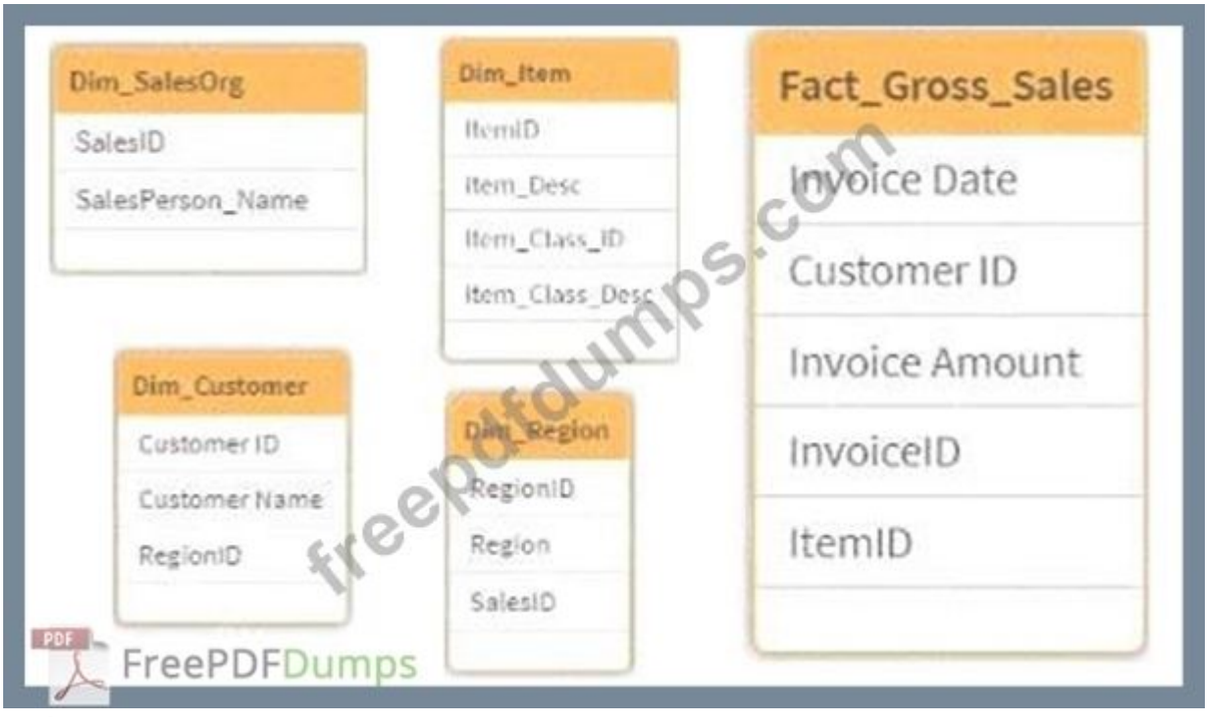
Qlik.QSDA2024.v2024-11-06.q18

Exam Code:	QSDA2024
Exam Name:	Qlik Sense Data Architect Certification Exam - 2024
Certification Provider:	Qlik
Free Question Number:	18
Version:	v2024-11-06
# of views:	233
# of Questions views:	180
https://www.freepdfdumps.com/Qlik.QSDA2024.v2024-11-06.q18.html	

NEW QUESTION: 1

Exhibit.



