POWER BI

PL-300

PART -4

Question 74:

You are building a financial report by using Power BI.
You have a table named financials that contains a column named Date and a column named Sales.
You need to create a measure that calculates the relative change in sales as compared to the previous quarter.
How should you complete the measure? To answer, select the appropriate options in the answer area.

Answer:

Answer Area

```
Sales QoQ% =
IF(
        ISFILTERED('financials'[Date]),
        ERROR("Uh oh."),
        VAR PREV_QUARTER =
                [CALCULATE / CALCULATETABLE / DATEADD / DIVIDE / FILTER / FIND]
                (SUM('financials'[Sales]),
                        [CALCULATE / CALCULATETABLE / DATEADD / DIVIDE / FILTER / FIND]
                        ('financials'[Date].[Date], -1, QUARTER)
                )
        RETURN
                [CALCULATE / CALCULATETABLE / DATEADD / DIVIDE / FILTER / FIND]
                (SUM('financials'[Sales]) - PREV_QUARTER, PREV_QUARTER)
)
```

ANSWER

1. **CALCULATE**
2. **DATEADD**
3. **DIVIDE**

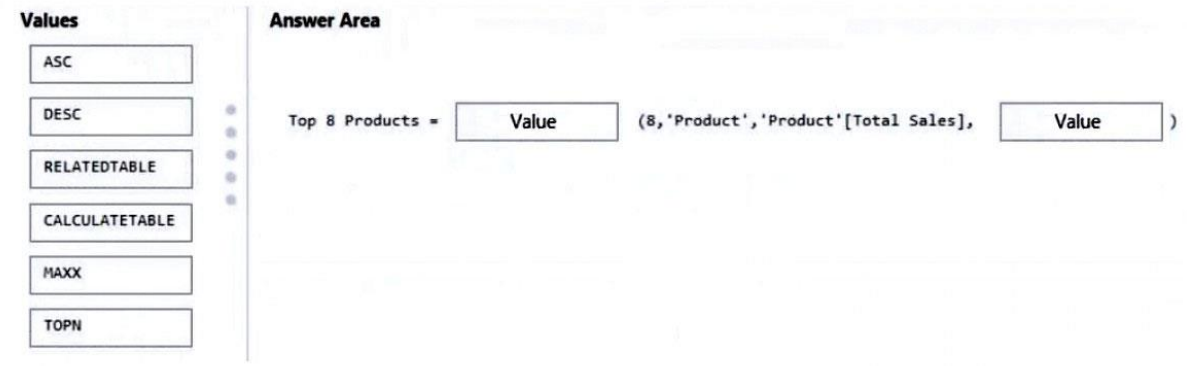
Question 75:

You are creating a Power BI model and report.
You have a single table in a data model named **Product**. The **Product** table contains the following fields:

- **ID**
- **Name**
- **Color**
- **Category**
- **Total Sales**

You need to create a **calculated table** that shows only the **top eight products** based on the highest value in **Total Sales**.

How should you complete the **DAX expression**?



Answer:

1. **TOPN**
2. **DESC**

Question 76:

You are creating a **sales report** in Power BI for the **NorthWest region** sales territory of your company. Data will come from a **view** in a **Microsoft SQL Server database**.

ID	ProductKey	OrderDate	ShipDate	CustomerKey	SalesTerritoryRegion	SalesOrderNumber	SalesOrderLineNumber	OrderQuantity	UnitPrice	SalesAmount	TaxAmount	Freight
1	310	2010-12-29	2011-01-05	21768	Canada	SO43697	1	1	3578.27	3578.27	286.2616	89.4568
2	346	2010-12-29	2011-01-05	27365	France	SO43698	1	1	3399.99	3399.99	271.9992	84.9998
3	346	2010-12-29	2011-01-05	76537	NorthWest	SO43699	1	1	3399.99	3399.99	271.9992	84.9998
4	336	2010-12-29	2011-01-05	34256	SouthWest	SO43700	1	1	699.0982	699.0982	55.9279	17.4775
5	346	2010-12-29	2011-01-05	34253	Australia	SO43701	1	1	3399.99	3399.99	271.9992	84.9998
6	311	2010-12-30	2011-01-06	12543	SouthWest	SO43702	1	1	3578.27	3578.27	286.2616	89.4568
7	310	2010-12-30	2011-01-06	76545	Australia	SO43703	1	1	3578.27	3578.27	286.2616	89.4568

The report will facilitate the following analysis:

- **The count of orders and the sum of total sales by Order Date**

☞ **The count of customers who placed an order**
☞ **The average quantity per order**

You need to **reduce data refresh times** and **report query times**.

Options:

A. Set the data type for **SalesOrderNumber** to **Decimal Number**.
B. Remove the **CustomerKey** and **ProductKey** columns.
C. Remove the **TaxAmt** and **Freight** columns.
D. **Filter the data** to only the **NorthWest** region sales territory.

Answer:

✅ **C. Remove the TaxAmt and Freight columns.**
✅ **D. Filter the data to only the NorthWest region sales territory.**

Solution Explanation:

1. **Removing unnecessary columns (TaxAmt, Freight) (Option C)**
    - Columns that are **not required** for the report should be removed to **reduce model size** and **improve performance**.
    - Since **TaxAmt and Freight** are not needed for the required analysis, removing them **optimizes data refresh** and **query times**.
2. **Filtering data to only the NorthWest region (Option D)**
    - The report is **only for the NorthWest region**, so **filtering the data at the source** will reduce the **dataset size** and **improve performance**.
    - Power BI will not need to process unnecessary rows, which results in **faster queries**.

Question 77:

You are creating a **Power BI model** that contains a table named **Store**. The table includes fields for:

- **City**
- **State/Province**
- **Country**

Name	Data type
Store ID	Whole Number
Store Name	Text
City	Text
State/Province	Text
Country	Text

You plan to create a **map visual** that will show **store locations** and provide the ability to **drill down** from **Country → State/Province → City**.

What should you do to ensure that the locations are **mapped properly**?

Options:

A. Change the **data type** of City, State/Province, and Country.
B. Set **Summarization** for City, State/Province, and Country to **Don't summarize**.
C. Set the **data category** of City, State/Province, and Country.
D. Create a **calculated column** that concatenates the values in City, State/Province, and Country.

