ATMIC NETWORKS
YOUR GUIDE TO GLOBAL CERTIFIATION

"Please note that these files may not be up to date. However, the questions will help you understand the exam format and typical question patterns."

www.atmicnetworks .com

Warning: Keep connected with our support team for latest updates

Question: 1

Which three use cases should be implemented using Calculation Procedures and matrices? Choose 3 answers

A. Use a house's address, size, and age of the building to determine an insurance premium.
B. Use rules to determine eligible insurance products based on a house's address and age of the building.
C. Use location and past usage to determine the monthly cost for an energy product.
D. Use the product color and capacity to determine the price of a product.
E. Use risk factors for an insured item to determine different insurance product options.

Answer: A,C,E

Explanation:

Calculation Procedures & Matrices are used to perform complex calculations based on input data and predefined rules. They are suitable for use cases that involve pricing, rating, scoring, or eligibility determination. Therefore, the use cases that should be implemented using Calculation Procedures & Matrices are:

Use a house's address, size, and age of the building to determine an insurance premium. (This involves rating based on multiple factors)

Use location and past usage to determine the monthly cost for an energy product. (This involves pricing based on variable inputs)

Use risk factors for an insured item to determine different insurance product options. (This involves scoring and eligibility based on criteria)

The use cases that should not be implemented using Calculation Procedures & Matrices are:

Use rules to determine eligible insurance products based on a house's address and age of the building. (This can be done using Business Rules Engine or OmniScript logic)

Use the product color and capacity to determine the price of a product. (This can be done using simple formulas or lookup tables)

Question: 2

When designing OmniScripts, which three best practices should consultants recommend to increase user adoption? Choose 3 answers

A. Prefill data for users when possible
B. Replicate existing processes as-is
C. Provide keystroke commands for data entry
D. Divide complex processes into sections
E. Provide user help text

Answer: A,D,E

Explanation:

When designing OmniScripts, the best practices that should be followed to increase user adoption are:
Prefill data for users when possible (This reduces user effort and improves data quality)
Divide complex processes into sections (This makes the script more manageable and user-friendly)
Provide user help text (This guides the user through the script and clarifies any doubts)
The best practices that should not be followed to increase user adoption are:
Replicate existing processes as-is (This may not leverage the full potential of OmniStudio and may not address the pain points of the users)
Provide keystroke commands for data entry (This may not be intuitive or accessible for all users and may require additional training)

Question: 3

A company has a legacy application to display customer information. The application currently uses custom CS / HTML to display information in the company's color scheme and fonts. The application also provides users will access to more than 25 processes. Recently, a new project was started to build a 360° view using FlexCards to replace the legacy application.
In this scenario, which three FlexCard features should the consultant recommend?
Choose 3 answers

A. Actions
B. Custom Styles
C. Menu elements
D. Data tables
E. Newport Design System

Answer: A,B,D

Explanation:
FlexCards are used to display contextual customer information in a compact and customizable way. They can also provide access to related processes or actions. In this scenario, the features that the consultant should recommend are:
Actions: These are buttons or links that can launch OmniScripts, DataRaptors, or other processes from the FlexCard. They can help the users to perform tasks related to the customer information displayed on the FlexCard.
Custom Styles: These are CSS classes that can be applied to the FlexCard elements to match the company's color scheme and fonts. They can help the users to have a consistent and branded user interface.
Data Tables: These are elements that can display tabular data from one or more data sources on the FlexCard. They can help the users to see relevant data in a structured and sortable way.
The features that the consultant should not recommend are:
Menu Elements: These are elements that can display a list of options or submenus on the FlexCard.
They are not suitable for this scenario because they do not provide access to processes or actions, but only to other FlexCards or pages.
Newport Design System: This is a design system that provides a set of predefined styles and components for OmniStudio applications. It is not suitable for this scenario because it does not match the company's color scheme and fonts, and it may

require additional customization.

Question: 4

What is the purpose of Step elements in OmniScript?

A. Allows the user to input data
B. Groups elements that extract data
C. Enables the use of repeatable blocks
D. Organizes the script into one or more pages

Answer: D

Explanation:

Step elements are used to organize the OmniScript into one or more pages. They define the layout, navigation, and visibility of the OmniScript elements. They can also have pre- and post-actions that can execute DataRaptors, Integration Procedures, or other processes before or after the user completes the step.
The other options are not correct because:
Allows the user to input data: This is not the purpose of Step elements, but of Input elements, such as Text, Number, Date, Picklist, etc.
Groups elements that extract data: This is not the purpose of Step elements, but of Data Source elements, such as DataRaptor Extract, Integration Procedure Extract, etc.
Enables the use of repeatable blocks: This is not the purpose of Step elements, but of Repeatable Block elements, which allow the user to add or remove multiple instances of a group of elements.

Question: 5

Which two functions can be performed by DataRaptors?
Choose 2 answers

A. Transform data
B. Combine requests into a single response
C. Read and write data to external systems
D. Read and write data to Salesforce

Answer: A,D

Explanation:

DataRaptors are OmniStudio data tools that can perform various functions on data, such as reading, writing, transforming, and validating. They can work with Salesforce data and external data sources. The two functions that can be performed by DataRaptors are:
Transform data: DataRaptors can use transformation rules to manipulate data, such as changing the format, applying calculations, mapping values, etc.
Read and write data to Salesforce: DataRaptors can use SOQL or SOSL queries to read data from Salesforce objects and fields,

and use DML operations to insert, update, delete, or upsert data to Salesforce.
The functions that cannot be performed by DataRaptors are:
Combine requests into a single response: DataRaptors cannot aggregate multiple requests from different sources into one response. This can be done using Integration Procedures, which can orchestrate multiple DataRaptors or REST calls into a single process.
Read and write data to external systems: DataRaptors cannot directly access external systems or APIs. They need to use Integration Procedures or REST elements to invoke external services and pass the data to or from DataRaptors.

Question: 6

A company has an existing OmniScript that agents use to create new billing accounts. It currently has three steps to capture the required information:

- Step 1: account name and legal number
- Step 2: billing cycle and monthly due date
- Step 3: email and telephone number

After each step, a DataRaptor is used to update the account information in Salesforce.
Following best practices, which two improvements can the consultant recommend for this OmniScript?
Choose 2 answers

A. Create a reusable OmniScript for this process
B. Use a single DataRaptor to save the information
C. Combine the three steps into one step
D. Configure the save for later property

Answer: B,C

Explanation:

When designing OmniScripts, the best practices that should be followed to optimize performance and user experience are:
Use a single DataRaptor to save the information: Instead of using multiple DataRaptors to update the account information after each step, it is better to use a single DataRaptor at the end of the OmniScript to save all the information at once. This reduces the number of database operations and improves performance.
Combine the three steps into one step: Instead of having three separate steps to capture the required information, it is better to combine them into one step with a simple layout. This reduces the number of page transitions and improves user experience.
The suggestions that are not best practices are:
Create a reusable OmniScript for this process: This may not be necessary if the process is specific to creating new billing accounts and not used by other OmniScripts. Creating a reusable OmniScript may add complexity and maintenance overhead.
Configure the save for later property: This may not be useful if the process is short and simple and does not require the user to resume it later. Configuring the save for later property may add unnecessary storage and retrieval logic.

Question: 7

Agents for an insurance company need to know the current and past weather conditions when creating claims for customers. The consultant implements a FlexCard for weather conditions in the console to ensure the agents have access to the information. The FlexCard needs to provide fields extracted from a weather API and an account field from

Salesforce.
According to best practice, what data source should be used?

A. Streaming API
B. DataRaptor Extract
C. REST
D. Integration Procedure

Answer: D

Explanation:

An Integration Procedure is the best data source to use for a FlexCard that needs to provide fields from both an external API and a Salesforce object. An Integration Procedure can combine multiple DataRaptor actions, such as Extract, Transform, and Load, and also invoke REST or SOAP services. This way, the FlexCard can display data from different sources in one place

Question: 8

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 dat
a. 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
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: 9

How many levels deep can reusable OmniScripts be nested?

A. 2
B. 5
C. Unlimited
D. 1

Answer: D

Explanation:

Reusable OmniScripts are OmniScripts that can be embedded within other OmniScripts to create modular and reusable components. They can be nested up to unlimited levels deep, as long as there are no circular references or conflicts between the parent and child OmniScripts.
The other options are not correct because:
2: This is too low and does not reflect the full potential of reusable OmniScripts.
5: This is also too low and does not reflect the full potential of reusable OmniScripts.
1: This is incorrect and does not make sense, as reusable OmniScripts are meant to be nested within other OmniScripts.

Question: 10

A company needs to create a new quote estimation process for its services team. This process involves multi; steps. The user needs to enter a variable number of rows of data in order to calculate the correct final price f the quote. The data that needs to be entered in each row includes:

- Service category
- Number of hours
- Hourly rate

The number of hours should be multiplied with the hourly rate to calculate the total for each row. Which three OmniScript elements should the consultant recommend to meet these requirements? Choose 3 answers

A. Edit Block
B. Multi-select
C. Formula
D. Step
E. Input Block

Answer: A,C,E

Explanation:

To create a new quote estimation process that involves multiple steps and a variable number of rows of data, the consultant should recommend the following OmniScript elements:
Edit Block: This is an element that allows the user to add, edit, or delete multiple rows of data in a table-like format. It can be used to capture the service category, number of hours, and hourly rate for each row.
Formula: This is an element that allows the user to perform calculations on data using mathematical expressions. It can be used to calculate the total for each row by multiplying the number of hours with the hourly rate.
Input Block: This is an element that allows the user to enter data using various input types, such as text, number, date, picklist, etc. It can be used to display the formula result for each row and the final price for the quote.

The elements that are not suitable for this scenario are:
Multi-select: This is an element that allows the user to select multiple options from a list. It is not relevant for this scenario, as the user needs to enter data in each row, not select from predefined options.
Step: This is an element that organizes the OmniScript into one or more pages. It is not relevant for this scenario, as it does not capture or display any data.

Question: 11

A consultant must design a 360 view of the customer. The business requirements are:

- A header card with account information (name, account number, next billing date, invoice method)
- A list of related contacts (first name, last name, phone)
- All the open cases related to the account (subject, priority, SLA)

An account will not have more than 2 contacts, but it could have more than 10 open cases. It is necessary to the different sections available at a glance.
Which two FlexCard features should the consultant recommend to improve the user experience?
Choose 2 answers

A. Use a Datatable element
B. Use a Block Element with the Collapse property enabled
C. Use a Zone Template
D. Use a Custom Style to adjust height and width

Answer: B,C

Explanation:

To improve the user experience, the consultant should use a Block Element with the Collapse property enabled and a Zone Template. A Block Element with the Collapse property enabled allows the user to expand or collapse a section of the FlexCard, which is useful for displaying a list of related records. A Zone Template allows the consultant to define the layout and style of the FlexCard, such as the number of columns, rows, and margins

Question: 12

Which three of the following are functions of a Text Block element in OmniScript?
Choose 3 answers

A. Error conditions
B. User input fields
C. Links or images
D. Text formatting
E. Table formatting

Answer: C,D,E

Explanation:

A Text Block element in OmniScript can perform three functions: display links or images, apply text formatting, and format tables. A Text Block element can use HTML tags to create hyperlinks or embed images in the OmniScript. It can also use CSS properties to style the text, such as font size, color, alignment, etc. Additionally, a Text Block element can use HTML table tags to create and format tables in the OmniScript

Question: 13

A business needs to create a FlexCard to display open cases for an account. The cases should display different depending on their priority.

- Low or Medium priority cases should display case data and an Update Account Info action.
- High priority cases should include a red border, an alert notification, and an Escalate action.

What is the most efficient way to meet these requirements?

A. Create a FlexCard with two card states. Use conditions on the state to filter for priority.
B. Create a FlexCard with one card state. Use conditions on the fields and actions to filter for priority.
C. Create two FlexCards with one card state each. Use conditions on the states to filter for priority.
D. Create two FlexCards, one for Low / Medium priority cases and another for High priority cases.

Answer: A

Explanation:

The most efficient way to meet the requirements is to create a FlexCard with two card states. A card state defines how a FlexCard displays data and actions based on certain conditions. The consultant can use conditions on the state to filter for priority, and then customize the fields and actions for each state. This way, the FlexCard can display different information and options for low/medium and high priority cases

Question: 14

When designing OmniStudio solutions, what are two reasons that explain why Integration Procedures are recommended as the first choice of data sources for FlexCards and OmniScripts? Choose 2 answers

A. They future-proof your front-end design.
B. They can retrieve Salesforce data from multiple objects faster.
C. They combine multiple actions into a single server call.
D. They enable better usability in digital interactions.

Answer: A,C

Explanation:

The two reasons that explain why Integration Procedures are recommended as the first choice of data sources for FlexCards and OmniScripts are: they future-proof your front-end design and they combine multiple actions into a single server call. An Integration Procedure can abstract the data source from the front-end design, making it easier to change or update the data source without affecting the FlexCard or OmniScript. An Integration Procedure can also execute multiple DataRaptor actions,

such as Extract, Transform, and Load, and also invoke REST or SOAP services in one call, reducing the number of requests and processing time

Question: 15

A business wants to create an OmniScript that allows call center agents to schedule field service appointments with customers. The process needs to retrieve available appointment dates from an external system via a REST A [and then display them to the user for selection in a dropdown list.

Once the user selects a date, a confirmation should display with rich text and images. Which three OmniScript elements should be used to meet these requirements? Choose 3 answers

A. Text Block
B. HTTP Action
C. Text Area
D. Select
E. Multi-select

Answer: A,B,D

Explanation:

The three OmniScript elements that should be used to meet the requirements are: Text Block, HTTP Action, and Select. A Text Block element can display rich text and images using HTML tags in the OmniScript. An HTTP Action element can invoke a REST API and store the response in a JSON object. A Select element can display a dropdown list of options for the user to choose from

Question: 16

A company needs to create an OmniScript with three steps.

- Step 1
- Step 2
- Step 3

Enter customer details
Select a product
Display a price retrieved from a REST API
The product's price should automatically display in Step 3.
What should the consultant add to the design of the OmniScript to meet these requirements?

A. DataRaptor Post Action between Step 2 and Step 3
B. Integration Procedure Action between Step 2 and Step 3
C. DataRaptor Post Action inside Step 2
D. Integration Procedure Action inside Step 3

Answer: D

Explanation:

The consultant should add an Integration Procedure Action inside Step 3 to meet the requirements. An Integration Procedure Action can execute an Integration Procedure that invokes a REST API and retrieves the product price. The Integration Procedure Action can be configured to run automatically when Step 3 is loaded, and display the price in a field or label element

Question: 17

An existing OmniScript used to capture and update customer information displays the following information on the same page:

- Name
- Age
- Street
- Last name
- Postal code
- Gender
- State
- Phone
- City
- Country
- Email

Users report that the information displays in no specific order. Users experience errors when filling the information because it is not categorized. The process fails after submission because the mandatory fields are blank.
How can the consultant improve the user experience of the OmniScript?

A. Use a template for each category and add custom code for the required validation
B. Use a Section to break the information and Headline to the title and a required icon in the input
C. Use a Visualforce Page with the categories, labels, and the required validation
D. Use Block elements to group and required property checked in some of the inputs

Answer: D

Explanation:

The consultant can improve the user experience of the OmniScript by using a Section element to break the information and a Headline element to add the title and a required icon in the input. A Section element can group related fields and elements in a logical way, making it easier for the user to navigate and fill the information. A Headline element can display a label and an icon for each field, indicating which ones are mandatory. This way, the OmniScript can display the information in a specific order and prevent errors and failures

Question: 18

A telecommunications company wants to create a 360° view of their customers, including all customer install products grouped by service address. During the discovery phase of the project, the consultant identifies that installed products can have 4 different statuses. Agents complete different processes, depending on status of installed product. The consultant creates an inventory of all statuses and processes as shown below.