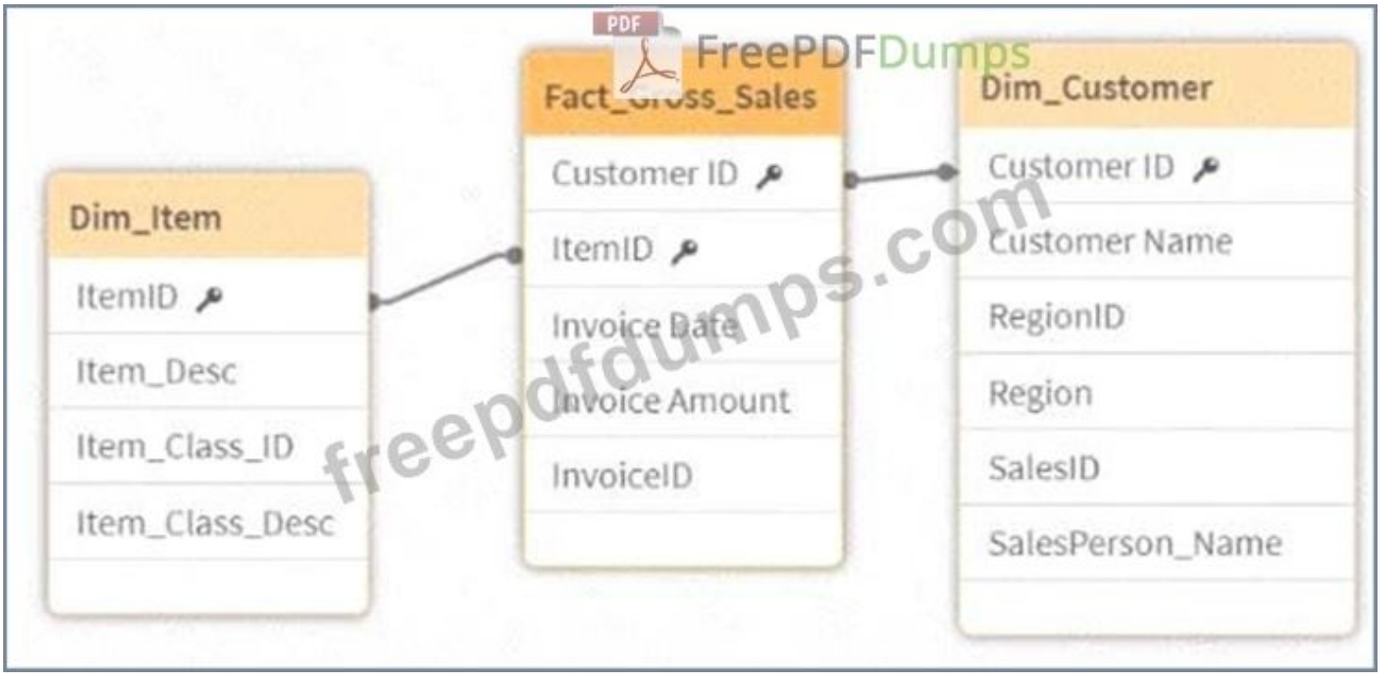
Refer to the exhibit.

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.

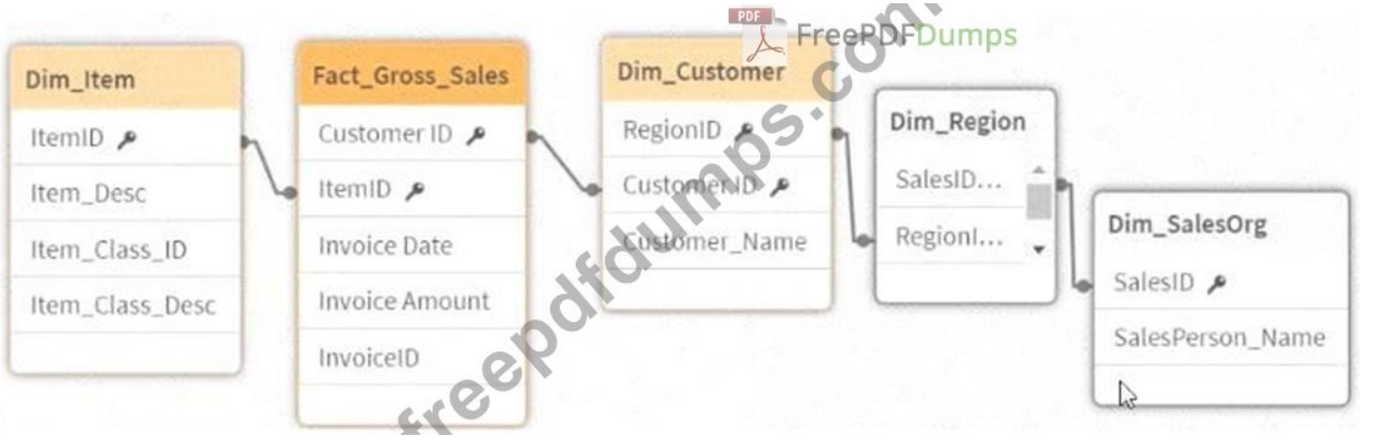
There is only one Sales Person in each Region and only one Region per Customer.

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

A.

