type of data that you want to filter.

Topics:

- Apply Range Filters
- Apply Top Bottom N Filters
- Apply List Filters
- Apply Date Range Filters
- Apply Relative Time Filters
- Apply Expression Filters

Apply Range Filters

You use Range filters for data elements that are numeric data types and have an aggregation rule set to something other than none.

Range filters are applied only to measure columns and limits data to a range of contiguous values, such as revenue of $100,000 to $500,000. Alternatively, you can create a range filter that excludes (as opposed to includes) a contiguous range of values. Such exclusive filters limit data to two noncontiguous ranges (for example, revenue of less than $100,000 or greater than $500,000).

If your workbook doesn't have a visualization, create one. See Begin to Build a Workbook and Create Visualizations.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the Visualize canvas, go to the filter bar or on-canvas filter, and click the filter to view the Range list.
3. In the Range list, click **By** to view the selected list of Attributes. You can perform any of the following steps:
   - Click a member to remove or add it to the selected list.
   - Click the **Plus (+)** icon to add a new member to the selected list.
   - Set the range that you want to filter on by moving the sliders in the histogram.
4. Click outside of the filter to close the filter panel.

Apply Top Bottom N Filters

Use the Top Bottom N filter to filter a measure or attribute and display its highest or lowest values.

1. To apply the Top Bottom N filter to the canvas and all visualizations in the workbook:
   a. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.

   b. In the Data panel of the visualization, select the attribute or measure that you want to filter on and drag and drop it to the filter bar.
   c. Click the filter menu and select **Filter Type**, then click **Top Bottom N**. You can only convert a range filter to Top Bottom N filter.
2. To apply the Top Bottom N filter to a specific visualization in the workbook:
   a. In the canvas, select the visualization that you want to filter.
   b. In the Data Panel, locate the attribute or measure that you want to filter on and drag and drop it to the Filter drop target in the Grammar Panel.
   c. In the Filter drop target, click the attribute or measure's name, hover over the **Filter Type** option, then click **Top Bottom N**.
3. To apply the Top Bottom N filter to an on-canvas filter:
   a. With the canvas selected, go to the Data Panel and click **Visualizations**, then click the **List Box** filter.
   b. In the Data Panel, locate the attribute or measure that you want to filter on, and drag and drop it to the List Box visualization that you just created.
   c. In the General Properties tab for the List Box, scroll down and select **Type**, and then click **Top Bottom N**.
4. To change the filter method (top or bottom), go to the **Method** value and click Top or Bottom to choose the method you want.
5. To specify the number of rows that are displayed, click the **Count** field and enter the number of rows.
6. To change which attribute or measure column to limit by, click the **By** field and select an attribute or measure or custom calculation included on the canvas. Or click **Plus (+)** to locate and select the attribute or measure or metric that you want to limit by.
7. Click outside of the filter to close the filter panel.

Apply List Filters

You apply list filters to text, non-aggregatable numbers, and dates, and you can choose which members to include or exclude from the filter.

If your workbook doesn't have a visualization, create one. See Begin to Build a Workbook and Create Visualizations.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the Visualize canvas, select a filter from the filter bar, or select an on-canvas filter to view the Selections list.
3. Locate the member you want to include and click it to add it to the Selections list. Alternatively, use the **Search** field and the **Search Options Menu** to find a member you want to add to the filter. Use the wildcards * and ? for searching.
4. Optional: Perform the following actions related to the Selections list:
   - Click a member to remove it from the list.
   - Click the eye icon next to a member to filter it out, but not remove it from the list.
   - Click **Menu** at the top, and select **Exclude Selections** to exclude members from the list.

- Click **Null** to include members with null values in the list.
- Click **Add** to add all members to the list.
- Click **Clear** to remove all members from the list.

5. Click outside of the filter to close the filter panel.

Apply Date Range Filters

Date range filters use calendar controls to adjust time or date selections. You can select a single contiguous range of dates, or use a date range filter to exclude dates within the specified range.

If your workbook doesn't have a visualization, create one. See Begin to Build a Workbook and Create Visualizations.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the Visualize canvas, select a filter from the filter bar or on-canvas filter to view the Calendar Date list.
3. In **Start**, select the date that begins the range that you want to filter.

   Use the **Previous** arrow and **Next** arrow to move backward or forward in time, or use the drop-down lists to change the month or year.
4. In **End**, select the date that ends the range that you want to filter.
5. Optional: To remove all dates for a filter in the filter bar, right-click the filter and select **Clear Filter Selections**.
6. Click outside of the filter to close the filter panel.

Apply Relative Time Filters

Use the relative time filter on a Date or Date/Time column to display data for a specified time period based upon the current date and time.

You can specify a relative time period as either an explicit number of past or future time units (for example 2 years), or you can specify a previous period. For example, Year To Date which includes data from 1-January this year to the current date, and Month To Date which includes data from the beginning of the month to the current date.

You can apply a Relative Time filter only to date columns that already exist in the data source, and not to derived columns such as Year, or Quarter. The Relative Time filter type supports Date (with no time of day portion) and DateTime (that is, TIMESTAMP with both date and time of day) column types.

The current date and time used in queries is the Oracle Analytics server host's date and time in the server's timezone (not the browser host's time or timezone). The starting day of the week (Sunday versus Monday) is based on the locale of the Oracle Analytics server which is set using the server configuration setting NLS_TERRITORY.

If your workbook doesn't have a visualization, create one. See Begin to Build a Workbook and Create Visualizations.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.

2. In the Visualize canvas, go to the filter bar or to the canvas and select a date range filter.
3. Click the filter context menu, select **Filter Type**, and **Relative Time** to switch to a relative time filter.
4. In the **Relative Time** filter, select a **Type** that defines the range that you want to filter.
   - **Last -** You specify a **Period** to apply to the selected **Time Level** (Years, Quarters, Months, Weeks, Days, and includes Hours, Minutes, and Seconds if the column time is TIMESTAMP) relative to today's date, to display records for the date values for that period.
     Last filters that are based on a DateTime column and which have a grain of Day or longer (for example, Year, Quarter, Month, Week, Day), retrieve data from the same time of day on the starting day. For example, if the server date/time is currently Thursday 3:15pm , a Last 2 Days filter on a DateTime column retrieves data with timestamps between Tuesday 3:15pm and Thursday 3:15pm in the server's timezone. Filter queries that are based on a DATE column type (which by definition have no time of day associated) only depend on the server host's date, not the time of day.
   - **Next -** You specify a future **Period** number to apply to the selected **Time Level** (Years, Quarters, Months, Weeks, Days, also Hours, Minutes, and Seconds if the column time is TIMESTAMP) relative to today's date, to display records for the date values for that period.
   - **To Date -** You specify a past **Time Level** (Year, Quarter, Month, Week, Day, also includes Hour, and Minute if the column time is TIMESTAMP) relative to today's date, that you want to apply to the date values in the visualization.
     A To Date filter retrieves data from the beginning of the chosen period, for example, Month to Date retrieves data from midnight of the first day of this month up until the current date and time (that is, Today or Now).
5. Click outside of the filter to close the filter panel.

Apply Expression Filters

Using expression filters, you can define more complex filters using SQL expressions. Expression filters can reference zero or more data elements.

For example, you can create the expression filter "Sample Sales"."Base Facts"."Revenue" < "Sample Sales"."Base Facts"."Target Revenue". After you apply the filter, you see the items that didn’t achieve their target revenue.

You create expressions using the Expression Filter panel. You can drag and drop data elements to the Expression Filter panel and then choose functions to apply. Expressions are validated for you before you apply them.

If your workbook doesn't have a visualization, create one. See Begin to Build a Workbook and Create Visualizations.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. In the Visualize canvas, go to the filter bar and click **Add Filter** and then select **Add Expression Filter**.
3. In the Expression Filter panel, compose an expression in the **Expression** field.
4. In the **Label** field, give the expression a name.
5. Click **Validate** to check if the syntax is correct.

6. When the expression filter is valid, then click **Apply**. The expression applies to the visualizations on the canvas.

9
Use Other Functions to Visualize Data

This topic describes other functions that you can use to visualize your data.

Topics:

- Build Stories
- Identify Content with Tooltips
- Manage Custom Plug-ins
- Compose Expressions
- Use Data Actions to Connect to Canvases, External URLs, Oracle Analytics Publisher Reports
- Visualize Data from the Home Page
- Save Your Changes Automatically
- Sort the Items in a Page

Build Stories

This topic covers how you capture insights and group them into stories.

Topics:

- Capture Insights
- Create a Story
- Add Notes to a Visualization
- View a Story in Present Mode

Capture Insights

As you explore data in visualizations, you can capture memorable information in one or more insights, which build your story. For example, you might notice before and after trends in your data that you'd like to add to a story to present to colleagues.

Tutorial

Using insights, you can take a snapshot of any information that you see in a visualization and keep track of any moments of sudden realization while you work with the data. You can share insights in the form of a story, but you don't have to. Your insights can remain a list of personal moments of realization that you can go back to, and perhaps explore more. You can combine multiple insights in a story. You can also link insights to visualizations using the Interaction property.

Insights don't take a snapshot of data. They take a snapshot of the workbook definition at a certain point in time.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click **Present**, and use the story designer options to add your insights:
   - To add a canvas to your story, in the **Canvases** panel, right click a canvas and select **Add To Story**.
     To locate other canvases in the workbook, use the **Search** option in the Canvases pane.
   - Use the tabs on the Properties pane in the bottom left-hand corner to further refine your story. For example, click **Presentation** to change the presentation style from compact to film strip.
   - To rename, duplicate, hide, or delete a page in your story, right click a canvas displayed in the canvas selector at the bottom of the design area, and select a canvas option.
   - To synchronize your story canvases with your visualizations, display the Visualize pane and right-click the canvas tabs to select **Canvas Properties**, then set **Synchronize Visualizations** to **On**.
3. Continue adding insights to build a story about your data exploration.

   The story builds in the Present canvas. Each insight has a tab.
4. Click **Save** to save your story as part of your workbook.

Create a Story

A workbook can have one story that includes one or more pages. Each page contains a canvas. You can cultivate the look and feel of the story to meet your presentation needs. For example, you can rearrange pages, include another page, or hide a page's title.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click **Present**.
3. In the Present panel, drag and drop one or more canvases to the story.
4. Use the following options to build your story:
   - Add one or more canvases to the story. After you add a canvas to the story, go to the bottom of the Present page, click the page, and use the Properties panel to specify how you want the page to display in the story.
   - To include or exclude a canvas' notes, select the canvas and go to the canvas' Properties panel. Click **Notes** and specify which notes to display or hide. To add notes, go to the Visualize page and click **Add Note**.
   - To delete a page from the story, right-click the page at the bottom of the Present page and click **Delete**.
   - To show or hide a page's titles or descriptions, go to the **Properties** panel, click **General**, and use the **Hide Page**, **Page Title**, and **Description** options.
   - To rearrange pages, drag and drop them in the pages area at the bottom of the Present page.

- To limit the data displayed in a page, go to the Properties panel and click **Filters**. If no filters are displayed, then go to the Visualize page and use the **Filters** bar to add filters.
- To include or exclude a canvas' filters, select a canvas and go to the canvas' Properties panel. Click **Filters**, and specify which filters to apply to the canvas.
- To rename a page, click the page's title and update it.
- To add the same page multiple times to a story, right-click the page at the bottom of the Present canvas and click **Duplicate**.
- To provide present mode functionality such as adding new filters or adjusting the existing filter values, or providing the context menu to export a visualization, sort a visualization, or drill within a visualization, go to the Properties panel, click the Presentation tab, and set the **Full Interactivity** field to **On**. The interactivity mode that you specify is applied to all pages in the story.

5. To display the story and check your work, click **Present**. Then click **Go back** to return to the **Present** panel.
6. Click **Save** to save the story as part of the workbook.

View a Story in Present Mode

You can use present mode to view and interact with a story without the visual clutter of the Present canvas toolbar and authoring options.

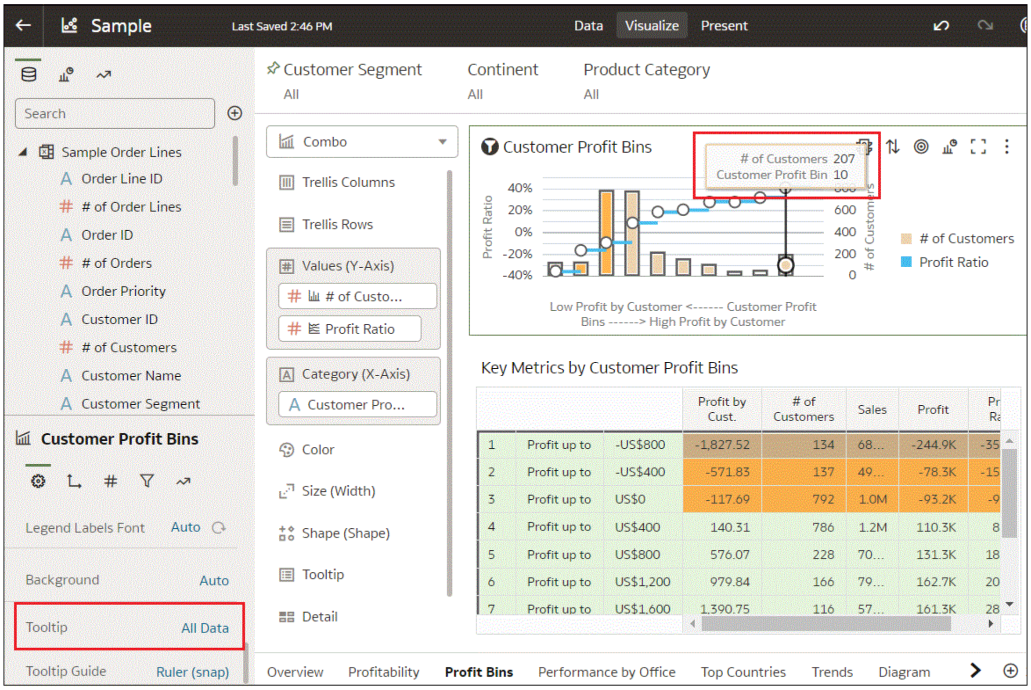
If the story's **Full Interactivity** field is set to **On**, then while in present mode, you can access and use context menus to explore a visualization's data. For example, you can add new filters or adjust the existing filter values, export to common formats such as Powerpoint, PDF, PNG, or CSV, or use the right-click menu to sort a visualization or drill within a visualization. The type of visualization and its data determine the available context menu options.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click **Present** and in the story, click the page that you want to view.
3. If the data is stale, click **Refresh Data**.
4. To print, email, or export the page, click **Export** and choose an export option.
5. To hide or show the page's notes, click **Hide Notes**.
6. To adjust the page's filter values, go to the filters bar and click the filter that you want to adjust the values for.
7. To add a filter to the page, go to the filter bar and click **Add Filter**, select the column that you want to filter, and specify the filter's values.
8. Right-click a visualization to perform visualization or data-specific tasks, such as apply a sort order, zoom in, or drill.
9. To return to the Present tab, go to the story's header and click **Go back**.

Identify Content with Tooltips

Use tooltips to make your visualizations more interactive without cluttering them with too much information.

In the visualization designer page, use the Tooltip pane to display measures and labels when users click on a measure or hover over a visualization.



Legend Labels Font Auto

Background Auto

Tooltip All Data

Tooltip Guide Ruler (snap)

Key Metrics by Customer Profit Bins

			Profit by Cust.	# of Customers	Sales	Profit	Pr Ra
1	Profit up to	-US$800	-1,827.52	134	68...	-244.9K	-35
2	Profit up to	-US$400	-571.83	137	49...	-78.3K	-15
3	Profit up to	US$0	-117.69	792	1.0M	-93.2K	-9
4	Profit up to	US$400	140.31	786	1.2M	110.3K	8
5	Profit up to	US$800	576.07	228	70...	131.3K	18
6	Profit up to	US$1,200	979.84	166	79...	162.7K	20
7	Profit up to	US$1,600	1,390.75	116	57...	161.3K	28

Manage Custom Plug-ins

You can upload, download, search for, and delete custom plug-ins. Plug-ins are custom visualization types or custom data actions that you create externally and then import into your system.

For example, you can upload a custom plug-in that provides a visualization type that you can use in workbooks.

Tutorial

1. On the Home page, click the **Navigator**, and then click **Console**.
2. Click **Extensions**.

   You use this page to upload, search for, delete, or download a custom plug-in.
3. To upload a custom plug-in, click **Extension** and perform one of the following actions.
   - Browse to the required plug-in file in your file system, and click **Open** to select the plug-in.
   - Drag the required plug-in file to the **Upload Custom Plugin** object.

   If the uploaded custom plug-in file name is the same as an existing custom plug-in, then the uploaded file replaces the existing one and is displayed in visualizations.
4. Perform any of the following tasks.
   - If the plug-in provides a visualization type, you can select that type from the list of available types when you create or switch the type of a visualization.
   - To search for a custom plug-in, enter your search criteria in the **Search** field and click **Return** to display search results.

- To delete a custom plug-in, click **Options** on the custom plug-in and select **Delete**, and click **Yes** to delete the custom plug-in.
  If you delete a custom visualization type that's used in a workbook, then that workbook displays an error message in place of the visualization. Either click **Delete** to remove the visualization, or upload the same custom plug-in so that the visualization renders correctly.
- To download a custom plug-in, click **Options** on the custom plug-in and select **Download**.

Compose Expressions

You can use the Expression window to compose expressions to use in expression filters or in calculations. Expressions that you create for expression filters must be Boolean (that is, they must evaluate to true or false).

While you compose expressions for both expression filters and calculations, the end result is different. A calculation becomes a new data element that you can add to your visualization. An expression filter, on the other hand, appears only in the filter bar and can't be added as a data element to a visualization. You can create an expression filter from a calculation, but you can't create a calculation from an expression filter. See Create Calculated Data Elements and Build Expression Filters.

You can compose an expression in various ways:

- Directly enter text and functions in the Expression window.
- Add data elements from the Data Elements pane (drag and drop, or double-click).
- Add functions from the function panel (drag and drop, or double-click).

See Expression Editor Reference.

Use Data Actions to Connect to Canvases, External URLs, and Use in Publisher Reports

A Data Action link can pass context values as parameters to external URLs, filters to other workbooks or to visualizations embedded in external containers.

When a link navigates to a workbook, the data context is displayed in the form of canvas scope filters in the filter bar. The links data context may include attributes associated with the selections or cell from which the link was initiated.

Topics:

- Create Data Actions to Connect Visualization Canvases
- Create Data Actions to Connect to External URLs from Visualization Canvases
- Create Data Actions to Connect to REST APIs from Visualization Canvases
- Use Data Actions to Connect to Oracle Business Intelligence Publisher Reports
- Invoke Data Actions from Visualization Canvases

Create Data Actions to Connect Visualization Canvases

You can create data actions to navigate to a canvas in the current workbook or to a canvas in another workbook.

▶ Video

You can also use data actions to transfer context-related information (for example, an order number) where the link displays details about an order number in another visualization or workbook.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click **Menu** on the workbook toolbar and click **Data Actions**.
3. Click **Add Action** and enter a name for the new navigation link.
   - You can use only letters and numbers in the navigation link's name.
   - You can add multiple navigation links.
4. Click the **Type** field and select **Analytics Link**.
5. Click the **Anchor To** field and select the columns from the current visualization to associate with this data action. Don't select measure columns or hidden columns. If you don't specify a value for the **Anchor To** field, then the data action applies to all data elements in the visualizations.
6. Click the **Workbook** field and select the workbook you want to use for the anchor:
   - **Use This Workbook** - Select if you want to link to a canvas in the active workbook.
     Columns that you select must be in the current visualization.
   - **Select from Catalog** - Select to browse for and select the workbook that you want to use.
7. Click the **Canvas Link** field and select the canvas that you want to use.
8. Click the **Pass Values** field and select which values you want the data action to pass.

   For example, if in the **Anchor To** field, you specified order number column, then in the **Pass Values** field, select **Anchor Data** to pass the specified column values.
   - **All** - Dynamically determines the intersection of the cell that you click and passes those values to the target.
   - **Anchor Data** - Ensures that the data action is displayed at runtime, but only if the required columns specified in the **Anchor To** field are available in the view context.
   - **None** - Opens the page (URL or canvas) but doesn't pass any data.
   - **Custom** - Enables you to specify a custom set of columns to pass.
9. Click **Supports Multiple Selection** to set the value.
   - **On** - The data action can be invoked when one or multiple data points are selected.
   - **Off** - The data action can only be invoked when a single data point is selected.

This setting is particularly useful when the selection of multiple data points might result in an error (for example, with some third-party REST APIs).

10. Click **OK** to save.

Create Data Actions to Connect to External URLs from Visualization Canvases

You can use data actions to navigate to an external URL from a canvas so that when you select a column such as the supplier ID, it displays a specific external website.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click **Menu** and select **Data Actions**.
3. Click **Add Action** and enter a name for the new navigation link.

   You can add multiple navigation links.
4. Click the **Type** field and select **URL Navigation**.
5. Click the **Anchor To** field and select the columns that you want the URL to apply to. If you don't specify a value for the **Anchor To** field, then the data action applies to all data elements in the visualizations.
6. Enter a URL address and optionally include notation and parameters.

   For example, where `http://www.example.com?q=${keyValuesForColumn:"COLUMN"}` is displayed like `www.oracle.com?q=${keyValuesForColumn:"Sales"."Products"."Brand"}` The column names that you select here are replaced with values when you invoke the data action.
7. Click **Supports Multiple Selection** to set the value.
   - **On** - The data action can be invoked when one or multiple data points are selected.
   - **Off** - The data action can only be invoked when a single data point is selected. This setting is particularly useful when the selection of multiple data points might result in an error (for example, with some third-party REST APIs).
8. Click **OK** to save.
9. In the **Canvas**, click a cell, or use Ctrl-click to select multiple cells.
10. Right-click and select from the menu the navigation name that you created earlier.

    Selecting the cells determines the values to pass to the parameters (that is, the URL tokens).

Create Data Actions to Connect to REST APIs from Visualization Canvases

You can use HTTP API data actions in a canvas so that when you select a column, it sends the value to a REST API which returns a response from the external website.

You must configure the domain that you're trying to connect to as a safe domain before you create an HTTP API data action. See Register Safe Domains.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click **Menu** and then select the **Data Actions** tab.

3. Click **Add Action** and enter a name for the new HTTP API data action. For example, enter `HTTP API Example`.

   You can add multiple HTTP API data actions.

4. Click the **Type** field and select **HTTP API**.

5. Click the **Anchor To** field and select the columns that you want the HTTP API data action to apply to. Don't select measure columns or hidden columns. If you don't specify a value for the **Anchor To** field, then the data action applies to all data elements in the visualizations.

6. Click the **HTTP Method** field and select an appropriate value (that is, GET, POST, PUT, DELETE) to send to the REST API.

7. Enter the URL for the REST API that starts with http or https and optionally includes replacement tokens.

   Replacement tokens in the URL pass contextual values to a data action (for example, data values, user name, workbook path, canvas name).

   For example, you can pass a column value (for example, a value for the Category column) to a REST API (for example, the Google books API) using: `http://www.googleapis.com/books/v1/volumes?q=${valuesForColumn:"Category"}` The value that you select from a cell in the Category column (for example, "Books") passes to the REST API.

8. If you selected the POST method, a field is displayed where you enter each parameter on a separate line.

   You enter each parameter as a name-value pair with the name and value separated by "**=**". You can use the same URL token syntax in the name-value pairs as required by the API that you're calling.

   For example:

   - `paramName1=paramValue1`
   - `paramName2=${valuesForColumn:"Product"}`

9. Click **Supports Multiple Selection** to set the value.

   - **On** - The data action can be invoked when one or multiple data points are selected.
   - **Off** - The data action can only be invoked when a single data point is selected. This setting is particularly useful when the selection of multiple data points might result in an error (for example, with some third-party REST APIs).

10. Click **OK** to save.

11. Click a data point in the visualization.

    For example, you might select "Books" from the Category column.

12. Right-click and select **HTTP API Example** to display the result.

    The selected data-points determine which values to pass to the URL tokens.

    A success or failure message is displayed confirming the REST API has been successfully invoked using the selected value.

Use Data Actions to Connect to Oracle Business Intelligence Publisher Reports

A data action link can pass context values as URL parameters in Oracle Business Intelligence Publisher reports.

When you click a link to open a BI Publisher report, the link might include attributes associated with the column selected in the visualization.

Topics:

- About Creating Data Actions to Connect to Oracle Business Intelligence Publisher Reports
- Create Data Actions to Connect to Oracle Business Intelligence Publisher Reports
- Create Custom Column Names in Data Actions to Pass in Oracle Business Intelligence Publisher Report URLs

About Creating Data Actions to Connect to Oracle Business Intelligence Publisher Reports

You can create a data action to link to an Oracle Business Intelligence Publisher report.

You configure a data action to pass selected column values from a visualization to display in a BI Publisher report.

- You must store your BI Publisher report in a local folder.
- When the BI Publisher report uses the analysis as the data model, prompts in the underlying analysis must have **User Input** value set to **Choice List** to enable selected values to pass to prompts in the BI Publisher report.
- You can pass lists of values and list filters to display as prompts in your BI Publisher report. However, you can't pass number filters, date filters, or expression filters.

Create Data Actions to Connect to Oracle Business Intelligence Publisher Reports

You can create an analytics data action link to transfer selected data points from an Oracle Analytics workbook to an Oracle Business Intelligence Publisher report.

The Oracle Analytics workbook, BI Publisher report, and analysis can be in different folders.

1. On the Home page, select a workbook that uses the data model used in the BI Publisher report, click the **Actions menu**, and then select **Open**.
2. Click **Menu** and select **Data Actions**.
3. Click **Add Action** and enter a name for the new navigation link.
4. Click the **Type** field and select **Analytics Link**.
5. Enter a name for the data action in the **Name** field.
6. Click the **Target** field, select **Select from Catalog**, then browse to select the BI Publisher report that you want the data action to pass data to, and click **OK**.
7. Verify that **Parameter Mapping** is set to `Default`.

8. Verify the **Pass Values** field value is set to `All`.
9. Click **Supports Multiple Selection** to set the value.
    - **On** - The data action can be invoked when one or multiple data points are selected.
    - **Off** - The data action can only be invoked when a single data point is selected. This setting is particularly useful when the selection of multiple data points might result in an error (for example, with some third-party REST APIs).
10. Click **OK**.
11. Select data points in the visualization and choose the data action to test that the values are passed to the BI Publisher report.