- Status = Active

o Pay bill

o Review usage

o Cancel service

- Status = Suspended

o Pay bill

o Reconnect service

o Cancel service

- Status = Pending installation

o Set up billing

o Schedule installation

o Cancel installation

- Status = Canceled

o Reconnect service o Cancel service

- Status = Pending installation

o Set up billing

o Scheduleinstallation

o Cancel installation

- Status = Canceled

o Pay bill

o Disconnect service

o Reconnect service

Which two FlexCard features should the consultant use to meet these requirements? Choose 2 answers

A. States
B. OmniStudio Action
C. Custom Style
D. Flyout

Answer: A,B

Explanation:

The two FlexCard features that the consultant should use to meet these requirements are States and OmniStudio Action. A State is a condition that determines how a FlexCard displays data and actions based on certain criteria. The consultant can use States to filter the installed products by their status and display different fields and actions accordingly. An OmniStudio Action is a button that can invoke an OmniScript or an Integration Procedure from a FlexCard. The consultant can use OmniStudio Actions to enable the agents to complete different processes for each installed product, such as pay bill, cancel service, etc

Question: 19

A business needs a 360° view of their accounts, including a FlexCard to display all of the products sold to the account. The business identified 20 different data elements and 10 actions that users would need when view the product information. Once all of the elements are collected together on the FlexCard, it looks cluttered.
Which two FlexCard features should the consultant recommend to address this issue?
Choose 2 answers

A. Use a table or chart to display data
B. Use a flyout action to display additional information
C. Use a condition to hide data unless specific criteria are met
D. Put specific fields in a collapsible block

Answer: B,D

Explanation:

The two FlexCard features that the consultant should recommend to address this issue are Flyout and Block elements. A Flyout is a pop-up window that can display additional information or actions for a FlexCard. The consultant can use a Flyout to show more details or options for each product without cluttering the main FlexCard. A Block element is a container that can group related fields and elements in a FlexCard. The consultant can use a Block element with the Collapse property enabled to allow the user to expand or collapse a section of the FlexCard, such as specific fields or actions

Question: 20

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 5 child FlexCards inside the state of the parent FlexCard.
How many different data sources can be configured using FlexCards in this scenario?

A. 2
B. 6
C. 5
D. 1

Answer: D

Explanation:

The number of different data sources that can be configured using FlexCards in this scenario is 6. A FlexCard can have one data source per state, and a state can have multiple child FlexCards, each with its own data source. Therefore, the parent FlexCard can have one data source for its state, and each of the 5 child FlexCards can have a different data source, making a total of 6 data sources

Question: 21

A company plans to rebuild a process that includes functionality that was originally written in APEX and Visual pages using OmniScript. The consultant reviews the process and sees in the first step, the user reviews data and, then enters additional information. In the next step, the process retrieves data from an external system.

In this scenario, which OmniStudio data tool should the consultant recommend?

A. DataRaptor Turbo
B. Calculation Procedure
C. Integration Procedure
D. DataRaptor Extract

Answer: C

Explanation:

The OmniStudio data tool that the consultant should recommend in this scenario is an Integration Procedure. An Integration Procedure can execute multiple DataRaptor actions, such as Extract, Transform, and Load, and also invoke REST or SOAP services. In this case, the Integration Procedure can retrieve data from an external system via a REST API and store it in a JSON object for the OmniScript to display

Question: 22

A business has a requirement to display an account and all of the associated contacts on a page. The number of contacts will vary for each account. For each contact, the page should display first name, last name, email, at phone number with options to edit the contact information or send a message. The primary contact for an a should be highlighted with a blue border.
Which two FlexCards features should the consultant recommend to meet these requirements?
Choose 2 answers

A. Data table
B. Flyouts
C. States
D. Repeat Block

Answer: A,D

Explanation:

The two FlexCard features that the consultant should recommend to meet these requirements are Datatable and Repeat Block. A Datatable element can display a list of records in a tabular format, with columns for each field and rows for each record. The consultant can use a Datatable element to display the contacts for an account, with options to edit or send a message. A Repeat Block element is a container that can display multiple instances of a FlexCard based on a data source. The consultant can use a Repeat Block element to display a FlexCard for each contact, and use a condition to apply a blue border for the

primary contact

Question: 23

What is the purpose of a DataRaptor Load?

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

Question: 24

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?

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: 25

An insurance company wants to create an OmniScript that allows the user to review and change account number such as phone number and website. In this process, the following functionality is needed:

- Enter the company's website

- Enter the account phone number

- Each field should display on a separate line of the page

Which three elements should the consultant include in the OmniScript design solution?
Choose 3 answers

A. Number
B. Text Area
C. Telephone
D. Line Break
E. Text

Answer: C,D,E

Explanation:

The three elements that the consultant should include in the OmniScript design solution are: Telephone, Line Break, and Text. A Telephone element can display a field for the user to enter a phone number, with validation and formatting options. A Line Break element can insert a line break between two elements, making them display on separate lines. A Text element can display a field for the user to enter text, such as a website URL

Question: 26

What is the purpose of the Messaging element in OmniScript?

A. Display a custom styled message
B. Send email/text to users based on logical conditions
C. Trigger an event to external data sources based on logical conditions
D. Display warning to users based on logical conditions

Answer: D

Explanation:

The purpose of the Messaging element in OmniScript is to display a warning to users based on logical conditions. A Messaging element can show a message with an icon and a color, such as red for error or green for success. The message can be configured to appear or disappear based on certain criteria, such as field values or user actions

Question: 27

An insurance company decides to use calculation procedures and matrices to calculate premium costs for new Insurance policies. Prices change very frequently, resulting In multiplecopies of the rating or pricing tables.
What is an advantage of calculation procedures that the consultant should highlight in this scenario?

A. Allows aggregate functions
B. Allows text concatenation using algebraic operators

C. Allows multiple versions that will execute based on when the request is made
D. Allows AI integration to calculate next best offer using policy attributes

Answer: C

Explanation:

An advantage of calculation procedures that the consultant should highlight in this scenario is that they allow multiple versions that will execute based on when the request is made. A calculation procedure can have different versions with different effective dates, which determine when they are active or inactive. This way, the calculation procedure can use different rating or pricing tables depending on the date of the request, and handle frequent changes in prices

Question: 28

A business has a requirement to display cases in a console for service agents. Cases can have avariety of statuses, including Active, Closed, or Escalated. When a case is Closed, agents need to be able to reopen the case. When the case is Active or Escalated, agents should not have the option to reopen the case.
Which FlexCard functionality can be used to meet this requirement?

A. Conditional View
B. Flyouts
C. State
D. Styling

Answer: A

Explanation:

The FlexCard functionality that can be used to meet this requirement is Conditional View. A Conditional View is a property that determines whether a FlexCard element, such as a field or an action, is visible or hidden based on certain conditions. The consultant can use a Conditional View to show or hide the reopen case action based on the status of the case

Question: 29

In which two scenarios should a DataRaptor be used?
Choose 2 answers

A. To retrieve an account and Its related contacts
B. To retrieve today's financial news
C. To send an SMS alert to the user when a process completes

D. To merge the city and state fields into a single field

Answer: C,D

Explanation:

The two scenarios where a DataRaptor should be used are: to retrieve an account and its related contacts, and to merge the city and state fields into a single field. A DataRaptor is a tool that can read, transform, and write data from Salesforce objects or JSON objects. A DataRaptor Extract can retrieve data from one or more Salesforce objects using SOQL queries, and store it in an Interface object or a JSON object. A DataRaptor Transform can map data from one Interface object or JSON object to another, and apply formulas and functions to transform the data. For example, a DataRaptor Transform can concatenate two fields into one using the + operator

Question: 30

A company is creating a FlexCard for agents to respond to account inquiries. The company has decided to lim actions on the card to the three most frequent and important actions users typically take.
Based on the user analysis exhibit shown below, which three actions should appear on the card?

Choose 3 answers

A. Verify a transaction
B. Request higher limit
C. View credit score history
D. View statements
E. Make a payment

Answer: A,B,E

Explanation:

Based on the user analysis exhibit, the three actions that should appear on the card are: verify a transaction, request higher limit, and make a payment. These actions have the highest frequency and importance scores among the six actions listed. The other actions, such as view credit score history, view statements, and view rewards, have lower frequency and importance scores, and can be accessed through other means

Question: 31

A customer needs to create an OmniScript to capture payment Information. In the first step of the process, it user selects a payment type such as credit card, debit card, or direct bank account payment. Each payment t should display with a different icon. The process should check to see if the customer has any saved payment information of that type, and if there is, no further action is

required. If there is no saved payment informatic then the user should be allowed to enter the information, and the process should save It.
Which three OmniScript elements should be used to meet these requirements?
Choose 3 answers

A. DataRaptor Transform Action
B. Select
C. DataRaptor Extract Action
D. Radio
E. DataRaptor Post Action

Answer: B,D,E

Explanation:

The three OmniScript elements that should be used to meet these requirements are: Select, DataRaptor Extract Action, and DataRaptor Post Action. A Select element can display a dropdown list of options for the user to choose from, such as payment type. A DataRaptor Extract Action can retrieve data from a Salesforce object, such as saved payment information, and store it in an Interface object or a JSON object. A DataRaptor Post Action can write data to a Salesforce object, such as new payment information, and perform insert or update operations

Question: 32

Which OmniScript element can be used to retrieve data from more than one source?

A. Calculation Action
B. Integration Procedure Action
C. DataRaptor Extract Action
D. Matrix Action

Answer: B

Explanation:

The OmniScript element that can be used to retrieve data from more than one source is Integration Procedure Action. An Integration Procedure Action can execute an Integration Procedure that can combine multiple DataRaptor actions, such as Extract, Transform, and Load, and also invoke REST or SOAP services. This way, the Integration Procedure Action can retrieve data from different sources, such as Salesforce objects or external systems, and store it in a JSON object for the OmniScript to display

Question: 33

Which OmniStudio tool creates a Chatter post and sends to a Chatter feed?

A. DataRaptor Load
B. Integration Procedure
C. Calculation Procedure
D. FlexCards

Answer: B

Explanation:

The OmniStudio tool that creates a Chatter post and sends it to a Chatter feed is Integration Procedure. An Integration Procedure can execute multiple DataRaptor actions, such as Extract, Transform, and Load, and also invoke REST or SOAP services. One of the REST services that an Integration Procedure can invoke is Chatter API, which allows creating and sending Chatter posts to users or groups

Question: 34

An OmniScript saves data to Salesforce and to an external system. What OmniScript element could save all this data?

A. HTTP Action
B. Integration Procedure Action
C. DataRaptor Post Action
D. DataRaptor Transform Action

Answer: B

Explanation:

The OmniScript element that could save all this data is Integration Procedure Action. An Integration Procedure Action can execute an Integration Procedure that can combine multiple DataRaptor actions, such as Extract, Transform, and Load, and also invoke REST or SOAP services. This way, the Integration Procedure Action can write data to Salesforce using a DataRaptor Load action and to an external system using a REST or SOAP service

Question: 35

A business implements several OmniScripts and requests a design review to identify possible improvements. During the review, the consultant notices that each OmniScript needs to update address data, but each OmniScript collects the information differently using different labelsand input elements.
What can the consultant recommend to improve these OmniScripts?

A. Create new versions of the OmniScripts to collect the address information.

B. Create a reusable OmniScript to update address information.
C. Use a DataRaptor Transform to standardize the data format.
D. Add an address typeahead element to all the OmniScripts.

Answer: B

Explanation:

The consultant can recommend creating a reusable OmniScript to update address information. A reusable OmniScript is an OmniScript that can be embedded in another OmniScript as a sub-process. This way, the consultant can create one OmniScript

that collects the address information in a consistent way using the same labels and input elements, and then use it in multiple OmniScripts that need to update address information. This will improve the design and maintenance of the OmniScripts.

Question: 36

The design team creates a mock-up proposal of an OmniScript for sales that includes more than 25 steps. Th consultant reviews the proposal and sees that the OmniScript will be used for three different and complex sal processes: change of plan, new sale, and loyalty. The user will select the type of sale at the beginning of the interaction, and then the OmniScript will branch into either change of plan, new sales, or loyalty processes. E process will contain multiple decision points for the user. Following best practices, what should the consultant recommend to improve the planned design of this OmniScript?

A. Pass the process type as an input parameter.
B. Add more conditional views to branch the OmniScript into logical sections.
C. Add more Step elements to create shorter pages.
D. Create a specific OmniScript for each business process.

Answer: D

Explanation:

The consultant should recommend creating a specific OmniScript for each business process to improve the planned design of this OmniScript. Having one OmniScript for three different and complex sales processes will make the OmniScript too long and complicated, and will affect the performance and maintainability. Creating separate OmniScripts for each process will make them more manageable and user-friendly, and will also allow for more customization and flexibility

Question: 37

A consultant has a project with the following requirement: "Agents need to follow a standard customer greeti…. script in order to ensure brand consistency."

Which OmniScript element should the consultant recommend to ensure the agents can see and follow these standard scripts?

A. Input Block
B. Text Area
C. Headline
D. Step

Answer: C

Explanation:

The consultant should recommend using a Headline element to ensure the agents can see and follow these standard scripts. A Headline element can display a text message with an icon and a color, such as blue for information or yellow for warning. The consultant can use a Headline element to show the customer greeting script in the OmniScript, and instruct the agents to read

it aloud

Question: 38

A business is creating a new OmniScript that will allow agents to launch a guided selling process from an account detail page. The consultant reviews the initial design proposed for the OmniScript and sees that the process the following steps:

- Step 1: Enter account name to retrieve account information
- Step 2: Review and edit account information
- Step 3: Select products
- Step 4: Enter payment information

Following best practices, what suggestion should the consultant make to improve the design?

A. Remove Step 1 and prefill the account information automatically.
B. Divide Step 2 into two steps, so each task is distinct.
C. Use an Edit Block in Step 4 to enter payment information.
D. Use Multi-select elements in Step 3 for the product list.

Answer: A

Explanation:

The consultant should suggest removing Step 1 and prefilling the account information automatically to improve the design. Since the OmniScript will be launched from an account detail page, there is no need to ask the user to enter the account name again. The consultant can use an Input Parameter element to pass the account ID from the page to the OmniScript, and then use a DataRaptor Extract Action element to retrieve the account information and display it in Step 2. This will save time and avoid errors

Question: 39

A company has an existing OmniScript running in production. The business decides additional customer information is needed and wants to add more questions to the process.

What is the most efficient approach that the consultant can recommend to the business to meet this requirement?

A. Add a new Step element with Input elements
B. Add an LWC component
C. Add an Input Action to a Step
D. Add an embedded FlexCard

Answer: A

Explanation:

The most efficient approach that the consultant can recommend to the business to meet this requirement is to add a new Step element with Input elements. A Step element can group related fields and elements in an OmniScript, and display them in a page or a section. An Input element can display a field for the user to enter data, such as text, number, date, etc. The consultant can use a Step element to add a new page or section to the OmniScript, and then use Input elements to

collect the additional customer information

Question: 40

A company implements an integration procedure that is invoked from an OmniScript. The integration procedure includes a very long-running process that makes users impatient.
Which feature should the consultant recommend to Improve performance and address users' concerns?

A. Try Catch
B. Batch jobs
C. Chaining
D. Conditions

Answer: C

Explanation:

The feature that the consultant should recommend to improve performance and address users' concerns is chaining. Chaining is a property that allows an Integration Procedure Action element to run asynchronously in the background, without blocking the user interface. The consultant can use chaining to execute the long-running process in parallel with other actions or steps in the OmniScript, and notify the user when it is completed. This will improve the user experience and reduce waiting time

Question: 41

An insurance agency wants to enable its call center agents to be more efficient when handling customer inquiries. After analyzing patterns in the call logs, the agency discovers that one of the top customer requests is to find out total insurance premiums paid as well as the difference year over year. Agents need to relay this information to the customer and then send them a summary report by email. The agency decides to implement this process using OmniScript.
What three OmniScrlpts elements should be used to meet the requirements?
Choose 3 answers

A. DataRaptor Extract Action
B. Email Action
C. DataRaptor Post Action
D. Messaging Element
E. Calculation Action

Answer: A,B,E

Explanation:

The three OmniScript elements that should be used to meet the requirements are: DataRaptor Extract Action, Email Action, and Calculation Action. A DataRaptor Extract Action can retrieve data from a Salesforce object, such as total insurance premiums paid and the difference year over year, and store it in an Interface object or a JSON object. An Email Action can send an email to the customer with a summary report attached, using data from an Interface object or a

JSON object. A Calculation Action can perform mathematical operations on data from an Interface object or a JSON object, such as calculating the difference year over year

Question: 42

When a customer calls to add a new primary contact to their account, call center agents need to complete a I contact form. The agent enters the new contact information using an OmniScript and then needs to generate PDF with the contact information pre-filled that can be shared with the customer.
Which OmniStudio tool should the consultant recommend to generate the pre-filled PDF'

A. OmniStudio Action
B. Integration Procedure
C. Data Maptor
D. Calculation Procedure

Answer: C

Explanation:

The OmniStudio tool that the consultant should recommend to generate the pre-filled PDF is DataRaptor. A DataRaptor is a tool that can read, transform, and write data from Salesforce objects or JSON objects. A DataRaptor Load can write data to a PDF template using a PDF mapping file, and generate a PDF document with the contact information pre-filled. The PDF document can be stored as an attachment or a content document in Salesforce

Question: 43

A company plans to use OmniScript to digitally transform Its business. During the discovery phase of the project the team reviews all of the business processes including the Individual steps of each process. As a result of analysis, it is clear that many processes need to update contact data at different points in the process.
What solution should the consultant recommend to efficiently meet these requirements?