Answer:

✅ **C. Set the data category of City, State/Province, and Country.**

Solution Explanation:

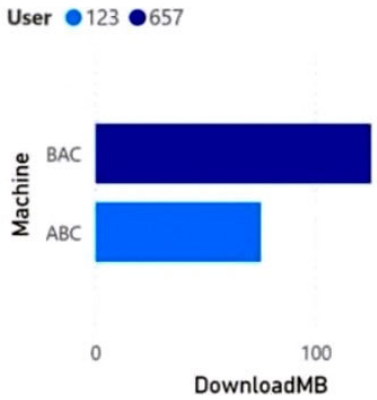
1. **Power BI's Map Visual requires proper geographic categorization**
    - To correctly **plot locations on a map**, Power BI needs to understand the meaning of **City, State/Province, and Country** fields.
    - **Setting the Data Category** (Option **C**) tells Power BI to treat them as **geographic locations**.
2. **How to Set Data Categories?**
    - Go to **Model View** in Power BI.
    - Select **City, State/Province, and Country** fields.
    - In the **Properties Pane**, set the **Data Category** to:
        - **Country** → *Country/Region*
        - **State/Province** → *State or Province*
        - **City** → *City*

Question 78:

You are building a data model for a Power BI report.
You have data formatted as shown in the following table.

Machine-User	DownloadMB
ABC-123	75
BAC-657	125

You need to create a clustered bar chart as shown in the following exhibit.



Option:

- A. From Power Query Editor, split the Machine-User column by using a delimiter.
- B. From Power Query Editor, create a column that contains the last three digits of the Machine-User column.
- C. In a DAX function, create two calculated columns named Machine and User by using the SUBSTITUTE function.
- D. In a DAX function, create two measures named Machine and User by using the SUBSTITUTE function.

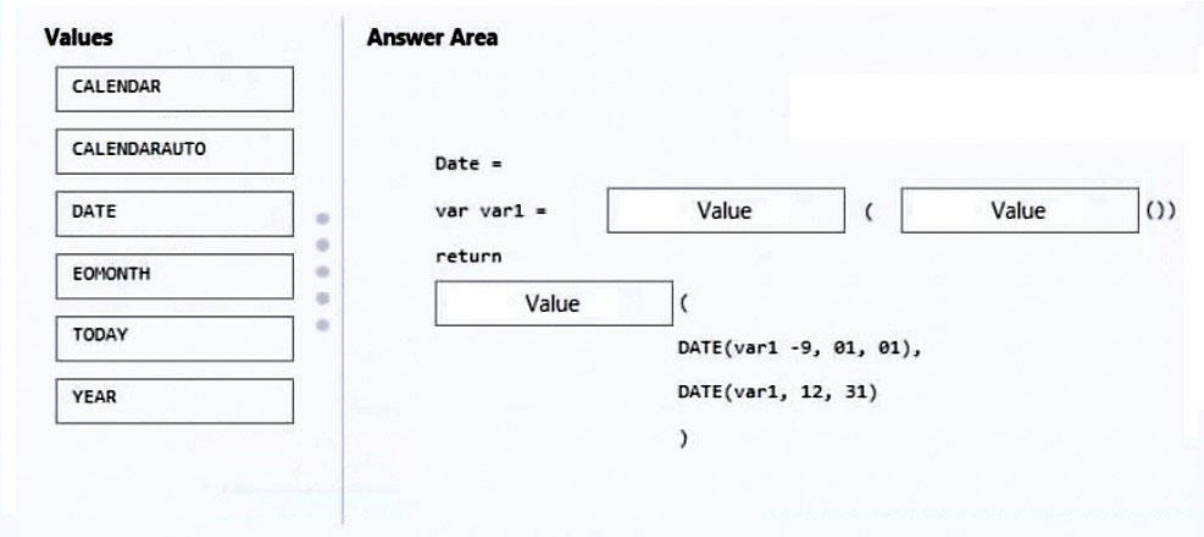
Answer:

✅ **A. From Power Query Editor, split the Machine-User column by using a delimiter**

Question 79:

You need create a date table in Power BI that must contain 10 full calendar years, including the current year.
How should you complete the DAX expression? To answer, select the appropriate options in the

Select and Place:



Answer:

✅ **A. YEAR**

✅ **B. TODAY**

✅ **C. CALENDER**

Question 80:

Question:

You have a Power BI report that imports a date table and a sales table from an Azure SQL database data source. The sales table has the following date foreign keys:

- **Due Date**
- **Order Date**
- **Delivery Date**

You need to support the analysis of sales over time based on all the date foreign keys.

Solution: You create measures that use the USERELATIONSHIP DAX function to filter sales on the active relationship between the sales table and the date table.

Does this meet the goal?

Options:

A. Yes
B. No

Answer:

❌ **B. No**

Solution Explanation:

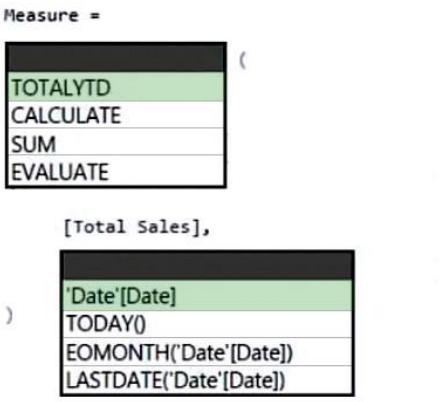
The USERELATIONSHIP() function in DAX is used to activate inactive relationships within a calculation. However, in this scenario, it is applied to an active relationship, which is not possible. Power BI automatically uses the active relationship without needing USERELATIONSHIP(). To analyze sales using all date foreign keys, the correct approach is to create multiple date tables (e.g., DueDateTable, OrderDateTable, DeliveryDateTable) and establish separate relationships for each. This allows filtering sales based on different dates without conflicts.

Question 81:

You have a Power BI report that contains a measure named Total Sales.
You need to create a new measure that will return the sum of Total Sales for a year up to a selected date.
How should you complete the DAX expression? To answer, select the appropriate options in the answer area.

Answer:

Answer Area



Answer:

✅ **A. TOTALYTD**

✅ **B. 'DATE'[DATE]**

Syntax

```
DAX
TOTALYTD(<expression>,<dates>[,<filter>][,<year_end_date>])
```

Parameters

Expand table

Parameter	Definition
`expression`	An expression that returns a scalar value.
`dates`	A column that contains dates.
`filter`	(optional) An expression that specifies a filter to apply to the current context.
`year_end_date`	(optional) A literal string with a date that defines the year-end date. The default is December 31.

Question 82:

You are modifying a Power BI model by using Power BI Desktop.
You have a table named Sales that contains the following fields.

