Qlik QSDA2024 Practice Questions

Shared by Stanley on 21-08-2026

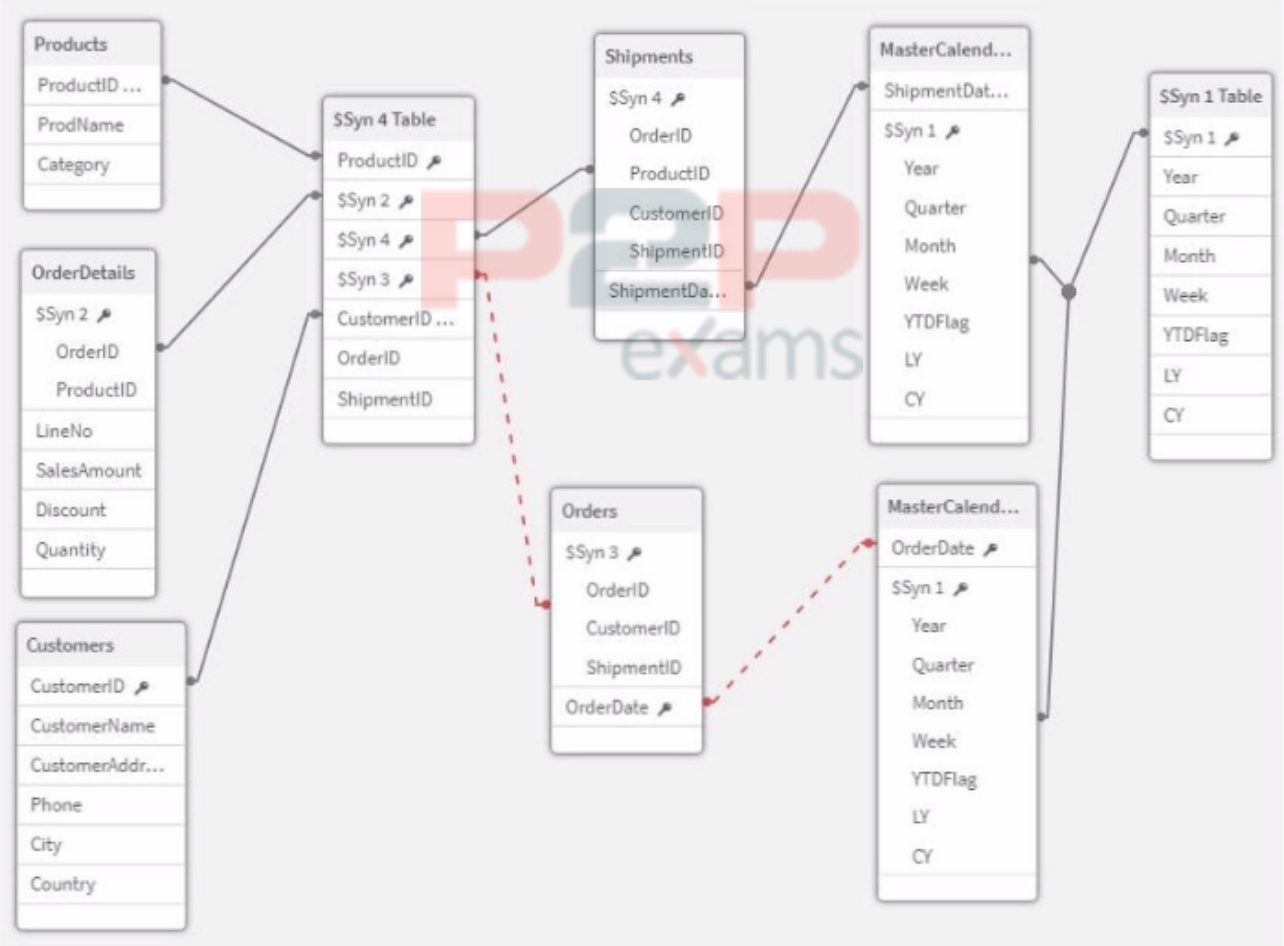
For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Exhibit.



Refer to the exhibit.

A data architect is working on a Qlik Sense app the business has created to analyze the company orders and shipments.

To understand the table structure, the business has given the following summary:

* Every order creates a unique orderlD and an order date in the Orders table

* An order can contain one or more order lines one for each product ID in the order details table

* Products In the order are shipped (shipment date) as soon as they are ready and can be shipped separately

* The dates need to be analyzed separately by Year, Month, and Quarter

The data architect realizes the data model has issues that must be fixed. Which steps should the data architect perform?

