POWER BI INTERVIEW QUESTIONS
0-3 YOE
8-13 LPA

1. Difference Between Calculated Columns and Measures

Both Calculated Columns and Measures are DAX (Data Analysis Expressions) formulas in Power BI, but they serve different purposes and behave differently.

Feature	Calculated Column	Measure
Evaluation Context	Row Context (evaluated row by row)	Filter Context (evaluated on aggregation/filtering)
Storage	Stored in memory; adds to the data model size	Calculated on the fly; does not increase data size
Use Case	When you need new data columns (e.g., categorizing sales into segments)	When you need aggregations (e.g., total, average, YTD)
Visibility	Appears as a column in data table and visual fields	Appears in the fields pane but not as a table column
Example	Profit = Sales[Revenue] - Sales[Cost]	Total Sales = SUM(Sales[Revenue])

🟧 **When to use Calculated Column:**

- To create slicers or filters based on new fields.
- To perform row-wise calculations (e.g., Age from DOB).

🟦 **When to use Measure:**

- To calculate KPIs, totals, averages, or percentages.
- For optimized performance in reports and visuals.

---

2. How Do You Handle Missing Values in Power BI?

Handling missing values ensures data accuracy and meaningful visuals. Here's how it's generally done in Power BI:

✅ **1. Power Query Editor (Pre-processing Stage)**

You can handle missing values using the **Power Query Editor** (Transform Data window):

- **Remove Nulls:**
  Right-click the column → "Remove Empty" or use filter to exclude nulls.
- **Replace Nulls:**
  - Replace with default value: null → 0, Unknown, or Not Available.
  - Right-click column → "Replace Values" → Replace null with desired value.
- **Fill Down/Up:**
  Useful for forward or backward filling in grouped data.
  Example: If dates are missing in rows, "Fill Down" can propagate previous values.

✅ **2. DAX (Calculated Fields or Measures)**

Use DAX functions to handle missing values during calculations:

- **IF(ISBLANK(...), ..., ...)**: Handle blanks in logic.

Adjusted Revenue = IF(ISBLANK(Sales[Revenue]), 0, Sales[Revenue])

- **COALESCE()**: Returns the first non-blank value.

Profit = COALESCE(Sales[Profit], 0)

✅ **3. Visual-Level Handling**

In visuals:

- Use filters to **exclude blank values**.
- Use **conditional formatting** to highlight missing or zero data.

Would you like a sample Power BI use case or visual explanation for either of these topics?

3. Difference Between SUM and SUMX in Power BI

Both SUM and SUMX are DAX functions used to perform addition, but **they work differently**.

Feature	SUM	SUMX
Function Type	Aggregator	Iterator
Purpose	Directly sums up values from a column	Iterates through a table row by row and sums up the expression result
Input	Single column name	Table and expression
Performance	Faster for simple column sums	Slightly slower due to row-by-row calculation

✅ **Usage Example:**

SUM Example:

Total Sales = SUM(Sales[Revenue])

- It simply adds up all the values in the Revenue column.

SUMX Example:

Total Revenue = SUMX(Sales, Sales[Quantity] * Sales[Price])

- It first multiplies Quantity * Price for each row and then sums the result — which SUM cannot do.

🟢 **When to use SUMX:**

- When the values to be summed are **derived** (like multiplication or conditions).
- When working with **related tables** and need row-wise calculation.

---

4. What is DAX? Give an Example. Which DAX Functions Did You Use in Projects?

✅ **What is DAX?**

DAX (Data Analysis Expressions) is a formula language used in Power BI, Excel Power Pivot, and SSAS Tabular models to create **calculated columns**, **measures**, and **custom tables**.
It is similar to Excel formulas but is **optimized for relational data models** and performs advanced analytics like filtering, row context, and aggregations.

