**Module 5) Creating Dashboard with Visualization Tool**

**Assignment**

### **What is Power BI and how does it differ from Excel?**

Power BI is a **data visualisation tool that lets you create interactive reports**. Excel is a spreadsheet program that lets you organize data, transform it and perform mathematical operations and calculations. **Some of the differences between Power BI and Excel are**:

**Power BI and Excel** have many similarities in terms of functionalities and how the data is presented or how we make the connection with the other data sources. Excel is much easier to use than Power BI, but Power BI has a certain upper hand, like better visualization. We should also remember that Excel is very limited to sharing reports which Power BI overcomes.

Power BI is a business analytics tool from Microsoft that helps build various dashboards and reports and can quickly deal with millions of rows of data. In contrast, Excel is also a tool from Microsoft with various built-in tools and functions that we can use for mathematical calculations, iterations, forecasting, and creating graphs and charts.

### **Explain the concept of data modeling in Power BI.**

Data modeling in Power BI is the practice of developing a conceptual representation of data and its relationships within a system, often through visual depictions. Data modeling involves creating structures, constraints, and rules for organizing and manipulating the data within this representation to ensure its integrity, improve quality, and facilitate efficient management and analysis.

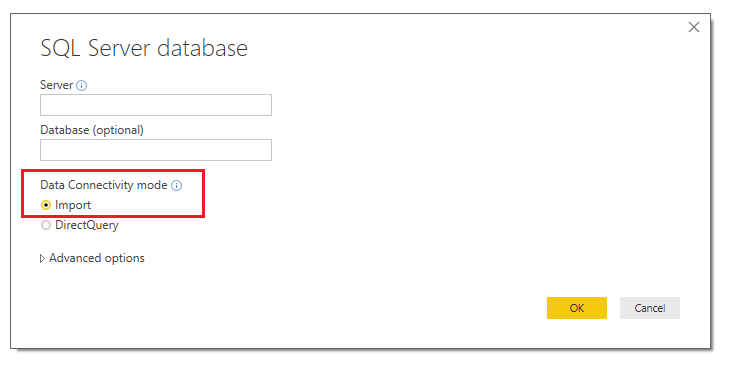
**Why Should You Use Data Modeling?**

Data modeling in Power BI offers many advantages that are essential for effective database design and administration:

* **Data Integrity**: Data modeling ensures data integrity by setting constraints and rules to enforce consistency, accuracy, and validity, helping prevent incorrect or inconsistent information from entering databases.
* **Efficient Queries and Analysis**: Data models facilitate efficient retrieval and analysis by creating relationships among entities, which allows for quick queries, joining, aggregation, and faster analysis times.
* **Data Integration:**Data modeling in Power BI facilitates data integration by creating standard definitions and relationships across different data sources, making data exchange and integration seamless, reducing redundancy, and improving the consistency of results.
* **Scalability and Flexibility**: Data models serve as the cornerstone for creating scalable and flexible database designs. Their flexible structure enables the addition or modification of data elements without disrupting their existing structures, providing adaptability to ever-evolving business requirements.
* **Types of Data Modelling in Power BI**
* There are three basic data models: relational, dimensional, and entity-relationship (ER). They all serve specific requirements and objectives. Each data modeling type offers its own approach to structuring and understanding data, meeting various organizational requirements.
* 1. **Relational Data Model (RDM)**
* The Relational Data Model (RDM) in Power BI is the foundation for organizing and structuring data. It arranges data into tables, where each table consists of rows and columns. Relationships between tables establish connections, aiding in combining and analyzing data effectively. RDM enhances data integrity, ensuring accuracy and consistency. Power BI leverages RDM to enable intuitive visualizations and insightful data analysis for informed decision-making.
* 2. **Dimensional Data Model**
* The Dimensional Data Model in Power BI is a structured framework for organizing and presenting data in a user-friendly manner. It involves two types of tables: dimension tables (containing attributes like time, and geography) and fact tables (holding quantitative data). This model simplifies data analysis by enabling users to drill down, slice, and aggregate information easily. It enhances reporting efficiency and facilitates intuitive exploration of data relationships within Power BI.
* 3. **Entity-Relationship Data Model (ERDM)**
* The Entity-Relationship Data Model (ERDM) in Power BI is a visual representation of data structures and their relationships. It employs entities (tables) connected by relationships, defining how data interconnects. ERDM enhances data analysis and visualization by illustrating data flows, aiding in the design of efficient queries, and creating intuitive reports. This model streamlines database management offers insights into data dependencies, and contributes to better decision-making through comprehensive data understanding.

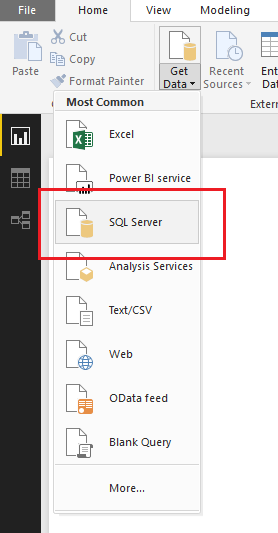
### **What are the different types of connections available in Power BI?**

There are really 3 main types of connections. The first is the most widely used, and is the default when connecting to most data sources. It is **Import**. This connection will ingest or pull the data from the data source and become part of the PBI Desktop file.  An example of where you would select import Is in the SQL Server dialog box.

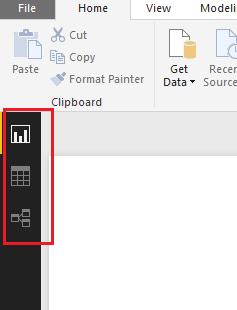


SQL Server Import

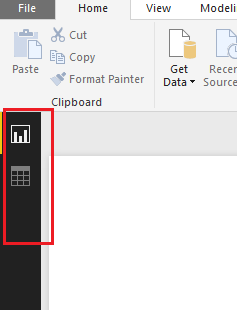
You can import data from a SQL Server by clicking **Get Data** on the **Home** ribbon.



The import connection type allows you to use the full capabilities of the Power BI Desktop and you can manipulate it however you see fit. A way to validate this is by looking at the left-hand navigation and you will see three selections.  The top selection which resembles a bar chart is the Report Page.  This is where you would place all your visuals and develop your report pages.  The second item from the top, which looks like a table is just that, the Data view in a table form.  This lets you see all the data contained with a loaded data table.  Finally, at the very bottom, the relationships selection.  This is where you will see multiple tables and the connections between the tables.  The relationships section feels like working SQL or in Microsoft Access.

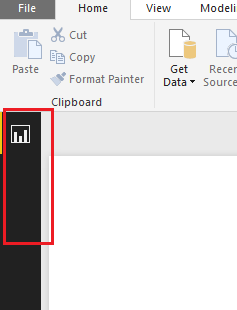


The 2nd connection type is **Direct Query**.  Notice in direct query mode the third item, relationships has been removed.  The direct query connection type is only available when you connect to certain data sources. The list of the data sources that are accessed using direct query can be found [here](https://powerbi.microsoft.com/en-us/documentation/powerbi-desktop-directquery-data-sources/).  This connection is unique in that the data does not get loaded into the PBI Desktop.  What happens, is that Power BI can communicate in the language of the data source and request information as you interact with your Power BI Visuals. The useful thing about this connection is that the data never leaves the data sources, it is only queried.  Direct Query does limit what you can do from a data manipulation perspective.  Power BI assumes you are already doing all the necessary data manipulations in your source. As a result, you don’t even have the option to mashup data and that selection is removed in the left-hand nav.

Direct Query Options

The 3rd type is **Live Connection**. There are only 3 data sources that support the live connection method at this time.  All of them are a type of (SSAS) SQL Server Analysis Services. Those types are Multidimensional, Azure Tabular and Tabular on premises. The live connection type is the most unique in that it recognizes the full model or cube that you’ve created.  Power BI Desktop turns off all data prep features.  Thus, the user is given a bare minimum in formatting and report side calculations.  All the heavy lifting is done on the server that supports the model and Power BI is only used as a reporting tool. This connection is used mainly by IT and enterprise implementations. If one looks at the left-hand navigation, you quickly realize that it is the most restrictive in terms of what can be done in the Desktop itself.

There is a fourth Live Connection that defaults to the connection type, and this occurs when you use the Power BI Service as a data source. This connection is using a SSAS connection, only the end users don’t need to set anything up other than having dataset to connect to in the Service.



### **How do you handle data transformation in Power BI?**

[**Data Transformation in Power BI**](https://www.bing.com/ck/a?!&&p=8e15a2143e67e3d92fdf9c9296a0586183b2d26c4b44cd53200a6feaffb081b0JmltdHM9MTczOTkyMzIwMA&ptn=3&ver=2&hsh=4&fclid=14da0609-4f2c-62fd-0346-12cc4e7e634a&psq=4)+How+do+you+handle+data+transformation+in+Power+BI%3f+&u=a1aHR0cHM6Ly93d3cuZ2Vla3Nmb3JnZWVrcy5vcmcvcXVlcnktZWRpdG9yLWluLXBvd2VyLWJpLWZvci1kYXRhLXRyYW5zZm9ybWF0aW9uLw&ntb=1)

Data transformation in Power BI is a crucial step to ensure that the data is in the desired format for analysis and reporting. Power BI provides a powerful tool called **Power Query Editor** for this purpose. The Power Query Editor allows users to connect, shape, and transform data from multiple sources according to their needs[1](https://www.bing.com/ck/a?!&&p=8e15a2143e67e3d92fdf9c9296a0586183b2d26c4b44cd53200a6feaffb081b0JmltdHM9MTczOTkyMzIwMA&ptn=3&ver=2&hsh=4&fclid=14da0609-4f2c-62fd-0346-12cc4e7e634a&psq=4)+How+do+you+handle+data+transformation+in+Power+BI%3f+&u=a1aHR0cHM6Ly93d3cuZ2Vla3Nmb3JnZWVrcy5vcmcvcXVlcnktZWRpdG9yLWluLXBvd2VyLWJpLWZvci1kYXRhLXRyYW5zZm9ybWF0aW9uLw&ntb=1).

**Key Data Transformation Operations**

**Rename Operations**

You can rename data sources, columns, and queries to make them more meaningful. For example, renaming a data source from "Sheet1" to "Movie Data" or a column from "TITLE" to "MOVIE NAME"[1](https://www.bing.com/ck/a?!&&p=8e15a2143e67e3d92fdf9c9296a0586183b2d26c4b44cd53200a6feaffb081b0JmltdHM9MTczOTkyMzIwMA&ptn=3&ver=2&hsh=4&fclid=14da0609-4f2c-62fd-0346-12cc4e7e634a&psq=4)+How+do+you+handle+data+transformation+in+Power+BI%3f+&u=a1aHR0cHM6Ly93d3cuZ2Vla3Nmb3JnZWVrcy5vcmcvcXVlcnktZWRpdG9yLWluLXBvd2VyLWJpLWZvci1kYXRhLXRyYW5zZm9ybWF0aW9uLw&ntb=1).

