



Free Questions for QSDA2024 by dumpshq

Shared by Baldwin on 10-09-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Exhibit.

OrderID	Product	Quantity	SupplierID
1	9	45	1
2	3	11	2
3	5	47	3
4	4	79	1
5	4	79	1
6	7	37	3
7	1	4	3
8	2	77	2
9	4	79	1
10	1	4	3

SupplierID	SupplierName	City
1	ABC	London
1	ABC	New York
2	XYZ	London
2	XYZ	Paris
3	QWERTY	Amsterdam
3	QWERTY	Barcelona

Refer to the exhibits.

The Orders table contains a list of orders and associated details. A data architect needs to replace the SupplierID with the SupplierName using the second table as the source.

The output must be a single table.

Which script should the data architect use?

A)

```
SuppliersMap:
MAPPING LOAD SupplierName, SupplierID;
SQL SELECT SupplierID, SupplierName FROM Suppliers;
LOAD *, APPLYMAP('SupplierName', SupplierID) AS SupplierName RESIDENT Orders;
DROP TABLE Orders;
```

B)

```
LEFT JOIN(Products)
LOAD SupplierID,
SupplierName;
SQL SELECT SupplierID, SupplierName FROM Suppliers;
```

C)

```
LEFT JOIN(Products)
LOAD DISTINCT SupplierID,
SupplierName;
SQL SELECT SupplierID, SupplierName FROM Suppliers;
```

D)

```
SuppliersNames:  
MAPPING LOAD ID, SupplierName as Name;  
SQL SELECT SupplierID as ID, SupplierName FROM Suppliers;  
LOAD *, APPLYMAP('SupplierNames', SupplierID) AS SupplierName RESIDENT Orders;  
DROP TABLE Orders;
```

Options:

- A- Option A
- B- Option B
- C- Option C
- D- Option D

Answer:

C

Explanation:

In this scenario, the data architect needs to replace the SupplierID in the Orders table with the corresponding SupplierName from the Suppliers table, and the desired output should be a single table that includes all the order details along with the SupplierName instead of the SupplierID.

Analyzing the Options:

Option A:

Uses a MAPPING LOAD followed by an APPLYMAP to replace SupplierID with SupplierName in the Orders table. However, the table is dropped afterward, which means it won't produce the required output.

The MAPPING LOAD approach is generally used to map values but is not necessary in this context as we are combining data from two tables directly.

Option B:

This option attempts to LEFT JOIN the Products table with the Suppliers table, but it does not directly address replacing SupplierID with SupplierName in the Orders table.

Additionally, it does not remove the SupplierID after the join, which is essential for the correct output.

Option C:

This option uses a LEFT JOIN with the DISTINCT keyword on the SupplierID field to avoid duplicates. The SupplierName is correctly joined to the Orders table, replacing the SupplierID.

This approach is the most appropriate because it results in a single table containing all order details with the SupplierName instead of the SupplierID.

Option D:

Similar to Option A, but it also introduces an unnecessary renaming step with MAPPING LOAD. It's redundant and does not improve the solution over Option C.

Correct Script Choice:

Option C is the correct script because:

It ensures that SupplierName replaces SupplierID in the Orders table using a LEFT JOIN.

The DISTINCT keyword is applied to the SupplierID field to prevent duplicate rows during the join.

The result is a single table containing the required information with SupplierName in place of SupplierID.

Qlik Sense Join Operations: Using the correct JOIN type and ensuring proper deduplication (with DISTINCT if necessary) is key to merging tables in Qlik Sense.

Question 2

Question Type: MultipleChoice

A data architect needs to load Table_A from an Excel file and sort the data by Reld_2.

Which script should the data architect use?

A)

Temp:

```
LOAD
Field_1,
Field_2,
Field_3
FROM [lib://Data/Table_A.xlsx]
(coxml, embedded labels, table is Sheet1);
```

Table_A:

```
LOAD *
resident Temp Order by Field_2 asc;

drop Table Temp;
```

B)

Table_A:

```
LOAD
Field_1,
Field_2,
Field_3
FROM [lib://Data/Table_A.xlsx]
(coxml, embedded labels, table is Sheet1)
Order by Field_2 asc;
```

C)

Temp:

```
LOAD
Field_1,
Field_2,
Field_3
FROM [lib://Data/Table_A.xlsx]
(ooxml, embedded labels, table is Sheet1);
```

NoConcatenate

Table_A:

```
LOAD *
resident Temp Order by Field_2 asc;
drop Table Temp;
```

D)

Table_A:

```
LOAD *
Order by Field_2 asc;
```

```
LOAD
Field_1,
Field_2,
Field_3
FROM [lib://Data/Table_A.xlsx]
(ooxml, embedded labels, table is Sheet1);
```


Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

A

Explanation:

In this scenario, the data architect needs to load Table_A from an Excel file and ensure that the data is sorted by Field_2. The key here is to correctly load and sort the data in the script.

