POWER BI ASSIGNMENT 6

1. **Explain DAX.**

Data Analysis Expressions (DAX) is a formula expression language used in Analysis Services, Power BI, and Power Pivot in Excel. DAX formulas include functions, operators, and values to perform advanced calculations and queries on data in related tables and columns in tabular data models.

Measures

Measures are dynamic calculation formulas where the results change depending on context. Measures are used in reporting that support combining and filtering model data by using multiple attributes such as a Power BI report or Excel PivotTable or PivotChart. Measures are created by using the DAX formula bar in the model designer.

A formula in a measure can use standard aggregation functions automatically created by using the Autosum feature, such as COUNT or SUM, or you can define your own formula by using the DAX formula bar. Named measures can be passed as an argument to other measures.

Unlike calculated columns, the syntax for a measure includes the measure's name preceding the formula. After you've created a measure, the name and its definition appear in the reporting client application Fields list, and depending on perspectives and roles is available to all users of the model.

Calculated columns

A calculated column is a column that you add to an existing table (in the model designer) and then create a DAX formula that defines the column's values. When a calculated column contains a valid DAX formula, values are calculated for each row as soon as the formula is entered. Values are then stored in the in-memory data model. For example, in a Date table, when the formula is entered into the formula bar:

```
= [Calendar Year] & " Q" & [Calendar Quarter]
```

A value for each row in the table is calculated by taking values from the Calendar Year column (in the same Date table), adding a space and the capital letter Q, and then adding the values from the Calendar Quarter column (in the same Date table). The result for each row in the calculated column is calculated immediately and appears, for example, as 2017 Q1. Column values are only recalculated if the table or any related table is processed (refresh) or the model is unloaded from memory and then reloaded, like when closing and reopening a Power BI Desktop file.

Calculated tables

A calculated table is a computed object, based on a formula expression, derived from all or part of other tables in the same model. Instead of querying and loading values into your new table's columns from a data source, a DAX formula defines the table's values.

Calculated tables can be helpful in a role-playing dimension. An example is the Date table, as OrderDate, ShipDate, or DueDate, depending on the foreign key relationship. By creating a calculated table for ShipDate explicitly, you get a standalone table that is available for queries, as fully operable as any other table. Calculated tables are also useful when configuring a filtered rowset, or a subset or superset of columns from other existing tables. This allows you to keep the original table intact while creating variations of that table to support specific scenarios.

Calculated tables support relationships with other tables. The columns in your calculated table have data types, formatting, and can belong to a data category. Calculated tables can be named, and

surfaced or hidden just like any other table. Calculated tables are re-calculated if any of the tables it pulls data from are refreshed or updated.

Row-level security

With row-level security, a DAX formula must evaluate to a Boolean TRUE/FALSE condition, defining which rows can be returned by the results of a query by members of a particular role. When defining row-level security by using DAX formula, you are creating an allowed row set. This does not deny access to other rows; rather, they are simply not returned as part of the allowed row set. Other roles can allow access to the rows excluded by the DAX formula. If a user is a member of another role, and that role's row-level security allows access to that particular row set, the user can view data for that row.

Row-level security formulas apply to the specified rows as well as related rows. When a table has multiple relationships, filters apply security for the relationship that is active. Row-level security formulas will be intersected with other formulas defined for related tables.

Queries

DAX queries can be created and run in SQL Server Management Studio (SSMS) and open-source tools like DAX Studio (daxstudio.org). Unlike DAX calculation formulas, which can only be created in tabular data models, DAX queries can also be run against Analysis Services Multidimensional models. DAX queries are often easier to write and more efficient than Multidimensional Data Expressions (MDX) queries.

A DAX query is a statement, similar to a SELECT statement in T-SQL. The most basic type of DAX query is an evaluate statement.

Formulas

DAX formulas are essential for creating calculations in calculated columns and measures, and securing your data by using row-level security. To create formulas for calculated columns and measures, use the formula bar along the top of the model designer window or the DAX Editor. To create formulas for row-level security, use the Role Manager or Manage roles dialog box.

Functions

A function is a named formula within an expression. Most functions have required and optional arguments, also known as parameters, as input. When the function is executed, a value is returned. DAX includes functions you can use to perform calculations using dates and times, create conditional values, work with strings, perform lookups based on relationships, and the ability to iterate over a table to perform recursive calculations. If you are familiar with Excel formulas, many of these functions will appear very similar; however, DAX formulas are different in the following important ways:

A DAX function always references a complete column or a table. If you want to use only particular values from a table or column, you can add filters to the formula.

If you need to customize calculations on a row-by-row basis, DAX provides functions that let you use the current row value or a related value as a kind of parameter, to perform calculations that vary by context. To understand how these functions work, see Context in this article.

