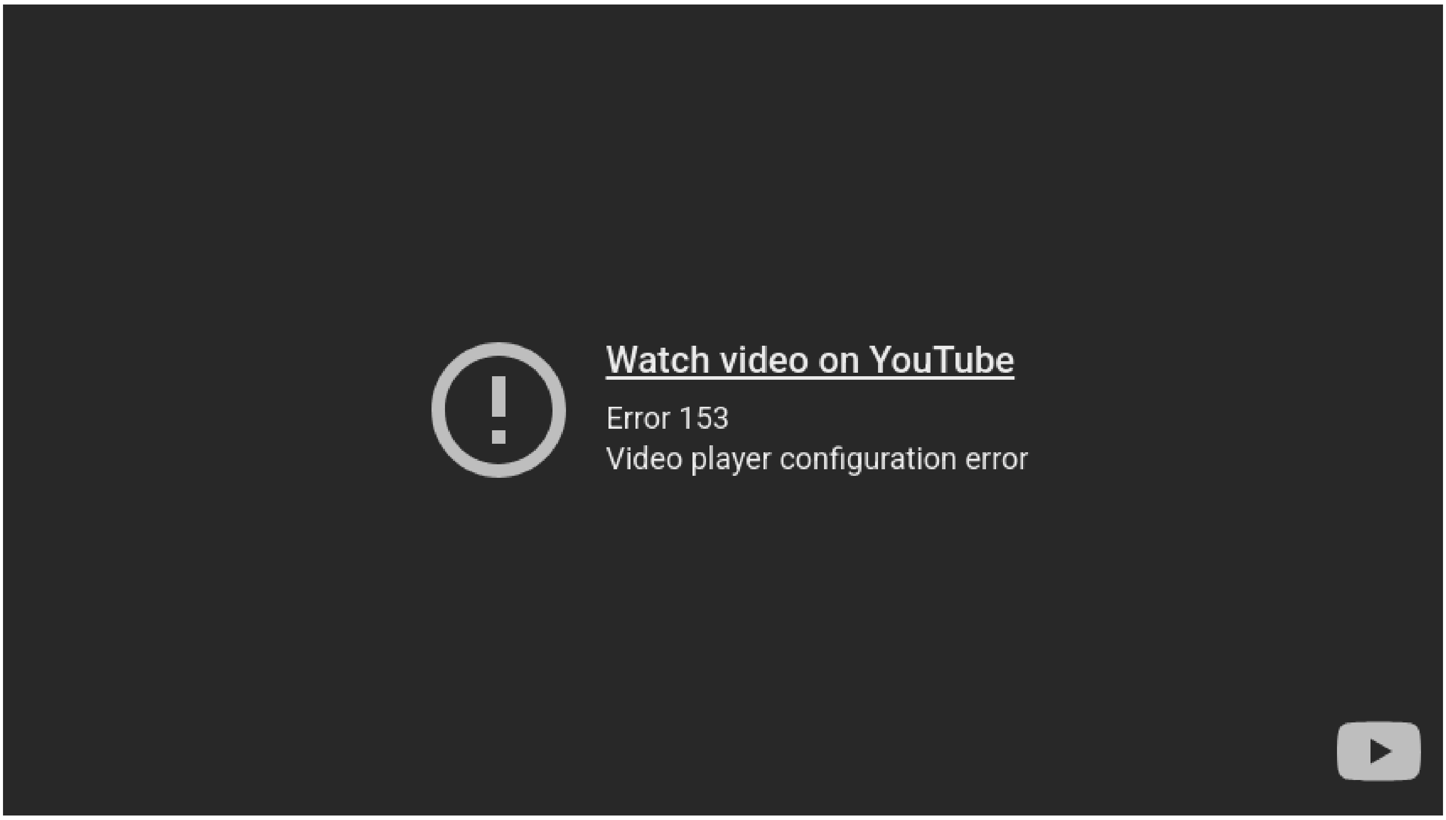
Data Analytics PowerBI

Power Bi All Questions


Watch video on YouTube

Error 153

Video player configuration error

- Q1. What is Power BI?
  Power BI is a collection of software services, apps, and connectors that work together to turn unrelated data sources into coherent, visually immersive, and interactive insights.
- Q2. What are the main components of Power BI?
  The main components are Power BI Desktop (report authoring), Power BI service (online SaaS for sharing and collaboration), Power BI Mobile apps, and Power BI Gateway for on-premises data connectivity.
- Q3. What type of tool is Power BI in the analytics ecosystem?
  Power BI is a self-service business intelligence and data visualization tool used for reporting, dashboards, and analytics, often by business analysts and data professionals.
- Q4. What are typical business applications of Power BI?
  It is used for interactive dashboards, KPI tracking, sales and marketing reports, financial analysis, operations monitoring, and executive management reports.
- Q5. Name three key capabilities of Power BI Desktop.
  It can connect to many data sources, transform and clean data using Power Query, and build data models with relationships and DAX measures to create rich visual reports.
- Q6. What types of data sources can Power BI connect to?
  Power BI connects to files (Excel, CSV), databases (SQL Server, Oracle, etc.), online services (SharePoint, OneDrive), web APIs, and many cloud services.
- Q7. How does Power BI help in decision-making?
  It provides real-time or regularly refreshed visual dashboards that summarize KPIs and

trends, enabling faster, data-driven decisions across departments.

- Q8. What is the difference between Power BI Desktop and Power BI service?
  Power BI Desktop is a Windows application used to build and design reports, while the Power BI service is the online platform for publishing, sharing, and collaborating on those reports.
- Q9. Why is Power BI called a "self-service" BI tool?
  Because business users without deep IT or coding skills can connect to data, create reports, and build dashboards themselves using a graphical interface.
- Q10. What is DAX in Power BI? (basic awareness for Unit I)
  DAX (Data Analysis Expressions) is a formula language used in Power BI to create calculated columns and measures for advanced calculations in data models.

Capabilities and Benefits

- Q11. What are some benefits of using Power BI?
  Benefits include rich interactive visuals, connectivity to diverse data sources, strong integration with Excel, robust data modeling, and easy sharing across the organization.
- Q12. How does Power BI handle large datasets?
  Power BI can import and compress large datasets (up to tens or hundreds of millions of rows) and offers DirectQuery to query some sources in real time.
- Q13. What is Power Query in Power BI?
  Power Query is the ETL (Extract, Transform, Load) engine in Power BI used to connect to data, clean it, merge tables, and perform transformations before loading into the model.
- Q14. How does Power BI integrate with Excel?
  Users can import Excel workbooks, use Excel files as data sources, pin Excel ranges to dashboards, and publish Excel-based models to Power BI.
- Q15. What is a Power BI dashboard?
  A dashboard is a single page, often called a canvas, containing tiles from one or more reports, giving a consolidated top-level view of metrics.
- Q16. What is a Power BI report?
  A report is a multi-page set of visuals built from a single dataset, created mainly in Power BI Desktop and then published to the service.
- Q17. How does Power BI support real-time analytics?
  Through streaming datasets, push datasets, and integration with services like Azure Stream Analytics to display live updating visuals.
- Q18. What is Power BI Embedded? (basic awareness)
  Power BI Embedded is a feature that lets developers embed Power BI dashboards and reports into custom applications and websites.
- Q19. Give one example of Power BI usage in finance.
  Finance teams use Power BI to build P&L dashboards, cash-flow reports, variance analysis, and KPI monitoring for revenue and expenses.
- Q20. Give one example of Power BI usage in sales and marketing.
  Sales teams track pipeline, conversion rates, and regional performance, while marketing analyzes campaign performance and customer segmentation.