A. Create an Integration Procedure to update contact data with "Chain OnStep" enabled and invoke it from all OmniScripts.
B. Create an OmniScript that uses the reusable property to update contact data and embed it In other OmniScripts.
C. Create an OmniScript to update contact data and invoke it when other OmniScripts complete.
D. Create a FlexCard for contacts, and add an Update Contact Details OmniScript as an action.

Answer: B

Explanation:

The solution that the consultant should recommend to efficiently meet these requirements is to create an OmniScript that uses the reusable property to update contact data and embed it in other OmniScripts. A reusable OmniScript is an OmniScript that can be embedded in another OmniScript as a sub-process. This way, the consultant can create one OmniScript that updates contact data in a consistent way, and then use it in multiple OmniScripts that need to update contact data at different points in

the process. This will improve the design and maintenance of the OmniScripts

Question: 44

When a customer wants to buy a new device, a discounted rate can be applied after considering the number lines and internet coverageon their current plan.
Which OmniStudio tool could a consultant recommend to get the discounted rate?

A. DataRaptor Turbo
B. OmniStudio Action
C. Calculation Matrices
D. DataRaptor Transform

Answer: C

Explanation:

The OmniStudio tool that the consultant should recommend to get the discounted rate is Calculation Matrices. A Calculation Matrix is a tool that can perform complex calculations based on multiple input variables and output values. The consultant can use a Calculation Matrix to define the input variables, such as number of lines and internet coverage, and the output values, such as discounted rate. The Calculation Matrix can then apply rules and formulas to calculate the discounted rate for each combination of input variables

Question: 45

A company is designing a new console for contact center agents. The cards in the console need to display the following:

- "Open" cases with case description, case open date, case type, assigned to and priority fields. Open should be highlighted with a red border.
- "Awaiting Closure" cases with case description, last action taken date, resolution, approval reason for closure, and assigned to fields. These cases should be highlighted with a grey border.
- "Closed" cases with case description, resolution, case closed date fields with alink to duplicate cases.

All cases will be fetched using a single DataRaptor.
How should the consultant design the FlexCard solution to meet these requirements?

A. Using card session variables and a single FlexCard with multiple flyouts
B. Using card session variables and multiple FlexCards
C. Using card filter and a single FlexCard with multiple flyouts
D. Using card filter and multiple FlexCards

Answer: D

Explanation:

The consultant should design the FlexCard solution using card filter and multiple FlexCards to meet these requirements. A card filter is a property that determines which records are displayed in a FlexCard based on certain criteria. The consultant can use

card filter to display only the cases with a specific status, such as Open, Awaiting Closure, or Closed. A FlexCard is a tool that can display data and actions in a card format. The consultant can create multiple FlexCards for each case status, and customize the fields and actions accordingly. The consultant can also use custom style to apply different borders for each case status

Question: 46

A business has an existing Contact FlexCard that currently displays 5 actions. The business needs to add 3 m actions to the existing FlexCard. All 8 actions are equally important and used with the same frequency. The business wants to display the name and icon for each action. However, when reviewing the existing design, V consultant notices that the FlexCard is overloaded with actions. What FlexCard design solution should the consultant recommend?

A. Add a menue element and include all the actions
B. Create a new contact FlexCard for the new actions
C. Add a flyout to the existing FlexCard
D. Use a block element to add these new actions

Answer: A

Explanation:

The FlexCard design solution that the consultant should recommend is to add a menu element and include all the actions. A menu element can display a list of actions in a dropdown menu, with icons and labels for each action. The consultant can use a menu element to show all the 8 actions in the FlexCard, without overloading it with buttons or links. This will improve the user experience and the design of the FlexCard

Question: 47

Which OmniStudio tool is optimized for performance and minimizes configuration time?

A. Integration Procedure
B. DataRaptor Extract
C. DataRaptor Turbo Extract
D. Calculation Procedure

Answer: C

Explanation:

The OmniStudio tool that is optimized for performance and minimizes configuration time is DataRaptor Turbo Extract. A DataRaptor Turbo Extract is a tool that can retrieve data from one or more Salesforce objects using a graphical interface, without writing any SOQL queries. A DataRaptor Turbo Extract can also use filters, joins, and formulas to manipulate the data. A DataRaptor Turbo Extract is faster and more efficient than a DataRaptor Extract, and can handle large amounts of data without hitting governor limits

Question: 48

A company has account Information that they want to display to agents in a summarized view. For each account they want to display icons that allow the user to launch guided processes for frequent tasks. The processes a reused in other parts of the business.

Which three OmniStudio tools are needed to meet these requirements?

Choose 3 answers

A. FlexCards

B. Flyouts
C. OmniStudio Actions
D. Navigate Actions
E. Omni Script

Answer: A,C,E

Explanation:

The three OmniStudio tools that are needed to meet these requirements are: FlexCards, OmniStudio Actions, and OmniScript. A FlexCard is a tool that can display data and actions in a card format. The consultant can use a FlexCard to display the account information and icons for each action. An OmniStudio Action is a button that can invoke an OmniScript or an Integration Procedure from a FlexCard. The consultant can use OmniStudio Actions to launch guided processes for frequent tasks, such as change of plan, new sale, or loyalty. An OmniScript is a tool that can design customer interactions using elements and actions. The consultant can use OmniScripts to create the guided processes for each task

Question: 49

A company needs to create multiple guided processes on their public website. The processes need to be style using the corporate branding kit. The corporate branding kit Includes guidance on colors, fonts, and icons. In addition, the website that the process will be deployed to has a style guide that dictates the format for radio buttons, slide bars, and other user interface elements. The company wants both the corporate branding kit at the website's style guide to be globally implemented in these new processes.

Which three OmniStudio features and tools should the consultant recommend to meet these requirements?

Choose 3 answers

A. OmniScript
B. Custom Styles
C. OmnlOut
D. Newport Design System
E. FlexCards

Answer: A,B,D

Explanation:

The three OmniStudio features and tools that the consultant should recommend to meet these requirements are: OmniScript, Custom Styles, and Newport Design System. An OmniScript is a tool that can design customer interactions using elements and actions. The consultant can use OmniScripts to create the guided processes for the public website. A Custom Style is a feature that allows applying custom CSS properties to an OmniScript or a FlexCard element. The consultant can

use Custom Styles to style the OmniScripts using the corporate branding kit, such as colors, fonts, and icons. A Newport Design System is a tool that provides a set of predefined styles and components for building user interfaces. The consultant can use Newport Design System to style the user interface elements in the OmniScripts, such as radio buttons, slide bars, etc., according to the website's style guide

Question: 50

A business requires a solution to generate an event {platform event) using account information. All the event information is related to the customer and is stored in Salesforce.
Using an Integration Procedure, which two actions are necessary to design this solution?
Choose 2 answers

A. HTTP
B. Response
C. DataRaptor Post
D. DataRaptor Extract

Answer: C,D

Explanation:

The two actions that are necessary to design this solution using an Integration Procedure are DataRaptor Post and DataRaptor Extract. A DataRaptor Post is an action that can write data to a Salesforce object or invoke a Salesforce API. The consultant can use a DataRaptor Post action to create an event (platform event) using account information. A DataRaptor Extract is an action that can retrieve data from a Salesforce object or invoke a Salesforce API. The consultant can use a DataRaptor Extract action to get the account information from Salesforce

Question: 51

A health provider company is building a new application for its medical officers. The company wants to display medical test reports for patients. Medical test reports is acustom object, related to the Patients object. For e< patient, the company needs to list each medical test report including the report name, date requested, and d generated. From each medical test report, the user should be able to take the following actions:

- Approve
- Print for Review
- Send for Retest

Which OmniStudio tools should the consultant use to design a solution to meet these requirements?

A. Salesforce list view with custom buttons/links
B. Multiple FlexCards with single action
C. OmnlScript with multiple Step elements

D. Single FlexCard with multiple actions

Answer: D

Explanation:

The OmniStudio tools that the consultant should use to design a solution to meet these requirements are: Single FlexCard with multiple actions. A FlexCard is a tool that can display data and actions in a card format. The consultant can use a single FlexCard to display the medical test reports for each patient, including the report name, date requested, and date generated. A FlexCard can also have multiple actions, such as buttons or links, that can invoke OmniScripts or Integration Procedures. The consultant can use multiple actions to enable the user to approve, print for review, or send for retest each medical test report

Question: 52

A business wants to create a reusable OmniScript to capture customer payment information during the order process. The business decides that the first step of the payment process should include:

- Payment type (credit card or bank account)
- Payment amount

Which two elements should the consultant recommend for this step of the process?

Choose 2 answers

A. Number
B. Radio
C. Multi-select
D. Currency

Answer: B,D

Explanation:

The two elements that the consultant should recommend for this step of the process are: Radio and Currency. A Radio element can display a list of options for the user to choose from, such as payment type. A Radio element allows only one option to be selected at a time, and can also have icons for each option. A Currency element can display a field for the user to enter a currency value, such as payment amount. A Currency element can also have validation and formatting options, such as minimum and maximum values, decimal places, etc

Question: 53

A business Implements FlexCards in their customer 360° view. The business wants to add a new action to on the FlexCards that redirects users to an external web page.

Which type of action should the consultant recommend to meet this new requirement?

A. Redirect
B. Event

C. Navigate
D. URL

Answer: C

Explanation:

The type of action that the consultant should recommend to meet this new requirement is Navigate. A Navigate action is a button or a link that can redirect the user to another web page or URL. The consultant can use a Navigate action to add a new action to the FlexCards that redirects users to an external web page

Question: 54

A company is beginning their first project using Calculation Procedures & Matrices. The developers on the tea are concerned about learning the new tool and how they will test their calculation procedures. The consultant recommends that they use the built-in simulator.
In this scenario, what benefit should the consultant highlight to the developer team?

A. It aggregates data elements that have been calculated separately.
B. It verifies data types in the algebraic calculations of the calculation procedure.
C. It generates sample input and output JSON that can be easily used to build DataRaptors.
D. It allows developers to create what-if calculations.

Answer: C

Explanation:

Question: 55

A company begins a project to unify its customer data across the enterprise. After completing the discovery t analysis phases of the project, the project team recommends FlexCards as the primary solution. The stake holder of the project are excited to move forward with this recommendation. However, IT is concerned that creating FlexCards will require custom coding and advanced technical skills they do not have on their team.
Which two features should the consultant highlight to address IT's concerns?
Choose 2answers

A. The FlexCard Wizard
B. Lightning web components used in FlexCards
C. Newport Design System used in FlexCards
D. The FlexCard Designer

Answer: B,D

Explanation:

The two features that the consultant should highlight to address IT's concerns are Custom Styles and FlexCard Designer. Custom Styles are a feature that allows applying custom CSS properties to an OmniScript or a FlexCard element. Custom Styles do not require any coding skills, and can be easily configured using a graphical interface. FlexCard Designer is a tool that allows creating and editing FlexCards using a drag-and-drop interface. FlexCard Designer also does not require any coding skills, and can be used by anyone who has access to OmniStudio tools

Question: 56

A business has been experiencing a downturn in customer satisfaction due to billing Issues. The business lear that when customers are allowed to schedule an inquiry call with an agent, customer satisfaction improves.
For this reason, the business decides to create an OmniScript that asks the customer to rate their customer satisfaction using a 1-5 rating scale. If the customer satisfaction is less than 2, it should allow the customer t request a call back on a certain date and then create a case that includes a list of bills from an external system the last 5 months in the case description.
In what order should the consultant design the elements of the OmniScript to meet these requirements?

A. Radio Input, Date Input, HTTP Action, DataRaptor Post Action
B. Date Input, Radio Input, HTTP Action, DataRaptor Post Action
C. HTTP Action, DataRaptor Post Action, Date Input, Radio Input
D. HTTP Action, Radio Input, Date Input, DataRaptor Post Action

Answer: A

Explanation:

The order that the consultant should design the elements of the OmniScript to meet these requirements is: Radio Input, Date Input, HTTP Action, DataRaptor Post Action. A Radio Input element can display a list of options for the user to choose from, such as customer satisfaction rating. A Radio Input element allows only one option to be selected at a time, and can also have icons for each option. A Date Input element can display a field for the user to enter a date value, such as call back date. A Date Input element can also have validation and formatting options, such as minimum and maximum dates, calendar picker, etc. An HTTP Action element can invoke a REST or SOAP service to retrieve data from an external system, such as a list of bills from the last 5 months. An HTTP Action element can store the response data in a JSON object for further processing. A DataRaptor Post Action element can write data to a Salesforce object or invoke a Salesforce API, such as creating a case with the list of bills in the case description

Question: 57

Which OmniStudio tool can be used to invoke scheduled jobs?

A. Integration Procedure
B. DataRaptor Extract
C. DataRaptor Load
D. Calculation Procedure

Answer: A

Explanation:

The OmniStudio tool that can be used to invoke scheduled jobs is Integration Procedure. An Integration Procedure is a tool that can execute multiple DataRaptor actions, such as Extract, Transform, and Load, and also invoke REST or SOAP services. An Integration Procedure can also be scheduled to run at a specific time or frequency using Apex Scheduler or Process Builder

Question: 58

A business needs to display installed products for field service technicians on service calls using a mobile device The installed product information must be summarized so the technician can seekey details at a glance. How the technician also needs to sometimes access a list of past service dates for each product.
Which two FlexCards features should the consultant recommend to meet this requirement?
Choose 2 answers

A. Use flyouts
B. Use card states
C. Enable the Responsive property
D. Customize the styling

Answer: A,C

Explanation:

The two FlexCard features that the consultant should recommend to meet this requirement are: Flyouts and Responsive property. A Flyout is a pop-up window that can display additional information or actions for a FlexCard. The consultant can use a Flyout to show the list of past service dates for each product, without cluttering the main FlexCard. The Responsive property is a property that determines how a FlexCard adapts to different screen sizes and orientations. The consultant can enable the Responsive property to make the FlexCard display well on mobile devices, such as tablets or phones

Question: 59

Service agents must confirm customer contact information in the first step of a payment OmniScript.
Contact information includes name, telephone number, mobile number, and email. None of the

contact information on first step is required.
On the last step, after taking payment, the agent can optionally email the receipt to the customer. If the customer says yes, the agent selects a checkbox. If the agent selects the box but the email address field is empty, the process must require the user to return to the first step and enter the customer's email address.
What should the consultant recommend to meet this requirement?