✅ **Simple DAX Example:**

Calculated Column:
Profit = Sales[Revenue] - Sales[Cost]
Measure:
Total Sales = SUM(Sales[Revenue])
Conditional Measure:
High Revenue Sales = CALCULATE([Total Sales], Sales[Revenue] > 100000)

✅ **DAX Functions Commonly Used in Projects:**

Category	Functions Used	Use Case
Aggregation	SUM, AVERAGE, COUNT, DISTINCTCOUNT	Total and average KPIs
Iterators	SUMX, AVERAGEX	Row-wise operations (e.g., dynamic profit margins)
Filter	FILTER, CALCULATE, ALL, REMOVEFILTERS	Custom filtering and context switching
Logical	IF, SWITCH, AND, OR	Conditional logic for KPIs or flags
Time Intelligence	DATEADD, SAMEPERIODLASTYEAR, TOTALYTD, DATESINPERIOD	YTD, MTD, YoY comparisons
Text	CONCATENATE, LEFT, RIGHT, FORMAT	Formatting labels or combining fields
Handling Missing Values	ISBLANK, COALESCE	Replacing or identifying blanks in visuals

---

📝 **Real-World Example in Projects:**
In a sales dashboard, I used CALCULATE with FILTER to create dynamic measures like “High Value Customers” and “YoY Sales Growth”. I also used SUMX to calculate row-level profitability using SUMX(Sales, Sales[Qty] * (Sales[Price] - Sales[Cost])).

5. How Do You Disable a Graph That Is Changing Dynamically?

In Power BI, a graph or visual can change dynamically due to interactions with **filters, slicers, or other visuals** on the report page. To **disable or control this dynamic behavior**, you can use **Edit Interactions** and other techniques.

✅ **Method 1: Use “Edit Interactions”**
This is the most common and direct method.

🔧 **Steps:**

1. Click on the visual **that is triggering the change** (e.g., a slicer or another graph).
2. Go to the **"Format" ribbon** and select **"Edit Interactions"**.
3. For the visual **you want to disable**, click on the **“None” (circle with a slash)** icon.
    - This stops it from responding to changes in the triggering visual.

🟢 **Use Case Example:**

- You have a pie chart and a bar chart. Clicking a slice in the pie chart changes the bar chart.
- You want the bar chart to **remain static** → Use *Edit Interactions* and disable the link from pie to

bar.

✅ **Method 2: Use Bookmarks and Selection Pane**

If you want to **hide or disable** a graph completely under certain conditions:

- Use **Bookmarks** to save the visibility state.
- Use the **Selection Pane** to control which visuals appear with each bookmark.
- Trigger bookmarks using **buttons** or **slicers**.

✅ **Method 3: Use Conditional Visibility**

Power BI doesn't natively support conditional visibility, but you can mimic it:

- Create a **transparent shape** that overlays the visual and blocks interaction.
- Use **measure-based transparency or visibility** in combination with bookmarks.

---

6. Explain RLS (Row-Level Security) and How Do You Implement It?

✅ **What is RLS?**

Row-Level Security (RLS) is a feature in Power BI that **restricts data access for users** based on their roles. It ensures that users **only see the data relevant to them**.

🎯 For example: A sales manager for Region A should not be able to see sales data for Region B.

✅ **How RLS Works:**

- You **define roles** with **DAX filters** that limit what rows are visible in a table.
- These filters apply at the **row level** during report interaction or sharing.

✅ **Steps to Implement RLS in Power BI:**

1. Go to Modeling → Manage Roles

- Create a new role (e.g., RegionManager).
- Apply a **DAX filter** to the relevant table.

Example:
[Region] = "West"
This ensures the user assigned to this role sees only data where Region = West.

---

2. Test the Role

- Click on **"View as Roles"** to test and validate if the filters are applied correctly.

---

3. Publish to Power BI Service

Once published:

- Go to **Power BI Service** → Workspace → Dataset → **Security**.
- Assign **email addresses (users)** to the defined role.

✅ **Advanced RLS with Dynamic Security**

Dynamic RLS uses a **mapping table** (e.g., UserEmail → Region), allowing one report to serve multiple users with personalized data.

- Example: SalesRegionAccess table

  UserEmail **Region**

UserEmail	Region
alice@example.com	West
bob@example.com	East

- DAX Filter:

[Region] IN LOOKUPVALUE(SalesRegionAccess[Region], SalesRegionAccess[UserEmail], USERPRINCIPALNAME())

✅ **Best Practices for RLS:**

- Always test roles using "View as Roles".
- Avoid applying RLS on calculated tables or complex relationships if unnecessary.
- Use dynamic RLS for scalability in enterprise-level reports.

---

7. Types of Filters in Power BI

Power BI offers **multiple types of filters** to control and customize data views at different levels within a report. These filters help in slicing the data for specific insights.

✅ **1. Visual-Level Filters**

- Applied to **individual visuals** (charts, tables, cards, etc.).
- Affects only **that one visual**.
- Useful when you want to filter a chart independently from others.

