CERT EMPIRE

Microsoft Power BI PL-300 Exam Questions

Total Questions: 420+
Demo Questions: 30
Version: Updated for 2025

Prepared and Verified by Cert Empire – Your Trusted IT Certification Partner

For Access to the full set of Updated Questions – Visit: PL-300 Exam Dumps by Cert Empire

Question: 1

You need to address the data concerns before creating the data model. What should you do in Power Query Editor?

A: Select Column distribution.

B: Select the sales_amount column and apply a number filter.

C: Select Column profile, and then select the sales_amount column.

D: Transform the sales_amount column to replace negative values with 0.

Correct Answer:

C

Explanation:

To address data quality concerns before modeling, the initial and most crucial step is to explore and understand the data's characteristics. In Power Query Editor, the Column profile feature provides the most comprehensive set of diagnostic tools for a specific column. By selecting the salesamount column and using Column profile, you can view detailed column statistics (such as min, max, average, count, errors, empty values) and a value distribution chart. This allows for a thorough investigation to identify issues like outliers, unexpected negative numbers (via the 'min' statistic), or data type inconsistencies, which directly addresses the need to understand "data concerns."

Why Incorrect Options are Wrong:

A: Column distribution is only one component of the Column profile. Choosing Column profile is more comprehensive as it includes distribution plus other vital statistics like min/max.

B: Applying a number filter is a data transformation step. This action should be taken after profiling the data and identifying a specific issue that a filter can resolve.

D: Transforming the column to replace values is a data cleaning action. This is a corrective measure performed only after an issue, such as the presence of negative values, has been identified through data profiling.

References:

Microsoft Learn. (n.d.). Use the data profiling tools. In Power Query documentation. Retrieved from https://learn.microsoft.com/en-us/power-query/data-profiling-tools

This document details the three data profiling tools in Power Query. It explains that Column profile provides the most in-depth view, stating, "This feature gives you a more in-depth look at the data in your column," and lists the "Column statistics" (count, error, empty, distinct, unique, min, max, etc.) it provides, which are essential for identifying the "data concerns" mentioned in the question.

Question: 2

You need to create a calculated column to display the month based on the reporting requirements. Which DAX expression should you use?

A: FORMAT('Date'[date], "MMM YYYY")

B: FORMAT('Date' [date], "M YY")

C: FORMAT('Date'[date_id], "MMM") & "" & FORMAT('Date'[year], "#")

D: FORMAT('Date' [date_id], "MMM YYYY")

Correct Answer:

A

Explanation:

The DAX FORMAT function is used to convert a value to text in a specified format. To create a calculated column that displays the month and year for reporting, you must apply a format string to a column with a date data type. The expression FORMAT('Date'[date], "MMM YYYY") correctly references a date column ('Date'[date]) and uses the format string "MMM YYYY". This string instructs DAX to return the abbreviated month name (e.g., "Jan") and the full four-digit year (e.g., "2023"), resulting in a user-friendly text value like "Jan 2023".

Why Incorrect Options are Wrong:

B: The format string "M YY" would produce a numeric month and a two-digit year (e.g., "1 23"), which is less common and less readable for standard business reports.

C: This expression incorrectly attempts to format a dateid column, which is typically an integer key, not a date data type. It also uses overly complex and unnecessary concatenation.

D: This expression is incorrect because it attempts to apply a date format string to a dateid column. The FORMAT function will not produce the desired month and year from a non-date data type like an integer key.

References:

1. Microsoft Documentation, DAX function reference, FORMAT function: "Converts a value to text according to the specified format... The format string for dates/times can be a predefined string (e.g., "Short Date") or a custom format string (e.g., "YYYY-MM-DD")." The

documentation lists "MMM" for the abbreviated month name and "YYYY" for the four-digit year as valid custom format specifiers.

2. Microsoft Learn, "Create calculated columns in Power BI Desktop," Module: "Add calculated tables and columns to Power BI Desktop models": This module explains the use of DAX to add new columns to a table. Creating a formatted date column, such as a month-year column for slicers or chart axes, is a primary use case for calculated columns. The FORMAT function is the standard tool for this task.

Question: 3

You need to create the required relationship for the executive's visual. What should you do before you can create the relationship?

A: Change the data type of Sales[region_id] to Whole Number.

B: In the Sales table, add a measure for sum(sales_amount).

C: Change the data type of sales[sales_id] to Text.

D: Change the data type of sales [region_id] to Decimal Number.

Correct Answer:

A

Explanation:

To create a relationship between two tables in a Power BI data model, the columns on which the relationship is based must have compatible data types. Foreign key columns, such as Sales[regionid], are used to link to a primary key in a dimension table (e.g., a Regions table). These key columns are almost always represented as integers (Whole Numbers). If Sales[regionid] was imported with an incorrect data type, such as Text or Decimal, it must be converted to Whole Number to match the data type of the primary key in the corresponding dimension table before a valid relationship can be established.

Why Incorrect Options are Wrong:

B: Creating a measure is a DAX calculation for analysis and visualization; it does not affect the underlying data model structure or the ability to create table relationships.

C: salesid is likely the primary key for the Sales table itself, not the foreign key needed to connect to a region dimension table.

D: ID columns represent discrete identifiers and should be Whole Number. Using Decimal Number is incorrect for a key column and would likely cause a data type mismatch.

References:

1. Microsoft Learn. (2023). Create and manage relationships in Power BI Desktop. In the "Troubleshoot relationships" section, it addresses common issues, including data type mismatches. It states, "You can't create a relationship between two columns if they have different data types. To create a relationship, you must change the data type for one of the

columns so they're the same." This directly supports correcting the data type as a prerequisite.

2. Microsoft Learn. (2023). Model relationships in Power BI Desktop. This document outlines the fundamentals of data modeling. While it doesn't explicitly list data type matching as a step-by-step instruction, the entire premise of creating relationships between key columns relies on the values and types being compatible for a join operation, with integer keys being the standard.

Question: 4

What should you create to meet the reporting requirements of the sales department?

A: a measure that uses a formula of SUM (Sales [sales_id])

B: a calculated column that use a formula of COUNTA(sales [sales_id])

C: a measure that uses a formula of COUNTROWS (Sales)

D: a calculated column that uses a formula of SUM (Sales [sales_id])

Correct Answer:

C

Explanation:

The requirement is to calculate the total number of sales for reporting. This is an aggregate calculation that needs to respond dynamically to filters in a report (e.g., show sales by year or by region). A DAX measure is the appropriate object for this type of on-the-fly aggregation. The COUNTROWS(Sales) function correctly counts the number of rows in the 'Sales' table, where each row represents a single sale. This provides an accurate and efficient calculation for the total number of sales transactions.

Why Incorrect Options are Wrong:

A: The SUM function would add the numerical values of the salesid column, which is a meaningless calculation and does not represent the number of sales.

B: A calculated column is evaluated at data refresh for each row and is not suitable for dynamic, report-level aggregations. It would be highly inefficient.

D: This option incorrectly uses both a calculated column for an aggregate measure and the SUM function on an identifier column, making it wrong on two levels.

References:

1. Calculated Columns vs. Measures: Microsoft Learn documentation explains that measures are used for calculations that are aggregated, such as sums, averages, or counts, and are evaluated based on the context of a report visual. Calculated columns, in contrast, compute a value for each row.

Source: Microsoft. (n.d.). Tutorial: Create calculated columns in Power BI Desktop. Microsoft Learn. Retrieved from https://learn.microsoft.com/en-us/power-bi/transform-model/desktop-tutorial-create-calculated-columns (See section: "Understand calculated columns" and "Understand measures").

2. COUNTROWS (DAX) function: The official DAX function reference defines COUNTROWS as the function that counts the number of rows in a specified table. This is the direct method for counting transactions when each transaction is a single row.

Source: Microsoft. (2024). COUNTROWS function (DAX). Microsoft Learn. Retrieved from https://learn.microsoft.com/en-us/dax/countrows-function-dax.

3. SUM (DAX) function: The official DAX function reference defines SUM as the function that adds all the numbers in a column. This confirms it is inappropriate for counting distinct transactions using an ID column.

Source: Microsoft. (2024). SUM function (DAX). Microsoft Learn. Retrieved from https://learn.microsoft.com/en-us/dax/sum-function-dax.

Question: 5

You need to create a relationship between the Weekly_Returns table and the Date table to meet the reporting requirements of the regional managers. What should you do?

A: In the Weekly.Returns table, create a new calculated column named date-id in a format of yyyymmdd and use the calculated column to create a relationship to the Date table.

B: Add the Weekly_Returns data to the Sales table by using related DAX functions.