Installation and Setup (Power BI Desktop)

- Q21. What is the first step to start using Power BI Desktop?
  You must install Power BI Desktop on a Windows machine, either from the Microsoft Store or by downloading the installer (.exe or MSI) from the official site.
- Q22. How can you install Power BI Desktop from the Microsoft Store?
  Search "Power BI Desktop" in Microsoft Store, click "Get", let it download and install, and then launch it from the Start menu.
- Q23. How can you install Power BI Desktop using a direct download?
  Go to the official Power BI Desktop download page, choose the correct language and 64-bit/32-bit version, download the .exe, run it, accept the license, choose the folder, and complete the setup wizard.
- Q24. Why is the Microsoft Store version often recommended?
  The Store version is recommended because it receives automatic updates along with other Windows Store apps, reducing manual maintenance.
- Q25. What are typical basic steps in the installation wizard?
  Run the installer, select language, accept the license agreement, choose installation location, click Install, and finish the wizard then launch Power BI Desktop.
- Q26. Does Power BI Desktop support multiple languages?
  Yes, the installation package includes all supported languages, and you can select the language during installation or change it later in options.
- Q27. On which operating system is Power BI Desktop primarily supported?
  Power BI Desktop is primarily supported on Windows; other platforms use the Power BI service in a browser instead of desktop.
- Q28. What is a gateway in the context of Power BI (high-level)?
  A gateway is a bridge that securely connects on-premises data sources to Power BI so that reports in the cloud can be refreshed.
- Q29. After installing Power BI Desktop, how do you open it?
  You can open it from the Windows Start menu, desktop shortcut, or directly from the Microsoft Store "Launch" button.
- Q30. Why is it important to install the correct (32-bit or 64-bit) version?
  It must match your OS and Office architecture to ensure compatibility and to handle larger datasets efficiently on 64-bit systems.

Interface and Views

- Q31. What are the three core views in Power BI Desktop?
  The three core views are Report view, Data view, and Model (Relationship) view.
- Q32. Where are the view icons located in Power BI Desktop?
  The icons for Report, Data, and Model views are located vertically on the left side of the Power BI Desktop window.
- Q33. What is the purpose of the Report view?
  Report view is used to design and arrange visuals on report pages, manage filters, and format charts and other visuals.
- Q34. What is the purpose of the Data view?
  Data view lets you see the loaded tables in tabular form so you can inspect columns, verify data, and create calculated columns and simple quick measures.

- Q35. What is the purpose of the Model view?
  Model view shows the tables and their relationships visually, allowing you to define and manage relationships and manage the overall data model.
- Q36. How would you describe the standard workflow using these three views?
  Typically you load and transform data, check tables in Data view, define relationships in Model view, and finally build visuals in Report view.
- Q37. What panes are visible in the Report view?
  Common panes are Fields (datasets and fields), Visualizations (visual types and formatting), Filters (page/report/visual filters), and the central report canvas.
- Q38. What is the report canvas?
  The report canvas is the main area in Report view where you place and arrange visuals to create report pages.
- Q39. What is a visual in Power BI?
  A visual is any graphical representation of data, such as bar charts, line charts, cards, tables, maps, and other custom visuals.
- Q40. Can you edit data directly in Data view?
  You typically cannot edit source data directly in Data view; data modifications are done via Power Query (Transform Data) and then re-loaded.

Basic Power BI Workflow

- Q41. What is the typical end-to-end workflow in Power BI Desktop?
  Connect to data → transform and clean data in Power Query → load data into the model → define relationships and calculations → build visuals and reports → publish to Power BI service.
- Q42. What is the first screen you see after loading data into Power BI Desktop?
  You usually see the Report view with a blank canvas and the loaded fields available in the Fields pane.
- Q43. How do you add a field to a visual?
  Select a visual type and then drag fields from the Fields pane into the appropriate wells such as Axis, Values, Legend, or Tooltip.
- Q44. How do you switch between Report, Data, and Model views?
  Click the corresponding icons on the left side navigation bar in Power BI Desktop.
- Q45. What is meant by “publishing” a report in Power BI?
  Publishing is the process of uploading a .pbix report from Power BI Desktop to the Power BI service so others can view and use it.
- Q46. What file extension does Power BI Desktop use to save reports?
  Power BI Desktop saves reports in files with the .pbix extension.
- Q47. What is a dataset in Power BI?
  A dataset is a collection of imported or connected data that underlies reports and dashboards and is composed of tables, relationships, and calculations.
- Q48. What is meant by “relationships” between tables?
  Relationships define how tables connect via keys (like primary-foreign key), enabling cross-filtering and combined analysis across multiple tables.
- Q49. How is filtering applied in a report?
  Filtering can be applied at visual level, page level, or report level using the Filters pane or

through slicers and visual interactions.

- Q50. What is a slicer?
  A slicer is a special visual used to filter data on a report page by allowing users to select values for a particular field.

Adjusting Initial Settings

- Q51. How do you open the options/settings in Power BI Desktop?
  Go to File → Options and settings → Options to access global and current file settings.
- Q52. How can you change the language of Power BI Desktop?
  In Options, under Regional Settings or General, you can select the display language and locale for Power BI Desktop.
- Q53. How do you change the regional settings (for example, date and number formats)?
  Under Options → Regional Settings, you can choose the model's locale and regional formatting to control date, time, and number formats.
- Q54. What are data load settings in Power BI Desktop?
  Data load settings control how data is imported, such as background data previews, automatic relationship detection, and whether to auto-date/time for time-intelligence.
- Q55. Why might you disable "auto date/time" in Power BI settings?
  Disabling auto date/time can reduce model size and give more control when you want to create your own date table for time intelligence.
- Q56. How can you control automatic relationship detection?
  In Options → Data Load, you can enable or disable automatic creation of relationships when new tables are loaded.
- Q57. Why are regional settings important when working with data?
  They ensure numbers, currency, and dates are interpreted correctly, which is critical when data comes from different countries or formats.
- Q58. How can you adjust default visual interaction and formatting behavior?
  Many default behaviors and themes can be adjusted in the Options dialog and via report themes to match organizational standards.
- Q59. Where do you configure automatic updates and preview features?
  In Options under Preview features and Updates, you can enable experimental features and manage the update behavior of Power BI Desktop.
- Q60. Why is it useful to review settings immediately after installing Power BI?
  Reviewing settings early ensures correct region, language, data load preferences, and governance policies, leading to consistent and reliable reports from the beginning.

Unit II

Front-End vs Back-End

- Q1. What is meant by Power BI front-end and back-end?
  Front-end refers to report view where visuals, dashboards, and user interactions are created, while back-end refers mainly to Power Query Editor and the data model where data is connected, cleaned, and modeled.
- Q2. Which part of Power BI is primarily responsible for data shaping and cleaning?
  Power Query Editor (back-end) is responsible for extracting, transforming, and loading

(ETL) data before it reaches the model.

- Q3. Which part of Power BI is used to design charts and dashboards?
  The Report view in Power BI Desktop (front-end) is used to create visuals, arrange them on pages, and manage filters and slicers.
- Q4. How does Power Query Editor relate to the data model and report?
  Queries in Power Query Editor produce tables that are loaded into the data model; these tables and relationships are then used in Report view for building visuals.
- Q5. Can you change data types in the front-end report view?
  Most data-type changes should be done in Power Query Editor or Model view; Report view mainly consumes already prepared fields for visualization.
- Q6. Why is it recommended to apply heavy data transformations in the back-end instead of the front-end?
  Doing transformations in Power Query reduces model complexity, improves performance, and centralizes logic so multiple visuals share the same clean data.
- Q7. What file component stores back-end queries inside a .pbix file?
  The .pbix file contains M queries (Power Query), the data model, and the report layout together, so back-end definitions are stored alongside front-end visuals.
- Q8. What happens in the report when you refresh queries in Power Query Editor?
  Data is re-fetched and transformed according to the query steps, updating the model; visuals in the report automatically reflect the new data.

Data Connectors and Importing Data

- Q9. What is a data connector in Power BI?
  A data connector defines how Power BI connects to a specific type of data source, handling authentication, connection parameters, and query translation.
- Q10. Name at least five common Power BI data connectors.
  Examples include Excel, Text/CSV, SQL Server, Oracle database, Web, SharePoint, and cloud services like Azure SQL Database and Google Analytics.
- Q11. Where do you start when connecting to data in Power BI Desktop?
  You start from the Home tab by selecting “Get data”, choosing a connector such as Excel, SQL Server, or Web, and then configuring connection details.
- Q12. What is the difference between file connectors and database connectors?
  File connectors read data from static files (Excel, CSV, XML, JSON), whereas database connectors communicate with relational or analytical engines like SQL Server or Oracle.
- Q13. What are Import and DirectQuery modes?
  Import mode loads a copy of the data into the Power BI model, while DirectQuery leaves data in the source and sends queries to it in real time.
- Q14. When would you choose DirectQuery instead of Import mode?
  DirectQuery is used when data is too large to import or when near real-time results are needed from sources like large SQL Server or data warehouses.
- Q15. What are some limitations of DirectQuery compared to Import?
  DirectQuery often has limited DAX functions, slower performance due to live queries, and stricter limits on visual types and aggregations compared to Import.
- Q16. How do credentials affect data connectors?
  Each connector requires appropriate credentials (Windows, database, OAuth, etc.), which

control security and what data Power BI is allowed to access.

