

POWER BI

PL-300



Question 1:

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?

Options:

- A. Microsoft Teams Personal Analytics
- B. SQL Server database
- C. Dataverse
- D. Dataflows

Answer:

- C. Dataverse

Solution:

Microsoft **Dataverse** is the preferred data storage solution for Power Apps, especially when hosted in **Microsoft Teams**. Since the project management app was built using **Power Apps**, it is highly likely that its data is stored in **Dataverse for Teams**. Power BI has a direct **Dataverse connector**, making it easy to retrieve and analyse app data. Dataverse is the preferred data storage solution for Power Apps and integrates seamlessly with Power BI for reporting.

Question 2:

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?

Options:

- A. Power BI dataset
- B. A SharePoint folder
- C. Power BI dataflows
- D. An Excel workbook

Answer:

- A. Power BI dataset

Solution:

A **Power BI dataset** is the best choice because the data has already been imported, processed, and includes measures in the existing report. By connecting to the **Power BI dataset**, you can **reuse the existing model** with minimal effort, avoiding the need to re-import or reprocess data.

Question 3:

You import two Microsoft Excel tables named **Customer** and **Address** into Power Query.

Customer Table Columns:

• Customer ID
• Customer Name
• Phone
• Email Address
• Address ID

Address Table 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.

Options:

- 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.

Answer:

- A. Merge the **Customer** and **Address** tables.

Solution:

To get **one row per customer** with their respective **City, State/Region, and Country**, you need to **merge** the **Customer** and **Address** tables using the **Address ID** column as the common key. **Merging** creates a single table that includes customer details along with location data.

Question 4:

You have two **Azure SQL databases** that contain the **same tables and columns**. For each database, you create a query that retrieves data from a table named **Customer**.

You need to **combine the Customer tables into a single table** while meeting the following requirements:

- **Minimize the size of the data model**
- **Support scheduled refresh in PowerBI.com**

What should you do?

Answer Area

Option to use to combine the Customer tables:

	▼
Append Queries	
Append Queries as New	
Merge Queries	
Merge Queries as New	

Action to perform on the original two SQL database queries:

	▼
Delete the queries	
Disable including the query in report refresh	
Disable loading the query to the data model	
Duplicate the queries	

Answer:

- ✓ **A. Append Queries as New**
- ✓ **B. Disable loading the query to the data model**

Box 1: Append Queries as New -

When you have additional rows of data that you'd like to add to an existing query, you append the query.

There are two append options:

- * Append queries as new displays the Append dialog box to create a new query by appending multiple tables.
- * Append queries displays the Append dialog box to add additional tables to the current query.

Box 2: Disable loading the query to the data model

By default, all queries from Query Editor will be loaded into the memory of Power BI Model. You can disable the load for some queries, especially queries that used as intermediate transformation to produce the final query for the model.

Disabling Load doesn't mean the query won't be refreshed, it only means the query won't be loaded into the memory. When you click on Refresh model in Power BI, or when a scheduled refresh happens even queries marked as Disable Load will be refreshed, but their data will be used as intermediate source for other queries instead of loading directly into the model. This is a very basic performance tuning tip, but very important when your Power BI model grows bigger and bigger.

Question 5:

In **Power Query Editor**, you have three queries:

1. ProductCategory
2. ProductSubCategory
3. Product

- Every **Product** has a **ProductSubCategory**.
- Not every **ProductSubCategory** has a parent **ProductCategory**.
- You need to merge the three queries into a **single query** while ensuring **best performance** in Power Query.

Solution:

To merge the tables efficiently, follow these steps:

1. **Merge Product with ProductSubCategory using an Inner Join on ProductSubCategoryID**
 - Since every **Product** has a corresponding **ProductSubCategory**, an **Inner Join** ensures only matching records are included, improving performance.
2. **Merge ProductSubCategory with ProductCategory using a Left Outer Join on ProductCategoryID**
 - Since **not every ProductSubCategory** has a **ProductCategory**, a **Left Outer Join** ensures that all subcategories are retained, even if they lack a parent category.

Final Mapping:

Query 1 (First Merge)	Query 2 (Second Merge)	Merge Type
Product	ProductSubCategory	Inner Join
ProductSubCategory	ProductCategory	Left Outer Join

Question 6:

You are building a **Power BI report** using data from an **Azure SQL database** named **erp1**.

You have the following tables:

• Product: Contains the product catalogue.
• Order: Contains high-level order information.
• Order Line Items: Contains Product ID, Quantity, and Price details of each order.