**Change Data Types**

Changing the data type of a column is essential for accurate data analysis. For instance, converting a column from "Whole Number" to "Decimal Number" can be done by right-clicking on the column and selecting the appropriate data type[1](https://www.bing.com/ck/a?!&&p=8e15a2143e67e3d92fdf9c9296a0586183b2d26c4b44cd53200a6feaffb081b0JmltdHM9MTczOTkyMzIwMA&ptn=3&ver=2&hsh=4&fclid=14da0609-4f2c-62fd-0346-12cc4e7e634a&psq=4)+How+do+you+handle+data+transformation+in+Power+BI%3f+&u=a1aHR0cHM6Ly93d3cuZ2Vla3Nmb3JnZWVrcy5vcmcvcXVlcnktZWRpdG9yLWluLXBvd2VyLWJpLWZvci1kYXRhLXRyYW5zZm9ybWF0aW9uLw&ntb=1).

**Format Operations**

Formatting options include changing text to lowercase or UPPERCASE, adding prefixes or suffixes, and removing leading/trailing whitespaces. These operations help in standardizing the data[1](https://www.bing.com/ck/a?!&&p=2455f8382f1ab94f141a0a1d6c402fda8af7ef9f3940058eb33df31bea6266ecJmltdHM9MTczOTkyMzIwMA&ptn=3&ver=2&hsh=4&fclid=14da0609-4f2c-62fd-0346-12cc4e7e634a&u=a1aHR0cHM6Ly93d3cuZ2Vla3Nmb3JnZWVrcy5vcmcvcXVlcnktZWRpdG9yLWluLXBvd2VyLWJpLWZvci1kYXRhLXRyYW5zZm9ybWF0aW9uLw&ntb=1).

**Removal Operations**

You can remove unwanted rows or columns using the "Reduce Rows" and "Manage Columns" features. This includes removing duplicates, blank rows, or specific columns that are not needed for analysis[1](https://www.bing.com/ck/a?!&&p=2455f8382f1ab94f141a0a1d6c402fda8af7ef9f3940058eb33df31bea6266ecJmltdHM9MTczOTkyMzIwMA&ptn=3&ver=2&hsh=4&fclid=14da0609-4f2c-62fd-0346-12cc4e7e634a&u=a1aHR0cHM6Ly93d3cuZ2Vla3Nmb3JnZWVrcy5vcmcvcXVlcnktZWRpdG9yLWluLXBvd2VyLWJpLWZvci1kYXRhLXRyYW5zZm9ybWF0aW9uLw&ntb=1).

**Merge and Split Columns**

Merging columns can combine multiple columns into one, while splitting columns can divide a single column into multiple columns based on a delimiter or character count. For example, merging "GENRE" and "RATING" columns into a single "Merged" column[1](https://www.bing.com/ck/a?!&&p=2455f8382f1ab94f141a0a1d6c402fda8af7ef9f3940058eb33df31bea6266ecJmltdHM9MTczOTkyMzIwMA&ptn=3&ver=2&hsh=4&fclid=14da0609-4f2c-62fd-0346-12cc4e7e634a&u=a1aHR0cHM6Ly93d3cuZ2Vla3Nmb3JnZWVrcy5vcmcvcXVlcnktZWRpdG9yLWluLXBvd2VyLWJpLWZvci1kYXRhLXRyYW5zZm9ybWF0aW9uLw&ntb=1).

**Pivot and Unpivot Columns**

Pivoting turns rows into columns, and unpivoting does the opposite. These operations are useful for reshaping the data to fit the desired analysis format[1](https://www.bing.com/ck/a?!&&p=2455f8382f1ab94f141a0a1d6c402fda8af7ef9f3940058eb33df31bea6266ecJmltdHM9MTczOTkyMzIwMA&ptn=3&ver=2&hsh=4&fclid=14da0609-4f2c-62fd-0346-12cc4e7e634a&u=a1aHR0cHM6Ly93d3cuZ2Vla3Nmb3JnZWVrcy5vcmcvcXVlcnktZWRpdG9yLWluLXBvd2VyLWJpLWZvci1kYXRhLXRyYW5zZm9ybWF0aW9uLw&ntb=1).

**Replace Values**

Replacing specific values in a column can help in cleaning the data. For example, replacing "null" values with "geeksforgeeks" in the "GENRE" column

### **What is DAX (Data Analysis Expressions) and why is it important in Power BI?**

DAX is a formula and query language that is designed to work with tabular data models and is primarily used to simplify data analysis and calculation tasks in [Power BI](https://www.geeksforgeeks.org/power-bi-tools-and-functionalities/), Microsoft PowerPivot, SQL, and Server Analysis Services (SSAS). It provides users with the ability to create sophisticated calculations, define custom metrics, and perform complex data manipulations. DAX has many powerful functions which Excel does not have.

**Importance of DAX**

DAX provides tools and features that enable flexible and customized data analysis, reporting, and modeling capabilities.

* **Advanced Calculations and Data Analysis:**DAX provides a comprehensive set of functions and operators for performing advanced calculations and data analysis tasks on large datasets. It allows users to create complex formulas and expressions, enabling calculations that go beyond simple arithmetic operations. With DAX, you can perform aggregations, apply conditional logic, perform statistical calculations, and more.
* **Integration with Microsoft Tools:**DAX is the formula language used in Microsoft’s business intelligence tools like Power BI and Power Pivot. It seamlessly integrates with these tools, allowing users to leverage DAX’s capabilities for data modeling, calculations, and analysis within a familiar environment.
* **Flexibility and Customization:**DAX offers a high degree of flexibility and customization options. Users can define calculated columns, measures, and tables to create custom calculations and derive insights specific to their business requirements. This flexibility enables users to tailor their analysis and reporting to the unique needs of their organization.
* **Time Intelligence Functions:**DAX includes a range of functions specifically designed for time-based analysis. These functions simplify complex calculations involving dates and time periods, such as year-to-date, quarter-to-date, and rolling averages. Time intelligence functions in DAX make it easier to analyze trends, compare data across different time periods, and perform time-based calculations.
* **Data Modeling Capabilities:**DAX plays a crucial role in data modeling, allowing users to define relationships between tables, create hierarchies, and implement calculated columns and measures. These capabilities enable users to build robust and efficient data models that accurately represent their business scenarios and support complex analysis and reporting requirements.
* **Performance Optimization:**DAX provides optimization techniques and best practices to improve query performance and enhance the overall responsiveness of analytical solutions. By optimizing DAX formulas, leveraging calculation dependencies, and applying performance-tuning strategies, users can optimize the execution speed of queries and improve the user experience.

DAX is built on a formula syntax similar to Excel but with additional functions and capabilities. It operates on tabular data models in Power BI, enabling users to create measures, calculated columns, and tables.

### **Can you explain the difference between calculated columns and measures in Power BI?**

In Power BI, **calculated columns** and **measures** are two essential concepts that help in data analysis and visualization. Understanding their differences is crucial for optimizing your Power BI models and reports.

**Calculated Column**

A **calculated column** is a column added to an existing table in the data model. It is computed during the data loading phase, and its values are stored in the Power BI model. The formula for a calculated column is evaluated for each row in a table. For example, to create a calculated column in Power BI Desktop, you can use a DAX expression like:

Total Cost = Sales[Price] \* Sales[Quantity]

This expression calculates the total cost by multiplying the Price and Quantity columns from the Sales table. The calculation is done row by row, and the values are stored in memory1.

**Measure**

A **measure** is a calculation created using DAX that is performed on the fly as you interact with your reports. Measures are used in visuals, pivot tables, and pivot charts. Unlike calculated columns, measures are not pre-computed and are calculated dynamically. For example, to create a measure in Power BI, you can use a DAX formula like:

Total Revenue = SUM(Sales[Total Cost])

This measure calculates the total revenue by summing the Total Cost column from the Sales table. Measures operate on aggregate values and are not stored in memory, making them faster and more efficient[1](https://www.bing.com/ck/a?!&&p=31b7e9efdcbdaa8e8a80a91abfe39535689e2b8803c92a7fb224f1dbd22a2b09JmltdHM9MTczOTkyMzIwMA&ptn=3&ver=2&hsh=4&fclid=14da0609-4f2c-62fd-0346-12cc4e7e634a&u=a1aHR0cHM6Ly93d3cuc3BndWlkZXMuY29tL3doYXQtaXMtdGhlLWRpZmZlcmVuY2UtYmV0d2Vlbi1jYWxjdWxhdGVkLWNvbHVtbi1hbmQtbWVhc3VyZS1pbi1wb3dlci1iaS8&ntb=1).

**Key Differences**

Here are the key differences between calculated columns and measures:

* **Calculation Time**: Calculated columns are computed during the data loading phase, while measures are calculated at query time.
* **Storage**: Calculated columns are stored in the data model, whereas measures are calculated on the fly.
* **Usage**: Calculated columns can be used in tables, charts, slicers, etc., while measures are mainly used in charts and visuals.
* **Formula Evaluation**: Calculated columns are evaluated row by row, while measures are aggregated over a table.
* **Performance Impact**: Calculated columns increase the model size, whereas measures have a minimal impact on performance[1](https://www.bing.com/ck/a?!&&p=31b7e9efdcbdaa8e8a80a91abfe39535689e2b8803c92a7fb224f1dbd22a2b09JmltdHM9MTczOTkyMzIwMA&ptn=3&ver=2&hsh=4&fclid=14da0609-4f2c-62fd-0346-12cc4e7e634a&u=a1aHR0cHM6Ly93d3cuc3BndWlkZXMuY29tL3doYXQtaXMtdGhlLWRpZmZlcmVuY2UtYmV0d2Vlbi1jYWxjdWxhdGVkLWNvbHVtbi1hbmQtbWVhc3VyZS1pbi1wb3dlci1iaS8&ntb=1).

**When to Use**

* **Row-level Calculations**: Use calculated columns.
* **Aggregated Analysis**: Use measures.
* **Limited Memory**: Use measures to save space.
* **Ease of Use**: Use calculated columns for simpler calculations

### **How do you handle relationships between tables in Power BI?**

### In Power BI, model relationships define how the tables are linked. In model relationships, you can create effective and insightful data visualization for easy understanding. There are different types of relationships:-

### **One-to-One Relationships**- Each row in the first table is connected to one and only one row in the second table.

### **Many-to-One Relationships** - Many rows in the first table are connected to one row in the second table.

### **One-to-Many Relationships** - Each row in the first table is connected to one or many rows in the second table.

### **Many-to-Many Relationships** - Each row in the first table can be connected to many rows in the second table. Many-to-many relationships are a type of relationship that requires an intermediate table that links the two main tables.