- Q17. What is a “data source” versus a “dataset” in Power BI?
  The data source is the external storage like Excel or SQL Server, while a dataset is the imported or connected data model used by Power BI reports and dashboards.
- Q18. Can you connect Power BI to online services like Google Analytics or Salesforce?
  Yes, Power BI provides specific connectors for many SaaS services such as Google Analytics, Salesforce, and others through the Online Services category.
- Q19. What is the role of a gateway when using on-premises data connectors?
  A gateway acts as a secure bridge that allows the Power BI service to refresh or query on-premises data sources like SQL Server.
- Q20. How can you see all supported data sources in Power BI Desktop?
  In Power BI Desktop, click Home → Get Data → More to open the data source dialog that lists all connectors organized by category.

Power Query Editor Concepts

- Q21. What is Power Query Editor?
  Power Query Editor is the interface where you define step-by-step transformations using the M language to clean, reshape, and combine data before loading.
- Q22. How do you open Power Query Editor in Power BI Desktop?
  Click Home → Transform data → Transform data to launch the Power Query Editor window.
- Q23. What pane lists all queries in Power Query Editor?
  The Queries pane on the left lists each query (table) that has been defined.
- Q24. What is the “Applied Steps” pane used for?
  The Applied Steps pane records each transformation step; you can rename, delete, or reorder these steps to adjust the ETL process.
- Q25. Are Power Query transformations destructive to the original source data?
  No, Power Query works in a read-only and step-based way, leaving the original source unchanged.
- Q26. What language is used internally by Power Query?
  Power Query uses the M (Power Query) formula language to represent queries and transformations.
- Q27. What is meant by “query folding” in Power Query?
  Query folding is the process where Power Query translates transformations back to the source system (like SQL) so they run there instead of locally, improving performance.
- Q28. How do you preview data while editing queries?
  The central grid in Power Query Editor shows a preview of the data after each transformation step.
- Q29. What is the purpose of “Close & Apply”?
  “Close & Apply” closes Power Query Editor and applies all query changes, loading or refreshing data into the Power BI model.
- Q30. Can you use parameters in Power Query?
  Yes, parameters allow you to externalize values (such as server names, file paths, or filters) to reuse and control them across multiple queries.

Basic Table Transformations

- Q31. How can you rename a column in Power Query?
  You can double-click the column header, right-click and choose Rename, or use Transform → Rename in the ribbon.
- Q32. Why is it important to give meaningful column names?
  Clear names make reports understandable, reduce confusion for users, and improve maintainability of queries and measures.
- Q33. How do you change a column's data type?
  Click the data-type icon in the column header or use Transform → Data Type and select the appropriate type such as Whole Number, Decimal, Date, or Text.
- Q34. What happens if the wrong data type is set on a column?
  Incorrect types can cause errors, truncation, wrong aggregations, or misinterpreted dates and numbers in visuals.
- Q35. How do you remove unwanted columns?
  Select the columns, right-click and choose "Remove Columns" or use Home → Remove Columns to drop them from the query.
- Q36. How do you filter rows in Power Query?
  Use the filter drop-downs in column headers to select values, ranges, or conditions such as "Text filters", "Number filters", or "Date filters".
- Q37. How do you sort data in Power Query Editor?
  Click on a column header and choose ascending or descending sort options from the drop-down.
- Q38. How can you split a column into multiple columns?
  Use the "Split Column" option (by delimiter, by number of characters, etc.) to separate values such as "City, Country" into two fields.
- Q39. How do you merge two queries (tables) in Power Query?
  Use Home → Merge Queries to join tables based on matching columns, choosing join type like Inner, Left Outer, or Full Outer.
- Q40. How do you append queries in Power Query?
  Use Home → Append Queries to stack tables with similar structure vertically into a single combined table.

Database Connections (SQL Server Example)

- Q41. How do you start a connection from Power BI Desktop to a SQL Server database?
  Go to Home → Get Data → SQL Server, enter the server and optional database name, then choose Import or DirectQuery mode.
- Q42. What information is required to connect to SQL Server?
  At minimum you need the server name, and optionally database name, plus authentication details such as Windows or SQL Server credentials.
- Q43. What is the difference between Import and DirectQuery when connecting to SQL Server?
  Import copies data into the Power BI model for in-memory analytics, while DirectQuery keeps data in SQL Server and queries it live.
- Q44. How do you select which tables to load from a database?
  After connecting, the Navigator window shows available tables and views; you tick the ones you want, and then choose Load or Transform Data.

- Q45. How can you use a custom SQL statement when connecting?
  In the SQL Server connector dialog, under Advanced options, you can paste a SQL query to pull only specific columns or rows.
- Q46. What is a live connection versus DirectQuery?
  A live connection usually refers to connecting to Analysis Services or semantic models where Power BI only sends queries and does not store a separate model, whereas DirectQuery targets relational sources like SQL Server.
- Q47. How does a data gateway support SQL Server connections from the Power BI service?
  The gateway relays encrypted queries and refresh requests from the cloud service to on-premises SQL Server and returns results.
- Q48. What performance considerations apply when connecting Power BI to databases?
  You should filter and aggregate in the source, leverage query folding, and avoid pulling unnecessary columns or rows to improve performance.
- Q49. What security measures should be respected when connecting to databases?
  Use least-privilege accounts, avoid embedding credentials in shared files, and configure row-level security or database permissions as needed.
- Q50. Can Power BI connect to cloud databases like Azure SQL or BigQuery using similar steps?
  Yes, Power BI provides dedicated connectors where you specify server/instance information and credentials, then load or DirectQuery data in a similar way.

Extracting Data from the Web

- Q51. How do you start importing data from a website in Power BI?
  Use Home → Get Data → Web, paste the URL, and let Power BI analyze the page to extract tables or structured data.
- Q52. What kind of web data can Power BI read by default?
  Power BI can detect HTML tables, some lists, and structured content returned by URLs, including certain APIs if they return JSON or XML.
- Q53. How do you select a specific table from a web page?
  After providing the URL, the Navigator window lists detected tables or document structures; you preview and select the desired one, then choose Load or Transform Data.
- Q54. How can you transform JSON data from a web API in Power Query?
  After connecting via Web → URL returning JSON, Power Query shows a record/list structure; you use “To Table”, “Expand” and other transformations to convert JSON fields into columns.
- Q55. What challenges might occur when extracting data from the Web?
  Issues include changing website structure, authentication requirements, pagination, dynamic content loaded by scripts, or rate limits.
- Q56. How can parameters help when calling web APIs?
  You can use parameters to store base URLs, API keys, or page numbers and reference them in query steps to manage and reuse API calls.
- Q57. Is it possible to schedule refresh of web-based data in the Power BI service?
  Yes, if the web source is accessible to the service or through a gateway, you can configure scheduled refresh so web data updates automatically.

- Q58. What should be considered about the reliability of web data sources?
  You should verify data quality, stability of URLs, compliance with terms of use, and ensure refresh failures are monitored.
- Q59. How can you handle multiple pages of web data (pagination)?
  Power Query can construct functions and loops over page numbers or offsets, combining results using “Invoke Custom Function” and append operations.
- Q60. Why is it important to document web queries and transformations?
  Clear documentation helps others understand the origin, refresh behavior, and transformation logic of web-sourced data, which is critical for governance and troubleshooting

Unit III

Data Modeling Fundamentals

- Q1. What is data modeling in Power BI?
  Data modeling is the process of organizing tables, relationships, and calculations to represent business data logically and efficiently for analysis and reporting.
- Q2. Why is good data modeling important in Power BI?
  A well-designed model improves performance, simplifies DAX, and ensures visuals return correct results with predictable filter behavior.
- Q3. What is a semantic model in Power BI?
  A semantic model is the curated layer of tables, relationships, measures, and hierarchies that exposes business-friendly data to report authors.
- Q4. What are the main elements of a Power BI data model?
  The main elements are tables, columns, relationships, measures, calculated columns, and hierarchies.
- Q5. What is the general best-practice layout for models?
  Best practice is a star-schema design with fact tables in the center and dimension tables around them, connected by one-to-many relationships.
- Q6. How does VertiPaq influence modeling choices?
  VertiPaq is Power BI’s columnar storage engine; it compresses data more effectively when columns are well-typed and models use tall fact tables with narrow dimensions.