DAX includes many functions that return a table, rather than a value. The table is not displayed in a reporting client, but is used to provide input to other functions. For example, you can retrieve a table and then count the distinct values in it, or calculate dynamic sums across filtered tables or columns.

Variables

You can create variables within an expression by using VAR. VAR is technically not a function, it's a keyword to store the result of an expression as a named variable. That variable can then be passed as an argument to other measure expressions.

Data types

You can import data into a model from many different data sources that might support different data types. When you import data into a model, the data is converted to one of the tabular model data types. When the model data is used in a calculation, the data is then converted to a DAX data type for the duration and output of the calculation. When you create a DAX formula, the terms used in the formula will automatically determine the value data type returned.

Context

Context is an important concept to understand when creating DAX formulas. Context is what enables you to perform dynamic analysis, as the results of a formula change to reflect the current row or cell selection and also any related data. Understanding context and using context effectively are critical for building high-performing, dynamic analyses, and for troubleshooting problems in formulas.

Formulas in tabular models can be evaluated in a different context, depending on other design elements:

- Filters applied in a PivotTable or report
- Filters defined within a formula
- Relationships specified by using special functions within a formula

There are different types of context: row context, query context, and filter context.

Operators

The DAX language uses four different types of calculation operators in formulas:

- Comparison operators to compare values and return a logical TRUE\FALSE value.
- Arithmetic operators to perform arithmetic calculations that return numeric values.
- Text concatenation operators to join two or more text strings.
- Logical operators that combine two or more expressions to return a single result.

2. Explain datasets, reports, and dashboards and how they relate to each other.

A Power BI dashboard is made up to visualize data summarizing. In contrast, a report is a detailed presentation of data. Therefore, the Power BI dashboard is easier to understand as a report can contain complex information that some users might not understand.

The key difference is that the Power BI dashboard is a collection of visuals to tell the story graphically, like charts and graphs, along with features to interact with the end-user. In contrast, a report is generally a detailed summary of the large data set as per the criteria given by the user.

The key differences between these two are as follows:

- Information Level: Reports are created on multiple pages, so every kind of detailed analysis and information is available with “Reports.” We drill through the reports.
  Dashboards include only important information on the large data set, which is critical for quick decision-making.

- Interactivity: Reports are embedded with slicers and filters, so if the summary table shows only monthly sales, then by adding the category field to the slicers, we can select each category individually and see how each category performs across months.
  Dashboards may not have this interactivity. We may see monthly and category-wise sales values in different tables or visuals. Users need to look at two different tables and find the differences.

Items	Dashboard	Report
Data Source	Dashboards are built based on multiple data tables connected in one or more ways.	Reports are generally created from a single data set table with no relationship to other tables.
No. of Pages	Dashboards are not allowed to cross more than one page. Hence, it is because it always shows the important reports on a single page.	Reports are generally built-in multiple pages.
Visualizations	Dashboards always concentrate on building insights into the data by using attractive visuals, graphs, charts, etc.	Reports are not concentrated on the visualization part of the data. Rather, it looks to create summary pages.
Template	Dashboards do not have any set template. It is up to the creator to visualize the data to fit the needs of the business.	Reports generally have a set template. According to the additional deletion of the data, the template will create reports if the formulas are applied from the data table.
Slicers and Filters	Since dashboards are limited to a single page, we cannot use filters and slicers.	In reports, we can filter and slice the data using slicers and many filtering options like cross-filtering, visual level filtering and page-level filtering.
Kind of Information	Dashboards may include limited information, which is only important to the end-users.	Reports are not limited to a single page so that it can have a detailed break up of each report category on multiple pages.

Items	Dashboard	Report
Reader Interactivity	Dashboards are pinned to the page so the reader can just read through the data.	Reports are created with filters and slicers so the user can interact with the data set.
Changes to Visuals	Dashboards are pinned to the page. Even if the report owner changes, it will not reflect on the page.	Reports usually come along with the data set, so if the reader wishes to change the visual type, they can change it at any time.
Alerts	Dashboards can create alerts to email when a specific condition or criteria is met, or a limit is crossed.	Reports cannot create alerts to email when a specific condition or criteria is met or a limit is crossed.
Data Set View	We cannot see the source data with dashboards because the reader only gets the single-page information.	Reports can see tables, data sets, and data fields in detail, i.e., raw data.

3. **How reports can be created in power BI, explain two ways with Navigation of each?**

You can think of any report as a dashboard. A widget is some kind of visual display: a chart, a table, or just a single metric displayed in a text box.

The basic procedure is to repeat this process for each widget:

1. Create a data source
2. Run a transformation (Optional)
3. Create a query or data model
4. Pick a visualization
5. Select fields
6. Arrange visualization on dashboard

How to create Power BI reports

To illustrate, let's move through each of these steps. First, we create a data source. That connects to a file, like a .csv, or a database.