By using auto-detect feature while creating a relationship, users can establish relationships between tables based on field name and interconnected to data model for report and analyze. Steps-by-Steps guidelines are:

Step 1: Load your Data using the “Home” tab and select “Get Data”.

**Step 2:** On the modeling view, Go to the “Manage Relationship”.

**Step 3:** Select “Autodetect” option.

**Step 4:**Then select the “OK” option.

**Create a relationship manually**

In Power BI, users can create table relationships manually. Cardinality and cross-filter direction are automatically set during creation but users can edit them as required. By manually creating relationships in Power BI, users have greater control and are able to customize the relationships between two tables that are interconnected to each other. Steps-by-Steps guidelines are:

**Step-1:** Load your Data using the “Home” tab and select “Get Data”.

**Step-2**: On the modeling view, go to the “Manage Relationship”.

**Step-3:** Select “New” option.

* Select the first table and then select the column you want to use in a relationship.
* Select the second table and then select the column you want to use in a relationship.
* Then select the “OK” option.

### **What is the purpose of a Power BI Gateway?**

The main goal of power BI is to build the dashboard and to keep the data up to date and make sure that the data is continuously fetched at every instance of time in order to track the changes at every instance of time.  
We can integrate our Power BI to several databases like –

1. File
2. Database
3. Power Platform
4. Azure

We need a gateway to connect to the Power BI services up to the Data Sources, and we can do it using 2 gateways –

1. Personal Gateway
2. On-Premises Data Gateway

**Personal Gateway :**  
It is used for personal usage and does not require any organization’s credentials to connect to the database. Anyone can set up the personal gateway and use it to refresh the data, configure the data and install it on a personal machine.

**Key Features of Personal Gateway –**

1. Works Only with power BI cloud services.
2. Import data and set up scheduled refresh from the given data sources.
3. Runs as an application for the users who are not administrators of the computer.
4. Runs as a single user using only your login credentials.

**On-Premise Data Gateway :**  
It is a paid service mostly used with an organization’s credentials to connect to the database. Users should be from the same organization to set up the gateway and work on it.

**Key Features of Personal Gateway –**

1. Works with power BI cloud services, PowerApps, Azure Logic Apps, Microsoft Flow.
2. Import data and set up scheduled refresh from the given data sources.
3. Provides access to multiple users along with the access control per data source.
4. Support for direct Query to SQL Server.
5. Monitoring and auditing the services.

### **How can you schedule data refresh in Power BI Service?**

Even we can schedule data refreshes in Power BI to happen automatically at regular or specific intervals. This method is suitable for larger datasets that require frequent updates.

You must have a Power BI Pro or Power BI Premium license in order to schedule a refresh in the software. Following these steps will allow you to arrange a refresh once you have the necessary license:

a. Launch the Power BI service and select the dataset.

b. Select “Settings” by clicking on the ellipsis (…) in the top right corner.

c. Select “Scheduled refresh” from the navigation menu on the left.

d. Set “Scheduled refresh” to “On” in the menu.

e. Decide if you want the data to be refreshed daily, weekly, or monthly.

f. Decide what time of day the refresh should happen.

g. Set up any other options, such as the maximum parallelism permitted or the notifications for refresh failures.

Power BI will automatically update the data at the chosen frequency and time once you have scheduled a refresh. When you set up a scheduled refresh, Power BI will automatically refresh the data at the chosen frequency and time. You can check the progress of the refresh process under the “Refresh history” page, where you can observe the refresh status, refresh duration, and any errors or warnings that occur. The “Refresh history” tab allows you to keep track of the refresh process’s progress. Here, you can see the refresh status, refresh time, and any issues or warnings that happened throughout the refresh process.

### **Explain the concept of row-level security in Power BI.**

Row-Level Security (RLS) in Power BI is a feature that allows data access control at the row level within a dataset. Its primary purpose is to restrict data visibility based on user roles or attributes, ensuring that users only see the data they are authorized to access.

Row-Level Security offers several key benefits:

1. Enhanced data protection
2. Improved compliance with data regulations
3. Simplified report distribution
4. Increased user trust

Row-Level Security is widely applicable across various business scenarios:

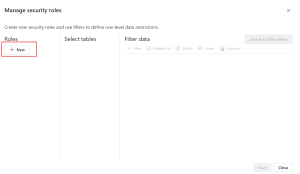
* **Sales teams**: Restricting access to regional data
* **HR departments**: Limiting visibility of employee information
* **Financial institutions**: Controlling access to customer account details
* **Healthcare providers**: Ensuring patient data confidentiality

**Define Roles and Rules in Power BI Desktop**

1. **Import or Connect to Data**:
   * Import data into your Power BI Desktop report or set up a DirectQuery connection.
   * **Note**: If you’re using an Analysis Services live connection, you’ll need to set up roles directly in the Analysis Services model instead of in Power BI Desktop.
2. **Open Manage Roles**:
   * From the Modeling tab, select**Manage Roles**.

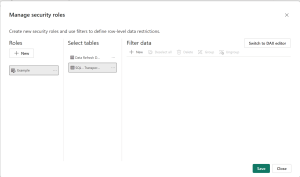
Raw Level Security

1. **Create a New Role**:
   * In the **Manage roles** window, select **New**to create a new role.

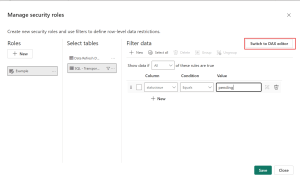


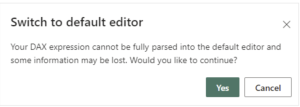
* + Name the role (for example, “EastRegion” or “HRViewOnly”) and press**Enter**.
  + **Note:** Avoid using commas in role names.

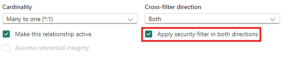
1. **Choose Tables and Set Filters**:
   * + Under **Select tables**, pick the table(s) you want to apply RLS to.
     + Under **Filter data**, you can define filters to restrict data for this role. The default editor provides a dropdown interface that creates expressions returning true or false values.



* + **Use the DAX Editor (Optional**):
    - If needed, switch to the DAX editor by selecting **Switch to DAX editor**. This is useful for setting up more complex or dynamic rules.



* + - For example, you can use expressions like [Entity ID] = “Value” or dynamic functions such as**username()** and **userprincipalname()**.
    - **Note:** In Power BI Desktop, username() follows the format DOMAIN\username, but in Power BI Service and Power BI Report Server, it uses the User Principal Name (UPN).
  + **Switch Editors if Necessary**:
    - You can switch back to the default editor by selecting **Switch to default editor**.
    - Changes made in either editor will generally carry over. However, if you define rules that only work in the DAX editor, you may receive a warning that some information could be lost if you switch back.                                                                                  
  + **Enable Bi-Directional Cross-Filtering (Optional)**:
    - By default, RLS filters apply single-directional filtering, even in bi-directional relationships.
    - To enable cross-filtering in both directions for RLS, select a relationship and check the Apply security filter in both directions .



* + - **Note:** If a table has multiple bi-directional relationships, you can only apply this setting to one.
  + **Save**:
    - Once your roles and rules are defined, select **Save**.
  + **Assign Users in Power BI Service**:
    - Publish the report to the Power BI Service.
    - Assign users or groups to the roles under the **Security** section of the dataset settings in Power BI Service.

1. **What is the Power BI Desktop and how does it differ from Power BI Service?**

**Power BI Service** is a seamless web-based platform that requires no installation, granting accessibility from anywhere with an internet connection. By simply using a web browser, users can effortlessly access **Power BI Service**, making it incredibly convenient for those who need to reach their reports on various devices.

**Power BI Service** redefines collaboration with its secure, user-friendly sharing capabilities. Once you’ve created reports and dashboards in the desktop application, utilizing a **Power BI Pro license**, sharing them with stakeholders becomes a breeze. They can access these resources through a web browser or a mobile app, ensuring that everyone collaborates with real-time data and fostering efficient decision-making.

Furthermore, **Power BI Service** places automation at the forefront with its scheduled data refresh capabilities. This feature guarantees that your reports continuously stay up-to-date with the latest data from diverse sources. Not only does it save time, but it also maintains the utmost accuracy in your reporting, ensuring that your insights are always based on the most current information available.

**Power BI Desktop** is a Windows application that you need to download and install on your local machine. It operates as a standalone tool primarily crafted for the creation of reports and dashboards. In essence, it’s your personal creative space for data work—a digital canvas where you can seamlessly import and transform data from various sources, craft interactive visualizations, and weave compelling data narratives. The **Power BI Desktop Download** is the foundational workhorse behind the analytical powerhouse, as is where the heavy lifting of data modeling occurs.

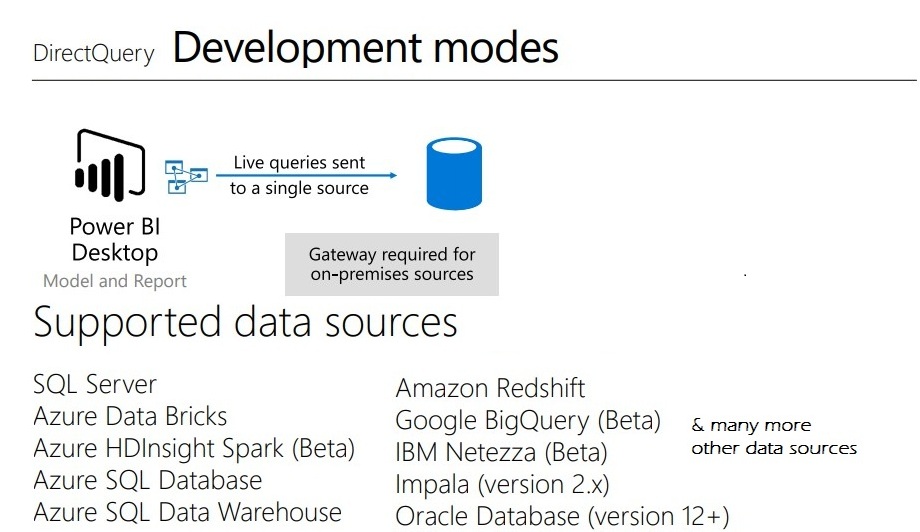
Keep in mind that **Power BI** is more than what’s just on your computer. Enter **Power BI Service**, which lives in the cloud. This part of **Power BI** is all about collaboration and sharing. It’s like a hub where you and your team can work together on data projects. You can easily share your reports and dashboards with colleagues or clients by sending them a link. Plus, it comes with features like comments, notes, and real-time collaboration, making it the go-to form of **Power B**I when you’re collaborating on data analysis projects.

1. **Explain the concept of Direct Query in Power BI.**

**Direct Query** – When you connect a data source using the Direct Query technique, your dashboard will query the data source immediately during runtime. Each filter and interaction with the report will result in a series of new queries. Since no data is imported into Power BI, the user can always query the data that already exists in the data source.

Each & every single request goes straight to the data source (which is more often SQL database) and pulls the data from there into Power BI.