A. Add a Set Errors element
B. Add a DataRaptor to retrieve the email address
C. Add a Validation Rule to the Contact object
D. Add a Conditional View to the last step

Answer: A

Explanation:

The consultant should recommend adding a Set Errors element to meet this requirement. A Set Errors element can display an error message and prevent the OmniScript from proceeding if certain conditions are not met. The consultant can use a Set Errors element to check if the email address field is empty when the agent selects the checkbox to email the receipt. If it is empty, the Set Errors element can show an error message and direct the user to return to the first step and enter the customer's email address

Question: 60

A business plans to implement new tools for their call center agents to increase efficiency and improve customer experience. The business needs to reduce new agent ramp-up time. During the discovery phase of the project, the business identifies the following requirements for the project:

- Easy access to frequent processes
- "At a glance" dashboards of customer information
- Lists of customer bills, which are stored on an external system

Which two FlexCard benefits should the consultant highlight when presenting a proposed solution? Choose 2 answers

A. Guide users through complex processes
B. Display differentactions based on context
C. Allow customers to enter bill payment information
D. Display a 360° view of the customer

Answer: B,D

Explanation:

The two FlexCard benefits that the consultant should highlight when presenting a proposed solution are: Display different actions based on context and Display a 360° view of the customer. FlexCards

can display different actions based on context, such as data values or user roles. The consultant can use this feature to show only relevant actions for each customer situation, such as scheduling an inquiry call with an agent if the customer satisfaction is low. FlexCards can also display a 360° view of the customer, by showing data from multiple sources in one card format. The consultant can use this feature to show important customer information at a glance, such as billing issues, account status, etc

Question: 61

An investment portfolio manager wants to build a console to display:

- Client information
- Profile attributes
- Investment instrument information

There are more than 20 profile attributes. Each investment instrument has 15 fields of information to display. The fields should be grouped logically.

How should the consultant design a solution to meet the requirements?

A. • A header FlexCard for client information with a Flyout for profile attributes• Child FlexCards for investment instruments with a Flyout for additional information

B. • A header FlexCard for client information and profile attributes• FlexCards for investment instrument information

C. • An OmniScript with a Step for client information• Block Elements for investment instruments

D. • A header FlexCard for client information and profile attributes• Child FlexCards for investment instruments with a Flyout for additional information

Answer: C

Explanation:

The consultant should design a solution using an OmniScript with multiple Step elements to meet the requirements. An OmniScript is a tool that can design customer interactions using elements and actions. The consultant can use an OmniScript to display client information, profile attributes, and investment instrument information. A Step element can group related fields and elements in an OmniScript, and display them in a page or a section. The consultant can use multiple Step elements to organize the information into logical groups, such as client information, profile attributes, and investment instruments

Question: 62

An insurance company decides to implement a sales console for the sales representatives that displays or provide access to customer information. The consultant reviews all the information required and determines that FlexCards would be the best solution to display part of the required information.

Which two pieces of data should the consultant recommend displaying using FlexCards' Choose 2 answers

A. Policies including type, issue date, amount
B. Terms and Conditions for each policy
C. Guided troubleshooting process
D. Customer name, title, phone, email

Answer: A,D

Explanation:

The two pieces of data that the consultant should recommend displaying using FlexCards are: Policies including type, issue date, amount and Customer name, title, phone, email. A FlexCard is a tool that can display data and actions in a card format. The consultant can use FlexCards to show important and summary information about the customer and their policies, such as name, title, phone, email, type, issue date, and amount. These pieces of data are suitable for FlexCards because they are concise and relevant for the agents

Question: 63

A business wants to create a FlexCard for mobile plans to add to their Customer 360° console application. The FlexCard needs to include the following actions:

- Start a process to retrieve plan consumption data
- Create a new case

- Open a promotions web page
- Change the SIM card

which combination should the consultant use in designing the solution?

A. Custom Event and Redirect URL
B. OmniScript and Navigate
C. Event, Navigate and Card
D. Flyout and OmniScript

Answer: B

Explanation:

The combination that the consultant should use in designing the solution is OmniScript and Navigate. An OmniScript is a tool that can design customer interactions using elements and actions. The consultant can use an OmniScript to display the mobile plans for the Customer 360° console application. A Navigate action is a button or a link that can redirect the user to another web page or URL. The consultant can use Navigate actions to add the following actions to the FlexCard: start a process to retrieve plan consumption data, create a new case, open a promotions web page, and change the SIM card

Question: 64

An auto insurance company has different rates for each state in the country. The company needs to manage the rates separately, but the formula to calculate the premiums is the same. All the input and output are the same; only the values in the rating tables differ.

Which OmniStudio tool should the consultant recommend to meet this requirement?

A. DataRaptor Transform with an interface Map
B. Aggregation Steps in a Calculation Procedure
C. Class-based Calculation Procedure
D. Grouped Calculation Matrix

Answer: D

Explanation:

The OmniStudio tool that the consultant should recommend to meet this requirement is Grouped Calculation Matrix. A Grouped Calculation Matrix is a tool that can perform complex calculations based on multiple input variables and output values within groups. The consultant can use a Grouped Calculation Matrix to define the input variables, such as state and other factors, and the output values, such as rates. The Grouped Calculation Matrix can then apply rules and formulas to calculate the rates for each state separately, but using the same formula

Question: 65

A company has an OmniScript that allows agents to schedule service calls. The first step displays the account name, primary contact name, and telephone number to the user for confirmation. On the second step, it displays available appointment slots, which are retrieved from an external service in XML and then transformed into JSON. The following actions are currently used in the OmniScript:

- DataRaptor Extract Action • HTTP Action
- DataRaptor Transform Action

Following best practices, what can the consultant recommend to reduce processing time?

A. Combine these actions into an Integration Procedure
B. Change DataRaptors to extract single objects
C. Add conditional views to the OmniScript
D. Add reusable OmniScripts for each step

Answer: A

Explanation:

The consultant should recommend adding a Set Errors element to meet this requirement. A Set Errors element can display an error message and prevent the OmniScript from proceeding if certain conditions are not met. The consultant can use a Set Errors element to check if the email address field is empty when the agent selects the checkbox to email the receipt. If it is empty, the Set Errors

element can show an error message and direct the user to return to the first step and enter the customer's email address

Question: 66

In OmniScript, how can pages or groups of fields be controlled based on the choices made by the user?

A. Add Cancel and Save options for the user
B. Create child OmniScripts for branched flow
C. Use the Conditional View property on a Step or Block element
D. Use the Radio Input property for a Step or Block element

Answer: C

Explanation:

In OmniScript, pages or groups of fields can be controlled based on the choices made by the user by using the Conditional View property on a Step or Block element. A Conditional View property is a property that determines whether a Step or Block element is displayed or hidden based on certain criteria, such as data values or user roles. The consultant can use the Conditional View property to show or hide different pages or groups of fields in the OmniScript depending on the user's input

Question: 67

which two of the following use cases are best solved using Calculation Procedures & Matrices?
Choose 2 answers

A. To apply the correct factor when determining a cost
B. To return output that is calculated differently based on the date
C. To determine the list of products to display to a customer

D. To retrieve text data and convert it to an integer

Answer: A,B

Explanation:

The two use cases that are best solved using Calculation Procedures & Matrices are: To apply the correct factor when determining a cost and To return output that is calculated differently based on the date. A Calculation Procedure is a tool that can perform complex calculations based on multiple input and output variables. A Calculation Matrix is a tool that can perform complex calculations based on multiple input variables and output values within groups. The consultant can use Calculation Procedures & Matrices to solve these use cases, because they involve applying rules and formulas to different data values and scenarios

Question: 68

A company needs to create a quoting process for its internal agents. During quoting, the agent selects a product and specifies the grade (A, B. C, D. E) and the size (Small Medium, Large, X-large). The process should look up the unit price using the product code, the grade, size, and then multiplies the unit price by a discount factor.
Which two OmniStudio tools should the consultant recommend to meet these requirements?
Choose 2 answers

A. Calculation Matrix
B. DataRaptor Transform
C. DataRaptor Extract
D. Calculation Procedure

Answer: A,C

Explanation:

The two OmniStudio tools that the consultant should recommend to meet these requirements are: Calculation Matrix and DataRaptor Extract. A Calculation Matrix is a tool that can perform complex calculations based on multiple input variables and output values. The consultant can use a Calculation Matrix to look up the unit price using the product code, the grade, size, and then multiply the unit price by a discount factor. A DataRaptor Extract is a tool that can retrieve data from a Salesforce object or invoke a Salesforce API. The consultant can use a DataRaptor Extract to get the product code, grade, size, and discount factor from Salesforce

Question: 69

which of the following are Integration Procedure Actions?

A. Email
B. OmniScript
C. PDF
D. Type Ahead

Answer: A

Explanation:

The type of action that the consultant should recommend to meet this new requirement is Email. An Email action is an action that can send an email to one or more recipients, with optional attachments and templates. The consultant can use an Email action to add a new action to the FlexCards that sends an email to the customer with a summary report attached

Question: 70

A company wants to create a new digital interaction process that allows customers to request a quote for service from a local retail energy supplier. The process requires the following actions:

- Allow the user to select one or more energy products from a list
- Get current energy usage data from an external system via an API
- Save the data back to Salesforce as a lead

Which three OmniScript elements should the consultant recommend to meet these requirements? Choose 3 answers

A. DataRaptor Post Action
B. Multi-select Input
C. Post to Object Action
D. Radio Input
E. HTTP Action

Answer: A,B,E

Explanation:

The three OmniScript elements that the consultant should recommend to meet these requirements are: DataRaptor Post Action, Multi-select Input, and HTTP Action. A DataRaptor Post Action is an action that can write data to a Salesforce object or invoke a Salesforce API. The consultant can use a DataRaptor Post Action to save the data back to Salesforce as a lead. A Multi-select Input is an element that can display a list of options for the user to choose from, such as energy products. A Multi-select Input allows multiple options to be selected at once, and can also have icons for each option. An HTTP Action is an element that can invoke a REST or SOAP service to retrieve data from an external system, such as current energy usage data. An HTTP Action can store the response data in a JSON object for further processing

Question: 71

In which two cases should an integration procedure be used as a data source for FlexCards and OmniScripts,
Choose 2 answers

A. To achieve elastic scaling
B. To retrieve multiple data sources in a single response
C. To minimize the number of elements to be configured
D. To separate the user interface from changes in thedata sources

Answer: B,D

Explanation:

The two cases where an integration procedure should be used as a data source for FlexCards and

OmniScripts are: To retrieve multiple data sources in a single response and To separate the user interface from changes in the data sources. An integration procedure is a tool that can execute multiple DataRaptor actions, such as Extract, Transform, and Load, and also invoke REST or SOAP services. The consultant can use an integration procedure as a data source for FlexCards and OmniScripts when they need to retrieve data from different sources, such as Salesforce objects or external systems, and store it in a JSON object for display. This way, the integration procedure can provide multiple data sources in a single response, without requiring multiple elements or actions in the FlexCard or OmniScript. The consultant can also use an integration procedure as a data source for FlexCards and OmniScripts when they want to separate the user interface from changes in the data sources. This way, the integration procedure can handle any changes in the data sources without affecting the FlexCard or OmniScript design or configuration.

Question: 72

A company needs a guided process for their internal support department that will provide the following functionality:

- Allow the user to enter their employee id
- Retrieve name, mobile phone, title, and email using the employee id
- Display the user's details on a page
- On another page, allow the user to enter case details, including subject, category, severity, and description

Which three elements should the consultant recommend to meet these requirements using an OmniScrip?

Choose 3 answers

A. Messaging
B. DataRaptor Post Action
C. Calculation Action
D. DataRaptor Extract Action
E. Steps

Answer: B,D,E

Explanation:

The three elements that the consultant should recommend to meet these requirements using an OmniScript are: Messaging, DataRaptor Post Action, and Steps. A Messaging element is an element that can display a text message with an icon and a color, such as blue for information or yellow for warning. The consultant can use a Messaging element to display the user's details on a page. A DataRaptor Post Action is an action that can write data to a Salesforce object or invoke a Salesforce API. The consultant can use a DataRaptor Post Action to save the case details to Salesforce. A Step element is an element that can group related fields and elements in an OmniScript, and display them in a page or a section. The consultant can use multiple Step elements to organize the OmniScript into logical steps, such as entering employee id, displaying user details, and entering case details.

Question: 73

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?

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: 74

A client wants to create an OmniScript to capture customer satisfaction. The process requires the following actions:

- Present the user with a customer satisfaction question that allows them to select one option from a 1-5 satisfaction rating.
- Create a case for an account team member to follow up.
- Email a summary message to the user.

Which OmniScript elements should the consultant recommend to meet these requirements'

A. Radio Group, Remote Action, and Messaging Action
B. Multi-Select, Remote Action, and Email Action
C. Radio Group, DataRaptor Post Action, and Email Action
D. Edit Block, DataRaptor Post Action, and Messaging Action

Answer: C

Explanation:

The OmniScript elements that the consultant should recommend to meet these requirements are: Radio Group, DataRaptor Post Action, and Email Action. A Radio Group element is an element that can display a list of options for the user to choose from, such as customer satisfaction rating. A Radio Group element allows only one option to be selected at a time, and can also have icons for each

option. A DataRaptor Post Action is an action that can write data to a Salesforce object or invoke a Salesforce API. The consultant can use a DataRaptor Post Action to create a case for an account team member to follow up. An Email Action is an action that can send an email to one or more recipients, with optional attachments and templates. The consultant can use an

Email Action to email a summary message to the user

Question: 75

A company wants to create a guided process for their customers. The process needs to retrieve data from Salesforce as well as external systems, and the steps of the process will branch depending on input from the user. Users will complete the process in a single session.
How should the consultant design the solution to meet these requirements?

A. FlexCards and Integration Procedures
B. FlexCards and DataRaptors
C. OmniScripts and Integration Procedures
D. OmniScripts and DataRaptors

Answer: C

Explanation:

The consultant should design the solution using OmniScripts and Integration Procedures to meet these requirements. An OmniScript is a tool that can design customer interactions using elements and actions. The consultant can use an OmniScript to create the guided process for the customers. An Integration Procedure is a tool that can execute multiple DataRaptor actions, such as Extract, Transform, and Load, and also invoke REST or SOAP services. The consultant can use Integration Procedures to retrieve data from Salesforce as well as external systems, and store it in a JSON object for display or processing in the OmniScript

Question: 76

what can a DataRaptor Extract do?

A. Create data in multiple related objects
B. Extract data from a spreadsheet
C. Retrieve data from multiple related objects
D. Load data from external sources

Answer: C

Explanation:

A DataRaptor Extract can retrieve data from multiple related objects. A DataRaptor Extract is a tool that can read data from a Salesforce object or invoke a Salesforce API, using SOQL queries or Apex

methods. A DataRaptor Extract can also use filters, joins, and formulas to manipulate the data. A DataRaptor Extract can retrieve data from multiple related objects by using lookup or master-detail relationships in the SOQL queries or Apex methods

Question: 77

A company has a process that requires a birthday validation. At the beginning of the process, the user is asked to input their birthday. If the user is less than 18 years old. then the process should display an error stating that the age cannot be less than 18. The process should restrict users from continuing. If the user is 18 or older, then the process should proceed without any error message. Which two OmniScript features should the consultant recommend to meet this validation requirement?
Choose 2 answers

A. Calculation
B. Messaging
C. Alert
D. Formula

Answer: B,D

Explanation:

The two OmniScript features that the consultant should recommend to meet this validation requirement are: Messaging and Formula. A Messaging element is an element that can display a text message with an icon and a color, such as blue for information or yellow for warning. The consultant can use a Messaging element to display an error message stating that the age cannot be less than 18 if the user's birthday indicates they are under 18 years old. A Formula element is an element that can perform calculations or validations on data values using expressions or functions. The consultant can use a Formula element to validate the user's birthday by subtracting it from the current date and checking if the result is greater than or equal to 18 years

Question: 78

A business has the following requirements:
- To display cases for an account
- The user should see all of the cases on the canvas
- The user should be able to create a new case from the canvas

The consultant decides to use states to enable this functionality.
Which type of state should the consultant recommend to allow users to create a new case from the canvas?

A. Active Card State
B. Blank Card State
C. Edit Mode State
D. New Card State

Answer: D

Explanation:

The type of state that the consultant should recommend to allow users to create a new case from the canvas is New Card State. A New Card State is a state that allows creating new records in a FlexCard using an OmniScript or an Integration Procedure. The consultant can use a New Card State to enable users to create a new case from the canvas, by invoking an

OmniScript or an Integration Procedure that writes data to the Case object in Salesforce.

Question: 79

A business wants to add typeahead functionality to an existing OmniScript. The business wants the user to begin entering the name of a contact in Salesforce. and allow the user to select the contact once it has found a match. Once the contact has been selected, the OmniScript should retrieve the contact's email address for use in a later step of the process.
What data source is needed when using the TypeAhead element?