Database Normalization

- Q7. What is database normalization?
  Normalization organizes data into multiple related tables to reduce redundancy and maintain data integrity, typically following normal forms.
- Q8. Why is highly normalized data not ideal directly for Power BI reporting?
  Highly normalized schemas can lead to many small tables and complex joins that hurt performance and make models harder for users to understand.
- Q9. How is normalization typically handled when building a Power BI model?
  Source systems remain normalized, but Power BI often “denormalizes” them into star schemas by combining lookup tables where appropriate.
- Q10. What is denormalization in the context of Power BI?
  Denormalization means flattening multiple related tables into fewer, wider dimension

tables to simplify relationships and improve query speed.
- Q11. When would you keep a more normalized (snowflake) structure in Power BI?
  Snowflakes may be kept when dimensions share reusable sub-dimensions or when governance rules demand separate tables for shared entities.

Fact and Dimension Tables

- Q12. What is a fact table?
  A fact table stores transactional or numeric event data, such as sales or orders, with foreign keys to related dimensions and numeric measures.
- Q13. What is a dimension table?
  A dimension table stores descriptive attributes such as customer, product, or date that provide context for analyzing facts.
- Q14. How can you quickly identify a fact table in a model?
  Fact tables are usually large, contain many rows, have foreign keys, and mostly numeric aggregatable columns like amounts or quantities.
- Q15. Why is a dedicated Date dimension recommended?
  A Date dimension supports consistent time intelligence, provides calendar attributes, and replaces auto date/time for better control.
- Q16. Can a model have multiple fact tables?
  Yes, multiple fact tables (e.g., Internet Sales and Reseller Sales) can share common dimensions like Date and Product.
- Q17. What is a degenerate dimension?
  A degenerate dimension is an identifier (like invoice number) stored in the fact table but not linked to a separate dimension table.

Relationships and Keys

- Q18. What is a relationship in Power BI?
  A relationship defines how rows in one table relate to rows in another, enabling filter propagation and combined analysis.
- Q19. What is a primary key and where is it used?
  A primary key uniquely identifies each row in a table, usually on the dimension side of a relationship.
- Q20. What is a foreign key?
  A foreign key is a column in one table that references the primary key of another table, typically found in fact tables.
- Q21. What is cardinality in Power BI relationships?
  Cardinality describes the nature of row matching: one-to-one, one-to-many, or many-to-many.
- Q22. What are active and inactive relationships?
  Only one relationship between two tables can be active (used by default in calculations); others remain inactive and can be activated in DAX with functions like USERELATIONSHIP.
- Q23. What are some prerequisites for creating a one-to-many relationship?
  The “one” side column must contain unique values, while the “many” side may contain

repeats; data types must match on both sides.

Managing Relationships

- Q24. How do you create or edit a relationship in Power BI Desktop?
  In Model view, drag a column from one table to another or open Manage Relationships to define cardinality, cross-filter direction, and active status.
- Q25. What is an automatic (detected) relationship?
  Power BI can infer relationships automatically based on column names and data patterns when the “Autodetect relationships” option is enabled.
- Q26. What is a one-to-one relationship?
  Each row in Table A matches at most one row in Table B and vice versa, often used for splitting large dimensions across multiple tables.
- Q27. What is a one-to-many relationship?
  One row in the dimension table corresponds to many rows in the fact table; this is the standard relationship in star schemas.
- Q28. What is a many-to-many relationship?
  Many rows in one table can relate to many rows in another, typically implemented via many-to-many cardinality or bridge tables.
- Q29. How can a bridge table help with many-to-many scenarios?
  A bridge table holds unique combinations of keys and connects to both sides with one-to-many relationships, simplifying filter behavior.
- Q30. Why should circular relationships be avoided?
  Circular references create ambiguity and can cause incorrect totals or model validation errors; models should be acyclic.

Star and Snowflake Schemas

- Q31. What is a star schema?
  A star schema has a central fact table connected directly to multiple dimension tables, forming a star-like layout.
- Q32. What advantages does a star schema provide in Power BI?
  Star schemas simplify relationships, improve query performance, and make DAX and filter behavior easier to reason about.
- Q33. What is a snowflake schema?
  A snowflake schema extends dimensions into further related tables (sub-dimensions), reflecting a more normalized structure.
- Q34. When is a star schema preferred over a snowflake?
  Star schema is preferred for most reporting because it reduces joins, improves performance, and is easier for users to navigate.
- Q35. Can you mix star and snowflake patterns in one model?
  Yes, many real models are “hybrid” with mostly star structures and a few snowflaked dimensions where needed.
- Q36. How does star schema help with DAX time intelligence and aggregation?
  It ensures clear paths from dimension filters to fact tables, allowing DAX measures to compute correctly across attributes like product and date.

Filter Context and Flow

- Q37. What is filter context in Power BI?
  Filter context is the set of filters applied to a calculation from visuals, slicers, page filters, and relationships at evaluation time.
- Q38. How do filters propagate through relationships?
  Filters typically flow from dimension tables on the “one” side to fact tables on the “many” side according to the relationship’s cross-filter direction.
- Q39. How does a slicer on a dimension affect a measure in a visual?
  Selecting values in a slicer filters the related dimension table, which then propagates filters to the connected fact table before the measure is evaluated.
- Q40. How does CALCULATE interact with filter context?
  CALCULATE modifies filter context by adding, replacing, or removing filters before evaluating its expression.
- Q41. What is row context versus filter context?
  Row context refers to per-row evaluation (e.g., in calculated columns), whereas filter context refers to subsets of rows determined by active filters.
- Q42. Why is understanding filter context essential for modeling?
  Without understanding filter flow, relationships and measures can yield unexpected results, especially with multiple fact tables or bi-directional filters.

Bi-Directional Filters and Ambiguity

- Q43. What is cross-filter direction in Power BI?
  Cross-filter direction defines whether a relationship propagates filters from one table to another in a single direction or both directions.
- Q44. What is a bi-directional relationship?
  A bi-directional relationship allows filters to flow in both directions between two tables, often used in many-to-many or role-playing scenarios.
- Q45. What is the risk of using too many bi-directional relationships?
  They can create ambiguous filter paths, causing double-counting, unexpected totals, or model validation issues.
- Q46. How can ambiguity be detected in a model?
  Ambiguity becomes evident when multiple paths exist between tables in Model view or when DAX returns inconsistent results for seemingly simple measures.
- Q47. What is a recommended strategy to reduce ambiguity?
  Use single-direction relationships from dimensions to facts, apply bridge tables for many-to-many, and rely on DAX (e.g., TREATAS) for advanced scenarios.
- Q48. When might bi-directional filtering be appropriate?
  It’s sometimes appropriate for small models with reporting tables like “Regions ↔ Customers” where both sides need to filter each other and ambiguity is controlled.

Data Formats, Categories, and Hierarchies

- Q49. Why are data formats important in the model?
  Correct formats ensure numbers, dates, and currencies display properly and aggregate

correctly in visuals.