Create Custom Column Names in Data Actions to Pass in Oracle Business Intelligence Publisher Report URLs

You can create a data action link that passes custom column names in the URL of an Oracle Business Intelligence Publisher report.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. Click **Menu** and select **Data Actions**.
3. Click **Add Action** and enter a name for the new navigation link.
4. Click the **Type** field and select **Analytics Link**.
5. Enter a name for the data action in the **Name** field.
6. Click the **Target** field, select **Select from Catalog**, then browse to select the BI Publisher report that you want the data action to pass data to, and click **OK**.
7. Click **Parameter Mapping** and select **Set Custom Mapping** to specify custom column names to pass as URL parameters to the BI Publisher report.
8. Click **Add Row** and click **Select a Column** for each column that you want to pass to the BI Publisher report using a custom name.
9. Click **Enter Parameter** and enter a custom name for each column name that you want to replace.

    The custom name passes to the BI Publisher report in the URL.
10. Verify the **Pass Values** field value is set to `All`.
11. Click **Supports Multiple Selection** to set the value.
    - **On** - The data action can be invoked when one or multiple data points are selected.
    - **Off** - The data action can only be invoked when a single data point is selected. This setting is particularly useful when the selection of multiple data points might result in an error (for example, with some third-party REST APIs).
12. Click **OK**.
13. Select data points in the visualization and choose the data action to verify that the custom column names are displayed in the URL of the BI Publisher report.

Invoke Data Actions from Visualization Canvases

You can invoke data actions from a canvas to other canvases and URLs.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. On the canvas that contains a Data Action link leading to another canvas or URL, perform the following steps:
    a. Right-click a data element, or select multiple elements (using Ctrl-click).
    b. Select **Data Actions** from the context menu.
    c. Complete the Workbook Properties dialog.

    The name of the data actions that apply in the current view context are displayed in the context menu.

    All the values defined in the **Anchor To** field must be available in the view context in order for a data action to be displayed in the context menu.

    The following rules apply to matching data elements passed as values with data elements on the target canvas:

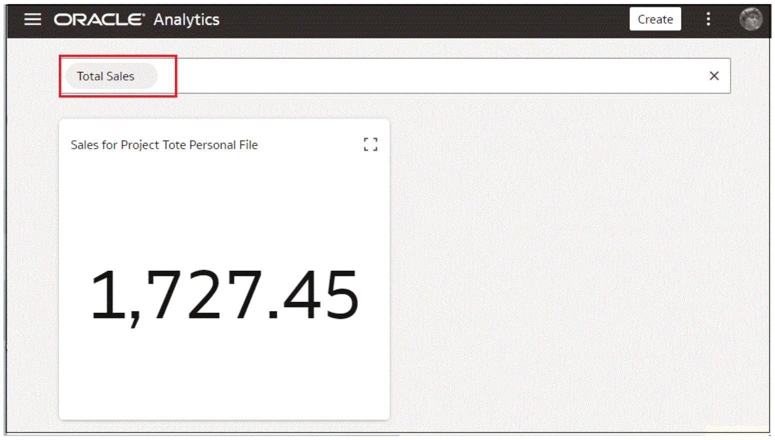
    - If the same data element is matched in the target workbook's canvas, and if the target canvas doesn't have an existing canvas filter for the data element, a new canvas filter is added. If there is an existing canvas filter, it’s replaced by the value from the source workbook's canvas.
    - If the expected dataset is unavailable but a different dataset is available, the match is made by using the column name and data type in the different dataset, and the filter is added to that.
    - If there are multiple column matches by name and data type, then the filter is added to all those columns in the target workbook or canvas.

    The data action navigates to the target cell or URL that is mapped and filters the data displayed based on the values specified in the Data Actions dialog.
    The Pass Values context, set in the **Pass Values** field, consists of data elements used in the visualization from which the data action is invoked. The Pass Values context doesn't include data elements in the workbook, canvas, or visualization level filters.

Visualize Data from the Home Page

Use the search bar on the Home page to find your data and quickly visualize it. You can use this functionality to perform impromptu visualizations without having to first build a workbook.

1. On the Home page, click the search bar.
2. Enter a search term (for example, "Office"), then press SHIFT + ENTER or select a specific result displayed in the drop-down list. You'll see your data visualized.
    - What you select determines the dataset for the visualization, and all other criteria that you enter is limited to columns or values in that dataset.



3. Build your visualization by searching for and selecting other items.
4. When you're happy with your visualization and you'd like to save it or further enhance it, right-click your visualization and choose **Open in Data Visualization**.

Find Data, Workbooks, and Visualizations

This topic describes how you can search for items such as objects, workbooks, and columns.

Topics:

- How Is Data Indexed?
- Search for Content
- Advanced Search Commands for Finding Content
- Search Tips

How Is Data Indexed?

When you search or visualize data from the Home page, your results are determined by what information has been indexed.

The system runs a process to index your saved objects, workbook content, and dataset column information. The indexing process also updates the index file to reflect any objects, workbooks, or datasets that you deleted from your system so that these items are no longer displayed in your search results.

For all datasets, the column metadata is indexed. For example, column name, the data type used in the column, aggregation type, and so on. Column data is indexed for Excel spreadsheet, CSV, and TXT dataset columns with 1,000 or fewer distinct rows. Note that no database column data is indexed and therefore that data isn’t available in your search results.

Search for Content

Use the search bar on the Home page to find items such as datasets, workbooks, and machine learning scripts.

1. On the Home page, locate the search bar.
2. Enter your search criteria. Note the following options:
   - Click in the search bar for a drop down list of all content types, such as workbook, report, and dataset. Click a content type to add it to the search bar. Or below the search bar, click one of the search tags to add it to the search bar.
   - Build or modify a search tag by adding or removing other items.
   - Specify the full or partial name of what you're looking for. The search is case-insensitive.
   - To clear your search terms, in the search bar click X or select search tags and delete.
3. In the search results, click an object to display it.

Advanced Search Commands for Finding Content

You can enter advanced search commands in the search bar to tailor your search results for exact matches, multi-term matches, and field-level matches.

You can combine multiple search terms with commands to narrow or widen your search. For example, name:(revenue AND Analysis). Search commands and search terms are case-insensitive.

Search Command	Description	Example
AND	Enter AND between search terms to only display content which contains all the search terms. All forms of AND, such as, and, &&, or entering two search terms together, will return the same results.	Revenue AND Forecast Revenue and Forecast Revenue && Forecast Revenue Forecast
OR	Enter OR between search terms to display content that contains any of the search terms.	Revenue OR Profit Revenue or Profit Revenue || Profit
NOT	After entering a search term, enter NOT followed by more search terms to exclude any content from the results which match the search terms entered after the NOT command.	Revenue NOT Product Revenue not Product
?	Enter the question mark (?) character in a search term as a wildcard to signify a single unknown character. This ensures that the search results include content with words matching the known characters from the search term. For example, searching for st?r would include results containing star and stir.	st?r
*	Enter the asterisk (*) character at the end of a partial search term or root word as a wildcard to find all content that contains the partial search term as well as content that contain variations of the root word. For example, searching for employ* would include results for employee, employment, or employer.	Employ*

Search Command	Description	Example
`name:`	Enter `name:` followed by a search term to search for content where the search term is part of the Name field.	`name:`Revenue Analysis
`description:`	Enter `description:` followed by a search term to search for content where the search term is contained in the Description field of the content.	`description:`template `desc:`template
`owner:`	Enter `owner:` followed by a search term to search for content where the search term is contained in the Owner field of the content.	`owner:`Admin
`columns:`	Enter `columns:` followed by a search term to search for objects that reference columns matching the search term.	`columns:`product
`text:`	Enter `text:` at the beginning of a search followed by a search term to search for content where the search term is contained in any of the content's fields.	`text:`Revenue
`""`	Enclose a search term with double quotation marks to search for content that includes phrases or stop words which match the search term.	"Balance Letter" "Research by analysis"

Search Tips

Use these tips to help you find your content.

- **Searching in Non-English Locales** - When you enter criteria in the search field, what displays in the drop-down list of suggestions can differ depending upon your locale setting. For example, if you're using an English locale and enter *sales*, then the drop-down list of suggestions contains items named *sale* and *sales*. However, if you're using a non-English locale such as Korean and type sales, then the drop-down list of suggestions contains only items that are named *sales* and items such as *sale* aren't included in the drop-down list of suggestions.
- **Searching for New Objects and Data -** If you create or save a workbook or create a dataset and then immediately try to search for it, then it's likely that your search results won't contain matches. If this happens, refresh your browser. If you still can't find the new object or data, then wait a few minutes for the indexing process to run, and retry your search. Users can access only the data they've been given permission to access.

Save Your Changes Automatically

You can use the auto save option to save your updates to a visualization workbook in real-time automatically.

If you've already saved your workbook in a specific location, the Save Workbook dialog isn't displayed after you click **Auto Save**.

1. On the Home page, select a workbook, click the **Actions menu**, and then select **Open**.
2. From the Save menu, select **Auto Save**.

3. In the Save Workbook dialog, enter the **Name** and optional **Description** to identify your workbook.
4. Select the folder where you want to save your workbook.
5. Click **Save**. Any workbook updates are saved in real-time.

Suppose that two users are updating the same workbook and **Auto Save** is enabled. The **Auto Save** option is automatically disabled when different types of updates are made to the workbook. A message is displayed that states that another user has updated the workbook.

Sort the Items in a Page

You can quickly reorganize items in the Catalog, Data, and Machine Learning pages by sorting the items based on their attributes.

As you add more data or workbooks, sorting items helps you to control the order in which they're listed. For example, you can reorganize datasets on the Data page based on their modification dates and times.

1. On the Home page, click **Navigator**, and then click **Console**.
2. Click either **Catalog**, **Data**, or **Machine Learning** depending on which items you want to sort.
3. Click the **Sort By** menu on the page toolbar and select a sort option such as **Modified** or **Reverse Order**.
4. Click the **List View** icon to switch to list view. In the table, click a column header to sort the items in that column in ascending or descending order.

The selected sort and column options are saved as user preferences.

10

Create Custom Data Action Plug-ins

You can create custom data action plug-ins to use in Oracle Analytics.

Data action plug-ins extend Oracle Analytics and enable users to select data-points in visualizations and to invoke specific actions. Oracle Analytics provides a core set of data actions that cover many common use cases, but by writing your own data action plug-in, you can extend this functionality even further.

The tutorial uses an example to help you understand how to create a custom data action plug-in. Tutorial

You must have a basic understanding of the following to create custom data action plug-ins:

- JavaScript
- RequireJS
- JQuery
- KnockoutJS

Topics:

About Data Action Plug-ins and the Data Actions Framework

Data action plug-ins leverage the data actions framework to provide custom, data-driven actions that are tightly integrated into the Oracle Analytics user interface.

When a user invokes a data action, the Data Action Manager passes the request context (for example, qualified data reference, measure values, filters and metadata) to the data action plug-in which is responsible for handling the request. Oracle provides four types of data action plug-ins: `CanvasDataAction`, `URLNavigationDataAction`, `HTTPAPIDataAction` and `EventDataAction`. You can extend these data action plug-in types along with their abstract base classes to provide your own data actions.

Topics:

Data Action Categories

The data action categories include Navigate to URL, HTTP API, Navigate to Canvas, and Event actions:

- **Navigate to URL**: Opens the specified URL in a new browser tab.
- **HTTP API**: Uses the `GET/POST/PUT/DELETE/TRACE` commands to target an HTTP API and doesn't result in a new tab. Instead the HTTP status code is examined and a transient success or failure message is displayed.
- **Navigate to Canvas**: Enables the user to navigate from a source canvas to a target canvas in either the same or a different visualization. Any filters that are in effect in the source canvas are passed to the target canvas as external filters. When the target canvas opens, it attempts to apply the external filters to the visualization. The mechanism by which external filters are applied isn't described here.
- **Event Actions**: Publishes an event using the Oracle Analytics event router. Any JavaScript code (for example, a third-party plug-in) can subscribe to these events and handle their custom response accordingly. This provides the maximum flexibility because the plug-in developer can choose how the data action responds. For example, they can choose to display a user interface or pass data to multiple services at once.

Both the **Navigate to URL** and **HTTP API** data action category types can use a token syntax to inject data or metadata from the visualization into the `URL` and `POST` parameters.

URL Token Replacement

HTTP data actions can replace tokens in URLs with values from the context passed to the data action. For example, qualified data reference values, filter values, username, workbook path, and canvas name.

Token	Notes	Replace With	Example	Result
`${valuesForColumn:COLUMN}`	NA	Column display values from the qualified data reference.	`${valuesForColumn:"Sales"."Products"."Brand"}`	`BizTech,FunPod`

Token	Notes	Replace With	Example	Result
`${valuesForColumn:COLUMN, separator:"/"}`	Any token that can potentially be replaced with multiple values supports the optional separator option. The `separator` defaults to a comma (,) but you can set it to any string. You can escape double quotes inside this string by using a backslash (\).	Column display values from the qualified data reference.	`${valuesForColumn:"Sales"."Products"."Brand"}`	`BizTech,FunPod`

`${valuesForColumn:COLUMN, separationStyle:individual}`	Any `separationStyle` defaults to `delimited` but you can set it to `individual` if the user needs to generate separate URL parameters for each value.	Column display values from the qualified data reference.	`&myParam=${valuesForColumn:"Sales"."Products"."Brand"}`	`&myParam=BizTech`
				`&myParam=FunPod`

`${keyValuesForColumn:COLUMN}`	NA	Column key values from the qualified data reference.	`${keyValuesForColumn:COLUMN}`	`10001,10002`
`${env:ENV_VAR}`	Supported environment variables are: `sProjectPath`, `sProjectName`, `sCanvasName`, `sUserID`, and `sUserName`.	An environment variable.	`${env:'sUserID'}`	`myUserName`

Data Action Context

You can define a context that is passed when the user invokes a data action.

You define how much of the context is passed to the data action when you create the data action.

Qualified Data Reference

When the data action is invoked a qualified data reference is generated for each marked data point using an array of `LogicalFilterTree` objects. A `LogicalFilterTree` consists of multiple `LogicalFilterNode` objects arranged in a tree structure. This object includes:

- The attributes on the row or column edges of the data layout.
- The specific measure on the measure edge that addresses each marked cell.
- The specific measure value for each marked cell.
- Key values and display values.

Environment Variables

In addition to the data and metadata describing each marked data point, certain data actions may need further context describing the environment from where the data action is invoked. Such environment variables include:

- Project Path
- Project Name
- Canvas Name
- User ID
- User Name

Data Action Code Design

You create data actions using API classes.

- There are four concrete classes of data action that inherit from the `AbstractDataAction` class:
  - `CanvasDataAction`
  - `URLNavigationDataAction`
  - `HTTPAPIDataAction`
  - `EventDataAction`
- You can create new types of data actions using the data action plug-in API. See Data Visualizer SDK Reference.
- The registry of data action types is managed by the `DataActionPluginHandler`.
- Code that creates, reads, edits, deletes, or invokes instances of data actions does so by publishing events.
- Events are handled by the `DataActionManager`.

Data Action Model Classes

There are several different types of data action model classes.

AbstractDataAction

This class is responsible for:

- Storing the Knockout Model (subclasses are free to extend this with their own properties).
- Defining the abstract methods that subclasses must implement:
  - `+ invoke(oActionContext: ActionContext, oDataActionContext:DataActionContext) <<abstract>>`
    Invokes the data action with the passed context - should only be called by the `DataActionManager`.
  - `+ getGadgetInfos(oReport): AbstractGadgetInfo[] <<abstract>>`
    Constructs and returns the `GadgetInfos` responsible for rendering the user interface fields for editing this type of data action.

  - `+ validate() : DataActionError`
    Validates the data action and returns null if valid or a `DataActionError` if it's invalid.

- Providing the default implementation for the following methods used to render generic parts of the data action user interface fields:
  - `+ getSettings():JSON`
    Serializes the data action's Knockout Model to JSON ready to be included in the report (uses komapping.toJS(_koModel)).
  - `+ createNameGadgetInfo(oReport) : AbstractGadgetInfo`
    Constructs and returns the `GadgetInfo` that can render the data action's **Name** field.
  - `+ createAnchorToGadgetInfo(oReport) : AbstractGadgetInfo`
    Constructs and returns the `GadgetInfo` that can render the data action's **Anchor To** field.
  - `+ createPassValuesGadgetInfo(oReport) : AbstractGadgetInfo`
    Constructs and returns the `GadgetInfo` that can render the data action's **Pass Values** field.

Subclasses may not need all of the `GadgetInfos` that the base class provides so they may not need to call all of these methods. By separating out the rendering of each field in this way, subclasses are free to pick and choose the gadgets they need. Some subclasses may even choose to provide a different implementation of these common data action gadgets.

CanvasDataAction, URLNavigationDataAction, HTTPAPIDataAction, EventDataAction

These are the concrete classes for the basic types of data actions. These classes work by themselves to provide the generic user interface for these types of data action. They can also act as convenient base classes for custom data action plug-ins to extend.

- **CanvasDataAction**: Used to navigate to a canvas.
- **URLNavigationDataAction**: Used to open a web page in a new browser window.
- **HTTPAPIDataAction**: Used to make a `GET/POST/PUT/DELETE/TRACE` request to an HTTP API and handle the `HTTP Response` programatically.
- **EventDataAction**: Used to publish JavaScript events through the Event Router.

Each class is responsible for:

- Implementing the abstract methods from the base class.
  - `invoke(oActionContext: ActionContext, oDataActionContext:DataActionContext)`
    This method should invoke the data action by combining the properties defined in the KOModel with the specified `DataActionContext` object.
  - `getGadgetInfos(oReport): AbstractGadgetInfo[]`
    This method should:
    - Create an array containing `AbstractGadgetInfos`.
    - Call individual `createXXXGadgetInfo()` methods pushing each `AbstractGadgetInfo` into the array.
    - Return the array.
- Providing the additional methods for creating the individual gadgets that are specific to the particular subclass of data action.

Subclasses of these concrete classes may not need to use all of the gadgets provided by their superclasses in their custom user interfaces. By separating out the construction of each gadget in this way, subclasses are free to pick and choose the gadgets they need.

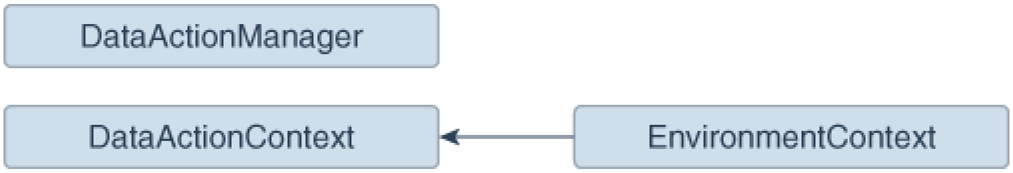
DataActionKOModel, ValuePassingMode

The `DataActionKOModel` class provides the base KOModel shared by the different subclasses of `AbstractDataAction`. See DataActionKOModel Class.

Data Action Service Classes

There are several different data action service classes.

DataActionManager



All communication with `DataActionManager` uses `ClientEvents.DataActionManager` which implements event handlers for:

- Managing the set of data actions defined in the current workbook.
- Invoking a data action.
- Retrieving all the data actions defined in the current workbook.
- Retrieving all the data actions that are applicable to the current marked data points.

DataActionContext, EnvironmentContext

When a data action is invoked, the `DataActionContext` class contains the context that's passed to the target.

- `getColumnValueMap()`
  Returns a map of attribute column values keyed by attribute column names. These define the qualified data reference for the data points that the data action is invoked from.
- `getLogicalFilterTrees()`
  Returns a `LogicalFilterTrees` object describing the qualified data references for the specific data points that the data action is invoked from (see the `InteractionService` for details).
- `getEnvironmentContext()`
  An instance of the `EnvironmentContext` class describing the source environment such as:
  - `getProjectPath()`
  - `getCanvasName()`

  - `getUserID()`
  - `getUserName()`
- `getReport()`
  Returns the report that the data action is invoked from.

DataActionHandler

The `DataActionHandler` class registers the various data action plug-ins. Its API is broadly consistent with the other plug-in handlers (for example, `VisualizationHandler`).

The `DataActionHandler` class provides the following public methods:

- `getClassName(sPluginType:String) : String`
  Returns the fully qualified class name for the specified data action type.
- `getDisplayName(sPluginType:String) : String`
  Returns the translated display name for the specified data action type.
- `getOrder(sPluginType:String) : Number`
  Returns a number used to sort lists of the types of data action into the preferred order.

The `DataActionHandler` class provides the following static methods:

- `getDependencies(oPluginRegistry:Object) : Object.<String, Array>`
  Returns a dependency map covering all the registered data action types.
- `getHandler(oPluginRegistry:Object, sExtensionPointName:String, oConfig:Object) : DataActionPluginHandler`
  Constructs and returns a new instance of the `DataActionHandler` class.

DataActionUpgradeHandler

The `DataActionUpgradeHandler` class is called by the `UpgradeService` when a report is opened.

The `DataActionHandler` class provides two main methods:

- `deferredNeedsUpgrade(sCurrentVersion, sUpgradeTopic, oDataActionJS, oActionContext) : Promise`
  Returns a `Promise` that resolves to a Boolean indicating whether the specified data action must be upgraded (`true`) or not (`false`). The method decides whether the data action must be upgraded by comparing the version number of the data action instance with the version number obtained from the data action's constructor. If the instance's version number is less than the constructor's version number, the data action instance must be upgraded.
- `performUpgrade(sCurrentVersion, sUpgradeTopic, oDataActionJS, oActionContext, oUpgradeContext) : Promise`
  Carries out the upgrade on the specified data action and resolves the `Promise`. The upgrade itself is carried out by calling the `upgrade()` method on the data action (only the specific subclass of data action being upgraded is qualified to upgrade itself).
- `getOrder(sPluginType:String) : Number`
  Returns a number used to sort lists of the types of data action into the preferred order.

Data Action Code Interactions

A data action interacts with Oracle Analytics code when it creates a user interface field, and when a user invokes a data action.

Create the Field for a New Data Action Instance

This interaction starts when Oracle Analytics wants to render a data action user interface field. To do so, it:

1. Creates a `PanelGadgetInfo` that acts as the parent `GadgetInfo` for the `GadgetInfos` that the data action returns.
2. Calls `getGadgetInfos()` on the data action.
3. Adds the data action's `GadgetInfos` as children of the `PanelGadgetInfo` created in the first step.
4. Creates the `PanelGadgetView` that renders the `PanelGadgetInfo`.
5. Sets the `HTMLElement` that's the container of the `PanelGadgetView`.
6. Registers the `PanelGadgetView` as a child `HostedComponent` of a `HostedComponent` that's already attached to the `HostedComponent` tree.
   This renders the data action's gadgets inside the Panel gadget in the order they appear in the array returned by `getGadgetInfos()`.

Invoke a Data Action

This interaction starts when the user invokes a data action through the Oracle Analytics user interface (for example, from the context menu on a data point in a visualization).

In response to the user interaction, the code:

1. Publishes an `INVOKE_DATA_ACTION` event containing the data action's ID, the `DataVisualization` that the data action is invoked from, and a `TransientVizContext` object.
2. The `DataActionManager` handles this event by:
   a. Obtaining the data action instance from its ID.
   b. Obtaining the `LogicalFilterTrees` for the marked data points in the specified `DataVisualization`.
   c. Constructing a `DataActionContext` that contains all the information to pass to the data action's target.
   d. Calling `invoke(oDataActionContext)` on the data action.

Example Data Action plugin.xml File

This topic shows an example plugin.xml file for a `CanvasDataAction` data action.

Example plugin.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:obiplugin xmlns:tns="http://plugin.frameworks.tech.bi.oracle"
```

```
               xmlns:viz="http://plugin.frameworks.tech.bi.oracle/extension-
points/visualization"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               id="obitech-currencyconversion"
               name="Oracle BI Currency Conversion"
               version="0.1.0.@qualifier@"
               optimizable="true"
               optimized="false">


   <tns:resources>
      <tns:resource id="currencyconversion" path="scripts/
currencyconversion.js" type="script" optimizedGroup="base"/>
      <tns:resource-folder id="nls" path="resources/nls" optimizable="true">
         <tns:extensions>
            <tns:extension name="js" resource-type="script"/>
         </tns:extensions>
      </tns:resource-folder>
   </tns:resources>


   <tns:extensions>
      <tns:extension id="oracle.bi.tech.currencyconversiondataaction" point-
id="oracle.bi.tech.plugin.dataaction" version="1.0.0">
         <tns:configuration>
         {
            "resourceBundle": "obitech-currencyconversion/nls/messages",
            "properties":
            {
               "className": "obitech-currencyconversion/
currencyconversion.CurrencyConversionDataAction",
               "displayName": { "key" : "CURRENCY_CONVERSION", "default" :
"Currency Conversion" },
               "order": 100
            }
         }
         </tns:configuration>
      </tns:extension>
   </tns:extensions>


</tns:obiplugin>
```

Data Action Plug-in Files and Folders

The following files and folders are used to implement data action plug-ins.

bitech/client/plugins/src/

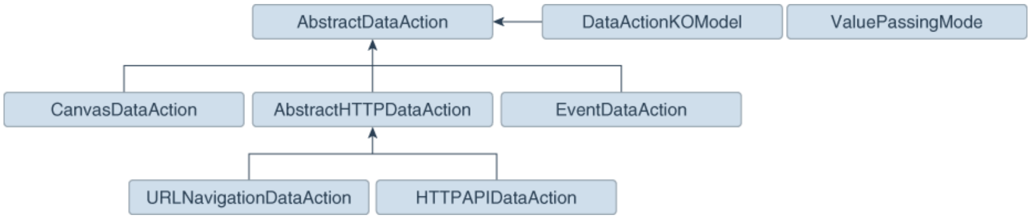
- report
  - obitech-report
    - scripts
      - dataaction

                - dataaction.js
                - dataactiongadgets.js
                - dataactionpanel.js
                - dataactionupgradehandler.js
- obitech-reportservice
    - scripts
        - dataaction
            - dataactionmanager.js
            - dataactionhandler.js

Choose the Best Data Action Class to Extend

Before you start writing your custom data action plug-in, decide which of the existing data action classes you want to extend. Choose the data action class that provides functionality that most closely matches what you want your data action to do.

Each data action inherits from the `AbstractDataAction` class as shown in the class diagram. The class diagram shows the two abstract data action classes (`AbstractDataAction` and `AbstractHTTPDataAction`) and the four concrete data action classes (`CanvasDataAction`, `URLNavigationDataAction`, `HTTPAPIDataAction`, and `EventDataAction`) that you can extend. Each data action that you provide must extend one of these classes. Which class you extend depends on the behavior you want to implement when you invoke your data action. Most third-party data actions are likely to extend either `URLNavigationDataAction`, `HTTPAPIDataAction` or `EventDataAction`.



Regardless of which class you extend, when your data action is invoked, you're provided with metadata describing the full context of the data-point from which the data action is invoked. See Data Action Context.

AbstractDataAction Class

`AbstractDataAction` is the abstract base class from which all types of data action inherit. It's responsible for providing common functionality and default behavior that the subclasses can use.

AbstractDataAction

All types of data action are subclasses of the `AbstractDataAction` base class. It provides the core set of functionality common to all data actions. Unless you're creating a complex data action that carries out multiple types of action when invoked, or you need to do something not supported by the concrete classes, you shouldn't extend this class directly. If you need to create a complex data action then consider extending the concrete class that most closely provides the functionality you require.

AbstractDataAction Syntax

```
+ AbstractDataAction(oKOModel)



+ getKOViewModel():DataActionKOModel



+ createFromJS(fDataActionConstructor, sClassName, oDataActionKOModelUS) :
AbstractDataAction


+ invoke(oActionContext, oDataActionContext)
+ getGadgetInfos(oReport) : AbstractGadgetInfo[]
+ validate() : DataActionError


+ getSettings() : Object
+ requiresActionContextToInvoke() : Boolean
+ isAllowedHere() : Boolean


# createNameGadgetInfo(oReport) : AbstractGadgetInfo
# createAnchorToGadgetInfo(oReport) : AbstractGadgetInfo
# createPassValuesGadgetInfo(oReport) : AbstractGadgetInfo
```

DataActionKOModel Class

Each subclass of `AbstractDataAction` is likely to create its own subclass of `DataActionKOModel`. The `DataActionKOModel` base class provides the following properties:

DataActionKOModel, ValuePassingMode

- `sID:String`
  The unique ID given to the data action instance.
- `sClass:String`
  The class name of this specific type of data action.
- `sName:String`
  The display name given to the data action instance.
- `sVersion`
- `sScopeID`
- `eValuePassingMode:ValuePassingMode`
  The mode used when passing context values. The mode can be one of the `ValuePassingMode` values (`ALL, ANCHOR_DATA, NONE, CUSTOM`).
- `aAnchorToColumns: ColumnKOViewModel[]`

The columns that this data action is anchored to. This is optional. If not supplied, then the data action is available on all columns.

- `aContextColumns : ColumnKOViewModel[]`
  The columns that this data action includes in the context passed to the data action target when the data action is invoked. If not supplied, all marked columns are included in the context.

CanvasDataAction Class

`CanvasDataAction` is a subclass of the `AbstractDataAction` base class. You can extend this concrete class to provide the functionality you require.

CanvasDataAction

Use the `CanvasDataAction` class to navigate from a data point in a visualization to a different canvas. The canvas you're navigating to can be in the same workbook or a different one. All the active filters for the source visualization are passed to the target canvas along with new filters that describe the Qualified Data Reference of the data point itself. If your data action needs to navigate to a different canvas then this is the class your data action should extend.