Next, you have the option to run a **transformation**. In our example, we use financial data. We will:

- Apply a filter to select only negative values (payments)
- Drop and rename columns
- Optionally apply a function, such as an aggregation

Step 3 is the natural result of step 2, because you have built up a query in stages.

Alternatively, at this point, you could create a data model. For example, if you have sales and inventory movements in two data sources you can model that. You would create a model to show the common element between tables: product number. (But in the example we're using, we only have a single data source.)

In step 4, you create visualization. In this example, we will have a table of transactions. A table is a row and column display. We will also have a single **card** (like a text box) to show a single number, the maximum transaction amount.

Next, we'll pick fields from step 4. Finally, in the last step, we'll position the visualizations on the dashboard.

Now, let's walk through an actual example.

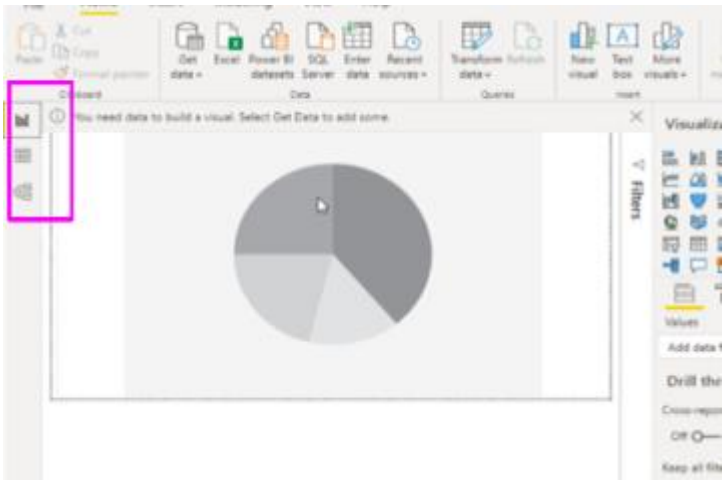
A hands-on tutorial

Here is the landing page for BI. By default, it shows a pie chart with no data. Notice the three icons on the left:

- Dashboard
- Queries
- Data model

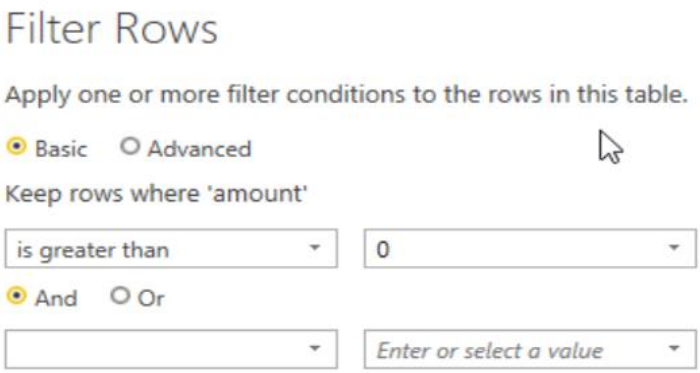
Adding the first visualization:-

The logical place to start is to select a **data source**.

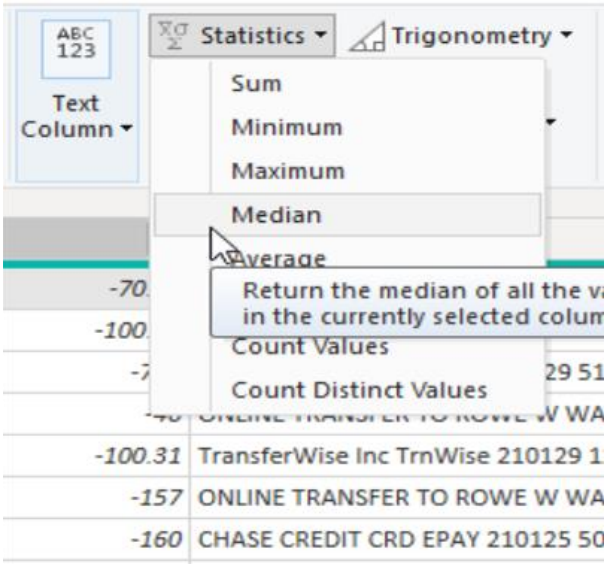


The basic procedure is to load and optionally select **transform**. In most cases, you would want to do a transformation.

