

# Power BI Assignment Q & A

29 July 2025 02:26 PM

## 1) What is Power BI, and how does it differ from Excel?

Power BI is Microsoft's self-service business intelligence platform used to ingest, model, analyze, and share interactive data visualizations. While Excel also offers analytics features (PivotTables, formulas, Power Query, Power Pivot), Power BI is purpose-built for:

| Aspect        | Power BI  | Excel  |
|---------------|---|--|
| Data capacity | Handles large, column-store (VertiPAQ) models that can exceed millions of rows with high compression. | Limited by worksheet row/column limits and memory.                       |
| Data model    | Central semantic model, reusable across many reports, multiple fact tables, relationships, and KPIs.  | Each workbook has its model; complex star/snowflake models are uncommon. |
| Sharing       | Cloud & on-prem publishing, dashboards, apps, row-level security, automatic refresh, governance.      | Email or SharePoint file sharing, and collaboration via co-authoring.    |
| Interactivity | Cross-filtering, drill-through, bookmarks, natural-language Q&A, real-time tiles.                     | Slicers, timelines; less seamless.                                       |
| Deployment    | Desktop → Service/Fabric → mobile, Teams, embedded.   | Desktop-centric; Power Automate or VBA for automation.                   |

## 2) Concept of data modelling in Power BI

Data modelling is the process of structuring imported tables into a relational (usually star-schema) model that the VertiPaq engine can query efficiently. It involves:

- Defining fact and dimension tables.
- Creating relationships (one-to-many, many-to-many, bi-directional when needed).
- Setting filter directions and cardinality.
- Creating calculated columns, measures, hierarchies, and role-playing dimensions (e.g., multiple date relationships).  
Good modeling yields faster queries, simpler DAX, and more intuitive visuals.

## 3) Types of connections available in Power BI

| Connection mode   | How it works  | Typical sources   |
|---|---|---|
| Import  | Data is loaded into the local VertiPaq cache (.pbix). Fastest query performance; requires refresh.                                | CSV, SQL, web APIs, etc.                                      |
| DirectQuery   | Queries the source at run-time; minimal local storage. Useful for very large or real-time sources; performance depends on source. | Azure Synapse, Snowflake, SAP HANA, etc.                      |
| Live Connection   | Connects to an Analysis Services/Power BI semantic model; can't create new data model in the report.                              | Azure/SQL Server Analysis Services, Power BI semantic models. |
| Composite/Hybrid (Import + DirectQuery) & DirectLake (Fabric) | Mix of Import and DirectQuery tables; DirectLake reads Delta tables in OneLake with in-memory speed.                              | Fabric OneLake, mixed enterprise sources.                     |

## 4) Handling data transformation in Power BI

Power Query Editor (M language) performs Extract-Transform-Load (ETL):

1. Connect to myriad sources.
2. Apply step-by-step transforms: filtering, splitting, pivots/unpivots, joins, data types, fuzzy matching, etc.
3. Leverage parameters and functions for reuse.
4. Load the cleaned tables into the model (Import) or leave transformations in the source (DirectQuery folding).  
Advanced: use dataflow Gen 2 / Fabric Data Factory for centralized transforms.

## 5) What is DAX and why is it important?

Data Analysis Expressions (DAX) is a functional, Excel-like formula language used to define calculated columns, measures, and table expressions in the model. It's crucial because:

- It adds business logic that raw data lacks (KPIs, YTD, MoM, ratios).
- It respects filter context, enabling dynamic calculations tied to slicers and visual interactions.
- It powers advanced analytics such as time-intelligence, ranking, parent-child hierarchies, and semi-additive measures.