You need to analyse:

1. **Orders sold over time**, including a **measure of total order value**.
2. **Orders by attributes of products sold**.

The solution must **minimize update times** when interacting with visuals.

Options:

- 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 using **DAX**.
- C. Calculate the **count of orders per product** using **DAX**.
- D. From Power Query, merge the Orders query and the Order Line Items query.**

Answer:

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

Solution Explanation:

- The "**Order Line Items**" table contains **Product ID, Quantity, and Price**, which are essential to calculate the **total order value**.
- The "**Orders**" table contains **order dates**, which are necessary for analyzing **orders over time**.
- Merging "**Orders**" and "**Order Line Items**" in **Power Query** creates a **single table** that allows **faster aggregations** (e.g., total revenue, order count).
- Merging in **Power Query** (before loading data into Power BI) reduces **runtime calculations**, improving **performance**.

Question 7:

You have a **Microsoft SharePoint Online** site that contains several **document libraries**. One of the **document libraries** stores **manufacturing reports** as **Microsoft Excel files**, all having the **same data structure**.

You need to **load only the manufacturing reports** into **Power BI Desktop** for analysis.

Options:

- 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**.

Answer:

- 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.**

Solution Explanation:

- **SharePoint folders** store files like Excel, Word, or CSV, making them ideal for **retrieving document-based data**.

- **The correct approach** is to use "**Get data from a SharePoint folder**" instead of a **SharePoint list**, since lists store structured data, not files.
- **Filtering by the folder path** ensures that **only manufacturing reports** are loaded, preventing unnecessary data from other document libraries.
- **Selecting "Transform" instead of "Combine & Load"** allows data cleaning before loading, improving performance.

Question 8:

You have a **Microsoft Excel workbook** with **two sheets**:

- **Sheet1** contains **Table1**
- **Sheet2** contains **Table2**

You need to use **Power Query Editor** to **combine** the products from **Table1 and Table2** into **one column** without duplicates.

Tables:

Table1 (Sheet1)

Product
Apple
Banana
Orange
Mango

Table2 (Sheet2)

Product
Banana
Grape
Apple
Pineapple

Solution Steps:

1 Load Table1 and Table2 into Power Query

- Go to **Power Query Editor** and import both **Table1** and **Table2**.

2 Append Queries

- Use the "**Append Queries**" feature to **combine both tables** into one. This will merge the product lists.

3 Remove Duplicates

- Apply the "**Remove Duplicates**" function in Power Query to keep **only unique product names**.

Question 9:

You have a **CSV file** containing **user complaints**. The file includes a column named **Logged**, which records the **date and time** each complaint occurred.

The format of **Logged** is: **2018-12-31 at 08:59**

You need to **analyze** the complaints based on the **logged date** and use a **built-in date hierarchy** in Power BI.

Options:

- A. **Apply a transformation** to extract the **last 11 characters** of the Logged column and set the data type to **Date**.
- B. **Change the data type** of the Logged column to **Date**.
- C. **Split the Logged column** by using **"at"** as the delimiter.
- D. **Apply a transformation** to extract the **first 11 characters** of the Logged column.

Answer:

- C. Split the Logged column by using "at" as the delimiter

Question 10:

You have a **Microsoft Excel file** stored in a **Microsoft OneDrive folder**. The file must be **imported into a Power BI dataset**.

You need to ensure that the **dataset can be refreshed** in **powerbi.com**.

Options:

- A. **Excel Workbook**
- B. **Text/CSV**
- C. **Folder**
- D. **SharePoint folder**
- E. **Web**

Answer:

✓ D. SharePoint folder

✓ E. Web

Solution Explanation:

1 **Power BI requires a cloud-accessible connection** to refresh datasets in **powerbi.com**.

- Since the **file is in OneDrive**, we need a connector that supports cloud-based refresh.

2 **Option D - SharePoint folder (✓ Correct)**

- OneDrive is **built on SharePoint**, so using the **SharePoint folder** connector provides a direct connection to OneDrive files.
- This ensures **automatic refresh** in Power BI Service.

3 **Option E - Web (✓ Correct)**

- The **Web connector** allows **direct access to OneDrive files** using their **URL**.
- If you use the **OneDrive for Business link**, Power BI can refresh the data automatically.

Question 11:

You have two **CSV files** named **Products** and **Categories**.

