# Power-BI
# Questions interview?

## ☺ What is Power BI?

Power BI is a cloud-based business analytics service by Microsoft that provides tools to transform, analyze, visualize, business intelligence capabilities, and share data. It enables users to create reports and dashboards from different data sources.

## ☺ What are the building blocks of Power BI?

Power BI consists of several key building blocks that allow users to analyze, visualize, and share data. These are:

1. **Visualizations**: Graphical representations of data such as charts, graphs, and maps.
2. **Datasets**: Collections of data imported or connected from various sources.
3. **Reports**: A collection of visualizations that appear together on one or more pages.
4. **Dashboards**: A single page containing visualizations from one or more reports, providing a high-level view of key insights.
5. **Tiles**: Individual visualizations within a report or dashboard.

These components work together to create interactive, insightful data analysis in Power BI.

## ☺ Differentiate between Power BI and Excel.

Power BI and Excel are both powerful tools for data analysis, but they serve different purposes and have distinct features:

1. Data Handling:
   - Excel: Best for handling smaller datasets and performing ad-hoc analysis. It allows users to explore data interactively but struggles with very large datasets.
   - Power BI: Designed to handle massive datasets efficiently through its in-memory processing engine, making it suitable for complex data models.
2. Visualization:
   - Excel: Offers basic charting and visualization tools, but they are less interactive and visually appealing compared to Power BI.
   - Power BI: Provides advanced, interactive visualizations and dashboards that enhance data storytelling and presentation.
3. Collaboration and Sharing:
   - Excel: Primarily a desktop application; sharing can be cumbersome and often relies on email or cloud storage.
   - Power BI: Built for collaboration, allowing users to publish reports and dashboards online for team access and real-time updates.

Overall, Excel is excellent for detailed analysis and quick calculations, while Power BI excels in data visualization and sharing insights across teams.

## ☺ What is a Power BI Desktop?

Power BI Desktop is a development tool that allows users to create data models, build visuals, and design reports that can be published to the Power BI Service.

## ☺ What data sources can Power BI connect to?

Power BI can connect to multiple data sources such as Excel, SQL Server, Oracle, Salesforce, Azure, Google Analytics, and more

## ☺ Explain the difference between Power BI and Excel?

Power BI is more advanced in data visualization, real-time reporting, and handling large datasets compared to Excel. Power BI is also better suited for sharing reports and dashboards online.

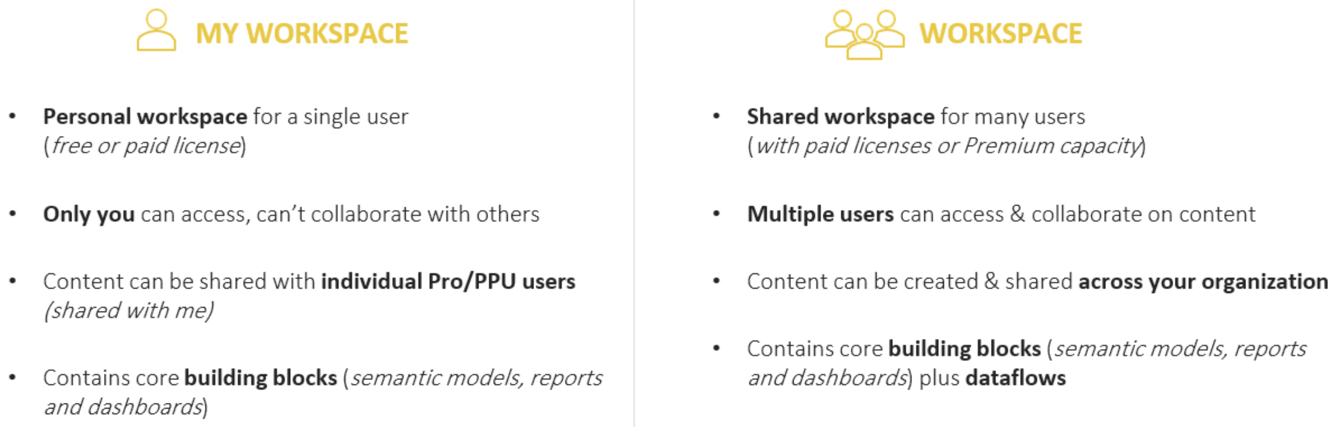
### ☺ What is a Power BI Dashboard?

A Power BI Dashboard is a single-page, visual representation of key metrics and data points from reports that are designed to give an overview of a business's performance.

### ☺ How can you share Power BI reports?

Reports in Power BI can be shared through Power BI Service by using dashboards, sharing reports via links, or embedding them in applications.

### ☺ What is the Difference between My Workspace and Workspace in Power BI Service?



**MY WORKSPACE**

- **Personal workspace** for a single user (*free or paid license*)
- **Only you** can access, can't collaborate with others
- Content can be shared with **individual Pro/PPU users** *(shared with me)*
- Contains core **building blocks** (*semantic models, reports and dashboards*)

**WORKSPACE**

- **Shared workspace** for many users (*with paid licenses or Premium capacity*)
- **Multiple users** can access & collaborate on content
- Content can be created & shared **across your organization**
- Contains core **building blocks** (*semantic models, reports and dashboards*) plus **dataflows**

### ☺ What is a DAX function in Power BI?

Data Analysis Expressions (DAX) is a formula language used in Power BI to perform calculations and build expressions for creating data models

### ☺ What is Power Query? Explain the role of Power Query in Power BI?

Power Query ( ETL tool) is a data connection technology in Power BI that allows users to discover, connect, combine, and refine data across a wide variety of sources. It is used for data transformation and shaping. It allows users to connect to various data sources, clean and transform data before loading it into Power BI for analysis.

### ☺ What are the main components of Power BI?

There are five different components of Power BI.

- Power Pivot: Fetches and cleans data and loads on to Power Query
- Power Query: Operates on the loaded data
- Power Q&A: Makes it possible for users to interact with reports using simple English language
- Power View: This lets users create interactive charts, graphs, maps, and other visuals
- Power Map: Enables the processing of accurate geographic locations in datasets

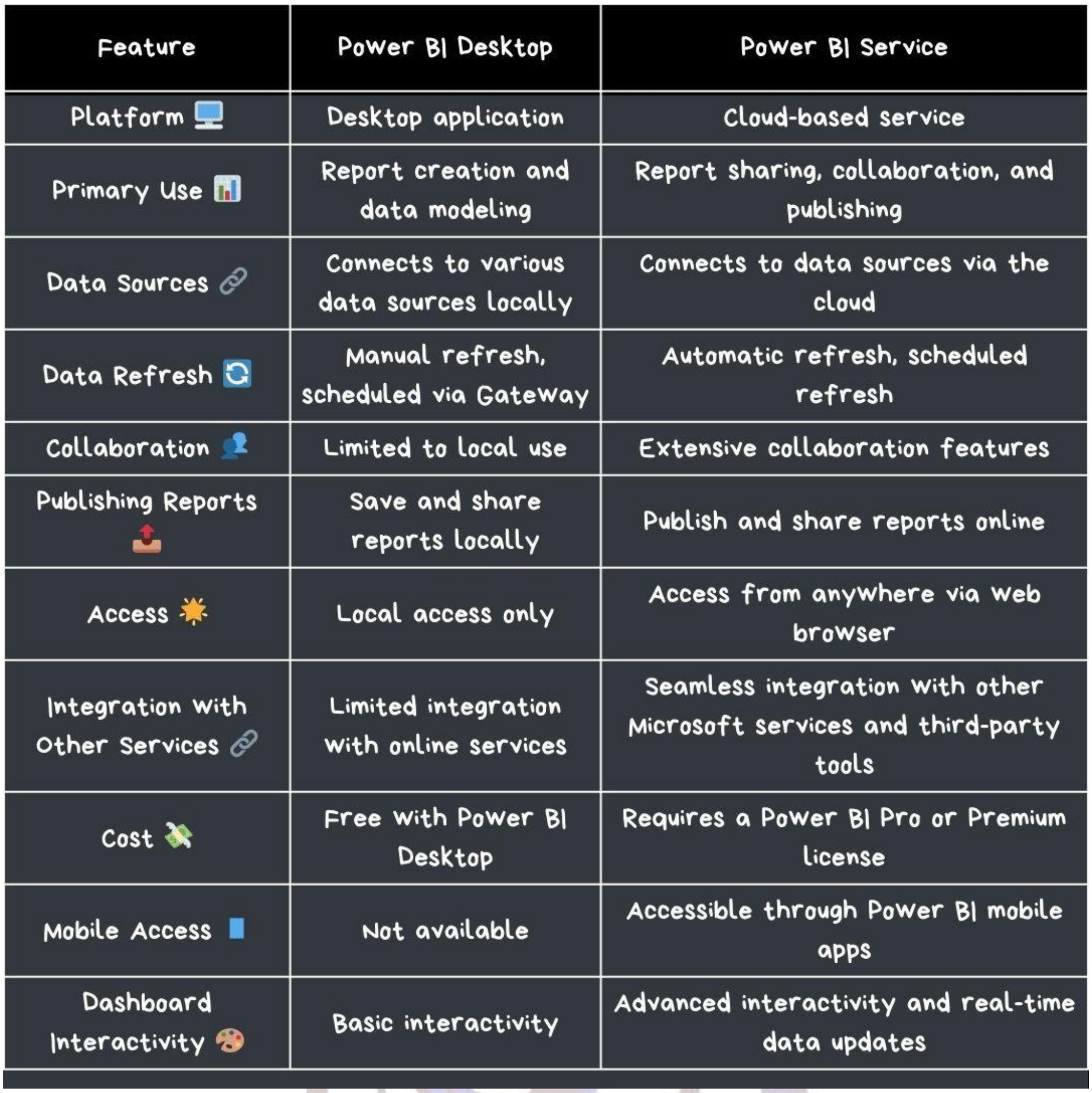
### ☺ What are the main data sources in Power BI?

Power BI supports a wide array of data sources, allowing users to connect, transform, and visualize data effectively. The main types of data sources include:

1. **Files**: Excel, CSV, XML, JSON, and text files.
2. **Databases**: SQL Server, Oracle, IBM Db2, and PostgreSQL.
3. **Cloud Services**: Azure SQL Database, Google BigQuery, and Salesforce.
4. **Online Services**: SharePoint, Dynamics 365, and web pages.
5. **Other Applications**: Power BI can connect to various other services and APIs.
This diverse range of sources enhances the flexibility and capability of Power BI for data analysis

### ☺ Differentiate between Power BI Desktop, Power BI Service, and Power BI Mobile.

Power BI Desktop is used for creating reports, Power BI Service (or Power BI Online) is the cloud service for sharing and collaborating on reports, and Power BI Mobile allows users to access reports on mobile devices.



| Feature | Power BI Desktop | Power BI Service |
|---|---|---|
| Platform 🖥️ | Desktop application | Cloud-based service |
| Primary Use 📊 | Report creation and data modeling | Report sharing, collaboration, and publishing |
| Data Sources 🔗 | Connects to various data sources locally | Connects to data sources via the cloud |
| Data Refresh 🔄 | Manual refresh, scheduled via Gateway | Automatic refresh, scheduled refresh |
| Collaboration 👥 | Limited to local use | Extensive collaboration features |
| Publishing Reports 📤 | Save and share reports locally | Publish and share reports online |
| Access 🌟 | Local access only | Access from anywhere via web browser |
| Integration with Other Services 🔗 | Limited integration with online services | Seamless integration with other Microsoft services and third-party tools |
| Cost 💸 | Free with Power BI Desktop | Requires a Power BI Pro or Premium license |
| Mobile Access 📱 | Not available | Accessible through Power BI mobile apps |
| Dashboard Interactivity 🎨 | Basic interactivity | Advanced interactivity and real-time data updates |

### ☺ What is the role of Power BI Service?

Power BI Service allows users to publish, share, and collaborate on reports created in Power BI Desktop, and access real-time reports from any location

### ☺ What is a Power BI Gateway? Explain the purpose of the Power BI Gateway.

A Power BI Gateway is used to securely transfer data between on-premises data sources and Power BI services. It facilitates refreshing datasets and running scheduled refreshes.

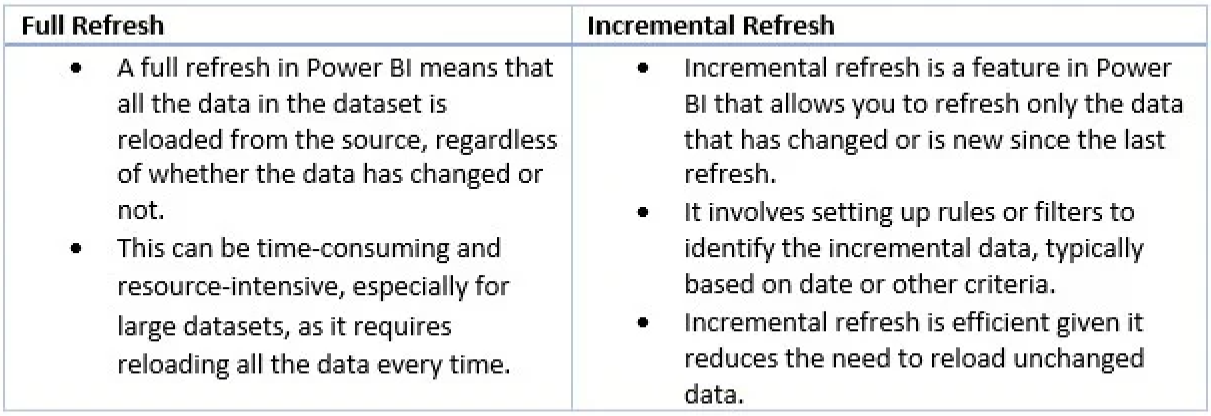
### ☺ What is Dataflow in Power BI Services?

Dataflows in Power BI are a self-service data integration feature that allows users to connect to various data sources, transform data, and store it in the cloud for use across multiple Power BI reports and datasets. Here are some key aspects:

1. **Data Transformation**: Dataflows enable users to clean and transform data before it is loaded into Power BI datasets, enhancing data quality and consistency.
2. **Reusability**: They promote the creation of reusable transformation logic, which can be shared among different reports and semantic models, reducing duplication of effort.
3. **Cloud-Based**: Dataflows run in the Power BI service, allowing for processing that is independent

of specific datasets or solutions, making them versatile for various data preparation needs. Overall, dataflows are essential for streamlining data preparation in Power BI, facilitating better data governance and collaboration.

### ☺ Explain the concept of Scheduled Refresh and Incremental Refresh?



| Full Refresh | Incremental Refresh |
|---|---|
| • A full refresh in Power BI means that all the data in the dataset is reloaded from the source, regardless of whether the data has changed or not.<br>• This can be time-consuming and resource-intensive, especially for large datasets, as it requires reloading all the data every time. | • Incremental refresh is a feature in Power BI that allows you to refresh only the data that has changed or is new since the last refresh.<br>• It involves setting up rules or filters to identify the incremental data, typically based on date or other criteria.<br>• Incremental refresh is efficient given it reduces the need to reload unchanged data. |

### ☺ What's incremental refresh and when should we avoid?

1. **Small Datasets:** If your dataset is small enough that a full refresh is quick and resource-efficient, incremental refresh may be unnecessary.
2. **Frequent Data Structure Changes:** If the schema or structure of your data changes often, managing incremental refresh can become complex and error-prone.
3. **Complex Queries:** Queries that involve complex transformations might not benefit from incremental refresh and could be slower.
4. **Data Source Limitations:** Some data sources may not support the necessary queries or capabilities for incremental refresh

### ☺ How can you schedule data refresh in Power BI?

Data refresh can be scheduled in Power BI Service to ensure that your datasets are updated at specified intervals. To schedule data refresh in Power BI, follow these steps:

1. **Open the Dataset Settings**:
   - Go to Power BI Service, find the dataset you want to refresh, and click the **Settings** icon.
2. **Configure Scheduled Refresh**:
   - In the **Scheduled Refresh** section, toggle the button to **On**. Set the frequency (daily, weekly, etc.), time zone, and specific time slots for the refresh to occur.
3. **Set Refresh Frequency**:
   - Power BI Premium allows more frequent refreshes (up to every 15 minutes). For non-Premium users, refreshes are limited to eight per day .
4. **Save and Apply**:
   - After configuring the refresh schedule, save the settings. The refresh will now automatically occur based on the schedule you set

### ☺ What is Query Folding in Power Query?

The process where Power Query translates transformations into native queries executed by the data source, optimizing performance

### ☺ Can you explain what a KPI in Power BI is?

A KPI (Key Performance Indicator) in Power BI is a visual that displays the progress toward a measurable goal.

### ☺ What are data alerts in Power BI?

Data alerts notify users when certain thresholds are met or exceeded in a report or dashboard.

### ☺ How do you create a hierarchy in Power BI?

Hierarchies in Power BI are created to organize data fields into parent-child relationships, allowing for drill-down functionality in reports.

### ☺ What is the use of Power BI Templates?

Power BI Templates allow you to save a report’s structure without saving the underlying data, making it easy to reuse report designs.

### ☺ What are the different views in Power BI Desktop?

Power BI Desktop has three views: Report View, Data View, and Model View, used for report creation, data inspection, and managing relationships, respectively.

### ☺ Can you explain drill-through functionality in Power BI? What are Drill through and Drilldown in Power BI?

In Power BI, both **Drill-through** and **Drilldown** are interactive features designed for deeper data exploration, but they serve different purposes:

1. **Drilldown**:
   - Drilldown allows users to navigate hierarchical levels within a single visual. For example, if you have a sales chart grouped by region, you can drill down into a specific region to see city-level data.
   - Drilldown focuses on exploring data within the same visual by revealing more granular details.
   - It is used in visuals with hierarchical fields, like date or geographical hierarchies.
2. **Drill-through**:
   - Drill-through enables users to navigate between report pages. You can pass data from one page to another to focus on a particular subset of data.
   - For instance, from a summary page, users can drill through to a detailed report on a specific customer or product.
   - Drill-through creates a new context by applying the filters from the source page to the destination page.

### ☺ What is meant by ‘cross-filtering’ in Power BI?

Cross-filtering in Power BI refers to how one visual can interact with another, filtering the data dynamically across visuals in the report.

### ☺ How can you do automation in Power BI?

Automation can be done with Power Automate. Power Automate is a cloud-based service that allows you to create workflows to automate business processes. With Power Automate, you are able to use out-of-the-box connectors that allow you to connect to more than 200 services and automate repetitive tasks, saving you a considerable amount of time and effort.

### ☺ How would you automate the process of appending new data to an existing dataset in Power Query?

To automate the process of appending new data to an existing dataset in Power Query, follow these steps:

1. Data Source Setup: Create a folder where the new data files (e.g., CSVs) will be saved. This can be a local or SharePoint folder.
2. Use Power Query to Connect: In Power BI, use Power Query to connect to the folder and load the files. You can use the "Folder" connector to pull in all files within that folder
3. Combine Queries: Within Power Query, you can use the "Combine" function to append new data to the existing dataset. Ensure that the structure of the new data matches the existing dataset.
4. Schedule Refresh: Finally, set up a scheduled refresh in Power BI to automatically fetch and append new data as it becomes available. This way, each time the dataset refreshes, it will include the newly added data.

By following these steps, you can effectively automate the appending of new data in Power Query.

### ☺ What are the benefits of using custom visuals in Power BI?

Custom visuals allow for tailored and enhanced visualizations that meet specific business requirements beyond the standard set provided by Power BI.

### ☺ How can you embed Power BI reports into an application?

Power BI reports can be embedded into applications using Power BI Embedded, a set of REST APIs for seamless integration.

### ☺ What is the use of quick measures in Power BI? Answer:

Quick measures allow users to quickly create commonly used calculations without writing DAX code.

### ☺ What is Q&A in Power BI?

Q&A is a natural language querying feature that allows users to ask questions about their data and get visual answers.

### ☺ How do you create and use calculated tables in Power BI?

Calculated tables are created using DAX expressions to derive new tables from existing data models.

### ☺ What is Power BI Report Builder?

Power BI Report Builder is a tool for creating paginated reports that are ideal for printing or exporting to PDF formats.

### ☺ How does Power BI handle data privacy levels?

Power BI enforces privacy levels to ensure that data from different sources cannot be combined unless the privacy level is compatible, protecting sensitive information.

### ☺ What are implicit and explicit measures in Power BI?

Implicit measures are created automatically by Power BI, while explicit measures are defined manually using DAX formulas.

### ☺ What is Merge and Append in power query editor?

Merge: Combines columns from two tables based on a matching column.
Append: Combines rows from two or more tables into a single table.

### ☺ What are slicers in Power BI?

Slicers are visual tools that allow users to filter data in a report by a selected dimension or attribute.
Learning Resource: Using Slicers

### ☺ What is a slicer sync in Power BI? If you want to apply the same slicer across multiple pages how will u achieve that?

Slicer sync allows you to synchronize slicers across multiple report pages, ensuring consistent filtering behavior. To apply the same slicer across multiple pages in Power BI, you can use the Sync Slicers feature. **Add the Slicer:** First, create the slicer on one of your report pages.

> **Enable Sync Slicers:** Go to the View tab in Power BI and click on Sync slicers. This will open the Sync Slicers pane.
>
> **Select the Pages:** In the Sync Slicers pane, select the slicer you want to sync, then check the boxes next to the pages where you want the slicer to apply. This ensures that the slicer selection on one page is synchronized across all selected pages.

**Save Your Changes:** After selecting the pages, save the report. Now, the slicer selections will be applied consistently across all synchronized pages

### ☺ What will be preferred slicer or filters? like according to u which is good to use. What is difference between Filters and Slicers?

The choice between slicers and filters in Power BI depends on the use case and user experience goals:

1. Slicers:
   - Best for User Interaction: Slicers are on-canvas visuals and provide an intuitive and interactive way for users to slice data directly within a report. They are more visually dynamic and work well for users who need quick, visible selections on the report page.
   - Preferred for Dashboards: If the goal is to create a user-friendly, visually appealing dashboard with easy access to selection options, slicers are preferable.
2. Filters:
   - Best for Complex Filtering: Filters offer more customization and flexibility, allowing developers to apply filtering logic at various levels (visual, page, report). Filters can be hidden from users, keeping the interface clean, and are better for detailed or background filtering needs.
   - Preferred for Detailed Reports: If you need precision and control over how data is filtered across the entire report, filters are more suitable.