Understanding the Options:

Option A:

First, it loads the data into a temporary table (Temp) from the Excel file.

Then, it loads the data from the temporary table (Temp) into Table_A, using the ORDER BY Field_2 ASC clause to sort the data by Field_2.

Finally, it drops the temporary table (Temp), leaving the sorted data in Table_A.

Option B:

Directly loads the data from the Excel file into Table_A and applies the ORDER BY Field_2 ASC clause in the same step.

However, the ORDER BY clause in a direct load from an external source like Excel might not work as expected because Qlik Sense does not support ORDER BY when loading directly from a file.

Option C:

Similar to Option A but uses the NoConcatenate keyword to prevent concatenation, which is unnecessary since Temp and Table_A have different names.

While this script works, the NoConcatenate keyword is redundant in this context.

Option D:

The ORDER BY Field_2 ASC is placed before the LOAD statement, which is not a correct usage in Qlik Sense script syntax.

Correct Script Choice:

Option A is the correct script because it correctly sorts the data after loading it into a temporary table and then loads the sorted data into Table_A. This method ensures that the data is sorted by Field_2 and avoids any issues related to sorting during the initial data load.

Qlik Sense Scripting Best Practices: When sorting data in Qlik Sense, the correct approach is to use a RESIDENT LOAD with an ORDER BY clause after loading the data into a temporary table.

Question 3

Question Type: MultipleChoice

Exhibit.

Orders:		
OrderID	LineNo	OrderDate
668	1	2019-06-01
668	2	2019-06-01
669	1	2019-06-02

Shipments:		
OrderID	LineNo	ShipmentDate
668	1	2019-06-01
669	1	2019-06-03
668	2	2019-06-02

Refer to the exhibit.

A data architect is loading the tables and a synthetic key is generated.

How should the data architect resolve the synthetic key?

Options:

- A-** Remove the LineNo field from Shipments and use the AutoNumber function on the OrderID field
- B-** Remove the LineNo field from both tables and use the AutoNumber function on the OrderID field
- C-** Create a composite key using OrderID and LineNo
- D-** Create a composite key using OrderID and LineNo, and remove OrderID and LineNo from Shipments

Answer:

C

Explanation:

In this scenario, the data architect is loading two tables, Orders and Shipments, into Qlik Sense, and a synthetic key is being generated due to the presence of shared fields (OrderID and LineNo) between these tables.

Understanding the Issue:

Synthetic Keys: Qlik Sense automatically creates synthetic keys when two or more tables share multiple fields with the same names. While synthetic keys aren't necessarily problematic, they can sometimes lead to incorrect or unexpected data associations and should be resolved when possible to maintain clarity and control over the data model.

The tables Orders and Shipments share the fields OrderID and LineNo. In this context, these fields together uniquely identify each record, so they are both necessary for accurate data linkage.

Correct Resolution Approach:

Option C: Create a composite key using OrderID and LineNo is the best approach.

Here's why:

Composite Key Creation:

By creating a composite key that combines OrderID and LineNo (e.g., OrderID & '-' & LineNo), you ensure that each line in the orders and shipments tables is uniquely identified. This composite key will accurately link the related records from the Orders and Shipments tables.

Avoiding Synthetic Keys:

By manually creating this composite key, you eliminate the need for Qlik Sense to generate a synthetic key, thereby simplifying the data model and ensuring that data associations are clear and controlled.

Retaining Both Fields:

This approach allows you to keep both OrderID and LineNo as separate fields in your tables if needed for other analyses or reporting purposes, while using the composite key for linking the tables.

Qlik Sense Data Modeling Best Practices: When dealing with multiple fields that are used together to uniquely identify records, it is recommended to create composite keys rather than relying on Qlik Sense's synthetic keys for clarity and better control.

Question 4

Question Type: MultipleChoice

Exhibit.

The diagram illustrates a data transformation process. It shows two tables: 'Input' and 'Output'. A blue arrow points from the 'Input' table to the 'Output' table, indicating the direction of data flow. Red circles highlight the 'Attribute' column in the 'Input' table and the 'Color', 'Diameter', 'Height', 'Length', 'Weight', and 'Width' columns in the 'Output' table.

Input		
Key	Attribute	Value
Ball	Color	Red
Ball	Diameter	4
Box	Height	2
Box	Length	6
Box	Weight	1
Box	Width	4
Dog	Color	Black
Dog	Weight	10
Rod	Diameter	1
Rod	Length	10

Output						
Key	Color	Diameter	Height	Length	Weight	Width
Ball	Red	-	-	-	-	-
Box	-	-	2	6	1	4
Dog	Black	-	-	-	10	-
-	-	-	1	10	-	-

Refer to the exhibit.