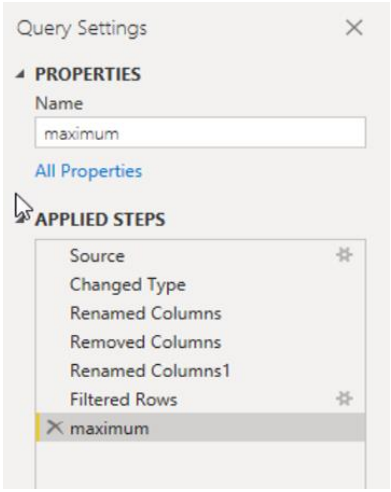
For example, let's click a column, and then apply a filter to only have negative values (payments):



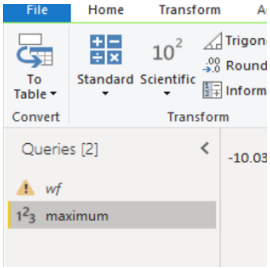
Here we select a numeric column, **amount**. Because it's a number, we can run a math or aggregation function on it.
We select the **Statistics** function **Maximum**:



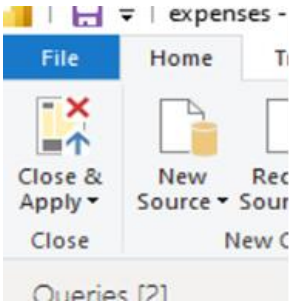
The result is a scalar (single value), as opposed to a row in a row-column table.
Now click on the new field and give it a meaningful name. Notice that BI keeps track of the steps we have taken.



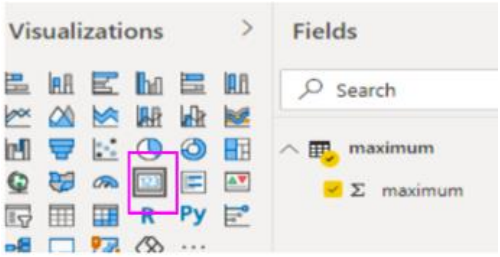
You'll also want to rename the query. At this point, BI calls the results of the transformation a **query**.



Click the **close & apply** button to close the Power BI editor and return to the dashboard view.



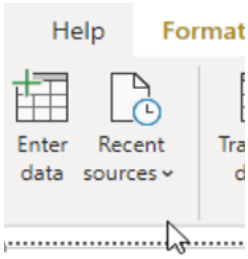
Select the **card** visualization, then select field **maximum** from the query **maximum**.



The card is added to the report:

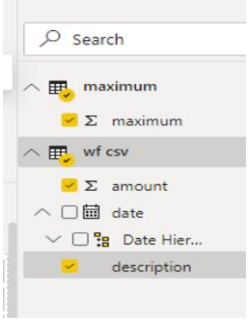
Adding more visualization:-

Now we can add another visualization to show how to build up your report. We will make a table. Select **recent sources** and pick the same .csv file. Importantly, we have to go all the way back to the beginning data source because we turned the first source into a query. (We can't use the query to make a table, since it's already transformed into a scalar.)

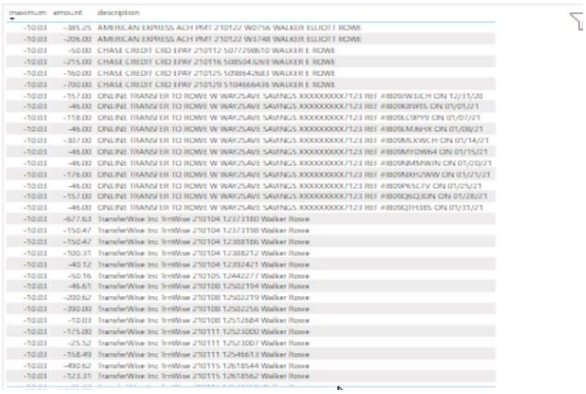


Now we have two queries:

- **wf** is a table
- **maximum** is the data source or the card visualization



Here's what the table looks like when attached to the dashboard:

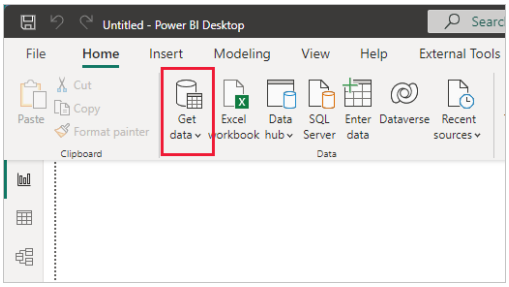


The text looks annoyingly small and graphic-like. It's not like a spreadsheet, which would be clear and easy to read. (We will show how to clean that up in an upcoming tutorial.)
Finally, move the card over to make room for the table. Select the corner so you can resize it.

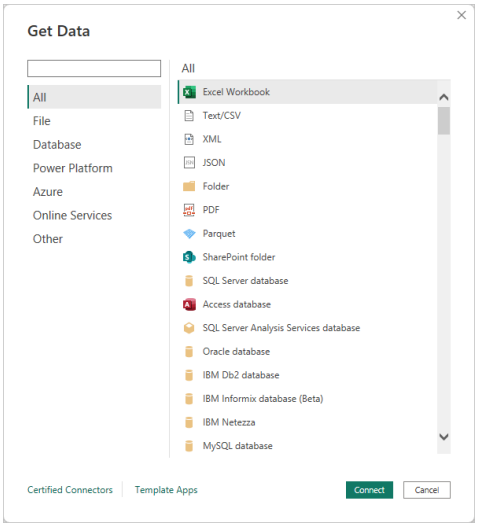
4. **How to connect to data in Power BI? How to use the content pack to connect to google analytics? Mention the steps.**

Connect to data in Power BI

With Power BI Desktop, you can connect to many different types of data. These sources include basic data sources, such as a Microsoft Excel file. You can connect to online services that contain all sorts of data, such as Salesforce, Microsoft Dynamics, Azure Blob Storage, and many more.
To connect to data, from the Home ribbon select Get data.