- Q50. What is a data category in Power BI?
  Data categories (like City, Web URL, Image URL, Latitude/Longitude) tell Power BI how to interpret a column for visuals such as maps or links.
- Q51. How do you set a column's data category?
  Select the column, then use Column tools → Data category to choose an appropriate category.
- Q52. What is a hierarchy in Power BI?
  A hierarchy is an ordered set of columns (e.g., Year → Quarter → Month) that enables drill-down and drill-up in visuals.
- Q53. How do you create a hierarchy?
  In Model or Data view, right-click a column (such as Year) and choose "Create hierarchy", then add additional levels like Quarter, Month, and Day.
- Q54. Give an example of a geographic hierarchy.
  A common geographic hierarchy is Country → State/Province → City.
- Q55. How do hierarchies affect filter context?
  Drilling down in a hierarchy adds more specific filters, narrowing the filter context from higher to lower levels.
- Q56. Why should you avoid mixing unrelated columns in a hierarchy?
  Mixing unrelated attributes confuses users and leads to illogical drill paths that don't reflect real business relationships.
- Q57. How can incorrect data categories break visuals like maps?
  Wrong categories can cause Power BI to misinterpret fields, leading to missing or mis-placed geographic points.
- Q58. What is a role-playing dimension (e.g., multiple date roles)?
  A role-playing dimension is reused for different roles, such as Order Date and Ship Date, typically implemented as separate copies or via inactive relationships.
- Q59. How do you model role-playing dates while avoiding ambiguity?
  Use a single Date table with multiple relationships and activate the relevant one in measures using USERELATIONSHIP, keeping filters mostly single-direction.
- Q60. Why is documentation of the data model important for teams?
  Documenting tables, relationships, and hierarchies helps others understand logic, maintain models, and troubleshoot DAX and performance issues

Unit IV

DAX Basics and Syntax

- Q1. What is DAX in Power BI?
  DAX (Data Analysis Expressions) is a formula language used in Power BI, Analysis Services, and Power Pivot to define calculations such as measures, calculated columns, and calculated tables.
- Q2. What are typical use cases of DAX?
  DAX is used to create KPIs, ratios, time-intelligence calculations, row-level derived columns, advanced filters, and custom aggregations that go beyond basic sums and averages.

- Q3. What is the basic syntax structure of a DAX formula?
  A DAX formula typically starts with a function name followed by parentheses containing arguments, for example Total Sales = SUM(Sales[SalesAmount]).
- Q4. How is a DAX measure typically defined in Power BI Desktop?
  In Report or Model view you select a table, choose "New measure", then type a formula like Total Sales = SUM(Sales[Amount]) in the formula bar.
- Q5. What is the role of functions and operators in DAX?
  Functions perform predefined operations, while operators (+, -, *, /, &&, ||, =, >, <, etc.) combine or compare values in expressions.
- Q6. What are common categories of DAX functions?
  Categories include aggregation, logical, statistical, text, date/time, filter, relationship, and iterator (X) functions.

Calculated Columns vs Measures

- Q7. What is a calculated column in DAX?
  A calculated column is a new column added to a table where the DAX expression is evaluated for each row and stored in the model at refresh time.
- Q8. What is a measure in DAX?
  A measure is a dynamic calculation evaluated at query time in the context of filters on the report (filter context) and is not stored row-by-row.
- Q9. When would you prefer a calculated column over a measure?
  Use a calculated column when you need a value per row for sorting, grouping, or as part of a relationship key, and the value does not depend on slicers.
- Q10. When is a measure preferred instead of a calculated column?
  Measures are preferred for aggregations (totals, averages, percentages) that must respond dynamically to filters and do not need to be stored per row.
- Q11. How do performance and storage differ between calculated columns and measures?
  Calculated columns consume model storage because values are materialized, whereas measures are computed on the fly and usually have smaller storage impact.
- Q12. How does context differ for calculated columns vs measures?
  Calculated columns use row context (current row), while measures use filter context (subset of rows defined by visuals and filters).

Implicit vs Explicit Measures

- Q13. What is an implicit measure?
  An implicit measure is automatically created by Power BI when a numeric column is dropped into a visual, applying an aggregation such as SUM or AVERAGE without DAX code.
- Q14. What is an explicit measure?
  An explicit measure is defined by the user with DAX, like Total Sales = SUM(Sales[SalesAmount]), and appears as a separate field in the Fields pane.
- Q15. What are limitations of implicit measures?
  Implicit measures cannot include complex logic, are not reusable across visuals, and are limited to basic aggregations.

- Q16. Why are explicit measures recommended for serious projects?
  Explicit measures provide reusability, documentation, control over filter context, and support advanced calculations like YTD, conditional logic, and time intelligence.
- Q17. Can implicit and explicit measures return different results?
  Yes, because explicit measures often modify filter context using functions like CALCULATE, while implicit measures always use the default aggregation on the column.

DAX Syntax, Operators, and Basic Functions

- Q18. What arithmetic operators are available in DAX?
  DAX supports addition (+), subtraction (-), multiplication (*), division (/), and exponentiation (^) for numeric expressions.
- Q19. Name common comparison operators in DAX.
  Common comparison operators are =, <> (not equal), >, <, >=, and <=.
- Q20. What logical operators does DAX use?
  Logical operators include AND (&&), OR (||), and NOT.
- Q21. Give examples of basic aggregation functions in DAX.
  Common aggregations are SUM, AVERAGE, MIN, MAX, COUNT, and DISTINCTCOUNT.
- Q22. How does the COUNT function differ from DISTINCTCOUNT?
  COUNT returns the number of non-blank values, whereas DISTINCTCOUNT returns the number of unique non-blank values.
- Q23. What is the syntax of the IF function in DAX?
  IF(<condition>, <true_result>, <false_result>) evaluates a logical test and returns one of two values.
- Q24. When would you use SWITCH instead of nested IFs?
  SWITCH is preferred when comparing one expression against multiple possible values, making the code more readable than nested IF statements.

Text and Date/Time Functions

- Q25. Give examples of common text functions in DAX.
  Examples include CONCATENATE (or & operator), LEFT, RIGHT, MID, UPPER, LOWER, and FORMAT.
- Q26. How can you combine first name and last name using DAX?
  You can use Full Name = Customers[FirstName] & " " & Customers[LastName] or CONCATENATE.
- Q27. What does the LEN function do in DAX?
  LEN returns the number of characters in a text string.
- Q28. Name some basic date/time functions in DAX.
  Common functions are TODAY, NOW, YEAR, MONTH, DAY, EOMONTH, and DATEDIFF.
- Q29. How would you calculate customer age in years using DAX?
  One approach is Age = DATEDIFF(Customers[BirthDate], TODAY(), YEAR).
- Q30. What is the difference between NOW and TODAY in DAX?
  TODAY returns the current date with no time component, whereas NOW returns the current date and time.

CALCULATE and FILTER

- Q31. What does CALCULATE do in DAX?
  CALCULATE evaluates an expression in a modified filter context, allowing you to add, replace, or remove filters before computing a result.
- Q32. What is the general syntax of CALCULATE?
  CALCULATE(<expression>, <filter1>, <filter2>, ...) where each filter modifies the current context.
- Q33. Give an example of CALCULATE for “Sales for 2024 only”.
  Sales 2024 = CALCULATE([Total Sales], 'Date'[Year] = 2024) restricts the measure to year 2024.
- Q34. What does the FILTER function return?
  FILTER returns a table containing rows from another table that meet the specified condition.
- Q35. How can FILTER be combined with CALCULATE?
  You can wrap a table in FILTER inside CALCULATE to apply complex conditions, such as CALCULATE([Total Sales], FILTER(ALL(Customer), Customer[Country] = "India")).
- Q36. Why is CALCULATE considered one of the most powerful DAX functions?
  Because it is the only function that can change filter context, enabling advanced scenarios like time intelligence, conditional totals, and custom rollups.

RELATED and Relationship-Aware Functions

- Q37. What does the RELATED function do?
  RELATED returns a single related value from another table, following an existing relationship from the current row context.
- Q38. Where is RELATED typically used: measures or calculated columns?
  RELATED is most often used in calculated columns to pull attributes from a dimension into a fact table.
- Q39. What precondition must be met for RELATED to work?
  There must be an active relationship between the current table and the related table with the correct direction.
- Q40. What is the reverse of RELATED that works from the “one” side of a relationship to the “many” side?
  Functions like RELATEDTABLE return table results from the many side in the current row context.
- Q41. Give an example of using RELATED for bringing in product category.
  Category = RELATED('Product'[Category]) on a Sales table copies the product category for each sales row.

Iterator (X) Functions: SUMX, AVERAGEX, etc.

- Q42. What is an iterator function in DAX?
  Iterator functions (ending with X) evaluate an expression row-by-row over a table and then aggregate the results, such as SUMX or AVERAGEX.