Conclusion:

- Use slicers for interactive, user-facing filtering.
- Use filters for more complex, behind-the-scenes filtering logic.

### ☺ How do you handle errors in Power Query?

In Power Query, handling errors is essential for ensuring smooth data transformation and reporting. Some common techniques to handle errors in Power Query include:

1. Using Try and Otherwise: Power Query offers the try keyword to catch errors. If an error occurs in a step, try will allow you to handle it, and otherwise define what to return if an error occurs.

**try [Expression] otherwise [AlternativeValue]**

2. Replacing Errors: You can replace errors directly in the Power Query editor by selecting Replace Errors and providing a default value.

**Table.ReplaceErrorValues([ColumnName], {"Error", "Default Value"})**

3. Conditional Columns: You can create conditional columns using if statements to avoid potential errors by checking for specific conditions beforehand.

**if [ColumnName] = null then "Default Value" else [ColumnName]**

4. Detecting and Removing Errors: Power Query provides the option to filter out rows that contain errors using the Remove Errors option in the editor.

These techniques help ensure the data flows smoothly through transformations and prevent disruptions in reports.

### ☺ What are tooltips in Power BI?

Tooltips provide additional information when hovering over visuals in a report, giving users more context without cluttering the main visual.

### ☺ What is the use of relationships in Power BI?

Relationships in Power BI connect two or more tables, enabling you to build a comprehensive data model that ensures accurate calculations and data retrieval.

### ☺ What is Conditional Formatting in Power BI? Describe how to set up conditional formatting for a table or matrix visual in Power BI.

Conditional Formatting in Power BI allows you to dynamically apply formatting based on the value of a field. This can include adjusting the background color, font color, icons, or data bars based on certain conditions. This feature is essential for highlighting trends, outliers, or key insights in your data.

**Steps to set up conditional formatting for a table or matrix visual in Power BI:**

1. **Select the visual**: Click on the table or matrix visual to which you want to apply the conditional formatting.
2. **Open the Format pane**: Go to the “Visualizations” pane and select the “Format” icon.
3. **Expand the Conditional Formatting section**: Scroll to find and expand the "Cell elements" or “Values” section.
4. **Choose the field to format**: Click the dropdown next to the field or value you want to format (e.g., background color, font color).
5. **Define the format style**: Choose the type of conditional formatting, such as using a color scale, rules, or field values.
6. **Apply the formatting**: Click "OK" to apply the conditional formatting and update the visual.

### ☺ What are some popular DAX functions you have used?

Some popular DAX functions include SUM, COUNT, DISTINCTCOUNT, RELATED, IF, and CALCULATE.

### ☺ How do you optimize performance in Power BI reports?

To optimize Power BI report performance, follow these best practices:

1. **Optimize the Data Model**:
   - Remove unnecessary columns and tables.
   - Use a star schema design to simplify data relationships.
   - Avoid calculated columns where possible and use measures instead.
2. **Reduce Data Volume**:
   - Filter out irrelevant data and aggregate it to reduce the size of datasets.
   - Use appropriate data types and minimize high-precision data types (like decimals).
3. **Turn off Auto Date/Time**:
   - Disabling auto date/time helps in optimizing refresh and calculation performance.
4. **Optimize DAX Queries**:
   - Simplify complex DAX expressions and avoid nested calculations.
   - Use variables in DAX for better readability and performance.
5. **Limit Visuals and Interactions**:
   - Reduce the number of visuals on a report page to improve rendering time.
   - Limit the number of slicers and interactive elements.
6. **Incremental Data Refresh**:
   - Implement incremental refresh to only update new or changed data instead of reloading the entire dataset.

### ☺ How do you implement dynamic row level security in Power BI?

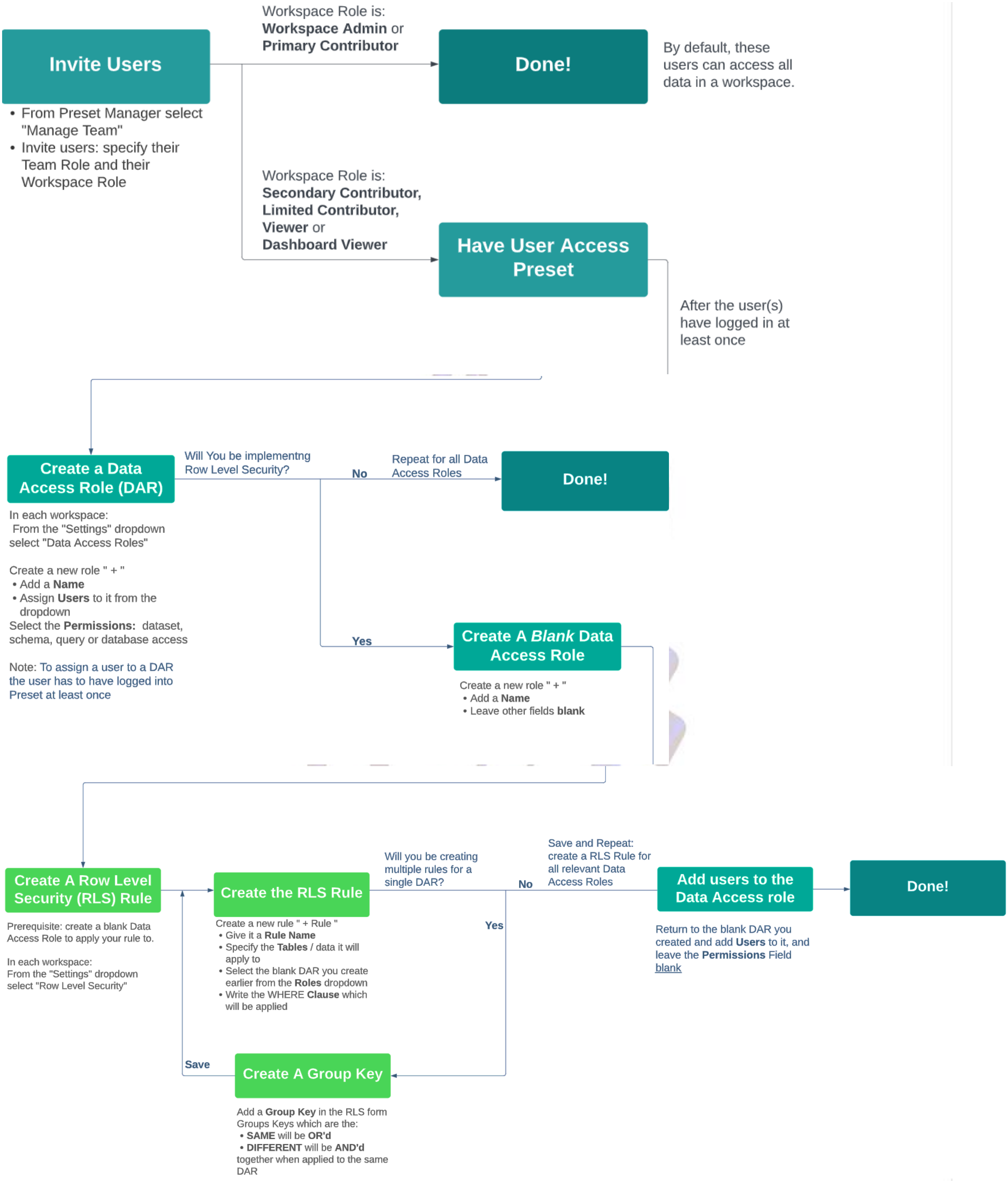
To implement dynamic row-level security (RLS) in Power BI, follow these steps:

1. Create a Security Table: Create a table that contains user roles and the corresponding filters, typically including user email or ID.
2. Define Roles: In Power BI Desktop, go to the "Modeling" tab and select "Manage Roles." Create a new role and set a DAX filter expression based on the security table that references the user’s identity.
3. Add DAX Filter: Use DAX expressions to filter data based on the logged-in user. For example: Table[UserEmail] = USERPRINCIPALNAME(), where Table is your security table.
4. Test the Roles: Use the "View as Role" feature to simulate how different users will see the data based on their roles.
5. Publish to Power BI Service: After testing, publish your report. Ensure that the security table is also published.
6. Assign Users: In the Power BI Service, assign users to the defined roles under the dataset settings.

Dynamic RLS allows you to restrict access dynamically based on user attributes, enhancing data security

## ☻ What is Row-Level Security (RLS) in Power BI?

Row-Level Security allows you to restrict data access for certain users based on filters defined in the dataset.



**Invite Users**

- From Preset Manager select "Manage Team"
- Invite users: specify their Team Role and their Workspace Role

Workspace Role is: **Workspace Admin** or **Primary Contributor**

**Done!**

By default, these users can access all data in a workspace.

Workspace Role is: **Secondary Contributor, Limited Contributor, Viewer** or **Dashboard Viewer**

**Have User Access Preset**

After the user(s) have logged in at least once

**Create a Data Access Role (DAR)**

In each workspace:
From the "Settings" dropdown select "Data Access Roles"

Create a new role " + "
- Add a **Name**
- Assign **Users** to it from the dropdown

Select the **Permissions:** dataset, schema, query or database access

Note: To assign a user to a DAR the user has to have logged into Preset at least once

Will You be implementng Row Level Security?

No — Repeat for all Data Access Roles

**Done!**

Yes

**Create A *Blank* Data Access Role**

Create a new role " + "
- Add a **Name**
- Leave other fields **blank**

**Create A Row Level Security (RLS) Rule**

Prerequisite: create a blank Data Access Role to apply your rule to.

In each workspace:
From the "Settings" dropdown select "Row Level Security"

**Create the RLS Rule**

Create a new rule " + Rule "
- Give it a **Rule Name**
- Specify the **Tables** / data it will apply to
- Select the blank DAR you create earlier from the **Roles** dropdown
- Write the WHERE **Clause** which will be applied

Will you be creating multiple rules for a single DAR?

No — Save and Repeat: create a RLS Rule for all relevant Data Access Roles

**Add users to the Data Access role**

Return to the blank DAR you created and add **Users** to it, and leave the **Permissions** Field blank

**Done!**

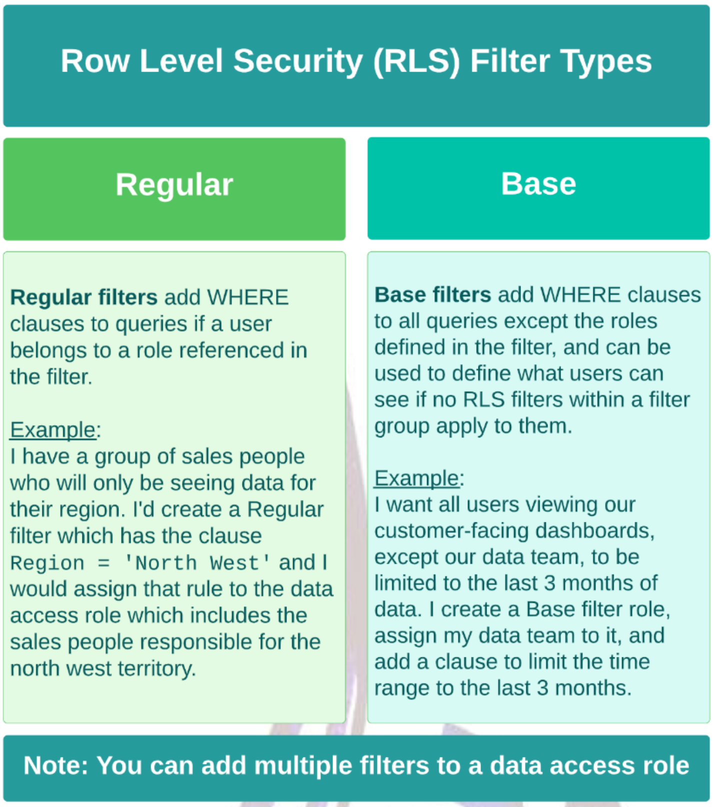
Yes

**Create A Group Key**

Save

Add a **Group Key** in the RLS form
Groups Keys which are the:
- **SAME** will be **OR'd**
- **DIFFERENT** will be **AND'd**

together when applied to the same DAR



Row Level Security (RLS) Filter Types

Regular

**Regular filters** add WHERE clauses to queries if a user belongs to a role referenced in the filter.

Example:
I have a group of sales people who will only be seeing data for their region. I'd create a Regular filter which has the clause `Region = 'North West'` and I would assign that rule to the data access role which includes the sales people responsible for the north west territory.

Base

**Base filters** add WHERE clauses to all queries except the roles defined in the filter, and can be used to define what users can see if no RLS filters within a filter group apply to them.

Example:
I want all users viewing our customer-facing dashboards, except our data team, to be limited to the last 3 months of data. I create a Base filter role, assign my data team to it, and add a clause to limit the time range to the last 3 months.

Note: You can add multiple filters to a data access role

### ☺ Explain Power BI's data refresh capabilities?

Power BI allows for scheduled data refreshments to keep the data updated in reports and dashboards. Data refresh can be configured in Power BI Service.

### ☺ What Will Happen If Many to Many Relationships Exist Between 2 Tables In Power BI In Data Modeling

In Power BI data modeling, many-to-many relationships can present several challenges that can impact the accuracy and performance of your reports and dashboards. Here are some key issues to consider:
**Filtering and Slicing Issues:** Filtering and slicing data may not behave as expected due to the potential for multiple matches. This can lead to unexpected results when visualizing data or creating measures.
**Cross Filtering:** In Power BI, relationships automatically enable cross-filtering between tables. With many-to-many relationships, cross-filtering can become more complex and may not provide the desired outcomes.
**Ambiguity in Aggregations:** Aggregations (such as sum, average, etc.) can become ambiguous when there are multiple related records. Power BI may not be able to determine which records to aggregate.
**Performance Impact:** Many-to-many relationships can have a performance impact, particularly with large datasets, as Power BI has to navigate through potentially complex relationships to fetch the data.
**DAX Measures and Calculations:** When working with many-to-many relationships, you may need to create more complex DAX measures and calculations to handle the ambiguity and provide accurate results.

### ☺ How do you handle many-to-many relationships in Power BI?

Many-to-many relationships can be handled by introducing a bridge table or using composite models and cross-filtering techniques. To handle many-to-many relationships in Power BI, you can use several strategies depending on the data model and the scenario:
**Bridge Table:** Create a bridge (junction) table that contains only the unique values from both sides of the relationship. This helps eliminate duplicate values and allows you to establish one-to-many relationships instead of many-to-many. For example, a table containing unique IDs for both tables (fact and dimension) can be created to link them.
**Direct Many-to-Many Relationships:** Power BI supports direct many-to-many relationships using a feature where you can define relationships with both tables as many. This method is useful when both

tables contain duplicate values for the same keys. However, it can introduce complexity and reduce performance. It's often recommended to use this sparingly as it can lead to ambiguous results [1].
**Bidirectional Filters:** You can enable bidirectional filtering to allow the filters to flow between tables. This can help handle many-to-many scenarios but can also complicate filtering logic.
Avoiding When Possible: It is generally best to avoid many-to-many relationships by redesigning the data model using techniques like denormalization or creating a bridge table, which improves performance and avoids calculation ambiguities

### ☺ What is the use of bookmarks in Power BI?

Bookmarks in Power BI provide a way to capture the current page of a report page, including any filters, slicers, visual configurations, or other custom settings that have been applied. This allows users to navigate between different views of the report with ease.
When a bookmark is created, it saves the following details:

- Active filters and slicers
- Current view of the visuals (including drill through states)
- Cross-highlighting between visuals
- Sorting order and page navigation setup

This feature is highly useful for building guided storylines, toggling between views, or creating custom navigation buttons in reports.
However, one must be aware that using bookmarks may reset the filters to whatever state was saved with the bookmark unless otherwise configured correctly

### ☺ How Bookmarks can be used in storytelling?

In Power BI and other data visualization tools, bookmarks are a powerful feature for storytelling. They allow users to capture a specific state of a report, including filters, slicers, and visual configurations, and use these bookmarks to craft a narrative or guide the audience through the data. Here's how bookmarks can enhance storytelling:

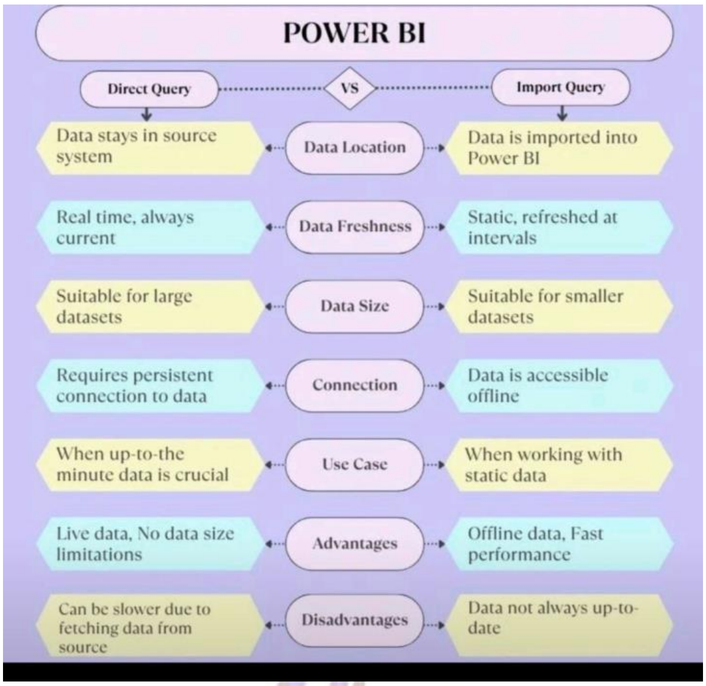
1. **Guided Navigation:** Bookmarks can be used to guide the viewer through a structured flow of insights, allowing them to focus on key data points.
2. **Scenario Comparison:** Multiple bookmarks can showcase different scenarios or "what-if" analyses, enabling easy comparison between data views.
3. **Highlight Key Insights:** Bookmarks can capture key visual moments in data storytelling, such as trends or anomalies, and present them in a clear and structured manner.
4. **Interactive Presentations:** With bookmarks, the presenter can toggle between views dynamically during presentations to illustrate different parts of the story, making the data more engaging.

This makes bookmarks an essential tool for building interactive and impactful data stories in reports and dashboards.

### ☻ What are the different connectivity modes in power BI? What is composite model? What is import and direct query mode? which one is better?

Power BI offers three primary connectivity modes to connect to data:

1. **Import Mode**: This is the default mode, where all data is loaded into Power BI's in-memory engine. It provides fast query performance but requires refreshing the dataset to reflect changes in the source data. This mode is ideal for small to medium datasets.
2. **Direct Query Mode**: In this mode, no data is imported into Power BI. Instead, queries are sent directly to the underlying data source every time a report or visualization is refreshed. This is suitable for large datasets or when real-time data is required, but it may result in slower performance depending on the data source.
3. **Live Connection**: This mode is used with certain data sources, such as SQL Server Analysis Services (SSAS). It allows Power BI to connect directly to the data source without importing data, providing real-time access to the underlying model.
4. **Composite Mode**: A hybrid approach that allows combining Import and Direct Query data sources in the same report, offering flexibility for complex scenarios.



POWER BI

| Direct Query | VS | Import Query |
|---|---|---|
| Data stays in source system | Data Location | Data is imported into Power BI |
| Real time, always current | Data Freshness | Static, refreshed at intervals |
| Suitable for large datasets | Data Size | Suitable for smaller datasets |
| Requires persistent connection to data | Connection | Data is accessible offline |
| When up-to-the minute data is crucial | Use Case | When working with static data |
| Live data, No data size limitations | Advantages | Offline data, Fast performance |
| Can be slower due to fetching data from source | Disadvantages | Data not always up-to-date |

### ☺ What is a composite model in Power BI?

A composite model in Power BI enables combining direct query and import modes in a single report, allowing flexibility in querying large datasets.

### ☺ Explain what incremental data load is and how to implement it in Power BI?

Incremental load refers to the practice of updating only the data that has changed instead of reloading the entire dataset, improving refresh performance.

### ☺ What is the Power BI REST API?

Power BI REST API allows developers to programmatically access Power BI dashboards, datasets, reports, and embed Power BI content into applications.

### ☺ How do you use Paginated Reports in Power BI?

Paginated Reports in Power BI are used to create pixel-perfect, printable reports that fit well across pages and handle large amounts of data.

### ☺ How does the XMLA endpoint in Power BI work?

The XMLA endpoint enables interaction with Power BI datasets via external tools for advanced data modeling, connecting directly to the data model through SQL Server Management Studio.

### ☺ Explain the difference between Power BI Pro and Premium.

- ✓ **Power BI Desktop:** Power BI Desktop is a free of cost development, authoring and publishing tool of Microsoft Power BI. It is a user interaction platform from where users can connect to multiple data sources, transform and clean data, visualize and create reports.
- ✓ **Power BI Pro:** Power BI Pro is a modern self-service BI having advanced features for collaboration, publishing, report sharing, and ad hoc analysis. It costs $9.99 per month per user.
- ✓ **Power BI Premium:** Power BI Premium is an enterprise BI solution with features and tools for advanced analytics, big data support, on-premises and cloud reporting, etc. It also provides dedicated cloud computing and storage facilities. It costs $ 20 per month per user.

Power BI Pro is an individual license that allows users to create and share reports, while Power BI Premium provides enterprise-level scalability, more capacity, and advanced features like AI and paginated reports.

### ☺ How do you create relationships between tables in Power BI?

In Power BI Desktop, go to the "Model" view, drag and drop fields from one table to another to create relationships based on common keys.

### ☺ What is the use of advanced analytics in Power BI?

Advanced analytics in Power BI allows users to create sophisticated visualizations using machine learning models and statistical algorithms, such as forecasting.