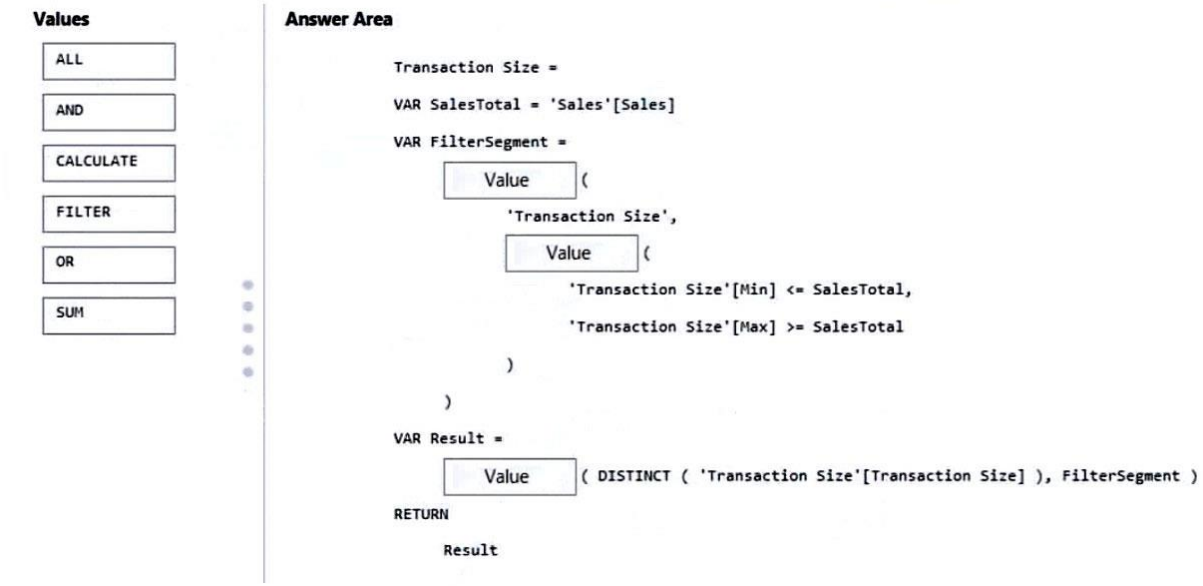
Name	Data type
Transaction ID	Whole Number
Customer Key	Whole Number
Sales Date Key	Date
Sales Amount	Whole Number

You have a table named Transaction Size that contains the following data.

Transaction Size ID	Transaction Size	Min	Max
1	Small	0	10,000
2	Medium	10,001	100,000
3	Large	100,001	999,999,999

You need to create a calculated column to classify each transaction as small, medium, or large based on the value in Sales Amount.
How should you complete the code? To answer, drag the appropriate values to the correct targets. Each value may be used once, more than once, or not at all.

Select and Place:



Answer:

✅ **Box 1: FILTER**

✅ **Box 2: AND**

✅ **Box 3: CALCULATE**

Question 83:

You have a Power BI report for the procurement department. The report contains data from the following tables.

Table name	Source	Description	Column name	Approximate record count
Suppliers	Microsoft Dynamics 365	A list of all the suppliers approved for use by the company.	• ID • Name • Country	100,000
LineItems	Microsoft Dynamics 365	All individual purchases made by employees across the company. An average of five line items per invoice.	• ID • Invoice ID • Invoice Date • Supplier ID • Description • Units • Price per Unit • Discount • Price	1,000,000,000

There is a one-to-many relationship from Suppliers to LineItems that uses the ID and Supplier ID columns.
The report contains the visuals shown in the following table.

Name	Used field	Filter

Supplier usage by count and value of invoices	`Suppliers[ID]`
	`Suppliers[Name]`
	`LineItems[Invoice ID]`
	`LineItems[Price]`
		None

Spend by supplier location	`Suppliers[Country]` `LineItems[Price]`	*None*
Top 10 largest invoices last month	`LineItems[Invoice ID]` `LineItems[Price]`	`LineItems[Invoice Date]` in last calendar month

You need to minimize the size of the dataset without affecting the visuals.
What should you do?

OPTIONS

A. Merge Suppliers and LineItems.

B. Remove the LineItems[Description] column.

C. Remove the rows from LineItems where LineItems[Invoice Date] is before the beginning of last month.

D. Group LineItems by LineItems[Invoice ID] and LineItems[Invoice Date] with a sum of LineItems[Price].

Answer:

✅ **B. Remove the LineItems[Description] column.**

Question 84:

You have a Power BI report for the marketing department. The report reports on web traffic to a blog and contains data from the following tables.

Table name	Source	Description	Column name
Posts	Blog RSS feed	An XML representation of all the blog posts from your company's website	• Publish Date • URL • Title • Full Text • Summary
Traffic	Website logs	Activity data from your company's entire website	• DateTime • URL Visited • IP Address • Browser Agent • Referring URL

There is a one-to-many relationship from Posts to Traffic that uses the URL and URL Visited columns.
The report contains the visuals shown in the following table.

Name	Used field	Filter
Top 10 blog posts of all time	`Posts[Title]` `Traffic[DateTime]`	*None*
Top 10 blog posts from the last seven days	`Posts[Title]` `Traffic[DateTime]`	`Traffic[DateTime]` is in the last 7 days
Blog visits over time	`Traffic[DateTime]` `Traffic[URL Visited]`	`Traffic[URL Visited]` contains "blog"
Top 10 external referrals to the blog of all time	`Traffic[Referring URL]`	`Traffic[URL Visited]` contains "blog" AND `Traffic[Referring URL]` does not start with "/"

The dataset takes a long time to refresh.
You need to modify Posts and Traffic queries to reduce load times.
Which two actions will reduce the load times? Each correct answer presents part of the solution.

OPTIONS:

- A. Remove the rows in Posts in which Posts[Publish Date] is in the last seven days.
- B. Remove the rows in Traffic in which Traffic[URL Visited] does not contain blog.
- C. Remove Traffic[IP Address], Traffic[Browser Agent], and Traffic[Referring URL].
- D. Remove Posts[Full Text] and Posts[Summary].
- E. Remove the rows in Traffic in which Traffic[Referring URL] does not start with /.