- Q43. What is the syntax of SUMX?
  SUMX(<table>, <expression>) iterates over each row of <table>, evaluates <expression> and sums the resulting values.
- Q44. Give a practical example of SUMX for line-amount calculation.
  Total Sales = SUMX(Sales, Sales[Quantity] * Sales[UnitPrice]) multiplies quantity by price per row and then sums across rows.
- Q45. How does AVERAGEX differ from AVERAGE?
  AVERAGE aggregates an existing column, whereas AVERAGEX evaluates an expression row-by-row and then averages the results.
- Q46. When would you choose SUMX over a simple SUM on a calculated column?
  SUMX lets you avoid storing an intermediate calculated column, improving model size and keeping logic centralized in the measure.
- Q47. How do iterator functions interact with filter context?
  Iterators respect the current filter context, iterating only over rows visible under existing filters and any additional filters applied in CALCULATE.

Combining DAX Concepts (Interview-Style)

- Q48. Explain the difference between row context and filter context with an example.
  Row context applies to a single row, such as in a calculated column LineAmount = Quantity * Price, while filter context is the set of filters active on a measure like Total Sales displayed for a specific year and region.
- Q49. How would you create a measure for “Sales Last Year” using DAX?
  A typical pattern is Sales LY = CALCULATE([Total Sales], SAMEPERIODLASTYEAR('Date'[Date])), using CALCULATE with a time-intelligence modifier.
- Q50. How can you calculate “Percentage of Total Sales” using measures?
  Define Total Sales and then Sales % of Total = DIVIDE([Total Sales], CALCULATE([Total Sales], ALL('Product'))).
- Q51. What are common pitfalls beginners face with DAX?
  Common issues include confusing calculated columns and measures, ignoring filter context, overusing implicit measures, and misusing row-by-row iterators.
- Q52. Why is understanding the data model crucial before writing DAX?
  DAX relies on relationships and filter propagation; without a solid model, even correct formulas can yield wrong answers due to ambiguous filter paths.
- Q53. In an interview, how would you justify preferring explicit measures and SUMX patterns?

Unit V

Dashboard Design Framework

- Q1. What is the goal of a Power BI dashboard?
  A dashboard should present the most important information at a glance, enabling quick monitoring and decision-making without scrolling or complex navigation.
- Q2. What principles define an effective dashboard design?
  Key principles are clarity, simplicity, consistency, appropriate use of color, a clear visual

hierarchy, and alignment with specific business questions.

- Q3. Why is understanding the target audience important for dashboard design?
  Audience understanding determines which metrics, level of detail, and interactions are necessary (e.g., executives need high-level KPIs, analysts need more detail and drill-downs).
- Q4. What is meant by “visual hierarchy” in dashboards?
  Visual hierarchy is arranging elements so the most important insights appear first (usually top-left), with supporting trends and details placed lower on the page.
- Q5. Why should scrolling be minimized on dashboards?
  Keeping everything critical on one screen supports at-a-glance monitoring and reduces the chance that key metrics are overlooked.
- Q6. How does consistent use of colors and fonts help?
  Consistency strengthens brand identity, reduces cognitive load, and makes readings across pages feel cohesive and predictable.

Sketching the Layout

- Q7. Why is sketching the layout before building important?
  Sketching helps plan which visuals go where, check alignment with business questions, and avoid clutter before investing time in Power BI.
- Q8. What is a common three-zone layout for a dashboard page?
  Top: key KPIs and summary cards; middle: trend and comparison charts; bottom: detailed tables or drill-down visuals.
- Q9. How should whitespace be used in a layout?
  Adequate whitespace between visuals improves readability, prevents overcrowding, and guides the eye between related elements.
- Q10. What considerations apply when designing multi-page reports?
  Each page should focus on a specific theme or set of questions, with an overview/landing page and clear navigation between related pages.
- Q11. How can alignment and grid systems improve layout quality?
  Using a grid and Power BI’s snap-to-grid features ensures neat alignment and consistent sizing, which makes dashboards look professional and easier to scan.

Report Pages and Objects

- Q12. How do you add or rename a report page in Power BI Desktop?
  Use the page tabs at the bottom: click the plus icon to add a page, right-click a tab to rename, duplicate, or delete it.
- Q13. What types of objects can be added to a report page besides visuals?
  You can add text boxes, shapes, images, buttons, and bookmarks to support explanations, branding, and navigation.
- Q14. How can bookmarks and buttons support storytelling?
  Bookmarks capture a specific view (filters, visibility) and buttons trigger navigation or highlight scenarios, enabling guided storytelling and “page-like” experiences.
- Q15. What is the purpose of a dedicated overview or landing page?
  It shows high-level KPIs and links to more detailed pages, helping users orient themselves

and navigate quickly.

Basic Chart Types (Line, Bar, Donut)

- Q16. When should you use a bar or column chart?
  Use bar/column charts to compare categories or discrete items, especially when you want users to compare lengths easily.
- Q17. When is a line chart more appropriate?
  Line charts are best for trends over time, showing how a metric changes by date, month, or year.
- Q18. What are typical drawbacks of donut charts?
  Donut charts are harder to read accurately, especially with many categories; they should be used sparingly for part-to-whole comparisons.
- Q19. How can donut charts be enhanced or repurposed?
  Donuts can be customized into gauges or KPI visuals by adding measures for whitespace and overlays, giving a more informative single-metric display.
- Q20. Why should 3D chart styles generally be avoided?
  3D effects distort perception of length/area, making comparisons difficult and reducing readability.

Advanced Visuals: KPIs, Gauges, Cards

- Q21. What is a card visual used for?
  Card visuals display a single aggregated value such as total sales, number of customers, or current inventory.
- Q22. How does a KPI visual differ from a simple card?
  KPI visuals combine an actual value, a target, and an indicator of trend or status, often compared to a goal or previous period.
- Q23. When is a gauge chart appropriate?
  Gauges are used to show progress toward a fixed goal (e.g., 70% of target), but should be limited to a few key metrics to avoid space waste.
- Q24. What best practices apply to KPI placement on a dashboard?
  KPIs should be placed at the top of the report, use consistent colors and scales, and focus on 3–5 key metrics rather than many.
- Q25. How can custom visuals enhance KPI communication?
  Custom visuals (like advanced cards or donut gauges) can combine KPIs and segments in a single tile, saving space and improving clarity.

Trend Lines, Forecasts, and Analytics

- Q26. How do you add a trend line to a visual?
  In many Power BI visuals you can use the Analytics pane to add a trend line based on linear regression over the existing data.
- Q27. What is the purpose of a trend line?
  Trend lines help highlight the underlying direction of data (upward, downward, stable) beyond short-term fluctuations.

- Q28. How can forecasting be applied in line charts?
  The Analytics pane can generate forecasts based on historical data, adding projected values and confidence intervals into the future.
- Q29. What cautions apply when using built-in forecasts?
  Forecasts rely on simple time-series models and may not capture seasonality or structural breaks, so they should be validated and clearly labeled as projections.

Table and Matrix Visuals

- Q30. What is the difference between a table and a matrix visual?
  Tables show flat data in columns, while matrices support row and column groups, hierarchies, and pivot-style cross-tab summaries.
- Q31. When are table or matrix visuals most appropriate?
  They are best for detailed data lookups, drill-downs, and financial reports where precise values matter.
- Q32. What is conditional formatting in tables/matrices?
  Conditional formatting applies color scales, data bars, icons, or custom rules to highlight important values within cells.
- Q33. Give an example of using conditional formatting in a matrix.
  You might color cells red for negative profit, amber for low margin, and green for high margin to quickly spot issues.
- Q34. How can Top-N filtering be applied to a matrix?
  You can use the filter pane to set “Top N” on a category field by a measure (e.g., top 10 products by sales), limiting rows to that N.

Map Visuals

- Q35. Which Power BI visuals support geographic data?
  Map, Filled map, and Azure Maps visuals support plotting data by city, country, latitude/longitude, or other geographic fields.
- Q36. What data preparation is needed for map visuals?
  Columns must be correctly categorized (e.g., City, Country/Region, Latitude, Longitude) so Power BI’s geocoding can locate them.
- Q37. When is a filled map preferable to a bubble map?
  Filled maps are useful when comparing values across regions like states or countries, while bubble maps emphasize individual locations or points.
- Q38. What best practices apply to mapping sensitive data?
  Aggregating data, avoiding exact addresses when unnecessary, and respecting privacy or regulatory requirements are key.