### ☺ How does Power BI work with AI visualizations?

Power BI integrates AI capabilities through built-in visualizations such as Key Influencers and Decomposition Tree, which help analyze and explain the data.

### ☺ What is Power BI Premium Per User?

Power BI Premium Per User provides the benefits of Premium, such as AI and paginated reports, to individual users at a lower cost compared to organizational Premium capacities.

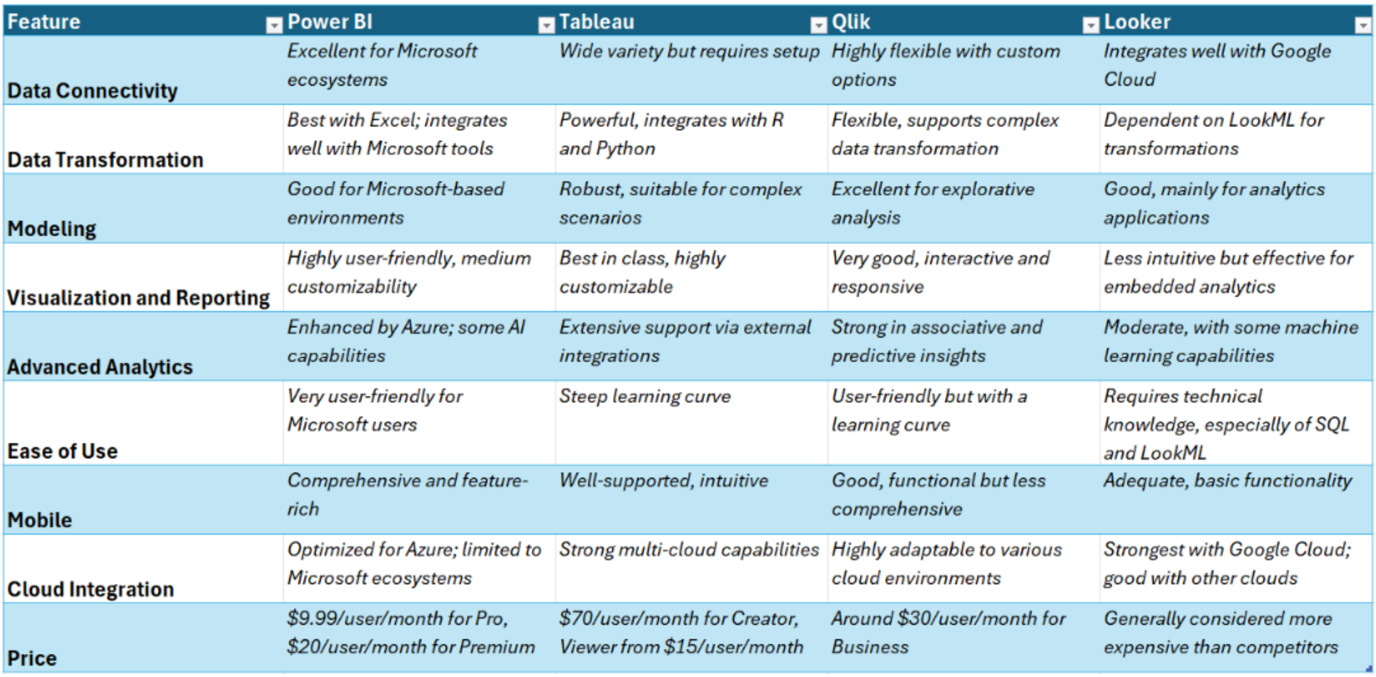
### ☺ What is sensitivity labeling in Power BI?

Sensitivity labels help in classifying and protecting data across Power BI, ensuring compliance with data governance policies.

### ☺ How do you enable and use Power BI dataflows?

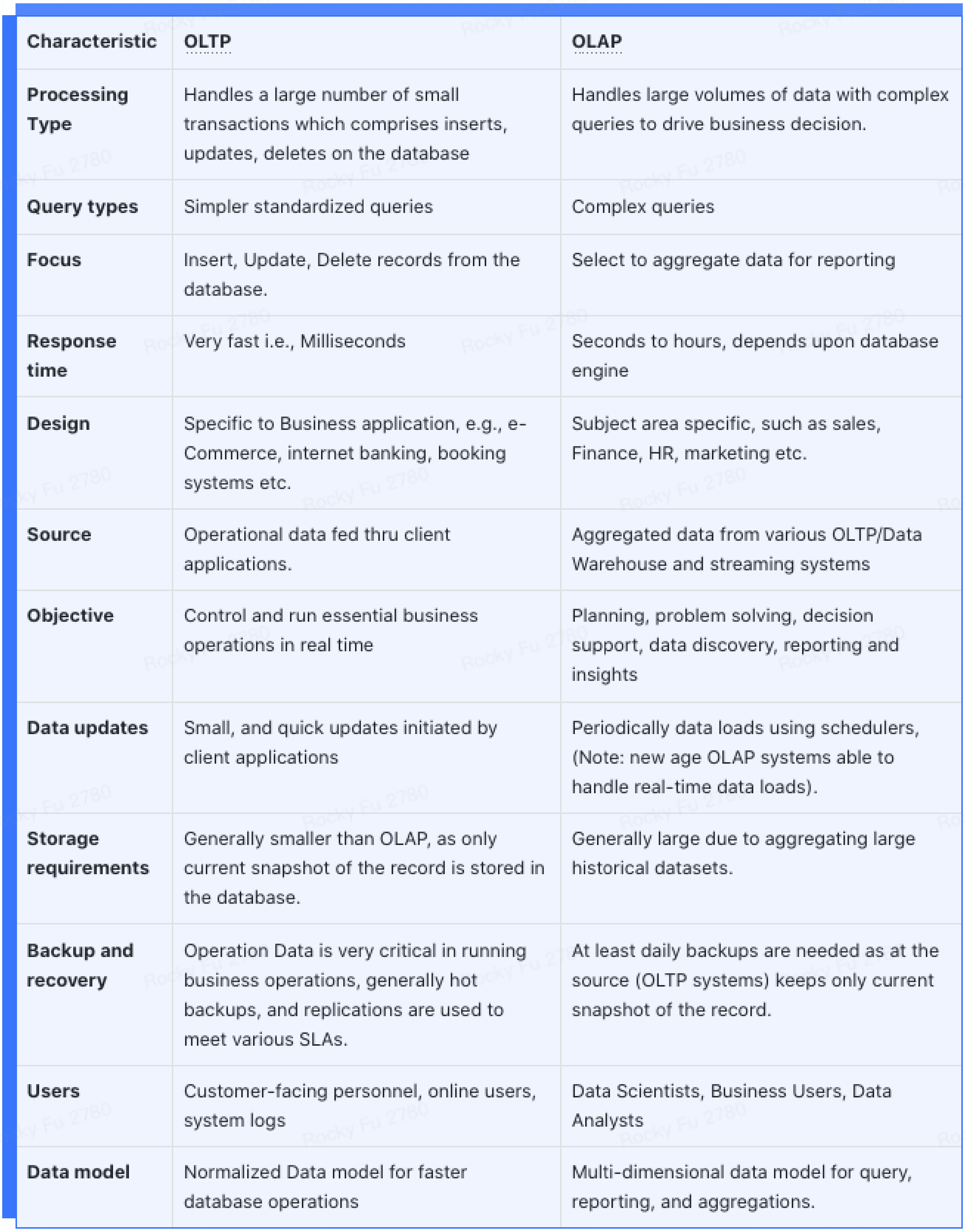
Dataflows are reusable, cloud-based data preparation entities in Power BI, allowing users to build ETL processes directly in the Power BI Service.

### ☺ Why power bi over other tools?



| Feature | Power BI | Tableau | Qlik | Looker |
|---|---|---|---|---|
| Data Connectivity | *Excellent for Microsoft ecosystems* | *Wide variety but requires setup* | *Highly flexible with custom options* | *Integrates well with Google Cloud* |
| Data Transformation | *Best with Excel; integrates well with Microsoft tools* | *Powerful, integrates with R and Python* | *Flexible, supports complex data transformation* | *Dependent on LookML for transformations* |
| Modeling | *Good for Microsoft-based environments* | *Robust, suitable for complex scenarios* | *Excellent for explorative analysis* | *Good, mainly for analytics applications* |
| Visualization and Reporting | *Highly user-friendly, medium customizability* | *Best in class, highly customizable* | *Very good, interactive and responsive* | *Less intuitive but effective for embedded analytics* |
| Advanced Analytics | *Enhanced by Azure; some AI capabilities* | *Extensive support via external integrations* | *Strong in associative and predictive insights* | *Moderate, with some machine learning capabilities* |
| Ease of Use | *Very user-friendly for Microsoft users* | *Steep learning curve* | *User-friendly but with a learning curve* | *Requires technical knowledge, especially of SQL and LookML* |
| Mobile | *Comprehensive and feature-rich* | *Well-supported, intuitive* | *Good, functional but less comprehensive* | *Adequate, basic functionality* |
| Cloud Integration | *Optimized for Azure; limited to Microsoft ecosystems* | *Strong multi-cloud capabilities* | *Highly adaptable to various cloud environments* | *Strongest with Google Cloud; good with other clouds* |
| Price | *$9.99/user/month for Pro, $20/user/month for Premium* | *$70/user/month for Creator, Viewer from $15/user/month* | *Around $30/user/month for Business* | *Generally considered more expensive than competitors* |

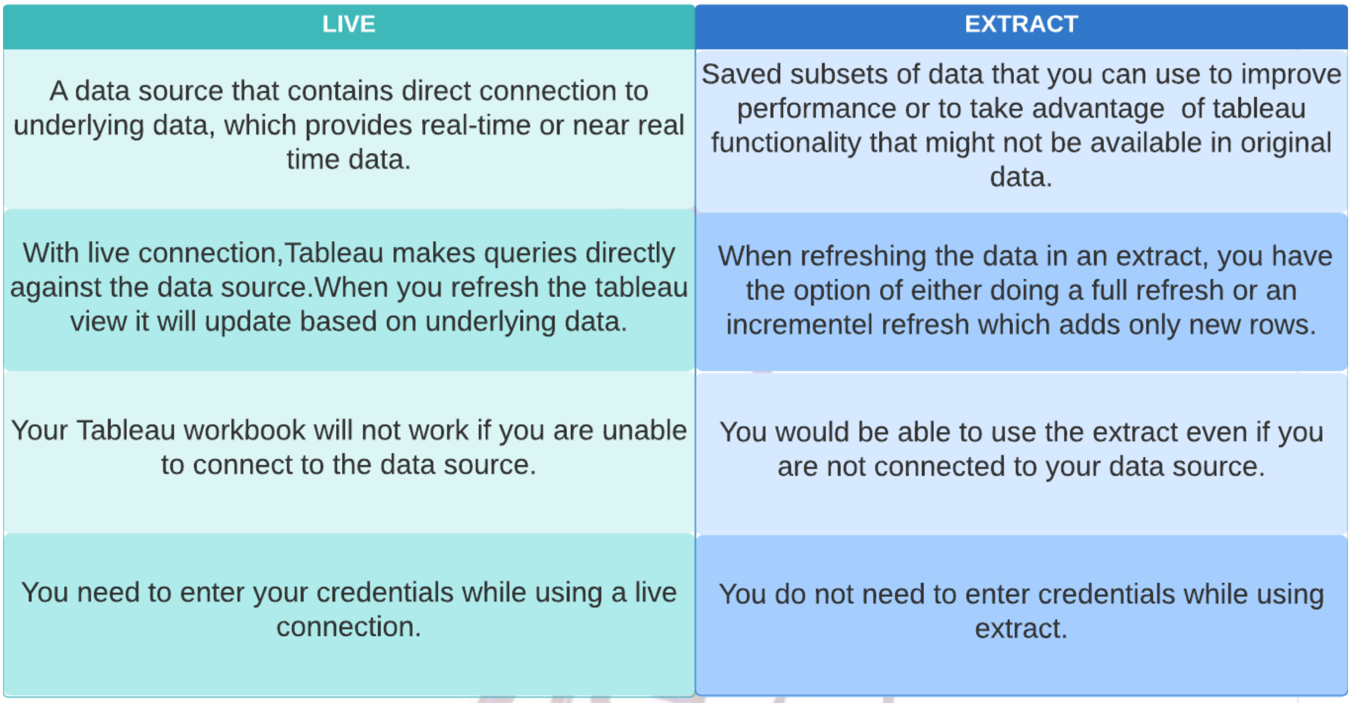
## What is the difference between OLAP and OLTP systems?



| Characteristic | OLTP | OLAP |
|---|---|---|
| Processing Type | Handles a large number of small transactions which comprises inserts, updates, deletes on the database | Handles large volumes of data with complex queries to drive business decision. |
| Query types | Simpler standardized queries | Complex queries |
| Focus | Insert, Update, Delete records from the database. | Select to aggregate data for reporting |
| Response time | Very fast i.e., Milliseconds | Seconds to hours, depends upon database engine |
| Design | Specific to Business application, e.g., e-Commerce, internet banking, booking systems etc. | Subject area specific, such as sales, Finance, HR, marketing etc. |
| Source | Operational data fed thru client applications. | Aggregated data from various OLTP/Data Warehouse and streaming systems |
| Objective | Control and run essential business operations in real time | Planning, problem solving, decision support, data discovery, reporting and insights |
| Data updates | Small, and quick updates initiated by client applications | Periodically data loads using schedulers, (Note: new age OLAP systems able to handle real-time data loads). |
| Storage requirements | Generally smaller than OLAP, as only current snapshot of the record is stored in the database. | Generally large due to aggregating large historical datasets. |
| Backup and recovery | Operation Data is very critical in running business operations, generally hot backups, and replications are used to meet various SLAs. | At least daily backups are needed as at the source (OLTP systems) keeps only current snapshot of the record. |
| Users | Customer-facing personnel, online users, system logs | Data Scientists, Business Users, Data Analysts |
| Data model | Normalized Data model for faster database operations | Multi-dimensional data model for query, reporting, and aggregations. |

### ☺ Explain the difference between live connection and extract connection in Tableau?

**LIVE AND EXTRACT CONNECTION**



| LIVE | EXTRACT |
|---|---|
| A data source that contains direct connection to underlying data, which provides real-time or near real time data. | Saved subsets of data that you can use to improve performance or to take advantage of tableau functionality that might not be available in original data. |
| With live connection,Tableau makes queries directly against the data source.When you refresh the tableau view it will update based on underlying data. | When refreshing the data in an extract, you have the option of either doing a full refresh or an incrementel refresh which adds only new rows. |
| Your Tableau workbook will not work if you are unable to connect to the data source. | You would be able to use the extract even if you are not connected to your data source. |
| You need to enter your credentials while using a live connection. | You do not need to enter credentials while using extract. |

### ☺ What is the need of Data Master Table?

Data master tables are crucial for managing and organizing key business information across various applications and systems. Here are the main reasons for their importance:

1. **Centralized Data Management**: Master tables serve as a single source of truth for critical business data, ensuring consistency and accuracy across the organization
2. **Data Integrity**: By consolidating data into master tables, organizations can maintain high data quality and integrity, reducing redundancy and minimizing errors in data entry
3. **Enhanced Reporting and Analysis**: Master tables facilitate efficient reporting and analysis by providing a structured format that integrates data from multiple sources. This aids in decision-making processes
4. **Support for Relationships**: They help in establishing relationships between different entities in a database, enabling more complex queries and data manipulations, particularly in master-detail relationships.

In summary, data master tables are essential for effective data governance, operational efficiency, and strategic analytics.

### ☺ What is the role of Metadata in BI environment?

Metadata is essential to the functionality of Business Intelligence (BI) systems, where it ensures efficient data management and analysis. Below are key types and roles of metadata in BI:

1. **Structural Metadata**: It outlines how data is organized, providing a framework for data warehouses, lakes, and other repositories. This helps maintain the structure and flow of data within BI environments, ensuring smooth data processing and management.
2. **Descriptive Metadata**: Acts as a guide to data assets by tagging and categorizing them. It makes data accessible and searchable, which is crucial in BI systems where users must quickly find and use relevant data.
3. **Administrative Metadata**: Manages data governance by recording information such as data ownership, access controls, and compliance rules. This type of metadata ensures the security of data and adherence to regulations within BI systems.
4. **Data Integration and Quality**: Metadata helps coordinate data during integration (ETL/ELT

processes), ensuring the accuracy, consistency, and reliability of data used in BI analysis.

5. **Security and Compliance**: Administrative metadata tracks data access and usage, enhancing security and regulatory compliance, vital in BI environments that handle sensitive data.

### ☺ Can you explain the concept of KPI's and how they are used in business analysis?

Key Performance Indicators (KPIs) are measurable values used to assess how effectively a company, project, or individual is achieving key business objectives. In the context of **business analysis**, KPIs are critical in tracking the success of projects, helping analysts gauge the performance and value of their activities.

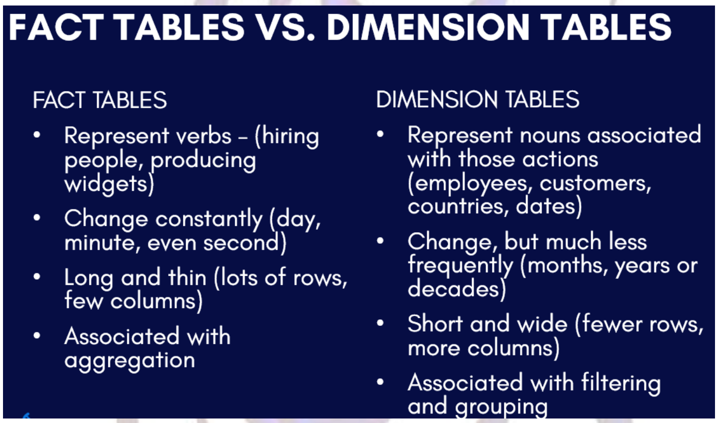
Here's how KPIs are used in business analysis:

1. **Measuring Project Success**: KPIs evaluate the impact of business analysis on project goals by measuring aspects like stakeholder satisfaction, requirement clarity, and the number of completed tasks.
2. **Performance Monitoring**: Business analysts use KPIs to monitor their own performance, such as the number of issues resolved, or the ratio of requirements gathered to project success.
3. **Decision Making**: KPIs help in making data-driven decisions by providing insights into areas needing improvement or attention, such as delays in project deliverables or gaps in stakeholder communication.

KPIs offer a structured way to measure success and adjust strategies in real time to ensure business goals are met.

### ☺ What is Fact and Dimension tables?

A Fact table contains measurable, quantitative data used in analysis. Dimension tables hold descriptive attributes related to fact data, helping in filtering and categorizing it



FACT TABLES VS. DIMENSION TABLES

FACT TABLES

- Represent verbs – (hiring people, producing widgets)
- Change constantly (day, minute, even second)
- Long and thin (lots of rows, few columns)
- Associated with aggregation

DIMENSION TABLES

- Represent nouns associated with those actions (employees, customers, countries, dates)
- Change, but much less frequently (months, years or decades)
- Short and wide (fewer rows, more columns)
- Associated with filtering and grouping

### ☺ Why is your Power BI dataset refresh failing intermittently?

Possible Reasons:
- Intermittent connectivity issues with the data source.
- Dataset size exceeding the available memory during refresh.
- Resource contention on the server hosting the data source.
- Inconsistent data schema (e.g., column renaming, data type changes).
- The data source is temporarily unavailable during scheduled refresh.

### ☺ Why are your Power BI visuals not updating after the data refresh?

Possible Reasons:
- Cached data is being used instead of the refreshed data.
- The data refresh did not complete successfully.
- Filters or slicers applied are restricting the data.
- Relationships in the data model are not correctly defined or are inactive.
- Data was refreshed, but the report was not republished.

### ☺ How to configure Gateway?

To configure a Power BI data gateway, follow these steps:

1. **Install the Gateway**:
   - Download and install the on-premises data gateway from the Power BI website.
   - Choose either the **Standard** or **Personal Mode** gateway, depending on your needs.
2. **Log in to Power BI**:
   - Sign in to the Power BI Service.
   - Go to the **Settings** (gear icon) in the top-right corner and select **Manage Connections and Gateways**.
3. **Add a Data Source**:
   - In the **Manage Gateways** section, add a data source by selecting your gateway.
   - Enter the connection details (server name, database name, credentials) to link your data source to the gateway.
4. **Configure Scheduled Refresh**:
   - Ensure that the gateway is connected to your dataset in the **Datasets** settings.
   - Under **Gateway Connection**, link the dataset to the gateway to schedule data refreshes.

### ☺ How to optimize Report Performance?

To optimize report performance in Power BI, follow these best practices:

1. **Optimize the Data Model**: Keep your data model as simple as possible. Remove unnecessary columns, reduce the number of calculated columns, and avoid many-to-many relationships.
2. **Use Aggregations**: Pre-aggregate data at the source level to reduce the volume of data being processed, which can speed up report rendering.
3. **Limit Visuals and Calculations**: Reduce the number of visuals and complex DAX calculations in the report. Heavy visuals can increase the load time of reports.
4. **Optimize Queries**: Use the Power Query Editor to simplify and clean your queries before loading them into the model. Avoid complex transformations that could slow down the loading process.
5. **Use Direct Query Wisely**: Direct Query can slow down reports if the source is not optimized. If possible, use Import mode for faster performance

### ☺ How to share multiple Reports in single go?

To share multiple Power BI reports efficiently, you can use the following methods:

1. **Share via Workspaces**: The easiest way to share multiple reports at once is to add users to a shared workspace. When you add users to a workspace, they gain access to all the reports and dashboards in that workspace. Assign appropriate roles (Viewer, Contributor, etc.) depending on their needs.
2. **App Creation**: Create a Power BI app, which bundles multiple reports and dashboards together. You can publish the app and share it with users, granting them access to all the included content in one go.
3. **External Sharing**: If you need to share with users outside your organization, Power BI allows sharing via email. You can send links to multiple reports or share via the app by selecting "Share" from the report or dashboard

### ☺ Different Types of Roles in Power BI? Suppose someone with contributor role who is not the part of your team Can able to view that data and dashboard. if yes what should be done? What are the roles available in workspace?