The **Products** file contains:

• ProductID
• ProductName
• SupplierID
• CategoryID

The **Categories** file contains:

• CategoryID
• CategoryName
• CategoryDescription

You need to create a **single table named Product** in Power BI with the following columns:

• ProductID
• ProductName
• SupplierID
• CategoryID
• CategoryName
• CategoryDescription

Options:

How should you combine the queries?

- Merge (✓ Correct)
- Append
- Transpose

What should you do on the Categories query?

- Disable the query load (✓ Correct)
- Delete the query
- Exclude the query from report refresh

Solution

1 Combining Queries - Merge (Correct)

- **Why Merge?**
 - We need to **combine** data from both tables **based on the CategoryID**.
 - The **Products table has CategoryID** but does **not** have CategoryName and CategoryDescription.
 - The **Categories table has CategoryID** along with CategoryName and CategoryDescription.
 - **Merging** will allow us to bring these extra columns into the **Products table**.

2 Disabling Query Load - Disable the query load (Correct)

- **Why Disable Query Load?**
 - Once we **merge** the Categories table into Products, we **no longer need the Categories table to load separately into the dataset**.
 - Disabling query load **prevents unnecessary data duplication**, improving performance.

Question 12:

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?

Options:

- A. Add a SQL statement.
- B. Set the Command timeout in minutes setting.

<https://www.linkedin.com/in/imratankj/>

C. Set Data Connectivity mode to Import.

D. Set Data Connectivity mode to DirectQuery.

Answer:

✓ D. Set Data Connectivity mode to DirectQuery.

Solution:

- **DirectQuery** allows Power BI to fetch data **directly from the database** in real-time or near real-time.
- This mode ensures that **any updates in Azure SQL** are reflected in reports **without manual refreshes**.

Since the **Azure SQL database is updated frequently** and **you need to see updates within five minutes**, **DirectQuery** is the best option.

Why DirectQuery?

- **Real-time or near real-time data:** DirectQuery **fetches data directly** from the source whenever a user interacts with a report.
- **No need to refresh manually:** Unlike Import mode, which requires scheduled refreshes, DirectQuery ensures the latest data is **always** shown in reports.
- **Ideal for large datasets:** Since no data is stored in Power BI, it avoids unnecessary memory consumption.

Question 13:

You have a folder that contains 100 CSV files.

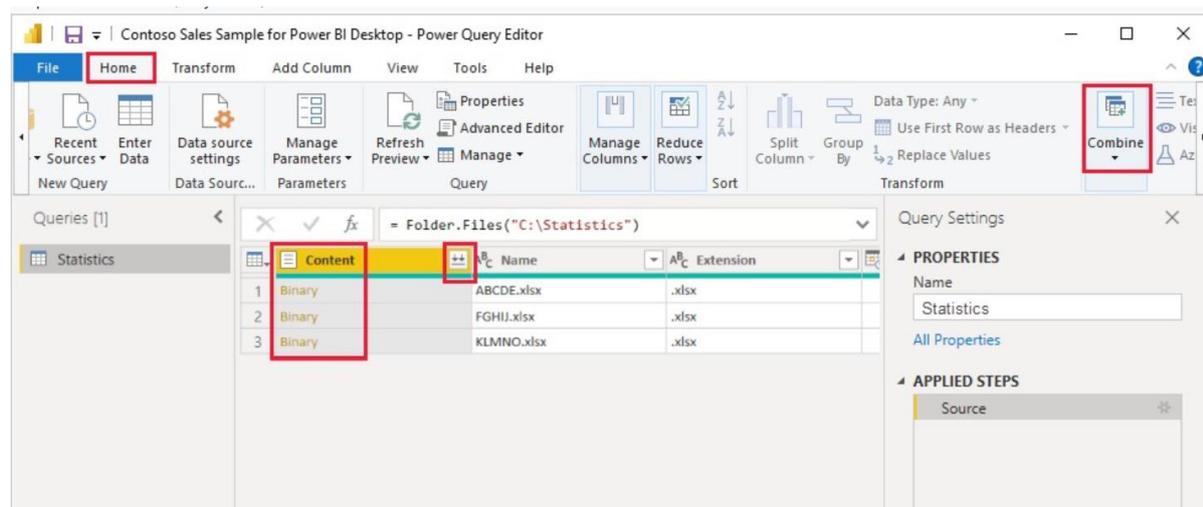
You need to make the file metadata available as a single dataset by using Power BI. The solution must **NOT** store the data of the CSV files.

Which three actions should you perform in sequence?