```
+ CanvasDataAction(oKOModel)



+ create(s)ID_sName) : CanvasDataAction
+ upgrade(oOldDataActionJS) : Object



+ invoke(oActionContext: ActionContext,
oDataActionContext:DataActionContext)
+ getGadgetInfos(oReport) : AbstractGadgetInfo[]
+ validate() : DataActionError


# createProjectGadgetInfo(oReport) : AbstractGadgetInfo
# createCanvasGadgetInfo(oReport) : AbstractGadgetInfo
```

EventDataAction Class

`EventDataAction` is a subclass of the `AbstractDataAction` base class. You can extend this concrete class to provide the functionality you require.

EventDataAction

Use the `EventDataAction` class to publish a client-side event. You can then register one or more subscribers that listen for that event and perform their own actions. Use this type of data action in more complex use cases where you've a large amount of code and can benefit from keeping your data action code loosely coupled to the code that performs the necessary actions when the data action is invoked.

```
+ EventDataAction(oKOModel)



+ create(sID_sName) : EventDataAction
+ upgrade(oOldDataActionJS) : Object


+ invoke(oActionContext: ActionContext, oDataActionContext:DataActionContext)
+ getGadgetInfos(oReport) : AbstractGadgetInfo[]
+ validate() : DataActionError


# createEventGadgetInfo(oReport) : AbstractGadgetInfo
```

AbstractHTTPDataAction Class

`AbstractHTTPDataAction` is the abstract base class that the `URLNavigationDataAction` and `HTTPAPIDataAction` subclasses inherit common functionality and default behavior from.

AbstractHTTPDataAction

The `AbstractHTTPDataAction` abstract base class is shared by both the `URLNavigationDataAction` and `HTTPAPIDataAction` classes. If your data action needs to open a web page in a new browser tab you must extend `URLNavigationDataAction`. If your data action needs to invoke an HTTP API then you should extend `HTTPAPIDataAction`. You may decide it's better to extend `AbstractHTTPDataAction` directly.

```
+ HTTPDataAction(oKOModel)


+ validate() : DataActionError


# createURLGadgetInfo(oReport) : AbstractGadgetInfo
```

URLNavigationDataAction Class

`URLNavigationDataAction` is a subclass or the `AbstractHTTPDataAction` base class.

URLNavigationDataAction

Use the `URLNavigationDataAction` class to open a specific URL in a new browser tab. You compose the URL using tokens that are replaced with values derived from data points that the user selects when they invoke the data action. The data point values are passed as part of the data action context to the external web page. For example, create a data action invoked using a CustomerID column that opens a customer's web page in your Customer Relations Management application such as Oracle Sales Cloud.

```
+ URLNavigationDataAction(oKOModel)


+ create(sID_sName) : URLNavigationDataAction
+ upgrade(oOldDataActionJS) : Object


+ invoke(oActionContext: ActionContext,
oDataActionContext:DataActionContext)
+ getGadgetInfos(oReport) : AbstractGadgetInfo[]
```

HTTPAPIDataAction Class

HTTPAPIDataAction is a subclass or the AbstractHTTPDataAction base class. You can extend this concrete class to provide the functionality you require.

HTTPAPIDataAction

Use the HTTPAPIDataAction class to invoke HTTP APIs by creating an asyncronous XMLHTTPRequest (XHR) and submitting it to the specified URL. The HTTP response code enables a message to be displayed briefly on the canvas. For example, you can customize the request to send JSON or XML payloads to a REST or SOAP server and you can customize the response handler to show a custom user interface.

For the HTTPAPIDataAction data action to work, you must add the URL of the HTTP API you want to access to your list of Safe Domains and grant it **Connect** access. See Register Safe Domains.

```
+ HTTPAPIDataAction(oKOModel)


+ create(sID_sName) : HTTPAPIDataAction
+ upgrade(oOldDataActionJS) : Object


+ invoke(oActionContext: ActionContext,
oDataActionContext:DataActionContext)
+ getGadgetInfos(oReport) : AbstractGadgetInfo[]


# createHTTPMethodGadgetInfo(oReport) : AbstractGadgetInfo
# createPostParamGadgetInfo(oReport) : AbstractGadgetInfo
```

Generate Data Action Plug-ins from a Template

You use a series of commands to generate a development environment and populate it with a HTTP API Data Action along with the necessary folders and files that you need to create a custom data action plug-in.

All plug-in files follow the same basic structure. You can manually create the files and folders or you can generate them from a template. The tools to do this are part of the Oracle Analytics Desktop software development kit (SDK) which is included with Oracle Analytics Desktop. See Oracle Analytics Desktop SDK Reference.

Use Oracle Analytics Desktop version 5.4 or later to access the classes required to create a custom data action plug-in.

Use these commands to generate your development environment and populate it with a HTTP API data action.

1. At a command prompt, specify the root folder of your Oracle Analytics Desktop installation:

   ```
   set DVDESKTOP_SDK_HOME=C:\Program Files\Oracle Analytics Desktop
   ```

2. Specify the location to store your custom plug-ins:

   ```
   set PLUGIN_DEV_DIR=C:\temp\dv-custom-plugins
   ```

3. Add the SDK command line tools to your path using:

   ```
   set PATH=%DVDESKTOP_SDK_HOME%\tools\bin;%PATH%
   ```

4. Create a folder for the directory used to store the custom plug-ins using:

   ```
   mkdir %PLUGIN_DEV_DIR%
   ```

5. Change the directory to the folder for storing custom plug-ins:

   ```
   cd %PLUGIN_DEV_DIR%
   ```

6. Create the environment variables:

   ```
   bicreateenv
   ```

7. Create the template files needed to start developing a custom HTTP API data action, for example:

   ```
   bicreateplugin -pluginxml dataaction -id company.mydataaction -subType
   httpapi
   ```

   Use the -`subType` option to specify the data action type that you want to create from: `httpapi`, `urlNavigation`, `canvasNavigation`, `event`, or `advanced`. The `advanced` option extends from the `AbstractDataAction` base class.

Generated Folders and Files

Your newly generated data action development environment contains these folders and files:

```
%PLUGIN_DEV_DIR%\src\customdataaction
   company-mydataaction\
      extensions\
```

```
4             oracle.bi.tech.plugin.dataaction\
5                company.mydataaction.json
6          nls\
7             root\
8                messages.js
9             messages.js
10          mydataaction.js
11          mydataactionstyles.css
12          plugin.xml
```

- **Line 2**: The `company-mydataaction` folder is the ID that you specify.
- **Line 6**: The `nls` folder contains the files for externalizing strings that enable your plug-in to provide Native Language Support.
- **Line 7**: The strings in the files under the `nls\root` folder are the default strings used when translations for a requested language aren't available.
- **Line 8**: The `messages.js` file contains externalized strings for your plug-in that you can add.
- **Line 9**: The `messages.js` file must contain an entry that you add for each additional language that you want to provide localized strings for. You must add a corresponding folder under the `nls` folder for each locale that you want to add translations for. Each folder must contain the same set of files, with the same file names as those added under the `nls\root` folder.
- **Line 10**: The `mydataaction.js` file is the newly generated JavaScript module template that provides a starting point to develop your custom data action.
- **Line 11**: The `mydataactionstyles.css` file can contain any CSS styles that you want to add, and which your data action's user interface can use.
- **Line 12**: The `plugin.xml` file registers your plug-in and its files with Oracle Analytics.

Extend a Data Action Base Class

Once you've chosen the subclass of data action that you want to extend and have generated the necessary folders and files, you're ready to start writing the code specific to your new data action.

You can find your newly generated data action code under `%PLUGIN_DEV_DIR%\src\dataaction`. See Generated Folders and Files for an explanation of the files and folder structure. The main file you must edit is the JavaScript file. For example, if your custom data action ID is `company.MyDataaction`, then the file you're looking for is `%PLUGIN_DEV_DIR%\src\dataaction\company-mydataaction\mydataaction.js`.

Extending Your Data Action's Knockout Model
If your data action has additional properties that need to be stored, then you must add them as observable properties to the Knockout Model. If your data action is given the ID `company.MyDataaction`, then the Knockout Model is called `mydataaction.MyDataActionKOModel` which is located near the top of `mydataaction.js`. By default, this Knockout Model is configured to extend the Knockout Model used by your data action's superclass so you only need to add additional properties to the model.

For a data action that's extending the `HTTPAPIDataAction` base class, use code similar to the following:

```
1 - mydataaction.MydataactionKOModel = function (sClass, sID, sName,
sVersion, sScopeID, aAnchorToColumns, eValuePassingMode, sURL,
        eHTTPMethod, sPOSTParams)
2 - {
3 - mydataaction.MydataactionKOModel.baseConstructor.call(this, sClass, sID,
sName, sVersion, sScopeID, aAnchorToColumns, eValuePassingMode, sURL,
eHTTPMethod, sPOSTParams);
4 - };
5 - jsx.extend(mydataaction.MydataactionKOModel,
dataaction.HTTPAPIDataActionKOModel);
```

- **Line 1**: This is the constructor for your Knockout Model. It accepts the properties that the model needs to store.
- **Line 3**: This is the superclass's constructor, otherwise known as the `baseConstructor` to which you pass the values for all of the properties that are handled by one of the Knockout Model's superclasses.
- **Line 5**: This sets the superclass for this Knockout Model class.

Use code similar to the following to add a string and an array to set properties that are persisted by the data action.

```
mydataaction.MydataactionKOModel = function (sClass, sID, sName,
sVersion, sScopeID, aAnchorToColumns, eValuePassingMode, sURL, eHTTPMethod,
sPOSTParams)
{
mydataaction.MydataactionKOModel.baseConstructor.call(this, sClass, sID,
sName, sVersion, sScopeID, aAnchorToColumns, eValuePassingMode, sURL,
eHTTPMethod, sPOSTParams);


// Set Defaults
sMyString = sMyString || "My default string value";
aMyArray = aMyArray || [];


// Asserts
jsx.assertString(sMyString, "sMyString");
jsx.assertArray(aMyArray, "aMyArray");


// Add observable properties
this.sMyString = ko.observable(sMyString);
this.aMyArray = ko.observableArray(aMyArray);
};
jsx.extend(mydataaction.MydataactionKOModel,
dataaction.HTTPAPIDataActionKOModel);
```

Choose Which Data Action Inherited Methods to Override

Each data action must implement various methods in order to function properly, so you only need to override those methods that implement behavior that you want to change.

If you're extending one of the concrete data actions classes, for example `HTTPAPIDataAction`, then most of the required methods are already implemented and you only need to override the methods that implement the behavior you want to change.

Generic Methods

This section describes the various methods and what's expected of them.

All types of data action must implement the methods that are described here.

create(sID, sName)

The `create()` static method is called when you're creating a new data action and select a **Data Action Type** from the drop-down menu. This method is responsible for:

- Constructing the Knockout Model class that your data action uses.
  The Knockout Model class must have the ID and name that's passed to the `create()` method along with sensible defaults for all other properties. For example, for a currency conversion data action you might want to set the default currency to convert into Dollars. The Knockout Model is the correct place to provide your default values.
- Constructing an instance of your data action from the Knockout Model.
- Returning the instance of your data action.

invoke(oActionContext, oDataActionContext)

The `invoke()` method is called when the user invokes your data action from the context menu for a data point in a visualization. The method passes the `DataActionContext` argument which contains metadata describing the selected data points, visualization, filters, workbook, and session. See Data Action Service Classes.

validate()

The `validate()` method is called on each data action when the user clicks **OK** in the Data Actions dialog. The `validate()` method returns a `null` to indicate that everything is valid or a `DataActionError` if something is invalid. If there's an error in one of the data actions in the dialog, the error prevents the dialog from closing and an error message is displayed to the user. This method validates the name of the data action using the `this.validateName()` method.

getGadgetInfos(oReport)

The `getGadgetInfos()` method is called to enable the user interface to display data action property fields. The method returns an array of `GadgetInfos` in the order you want them to appear in the user interface. Gadgets are provided for all of the most common types of fields (for example, text, drop-down, password, multi-select, radio button, check box) but you can create custom gadgets if you want more complicated fields (for example, where multiple gadgets are grouped together, or where different gadget fields display depending on which option you select). It's a best practice to create a method that constructs each `GadgetInfo` you want in your array, as it makes it easier for potential subclasses to pick and choose from the GadgetInfos you've provided. If you follow this best practice there are already various methods implemented by the different data action base classes that can return a GadgetInfo for each of the fields that they use in their user interfaces. If you also need one of these GadgetInfos then you call the corresponding create****GadgetInfo() method and push its return value into your array of gadgets.

isAllowedHere(oReport)

The isAllowedHere() method is called when the user right-clicks on a data-point in a visualization and the user interface starts to generate the context menu. If a data action exists that's relevant to the selected data-points, then the method returns true and the data action appears in the context menu. If the method returns false, then the data action doesn't appear in the context menu. Consider accepting the default behavior inherited from the superclass.

upgrade(oOldDataActionJS)

If you're creating your first data action then don't use the upgrade(oOldDataActionJS) method. Only use this method after you've created your first Knockout Model and are making significant changes to properties for a second version of your Knockout Model. For example, if the first version of your data action stores a URL in its Knockout Model, but you decide that the next version will store URL component parts in separate properties (for example, protocol, hostname, port, path, queryString and bookmark).

The second version of your Knockout Model code would request to open a data action that had been saved with the first version of your Knockout Model code which can cause problems. To resolve this issue, the system identifies that your current data action code version is newer than that of the data action being opened and it calls the upgrade() method on your new data action class and passes in the old data action Knockout Model (serialized to a JSON object). You can then use the old JSON object to populate your new Knockout Model and return an upgraded version of the JSON object. This ensures that old data action metadata continues to work as you improve your data action code.

HTTPAPIDataAction Methods

If you're extending the HTTPAPIDataAction class, then it provides the following additional method that you may choose to override:

getAJAXOptions(oDataActionContext)

The getAJAXOptions() method is called by the data action's invoke() method. The getAJAXOptions() method creates the AJAX Options object that describes the HTTP request that you want your data action to make. The getAJAXOptions() method is passed the oDataActionContext object that contains the metadata describing the selected data-points, visualization, filters, workbook, and session. Set the AJAX Options as required by the HTTP API you're trying to integrate with and specify the functions you want to be called when the HTTPRequest is successful or results in an error. See the JQuery website for an explanation of the jQuery.ajax object and its properties.

The following implementation is inherited from the HTTPAPIDataAction class. You need to rewrite the inherited method to specify requirements. For example, forming the HTTP request, and the code that handles the HTTP response. This implementation is useful as it shows the parameters passed to the `getAJAXOptions()` function, the object that it's expected to return, and gives a clear example of how to structure the code inside the method.

```
/**
 * This method returns an object containing the AJAX settings used when the data action is invoked.
 * Subclasses may wish to override this method to provide their own behavior.
 * @param {module:obitech-reportservices/dataactionmanager.DataActionContext} oDataActionContext The context metadata describing where the data action was invoked from.
 * @returns {?object} A JQuery AJAX settings object (see http://api.jquery.com/jQuery.ajax/ for details) - returns null if there is a problem.
 */
dataaction.HTTPAPIDataAction.prototype.getAJAXOptions = function (oDataActionContext)
{
   jsx.assertInstanceOfModule(oDataActionContext, "oDataActionContext", "obitech-reportservices/dataactionmanager", "DataActionContext");

   var oAJAXOptions = null;
   var oKOViewModel = this.getKOViewModel();
   var sURL = oKOViewModel.sURL();
   if (sURL)
   {
      // Parse the URL
      var sResultURL = this._parseURL(sURL, oDataActionContext);
      if (sResultURL)
      {
         // Parse the POST parameters (if required)
         var eHTTPMethod = oKOViewModel.eHTTPMethod()[0];
         var sData = null;
         if (eHTTPMethod === dataaction.HTTPDataActionKOModel.HTTPMethod.POST)
         {
            var sPOSTParams = oKOViewModel.sPOSTParams();
            sData = sPOSTParams.replace(dataaction.AbstractHTTPDataAction.RegularExpressions.LINE_END, "&");
            sData = this._parseURL(sData, oDataActionContext, false);
         }
         oAJAXOptions = {
            type: eHTTPMethod,
            url: sResultURL,
            async: true,
            cache: false,
            success: function (/*oData, sTextStatus, oJQXHR*/)
            {
               oDataActionContext.getReport().displaySuccessMessage(messages.HTTP_API_DATA_ACTION_INVOCATION_SUCCESSFUL.format(oKOViewModel.sName()));
            },
            error: function (oJQXHR/*, sTextStatus, sError*/)
            {
 oDataActionContext.getReport().displayErrorMessage(messages.HTTP_API_DATA_ACTION_INVOCATION_FAILED.format(oKOViewModel.sName(), oJQXHR.statusText, oJQXHR.status));
            }
         };
         if (sData)
         {
            oAJAXOptions.data = sData;
         }
      }
   }
   return oAJAXOptions;
 };
```

Test, Package, and Install Your Data Action

You use Oracle Analytics Desktop to test your data action from its source location before you install it.

1. If Oracle Analytics Desktop is currently running, close it.
2. If you're working behind a proxy, set the proxy settings in `%PLUGIN_DEV_DIR%\gradle.properties`. For information about accessing the web through HTTP proxy, see Gradle User Manual.
3. Run Oracle Analytics Desktop in SDK mode by using the command prompt you started in Choose Which Data Action Inherited Methods to Override and enter the following commands:

   ```
   cd %PLUGIN_DEV_DIR%
   ```

   ```
   .\gradlew run
   ```

   Oracle Analytics Desktop starts in SDK mode. Your data action plug-in appears in the Console | Extensions page.

   Create a workbook and test your data action. If you find any issues, you can debug your code using your browser's built-in developer tools.
4. If you created an HTTP API data action:
   a. Go to the Console and display the Safe Domains page.
   b. Add each domain that you want to access.

      For example, if you need access to the apilayer.com APIs, add `apilayer.net` to the list of safe domains.
   c. Click the **Connect** column checkbox for the selected domain.
   d. Reload the Safe Domains page in your browser for the changes to take effect.
5. If you want to prepare your data action plug-in to distribute to other people or to install in Oracle Analytics:
   - Package all of the files into a single ZIP file containing the `%PLUGIN_DEV_DIR%\src\customdataaction` folder and its contents.

- Name the zip using the same ID you gave to your data action plug-in when you created it.

6. Install your data action plug-in. See Manage Custom Plug-ins.

Use an Upgrade Handler for Knockout Model Changes

For some Knockout Model changes you need to upgrade your data action plug-in using an upgrade handler.

When you're making improvements to your data action plug-in without making changes to the Knockout Model you normally edit your JavaScript or CSS files, create a new ZIP file, and replace the existing data action plug-in with the new ZIP file. However, if you've made changes to your data action's Knockout Model then you might need to change the data action `VERSION` property and provide an upgrade handler.

Decide whether you need to use an upgrade handler:

Upgrade Handler Required

- If you rename a property in your Knockout Model.
- If you combine multiple properties into a single property in your Knockout Model.
- If you split a single property into multiple properties in your Knockout Model.
- If you add a new property to the Knockout Model and the correct default value for it depends on other values in the Knockout Model.

Upgrade Handler Not Required

- If you add a new property to the Knockout Model and can provide a default value that's correct for all existing usages of your data action.
- If you remove a property from the Knockout Model because it's no longer used by your data action code.

Upgrade Data Action Plug-ins

Upgrade your data action plug-ins to improve the data action code or upgrade the metadata to enable existing data actions to work with new data action code.

Use an upgrade handler to upgrade a data action plug-in.

1. Increase the version number of your data action.

   For example, if your data action is called company.MyDataAction, then search `mydataaction.js` for the `mydataaction.MyDataAction.VERSION` property. If it's currently set to `1.0.0` then change it to `1.0.1`.

2. Add a `static upgrade(oOldDataActionJS)` method to your data action's class.

   If the `VERSION` property differs from the `sVersion` value stored in the data action metadata then the Data Action Manager calls the `static upgrade()` method on your data action's class.

3. Implement your `upgrade()` method by calling the `upgrade()` method on the superclass and capture its response.

4. Continue to implement your `upgrade()` method by making further edits to the partially upgraded data action JSON returned by the superclass, until the object matches the correct set of properties required by your latest Knockout Model.
5. To finish call `var oUpgradedDataAction = dataaction.AbstractDataAction.createFromJS(fDataActionClass, sFullyQualifiedDataActionClassName, oUpgradedDataActionJS)`.

   This command constructs a new instance of your data action from the upgraded data action JSON and returns `oUpgradedDataAction.getSettings()`.

Data Action Plug-in File Reference

Each data action plug-in requires a plugin.xml file and each plugin.xml file can contain any number of data actions.

Topics:

- Data Action plugin.xml File Example
- Data Action plugin.xml File Properties Section - tns:obiplugin
- Data Action plugin.xml File Resources Section - tns:resources
- Data Action plugin.xml File Extensions Section - tns:extension

Data Action plugin.xml File Example

The plugin.xml file has three main sections, `tns:obiplugin`, `tns:resources`, and `tns:extension`.

Example plugin.xml

This example shows a typical plugin.xml file for one data action.

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:obiplugin xmlns:tns="http://plugin.frameworks.tech.bi.oracle"
               id="obitech-currencyconversion"
               name="Oracle BI Currency Conversion"
               version="0.1.0.@qualifier@"
               optimizable="true"
               optimized="false">


   <tns:resources>
      <tns:resource id="currencyconversion" path="scripts/
currencyconversion.js" type="script" optimizedGroup="base"/>
      <tns:resource-folder id="nls" path="resources/nls"
optimizable="true">
         <tns:extensions>
            <tns:extension name="js" resource-type="script"/>
         </tns:extensions>
      </tns:resource-folder>
   </tns:resources>


   <tns:extensions>
```

```
     <tns:extension id="oracle.bi.tech.currencyconversiondataaction" point-id="oracle.bi.tech.plugin.dataaction" version="1.0.0">
        <tns:configuration>
        {
           "host": { "module": "obitech-currencyconversion/currencyconversion" },
           "resourceBundle": "obitech-currencyconversion/nls/messages",
           "properties":
           {
              "className": "obitech-currencyconversion/currencyconversion.CurrencyConversionDataAction",
              "displayName": { "key" : "CURRENCY_CONVERSION", "default" : "Currency Conversion" },
              "order": 100
           }
        }
        </tns:configuration>
     </tns:extension>
  </tns:extensions>

</tns:obiplugin>
```

Data Action plugin.xml File Properties Section - tns:obiplugin

The `tns:obiplugin` section defines properties common to all types of plug-ins.

Plug-in Properties

The `tns:obiplugin` section defines properties common to all types of plug-ins.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <tns:obiplugin xmlns:tns="http://plugin.frameworks.tech.bi.oracle"
3                id="obitech-currencyconversion"
4                name="Oracle BI Currency Conversion"
5                version="0.1.0.@qualifier@"
6                optimizable="true"
7                optimized="false">
```

- **Line 1**: The XML declaration.
- **Line 2**: The opening tag for the plug-in's root XMLElement and the declaration for the `tns` namespace that's used throughout plugin.xml files.
- **Line 3**: The plug-in's unique ID.
- **Line 4**: The plug-in's default display name (used when a localized version isn't available).
- **Line 5**: The plug-in's version number.
- **Line 6**: A boolean indicating whether or not the JS/CSS can be optimized (compressed).
- **Line 7**: A boolean indicating whether or not the JS/CSS has been optimized (compressed).

Data Action plugin.xml File Resources Section - tns:resources

The `tns:resources` section registers all of the files that contribute to your plug-in.

Resources

```
<tns:resources>
   <tns:resource id="currencyconversion" path="scripts/
currencyconversion.js" type="script" optimizedGroup="base"/>
   <tns:resource-folder id="nls" path="resources/nls" optimizable="true">
      <tns:extensions>
         <tns:extension name="js" resource-type="script"/>
      </tns:extensions>
   </tns:resource-folder>
</tns:resources>
```

You need to register each JavaScript, CSS, Image, and Translation Resource File here. The section is contained within the `<tns:resources>` element and contains any number of the following elements:

- `<tns:resource>`
  These elements are used to register a single file (for example, a JavaScript or CSS file).
- `<tns:resource-folder>`
  These elements are used to register all the files under a specified folder at the same time. For example, an image folder or the folder containing the resource files for Native Language Support.

More information on how to register each type of file is provided in the following sections.

JavaScript Files

Each JavaScript file in your plug-in must be registered with a line similar to the one shown below.