Power BI provides four main roles within workspaces, each with different levels of access:

1. **Admin**: Full control over the workspace. Admins can add/remove users, modify content, and manage permissions.
2. **Member**: Members can create, edit, and share content. They can also add other members but cannot change the overall workspace settings.
3. **Contributor**: Contributors can add new content, such as reports and dashboards, and interact with the workspace's existing content. They cannot manage users or modify permissions.

4. **Viewer**: This is a read-only role. Viewers can access and view reports and dashboards but cannot edit or share them.

These roles ensure secure and organized collaboration within Power BI workspaces.

### ☺ How do you stay updated on the latest trend and technologies in business intelligence?

1. Read BI-Specific Publications and Blogs
2. Attend BI Conferences and Webinars
3. Follow BI Thought Leaders on Social Media
4. Online Learning Platforms
5. Engage in Professional Communities

### ☻ Suppose Scheduled refresh took yesterday 10 min to refresh & today took more 30 min, also all internet, credentials and all other settings are ok then what was reason for it?

Several factors could explain why the scheduled refresh took longer today despite everything being correctly set up. Here are the most common reasons:

1. **Data Volume Increase**: If there is more data to refresh today compared to yesterday, such as new data rows or increased complexity, it can increase the refresh time.
2. **System Load**: The system or server where the refresh is running could have been busier today due to higher demand or background processes running simultaneously, leading to longer processing times.
3. **Backend Optimization**: There might be changes in the backend system or the service running the refresh, causing variations in performance.
4. **Data Source Performance**: The performance of the data source itself (e.g., SQL server or cloud service) may vary, which can slow down data retrieval for Power BI.

Even if the credentials, settings, and internet are fine, the above factors could impact the refresh duration on different days.

### ☻ Your PBIX file is not getting published, what are the reasons?

1. **Cache Issues**: Sometimes, Power BI cache data can cause problems. Clearing the cache in Power BI Desktop may help resolve this issue.
2. **Data Load or Connection Problems**: If there’s an issue with the connection to the data source or if the data is too large or complex, publishing may fail.
3. **Browser Cache**: Publishing issues can also arise due to old data being stored in your browser cache. Clearing your browser’s cache and trying again might fix the problem.
4. **Software Version**: Using an outdated version of Power BI Desktop may prevent you from publishing. Ensure that you are using the latest version of Power BI.
5. **Images in Reports**: Sometimes, reports with embedded images can fail to be published. Removing or replacing them could resolve the issue.
6. file with same name open two times in Power BI Desktop
7. Internet Issues
8. Don't have permission of workspace
9. Outdated Version of PBI Desktop.
10. Refresh going at Backend and data is too Huge in cage of Direct Query
11. If your account is deactivated or restricted
12. Heavy Cookies & Caches in PBI Desktop

### ☺ Why is your Power BI report taking too long to load in the Power BI Service?

There are several reasons why your Power BI report may be taking too long to load in the Power BI Service:

1. **Data Model Size**: Large or unoptimized data models can slow downloading times. Reducing data size, removing unnecessary columns, and optimizing measures can help.
2. **Network Connection**: A slow or unreliable network can significantly impact report loading times. Ensure that your internet connection is stable.
3. **Data Source Performance**: The performance of the data source itself affects query times. Slow queries from the data source will result in slower report rendering.
4. **Gateway Location**: If you are using an on-premises gateway, its location relative to the data sources and Power BI service can impact performance. Ensure that the gateway is properly configured and close to the data source.
5. **Too Many Visuals**: Overloading the report with complex or numerous visuals can increase load times. Consider reducing the number of visuals or optimizing them
6. Inefficient DAX queries or poorly optimized data model.
7. Using too many visuals in a single report.
8. Heavy use of Direct Query with slow data sources.
9. Excessive use of high-cardinality columns in slicers and filters.
10. Insufficient memory allocated on the Power BI Service.

### ☺ How do you handle data type conversion in Power Query, especially when dealing with date and time data?

In Power Query, handling data type conversion, particularly for date and time, involves a few key steps:

1. **Identify Data Type**: Check the current data type of your column. If it shows as "Text" or an incorrect format, you'll need to convert it.
2. **Change Data Type**: Right-click on the column header, select "Change Type," and then choose "Date," "Time," or "Date/Time" as needed. This helps in standardizing the format.
3. **Using M Code**: If the format is unusual (e.g., strings like "Sep 10 2005 12:03:34:716 PM"), you may need to transform it using custom M code. You can apply functions like Date. FromText or DateTime.FromText to convert strings into the desired format.
4. **Handling Errors**: If conversion results in errors (e.g., invalid dates), check your source data for inconsistencies and clean it up before conversion.
5. **Data Type Summary**: Always keep in mind that properly defined data types facilitate better data modeling and analysis

### ☺ What is column profiling?

Column profiling is a method used in data profiling to analyze individual columns within a dataset. It helps assess the structure and quality of the data by identifying the data type, data format, range of values, unique values, and frequency distribution of values within a column. This process is essential for understanding how data is distributed, spotting inconsistencies, and ensuring data quality. It is commonly used in data cleaning, transformation, and validation processes to ensure the dataset is reliable for analysis or reporting

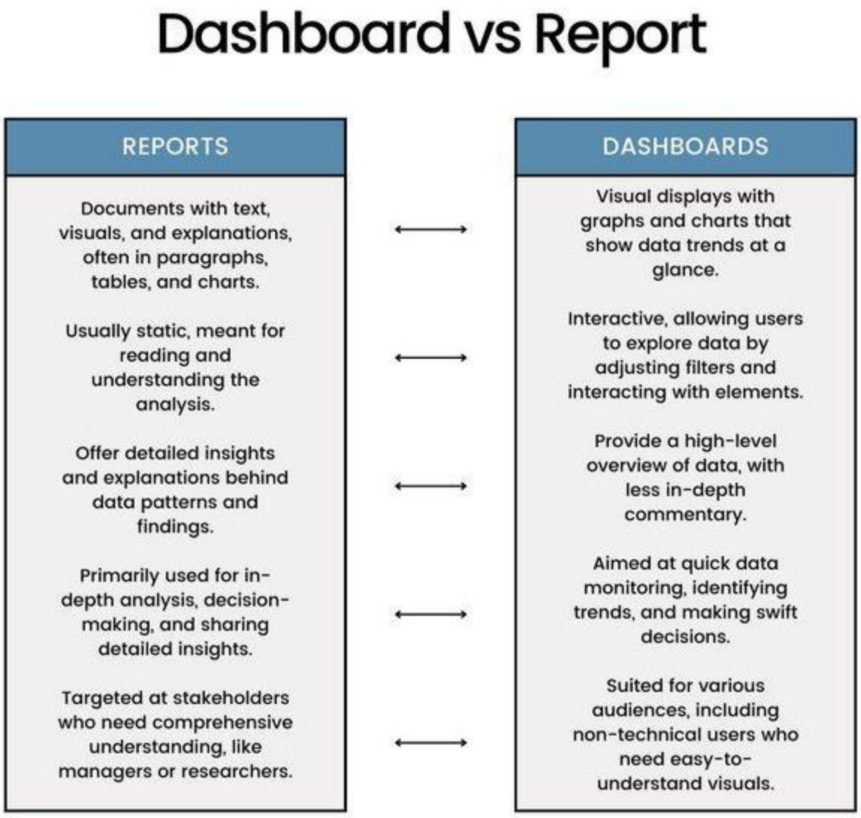
### ☺ You have data from diff data sources like one is from google sheet, other is from SQL server and other is from previous project like dataset what will the approach to bring in power bi desktop?

To bring data from different sources (Google Sheets, SQL Server, and a previous Power BI dataset) into Power BI Desktop, follow these steps:

1. Connect to Google Sheets:
   - Use the Google Sheets connector available in Power BI. Navigate to Home > Get Data > More > Google Sheets, and authenticate using your Google account. You can also use Connected Sheets to link with BigQuery if needed.
2. Connect to SQL Server:

   - In Power BI, go to Home > Get Data > SQL Server. Enter the server and database details and choose the tables or views you need from SQL Server. You may need to authenticate using your credentials.
3. Load Previous Project Dataset:
   - For the previous Power BI project dataset, if it's available in Power BI Service, use the Power BI datasets connector under Get Data > Power BI datasets. This allows you to connect to and combine data from existing reports and datasets.
4. Data Modeling:
   - Once all data sources are imported, model the data by creating relationships between tables as needed. You can also clean and transform the data using Power Query to ensure consistency across datasets.

### ☺ Difference between dashboard and report?



Dashboard vs Report

| REPORTS | | DASHBOARDS |
|---|---|---|
| Documents with text, visuals, and explanations, often in paragraphs, tables, and charts. | ⟷ | Visual displays with graphs and charts that show data trends at a glance. |
| Usually static, meant for reading and understanding the analysis. | ⟷ | Interactive, allowing users to explore data by adjusting filters and interacting with elements. |
| Offer detailed insights and explanations behind data patterns and findings. | ⟷ | Provide a high-level overview of data, with less in-depth commentary. |
| Primarily used for in-depth analysis, decision-making, and sharing detailed insights. | ⟷ | Aimed at quick data monitoring, identifying trends, and making swift decisions. |
| Targeted at stakeholders who need comprehensive understanding, like managers or researchers. | ⟷ | Suited for various audiences, including non-technical users who need easy-to-understand visuals. |

### ☺ How will you refresh single table in Power BI?

To refresh a single table in Power BI, follow these methods:

1. Using XMLA Endpoints (Premium Feature):
   - If you're using Power BI Premium (or Premium per User), you can refresh a single table by connecting to the dataset via XMLA endpoints.
   - You can use SQL Server Management Studio (SSMS) to connect to the Power BI service, expand the dataset, and refresh a specific table by selecting and right-clicking on it, then choosing Process Table.
2. Using SSMS (SQL Server Management Studio):
   - In Premium environments, you can refresh one table by connecting to the dataset via SSMS. Navigate to the desired table in the dataset, right-click, and select Process Table.
3. Power BI Desktop:
   - In Power BI Desktop, refreshing a single table directly is not supported. You need to refresh the entire dataset or use DAX Studio to refresh data in a specific query

### ☺ You want to give permission to 50 users in power bi how will you do that?

To give permission to 50 users in Power BI efficiently, follow these steps:

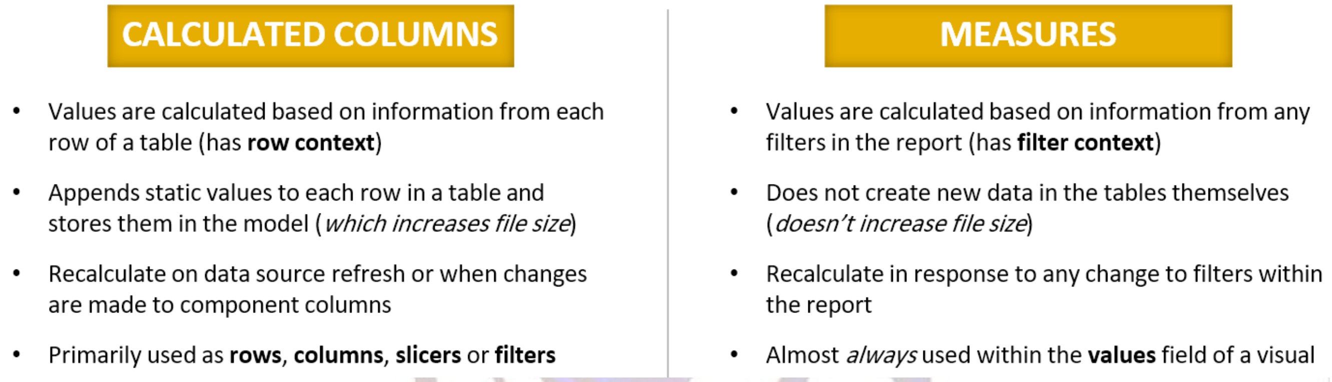
1. Use Security Groups:
    - Instead of manually adding 50 users individually, create or use an Azure Active Directory (AAD) security group. Add all 50 users to this group.
    - In Power BI, you can then share the report or dashboard by granting access to the entire group, reducing the manual effort of adding users one by one.
2. Manage Permissions:
    - Go to the Permissions section of the report or dashboard in Power BI.
    - From there, enter the name of the group in the Direct Access or Manage Permissions pane, then select Grant access to share the report or dashboard with the entire group.
3. Row-Level Security (Optional):
    - If you need to apply specific Row-Level Security (RLS) for these users, set up RLS roles in Power BI and assign the group to a role to control the data they can see.

### ☺ What are Calculated Columns in Power BI?

A Calculated Column is a new column that you create using DAX formulas based on data from other columns in a table.

### ☺ Explain the difference between calculated columns and measures in Power BI?

Calculated columns is a column added to a table, computed row by row, store data and are recalculated as soon as the data changes while a measure is a formula applied to a set of data, providing a dynamic calculation based on the context.



| CALCULATED COLUMNS | MEASURES |
|---|---|
| Values are calculated based on information from each row of a table (has **row context**) | Values are calculated based on information from any filters in the report (has **filter context**) |
| Appends static values to each row in a table and stores them in the model (*which increases file size*) | Does not create new data in the tables themselves (*doesn't increase file size*) |
| Recalculate on data source refresh or when changes are made to component columns | Recalculate in response to any change to filters within the report |
| Primarily used as **rows**, **columns**, **slicers** or **filters** | Almost *always* used within the **values** field of a visual |

### ☺ What is DAX in Power BI, and why is it important?

DAX (Data Analysis Expressions) is a formula language used for creating custom calculations in Power BI. It is important as it enables users to create sophisticated measures and calculated columns.

### ☺ How will you join two tables with the help of DAX function?

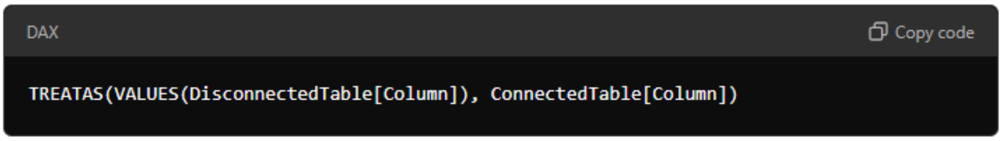
In DAX, joining two tables can be done using explicit join functions or by leveraging relationships between tables. Here are the two primary methods:

1. Using Relationships: If the tables are related in the data model, DAX can implicitly perform a join by referencing fields from both tables in a measure or column calculation. This is the most common and efficient way to join tables in DAX.
2. Using Join Functions: DAX provides explicit functions for performing joins:
    - **NATURALINNERJOIN**: This performs an inner join between two tables based on matching column names.
    - **NATURALLEFTOUTERJOIN**: This performs a left outer join between two tables.

For more complex joins, functions like GENERATE and GENERATEALL can be used to create combinations of rows from two tables

### ☺ Suppose you have 2 disconnected tables what DAX function u will send to Get the data from another table?

To get data from one table and apply it to another disconnected table, you can use the TREATAS function in DAX. The TREATAS function transfers a filter context from one table to another without requiring an explicit relationship between them. This is useful when you have two disconnected tables, and you want to filter or create a measure based on the values of one table applied to the other.



```
TREATAS(VALUES(DisconnectedTable[Column]), ConnectedTable[Column])
```

### ☺ Difference between DATA TABLES VS. LOOKUP TABLES

Models generally contain two types of tables: data (or “fact”) tables, and lookup (or “dimension”) tables

- Data tables contain measurable values or metrics about the business (quantity, revenue, pageviews, etc.)
- Lookup tables provide descriptive attributes about each dimension in your model (customers, products, etc.)

### ☺ Difference between lookup and related function?

The LOOKUPVALUE and RELATED functions in DAX both retrieve values from other tables but differ in how and when they should be used:

1. **LOOKUPVALUE:**
   - Purpose: Used to retrieve a value from another table without needing an established relationship between the tables.
   - Use case: It allows for multiple search criteria, making it flexible in more complex scenarios.
   - Performance: Slower when a relationship exists between the tables because it manually performs the lookup.
2. **RELATED**:
   - Purpose: Fetches values from another table, but only when there is a defined relationship between the current and related table.
   - Use case: Best used when there is already a defined relationship, providing faster performance than LOOKUPVALUE in these cases.

In summary, LOOKUPVALUE is for flexible lookups without relationships, while RELATED is more efficient when relationships are in place.

### ☺ Difference between all and remove function?

The ALL and REMOVEFILTERS functions in Power BI both clear filters, but they have different purposes and use cases:

1. ALL:
   - Function Type: Can be used both as a table function and a filter modifier in CALCULATE.
   - Purpose: Removes filters from the specified columns, tables, or the entire dataset, but can also return a table for calculations.
   - Use Case: Typically used in measures that require ignoring certain filters and performing calculations on the full dataset or specific columns.
2. REMOVEFILTERS:
   - Function Type: Primarily used as a CALCULATE modifier.
   - Purpose: Removes filters only on the specified columns or tables without returning a table.
   - Use Case: Used mainly to clear filters without the added functionality of returning tables like ALL does.

Key Difference: ALL can return a table and modify filters, while REMOVEFILTERS is used strictly to remove filters without returning a table

### ☺ What is ALL and ALLEXCEPT function?

The ALL and ALLEXCEPT functions in Power BI (DAX) are filter-modifying functions that work within the context of data calculations, helping control which rows or columns should be included or excluded.

1. ALL: The ALL function removes all filters from a table or column, allowing a calculation to consider every row or value, regardless of any filters applied. For example, ALL(Table) will ignore all filters on the table during calculation.
2. ALLEXCEPT: The ALLEXCEPT function removes filters from all columns in a table except for the specified columns. This means you can keep filters on certain columns while ignoring others. For example, ALLEXCEPT (Table, Column1) will ignore filters on all columns except Column1.

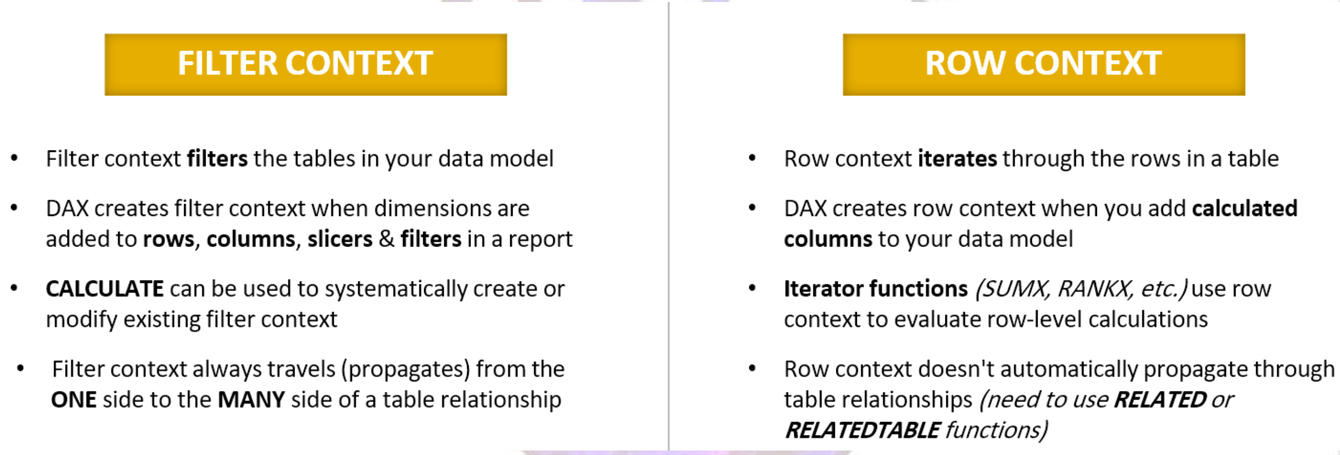
These functions are useful when you need to control how filters affect your data, such as in complex calculations with CALCULATE.

### ☺ Difference between row context and filter context?

In DAX, Row Context and Filter Context are key concepts that impact how calculations are performed.

1. **Row Context:**
   - Definition: Row context refers to the current row in a table when evaluating a formula. It exists naturally when iterating over rows in a table (e.g., in calculated columns or using functions like SUMX).
   - Behavior: It affects each row individually, and the calculation is based on the current row's data.
2. **Filter Context:**
   - Definition: Filter context refers to the filters applied to a table or column when calculating a measure. These filters can be applied by slicers, page filters, or manually in the DAX query.
   - Behavior: It affects the entire dataset based on the specified filters and can propagate through relationships between tables.

Key Difference: Row context operates on one row at a time in a table, while filter context applies to multiple rows or an entire dataset, allowing filtered data to affect calculations.



**FILTER CONTEXT**

- Filter context **filters** the tables in your data model
- DAX creates filter context when dimensions are added to **rows**, **columns**, **slicers** & **filters** in a report
- **CALCULATE** can be used to systematically create or modify existing filter context
- Filter context always travels (propagates) from the **ONE** side to the **MANY** side of a table relationship

**ROW CONTEXT**

- Row context **iterates** through the rows in a table
- DAX creates row context when you add **calculated columns** to your data model
- **Iterator functions** *(SUMX, RANKX, etc.)* use row context to evaluate row-level calculations
- Row context doesn't automatically propagate through table relationships *(need to use **RELATED** or **RELATEDTABLE** functions)*

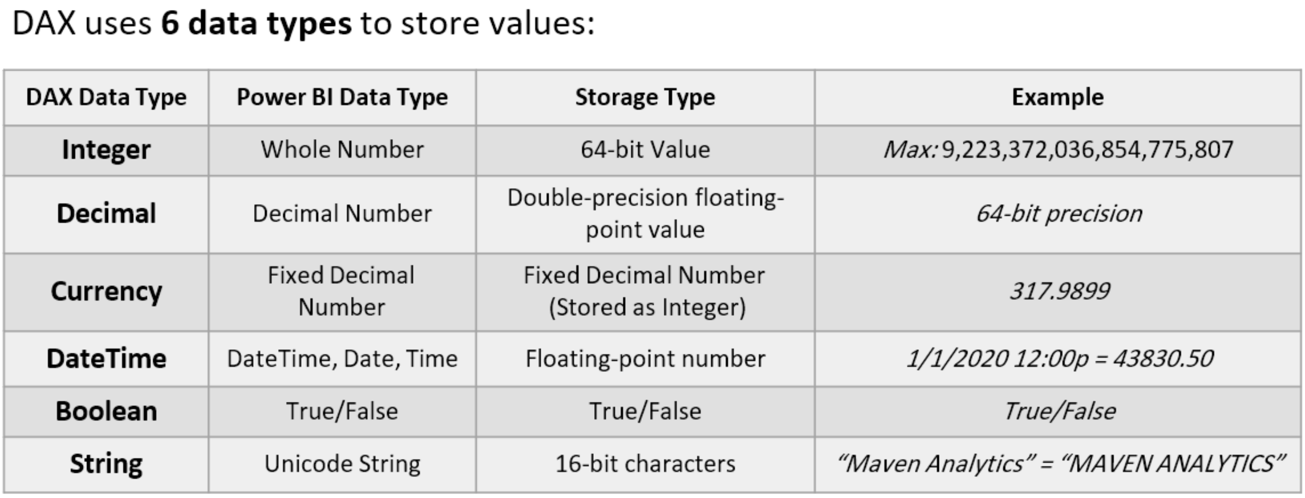
### ☺ How will optimize the Dax code?