Example: Show only “Product Category = Electronics” in a specific bar chart.

✅ **2. Page-Level Filters**

- Applied to **all visuals** on a single report page.
- Does **not** affect visuals on other pages.

Example: Apply “Country = India” on a page to show only Indian sales data across all visuals.

✅ **3. Report-Level Filters**

- Applied to **all pages** and **all visuals** in the report.
- Great for global filters like Time Period or Brand.

Example: Filter for “Year = 2024” across the entire report.

✅ **4. Drillthrough Filters**

- Allows users to **right-click a data point** and **navigate to a different page** with **context-specific data**.
- Applied to the **drillthrough page** only when a user triggers it.

Example: From a sales summary page, right-click on “West Region” and drillthrough to a page that shows detailed West Region sales.

✅ **5. Slicer Filters**

- Visual controls (dropdown, list, etc.) added by the developer.
- Allow **end-users** to control filtering dynamically.

Example: A date slicer that lets the user select a range and filters all connected visuals accordingly.

✅ **6. Cross Filtering and Cross Highlighting**

- Happens when you **click a data point** in one visual and it **automatically filters or highlights** related visuals.
- Controlled using **"Edit Interactions"**.

✅ **7. URL Filters (in Power BI Service)**

- Enables filtering a report page through URL query strings.
- Useful for **embedding** reports or passing filters from external systems.

---

8. Difference Between Drill Down and Drill Through in Power BI

These are both **navigation tools** used to explore data at different levels of granularity, but they function differently:

Feature	Drill Down	Drill Through
Purpose	Navigate to **lower levels** in the same visual (e.g., Year → Month → Day)	Navigate to a **different page** with detailed data
Interaction	Done by **clicking a data point** (usually arrows or right-clicking)	Done by **right-clicking a data point** and choosing a drillthrough page
Where It Happens	Inside the **same visual or report page**	Takes you to a **new report page** designed for detailed insights
Data Flow	Uses **hierarchies** (e.g., Date, Geography)	Uses **filters passed between pages**
Use Case	Explore sales over time or breakdowns by category	Show a customer's full transaction history when clicked from summary page

---

✅ **Drill Down Example:**

You have a bar chart showing **Total Sales by Year**. When you drill down, it changes to show **Sales by Quarter**, then **by Month**, and so on.
Hierarchy setup: Year → Quarter → Month → Day

✅ **Drill Through Example:**

You have a **summary report** showing Sales by Region. You right-click on "North Region" and select **Drill Through → Region Details**. It navigates to a different page showing detailed charts for only the **North Region**.
You must configure **Drillthrough filters** on the target page.

---

9. Explain the Process of Publishing the Report in Power BI

Publishing a report means taking your Power BI Desktop (.pbix) file and making it available in the **Power BI Service (cloud)** for sharing, collaboration, and scheduled refresh.

✅ **Step-by-Step Process to Publish a Report:**

🔹 **1. Save Your Report**

- File is saved with a .pbix extension in Power BI Desktop.
- Ensure the data model and visuals are finalized.

🔹 **2. Sign in to Power BI**

- Open Power BI Desktop.
- Click **Sign In** (top right) using your Power BI account (usually work or school email).

🔹 **3. Click on Publish**

- Go to the **"Home" ribbon** → Click **"Publish"**.
- Select the **Workspace** where you want to publish the report.

📌 **Note:** Workspaces are like folders in Power BI Service that hold datasets, reports, and dashboards.

🔹 **4. Publishing Confirmation**

- After publishing, a message appears:
  "Successfully published to Power BI".
- Option to **open in Power BI Service** directly via a link.

🔹 **5. Post-Publish Tasks (in Power BI Service):**

- **Set up scheduled refresh** (especially for Excel/SQL/dataflow sources).
- Share report via:
  - Share button (email invite)
  - Embedding in apps/Teams
  - Publish to web (for public reports only)
- Create **dashboards** by pinning visuals from the report.

✅ **Best Practices:**

- Use **gateways** for on-premise data refresh.
- Apply **Row-Level Security (RLS)** before publishing.
- Test performance and access permissions.

---

10. How Do You Create Relationships in Power BI? What Are the Types of Cardinality?

✅ **What is a Relationship?**

Relationships in Power BI define how data in one table relates to data in another. This is essential for building a proper data model.

✅ **How to Create Relationships:**

🔹 **Method 1: Auto-Detect**

- Power BI **automatically creates relationships** based on common column names and data types.
- Appears in the **Model View** as lines between tables.