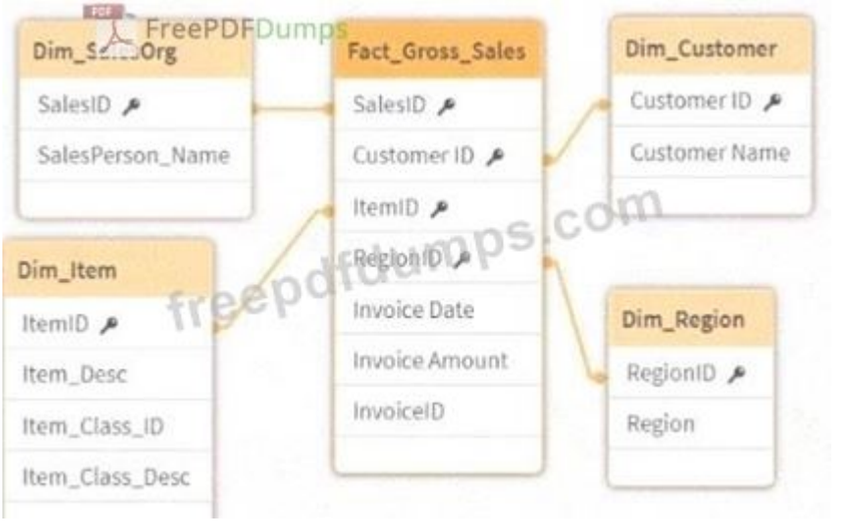


B.



C.

D.



Answer: (SHOW ANSWER)

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 Dis 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.
References:
* 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.

NEW QUESTION: 2
A data architect executes the following script:

```
Table_A:
LOAD * INLINE [
Field_1, Field_2, Field_3
01, AB, 10
01, AC, 50
02, AD, 75
];

Join(Table_A)
Table_B:
LOAD * INLINE [
Field_1, Field_4, Field_5
01, 30%, 500
03, 60%, 1000
];
```

What will be the result of Table.A?

A.

Preview of data

Field_1	Field_2	Field_3	Field_4	Field_5
01	AB	10	30%	500
01	AC	50	30%	500

B.

Preview of data

Field_1	Field_2	Field_3	Field_4	Field_5
01	AB	10	30%	500
01	AC	50	30%	500

C.

Preview of data

Field_1	Field_2	Field_3	Field_4	Field_5
01	AB	10	30%	500
01	AC	50	30%	500
02	AD	75	-	-
03	-	-	60%	1000

D.

Preview of dat

Field_1	Field_2	Field_3	Field_4	Field_5
01	AB	10	30%	500
01	AC	50	30%	500
02	AD	75	-	-

Answer: (SHOW ANSWER)

In the script provided, there are two tables being loaded inline: Table_A and Table_B. The script uses the Join function to combine Table_B with Table_A based on the common field Field_1. Here's how the join operation works:

* Table_Ainitially contains three records with Field_1 values of 01, 01, and 02.
* Table_Bcontains two records with Field_1 values of 01 and 03.

When Join(Table_A) is executed, Qlik Sense will perform an inner join by default, meaning it will join rows from Table_B to Table_A where Field_1 matches in both tables. The result is:

* For Field_1 = 01, there are two matches in Table_A and one match in Table_B. This results in two records in the joined table where Field_4 and Field_5 values from Table_B are repeated for each match in Table_A.
* For Field_1 = 02, there is no corresponding Field_1 = 02 in Table_B, so the Field_4 and Field_5 values for this record will be null.
* For Field_1 = 03, there is no corresponding Field_1 = 03 in Table_A, so the record from Table_B with Field_1 = 03 is not included in the final joined table.

Thus, the correct output will look like this:

* Field_1 = 01, Field_2 = AB, Field_3 = 10, Field_4 = 30%, Field_5 = 500
* Field_1 = 01, Field_2 = AC, Field_3 = 50, Field_4 = 30%, Field_5 = 500
* Field_1 = 02, Field_2 = AD, Field_3 = 75, Field_4 = null, Field_5 = null

NEW QUESTION: 3

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?
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 RegionalAcctMgrID
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 CustomerID
2. Use a Mapping Load for the Employees table
3. Load the Sales table and use ApplyMap to get the names for SalesPersonID and RegionalAcctMgrID
C. 1. Load the Sales table
2. Load the Customers table
3. Load the Employees table twice; name it and alias the EmployeeID field appropriately each time
D. 1. Load the Customers table and alias the CustID field as CustomerID
2. Load the Employees table
3. Load the Sales table and alias the SalesPersonID and RegionalAcctMgrID fields as EmployeeID
Answer: (SHOW ANSWER)
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 whyOption Cis 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 AandOption Bpropose using a mapping load and ApplyMap, which can complicate the model and does not directly address all the business requirements.
* Option Dinvolves aliasing fields in a way that could create unnecessary complexity and might not accurately reflect the relationships in the data.
Thus,Option Cis the correct answer as it best meets the requirements while maintaining a clear and functional data model.