To optimize DAX code in Power BI, you can follow these key strategies:

1. Avoid Calculated Columns:
   - Instead of using calculated columns, try using measures as they are computed on demand, making them more efficient.
2. Use Variables:
   - Use variables to store the result of repeated calculations or expressions. This reduces the number of calculations and improves performance.
3. Minimize Row-by-Row Calculations:

- Avoid iterating functions like SUMX, FILTER, and EARLIER where possible, as they can slow down performance. Use aggregation functions like SUM, AVERAGE, etc., which are optimized for performance.

4. Reduce Redundancy:
   - Simplify and reduce repeated logic in your DAX expressions. Consolidate multiple expressions into fewer steps to make the code cleaner and faster.
5. Optimize Filter Context:
   - Ensure that filters are applied efficiently, and use functions like KEEPFILTERS or REMOVEFILTERS appropriately to manage context transitions



DAX uses **6 data types** to store values:

| DAX Data Type | Power BI Data Type | Storage Type | Example |
|---|---|---|---|
| **Integer** | Whole Number | 64-bit Value | *Max:* 9,223,372,036,854,775,807 |
| **Decimal** | Decimal Number | Double-precision floating-point value | *64-bit precision* |
| **Currency** | Fixed Decimal Number | Fixed Decimal Number (Stored as Integer) | *317.9899* |
| **DateTime** | DateTime, Date, Time | Floating-point number | *1/1/2020 12:00p = 43830.50* |
| **Boolean** | True/False | True/False | *True/False* |
| **String** | Unicode String | 16-bit characters | *"Maven Analytics" = "MAVEN ANALYTICS"* |

### ☺ How do you create a semantic model in Power BI?

Creating a semantic model in Power BI involves several steps to ensure that data is organized, accessible, and meaningful for analysis. Here's a simplified process:

1. **Connect to Data Sources**: Import or connect to your required data sources using methods like Import mode, Direct Query, or composite models.
2. **Data Modeling**: Use a star schema approach to simplify the model. This involves creating fact tables (which contain measurable, quantitative data) and dimension tables (which contain descriptive attributes related to the facts).
3. **Define Relationships**: Establish relationships between tables to enable effective data interaction. Power BI allows you to create one-to-many or many-to-many relationships based on keys.
4. **Create Measures and Calculated Columns**: Use DAX (Data Analysis Expressions) to create measures for aggregating data and calculated columns for additional data manipulation.
5. **Visualizations**: Finally, create reports and dashboards using the semantic model to visualize insights effectively

### ☺ Why use an additional date table if an existing table has a date field?

Using an additional date table in Power BI, even when an existing table has a date field, is advantageous for several reasons:

1. **Enhanced Date Management**: A dedicated date table provides better control over date-related calculations and functions. It allows for more complex date filtering, grouping, and aggregating, improving the overall analysis.
2. **Avoiding Implicit Relationships**: Power BI often creates hidden date tables when date fields are present in multiple tables. This can lead to confusion and unexpected results. A dedicated date table simplifies the model and prevents this issue.
3. **Custom Scenarios**: A separate date table offers flexibility for custom date scenarios, such as fiscal years or specific date ranges, which might not align with the standard calendar.
4. **Performance Optimization**: Using a single date table can enhance performance, especially in large models, by reducing the memory footprint compared to multiple date fields scattered across various tables

### ☺ How would you manage slowly changing dimensions in a Power BI data model?

To manage Slowly Changing Dimensions (SCD) in Power BI, follow these steps:

1. Use the Star Schema:
   - Implement a star schema with fact and dimension tables. Power BI handles slowly changing dimensions through proper modeling, which involves joining fact tables with dimension tables that track changes over time.
2. SCD Type 1:
   - In Type 1, the old data is overwritten with the new data. You can update the dimension tables using SQL or ETL processes, then refresh the model in Power BI to reflect the changes.
3. SCD Type 2:
   - For Type 2, maintain historical data by adding new rows with start and end dates for the dimensional changes. In Power BI, you can track the changes by loading both historical and current records and using DAX or Power Query to filter based on effective dates.
4. Temporal Tables:
   - Another option is using temporal tables in SQL Server, which automatically tracks historical data changes. You can then import these tables into Power BI for better management of SCD

### ☺ Difference between Star Schema and Snowflake Schema?

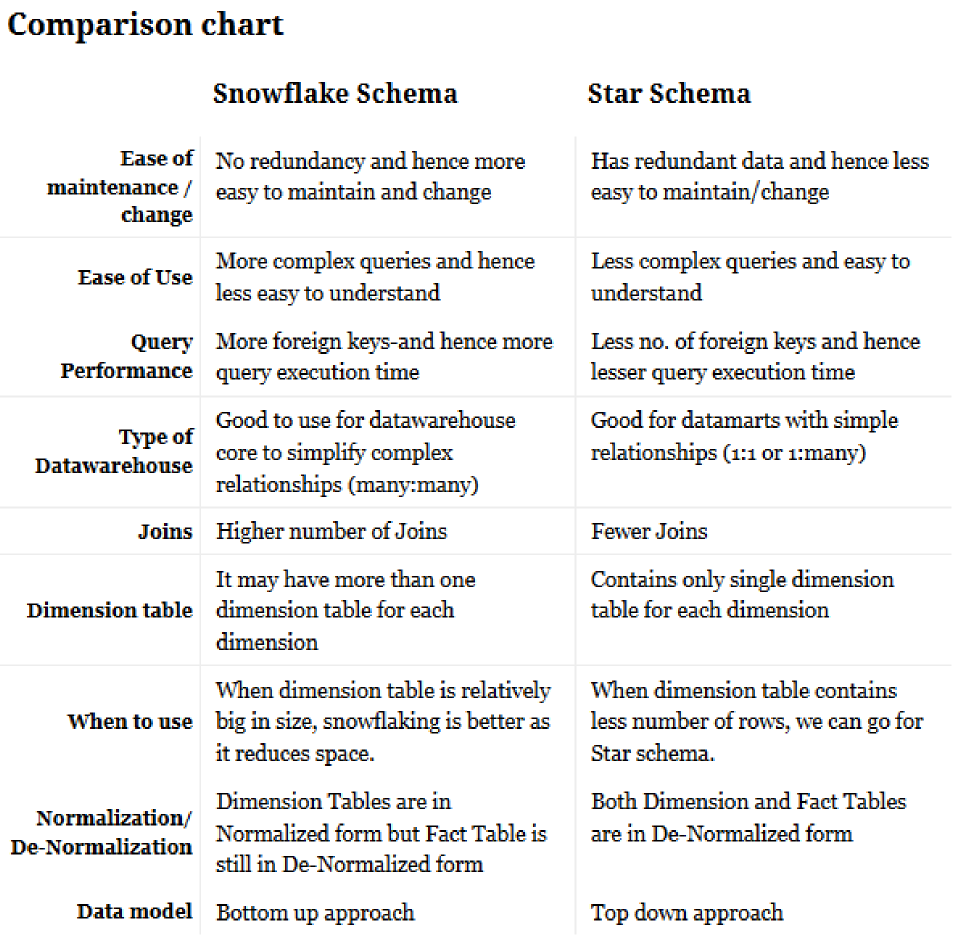
Star Schema: Central fact table connected directly to dimension tables.
Snowflake Schema: Extension of Star Schema where dimension tables are further normalized into multiple related tables.

**When to Use:**

- **Star Schema**: Use it when prioritizing **performance and simplicity** in querying, particularly with smaller or moderately sized datasets.
- **Snowflake Schema**: Choose it for **large, complex datasets** where **storage efficiency** and **normalized data** structures are important

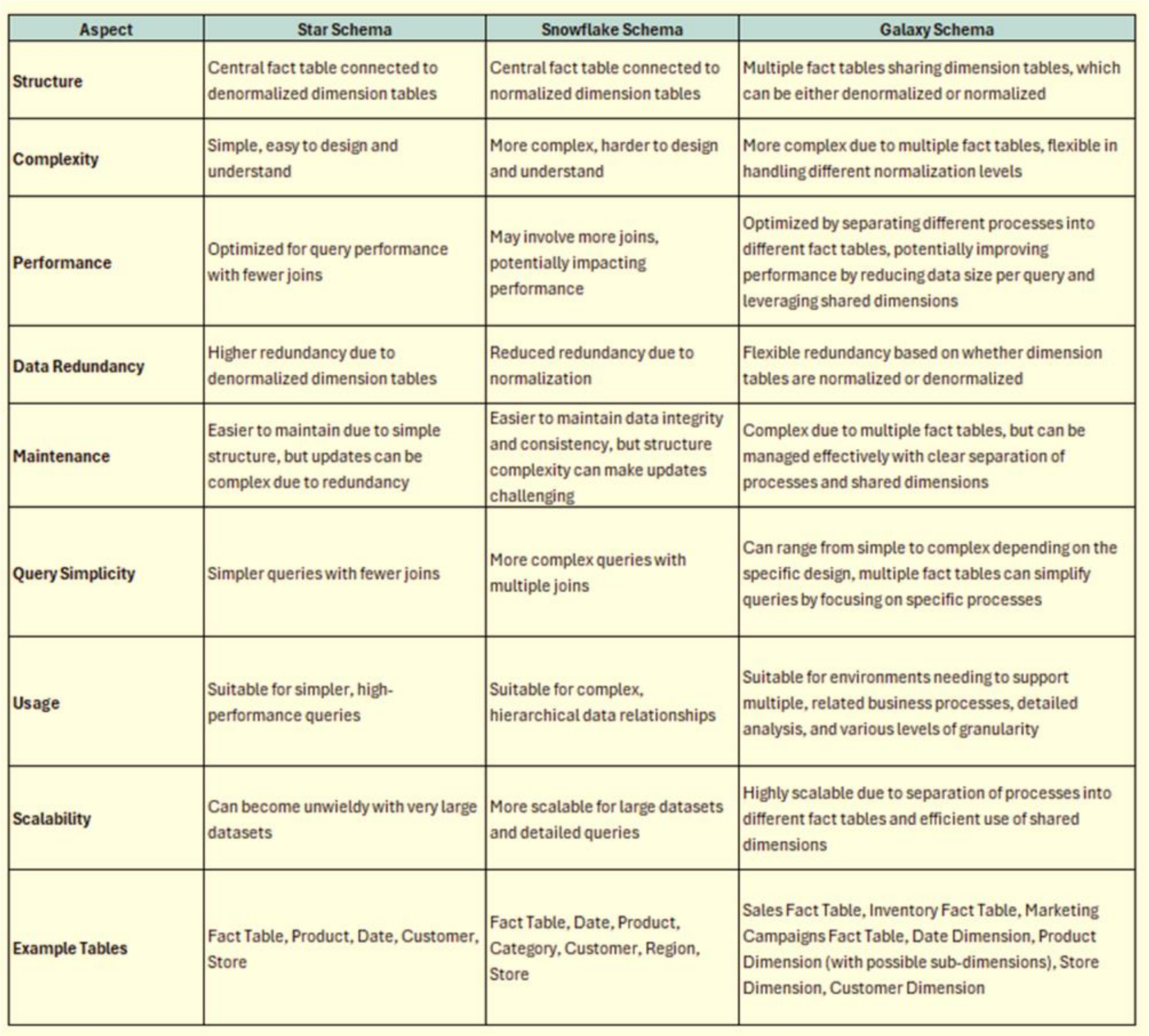
**Star Schema:** It is easier to query and generally performs faster due to fewer joins, making it ideal for smaller datasets or simple business queries. **Use Case**: Best suited for businesses prioritizing **simplicity** and **speed** in queries, especially with small to moderately sized data.

**Galaxy (Fact Constellation) Schema: duplication** and **slower query performance** due to additional joins. **Use Case**: Ideal for **large and complex datasets** where multiple facts (e.g., sales, inventory) need to be analyzed together across shared dimensions



## Comparison chart

| | Snowflake Schema | Star Schema |
|---|---|---|
| **Ease of maintenance / change** | No redundancy and hence more easy to maintain and change | Has redundant data and hence less easy to maintain/change |
| **Ease of Use** | More complex queries and hence less easy to understand | Less complex queries and easy to understand |
| **Query Performance** | More foreign keys-and hence more query execution time | Less no. of foreign keys and hence lesser query execution time |
| **Type of Datawarehouse** | Good to use for datawarehouse core to simplify complex relationships (many:many) | Good for datamarts with simple relationships (1:1 or 1:many) |
| **Joins** | Higher number of Joins | Fewer Joins |
| **Dimension table** | It may have more than one dimension table for each dimension | Contains only single dimension table for each dimension |
| **When to use** | When dimension table is relatively big in size, snowflaking is better as it reduces space. | When dimension table contains less number of rows, we can go for Star schema. |
| **Normalization/ De-Normalization** | Dimension Tables are in Normalized form but Fact Table is still in De-Normalized form | Both Dimension and Fact Tables are in De-Normalized form |
| **Data model** | Bottom up approach | Top down approach |



| Aspect | Star Schema | Snowflake Schema | Galaxy Schema |
|---|---|---|---|
| **Structure** | Central fact table connected to denormalized dimension tables | Central fact table connected to normalized dimension tables | Multiple fact tables sharing dimension tables, which can be either denormalized or normalized |
| **Complexity** | Simple, easy to design and understand | More complex, harder to design and understand | More complex due to multiple fact tables, flexible in handling different normalization levels |
| **Performance** | Optimized for query performance with fewer joins | May involve more joins, potentially impacting performance | Optimized by separating different processes into different fact tables, potentially improving performance by reducing data size per query and leveraging shared dimensions |
| **Data Redundancy** | Higher redundancy due to denormalized dimension tables | Reduced redundancy due to normalization | Flexible redundancy based on whether dimension tables are normalized or denormalized |
| **Maintenance** | Easier to maintain due to simple structure, but updates can be complex due to redundancy | Easier to maintain data integrity and consistency, but structure complexity can make updates challenging | Complex due to multiple fact tables, but can be managed effectively with clear separation of processes and shared dimensions |
| **Query Simplicity** | Simpler queries with fewer joins | More complex queries with multiple joins | Can range from simple to complex depending on the specific design, multiple fact tables can simplify queries by focusing on specific processes |
| **Usage** | Suitable for simpler, high-performance queries | Suitable for complex, hierarchical data relationships | Suitable for environments needing to support multiple, related business processes, detailed analysis, and various levels of granularity |
| **Scalability** | Can become unwieldy with very large datasets | More scalable for large datasets and detailed queries | Highly scalable due to separation of processes into different fact tables and efficient use of shared dimensions |
| **Example Tables** | Fact Table, Product, Date, Customer, Store | Fact Table, Date, Product, Category, Customer, Region, Store | Sales Fact Table, Inventory Fact Table, Marketing Campaigns Fact Table, Date Dimension, Product Dimension (with possible sub-dimensions), Store Dimension, Customer Dimension |

## ☺ Describe how to implement role-playing dimensions in Power BI?

Role-playing dimensions are dimensions that can be used in different contexts within a model. For example, a Date dimension can be used for Order Date, Ship Date, and Due Date. To implement role-playing dimensions in Power BI, follow these steps:

1. Duplicate the Dimension Table:
   - Create multiple instances of the same dimension (like Date) by duplicating the table in Power BI. Each instance will play a different role in filtering your data (e.g., Order Date, Ship Date).
2. Create Relationships:
   - For each role, define a separate relationship between the fact table and the duplicated dimension tables. For example, relate the Order Date in the fact table to the Order Date dimension, and similarly for other roles.
3. Use DAX with USERELATIONSHIP:
   - If you don't want to duplicate the table, use the USERELATIONSHIP function in DAX to switch between different relationships dynamically in your measures. This is particularly useful when handling multiple date fields in reports.
4. Use in Visualizations:

  - Ensure that you use the correct dimension (role) in your visualizations to reflect the right data context (e.g., showing sales based on Order Date vs. Ship Date).

This method provides flexibility and clarity when handling multiple instances of the same dimension in various contexts.

### ☺ How would you manage circular dependencies when they occur in your data model?

Circular dependencies occur when calculations or relationships in a data model depend on each other, creating a loop that Power BI cannot resolve. Here are some strategies to manage circular dependencies:

1. **Use Measures Instead of Calculated Columns**:
   - Calculated columns are evaluated when data is loaded, leading to circular dependencies. Measures, however, are computed at query time and can avoid circular dependencies as they don't persist in the model.
2. **Rearrange or Simplify Dependencies**:
   - Review the relationships and calculations that are creating circular dependency. Simplify the logic or ensure the dependencies flow in one direction by breaking complex chains of calculated columns.
3. **Reorganize the Data Model**:
   - If possible, change the data model's structure by introducing intermediate tables or breaking up complex relationships to avoid circular dependencies.

### ☺ What is the CALCULATE function in DAX?

The CALCULATE function in DAX (Data Analysis Expressions) is a powerful and essential function used to modify the filter context of a calculation. Its primary purpose is to evaluate an expression while applying one or more filters to it. This allows users to create complex calculations that go beyond simple aggregations.

**Key Features:**

1. **Context Transition**: CALCULATE changes the row context to a filter context, enabling it to work with more complex data models.
2. **Multiple Filters**: You can apply multiple filters in a single CALCULATE function, enhancing flexibility in data analysis.
3. **Aggregations**: It's often used to create aggregations that are dependent on specific criteria, such as calculating sales totals for a specific region or time period.

Overall, CALCULATE is integral to creating insightful reports and dashboards in Power BI, enabling users to derive meaningful insights from their data.

### ☺ What is difference between SUM and SUMX?

- **SUM**: A simple aggregation function that adds up all the values in a single column. It is used when you want a straightforward sum without any row-level context.
- **SUMX**: An iterator function that performs calculations row by row. It evaluates an expression for each row in a table and then sums the results. This makes it more flexible for complex calculations involving multiple columns.

**Usage**:

- **SUM** is suitable for direct summation of numeric columns.
- **SUMX** is preferred when you need to apply an expression that involves more than one column or to apply a filter or condition to the sum

### ☺ What is use case of SUMMARIZE function?

The SUMMARIZE function in DAX is primarily used for creating summary tables by grouping data. Here are key use cases:

1. **Data Aggregation**: SUMMARIZE is utilized to aggregate data from a detailed dataset, allowing users to create summary statistics such as totals, averages, or counts based on specific columns. For instance, you can summarize sales data by product category and region.
2. **Creating Crosstab Reports**: It helps in generating crosstab reports where you need to analyze data across multiple dimensions. This is particularly useful for visualizations in Power BI that

require grouped data for comparisons.

3. **Customized Reporting**: By specifying grouping columns and additional calculations, users can customize reports to fit specific analytical needs, providing tailored insights into the dataset

### ☺ Can you explain the use case of SAMEPERIODLASTYEAR?

The SAMEPERIODLASTYEAR function in Power BI is primarily used for comparative time analysis, specifically to assess performance over the same period in the previous year. Here are key use cases:

1. **Year-over-Year (YoY) Analysis**: It helps businesses evaluate performance changes by comparing current year metrics with the same period from the last year, such as sales or revenue.
2. **Trend Analysis**: Analysts can identify trends over time, making it easier to see seasonal patterns or growth by evaluating how values change from year to year.
3. **Forecasting and Planning**: Organizations can use insights gained from these comparisons to inform future strategies and forecasts, helping in resource allocation and budget planning.

In summary, SAMEPERIODLASTYEAR is an essential DAX function for businesses seeking to perform historical comparisons and understand trends effectively.

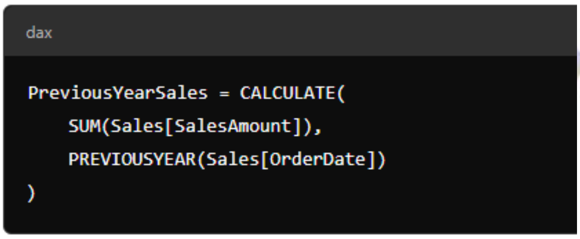
### ☺ What do you mean by MTD , QTD & YTD?

MTD, QTD, and YTD are financial metrics used to measure performance over specific time frames:

1. **MTD (Month-to-Date)**: This measures the performance from the beginning of the current month up to the current date. It's useful for assessing how well a business is doing in the current month.
2. **QTD (Quarter-to-Date)**: This metric covers the period from the start of the current quarter (typically three months) up to the current date. It allows for evaluation of performance in the current quarter.
3. **YTD (Year-to-Date)**: YTD measures the performance from the beginning of the current year to the current date. This metric is often used for longer-term assessments, such as annual budgeting and forecasting.