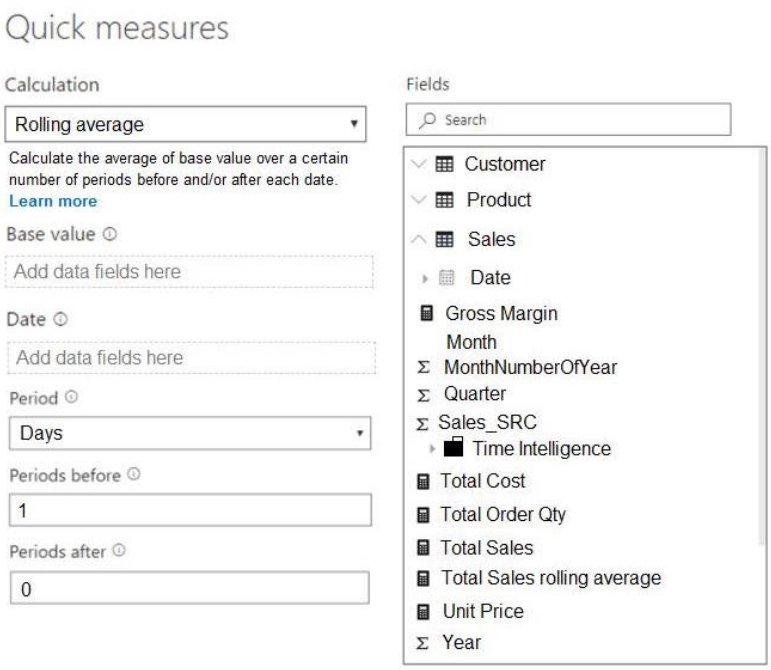
Answer:

✅ **B. Remove the rows in Traffic in which Traffic[URL Visited] does not contain blog.**

✅ **D. Remove Posts[Full Text] and Posts[Summary].**

Question 85:

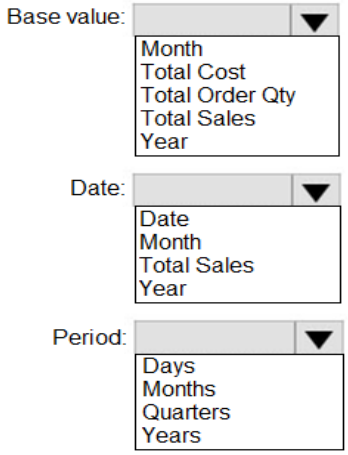
You are creating a quick measure as shown in the following exhibit.



You need to create a monthly rolling average measure for Sales over time.

How should you configure the quick measure calculation? To answer, select the appropriate options in the answer area.

Answer Area



Answer:

A. Total Sales

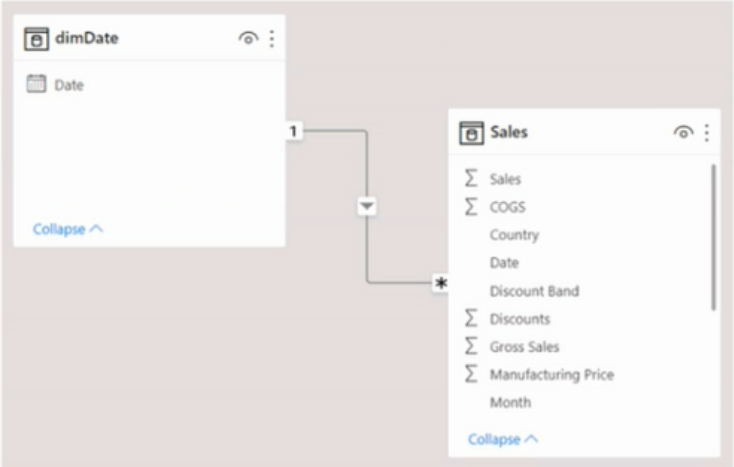
B. Date

C. Months

Question 86:

Question:

You have the Power BI data model shown in the exhibit.



The **Sales** table contains records of sales by day from the last five years up until today's date.

You plan to create a measure to return the **total sales of March 2021 when March 2022 is selected**.

Which DAX expression should you use?

Options:

A. CALCULATE(SUM(Sales[Sales]), PREVIOUSYEAR(dimDate[Date]))
B. TOTALYTD(SUM(Sales[Sales]), dimDate[Date])
C. CALCULATE(SUM(Sales[Sales]), SAMEPERIODLASTYEAR(dimDate[Date]))
D. SUM(Sales[Sales])

Answer:

✅ **C. CALCULATE(SUM(Sales[Sales]), SAMEPERIODLASTYEAR(dimDate[Date]))**

Solution Explanation:

- The function **SAMEPERIODLASTYEAR()** shifts the selected period (March 2022) back by one year to return the equivalent period in the previous year (March 2021).
- **CALCULATE()** is used to modify the context of the calculation to apply the time intelligence filter.
- **PREVIOUSYEAR()** (Option A) would return the total sales for the entire previous year, not just March.
- **TOTALYTD()** (Option B) calculates year-to-date values and is not suitable for this case.
- **SUM(Sales[Sales])** (Option D) simply sums all sales without applying any date filtering.
- Hence, Option **C** is the correct choice for retrieving sales data for the same period in the previous year.

Question 87:

Question:

You use **Power BI Desktop** to load data from a **Microsoft SQL Server** database.

While waiting for the data to load, you receive the following error:
ERROR [08001] TIMEOUT EXPIRED

You need to resolve the error.

Which two ways can achieve the goal? *(Each correct selection is worth one point.)*

Options:

A. Reduce the number of rows and columns returned by each query.
B. Split long-running queries into subsets of columns and use Power Query to merge the queries.
C. Use Power Query to combine long-running queries into one query.
D. Disable query folding on long-running queries.

Answer:

✅ **A. Reduce the number of rows and columns returned by each query.**
✅ **B. Split long-running queries into subsets of columns and use Power Query to merge the queries.**

Solution Explanation:

- **Option A:** Reducing the number of rows and columns in queries improves performance by decreasing data transfer size, reducing the likelihood of a timeout.
- **Option B:** Breaking large queries into smaller subsets makes data retrieval more efficient and avoids excessive query execution time. Power Query can then merge the results after loading.
- **Option C:** Combining multiple queries into one can **increase** query complexity and execution time, making the timeout issue worse.
- **Option D: Disabling query folding** can result in fetching all data before transformations are applied, potentially worsening performance instead of solving the timeout issue.
- Hence, **Options A and B** are the best solutions for resolving the timeout error.

Question 88:

Question:

From **Power Query Editor**, you profile the data shown in the exhibit.

The **IoT GUID** and **IoT ID** columns are unique to each row in the query.

You need to analyze **IoT events by the hour and day of the year**. The solution must improve dataset performance.

Solution: You split the **IoT DateTime** column into a column named **Date** and a column named **Time**.

Does this meet the goal?

Options:

A. Yes
B. No

Answer:

✅ **A. Yes**

Solution Explanation:

- **Splitting the DateTime column** into separate **Date** and **Time** columns improves performance because each new column has **fewer unique values** than the original DateTime column.
- **Why does this improve performance?** Power BI's **VertiPaq engine** compresses data better when there are fewer unique values.
- The **Date column** will have at most **365 unique values per year**, while the **Time column** will have at most **24 unique values** (if storing only hours).
- **Reducing cardinality (number of unique values)** helps in improving aggregation and filtering efficiency.
- Even though this process **adds an extra column**, the overall **compression and query performance** improve significantly.
- **Therefore, splitting the DateTime column does meet the goal, making the correct answer A (Yes).**

Question 89:

Question:

You have a **Microsoft Power BI** report with a **PBIX file size of 550 MB**. The report is accessed using an **App workspace in shared capacity** on **PowerBI.com**.