NEW QUESTION: 4
Exhibit.

```
Section Access;
SecurityTable:
Load * INLINE [
ACCESS, USERID, LINK, OMIT
ADMIN, ABC\QSERVICE, LEVEL
USER, ABC\EFN, *,
USER, ABC\JGS, *,
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?
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: (SHOW ANSWER)
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 Accessis 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 columnis the correct outcome.
* Explanation: When user ABC\PPP opens the app, the LEVEL field is omitted from their view. Any table or visualization that uses the LEVEL field will have that field excluded from display. The rest of the data and columns in the table will be visible, but the LEVEL column will not be shown.
References:
* 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.

NEW QUESTION: 5
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?
A. 1
B. 3
C. 4
D. 6
Answer: D (LEAVE A REPLY)
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 tableswill 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 create6 separate tables: one for each unique combination of Object and Attribute.
References:
* 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.

NEW QUESTION: 6
Refer to the exhibit.

OrderID	OrderDate	CustomerI	EmployeeID	Sales
10251	3/3/2016	CST1	EMP7	21.36
10251	3/3/2016	CST1	EMP7	332.64
10251	3/3/2016	CST1	EMP7	185.2
10277	11/5/2016	CST1	EMP7	889.6
10277	11/5/2016	CST1	EMP7	360.96
10289	11/21/2015	CST1	EMP8	616.2
10289	11/21/2015	CST1	EMP8	320.4
10290	11/23/2016	CST1	EMP7	131.4
10290	11/23/2016	CST1	EMP7	1890.45
10290	11/23/2016	CST1	EMP7	294.9
10290	11/23/2016	CST1	EMP7	134.9
10338	1/21/2014	CST1	EMP8	524.8
10338	1/21/2014	CST1	EMP8	381
10349	2/3/2016	CST1	EMP7	820.32
10463	2/28/2017	CST1	EMP8	203.07

A data architect needs to build a dashboard that displays the aggregated sates for each sales representative. All aggregations on the data must be performed in the script.
Which script should the data architect use to meet these requirements?

A.

```
Data:
LOAD
    [OrderID],
    [OrderDate],
    [CustomerID],
    [EmployeeID],
    [Sales]
 FROM [lib://Certification Exam/Sample Daa.xlsx]
(ooxml, embedded labels, table is Sales);

Left Join (Data)
LOAD
    EmployeeID,
    EmployeeName
FROM [lib://Certification Exam/Sample Daa.xlsx]
(ooxml, embedded labels, table is Emp);

Summary:
LOAD
    EmployeeName,
    sum([Sales]) as TotalSales
 FROM [lib://Certification Exam/Sample Daa.xlsx]
(ooxml, embedded labels, table is Sales);
```

```
Data:
LOAD
    [OrderID],
    [OrderDate],
    [CustomerID],
    [EmployeeID],
    [Sales]
 FROM [lib://Certification Exam/Sample Daa.xlsx]
(ooxml, embedded labels, table is Sales);

Left join (Data)
LOAD
    EmployeeID,
    EmployeeName
FROM [lib://Certification Exam/Sample Daa.xlsx]
(ooxml, embedded labels, table is Emp);


Summary:
LOAD
    EmployeeName,
    sum([Sales]) as TotalSales
 Resident Emp Group by (EmployeeName) ;
```


B.


```
Data:
LOAD
    [OrderID],
    [OrderDate],
    [CustomerID],
    [EmployeeID],
    [Sales]
 FROM [lib://Certification Exam/Sample Daa.xlsx]
(ooxml, embedded labels, table is Sales);

Emp:
LOAD
    EmployeeID,
    EmployeeName
FROM [lib://Certification Exam/Sample Daa.xlsx]
(ooxml, embedded labels, table is Emp);


Summary:
LOAD
    EmployeeName,
    sum([Sales]) as TotalSales
 Resident Data Group by (EmployeeName) ;
```


C.

D.