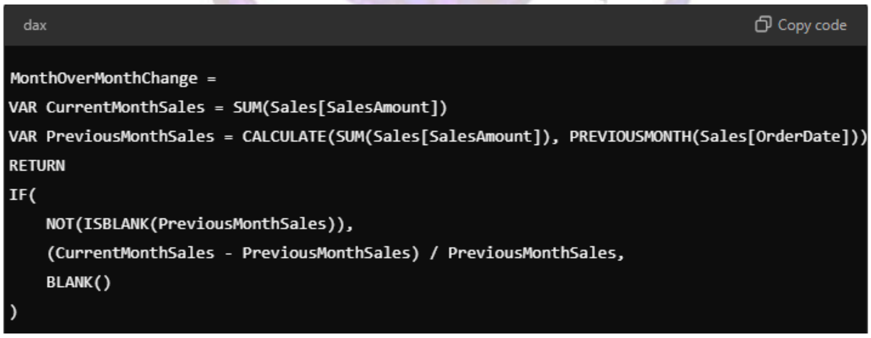
These metrics help businesses track their progress and make informed decisions based on performance trends.

### ☺ Write a DAX formula to calculate the previous year's sales.



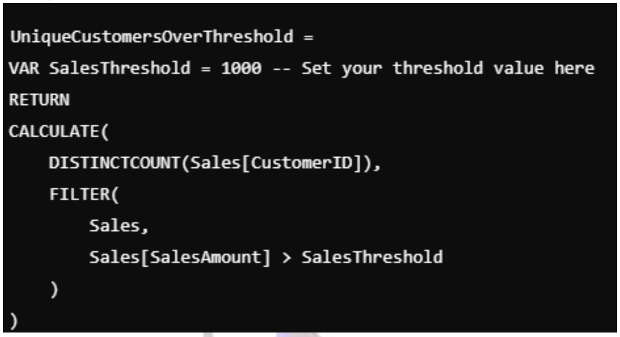
```
PreviousYearSales = CALCULATE(
    SUM(Sales[SalesAmount]),
    PREVIOUSYEAR(Sales[OrderDate])
)
```

### ☺ How would you create a measure to calculate the percentage change in sales month-over-month?



```
MonthOverMonthChange =
VAR CurrentMonthSales = SUM(Sales[SalesAmount])
VAR PreviousMonthSales = CALCULATE(SUM(Sales[SalesAmount]), PREVIOUSMONTH(Sales[OrderDate]))
RETURN
IF(
    NOT(ISBLANK(PreviousMonthSales)),
    (CurrentMonthSales - PreviousMonthSales) / PreviousMonthSales,
    BLANK()
)
```

### ☺ Write a DAX expression to count the number of unique customers who made purchases over a certain threshold.



```
UniqueCustomersOverThreshold =
VAR SalesThreshold = 1000 -- Set your threshold value here
RETURN
CALCULATE(
    DISTINCTCOUNT(Sales[CustomerID]),
    FILTER(
        Sales,
        Sales[SalesAmount] > SalesThreshold
    )
)
```

### ☺ Describe how you would split a single column into multiple columns based on a delimiter in Power Query.

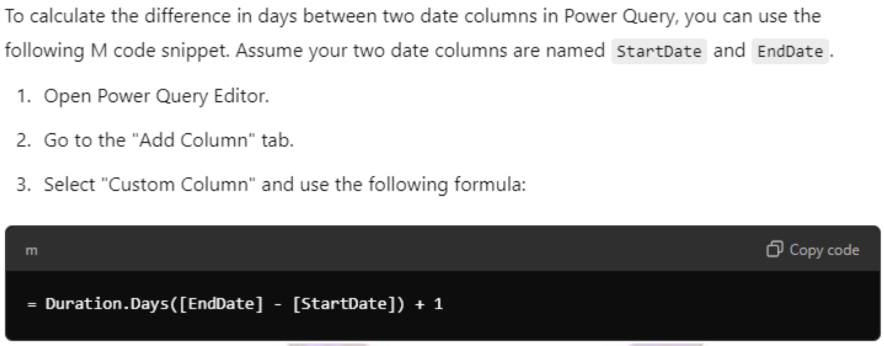
To unpivot a table in Power Query, follow these steps:

1. **Select Columns**: Choose the columns you want to unpivot. Typically, these are the columns that contain attribute data you want to convert into rows, leaving the identifier columns (e.g., ID, names) intact.
2. **Transform to Unpivot**: Go to the "Transform" tab, and click on "Unpivot Columns." This will convert the selected columns into attribute-value pairs, creating two new columns: one for the attribute names and another for their corresponding values.
3. **Finalize Your Table**: You can then rename the new columns as necessary and perform any additional transformations.

**Scenarios for Unpivoting:**

- **Normalizing Data**: When working with data that is in a wide format (e.g., multiple columns for sales in different months), unpivoting can help normalize it into a more analyzable long format.
- **Facilitating Analysis**: Unpivoting makes it easier to aggregate or analyze data, especially for visualizations that require a single value column.
- **Preparing for Data Models**: When feeding data into models that require a tidy dataset, unpivoting is essential

### ☺ Write a Power Query M code to create a new column that calculates the difference in days between two date columns.

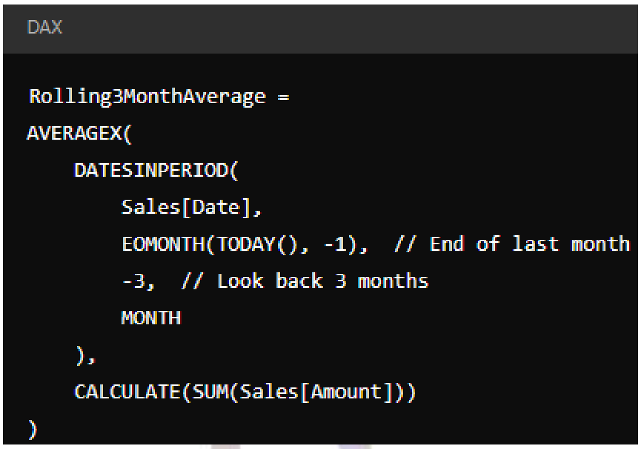


To calculate the difference in days between two date columns in Power Query, you can use the following M code snippet. Assume your two date columns are named `StartDate` and `EndDate`.

1. Open Power Query Editor.
2. Go to the "Add Column" tab.
3. Select "Custom Column" and use the following formula:

```m
= Duration.Days([EndDate] - [StartDate]) + 1
```

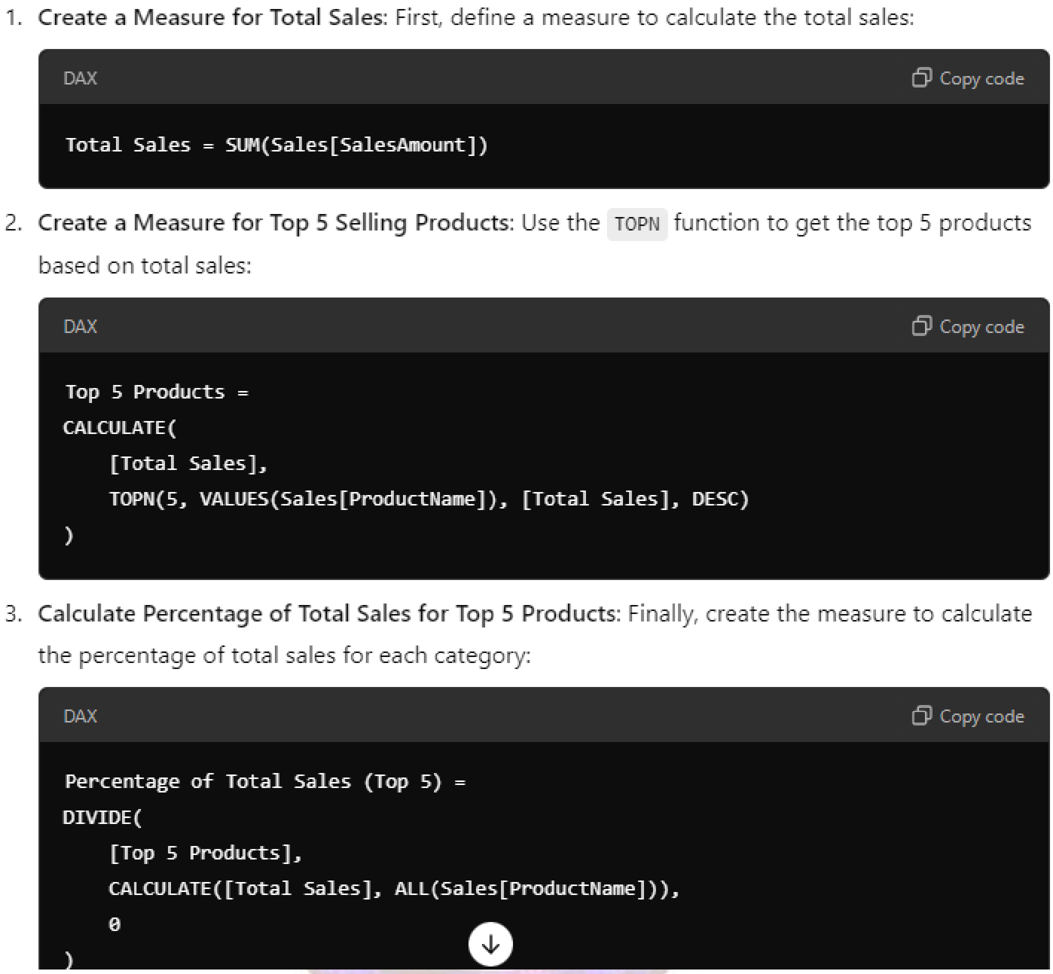
### ☺ How would you create a rolling 3-month average of sales, excluding the current month?



DAX

```
Rolling3MonthAverage =
AVERAGEX(
    DATESINPERIOD(
        Sales[Date],
        EOMONTH(TODAY(), -1),  // End of last month
        -3,  // Look back 3 months
        MONTH
    ),
    CALCULATE(SUM(Sales[Amount]))
)
```

### ☺ Write a measure to calculate the percentage of total sales for each product category, but only for the top 5 selling products overall.



1. **Create a Measure for Total Sales:** First, define a measure to calculate the total sales:

DAX  Copy code

```
Total Sales = SUM(Sales[SalesAmount])
```

2. **Create a Measure for Top 5 Selling Products:** Use the `TOPN` function to get the top 5 products based on total sales:

DAX  Copy code

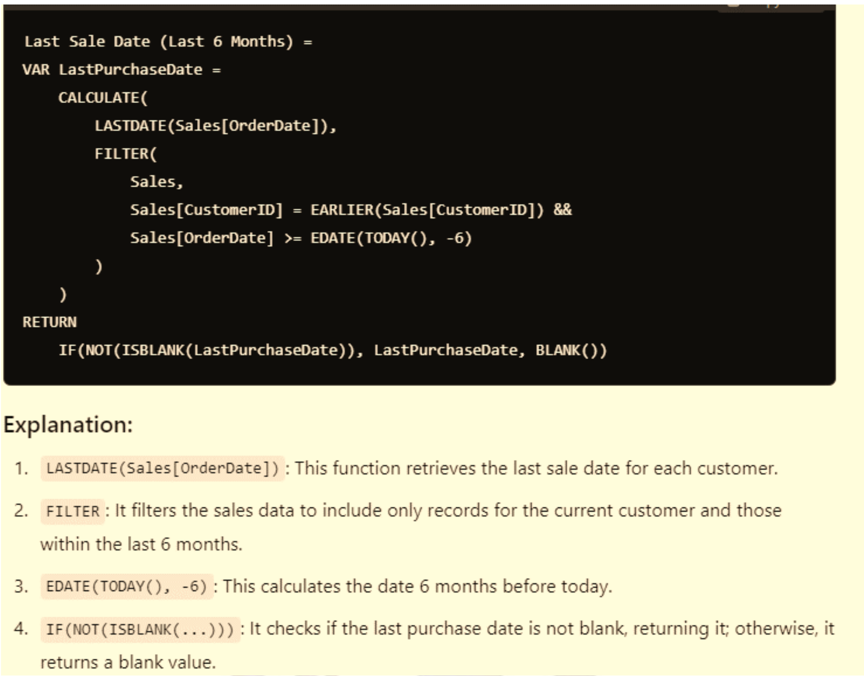
```
Top 5 Products =
CALCULATE(
    [Total Sales],
    TOPN(5, VALUES(Sales[ProductName]), [Total Sales], DESC)
)
```

3. **Calculate Percentage of Total Sales for Top 5 Products:** Finally, create the measure to calculate the percentage of total sales for each category:

DAX  Copy code

```
Percentage of Total Sales (Top 5) =
DIVIDE(
    [Top 5 Products],
    CALCULATE([Total Sales], ALL(Sales[ProductName])),
    0
)
```

☺ **Create a DAX expression to find the date of the last sale for each customer, but only if they've made a purchase in the last 6 months.**

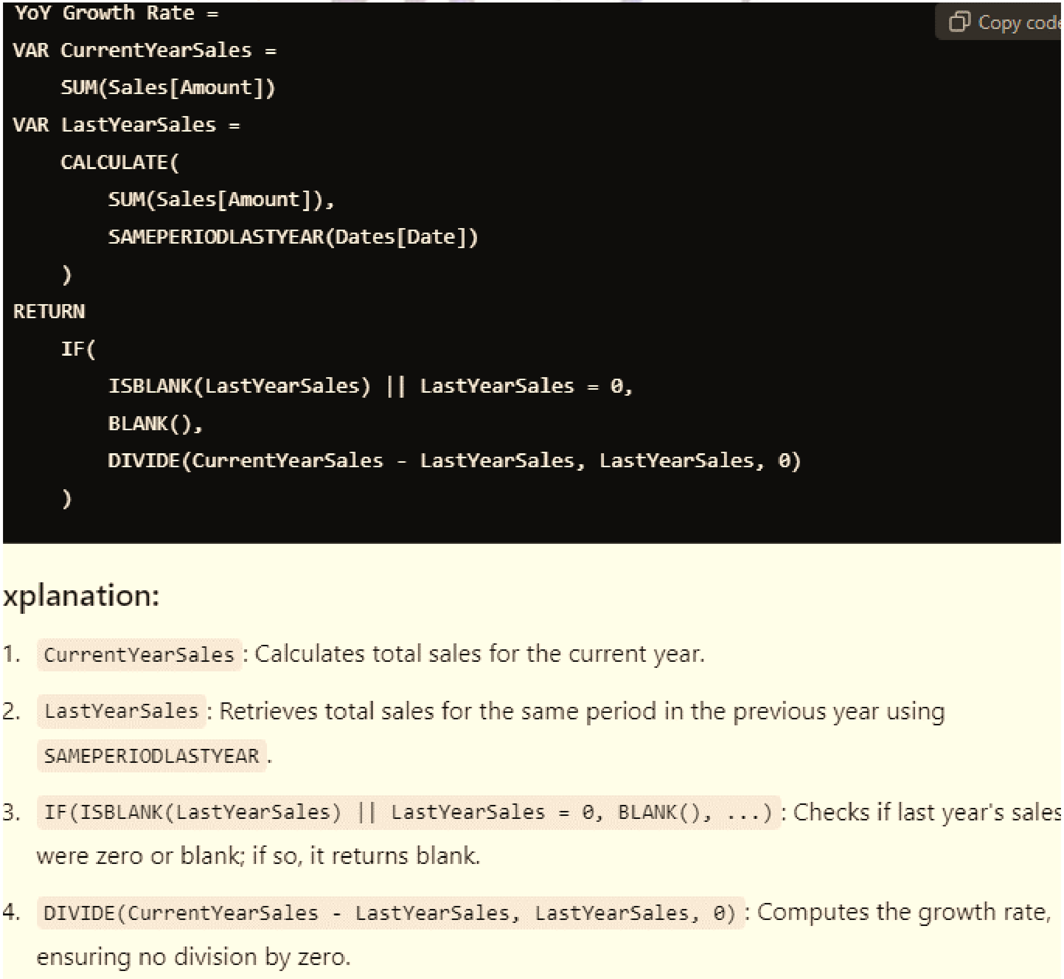


```
Last Sale Date (Last 6 Months) =
VAR LastPurchaseDate =
    CALCULATE(
        LASTDATE(Sales[OrderDate]),
        FILTER(
            Sales,
            Sales[CustomerID] = EARLIER(Sales[CustomerID]) &&
            Sales[OrderDate] >= EDATE(TODAY(), -6)
        )
    )
RETURN
    IF(NOT(ISBLANK(LastPurchaseDate)), LastPurchaseDate, BLANK())
```

Explanation:

1. `LASTDATE(Sales[OrderDate])` : This function retrieves the last sale date for each customer.
2. `FILTER` : It filters the sales data to include only records for the current customer and those within the last 6 months.
3. `EDATE(TODAY(), -6)` : This calculates the date 6 months before today.
4. `IF(NOT(ISBLANK(...)))` : It checks if the last purchase date is not blank, returning it; otherwise, it returns a blank value.

☺ **Develop a measure that calculates the year-over-year growth rate for each product, handling scenarios where there were no sales in the previous year.**

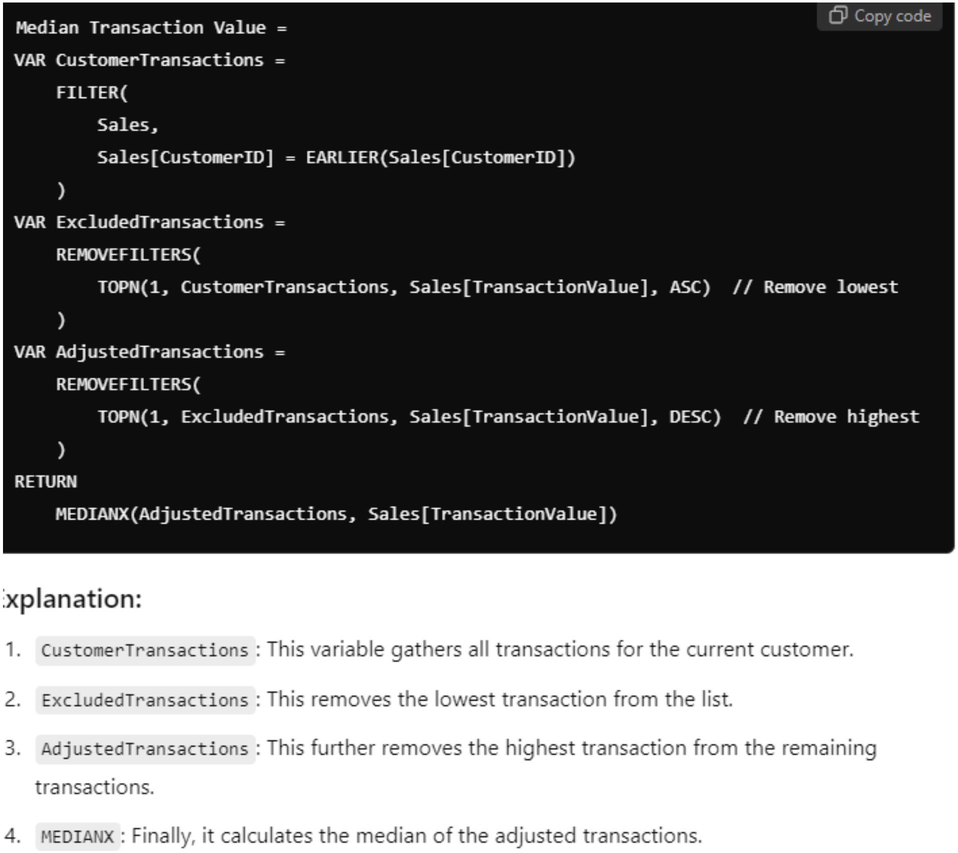


```
YoY Growth Rate =
VAR CurrentYearSales =
    SUM(Sales[Amount])
VAR LastYearSales =
    CALCULATE(
        SUM(Sales[Amount]),
        SAMEPERIODLASTYEAR(Dates[Date])
    )
RETURN
    IF(
        ISBLANK(LastYearSales) || LastYearSales = 0,
        BLANK(),
        DIVIDE(CurrentYearSales - LastYearSales, LastYearSales, 0)
    )
```

xplanation:

1. `CurrentYearSales` : Calculates total sales for the current year.
2. `LastYearSales` : Retrieves total sales for the same period in the previous year using `SAMEPERIODLASTYEAR` .
3. `IF(ISBLANK(LastYearSales) || LastYearSales = 0, BLANK(), ...)` : Checks if last year's sales were zero or blank; if so, it returns blank.
4. `DIVIDE(CurrentYearSales - LastYearSales, LastYearSales, 0)` : Computes the growth rate, ensuring no division by zero.

## ☺ Write a calculation to determine the median transaction value per customer, excluding their highest and lowest purchases.