**Direct Mode Overview**  
Direct query supports a number of data sources like SQL server, Azure data  bricks, Amazon Redshift & more as shown in below image.



*Img source: pbiusergroup.com*

* Data Scheduling: Data is queried from the data source, ensuring that you get the most up-to-date information. Every 15 minutes, the report is refreshed.
* Files with a smaller footprint: The Power BI Desktop files are much smaller and easier to work with because you are not caching your data when using Direct Query (faster saving, publishing etc.)
* Storage space is limited
* You don’t have to store that compressed data on Power BI Service if you don’t have a cache, thus you don’t need as much storage capacity on Service.

1. **What are Power BI templates and how are they useful?**

*Power BI templates are pre-built files that contain a set of predefined configurations, data models, queries, and visualizations. These templates can be used to streamline the process of creating reports and dashboards in Power BI Desktop based on existing templates.*

Templates help ensure consistency in report design and can save time by providing a starting point for new projects, report layouts, designing data models, and building queries. You can standardize your report easily through these templates.

* Power BI templates streamline report creation by providing predefined layouts, data models, queries, and visualizations.
* Templates promote consistency, save time, and help standardize report development within an organization.
* Care should be taken to consider data source compatibility, version issues, and security when using or sharing templates.
* Ensure that the Power BI Desktop version used to create the template is compatible with the version being used to open it.

The key features of Power BI Templates are highlighted below:

* **Predefined Layouts:** Templates often include predefined layouts and designs for reports, dashboards, and report pages, ensuring a consistent and professional look.
* **Data Models:** Templates can include predefined data models, which are relationships and structures for organizing and analyzing data, data model definitions (e.g., schema, measures, and other definitions capturing data model information)
* **Queries:** Templates may contain predefined queries that fetch and transform data from various sources and query definitions (e.g., query parameters and other query elements)
* **Visualizations:** Templates often include pre-configured charts, graphs, and other visualizations to help users quickly create informative reports.
* **Themes:** Templates may include predefined color schemes and themes to maintain a consistent visual style.

14) **How do you handle incremental data refresh in Power BI?**

*Incremental Refresh in Power BI Desktop or Service is a feature that allows you to optimize data refresh processes for large datasets. Instead of refreshing all the data every time, Incremental Refresh enables you to update only the data that has changed or is new since the last refresh. Helps in reducing data refresh times and resource consumption, making it practical to work with large datasets.*

* Incremental Refresh is a powerful feature for optimizing data refresh in Power BI.
* It’s essential to understand your data and have a clear refresh key.
* Careful configuration and monitoring are crucial for successful implementation.
* Use it with **Power BI Premium** or Premium Per User for optimal results, especially with large datasets.
* You can configure incremental refresh in Power BI Desktop or Power BI Service
* Be wary of the limitations of incremental refresh while configuring it for your datasets

**Pre-requisites for Incremental Refresh in Power BI**

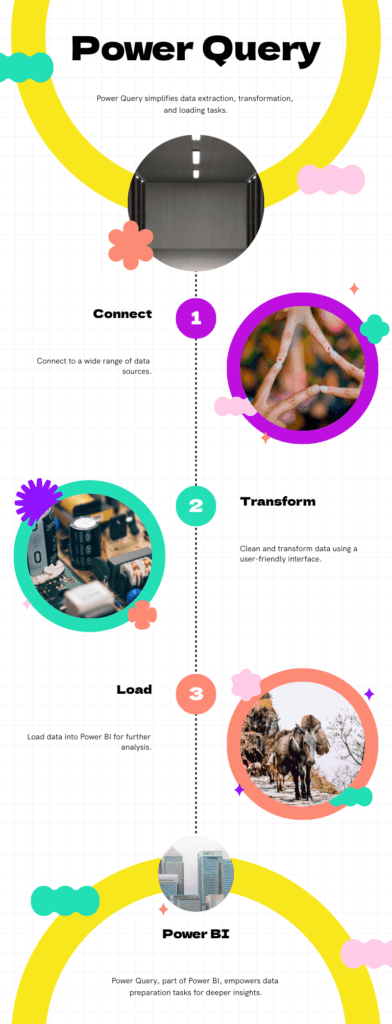
Some of the key pre-requisites to implement Incremental Refresh in **Power BI Desktop** include:

* Power BI Premium, Premium Per User (PPU), Power BI Pro or **Power BI Embedded** datasets
* Your dataset must be structured, logically related data sources such as relational databases (e.g. SQL, Azure Synapse)
* Ensure that your dataset has a date column or pass date parameters to enable filtering.
* A Date or DateTime column in your data source that represents the “refresh key” to determine what data should be refreshed.
* Knowledge of M Query Language to create custom functions for data transformation.
* A data model with tables that are suitable for incremental refresh
* Time limits are applicable to Power BI Pro or Power BI Premium, depending on what you choose to use.
* If you are using incremental refresh in Power BI Service, ensure that the current date, time, and timezone are accurately updated.

15) **What is the role of Power Query in Power BI?**

Power Query is an intuitive data transformation and data preparation tool integrated into Power BI. It enables users to:

* Connect to a wide range of data sources.
* Clean and transform data using a user-friendly interface.
* Load transformed data into Power BI for further analysis.



**Key Features of Power Query**

1. **Data Connectivity**: Connect to multiple data sources including databases, web services, and various file formats.
2. **Data Transformation**: Apply various transformations such as filtering, sorting, merging, and pivoting data.
3. **User Interface**: An easy-to-use graphical interface that allows users to perform transformations without writing code.
4. **M Language**: For advanced users, Power Query offers the M language for complex transformations.

**Step 1: Accessing Power Query**

1. **Open Power BI Desktop**: Launch Power BI Desktop to start working with Power Query.
2. **Get Data**: Click on the “Home” tab and select “Get Data”. This opens a window where you can choose your data source.

**Step 2: Connecting to a Data Source**

1. **Select a Data Source**: Choose a data source such as Excel, SQL Server, or a web page. For example, select “Excel” to connect to an Excel file.
2. **Load Data**: Navigate to the file location and select the file. Click “Load” to import the data into Power Query Editor.

**Step 3: Using Power Query Editor**

Once the data is loaded, Power Query Editor opens. Here, you can perform various data transformations.

1. **Preview Pane**: View a sample of the data to understand its structure.
2. **Query Settings**: The right pane shows the applied steps and properties of the query.
3. **Ribbon**: The top menu offers various transformation options.

**Step 4: Performing Data Transformations**

**1. Removing Columns**

1. **Select Columns**: Click on the columns you want to remove.
2. **Remove Columns**: Go to the “Home” tab and click “Remove Columns”.

**2.Filtering Rows**

1. **Select Column**: Click on the column header you want to filter.
2. **Apply Filter**: Click the dropdown arrow in the column header, choose the filter criteria (e.g., “Greater than 1000”), and apply.

**3. Splitting Columns**

1. **Select Column**: Click on the column you want to split.
2. **Split Column**: Go to the “Transform” tab, select “Split Column”, and choose the delimiter or number of characters.

**4. Merging Queries**

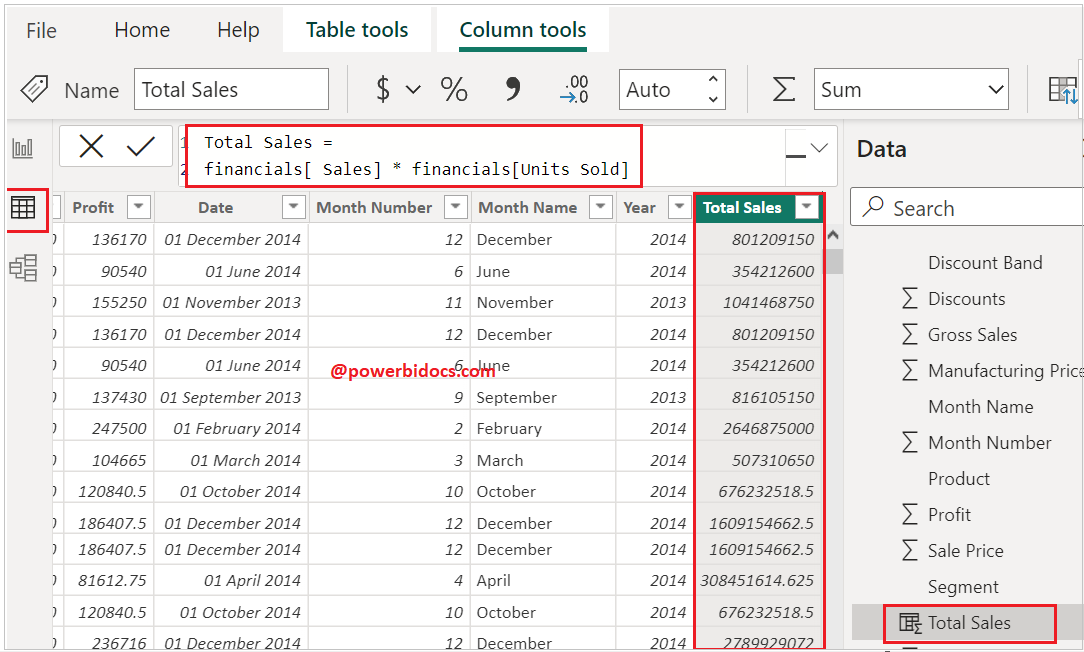
1. **Merge Queries**: Go to the “Home” tab, click “Merge Queries”, and select the queries you want to merge.
2. **Join Type**: Choose the type of join (e.g., Inner Join, Left Outer Join).

16) **Explain the difference between calculated columns and calculated tables in Power BI.**

A calculated column is a column that is added to a table in the data model, and the calculation is performed on each row of the table.

The result is stored in the column, and the calculation is performed only once when the data is loaded or refreshed.

Calculated columns are useful when you want to add a new column to the table that is derived from the values in other columns, and you want the result to be available for all further calculations.



A calculated table differs from other tables you might have in your report because it is created by pulling from data that is already in your model, whereas most other tables are imported from an outside source.

A calculated table can have a connection with other tables. You may choose to join two tables, or extract data from multiple tables to be displayed together. Like calculated columns, these tables are stored in memory and will update whenever the source data is updated.

How to create a calculated table:

* From Report View or Data View, go to the Modeling tab
* Within Modeling, find the section for Calculations
* Click “New Table”

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Calculated Columns** | **Calculated Tables** |  |
| Evaluation Context | Row context | N/A (entire table) |  |
| Storage | Stored in memory | Stored in memory |  |
| Visibility | Visible in the data view | Visible in the data view |  |
| Use Case | Adding new data to existing tables | Creating new tables from existing data |  |
| Performance Impact | Increases model size | Increases model size |  |

17) **How do you create custom visuals in Power BI?**

Custom visuals are created by developers using the custom visuals SDK. Developers use **JavaScript libraries** such as jQuery, D3, R-language scripts, etc. to create custom visuals from scratch. Once a custom visual is ready, it is then tested and debugged. After it is ready and secure, a developer packages it in the. pbiviz file and share it within an organization or upload it on **AppSource** as an open-source data visualization. From the AppSource, any Power BI user (author) can import the custom visual to use it in their report.