C: Create a new table based on the Date table where date-id is unique, and then create a manyto-many relationship to Weekly_Return.

Correct Answer:

A

Explanation:

To create a valid relationship between a fact table (WeeklyReturns) and a dimension table (Date), both tables must have a key column with a matching data type and format. The Date dimension table typically uses an integer key in the yyyymmdd format for performance. The WeeklyReturns table likely has a standard datetime column. Therefore, creating a new calculated column in WeeklyReturns that formats its date into the yyyymmdd format is the necessary step to prepare the data for a standard one-to-many relationship. This is a fundamental and best-practice data modeling technique in Power BI.

Why Incorrect Options are Wrong:

B: DAX functions like RELATED are used to retrieve data after a relationship has been created; they cannot be used to establish the relationship itself.

C: A Date dimension table is designed to have a unique key (date-id), so creating a new table is redundant. A many-to-many relationship is less efficient and should be avoided here.

References:

1. Microsoft Learn. (2023). Create and manage relationships in Power BI Desktop. This document outlines the requirements for creating relationships, stating that the columns used in a relationship must have the same data type, or at least compatible data types. Creating a calculated column as described in option A is a standard method to meet this requirement. (See section: "Understand relationships").

2. Microsoft Learn. (2023). Model data in Power BI - Create model relationships. This learning module explains the fundamentals of creating relationships in a star schema, emphasizing the connection between fact tables and dimension tables (like Date) via a common key column. The process in option A aligns directly with this principle. (See unit: "Create model relationships").

3. Microsoft Learn. (2024). FORMAT function (DAX). This official documentation describes the FORMAT function, which is the specific DAX function that would be used to create the date-id calculated column in the yyyymmdd format as proposed in the correct answer. (See section: "Syntax" and "Examples").

Question: 6

You have a project management app that is fully hosted in Microsoft Teams. The app was developed by using Microsoft Power Apps. You need to create a Power BI report that connects to the project management app. Which connector should you select?

A: Microsoft Teams Personal Analytics

B: SQL Server database

C: Dataverse

D: Dataflows

Correct Answer:

C

Explanation:

Power Apps created within the Microsoft Teams environment store their data in Microsoft Dataverse for Teams. Dataverse for Teams is a built-in, low-code data platform that provides relational data storage, rich data types, and enterprise-grade governance directly within Teams. To create a Power BI report that analyzes data from such an app, you must use the Dataverse connector. This connector allows Power BI Desktop to directly access the tables and data stored in the specific Dataverse for Teams environment associated with the team where the app is hosted.

Why Incorrect Options are Wrong:

A: Microsoft Teams Personal Analytics: This connector is for analyzing an individual's personal Teams activity (e.g., meeting and chat habits), not for accessing data from a custom Power App.

B: SQL Server database: While a Power App can connect to an external SQL Server, an app "fully hosted in Microsoft Teams" uses the integrated Dataverse for Teams by default.

D: Dataflows: Dataflows are a data preparation tool used to ingest and transform data. The source connector to get the app's data would still be Dataverse, not the Dataflow itself.

References:

1. Microsoft Learn. (2023). About the Microsoft Dataverse for Teams environment. "Power Apps in Teams uses Dataverse for Teams to store its data." Retrieved from https://learn.microsoft.com/en-us/power-platform/teams/about-teams-environment

2. Microsoft Learn. (2023). Connect to Dataverse for Teams tables in Power BI Desktop. "To connect to Dataverse for Teams tables from Power BI Desktop, you use the Dataverse connector." Retrieved from https://learn.microsoft.com/en-us/power-apps/teams/data-platform-powerbi-connector

3. Microsoft Learn. (2024). Use the Dataverse connector. "The Dataverse connector lets you connect to your tables in Dataverse from Power BI Desktop." Retrieved from https://learn.microsoft.com/en-us/power-bi/connect-data/service-dataverse-connector

Question: 7

For the sales department at your company, you publish a Power BI report that imports data from a Microsoft Excel file located in a Microsoft SharePoint folder. The data model contains several measures. You need to create a Power BI report from the existing data. The solution must minimize development effort. Which type of data source should you use?

A: Power BI dataset

B: a SharePoint folder

C: Power BI dataflows

D: an Excel workbook

Correct Answer:

A

Explanation:

The most efficient way to create a new report from an existing data model is to connect to the published Power BI dataset. When the original report was published to the Power BI service, its underlying data model, including all tables, relationships, and DAX measures, was created as a reusable dataset. By creating a new report with a live connection to this existing dataset, you leverage all the prior development work. This approach directly uses the existing data model and measures, which is the definition of minimizing development effort.

Why Incorrect Options are Wrong:

B: a SharePoint folder: This would require re-implementing all the data import, transformation, data modeling, and measure creation steps from the beginning, maximizing effort.

C: Power BI dataflows: Dataflows are for reusable data preparation (ETL) and contain transformed data tables, but they do not include the data model's relationships or DAX measures.

D: an Excel workbook: Similar to connecting to a SharePoint folder, this would mean starting the entire data modeling and measure creation process from scratch.

References:

1. Microsoft Learn. (2023). Connect to datasets in the Power BI service from Power BI Desktop. This document states, "When you create a Power BI Desktop (.pbix) file with a Live Connection to a Power BI dataset, you can create many different reports from that same, single dataset... This approach provides a great way to create and manage your data model, and then have many different report creators build a variety of unique reports from that same, solid data model." This directly supports using a Power BI dataset to reuse an existing model and minimize effort.

2. Microsoft Learn. (2023). Datasets in the Power BI service. This article clarifies the components of a dataset: "A dataset is a collection of data that you import or connect to. Power BI lets you connect to and import all sorts of datasets and bring all of it together in one place... A dataset can also contain a data model, a collection of tables and their relationships." This confirms the dataset contains the full model needed.

3. Microsoft Learn. (2023). Create and use dataflows in Power BI. This source distinguishes dataflows from datasets: "A dataflow is a collection of tables that are created and managed in workspaces in the Power BI service... You use dataflows to ingest, transform, integrate, and enrich big data." This highlights that dataflows are for data preparation, not for storing the complete semantic model with measures, which is the function of a dataset.

Question: 8

You import two Microsoft Excel tables named Customer and Address into Power Query. Customer contains the following columns: ✑ Customer ID ✑ Customer Name ✑ Phone ✑ Email Address ✑ Address ID Address contains the following columns: ✑ Address ID ✑ Address Line 1 ✑ Address Line 2 ✑ City ✑ State/Region ✑ Country ✑ Postal Code Each Customer ID represents a unique customer in the Customer table. Each Address ID represents a unique address in the Address table. You need to create a query that has one row per customer. Each row must contain City, State/Region, and Country for each customer. What should you do?

A: Merge the Customer and Address tables.

B: Group the Customer and Address tables by the Address ID column.

C: Transpose the Customer and Address tables.

D: Append the Customer and Address tables.

Correct Answer:

A

Explanation:

The goal is to combine columns from the Customer table and the Address table into a single query, with one row per customer. The two tables share a common column, Address ID, which links customers to their respective addresses. In Power Query, the Merge Queries operation is used to join two existing tables together based on matching values in one or more columns. Merging the Customer table with the Address table on the Address ID column will add the address details (City, State/Region, Country) to each corresponding customer row, fulfilling the requirement.

Why Incorrect Options are Wrong:

B: Grouping is an aggregation operation that summarizes rows based on a column's values. It would not combine the detailed address columns for each unique customer.

C: Transposing swaps rows with columns. This operation is used for reshaping a single table, not for combining two different tables based on a common key.

D: Appending stacks tables vertically (adds rows). This is appropriate for tables with the same column structure, not for joining tables with different columns side-by-side.

References:

1. Microsoft Learn. (2023). Merge queries overview. Power Query documentation. "In Power Query, you can merge two or more queries by using a Merge queries operation. The merge operation joins two tables based on matching values in one or more columns."

2. Microsoft Learn. (2023). Append queries. Power Query documentation. "The append operation creates a single table by adding the contents of one or more tables to another, and aggregates the column headers from the tables to create the schema for the new table."

3. Microsoft Learn. (2023). Group or summarize rows. Power Query documentation. "The Group by feature in Power Query enables you to group values in various rows into a single value by grouping the rows according to the values in one or more columns."

Question: 9

You have a PBIX file that imports data from a Microsoft Excel data source stored in a file share on a local network. You are notified that the Excel data source was moved to a new location. You need to update the PBIX file to use the new location. What are three ways to achieve the goal? Each correct answer presents a complete solution.

A: From the Datasets settings of the Power BI service, configure the data source credentials.