```
Data:
LOAD
    [OrderID],
    [OrderDate],
    [CustomerID],
    [EmployeeID],
    [Sales]
 FROM [lib://Certification Exam/Sample Daa.xlsx]
(ooxml, embedded labels, table is Sales);

Left join (Data)
LOAD
    EmployeeID,
    EmployeeName
FROM [lib://Certification Exam/Sample Daa.xlsx]
(ooxml, embedded labels, table is Emp);


Summary:
LOAD
    EmployeeName,
    sum([Sales]) as TotalSales
 Resident Data Group by (EmployeeName) ;
```


Answer: (SHOW ANSWER)

The goal is to display the aggregated sales for each sales representative, with all aggregations being performed in the script. Option C is the correct choice because it performs the aggregation correctly using a Group by clause, ensuring that the sum of sales for each employee is calculated within the script.

* Data Load:
* The Data table is loaded first from the Sales table. This includes the OrderID, OrderDate, CustomerID, EmployeeID, and Sales.
* Next, the Emp table is loaded containing EmployeeID and EmployeeName.
* Joining Data:
* A Left Join is performed between the Data table and the Emp table on EmployeeID, enriching the data with EmployeeName.
* Aggregation:
* The Summary table is created by loading the EmployeeName and calculating the total sales using the sum([Sales]) function.
* The Resident keyword indicates that the data is pulled from the existing tables in memory, specifically the Data table.
* The Group by clause ensures that the aggregation is performed correctly for each EmployeeName, summarizing the total sales for each employee.

Key Qlik Sense Data Architect References:

* Resident Load: This is a method to reuse data that is already loaded into the app's memory. By using a Resident load, you can create new tables or perform calculations like aggregation on the existing data.

* Group by Clause: The Group by clause is essential when performing aggregations in the script. It groups the data by specified fields and performs the desired aggregation function (e.g., sum, count).
* Left Join: Used to combine data from two tables. In this case, Left Join is used to enrich the sales data with employee names, ensuring that the sales data is associated correctly with the respective employee.
Conclusion:Option C is the most appropriate script for this task because it correctly performs the necessary joins and aggregations in the script. This ensures that the dashboard will display the correct aggregated sales per employee, meeting the data architect's requirements.

NEW QUESTION: 7
Exhibit.

A large electronics company re-assigns sales people once per year from one Department to another.
SPID is the Salesperson ID; the SPID for each individual sales person Name remains constant.
The Department for a SPID may change; each change is stored in the Dynamic Dimension data.
Four tables need to be linked correctly: a transaction table, a dynamic salesperson dimension, a static salesperson dimension, and a department dimension.
Which script prefix should the data architect use?
A. Merge
B. IntervalMatch
C. Partial Reload
D. Semantic
Answer: (SHOW ANSWER)
In the scenario described, the Dynamic Dimension data tracks changes in department assignments for salespeople over time. To correctly link the transaction data with the salesperson data and ensure that sales are associated with the correct department based on the date, an IntervalMatch function should be used.
IntervalMatchis designed to match discrete data (like transaction dates) with a range of dates. In this case, each salesperson's department assignment is valid over a period of time, and the IntervalMatch function can be used to link the transaction data with the correct department for each salesperson based on the transaction date.
* Option A (Merge):This option is incorrect as it refers to combining data sets, which doesn't address the need to handle the dynamic, date-based department assignments.

* Option B (IntervalMatch):This is the correct choice because it allows you to match each transaction with the correct department assignment based on the ChangeDate in the Dynamic Dimension data.
* Option C (Partial Reload):This refers to reloading only part of the data, which is not relevant to linking tables based on date ranges.
* Option D (Semantic):This option is not applicable as it refers to a broader approach to data modeling and interpretation rather than specifically linking data based on time intervals.
Thus,IntervalMatchis the correct method for linking the transaction data with the dynamic salesperson dimension, ensuring that each transaction is associated with the correct department based on the historical assignment data.

NEW QUESTION: 8
A startup company is about have its Initial Public Offering (IPO) on the New York Stock Exchange.
This startup company has used Qlik Sense for many years for data-based decision making for Sales and Marketing efforts, as well as for input into Financial Reporting. The startup's Qlik Sense applications use variables that have different values at different points in time.
Due to the increased rigor required in record keeping for public companies, these variables must be clearly recorded in the script reload logs of the Qlik Sense applications. These logs are refreshed daily.
The data architect wants to have the variables names, with their current values,writteninto the script reload logs. Which script statement should the data architect use?
A. LogDetail
B. Trace
C. Tag
D. REM
Answer: B (LEAVE A REPLY)
In the scenario where the startup company is preparing for an IPO, there is an increased need for meticulous record-keeping, including the recording of variable values used in Qlik Sense applications. The TRACE statement is the most suitable option for logging variable values during script execution.
* TRACE: This statement writes custom messages, including variable values, to the script execution log.
By using TRACE, you can ensure that every reload log contains the names and current values of all relevant variables, providing the necessary transparency and traceability.
For example, the script could include:
TRACE $(VariableName);
This command will output the variable's value in the script log, ensuring it is recorded for audit purposes.

NEW QUESTION: 9

A data architect receives an error while running script.
What will happen to the existing data model?
A. The data model will be removed from the application.
B. The latest error-free data model will be maintained.
C. Newly loaded tables will be merged with the existing data model until the error is resolved.
D. The data model will be replaced with the tables that were successfully loaded before the error.
Answer: B (LEAVE A REPLY)
In Qlik Sense, when a data load script is executed and an error occurs, the script execution is halted immediately, and any tables that were being loaded at the time of the error are discarded. However, the existing data model-i.e., the last successfully loaded data model-remains intact and is not affected by the failed script. This ensures that the application retains the last known good state of the data, avoiding any partial or inconsistent data loads that could occur due to an error.
When the script encounters an error:
* The tables that were successfully loaded prior to the error are retained in the session, but these tables are not merged with the existing data model.
* The existing data model before the script was executed remains unchanged and is maintained.
* No partial or incomplete data is loaded into the application; hence, the data model remains consistent and reliable.
Qlik Sense Data Architect ReferencesThis behavior is designed to protect the integrity of the data model. In scenarios where script execution fails, the user can debug and fix the script without risking the data integrity of the existing application. The key references include:
* Qlik Help Documentation: Provides detailed information on how Qlik Sense handles script errors, highlighting that the existing data model remains unchanged after an error.
* Data Load Editor Practices: Best practices dictate ensuring that the script is fully functional before executing it to avoid data inconsistency. In cases where an error occurs, understanding that the current data model is maintained helps in strategic debugging and script correction.

NEW QUESTION: 10
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?
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 (LEAVE A REPLY)

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 Cis 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.
References:
* 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.

NEW QUESTION: 11
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 ####;
```

