Free Salesforce Plat-Con-201 Practice Questions

Shared by Hansen on 13-08-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

What is best practice regarding the number of fields returned by a data source to a FlexCard?

Options:

A- Return all available fields to ensure future flexibility.
B- Return all fields as a single concatenated value.
C- Return fields in a standardized XML structure.
D- Return only the fields required for display or processing.

Answer:

D

Explanation:

FlexCard performance is directly influenced by the amount of data transferred between the server and the client. Returning only the fields required for display, calculations, or actions minimizes payload size, improves rendering speed, reduces network traffic, and simplifies maintenance. Returning every available field may seem flexible, but it increases response size and can negatively affect performance, particularly on mobile devices and slower networks. Concatenating fields into a single value reduces usability and limits downstream processing options. XML standardization is unrelated to FlexCard optimization. Salesforce recommends trimming data returned from Integration Procedures and Data Mappers so that only necessary information is passed to consuming components. This principle is a core OmniStudio performance best practice and contributes to faster, more scalable user experiences.

==========

Question 2

Question Type: MultipleChoice

A business is creating an agent console with FlexCards to provide a 360 view of their customers. The business wants the following information displayed:

* Account information including account name, phone, and website

* Active opportunities related to the account

* Active contracts related to the account

* The ability to view and renew contracts

An Integration Procedure will be used to retrieve Account, Opportunity, and Contract data.

How should the consultant design the FlexCards to meet these requirements?

Options:

A- Parent FlexCard with multiple Child and Card Actions
B- Parent FlexCard with multiple Child and different Card States
C- Parent FlexCard with single Child and multiple Card States
D- Parent FlexCard with single Child and Card Actions

Answer:

A

Explanation:

The consultant should design the FlexCards using a Parent FlexCard with multiple Child and Card Actions to meet these requirements. A Parent FlexCard is a FlexCard that can display data and actions in a card format, and also contain one or more Child FlexCards. The consultant can use a Parent FlexCard to display the account information, including account name, phone, and website. A Child FlexCard is a FlexCard that can display data and actions in a card format within a Parent FlexCard. The consultant can use multiple Child FlexCards to display the active opportunities and contracts related to the account. A Card Action is a button or a link that can invoke an OmniScript or an Integration Procedure from a FlexCard.The consultant can use multiple Card Actions to enable the user to view and renew contracts

Question 3

Question Type: MultipleChoice

A business wants to display cases for an account using FlexCards. The cases should be organized in a tabular list by case status, so that Escalated cases appear first, followed by Active and Closed cases.

Which FlexCard feature should the consultant recommend to meet this requirement?

Options:

A- flyout with child FlexCards with conditions to filter by case status
B- States with conditions to filter by case status
C- A Datatable element with the Group By property
D- Child FlexCards in a parent FlexCard with the Group By property

Answer:

C

Explanation:

The FlexCard feature that the consultant should recommend to meet this requirement is a Datatable element with the Group By property. A Datatable element is an element that can display data in a tabular format, with columns and rows. The consultant can use a Datatable element to display the cases for an account, with columns for case status, case number, subject, etc. The Group By property is a property that allows grouping data by one or more fields, such as case status.The consultant can use the Group By property to organize the cases by case status, so that Escalated cases appear first, followed by Active and Closed cases

Question 4

Question Type: MultipleChoice

A company has a requirement to create a 360 view of their customers using FlexCards. At this company, customer data is stored in Salesforce but also in external legacy systems. A consultant reviews the use cases needed and recommends a FlexCard canvas that contains multiple child cards inside the state of a parent FlexCard.

What is the advantage of embedding multiple child cards in this scenario?

Options:

A- Enables easier global styling changes
B- Allows multiple data sources to be used
C- Enables the FlexCard to be used on a community page
D- Allows conditional views with the Toggle element on each child card

Answer:

D

Explanation:

The advantage of embedding multiple child cards in this scenario is that it allows multiple data sources to be used. A child card can have its own data source independent of the parent card or other child cards.This way, the consultant can use different data sources for each child card, such as Salesforce objects or external systems, and display them in one FlexCard canvas

Question 5

Question Type: MultipleChoice

Which element allows a user to retrieve data from a single field and display it in a dropdown list?