B: From the Data source settings in Power BI Desktop, configure the file path.

C: From Current File in Power BI Desktop, configure the Data Load settings.

D: From Power Query Editor, use the formula bar to configure the file path for the applied step.

E: From Advanced Editor in Power Query Editor, configure the file path in the M code.

Correct Answer:

B, D, E

Explanation:

To update the location of an Excel data source within a PBIX file, all modifications must be made in Power BI Desktop. There are three primary methods:

1. Data Source Settings (B): The most straightforward method is using the "Data source settings" dialog in Power BI Desktop. This interface allows you to select the specific data source and use the "Change Source" button to browse to the new file path.

2. Power Query Formula Bar (D): Inside the Power Query Editor, selecting the "Source" step in the "Applied Steps" pane displays the corresponding M code in the formula bar. The file path can be directly edited in this formula.

3. Power Query Advanced Editor (E): The Advanced Editor provides access to the complete M code for a query. You can locate the Source step and manually edit the file path string within the M code.

Why Incorrect Options are Wrong:

A: Dataset settings in the Power BI service are for published datasets, often for configuring gateway connections or cloud credentials, not for editing the source path within the PBIX file itself.

C: The "Data Load" settings under "Current File" options control how data is loaded into the model (e.g., relationship detection), not the location of the external data sources.

References:

1. Microsoft Documentation (for option B): "Manage data source settings in Power BI Desktop". Microsoft Learn. This document explicitly describes the process: "In the Data source settings dialog box, select the data source you want to change, and then select Change Source... A dialog box appears, similar to the data source connection window... you can enter the new location information."

2. Microsoft Documentation (for option D): "Use the Power Query Formula Bar". Microsoft Learn. This page explains, "The formula bar displays the M formula associated with the currently selected step... You can edit the existing formula or add a new one." This confirms the ability to edit the file path in the Source step's formula.

3. Microsoft Documentation (for option E): "Use the Advanced Editor in Power Query". Microsoft Learn. This document states, "You can use the Power Query Advanced Editor to see the code that Power Query Editor is creating with each step... You can also create your own shaping code from scratch." This includes modifying the file path in the M code.

Question: 10

You have a CSV file that contains user complaints. The file contains a column named Logged. Logged contains the date and time each complaint occurred. The data in Logged is in the following format: 2018-12-31 at 08:59. You need to be able to analyze the complaints by the logged date and use a built-in date hierarchy. What should you do?

A: Apply the Parse function from the Data transformations options to the Logged column.

B: Change the data type of the Logged column to Date.

C: Split the Logged column by using at as the delimiter.

D: Create a column by example that starts with 2018-12-31.

Correct Answer:

C

Explanation:

The Logged column is a text (string) data type because its format, YYYY-MM-DD at HH:MM, is not a standard date/time format that Power BI automatically recognizes. To enable the built-in date hierarchy, this text column must be converted into a column with a Date or Date/Time data type. The most direct and robust method to resolve this is to treat "at" as a delimiter. By using the Split Column transformation on the "at" delimiter, you can isolate the date portion (2018-12-31) into a new column. After splitting, you can then change the data type of this new column to Date, which allows Power BI to create the desired date hierarchy.

Why Incorrect Options are Wrong:

A: The Parse function in Power Query is used for processing structured text formats like JSON or XML, not for converting custom date/time strings.

B: Directly changing the data type to Date will likely result in errors because Power BI's engine does not natively recognize "at" as a valid separator in a date/time string.

D: While "Column by Example" could also work by inferring the pattern, Split Column is the more fundamental and explicit transformation for handling a known, consistent delimiter.

References:

1. Microsoft Documentation, "Split a column of text (Power Query)": This document details the procedure for splitting a text column by a delimiter, which is the core action required in

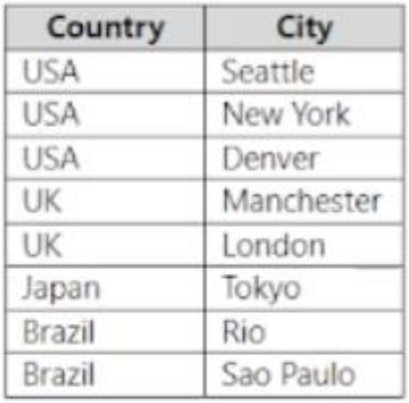
this scenario. It states, "You can split a column with a text data type into two or more columns... by a delimiter." This directly supports using "at" as the delimiter.

2. Microsoft Documentation, "Data types in Power BI Desktop": This source explains the different data types available in Power BI. For the built-in date hierarchy to function, a column must be of the Date or Date/Time data type. The document notes, "Power BI Desktop automatically creates a hierarchy for you when you add a column with a Date or Date/Time data type to a visual." This confirms the prerequisite for the solution.

3. Microsoft Documentation, "Tutorial: Shape and combine data in Power BI Desktop": This tutorial provides practical examples of data transformation, including splitting columns and changing data types, which are the essential steps to solve the problem described in the question. It reinforces that splitting columns is a standard step in cleaning and preparing data for analysis.

Question: 11

You are creating a query to be used as a Country dimension in a star schema. A snapshot of the source data is shown in the following table. You need to create the dimension. The dimension must contain a list of unique countries. Which two actions should you perform? Each correct answer presents part of the solution.



Country	City
USA	Seattle
USA	New York
USA	Denver
UK	Manchester
UK	London
Japan	Tokyo
Brazil	Rio
Brazil	Sao Paulo

A: Delete the Country column.

B: Remove duplicates from the table.

C: Remove duplicates from the City column.

D: Delete the City column.

E: Remove duplicates from the Country column.

Correct Answer:

D, E

Explanation:

The objective is to create a Country dimension table, which, by definition in a star schema, must contain a unique list of countries. The source data contains two columns: Country and City.

1. Delete the City column (D): The City column provides a lower level of granularity than required for a Country dimension. Its presence would cause rows with the same country but different cities (e.g., "United States", "New York"; "United States", "Los Angeles") to be treated as unique, which is incorrect for this dimension. Removing this column is necessary to isolate the country attribute.

2. Remove duplicates from the Country column (E): After removing the City column, the table will consist of a single Country column with repeated values. The final step is to

remove these duplicates to ensure each country is represented only once, satisfying the uniqueness requirement of a dimension table's key.

Why Incorrect Options are Wrong:

A: Delete the Country column. This is incorrect because the Country column is the essential attribute needed for the dimension table.

B: Remove duplicates from the table. This action considers all columns. Since rows like ("United States", "New York") and ("United States", "Los Angeles") are unique, this would not create a unique list of countries.

C: Remove duplicates from the City column. This is irrelevant to the goal of creating a unique list of countries and would not resolve the issue of multiple rows for the same country.

References:

1. Microsoft Learn I Power BI Documentation. Star schema and the importance for Power BI. This document explains that dimension tables contain a key column with unique values that describe business entities. To create a Country dimension, the country name must be the unique key. This supports the need for action E.

Reference: In the section "Dimension tables," it states, "A dimension table has a key column (or columns) that acts as a unique identifier...".

2. Microsoft Learn I Power BI Documentation. Tutorial: Shape and combine data in Power BI Desktop. This tutorial demonstrates common data transformation steps in Power Query, including removing unnecessary columns and reducing rows to create clean tables for modeling.

Reference: The sections on "Shape data" and "Reduce data" cover the actions of removing columns and removing duplicate rows, which are precisely the steps required in this scenario (D and E).

3. Microsoft Learn I Power Query Documentation. Table.Distinct function. The documentation for the function underlying the "Remove Duplicates" feature explains its use for creating a table with unique rows, which can be applied to a single column after other columns have been removed.

Reference: The function Table.Distinct is the M code generated by the "Remove Duplicates" command in the Power Query Editor UI.

Question: 12

You attempt to connect Power BI Desktop to a Cassandra database. From the Get Data connector list, you discover that there is no specific connector for the Cassandra database. You need to select an alternate data connector that will connect to the database. Which type of connector should you choose?

A: Microsoft SQL Server database

B: ODBC

C: OLE DB

D: OData

Correct Answer:

B

Explanation:

When Power BI Desktop does not have a native connector for a specific database, such as Cassandra, the standard approach is to use a generic data access interface. Open Database Connectivity (ODBC) is a widely adopted application programming interface (API) for accessing database management systems. By installing a Cassandra-specific ODBC driver on the machine running Power BI Desktop, the generic ODBC connector in Power BI can be configured with a Data Source Name (DSN) or connection string to connect to and import data from the Cassandra database.

Why Incorrect Options are Wrong:

A: Microsoft SQL Server database: This is a specific connector designed exclusively for connecting to Microsoft SQL Server databases and cannot be used for other database types like Cassandra.