- The report uses an **imported dataset** containing **one fact table** with **12 million rows**.
- The dataset is **refreshed twice a day** at **08:00 AM** and **05:00 PM**.
- The report consists of **one page with 15 AppSource visuals and 10 default visuals**.
- **Users report that the visuals load slowly** when interacting with the report.

You need to recommend a solution to **improve performance**.

Options:

A. Change any DAX measures to use iterator functions.
B. Remove unused columns from tables in the data model.

C. Replace the default visuals with AppSource visuals.
D. Increase the number of times that the dataset is refreshed.

Answer:

✅ **B. Remove unused columns from tables in the data model.**

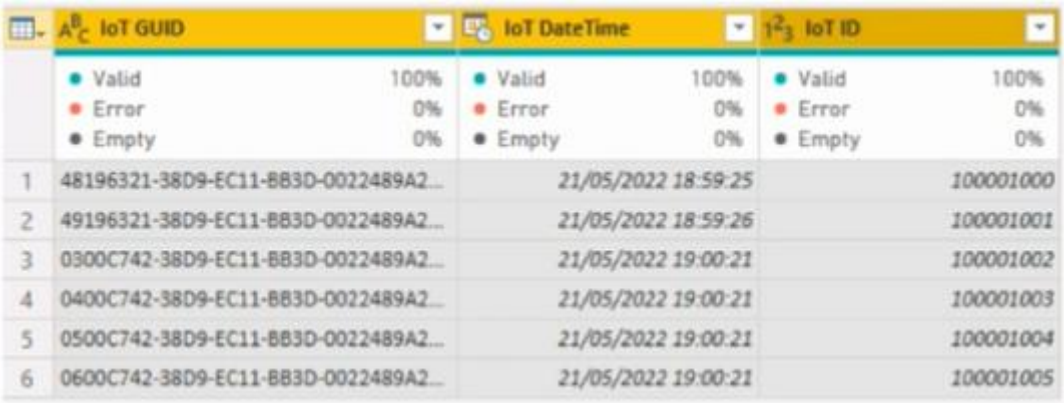
Solution Explanation:

- **Large dataset size (550 MB PBIX file)** suggests that excessive columns may be increasing **memory consumption** and **query execution time**.
- **Reducing the number of columns** in the fact table **lowers the data model size**, improves **compression efficiency**, and enhances **report performance**.
- **Option A (Using iterator functions)** often decreases performance as iterator functions (SUMX, FILTER, etc.) process row by row instead of using optimized columnar operations.
- **Option C (Replacing default visuals with AppSource visuals)** is **not recommended** since AppSource visuals are often **slower** due to extra processing.
- **Option D (Increasing refresh frequency)** does **not improve report interaction speed** since slow visuals are due to the **data model size**, not refresh rate.

Question 90:

Question:

From Power Query Editor, you profile the data with the following details:



	IoT GUID		IoT DateTime		IoT ID	
	Valid	100%	Valid	100%	Valid	100%
	Error	0%	Error	0%	Error	0%
	Empty	0%	Empty	0%	Empty	0%
1	48196321-38D9-EC11-BB3D-0022489A2...		21/05/2022 18:59:25		100001000	
2	49196321-38D9-EC11-BB3D-0022489A2...		21/05/2022 18:59:26		100001001	
3	0300C742-38D9-EC11-BB3D-0022489A2...		21/05/2022 19:00:21		100001002	
4	0400C742-38D9-EC11-BB3D-0022489A2...		21/05/2022 19:00:21		100001003	
5	0500C742-38D9-EC11-BB3D-0022489A2...		21/05/2022 19:00:21		100001004	
6	0600C742-38D9-EC11-BB3D-0022489A2...		21/05/2022 19:00:21		100001005	

- **The IoT GUID and IoT ID columns are unique to each row.**
- **You need to analyze IoT events by the hour and day of the year.**
- **The solution must improve dataset performance.**

Proposed Solution:

You change the IoT DateTime column to the Date data type.

Options:

A. Yes
B. No

Answer:

❌ **B. No**

Solution Explanation:

- **Changing the IoT DateTime column to Date removes the time component, making it impossible to analyze IoT events by the hour.**
- **Instead, the recommended approach is to split the DateTime column into separate Date and Time columns.**
- **Date-only data type does not allow hourly-level analysis, so a separate Time column will help in performance optimization while allowing analysis at an hourly level.**
- **Keeping DateTime as it is would lead to higher cardinality, increasing the dataset size and reducing performance.**

Correct Approach:

✅ **Split the DateTime column into a Date column and a Time column to improve dataset performance while enabling analysis by hour and day.**

Question 91:

Question:

From Power Query Editor, you profile the data with the following details:

- **The IoT GUID and IoT ID columns are unique to each row.**
- **You need to analyze IoT events by the hour and day of the year.**
- **The solution must improve dataset performance.**

Proposed Solution:

You remove the IoT GUID column and retain the IoT ID column.

Options:

A. Yes
B. No

Answer:

✅ **A. Yes**

Solution Explanation:

- **Both IoT GUID and IoT ID are unique, meaning either can serve as a primary key.**

- **Removing the GUID column improves performance by reducing memory usage, especially if GUID is a 16-byte binary data type, whereas IoT ID is likely a 4-byte integer.**
- **Keeping IoT ID allows for the required analysis of IoT events by the hour and day of the year without affecting data integrity.**
- **Removing unnecessary columns is a best practice in Power BI for better compression and faster query execution.**

Question 92:

You have a **Power BI data model** with two tables:

- **Products**
- **Sales**
  A **one-to-many relationship** exists between these tables.

You have a **report-level filter** applied to **Products**.

You need to create a **measure** that returns the **percent of total sales for each product**, ensuring that the measure respects the **report-level filter** when calculating the total.

DAX Function

- ALL
- ALLSELECTED
- CALCULATE
- FILTER
- SELECTEDVALUE

Answer Area

```
Percent of Product Sales =
VAR ProductSales = SUM ('Sales'[Sales])
VAR AllSales =
[          ](SUM('Sales'[Sales]),[          ]('Products'[Product]))
RETURN
DIVIDE (ProductSales, AllSales)
```

Answer:

✅ **A. Calculate**

✅ **A. ALLSELECTED**

Why ALLSELECTED()?

- **ALLSELECTED(Products)** removes **row-level filters** but **preserves report-level filters**, ensuring the total respects the **filters applied in the report**.