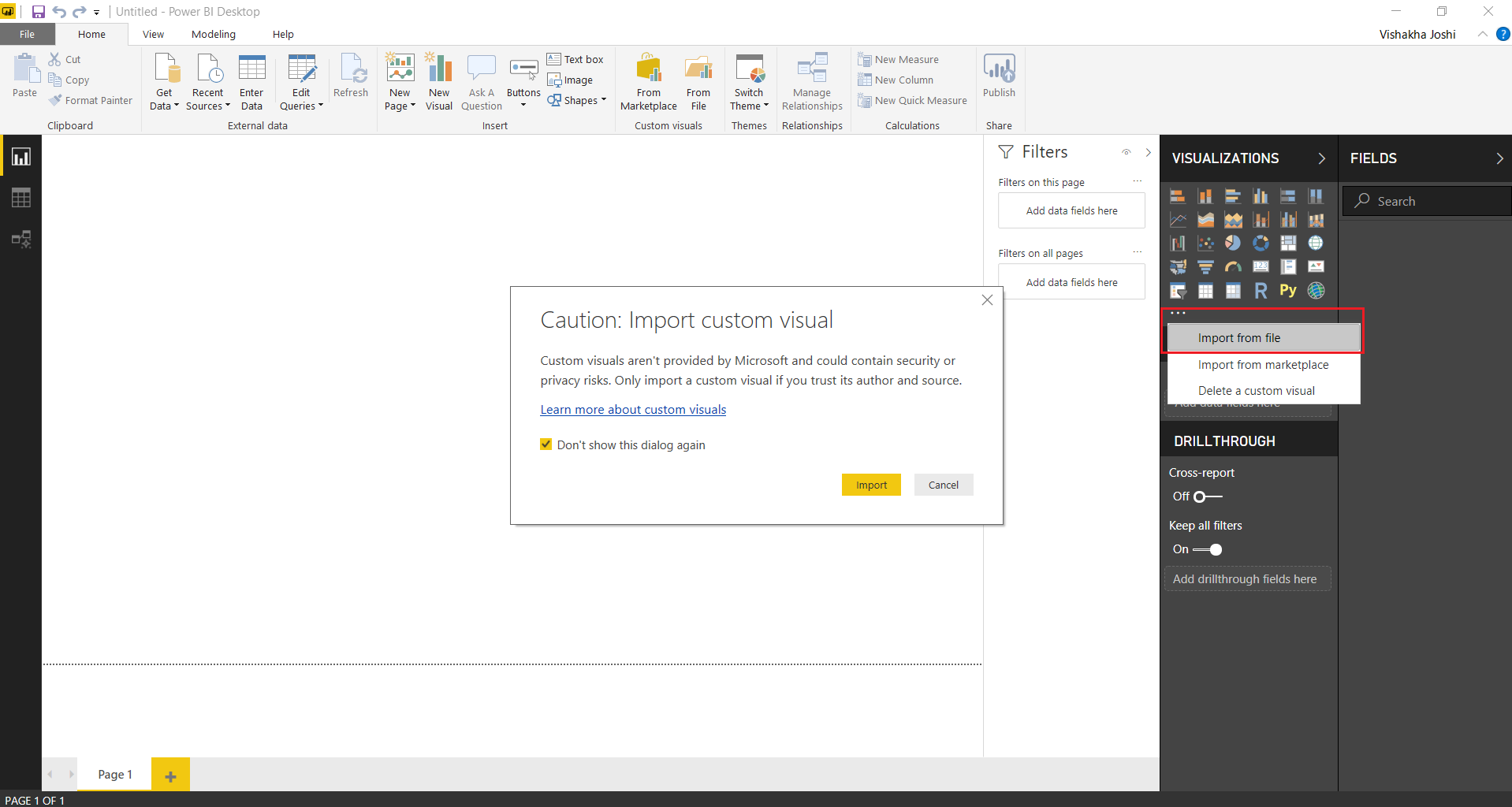
Generally, Power BI custom visuals are divided into three categories based on the way they are deployed.

* Custom visual files
* Organizational visuals
* Marketplace visuals

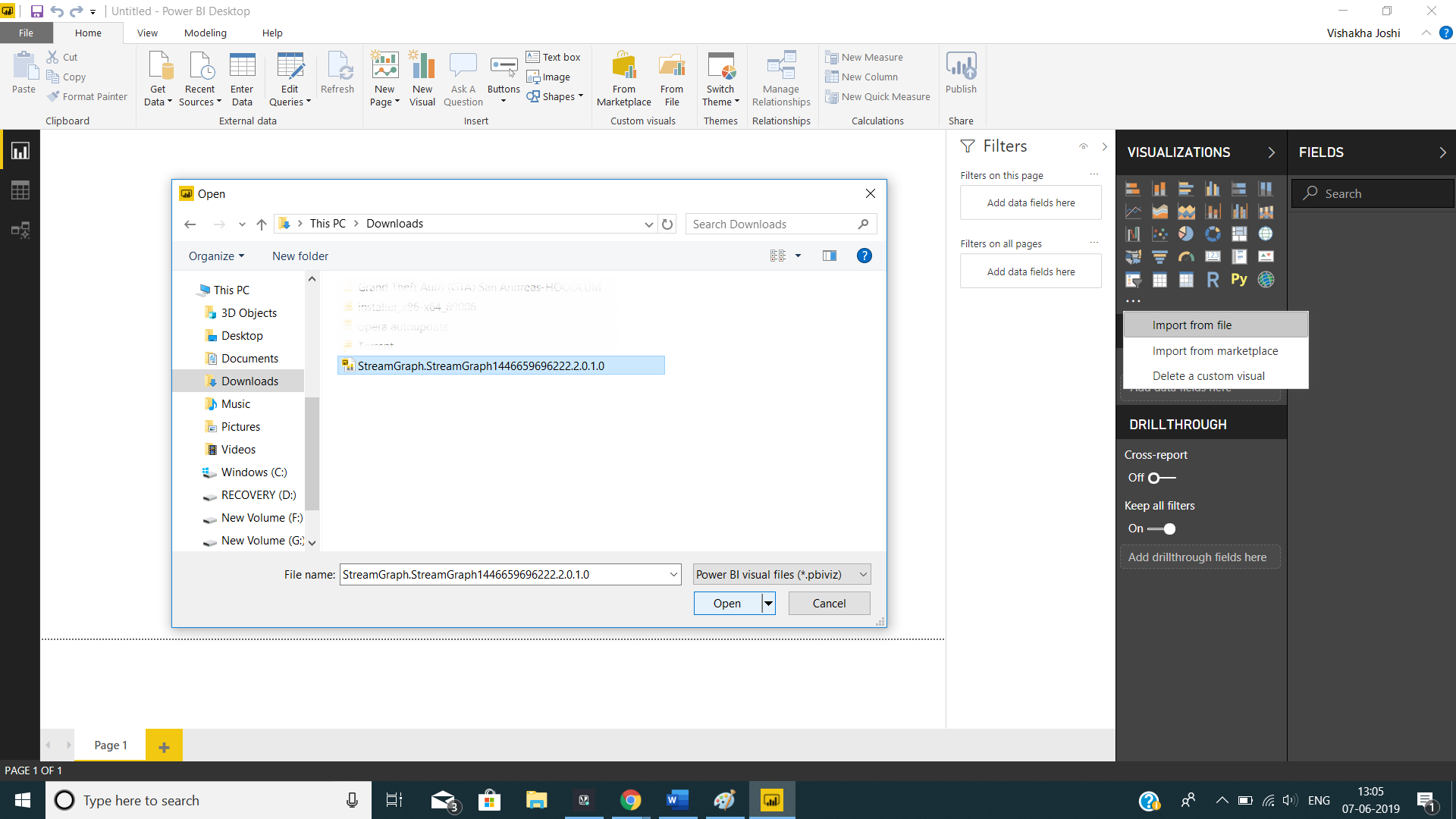
The custom visuals that you’ll download from AppSource will get downloaded in your system at a default location. We will get those custom visuals in [Power BI](https://powerbi.microsoft.com/en-us/what-is-power-bi/) Desktop through**Import from file** option.

To do this, follow these steps.

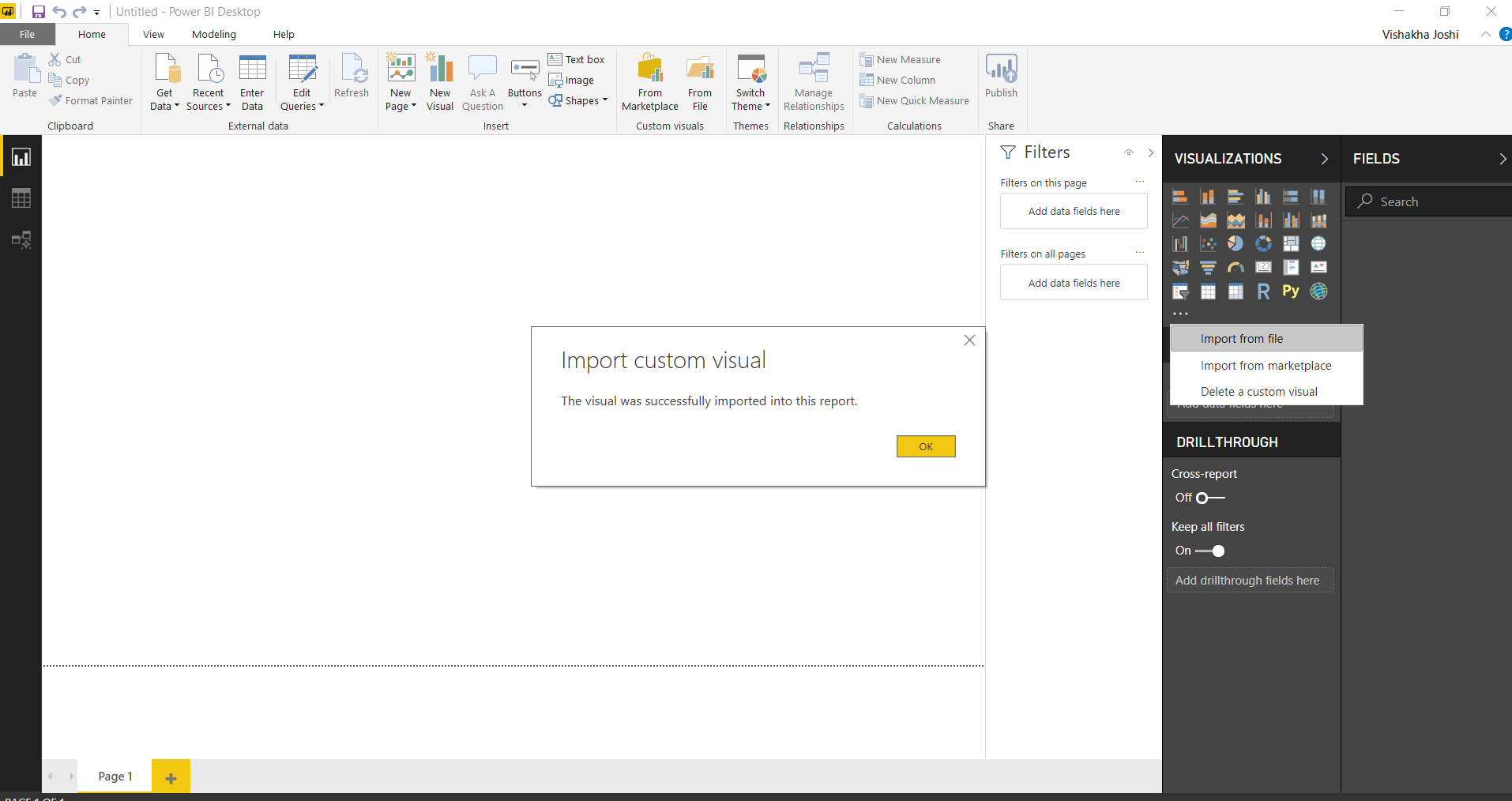
**Step 1:** Open Power BI Desktop in your system. From the Visualizations pane, click on the **ellipses …** icon. Then, select **Import from file** option.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/08/import-file.png)