🔹 **Method 2: Manual Creation**

1. Go to **Model View** in Power BI Desktop.
2. Drag a column from one table to another (usually dimension → fact).
3. Or click on **"Manage Relationships"** → **"New"**.
4. Define:
   - **Table and columns**

- **Cardinality**
- **Cross filter direction**

✅ **Types of Cardinality in Power BI:**

Cardinality describes the **nature of the relationship** between two tables.

Cardinality Type	Description	Example
One-to-Many (1:*)	A single value in one table matches multiple in another. Most common.	One customer → many orders
Many-to-One (*:1)	In reverse, often interpreted same as 1:* (depends on drag direction).	Many orders → one product
Many-to-Many (:)	Both sides contain duplicate values; used with care.	Products sold in multiple regions with multiple overlaps
One-to-One (1:1)	Each row in Table A matches only one row in Table B.	Employee table ↔ HR details table

---

✅ **Cross Filter Direction:**

- **Single**: Default. Filters flow from dimension to fact.
- **Both**: Enables bi-directional filtering. Needed for complex relationships like M:M but should be used cautiously.

✅ **Important Notes:**

- **Avoid circular dependencies** in relationships.
- Always ensure that **key columns** (like IDs) are unique in the dimension table.
- Review relationships using the **Model View** to keep the schema clean.

---

11. How Do You Schedule a Refresh in Power BI?

Scheduled Refresh in Power BI ensures your reports and dashboards stay updated with the latest data at set intervals. This is done after publishing your report to the **Power BI Service**.

✅ **Steps to Schedule a Refresh:**

🔹 **1. Publish the Report**

- First, publish your .pbix file from Power BI Desktop to the **Power BI Service**.

🔹 **2. Open Power BI Service**

- Go to https://app.powerbi.com
- Navigate to the **workspace** where your dataset is published.

🔹 **3. Go to Dataset Settings**

- Click on **"Datasets + Dataflows"**.
- Click the **more options (⋮)** next to your dataset → choose **"Settings"**.

🔹 **4. Set Up Gateway (if using on-premises data)**

- If your data source is on-prem (e.g., SQL Server, Excel), you must install and configure a **Power BI Gateway**.
- Choose the appropriate **data gateway connection**.

🔹 **5. Configure Credentials**

- Under "Data Source Credentials", click **Edit Credentials** and choose the correct authentication method:
  - Windows
  - Basic
  - OAuth2
  - Web API

🔹 **6. Enable Scheduled Refresh**

- Scroll down to the **Scheduled Refresh** section.
- Turn the **"Keep data updated"** switch ON.
- Set the **frequency**:
  - **Daily** or **Weekly**
  - Up to **8 times/day** for Pro users and **48 times/day** for Premium capacity.
- Set **time zones**, add failure notifications if needed.

🔹 **7. Save Settings**

- Click **Apply** or **Save** to complete the refresh setup.

✅ **Best Practices:**

- Always test manual refresh before scheduling.
- Monitor refresh failures in the **Refresh history** section.
- Avoid large datasets with heavy transformations in Power Query for smoother refreshes.

---

12. What Type of Account Did You Use While Working on the Project? Explain the Features.

Power BI offers different types of user accounts and licenses. Your answer depends on your **project environment**, but here's a structured explanation:

✅ **Common Account Types in Power BI:**

🔹 **1. Power BI Free Account**

- Used mainly for building reports in **Power BI Desktop**.
- Cannot **share or publish** to Power BI Service.
- No access to **workspaces**, dashboards, or collaboration.
- Suitable for **personal use** or learning.

🔹 **2. Power BI Pro Account** ✅ ***(Most Common in Projects)***

- Allows full capabilities:
  - Publish reports to Power BI Service
  - Share dashboards and reports with others
  - Create and manage **workspaces**
  - Set up **Scheduled Refresh**
  - Use **RLS (Row-Level Security)**
  - Use **Power BI apps**

Example Use in Project:
"We used Power BI Pro accounts to build and publish reports to shared workspaces. This allowed collaboration among stakeholders, automated refreshes, and implementation of RLS for secure access."

🔹 **3. Power BI Premium Per User (PPU)**

- Same features as Pro + extra:
  - Larger datasets (100 GB)
  - 48 refreshes/day
  - Paginated reports
  - AI capabilities (AutoML, Cognitive Services)
- Used for enterprise-grade performance.