Report Slicers and Filtering

- Q39. What is a slicer in Power BI?
  A slicer is a visual used to filter other visuals on a report page based on selection of one or more field values.
- Q40. When should you use slicers instead of the filter pane?
  Use slicers when end users need obvious, interactive controls on the canvas, whereas the

filter pane is better for predefined or less frequently changed filters.

- Q41. What types of slicers are available?
  Power BI provides list, dropdown, numeric range, date range, and hierarchical slicers, among others.
- Q42. How can slicers support navigation?
  Slicers combined with bookmarks can simulate menu systems or allow users to switch views (e.g., by region or scenario).

Top-N Filtering and Focus

- Q43. What is Top-N filtering?
  Top-N filtering limits a visual to the top (or bottom) N categories based on a chosen measure, such as top 10 products by revenue.
- Q44. Why is Top-N filtering useful in dashboards?
  It reduces clutter by focusing attention on the most important contributors, such as top customers or worst-performing regions.
- Q45. How can Top-N be made dynamic for users?
  You can combine a “What-If” parameter with measures and use it in a filter, allowing users to adjust N interactively.
- Q46. What must you specify when configuring a Top-N filter?
  You choose the number N, the measure used for ranking, and whether to show top or bottom items.

Conditional Formatting and Visual Cues

- Q47. What is conditional formatting at the visual level?
  It is applying color or style changes in charts, cards, or tables based on expressions or rules, such as varying bar color by performance.
- Q48. What options are available for table/matrix conditional formatting?
  Options include font color, background color, data bars, icons, web-URL formatting, and field value-driven formatting.
- Q49. How can conditional formatting support KPIs?
  You can color KPI cards or matrix values based on thresholds (e.g., red below target, amber near target, green above target), making status immediately visible.
- Q50. What are best practices for using colors in conditional formatting?
  Use a limited, consistent palette, ensure sufficient contrast, and reserve strong colors like red/green for exceptions or status indicators.
- Q51. How can overuse of conditional formatting harm a report?
  Too many colors, icons, or rules create visual noise, making it harder for users to focus on truly important signals.

Unit VI

Optimize Ribbon and Presets

- Q1. What is the Optimize ribbon in Power BI Desktop?
  The Optimize ribbon is a set of tools that lets authors control visual refresh, apply

optimization presets, and access Performance Analyzer to improve report performance.

- Q2. Which key features are available on the Optimize ribbon?
  Main features include Pause visuals, Refresh visuals, Optimization presets (Query reduction, Interactivity, Customize), Apply all slicers, and a shortcut to Performance Analyzer.
- Q3. What does “Pause visuals” do and when is it useful?
  Pause visuals temporarily stops visuals from refreshing while you edit the report, which speeds up authoring on slow or DirectQuery reports.
- Q4. How is “Refresh visuals” different from a full report refresh?
  Refresh visuals lets you refresh only visuals currently paused or selected, instead of reloading the entire page or dataset.
- Q5. What are Optimization presets?
  Optimization presets are predefined combinations of settings (Query reduction, Interactivity, Customize) that tune visual interactions and filter behavior for performance or responsiveness.
- Q6. Describe the Query reduction preset.
  Query reduction turns off automatic cross-highlighting/cross-filtering and adds Apply buttons for filters/slicers, reducing the number of queries sent—ideal for DirectQuery scenarios.
- Q7. Describe the Interactivity preset.
  Interactivity keeps visuals fully interactive with real-time cross-filtering and slicer effects, which is best for import-mode reports where performance is less constrained.
- Q8. What does the Customize preset allow you to do?
  Customize opens detailed query-reduction options so you can selectively enable or disable features like Apply buttons and instant slicer application.
- Q9. How does “Apply all slicers” help performance?
  Apply all slicers defers slicer changes until users click a single button, avoiding multiple refreshes while they adjust several slicers.
- Q10. In an interview, how would you explain when to enable Query reduction vs Interactivity?
  Use Query reduction for DirectQuery or large models to minimize query load, and Interactivity for small/import models where instant feedback is more important than query minimization.

Performance Analyzer: Concepts and Usage

- Q11. What is Performance Analyzer in Power BI Desktop?
  Performance Analyzer is a diagnostic tool that measures how long each visual takes to refresh and breaks timing into DAX query, visual display, and other components.
- Q12. How do you open Performance Analyzer?
  Go to the Optimize ribbon (or View tab) and select Performance Analyzer to open its pane beside the report canvas.
- Q13. How do you record performance with Performance Analyzer?
  Click Start recording, interact with the report (filters, slicers, navigation), then Stop recording to see timings for each affected visual.

- Q14. What key metrics does Performance Analyzer show for each visual?
  It lists DAX query time, visual display time, and "other" processing time, helping pinpoint whether delays come from queries or rendering.
- Q15. How can you use Performance Analyzer data outside Power BI?
  You can copy or export DAX queries and timings, then analyze them in external tools like DAX Studio for deeper optimization.
- Q16. How does Performance Analyzer help debug slow DAX?
  By highlighting visuals with long query times, it directs you to specific measures or calculations that may need simplification or better model design.
- Q17. What is a systematic approach to improving a slow report using Performance Analyzer?
  Start with Performance Analyzer to find slow visuals, inspect their DAX and data model, then apply optimizations such as simpler measures, fewer visuals, and improved relationships.

General Power BI Performance Best Practices

- Q18. Why is data modeling critical for performance?
  A well-designed model (star schema, proper cardinality, minimized columns) reduces memory usage and speeds up query execution.
- Q19. How does reducing column cardinality help?
  Fewer distinct values in columns improve VertiPaq compression, which usually leads to faster aggregations and smaller models.
- Q20. Why should you prefer measures over calculated columns when possible?
  Measures are evaluated at query time and do not store extra data, keeping models lighter and often faster than equivalent calculated columns.
- Q21. What DAX optimization practices are commonly recommended?
  Use variables, avoid unnecessary row-by-row iterators, reduce nested CALCULATE calls, and reuse base measures rather than repeating logic.
- Q22. How can limiting visuals on a page improve performance?
  Each visual sends queries and renders; reducing the number of visuals decreases total query workload and rendering time.
- Q23. What are some DirectQuery-specific performance tips?
  Use Query reduction, limit visuals, filter data early, avoid complex DAX on large granular tables, and ensure the underlying database is indexed.
- Q24. How does aggregating data improve report speed?
  Using summary tables or aggregation tables reduces the number of rows scanned, leading to faster queries and smaller memory footprint.
- Q25. Why is gateway performance relevant to report speed?
  If an on-premises data gateway is overloaded or poorly configured, refresh and query times increase even if the report and model are optimized.

Practical Topics from the Unit

- Q26. How can thoughtful dashboard layout improve performance?
  Grouping critical visuals on the first page and offloading heavy detail tables to secondary

pages prevents loading unnecessary visuals at initial view.

- Q27. What formatting choices can affect performance?
  Excessive conditional formatting, many high-cardinality legends, and overly detailed tooltips can increase rendering time for visuals.
- Q28. How can DAX choices slow down a report?
  Complex nested iterators over large tables, repeated calculations without variables, and poorly scoped filters can cause long DAX query times.
- Q29. What special care is needed when modeling multiple fact tables?
  You need clear shared dimensions, avoid ambiguous many-to-many paths, and ensure relationships are optimized so filter propagation stays simple and efficient.
- Q30. How does the Optimize ribbon support the development lifecycle?
  During development you can pause visuals to design layouts quickly, use presets to mimic production load, and launch Performance Analyzer to validate performance before publishing

Additional

Q1. Why did you choose Power BI over Tableau or Excel for a project?
Power BI integrates tightly with the Microsoft stack (Excel, Azure, Office 365), which simplifies data access, security, and user adoption in organizations already using these tools. It also offers strong self-service capabilities at a comparatively low license cost, with robust modeling (DAX, Power Query) and easy sharing via the Power BI service, whereas Excel is weaker for governed semantic models and Tableau may be more expensive or less integrated in a Microsoft environment.

Q2. Describe a challenging dashboard you built and how you handled user feedback.
A good answer explains the business problem, not just visuals: for example, consolidating sales, inventory, and finance KPIs into one executive dashboard with conflicting stakeholder expectations. You might describe how you gathered requirements iteratively, used wireframes, prioritized a minimal first release, and then adjusted layout, KPIs, and drill-downs based on feedback, emphasizing communication, version control, and data validation to build trust.