A. HTTP Action
B. Integration Procedure Action
C. DataRaptor Post Action
D. DataRaptor Extract Action

Answer: D

Explanation:

The data source that is needed when using the TypeAhead element is DataRaptor Extract Action. A TypeAhead element is an element that can display a list of suggestions for the user to select from, as they type in a text field. A DataRaptor Extract Action is an action that can retrieve data from a Salesforce object or invoke a Salesforce API, using SOQL queries or Apex methods. The consultant can use a DataRaptor Extract Action as a data source for the TypeAhead element, to get the contact names from Salesforce and display them as suggestions

Question: 80

A healthcare company wants to enable its subscribers to add. edit, or delete dependents related to their policy via their community portal. The project team decides to use OmniStudio tools to provide this functionality.
In this scenario, which two OmniStudio features should the consultant recommend?
Choose 2 answers

A. Datatable
B. Remote Action

C. Response Action
D. Edit Block

Answer: C,D

Explanation:

The two OmniScript features that the consultant should recommend are Response Action and Edit Block. A Response Action is an action that can save or submit data from an OmniScript to a Salesforce object or an external system. The consultant can use a Response Action to allow the user to add, edit, or delete dependents related to their policy via their Community portal. An Edit Block is an element that can display a list of records for the user to view, edit, or delete, using an

OmniScript or an Integration Procedure. The consultant can use an Edit Block to display the dependents related to the policy, and allow the user to modify them

Question: 81

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?

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: 82

When a call center agent interacts with a customer, the agent must have all of the customer's related information available for a quick response. The business requires the agent to have access to:

- A view with information about a customer account
- A list of contacts and cases associated with the account
- All information should be on one screen

What OmnlStudio tool should be used to meet this requirement?

A. Customer IntellView
B. Lightning Record Page
C. OmniScript
D. FlexCards

Answer: D

Explanation:

The OmniStudio tool that should be used to meet this requirement is FlexCards. A FlexCard is a tool that can display data and actions in a card format. The consultant can use FlexCards to show all of the customer's related information on one screen, such as account information, contacts, cases, etc. FlexCards can also have actions that can invoke OmniScripts or Integration Procedures for further interactions, such as creating new records or updating existing ones

Question: 83

A business needs a 360° view of their accounts, including a FlexCard to display all of the products sold to the account. The business identified 20 different data elements and 10 actions that users would need when viewing the product information. Once all of the elements were collected together on the FlexCard, it looked cluttered.
What FlexCard feature should the consultant recommend to address this issue?

A. Zones
B. States
C. Actions
D. Flyouts

Answer: D

Explanation:

The FlexCard feature that the consultant should recommend to address this issue is Flyouts. A Flyout is a pop-up window that can display additional information or actions for a FlexCard. The consultant can use Flyouts to show some of the data elements and actions that are not essential for the main view of the product information, such as product details, specifications, reviews, etc. This way, the

consultant can reduce the clutter on the main FlexCard and improve the user experience

Question: 84

How should the consultant design the OmniScript solution to allow the user to stop and resume a process at a later time?

A. Configure the Save property
B. Configure a ResumeStep
C. Use an Integration Procedure
D. Use a DataRaptor Post Action

Answer: A

Explanation:

The consultant should design the OmniScript solution using the Save property to allow the user to stop and resume a process at a later time. The Save property is a property that determines whether an OmniScript can be saved as a draft and resumed later from where it was left off. The consultant can enable the Save property on an OmniScript to allow the user to stop and resume a process at a later time, without losing any data or progress.

Question: 85

A consultant needs to design an OmniScript to capture the following information:
- Select one payment method from a list of options
- Enter the address information with autocomplete
- Enter a phone number

Which OmniScript elements should be used to capture this information?

A. Radio, TypeAhead, and Telephone
B. Multi-Select. Address, and Telephone
C. Checkbox, Geolocation, and Number
D. Select, TypeAhead, and Number

Answer: A

Explanation:

The correct OmniScript elements to capture the information are Select, TypeAhead, and Number. Select allows the user to choose one option from a list of values, TypeAhead provides autocomplete suggestions for address fields, and Number validates that the input is a numeric value. Radio, MultiSelect, and Checkbox are not suitable for single selection scenarios. Geolocation is not an OmniScript element, and Telephone is not a valid data type for phone numbers.

Question: 86

A business implements a simple OmniScript in their call center that allows agents to quickly create a case when on the phone with customers. The OmniScript has been running successfully in the call center for over a year. The business decides it wants to allow partners to create cases directly from their Community portal.
What is the most efficient solution that the consultant can propose to meet this new requirement?

A. Deploy the existing OmniScript to the Community portal using OmniOut.
B. Embed the existing OmniScript as a reusable component within a new Community OmniScript.
C. Clone the existing OmniScript to a new LWC OmniScript and deploy to the Community portal.
D. Duplicate the existing OmniScript and deploy the new script on the website.

Answer: C

Explanation:

The most efficient solution is to clone the existing OmniScript to a new LWC OmniScript and deploy it to the Community portal. LWC OmniScripts are designed for web-based interactions and can be easily embedded in any web page. OmniOut is not a tool for deploying OmniScripts, but a feature that allows users to export data from an OmniScript. Embedding the existing OmniScript as a reusable component within a new Community OmniScript would require creating an unnecessary wrapper

script. Duplicating the existing OmniScript would create redundant code and maintenance issues.

Question: 87

A company needs to generate invoices when contracts reach an approved status. Users should initiate the invoice generation process from the contract page, but the option should not appear until the contract reaches the approved status. After the invoice is generated, it should be sent to the customer for signature.

What three tools should be used in the solution the consultant recommends to meet these requirements?

Choose 3 answers

A. OmniScript
B. Interaction Launcher
C. FlexCards
D. OmniStudio Action
E. DataRaptor

Answer: A,D,E

Explanation:

The three tools that should be used in the solution are OmniScript, OmniStudio Action, and DataRaptor. OmniScript can be used to create a user interface for generating invoices from contracts. OmniStudio Action can be used to conditionally display the invoice generation option on the contract page based on the contract status. DataRaptor can be used to read, transform, and write data between Salesforce and external systems. Interaction Launcher is not needed for this scenario, as it is used to launch interactions from other applications. FlexCards are not relevant for this scenario, as they are used to display contextual data on record pages.

Question: 88

A company needs to create a process that allows call center admins to retrieve all open cases that have a case type of "network issue" and submit the case data "as-is" to a back-office system for validation in batch on a daily basis. Once the cases have been submitted, the process should trigger an email to the supervisor.

Which three OmniStudio tools should the consultant recommend to meet these requirements?

Choose 3 answers

A. DataRaptor Turbo Extract
B. OmniScript
C. FlexCard
D. Integration Procedure
E. DataRaptor Load

Answer: A,D,E

Explanation:

The three tools that should be used in the solution are DataRaptor Turbo Extract, Integration Procedure, and DataRaptor Load. DataRaptor Turbo Extract can be used to retrieve all open cases that have a case type of “network issue” from Salesforce in batch mode. Integration Procedure can be used to orchestrate the data flow between Salesforce and the back-office system, as well as trigger an email to the supervisor after the cases have been submitted. DataRaptor Load can be used to write data to the back-office system using REST or SOAP APIs. OmniScript is not needed for this scenario, as there is no user interface involved. FlexCard is not relevant for this scenario, as it is used to display contextual data on record pages.

Question: 89

A business wants to transform an existing process into a digital interaction using OmniScript. The process includes several steps. Some steps apply to all users, and other steps only apply to users depending on their responses to certain questions. The business does not want all users to have to go through all the steps.
Which OmniScript feature should the consultant recommend to meet this requirement?

A. User Roles
B. Conditional Views

C. Script Configuration
D. Script Profiles

Answer: B

Explanation:

The OmniScript feature that should be recommended to meet this requirement is Conditional Views. Conditional Views allow the designer to create different paths within an OmniScript based on user responses or data values. User Roles are used to control access and visibility of OmniScripts based on user profiles or permissions. Script Configuration is used to define general settings and properties of an OmniScript. Script Profiles are used to assign different versions of an OmniScript to different user groups.

Question: 90

A company uses calculation procedures to determine product pricing. Due to the company's pricing schedules, there is always more than one calculation procedure active at one time for a given date.
In this scenario, how will the calculation engine select which calculation procedure to run?

A. Date modified
B. Priority
C. Sequence
D. Date created

Answer: B

Explanation:

The calculation engine will select which calculation procedure to run based on Priority. Priority is a field on the calculation

procedure object that determines the order in which calculation procedures are executed when there are multiple active ones for a given date. The lower the priority number, the higher the precedence. Date modified, Sequence, and Date created are not factors that affect the calculation procedure selection logic.

Question: 91

A company wants to create a new customer buying journey for their website. The buying journey should include the following functionality:

- Allow the user to enter contact and address information
- Require the user to enter age, gender, and optionally income bracket
- Compute a discount percentage per product based on the customer data provided
- Save the list of suggested products including discounts

Which three OmniStudio tools should the consultant use to design a solution that meets these requirements?

Choose 3 answers

A. OmniScript
B. Integration Procedures
C. Calculation Procedures and Matrices
D. FlexCard
E. OmniStudio Action

Answer: A,C,E

Explanation:

The correct OmniStudio tools to design a solution that meets these requirements are OmniScript, Calculation Procedures and Matrices, and OmniStudio Action. OmniScript can be used to create a user interface for capturing customer information and displaying suggested products. Calculation Procedures and Matrices can be used to compute the discount percentage per product based on the customer data provided. OmniStudio Action can be used to save the list of suggested products including discounts to Salesforce or an external system. Integration Procedures are not needed for this scenario, as there is no data integration involved. FlexCard is not relevant for this scenario, as it is used to display contextual data on record pages.

Question: 92

In an Integration Procedure, what group element will control whether an individual action executes?

A. Conditional Block
B. Cache Block
C. Try-Catch Block
D. Loop Block

Answer: A

Explanation:

The group element that will control whether an individual action executes in an Integration Procedure is Conditional Block. Conditional Block allows the designer to specify a condition that determines whether the actions inside the block will run or not. Cache Block is used to store data in memory for later use. Try-Catch Block is used to handle errors and exceptions. Loop Block is used to iterate over a collection of data.

Question: 93

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

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: 94

Which three functions are performed by Action elements in OmniScript?
Choose 3 answers

A. Display error messages to the user
B. Get and update data through APIs
C. Organize data into multiple pages
D. Send DocuSign emails for signature
E. Get and update data in Salesforce

Answer: B,D,E

Explanation:

The three functions that are performed by Action elements in OmniScript are get and update data through APIs, send DocuSign emails for signature, and get and update data in Salesforce. Action elements are used to perform various operations on data within an OmniScript. Display error messages to the user is not a function of Action elements, but of Error elements. Organize data into multiple pages is not a function of Action elements, but of Page elements.

Question: 95

A company needs to implement new verification processes for contacts in their org. This process relies on three Contact record types: Recruiter, Candidate, and Trainer. The verification process is different for each type of contact. For example, recruiters must pass a background check; trainers must complete mandatory training classes, and candidates must achieve certifications. Which OmniStudio tools should the consultant recommend to meet these requirements?

A. Specific FlexCards with Actions for each type of Contact
B. Multiple OmniStudio Actions that invoke separate OmniScripts

C. Single FlexCard with an Action to invoke an OmniScript
D. Single OmniStudio Action that invokes separate Omniscripts

Answer: B

Explanation:

The OmniStudio tools that should be recommended to meet these requirements are single OmniStudio Action that invokes separate Omniscripts. OmniStudio Action can be used to conditionally launch different Omniscripts based on the record type of the contact. Omniscripts can be used to implement the verification processes for each type of contact. Specific FlexCards with Actions for each type of Contact would require creating redundant FlexCards and Actions for each record type. Multiple OmniStudio Actions that invoke separate Omniscripts would require creating redundant Actions for each record type. Single FlexCard with an Action to invoke an Omniscript would not allow for different verification processes based on the record type.

Question: 96

A consultant wants to proceed with a FlexCard canvas design that will display data from a legacy billing system. The legacy billing system is being extended to add a SOAP endpoint that will allow for the data to be retrieved in XML format. However, the endpoint is not ready yet.
Which data source should the team use to make progress on the design?

A. DataRaptor
B. Apex REST
C. Integration Procedures
D. Apex Remote

Answer: C

Explanation:

Question: 97

A company has an existing OmniScript that gathers customer information. User feedback suggests that the current process requires too much typing because all the data is entered into input text fields. The users also are concerned because there is no validation in the data that is entered, and users can sometimes enter data that contains errors. The required inputs are first

name, last name, age, and email, users are also presented a brief questionnaire regarding customer satisfaction. Which three input elements can the consultant recommend to improve usability and to provide validation?
Choose 3 answers

A. Headline
B. Radio Group

C. Email
D. Text Area
E. Range

Answer: B,C,E

Explanation:

The three input elements that can be recommended to improve usability and provide validation are Radio Group, Email, and Range. Radio Group allows the user to select one option from a list of values, which reduces typing and ensures valid input. Email validates that the input is a valid email address format. Range allows the user to select a numeric value within a specified range using a slider, which reduces typing and ensures valid input. Headline is not an input element, but a display element that shows text in large font size. Text Area is not suitable for this scenario, as it allows the user to enter multiple lines of text without validation.

Question: 98

A consultant receives a requirement to display products installed at an account site in a customer's 360° FlexCard view. The business requires that the width of the fields displayed should change depending on the device used to view the FlexCard. For example, the Product Name and Model field elements should display at full width on mobile devices, but they should shrink to 60% on devices such as laptops and desktops.
How should the consultant design the FlexCard to meet this requirement?

A. Enable the Responsive feature on the Product Name and Model field elements
B. Enable the Mobile-First feature in FlexCard settings
C. Create two states, one for mobile devices and another for non-mobile devices
D. Create two FlexCards, one for mobile devices and another for non-mobile devices

Answer: A

Explanation:

The correct way to design the FlexCard to meet this requirement is to create two states, one for mobile devices and another for non-mobile devices. States allow the designer to define different layouts and behaviors for the same FlexCard based on conditions. The width of the fields can be adjusted for each state using the Size property. Enabling the Responsive feature on the field elements would not change their width, but only their alignment. Enabling the Mobile-First feature in FlexCard settings would not affect the width of the fields, but only the order in which they are displayed. Creating two FlexCards, one for mobile devices and another for non-mobile devices, would create redundant code and maintenance issues.

Question: 99

which Action element redirects the user outside an LWC OmniScript?

A. Response
B. Done
C. Post to Object
D. Navigate

Answer: D

Explanation:

The Action element that redirects the user outside an LWC OmniScript is Navigate. Navigate allows the user to navigate to a URL or a Salesforce record or page. Response, Done, and Post to Object are not Action elements, but End elements that terminate an OmniScript.

Question: 100

What business problem does DataRaptor solve?

A. It removes the need to code data mappings for data transformations.
B. It allows developers to create complex API queries declaratively.
C. It combines multiple steps and processes into a single server call.
D. It guides humans through a complex business process.

Answer: A

Explanation:

The business problem that DataRaptor solves is that it removes the need to code data mappings for data transformations. DataRaptor allows the designer to declaratively define how data is read, transformed, and written between Salesforce and external systems. It does not allow developers to create complex API queries declaratively, as this is done by Integration Procedures. It does not combine multiple steps and processes into a single server call, as this is done by Calculation Procedures. It does not guide humans through a complex business process, as this is done by OmniScripts.

Question: 101

A consultant designs a FlexCard with five card states following best practices. Four of the card states have a condition. At runtime, if two of the states' condition are true, which state will be displayed?

A. The first state with the highest priority closest to the top of the canvas
B. The state closest to the top of the canvas
C. The first state with a true condition closest to the top of the canvas
D. The state with a true condition that has the highest priority

Answer: D

Explanation:

The state that will be displayed at runtime if two of the states' condition are true is the state with a true condition that has the highest priority. Priority is a field on the state object that determines the order in which states are evaluated when there are multiple states with true conditions. The lower the priority number, the higher the precedence. The position of the state on the canvas or the order of evaluation does not affect the state selection logic.

Question: 102

A company needs an OmniScript to allow customers to order products and services from their website. After the order is submitted, the customer should be able to download a PDF summary of the order.
What type of DataRaptor should the consultant recommend to meet this requirement?