## 6) Calculated columns vs. measures

| Feature         | Calculated Column                                    | Measure  |
|-----------------|--|--|
| Evaluation time | Calculated during data refresh; stored in the model. | Calculated at query time, respecting current filter context. |
| Storage         | Consumes RAM in VertiPaq.                            | Virtual—no extra storage.                                    |
| Granularity     | Row context (one value per row).                     | Aggregate context (one value per cell in a visual).          |
| Use cases       | Data prep: classifications, concatenations, keys.    | Metrics: totals, averages, YoY %, dynamic ratios.            |

## 7) Handling relationships between tables

- Create relationships in the Model view.
- Choose correct cardinality (1:\* , :) and, when needed, many-to-many via a bridge table.
- Control filter direction (single vs. bi-directional) to prevent ambiguity/performance issues.
- Enable “Assume referential integrity” for DirectQuery when the keys truly match—improves query folding.
- Use USERELATIONSHIP in DAX for inactive relationships (e.g., Order Date vs. Ship Date).

## 8) Purpose of a Power BI Gateway

The on-premises data gateway securely bridges cloud Power BI Service (and Fabric, Power Apps, Power Automate) with on-prem or VNet-isolated data sources:

- Encrypts queries in transit, no inbound firewall holes.
- Supports both Standard (personal mode per user) and Enterprise (central, high-availability clusters).
- Required for scheduled refresh and DirectQuery/live reports against on-prem sources.

## 9) Scheduling data refresh in Power BI Service

1. Publish the dataset (.pbix) to a workspace.
2. In Dataset → Settings → Scheduled refresh, set frequency (up to 8× daily for Pro, 48× for Premium/Fabric).
3. Provide credentials and gateway mapping (if on-prem).
4. Configure failure notifications and incremental refresh policies for large fact tables (refresh only new/changed partitions).

## 10) Row-level security (RLS)

RLS restricts data a user can see to only their authorized subset:

- Define DAX filter rules on tables (e.g., [Region] = USERPRINCIPALNAME ()).
- Assign users or AAD groups to roles in the Service or via apps.
- Works in Import, DirectQuery, and Hybrid; with Azure Analysis Services/Power BI semantic models, can combine with object-level security (OLS) for column/table hiding.

## 11) Power BI Desktop vs. Power BI Service

| Feature            | Power BI Desktop   | Power BI Service / Fabric  |
|--------------------|--|--|
| Purpose            | Authoring tool: connect, transform, model, design visuals; free Windows app. | Cloud platform: sharing, collaboration, refresh, governance, metrics, apps, Fabric workloads.      |
| Data model editing | Full (relationships, measures, calculation groups).                          | Limited (rename measures, change Q&A synonyms; no relationship edits).                             |
| DAX authoring      | Full editor, Performance Analyzer, DAX Query View (since 2024).              | Quick measures only; no advanced editor.   |
| Deployment         | Publish to workspace (.pbix or .pbit).                                       | Host datasets and thin reports; manage security, refresh, subscriptions, pipelines.                |
| Extras             | Local preview features, external tools (Tabular Editor, DAX Studio).         | Dashboards (pin visuals), dataflows, paginated reports, Scorecards, mobile, embed, XMLA endpoints. |

## 12) Explain the concept of DirectQuery in Power BI.

DirectQuery is a data connection method where Power BI doesn't import data into its model but queries the data source in real-time whenever a user interacts with a visual.

### Key Points:

- Data stays in the source (e.g., SQL Server, Snowflake, SAP).
- Minimal local storage; suitable for large datasets.
- Can reflect real-time changes.
- Slower performance than Import mode.
- Limited DAX & Power Query functionality.
- Requires optimized source queries and indexing.

## 13) What are Power BI templates and how are they useful?

A Power BI Template (.pbit file) is a reusable file that saves:

- Report layout
  - Data model
  - Queries
- But not the actual data.

### Use Cases:

- Sharing reusable report structures with clients.
- Creating standardized reporting formats across teams.
- Parameterized reports where users input their own source paths.

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

Incremental Refresh loads only new or changed data instead of the full dataset every time.