Question 93:

You have a **Power BI data model** analyzing **product sales over time** with the following conditions:

Table name	Column name	Data type
Product	Product ID	Whole number
	Product Name	Text
	Product Category	Text
Sales	Product ID	Whole number
	Order Date	Date
	Ship Date	Date
	Delivered Date	Date
	Invoice Number	Whole number
	Quantity	Whole number
	Sales Amount	Decimal number

- **One-to-many relationship** exists between tables.
- **Auto date/time** option is **enabled**.

You need to **reduce the size of the data model** while maintaining the ability to analyze **product sales by month and quarter**.

Options:

A. **Create a relationship between the Date table and the Sales table.**
B. **Disable the auto date/time option.** ✅
C. **Create a Date table and select Mark as Date Table.** ✅
D. **Disable the load on the Date table.**
E. **Remove the relationship between the Product table and the Sales table.**

Answer:

✅ **B. Disable the auto date/time option.**
✅ **C. Create a Date table and select Mark as Date Table.**

Explanation:

1. **Disabling Auto Date/Time (B)**
    - **Auto date/time** creates an **internal hidden table** for every date column, which increases the **size of the data model** significantly.
    - **Disabling** this option **reduces memory usage** and **improves performance**.
2. **Creating a Custom Date Table & Marking it as Date Table (C)**
    - Instead of relying on **auto date/time**, creating a **custom Date table** allows **better control** over time intelligence calculations.
    - Marking it as a **Date Table** ensures that **DAX time intelligence functions** work correctly.

Why Not Other Options?

❌ **A. Create a relationship between the Date table and Sales table:**

- While necessary for time-based analysis, it **does not** reduce the model size.

❌ **D. Disable load on the Date table:**

- This would **remove** the Date table from the model, **breaking** time-based analysis.

❌ **E. Remove the relationship between Product and Sales:**

- This would **break product-wise sales analysis**, which is not required.

Question 94:

You have a **Power BI report** with the following conditions:

- **PBIX file size: 550 MB**
- **Fact table rows: 12 million**
- **Scheduled refresh: Twice a day (08:00 & 17:00)**
- **Contains: 15 AppSource visuals** and **10 default visuals**
- **Users report slow performance when interacting with visuals**

You need to **improve the performance of the report**.

Options:

A. **Implement row-level security (RLS).**
B. **Remove unused columns from tables in the data model.** ✅
C. **Replace the default visuals with AppSource visuals.**
D. **Enable visual interactions.**

Answer:

✅ **B. Remove unused columns from tables in the data model.**

Explanation:

1. **Reducing Data Model Size (B - Correct)**
    - **Removing unused columns reduces memory usage** and improves performance.
    - A smaller model size results in **faster queries** and **faster visual rendering**.
2. **Why Not Other Options?**
   ❌ **A. Implement Row-Level Security (RLS):**
    - RLS restricts access **but does not** directly improve report performance.
    - In fact, **it can slow down performance** due to added query filtering.

❌ **C. Replace default visuals with AppSource visuals:**

- AppSource visuals **often require additional processing**, making the report **even slower**.
- **Default visuals are optimized** for better performance.

❌ **D. Enable Visual Interactions:**

- **Visual interactions are enabled by default**, and modifying them **does not significantly impact performance**.
- If anything, excessive **cross-filtering can slow down performance**.

Question 95:

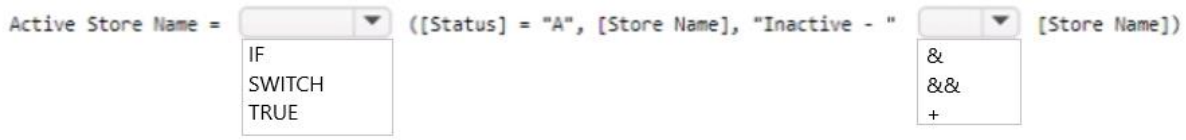
You have a Power BI data model that contains a table named **Stores**. The table has the following columns:

• **Store Name**
• **Open Date**
• **Status**
• **State**
• **City**

You need to create a calculated column named **Active Store Name** that meets the following requirements:

- When the value of the **Status** column is **“A”**, the value in the **Store Name** column must be returned.
- When the value of the **Status** column is NOT **“A”**, the value in the **Store Name** column must be prefixed with **"Inactive - "**.

How should you complete the DAX expression?



Answer:

✅ **A. IF**
✅ **B. &**

Explanation:

The IF function checks if the Status column equals "A".

If true, it returns the value from the Store Name column.

If false, it concatenates "Inactive - " with the Store Name using the & operator.

This approach ensures that the **Active Store Name** column meets the specified requirements.

Question 96:

You have a CSV file that contains user complaints. The file contains a column named **Logged**. The **Logged** column contains the date and time each complaint occurred 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?

Options:

A. Apply a transformation to extract the first 11 characters of the Logged column.
B. Add a conditional column that outputs **2018** if the Logged column starts with **2018** and set the data type of the new column to **Whole Number**.
C. Create a column by example that starts with **2018-12-31** and set the data type of the new column to **Date**.
D. Apply a transformation to extract the last 11 characters of the Logged column and set the data type of the new column.

Answer:

✅ **C. Create a column by example that starts with 2018-12-31 and set the data type of the new column to Date.**

Explanation:

- The **Logged** column contains both **date and time**, but to use a **built-in date hierarchy**, we need only the **date** part.
- **Column by Example** in Power Query allows users to manually type a sample value, and Power Query will infer the transformation logic.
- By typing **2018-12-31**, Power Query understands that we want to extract only the **date part (YYYY-MM-DD)** from the column.
- After extracting the date, changing the **data type to Date** ensures that it integrates properly with **Power BI's built-in date hierarchy** for time-based analysis.

Why Other Options Are Incorrect:

- **A:** Extracting the first 11 characters would work, but the **data type conversion to Date** is not explicitly mentioned.
- **B:** Extracting just the **year (2018)** and converting it to a whole number would not allow full **date hierarchy analysis** (year, quarter, month, day).
- **D:** Extracting the last 11 characters would include the **time (08:59)** and text **("at")**, making conversion to Date invalid.

Why Option A is Incorrect and Option C is Correct?

Option A: "Apply a transformation to extract the first 11 characters of the Logged column."

🔴 **Issue:**

- Extracting the **first 11 characters** (2018-12-31) will indeed give the **date portion**.
- However, the extracted value remains as **text** unless you manually change its **data type to Date**.
- The option **does not explicitly mention** changing the data type, so the value will still be treated as text.

🛑 **Power BI's built-in date hierarchy only works with Date-type columns, not text.**

Option C: "Create a column by example that starts with 2018-12-31 and set the data type of the new column to Date."