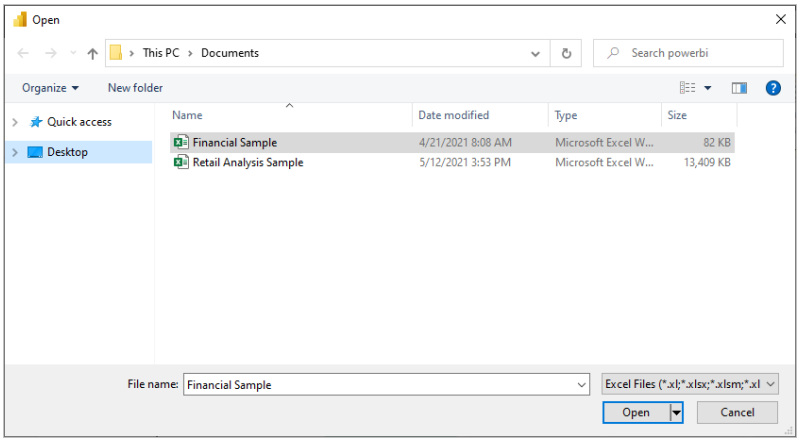


The Get Data window appears. You can choose from the many different data sources to which Power BI Desktop can connect. In this quick start, use the Excel workbook that you downloaded in Prerequisites.

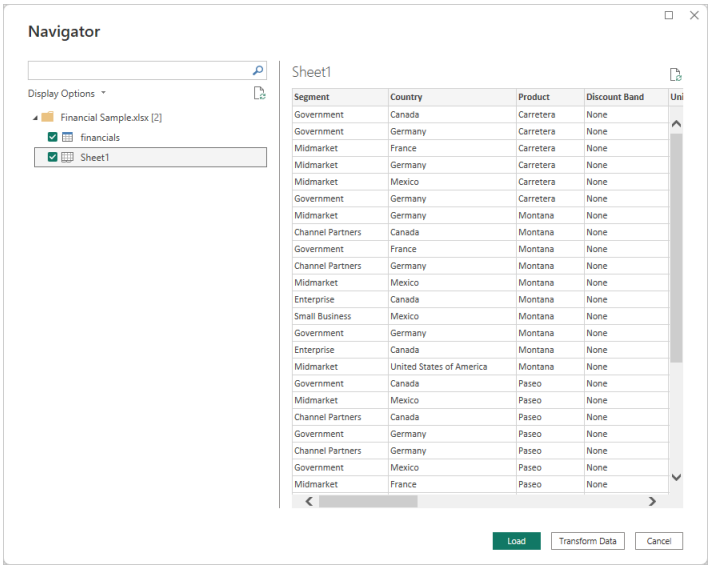


Since this data source is an Excel file, select Excel from the Get Data window, then select the Connect button.

Power BI prompts you to provide the location of the Excel file to which to connect. The downloaded file is called *Financial Sample*. Select that file, and then select Open.



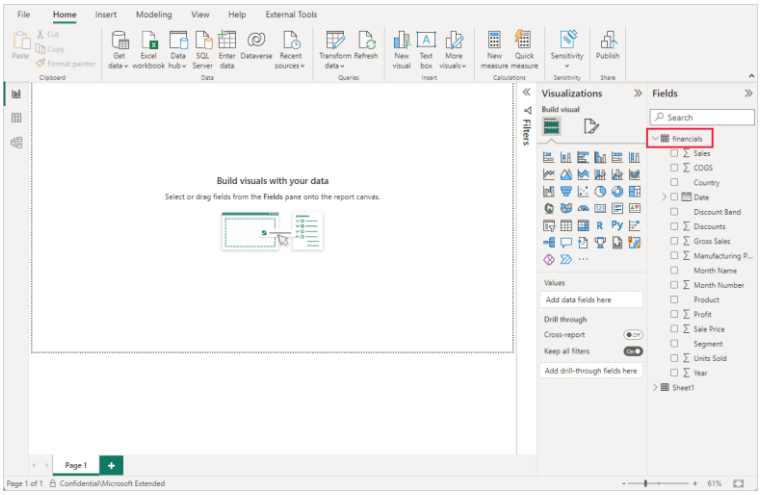
Power BI Desktop then loads the workbook and reads its contents, and shows you the available data in the file using the Navigator window. In that window, you can choose which data you would like to load into Power BI Desktop. Select the tables by marking the checkboxes beside each table you want to import. Import both available tables.