```
<tns:resource id="currencyconversion" path="scripts/currencyconversion.js"
type="script" optimizedGroup="base"/>
```

Where:

- **id** is the ID given to the file.
  Set the ID to match the JavaScript filename without the .js extension.
- **path** is the relative path to the JavaScript file from the plugin.xml file. JavaScript files should be stored under your plug-in's `scripts` directory.
  Use all lowercase for your JavaScript files with no special characters (for example, underscore, hyphen).
- **type** is the type of file being registered. It must be set to `script` for JavaScript files.
- **optimizedGroup** groups multiple JavaScript files into a single compressed file. Third-party plug-ins must leave this set to `base`.

CSS Files

Each CSS file in your plug-in must be registered with a line similar to the one shown below.

```
<tns:resource id="currencyconversionstyles" path="resources/
currencyconversion.css" type="css"/>
```

Where:

- **id** is the ID given to the file.
  Set the ID to match the CSS filename without the .css extension.
- **path** is the relative path to the CSS file from the plugin.xml file. CSS files should be stored under your plug-in's `resources` directory.
  Use all lowercase for your CSS files with no special characters (for example, underscore, hyphen).
- **type** is the type of file being registered. It should always be set to `css` for CSS files.

Image Folders

If your plug-in has images that you need to refer to from within your JavaScript code, then put them in a `resources/images` directory within your plug-in's directory structure and add a `<tns:resource-folder>` element to your `plugin.xml` as follows:

```
<tns:resource-folder id="images" path="resources/images"
optimizable="false"/>
```

If your images are only referenced by your CSS files, then you don't need to add this `<tns:resource-folder>` element to your `plugin.xml` file. In this case, you must still add them to the `resources/images` directory so that you can then refer to them using a relative path from your CSS file.

Native Language Support Resource Folders

Oracle Analytics implements Native Language Support. This requires developers to externalize the strings they display in their user interface into separate JSON resource files. You can then provide different localized versions of those files in a prescribed directory structure and Oracle Analytics automatically uses the correct file for the user's chosen language. You can provide as many translated versions of the resource files as needed. A Native Language Support resource folder points Oracle Analytics to the root of the prescribed Native Language Support directory structure used by your plug-in. All plug-ins that use Native Language Support resource files must have a `<tns:resource-folder>` entry that looks exactly like the example below.

```
<tns:resource-folder id="nls" path="resources/nls" optimizable="true">
   <tns:extensions>
      <tns:extension name="js" resource-type="script"/>
   </tns:extensions>
</tns:resource-folder>
```

See Generated Folders and Files for details about the contents of the files and the prescribed directory structure that you should follow.

Data Action plugin.xml File Extensions Section - tns:extension

For each data action you want your plug-in to provide, you must register a data action extension using a <tns:extension> element similar to this:

```
<tns:extension id="oracle.bi.tech.currencyconversiondataaction" point-
id="oracle.bi.tech.plugin.dataaction" version="1.0.0">
   <tns:configuration>
   {
      "host": { "module": "obitech-currencyconversion/currencyconversion" },
      "resourceBundle": "obitech-currencyconversion/nls/messages",
      "properties":
      {
         "className": "obitech-currencyconversion/
currencyconversion.CurrencyConversionDataAction",
         "displayName": { "key" : "CURRENCY_CONVERSION", "default" :
"Currency Conversion" },
         "order": 100
      }
   }
   </tns:configuration>
</tns:extension>
```

Where:

- **id** is the unique ID you give to your data action.
- **point-id** is the type of extension you want to register. For data action extensions, this must be set to oracle.bi.tech.plugin.dataaction.
- **version** is the extension API version that your extension definition uses (leave this set to **1.0.0**).

The <tns:configuration> element contains a JSON string that defines:

- **host.module** - This is the fully qualified name of the module containing your data action. This fully qualified module name is formulated as %PluginID%/%ModuleName%, where:
  - %PluginID% must be replaced with the plug-in ID you specified in the id attribute of the <tns:obiplugin> element.
  - %ModuleName% must be replaced with the resource ID you specified in the id attribute of the <tns:resource> element for the JavaScript file containing your data action.
- **resourceBundle** - This is the Native Language Support path to the resource file that contains this data action's localized resources. If your resource files are named messages.js and stored correctly in the prescribed nls directory structure, then set this property to %PluginID%/nls/messages (where %PluginID% must be replaced with the plug-in ID you specified in the id attribute of the <tns:obiplugin> element at the top of the plugin.xml file).
- **properties.className** - This is the fully qualified class name given to the data action you're registering. This fully qualified class name is formulated as %PluginID%/%ModuleName%.%ClassName%, where:
  - %PluginID% must be replaced with the plug-in ID you specified in the id attribute of the <tns:obiplugin> element.

  - `%ModuleName%` must be replaced with the resource ID you specified in the id attribute of the `<tns:resource>` element for the JavaScript file containing your data action.
  - `%ClassName%` must be replaced with the name you gave to the data action class in your JavaScript file.
- **properties.displayName** - This property contains an object and two further properties:
  - **key** is the Native Language Support message key that can be used to lookup the data action's localized display name from within the specified `resourceBundle`.
  - **default** is the default display name to use if for some reason the localized version of the display name can't be found.
- **properties.order** - This property enables you to provide a hint that's used to determine the position that this data action should appear when shown in a list of data actions. Data actions with lower numbers in their order property appear before data actions with higher numbers. When there's a tie, the data actions are displayed in the order they're loaded by the system.

11

Use Oracle Analytics Predictive Models and Oracle Machine Learning Models

Predict data with Oracle Analytics predictive models or machine learning models that reside in Oracle Database or Oracle Autonomous Data Warehouse.

Topics:

- Create and Use Oracle Analytics Predictive Models
- Register and Use Oracle Machine Learning Models
- Apply a Predictive or Registered Oracle Machine Learning Model to a Dataset

Create and Use Oracle Analytics Predictive Models

Oracle Analytics predictive models use several embedded machine learning algorithms to mine your datasets, predict a target value, or identify classes of records. Use the data flow editor to create, train, and apply predictive models to your data.

Topics:

- What Are Oracle Analytics Predictive Models?
- How Do I Choose a Predictive Model Algorithm?
- Create and Train a Predictive Model
- Inspect a Predictive Model
- Add a Predictive Model to a Workbook
- Evaluate Machine Learning Models Using Lift and Gain Charts

What Are Oracle Analytics Predictive Models?

An Oracle Analytics predictive model applies a specific algorithm to a dataset to predict values, predict classes, or to identify groups in the data.

You can also use Oracle machine learning models to predict data. See How Can I Use Oracle Machine Learning Models in Oracle Analytics?.

Oracle Analytics includes algorithms to help you train predictive models for various purposes. Examples of algorithms are classification and regression trees (CART), logistic regression, and k-means.

You use the data flow editor to first train a model on a training dataset. After the predictive model has been trained, you apply it to the datasets that you want to predict.

You can make a trained model available to other users who can apply it against their data to predict values. In some cases, certain users train models, and other users apply the models.

> **Note:**
>
> If you're not sure what to look for in your data, you can start by using Explain, which uses machine learning to identify trends and patterns. Then you can use the data flow editor to create and train predictive models to drill into the trends and patterns that Explain found. See What is Explain?

You use the data flow editor to train a model:

- First, you create a data flow and add the dataset that you want to use to train the model. This training dataset contains the data that you want to predict (for example, a value like sales or age, or a variable like credit risk bucket).
- If needed, you can use the data flow editor to edit the dataset by adding columns, selecting columns, joining, and so on.
- After you've confirmed that the data is what you want to train the model on, you add a training step to the data flow and choose a classification (binary or multi), regression, or cluster algorithm to train a model. Then name the resulting model, save the data flow, and run it to train and create the model.
- Examine the properties in the machine learning objects to determine the quality of the model. If needed, you can iterate the training process until the model reaches the quality you want.

Use the finished model to score unknown, or unlabeled, data to generate a dataset within a data flow or to add a prediction visualization to a workbook.

Example

Suppose you want to create and train a multi-classification model to predict which patients have a high risk of developing heart disease.

1. Supply a training dataset containing attributes on individual patients like age, gender, and if they've ever experienced chest pain, and metrics like blood pressure, fasting blood sugar, cholesterol, and maximum heart rate. The training dataset also contains a column named "Likelihood" that is assigned one of the following values: absent, less likely, likely, highly likely, or present.
2. Choose the CART (Decision Tree) algorithm because it ignores redundant columns that don't add value for prediction, and identifies and uses only the columns that are helpful to predict the target. When you add the algorithm to the data flow, you choose the Likelihood column to train the model. The algorithm uses machine learning to choose the driver columns that it needs to perform and output predictions and related datasets.
3. Inspect the results and fine tune the training model, and then apply the model to a larger dataset to predict which patients have a high probability of having or developing heart disease.

How Do I Choose a Predictive Model Algorithm?

Oracle Analytics provides algorithms for any of your machine learning modeling needs: numeric prediction, multi-classifier, binary classifier, and clustering.

Oracle's machine learning functionality is for advanced data analysts who have an idea of what they're looking for in their data, are familiar with the practice of predictive analytics, and understand the differences between algorithms.

Normally users want to create multiple prediction models, compare them, and choose the one that's most likely to give results that satisfy their criteria and requirements. These criteria can vary. For example, sometimes users choose models that have better overall accuracy, sometimes users choose models that have the least type I (false positive) and type II (false negative) errors, and sometimes users choose models that return results faster and with an acceptable level of accuracy even if the results aren't ideal.

Oracle Analytics contains multiple machine learning algorithms for each kind of prediction or classification. With these algorithms, users can create more than one model, or use different fine-tuned parameters, or use different input training datasets and then choose the best model. The user can choose the best model by comparing and weighing models against their own criteria. To determine the best model, users can apply the model and visualize results of the calculations to determine accuracy, or they can open and explore the related datasets that Oracle Analytics used the model to output. See What Are a Predictive Model's Related Datasets?

Consult this table to learn about the provided algorithms:

Name	Type	Category	Function	Description
CART	Classification Regression	Binary Classifier Multi-Classifier Numerical	-	Uses decision trees to predict both discrete and continuous values. Use with large datasets.
Elastic Net Linear Regression	Regression	Numerical	ElasticNet	Advanced regression model. Provides additional information (regularization), performs variable selection, and performs linear combinations. Penalties of Lasso and Ridge regression methods. Use with a large number of attributes to avoid collinearity (where multiple attributes are perfectly correlated) and overfitting.

Name	Type	Category	Function	Description
Hierarchical	Clustering	Clustering	AgglomerativeClustering	Builds a hierarchy of clustering using either bottom-up (each observation is its own cluster and then merged) or top down (all observations start as one cluster) and distance metrics. Use when the dataset isn't large and the number of clusters isn't known beforehand.
K-Means	Clustering	Clustering	k-means	Iteratively partitions records into k clusters where each observation belongs to the cluster with the nearest mean. Use for clustering metric columns and with a set expectation of number of clusters needed. Works well with large datasets. Result are different with each run.
Linear Regression	Regression	Numerical	Ordinary Least Squares Ridge Lasso	Linear approach for a modeling relationship between target variable and other attributes in the dataset. Use to predict numeric values when the attributes aren't perfectly correlated.
Logistic Regression	Regression	Binary Classifier	LogisticRegressionCV	Use to predict the value of a categorically dependent variable. The dependent variable is a binary variable that contains data coded to 1 or 0.
Naive Bayes	Classification	Binary Classifier Multi-Classifier	GaussianNB	Probabilistic classification based on Bayes' theorem that assumes no dependence between features. Use when there are a high number of input dimensions.

Name	Type	Category	Function	Description
Neural Network	Classification	Binary Classifier Multi-Classifier	MLPClassifier	Iterative classification algorithm that learns by comparing its classification result with the actual value and returns it to the network to modify the algorithm for further iterations. Use for text analysis.
Random Forest	Classification	Binary Classifier Multi-Classifier Numerical	-	An ensemble learning method that constructs multiple decision trees and outputs the value that collectively represents all the decision trees. Use to predict numeric and categorical variables.
SVM	Classification	Binary Classifier Multi-Classifier	LinearSVC, SVC	Classifies records by mapping them in space and constructing hyperplanes that can be used for classification. New records (scoring data) are mapped into the space and are predicted to belong to a category, which is based on the side of the hyperplane where they fall.

Create and Train a Predictive Model

Based on the problem that needs to be solved, an advanced data analyst chooses an appropriate algorithm to train a predictive model and then evaluates the model's results.

Arriving at an accurate model is an iterative process and an advanced data analyst can try different models, compare their results, and fine tune parameters based on trial and error. A data analyst can use the finalized, accurate predictive model to predict trends in other datasets, or add the model to workbooks.

Oracle Analytics provides algorithms for numeric prediction, multi-classification, binary-classification and clustering. For information about how to choose an algorithm, see How Do I Choose a Predictive Model Algorithm?

The algorithms aren't available until you install Oracle machine learning into your local Oracle Analytics Desktop directory. See How do I install Machine Learning for Desktop?

1. On the Home page, click **Create**, and then select **Data Flow**.

2. Select the dataset that you want to use to train the model. Click **Add**.

   Typically you'll select a dataset that was prepared specifically for training the model and contains a sample of the data that you want to predict. The accuracy of a model depends on how representative the training data is.

3. In the data flow editor, click **Add a step (+)**.

   After adding a dataset, you can either use all columns in the dataset to build the model or select only the relevant columns. Choosing the relevant columns requires an understanding of the dataset. Ignore columns that you know won't influence the outcome behavior or that contain redundant information. You can choose only relevant columns by adding the **Select Columns** step. If you're not sure about the relevant columns, then use all columns.

4. Navigate to the bottom of the list and click the train model type that you want to apply to the dataset.

5. Select an algorithm and click **OK**.

6. If you're working with a supervised model like prediction or classification, then click **Target** and select the column that you're trying to predict. For example, if you're creating a model to predict a person's income, then select the Income column.

   If you're working with an unsupervised model like clustering, then no target column is required.

7. Change the default settings for your model to fine tune and improve the accuracy of the predicted outcome. The model you're working with determines these settings.

8. Click the **Save Model** step and provide a name and description. This will be the name of the generated predictive model.

9. Click **Save**, enter a name and description of the data flow, and click **OK** to save the data flow.

10. Click **Run Data Flow** to create the predictive model based on the input dataset and model settings that you provided.

Inspect a Predictive Model

After you create the predictive model and run the data flow, you can review information about the model to determine its accuracy. Use this information to iteratively adjust the model settings to improve its accuracy and predict better results.

Topics:

- View a Predictive Model's Details
- Assess a Predictive Model's Quality
- What Are a Predictive Model's Related Datasets?
- Find a Predictive Model's Related Datasets

View a Predictive Model's Details

A predictive model's detail information helps you understand the model and determine if it's suitable for predicting your data. Model details include its model class, algorithm, input columns, and output columns

1. On the Home page, click **Navigator**, and then click **Machine Learning**.
2. Click the **Models** tab.
3. Click the menu icon for a training model and select **Inspect**.
4. Click the **Details** to view the model's information.

Assess a Predictive Model's Quality

View information that helps you understand the quality of a predictive model. For example, you can review accuracy metrics like model accuracy, precision, recall, F1 value, and false positive rate.

Oracle Analytics provides similar metrics irrespective of the algorithm used to create the model, thereby making comparison between different models easy. During the model creation process, the input dataset is split into two parts to train and test the model based on the Train Partition Percent parameter. The model uses the test portion of the dataset to test the accuracy of the model that is built.

Based on your findings in the **Quality** tab, you may need to adjust the model parameters and retrain it. See Create and Train a Predictive Model.

1. On the Home page, click **Navigator**, and then click **Machine Learning**.
2. Click the **Models** tab.
3. Click the menu icon for a training model and select **Inspect**.
4. Click the **Quality** tab to review the model's quality metrics.

What Are a Predictive Model's Related Datasets?

When you run the data flow to create the Oracle Analytics predictive model's training model, Oracle Analytics creates a set of related datasets. You can open and create workbooks on these datasets to learn about the accuracy of the model.

See Find a Predictive Model's Related Datasets.

Depending on the algorithm you chose for your model, related datasets contain details about the model such as prediction rules, accuracy metrics, confusion matrix, and key drivers for prediction. You can use this information to fine tune the model to get better results, and you can use related datasets to compare models and decide which model is more accurate.

For example, you can open a Drivers dataset to discover which columns have a strong positive or negative influence on the model. By examining those columns, you find that some columns aren't treated as model variables because they aren't realistic inputs or that they're too granular for the forecast. You use the data flow editor to open the model and based on the information you discovered, you remove the irrelevant or too-granular columns, and regenerate the model. You check the Quality and Results tab and verify if the model accuracy is improved. You continue this process until you're satisfied with the model's accuracy and it's ready to score a new dataset.

Different algorithms generate similar related datasets. Individual parameters and column names may change in the dataset depending on the type of algorithm, but the functionality of the dataset stays the same. For example, the column names in a statistics dataset may change from Linear Regression to Logistic Regression, but the statistics dataset contains accuracy metrics of the model.

These are the related datasets:

CARTree

This dataset is a tabular representation of CART (Decision Tree), computed to predict the target column values. It contains columns that represent the conditions and the conditions' criteria in the decision tree, a prediction for each group, and prediction confidence. The Inbuilt Tree Diagram visualization can be used to visualize this decision tree.

The CARTree dataset is outputted when you select these model and algorithm combinations.

Model	Algorithm
Numeric	CART for Numeric Prediction
Binary Classification	CART (Decision Tree)
Multi Classification	CART (Decision Tree)

Classification Report

This dataset is a tabular representation of the accuracy metrics for each distinct value of the target column. For example, if the target column can have the two distinct values Yes and No, this dataset shows accuracy metrics like F1, Precision, Recall, and Support (the number of rows in the training dataset with this value) for every distinct value of the target column.

The Classification dataset is outputted when you select these model and algorithm combinations.

Model	Algorithms
Binary Classification	Naive Bayes Neural Network Support Vector Machine
Multi Classification	Naive Bayes Neural Network Support Vector Machine

Confusion Matrix

This dataset, which is also called an error matrix, is a pivot table layout. Each row represents an instance of a predicted class, and each column represents an instance in an actual class. This table reports the number of false positives, false negatives, true positives, and true negatives, which are used to compute precision, recall, and F1 accuracy metrics.

The Confusion Matrix dataset is outputted when you select these model and algorithm combinations.

Model	Algorithms
Binary Classification	Logistics Regression CART (Decision Tree) Naive Bayes Neural Network Random Forest Support Vector Machine
Multi Classification	CART (Decision Tree) Naive Bayes Neural Network Random Forest Support Vector Machine

Drivers

This dataset provides information about the columns that determine the target column values. Linear regressions are used to identify these columns. Each column is assigned coefficient and correlation values. The coefficient value describes the column's weight-age used to determine the target column's value. The correlation value indicates the relationship direction between the target column and dependent column. For example, if the target column's value increases or decreases based on the dependent column.

The Drivers dataset is outputted when you select these model and algorithm combinations.

Model	Algorithms
Numeric	Linear Regression Elastic Net Linear Regression
Binary Classification	Logistics Regression Support Vector Machine
Multi Classification	Support Vector Machine

Hitmap

This dataset contains information about the decision tree's leaf nodes. Each row in the table represents a leaf node and contains information describing what that leaf node represents, such as segment size, confidence, and expected number of rows. For example, expected number of correct predictions = Segment Size * Confidence.

The Hitmap dataset is outputted when you select these model and algorithm combinations.

Model	Algorithm
Numeric	CART for Numeric Prediction

Residuals

This dataset provides information on the quality of the residual predictions. A residual is the difference between the measured value and the predicted value of a regression model. This dataset contains an aggregated sum value of absolute difference between the actual and predicted values for all columns in the dataset.

The Residuals dataset is outputted when you select these model and algorithm combinations.

Model	Algorithms
Numerics	Linear Regression Elastic Net Linear Regression CART for Numeric Prediction
Binary Classification	CART (Decision Tree)
Multi Classificatin	CART (Decision Tree)

Statistics

This dataset's metrics depend upon the algorithm used to generate it. Note this list of metrics based on algorithm:

- Linear Regression, CART for Numeric Prediction, Elastic Net Linear Regression - These algorithms contain R-Square, R-Square Adjusted, Mean Absolute Error(MAE), Mean Squared Error(MSE), Relative Absolute Error(RAE), Related Squared Error(RSE), Root Mean Squared Error(RMSE).
- CART(Classification And Regression Trees), Naive Bayes Classification, Neural Network, Support Vector Machine(SVM), Random Forest, Logistic Regression - These algorithms contain Accuracy, Total F1.

This dataset is outputted when you select these model and algorithm combinations.

Model	Algorithm
Numeric	Linear Regression Elastic Net Linear Regression CART for Numeric Prediction
Binary Classification	Logistics Regression CART (Decision Tree) Naive Bayes Neural Network Random Forest Support Vector Machine
Multi Classification	Naive Bayes Neural Network Random Forest Support Vector Machine

Summary

This dataset contains information such as Target name and Model name.

The Summary dataset is outputted when you select these model and algorithm combinations.

Model	Algorithms
Binary Classification	Naive Bayes Neural Network Support Vector Machine
Multi Classification	Naive Bayes Neural Network Support Vector Machine

Find a Predictive Model's Related Datasets

Related datasets are generated when you train a predictive model.

Depending on the algorithm, related datasets contain details about the model like: prediction rules, accuracy metrics, confusion matrix, key drivers for prediction, and so on. These parameters help you understand the rules the model used to determine the predictions and classifications.

See What Are a Predictive Model's Related Datasets?

1. On the Home page, click **Navigator**, and then click **Machine Learning**.
2. Click the **Models** tab.
3. Click the menu icon for a training model and select **Inspect**.
4. Click the **Related** tab to access the model's related datasets.
5. Double-click a related dataset to view it or to use it in a workbook.

Add a Predictive Model to a Workbook

When you create a scenario in a workbook, you apply a predictive model to the workbook's dataset to reveal the trends and patterns that the model was designed to find.

> **Note:**
>
> You can't apply an Oracle machine learning model to a workbook's data.

After you add the model to the workbook and map the model's inputs to the dataset's columns, the Data Panel contains the model's objects, which you can drag and drop onto the canvas. Machine learning generates the model's values based on the visualization's corresponding data columns.

1. On the Home page, click **Create**, and then click **Workbook**.
2. Select the dataset that you want to use to create the workbook and click **Add to Workbook**.
3. In the Data pane, click **Add**, and select **Create Scenario**.
4. In the Create Scenario - Select Model dialog, select a model and click **OK**.

   You can only apply a predictive model. You can't apply an Oracle machine learning model.

If each model input can't be matched to a data element, then the Map Your Data to the Model dialog is displayed.

5. If the Map Your Data to the Model dialog is displayed, then in the **Dataset** field, select the dataset to use with the model.
6. Match the model input and data elements as needed. Click **Done**.

   The scenario is displayed as a dataset in the Data Elements pane.
7. Drag and drop elements from the dataset and the data model onto the Visualize canvas.
8. To adjust the scenario, right-click the scenario in the Data Elements pane and select **Edit Scenario**.
9. Change the dataset and update the model input and data elements mapping as needed.
10. Click **Save** to save the workbook.

Evaluate Machine Learning Models Using Lift and Gain Charts

Lift and gain charts enable you to compare different classification machine learning models to determine the most accurate model.

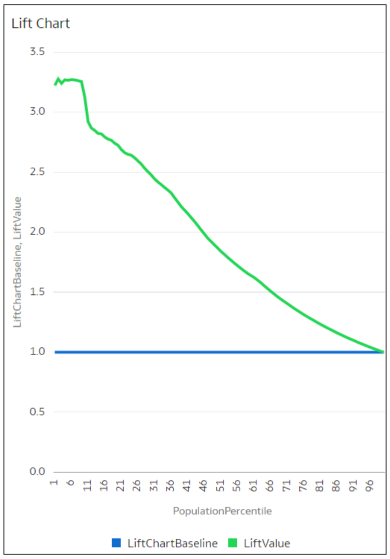
Topics:

- Overview to Using Lift and Gain Charts
- Generate Predictive Data for Lift and Gain Charts
- Evaluate a Machine Learning Model Using a Lift and Gain Chart

Overview to Using Lift and Gain Charts

Lift and gain charts enable you to evaluate predictive machine learning models by charting modeling statistics in a visualization in Oracle Analytics.

When you use a data flow to apply a classification model to a dataset, Oracle Analytics enables you to compute lift and gain values. You can then visualize this data in a chart to help you assess the accuracy of predictive models and determine the best one to use.



Prequisites

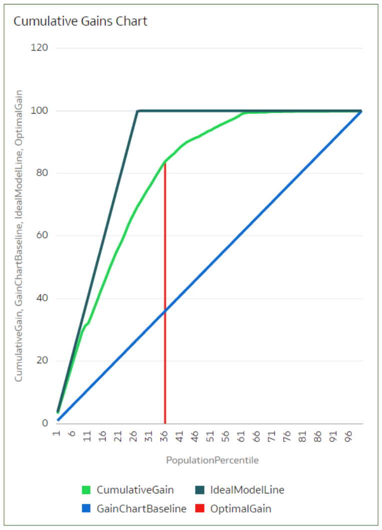
- Oracle Database or Oracle Autonomous Data Warehouse
- A classification predictive model, for example, based on Binary Classifier or Multi-classifier.
  You access pre-created predictive models in the Machine Learning area in Oracle Analytics.

Statistics Generated For List and Gain Analysis

When you apply a predictive classification model to a dataset and generate lift and gain statistics, you produce a dataset named <*Data flow name*>_LIFT with these columns:

- PopulationPercentile - The dataset population split into 100 equal groups.
- CumulativeGain - The ratio of the cumulative number of positive targets up to that percentile, to the total number of positive targets. The closer the cumulative gains line is to the top-left corner of the chart, the greater the gain; the higher the proportion of the responders that are reached for the lower proportion of customers contacted.
- GainChartBaseline - The overall response rate : the line represents the percentage of positive records we expect to get if we selected records randomly. For example, in a marketing campaign, if we contact X% of the customers randomly, we will receive X% of the total positive response.
- LiftChartBaseline - Value of 1 and used as a baseline for lift comparison.
- LiftValue - The cumulative lift for a percentile. Lift is the ratio of the cumulative positive records density for the selected data, to the positive density over all the test data.
- IdealModelLine - The ratio of the cumulative number of positive targets to the total number of positive targets.
- OptimalGain - This indicates the optimum number of customers to contact. The cumulative gain curve will flatten beyond this point.

You can then visualize the <*Data flow name*>_LIFT dataset in an Oracle Analytics chart. For example, to analyze gains, you might plot PopulationPercentile on the x-axis, and CumulativeGain, GainChartBaseline, IdealModelLine, and OptimalGain on the y-axis.



Generate Predictive Data for Lift and Gain Charts

When you use a data flow to apply a classification model to a dataset, Oracle Analytics enables you to compute statistics that you can visualize in lift and gain charts. Oracle Analytics generates the statistics in a dataset named <*`Data flow name`*>_LIFT.

1. On the Home page, click **New**, then click **Data Flow**.
2. Select a data source, then click **Add**.
3. Click **Add a Step**, and select **Apply Model**.
4. At Select Model, select a predictive model, then click **OK**.
5. In Apply Model, in the **Parameters** section:
   - In **Compute lift and gain**, select **Yes**.
   - In **Target column to compute lift**, select the column name of the value being predicted. For example, if your model predicts whether customers will sign up for a membership using a column named SIGNUP, select SIGNUP.
   - In **Positive class to compute**, specify the case-sensitive data value representing the positive class (or the preferred outcome) in the prediction. For example, if your model predicts whether customers will sign-up for a membership using a column named SIGNUP with values YES or NO, specify YES.
6. Add a **Save Data** node to your data flow.
7. Execute this data flow.

The data flow produces a dataset named <*`Data flow name`*>_LIFT that contains lift and gain statistics, which you can evaluate. See Evaluate a Machine Learning Model Using a Lift and Gain Chart.

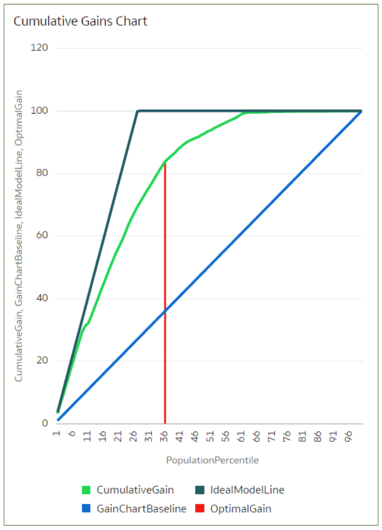
Evaluate a Machine Learning Model Using a Lift and Gain Chart

Use a chart to analyze statistics generated by machine learning classification models to determine the best model to use.

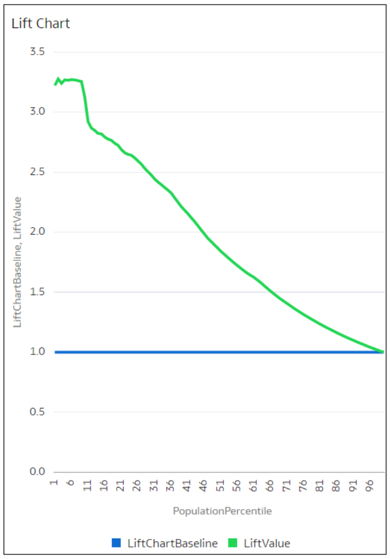
Before you start, apply a predictive model to your data and generate lift and gain statistics. See Generate Predictive Data for Lift and Gain Charts.

1. On the Home page, click **Create**, then click **Workbook**.
2. In **Add Dataset**, select the <*Data flow name*>_LIFT dataset that you generated in the previous task, then click **Add to Workbook**.
3. In the **Visualize** panel, select the statistics to analyze, then right-click and select **Pick Visualization**, and choose **Line Chart**.

   For example, to analyze gains, you might place **PopulationPercentile** on the x-axis, and place **CumulativeGain**, **GainChartBaseline, IdealModelLine**, and **OptimalGain** on the y-axis.

   To analyze lift, you might place **PopulationPercentile** on the x-axis, and place **LiftChartBaseline**, and **LiftValue** on the y-axis.



Register and Use Oracle Machine Learning Models

You can register and use Oracle machine learning models from Oracle Database or Oracle Autonomous Data Warehouse to score data in Oracle Analytics. Use the data flow editor to apply the machine learning models to your data.

Topics:

- How Can I Use Oracle Machine Learning Models in Oracle Analytics?
- Register Oracle Machine Learning Models in Oracle Analytics
- Inspect Registered Oracle Machine Learning Models
- Visualize a Registered Oracle Machine Learning Model's View

How Can I Use Oracle Machine Learning Models in Oracle Analytics?

Oracle Analytics allows you to register and use Oracle machine leaning models from Oracle Database or Oracle Autonomous Data Warehouse.

Using Oracle machine learning models with Oracle Analytics greatly increases the level of predictive analytics that you can perform on datasets because the data and the model reside in the database, the data scoring is performed in the database, and the resulting dataset is stored in the database. This allows you to use the Oracle machine learning execution engine to score large datasets.

You can register and use Oracle machine learning models from these database data sources:

- Oracle Autonomous Data Warehouse
- Oracle Database

In Oracle Analytics you can register any of the database's Oracle machine learning models in the mining classes Classification, Regression, Clustering, Anomaly, or Feature Extraction that were created using the Oracle Machine Learning for SQL API (OML 4SQL). Your database permissions determine the Oracle machine learning models that are available for you to register and use.

You can also create predictive models in Oracle Analytics. See What Are Oracle Analytics Predictive Models?

Register Oracle Machine Learning Models in Oracle Analytics

The Oracle machine learning models must be registered in Oracle Analytics before you can use them to predict data. You can register and use models that reside in your Oracle Database or Oracle Autonomous Data Warehouse data sources.

1. On the Home page, click **Page Menu**, and then click **Register ML Model**.
2. In the Register an ML Model dialog, select a connection.

   In the Select a Model to Register dialog, you'll see the database's Oracle machine learning models in the mining classes Classification, Regression, Clustering, Anomaly, or Feature Extraction that were created using the Oracle Machine Learning for SQL API (OML 4SQL).
   If needed, click **Create Connection** to create a connection to the Oracle Database or Oracle Autonomous Data Warehouse data source containing the Oracle machine learning model that you want to use.
3. In the Select a Model to Register dialog, click the model that you want to register and review the model's information. For example, the model class and algorithm used to build the model, the target the model predicts, the columns the model is trained on, model predictions, and parameters.
4. Click **Register**.
5. From the Home page, click **Navigator**, and then click **Machine Learning** to confirm that the model was imported successfully.

Inspect Registered Oracle Machine Learning Models

You can access and review information about the Oracle machine learning models that you registered in Oracle Analytics.

Topics:

- View a Registered Model's Details
- What Are a Registered Model's Views?
- View a Registered Model's Views List

View a Registered Model's Details

View an Oracle machine learning model's detail information to help you understand the model and determine if it's suitable for predicting your data. Model details include model class, algorithm, input columns, output columns, and parameters.

When you register a model, its detail information is included. This information is gotten from Oracle Database or Oracle Autonomous Data Warehouse.

1. On the Home page, click **Navigator**, and then click **Machine Learning**.
2. Click the **Models** tab.
3. Select a model, click its **Actions** menu, and then select **Inspect**.
4. Click **Details** to view the model's information.

What Are a Registered Model's Views?

When an Oracle machine learning model is created, views containing specific information about the model are generated and stored in the database. Use Oracle Analytics to access a list of a model's views, and then build datasets that you can use to visualize the information contained in the views.

Views contain information about the registered model such as model statistics, target value distribution, and algorithm settings. The number and kind of views created are determined by the model's algorithm. So a model built from the Naive Bayes algorithm has one set of views and a model built from the Decision Tree algorithm has a different set of views. For example, some of the views generated for a Decision Tree model are:

- Scoring Cost Matrix - Describes the scoring matrix for classification models. The view contains `actual_target_value`, `predicted_target_value`, and `cost`.
- Global Name-Value Pairs - Describes global statistics related to the model like number of rows used in the model build and convergence status.
- Decision Tree Statistics - Describes the statistics associated with individual nodes in the decision tree. The statistics include a target histogram for the data in the node. For every node in the tree, this view has information about `predicted_target_value`, `actual_target_value`, and `node` support.

Each view's name is unique, for example DM$VCDT_TEST. The format used to generate view names is DM$V*Alphabet_Model Name* where:

- DM$V - Represents a prefix for views generated from a registered model.
- *Alphabet* - Represents a value that indicates the type of output model. For example, C indicates that the view type is Scoring Cost Matrix, and G indicated that the view type is Global Name-Value Pair.
- *Model Name* - Holds the name of the registered Oracle machine learning model and its view. For example, DT_TEST.

For more information about views, see the documentation for your Oracle database version.

Oracle Analytics provides a list of any registered model's views. However, you can only access and visualize views for Oracle Database 12*c* Release 2 or later. If you're working with an early version of Oracle Database, then you can't use Oracle Analytics to access and visualize views.

View a Registered Model's Views List

A registered model's views are stored in the database, but you can use Oracle Analytics to display a list of the model's views.

Views contain information such as a model's size, settings, and the attributes used in the model. This information can help you better understand and utilize the model.

> **Note:**
>
> You can access and visualize views for Oracle Database 12*c* Release 2 or later. If you're working with an earlier version of Oracle Database, then these views don't exist in the database and you can't use Oracle Analytics to access and visualize them.

See What Are a Registered Model's Views?

1. On the Home page, click **Navigator**, and then click **Machine Learning**.
2. Click the **Models** tab.
3. Locate and click a registered model's **Actions menu**, and then click **Inspect**.
4. Click the **Related** tab to view a list of the model's views.

Visualize a Registered Oracle Machine Learning Model's View

Visualize any of a registered model's views to discover information that helps you better understand and utilize the model.

> **Note:**
>
> You can access and visualize views for Oracle Database 12*c* Release 2 or later. If you're working with an earlier version of Oracle Database, then these views don't exist in the database and you can't use Oracle Analytics to access and visualize them.

When creating the dataset, you'll need to know the model's view name and the database schema name. Use the following task to find these names, create the dataset, and visualize the view's information.

1. On the Home Page, click **Navigator**, and then click **Machine Learning**.
2. Locate the registered machine learning model and click its **Actions menu**. Click **Inspect**.
3. Click **Details** and confirm that the **Model Info** section is expanded. Go to the **DB Model Owner** field and record the database schema name.
4. Click **Related** and locate and record the name of the view. Click **Close**.
5. On the Home page, click **Create** and click **Dataset**.
6. Select the connection that contains the machine learning model and its views.
7. In the Dataset editor, browse for and click the database schema name that you located on the **Details** tab.
8. Select the view that you located on the **Related** tab, and double-click columns to add them to the dataset. Click **Add**.
9. Click **Create Workbook** to build visualizations.

Apply a Predictive or Registered Oracle Machine Learning Model to a Dataset

Use the data flow editor to score a predictive model on any dataset, or score a registered Oracle machine learning model on a dataset in its corresponding database.

Running the model outputs a new dataset with columns containing predicted values that you can use for analysis and visualization.

When you run a predictive model, the data is moved into and processed by Oracle Analytics. When you run a registered Oracle machine learning model, data isn't moved from the database into Oracle Analytics. Instead the model resides, is processed, and the output dataset is stored in the database.

Use this information to understand the data flow editor and **Apply Model** step options:

- The registered models are displayed and available for review and analyses. Unregistered models aren't displayed.
- The available output columns are specific to the model type. For example, for numeric prediction, output columns include PredictedValue and PredictedConfidence, and for clustering, output columns include the clusterId.
- The available parameters are specific to the model type. For example, if you use a clustering model for scoring, maximum null values is a parameter that you can provide for the scoring process. This parameter is used in the missing value imputation.
- The model and the mapped input data types must match when you're working with an Oracle machine learning model. See View a Registered Model's Details.

1. On the Home page, click **Create**, and then click **Data Flow**.
2. Select the dataset that you want to apply the model to. Click **Add**.
3. In the Data Flow editor, click **Add a step (+)**.
4. From the Data Flow Steps pane, double-click **Apply Model**, and then select the model to use.
5. In Apply Model, go to the Inputs section, and then select a column as the input.
6. In Apply Model, go to the Outputs section, and then select the columns that you want created with the dataset, and update the **Column Name** fields as needed.
7. In the data flow editor, click **Add a step (+)** and select **Save Data**.
8. Enter a name. In the **Save data to** field, specify the location for saving the output data.

   If you're working with an Oracle machine learning model, then the dataset's connection information defaults to the input dataset's connection.
9. Set data preferences as needed in the **Treat As** and **Default Aggregation** fields.

   When you save data, the apply model appends the model's output columns that you selected to the dataset.
10. Click **Save**, enter a name and description for the data flow, and click **OK** to save the data flow.
11. Click **Run Data Flow** to create the dataset.

12

Import, Export, and Share

This topic describes how to import, export, and share your workbooks, visualizations, and stories with other users.

Topics:

Import a Workbook File

You can import a workbook file (.dva file) that was exported from Analytics Cloud or Oracle Analytics Desktop. You can also import workbooks that were exported from Oracle Cloud Applications.

Depending upon how the file was exported, you'll be prompted to enter a password to decrypt the workbook's data or to access the workbook's data source.

The import includes everything that you need to use the workbook such as associated datasets, connection string, connection credentials, and stored data.

1. On the Home page, click the **Page Menu** icon and then select **Import Workbook/Flow**.
2. In the Import Workbook/Flow dialog, click **Select File** or drag a workbook file onto the dialog, and then click **Import**.

   Following a successful import, check that your imported connections are valid by opening and saving them. See Edit a Data Source Connection.

Export a Workbook or Folder as a File

You can export a workbook or folder as an archive file (.DVA) that another user can import.

The .DVA file includes items that you specify such as the associated datasets, connection strings, connection credentials, and stored data.

1. On the Home page, click **Navigator**, and then click **Catalog**.
2. In the Catalog page, highlight the workbook or folder that you want to share and click **Actions menu**, then select **Export** to open the Export dialog.
3. Click **File**.
4. For **Name**, keep the default name or enter a new name for the export file (.DVA file).

5. Enable the **Include Data** option to include the data when sharing a workbook or folder.
6. Enable the **Include Connection Credentials** option so that users can open the workbook without having to sign in. Use the following guidelines to set this field:
   - **Excel, CSV, or TXT data source -** These data sources don't use a data connection, therefore you can clear the **Include Connection Credentials** option.
   - **Database data source -** If you enable the **Include Connection Credentials** option, the user must provide a valid user name and password to load data into the imported workbook.
   - **Oracle Applications or Oracle Essbase data source** - Selecting the **Include Connection Credentials** option works if on the connection setup's Create Connection dialog you specified the **Always use these credentials** option in the **Authentication** field.

     If you clear the **Include Connection Credentials** option or specify the **Require users to enter their own credentials** option in the **Authentication** field, then the user must provide a valid user name and password to load data into the imported workbook.
7. If you enable **Include Data** or **Include Connection Credentials**, enter and confirm a password that the user must provide to import the workbook or folder and decrypt its connection credentials and data.
8. Click **Save** to export to a .DVA file.
9. Click **OK** to specify that you want to save the .DVA file rather open it in your browser.
10. Browse to a folder where you want to save the .DVA file.
11. Click **Save** to save the .DVA file to the folder that you specified.

Export a Visualization or Story

You can export a visualization or story in a variety of formats: DVA (a visualization workbook file), Powerpoint (PPTX), Adobe Acrobat (PDF), PNG, or CSV (data only).

1. On the Home page, select a workbook containing the visualization or story that you want to export, click the **Actions menu**, and then select **Open**.
2. Go to the Visualize or Present canvas and locate the visualization or story that you want to export.
3. In the visualization's toolbar, click **Menu**, and **Export**, and then click **File**.
4. In the **Format** field, select the output format you want and specify the output options:
   - For **Powerpoint (pptx)**, **Acrobat (pdf)**, and **Image (png)** - Specify the file name, paper size, and orientation.
     When you share any of these visual formats, the visualization or pages are re-rendered based on the size and orientation you select. So if you're sharing a table, then your output file might not contain all of the table's rows and columns displayed in your visualization, canvas, or story.
   - For **Data (csv)** - Specify the output file name. This option only includes the data used in the workbook. The outputted file uses the data delimiter for your

computer's locale. For example, if your locale is set to Brazil, then the delimiter for numeric decimals is a comma instead of a period, which is used when your locale is set to United States.

- For **Package (dva)** - Specify whether to include workbook data and connection credentials. To enable users to open the workbook DVA file without having to enter a password, click **Include Connection Credentials** and specify the password.

5. In the **Include** field, select **Active Visual**.
6. Click **Save**.

Export a Visualization's Data to a CSV File

You can export the data from a visualization to a CSV file. This allows you to open the file and work with the data in an application such as Excel.

1. Open the workbook that contains the visualization that you want to export data for.
2. In the Visualize or Present canvas, click the visualization that you want to export data for to select it.
3. Go to the workbook toolbar, and then click **Export**.
4. In the File dialog, go to the **Format** field and select Data (csv).
5. Click **Save**.

   The exported file uses the data delimiter for your computer's locale. For example, if your locale is set to Brazil, then the delimiter for numeric decimals is a comma.

Email Workbooks and Folders

You can email the .DVA file of a workbook or folder to other users.

Selecting the option to email a workbook or folder initiates an export process that produces a .DVA file. The .DVA file includes everything needed to use the workbook or folder (such as associated datasets, the connection string and credentials, and stored data)

1. On the Home page, click **Navigator**, and then click **Catalog**.
2. On the Catalog page select the workbook or folder that you want to share and click **Actions menu**, then select **Export** to open the Export dialog.
3. Click **Email** to open the Email dialog.
4. Enable the **Include Data** option if you’re sharing a workbook or folder that uses an Excel data source and you want to include the data with the export.
5. Enable the **Connection Credentials** option if retrieving the data requires connection credentials. Then enter and confirm the password.

   If your workbook or folder includes data from Oracle Applications or a database and you've selected the **Include Data** option, then enter a password to send to the database for authentication when the user opens the application to access the data. Disable the **Include Data** option if you want users to enter the password when they open the application to access the data.
6. Click **Email**.

   Your email client opens a new partially composed email with the .DVA file attached.

Email a File of a Visualization, Canvas, or Story

You can email visualizations, canvases, or stories in formats like Powerpoint (PPTX), Acrobat (PDF), Image (PNG), CSV (data only), or Package (the whole workbook including connection credentials).

1. On the Home page, select a workbook containing the visualization, canvas, or story to export, click the **Actions menu**, and then select **Open**.
2. On the Visualize or Present canvas, click the **Export** icon on the workbook toolbar, then click **Email**.
3. Use the **Format** option to select the output format you want, and specify output options:
   - For **Powerpoint (pptx)**, **Acrobat (pdf)**, and **Image (png)** - Specify the file name, paper size, and orientation.
     When you email any of these visual formats, the visualization or pages are re-rendered based on the size and orientation you select. So if you're emailing a table, then your output file might not contain all of the table's rows and columns included in your visualization, canvas, or story.
   - For **Data (csv)** - Specify the output file name. This option only includes the data used in the workbook. The outputted file uses the data delimiter for your computer's locale. For example, if your locale is set to Brazil, then the delimiter for numeric decimals is a comma instead of a period, which is used when your locale is set to United States.
   - For **Package (dva)** - Specify whether to include workbook data, connection credentials,). To enable users to open the workbook DVA file without having to enter a password, click **Include Connection Credentials** and specify the password.
4. Click **Email**.

   Your email client opens a new partially composed email with the .DVA file attached.

Print a Visualization, Canvas, or Story

You can print your workbook's visualizations, canvases, or stories.

When you print, the visualization or pages are re-rendered based on the size and orientation you selected. So if you're printing a table, then your printed copy might not contain all of the table's rows and columns included in your visualization, canvas, or story.

1. On the Home page, select a workbook containing the visualization, canvas, or story to print, click the **Actions menu**, and then select **Open**.
2. On the Visualize or Present canvas, click **Menu**, and **Export**, and then click **Print**.
3. Specify a name, and select an option from the include list.
   - **Name** - You can update the name if needed.
   - **Include** - You can select to print the Active Visual, Active Canvas, or All Canvases. You can also click **Include Filters** (if there are filters), and **Include Title** to include them in the printed output.

- **Size** - You can use the **Custom** setting (the default size) to print using the height and width displayed on your screen, and optionally select **Scale Proportionally**, or you can select a different size option (for example, US Letter, A4).
- **Orientation** - You can select whether to print in Landscape or Portrait format.

4. Optional: If filters are in the visualization or workbook, and you want to include them, click **Include Filters**.
5. Optional: If you want to include the title, click **Include Title**.
6. Optional: From the **Size** list, select the paper to use. When using **Custom**, specify the width and height in inches (in), pixels (px), or millimeters (mm).
7. Optional: Select the print format orientation.
8. Click **Print**.

A

Frequently Asked Questions

This reference provides answers to frequently asked questions for Oracle Analytics Desktop.

Topics:

- Oracle Analytics Desktop Installation FAQs
- Oracle Analytics Desktop Workbook and Data Source FAQs
- Oracle Analytics Desktop Printing and Exporting FAQs

Oracle Analytics Desktop Installation FAQs

This topic answers common installation questions.

How do I install Machine Learning and Advanced Analytics?

Machine learning and advanced analytics are optional components that aren't included in the Oracle Analytics Desktop installation. You must install machine learning to use Diagnostics Analytics (Explain), Machine Learning Studio, or advanced analytics.

Follow these steps to install Machine Learning Framework on Windows.

1. Go to the Windows Start menu, browse to Oracle, and click **Install DVML**.
2. The installer starts on completion of the download. Follow the displayed instructions to install machine learning to the selected install path.
3. Click **Finish** to close the installer.
4. When prompted, press any key to close the terminal window.
5. If Oracle Analytics Desktop was running during the installation, then restart it.

Follow these steps to install Machine Learning Framework on Apple Mac.

1. Double-click the application **Oracle Analytics Configure Python** in Finder under Applications or in Launchpad.
2. The installer starts on completion of the download. Follow the displayed instructions to install machine learning to the selected install path. Enter an administrator user name and password to run the installation.
3. Click **Close** after the installation is completed.
   The Machine Learning Framework was installed in `/Library/Frameworks/DVMLruntime.framework`
4. If Oracle Analytics Desktop was running during the installation, then restart it.

Why can’t I install Oracle Analytics Desktop?

To perform the installation, you must have administrator privileges. If you try to install without administrator privileges, the following error message is displayed: `Error in creating registry key. Permission denied.`

To check to see if you've the required administrator privileges, go to Windows Control Panel and check your user accounts. If you don't have administrator privileges, then see your administrator to help you set up the required privileges.

Why can't I successfully upgrade?

If you have issues upgrading, then delete the previous version and try to the installation again.

How will I know when to upgrade?

You'll see a message when a newer version is available. The message will guide you to Oracle Technology Network where you can download the latest installer. See Oracle Analytics Desktop Installation Download.

Oracle Analytics Desktop Workbook and Data Source FAQs

This topic answers common questions about workbooks and data sources.

What data sources are supported?

You can use data only from specific types and versions of sources. See Supported Data Sources.

What if I'm using an unsupported version of Teradata?

If you're using an unsupported version of Teradata, then you must update the extdriver.paths configuration file before you can successfully build a connection to Teradata. This configuration file is located here: `C:\<your directory>\AppData\Local\OracleAnalyticsDesktop\extdrvier.paths`. For example, `C:\Users\jsmith\AppData\Local\OracleAnalyticsDesktop\extdriver.paths`.

When updating the extdriver.paths configuration file, remove the default Teradata version number and replace it with the Teradata version number that you're using. Make sure that you include `\bin` in the path. For example if you're using Teradata 14.10, change `C:\Program Files\Teradata\Client\15.10\bin` to `C:\Program Files\Teradata\Client\14.10\bin`.

Oracle Analytics Desktop Printing and Exporting FAQs

This topic answers common questions about printing and exporting.

Why don't I see images in workbooks or from background maps when I print pages or when I export images in formats such as PDF, PPT, and PNG?

You or a visualization builder might have added an image to a workbook or background map by referencing that image with a URL. For the image to print or be exported in various formats, the external website hosting that image must have the Access-Control-Allow-Origin header from the host server, to ensure proper security. If a map background includes an image reference that's taken from an external website that doesn't have this header, you won't see the image.

For more information about this header, see `https://www.w3.org/wiki/CORS_Enabled`.

B
Troubleshoot Visualization Issues

This topic describes common problems that you might encounter when working with visualizations and explains how to solve them.

When I import a workbook, I get an error stating that the workbook, data source, or connection already exists

When you're trying to import a workbook, you might receive the following error message:

"There is already a workbook, data source or connection with the same name as something you're trying to import. Do you want to continue the import and replace the existing content?"

This error message is displayed because one or more of the components exported with the workbook is already on your system. When a workbook is exported, the outputted .DVA file includes the workbook's associated data sources and connection string. To resolve this error, you can either click **OK** to replace the components on your system, or you can click **Cancel** and go into your system and manually delete the components.

This error message is also displayed when the workbook you're trying to import contains no data. When you export a workbook without data, the workbook's and data sources' metadata are included in the .DVA. To resolve this issue, you can click **OK** to replace the components on your system, or you can click **Cancel** and go into your system and manually delete the data source or connection that's causing the error.

When I try to build a connection to Teradata, I get an error and the connection is not saved

When you're trying to create a connection to Teradata, you might receive the following error message:

"Failed to save the connection. Cannot create a connection since there are some errors. Please fix them and try again."

This error message is displayed because the version of Teradata that you're using is different from the version supported by Oracle Analytics Desktop. To resolve this issue, update the extdriver.paths configuration file. This configuration file is located here: `C:\<your directory>\AppData\Local\OracleAnalyticsDesktop\extdrvier.paths`. For example, `C:\Users\jsmith\AppData\Local\OracleAnalyticsDesktop\extdriver.paths`.

To update the extdriver.paths configuration file, remove the default Teradata version number and replace it with the Teradata version number that you're using. Make sure that you include \bin in the path. For example if you're using Teradata 14.10, then change `C:\Program Files\Teradata\Client\15.10\bin` to `C:\Program Files\Teradata\Client\14.10\bin`. See What if I'm using a Teradata version different that the one supported by Oracle Analytics Desktop?

I have issues when I try to refresh data for file-based data sources

Keep in mind the following requirements when you refresh data for Microsoft Excel, CSV, or TXT data sources:

- To refresh an Excel file, ensure that the newer spreadsheet file contains a sheet with the same name as the original file you uploaded. If a sheet is missing, then you must fix the file to match the sheets in the original uploaded file.
- If the Excel, CSV, or TXT file that you reload is missing some columns, then you'll get an error stating that your data reload has failed. If this happens, then you must fix the file to match the columns in the original uploaded file.
- If the Excel, CSV, or TXT file you used to create the data source was moved or deleted, then the connection path is crossed out in the Data Source dialog. You can reconnect the data source to its original source file, or connect it to a replacement file, by right-clicking the data source in the Display pane and in the Options menu select **Reload Data**. You can then browse for and select the file to load.
- If you reloaded an Excel, CSV, or TXT file with new columns, then the new columns are marked as hidden and don't display in the Data Panel for existing workbooks using the dataset. To unhide these columns, click the **Hidden** option.

Your Excel spreadsheets must have a specific structure. See About Files for Datasets.

I can't refresh data from a MongoDB data source

The first time you connect to MongoDB, the MongoDB driver creates a cache file. If the MongoDB schema was renamed and you try to reload a MongoDB data source or use the data source in a workbook, then you might get an error or Oracle Analytics doesn't respond.

To correct this error, you need to clear the MongoDB cache. To clear the cache, delete the contents of the following directory: `C:\<your directory>\AppData\Local\Progress\DataDirect\MongoDB_Schema`. For example, `C:\Users\jsmith\AppData\Local\Progress\DataDirect\MongoDB_Schema`

Oracle Support needs a file to help me diagnose a technical issue

If you're working with the Oracle Support team to resolve a specific issue, they may ask you to generate a diagnostic dump file. This file contains the following information:

- Version information
- Installer logs
- Application component logs containing status information for Oracle Business Intelligence Presentation Server, Oracle Business Intelligence Server, and other critical components
- Jetty logs
- Data Security Standard (DSS) logs
- Webcat metadata plugin logs
- Derby logs
- Server Administration Workbench (SAW) Server logs
- Oracle Business Intelligence Presentation Server logs
- Oracle Business Intelligence Server logs

1. Open the command prompt and change the directory to the Oracle Analytics Desktop installation directory (for example, C:\Program Files\Oracle Analytics Desktop).
2. Type `diagnostic_dump.cmd` and then provide a name for the .zip output file (for example, output.zip).
3. Press Enter to execute the command.
   You can find the diagnostic output .zip file in your installation directory.

I need to find more information about a specific issue

The community forum is another great resource that you can use to find out more information about the problem you're having.

You can find the forum here: Oracle Community Forum.

C

Accessibility Features and Tips

This topic describes accessibility features and information for Oracle Analytics Desktop.

Topics:

- Start Oracle Analytics Desktop with Accessibility Features Enabled
- Keyboard Shortcuts for Visualizations
- Keyboard Shortcuts for Data Flow

Start Oracle Analytics Desktop with Accessibility Features Enabled

You can enable features that improve navigation and make the interface accessible.

To enable the accessibility features, you must start Oracle Analytics Desktop from the command line. Open a command window and enter the following:

On Windows:

```
dvdesktop.exe - sdk
```

On Mac:

```
open /Applications/dvdesktop.app --args -sdk
```

When you run the command, Oracle Analytics Desktop opens in a web browser.

Keyboard Shortcuts for Visualizations

You can use keyboard shortcuts to navigate and to perform actions in visualizations.

Use these keyboard shortcuts for working with a workbook in the Visualize Canvas.

Task	Keyboard Shortcut
Add data columns to a workbook.	Shift + F10
Export content to PDF.	Ctrl + Shift + E (Win) Command + Shift + E (Mac)
Export content to PPT.	Ctrl + Shift + X (Win) Command + Shift + X (Mac)
Print content.	Ctrl + P (Win) Command + Shift + P (Mac)
Reverse the last undo.	Ctrl + Y
Save a newly created workbook with a specific name.	Ctrl + Shift + S
Save a workbook with the changes.	Ctrl + S

Task	Keyboard Shortcut
Undo the last change.	Ctrl + Z

Use these keyboard shortcuts while working on a visualization in the Visualize canvas.

Task	Keyboard Shortcut
Copy a visualization to paste it to another canvas in the workbook or to a canvas in another workbook.	Ctrl + C
Paste the visualization into a canvas in the workbook or into a canvas in another workbook.	Ctrl + V
Duplicate a visualization.	Ctrl + D
Delete a visualization.	Delete key

Use these keyboard shortcuts while working with a filter in the filter panel on the filter bar.

Task	Keyboard Shortcut
Add the search string to the selection list.	Ctrl + Enter

Use these keyboard shortcuts when you want to open, create, or edit artifacts such as datasets, workbooks, data flows, and sequences in a new tab or window.

Task	Keyboard Shortcut
Open an artifact in a new browser tab.	Ctrl + Click the artifact
Open an artifact in a new browser window.	Shift + Click the artifact

Keyboard Shortcuts for Data Flow

Use these keyboard shortcuts to perform actions in the data flow editor.

Task	Keyboard Shortcut
Undo the last change.	Ctrl + Z / Command + Z
Reverse the last undo.	Ctrl + Y / Command + Y

D

Data Sources and Data Types Reference

Find out about supported data sources, databases, and data types.

Topics

- Supported Data Sources
- Oracle Applications Connector
- Certification - Supported Data Types

Supported Data Sources

With Oracle Analytics Desktop, you can connect to many different data sources. Data sources are sorted alphabetically by Oracle databases first, then other databases.

Data Source	Version	Oracle Analytics Desktop for Windows	Oracle Analytics Desktop for Mac	More Information
Oracle Applications	11.1.1.9+ or Fusion Applications Release 8 and later	Yes	Yes	Connector supports several Oracle SaaS Applications. See Oracle Applications Connector. See also Create an Oracle Applications Connection.
Oracle Autonomous Data Warehouse	-	Yes	Yes	Connection to public IP address only. You can connect to multiple Oracle Autonomous Data Warehouse data sources. Upload a wallet for each connection. Supports saving output from data flows. See Connect to Oracle Autonomous Data Warehouse .

Data Source	Version	Oracle Analytics Desktop for Windows	Oracle Analytics Desktop for Mac	More Information
Oracle Database	11.2.0.4+ 12.1+ 12.2+ 18+ 19+	Yes	Yes	Use the Oracle Database connection type to connect to Oracle Database Classic Cloud Service. You can connect to multiple database services. Upload a wallet for each connection. Supports saving output from data flows. Ensure that the appropriate security access rules are in place to allow a network connection to the database service on the database listening port. See Create a Connection for an Oracle Database.
Oracle Essbase	11.1.2.4.0+ 21c	Yes	Yes	See Create a Connection to Oracle Essbase. You can't use Oracle Essbase datasets in Data Flows. You can't blend datasets that use Oracle Essbase data sources.
Oracle Netsuite	Netsuite Release 2019.2 (JDBC Driver 8.10.85.0)	Yes	Yes	See Connect to NetSuite .
Oracle B2C Service	1.2	Yes	No	-
Oracle Talent Acquisition	-	Yes	Yes	-
Actian Ingres	5.0+	Yes	No	-
Actian Matrix	5.0+	Yes	No	-
Actian Vector	5.0+	Yes	No	-
Amazon Aurora	-	Yes	No	-

Data Source	Version	Oracle Analytics Desktop for Windows	Oracle Analytics Desktop for Mac	More Information
Amazon EMR	Amazon EMR 4.7.2 running Amazon Hadoop 2.7.2 and Hive 1.0.0 Amazon EMR (MapR) - Amazon Machine Image (AMI) 3.3.2 running MapR Hadoop M3 and Hive 0.13.1	Yes	No	Complex data types not supported.
Amazon Redshift	1.0.1036 +	Yes	No	-
Apache Drill	1.7+	Yes	No	-
Apache Hive	2.3.0+ 3.0+	Yes	No	Supports Kerberos. Supports saving output from data flows.
Cassandra	3.10	Yes	No	-
CSV File	-	Yes	Yes	-
DB2	10.1+ 10.5+	Yes	No	-
DropBox	-	Yes	No	-
Elastic Search	5.6.4+	Yes	-	-
Google Analytics	-	Yes	No	-
Google Cloud	-	Yes	No	-
Google Drive	-	Yes	No	-
GreenPlum	4.3.8+	Yes	No	-
HortonWorks Hive	1.2+	Yes	No	Supports Kerberos. Supports saving output from data flows.
HP Vertica	7+	Yes	No	-
IBM BigInsights Hive	1.2+	Yes	No	Supports Kerberos.
Impala	2.7+	Yes	No	
Informix	12.1+	Yes	No	-
JDBC	Generic JDBC driver support	Yes	No	See Create JDBC Connections.

Data Source	Version	Oracle Analytics Desktop for Windows	Oracle Analytics Desktop for Mac	More Information
MapR Hive	1.2+	-	-	Supports Kerberos. Supports saving output from data flows.
Microsoft Access	2013 2016	Yes	No	-
Microsoft Azure SQL Database	-	Yes*	TBD	* Use the **SQL Server** connection type on the Create Connection page.
Microsoft Azure Synapse Analytics	-	Yes	Yes	-
Microsoft Excel	-	Yes	Yes	Only XLSX files (and XLS with unpivoted data).
MonetDB	5+	Yes	No	-
MongoDB	3.2.5	Yes	No	-
MySQL	5.6+ 5.7+	Yes	No	Connections to MySQL Community Edition aren't supported.
Netezza	7	Yes	No	-
OData	4.0+	Yes	No	-
ODBC	Generic ODBC driver support	-	-	-
Pivotal HD Hive	-	Yes	No	Supports Kerberos.
PostgreSQL	9.0+	Yes	No	-
Presto	-	Yes	No	-
Salesforce	-	Yes	No	-
Spark	1.6+	Yes	No	Supports saving output from data flows.
SQL Server	2014 2016	Yes	No	-
Sybase ASE	15.7+	Yes	No	-
Sybase IQ	16+	Yes	No	-
Teradata	14 15 16 16.10	Yes	No	-
Teradata Aster	6.10+	Yes	No	-

Oracle Applications Connector

Oracle Applications Connector supports Oracle Applications Cloud. You can also use Oracle Applications Connector to connect to your on-premises Oracle BI Enterprise

Edition deployments (if patched to an appropriate level) and another Oracle Analytics service.

Oracle Applications Cloud applications you can connect to:

- Oracle Sales
- Oracle Fusion Cloud Financials
- Oracle Fusion Cloud Human Capital Management
- Oracle Supply Chain Cloud
- Oracle Procurement Cloud
- Oracle Project Cloud
- Oracle Loyalty Cloud

> **Note:**
>
> When you connect to Oracle Applications, you access the data from an Oracle Transactional Business Intelligence (OTBI) report. These reports are subject to caching in OTBI, and the data available in Oracle Analytics is based on the cached data. You can't control the cache behavior in OTBI from Oracle Analytics.

Certification - Supported Data Types

Here're the supported data types for the latest version of Oracle Analytics Server.

Topics:

- Supported Base Data Types
- Supported Data Types by Database

Supported Base Data Types

When reading from a data source, Oracle Analytics attempts to map incoming data types to the supported data types.

For example, a database column that contains only date values is formatted as a DATE, a spreadsheet column that contains a mix of numerical and string values is formatted as a VARCHAR, and a data column that contains numerical data with fractional values uses DOUBLE or FLOAT.

In some cases Oracle Analytics can’t convert a source data type. To work around this data type issue, you can manually convert a data column to a supported type by entering SQL commands. In other cases, Oracle Analytics can't represent binary and complex data types such as BLOB, JSON, and XML.

Note that some data types aren't supported. You'll see an error message if the data source contains unsupported data types.

Oracle Analytics supports the following base data types:

- **Number Types** — SMALLINT, SMALLUNIT, TINYINT, TINYUINT, UINT, BIT, FLOAT, INT, NUMERIC, DOUBLE
- **Date Types** — DATE, DATETIME, TIMESTAMP, TIME

- **String Types** — LONGVARCHAR, CHAR, VARCHAR

Supported Data Types by Database

Oracle Analytics supports the following data types.

Database Type	Supported Data Types
Oracle	BINARY DOUBLE, BINARY FLOAT CHAR, NCHAR CLOB, NCLOB DATE FLOAT NUMBER, NUMBER (p,s), NVARCHAR2, VARCHAR2 ROWID TIMESTAMP, TIMESTAMP WITH LOCAL TIMEZONE, TIMESTAMP WITH TIMEZONE
DB2	BIGINT CHAR, CLOB DATE, DECFLOAT, DECIMAL, DOUBLE FLOAT INTEGER LONGVAR NUMERIC REAL SMALLINT TIME, TIMESTAMP VARCHAR
SQL Server	BIGINT, BIT CHAR DATE, DATETIME, DATETIME2, DATETIMEOFFSET, DECIMAL FLOAT INT MONEY NCHAR, NTEXT, NUMERIC, NVARCHAR, NVARCHAR(MAX) REAL SMALLDATETIME, SMALLINT, SMALLMONEY TEXT, TIME, TINYINT VARCHAR, VARCHAR(MAX) XML

Database Type	Supported Data Types
MySQL	BIGINT, BIGINT UNSIGNED CHAR DATE, DATETIME, DECIMAL, DECIMAL UNSIGNED, DOUBLE, DOUBLE UNSIGNED FLOAT, FLOAT UNSIGNED INTEGER, INTEGER UNSIGNED LONGTEXT MEDIUMINT, MEDIUMINT UNSIGNED, MEDIUMTEXT SMALLINT, SMALLINT UNSIGNED TEXT, TIME, TIMESTAMP, TINYINT, TINYINT UNSIGNED, TINYTEXT VARCHAR YEAR
Apache Spark	BIGINT, BOOLEAN DATE, DECIMAL, DOUBLE FLOAT INT SMALLINT, STRING TIMESTAMP, TINYINT VARCHAR
Teradata	BIGINT, BYTE, BYTEINT CHAR, CLOB DATE, DECIMAL, DOUBLE FLOAT INTEGER NUMERIC REAL SMALLINT TIME, TIMESTAMP VARCHAR

E

Data Preparation Reference

This topic describes the set and types of recommendation and options you can use to perform data transform changes to a dataset.

Topics:

- Transform Recommendation Reference
- Column Menu Options for Quick Data Transformations

Transform Recommendation Reference

Find out about the data transform options.

Option	Description
Edit	Edits the column. For example, you can change the name, select another column, or update functions.
Hide	Hides the column in the Data Panel and in the visualizations. If you want to see the hidden columns, click **Hidden columns** (ghost icon) on the page footer. You can then unhide individual columns or unhide all the hidden columns at the same time.
Group, Conditional Group	Select **Group** to create your own custom groups. For example, you can group States together with custom regions, and you can categorize dollar amounts into groups indicating small, medium, and large.
Split	Splits a specific column value into parts. For example, you can split a column called, Name, into first and last name.
Uppercase	Updates the contents of a column with the values in all uppercase letters.
Lowercase	Updates the contents of a column with the values all in lowercase letters.
Sentence Case	Updates the contents of a column to make the first letter of the first word of a sentence uppercase.
Rename	Allows you to change the name of any column.
Duplicate	Creates a column with identical content of the selected column.
Convert to Text	Changes the data type of a column to text.
Replace	Changes specific text in the selected column to any value that you specify. For example, you can change all instances of *Mister* to *Mr.* in the column.
Create	Creates a column based on a function.
Convert to Number	Changes the data type of the column to number, which deletes any values that aren't numbers from the column.
Convert to Date	Changes the data type of the column to date and deletes any values that aren’t dates from the column.
Bin	Creates your own custom groups for number ranges. For example, you can create bins for an Age column with age ranges binned into Pre-Teen, Young Adult, Adult, or Senior based on custom requirements.
Log	Calculates the natural logarithm of an expression.
Power	Raises the values of a column to the power that you specify. The default power is 2.

Option	Description
Square Root	Creates a column populated with the square root of the value in the column selected.

Column Menu Options for Quick Data Transformations

You can use the following column menu options to transform data in a data flow.

Options	Description
Bin	Create your custom groups for number ranges.
Convert to Number	Change the data type of the column to number and delete any values that aren't numbers.
Convert to Text	Change the data type of a column to text.
Delete	Select and remove a column from the dataset.
Duplicate	Create a column with data that’s identical to the selected column.
Group	Create a custom group to combine related values. For example, you can group states with custom regions and categorize dollar amounts into groups showing small, medium, and large.
Lowercase	Convert all the text in the column to lowercase.
Merge Columns	Combine two or more columns to display as one.
Rename	Change the column name.
Sentence Case	Convert the first letter of the first word to uppercase on each row in a column.
Transform	Modify the column data by using an expression.
Trim	Remove leading and trailing spaces from text data. Oracle Analytics doesn't show leading or trailing spaces in the data preview, but if the original data source contained leading or trailing spaces, this can affect queries.
Uppercase	Convert all the text in the column to uppercase.

Developer Reference

The Visualize canvas includes developer options that enable you to embed analytics content in other applications, and review performance statistics for visualization components.

Note:

To use developer options, you must have administrator or content author privileges.

To display Developer options, open a workbook, display the Visualize pane, and click **Menu**, then **Developer**. If the **Developer** option isn't displayed, enable Developer Options before you start. See Enable Developer Options.

Tab	Description
Performance Tools	Review performance statistics such as query time, server time, and streaming time for visualization components in a workbook. Click **Refresh** to get the latest statistics. You can then use the following tabs to analyse performance: Summary, Logical SQL, Query Generation, Execution Log.
Embed	Embed Oracle Analytics content in other applications.
JSON	Look up a column reference for an embed that uses filters.
XML, Datasets, Prep Datasets, Prep UIOptions	Don't use these tabs. They're for internal diagnostics only.

Enable Developer Options

Developer options aren't available if you log into Oracle Analytics with consumer privileges only. However, you can analyze the performance of a visualization canvas by appending a workbook URL in your browser with `&devtools=true`.

1. In the Home page, click the user icon, then click **Profile**.
2. On the Admin page, click **Advanced**, and click **Enable Developer Options**.
3. Click **Save**, then **Close**.

   If you have administrator or content author privileges, you can now display developer options below the Visualize pane in workbook designer (click **Menu**, then **Developer**).

F

Expression Editor Reference

This section describes the expression elements that you can use in the Expression Editor.

Topics:

- SQL Operators
- Conditional Expressions
- Functions
- Constants
- Types

SQL Operators

SQL operators are used to specify comparisons between expressions.

You can use various types of SQL operators.

Operator	Example	Description	Syntax
BETWEEN	`"COSTS"."UNIT_COST" BETWEEN 100.0 AND 5000.0`	Determines if a value is between two non-inclusive bounds. `BETWEEN` can be preceded with `NOT` to negate the condition.	`BETWEEN [LowerBound] AND [UpperBound]`
IN	`"COSTS"."UNIT_COST" IN(200, 600, 'A')`	Determines if a value is present in a set of values.	`IN ([Comma Separated List])`
IS NULL	`"PRODUCTS"."PROD_NAME" IS NULL`	Determines if a value is null.	`IS NULL`
LIKE	`"PRODUCTS"."PROD_NAME" LIKE 'prod%'`	Determines if a value matches all or part of a string. Often used with wildcard characters to indicate any character string match of zero or more characters (%) or any single character match (_).	`LIKE`
+	`(FEDERAL_REVENUE + LOCAL_REVENUE) - TOTAL_EXPENDITURE`	Plus sign for addition.	`+`
-	`(FEDERAL_REVENUE + LOCAL_REVENUE) - TOTAL_EXPENDITURE`	Minus sign for subtraction.	`-`

Operator	Example	Description	Syntax
* *or* X	`SUPPORT_SERVICES_EXPENDITURE * 1.5`	Multiply sign for multiplication.	`*` `X`
/	`CAPITAL_OUTLAY_EXPENDITURE/ 1.05`	Divide by sign for division.	`/`
%		Percentage	`%`
||	`STATE|| CAST(YEAR AS CHAR(4))`	Character string concatenation.	`||`
(	`(FEDERAL_REVENUE + LOCAL_REVENUE) - TOTAL_EXPENDITURE`	Open parenthesis.	`(`
)	`(FEDERAL_REVENUE + LOCAL_REVENUE) - TOTAL_EXPENDITURE`	Close parenthesis.	`)`
>	`YEAR > 2000 and YEAR < 2016 and YEAR <> 2013`	Greater than sign, indicating values higher than the comparison.	`>`
<	`YEAR > 2000 and YEAR < 2016 and YEAR <> 2013`	Less than sign, indicating values lower than the comparison.	`<`
=		Equal sign, indicating the same value.	`=`
>=		Greater than or equal to sign, indicating values the same or higher than the comparison.	`>=`
<=		Less than or equal to sign, indicating values the same or lower than the comparison.	`<=`
<>	`YEAR > 2000 and YEAR < 2016 and YEAR <> 2013`	Not equal to, indicating values higher or lower, but different.	`<>`
,	`STATE in ('ALABAMA','CALIFORNIA')`	Comma, used to separate elements in a list.	`,`

Functions

There are various types of functions that you can use in expressions.

Topics:

- Aggregate Functions
- Analytics Functions
- Calendar Functions
- Conversion Functions
- Display Functions
- Evaluate Functions
- Mathematical Functions
- Running Aggregate Functions
- String Functions
- System Functions
- Time Series Functions

Aggregate Functions

Aggregate functions perform operations on multiple values to create summary results.

The following list describes the aggregation rules that are available for columns and measure columns. The list also includes functions that you can use when creating calculated items for analyses.

- **Default** — Applies the default aggregation rule as in the data model or by the original author of the analysis. Not available for calculated items in analyses.
- **Server Determined** — Applies the aggregation rule that's determined by the Oracle Analytics (such as the rule that is defined in the data model). The aggregation is performed within Oracle Analytics for simple rules such as Sum, Min, and Max. Not available for measure columns in the Layout pane or for calculated items in analyses.
- **Sum** — Calculates the sum obtained by adding up all values in the result set. Use this for items that have numeric values.
- **Min** — Calculates the minimum value (lowest numeric value) of the rows in the result set. Use this for items that have numeric values.
- **Max** — Calculates the maximum value (highest numeric value) of the rows in the result set. Use this for items that have numeric values.
- **Average** — Calculates the average (mean) value of an item in the result set. Use this for items that have numeric values. Averages on tables and pivot tables are rounded to the nearest whole number.
- **First** — In the result set, selects the first occurrence of the item for measures. For calculated items, selects the first member according to the display in the Selected list. Not available in the Edit Column Formula dialog.

- **Last** — In the result set, selects the last occurrence of the item. For calculated items, selects the last member according to the display in the Selected list. Not available in the Edit Column Formula dialog.
- **Count** — Calculates the number of rows in the result set that have a non-null value for the item. The item is typically a column name, in which case the number of rows with non-null values for that column are returned.
- **Count Distinct** — Adds distinct processing to the Count function, which means that each distinct occurrence of the item is counted only once.
- **None** — Applies no aggregation. Not available for calculated items in analyses.
- **Report-Based Total (when applicable)** — If not selected, specifies that the Oracle Analytics should calculate the total based on the entire result set, before applying any filters to the measures. Not available in the Edit Column Formula dialog or for calculated items in analyses. Only available for attribute columns.

Function	Example	Description	Syntax
AGGREGATE AT	`AGGREGATE(sales AT month, region)`	Aggregates columns based on the level or levels you specify. *measure* is the name of a measure column. level is the level at which you want to aggregate. You can optionally specify more than one level. You can't specify a level from a dimension that contains levels that are being used as the measure level for the measure you specified in the first argument. For example, you can't write the function as AGGREGATE(yearly_sales AT month) because *month* is from the same time dimension that's being used as the measure level for *yearly_sales*.	`AGGREGATE(measure AT level [, level1, levelN])`
AVG	`Avg(Sales)`	Calculates the average (mean) of a numeric set of values.	`AVG(expr)`
AVGDISTINCT		Calculates the average (mean) of all distinct values of an expression.	`AVG(DISTINCT expr)`

Function	Example	Description	Syntax
BIN	`BIN(revenue BY productid, year WHERE productid > 2 INTO 4 BINS RETURNING RANGE_LOW)`	Classifies a given numeric expression into a specified number of equal width buckets. The function can return either the bin number or one of the two end points of the bin interval. numeric_expr is the measure or numeric attribute to bin. BY grain_expr1,…, grain_exprN is a list of expressions that define the grain at which the numeric_expr is calculated. BY is required for measure expressions and is optional for attribute expressions. WHERE a filter to apply to the numeric_expr before the numeric values are assigned to bins INTO number_of_bins BINS is the number of bins to return BETWEEN min_value AND max_value is the min and max values used for the end points of the outermost bins RETURNING NUMBER indicates that the return value should be the bin number (1, 2, 3, 4, etc.). This is the default. RETURNING RANGE_LOW indicates the lower value of the bin interval RETURNING RANGE_HIGH indicates the higher value of the bin interval	`BIN(numeric_expr [BY grain_expr1, ..., grain_exprN] [WHERE condition] INTO number_of_bins BINS [BETWEEN min_value AND max_value] [RETURNING {NUMBER | RANGE_LOW | RANGE_HIGH}])`
BottomN		Ranks the lowest n values of the expression argument from 1 to n, 1 corresponding to the lowest numerical value. *expr* is any expression that evaluates to a numerical value. integer is any positive integer. Represents the bottom number of rankings displayed in the result set, 1 being the lowest rank.	`BottomN(expr, integer)`
COUNT	`COUNT(Products)`	Determines the number of items with a non-null value.	`COUNT(expr)`
COUNTDISTINCT		Adds distinct processing to the COUNT function. *expr* is any expression.	`COUNT(DISTINCT expr)`
COUNT*	`SELECT COUNT(*) FROM Facts`	Counts the number of rows.	`COUNT(*)`
First	`First(Sales)`	Selects the first non-null returned value of the expression argument. The `First` function operates at the most detailed level specified in your explicitly defined dimension.	`First([NumericExpression)]`
Last	`Last(Sales)`	Selects the last non-null returned value of the expression.	`Last([NumericExpression)]`
MAVG		Calculates a moving average (mean) for the last n rows of data in the result set, inclusive of the current row. *expr* is any expression that evaluates to a numerical value. integer is any positive integer. Represents the average of the last n rows of data.	`MAVG(expr, integer)`

Function	Example	Description	Syntax
MAX	`MAX(Revenue)`	Calculates the maximum value (highest numeric value) of the rows satisfying the numeric expression argument.	`MAX(expr)`
MEDIAN	`MEDIAN(Sales)`	Calculates the median (middle) value of the rows satisfying the numeric expression argument. When there are an even number of rows, the median is the mean of the two middle rows. This function always returns a double.	`MEDIAN(expr)`
MIN	`MIN(Revenue)`	Calculates the minimum value (lowest numeric value) of the rows satisfying the numeric expression argument.	`MIN(expr)`
NTILE		Determines the rank of a value in terms of a user-specified range. It returns integers to represent any range of ranks. NTILE with numTiles=100 returns what is commonly called the "percentile" (with numbers ranging from 1 to 100, with 100 representing the high end of the sort). *expr* is any expression that evaluates to a numerical value. numTiles is a positive, nonnull integer that represents the number of tiles.	`NTILE(expr, numTiles)`
PERCENTILE		Calculates a percentile rank for each value satisfying the numeric expression argument. The percentile rank ranges are between 0 (0th percentile) to 1 (100th percentile). *expr* is any expression that evaluates to a numerical value.	`PERCENTILE(expr)`
RANK	`RANK(chronological_key, null, year_key_columns)`	Calculates the rank for each value satisfying the numeric expression argument. The highest number is assigned a rank of 1, and each successive rank is assigned the next consecutive integer (2, 3, 4,...). If certain values are equal, they'reare assigned the same rank (for example, 1, 1, 1, 4, 5, 5, 7...). *expr* is any expression that evaluates to a numerical value.	`RANK(expr)`

STDDEV	`STDDEV(Sales)`
	`STDDEV(DISTINCT Sales)`
		Returns the standard deviation for a set of values. The return type is always a double.	`STDDEV(expr)`

STDDEV_POP	`STDDEV_POP(Sales)`
	`STDDEV_POP(DISTINCT Sales)`
		Returns the standard deviation for a set of values using the computational formula for population variance and standard deviation.	`STDDEV_POP([NumericExpression])`

SUM	`SUM(Revenue)`	Calculates the sum obtained by adding up all values satisfying the numeric expression argument.	`SUM(expr)`

Function	Example	Description	Syntax
SUMDISTINCT		Calculates the sum obtained by adding all of the distinct values satisfying the numeric expression argument. *expr* is any expression that evaluates to a numerical value.	`SUM(DISTINCT expr)`
TOPN		Ranks the highest n values of the expression argument from 1 to n, 1 corresponding to the highest numerical value. *expr* is any expression that evaluates to a numerical value. integer is any positive integer. Represents the top number of rankings displayed in the result set, 1 being the highest rank.	`TOPN(expr, integer)`

Analytics Functions

Analytics functions allow you to explore data using models such as trendline and cluster.

Function	Example	Description	Syntax
TRENDLINE	`TRENDLINE(revenue, (calendar_year, calendar_quarter, calendar_month) BY (product), 'LINEAR', 'VALUE')`	Oracle recommends that you apply a Trendline using the **Add Statistics** property when viewing a visualization. See Adjust Visualization Properties. Fits a linear, polynomial, or exponential model, and returns the fitted values or model. The *numeric_expr* represents the Y value for the trend and the *series* (time columns) represent the X value.	`TRENDLINE(numeric_expr, ([series]) BY ([partitionBy]), model_type, result_type)`
CLUSTER	`CLUSTER((product, company), (billed_quantity, revenue), 'clusterName', 'algorithm=k-means;numClusters=%1;maxIter=%2;useRandomSeed=FALSE;enablePartitioning=TRUE', 5, 10)`	Collects a set of records into groups based on one or more input expressions using K-Means or Hierarchical Clustering.	`CLUSTER((dimension_expr1 , ... dimension_exprN), (expr1, ... exprN), output_column_name, options, [runtime_binded_options])`
OUTLIER	`OUTLIER((product, company), (billed_quantity, revenue), 'isOutlier', 'algorithm=kmeans')`	Classifies a record as Outlier based on one or more input expressions using K-Means or Hierarchical Clustering or Multi-Variate Outlier detection Algorithms.	`OUTLIER((dimension_expr1 , ... dimension_exprN), (expr1, ... exprN), output_column_name, options, [runtime_binded_options])`

Function	Example	Description	Syntax
REGR	`REGR(revenue, (discount_amount), (product_type, brand), 'fitted', '')`	Fits a linear model and returns the fitted values or model. This function can be used to fit a linear curve on two measures.	`REGR(y_axis_measure_expr, (x_axis_expr), (category_expr1, ..., category_exprN), output_column_name, options, [runtime_binded_options])`