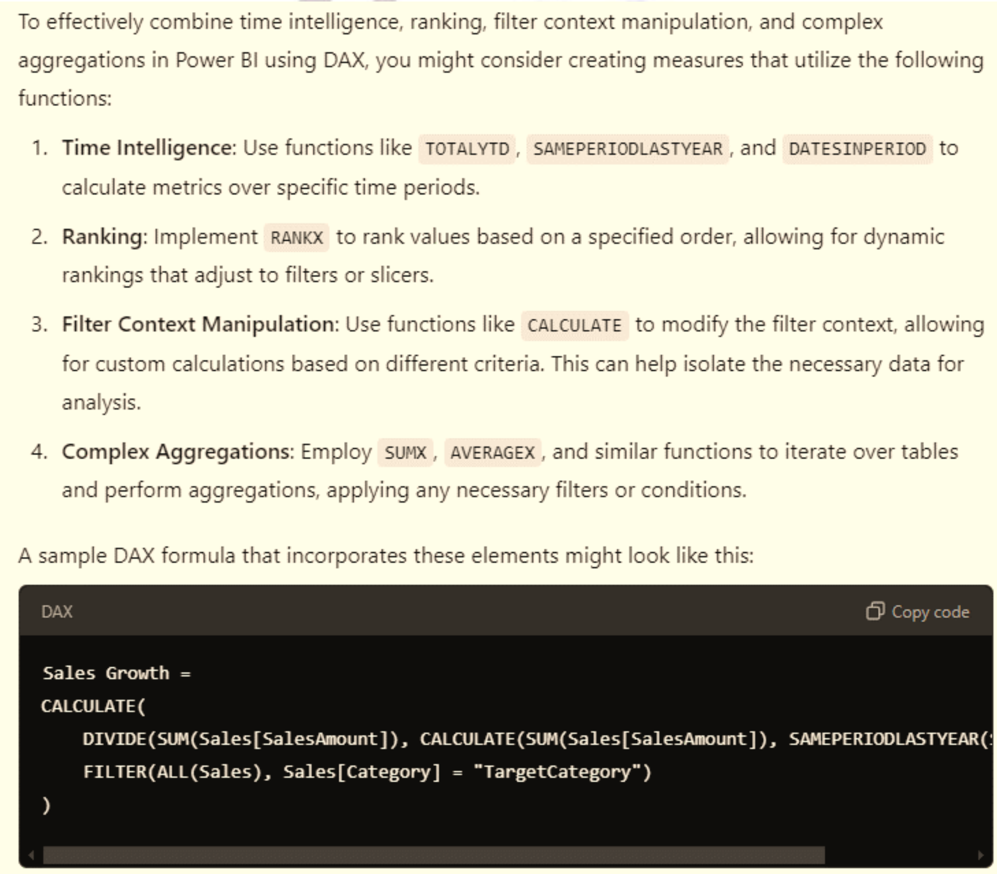


```
Median Transaction Value =
VAR CustomerTransactions =
    FILTER(
        Sales,
        Sales[CustomerID] = EARLIER(Sales[CustomerID])
    )
VAR ExcludedTransactions =
    REMOVEFILTERS(
        TOPN(1, CustomerTransactions, Sales[TransactionValue], ASC)  // Remove lowest
    )
VAR AdjustedTransactions =
    REMOVEFILTERS(
        TOPN(1, ExcludedTransactions, Sales[TransactionValue], DESC)  // Remove highest
    )
RETURN
    MEDIANX(AdjustedTransactions, Sales[TransactionValue])
```

xplanation:

1. `CustomerTransactions`: This variable gathers all transactions for the current customer.
2. `ExcludedTransactions`: This removes the lowest transaction from the list.
3. `AdjustedTransactions`: This further removes the highest transaction from the remaining transactions.
4. `MEDIANX`: Finally, it calculates the median of the adjusted transactions.

## ☺ These questions test your ability to combine time intelligence, ranking, filter context manipulation, and complex aggregations.



To effectively combine time intelligence, ranking, filter context manipulation, and complex aggregations in Power BI using DAX, you might consider creating measures that utilize the following functions:

1. **Time Intelligence**: Use functions like `TOTALYTD`, `SAMEPERIODLASTYEAR`, and `DATESINPERIOD` to calculate metrics over specific time periods.
2. **Ranking**: Implement `RANKX` to rank values based on a specified order, allowing for dynamic rankings that adjust to filters or slicers.
3. **Filter Context Manipulation**: Use functions like `CALCULATE` to modify the filter context, allowing for custom calculations based on different criteria. This can help isolate the necessary data for analysis.
4. **Complex Aggregations**: Employ `SUMX`, `AVERAGEX`, and similar functions to iterate over tables and perform aggregations, applying any necessary filters or conditions.

A sample DAX formula that incorporates these elements might look like this:

```
Sales Growth =
CALCULATE(
    DIVIDE(SUM(Sales[SalesAmount]), CALCULATE(SUM(Sales[SalesAmount]), SAMEPERIODLASTYEAR(
    FILTER(ALL(Sales), Sales[Category] = "TargetCategory")
)
```

YouTube channels to learn Power BI from:

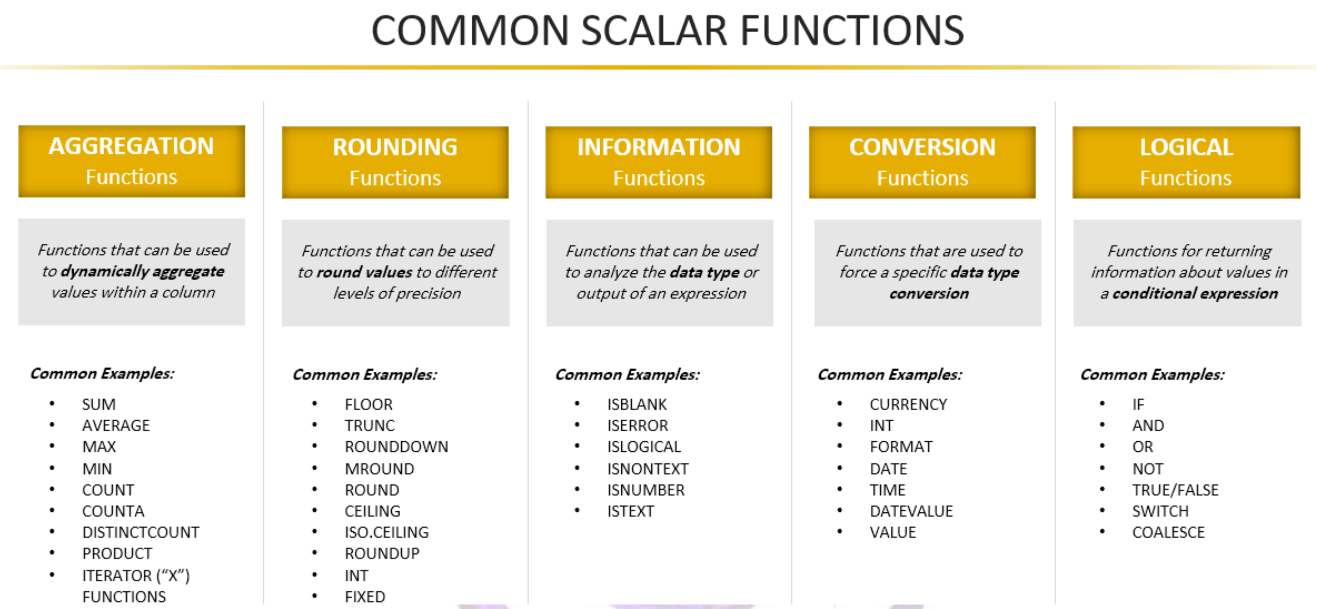
1. Curbal
2. Guy in a Cube
3. RADACAD
4. How to Power BI
5. Enterprise DNA

---

References

1. https://docs.preset.io/docs/row-level-security-rls
2. https://www.linkedin.com/in/raj-maradiya/
3. https://www.datacamp.com/cheat-sheet/dax-cheat-sheet
4. ***docs.microsoft.com***
5. ***https://mavenanalytics.io/***
6. https://mastering-da.com/business-intelligence-program/
7.

https://www.sontai.com/post/top-5-power-bi-charts-and-how-to-use-them
https://www.youtube.com/@datalab365
https://www.youtube.com/@AlexTheAnalyst/playlists



## COMMON SCALAR FUNCTIONS

**AGGREGATION** Functions

*Functions that can be used to **dynamically aggregate** values within a column*

***Common Examples:***

- SUM
- AVERAGE
- MAX
- MIN
- COUNT
- COUNTA
- DISTINCTCOUNT
- PRODUCT
- ITERATOR ("X") FUNCTIONS

**ROUNDING** Functions

*Functions that can be used to **round values** to different levels of precision*

***Common Examples:***

- FLOOR
- TRUNC
- ROUNDDOWN
- MROUND
- ROUND
- CEILING
- ISO.CEILING
- ROUNDUP
- INT
- FIXED

**INFORMATION** Functions

*Functions that can be used to analyze the **data type** or output of an expression*

***Common Examples:***

- ISBLANK
- ISERROR
- ISLOGICAL
- ISNONTEXT
- ISNUMBER
- ISTEXT

**CONVERSION** Functions

*Functions that are used to force a specific **data type conversion***

***Common Examples:***

- CURRENCY
- INT
- FORMAT
- DATE
- TIME
- DATEVALUE
- VALUE

**LOGICAL** Functions

*Functions for returning information about values in a **conditional expression***

***Common Examples:***

- IF
- AND
- OR
- NOT
- TRUE/FALSE
- SWITCH
- COALESCE

# COMMON TABLE & FILTER FUNCTIONS

## Filter Data

*Functions used to return filtered tables or filter results of measures*

***Common Examples:***

- ALL
- FILTER
- DISTINCT
- VALUES
- ALLEXCEPT
- ALLSELECTED

## Add Data

*Functions used to specify or add columns based on existing data in the model*

***Common Examples:***

- SELECTCOLUMNS
- ADDCOLUMNS
- SUMMARIZE

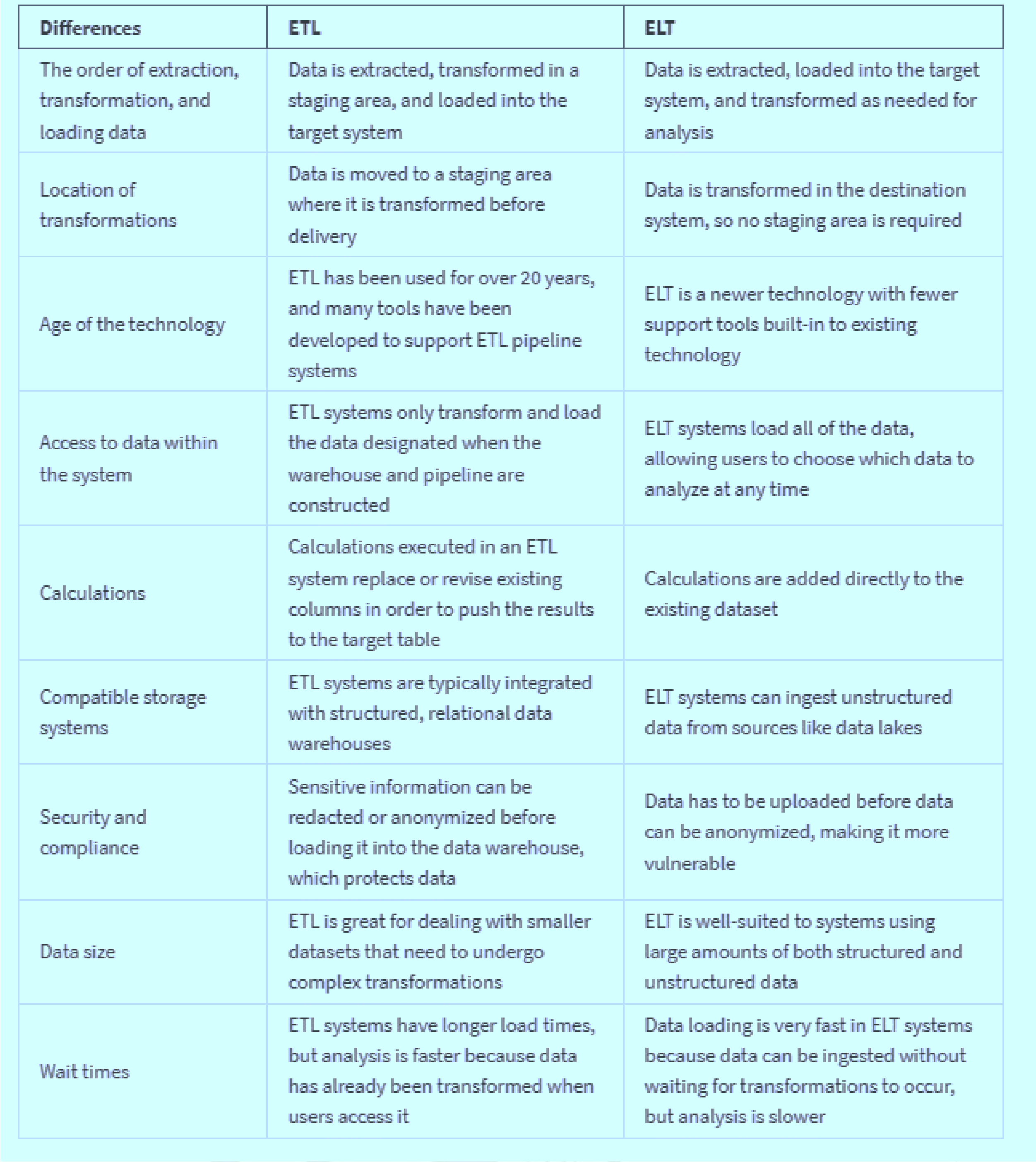
## Create Data

*Functions used to generate new rows, columns & tables from scratch*

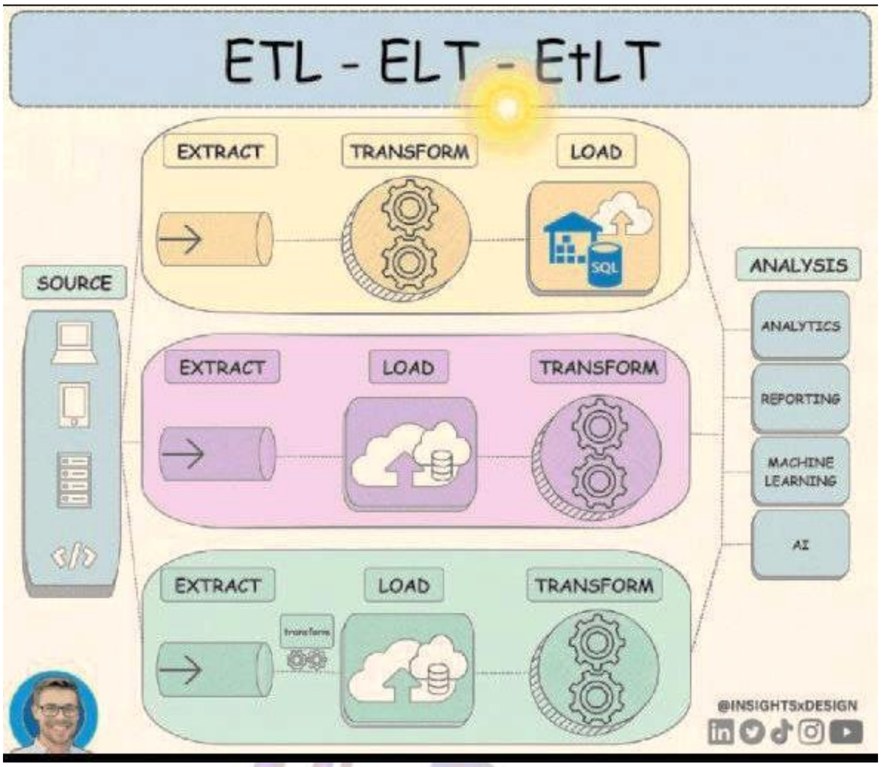
***Common Examples:***

- ROW
- DATATABLE
- GENERATESERIES
- { } Table Constructor

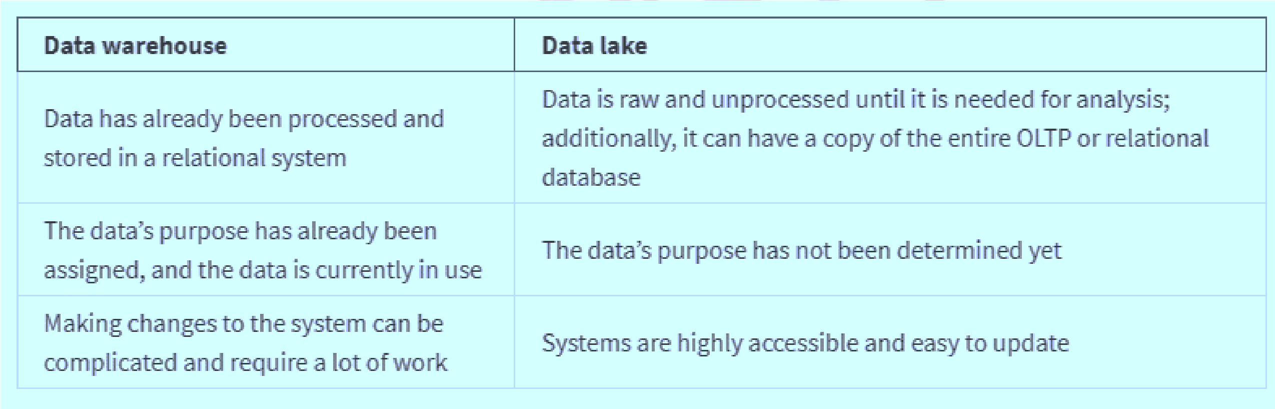
- **What is the difference between ETL vs. ELT**



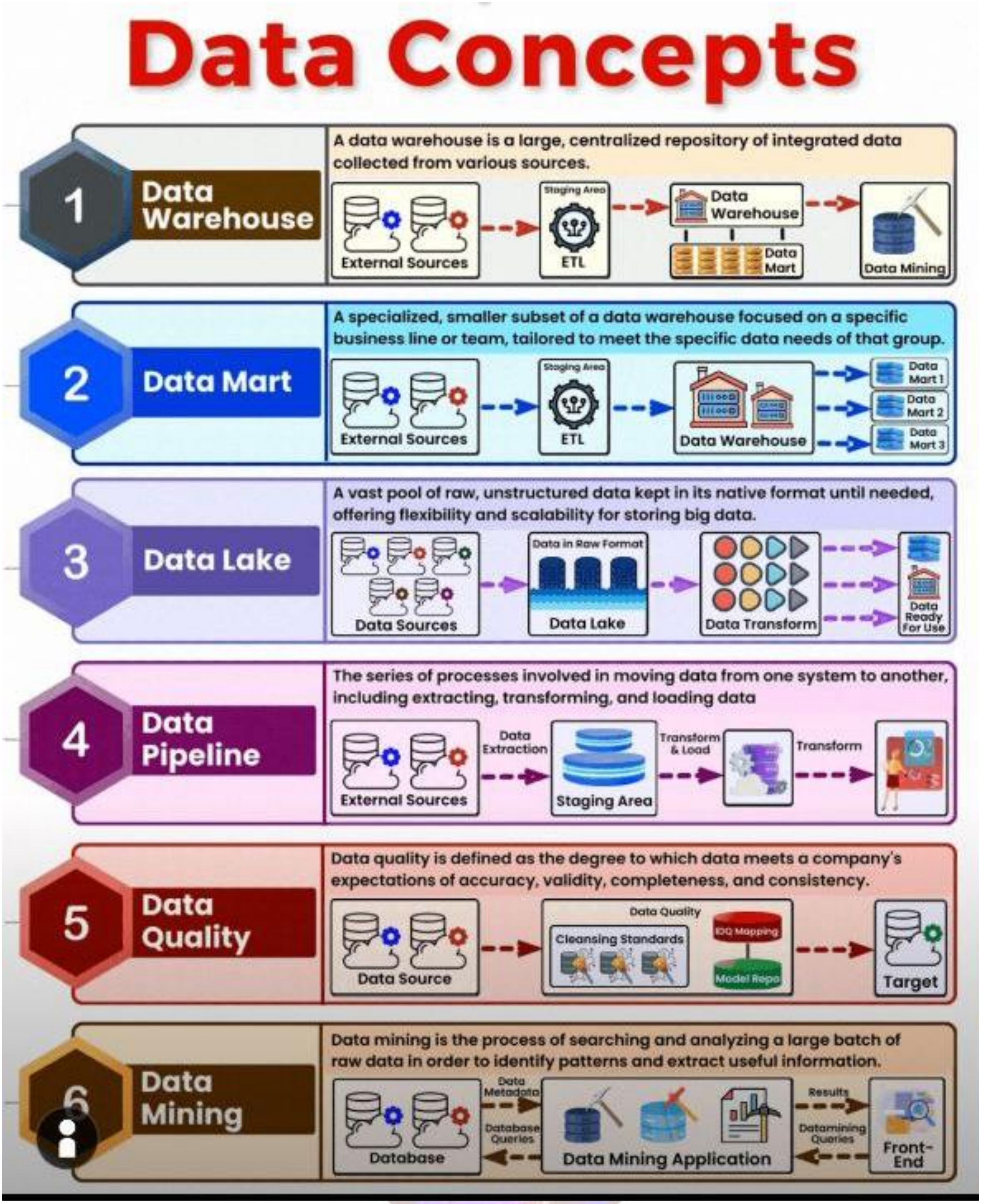
| Differences | ETL | ELT |
|---|---|---|
| The order of extraction, transformation, and loading data | Data is extracted, transformed in a staging area, and loaded into the target system | Data is extracted, loaded into the target system, and transformed as needed for analysis |
| Location of transformations | Data is moved to a staging area where it is transformed before delivery | Data is transformed in the destination system, so no staging area is required |
| Age of the technology | ETL has been used for over 20 years, and many tools have been developed to support ETL pipeline systems | ELT is a newer technology with fewer support tools built-in to existing technology |
| Access to data within the system | ETL systems only transform and load the data designated when the warehouse and pipeline are constructed | ELT systems load all of the data, allowing users to choose which data to analyze at any time |
| Calculations | Calculations executed in an ETL system replace or revise existing columns in order to push the results to the target table | Calculations are added directly to the existing dataset |
| Compatible storage systems | ETL systems are typically integrated with structured, relational data warehouses | ELT systems can ingest unstructured data from sources like data lakes |
| Security and compliance | Sensitive information can be redacted or anonymized before loading it into the data warehouse, which protects data | Data has to be uploaded before data can be anonymized, making it more vulnerable |
| Data size | ETL is great for dealing with smaller datasets that need to undergo complex transformations | ELT is well-suited to systems using large amounts of both structured and unstructured data |
| Wait times | ETL systems have longer load times, but analysis is faster because data has already been transformed when users access it | Data loading is very fast in ELT systems because data can be ingested without waiting for transformations to occur, but analysis is slower |



- ☻ **What is the difference between Data warehouse and data lake**



| Data warehouse | Data lake |
|---|---|
| Data has already been processed and stored in a relational system | Data is raw and unprocessed until it is needed for analysis; additionally, it can have a copy of the entire OLTP or relational database |
| The data's purpose has already been assigned, and the data is currently in use | The data's purpose has not been determined yet |
| Making changes to the system can be complicated and require a lot of work | Systems are highly accessible and easy to update |



# Data Concepts

## 1 Data Warehouse

A data warehouse is a large, centralized repository of integrated data collected from various sources.

## 2 Data Mart

A specialized, smaller subset of a data warehouse focused on a specific business line or team, tailored to meet the specific data needs of that group.