Answer: B (LEAVE A REPLY)
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 References:
* 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.

NEW QUESTION: 12
A company needs to analyze daily sales data from different countries. They also need to measure customer satisfaction of products as reported on a social media website. Thirty (30) reports must be produced with an average of 20,000 rows each. This process is estimated to take about 3 hours.
Which option should the data architect use to build this solution?
A. Qlik REST Connector
B. Microsoft SQL Server
C. Qlik GeoAnalytics
D. Mailbox IMAP
Answer: (SHOW ANSWER)
In this scenario, the company needs to analyze daily sales data from different countries and also measure customer satisfaction of products as reported on a social media website. This suggests that the data is likely coming from different sources, including possibly an API or a web service (social media website).
TheQlik REST Connectoris the appropriate tool for this job. It allows you to connect to RESTful web services and retrieve data directly into Qlik Sense. This is especially useful for integrating data from various online sources, such as social media platforms, which typically expose data via REST APIs. The REST Connector enables the extraction of large datasets from these sources, which is necessary given the requirement to produce 30 reports with an average of 20,000 rows each.
* Microsoft SQL Serveris not suitable for fetching data from web services or social media platforms.
* Qlik GeoAnalyticsis used for mapping and geographical data visualization, not for connecting to RESTful services.
* Mailbox IMAPis for connecting to email servers and is not applicable to the data extraction needs described here.

Thus,Qlik REST Connectoris the correct answer for this scenario.

NEW QUESTION: 13
Refer to the exhibit.

A system creates log files and csv files daily and places these files in a folder. The log files are named automatically by the source system and change regularly. All csv files must be loaded into Qlik Sense for analysis.
Which method should be used to meet the requirements?