Once you've made your selections, select Load to import the data into Power BI Desktop.

View data in the Fields pane

Once you've loaded the tables, the Fields pane shows you the data. You can expand each table by selecting the arrow beside its name. In the following image, the *financials* table is expanded, showing each of its fields.



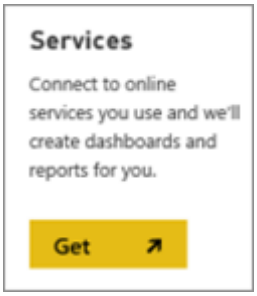
And that's it! You've connected to data in Power BI Desktop, loaded that data, and now you can see all the available fields within those tables.

Connecting Power BI with Google Analytics

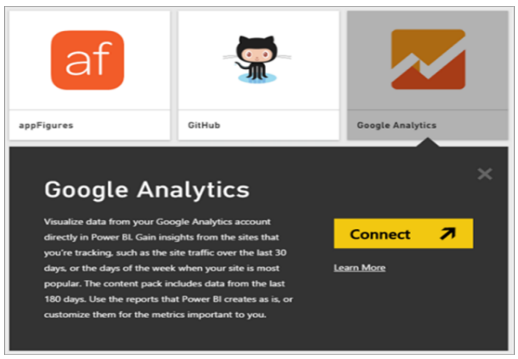
In Power BI, it's straightforward to connect to the Google Analytics content pack.

1. In the left navigation pane, click Get Data.

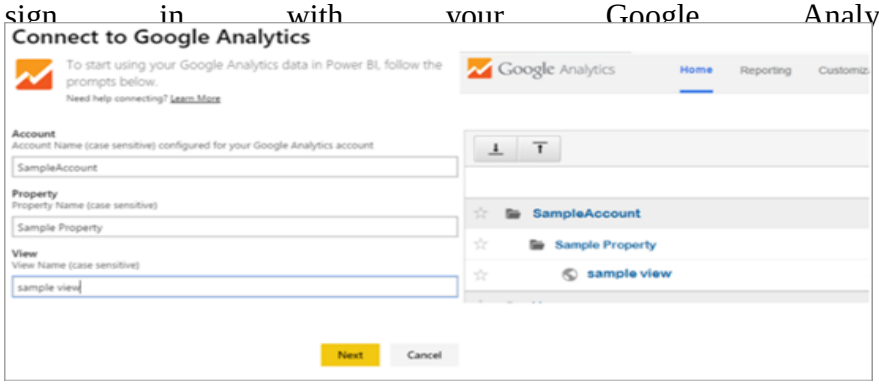
2. In the Services box, click Get.



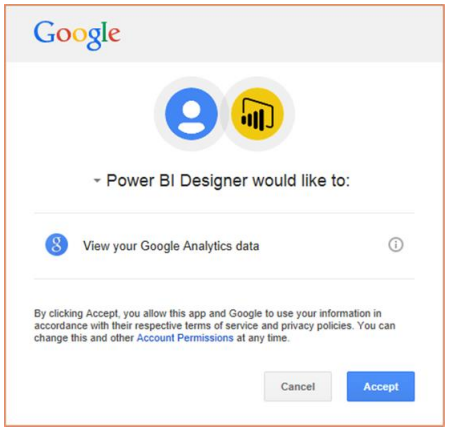
3. From the menu of online services, select Google Analytics, and then click Connect.



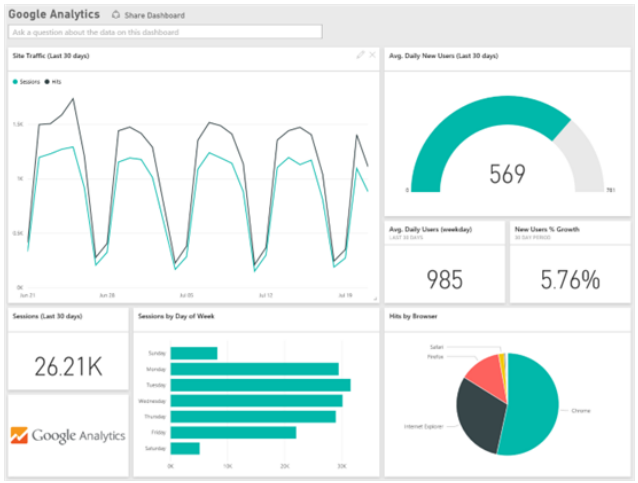
4. Enter the Google Analytics account, property, and view that you want to connect to. Then sign in with your Google Analytics credentials.