EVALUATE_SCRIPT	`EVALUATE_SCRIPT('filerepo://obiee.Outliers.xml', 'isOutlier', 'algorithm=kmeans;id=%1;arg1=%2;arg2=%3;useRandomSeed=False;', customer_number, expected_revenue, customer_age)`	Executes a Python script as specified in the *script_file_path*, passing in one or more columns or literal expressions as input. The output of the function is determined by the *output_column_name*.	`EVALUATE_SCRIPT(script_file_path, output_column_name, options, [runtime_binded_options])`

Calendar Functions

Calendar functions manipulate data of the data types `DATE` and `DATETIME` based on a calendar year.

Function	Example	Description	Syntax
CURRENT_Date	`CURRENT_DATE`	Returns the current date. The date is determined by the system in which the Oracle BI is running.	`CURRENT_DATE`
CURRENT_TIME	`CURRENT_TIME(3)`	Returns the current time to the specified number of digits of precision, for example: HH:MM:SS.SSS If no argument is specified, the function returns the default precision.	`CURRENT_TIME(expr)`
CURRENT_TIMESTAMP	`CURRENT_TIMESTAMP(3)`	Returns the current date/timestamp to the specified number of digits of precision.	`CURRENT_TIMESTAMP(expr)`
DAYNAME	`DAYNAME(Order_Date)`	Returns the name of the day of the week for a specified date expression.	`DAYNAME(expr)`
DAYOFMONTH	`DAYOFMONTH(Order_Date)`	Returns the number corresponding to the day of the month for a specified date expression.	`DAYOFMONTH(expr)`
DAYOFWEEK	`DAYOFWEEK(Order_Date)`	Returns a number between 1 and 7 corresponding to the day of the week for a specified date expression. For example, 1 always corresponds to Sunday, 2 corresponds to Monday, and so on through to Saturday which returns 7.	`DAYOFWEEK(expr)`
DAYOFYEAR	`DAYOFYEAR(Order_Date)`	Returns the number (between 1 and 366) corresponding to the day of the year for a specified date expression.	`DAYOFYEAR(expr)`

Function	Example	Description	Syntax
DAY_OF_QUARTER	`DAY_OF_QUARTER(Order_Date)`	Returns a number (between 1 and 92) corresponding to the day of the quarter for the specified date expression.	`DAY_OF_QUARTER(expr)`
HOUR	`HOUR(Order_Time)`	Returns a number (between 0 and 23) corresponding to the hour for a specified time expression. For example, 0 corresponds to 12 a.m. and 23 corresponds to 11 p.m.	`HOUR(expr)`
MINUTE	`MINUTE(Order_Time)`	Returns a number (between 0 and 59) corresponding to the minute for a specified time expression.	`MINUTE(expr)`
MONTH	`MONTH(Order_Time)`	Returns the number (between 1 and 12) corresponding to the month for a specified date expression.	`MONTH(expr)`
MONTHNAME	`MONTHNAME(Order_Time)`	Returns the name of the month for a specified date expression.	`MONTHNAME(expr)`
MONTH_OF_QUARTER	`MONTH_OF_QUARTER(Order_Date)`	Returns the number (between 1 and 3) corresponding to the month in the quarter for a specified date expression.	`MONTH_OF_QUARTER(expr)`
NOW	`NOW()`	Returns the current timestamp. The `NOW` function is equivalent to the `CURRENT_TIMESTAMP` function.	`NOW()`
QUARTER_OF_YEAR	`QUARTER_OF_YEAR(Order_Date)`	Returns the number (between 1 and 4) corresponding to the quarter of the year for a specified date expression.	`QUARTER_OF_YEAR(expr)`
SECOND	`SECOND(Order_Time)`	Returns the number (between 0 and 59) corresponding to the seconds for a specified time expression.	`SECOND(expr)`
TIMESTAMPADD	`TIMESTAMPADD(SQL_TSI_MONTH, 12,Time."Order Date")`	Adds a specified number of intervals to a timestamp, and returns a single timestamp. Interval options are: *SQL_TSI_SECOND*, *SQL_TSI_MINUTE*, *SQL_TSI_HOUR*, *SQL_TSI_DAY*, *SQL_TSI_WEEK*, *SQL_TSI_MONTH*, *SQL_TSI_QUARTER*, *SQL_TSI_YEAR*	`TIMESTAMPADD(interval, expr, timestamp)`
TIMESTAMPDIFF	`TIMESTAMPDIFF(SQL_TSI_MONTH, Time."Order Date",CURRENT_DATE)`	Returns the total number of specified intervals between two timestamps. Use the same intervals as TIMESTAMPADD.	`TIMESTAMPDIFF(interval, expr, timestamp2)`
WEEK_OF_QUARTER	`WEEK_OF_QUARTER(Order_Date)`	Returns a number (between 1 and 13) corresponding to the week of the quarter for the specified date expression.	`WEEK_OF_QUARTER(expr)`
WEEK_OF_YEAR	`WEEK_OF_YEAR(Order_Date)`	Returns a number (between 1 and 53) corresponding to the week of the year for the specified date expression.	`WEEK_OF_YEAR(expr)`
YEAR	`YEAR(Order_Date)`	Returns the year for the specified date expression.	`YEAR(expr)`

Conversion Functions

Conversion functions convert a value from one form to another.

Function	Example	Description	Syntax
CAST	`CAST(hiredate AS CHAR(40)) FROM employee`	Changes the data type of an expression or a null literal to another data type. For example, you can cast a *customer_name* (a data type of CHAR or VARCHAR) or *birthdate* (a datetime literal).	`CAST(expr AS type)`
IFNULL	`IFNULL(Sales, 0)`	Tests if an expression evaluates to a null value, and if it does, assigns the specified value to the expression.	`IFNULL(expr, value)`
INDEXCOL	`SELECT INDEXCOL(VALUEOF (NQ_SESSION.GEOGRAPHY _LEVEL), Country, State, City), Revenue FROM Sales`	Uses external information to return the appropriate column for the signed-in user to see.	`INDEXCOL([integer literal], [expr1] [, [expr2], ?-])`
NULLIF	`SELECT e.last_name, NULLIF(e.job_id, j.job_id) "Old Job ID" FROM employees e, job_history j WHERE e.employee_id = j.employee_id ORDER BY last_name, "Old Job ID";`	Compares two expressions. If they're equal, then the function returns NULL. If they're not equal, then the function returns the first expression. You can't specify the literal NULL for the first expression.	`NULLIF([expression], [expression])`
To_DateTime	`SELECT To_DateTime ('2009-03-0301:01:00', 'yyyy-mm-dd hh:mi:ss') FROM sales`	Converts string literals of *DateTime* format to a *DateTime* data type.	`To_DateTime([expression], [literal])`
VALUEOF	`SalesSubjectArea.Customer.Region = VALUEOF("Region Security"."REGION")`	References the value of an Oracle BI repository variable in a filter. Use *expr* variables as arguments of the VALUEOF function. Refer to static repository variables by name.	`VALUEOF(expr)`

Display Functions

Display functions operate on the result set of a query.

Function	Example	Description	Syntax
BottomN	`BottomN(Sales, 10)`	Returns the *n* lowest values of expression, ranked from lowest to highest.	`BottomN([NumericExpression], [integer])`
FILTER	`FILTER(Sales USING Product = 'widget')`	Computes the expression using the given preaggregate filter.	`FILTER(measure USING filter_expr)`

Function	Example	Description	Syntax
MAVG	`MAVG(Sales, 10)`	Calculates a moving average (mean) for the last *n* rows of data in the result set, inclusive of the current row.	`MAVG([NumericExpression], [integer])`
MSUM	`SELECT Month, Revenue, MSUM(Revenue, 3) as 3_MO_SUM FROM Sales`	Calculates a moving sum for the last *n* rows of data, inclusive of the current row. The sum for the first row is equal to the numeric expression for the first row. The sum for the second row is calculated by taking the sum of the first two rows of data, and so on. When the *n* th row is reached, the sum is calculated based on the last *n* rows of data.	`MSUM([NumericExpression], [integer])`
NTILE	`NTILE(Sales, 100)`	Determines the rank of a value in terms of a user-specified range. It returns integers to represent any range of ranks. The example shows a range from 1 to 100, with the lowest sale = 1 and the highest sale = 100.	`NTILE([NumericExpresssion], [integer])`
PERCENTILE	`PERCENTILE(Sales)`	Calculates a percent rank for each value satisfying the numeric expression argument. The percentile rank ranges are from 0 (1st percentile) to 1 (100th percentile), inclusive.	`PERCENTILE([NumericExpression])`
RANK	`RANK(Sales)`	Calculates the rank for each value satisfying the numeric expression argument. The highest number is assigned a rank of 1, and each successive rank is assigned the next consecutive integer (2, 3, 4,...). If certain values are equal, they're assigned the same rank (for example, 1, 1, 1, 4, 5, 5, 7...).	`RANK([NumericExpression])`
RCOUNT	`SELECT month, profit, RCOUNT(profit) FROM sales WHERE profit > 200`	Takes a set of records as input and counts the number of records encountered so far.	`RCOUNT([NumericExpression])`
RMAX	`SELECT month, profit, RMAX(profit) FROM sales`	Takes a set of records as input and shows the maximum value based on records encountered so far. The specified data type must be one that can be ordered.	`RMAX([NumericExpression])`
RMIN	`SELECT month, profit, RMIN(profit) FROM sales`	Takes a set of records as input and shows the minimum value based on records encountered so far. The specified data type must be one that can be ordered.	`RMIN([NumericExpression])`
RSUM	`SELECT month, revenue, RSUM(revenue) as RUNNING_SUM FROM sales`	Calculates a running sum based on records encountered so far. The sum for the first row is equal to the numeric expression for the first row. The sum for the second row is calculated by taking the sum of the first two rows of data, and so on.	`RSUM([NumericExpression])`
TOPN	`TOPN(Sales, 10)`	Returns the *n* highest values of expression, ranked from highest to lowest.	`TOPN([NumericExpression], [integer])`

Evaluate Functions

Evaluate functions are database functions that can be used to pass through expressions to get advanced calculations.

Embedded database functions can require one or more columns. These columns are referenced by %1 ... %N within the function. The actual columns must be listed after the function.

Function	Example	Description	Syntax
EVALUATE	`SELECT EVALUATE('instr(%1, %2)', address, 'Foster City') FROM employees`	Passes the specified database function with optional referenced columns as parameters to the database for evaluation.	`EVALUATE([string expression], [comma separated expressions])`
EVALUATE_AGGR	`EVALUATE_AGGR('REGR_SLOPE(%1, %2)', sales.quantity, market.marketkey)`	Passes the specified database function with optional referenced columns as parameters to the database for evaluation. This function is intended for aggregate functions with a `GROUP BY` clause.	`EVALUATE_AGGR('db_agg_function(%1...%N)' [AS datatype] [, column1, columnN])`

Mathematical Functions

The mathematical functions described in this section perform mathematical operations.

Function	Example	Description	Syntax
ABS	`ABS(Profit)`	Calculates the absolute value of a numeric expression. *expr* is any expression that evaluates to a numerical value.	`ABS(expr)`
ACOS	`ACOS(1)`	Calculates the arc cosine of a numeric expression. *expr* is any expression that evaluates to a numerical value.	`ACOS(expr)`
ASIN	`ASIN(1)`	Calculates the arc sine of a numeric expression. *expr* is any expression that evaluates to a numerical value.	`ASIN(expr)`
ATAN	`ATAN(1)`	Calculates the arc tangent of a numeric expression. *expr* is any expression that evaluates to a numerical value.	`ATAN(expr)`
ATAN2	`ATAN2(1, 2)`	Calculates the arc tangent of *y* /*x*, where *y* is the first numeric expression and *x* is the second numeric expression.	`ATAN2(expr1, expr2)`

Function	Example	Description	Syntax
CEILING	`CEILING(Profit)`	Rounds a non-integer numeric expression to the next highest integer. If the numeric expression evaluates to an integer, the `CEILING` function returns that integer.	`CEILING(expr)`
COS	`COS(1)`	Calculates the cosine of a numeric expression. *expr* is any expression that evaluates to a numerical value.	`COS(expr)`
COT	`COT(1)`	Calculates the cotangent of a numeric expression. *expr* is any expression that evaluates to a numerical value.	`COT(expr)`
DEGREES	`DEGREES(1)`	Converts an expression from radians to degrees. *expr* is any expression that evaluates to a numerical value.	`DEGREES(expr)`
EXP	`EXP(4)`	Sends the value to the power specified. Calculates *e* raised to the n-th power, where *e* is the base of the natural logarithm.	`EXP(expr)`
ExtractBit	`Int ExtractBit(1, 5)`	Retrieves a bit at a particular position in an integer. It returns an integer of either 0 or 1 corresponding to the position of the bit.	`ExtractBit([Source Number], [Digits])`
FLOOR	`FLOOR(Profit)`	Rounds a non-integer numeric expression to the next lowest integer. If the numeric expression evaluates to an integer, the `FLOOR` function returns that integer.	`FLOOR(expr)`
LOG	`LOG(1)`	Calculates the natural logarithm of an expression. *expr* is any expression that evaluates to a numerical value.	`LOG(expr)`
LOG10	`LOG10(1)`	Calculates the base 10 logarithm of an expression. *expr* is any expression that evaluates to a numerical value.	`LOG10(expr)`
MOD	`MOD(10, 3)`	Divides the first numeric expression by the second numeric expression and returns the remainder portion of the quotient.	`MOD(expr1, expr2)`
PI	`PI()`	Returns the constant value of pi.	`PI()`
POWER	`POWER(Profit, 2)`	Takes the first numeric expression and raises it to the power specified in the second numeric expression.	`POWER(expr1, expr2)`
RADIANS	`RADIANS(30)`	Converts an expression from degrees to radians. *expr* is any expression that evaluates to a numerical value.	`RADIANS(expr)`
RAND	`RAND()`	Returns a pseudo-random number between 0 and 1.	`RAND()`

Function	Example	Description	Syntax
RANDFromSeed	`RAND(2)`	Returns a pseudo-random number based on a seed value. For a given seed value, the same set of random numbers are generated.	`RAND(expr)`
ROUND	`ROUND(2.166000, 2)`	Rounds a numeric expression to *n* digits of precision. *expr* is any expression that evaluates to a numerical value. *integer* is any positive integer that represents the number of digits of precision.	`ROUND(expr, integer)`
SIGN	`SIGN(Profit)`	Returns the following: • 1 if the numeric expression evaluates to a positive number • -1 if the numeric expression evaluates to a negative number • 0 if the numeric expression evaluates to zero	`SIGN(expr)`
SIN	`SIN(1)`	Calculates the sine of a numeric expression.	`SIN(expr)`
SQRT	`SQRT(7)`	Calculates the square root of the numeric expression argument. The numeric expression must evaluate to a nonnegative number.	`SQRT(expr)`
TAN	`TAN(1)`	Calculates the tangent of a numeric expression. *expr* is any expression that evaluates to a numerical value.	`TAN(expr)`
TRUNCATE	`TRUNCATE(45.12345, 2)`	Truncates a decimal number to return a specified number of places from the decimal point. *expr* is any expression that evaluates to a numerical value. *integer* is any positive integer that represents the number of characters to the right of the decimal place to return.	`TRUNCATE(expr, integer)`

Running Aggregate Functions

Running aggregate functions perform operations on multiple values to create summary results.

Function	Example	Description	Syntax
MAVG		Calculates a moving average (mean) for the last *n* rows of data in the result set, inclusive of the current row. *expr* is any expression that evaluates to a numerical value. *integer* is any positive integer. Represents the average of the last *n* rows of data.	`MAVG(expr, integer)`

Function	Example	Description	Syntax
MSUM	`select month, revenue, MSUM(revenue, 3) as 3_MO_SUM from sales_subject_area`	Calculates a moving sum for the last n rows of data, inclusive of the current row. *expr* is any expression that evaluates to a numerical value. *integer* is any positive integer. Represents the sum of the last n rows of data.	`MSUM(expr, integer)`
RSUM	`SELECT month, revenue, RSUM(revenue) as RUNNING_SUM from sales_subject_area`	Calculates a running sum based on records encountered so far. *expr* is any expression that evaluates to a numerical value.	`RSUM(expr)`
RCOUNT	`select month, profit, RCOUNT(profit) from sales_subject_area where profit > 200`	Takes a set of records as input and counts the number of records encountered so far. *expr* is an expression of any datatype.	`RCOUNT(expr)`
RMAX	`SELECT month, profit,RMAX(profit) from sales_subject_area`	Takes a set of records as input and shows the maximum value based on records encountered so far. *expr* is an expression of any datatype.	`RMAX(expr)`
RMIN	`select month, profit,RMIN(profit) from sales_subject_area`	Takes a set of records as input and shows the minimum value based on records encountered so far. *expr* is an expression of any datatype.	`RMIN(expr)`

String Functions

String functions perform various character manipulations. They operate on character strings.

Function	Example	Description	Syntax
ASCII	`ASCII('a')`	Converts a single character string to its corresponding ASCII code, between 0 and 255. If the character expression evaluates to multiple characters, the ASCII code corresponding to the first character in the expression is returned. *expr* is any expression that evaluates to a character string.	`ASCII(expr)`
BIT_LENGTH	`BIT_LENGTH('abcdef')`	Returns the length, in bits, of a specified string. Each Unicode character is 2 bytes in length (equal to 16 bits). *expr* is any expression that evaluates to a character string.	`BIT_LENGTH(expr)`

Function	Example	Description	Syntax
CHAR	`CHAR(35)`	Converts a numeric value between 0 and 255 to the character value corresponding to the ASCII code. *expr* is any expression that evaluates to a numerical value between 0 and 255.	`CHAR(expr)`
CHAR_LENGTH	`CHAR_LENGTH(Customer_Name)`	Returns the length, in number of characters, of a specified string. Leading and trailing blanks aren't counted in the length of the string. *expr* is any expression that evaluates to a character string.	`CHAR_LENGTH(expr)`
CONCAT	`SELECT DISTINCT CONCAT ('abc', 'def') FROM employee`	Concatenates two character strings. *exprs* are expressions that evaluate to character strings, separated by commas. You must use raw data, not formatted data, with `CONCAT`.	`CONCAT(expr1, expr2)`
INSERT	`SELECT INSERT('123456', 2, 3, 'abcd') FROM table`	Inserts a specified character string into a specified location in another character string. *expr1* is any expression that evaluates to a character string. Identifies the target character string. *integer1* is any positive integer that represents the number of characters from the beginning of the target string where the second string is to be inserted. *integer2* is any positive integer that represents the number of characters in the target string to be replaced by the second string. *expr2* is any expression that evaluates to a character string. Identifies the character string to be inserted into the target string.	`INSERT(expr1, integer1, integer2, expr2)`
LEFT	`SELECT LEFT('123456', 3) FROM table`	Returns a specified number of characters from the left of a string. *expr* is any expression that evaluates to a character string *integer* is any positive integer that represents the number of characters from the left of the string to return.	`LEFT(expr, integer)`
LENGTH	`LENGTH(Customer_Name)`	Returns the length, in number of characters, of a specified string. The length is returned excluding any trailing blank characters. *expr* is any expression that evaluates to a character string.	`LENGTH(expr)`

Function	Example	Description	Syntax
LOCATE	`LOCATE('d' 'abcdef')`	Returns the numeric position of a character string in another character string. If the character string isn't found in the string being searched, the function returns a value of 0. *expr1* is any expression that evaluates to a character string. Identifies the string for which to search. *expr2* is any expression that evaluates to a character string. Identifies the string to be searched.	`LOCATE(expr1, expr2)`
LOCATEN	`LOCATEN('d' 'abcdef', 3)`	Like LOCATE, returns the numeric position of a character string in another character string. LOCATEN includes an integer argument that enables you to specify a starting position to begin the search. *expr1* is any expression that evaluates to a character string. Identifies the string for which to search. *expr2* is any expression that evaluates to a character string. Identifies the string to be searched. *integer* is any positive (nonzero) integer that represents the starting position to begin to look for the character string.	`LOCATEN(expr1, expr2, integer)`
LOWER	`LOWER(Customer_Name)`	Converts a character string to lowercase. *expr* is any expression that evaluates to a character string.	`LOWER(expr)`
OCTET_LENGTH	`OCTET_LENGTH('abcdef')`	Returns the number of bytes of a specified string. *expr* is any expression that evaluates to a character string.	`OCTET_LENGTH(expr)`
POSITION	`POSITION('d', 'abcdef')`	Returns the numeric position of *strExpr1* in a character expression. If *strExpr1* isn't found, the function returns 0. *expr1* is any expression that evaluates to a character string. Identifies the string to search for in the target string. *expr2* is any expression that evaluates to a character string. Identifies the target string to be searched.	`POSITION(expr1 IN expr2)`
REPEAT	`REPEAT('abc', 4)`	Repeats a specified expression *n* times. *expr* is any expression that evaluates to a character string *integer* is any positive integer that represents the number of times to repeat the character string.	`REPEAT(expr, integer)`

Function	Example	Description	Syntax
REPLACE	`REPLACE('abcd1234', '123', 'zz')`	Replaces one or more characters from a specified character expression with one or more other characters. *expr1* is any expression that evaluates to a character string. This is the string in which characters are to be replaced. *expr2* is any expression that evaluates to a character string. This second string identifies the characters from the first string that are to be replaced. *expr3* is any expression that evaluates to a character string. This third string specifies the characters to substitute into the first string.	`REPLACE(expr1, expr2, expr3)`
RIGHT	`SELECT RIGHT('123456', 3) FROM table`	Returns a specified number of characters from the right of a string. *expr* is any expression that evaluates to a character string. *integer* is any positive integer that represents the number of characters from the right of the string to return.	`RIGHT(expr, integer)`
SPACE	`SPACE(2)`	Inserts blank spaces. *integer* is any positive integer that indicates the number of spaces to insert.	`SPACE(expr)`
SUBSTRING	`SUBSTRING('abcdef' FROM 2)`	Creates a new string starting from a fixed number of characters into the original string. *expr* is any expression that evaluates to a character string. *startPos* is any positive integer that represents the number of characters from the start of the left side of the string where the result is to begin.	`SUBSTRING([SourceString] FROM [StartPostition])`
SUBSTRINGN	`SUBSTRING('abcdef' FROM 2 FOR 3)`	Like SUBSTRING, creates a new string starting from a fixed number of characters into the original string. *SUBSTRINGN* includes an integer argument that enables you to specify the length of the new string, in number of characters. *expr* is any expression that evaluates to a character string. *startPos* is any positive integer that represents the number of characters from the start of the left side of the string where the result is to begin.	`SUBSTRING(expr FROM startPos FOR length)`

Function	Example	Description	Syntax
TrimBoth	`Trim(BOTH '_' FROM '_abcdef_')`	Strips specified leading and trailing characters from a character string. *char* is any single character. If you omit this specification (and the required single quotes), a blank character is used as the default. *expr* is any expression that evaluates to a character string.	`TRIM(BOTH char FROM expr)`
TRIMLEADING	`TRIM(LEADING '_' FROM '_abcdef')`	Strips specified leading characters from a character string. *char* is any single character. If you omit this specification (and the required single quotes), a blank character is used as the default. *expr* is any expression that evaluates to a character string.	`TRIM(LEADING char FROM expr)`
TRIMTRAILING	`TRIM(TRAILING '_' FROM 'abcdef_')`	Strips specified trailing characters from a character string. *char* is any single character. If you omit this specification (and the required single quotes), a blank character is used as the default. *expr* is any expression that evaluates to a character string.	`TRIM(TRAILING char FROM expr)`
UPPER	`UPPER(Customer_Name)`	Converts a character string to uppercase. *expr* is any expression that evaluates to a character string.	`UPPER(expr)`

System Functions

The `USER` system function returns values relating to the session.

It returns the user name you signed in with.

Function	Example	Description	Syntax
DATABASE		Returns the name of the Oracle BI subject area to which you're logged on.	`DATABASE()`
USER		Returns the user name for the Oracle BI Repository to which you're logged on.	`USER()`

Time Series Functions

Time series functions are aggregate functions that operate on time dimensions.

Time dimension members must be at or below the level of the function. Because of this, one or more columns that uniquely identify members at or below the given level must be projected in the query.

Function	Example	Description	Syntax
AGO	`SELECT Year_ID, AGO(sales, year, 1)`	Calculates the aggregated value of a measure from the current time to a specified time period in the past. For example, `AGO` can produce sales for every month of the current quarter and the corresponding quarter-ago sales.	`AGO(expr, time_level, offset)`
PERIODROLLING	`SELECT Month_ID, PERIODROLLING (monthly_sales, -1, 1)`	Computes the aggregate of a measure over the period starting *x* units of time and ending *y* units of time from the current time. For example, `PERIODROLLING` can compute sales for a period that starts at a quarter before and ends at a quarter after the current quarter. *measure* is the name of a measure column. *x* is an integer that specifies the offset from the current time. *y* specifies the number of time units over which the function computes. *hierarchy* is an optional argument that specifies the name of a hierarchy in a time dimension, such as *yr*, *mon*, *day*, that you want to use to compute the time window.	`PERIODROLLING(measure, x [,y])`
TODATE	`SELECT Year_ID, Month_ID, TODATE (sales, year)`	Aggregates a measure from the beginning of a specified time period to the currently displayed time. For example, this function can calculate Year to Date sales. *expr* is an expression that references at least one measure column. *time_level* is the type of time period, such as quarter, month, or year.	`TODATE(expr, time_level)`

FORECAST Function

Creates a time-series model of the specified measure over the series using Exponential Smoothing (ETS) or Seasonal ARIMA or ARIMA, and outputs a forecast for a set of periods as specified by *numPeriods*.

Syntax `FORECAST(`*`numeric_expr`*`, ([series]), output_column_name, options, [runtime_binded_options])])`

Where:

- *numeric_expr* indicates the measure to forecast, for example, revenue data to forecast.
- *series* indicates the time grain at which the forecast model is built. This is a list of one or more time dimension columns. If you omit series, then the time grain is determined from the query.
- *output_column_name* indicates the output column. The valid values are *forecast*, *low*, *high*, and *predictionInterval*.
- *options* indicates a string list of name/value pairs separated by a semi-colon (;). The value can include *%1 ... %N* specified in `runtime_binded_options`.

- *runtime_binded_options* indicates a comma separated list of runtime-binded columns and options.

FORECAST Function Options The following table list available options to use with the `FORECAST` function.

Option Name	Values	Description
numPeriods	Integer	The number of periods to forecast
predictionInterval	0 to 100, where higher values specify higher confidence	The confidence level for the prediction.
modelType	ETS SeasonalArima ARIMA	The model to use for forecasting.
useBoxCox	TRUE FALSE	If *TRUE*, then use Box-Cox transformation.
lambdaValue	Not applicable	The Box-Cox transformation parameter. Ignore if NULL or when `useBoxCox` is *FALSE*. Otherwise the data is transformed before the model is estimated.
trendDamp	TRUE FALSE	This is a parameter for ETS model. If *TRUE*, then use damped trend. If FALSE or NULL, then use non-damped trend.
errorType	Not applicable	This is a parameter for ETS model.

trendType	none("N")
	additive("A")
	multiplicative("M")
	automatically selected("Z")
		This is a parameter for ETS model.

seasonType	none("N")
	additive("A")
	multiplicative("M")
	automatically selected("Z")
		This is a parameter for ETS model.

modelParamIC	ic_auto ic_aicc ic_bic ic_auto (this is the default)	The information criterion (IC) used in the model selection.

Revenue Forecast by Day Example

This example selects revenue forecast by day.