### Steps:

1. Define RangeStart and RangeEnd parameters in Power Query.
2. Apply filters on date column using these parameters.
3. Configure refresh policy in Power BI Service (not Desktop):
  - e.g., keep last 5 years of data, refresh last 10 days only.
4. Publish to Premium or PPU workspace (required for full functionality).

Benefit: Faster refresh, lower memory usage, better performance.

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

Power Query (also known as Get & Transform) is the ETL tool within Power BI Desktop.

It allows users to:

- Connect to various data sources.
- Apply transformations using a step-based UI.
- Clean, shape, and combine data.
- Use the M language behind the scenes for custom transformations.

It is the first stage of Power BI's data pipeline before modeling and visualization.

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

| Feature       | Calculated Column                  | Calculated Table                 |
|---------------|------------------------------------|----------------------------------|
| Scope         | Adds a column to an existing table | Creates a new table in the model |
| Context       | Evaluated row by row (row context) | Evaluated once at load/refresh   |
| Use cases     | Categorization, new keys, flags    | Summarized tables, bridge tables |
| Storage       | Takes space in model               | Entire table stored              |
| Created Using | DAX (e.g., =IF(...))               | DAX (e.g., SUMMARIZE(...))       |

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

There are 3 ways:

1. Marketplace Visuals – Download from AppSource (e.g., Sankey, WordCloud).
2. Custom-developed visuals – Create using TypeScript, D3.js, and Power BI Developer Tools (pbiviz) and upload manually.
3. R and Python Visuals – Use custom script visuals for statistical or advanced plots.

Use Case: When native visuals do not meet complex design or interactivity needs.

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

- Use Import mode whenever possible
- Follow star schema modeling
- Reduce columns & cardinality
- Use measures, not calculated columns
- Filter data in Power Query, not visuals
- Disable unnecessary auto-date/time
- Use summary tables and aggregations
- Optimize DAX using variables, avoid CALCULATE misuse
- Enable Performance Analyzer and DAX Studio to detect bottlenecks
- Apply incremental refresh for large datasets

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

Power BI + Azure:

- Azure Synapse: For big data integration.
- Azure SQL DB: DirectQuery or Import connections.
- Azure ML: Predictive modeling integration.
- Azure Data Lake: Power BI can read parquet/Delta files.
- Azure Active Directory: Role-based access with RLS.

Power BI + Office 365:

- Excel: Import data/models, Analyze in Excel.
- SharePoint/OneDrive: Auto-refresh from cloud storage.
- Microsoft Teams: Embed reports in chats.
- Outlook: Subscribe and receive email reports.
- Power Automate: Automate alerts and actions.

20) Explain the concept of aggregations in Power BI.

Aggregations are summary tables used to optimize performance by reducing the granularity of queries.

Steps:

- Create a summary table (e.g., Sales by Year, Region).
- Define it as Aggregation table in Power BI.
- Configure it to handle SUM, COUNT, etc., via Manage Aggregations.
- Power BI engine will redirect queries to this small table when possible.

Result: Faster performance on large models without losing detail.

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

In Power Query:

- Use “Keep Errors” or “Remove Errors” steps.
- Apply data type validations.
- Replace or fill nulls using "Replace Errors" or "Fill Down".
- Create custom if-error handling using try...otherwise.

Data Profiling Tools:

- Column distribution, quality, and profiling features in Power Query.

Post-load checks:

- Use DAX checks (e.g., ISBLANK (), HASONEVALUE ())
- Visual alerts for missing/invalid data

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

Power BI Embedded is a PaaS (Platform-as-a-Service) offering for ISVs and developers to embed Power BI reports/dashboards into their own applications or websites.

Use cases:

- Customer-facing portals (e.g., SaaS apps).
- No need for users to have Power BI licenses.
- Full control over branding, UI, and navigation.
- Pay-per-use or capacity-based pricing.

Used when you want interactive analytics inside your own product, not the Power BI platform.