5. To permit Power BI to connect to Google Analytics, click Accept.



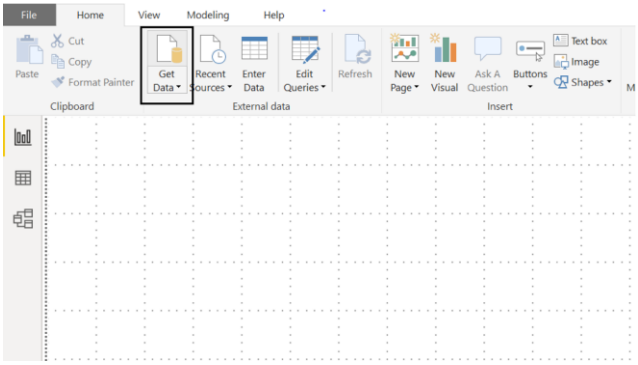
6. When the import process completes, you will see a new dashboard, report, and model in the Navigation Pane. Select the dashboard to view your imported data.



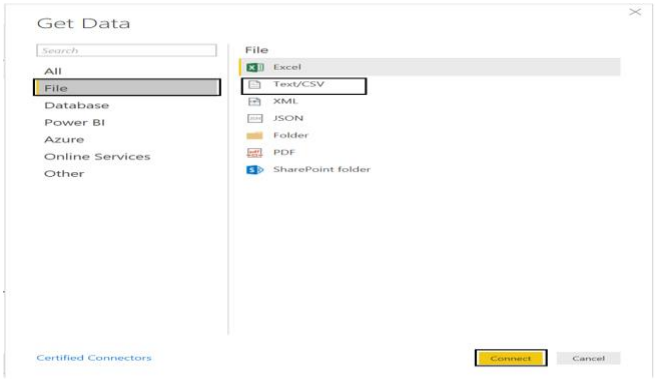
5. **How to import Local files in Power BI? Mention the Steps.**

Power BI Desktop has a Get Data button from the ribbon on the Home tab. In Power BI, there are all sorts of different data sources available. Select a source to establish a connection. Depending on your selection, you will be asked to find the source on your computer or network, or be prompted to sign in to a service to authenticate your request.

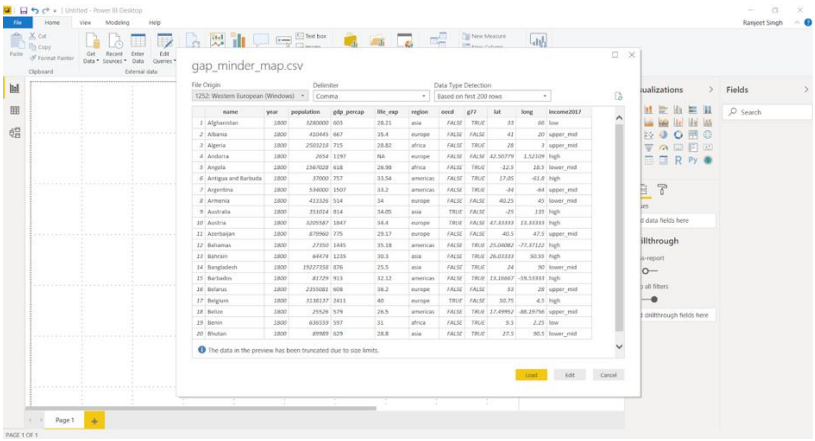
As our first step to import the dataset/file into Power BI, we click on the Get Data icon on the ribbon of Home tab.

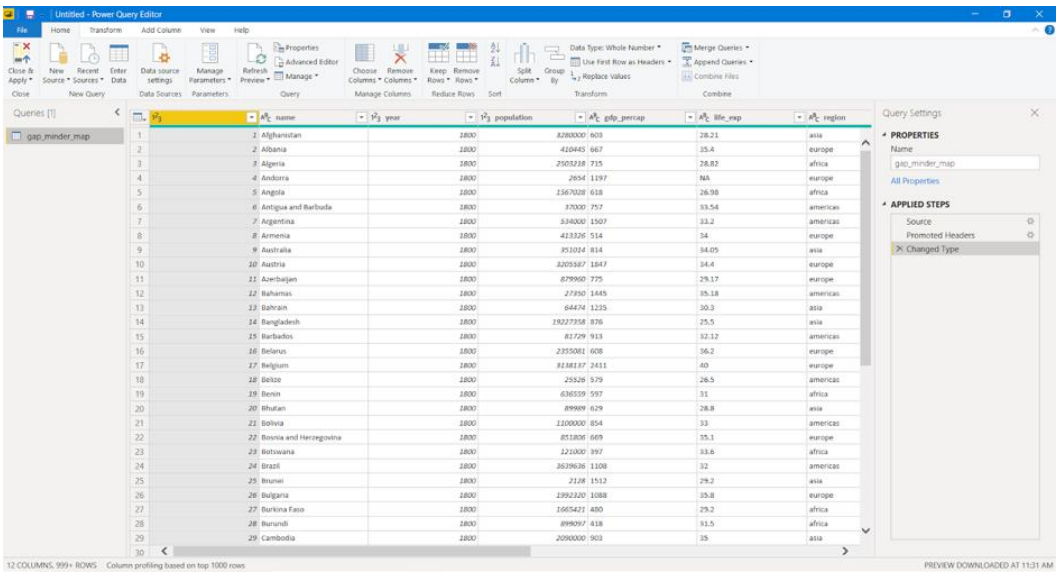


Once we select this, we go ahead and select the CSV option under the file subheading.