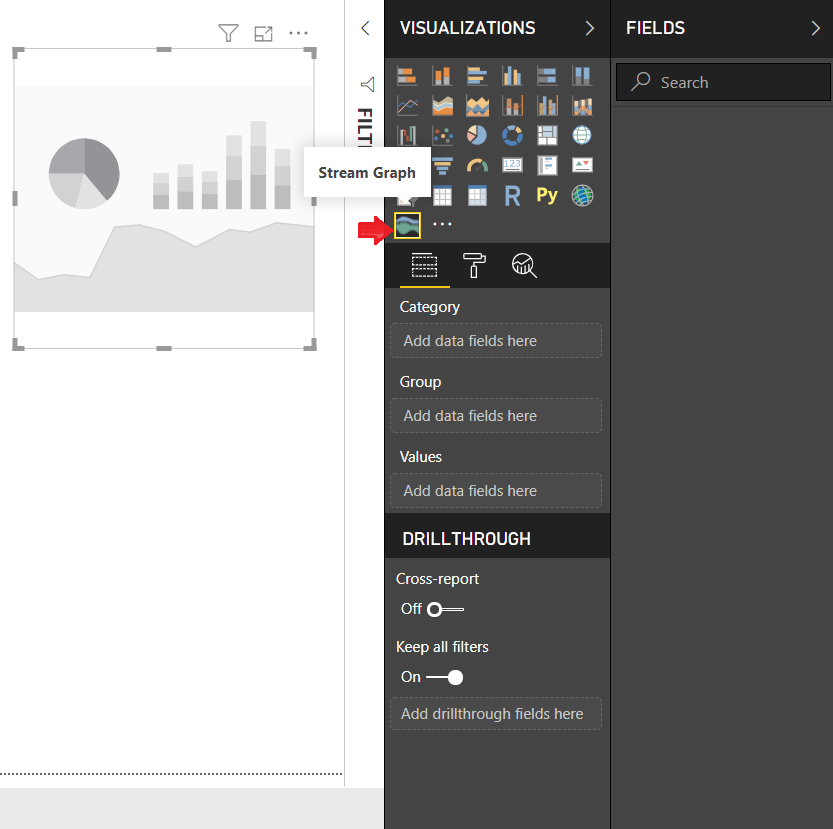
**Step 2**: Select the **.pbiviz** custom visual file from the destination. Then click on **Open**.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/08/open-file.png)

**Step 3:** A message prompt indicating successful import of file will appear on Power BI Desktop.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/08/import-successful.png)

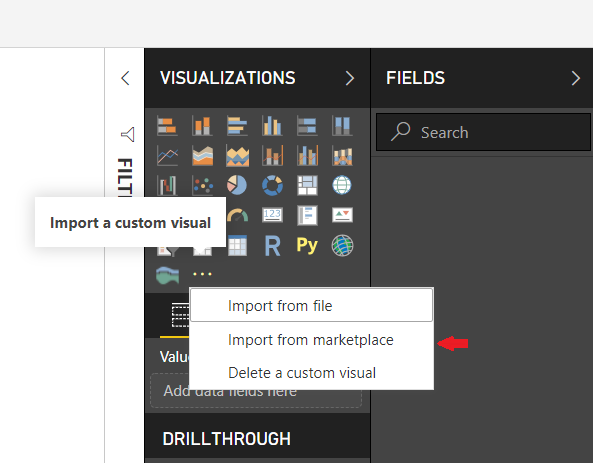
**Step 4:** As shown in the screenshot below, the Stream graph that we imported is shown in Visualizations pane and is ready to be used.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/08/custom-visual-ready.png)

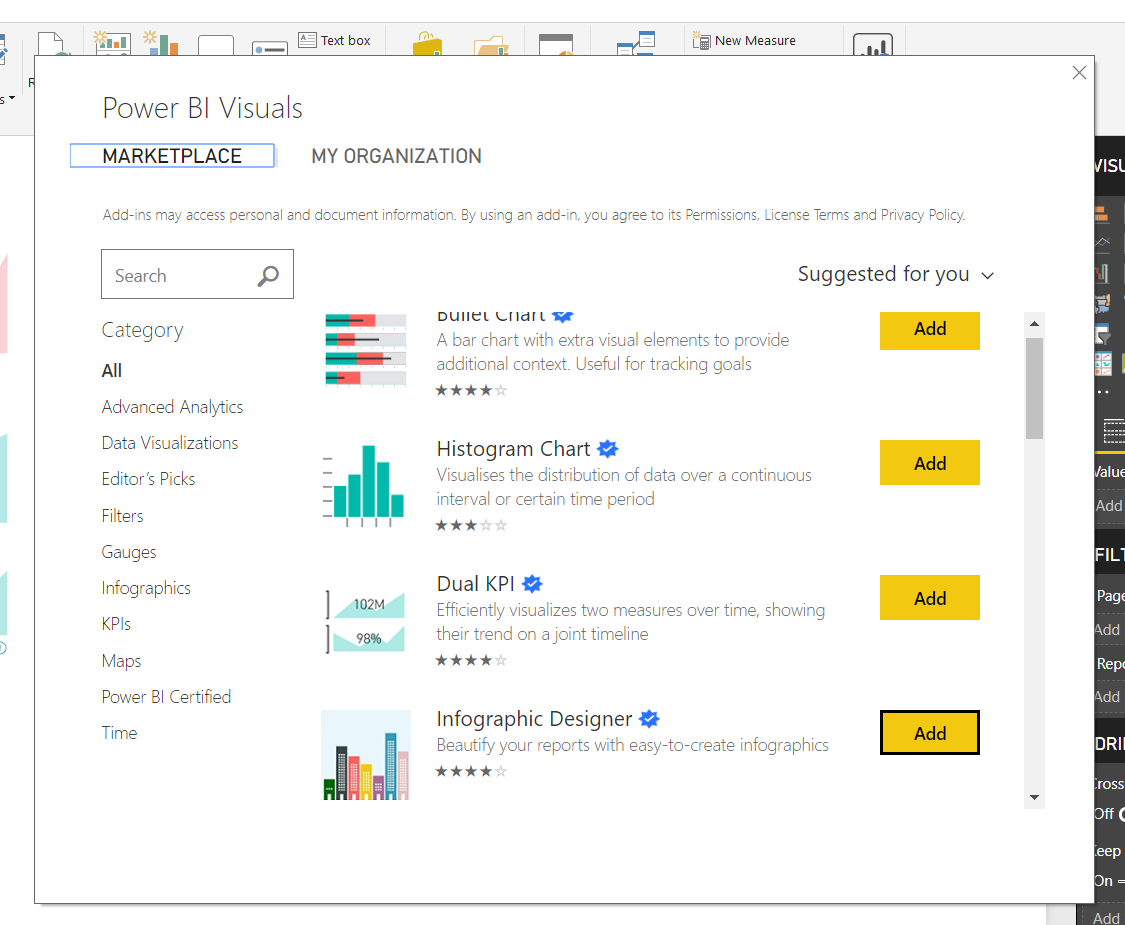
2. Importing custom visuals from marketplace

To import custom visuals available on the marketplace, follow the steps given here.

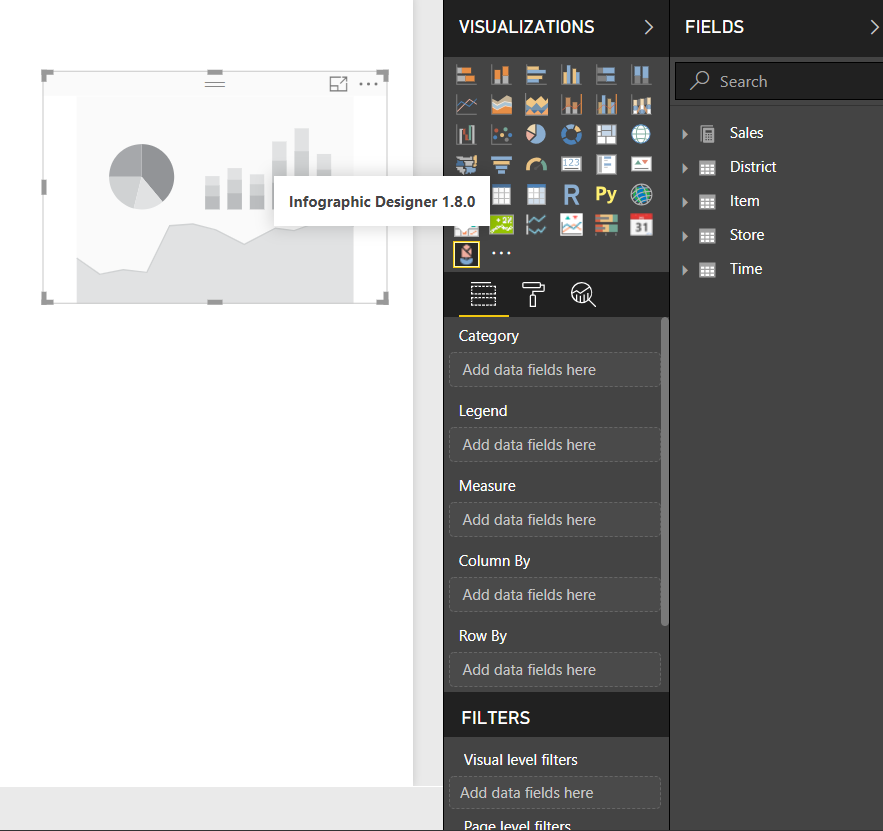
**Step 1:** Click on the **ellipses icon …** in the Visualizations pane. Select**Import from marketplace** option.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/08/import-from-marketplace.png)