Options:

A- 1. Create a key with OrderlD and ProductID in the OrderDetails table and in the Shipments table
2. Delete the ShipmentID in the Orders table
3. Delete the ProductID and OrderlD in the Shipments table
4. Left join Orders and OrderDetails
5. Use Derive statement with the MasterCalendar table and apply the derive fields to OrderDate and ShipmentDate
B- 1. Create a key with OrderlD and ProductID In the OrderDetails table and in the Orders table
2. Delete the ShipmentID in the Shipments table
3. Delete the ProductID and OrderlD in the OrderDetails table
4. Concatenate Orders and OrderDetails
5. Create a link table using the MasterCalendar table and create a concatenated field between OrderDate and ShipmentDate
C- 1. Create a key with OrderlD and ProductID in the OrderDetails table and in the Shipments table
2. Delete the ShipmentID in the Orders table
3. Delete the ProductID and OrderlD In the Shipments table
4. Concatenate Orders and OrderDetails
5. Create a link table using the MasterCalendar table and create a concatenated field between OrderDate and ShipmentDate
D- 1. Create a key with OrderlD and ProductID in the OrderDetails table and in the Orders table
2. Delete the ShipmentID in the Shipments table
3. Delete the ProductID and OrderlD in the OrderDetails table
4. Left join Orders and OrderDetails
5. Use Derive statement with the MasterCalendar table and apply the derive fields to OrderDate and ShipmentDate

Answer:

C

Explanation:

In the given data model, there are several issues related to table relationships and key fields that need to be addressed to create a functional and optimized data model. Here's how each step in the chosen solution (Option C) resolves these issues:

Create a key with OrderID and ProductID in the OrderDetails table and in the Shipments table:

By creating a composite key with OrderID and ProductID, you uniquely identify each line item in both the OrderDetails and Shipments tables. This step is crucial for ensuring that each product within an order is correctly associated with its respective shipment.

Delete the ShipmentID in the Orders table:

The ShipmentID in the Orders table is redundant because the Shipments table already captures this information at a more granular level (i.e., at the product level). Removing ShipmentID avoids potential circular references or synthetic keys.

Delete the ProductID and OrderID in the Shipments table:

After creating the composite key in step 1, the individual ProductID and OrderID fields in the Shipments table are no longer necessary for joins. Removing them reduces redundancy and simplifies the table structure.

Concatenate Orders and OrderDetails:

Concatenating Orders and OrderDetails into a single table creates a unified table that contains all necessary order-related information. This helps in simplifying the model and avoiding issues related to managing separate but related tables.

Create a link table using the MasterCalendar table and create a concatenated field between OrderDate and ShipmentDate:

A link table is created to associate the combined table with the MasterCalendar. By creating a concatenated field that combines OrderDate and ShipmentDate, you ensure that both dates are properly linked to the calendar, allowing for accurate time-based analysis.

Question 2

Question Type: MultipleChoice

A company generates l GB of ticketing data daily. The data is stored in multiple tables. Business users need to see trends of tickets processed for the past 2 years. Users very rarely access the transaction-level data for a specific date. Only the past 2 years of data must be loaded, which is 720 GB of data.

Which method should a data architect use to meet these requirements?

Options:

A- Load only 2 years of data in an aggregated app and create a separate transaction app for occasional use
B- Load only 2 years of data and use best practices in scripting and visualization to calculate and display aggregated data
C- Load only aggregated data for 2 years and use On-Demand App Generation (ODAG) for transaction data
D- Load only aggregated data for 2 years and apply filters on a sheet for transaction data

Answer:

C

Explanation:

In this scenario, the company generates 1 GB of ticketing data daily, accumulating up to 720 GB over two years. Business users mainly require trend analysis for the past two years and rarely need to access the transaction-level data. The objective is to load only the necessary data while ensuring the system remains performant.

Option C is the optimal choice for the following reasons:

Efficiency in Data Handling:

By loading only aggregated data for the two years, the app remains lean, ensuring faster load times and better performance when users interact with the dashboard. Aggregated data is sufficient for analyzing trends, which is the primary use case mentioned.