Options:

A- Calculation Action
B- DataRaptor Extract Action
C- Lookup
D- Select

Answer:

C

Explanation:

The element that allows a user to retrieve data from a single field and display it in a dropdown list is Lookup. Lookup allows the user to search for a value from a Salesforce object or an external system and select it from a list of suggestions. Calculation Action is used to perform calculations on data. DataRaptor Extract Action is used to retrieve data from multiple fields or records. Select is used to display a list of predefined values.

Question 6

Question Type: MultipleChoice

A company has deployed an OmniScript that is working as designed. During the process, the OmniScript has multiple steps that allow the user to review lists of cases, contacts, quotes, orders, and contracts for an accour

Eachstep uses DataRaptor Extract Actions to retrieve Salesforce data. However, the OmniScript is reaching performance limitations, and the development team warns that they may begin hitting Governor limits.

Which two suggestions should the consultant propose to address this concern?

Choose 2 answers

Options:

A- Replace the DataRaptor Extracts with DataRaptor Turbo Extracts
B- Add Limit filter in DataRaptor Extract
C- Combine the DataRaptors into an Integration Procedure
D- Add Order By filter in DataRaptor Extract

Answer:

A, C

Explanation:

To improve the performance and avoid hitting Governor limits, the consultant should replace the DataRaptor Extracts with DataRaptor Turbo Extracts and combine them into an Integration Procedure. DataRaptor Turbo Extracts are optimized for speed and efficiency, and can retrieve large amounts of data without using SOQL queries.An Integration Procedure can execute multiple DataRaptor actions in one call, reducing the number of requests and processing time

Question 7

Question Type: MultipleChoice

A business process requires the user to upload a file in the OmniScript. The file ID needs to be linked to the parent record being created.

Which approach ensures the link is established correctly?

Options:

A- The Data Mapper Load that creates the parent record should output its ID.
B- OmniScripts cannot handle file uploads.
C- Use a Data Mapper Extract to find the file on the user's computer.
D- The file upload happens automatically without any ID linking.

Answer:

A

Explanation:

The parent record's Salesforce ID is required to create the correct relationship between the uploaded file and the newly created parent record. Therefore, the Data Mapper Load that creates the parent record must return/output the new record ID so the OmniScript or subsequent data action can use it to link the file properly. OmniScripts can support file upload use cases, so option B is false. A Data Mapper Extract cannot search the user's local computer; it retrieves Salesforce data. The link is not guaranteed automatically in every multi-step process unless the parent record ID is available and passed into the file-linking step. Salesforce Data Mapper Load examples show that linking related records requires carrying the created parent ID into the related mapping/linking process.

==========

Question 8

Question Type: MultipleChoice

A company needs to create a quoting process for its internal agents. During quoting the agent select a product that is passed to the Omniscript with product details including the unit price, and specifies the grade (A, B, C, D, E)

size (Small, Medium, Large, X-Large). The process should use the grade and size to look up a discount factor, then multiply the unit price by that discount to retumn the quote.

Which two tools should the to meet these

Choose 2 answers

Options:

A- Data Mapper Transform
B- Expression Set
C- Decision Matrix

D- Data Mapper Extract

Answer:

C, D

Explanation:

The requirement involves creating a quoting process within an OmniScript where an agent selects a product (with a unit price), specifies grade and size, looks up a discount factor based on those inputs, and calculates the final quote. Two OmniStudio tools are needed to achieve this: Decision Matrix and DataRaptor Extract.

Here's why C. Decision Matrix and D. DataRaptor Extract are the correct answers:

C . Decision Matrix:

Purpose: A Decision Matrix in OmniStudio is a tool used to look up values based on multiple input conditions, returning a result that can be used in calculations or processes. It's essentially a configurable lookup table stored as a Salesforce custom object (DecisionMatrixDefinition).

Application: In this scenario, the Decision Matrix is ideal for determining the discount factor based on the combination of grade (A, B, C, D, E) and size (Small, Medium, Large, X-Large). For example:

Grade A + Small = 0.9 (10% discount)

Grade B + Large = 0.85 (15% discount)

And so on for all 20 combinations (5 grades 4 sizes).

Integration with OmniScript: The OmniScript can call the Decision Matrix via an Integration Procedure or directly reference it in a Calculation Action, passing Grade and Size as inputs and receiving the DiscountFactor as output.