✅ **Why Correct?**

- **Column by Example** in Power Query allows users to provide a sample (2018-12-31), and Power Query **automatically extracts the correct part** of the column.
- It ensures that only the **date part (YYYY-MM-DD)** is extracted while **ignoring the time and unwanted text**.
- The option **explicitly mentions changing the data type to Date**, which is **necessary for using Power BI's built-in date hierarchy** (Year, Quarter, Month, Day).

Key Difference:

- **Option A extracts only text**, requiring an **extra manual step** to convert it to Date.
- **Option C extracts the correct format AND explicitly sets it as a Date column, making it immediately usable in Power BI.**

Question 97:

From Power Query Editor, you profile the data shown in the exhibit.

	IoT GUID	IoT DateTime	IoT ID
	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%
1	48196321-38D9-EC11-BB3D-0022489A2...	21/05/2022 18:59:25	100001000
2	49196321-38D9-EC11-BB3D-0022489A2...	21/05/2022 18:59:26	100001001
3	0300C742-38D9-EC11-BB3D-0022489A2...	21/05/2022 19:00:21	100001002
4	0400C742-38D9-EC11-BB3D-0022489A2...	21/05/2022 19:00:21	100001003
5	0500C742-38D9-EC11-BB3D-0022489A2...	21/05/2022 19:00:21	100001004
6	0600C742-38D9-EC11-BB3D-0022489A2...	21/05/2022 19:00:21	100001005

The **IoT GUID** and **IoT ID** columns are unique to each row in the query.

You need to analyze **IoT events by the hour and day of the year**. The solution must improve dataset performance.

Solution:

You create a **custom column** that **concatenates** the IoT GUID and IoT ID columns and then **delete the IoT GUID and IoT ID columns**.

Does this meet the goal?

Options:

A. **Yes**
B. **No**

Answer:

B. No ❌

Explanation:

1. **Goal Misalignment** 🚫
    - The goal is to analyze **IoT events by hour and day of the year**.
    - Concatenating **IoT GUID and IoT ID** does **not** help in this type of analysis.
    - Instead, you need to **extract the date and time components (hour, day, etc.)** from the timestamp column.
2. **Performance Issue** ⚠️
    - GUIDs are unique identifiers, and concatenating them **does not reduce dataset size** significantly.
    - This transformation **does not improve performance** since it adds a new computed column **without optimizing the existing data structure**.
3. **Correct Approach:** ✅
    - Instead of concatenating IoT GUID and IoT ID, you should:
        - Extract **Hour** and **Day of the Year** from the timestamp column.
        - Create separate **columns for date-based analysis**.
        - Remove unnecessary columns that do not contribute to the analysis.

Thus, the proposed solution **does not meet the goal**, and **Option B (No) is correct.** ✅

Question 98:

You have a **Power BI model** that contains a table named **Employee**. The table includes the following data:

Name	EmployeeID	ParentEmployeeID
David	100	100
Simon	101	100
Wenanta	102	100
Conrad	103	101
Priyish	104	103
Sunil	105	103
Pavel	106	102

- Each **employee has one manager**, as shown in the **ParentEmployeeID** column.
- All reporting paths **lead to the CEO** at the top of the organizational hierarchy.

You need to **create a calculated column** that **returns the count of levels from each employee to the CEO**.

Options:

A. PATHLENGTH(PATH(Employee[EmployeeID], Employee[ParentEmployeeID]))
B. PATHITEM(PATH(Employee[EmployeeID], Employee[ParentEmployeeID]),1,INTEGER)
C. PATHCONTAINS(PATH(Employee[EmployeeID], Employee[ParentEmployeeID]),1)
D. PATH(Employee[EmployeeID], Employee[ParentEmployeeID])

Answer:

✅ **A. PATHLENGTH(PATH(Employee[EmployeeID], Employee[ParentEmployeeID]))**

Explanation:

1. **Understanding the Hierarchy Structure** 🏢
    - The **EmployeeID** column identifies each employee.
    - The **ParentEmployeeID** column identifies the **manager** of each employee.
    - The **CEO** is at the top, meaning their **ParentEmployeeID is blank or null**.
2. **Using PATH(Employee[EmployeeID], Employee[ParentEmployeeID])** 🔗
    - The **PATH function** returns a **text string** representing the hierarchy path from the employee to the CEO.
    - Example: If **Employee 5** reports to **Employee 3**, and **Employee 3** reports to the **CEO (Employee 1)**, the **PATH output** would be:
    - 5 | 3 | 1
3. **Using PATHLENGTH(PATH(...)) to Count Levels** 🔢

- PATHLENGTH counts the **number of levels** in the hierarchy.
- If an employee has a hierarchy path like "5 | 3 | 1", the PATHLENGTH will return 3, meaning they are **3 levels away from the CEO**.

Question 99:

You have a **Microsoft Power BI** report with the following characteristics:

- The **PBIX file size is 550 MB**.
- The report is accessed via an **App workspace in shared capacity on PowerBI.com**.
- It uses an **imported dataset** with **one fact table containing 12 million rows**.
- The dataset **refreshes twice a day** at **08:00 and 17:00**.
- The **report contains 15 AppSource visuals and 10 default visuals**.
- **Users report slow performance** when **loading and interacting** with the report.

You need to **recommend a solution** to **improve performance**.

Options:

A. **Replace the default visuals with AppSource visuals.**
B. **Remove unused columns from tables in the data model.**
C. **Change the imported dataset to DirectQuery.**
D. **Increase the number of times that the dataset is refreshed.**

Answer:

✅ **B. Remove unused columns from tables in the data model.**

Explanation:

1. **Performance Issues in Power BI Reports**
   - The **PBIX file is 550 MB**, which is **too large for shared capacity**.
   - The **fact table has 12 million rows**, which impacts performance.
   - **Many visuals (15 AppSource + 10 default)** can slow down rendering.
2. **Why Removing Unused Columns Helps**
   - **Power BI loads all columns into memory**, even if they are unused.
   - **Reducing column count reduces memory usage and improves performance.**
   - **Columnar storage works better with fewer columns**, making calculations faster.