C: OLE DB: While also a generic data access interface, ODBC is more commonly used and recommended for connecting to a wide range of non-native data sources in Power BI.

D: OData: This connector is for consuming data from OData feeds, which are web-based services, not for making a direct connection to a database management system like Cassandra.

References:

Microsoft Power BI Documentation. (2023). ODBC connector for Power BI. Microsoft Learn. This document states, "With the ODBC connector, you can import data from any third-party ODBC driver simply by specifying a Data Source Name (DSN) or a connection string. As an option, you can also specify a SQL statement to execute against the driver." This confirms its role as a generic connector for sources without native support.

Microsoft Power BI Documentation. (2023). Power Query connectors. Microsoft Learn. This page lists the available connectors. The absence of a Cassandra connector and the presence of the generic ODBC connector supports the use of ODBC as the appropriate alternative for unsupported databases that have an ODBC driver.

Question: 13

You have an Azure SQL database that contains sales transactions. The database is updated frequently. You need to generate reports from the data to detect fraudulent transactions. The data must be visible within five minutes of an update. How should you configure the data connection?

A: Add a SQL statement.

B: Set the Command timeout in minutes setting.

C: Set Data Connectivity mode to Import.

D: Set Data Connectivity mode to DirectQuery.

Correct Answer:

D

Explanation:

The requirement is to view data within five minutes of an update from a frequently changing source. The DirectQuery data connectivity mode is designed for this scenario. It does not import and store a copy of the data. Instead, it sends queries directly to the source database (Azure SQL database) each time a user interacts with a report. This ensures that the visuals always display the most current data from the source, meeting the near real-time requirement for fraud detection without waiting for a scheduled refresh.

Why Incorrect Options are Wrong:

A: A SQL statement specifies what data to retrieve but does not determine the connection method or data freshness.

B: The command timeout setting defines how long Power BI waits for a query to complete, not how frequently the data is updated.

C: Import mode copies data into the Power BI model. Viewing updates requires a dataset refresh, which is scheduled and may not meet the five-minute visibility requirement.

References:

1. Microsoft Learn. (2023). Use DirectQuery in Power BI Desktop. In the "When is DirectQuery a good solution?" section, it states, "DirectQuery provides near real-time query results... Use DirectQuery when: Data is changing frequently, and near real-time reporting is needed." This directly supports using DirectQuery for the scenario described.

2. Microsoft Learn. (2023). Dataset modes in the Power BI service. This document contrasts the modes, explaining that "With DirectQuery, no data is imported or copied into Power BI Desktop. When you interact with a visualization, Power BI Desktop queries the underlying data source, so you're always viewing current data." This confirms its suitability for near real-time needs.

3. Microsoft Learn. (2023). Apply query folding. While discussing query folding, the documentation implicitly highlights the nature of DirectQuery. It states, "In DirectQuery mode, all report and ad-hoc queries must be able to be folded." This reinforces that DirectQuery operates by sending live queries to the source.

Question: 14

You have a Microsoft Excel file in a Microsoft OneDrive folder. The file must be imported to a Power BI dataset. You need to ensure that the dataset can be refreshed in powerbi.com. Which two connectors can you use to connect to the file? Each correct answer presents a complete solution.

A: Excel Workbook

B: Text/CSV

C: Folder

D: SharePoint folder

E: Web

Correct Answer:

D, E

Explanation:

To connect to an Excel file stored in OneDrive for Business and ensure it is refreshable in the Power BI service, a cloud-aware connector must be used.

1. Web Connector (E): This is the most direct method for a single file. You obtain the URL for the Excel file from OneDrive and use the "Get Data" > "Web" connector in Power BI Desktop. This creates a direct link to the cloud source that the Power BI service can refresh.

2. SharePoint folder Connector (D): OneDrive for Business is built on SharePoint technology. You can use the "SharePoint folder" connector with the root URL of your OneDrive site. This connector lists all files and folders, from which you can then select and load the specific Excel file. This method is also fully refreshable in the Power BI service.

Why Incorrect Options are Wrong:

A: Excel Workbook: This connector is primarily designed for local files. While it may work with a locally synced OneDrive folder, it is not the recommended or most direct method for a cloud source.

B: Text/CSV: This connector is used for importing data from .txt or .csv files, not from Excel (.xlsx) workbooks.

C: Folder: This connector is used to combine multiple files from a local or network file system folder, not for connecting to a cloud repository like OneDrive for Business.

References:

1. Microsoft Documentation, Power BI Docs, "Use OneDrive for Business links in Power BI Desktop": This document explicitly states, "In Power BI Desktop, on the Home ribbon, select Get data > Web... Paste the link into the dialog box". It also mentions, "You can also use the SharePoint folder connector to connect to the folder and get to the file." This supports both the Web and SharePoint folder connectors.

2. Microsoft Documentation, Power BI Docs, "Get data from SharePoint folder": This page clarifies that the SharePoint folder connector can be used for files in SharePoint Online, which technologically includes OneDrive for Business. It states, "The SharePoint folder connector... allows you to import data from multiple files from a SharePoint folder."

3. Microsoft Documentation, Power BI Docs, "Refresh a dataset created from a Power BI Desktop file on OneDrive or SharePoint Online": This document explains the refresh mechanism, confirming that when Power BI connects to files on OneDrive or SharePoint Online, "any changes to the files are automatically reflected in Power BI, usually within about an hour." This is enabled by using the correct cloud-aware connectors like Web or SharePoint folder.

Question: 15

You have a Microsoft SharePoint Online site that contains several document libraries. One of the document libraries contains manufacturing reports saved as Microsoft Excel files. All the manufacturing reports have the same data structure. You need to use Power BI Desktop to load only the manufacturing reports to a table for analysis. What should you do?

A: Get data from a SharePoint folder and enter the site URL Select Transform, then filter by the folder path to the manufacturing reports library.

B: Get data from a SharePoint list and enter the site URL. Select Combine & Transform, then filter by the folder path to the manufacturing reports library.

C: Get data from a SharePoint folder, enter the site URL, and then select Combine & Load.

D: Get data from a SharePoint list, enter the site URL, and then select Combine & Load.

Correct Answer:

A

Explanation:

The correct method to import multiple files with the same structure from a specific SharePoint document library is to use the "SharePoint folder" connector. After providing the site URL, Power BI lists all files from the entire site. To isolate only the manufacturing reports, you must select "Transform Data" to open the Power Query Editor. Inside the editor, you filter the Folder Path column to select the specific document library containing the reports. After filtering, you can then use the "Combine Files" feature to merge them into a single table.

Why Incorrect Options are Wrong:

B: The "SharePoint list" connector is used for importing data from SharePoint lists, not for accessing and combining the content of files stored within a document library.

C: Selecting "Combine & Load" immediately would attempt to combine all files from the entire SharePoint site, not just the required manufacturing reports from the specific library, failing the requirement.

D: This option uses the incorrect connector ("SharePoint list") and skips the essential filtering step, making it incorrect on two counts for this scenario.

References:

1. Microsoft Documentation - Power Query SharePoint folder connector: "After you enter the site URL, you're taken to a navigator view that shows all the lists of files in your SharePoint site. ... It's best to select the Transform Data button and filter the data in the Power Query Editor. ... To filter for just the files in that subfolder, you can use the Folder Path column." This document explicitly outlines the process of connecting, transforming (filtering), and then combining.

Source: Microsoft Docs, Power Query SharePoint folder connector, "Combine files" section.

2. Microsoft Documentation - Combine files in Power BI Desktop: "The combine files experience starts by going to Get Data, selecting File > Folder, and then selecting Connect. ... After you select the folder you want to use, the file information about all the files in that folder is shown in a dialog window. Select Transform Data to open Power Query." This general guidance for combining files from a folder (which applies to SharePoint folders as well) emphasizes transforming (filtering) before combining.

Source: Microsoft Docs, Combine files in Power BI Desktop, "Combine files experience" section.

Question: 16

You use Power Query to import two tables named Order Header and Order Details from an Azure SQL database. The Order Header table relates to the Order Details table by using a column named Order ID in each table. You need to combine the tables into a single query that contains the unique columns of each table. What should you select in Power Query Editor?

A: Merge queries

B: Combine files

C: Append queries

Correct Answer:

A

Explanation:

The Merge queries operation in Power Query is used to join two or more tables based on matching values in one or more common columns. In this scenario, the Order Header and Order Details tables share a common Order ID column. Merging these two tables on the Order ID column will create a single, wider query that includes columns from both original tables, fulfilling the requirement to combine them into a single query containing the unique columns of each. This operation is analogous to a JOIN in SQL.