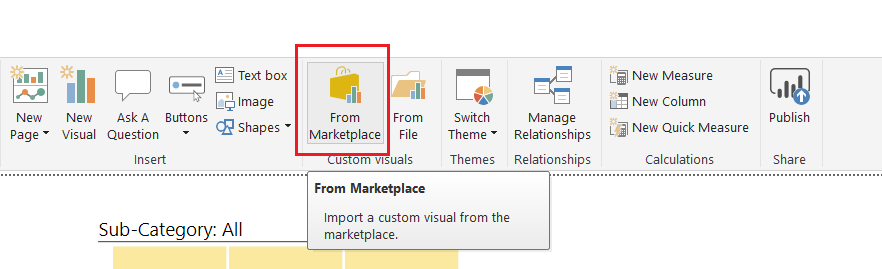
**Step 2:** The Power BI visuals dialog box will open. You will find two tabs at the top; *Marketplace* and *My Organization*. Click on **Marketplace**. You will see a list of all the available custom visuals. Also, you can select a category and then browse for the custom visuals. You can also directly type the name of a visual or category that you are looking for in the search area.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/08/marketplace.png)

**Step 3:** Select a custom visual from the list and click on **Add**. The visual will be added in your Power BI Desktop’s Visualizations pane. For instance, we have selected the **Infographic Designer** custom visual from the marketplace.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/08/infographic-designer.png)

**Step 4:** There is one more way by which you can access the marketplace. Select **From Marketplace** option given under the **Home** tab.

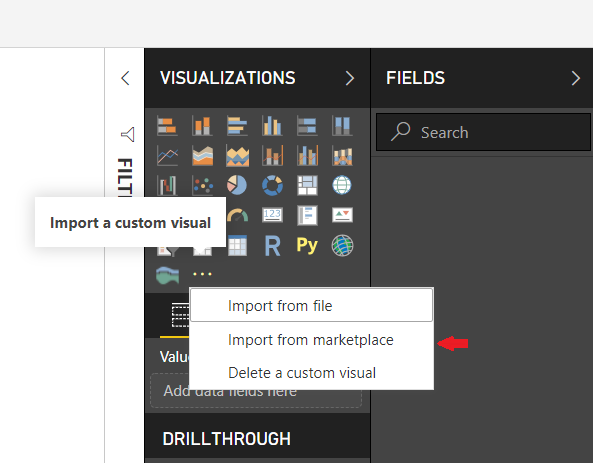
[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/08/from-marketplace.png)

3. Importing Organizational Custom Visuals

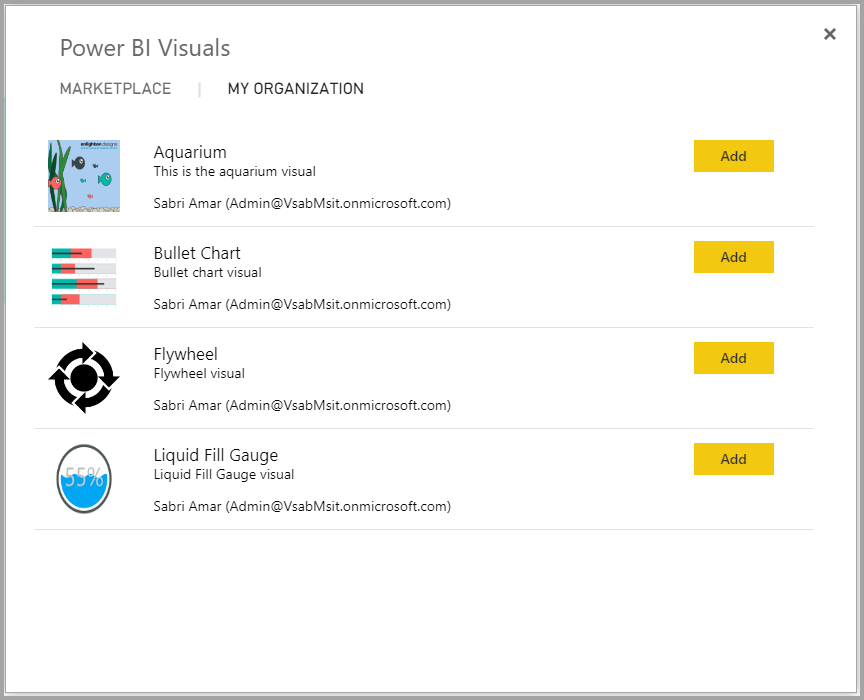
Just like we saw how to import custom visuals from the Marketplace, we’ll learn how to import organizational custom visuals.

To import custom visuals available in your organization’s repository, follow the steps given here.

**Step 1:** Click on the**ellipses icon …** in the Visualizations pane. Select **Import from marketplace** option.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/08/import-from-marketplace-1.png)

**Step 2:** Select the **MY ORGANIZATION** section. In this section, you’ll find all the custom visuals developed and uploaded by the Power BI Admin from your organization.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/08/organizational-visuals.png)

18) **What are the best practices for optimizing performance in Power BI?**

**Understanding Power BI Performance Optimization**

* **Key Performance Indicators:**Understanding and optimizing key performance indicators (KPIs) in [**Power BI**](https://anexas.net/course/power-bi-certification/) is crucial for creating efficient and responsive reports. The primary KPIs include report loading times, data refresh times, query performance, and user interaction responsiveness. Report loading times refer to the duration it takes for a report to be fully rendered and ready for user interaction. Optimizing loading times is essential for ensuring users do not experience delays when accessing their data. Data refresh times pertain to the time required to update the dataset with the latest information. Efficient data refresh processes ensure that the data remains up-to-date without causing significant downtime. Query performance is another critical KPI, involving the speed at which queries retrieve and process data from the underlying data sources. Optimizing queries can drastically reduce the time it takes to generate insights from large datasets. Finally, user interaction responsiveness measures how quickly the report responds to user actions, such as filtering, slicing, or drilling down into data. Ensuring quick responses to these interactions enhances the overall user experience and allows for seamless data exploration.
  + **Report loading times:** Understanding and optimizing report loading times is crucial for ensuring a smooth user experience. Report loading times refer to the duration it takes for a Power BI report to be fully rendered and ready for user interaction. Long loading times can be caused by complex visuals, large datasets, or inefficient data models. Optimizing loading times involves simplifying visuals, reducing the volume of data processed, and improving the efficiency of data models. This ensures that users can quickly access and interact with their reports without frustrating delays.
  + **Data refresh times:** Data refresh times pertain to the time required to update the dataset with the latest information. Efficient data refresh processes are essential to keep the data current without causing significant downtime. Strategies to optimize data refresh times include incremental data refresh, which updates only the data that has changed, and scheduling refreshes during off-peak hours. These techniques help maintain up-to-date data while minimizing the impact on performance.
  + **Query performance:** Query performance involves the speed at which queries retrieve and process data from the underlying data sources. Optimizing queries is vital for reducing the time it takes to generate insights from large datasets. Techniques to improve query performance include query folding, which pushes data transformations back to the data source, and writing efficient DAX queries. These practices help ensure that data retrieval is swift and efficient, enhancing overall performance.
  + **User interaction responsiveness:** User interaction responsiveness measures how quickly the report responds to user actions, such as filtering, slicing, or drilling down into data. Ensuring quick responses to these interactions is crucial for a seamless user experience. Optimizing user interaction involves reducing the number of visuals per page, using simpler visuals, and configuring visual interactions wisely. These steps help maintain a responsive and interactive report, allowing users to explore data without delays.
* Common Performance Issues
  + **Slow report loading:**One of the most prevalent performance issues in [**Power BI**](https://anexas.net/course/power-bi-certification/) is slow report loading. This issue can frustrate users and hinder productivity. Slow report loading often arises from complex or numerous visuals on a single page, excessive data volumes, or inefficient data models. To address this issue, it’s important to simplify visuals, reduce data volumes, and optimize data models using best practices like star schema design and appropriate data granularity.
  + **Inefficient data models:** Inefficient data models are another significant concern that can lead to prolonged data processing times and increased memory usage. Poorly designed data models can slow down report loading and query performance. Optimizing data models involves organizing data into fact and dimension tables, choosing the right level of data granularity, and reducing the use of calculated columns and measures. These practices help create efficient data models that enhance overall performance.
  + **High memory usage:** High memory usage can degrade performance, particularly in environments with limited resources. This issue can result from large datasets, excessive use of calculated columns, or inefficient data storage techniques. To mitigate high memory usage, it’s important to optimize data storage by using columnar storage where appropriate, aggregating data, and minimizing the use of calculated columns. These steps help reduce memory consumption and improve performance.
  + **Poorly optimized queries:** Poorly optimized queries can severely impact performance by taking longer to execute and consuming more resources. Inefficient queries can slow down the overall system and delay data retrieval. Optimizing queries involves writing efficient DAX queries, using query reduction techniques, and leveraging DirectQuery or Import Mode based on the specific use case. These practices help ensure that queries are executed efficiently, enhancing overall performance.

19) **How can you integrate Power BI with other Microsoft products like Azure and Office 365?**

Azure, Microsoft’s cloud computing service, provides many services that complement Power BI’s capabilities, particularly in data storage, processing, and advanced analytics.

**Azure SQL Database and Power BI**

Integrating Power BI with Azure SQL Database allows real-time data analytics and reporting. This combination enables businesses to leverage Azure’s scalability and security while utilizing Power BI’s visualization and BI capabilities. By connecting Power BI to Azure SQL Database, organizations can easily create interactive reports and dashboards that reflect the latest data, providing a dynamic analytics experience.

**Azure Machine Learning and Power BI**

Azure Machine Learning (AML) offers advanced predictive analytics and machine learning capabilities. By integrating AML with Power BI, businesses can embed predictive models directly into their reports and dashboards, enabling forward-looking insights and decision-making. For instance, you can use AML to predict customer churn or sales trends and visualize these predictions in Power BI, making complex data models accessible and understandable to business users.

Office 365’s suite of productivity tools, including SharePoint, Teams, and Excel, works seamlessly with Power BI, enhancing collaboration and accessibility of BI reports.

**Power BI and SharePoint Online**

Integrating Power BI reports into SharePoint Online enables organizations to disseminate insights more effectively across the team or organization. Users can embed Power BI reports directly into SharePoint pages, ensuring that stakeholders have easy access to the latest business insights within the context of their intranet or team sites.

**Power BI and Microsoft Teams**

Microsoft Teams, Office 365’s collaboration hub, allows for the integration of Power BI reports into team channels. This integration facilitates data-driven discussions and decisions, as team members can access and interact with BI reports without leaving the Teams environment. It promotes a culture of transparency and informed decision-making by making analytics readily accessible during meetings or chats.