On-Demand App Generation (ODAG):

ODAG is a feature in Qlik Sense designed for scenarios like this one. It allows users to generate a smaller, transaction-level dataset on demand. Since users rarely need to drill down into transaction-level data, ODAG is a perfect fit. It lets users load detailed data for specific dates only when needed, thus saving resources and keeping the main application lightweight.

Performance Optimization:

Loading only aggregated data ensures that the application is optimized for performance. Users can analyze trends without the overhead of transaction-level details, and when they need more detailed data, ODAG allows for targeted loading of that data.

Qlik Sense Best Practices: Using ODAG is recommended when dealing with large datasets where full transaction data isn't frequently needed but should still be accessible.

Qlik Documentation on ODAG: ODAG helps in maintaining a balance between performance and data availability by providing a method to load only the necessary details on demand.

Question 3

Question Type: MultipleChoice

A data architect wants reflect a value of the variable in the script log for tracking purposes. The variable is defined as:

```
Let vMaxDate = Date(Peek('MaxDate'),'DD/MMM/YYYY');
```

Which statement should be used to track the variable's value?

A)

```
Trace #### vMaxDate ####;
```

B)

```
Trace #### $(=vMaxDate) ####;
```

C)

```
Trace #### $(vMaxDate) ####;
```

D)

```
Trace #### =vMaxDate ####;
```

Options:

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

Answer:

B

Explanation:

In Qlik Sense, the TRACE statement is used to print custom messages to the script execution log. To output the value of a variable, particularly one that is dynamically assigned, the correct syntax must be used to ensure that the variable's value is evaluated and displayed correctly.

The variable vMaxDate is defined with the LET statement, which means it is evaluated immediately, and its value is stored.

When using the TRACE statement, to output the value of vMaxDate, you need to ensure the variable's value is expanded before being printed. This is done using the $() expansion syntax.

The correct syntax is TRACE #### $(vMaxDate) ####; which evaluates the variable vMaxDate and inserts its value into the log output.

Key Qlik Sense Data Architect Reference:

Variable Expansion: In Qlik Sense scripting, $(variable_name) is used to expand and insert the value of the variable into expressions or statements. This is crucial when you want to output or use the value stored in a variable.

TRACE Statement: The TRACE command is used to write messages to the script log. It is commonly used for debugging purposes to track the flow of script execution or to verify the values of variables during script execution.

Question 4

Question Type: MultipleChoice

Refer to the exhibit

A large transport company (Company A) acquires a smaller rival (Company B).

Company A has been using Qlik Sense tor 6 years to track revenue per ship journey. Ship journeys with no revenue (such as journeys to shipyards for repair) always show revenue of $0.

Company A wants to combine its data set with the data set of the acquired Company B. Company B's ship journey data shows $0 revenue in one of the following ways:

* A NULL value

* A value with one or more blank spaces (ASCII char code 32)

The data architect wants to conform the Company B data to the Company A standard, specifically regarding the use of an explicit $0 for journeys without revenue. Which script line should the data architect use?

A)

```
money ( replace(Revenue, chr(32), 0) ) AS [Revenue Conformed]
```

B)

```
if (len (trim(Revenue) ) < 1,="" money(0),="" revenue="" )="" as="" [revenue="" conformed]="" 1,="" money(0),="" revenue="" )="" as="" [revenue="" />
```

C)

```
money ( replace(Revenue, Null(), 0) ) AS [Revenue Conformed]
```

D)

```
replace(Revenue, chr(32), money(0) ) AS [Revenue Conformed]
```

Options:

A- Option A

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

Answer:

A

Explanation:

In this scenario, the data architect needs to conform the revenue data from Company B to match the data standard of Company A, where $0 is explicitly used to represent journeys without revenue.

Explanation of the Correct Script:

Option A: money(replace(Revenue, chr(32), 0)) AS [Revenue Conformed]

replace(Revenue, chr(32), 0): This part of the expression replaces any spaces (ASCII character code 32) in the Revenue field with 0.

money(...): This function formats the resulting value as currency. Since Company B may have either null values or spaces where 0 should be, this script ensures that any blanks are replaced with 0 and then formatted as currency.

Why Option A is Correct:

Handling Spaces: The replace() function is effective in replacing spaces with 0, conforming to Company A's standard of using $0 for non-revenue journeys.

Handling NULL Values: The money() function is used to ensure the final output is formatted as currency. However, it's important to note that NULL values are not directly handled by the replace() function, which is why it is applied before money() to deal with spaces.