Why Incorrect Options are Wrong:

B: Combine files: This operation is used to consolidate data from multiple files (e.g., CSVs, Excel workbooks) that have the same structure into a single table, not for joining database tables.

C: Append queries: This operation stacks rows from two or more tables on top of each other to create a single, longer table. It does not combine columns based on a key.

References:

Microsoft Power BI Documentation. (2023). Merge queries overview. Microsoft Learn. Retrieved from https://learn.microsoft.com/en-us/power-query/merge-queries-overview. This document states, "In Power Query, you can merge two or more queries into a single query... Merging queries creates a new query from two queries in a join operation."

Microsoft Power BI Documentation. (2023). Append queries. Microsoft Learn. Retrieved from https://learn.microsoft.com/en-us/power-query/append-queries. This source clarifies, "The append operation creates a single table by adding the contents of one or more tables to another."

Microsoft Power BI Documentation. (2023). Combine files overview. Microsoft Learn. Retrieved from https://learn.microsoft.com/en-us/power-query/combine-files. This document explains that combining files is for when "you have multiple files that have the same schema."

Question: 17

From Power Query Editor, you attempt to execute a query and receive the following error message. Datasource.Error: Could not find file. What are two possible causes of the error? Each correct answer presents a complete solution.

A: You do not have permissions to the file.

B: An incorrect privacy level was used for the data source.

C: The file is locked.

D: The referenced file was moved to a new location.

Correct Answer:

A, D

Explanation:

The error message Datasource.Error: Could not find file indicates that the Power Query engine was unable to locate the specified file at the given path. This is a file system-level issue. The two most common causes are:

1. The path to the file is no longer valid because the file was moved, renamed, or deleted.

2. The account running the query (either the user's account in Power BI Desktop or the service account for an on-premises data gateway) lacks the necessary file system permissions to access the folder path or the file itself. In many operating systems, a lack of permissions on a folder in the path will result in a "not found" error rather than an "access denied" error.

Why Incorrect Options are Wrong:

B: An incorrect privacy level would generate a Firewall.Error related to combining data sources, not a Datasource.Error about a missing file.

C: A locked file typically generates a different error, such as a sharing violation or a message stating the file is "in use by another process."

References:

1. Microsoft Learn. (2023). Troubleshooting refresh scenarios. In the section "Refresh fails because of data source errors," the documentation addresses the "We couldn't find the file"

error. It explicitly states, "This error is a result of the file being moved or the file permissions have changed." This directly supports options A and D.

2. Microsoft Learn. (2023). Manage an on-premises data gateway. In the section "Account," it details the permissions required for the gateway service account. It notes that the account must have read permissions on the file and its parent folders. A failure to meet this requirement would prevent the service from locating the file, leading to a "Could not find file" error during a scheduled refresh.

Question: 18

A business intelligence (BI) developer creates a dataflow in Power BI that uses DirectQuery to access tables from an on-premises Microsoft SQL server. The Enhanced Dataflows Compute Engine is turned on for the dataflow. You need to use the dataflow in a report. The solution must meet the following requirements: ✑ Minimize online processing operations. ✑ Minimize calculation times and render times for visuals. ✑ Include data from the current year, up to and including the previous day. What should you do?

A: Create a dataflows connection that has DirectQuery mode selected.

B: Create a dataflows connection that has DirectQuery mode selected and configure a gateway connection for the dataset.

C: Create a dataflows connection that has Import mode selected and schedule a daily refresh.

D: Create a dataflows connection that has Import mode selected and create a Microsoft Power Automate solution to refresh the data hourly.

Correct Answer:

C

Explanation:

To meet the performance requirements of minimizing online processing and reducing calculation/render times, Import mode is the superior choice. Import mode caches the data within the Power BI dataset, allowing the in-memory VertiPaq engine to handle all queries and calculations rapidly, without needing to query the upstream dataflow or the on-premises SQL server for every visual interaction.

The data freshness requirement is "up to and including the previous day," which is perfectly met by a scheduled daily refresh. This ensures the imported data is updated once per day, satisfying the requirement without the unnecessary overhead of more frequent refreshes.

Why Incorrect Options are Wrong:

A: DirectQuery mode would increase online processing and visual render times, as each interaction would send a query to the dataflow, directly contradicting the performance requirements.

B: Adding a gateway to the dataset is irrelevant for a dataflow connection and does not change the fact that DirectQuery mode fails to meet the performance requirements.

D: An hourly refresh using Power Automate is excessive. The requirement is only for data up to the previous day, making a simpler scheduled daily refresh more efficient and appropriate.

References:

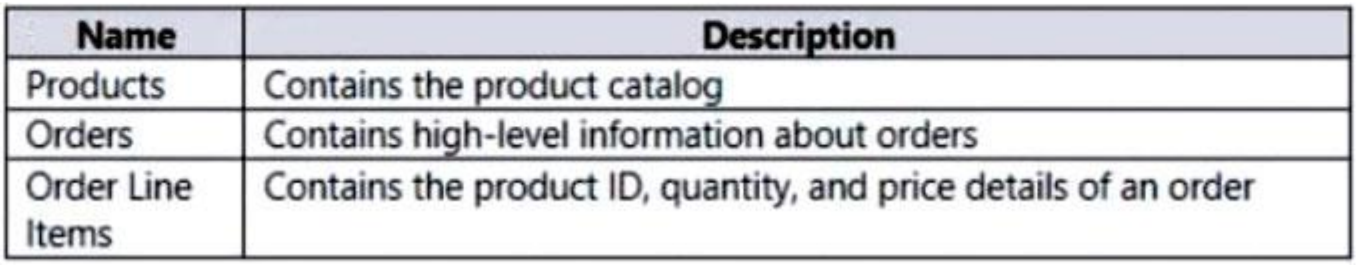
1. Microsoft Power BI Documentation, "Use DirectQuery in Power BI Desktop": This document contrasts DirectQuery and Import modes. It states, "When you use Import mode, Power BI Desktop stores a copy of the data. This copy of the data is used to create your report visuals... Because you're interacting with a local copy of the data, interactions are very fast." This supports using Import for performance. (Reference section: "Import connections versus DirectQuery connections").

2. Microsoft Power BI Documentation, "Configure scheduled refresh": This document details how to set up scheduled refreshes for datasets. It explains, "Configuring scheduled refresh for your datasets makes sure your data is not stale." A daily refresh aligns with the requirement for data "up to and including the previous day." (Reference section: "Configure scheduled refresh").

3. Microsoft Power BI Documentation, "Connect to dataflows in Power BI Desktop": This document explains the options for connecting to a dataflow. It confirms that both Import and DirectQuery are possible connection modes from a dataset to a dataflow (when the enhanced compute engine is on), leaving the choice dependent on the specific scenario's requirements for performance versus data recency. (Reference section: "Dataflow connectors in Power BI Desktop").

Question: 19

You are building a Power BI report that uses data from an Azure SQL database named erp1. You import the following tables.



Name	Description
Products	Contains the product catalog
Orders	Contains high-level information about orders
Order Line Items	Contains the product ID, quantity, and price details of an order

You need to perform the following analyses: ✑ Orders sold over time that include a measure of the total order value Orders by attributes of products sold The solution must minimize update times when interacting with visuals in the report. What should you do first?

A: From Power Query, merge the Order Line Items query and the Products query.

B: Create a calculated column that adds a list of product categories to the Orders table by using a DAX function.

C: Calculate the count of orders per product by using a DAX function.

D: From Power Query, merge the Orders query and the Order Line Items query.

Correct Answer:

D

Explanation:

The goal is to create a data model that supports the required analyses while ensuring high performance for report interactions. The best practice for performance in Power BI is to build a star schema, which consists of a central fact table and surrounding dimension tables.

In this scenario, Order Line Items represents the lowest level of detail (the grain) and should be the basis for the fact table. The Orders table contains header information (like OrderDate) that applies to each line item. By merging the Orders query into the Order Line Items query in Power Query, you denormalize the OrderDate into the fact table. This creates a single, comprehensive fact table that can be directly related to the Products dimension table, forming an efficient star schema. This structure minimizes joins at query time, directly addressing the performance requirement.

Why Incorrect Options are Wrong:

A: Merging Products into Order Line Items is a valid denormalization step, but it leaves you with two transactional tables (Orders and the new merged table), which is less optimal than a single fact table.

B: Using DAX for this type of data shaping is inefficient. Calculated columns are less compressed and evaluated after the data is loaded, which is contrary to the best practice of performing transformations in Power Query.