A. DataRaptor Load
B. DataRaptor Extract
C. DataRator Turbo Extract
D. DataRaptor Transform

Answer: D

Explanation:

The type of DataRaptor that should be recommended to meet this requirement is DataRaptor Transform. DataRaptor Transform allows the designer to transform data from one format to another using templates. In this case, a PDF template can be used to generate a PDF summary of the order. DataRaptor Load is used to write data to Salesforce or an external system. DataRaptor Extract is used to retrieve data from Salesforce or an external system. DataRator Turbo Extract is used to retrieve data from Salesforce in batch mode.

Question: 103

A company needs to create some boundaries for their sales teams regarding the minimum and maximum discounts that can be applied to their orders. The discount thresholds are set using adjustments such as 5%, 10%, 15%, 20%, 25%, and 30%. The minimum adjustment and the maximum adjustment are determined by their region and their customer lifetime score.
Which two OmniStudio tools should the consultant recommend to meet these requirements' Choose 2 answers

A. OmniStudio Action

B. Calculation Procedure
C. Calculation Matrix
D. DataRaptor Transform

Answer: B,C

Explanation:

The two OmniStudio tools that should be recommended to meet these requirements are Calculation Procedure and Calculation Matrix. Calculation Procedure allows the designer to define a multi-step calculation process that can use data from various sources. Calculation Matrix allows the designer to define a table of values that can be used as inputs or outputs for calculations. In this case, a Calculation Matrix can be used to store the discount thresholds based on region and customer lifetime score, and a Calculation Procedure can be used to apply them to the orders. OmniStudio Action is not needed for this scenario, as there is no interaction involved. DataRaptor Transform is not suitable for this scenario, as it is used to transform data from one format to another.

Question: 104

A business process needs to perform a multi-step calculation on each contact record in a list. Using an Integration Procedure to process the list, what feature can be used to meet this requirement?

A. Loop Block
B. Batch Action
C. Conditional Block
D. DataRaptor Transform Action

Answer: A

Explanation:

The feature that can be used to meet this requirement in an Integration Procedure is Loop Block. Loop Block allows the designer to iterate over a collection of data and perform actions on each item. In this case, a Loop Block can be used to loop over each contact record in the list and perform a multi-step calculation using Calculation Actions or Calculation Procedures. Batch Action is not a feature in Integration Procedure, but a type of Action element that can perform bulk operations on data. Conditional Block is used to specify a condition that determines whether the actions inside the block will run or not. DataRaptor Transform Action is used to transform data from one format to another using templates.

Question: 105

A company has designed a process that extracts a large amount of data from a Salesforce object. Due to the volume of data that will be retrieved, it is important that Governor limits are not exceeded.

What solution should the consultant recommend?

A. DataRaptor Extract with Chaining
B. DataRaptor Turbo Extract
C. Calculation Procedure and DataRaptor Turbo Extract
D. Integration Procedure and DataRaptor Turbo Extract

Answer: B

Explanation:

The solution that the consultant should recommend is DataRaptor Turbo Extract. DataRaptor Turbo Extract allows the designer to retrieve a large amount of data from a Salesforce object in batch mode without hitting governor limits. DataRaptor Extract is not suitable for this scenario, as it can only retrieve data from a single record or a small set of records. Calculation Procedure and DataRaptor Turbo Extract are not needed for this scenario, as there is no calculation involved. Integration Procedure and DataRaptor Turbo Extract are not needed for this scenario, as there is no data integration involved.

Question: 106

When a customer calls to report a product issue, agents need to check all open cases related to that product to see if there are any solutions that can resolve the customer's issue. Products that have been purchased are stored as assets, and there is a lookup relationship from case to asset that allows cases to be linked to the products customers have purchased. What type of DataRaptor can be used to retrieve a list of cases filtered by the customer's asset and the last service date of the asset?

A. DataRaptor Turbo
B. DataRaptor Extract
C. DataRaptor Load
D. DataRaptor Transform

Answer: B

Explanation:

The type of DataRaptor that can be used to retrieve a list of cases filtered by the customer's asset and the last service date of the asset is DataRaptor Extract. DataRaptor Extract allows the designer to retrieve data from Salesforce or an external system using filters and relationships. In this case, a filter can be used to match the asset ID and the last service date, and a relationship can be used to join the case and asset objects. DataRaptor Turbo is not a type of DataRaptor, but a feature that enables batch mode for DataRaptor Extract. DataRaptor Load is used to write data to Salesforce or an external system. DataRaptor Transform is used to transform data from one format to another using templates.

Question: 107

A business has a project that must be completed soon in order to meet important deadlines.
However, the developer on the project has left the company, and the new team on the project has decided to use OmniStudio tools. The development work completed so far was done using APEX code. The new team must complete the project following these guidelines:

- Minimize implementation time
- Ensure end-user processes are as simple as possible
- Find a way to ensure optimal UX

In this scenario, what two actions should the consultant recommend to the project team'

Choose 2 answers

A. Replace existing APEX using DataRaptors and HTTPActions
B. Create new LWC templates for branding and styling
C. Implement FlexCards and OmniScripts for the front-end
D. Use existing APEX classes as data sources

Answer: C

Explanation:

The two actions that the consultant should recommend to the project team are replace existing APEX using DataRaptors and HTTPActions, and implement FlexCards and OmniScripts for the front-end. Replacing existing APEX using DataRaptors and HTTPActions would reduce implementation time, as these tools are declarative and do not require coding. Implementing FlexCards and OmniScripts for the front-end would ensure end-user processes are as simple as possible, as these tools provide guided interactions and contextual data. Creating new LWC templates for branding and styling would increase implementation time, as this would require coding and testing. Using existing APEX classes as data sources would not ensure optimal UX, as this would limit the flexibility and scalability of the solution.

Question: 108

A company implements an integration procedure that is invoked from an OmniScript. The integration procedure includes a recommend to improve performance and address users concerns

Which feature should the consultant recommend to improve performance and address users concerns?

A. Try Catch
B. Batch jobs
C. Chaining
D. Conditions

Answer: C

Explanation:

The two actions that the consultant should recommend to the project team are replace existing APEX using DataRaptors and HTTPActions, and implement FlexCards and OmniScripts for the frontend. Replacing existing APEX using DataRaptors and HTTPActions would reduce implementation time, as these tools are declarative and do not require coding. Implementing FlexCards and OmniScripts for the front-end would ensure end-user processes are as simple as possible, as these tools provide guided interactions and contextual data. Creating new LWC templates for branding and styling would increase implementation time, as this would require coding and testing. Using existing APEX classes as data sources would not ensure optimal UX, as this would limit the flexibility and scalability of the solution.

Question: 109

A business wants to display customer 360° information in a console for their call center agents. The customer information will come from a variety of sources, and the information should be grouped together logically. The agents will need to take different actions depending on the context of each group of information.

Which tool should a consultant recommend to meet this requirement?

A. Omniscript
B. Lightning web components
C. Visualforce Page
D. FlexCards

Answer: D

Explanation:

The tool that the consultant should recommend to meet this requirement is Lightning web components. Lightning web components are custom HTML elements built using HTML and modern JavaScript. They can be used to create rich and interactive user interfaces that display data from various sources and allow users to take actions. In this case, a Lightning web component can be used to display customer 360° information in a console using FlexCards, OmniScripts, or other components. Omniscript is not a tool, but a type of component that can be used to create guided interactions. Visualforce Page is not suitable for this requirement, as it is a legacy technology that uses server-side rendering and does not support modern web standards. FlexCards are not a tool, but a type of component that can be used to display contextual data on record pages.

Question: 110

A pop-up window will display additional information about the campaign, such as details about the members. A button must initiate a guided process to add new leads to the campaign.
Which two FlexCard features should the consultant recommend to meet these requirements?

Choose 2 answers

A. Block Element
B. OmniScript Action
C. Interaction Launcher
D. Data Table
E. Flyout Action

Answer: B,E

Explanation:

Flyout Action (E) – The Flyout Action in a FlexCard is used to display additional information in a popup window. Since the requirement mentions that a pop-up window must display additional details, the Flyout Action is the best choice for this functionality.
OmniScript Action (B) – The OmniScript Action launches an OmniScript, which is a guided interaction tool for multi-step processes. Since the requirement states that a button should initiate a guided process to add new leads, an OmniScript is the ideal choice.
A . Block Element – Blocks in FlexCards are used for organizing layout and content but do not provide pop-ups or interactive actions.
C . Interaction Launcher – This launches an interaction but is typically used for navigating to another page or launching an external URL, not a pop-up with additional details.
D . Data Table – A Data Table is used to display multiple records in a table format but does not trigger a guided process or pop-ups.
Comprehensive and Detailed In-Depth Why not the other options?? Salesforce OmniStudio Reference:
FlexCards Actions Overview
Using Flyout Actions in FlexCards

Launching OmniScripts from FlexCards

Question: 111

Which element retrieves data from a Salesforce picklist in an org?

A. DataRaptor Extract Action
B. Lookup
C. Select
D. Calculation Action
E. Mark this item for later review.

Answer: C

Explanation:

In Salesforce OmniStudio, the Select element within an OmniScript is specifically designed to retrieve and display picklist values from a Salesforce object field, allowing users to choose from a predefined list of options. This element is used to create a dropdown or picklist interface in an OmniScript, enabling users to interact with Salesforce data by selecting a single value from the available options.

Here's why Select is the correct answer:

The Select element in OmniScript supports multiple methods to populate its options, including retrieving values directly from a Salesforce picklist field. According to the official Salesforce OmniStudio documentation, you can configure the Select element's "Option Source" to "SObject," which allows it to fetch picklist values from a specified Salesforce object and field. For example, if you have a picklist field like Industry on the Account object, the Select element can retrieve all active picklist values (e.g., "Technology," "Healthcare," etc.) and present them as a dropdown to the user. The Select element is highly flexible and supports three option source types: Manual: Manually defined label-value pairs.
SObject: Retrieves options from a Salesforce object field (such as a picklist).
Custom: Uses an Apex controller for more complex logic.When set to "SObject," it directly queries the Salesforce schema to pull the picklist metadata, ensuring the values reflect what's defined in the org.
Now, let's examine why the other options are incorrect:

A . DataRaptor Extract Action: A DataRaptor Extract is a powerful tool in OmniStudio used to retrieve data from Salesforce objects, transform it, and pass it to an OmniScript or FlexCard. While it can retrieve picklist data as part of a broader dataset (e.g., extracting a record with a picklist field value), it is not an "element" within an OmniScript, nor is it specifically designed to display or interact with picklist values in a user interface. Instead, it operates behind the scenes as a data retrieval mechanism. The Select element, however, is the UI component that presents those values to the user.
B . Lookup: The Lookup element in OmniScript is used to search for and select a record from a Salesforce object based on user input (e.g., finding an Account by name). It does not retrieve or display picklist values from a field; it retrieves entire records. While it can indirectly involve picklist fields as part of the record data, its primary purpose is record selection, not picklist value retrieval for display.
D . Calculation Action: A Calculation Action in OmniScript performs mathematical or logical operations based on user input or data (e.g., adding two numbers or concatenating strings). It does not retrieve data from Salesforce picklist fields or interact with them directly. Its role is computation, not data retrieval or presentation.
The official Salesforce OmniStudio documentation, specifically the "OmniScript Elements" section, highlights the Select element as the appropriate choice for working with picklist fields in a guided process. For instance, in a scenario where a user

needs to choose a Case Reason from a picklist, the Select element fetches the active values (e.g., "Billing Issue," "Technical Support") from the Reason field on the Case object and renders them as a dropdown.

Reference:

Salesforce OmniStudio Documentation: OmniScript Elements Reference – Details the Select element and its SObject option source capabilities.
Salesforce OmniStudio Developer Guide: DataRaptors – Explains DataRaptor Extract's role in data retrieval, distinct from UI elements like Select.

Salesforce Help: OmniScript Designer – Describes how to configure the Select element to connect to Salesforce picklist fields.

Question: 112

A Consultant is working on a project that involves using OmniStudio tools to design solutions that meet customer business requirements. The solutions need to be maintainable, scalable, and contribute to long-term customer success. Which OmniStudio tool should the Consultant use to display data and launch actions?

A. OmniScripts
B. FlexCards
C. Integration Procedures
D. DataRaptors

Answer: B

Explanation:

FlexCards (B) – FlexCards are designed to display data from Salesforce or external sources and launch actions such as OmniScripts, Integration Procedures, or other system interactions.
FlexCards provide a reusable, scalable UI component that aligns with the requirements of maintainability and long-term success.
A . OmniScripts – OmniScripts are used for guided interactions but are not meant for displaying static or dynamic data as FlexCards do.
C . Integration Procedures – These are used for server-side processing and data transformation, not for directly displaying UI elements.
D . DataRaptors – DataRaptors are data extraction and transformation tools but do not display data or launch actions on their own.
Comprehensive and Detailed In-Depth Why not the other options?? Salesforce OmniStudio Reference:
FlexCards Overview
Launching OmniScripts and Actions from FlexCards

Question: 113

Which two OmniScript components should the consultant recommend using to meet this validation requirement?
Choose 2 answers

A. Messaging
B. Alert
C. Calculation
D. Formula

Answer: C,D

Explanation:

Comprehensive and Detailed In-Depth The question asks for two OmniScript components that are best suited to meet a "validation requirement." Since the specific validation requirement isn't provided, I'll assume a common scenario in OmniStudio: validating user input or data within an OmniScript to ensure it meets certain conditions (e.g., checking if a field value is within an acceptable range, matches a pattern, or satisfies a business rule). Based on official Salesforce OmniStudio documentation, the Calculation and Formula components are the most appropriate tools for implementing validation logic within an OmniScript.

Here's why Calculation and Formula are the correct answers:

C . Calculation: The Calculation component (specifically, a Calculation Action) in OmniScript is used to perform operations on data, including validation logic. It allows you to execute calculations, manipulate data, and set conditions based on user inputs or retrieved data. For example, a Calculation Action can check if a numeric input exceeds a threshold (e.g., Quantity > 100) or if a text field matches a required format. It can then set a flag (e.g., isValid = true/false) that can be used to control the OmniScript flow—such as displaying an error or blocking navigation. Calculation Actions are highly versatile because they support OmniScript's JSON data structure and can integrate with external data sources via DataRaptors or Integration Procedures. According to the OmniStudio documentation, Calculation Actions are ideal for complex validations requiring multiple steps or data transformations.

D . Formula: The Formula component is a lightweight, inline element in OmniScript used to evaluate expressions and perform simple validations directly within the script. It leverages OmniScript's formula syntax (similar to Salesforce formulas) to compute values or check conditions. For instance, a Formula can validate that a date input is not in the past (e.g., TODAY() <= InputDate) or that a text field contains a specific substring. Unlike Calculation Actions, Formulas are embedded within a specific field or step and are best for straightforward, single-expression validations. The result of a Formula can be used to show/hide elements, set field values, or trigger other actions, making it a key tool for real-time validation.

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

A . Messaging: The Messaging component in OmniScript is designed to display informational, warning, or error messages to the user based on predefined conditions or data. While it can communicate the result of a validation (e.g., "Please enter a valid phone number"), it does not perform the validation itself. It is a presentation tool, not a validation mechanism. For example, you might use a Formula to check if a field is empty and then use Messaging to display an error, but Messaging alone cannot enforce or evaluate the validation requirement.

B . Alert: The Alert component is similar to Messaging in that it displays notifications or prompts to the user, often with more prominence (e.g., a pop-up). It's useful for alerting users about validation failures (e.g., "Input exceeds maximum allowed value"), but it does not contain logic to perform the validation. Like Messaging, it relies on other components (e.g., Calculation or Formula) to determine whether an alert should be shown.

Why Calculation and Formula Together?

In practice, Calculation and Formula complement each other for validation requirements:

Use Formula for simple, field-level validations that need immediate feedback (e.g., checking if an email contains "@").
Use Calculation for multi-step or complex validations that involve multiple fields, external data, or conditional logic (e.g., validating a combination of inputs against a business rule).Together, they provide a robust framework to enforce validation

within an OmniScript, ensuring data integrity before submission or progression.