Question 5

Question Type: MultipleChoice

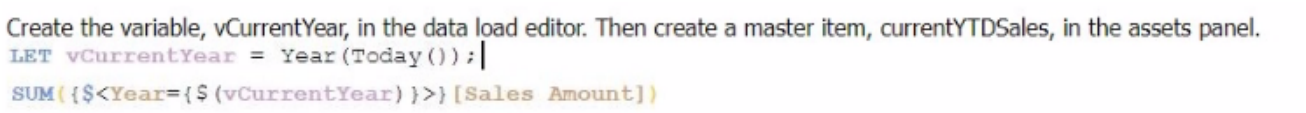
Sales managers need to see an overview of historical performance and highlight the current year's metrics. The app has the following requirements:

* Display the current year's total sales

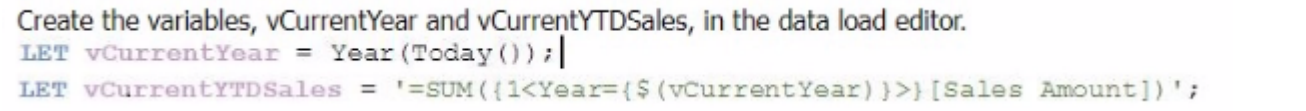
* Total sales displayed must respond to the user's selections

Which variables should a data architect create to meet these requirements?

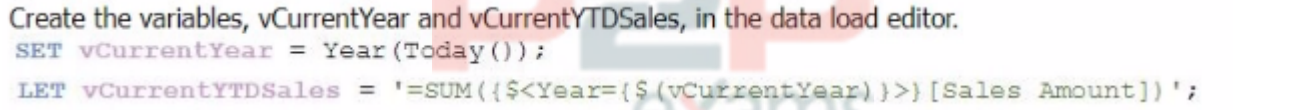
A)



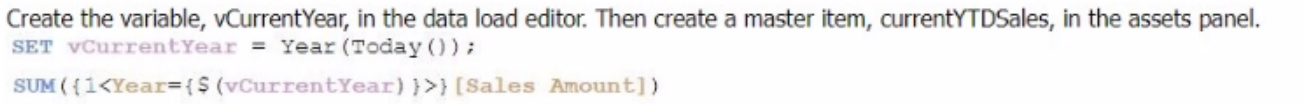
B)



C)



D)



Options:

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

Answer:

C

Explanation:

To meet the requirements of displaying the current year's total sales in a way that responds to user selections, the correct approach involves using both SET and LET statements to define the necessary variables in the data load editor.

Explanation of Option C:

SET vCurrentYear = Year(Today());

The SET statement is used here to assign the current year to the variable vCurrentYear. The SET statement treats the variable as a text string without evaluation. This is appropriate for a variable that will be used as part of an expression, ensuring the correct year is dynamically set

based on the current date.

LET vCurrentYTDSales = '=SUM({$<Year={'$(vCurrentYear)'}>} [Sales Amount])';

The LET statement is used here to assign an evaluated expression to the variable vCurrentYTDSales. This expression calculates the Year-to-Date (YTD) sales for the current year by filtering the Year field to match vCurrentYear. The LET statement ensures that the expression inside the variable is evaluated, meaning that when vCurrentYTDSales is called in a chart or KPI, it dynamically calculates the YTD sales based on the current year and any user selections.

Key Points:

Dynamic Year Calculation: Year(Today()) dynamically calculates the current year every time the script runs.

Responsive to Selections: The set analysis syntax {$<Year={'$(vCurrentYear)'}>} ensures that the sales totals respond to user selections while still focusing on the current year's data.

Appropriate Use of SET and LET: The combination of SET for storing the year and LET for storing the evaluated sum expression ensures that the variables are used effectively in the application.

Question 6

Question Type: MultipleChoice

Refer to the exhibit.

Sales	Customers	Employees
SaleID	CustID	EmployeeID
CustomerID	CustName	EmployeeName
Amount	Address	MgrID
SaleDate	City	
SalesPersonID	State	
RegionalAcctMgrID	Zip	

Refer to the exhibit.

A data architect needs to create a data model for a new app. Users must be able to see:

* Total sales for each customer

* Total sales for a given state

* Customers that have not had any sales

* Names of salesperson and regional account managers

* Total number of sales by date

Which steps should the data architect perform to meet these requirements?

Which steps should the data architect perform to meet these requirements?