**Excel and Power BI: A Seamless Data Journey**

Excel is often the starting point for data analysis in many organizations. Power BI’s integration with Excel enhances this familiar tool’s capabilities, allowing users to publish their Excel datasets to Power BI or import Excel reports into Power BI dashboards. This seamless transition from Excel to Power BI ensures that businesses can leverage their existing Excel expertise while benefiting from Power BI’s advanced data visualization and analysis features.

20) **Explain the concept of aggregations in Power BI.**

*Aggregate functions in Power BI are used to calculate a set of values within a column or a table and return a single aggregated result. These functions are essential for summarizing and analyzing data, especially with large datasets. Power BI offers several built-in aggregate functions that help users create meaningful insights and visualizations from their data.*

* Aggregate functions are essential for summarizing and analyzing data in Power BI.
* They allow you to calculate totals, averages, counts, and other summary metrics.
* Use appropriate aggregate functions based on the nature of your data and the insights you’re trying to gain.
* Combine aggregate functions with visualizations to create meaningful reports and dashboards.

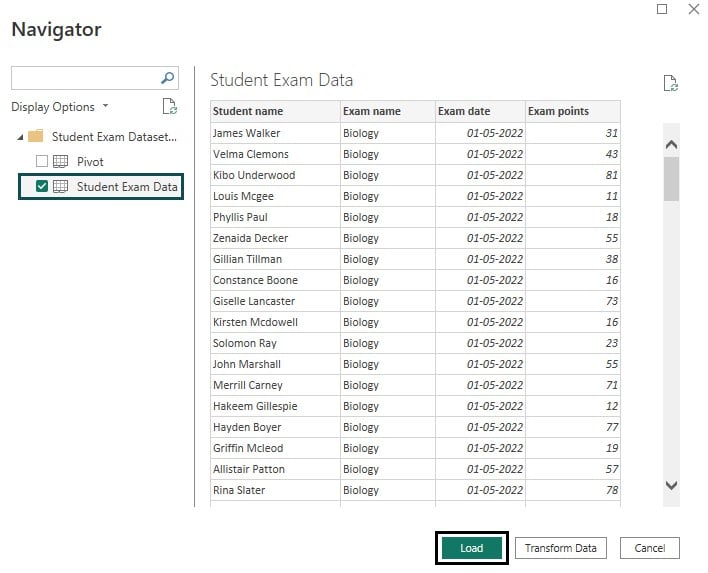
**Examples of Aggregate Functions in Power BI**

In this section, we will apply different Aggregate functions in Power BI to create data visualization in Power BI.

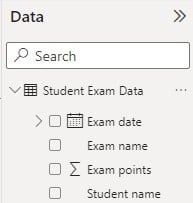
**#1 – SUM**

In this example, we will calculate the total marks scored in a subject using the aggregate function SUM in Power BI using the **Student Exam Data** dataset. Student Exam Data contains scores for each student per subject. To create a visual using the SUM function in Power BI, follow the steps highlighted below:

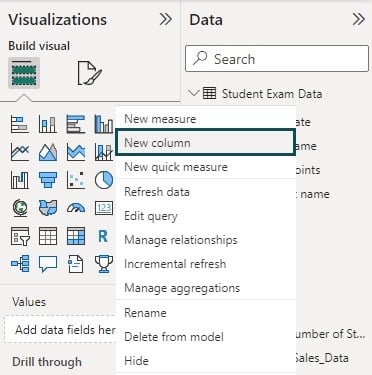
**Step 1:** Open the Power BI Desktop, import the **Student Exam Data** dataset using the **Get Data** option, and click the **Load** button. You can choose the **Transform Data** option to perform data cleaning, filtering, and transformation as needed.



Once the data is loaded, you shall be able to view it by navigating to the **Fields** pane.



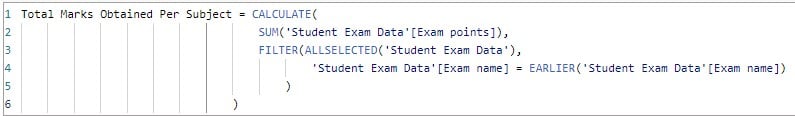
**Step 2:** Navigate to the **Fields** pane, right-click on the **Student Exam Data** table, and choose the **New column** option from the menu.



**Step 3**: Navigate to the **Formula** tab, and enter the [**Power BI DAX**](https://www.excelmojo.com/power-bi-dax/) aggregate functions for the SUM function.

Here we have created a calculated column **Total Marks Obtained Per Subject** to calculate the total marks scored in a subject.

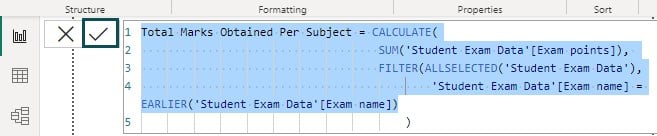
**Total Marks Obtained Per Subject = CALCULATE(SUM(‘Student Exam Data'[Exam points]),FILTER(ALLSELECTED(‘Student Exam Data’),’Student Exam Data'[Exam name] = EARLIER(‘Student Exam Data'[Exam name])))**



**Note:** We have used the aggregate functions DAX, such as [**Power BI CALCULATE**](https://www.excelmojo.com/power-bi-calculate/), FILTER, and EARLIER along with the SUM function. It is to calculate the SUM in the context of the table filter.

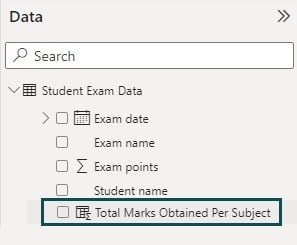
* EARLIER is a filter function primarily used for nested calculations by taking specific values as input and then producing calculations based on that input.
* CALCULATE is another filter function that evaluates an expression in a modified filter context.

**Step 4:** Once the DAX is specified, click on the **Commit** icon to save the changes. This shall create a new column in the data table.



You can see the underlying values of this new column by navigating to the **Data** view tab.

**Step 5:** Navigate to the **Fields** pane to view the newly created column.



21) **How do you handle error handling and data quality in Power BI?**

error handling in power BI is a critical aspect of developing robust and reliable reports. It ensures that your data visualizations remain accurate and informative even when faced with unexpected data or user input errors. One of the most powerful tools for error handling in power BI is the SWITCH function. This function allows report developers to define specific responses to different types of errors, making it possible to manage a wide range of scenarios gracefully. By using SWITCH, you can provide custom error messages, default values, or even complex logic to handle errors in your data.

Let's delve into the specifics of how the SWITCH function can be utilized for error handling in Power BI:

1. Understanding the SWITCH Function: At its core, the SWITCH function evaluates an expression and returns a result based on the first matching case. If no cases match, it can return a default value. The syntax is straightforward: `SWITCH(expression, value1, result1, value2, result2, ..., default)`. This structure makes it easy to map out various error conditions and their corresponding handling strategies.

2. Common Error Scenarios: Common errors in Power BI reports include missing data, invalid data types, or unexpected null values. For each of these scenarios, SWITCH can be used to check the condition and return an appropriate response, such as a warning message or a substitute value.

3. Custom Error Messages: By using SWITCH, you can create custom error messages that are more informative than generic system messages. For example, if a measure expects a numerical value but encounters text, you could use SWITCH to return a message like "Invalid input: Please enter a number."

4. Handling Null Values: Null values can cause confusion in reports. With SWITCH, you can detect nulls and replace them with a more meaningful placeholder, such as "Data not available."

5. Complex Logic for Error Handling: Sometimes, errors require more than just a simple message or value replacement. SWITCH can be combined with other DAX functions to implement complex error handling logic that takes into account multiple factors.

For instance, consider a scenario where you're calculating the average sales per region, but some regions have no sales data. A basic error handling approach might replace the missing values with zero, but this could skew the overall average. Instead, you could use SWITCH to exclude these regions from the calculation or to prompt the user to check the data source.

Here's an example of how SWITCH can be used in a measure to handle different error conditions:

```DAX

ErrorHandledMeasure =

SWITCH(

TRUE(),

ISBLANK([YourMeasure]), "No data available",

ISERROR([YourMeasure]), "Calculation error",

[YourMeasure]

In this measure, if `[YourMeasure]` is blank, it returns "No data available." If there's an error in the calculation, it returns "Calculation error." Otherwise, it simply returns the value of `[YourMeasure]`.

By **incorporating the SWITCH function into your power** BI reports, you can significantly improve the user experience by providing clear, actionable information when errors occur. This proactive approach to error handling not only makes your reports more reliable but also instills confidence in the users that they are making decisions based on accurate data.

22) What is the purpose of Power BI Embedded and when would you use it?

*Power BI Embedded is a Microsoft service for developers to integrate dashboards and interactive Power BI reports into their applications, Power BI Embedded in website, or custom solutions.*

Unlike **Power BI Desktop** or Power BI Service, which are intended for end-users, Power BI Embedded is designed for developers to embed Power BI content seamlessly within their applications, providing data visualization capabilities to their users without requiring them to have a Power BI account.

* Power BI Embedded allows developers to integrate Power BI reports into custom applications.
* It requires an Azure subscription, Power BI Pro licenses, and a good understanding of REST APIs.
* You can use it for internal or external use cases, providing interactive, data-driven insights.
* Ensure proper security and customization to create a seamless user experience.
* Assess the pricing and usage to manage costs effectively.

**Requirements:**

As a pre-requisite, you need to have one of the following

* You need an Azure AD tenant to use Embedded.
* To create and publish reports, you’ll need a Power BI Pro license.
* To use Power BI Embedded, follow these general steps:
* **Step 1:** Set up an Azure AD tenant. You can sign up for an Azure account in case you don’t have one.
* **Step 2:**Create a Power BI workspace using the Power BI service. It is required to organize your reports and datasets and publish any content you want to embed into this workspace.
* **Step 3:** Publish the Power BI reports and dashboards you want to embed. Make sure these reports are configured as you want them to appear in your application.
* **Step 4:** Register your application in Azure AD using the [**Power BI embedded analytics set up tool**](https://app.powerbi.com/embedsetup) to access these embedded reports securely.
* **Step 5:** Obtain an embed token using the [**Power BI REST API**](https://www.excelmojo.com/power-bi-rest-api/) to allow your users to view the embedded content.
* **Step 6:** Use the Power BI JavaScript API, depending on the platform (web, mobile, desktop, etc.). to embed the report into your application. Include the Power BI JavaScript library and use the powerbi.embed namespace to embed the report.
* **Step 7:** Use Azure AD to authenticate users and control their access to the embedded Power BI reports.
* Microsoft offers Embedded Playground to help developers get started with embedding Power BI content. It offers a user-friendly interface for generating code samples configuring settings for Embedded reports and dashboards, and understanding how to embed Power BI content in your applications.