## 3 Data Lake

A vast pool of raw, unstructured data kept in its native format until needed, offering flexibility and scalability for storing big data.

## 4 Data Pipeline

The series of processes involved in moving data from one system to another, including extracting, transforming, and loading data

## 5 Data Quality

Data quality is defined as the degree to which data meets a company's expectations of accuracy, validity, completeness, and consistency.

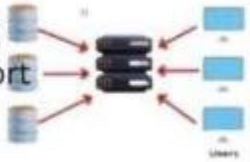
## 6 Data Mining

Data mining is the process of searching and analyzing a large batch of raw data in order to identify patterns and extract useful information.

# 6 KEY DATA TERMS
# YOU SHOULD KNOW

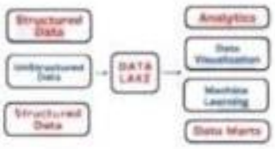
**DATA WAREHOUSE**

a central repository for storing and managing data from various sources; designed to support complex queries and analysis.



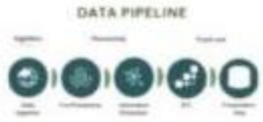
**DATA LAKE**

a central repository for storing data in its raw format; often used to store large amounts of data from a variety of sources, including structured, unstructured, and semi-structured data.



**DATA PIPELINE**

a series of steps that are used to move data from one system to another. It can be used to collect, process, and store data.



**BIG DATA PROCESSING**

the process of managing and analyzing large datasets; involves using specialized tools and techniques to handle the volume, velocity, and variety of big data.

**DATA GOVERNANCE**

the process of managing the flow of data through an organization. It includes policies and procedures for collecting, storing, processing, and using data.

**CLOUD COMPUTING**

the delivery of computing resources over the internet. It can be used to store, process, and analyze data on a large scale.

CLOUD COMPUTING

# DAX Functions Cheat Sheet

YOUR CHEAT SHEET

## Aggregation

Summarizes a set of values into a single result

| | | |
|---|---|---|
| SUM | Adds up all the values in a column. | = SUM(ColumnName) |
| AVERAGE | Calculates the average of all the values in a column. | = AVERAGE(ColumnName) |
| MAX | Returns the largest value in a column. | = MAX(ColumnName) |
| MIN | Returns the smallest value in a column. | = MIN(ColumnName) |
| COUNT | Counts the number of rows in a table or the number of non-blank values in a column. | = COUNT(ColumnName) |
| DIVIDE | Performs division and returns a result. It is often used to handle division by zero situations. | = DIVIDE(Numerator, Denominator, [AlternateResult]) |

## Iterator

Applies a calculation or expression to each row in a table

| | | |
|---|---|---|
| SUMX | Returns the sum of an expression evaluated for each row in a table. | = SUMX(Table, Expression) |
| AVERAGEX | Calculates the average of a set of expressions evaluated over a table. | = AVERAGEX(Table, Expression) |
| MAXX | Returns the highest value that results from evaluating an expression for each row of a table. | = MAXX(Table, Expression, [Variant]) |
| MINX | Returns the lowest value that results from evaluating an expression for each row of a table. | = MINX(Table, Expression, [Variant]) |
| COUNTX | Count the number of rows from an expression that evaluates to a non-blank value. | = COUNTX(Table, Expression) |

## Date & Time

Create calculations based on dates and time

| | | |
|---|---|---|
| CALENDER | Returns a table with a single column named "Date" that contains a contiguous set of dates. | = CALENDER(StartDate, EndDate) |
| DATE | Returns the specified date in datetime format. | = DATE(Year, Month, Day) |
| DATEDIFF | Returns the number of interval boundaries between two dates. | = DATEDIFF(Date1, Date2, Interval) |
| DATEVALUE | Converts a date in the form of text to a date in datetime format. | = DATEVALUE(DateText) |
| EOMONTH | Returns the date in datetime format of the last day of the month, before or after a specified number of months. | = EOMONTH(StartDate, Month) |
| NETWORKDAYS | Returns the number of whole workdays between two dates. | = NETWORKDAYS(StartDate, EndDate, [Weekend], [Holidays]) |
| WEEKDAY | Returns a number from 1 to 7 identifying the day of the week of a date. | = WEEKDAY(Date, [ReturnType]) |
| WEEKNUM | Returns the week number for the given date and year. | = WEEKNUM(Date, [ReturnType]) |

## Logical

Performs logical operations

| | | |
|---|---|---|
| IF | Checks a condition, and returns one value when TRUE, otherwise it returns a second value. | = IF(LogicalTest, ResultIfTrue, [ResultIfFalse]) |
| AND | Checks whether both arguments are TRUE, and returns TRUE if both arguments are TRUE. | = AND(Logic1, Logic2) |
| OR | Checks whether one of the arguments is TRUE to return TRUE. | = OR(Logic1, Logic2) |
| NOT | Changes FALSE to TRUE, or vice versa. | = NOT(Logic) |
| SWITCH | Evaluates an expression against a list of values and returns one of multiple possible result expressions. | = SWITCH(Expression, Value1, Result1, Value2, Result2, ..., [Else]) |
| IFERROR | Evaluates an expression and returns a specified value if the expression returns an error. | = IFERROR(Value, ValueIfError) |

## Text

Manipulates and formats text values within table

| | | |
|---|---|---|
| COMBINEVALUES | Joins two or more text strings into one text string. | = COMBINEVALUES(Delimiter, Expression1, ...) |
| CONCATENATE | Joins two text strings into one text string. | = CONCATENATE(Text1, Text2) |
| FIND | Returns the starting position of one text string within another text string. | = FIND(FindText, WithinText, [StartPosition], [NotFoundValue]) |
| FORMAT | Converts a value to text according to the specified format. | = FORMAT (Value, Format, [LocalName]) |
| REPLACE | Replaces part of a text string, based on the number of characters you specify, with a different text string. | = REPLACE(OldText, StartPosition, NumberofCharacters, NewText) |
| TRIM | Removes the extra spaces from the text. | = TRIM(Text) |
| UPPER / LOWER | Converts all letters in a text string to lower/upper case. | = UPPER / LOWER (Text) |
| SUBSTITUTE | Replace existing text with new text in a string. | = SUBSTITUTE(Text, OldText, NewText, [InstanceNumber]) |

## Relationship

Managing and utilizing relationships between tables

| | | |
|---|---|---|
| CROSSFILTER | Specifies the cross-filtering direction for a relationship between two columns in a calculation. | = CROSSFILTER(LeftColumnName, RightColumnName, CrossFilterType) |
| RELATED | Retrieves a related value from another table. | = RELATED(ColumnName) |
| RELATEDTABLE | Evaluates a table expression in a context modified by the given filters. | = RELATEDTABLE(TableName) |
| USERELATIONSHIP | Specifies a relationship for a specific calculation between columnName1 and columnName2. | = USERELATIONSHIP(ColumnName1, ColumnName2) |

## Filter

Modify tables by selecting/excluding rows based on conditions

| | | |
|---|---|---|
| ALL | Removes filters from a table or column, returning all values and disregarding existing context. | = ALL(TableName or ColumnName, [ColumnName1, ColumnName2, ...]) |
| LOOKUPVALUE | Retrieves a value for a row meeting specified criteria. | = LOOKUPVALUE(ResultColumnName, SearchColumnName1, SearchValue1, ..., [AlternateResult]) |
| ALLEXCEPT | Removes all context filters in the table except filters that have been applied to the specified columns. | = ALLEXCEPT(TableName, ColumnName1, ...) |
| ALLNONBLANKROW | Returns all rows or distinct column values from the parent table, excluding the blank row, and ignores context filters. | = ALLNONBLANKROW(TableName, ColumnName1, ColumnName2, ...) |
| ALLSELECTED | Removes context filters, preserving other filters in the current query. | = ALLSELECTED(TableName, ColumnName1, ColumnName2, ...) |
| CALCULATE | Evaluates an expression in a modified filter context. | = CALCULATE(Expression, Filter1, Filter2, ...) |
| FILTER | Returns a table that represents a subset of another table or expression. | = FILTER(Table, FilterExpression) |
| REMOVEFILTERS | Clears filters from the specified tables or columns. | = REMOVEFILTERS(TableName, ColumnName1, ColumnName2, ...) |
| SELECTEDVALUE | Returns the value for a filtered context, Otherwise returns alternate result. | = SELECTEDVALUE(ColumnName, [AlternateResult]) |

## Table Manipulation

These functions return a table or manipulate tables

| | | |
|---|---|---|
| ADDCOLUMNS | Adds calculated columns to the given table or table expression. | = ADDCOLUMNS(Table, Name1, Expression1, Name2, Expression2, ...) |
| CROSSJOIN | Returns a table with the Cartesian product of rows from specified tables. | = CROSSJOIN(Table1, ...) |
| EXCEPT | Returns rows from one table that are not present in another table. | = EXCEPT(LeftTable, RightRable) |
| GENERATESERIES | Returns a single-column table with values from an arithmetic series. | = GENERATESERIES(StartValue, EndValue, [IncrementValue]) |
| GROUPBY | Similar to SUMMARIZE but does not implicitly CALCULATE for extension columns. | = GROUPBY(Table, [Groupby_ColumnName], ..., [Name1], [Expression1, ...]) |
| INTERSECT | Returns the intersection of two tables, keeping duplicate rows. | = INTERSECT(LeftTable, RightTable) |
| NATURALINNERJOIN | Join two tables using an inner join. | = NATURALINNERJOIN(LeftTable, RightTable) |
| NATURALLEFT-OUTERJOIN | Performs a join of the Left Table with the Right Table. | = NATURALLEFTOUTERJOIN(LeftTable, RightTable) |
| ROW | Returns a table with a single row containing values resulting from expressions given to each column. | = ROW(Name1, Expression1, ...) |
| SELECTCOLUMNS | Returns a table with selected column from the table | = SELECTCOLUMNS((Table, Name1, Expression1, Name2, Expression2, ...) |
| SUMMARIZE | Returns a summary table for the requested totals over a set of groups. | = SUMMARIZE(Table, [Groupby_ColumnName], ..., [Name1], [Expression1, ...]) |
| TREATAS | Applies the result of a table expression as filters to columns from an unrelated table. | = TREATAS(Expression, ColumnName1, ...) |
| UNION | Combines tables vertically, preserving same rows. | = UNION(Table1, Table2, ...) |
| VALUES | Returns a one-column table with unique values from the specified table or column. | = VALUES(TableName or ColumnName) |

## Time Intelligence

Manipulates data for comparing calculations over time

| | | |
|---|---|---|
| DATEADD | Returns a table with dates shifted by a specified number of intervals from the current context. | = DATEADD(Dates, NumberofIntervals, Interval) |
| DATESBETWEEN | Returns a table with dates between a specified start date and end date, inclusive. | = DATESBETWEEN(Dates, StartDate, EndDate) |
| DATESINPERIOD | Generates a table with dates from a specified start date for a specified number and type of intervals. | **= DATESINPERIOD(Dates, StartDate, NumberofIntervals, Interval)** |
| PARALLELPERIOD | Returns a table with dates parallel to a specified column, shifted by a number of intervals. | = PARALLELPERIOD(Dates, NumberofIntervals, Interval) |
| SAMEPERIOD-LASTYEAR | Returns a table with dates representing the same period one year back from the specified column in the current context. | = SAMEPERIODLASTYEAR(Dates) |
| TOTALYTD/QTD/MTD | TOTALYTD calculates the total year-to-date, TOTALQTD calculates the total quarter-to-date, and TOTALMTD calculates the total month-to-date across all dates. | = TOTALYTD(Expression, Dates, [Filter], [YearEndDate])<br>= TOTALQTD(Expression, Dates, [Filter])<br>= TOTALMTD(Expression, Dates, [Filter]) |
| PREVIOUSYEAR/<br>PREVIOUSQUARTER/<br>PREVIOUSMONTH | PREVIOUSYEAR calculates the period from the last date to the same date in the previous year, PREVIOUSQUARTER calculates the first date to one day before the end of the previous quarter, and PREVIOUSMONTH calculates the first date to one day before the end of the previous month in the current context. | = PREVIOUSYEAR(Date, [YearEndDate])<br>= PREVIOUSQUARTER(Date)<br>= PREVIOUSMONTH(Date) |

## Information

Check if a value in a cell or row matches the expected type

| | | |
|---|---|---|
| HASONEVALUE | Returns TRUE if the context has one distinct value; otherwise, FALSE. | =HASONEVALUE(ColumnName) |
| ISBLANK | Checks whether a value is blank, and returns TRUE or FALSE. | = ISBLANK(Value) |
| ISCROSSFILTERED | Returns TRUE if the specified column or related column is being cross-filtered; otherwise, FALSE. | = ISCROSSFILTERED(ColumnName or TableName) |
| ISERROR | Checks if a value is an error, returning TRUE or FALSE. | = ISERROR(Value) |
| ISFILTERED | Returns TRUE when the specified column is being directly filtered. | = ISFILTERED(ColumnName or TableName) |
| ISINSCOPE | Returns TRUE when the specified column is the level in a hierarchy. | = ISINSCOPE(ColumnName) |
| SELECTEDMEASURE | Used by calculation items to reference the measure in context. | = SELECTEDMEASURE() |

Learn DAX : https://learn.microsoft.com/en-us/dax/

Download this in high resolution: check the link in the description

Raj Maradiya

follow me for more content on **Power BI**

# Power BI Desktop Cheat Sheet

Download this in high resolution:

check the link in the description

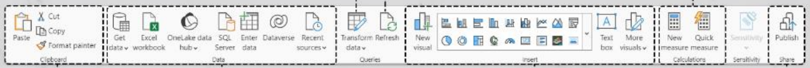
YOUR CHEAT SHEET

## Home Tab

Use Power Query editor to connect, prepare, transform the data

Update your reports with the latest data from connected sources

Use DAX to create new measure/metrics, or use Quick measure to create complex calculation



Copy, Paste & Format of visuals or objects

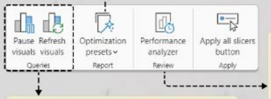
Connect to the various data sources & get the data from that sources

Add new visuals from visuals gallery , Add text box for comment & Add custom visuals from AppSource

Publish the report online in the Power BI Service

## Optimize Tab

Optimize your report setting for interactivity or for query reduction



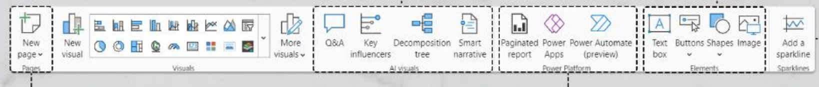
Assess your report's performance & identify areas for improvement

Queries allows you to pause or resume visuals for efficient performance during report editing

## Insert Tab

Q&A explores data with natural language, Key Influencer identifies metric influencers, Decomposition tree visualizes hierarchical data & Smart narrative automates insightful data summaries

Add different elements to the report



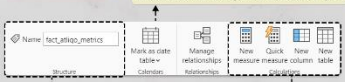
Add Blank page or Duplicate page to the report

Integrate PBI reports, Power Apps or Power Automate. This allows update to other reports, access apps or trigger workflows without leaving Power BI

sparkline is like a tiny trend chart you can place in a cell in a table or matrix visuals

## Table Tool

Choose a table to represent dates in your report. It'll be the reference for creating hierarchies and doing time-related analyses



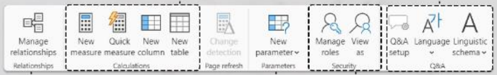
Change the name of the table

Create new metrics, column & tables using this calculations tools

## Modeling Tab

Create new metrics, column & tables using this calculations tools

Q&A Set up



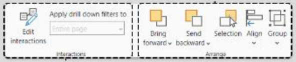
Add, edit or remove relationship between tables

Create new parameters for dynamic adjustments

Use this options for Row Level Security

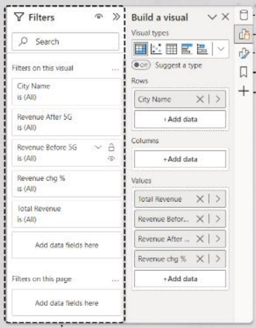
## Format Tab

Arrange visuals & objects using this options



Edit interactions to control the visual interactivity

## Filter/Task pane in Report view



Data

Build a visual

Format

Bookmarks

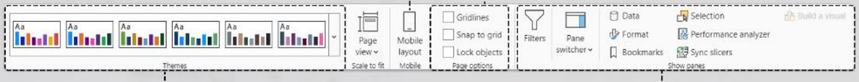
Customize pane switcher

Utilize the filter pane to manage & apply filters for focused analysis on visuals, pages, or the entire report

## View Tab

Design your report for Mobile devices

Use gridlines for alignment, snap to grid for precision & lock objects to secure report elements



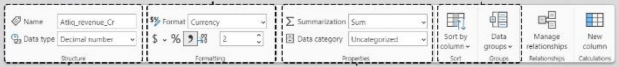
This option allows user to apply pre-defined or custom color schemes to enhance the visual aesthetics of their reports

Show panes is used to Show/Hide this options in report view

## Column Tool

Change the format of the column

Sort by column is use to arrange the data based on the values of another column & Data groups is use to create new group to combine multiple values into one column



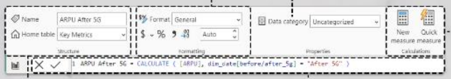
Use this options to change the name & data types of the column

Summarize sets aggregation for numerical columns, while "Data category" categorize columns for optimize visual representation

## Measure Tool

Change the format of the measure/metrics

Change the category of the measure/metrics



```
1 ARPU After 5G = CALCULATE ( [ARPU], dim_date[before/after_5g] = "After 5G" )
```

Use this options to change the name & Destination of the measure/metrics

Formula Bar

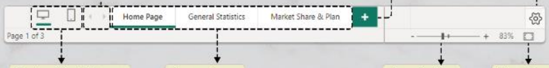
Use DAX to create new measure/metrics, or use Quick measure to create complex calculation

## Footer

Previous/Next page

New Page

Setting



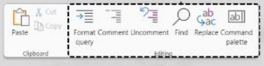
Desktop/Mobile layout

Report Pages

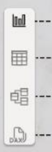
Zoom In/Out

Fit to Page

## DAX Query Tab



1. Format the DAX Query to improve readability
2. Comment/uncomment the selected lines
3. Find & Replace the text in the current DAX Query
4. Search for commands in DAX Query view



Report view

Table view

Model view

DAX Query view

https://www.novypro.com/profile_about/rajmaradiya

Download this in high resolution: check the link in the description

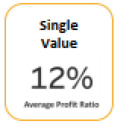
**Raj Maradiya**

follow me for more content on **Power BI**

# DATAcated® Chart Selector Guide

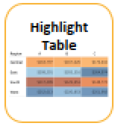
## What type of data would you like to present?

### Specific Value



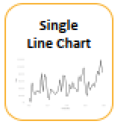
**Single Value**: Show the raw number prominently displayed

**Table**: Show the exact values and compare pairs of related values

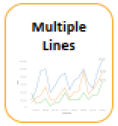


**Highlight Table**: Show the exact values and use color to convey relative magnitude

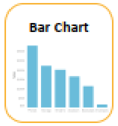
### Comparison



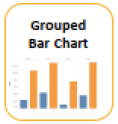
**Single Line Chart**: Display trends over a period of time for a single category



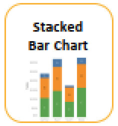
**Multiple Lines**: Display trends over a period of time for multiple categories



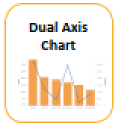
**Bar Chart**: Shows comparisons among discrete categories



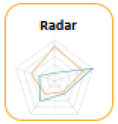
**Grouped Bar Chart**: Shows comparisons among discrete categories and sub-categories



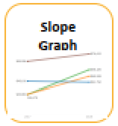
**Stacked Bar Chart**: Shows comparisons among discrete categories and sub-categories



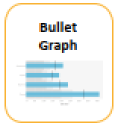
**Dual Axis Chart**: Show the relationship between two variables with different magnitudes and scales



**Radar**: Plots one or more series of values over multiple quantitative variables

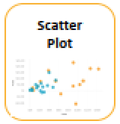


**Slope Graph**: Compares a data point, typically between two points in time

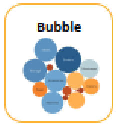


**Bullet Graph**: Compares data against historical performance or pre-assigned thresholds

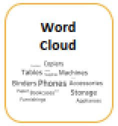
### Relationship



**Scatter Plot**: Shows the relationship between two variables

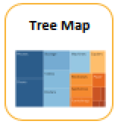


**Bubble**: Shows relational value without regards to axes

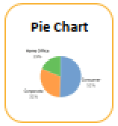


**Word Cloud**: Shows the relative frequency of words in our data

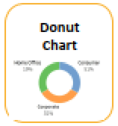
### Composition



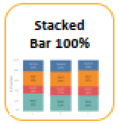
**Tree Map**: Shows a hierarchical part-to-whole relationship



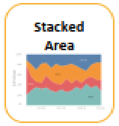
**Pie Chart**: Shows a part-to-whole relationship



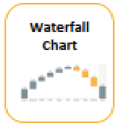
**Donut Chart**: Shows a part-to-whole relationship



**Stacked Bar 100%**: Shows a part-to-whole relationship across categories

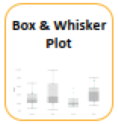


**Stacked Area**: Shows a part-to-whole relationship over a period of time

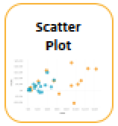


**Waterfall Chart**: Shows how a value changes by various factors that either increase the value, or decrease it

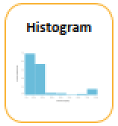
### Distribution



**Box & Whisker Plot**: Displays the data distribution through quartiles



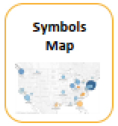
**Scatter Plot**: Shows the relationship between two variables



**Histogram**: Show the underlying shape of a set of continuous data

### Geographic

**Filled Map**: Shows geographic data using shading on a country or state basis to indicate relationships



**Symbols Map**: Shows geographic data using a symbol plotted over a longitude and latitude

DATAcated® Academy

The *DATAcated® Chart Selector Guide* is meant to be used as a guide selecting charts to use in your data visualizations