Why It Fits: The requirement explicitly states ''look up a discount factor,'' and Decision Matrix is purpose-built for such multi-variable lookups, making it more efficient than manual coding or other tools.

D . DataRaptor Extract:

Purpose: A DataRaptor Extract retrieves data from Salesforce objects and maps it into a JSON format usable by OmniScripts or other components.

Application: The product details, including the UnitPrice, are passed to the OmniScript when the agent selects a product. A DataRaptor Extract is needed to fetch this product data from a Salesforce object (e.g., Product2 or a custom object like QuoteLineItem__c) based on the selected product's ID. The extracted data (e.g., { 'ProductId': '01t...', 'UnitPrice': 100 }) is then available in the OmniScript's data JSON.

Role in Calculation: After the Decision Matrix provides the discount factor, the OmniScript can

use a Calculation Action to multiply the UnitPrice (from the DataRaptor Extract) by the DiscountFactor (from the Decision Matrix) to compute the quote (e.g., 100 * 0.9 = 90).

Why It Fits: The process starts with product selection, and DataRaptor Extract is the standard OmniStudio tool for retrieving Salesforce data like unit price.

How They Work Together:

DataRaptor Extract: Fetches product details (e.g., UnitPrice) when the agent selects a product.

Decision Matrix: Looks up the discount factor based on Grade and Size inputs.

Calculation Action in OmniScript: Multiplies UnitPrice by DiscountFactor to return the quote.

Now, let's examine why the other options are incorrect:

A . DataRaptor Transform: A DataRaptor Transform manipulates or reshapes data (e.g., converting JSON structures or applying formulas) but doesn't retrieve data from Salesforce or perform lookups like a Decision Matrix. While it could theoretically calculate the quote after data is fetched, it's not needed here since a Calculation Action within OmniScript can handle the multiplication, and it doesn't address the lookup requirement.

B . Expression Set: An Expression Set defines reusable formulas or conditions in OmniStudio, often used in Integration Procedures or Calculations. While it could compute the final quote (e.g., UnitPrice * DiscountFactor), it doesn't retrieve data or perform lookups based on grade and size. It's a supporting tool, not a primary solution for this requirement.

Salesforce OmniStudio Documentation: Decision Matrix Overview -- Describes using Decision Matrices for multi-condition lookups.

Salesforce OmniStudio Developer Guide: DataRaptor Extract -- Details retrieving Salesforce data for OmniScript use.

Question 9

Question Type: MultipleChoice

An Integration Procedure calls a REST API to retrieve weather data. The API requires certain parameters, such as the city and country, to be passed as query parameters in the URL.

How should the OmniStudio Consultant manage the configuration of these parameters in the HTTP Action?

Options:

A- Define the parameters only in the HTTP Action Request Body, regardless of the GET method.

B- Input the parameters as Query Parameters within the HTTP Action properties, which automatically appends them to the endpoint URL.
C- Use a Data Mapper Load action to send the parameters directly to the external API, bypassing the HTTP Action.
D- Manually concatenate the parameters into the endpoint URL string using a Set Values action.

Answer:

B

Explanation:

HTTP Actions provide native support for query parameters and are designed to handle REST endpoint construction correctly. When the external service expects values such as city and country in the URL query string, the consultant should configure them through the HTTP Action's Query Parameter settings. This approach improves maintainability, readability, and parameter management while avoiding manual URL construction errors. Request bodies are generally associated with POST, PUT, or PATCH operations rather than standard GET query-string requirements. A Data Mapper Load writes data to Salesforce and cannot replace an HTTP Action for external REST communication. Manual string concatenation is technically possible but is not a best practice because it creates maintenance challenges and increases the risk of malformed URLs. Integration Procedures are intended to orchestrate such external-service interactions efficiently.

==========

Question 10

Question Type: MultipleChoice

What is the purpose of a DataRaptor Load?

Options:

A- Write data to Salesforce objects
B- Post data to Salesforce APIs
C- Load data to an Interface object
D- Send data to a PDF template

Answer:

A

Explanation:

The purpose of a DataRaptor Load is to write data to Salesforce objects. A DataRaptor Load can map data from an Interface object or a JSON object to one or more Salesforce objects, and perform insert, update, upsert, or delete operations.A DataRaptor Load can also use formulas and functions to transform the data before writing it to Salesforce

P2P exams