Correct Answer:

Actions	Answer Area
From Power BI Desktop, select Get Data , and then select Folder.	From Power BI Desktop, select Get Data , and then select Folder.
From Power Query Editor, expand the Attributes column.	From Power Query Editor, expand the Attributes column.
From Power Query Editor, remove the Content column.	From Power Query Editor, combine the Content column.
From Power Query Editor, remove the Attributes column.	
From Power BI Desktop, select Get Data, and then select Text/CSV.	
From Power Query Editor, combine the Content column.	

Navigation arrows: > and <



Step 1: From Power BI Desktop, Select Get Data, and then Select Folder.

Open Power BI Desktop and then select Get Data\More€ and choose Folder from the All options on the left.

Enter the folder path, select OK, and then select Transform data to see the folder's files in Power Query Editor.

Step 2: From Power Query Editor, expand the Attributes column.

Step 3: From Power Query Editor, combine the Content column.

Combine files behavior -

To combine binary files in Power Query Editor, select Content (the first column label) and select Home > Combine Files. Or you can just select the Combine Files icon next to Content.

Question 14:

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?

Options:

- 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.

Answer:

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

Solution Explanation:

- **DirectQuery mode** would result in frequent online queries, increasing processing load and slowing visual rendering.
- **Import mode (Best Choice)** loads data **once per refresh** and improves performance since calculations happen in-memory.
- **A daily refresh** ensures data is **updated up to the previous day** without frequent querying, meeting the requirement efficiently.

Question 15:

You publish a dataset that contains data from an **on-premises Microsoft SQL Server database**.

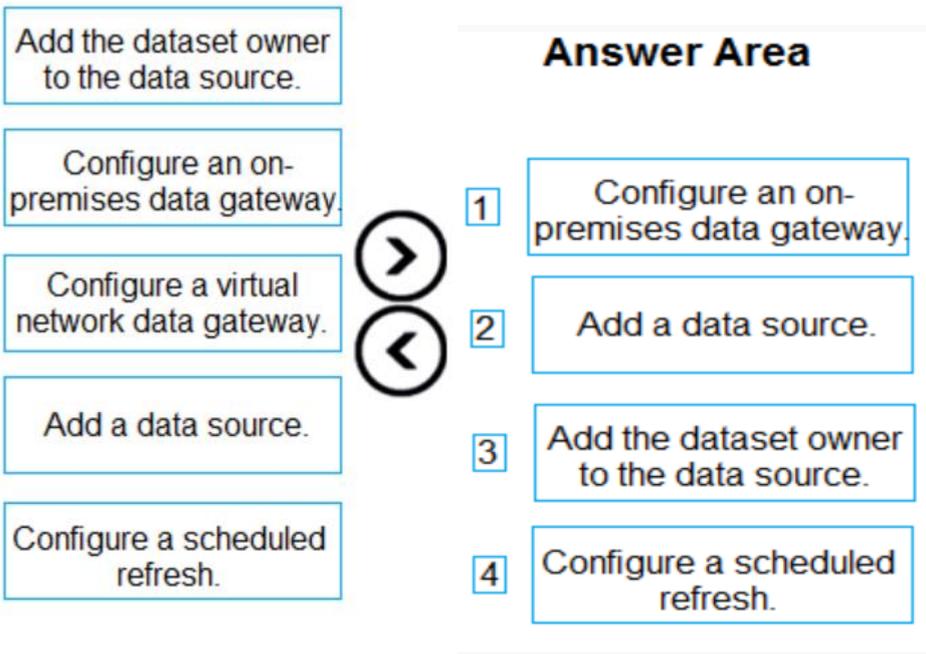
The dataset must be **refreshed daily**.

You need to ensure that the **Power BI service** can connect to the database and refresh the dataset.

Actions to Perform

- 1 **Install and configure an on-premises data gateway** on the server hosting the SQL database.
- 2 **Register the gateway in the Power BI service** under the "Manage Gateways" section.
- 3 **Configure the dataset settings** in Power BI to use the gateway for the SQL Server connection.
- 4 **Schedule the dataset refresh** in Power BI Service to ensure daily updates.

Actions



Question 16:

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 Cassandra.

You need to select an **alternate data connector** that will connect to the database.

Options:

- A. Microsoft SQL Server database
- B. ODBC**
- C. OLE DB
- D. OData

Answer:

- B. ODBC**

Solution Explanation:

- **Apache Cassandra** does not have a **native connector** in Power BI.
- However, Cassandra supports **ODBC drivers**, which allow Power BI to establish a connection.
- **ODBC (Open Database Connectivity)** provides a **generic way** to connect to databases without native support.
- **OLE DB** is mainly for **SQL-based** databases and is **less commonly** used for Cassandra.