C: Calculating a measure is an analysis step performed after the data model is properly structured. It is not the first step in preparing the data model itself.

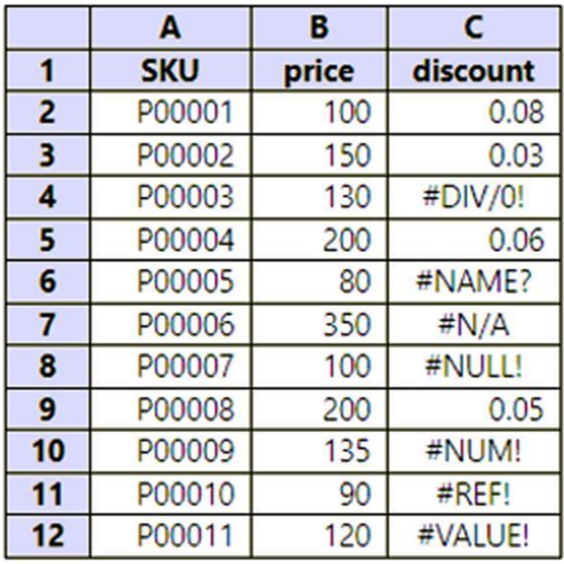
References:

1. Microsoft Power BI Documentation, "Understand star schema and the importance for Power BI": This document explains that a star schema is the most effective data model structure for Power BI. It emphasizes centralizing numeric measures in a fact table and descriptive attributes in dimension tables. Merging the order header (Orders) into the order details (Order Line Items) is a standard technique to create a single, robust fact table. (See the section "Data modeling best practices").

2. Microsoft Power BI Documentation, "Power Query M formula language": The documentation on Power Query transformations, such as Table.NestedJoin (the function behind the Merge Queries UI), establishes it as the primary tool for ETL and data shaping operations before the data is loaded into the model. This is more performant for the end-user than using DAX for structural changes.

Question: 20

You have data in a Microsoft Excel worksheet as shown in the following table.



	A	B	C
1	SKU	price	discount
2	P00001	100	0.08
3	P00002	150	0.03
4	P00003	130	#DIV/0!
5	P00004	200	0.06
6	P00005	80	#NAME?
7	P00006	350	#N/A
8	P00007	100	#NULL!
9	P00008	200	0.05
10	P00009	135	#NUM!
11	P00010	90	#REF!
12	P00011	120	#VALUE!

You need to use Power Query to clean and transform the dataset. The solution must meet the following requirements: • If the discount column returns an error, a discount of 0.05 must be used. • All the rows of data must be maintained. • Administrative effort must be minimized. What should you do in Power Query Editor?

A: Select Replace Errors.

B: Edit the query in the Query Errors group.

C: Select Remove Errors.

D: Select Keep Errors.

Correct Answer:

A

Explanation:

The Replace Errors feature in the Power Query Editor is the most direct and appropriate solution. This function allows you to specify a value (in this case, 0.05) to substitute for any errors found within a selected column. This action meets all requirements: it replaces the error with the specified value, maintains all original rows of data by only transforming the erroneous cells, and is a single, straightforward command, thus minimizing administrative effort.

Why Incorrect Options are Wrong:

B: "Query Errors group" is not a specific feature or command in the Power Query Editor ribbon. The correct action, Replace Errors, is a more precise and actionable choice.

C: Remove Errors would delete the entire row where an error is found in the discount column, which violates the requirement that all rows of data must be maintained.

D: Keep Errors would remove all rows that do not contain an error, which is the opposite of the desired outcome and violates the requirement to maintain all data.

References:

Microsoft Power Query Documentation, "Handle errors (Power Query)": This official document details the methods for managing data errors. The "Replace errors" section explicitly describes the process of right-clicking a column and selecting Replace Errors to substitute error values with a specified replacement value, while keeping all rows. It also explains that Remove Errors and Keep Errors are row-level operations that filter the dataset.

Reference: https://learn.microsoft.com/en-us/power-query/handle-errors (Refer to the section "Replace errors").

Question: 21

You plan to use Power BI Desktop to create a bug tracking dashboard that will pull data from Analytics in Azure DevOps. From Power BI Desktop, you need to configure a data connector to authenticate to Azure DevOps. The solution must meet the following requirements: • Use Analytics views. • Filter data from the cloud. Which connector should you use?

A: OData queries

B: Azure DevOps (Boards only)

C: Azure DevOps Server (Boards only)

D: OData Feed

Correct Answer:

B

Explanation:

The Azure DevOps (Boards only) connector is the purpose-built and recommended method for connecting Power BI Desktop to Azure DevOps Services. This connector is specifically designed to work with Analytics views, which are a prerequisite for this scenario. Using Analytics views allows you to define and apply filters on the server side (in the cloud) before the data is loaded into Power BI. This directly meets the requirement to "Filter data from the cloud" and simplifies the connection and authentication process compared to more generic methods.

Why Incorrect Options are Wrong:

A: OData queries: This describes the query language used to interact with the Analytics service, not the specific Power BI connector itself. The connector executes these queries.

C: Azure DevOps Server (Boards only): This connector is for the on-premises version of Azure DevOps, not the cloud-based service implied by the question.

D: OData Feed: While this generic connector can connect to Azure DevOps Analytics, the dedicated Azure DevOps (Boards only) connector is the recommended, more specific, and user-friendly option for using Analytics views.

References:

1. Microsoft Learn. (2023). Connect with the Power BI Data Connector. In "Azure DevOps documentation". This document explicitly states, "The Power BI Data Connector uses Analytics views... Analytics views provide a simplified way to specify the filter criteria for a Power BI report based on Analytics data." It also shows that the connector to select in Power BI is named "Azure DevOps (Boards only)".

2. Microsoft Learn. (2023). About Power BI integration. In "Azure DevOps documentation". This page outlines the different methods for connecting Power BI to Azure DevOps, highlighting the Data Connector as the primary method. It states, "The simplest way to get data from Analytics into Power BI is to use the Power BI Data Connector."

3. Microsoft Learn. (2023). Create an Analytics view in Azure DevOps. In "Azure DevOps documentation". This guide details how to create Analytics views, which are used to "define a filtered set of data and share it with Power BI." This confirms that Analytics views are the mechanism for server-side filtering.

Question: 22

You have a data source that contains a column. The column contains case sensitive data. You have a Power BI semantic model in DirectQuery mode. You connect to the model and discover that it contains undefined values and errors. You need to resolve the issue. Solution: You implicitly convert the values into the required type. Does this meet the goal?

A: Yes

B: No

Correct Answer:

B

Explanation:

The problem described is due to a case-sensitivity mismatch between the data source and the Power BI engine, which is case-insensitive by default. This can cause relationship and join failures in DirectQuery mode, leading to errors and undefined values. The proposed solution, "implicitly convert the values into the required type," addresses data type issues (e.g., converting text to a number). It does not resolve problems related to the casing of text data. The correct approach would involve explicitly standardizing the case of the data, for example, by using functions like UPPER or LOWER in the source query, which is a different action from type conversion.

Why Incorrect Options are Wrong:

A: Yes: This is incorrect because the proposed solution (implicit type conversion) is irrelevant to the root problem (case-sensitivity). Changing a column's data type will not standardize the case of its text values.

References:

1. Microsoft Learn. (2023). DirectQuery model guidance in Power BI Desktop.

Section: Data type handling

Content: This document notes that "The Power BI engine is case-insensitive, while many data sources are case-sensitive." It recommends resolving such issues in the source query, for instance, by applying UPPER or LOWER functions. This highlights that the problem is about casing, not data types.

2. Microsoft Learn. (2023). Data types in Power BI Desktop.

Section: Implicit and explicit data type conversion

Content: This documentation defines implicit conversion as an automatic process where Power BI converts data types when loading data or in DAX expressions (e.g., converting a text "10" to a number 10 for a calculation). This confirms that type conversion is unrelated to resolving case-sensitivity issues in text strings.

Question: 23

You have a data source that contains a column. The column contains case sensitive data. You have a Power BI semantic model in DirectQuery mode. You connect to the model and discover that it contains undefined values and errors. You need to resolve the issue. Solution: You change the semantic model mode. Does this meet the goal?

A: Yes

B: No

Correct Answer:

A

Explanation:

The issue stems from the interaction between the case-sensitive data source and Power BI's DirectQuery mode. In DirectQuery, the case sensitivity is determined by the source database. This can lead to relationship evaluation failures and errors if, for example, a fact table contains 'ID1' and a dimension table contains 'id1'.

By changing the semantic model mode to Import, the data is loaded into the Power BI VertiPaq engine. The VertiPaq engine is case-insensitive by default. This change forces all data to be treated in a case-insensitive manner, thereby resolving the mismatches that were causing the undefined values and errors.