A data architect wants to transform the input data set to the output data set. Which prefix to the Qlik Sense LOAD command should the data architect use?

Options:

A- Hierarchy Be longsTo

B- Peek

C- Generic

D- PivotTable

Answer:

C

Explanation:

In this scenario, the data architect wants to transform the input dataset, which is in a key-value pair structure, into a table where each attribute becomes a column with its corresponding value under the relevant key.

Understanding the Requirement:

The input data consists of three fields: Key, Attribute, and Value.

The desired output structure has the Key as a primary identifier, and the Attributes (like Color, Diameter, Height, etc.) are spread across the columns, with corresponding values filled in each row.

Best Method to Achieve this Transformation:

The appropriate method to convert key-value pairs into a structured table where each unique attribute becomes a separate column is the Generic Load function in Qlik Sense.

Why Generic?

Generic Load is specifically designed for situations where data is stored in a key-value format (like the one provided) and needs to be converted into a more traditional tabular format, with attributes as columns.

It creates a separate table for each combination of Key and Attribute, effectively 'pivoting' the attribute values into columns in the output table.

How it Works:

When applying a GENERIC LOAD to the input dataset, Qlik Sense will generate multiple tables, one for each Attribute. However, in the final data model, Qlik Sense automatically joins these tables by the Key field, effectively producing the desired output structure.

Qlik Sense Documentation on Generic Load: The documentation outlines how to use the Generic Load to handle key-value pairs and pivot them into a more traditional table format.

Question 5

Question Type: MultipleChoice

Exhibit.


```
Section Access;  
SecurityTable:  
Load * INLINE [  
ACCESS, USERID, LINK, OMIT  
ADMIN, ABC\QSERVICE, LEVEL  
USER, ABC\EFN, *,  
USER, ABC\JCS, *,  
USER, ABC\MMD, NA,  
USER, ABC\MMD, SA,  
USER, ABC\HDD, EMEA,  
USER, ABC\PPP, * , LEVEL  
];
```

The Section Access security table for an app is shown. User ABC\PPP opens a Qlik Sense app with a table using the field called LEVEL on one of the table columns.

Which is the result?

Options:

- A-** The table is removed from the user interface.
- B-** The user gets an 'Incomplete visualization' error.
- C-** The user gets a 'Field not found' error.
- D-** The table is displayed without the LEVEL column.

Answer:

D

Explanation:

In this scenario, the Section Access security table controls user access to data within the Qlik Sense app. The user in question, ABC\PPP, has a specific entry in the security table that determines their access rights to the LEVEL field.

Understanding Section Access:

Section Access is used to enforce security by restricting access to certain data based on the user's credentials.

In the security table provided, the USER role for ABC\PPP is set to have access to all data (* in the LINK field), but the OMIT field is set to LEVEL. The OMIT field in Section Access specifies fields that should be omitted from the user's view.

Outcome:

Since the OMIT field for user ABC\PPP is set to LEVEL, this user will not have access to the LEVEL field in the Qlik Sense application.

Option D: The table is displayed without the LEVEL column is the correct outcome.

Qlik Sense Security and Section Access Documentation: The OMIT functionality in Section Access is specifically designed to remove fields from the user's access, ensuring that sensitive or unnecessary data is not exposed.

Question 6

Question Type: MultipleChoice

A Chief Information Officer has hired Qlik to enhance the organization's inventory analytics. In the initial meeting, the client's focus was determined to be forecasting inventory levels.

Which stakeholder should be consulted first when gathering requirements?

Options:

- A- Product Buyer
- B- SQL Developer
- C- Vice President of Marketing
- D- Chief Information Officer

Answer:

A

Explanation:

In this scenario, the focus of the project is to enhance inventory analytics, specifically targeting forecasting inventory levels. The primary goal is to understand the factors influencing inventory management and to build a model that helps in predicting future inventory needs.

Option A: Product Buyer is the correct stakeholder to consult first.

Here's why:

Direct Involvement in Inventory Management:

The Product Buyer is typically responsible for making decisions related to purchasing and maintaining inventory levels. They have a deep understanding of the factors that influence inventory needs, such as lead times, supplier reliability, demand forecasting, and purchasing cycles.

Knowledge of Inventory Requirements:

Since the project's primary focus is forecasting inventory levels, the Product Buyer will provide crucial insights into the variables that affect inventory and the data needed for accurate forecasting. They can guide what historical data is essential and what external factors might need to be considered in the forecasting model.