```
FORECAST("A - Sample Sales"."Base Facts"."1- Revenue" Target,
("A - Sample Sales"."Time"."T00 Calendar Date"),'forecast',
'numPeriods=30;predictionInterval=70;') ForecastedRevenue
```

Revenue Forecast by Year and Quarter Example

This example selects revenue forecast by year and quarter.

```
FORECAST("A - Sample Sales"."Base Facts"."1- Revenue",
("A - Sample Sales"."Time"."T01 Year" timeYear, "A - Sample Sales"."Time"."T02
Quarter" TimeQuarter),'forecast', 'numPeriods=30;predictionInterval=70;')
ForecastedRevenue
```

Constants

You can use constants to include specific dates and times in expressions.

Available constants include Date, Time, and Timestamp.

Constant	Example	Description	Syntax
DATE	`DATE [2014-04-09]`	Inserts a specific date.	`DATE [yyyy-mm-dd]`
TIME	`TIME [12:00:00]`	Inserts a specific time.	`TIME [hh:mi:ss]`
TIMESTAMP	`TIMESTAMP [2014-04-09 12:00:00]`	Inserts a specific timestamp.	`TIMESTAMP [yyyy-mm-dd hh:mi:ss]`

Types

You can use data types, such as `CHAR`, `INT`, and `NUMERIC` in expressions.

For example, you use types when creating `CAST` expressions that change the data type of an expression or a null literal to another data type.

Conditional Expressions

You use conditional expressions to create expressions that convert values.

The conditional expressions described in this section are building blocks for creating expressions that convert a value from one form to another.

Follow these rules:

- In `CASE` statements, `AND` has precedence over `OR`.
- Strings must be in single quotes.

Expression	Example	Description	Syntax

CASE (If)	`CASE`
	`WHEN score-par < 0 THEN 'Under Par'`
	`WHEN score-par = 0 THEN 'Par'`
	`WHEN score-par = 1 THEN 'Bogey'`
	`WHEN score-par = 2 THEN 'Double Bogey'`
	`ELSE 'Triple Bogey or Worse'`
	`END`
		Evaluates each `WHEN` condition and if satisfied, assigns the value in the corresponding `THEN` expression.
		If none of the `WHEN` conditions are satisfied, it assigns the default value specified in the `ELSE` expression. If no `ELSE` expression is specified, the system automatically adds an `ELSE NULL`.
		Note: See *Best Practices for using CASE statements in Analyses and Visualizations*.
			`CASE WHEN request_condition1 THEN expr1 ELSE expr2 END`

CASE (Switch)	`CASE Score-par`
	`WHEN -5 THEN 'Birdie on Par 6'`
	`WHEN -4 THEN 'Must be Tiger'`
	`WHEN -3 THEN 'Three under par'`
	`WHEN -2 THEN 'Two under par'`
	`WHEN -1 THEN 'Birdie'`
	`WHEN 0 THEN 'Par'`
	`WHEN 1 THEN 'Bogey'`
	`WHEN 2 THEN 'Double Bogey'`
	`ELSE 'Triple Bogey or Worse'`
	`END`
		Also referred to as `CASE (Lookup)`. The value of the first expression is examined, then the `WHEN` expressions. If the first expression matches any `WHEN` expression, it assigns the value in the corresponding `THEN` expression.
		If none of the `WHEN` expressions match, it assigns the default value specified in the `ELSE` expression. If no `ELSE` expression is specified, the system automatically adds an `ELSE NULL`.
		If the first expression matches an expression in multiple `WHEN` clauses, only the expression following the first match is assigned.
		Note See *Best Practices for using CASE statements in Analyses and Visualizations*.
			`CASE expr1 WHEN expr2 THEN expr3 ELSE expr4 END`

IfCase > ELSE	-	-	`ELSE [expr]`
IfCase > IFNULL	-	-	`IFNULL([expr], [value])`
IfCase > NULLIF	-	-	`NULLIF([expr], [expr])`
IfCase > WHEN	-	-	`WHEN [Condition] THEN [expr]`
IfCase > CASE	-	-	`CASE WHEN [Condition] THEN [expr] END`

Expression	Example	Description	Syntax
SwitchCase > ELSE	-	-	`ELSE [expr]`
SwitchCase >IFNULL	-	-	`IFNULL([expr], [value])`
SwitchCase > NULLIF	-	-	`NULLIF([expr], [expr])`
SwitchCase > WHEN	-	-	`WHEN [Condition] THEN [expr]`

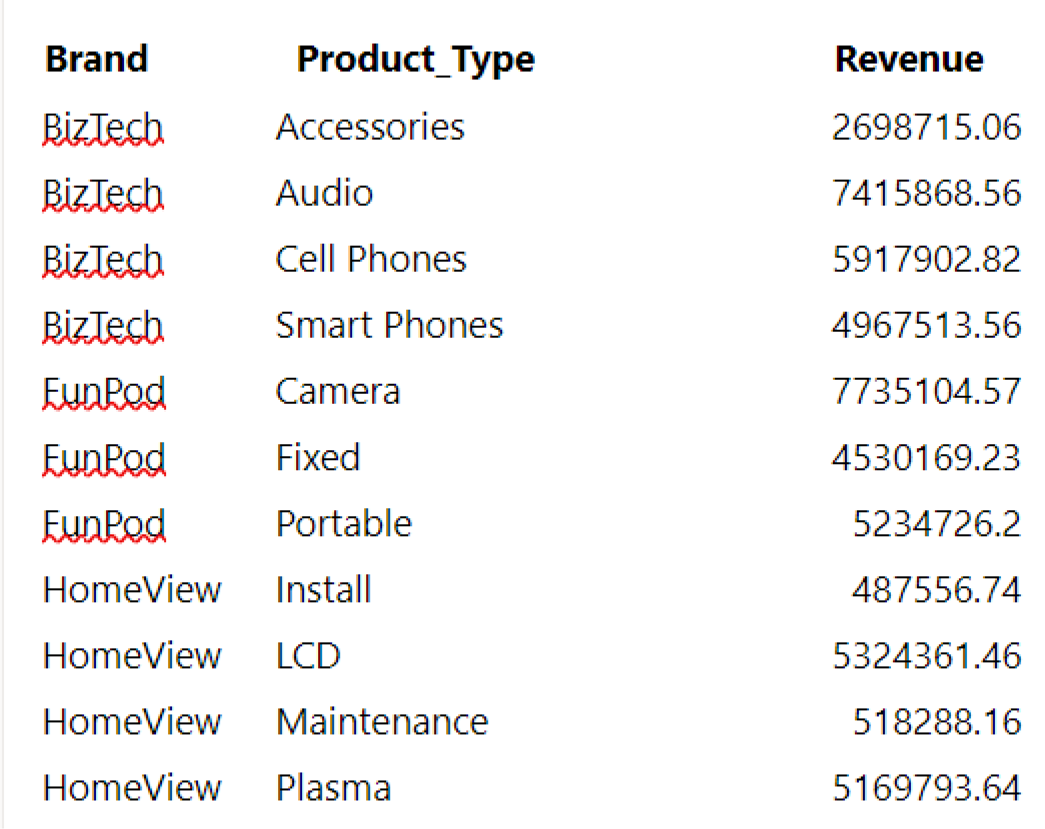
Best Practices For Using CASE statements in Analyses and Visualizations

When using CASE statements in reports and workbooks, you should consider the report columns and the order of aggregation, as these affect how expressions and sub-totals are calculated.

- In general, when using CASE expressions, make sure that any columns used in the expression are included in the report.
- If the order of aggregation is important, then change the report aggregation rule from Default to Sum.
- If a condition has to be evaluated before the aggregation is computed (for example, if the column filtered is not also displayed in the report), then use the FILTER function.

Example

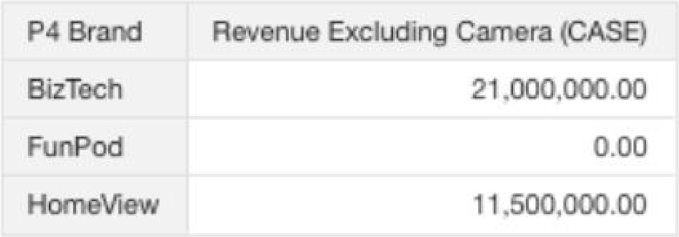
In this example, we have data for Brand, Product Type, and Revenue.



Brand	Product_Type	Revenue
BizTech	Accessories	2698715.06
BizTech	Audio	7415868.56
BizTech	Cell Phones	5917902.82
BizTech	Smart Phones	4967513.56
FunPod	Camera	7735104.57
FunPod	Fixed	4530169.23
FunPod	Portable	5234726.2
HomeView	Install	487556.74
HomeView	LCD	5324361.46
HomeView	Maintenance	518288.16
HomeView	Plasma	5169793.64

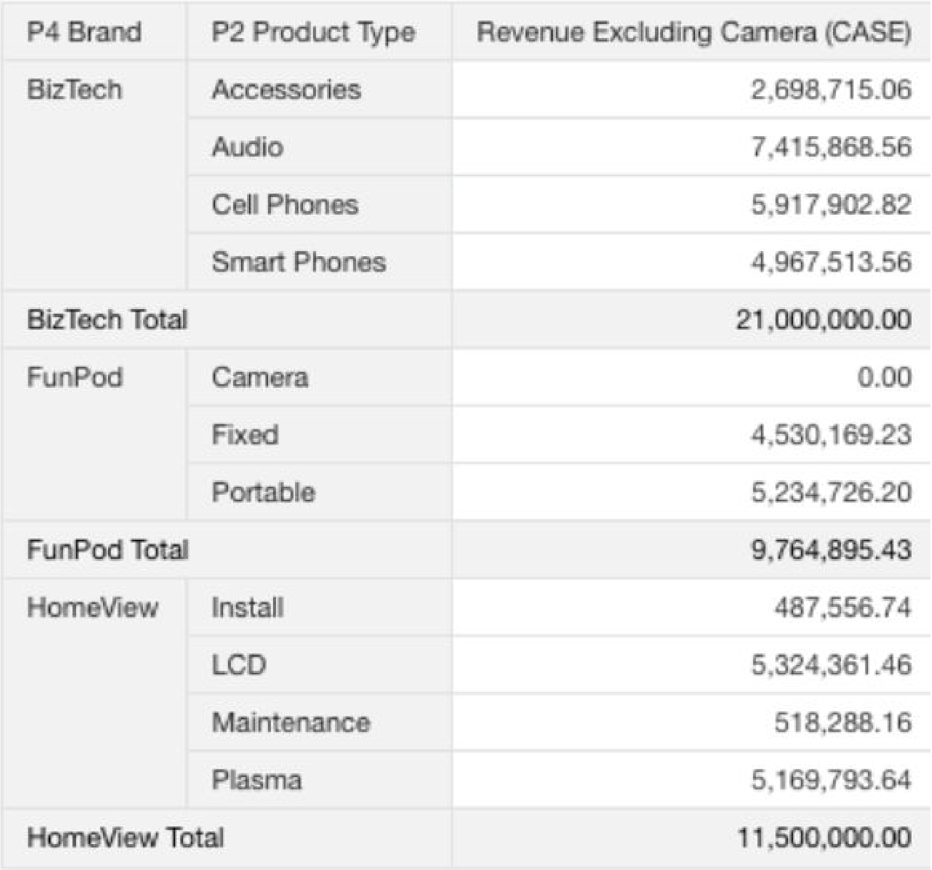
To apply a condition to set Camera revenue to 0, we create the following conditional expression: `CASE WHEN Product_Type = ‘Camera’ THEN 0 ELSE Revenue END`.

When Product_Type is excluded from the report, data is not available to completely evaluate the expression because it contains Product_Type, and the results are:



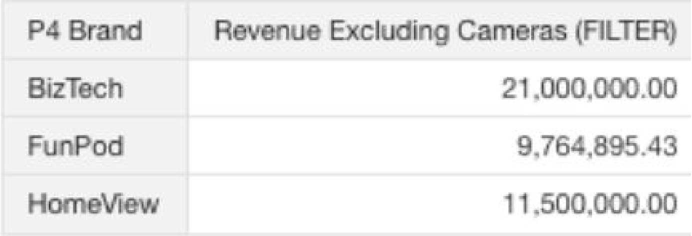
P4 Brand	Revenue Excluding Camera (CASE)
BizTech	21,000,000.00
FunPod	0.00
HomeView	11,500,000.00

By adding PRODUCT_TYPE to the report and setting report aggregation to SUM, the result aggregates after the base level values are calculated:



P4 Brand	P2 Product Type	Revenue Excluding Camera (CASE)
BizTech	Accessories	2,698,715.06
	Audio	7,415,868.56
	Cell Phones	5,917,902.82
	Smart Phones	4,967,513.56
BizTech Total		21,000,000.00
FunPod	Camera	0.00
	Fixed	4,530,169.23
	Portable	5,234,726.20
FunPod Total		9,764,895.43
HomeView	Install	487,556.74
	LCD	5,324,361.46
	Maintenance	518,288.16
	Plasma	5,169,793.64
HomeView Total		11,500,000.00

An alternative to using CASE is to use a filter expression: FILTER(Revenue using Product_Type != ‘Camera’). The expression is calculated independently of report columns, and aggregations are applied after calculation:



P4 Brand	Revenue Excluding Cameras (FILTER)
BizTech	21,000,000.00
FunPod	9,764,895.43
HomeView	11,500,000.00

G

Oracle Analytics Desktop SDK Reference

This topic describes the software development kit (SDK) that you can use to develop and deploy visualization plug-ins to your Oracle Analytics Desktop installation.

Topics:

Oracle Analytics Desktop SDK

The SDK provides a development environment where you can create and develop custom visualization plug-ins and deploy them to your Oracle Analytics Desktop installation.

Topics:

Scripts

Your installation includes the scripts that you use to create a development environment and create skeleton visualization plug-ins.

The scripts are located in this directory: `<your_installation_directory>\Oracle Analytics Desktop\tools\bin`.

For example, `C:\Program Files\Oracle Analytics Desktop\tools\bin`
Note the following script names and descriptions:

- **bicreateenv** - Run this script to create the development environment where you develop your plug-ins.
- **bicreateplugin** - Run this script to create a skeleton visualization to quickly get started on developing your custom plug-in.
- **bideleteplugin** - Run this script to delete a plug-in from your development environment.

- **bivalidate** - Run this script with the `gradlew validate` command to call the bivalidate script. The bivalidate script validates whether the JSON configuration files are properly formatted and contain appropriate visualization configuration.

Other Resources

You can use resources other than scripts to develop your custom visualization plug-ins.

Note the following available resources:

- **circlePack sample** - The circlePack sample is included in your development environment. You can deploy and use this sample immediately. However, the sample is designed for you to use with the provided tutorial to learn how to develop a visualization plug-in. You can also copy the sample and use it as a template for the visualization plug-ins that you want to create.

  The circlePack sample is located in `<your_development_directory>\src\sampleviz\sample-circlepack`

  For example, `C:\OracleDVDev\src\sampleviz\sample-circlepack`
- **Other visualization plug-in samples** - You can download plug-in examples from the Oracle Analytics Library.
- **Tutorial** - The tutorial contains information and instructions to help you understand how to create a robust visualization plug-in. This tutorial provides step-by-step instructions for modifying the circlePack sample included in your plug-in development environment.

  Tutorial
- **JS API documentation** - This documentation contains JavaScript reference information that you need to develop a visualization plug-in. See Oracle Analytics Desktop SDK JavaScript Reference.

Create the Visualization Plug-in Development Environment

You need to set the PATH environment variable and create the development environment before you can create visualization plug-ins.

1. Using the command prompt, create an empty development directory. For example, `C:\OracleAnalyticsDev`.
2. Set the PATH environment variable. For example,

```
set ANALYTICSDESKTOP_SDK_HOME="C:\Program Files\Oracle Analytics
Desktop"
set PLUGIN_DEV_DIR=C:\OracleAnalyticsDev
REM add tools\bin to path:
set PATH=%ANALYTICSDESKTOP_SDK_HOME%\tools\bin;%PATH%
```

3. Run the **bicreateenv** script included in your installation to create the development environment in the empty directory. For example,

```
cd C:\OracleAnalyticsDev
bicreateenv
```

For information about the options available for running this script, see the script's command-line help. For example,

```
C:\OracleAnalyticsDev>bicreateenv -help
```

The complete development environment, including build.gradle and gradlew, is created in the directory that you specified.

4. Optional: If you're working behind a web proxy, then you need to set gradle.properties to point to your proxy. The gradle.properties are located in your development environment, for example `C:\OracleAnalyticsDev\gradle.properties`.

   Use the following example to set your gradle.properties:

   ```
   systemProp.https.proxyHost=www-proxy.somecompany.com
   systemProp.https.proxyPort=80
   systemProp.https.nonProxyHosts=*.somecompany.com|*.companyaltname.com
   ```

Create a Skeleton Visualization Plug-in

After you create a skeleton visualization plug-in in your development environment, you then develop it into a robust visualization plug-in and deploy it to your Oracle Analytics Desktop environment.

1. Run the **bicreateplugin** script included in your installation to create a skeleton visualization. Use the following syntax:

   ```
   bicreateplugin viz -<subType> -<id> -<name>
   ```

   - `<subType>` is the type of visualization that you want to create. Your choices are:
     - **basic** - Use this option to create a visualization that doesn't use any data from Oracle Analytics Desktop or use any data model mapping. This is like the Image and Text visualization types delivered with Oracle Analytics Desktop. For example, you can use this visualization type to show an image or some text that's coded into the plug-in or from a configuration. You can use this type of visualization to improve formatting.
     - **dataviz** -This type renders data from data sources registered with Oracle Analytics Desktop into a chart or table or some other representation on the screen. It also respond to marking events from other visualizations on the same canvas and publish interaction events to affect other visualizations on the same canvas.
     - **embeddableDataviz** - This type renders data from data sources registered with Oracle Analytics Desktop into the cells of a trellis visualization. It also responds to marking events from other visualizations on the same canvas and publish interaction events to affect other visualizations on the same canvas.
   - `<id>` is your domain and the name that you want to give the visualization directory and components in your development environment. For example, `com-company.basicviz`.
   - `<name>` is the name of the visualization plug-in that you test, deploy, and use in workbooks.

   For example to create a basic visualization, name its development directory com-company-basicviz, and name the visualization plug-in helloViz, enter and run the following command:

```
C:\OracleDevDir>bicreateplugin viz –subType basic –id
com.company.basicviz —name helloViz
```

2. Optional: Open the script's command-line help for information about the options available for running this script. For example, `C:\OracleDVDev> bicreateplugin -help`

When you run the `bicreateplugin -viz` command for the first time, the system creates the customviz directory in the following location.
`<your_development_environment>\src\customviz`
All custom visualization development directories that you create are added to this directory.

For example, `C:\OracleDVDev\src\customviz\com-company-basicviz`

Create a Skeleton Skin or Unclassified Plug-in

The `bicreateplugin -unclassified` command creates an empty plug-in with plugin.xml, localization bundles. The `bicreateplugin -skin` command creates a skeleton skin plug-in.

- Run the **createplugin** script included in your installation to create a skeleton plug-in. Use one of the following syntaxes:

  ```
  bicreateplugin -skin -<id>
  ```

  ```
  bicreateplugin -unclassified -<id>
  ```

  - `<id>` is your domain and the name that you want to give the visualization. For example, `com-company.newskin`

  For example, to create a skin plug-in, enter and run the following command:
  `C:\OracleDevDir>bicreateplugin skin –id com.company.newskin`

Develop a Visualization Plug-in

After you create the skeleton visualization plug-in, you can use resources provided by Oracle to help you develop your plug-in.

The directories for dataviz and embeddableDataviz types include the datamodelhandler.js file, which contains the physical-to-logical data mapping format. This file also determines how Oracle Analytics Desktop renders and passes user interactions to the server.

- Use the tutorial to learn how to perform development tasks such as implement data mapping.

  Tutorial

- Use the .JS API documentation to learn how to add dependencies. See Oracle Analytics Desktop SDK JavaScript Reference.

Run in SDK Mode and Test the Plug-in

You can run Oracle Analytics Desktop in SDK mode from your browser when you’re developing your visualization plug-in or when you want to test your visualization plug-in.

1. Execute the `gradlew run` command. For example, `C:\OracleDevDir>gradlew run`

   After you run the command, note the following results:

   - Oracle Analytics Desktop opens in SDK mode in your default browser. Use the browser's JavaScript debugger to test and debug the application.
   - The visualization that you created is available in the Visualizations pane of Oracle Analytics Desktop.
   - A system tray is displayed in the operating system's toolbar and includes three links: Launch Browser, which you use to launch or relaunch your default browser to display Oracle Analytics Desktop; Copy URL to Clipboard, which you can use to copy the URL and paste it into a different browser; and Shutdown, which you use to shut down the development browser.

2. Test your visualization by dragging and dropping it to a workbook’s canvas and adding data elements.
3. If necessary, continue developing the visualization plug-in. When working in SDK mode in the browser, you can update the .JS definition and refresh the browser to see your changes.

Validate the Visualization Plug-in

After you’ve tested your visualization plug-in and before you can package and deploy it, you must validate it.

1. Run the `gradlew validate` command. For example,

   ```
   cd C:\OracleDVDev
   .\gradlew validate
   ```

   This step validates whether the JSON configuration files are properly formatted and contain appropriate visualization configuration. If the validation discovers any errors, then the system displays error messages.

2. To check for errors in the JavaScript source files, use your browser’s development tools.

Build, Package, and Deploy the Visualization Plug-in

After you validate the visualization plug-in, you've to build and package it, and then copy the resulting distributions into your installation directory.

The build and package process runs for all of the visualizations in your development directory, and each plug-in is contained in its own zip file. There’s no way to build and package specific visualizations. If you want to exclude visualizations from the build and package process, then you've to move the visualizations that you want to exclude out of your development directory, or delete them from the directory before you perform the build. See Delete Plug-ins from the Development Environment.

1. Run the `gradlew build` command. For example,

   ```
   cd C:\OracleDVDev
   .\gradlew clean build
   ```

A build directory is added to your development environment. For example, `C:\OracleDVDev\build\distributions`. This directory contains a zip file for each visualization. The zip file's name is the one that you gave the visualization when you created its skeleton. For example, basicviz.zip.

2. Copy the zip files to your installation directory. For example, `%localappdata%\OracleAnalyticsDesktop\plugins`.

Delete Plug-ins from the Development Environment

You can use the `bideleteplugin` script to delete the unneeded plug-ins from your development environment.

The build and package process includes all of the visualizations contained in your development directory. There is no way to build and package specific visualizations. To exclude any unwanted visualizations from the build, you can delete them before you perform the build and package process.

1. If you want to delete a visualization plug-in, then run the `bideleteplugin` command, using the following syntax:

   ```
   cd C:\<your_development_directory>
   bideleteplugin viz -id <name_of_your_domain>.<name_of_viz_plugin>
   ```

2. If you want to delete an unclassified plug-in, then run the `bideleteplugin` command, using the following syntax:

   ```
   cd C:\<your_development_directory>
   bideleteplugin unclassified -id
   <name_of_your_domain>.<name_of_unclassified_plugin>
   ```

3. If you want to delete a skin plug-in, then run the `bideleteplugin` command, using the following syntax:

   ```
   cd C:\<your_development_directory>
   bideleteplugin skin -id <name_of_your_domain>.<name_of_skin_plugin>
   ```