Why Incorrect Options are Wrong:

B: No: This is incorrect because changing the model to Import mode is a direct and effective solution for resolving case-sensitivity conflicts within the Power BI model's relationships and data evaluation.

References:

1. Microsoft Learn. (2023). Case sensitivity in Power BI. "In Power BI, case-sensitivity depends on where your data is being stored. [...] The Power BI engine is case-insensitive. This means that if you import data into Power BI, it will treat "Apple" and "apple" as the same value. [...] If you're using DirectQuery, case-sensitivity will be determined by the source database." This documentation directly confirms that switching from DirectQuery (source-dependent sensitivity) to Import (case-insensitive engine) resolves such issues.

2. Microsoft Learn. (2024). Model relationships in Power BI Desktop. "When you use DirectQuery, Power BI doesn't import data. Instead, each time a visual needs data, Power

BI queries the underlying data source." This highlights that the source's behavior, including its collation and case-sensitivity rules, dictates the results in DirectQuery mode. Changing the mode alters this fundamental behavior.

Question: 24

You have a data source that contains a column. The column contains case sensitive data. You have a Power BI semantic model in DirectQuery mode. You connect to the model and discover that it contains undefined values and errors. You need to resolve the issue. Solution: You normalize casing in the source query or Power Query Editor. Does this meet the goal?

A: Yes

B: No

Correct Answer:

A

Explanation:

The issue described, "undefined values and errors," in a DirectQuery model connected to a case-sensitive data source is a classic symptom of data integrity problems caused by case mismatches. In DirectQuery, Power BI sends queries to the source, and the source's case-sensitivity rules apply. If relationships are built on columns with inconsistent casing (e.g., 'USA' vs 'usa'), the joins can fail, leading to errors.

Normalizing the casing—by applying functions like UPPER or LOWER either in the native source query or through a foldable Power Query transformation—ensures that these values are treated consistently. This transformation is pushed down to the data source for execution, which is compatible with DirectQuery mode and effectively resolves the root cause of the problem.

Why Incorrect Options are Wrong:

B: This is incorrect. Normalizing case is a direct and effective solution for resolving data integrity and relationship issues caused by case-sensitivity in a DirectQuery source.

References:

1. Microsoft Learn. (2023). DirectQuery model guidance in Power BI Desktop. In the "Data modeling" section, it is noted that while the Power BI engine can be case-insensitive, DirectQuery sources can be case-sensitive. This mismatch necessitates handling at the source. The document states, "It's important to ensure there are no duplicate values in dimension-type table columns... For DirectQuery models, it's also important they don't have

text columns with only a difference in case." This implies that case normalization is a required step to ensure model integrity.

2. Microsoft Learn. (2023). Use DirectQuery in Power BI Desktop. Under the "Implications of using DirectQuery" section, it discusses limitations on data transformations. However, many simple transformations, including case changes, are supported through query folding. The article states, "The transformations that you apply in Power Query Editor are applied to the underlying source," which confirms that normalizing case in Power Query Editor is a valid approach for DirectQuery.

3. Microsoft Learn. (2023). Power Query query folding. This document details which Power Query transformations can be translated into the native language of the data source. Under "Supported operations," transformations on text columns such as Text.Lower, Text.Upper, and Text.Proper are listed as generally foldable, meaning they can be executed by the source database in a DirectQuery scenario.

Question: 25

You have a data source that contains a column. The column contains case sensitive data. You have a Power BI semantic model in DirectQuery mode. You connect to the model and discover that it contains undefined values and errors. You need to resolve the issue. Solution: You add an index key and normalize casing in the data source. Does this meet the goal?

A: Yes

B: No

Correct Answer:

A

Explanation:

In DirectQuery mode, Power BI sends queries directly to the source database, which inherits the source's properties, including case sensitivity. If a relationship is built on a column with case-sensitive data (e.g., 'Apple' and 'apple'), the source database treats these as distinct values. This creates ambiguity for the case-insensitive Power BI engine, leading to non-deterministic relationship behavior, errors, and undefined values.

Normalizing the casing in the data source (e.g., converting all text to uppercase) ensures that values in the key column are unique and consistent, which resolves the root cause of the errors. Adding an index is a critical performance optimization for DirectQuery, ensuring that the joins required for the relationship are efficient. Therefore, this combined solution effectively resolves the issue.

Why Incorrect Options are Wrong:

B: This option is incorrect because the proposed solution directly addresses both the data integrity problem (casing) and the performance aspect (indexing) that are critical for a functional DirectQuery model with case-sensitive data sources.

References:

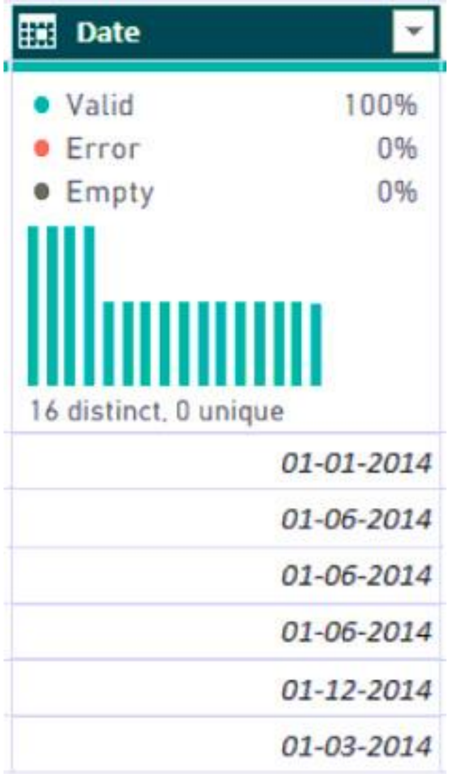
1. Microsoft Learn. (2023). DirectQuery model guidance in Power BI Desktop. In the "Data source guidance" section, it is emphasized that for relational sources, "columns used in relationships... should be indexed." This supports the "add an index key" part of the solution for performance.

2. Microsoft Learn. (2023). Model relationships in Power BI Desktop. The documentation on "Relationship cardinality" implicitly requires unique values on the "one" side of a one-to-many relationship. Case sensitivity in the source can violate this uniqueness from Power BI's perspective (e.g., 'A' and 'a' are different in the source but the same to Power BI), causing errors. Normalizing the casing resolves this violation.

3. Microsoft Learn. (2024). DAX overview. The documentation states, "DAX, like Microsoft Excel, is not case-sensitive." This highlights the conflict between the case-insensitive DAX engine and a case-sensitive DirectQuery source, which is the problem the solution addresses by modifying the source data.

Question: 26

You use Power Query Editor to preview a column named Date as shown in the following exhibit. You need to change the Date column to contain only the year. The solution must minimize administrative effort. What should you do?



Date

● Valid	100%
● Error	0%
● Empty	0%

16 distinct, 0 unique

Date
01-01-2014
01-06-2014
01-06-2014
01-06-2014
01-12-2014
01-03-2014

A: Split the column by delimiter.

B: Split the column by number of characters.

C: Extract the text after the delimiter.

D: Transform the column to contain only the year.

Correct Answer:

D

Explanation:

The most efficient method to change a column with a Date data type to contain only the year is to use the built-in date transformation feature in Power Query. By selecting the Date column and navigating to the Transform tab, you can select Date > Year > Year. This single action directly modifies the existing column to replace the full date values with their corresponding four-digit year, perfectly aligning with the requirement to minimize administrative effort.

Why Incorrect Options are Wrong:

A: Split the column by delimiter. This is a text-based operation that would require converting the column to text first and would create multiple new columns, adding unnecessary steps.

B: Split the column by number of characters. This is an unreliable text-based method, as the format and character count of dates can vary. It is not the intended way to parse date components.

C: Extract the text after the delimiter. This is also a text-based operation. It is less direct and efficient than using the dedicated date transformation functions available in Power Query.

References:

1. Microsoft Learn. (n.d.). Date and time column operations. In Power Query documentation. Retrieved from https://learn.microsoft.com/en-us/power-query/date-time-column-operations.

Reference Details: The "Date transformations" section describes the available operations on date columns. It explicitly lists "Year" as a transformation that can be applied directly to a date column from the Transform tab, which extracts the year component. This confirms that a dedicated, efficient function exists for the requested task.

2. Microsoft Learn. (n.d.). Tutorial: Shape and combine data in Power BI Desktop. Retrieved from https://learn.microsoft.com/en-us/power-bi/transform-model/desktop-shape-and-combine-data.