Alignment with Business Objectives:

By consulting the Product Buyer, the project can ensure that the inventory forecasting models align with the company's inventory management objectives, avoiding overstocking or understocking, and thus optimizing costs.

Qlik Project Management Best Practices: In analytics projects, particularly those focused on specific operational areas like inventory management, consulting the stakeholders who are closest to the operational data and decision-making processes ensures that the solution will be relevant and effective.

Question 7

Question Type: MultipleChoice

exhibit.

```
Data_02:  
LOAD  
  *  
  Resident  
  Data  
  Where  
    [Date] > '$(vMaxDate)'  
;
```

A data architect is validating that the script section, as shown in the exhibit, is working properly. They need to stop the script with a preview of the value used with the Load statement.

Where should the data architect put the debugger breakpoint?

A)

```
Data_02:  
LOAD  
  *  
  Resident  
  Data  
Where  
  [Date] > '$(vMaxDate)'  
;
```

B)

```
Data_02:  
LOAD  
  *  
  Resident  
  Data  
Where  
  [Date] > '$(vMaxDate)'  
;
```

C)

```
Data_02:  
LOAD  
  *  
  Resident  
  Data  
Where  
  [Date] > '$(vMaxDate)'  
;
```

D)

```
Data_02:  
LOAD  
*  
Resident  
Data  
where  
[Date] > '$(vMaxDate)'  
;
```

Options:

- A- Option A
- B- Option B
- C- Option C
- D- Option D

Answer:

A

Explanation:

In this scenario, the data architect needs to validate the script and specifically ensure that the vMaxDate variable is being correctly utilized in the LOAD statement. The goal is to stop the script execution at a point where the variable's value can be previewed.

Understanding the Options:

Option A places the breakpoint just after the assignment of the variable vMaxDate in the Where clause but before any data is loaded.

Option B, C, and D represent placements of the breakpoint after the LOAD statement begins processing the Resident table, which means that the variable vMaxDate would have already been utilized.

Correct Breakpoint Placement:

Option A is the correct choice because placing the breakpoint at this point allows you to preview the value of vMaxDate right before it is used in the Where clause. This placement ensures that the script execution halts before loading the data, allowing you to validate whether vMaxDate is correctly defined and whether it correctly filters the data based on the [Date] field.

If the breakpoint were placed after the LOAD statement (as in Options B, C, or D), the script would have already attempted to load the data, making it too late to inspect the variable's value before it's used.

Qlik Sense Debugging Best Practices: When debugging, it is crucial to set breakpoints before the execution of a critical operation where the values of variables or fields are used to ensure that they hold the expected data.

Question 8

Question Type: MultipleChoice

Exhibit.

Object	Attribute	Value
circle	color	red
circle	diameter	10
rectangle	color	black
rectangle	length	20
rectangle	width	10
square	color	peach
square	length	45

While performing a data load from the source shown, the data architect notices it is NOT appropriate for the required analysis.

The data architect runs the following script to resolve this issue:

```
Shapes:
GENERIC LOAD
Object,
"Attribute",
Value
FROM [lib://Data/products.xlsx]
(ooxml, embedded labels, table is Shapes);
```

How many tables will this script create?

Options:

A- 1

B- 3

C- 4

D- 6

Answer:

D

Explanation:

In this scenario, the data architect is using a GENERIC LOAD statement in the script to handle the data structure provided. A GENERIC LOAD is used in Qlik Sense when you have data in a key-value pair structure and you want to transform it into a more traditional table structure, where each attribute becomes a column.

Given the input data table with three columns (Object, Attribute, Value), and the attributes in the Attribute field being either color, diameter, length, or width, the GENERIC LOAD will create separate tables based on the combinations of Object and each Attribute.

Here's how the GENERIC LOAD works:

For each unique object (circle, rectangle, square), the GENERIC LOAD creates separate tables based on the distinct values of the Attribute field.

Each of these tables will contain two fields: Object and the specific attribute (e.g., color, diameter, length, width).

Breakdown:

Table for circle:

Fields: Object, color, diameter

Table for rectangle:

Fields: Object, color, length, width

Table for square:

Fields: Object, color, length

Each distinct attribute (color, diameter, length, width) and object combination generates a separate table.

Final Count of Tables:

The script will create 6 separate tables: one for each unique combination of Object and Attribute.

Qlik Sense Documentation on Generic Load: Generic loads are used to pivot key-value pair data structures into multiple tables, where each key (in this case, the Attribute field values) forms a new column in its own table.

To Get Premium Files for QSDA2024 Visit

<https://www.p2pexams.com/products/qsda2024>

For More Free Questions Visit

<https://www.p2pexams.com/qlik/pdf/qsda2024>

