DUMPSBOSS.

Qlik Sense Data Architect Certification Exam - February 2021 Release

Qlik QSDA2021

Version Demo

Total Demo Questions: 9

Total Premium Questions: 93

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Data Modeling	31
Topic 2, Data Loading	20
Topic 3, Data Transformation	36
Topic 4, Data Governance	6
Total	93

QUESTION NO: 1

A data architect needs to develop multiple apps for various departments. More departments are requesting apps over time The company uses specific requirements for the number interpretation variables (e.g., ThousandSep, DecimalSep) found at the beginning of a LOAD script.

The data architect wants to reduce duplicate scripts but does not want to copy and paste the number interpretation variables each time new app is created. The data architect prefers to use the least amount of script in Qlik Sense.

How should the data architect meet these requirements?

A. Save the script for the number interpretation variables in a text file and use the CALL function to insert the variables.

B. Create an Excel file with the number interpretation variables and use a FOR Next loop to load the variables into Qlik Sense

C. Save the script for the number interpretation variables in a text file and use the INCLUDE function to insert the variables.

D. Create an Excel file with the number interpretation variables and apply the variables to the app using a mapping table

ANSWER: C

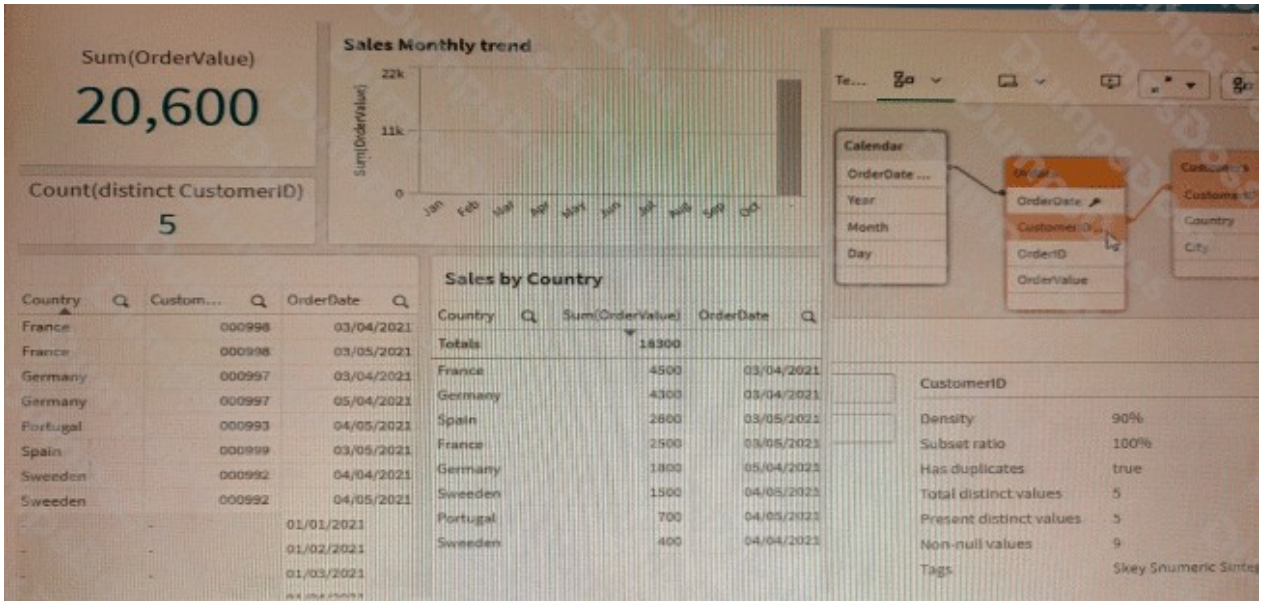
Explanation:

Save the script for the number interpretation variables in a text file and use the INCLUDE function to insert the variables. This approach centralizes the common script settings, including number interpretation variables such as `ThousandSep` and `DecimalSep`, in one externally maintained file. Each departmental app can then reference that file at reload time rather than carrying a copied version of the same settings. This keeps individual load scripts short, prevents repetitive maintenance, and helps ensure all newly created apps use the company's required interpretation standards consistently.

In Qlik Sense script, the `$(Include=filename)` expansion inserts the contents of an external text file into the script at the location where it is referenced. The included file can contain normal script statements, so it is appropriate for reusable variable and environment-setting blocks placed at the start of an app's load script. When a standard changes, the architect updates the shared include file rather than editing every app separately. Qlik documents the include mechanism and its file-reference requirements in the Qlik Sense Include statement documentation; the relevant number interpretation variables are listed in the Number interpretation variables documentation.

QUESTION NO: 2

Refer to the exhibit.



A data architect is working with an app and creates some visualizations to check the data. Some visualizations show issues in the data set.

* The Sales by Country table shows a total OrderValue of 18,300 sales while the KPI shows a total OrderValue of 20,600.

* The Sales monthly trend bar chart does not work with the Month field.

Which two data issues should the data architect fix in the app? (Select two.)