A.
```
SET vFileList = 'Day1','Day2','Day3';
FOR EACH vFile in $(vFileList)
LOAD * FROM [lib://Data (User1)/multi/$(vFile).csv];
NEXT
```

B.
```
LOAD *
FROM [lib://Data (User1)/multi/*]
(csv);
```

C.
```
LOAD *
FROM [lib://Data (User1)/multi/*]
(txt);
```

D.
```
SET vLib = chr(39) & 'lib://Data (User1)/multi/*.csv' & chr(39);
FOR EACH File in filelist($(vLib))
LOAD * FROM [$(File)];
NEXT
```

Answer: B (LEAVE A REPLY)
In the scenario described, the goal is to load all CSV files from a directory into Qlik Sense, while ignoring the log files that are also present in the same directory. The correct approach should allow for dynamic file loading without needing to manually specify each file name, especially since the log files change regularly.
Here's whyOption Bis the correct choice:
* Option A:This method involves manually specifying a list of files (Day1, Day2, Day3) and then iterating through them to load each one. While this method would work, it requires knowing the exact file names in advance, which is not practical given that new files are added regularly. Also, it doesn't handle dynamic file name changes or new files added to the folder automatically.
* Option B:This approach uses a wildcard (*) in the file path, which tells Qlik Sense to load all files matching the pattern (in this case, all CSV files in the directory). Since the csv file extension is

explicitly specified, only the CSV files will be loaded, and the log files will be ignored. This method is efficient and handles the dynamic nature of the file names without needing manual updates to the script.
* Option C:This option is similar to Option B but targets text files (txt) instead of CSV files. Since the requirement is to load CSV files, this option would not meet the needs.
* Option D:This option uses a more complex approach with filelist() and a loop, which could work, but it's more complex than necessary. Option B achieves the same result more simply and directly.
Therefore,Option Bis the most efficient and straightforward solution, dynamically loading all CSV files from the specified directory while ignoring the log files, as required.

NEW QUESTION: 14
A data architect executes the following script:

Which values does the OrderDate field contain after executing the script?
A. 20210131, 2020/01/31, 31/01/2019
B. 20210131, 2020/01/31, 31/01/2019, 9999
C. 20210131, 2020/01/31, 31/01/2019, 0
D. 20210131, 2020/01/31, 31/01/2019, 31/12/2022
Answer: D (LEAVE A REPLY)
In the script provided, the alt() function is used to handle various date formats. The alt() function in Qlik Sense evaluates a list of expressions and returns the first valid expression. If none of the expressions are valid, it returns the last argument provided (in this case, '31/12/2022').
Step-by-step breakdown:
* The alt() function checks the Date field for three different formats:
* YYYYMMDD
* YYYY/MM/DD
* DD/MM/YYYY
* If none of these formats match the value in the Date field, the default date '31/12/2022' is assigned.
Values in the Date field:
* 20210131: Matches the first format YYYYMMDD.

* 2020/01/31: Matches the second format YYYY/MM/DD.
* 31/01/2019: Matches the third format DD/MM/YYYY.
* 9999: Does not match any of the formats, so the alt() function returns the default value '31/12/2022'.

NEW QUESTION: 15
A data architect needs to develop a script to export tables from a model based upon rules from an independent file. The structure of the text file with the export rules is as follows:

```
TableToExport,StoreAsFilename,NumberOfCopies
Customers,Clients,3
Orders,SalesOrders,1
Regions,Countries,2
```

These rules govern which table in the model to export, what the target root filename should be, and the number of copies to export.
The TableToExport values are already verified to exist in the model.
In addition, the format will always be QVD, and the copies will be incrementally numbered.
For example, the Customers table would be exported as:

```
Clients1.qvd
Clients2.qvd
Clients3.qvd
```

What is the minimum set of scripting strategies the data architect must use?
A. One loop and two IF statements
B. One loop and one SELECT CASE statement
C. Two loops and one IF statement
D. Two loops without any conditional statements
Answer: A (LEAVE A REPLY)
In the provided scenario, the goal is to export tables from a Qlik Sense model based on rules specified in an external text file. The structure of the text file indicates which table to export, the filename to use, and how many copies to create.
Given this structure, the data architect needs to:
* Loop through each row in the text file to process each table.
* Use an IF statement to check whether the specified table exists in the model (though it's mentioned they are verified to exist, this step may involve conditional logic to ensure the rules are correctly followed).
* Use another IF statement to handle the creation of multiple copies, ensuring each file is named incrementally (e.g., Clients1.qvd, Clients2.qvd, etc.).
Key Script Strategies:
* Loop: A loop is necessary to iterate through each row of the text file to process the tables specified for export.
* IF Statements: The first IF statement checks conditions such as whether the table should be exported (based on additional logic if needed). The second IF statement handles the creation of multiple copies by incrementing the filename.