Question 100:

Question:

You have a **CSV file** containing **user complaints**. The file has a **Logged** column, which stores **date and time** in the following format:

📌 **2018-12-31 at 08:59**

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

Options:

A. **Change the data type of the Logged column to Date.**
B. **Split the Logged column by using "at" as the delimiter.**
C. **Add a conditional column that outputs 2018 if the Logged column starts with 2018 and set the data type of the new column to Whole Number.**
D. **Apply the Parse function from the Date transformations options to the Logged column.**

Answer:

✅ **B. Split the Logged column by using "at" as the delimiter.**

Explanation:

1. **Understanding the Problem:**
    - The **Logged column** contains both **date and time** in a **single column**.
    - To **use a built-in date hierarchy**, **Power BI requires a Date data type**.
    - The phrase **"at"** separates the **date and time**, which makes it easy to split.
2. **Why Option B is Correct:**
    - **Splitting the column using "at" as a delimiter** will result in two separate columns:
        - **One column with just the date (2018-12-31).**
        - **One column with just the time (08:59).**
    - After splitting, you can **change the date column's data type to Date**, allowing it to be **used with Power BI's built-in date hierarchy**.

Question 101:

You have the Power BI data model shown in the following exhibit.

You need to create a measure to count the number of product categories that had products sold during a selected period.

How should you complete the DAX expression? To answer, select the appropriate options in the answer area.

```
Product Categories Sold =
CALCULATE (
```

DISTINCTCOUNT('Product'[ProductCategory]),
COUNT('Product'[ProductCategory]),
DISTINCTCOUNT('Sales'[ProductID]),
SUM('Sales'[SalesQuantity]),

'Sales'
'Product'
'Product'[ProductCategory]
'Date'

)

Answer:

Verified Answer

```
Product Categories Sold =
CALCULATE (
```

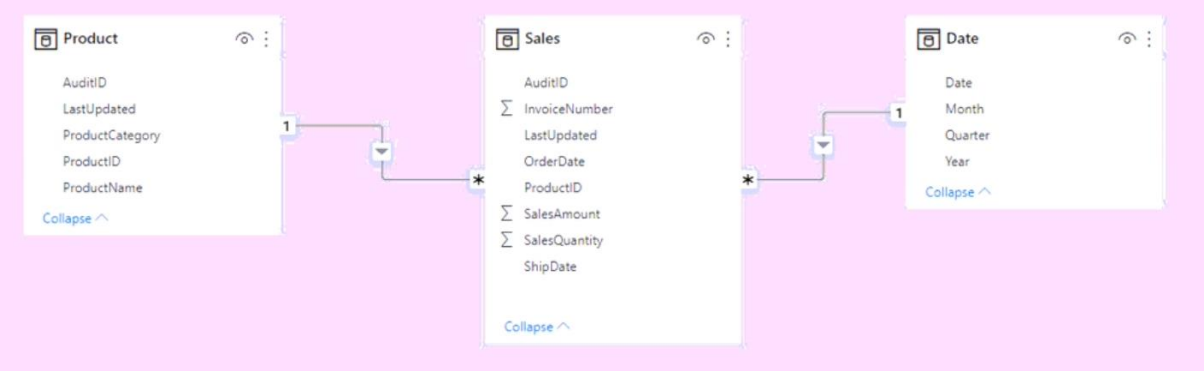
DISTINCTCOUNT('Product'[ProductCategory]),
COUNT('Product'[ProductCategory]),
DISTINCTCOUNT('Sales'[ProductID]),
SUM('Sales'[SalesQuantity]),

'Sales'
'Product'
'Product'[ProductCategory]
'Date'

)

Question 102:

You have the Power BI data model shown in the following exhibit.



The Sales table has the following columns.

Name	Data type	Sample value
ProductID	Whole number	1
InvoiceNumber	Whole number	100005
OrderDate	Date	2022-05-09
ShipDate	Date	2022-05-12
SalesAmount	Decimal number	1500.75
SalesQuantity	Whole number	3
LastUpdated	Date/time	5/22/2022 11:45:30 AM
AuditID	Whole number	123212

The data model must support the following analysis:

- Total sales by product by month in which the order was placed
- Quantities sold by product by day on which the order was placed
- Number of sales transactions by quarter in which the order was placed

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

Statements	Yes	No
Removing the LastUpdated column from the Sales table reduces the model size while still supporting the required analysis.	○	○
Removing the ProductID column from the Sales table reduces the model size while still supporting the required analysis.	○	○
Removing the ShipDate column from the Sales table reduces the model size while still supporting the required analysis.	○	○

Answer:

Statements	Yes	No
Removing the LastUpdated column from the Sales table reduces the model size while still supporting the required analysis.	○	○
Removing the ProductID column from the Sales table reduces the model size while still supporting the required analysis.	○	○
Removing the ShipDate column from the Sales table reduces the model size while still supporting the required analysis.	○	○

Explanation:

1.Removing the LastUpdated column from the Sales table reduces the model size while still supporting the required analysis.

- **Analysis:**
- The LastUpdated column is typically used for tracking record modifications and is not essential for sales metrics analysis.
- The required analysis focuses on sales metrics by **product** and **order dates**, so the absence of LastUpdated does not hinder the analysis.
- **Conclusion: Yes**, removing this column reduces the model size and does not impact the required analysis.

2. **Removing the ProductID column from the Sales table reduces the model size while still supporting the required analysis.**

- **Analysis:**
- The ProductID column is a key column that links the Sales table to the Product table, allowing analyses by **product**.
- Without ProductID, it would be impossible to perform analyses such as total sales, quantity sold, or number of transactions **by product**, which is part of the requirement.
- **Conclusion: No**, removing this column would break critical relationships and prevent the required analysis.

3. **Removing the ShipDate column from the Sales table reduces the model size while still supporting the required analysis.**

- **Analysis:**
- The analysis requirements specify that metrics are based on **order dates**, not shipping dates.
- Since ShipDate is not required for the specified analyses, it can be removed without affecting the ability to perform the required analysis.

Question 103:

You have a **Power BI report** with the following details:

- **PBIX file size: 550 MB**
- **Dataset: Imported mode** with **12 million rows** in one **fact table**
- **Refresh schedule: Twice a day (08:00 & 17:00)**
- **Report Layout:**
  - **One page**
  - **15 AppSource visuals**
  - **10 default visuals**
- **Problem:** Report is **slow to load visuals** when users access or interact with it.

Answer:

✅ **D. Remove unused columns from tables in the data model.**

Explanation:

1. **Understanding the Problem:**
   - **Large dataset (12 million rows)** → Increases **memory usage and processing time**.
   - **Imported mode (550 MB PBIX file)** → Data is **loaded into memory**, affecting performance.
   - **Too many visuals (15 AppSource + 10 default)** → Increases the time required for rendering and calculations.
2. **Why Option D is Correct (Remove Unused Columns)?**
   - **Every column in the model consumes memory** in Power BI's **VertiPaq engine**.
   - **Unused columns increase file size and slow down query execution.**
   - **Removing unnecessary columns reduces memory usage, improves performance**, and speeds up interactions.
   - **Smaller data models load faster** and perform calculations more efficiently.