Options:

A- 1. Use a Mapping Load for the Employees table
2. Load the Sales table and use ApplyMap to get the names for SalesPersonID and RegionalAcctMgrlD
3. Use a Left Join Load to add the customer details for the Sales table
B- 1. Load the Customers table and alias the CustID field as CustomerlD
2. Use a Mapping Load for the Employees table
3. Load the Sales table and use ApplyMap to get the names for SalesPersonID and RegionalAcctMgrlD
C- 1. Load the Sales table
2. Load the Customers table
3. Load the Employees table twice; name it and alias the EmployeelD field appropriately each time
D- 1. Load the Customers table and alias the CustID field as CustomerlD
2. Load the Employees table
3. Load the Sales table and alias the SalesPersonID and RegionalAcctMgrlD fields as EmployeelD

Answer:

C

Explanation:

In the provided scenario, the data architect needs to create a data model that supports various analyses, including total sales for each customer, total sales by state, identifying customers with no sales, and displaying the names of salespersons and regional account managers.

Here's why Option C is the correct choice:

Loading the Sales Table: The Sales table contains key information related to sales transactions, including SaleID, CustomerID, Amount, SaleDate, SalesPersonID, and RegionalAcctMgrID. This table must be loaded first as it will be central to the analysis.

Loading the Customers Table: The Customers table includes customer details such as CustID, CustName, Address, City, State, and Zip. Loading this table and linking it to the Sales table via the CustomerID field allows you to perform analyses such as total sales per customer and total sales by state. Importantly, loading the customers separately will also allow the identification of customers without any sales.

Loading the Employees Table Twice: The Employees table must be loaded twice because it is

used to look up two different roles in the sales process: the SalesPersonID and the RegionalAcctMgrID. When loading the table twice:

The first instance of the Employees table will be used to map the SalesPersonID to EmployeeName.

The second instance will be used to map the RegionalAcctMgrID to EmployeeName.

Aliasing the EmployeeID field appropriately in each instance is crucial to prevent creating synthetic keys and to ensure the correct association with the roles in the sales process.

This approach ensures that the data model will correctly support all the required analyses, including identifying customers without sales, which is crucial for meeting the business requirements.

Option A and Option B propose using a mapping load and ApplyMap, which can complicate the model and does not directly address all the business requirements.

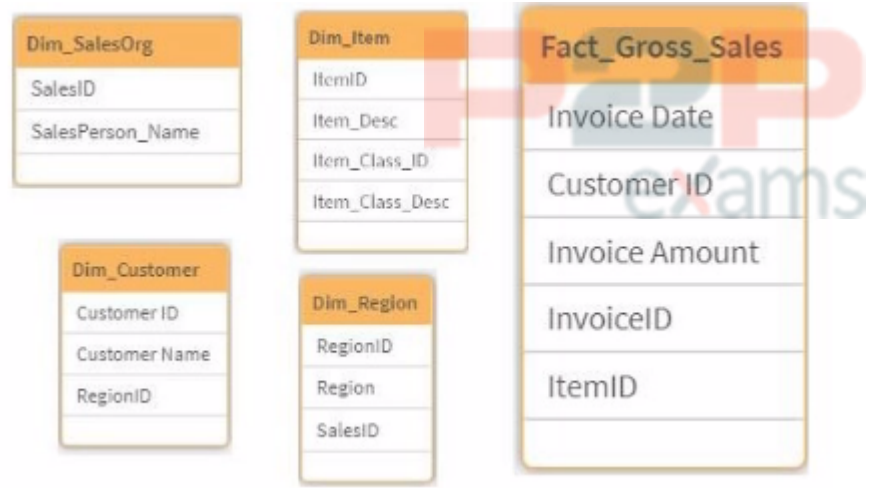
Option D involves aliasing fields in a way that could create unnecessary complexity and might not accurately reflect the relationships in the data.

Thus, Option C is the correct answer as it best meets the requirements while maintaining a clear and functional data model.

Question 7

Question Type: MultipleChoice

Exhibit.