Reference Details: Under the section "Shape data," the tutorial demonstrates using the Power Query Editor's ribbon commands to transform data. It shows how to use the Transform tab for in-place modifications, including operations on date columns, which supports option D as the most direct approach. The text-based operations (Split, Extract) are shown in the context of text columns, not date columns.

Question: 27

You use Power BI Desktop to import two tables named Customer and Contacts. The Customer table contains the following columns: • Customer_Name • Customer ID • Website The Contacts table contains the following columns: • Contact ID • Contact Email • Contact Name • Customer Name A web-based contact form is used to fill the Contacts table. The data is not sanitized. You need to create a merge for the Customer and Contacts tables. What should you do?

A: Disable fuzzy matching.

B: Enable fuzzy matching.

C: Set Join Kind to Left Outer.

Correct Answer:

B

Explanation:

The scenario states that the Contacts table is populated from a web form and the data is "not sanitized." This implies that the Customer Name column may contain typos, variations in spelling, or different casing compared to the CustomerName column in the Customer table. A standard merge requires an exact match, which would fail in these cases. Enabling fuzzy matching in Power Query allows for merging tables based on columns that are similar but not identical, effectively resolving the data quality issues caused by the unsanitized input.

Why Incorrect Options are Wrong:

A: Disable fuzzy matching. This would perform a standard, exact-match merge, which would fail to link many records due to the unsanitized, inconsistent customer names.

C: Set Join Kind to Left Outer. The Join Kind (e.g., Left Outer, Inner) determines which rows are kept after a match is found, but it does not change the matching logic itself. It does not solve the core problem of matching inconsistent text values.

References:

1. Microsoft Documentation, Power Query. "Fuzzy merge in Power Query". This document explains the feature's purpose: "With a fuzzy merge, you can apply fuzzy matching algorithms to find matches across the tables that you're merging... A common use case for

fuzzy matching is with freeform text fields, such as in a survey." This directly addresses the scenario of unsanitized data from a web form.

2. Microsoft Documentation, Power Query. "Overview of merge queries". This page details the different Join Kind options, illustrating that they control the output of the merge (which rows are included) rather than the logic for identifying a match between two columns. This differentiates the function of Join Kind from fuzzy matching.

Question: 28

You need to create the relationship between the product list and the revenue results. The solution must minimize the time it takes to render visuals. What should you set as the relationship cardinality?

A: One to one

B: Many to one

C: One to many

D: Many to many

Correct Answer:

C

Explanation:

The scenario describes a classic star schema model. The 'product list' table is a dimension table, containing a unique row for each product, representing the "one" side of the relationship. The 'revenue results' table is a fact table, containing transactional data where a product can appear multiple times, representing the "many" side.

To optimize performance and minimize visual rendering times, a one-to-many cardinality is the most efficient and standard choice for this model. This relationship type is highly optimized in the Power BI engine, unlike many-to-many relationships which can introduce ambiguity and performance degradation.

Why Incorrect Options are Wrong:

A: One to one: This is incorrect because a single product is expected to have multiple revenue entries over time, not just a single corresponding record.

B: Many to one: While this describes the same relationship from the perspective of the fact table, the standard convention is to define the relationship from the dimension table ('one' side) to the fact table ('many' side).

D: Many to many: This cardinality is less performant and should be avoided unless necessary. It is not appropriate here as a standard one-to-many relationship correctly models the data.

References:

1. Microsoft Documentation, "Model relationships in Power BI Desktop": Under the "Cardinality" section, it states, "A one-to-many or many-to-one relationship is the most common, default type. It describes a relationship between a dimension and a fact table." This directly supports the chosen answer for this scenario.

2. Microsoft Documentation, "Understand star schema and the importance for Power BI": This document emphasizes that Power BI is optimized for star schema models, which are built upon relationships between dimension tables and fact tables. The document states, "Each dimension table is related to a fact table by using a one-to-many relationship." This reinforces that one-to-many is the correct and performant choice.

3. Microsoft Documentation, "One-to-many relationships guidance": This page provides specific guidance on using one-to-many relationships, noting they are the most common type and are used to connect lookup tables (dimensions) and fact tables, which is the exact scenario in the question.

Question: 29

You plan to develop a Power BI report that has a bar chart to display the number of customers by location. You have a table named Customer that has the following columns: * Customer ID * CustomerName * Address * City * ProvState * Country You need to allow users to drill down by location. The report will display the number of each customer by Country, and drill down to ProvState, and then to City. How should you configure the drill down in the bar chart?

A: In the Value field, add Country. In the Legend field, add ProvState at the top, followed by City.

B: In the Legend field, add Country. In the Axis field, add ProvState at the top, followed by City.

C: In the Axis field, add Country at the top, followed by ProvState, and then City.

D: In the Value field, add Country at the top, followed by ProvState, and then City.

Correct Answer:

C

Explanation:

To enable a drill-down feature in a Power BI bar chart, you must create a hierarchy in the visual's field wells. The Axis field is designated for the categorical data that forms the levels of the chart. By placing Country, ProvState, and City into the Axis field in that specific order (from top to bottom), you establish the desired drill-down path. The Value field would then be used for the numeric data to be measured, such as a count of Customer IThis configuration allows users to start at the Country level and progressively drill down to ProvState and then City.

Why Incorrect Options are Wrong:

A: The Value field is for numeric measures, not categorical fields. The Legend field is used for segmenting data, not for creating the primary drill-down axis.

B: The primary hierarchy should be built entirely within the Axis field. Placing the top level (Country) in the Legend field would incorrectly segment the data.

D: The Value field is exclusively for numeric data that will be aggregated (e.g., sum, count, average), not for the categorical fields that define the drill-down hierarchy.

References:

1. Microsoft Learn. (2023). Drill down in a visual in Power BI. This document explicitly shows how to create a drill-down by adding multiple fields to the Axis well of a visual. The example demonstrates creating a hierarchy by dragging fields under the Axis section of the Visualizations pane, which directly supports the correct answer.

2. Microsoft Learn. (2023). Create column charts in Power BI. In the section "Prerequisites," the tutorial guides the user to place a categorical field (e.g., Store > Chain) on the X-axis and a numeric field (e.g., Sales > This Year Sales > Value) on the Y-axis. This confirms the distinct roles of the Axis (for categories) and Value (for measures) fields.

3. Microsoft Learn. (2023). Use drillthrough in Power BI Desktop. While discussing a different feature, this page reinforces the concept of visual configuration, stating, "In the Fields section of the Visualizations pane, drag the field that you want to enable drillthrough for into the Drillthrough well." This illustrates that specific wells (like Axis, Legend, Value, Drillthrough) have designated purposes for fields.

Question: 30

You need to create the dataset. Which dataset mode should you use?

A: DirectQuery

B: Import

C: Live connection

D: Composite

Correct Answer:

B

Explanation:

The Import storage mode is the default and most frequently used option in Power BI. It involves loading a copy of the data into the Power BI model, where it is compressed and stored in-memory. This approach provides the fastest query performance because all data is available locally within the .PBIX file or the Power BI service. It also allows for the full use of Power Query for data transformation and the entire suite of DAX functions for creating complex calculations. Without specific requirements necessitating other modes, Import is the standard and recommended choice.

Why Incorrect Options are Wrong:

A: DirectQuery: This mode is used for very large datasets that cannot be imported or when near real-time data is required. It is not the default choice due to potential performance latency.

C: Live connection: This is a specific type of direct connection used only for existing semantic models, such as SQL Server Analysis Services (SSAS) or other Power BI datasets, not for creating a new model from a source like a database.

D: Composite: This is an advanced mode that combines Import and DirectQuery sources. It is used for specific, complex scenarios and is not the standard starting point for creating a dataset.

References:

1. Microsoft Learn. (2023). Dataset modes in the Power BI service.

Section: "Import and DirectQuery modes"

Content: This document explicitly states, "For smaller datasets, the default and most common mode is import mode." It details that Import mode provides the fastest performance and supports the full Power Query M and DAX functionalities.

2. Microsoft Learn. (2023). Use DirectQuery in Power BI Desktop.

Section: "When to use DirectQuery"

Content: This documentation outlines the specific scenarios where DirectQuery is a better solution than Import mode, such as when data changes frequently and near real-time reporting is needed, or when the data is very large. This implicitly positions Import as the standard for cases not meeting these criteria.

3. Microsoft Learn. (2023). Use composite models in Power BI Desktop.

Section: "Use composite models"

Content: This source describes composite models as a feature that allows a report to have two or more data connections from different source groups, such as one or more DirectQuery connections and an import connection. This highlights its use for more complex, mixed-source scenarios rather than a primary, default choice.