This approach covers all the necessary logic with the minimum set of scripting strategies, ensuring that each table is exported according to the rules defined.

NEW QUESTION: 16
A data architect needs to retrieve data from a REST API. The data architect needs to loop over a series of items that are being read using the REST connection.
What should the data architect do?
A. Recreate the SQL Statement with the correct parameters
B. Use the REST Connector with pagination mechanism
C. Use pagination of the REST Connector to create a template of the desired data
D. Use With Connection to pass a parameter to the REST URL
Answer: B (LEAVE A REPLY)
When retrieving data from a REST API, particularly when the dataset is large or the data is segmented across multiple pages (which is common in REST APIs), the REST Connector in Qlik Sense needs to be configured to handle pagination.
Pagination is the process of dividing the data retrieved from the API into pages that can be loaded sequentially or as required. Qlik Sense's REST Connector supports pagination by allowing the dataarchitect to set parameters that will sequentially retrieve each page of data, ensuring that the complete dataset is retrieved.
Key Steps:
* REST Connector Setup: Configure the REST connector in Qlik Sense and specify the necessary API endpoint.
* Pagination Mechanism: Use the built-in pagination mechanism to define how the connector should retrieve the subsequent pages (e.g., by using query parameters like page or offset).

NEW QUESTION: 17

```
CountryTable:
load * inline [
country,          Total_Survey_Score
U.S,              2005
US,               2389
United States,    1890
DE,               605
IT,               764
FR,               1045
];

Fact_Table:
NoConcatenate
load
    applymap('MAP_COUNTRY', country) as country,
    Total_Survey_Score
resident CountryTable;

drop table CountryTable;
```

Country	Total Survey Score
Totals	8.698
FRANCE	1.045
GERMANY	605
ITALY	764
US	2.389
USA	3.895

Refer to the exhibits.

On executing a load script of an app, the country field needs to be normalized. The developer uses a mapping table to address the issue. The script runs successfully but the resulting table is not correct.

What should the data architect do?

A. Create two different mapping tables

B. Use LOAD DISTINCT on the mapping table

C. Use a LEFT JOIN Instead of the APPLYMAP

D. Review the values of the source mapping table

Answer: (SHOW ANSWER)

In this scenario, the issue arises from using the applymap() function to normalize the country field values, but the result is incorrect. The reason is most likely related to the values in the source mapping table not matching the values in the Fact_Table properly.

The applymap() function in Qlik Sense is designed to map one field to another using a mapping table. If the source values in the mapping table are inconsistent or incorrect, the applymap() will not function as expected, leading to incorrect results.

Steps to resolve:

* Review the mapping table (MAP_COUNTRY): The country field in the CountryTable contains values such as "U.S.", "US", and "United States" for the same country. To correctly normalize the country names, you need to ensure that all variations of a country's name are consistently mapped to a single value (e.g., "USA").
* Apply Mapping: Review and clean up the mapping table so that all possible variants of a country are correctly mapped to the desired normalized value.
Key References:
* Mapping Tables in Qlik Sense: Mapping tables allow you to substitute field values with mapped values. Any mismatches or variations in source values should be thoroughly reviewed.
* Applymap() Function: This function takes a mapping table and applies it to substitute a field value with its mapped equivalent. If the mapped values are not correct or incomplete, the output will not be as expected.

NEW QUESTION: 18
A data architect needs to load data from two different databases. Additional data will be added from a folder that contains QVDs, text files, and Excel files.
What is the minimum number of data connections required?
A. Four
B. Two
C. Three
D. Five
Answer: B (LEAVE A REPLY)
In the scenario, the data architect needs to load data from two different databases, and additional data is located in a folder containing QVDs, text files, and Excel files.
Minimum Number of Data Connections Required:
* Database Connections:
* Each database requires a separate data connection. Therefore, two data connections are needed for the two databases.
* Folder Connection:
* A single folder data connection can be used to access all the QVDs, text files, and Excel files in the specified folder. Qlik Sense allows you to create a folder connection that can access multiple file types within that folder.
Total Connections:
* Two Database Connections: One for each database.
* One Folder Connection: To access the QVDs, text files, and Excel files.
Therefore, the minimum number of data connections required istwo.