Then browse the file and select the necessary file type. Press on Connect to have a quick preview of the file. Once we click on Load, Power BI will successfully import the file. Any errors will then pop up ready to be analysed and fixed.





If there are any errors in the data load, select View errors to check any detected errors. Alternatively, close the pop up and click Edit queries to check for any errors. This will query and list the errors in the data.

6. **In Power BI visualization, what are Reading View and editing view?**

You can create and edit reports in both the Power BI service and Power BI Desktop. In the Power BI service, you create and edit reports in editing view. In Power BI Desktop, you create and edit reports in Report view.

The Power BI service has two different modes for interacting with reports: Reading view for report business users and Editing view for report owners and creators. You need a Power BI Pro or Premium Per User (PPU) license to share reports and to edit reports created by others. Without a Pro or Premium Per User (PPU) license, you can still create reports in your My Workspace, but you can't share them.

In report editing view, you have flexibility in both exploring and designing a report. All the Reading view functionality is available plus much more.

<u>Prerequisites</u>

Assign workspace roles to users so that they can edit reports. Editing view is only available to the person who created the report or to people who are assigned the member, admin, or contributor role in the workspace where the report is stored. If you share a report, the user's access will be limited to their assigned workspace role. Users who have only the viewer role can't edit reports in the workspace. See Roles in workspaces for details on permissions.

<u>Functionality only available in Editing view</u>

Take a look at the list of articles under the Power BI reports header in the Table of Contents. It's a long list and many of the articles cover functionality only available if you have editing permissions for a report.

To help you navigate the Table of Contents, Editing view is required for the following actions:

- Creating, editing, renaming, sharing, and deleting reports.
- Adding, renaming, rearranging, and deleting report pages.
- Formatting reports.
- Adding visualizations, text boxes, shapes, and buttons to a report.
- Adding visual-level, page-level, and report-level filters and setting visual interactions.
- Creating refresh schedules.
- Using Q&A functionality to create visuals in reports.
- Showing data used to create the visualization.
- Setting up drill through.
- Duplicating a report page.
- Using report settings to control your readers' interactions with reports.

Considerations and troubleshooting

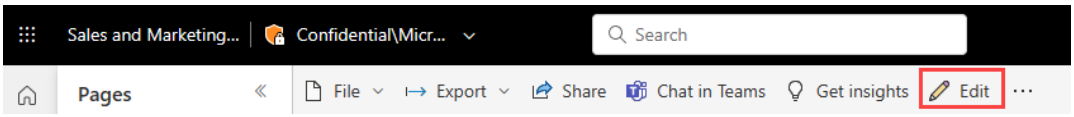
A Power BI Pro or Premium per User (PPU) license is required to edit reports created by others and to share your reports with others. If you don't have a Pro or Premium per User (PPU) license, you can still create reports, but you can't share them.

Reading view

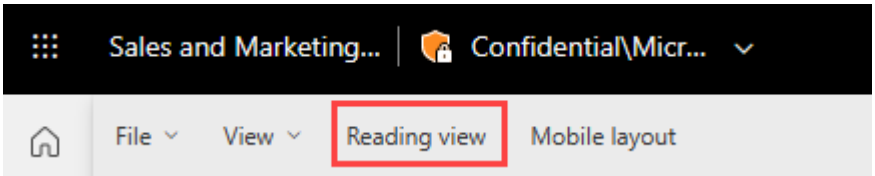
There are two modes for interacting with reports in the Power BI service: Editing view and Reading view. If you are a business user, then you are more likely to use Reading view to consume reports created by others. Editing view is used by report designers, who create the reports and share them with you. Reading view is your way to explore and interact with reports created by colleagues.

Select Editing view and Reading view

Most reports open in Reading view. To switch from Reading view to Editing view, select Edit from the action bar. If Edit is grayed out, that means that you don't have permissions to edit the report.



To switch back to Reading view, select Reading view from the action bar.



Even in Reading view, the content isn't static. You can dig in, looking for trends, insights, and other business intelligence. Slice and dice the content, and even ask it questions using your own words. Or, sit back and let your data discover interesting insights for you; send you alerts when data changes, and email reports to you on a schedule you set.