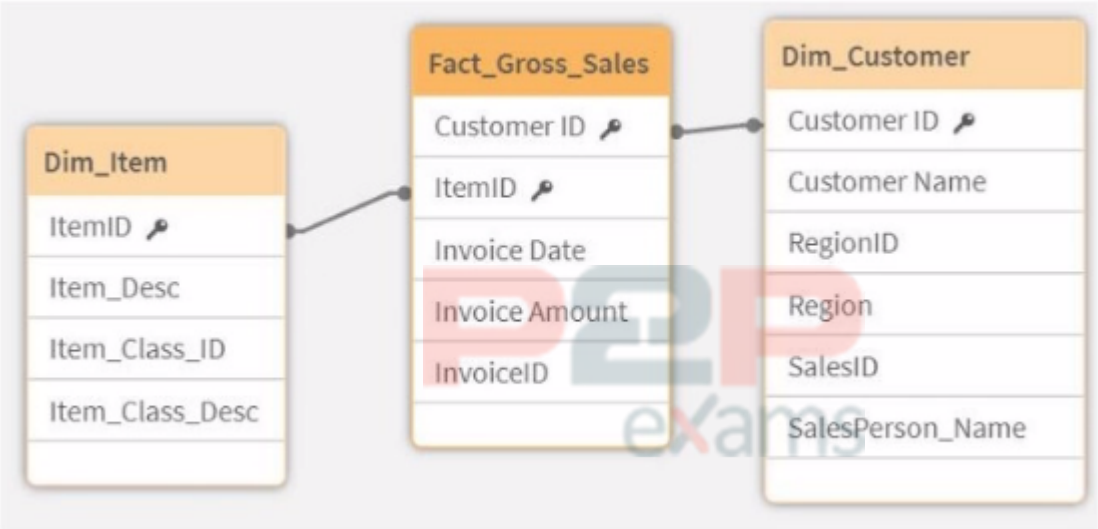
Refer to the exhibit.

A data architect is provided with five tables. One table has Sales Information. The other four tables provide attributes that the end user will group and filter by.

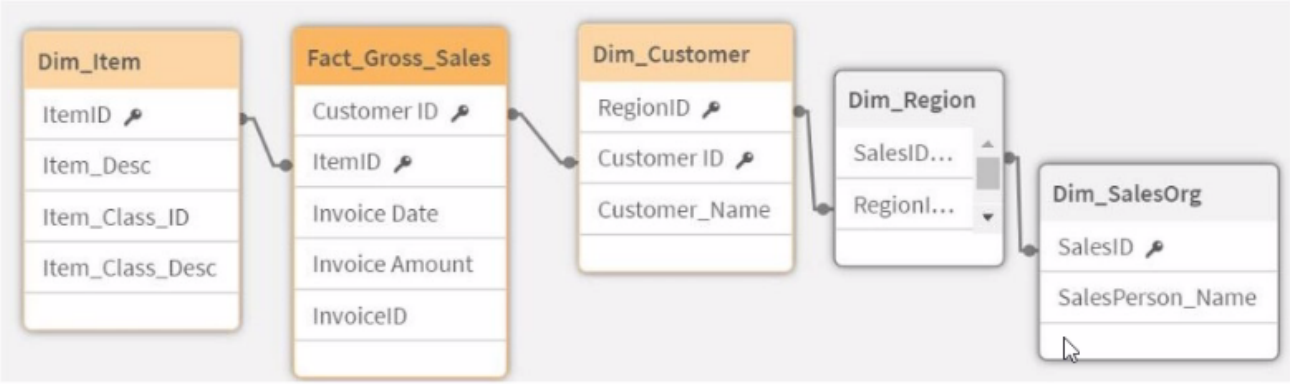
There is only one Sales Person in each Region and only one Region per Customer.

Which data model is the most optimal for use in this situation?

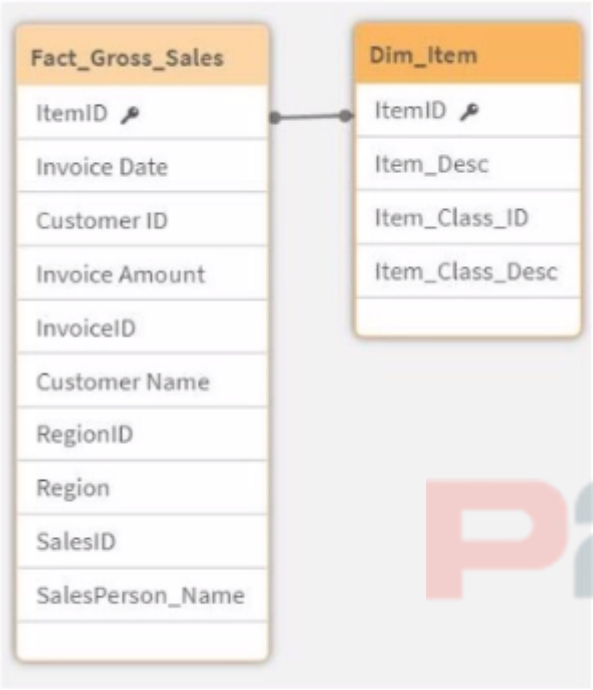
A)