Scenario-based modeling & DAX

Q3. Given messy sales data and slowly changing dimensions, how would you model it and which DAX patterns would you use?
First, clean and standardize sales transactions in Power Query (fix data types, remove duplicates, standardize product and customer IDs), then build a star schema: a central Sales fact table with Date, Product, Customer, and Region dimensions. For slowly changing dimensions (e.g., customer segment changes) you can store historical versions with effective-from/to dates and relate using a surrogate key or use DAX patterns such as USERELATIONSHIP and filter tables to select the correct version for a given date; measures may use CALCULATE with date filters or TREATAS to ensure the right historical attributes apply.

End-to-end project questions

Q4. Explain an end-to-end workflow: requirements → data → model → design → publish → governance.
Begin with requirements gathering: meet stakeholders, define business questions, KPIs, and success criteria, and identify source systems. Next, connect to data and perform cleaning and transformation (Power Query), then design a data model using star schema with well-defined relationships and core measures in DAX. After that, design report pages and dashboards with appropriate visuals, interactions, and performance tuning, then publish to the Power BI service, set up workspaces, and configure refresh schedules; finally, implement governance with access control, row-level security if needed, documentation, and a feedback/iteration process.

Governance and deployment

Q5. How would you manage row-level security, workspace structure, and dataset refresh schedules in production?
For row-level security, define roles (e.g., Region Manager, Country Manager) and DAX filters on dimension tables such as Region, then assign users or security groups to roles and test with “View as role.” Workspace structure typically separates development, test, and production workspaces, or uses “golden” shared datasets in a central workspace consumed by multiple thin report workspaces. Refresh schedules are configured based on data latency needs and source limitations, using incremental refresh for large fact tables, monitoring refresh history and failures, and coordinating refresh times with gateway capacity and downstream usage to avoid performance issues.

HR / Behavioral Questions

- Q1. Tell me about yourself and your experience with Power BI.
  Focus on roles, projects, and technologies: how long you have used Power BI, typical data sources (Excel/SQL/web APIs), and the type of dashboards or domains (finance, sales, operations).
- Q2. Describe your most challenging Power BI project. What made it difficult and how did you solve it?
  Mention data issues (messy data, multiple sources), performance problems, or changing requirements, then explain concrete actions: modeling into star schema, cleaning with Power Query, optimizing DAX, and iterating with stakeholders.
- Q3. Why did you choose Power BI over other BI tools (Tableau, Excel, etc.)?
  Explain integration with Microsoft ecosystem, DAX+Power Query for strong modeling, lower licensing cost, and easy publishing and collaboration through the Power BI service.
- Q4. How do you handle conflicting requirements from stakeholders?
  Say you clarify business goals, prioritize must-have vs good-to-have, create mockups, validate frequently, and document trade-offs so everyone understands the final design.

- Q5. How do you ensure data quality and trust in your reports?
  Talk about validating against source systems, adding data-quality checks in Power Query, using reconciliation totals, and documenting assumptions and refresh times in the report.
- Q6. Describe a time when your dashboard got negative feedback. What did you do?
  Show openness: you listened, asked what decisions the users needed to make, simplified visuals, improved performance or layout, and re-tested with them.
- Q7. How do you keep your Power BI skills up to date?
  Mention official docs, blogs, video courses, practice projects, and experimenting with new features like Optimize ribbon, Field parameters, etc.

Core Technical Questions

- Q8. Explain the difference between Power BI Desktop and Power BI Service.
  Desktop is for building models and reports; the Service is for hosting, sharing, security, refresh, and dashboards.
- Q9. What is a star schema and why is it recommended?
  A central fact table with surrounding dimension tables (one-to-many). It simplifies relationships, improves performance, and makes DAX and filter behavior more predictable.
- Q10. Difference between a calculated column and a measure.
  Calculated columns are computed row-by-row at refresh and stored in the model; measures are calculated at query time based on filter context and are not stored per row.
- Q11. What is the difference between Import, DirectQuery, and Composite models?
  Import loads data into VertiPaq for fast in-memory analysis; DirectQuery leaves data in the source and queries it live; Composite models combine both for flexibility and performance.
- Q12. What is DAX? Give an example.
  DAX (Data Analysis Expressions) is the calculation language for measures and calculated columns, e.g. Total Sales = SUM(Sales[Amount]).
- Q13. Explain filter context and row context in DAX.
  Row context is the current row when evaluating a calculated column or iterator; filter context is the set of filters from visuals, slicers, and CALCULATE that determines which rows are aggregated.
- Q14. What does CALCULATE do in DAX?
  It evaluates an expression in a modified filter context (adding/removing filters), which is the basis for time intelligence and conditional aggregations.
- Q15. When would you use SUMX instead of SUM?
  When you need to sum an expression per row (e.g. Quantity * Price) without storing an intermediate column: SUMX(Sales, Sales[Quantity] * Sales[UnitPrice]).

Data Modeling & Relationships

- Q16. How do you decide what should be a fact table vs a dimension table?
  Facts are transactional/numeric events (sales, orders); dimensions are descriptive attributes (Date, Product, Customer) used for slicing and grouping.

- Q17. What relationship types does Power BI support?
  One-to-many, many-to-one, one-to-one, and many-to-many; relationships can be single-direction or bi-directional, with one active relationship between any table pair.
- Q18. When would you use a many-to-many relationship or a bridge table?
  When categories share multiple items on both sides (e.g., customers and accounts) and a simple one-to-many doesn't work; a bridge table with unique combinations can resolve it.
- Q19. How do you handle Slowly Changing Dimensions (e.g., customer moves to new region)?
  Either model multiple versions of the dimension row with effective dates and join appropriately, or handle at query/DAX level using date filters and relationship patterns (USERELATIONSHIP, surrogate keys).

Power Query & Data Preparation

- Q20. What kinds of transformations can you perform in Power Query?
  Changing data types, splitting/merging columns, removing duplicates, filtering rows, pivot/unpivot, appending/merging tables, and parameterized queries.
- Q21. What is query folding and why is it important?
  Query folding pushes transformations back to the source (like SQL Server) so they run there; it improves performance and reduces data transferred to Power BI.
- Q22. How would you handle missing or inconsistent data?
  Use Power Query to replace nulls, standardize categories, remove or impute invalid records, and create data-quality flags if necessary.

Visuals, UX, and Performance

- Q23. How do you choose appropriate visuals for a requirement?
  Map business questions to visual types: bar/column for category comparison, line for trends, card/KPI for single metrics, matrix for detail; avoid clutter and 3D effects.
- Q24. What is Top-N filtering and how do you implement it?
  Top-N shows only the top or bottom N categories by a measure. In the filter pane, set "Filter type: Top N", specify N and the measure (e.g., Total Sales).
- Q25. How do you implement conditional formatting in tables or matrices?
  In the Formatting pane you choose Conditional formatting (background, font color, data bars, icons) and base rules on a measure or field value.
- Q26. What are some common techniques to improve report performance?
  Star schema modeling, remove unused columns, reduce cardinality, prefer measures over calculated columns, limit visuals per page, use aggregations/incremental refresh, and use Optimize ribbon/Performance Analyzer to find slow visuals.

Service, Security, and Deployment

- Q27. Explain the steps to publish and share a report.
  Publish from Desktop to the Service, place in the correct workspace, configure dataset credentials and refresh, build dashboards if needed, and share via workspaces, apps, or direct share links.

- Q28. What is Row-Level Security (RLS) and how do you implement it?
  RLS restricts which rows a user can see. Create roles with DAX filters on dimension tables (e.g., 'Region'[Region] = USERPRINCIPALNAME() mapping table), assign users or security groups to roles in the Service, and test with “View as”.
- Q29. How would you structure workspaces for a production environment?
  Often use Dev/Test/Prod workspaces; keep one certified “golden dataset” in a central workspace and build thin reports on top; apply proper permissions (Viewer, Contributor, etc.).
- Q30. How do you schedule and monitor data refresh?
  Configure refresh on the dataset (daily/hourly as needed), ensure gateway connections for on-prem sources, monitor refresh history and failure alerts, and use incremental refresh for large tables.