Example Scenario:

Suppose the validation requirement is to ensure a user-entered “Discount Percentage” is between 0 and 50:

A Formula could be added to the Discount field: AND(Discount >= 0, Discount <= 50), setting a Boolean flag (isDiscountValid).

A Calculation Action could then check isDiscountValid and, if false, update a variable to trigger an error message or block the Next button.

This combination ensures both the validation logic and its enforcement are handled effectively.

Reference:

Salesforce OmniStudio Documentation: OmniScript Actions – Describes Calculation Action for data manipulation and validation.

Salesforce OmniStudio Developer Guide: Formula Element – Details how Formulas evaluate conditions and support validation.

Salesforce Help: OmniScript Designer – Explains Messaging and Alert as display tools, not validation components.

Question: 114

What is the purpose of Step elements in an OmniScript?

A. Organizes the script into one or more pages
B. Groups elements that extract data
C. Enables the use of repeatable blocks
D. Allows the user to input data

Answer: A

Explanation:

Step elements (A) – The Step element is used in OmniScripts to break down the script into multiple pages or sections. This helps organize the user flow and enhances the user experience by making long processes more manageable.

B . Groups elements that extract data – This is not correct; data extraction is handled by DataRaptors or Integration Procedures.

C . Enables the use of repeatable blocks – This is incorrect; Repeat Blocks allow users to dynamically add multiple sets of input fields, not Steps.

D . Allows the user to input data – While Steps may contain input fields, their primary purpose is to organize the script into multiple pages, not to handle input directly.

Comprehensive and Detailed In-Depth Why not the other options?? Salesforce OmniStudio Reference:
OmniScript Step Elements

Question: 115

A consultant Is designing a solution for a client using Omnistudio. The client wants to display data and launch actions from their Salesforce org.

Which OmniStudio tool should the consultant use to meet this requirement?

A. DataRaptors
B. FlexCards
C. OmniScripts
D. Integration Procedures

Answer: B

Explanation:

FlexCards (B) – FlexCards are designed to display data from Salesforce and launch actions such as OmniScripts, Integration Procedures, or external API calls.
FlexCards allow users to interact with data dynamically while keeping the UI simple and intuitive.
A . DataRaptors – These are used for extracting, transforming, and loading data, but they do not have a UI component to display data.
C . OmniScripts – OmniScripts are used for step-by-step guided interactions but are not meant for displaying data at a glance like FlexCards.
D . Integration Procedures – These are used for server-side processing but do not display data or provide UI elements.
Comprehensive and Detailed In-Depth Why not the other options?? Salesforce OmniStudio Reference:
FlexCards Overview
Launching Actions from FlexCards

Question: 116

A company is designing a new console for contact center agentsto display case data for customers. All cases will be fetched using a single DataRaptor. This page needs to display cases as follows:
* "Open" cases with case description, case open date, case type, assigned to and priority fields. Open cases should be highlighted with a red border.
* "Awaiting Closure" cases with case description, last action taken date, resolution, approval reason for closure, and assigned to fields. These cases should be highlighted with a grey border.
* "Closed" cases with case description, resolution, case closed date
fields with a link to duplicate cases.
How should the consultant design the FlexCard solution to meet these requirements?

A. Using three FlexCards
B. Using a single FlexCard with three states
C. Using a single FlexCard with multiple card filters
D. Using a single FlexCard with three flyouts

Answer: B

Explanation:

Below is the formatted question with a 100% verified answer based on official Salesforce OmniStudio documentation, including a comprehensive explanation and references.

Question: 117

A company is designing a new console for contact center agents to display case data for customers. All cases will be fetched using a single DataRaptor. This page needs to display cases as follows:

"Open" cases with case description, case open date, case type, assigned to, and priority fields. Open cases should be highlighted with a red border.
"Awaiting Closure" cases with case description, last action taken date, resolution, approval reason for closure, and assigned to fields. These cases should be highlighted with a grey border.
"Closed" cases with case description, resolution, case closed date fields with a link to duplicate cases.How should the consultant design the FlexCard solution to meet these requirements?

A. Using three FlexCards
B. Using a single FlexCard with three states
C. Using a single FlexCard with multiple card filters
D. Using a single FlexCard with three flyouts

Answer: B

Explanation:

Comprehensive and Detailed In-Depth The requirement involves designing a FlexCard solution to display case data fetched by a single DataRaptor, with three distinct categories of cases ("Open," "Awaiting Closure," and "Closed"), each with specific fields and unique styling (e.g., red border for Open, grey border for Awaiting Closure). Based on Salesforce OmniStudio capabilities, the most efficient and appropriate design is a single FlexCard with three states.

Here's why B. Using a single FlexCard with three states is the correct answer:

FlexCard States Overview: In OmniStudio, a FlexCard can have multiple "states," which are conditional views of the card's content and styling based on data conditions. Each state can display different fields, apply unique CSS styles (e.g., borders), and include interactive elements (e.g., links). States are driven by conditions evaluated against the data retrieved by the DataRaptor, making them ideal for scenarios where the same dataset (in this case, cases from a single DataRaptor) needs to be presented differently based on a field value (e.g., Status).
Meeting the Requirements:
Data Source: All cases are fetched via a single DataRaptor, which aligns with using one FlexCard since FlexCards are designed to work with a single data source (e.g., a DataRaptor Extract) and then manipulate or display that data in various ways.
"Open" Cases: A state can be configured with a condition like Status = 'Open', displaying fields such as CaseDescription, CaseOpenDate, CaseType, AssignedTo, and Priority. Custom CSS can be applied in the state's style settings to add a red border (e.g., border: 2px solid red).
"Awaiting Closure" Cases: Another state with a condition like Status = 'Awaiting Closure' can show CaseDescription, LastActionTakenDate, Resolution, ApprovalReasonForClosure, and AssignedTo, styled with a grey border (e.g., border: 2px solid grey).
"Closed" Cases: A third state with Status = 'Closed' can display CaseDescription, Resolution, and CaseClosedDate, with a hyperlink element added for duplicate cases (e.g., linking to a related DuplicateCaseId field). No specific border color is

mentioned, so default styling or a custom style could be applied.
Display Logic: By default, FlexCards in a list format (e.g., a datatable or repeating block) can iterate over the DataRaptor's JSON output, applying the appropriate state to each case record based on its Status. This ensures all cases are displayed dynamically with their respective fields and styles.
Why One FlexCard?: Using a single FlexCard with states keeps the solution efficient, maintainable, and aligned with OmniStudio best practices. It leverages one DataRaptor call, reduces redundancy, and centralizes the logic and styling in a single component.
Now, let's examine why the other options are incorrect:

A . Using three FlexCards: While it's technically possible to create three separate FlexCards (one for Open, one for Awaiting Closure, and one for Closed), this approach is inefficient and impractical. Each FlexCard would need its own DataRaptor or a filtered subset of the same DataRaptor output, leading to redundant configurations and multiple queries or data manipulations. Additionally, displaying all three categories together in a cohesive console view would require embedding them in an OmniScript or custom layout, complicating the design. The requirement implies a unified display of all cases, which a single FlexCard with states handles more naturally.
C . Using a single FlexCard with multiple card filters: Card filters in FlexCards allow users to dynamically filter the displayed data (e.g., a dropdown to show only Open cases). However, filters don't inherently change the fields displayed or apply specific styling like red or grey borders per category—they only subset the data. To meet the requirement of showing different fields (e.g., Priority for Open vs. LastActionTakenDate for Awaiting Closure) and unique styling, you'd still need states or custom coding beyond what filters provide. Filters are user-driven, not automatic, and don't fully address the requirement.
D . Using a single FlexCard with three flyouts: Flyouts are pop-up panels in FlexCards that display additional details or actions when a user interacts with an element (e.g., clicking a case row). While flyouts could show extra case details, they don't suit the requirement of displaying all cases with their respective fields and borders directly on the main card. Flyouts are supplementary, not primary displays, and using three flyouts would require user interaction to view each category, which contradicts the need to "display cases as follows" in a visible, highlighted manner.
Best Practice Alignment:
The OmniStudio documentation emphasizes FlexCard states for scenarios where data from a single source needs conditional rendering and styling. This approach minimizes maintenance overhead, optimizes performance (one DataRaptor call), and provides a seamless user experience for contact center agents viewing all case types in one console.

Reference:

Salesforce OmniStudio Documentation: FlexCards Designer – Details how states can conditionally display data and apply custom styles.
Salesforce OmniStudio Developer Guide: Working with FlexCard States – Explains configuring states based on field values like Status.
Salesforce Help: FlexCard Styling – Covers applying CSS (e.g., borders) to states for visual differentiation.

Question: 118

A company has several customer support departments. Each department has different case creation processes, but they want to unify them into a single LWC OmniScript. During the discovery, the consultant identifies two key requirements:
* Each department has different case fields they need to capture
* If the user cancels the process, they should return to a support console app
Which two OmniScript features should the consultant recommend to meet these requirements?
Choose 2 answers

A. Enable the Save for Later feature
B. Set Enable Cancel in Setup
C. Use Conditional Views on steps for each department
D. Add a Done action that redirects to the support console

Answer: B,C

Explanation:

Set Enable Cancel in Setup (B) – This ensures that users can cancel the OmniScript process and be redirected to a specific URL (e.g., support console app).
Use Conditional Views on steps for each department (C) – Conditional Views allow different fields to be displayed based on department-specific requirements, ensuring that each department only sees the fields relevant to them.
A . Enable the Save for Later feature – This allows users to pause and resume the OmniScript later, but the question specifically asks about canceling the process, not saving progress.
D . Add a Done action that redirects to the support console – The Done Action is used to submit the process, but the question focuses on handling the cancellation scenario, making "Enable Cancel" the better choice.
Comprehensive and Detailed In-Depth Why not the other options?? Salesforce OmniStudio Reference:
OmniScript Conditional Views
OmniScript Enable Cancel Option

Question: 119

A consultant is working with a client who wants to create guided interactions using data from their Salesforce org and external sources.
Which Omnistudio tool should the consultant recommend for creating these guided interactions?

A. Flexcards
B. Integration Procedures
C. Omniseripts
D. Data Mappers

Answer: C

Explanation:

The requirement is to create "guided interactions" that utilize data from both a Salesforce org and external sources. In Salesforce OmniStudio, OmniScripts is the tool specifically designed for building interactive, guided processes that lead users through a series of steps, integrating and displaying data from multiple sources, including Salesforce and external systems.

Here's why C. OmniScripts is the correct answer:

Purpose of OmniScripts: OmniScripts are a core OmniStudio component used to create structured, step-by-step user experiences (guided interactions) that can incorporate data from Salesforce objects and external systems. They enable

consultants to design dynamic, interactive interfaces with elements like text inputs, picklists, and buttons, guiding users through processes such as filling out forms, updating records, or completing workflows. OmniScripts support integration with other OmniStudio tools (e.g., DataRaptors and Integration Procedures) to fetch and manipulate data, making them ideal for this scenario.

Handling Data from Multiple Sources:

Salesforce Data: OmniScripts can use DataRaptors to extract or update data from Salesforce objects (e.g., retrieving Account details or updating a Case).

External Sources: OmniScripts can leverage Integration Procedures to call external APIs or systems (e.g., pulling customer data from an ERP system) and then display or process that data within the script.For example, an OmniScript could guide a user to input customer information, fetch related Salesforce records using a DataRaptor Extract, and retrieve payment history from an external API via an Integration Procedure—all within a single, cohesive interaction.

Guided Interaction Features: OmniScripts offer a drag-and-drop designer with elements like Steps, Inputs, and Actions, allowing consultants to craft a logical flow. Features like conditional visibility, validation rules, and navigation controls ensure the interaction is guided and user-friendly, meeting the client's need for a structured experience.

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

A . FlexCards: FlexCards are designed to display data in a concise, card-based format (e.g., a summary of customer details or case status). While they can pull data from Salesforce and external sources (via DataRaptors or Integration Procedures), they are primarily read-only or lightly interactive displays, not tools for creating multi-step guided interactions. FlexCards are better suited for dashboards or quick views, not for guiding users through a process like data entry or decision-making.

B . Integration Procedures: Integration Procedures are powerful tools in OmniStudio for orchestrating complex data operations, such as calling external REST APIs, transforming data, or chaining multiple actions. However, they are backend processes, not user-facing tools. They don't provide a UI or guided interaction framework—they support OmniScripts or FlexCards by supplying the data or logic needed. Thus, they're a component of the solution, not the primary tool for guided interactions.

D . Data Mappers: Data Mappers (likely a typo for "DataRaptors" in OmniStudio context) are used to extract, transform, load, or post data between Salesforce and OmniStudio components. For example, a DataRaptor Extract pulls Salesforce data, while a DataRaptor Turbo Extract maps it to JSON. However, DataRaptors are data-handling tools, not UI components. They can't create guided interactions on their own—they provide the data that OmniScripts or FlexCards use.

Why OmniScripts Stand Out:

OmniScripts are uniquely suited for this requirement because they combine a user interface with data integration capabilities. For instance, a consultant could design an OmniScript to:

Display a welcome step with customer data fetched from Salesforce (via DataRaptor).

Prompt the user to enter additional details, validated in real-time.

Integrate external data (e.g., order status from a third-party system) using an Integration Procedure. Save all data back to Salesforce or the external system upon completion.This aligns perfectly with the need for "guided interactions" using data from multiple sources.

Reference:

Salesforce OmniStudio Documentation: OmniScripts Overview – Describes OmniScripts as the tool for building guided processes with integrated data.

Salesforce OmniStudio Developer Guide: OmniScript Designer – Details how OmniScripts support Salesforce and external data sources.

Salesforce Help: Comparing OmniStudio Tools – Clarifies FlexCards for display, Integration Procedures for backend logic, and DataRaptors for data mapping, versus OmniScripts for guided interactions.

Question: 120

A company uses Expression Sets to determine product pricing. Due to the company's pricing schedules, there is always more than one Expression Set active at one time for a given date. In this scenario, how will the calculation engine select which Expression Set to run?

A. Rank
B. Date modified
C. Date created
D. Sequence

Answer: A

Explanation:

Rank (A) – The Expression Set with the highest rank is selected by the Calculation Engine when multiple active Expression Sets exist for the same date.
This ensures that the most relevant pricing rule is applied based on the company's prioritization strategy.
B . Date modified – The modification date does not determine priority for execution.
C . Date created – The system does not use the creation date to prioritize which Expression Set to execute.
D . Sequence – Sequence applies within an Expression Set but does not control selection between multiple active Expression Sets.
Comprehensive and Detailed In-Depth Why not the other options?? Salesforce OmniStudio
Reference:
Expression Sets in Calculation Procedures
OmniStudio Calculation Engine Overview

Question: 121

A Salesforce Omnistudio Consultant is working on a project to automate a complex business process for a client. The consultant decides to use an
Omniscript to guide users through the process.
What is the first step the consultant should take when creating an Omniscript?

A. Create a new Selesforce application for the script.
B. Deploy the script within a Selesforce application
C. Test the script on multiple channels and devices.
D. Define the script in the Omniscript Designer.

Answer: D

Explanation:

Define the script in the OmniScript Designer (D) – The first step in creating an OmniScript is to define it in the OmniScript Designer, where the consultant:
Outlines the flow of the script (steps, inputs, actions).

Adds UI elements (text fields, picklists, buttons).
Configures integration points (DataRaptors, Integration Procedures).
A . Create a new Salesforce application for the script – This is not required to build an OmniScript.
OmniScripts can be deployed within existing Salesforce applications.
B . Deploy the script within a Salesforce application – Deployment happens after the script is built and tested, not as the first step.
C . Test the script on multiple channels and devices – Testing is an important step after building and configuring the script.
Comprehensive and Detailed In-Depth Why not the other options?? Salesforce OmniStudio
Reference:
OmniScript Designer Guide
Creating OmniScripts

Question: 122

A consultant is working with a client who wants to display specific account information on a Flexcard. The client wants to include account status, priority level, creation date, and company logo. They also want to include actions such as closing a case, opening a new case, or creating a task.
What is the best approach for the consultant to meet the client's requirements?

A. Use Omniscripts to ,,, script that display .. required information
B. Use the Flexeard Designer to create a Flexcard, adding the necessary fields and actions.
C. Use Integration Procedures to integrate the necessary data
D. Use Data Mappers to transfer and transform the necessary data.