🔹 **4. Power BI Premium Capacity**

- Organizational license, not individual.
- Enables deployment to **dedicated capacity**.
- Needed for larger deployments, external sharing without Pro licenses for viewers.

✅ **Key Features of Power BI Pro Account (commonly used):**

Feature	Description
Collaboration	Create, publish, and share reports in workspaces
Scheduled Refresh	Up to 8 refreshes per day
RLS Support	Restrict data access at user level
App Workspaces	Organize content for teams/projects
Export & Embed	Export to PDF/PPT or embed in Teams/SharePoint
Usage Metrics	Track report and dashboard usage

---

13. How Do You Perform Join Operations in Power BI?

In Power BI, **join operations** are primarily done using:

✅ **1. Power Query (M Language) – Merge Queries**

Power Query allows you to **join tables** using the **Merge Queries** option.

🔹 **Steps:**

1. Go to **Home → Transform Data**.
2. In the Power Query Editor, click on **"Merge Queries"**.
3. Choose the two tables you want to join.

4. Select the matching columns from both.
5. Choose the **Join Type** (explained below).
6. Expand the merged column to include required fields.

🔸 **Types of Joins:**

Join Type	Description
Left Outer	All rows from the first table, matching rows from the second
Right Outer	All rows from the second table, matching rows from the first
Inner	Only matching rows from both tables
Full Outer	All rows from both tables
Anti Join (Left/Right)	Non-matching rows only
Inner Anti Join	Opposite of Inner Join

✅ **2. Relationships in Data Model**

Instead of physically joining tables, Power BI prefers establishing **relationships** between tables.

🔹 **Steps:**

- Go to **Model View**.
- Drag a column from one table to another (usually keys).
- Define relationship type and cardinality (e.g., One-to-Many).
- Relationships enable filtering across visuals.

✅ **3. DAX Functions for Joins**

- Use DAX for calculated tables or columns:

```
JoinedTable = NATURALINNERJOIN(Table1, Table2)
```

Or with RELATED() and LOOKUPVALUE() to fetch values from related tables.

---

14. What Are the Best Practices in Power BI?

Following best practices ensures a **high-performance**, **readable**, and **maintainable** Power BI solution.

✅ **Data Modeling Best Practices**

- Use **star schema**: fact + dimension tables.
- Avoid snowflake schema unless necessary.
- Create **proper relationships** (prefer One-to-Many).
- Avoid bi-directional relationships unless essential.

✅ **Data Loading**

- Disable **auto date/time** (in Options → Data Load).
- Remove unnecessary columns/rows in Power Query.
- Use **query folding** wherever possible.

✅ **DAX & Measures**

- Prefer **measures over calculated columns** (memory efficient).
- Use CALCULATE() and FILTER() wisely to avoid performance issues.

- Name measures clearly (Total Sales, Avg Discount, etc.).

✅ **Visual Design**

- Avoid overloading reports with too many visuals.
- Maintain **consistent color theme**.
- Use **tooltips** and **titles** for clarity.
- Follow **report layout best practices** (Z-pattern or top-down logic).

✅ **Performance Optimization**

- Use **aggregated tables** where possible.
- Avoid using columns with high cardinality (e.g., GUIDs).
- Optimize **DAX code** with variables and minimal row context.
- Monitor performance using **Performance Analyzer**.

✅ **Security and Sharing**

- Implement **Row-Level Security (RLS)** as needed.
- Keep **sensitive data encrypted** or masked.
- Share via **Power BI Service Workspaces**, not by sending .pbix files.

---

15. Difference Between Slicer and Filter in Power BI

Both **Slicers** and **Filters** are used to **restrict data visibility**, but they differ in scope, UI, and use cases.

Aspect	Slicer	Filter
Location	Added as a **visual** on the report canvas	Applied through **Filter pane**
Visibility	Always visible to users	Can be hidden from report viewers
User Interaction	Users can interact directly with slicers	Filters may not be interactive for users
Scope	Can apply to visual, page, or all pages	Can apply to visual, page, or report level
UI Elements	Visual controls like dropdowns, list, sliders	No visual control, used in pane
Types	Basic, hierarchical, relative date, etc.	Manual, advanced, top N, relative, etc.

✅ **Example Use:**

- Use **Slicer** when you want users to actively select filters (e.g., filter by region).
- Use **Filter** when you want to **limit data silently** without user control (e.g., exclude discontinued products).