A. The Month field does not exist in the Orders table and needs to be incorporated in the table using he Calendar table.

B. In the Orders table, some CustomerID values are null because there are orders with no customer.
and needs to be incorporated in the table using the Calendar table, null because there are orders with no customer

C. In the Orders table, some values in the CustomerID field do not exist in the Customers table.

D. The OrderDate field values in the Calendar table do not match with the values in the OrderDate field from the Orders table

ANSWER: C D

Explanation:

In the Orders table, some values in the CustomerID field do not exist in the Customers table. This must be fixed because Sales by Country relies on the association from orders to customers, where the country attribute is held. Orders whose customer key has no matching customer record remain unassociated with a country. A KPI that simply sums OrderValue can still include all order rows, while a table grouped by Country will omit or separate the unmatched orders, producing the observed difference between the two totals. Qlik associations are based on common field names and matching field values, so referential integrity of the customer key is essential. See Qlik documentation on associations in the data model.

The OrderDate field values in the Calendar table do not match with the values in the OrderDate field from the Orders table. The Month dimension is supplied by the calendar, and it can filter or group order measures only when its date key associates correctly to order records. Inconsistent date formatting, timestamps versus date-only values, or incompatible parsed values prevents that association and leaves the monthly trend without the expected order data. The architect should standardize both fields to the same canonical date value before creating the association. Qlik guidance on interpreting and formatting dates is available in the Date function documentation.

QUESTION NO: 3

Refer to the exhibit

A data architect is working on an app that contains orders, invoices, and shipping data. There are three different date fields within the data:

* OrderDate

* InvoiceDate

* ShippingDate

The business analyst needs to replicate the chart above to show Order and Shipping amounts on the same Month axis.

What should the data architect do?

A. Create a Month field for each of the three dates in the fact table and use that in the chart

B. Load the key field and the three date fields into a concatenated bridge table that contains KeyField and Date

C. Left Join the three date fields onto one bridge table using the key field containing KeyField and Date

ANSWER: B

Explanation:

Loading the key field and the three date fields into a concatenated bridge table that contains KeyField and Date creates a canonical-date structure. Each business event date is represented as a value in one common Date field, associated with the relevant record through KeyField. A master calendar can then be linked to that common Date field, making its Month dimension available as a single, consistent axis for analysis.

This design lets the chart aggregate order amounts and shipping amounts by the same calendar month while preserving each measure's association with its originating event and date. For example, an order contributes using its OrderDate-derived bridge record, while a shipment contributes using its ShippingDate-derived bridge record. The model therefore supports analysis across multiple date roles without requiring separate month dimensions or disconnected calendar selections. Qlik describes using a link table when multiple tables need to be associated through shared key values, and its data-model guidance emphasizes designing associations deliberately to avoid ambiguity. See Qlik Help: Understanding data models and Qlik Help: Date and time functions.

QUESTION NO: 4

A data architect is creating a reusable QVD architecture for several departmental apps.

The source systems should be accessed as few times as possible, and each department requires its own transformed data model.

Which two practices should the data architect use? (Select two.)

A. Extract source-system data into QVD files

B. Use QVD files as input for transformation layers

C. Connect every departmental app directly to each source system

D. Store all departmental data in one unfiltered fact table

E. Use a separate database connection for every QVD file

ANSWER: A B

Explanation:

The data architect should extract source-system data into QVD files and use those QVD files as the input for transformation and departmental application layers. This reduces repeated access to operational source systems and allows multiple apps to reuse a governed extracted dataset.

Transforming data from QVD files also separates source extraction from business-specific modeling. Departmental apps can then load curated QVDs appropriate to their requirements without each app independently querying and transforming the original operational data.

QUESTION NO: 5

A company generates 1 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 aggregated data for 2 years and use On-Demand App Generation (ODAG) for transaction data

B. Load only aggregated data for 2 years and apply filters on a sheet for transaction data

C. Load only 2 years of data in an aggregated app and create a separate transaction app for occasional use

D. Load only 2 years of data and use best practices in scripting and visualization to calculate and display aggregated data

ANSWER: A

Explanation:

Load only aggregated data for 2 years and use On-Demand App Generation (ODAG) for transaction data is the appropriate architecture because it separates the common analytical workload from the exceptional detailed-data workload. Users primarily need trend analysis over a two-year period, so a compact aggregated application can preload the measures and dimensions needed for responsive everyday exploration without requiring the full 720 GB transaction dataset to be held in the app's associative in-memory model. When a user has a legitimate, infrequent need to investigate transactions for a particular date or selection, an ODAG selection app can pass those selections to an on-demand template and generate a detailed app containing only the relevant subset of records. This gives users governed access to detail while controlling reload time, memory consumption, and app size. ODAG is specifically intended for scenarios where the complete detailed dataset is too large to load into a single Qlik Sense app but users still need to generate focused detailed analyses from an aggregated source app. The aggregate app also provides a natural place to define the selection criteria and limits that govern each generated detail app. See Qlik's On-demand app generation documentation and ODAG overview.