Question 17:

You receive **annual sales data** that must be included in **Power BI reports**.

From **Power Query Editor**, you connect to the **Microsoft Excel** source shown in the exhibit.

You need to create a **report** that meets the following requirements:

- ✓ **Visualizes the Sales value** over a **period of years and months**.
- ✓ **Adds a slicer** for the **month**.
- ✓ **Adds a slicer** for the **year**.

	Month	MonthNumber	2019	2020	2021
1	Jan	1	345	5526	3456
2	Feb	2	758	773	0
3	Mar	3	37763	570	null
4	Apr	4	8364	9417	null
5	May	5	58256	276	null
6	June	6	6722	235	null
7	July	7	55225	6297	null
8	Aug	8	673	63	null
9	Sep	9	552	357	null
10	Oct	10	7838	24214	null
11	Nov	11	83544	257	null
12	Dec	12	32455	389	null

Actions

- Select the Month and MonthNumber columns.
- Select **Unpivot other columns**.
- Rename the Attribute column as Year and the Value column as Sales.
- Select the 2019, 2020, and 2021 columns.
- Select **Transpose**.



Answer Area

- 1 Select the Month and MonthNumber columns.
- 2 Select **Unpivot other columns**.
- 3 Rename the Attribute column as Year and the Value column as Sales.

Question 18:

You are using Power BI Desktop to connect to an Azure SQL database. The connection is configured as shown in the following exhibit.

SQL Server database

Server

Database (optional)

Data Connectivity mode Import DirectQuery

Advanced options

Command timeout in minutes (optional)

SQL statement (optional, requires database)

Include relationship columns

Navigate using full hierarchy

Enable SQL Server Failover support

The default timeout for the connection from Power BI Desktop to the database will be

- unlimited
- one minute
- 10 minutes

The Navigator will display

- all the tables
- only tables that contain data
- only tables that contain hierarchies

Answer:

- 10 minutes
- Only tables that contain data

Question 19:

You have the Azure SQL databases shown in the following table.

Name	Stage	Server URL
db-powerbi-dev	Development	dev.database.windows.net
db-powerbi-uat	Test	uat.database.windows.net
db-powerbi-prod	Production	prod.database.windows.net

You plan to build a single PBIX file to meet the following requirements:

- Data must be consumed from the database that corresponds to each stage of the development lifecycle.
- Power BI deployment pipelines must **NOT** be used.
- The solution must minimize administrative effort.

What should you do?

Options:

Create:

- A. One parameter
- B. Two parameters
- C. Three parameters

Parameter Type:

- A. Text
- B. True/False
- C. Decimal Number

Answer:

- **Create:** One parameter
- **Parameter Type:** Text

Solution Explanation:

1. Since the database names follow a structured pattern corresponding to different stages (e.g., Dev, Test, Prod), you only need **one parameter** to dynamically switch between them.
2. The **Text** data type is the best choice because the parameter will store the **database name or connection string**, which is a string value.
3. Using a **single parameter** simplifies administration, as it allows users to switch between databases without manually modifying queries.

4. When configuring Power Query, you can use this parameter to define the database source dynamically.
5. This approach **avoids the need for multiple queries** or hardcoded database connections, making the solution more flexible.
6. Since **Power BI deployment pipelines are not being used**, setting up a parameterized connection manually is the best approach.

Question 20:

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.

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

You need to create the **dimension**. The dimension must contain a **list of unique countries**.

Options:

- 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.

Answer:

- **D. Delete the City column**
- **E. Remove duplicates from the Country column**

Solution Explanation:

1. A **Country dimension** in a star schema should contain only **unique country names** without additional unnecessary columns.

2. **Deleting the City column (D)** ensures that only country-level data is retained, making the dimension table more efficient.
3. **Removing duplicates from the Country column (E)** ensures that each country appears only once in the dimension table.

Question 21:

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?

Options:

- 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.

Answer:

- A. You do not have permissions to the file.
- D. The referenced file was moved to a new location.

Solution Explanation:

1. **(A) You do not have permissions to the file**
 - If Power BI does not have the required **read permissions**, it will not be able to access the file, resulting in a **"Could not find file"** error.
 - This can happen if the file is in a restricted folder or requires administrator access.
2. **(D) The referenced file was moved to a new location**
 - If the file path has changed or the file was deleted, Power Query will **fail to locate** it and return this error.
 - This is common when files are **renamed, moved to another folder, or deleted**.