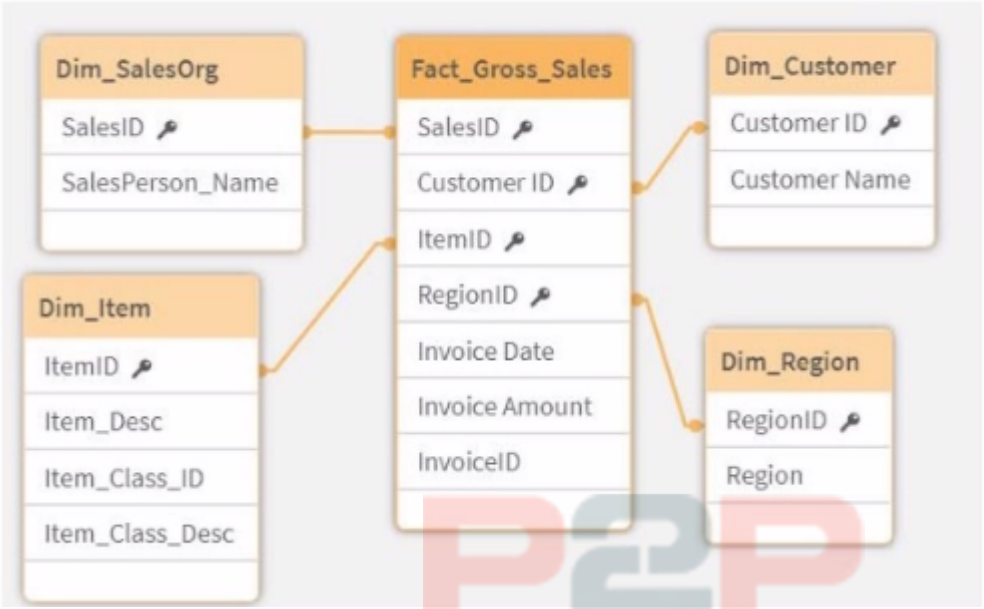
B)



C)



D)



Options:

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

Answer:

D

Explanation:

In the given scenario, where the data architect is provided with five tables, the goal is to design the most optimal data model for use in Qlik Sense. The key considerations here are to ensure a proper star schema, minimize redundancy, and ensure clear and efficient relationships among the tables.

Option D is the most optimal model for the following reasons:

Star Schema Design:

In Option D, the Fact_Gross_Sales table is clearly defined as the central fact table, while the other tables (Dim_SalesOrg, Dim_Item, Dim_Region, Dim_Customer) serve as dimension tables. This layout adheres to the star schema model, which is generally recommended in Qlik Sense for performance and simplicity.

Minimization of Redundancies:

In this model, each dimension table is only connected directly to the fact table, and there are no unnecessary joins between dimension tables. This minimizes the chances of redundant data and ensures that each dimension is only represented once, linked through a unique key to the fact table.

Clear and Efficient Relationships:

Option D ensures that there is no ambiguity in the relationships between tables. Each key field (like Customer ID, SalesID, RegionID, ItemID) is clearly linked between the dimension and fact tables, making it easy for Qlik Sense to optimize queries and for users to perform accurate aggregations and analysis.

Hierarchical Relationships and Data Integrity:

This model effectively represents the hierarchical relationships inherent in the data. For example, each customer belongs to a region, each salesperson is associated with a sales organization, and each sales transaction involves an item. By structuring the data in this way, Option D maintains the integrity of these relationships.

Flexibility for Analysis:

The model allows users to group and filter data efficiently by different attributes (such as salesperson, region, customer, and item). Because the dimensions are not interlinked directly with each other but only through the fact table, this setup allows for more flexibility in creating visualizations and filtering data in Qlik Sense.

Qlik Sense Best Practices: Adhering to star schema designs in Qlik Sense helps in simplifying the data model, which is crucial for performance optimization and ease of use.

Data Modeling Guidelines: The star schema is recommended over snowflake schema for its simplicity and performance benefits in Qlik Sense, particularly in scenarios where clear relationships are essential for the integrity and accuracy of the analysis.