QUESTION NO: 6

A data architect loads a Product table containing active products and an Orders table containing several years of order history.

Only orders for active products must be loaded. Product attributes must remain in a separate Product table and must NOT be joined into the Orders table.

Which two actions should the data architect take? (Select two.)

A. Load the Product table before the Orders table

B. Use Inner Keep between Product and Orders

C. Use Inner Join between Product and Orders

D. Use Concatenate to add Product records to Orders

E. Load Orders before the Product table

ANSWER: A B

Explanation:

The Product table should be loaded before the Orders table so that its ProductID values are available when the related Orders table is processed. The data architect should then use an `Inner Keep` operation between the tables.

An `Inner Keep` reduces both tables to records having matching field values, while retaining the tables as separate logical tables. In this case, orders for inactive or unknown products are removed and the Product table contains only products having matching orders. Unlike an `Inner Join`, Keep does not combine the product attributes into the Orders table.

QUESTION NO: 7

A data architect needs to develop multiple apps for various departments. More departments are requesting apps over time The company uses specific requirements for the number interpretation variables (e.g., ThousandSep, DecimalSep) found at the beginning of a LOAD script.

The data architect wants to reduce duplicate scripts but does not want to copy and paste the number interpretation variables each time new app is created. The data architect prefers to use the least amount of script in Qlik Sense.

How should the data architect meet these requirements?

A. Save the script for the number interpretation variables in a text file and use the CALL function to insert the variables.

B. Create an Excel file with the number interpretation variables and use a FOR Next loop to load the variables into Qlik Sense

C. Save the script for the number interpretation variables in a text file and use the INCLUDE function to insert the variables

D. Create an Excel file with the number interpretation variables and apply the variables to the app using a mapping table

ANSWER: C

Explanation:

Save the script for the number interpretation variables in a text file and use the INCLUDE function to insert the variables is correct because Qlik Sense supports external script inclusion through the `$(Include=filename)` directive. The architect can place the shared number-interpretation statements, such as `ThousandSep`, `DecimalSep`, date formats, and other locale-dependent settings, in one centrally maintained text file. Each app then needs only a short INCLUDE statement near the start of its load script to bring in that standard configuration.

This approach keeps application scripts concise while ensuring that every department's app begins with the same approved number-interpretation settings. It also improves maintainability: when the organization changes a standard separator or format requirement, the architect updates the shared file rather than editing the repeated settings in every individual app. Qlik documents that the Include statement inserts the contents of a file into the script during script execution, which is specifically useful for reusing common script sections. The shared file must be stored in an accessible data connection or otherwise referenced according to the deployment's permitted file-access configuration. See the Qlik documentation for the Include statement and number interpretation variables.

QUESTION NO: 8

A data architect is implementing section access for a published app. Regional managers must see data only for their assigned Region values.

Which two requirements must be met for data reduction to work correctly? (Select two.)

A. The reduction field must exist in Section Access and in the application data model

B. The reduction values must match the application data values

C. The reduction field must be a master dimension

D. The app must contain a data island for each region

E. All users must be assigned ADMIN access

ANSWER: A B

Explanation:

The reduction field must exist in both the Section Access table and the application data model, and the values in the reduction field must match the values loaded into the application. Qlik uses the common reduction field to associate a user or group with the data that remains available after the app is opened.

Section access field names and values are handled in uppercase by Qlik. Converting the reduction values consistently during loading is a common way to avoid mismatches. The access control fields such as `ACCESS` and `USERID` define authorization, but they do not replace the shared reduction field required for row-level data reduction. See Qlik guidance for managing security with section access.

QUESTION NO: 9

Adata architect needs to efficiently prepare a data model for a meeting in an hour.

The data source to be used contains five date fields The app needs to display sales trends and compare the current year to date (CYTD) to last year to date (LYTD) The app is NOT going to be published It will only be used for this meeting and a single user's ad-hoc analysis.

What should the data architect do to meet these requirements?

A. Use the data manager

B. Load a calendar island

C. Create a canonical calendar

D. Create five master calendars

ANSWER: A

Explanation:

Use the data manager is the appropriate choice because the app is a short-lived, single-user analysis that must be prepared quickly. Qlik Sense Data Manager is designed for rapidly adding and profiling data, associating tables, and making common data-preparation changes through the user interface rather than requiring a more elaborate scripted model. It also recognizes date and time fields and can generate calendar-related fields that support analysis by periods such as year, month, week, and day. Those derived calendar fields give the user the dimensions needed to create sales-trend visualizations promptly.

For the stated comparison, the analyst can use the generated date-related fields with Qlik expressions to calculate current-year-to-date and prior-year-to-date sales. Since the application is not intended for publication, reuse, or governed multi-user reporting, the speed and convenience of Data Manager are particularly valuable. It provides sufficient date analysis capability for an ad-hoc meeting without investing time in a more engineered, reusable calendar architecture. Qlik documents Data Manager as the visual interface for loading and preparing data, including its support for working with date fields and generated calendar information. See the Qlik Sense Data Manager documentation and Qlik's derived fields documentation.