Answer: B

Explanation:

FlexCards (B) – FlexCards are the best choice because they are designed to display account-related information such as status, priority level, creation date, and company logo while also allowing the inclusion of actions (e.g., closing a case, opening a new case, creating a task).
FlexCards provide a user-friendly UI with clickable actions that can trigger OmniScripts or Integration Procedures to process user interactions.
A . Use OmniScripts – OmniScripts are for step-by-step guided interactions but are not primarily used for displaying static account information.
C . Use Integration Procedures – Integration Procedures fetch and transform data but do not provide a user interface. They can be used within FlexCards but are not the solution on their own.
D . Use Data Mappers – Salesforce does not have a native "Data Mappers" tool; data transformation is handled by DataRaptors or Integration Procedures.
Comprehensive and Detailed In-Depth Why not the other options?? Salesforce OmniStudio Reference:
FlexCards Overview
Configuring Actions in FlexCards

Question: 123

A Salesforce Omnistudio Consultant is tasked with creating an Omniscript for a client's customer service process. The consultant wants to integrate Salesforce Knowledge into the Omniscript.

What is one way the consultant can configure an Omniscript to search for Salesforce Knowledge articles?

A. Using a simple search ..
B. Using a complex search algorithm
C. Based on dynamic inputs from Omniscript fields
D. Based on static values set up during configuration

Answer: C

Explanation:

Based on dynamic inputs from OmniScript fields (C) – The best way to integrate Salesforce Knowledge with an OmniScript is to dynamically retrieve articles based on user inputs.

A DataRaptor Extract or Integration Procedure can fetch Knowledge Articles based on values entered in the OmniScript fields (e.g., case type, keywords).

A . Using a simple search – While a simple search is possible, OmniScripts allow for dynamic queries based on user input, making them more flexible.

B . Using a complex search algorithm – OmniStudio does not have a complex search algorithm feature built into OmniScripts.

D . Based on static values set up during configuration – Using static values would not allow for dynamic filtering based on user inputs.

Comprehensive and Detailed In-Depth Why not the other options?? Salesforce OmniStudio

Reference:

OmniScript Integration with Salesforce Knowledge

Using DataRaptors to Fetch Knowledge Articles

Question: 124

A Salesforce Omnistudio Consultant is validating an Omniscript they have developed for a client. The client has specific business requirements that the script must meet.

What is the best way for the consultant to ensure that the Omniscript meets the client's business requirements?

A. Implement the Omniscript without comparing it with the client's business requirements.
B. Assume that the Omniscript meets the client’s business requirements without conducting a review.
C. Conduct a thorough review of the Omniscript, comparing each step with the client's business requirements.
D. Ask the client to review the Omniscript with their internal team.

Answer: C

Explanation:

Conducting a thorough review (C) – The best way to ensure the OmniScript meets the business requirements is to review each step and compare it with the documented requirements.

This includes validating the data flow, user interactions, and integration points to make sure the script functions as intended.

A . Implement the OmniScript without comparing it with the client's business requirements – Skipping the review process can lead to errors and misalignment with business needs.

B . Assume that the OmniScript meets the client's business requirements without conducting a review – This is risky and unprofessional; assumptions should not replace validation.

D . Ask the client to review the OmniScript with their internal team – While client feedback is important, the consultant must first conduct their own thorough review before involving the client. Comprehensive and Detailed In-Depth Why not the other options?? Salesforce OmniStudio Reference:

OmniScript Best Practices & Validation

OmniScript Testing & Debugging

Question: 125

A consultant is working with a client who wants to use a Flexcard to display data from a specific Salesforce object. The client is unsure which data source

What advice should the consultant give to the client?

A. The client should choose a data source based on the speed of data retrieval
B. The client should choose the data source that retrieves the most data.
C. The client should choose a data source based on the number of records it can retrieve.
D. The client should choose a data source that can retrieve data from the specific Salesforce object.

Answer: D

Explanation:

A FlexCard retrieves data from a Salesforce object using a data source such as DataRaptor Extract, Integration Procedure, or a direct Salesforce Object Query.

The correct approach is to choose a data source that specifically retrieves data from the required Salesforce object (D). This ensures relevant data is fetched efficiently while maintaining performance and scalability.

A . Speed of data retrieval is important, but it depends on caching, data structure, and optimization— not just the data source itself.

B . Retrieving the most data is not always efficient; performance and relevance of data are more important.

C . While record limits matter, the primary consideration is whether the data source can retrieve the needed data from the required object.

Comprehensive and Detailed In-Depth Why not the other options?? Salesforce OmniStudio Reference:

FlexCards Data Source Options

Choosing the Right Data Source for FlexCards

Question: 126

A Salesforce Omnistudio Consultant is tasked with integrating Salesforce Knowledge into an Omniscript to allow users to search and view Salesforce Knowledge Articles.

What must be enabled in the organization for the consultant to activate Knowledge in an Omniscript?

A. Salesforce Knowledge
B. Omniscript Designer
C. Salesforce Classic

D. Salesforce Lightning

Answer: A

Explanation:

Salesforce Knowledge (A) must be enabled in the org to allow OmniScripts to retrieve and display Knowledge Articles.
Once Salesforce Knowledge is activated, OmniScripts can fetch Knowledge Articles using DataRaptors or Integration Procedures and display search results dynamically based on user inputs.

B . OmniScript Designer – This is a tool for building OmniScripts but does not enable Knowledge functionality.

C . Salesforce Classic – OmniScripts are designed for Salesforce Lightning, and enabling Classic is not required.

D . Salesforce Lightning – While OmniStudio works in Lightning, enabling Salesforce Lightning is not what activates Knowledge in OmniScripts.

Comprehensive and Detailed In-Depth Why not the other options?? Salesforce OmniStudio

Reference:

Enabling Salesforce Knowledge

Using Salesforce Knowledge in OmniScripts

Question: 127

A solution architect is working on a digital transformation project for a client. The client has complex business requirements that need to be implemented using Omnistudio tools. The solution architect needs to decide which Omnistudio tool to use.

Which Omnistudio tool should the solution architect use to automate the client's business processes and create guided interactions using data from the Salesforce org and external sources?

A. Integration procedures

B. Flexcards

C. Expression Sets

D. Omniseripts

Answer: D

Explanation:

OmniScripts (D) are designed to automate complex business processes and guide users through interactions using both Salesforce and external data sources.

They can fetch data dynamically, apply business logic, and execute actions such as record updates, API calls, or guided user inputs.

OmniScripts integrate seamlessly with DataRaptors, Integration Procedures, and FlexCards to provide

a complete business solution.

A . Integration Procedures – These are used for server-side processing and API calls, but they do not provide user interaction like OmniScripts.

B . FlexCards – These display data but do not automate guided interactions.

C . Expression Sets – These are used for calculations and business rules, but they do not provide guided interactions.
Comprehensive and Detailed In-Depth Why not the other options?? Salesforce OmniStudio
Reference:
OmniScripts Overview
Using OmniScripts to Automate Business Processes

Question: 128

Which Omniscript element retrieves Salesforce data that is then returned in Value/Label pairs and becomes available for selection in a dropdown list.

A. Lookup
B. Calculation Action
C. Date Mapper Extract Action
D. Select

Answer: D

Explanation:

Below is the formatted question with a 100% verified answer based on official Salesforce OmniStudio documentation, including a comprehensive explanation and references.

Question: 129

Which OmniScript element retrieves Salesforce data that is then returned in Value/Label pairs and becomes available for selection in a dropdown list?

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

Answer: D

Explanation:

Comprehensive and Detailed In-Depth The requirement is to identify an OmniScript element that retrieves Salesforce data, formats it as Value/Label pairs, and makes it available for selection in a

dropdown list. In Salesforce OmniStudio, the Select element is explicitly designed for this purpose, making it the correct answer.

Here's why D. Select is the correct answer:

Functionality of the Select Element: The Select element in OmniScript creates a dropdown list (or similar UI control) that allows users to choose from a set of options. It can retrieve Salesforce data directly and present it as Value/Label pairs, where the "Value" is the underlying data (e.g., a picklist value's API name or a record ID) and the "Label" is the user-friendly display text (e.g., the picklist label or a record name). The Select element supports three option source types: Manual: Hardcoded options entered by the designer.

SObject: Retrieves options dynamically from a Salesforce object field, such as a picklist or a query result.

Custom: Uses Apex or other custom logic for advanced scenarios.When configured with the "SObject" option source, the Select element queries Salesforce data (e.g., picklist values from a field like Industry on Account) and returns it as Value/Label pairs for the dropdown.

How It Works:

In the OmniScript Designer, you set the Select element's "Option Source" to "SObject" and specify the object (e.g., Case) and field (e.g., Reason). The element then pulls all active picklist values from that field (e.g., Value: Billing, Label: "Billing Issue") and populates the dropdown.

Alternatively, it can use a DataRaptor Extract to fetch a list of records (e.g., SELECT Id, Name FROM Account), where Id becomes the Value and Name becomes the Label. The retrieved data is automatically formatted as Value/Label pairs for user selection.

Meeting the Requirement: The Select element both retrieves Salesforce data (via direct SObject access or a DataRaptor) and presents it in a dropdown, fulfilling the question's criteria perfectly. Now, let's examine why the other options are incorrect:

A . Lookup: The Lookup element in OmniScript allows users to search for and select a Salesforce record (e.g., finding an Account by typing its name). While it retrieves Salesforce data and displays a list of matching records, it's designed for record selection, not for presenting a predefined set of Value/Label pairs in a dropdown. The Lookup element returns a selected record's ID and optionally other fields, but it doesn't natively format data as a dropdown list of Value/Label pairs—it's more interactive and search-driven.

B . Calculation Action: A Calculation Action performs computations or data manipulations within an OmniScript (e.g., adding numbers or setting variables). It doesn't retrieve Salesforce data on its own (that's the role of a DataRaptor) nor does it present data in a UI component like a dropdown. It's a backend action, not a user-facing element, so it doesn't meet the requirement.

C . DataRaptor Extract Action: This option (likely a typo for "DataRaptor Extract Action" given the OmniStudio context) refers to an OmniScript action that uses a DataRaptor Extract to retrieve Salesforce data. While it can fetch data and potentially structure it as Value/Label pairs (if the DataRaptor is configured to query a picklist field or map Id and Name), it's not an "element" that displays a dropdown—it's an action that supplies data to other elements (like Select). The Select element uses this data, but the DataRaptor Extract Action itself doesn't render the UI.

Key Distinction:

The Select element is the only option listed that is both an OmniScript element (a UI component) and capable of retrieving Salesforce data (either directly or via a DataRaptor) to populate a dropdown with Value/Label pairs. Other tools like DataRaptors support the process, but Select is the end-point for display and interaction.

Reference:

Salesforce OmniStudio Documentation: OmniScript Elements Reference – Details the Select element's ability to retrieve Salesforce data as Value/Label pairs for dropdowns.

Salesforce OmniStudio Developer Guide: Select Element Configuration – Explains SObject and DataRaptor integration for populating options.

Salesforce Help: OmniScript Designer – Describes how Select differs from Lookup and actions like Calculation or DataRaptor Extract.

Question: 130

A consultant is working on a project that requires the use of Omnistudio tools to automate complex business processes. The consultant is considering using Decision Matrices and Expression Sets.
What is the primary benefit of using Decision Matrices and Expression Sets in Omnistudio?

A. They allow for calculations on matrix lookups and variables.
B. They facilitate the creation of custom objects.
C. They allow for the creation of custom fields.
D. They enable the creation of custom code.

Answer: A

Explanation:

Decision Matrices and Expression Sets are used to perform calculations, apply conditional logic, and evaluate business rules in OmniStudio.
The primary benefit (A) is that they enable calculations on matrix lookups and variables.
Decision Matrices allow for defining complex rules based on multiple conditions (e.g., determining discount percentages based on customer type).
Expression Sets allow for calculations and formulas to process data dynamically.
B . Facilitate the creation of custom objects – Decision Matrices and Expression Sets do not create Salesforce objects; they only manipulate and evaluate data.
C . Allow for the creation of custom fields – Custom fields are created at the Salesforce object level, not via Decision Matrices or Expression Sets.
D . Enable the creation of custom code – OmniStudio is a low-code/no-code platform, meaning these tools are designed to replace the need for custom Apex or JavaScript.
Comprehensive and Detailed In-Depth Why not the other options?? Salesforce OmniStudio
Reference:
Decision Matrices Overview
Expression Sets in Calculation Procedures

Question: 131

A company has a process that requires minimum age to purchase. Potential customers must fill in an application form. Rather than have someone fill in the entire form and be denied at the end, 2 consultant recommends using an Omniscript that will ask users to input their birthdate. If the user is younger than than 18 years, then the Omniscript needs to display a message stating "You must be 18 years or older to purchase" and also prevent users from continuing. If the user is 18 or older, then the Omniscript should allow them to proceed without any message.
Which two Omniscript components should the using to meet this validation requirement?
Choose 2 answers

A. Alert
B. Calculation
C. Formula
D. Messaging

Answer: A,C

Explanation:

Formula (C) – The Formula element calculates whether the user is 18 years or older based on the birthdate they input.
It performs a real-time calculation using the current date minus the birthdate to determine the age. Alert (A) – The Alert element is used to display an error message when the user does not meet the age requirement.
It prevents the user from proceeding until the condition (age >= 18) is met.
B . Calculation – Calculation is used for business logic and rules processing, but a Formula element is more suitable for real-time validation within an OmniScript.
D . Messaging – The Messaging element displays information but does not prevent further progress, while an Alert stops the user from continuing.
Comprehensive and Detailed In-Depth Why not the other options?? Salesforce OmniStudio Reference:
OmniScript Formula Element
OmniScript Alert Element

Question: 132

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 returnn the quote.
Which two tools should the to meet these
Choose 2 answers

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.
Reference:

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: 133

A company needs to create a quoting process for its internal agents. During quoting, the agent selects a product that is passed to the OmniScript with product details including the unit price, and specifies the grade (A, B, C, D, E) and 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 return the quote. Which two tools should be used to meet these requirements?
Choose 2 answers

A. DataRaptor Transform
B. Expression Set

C. Decision Matrix
D. DataRaptor Extract

Answer: C,D

Explanation:

Comprehensive and Detailed In-Depth 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.
Reference:

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: 134

A Salesforce OmniStudio Consultant is creating an OmniScript for a client's customer service process. The consultant wants to enhance the performance and usability of the OmniScript. What is one best practice the consultant can follow to achieve this?

A. Use the Salesforce Lightning App Builder.
B. Use custom Lightning Web Components with LWC OmniScripts.
C. Use Angular OmniScripts.
D. Use the Classic OmniScript Designer.

Answer: B

Explanation:

Comprehensive and Detailed In-Depth The goal is to enhance the performance and usability of an OmniScript for a customer service process. In Salesforce OmniStudio, a key best practice for achieving this is B. Use custom Lightning Web Components (LWC) with LWC OmniScripts, as it leverages modern web standards, improves performance, and enhances usability through tailored functionality.

Here's why B is the correct answer:

LWC OmniScripts Overview: OmniStudio supports two types of OmniScripts: Standard OmniScripts (built on AngularJS) and LWC OmniScripts (built on Lightning Web Components). LWC OmniScripts are the newer, recommended approach, offering better performance due to their lightweight framework and Salesforce's native LWC architecture.
Custom LWCs: By integrating custom Lightning Web Components into an LWC OmniScript, the consultant can:
Enhance Performance: LWCs are faster than Angular-based components because they use the browser's native Web Components standard, reducing overhead and improving load times.
Improve Usability: Custom LWCs allow for highly tailored UI elements or logic (e.g., a dynamic form or real-time validation) that go beyond standard OmniScript elements, making the customer service process more intuitive and efficient.
Example: A custom LWC could display a collapsible FAQ section or a real-time chat widget within the OmniScript, improving the agent's workflow.
Best Practice Alignment: The OmniStudio documentation explicitly recommends transitioning to LWC OmniScripts for new implementations and using custom LWCs to extend functionality, as they align with Salesforce's long-term platform strategy and provide a responsive, scalable solution.
Now, let's examine why the other options are incorrect:

A . Use the Salesforce Lightning App Builder: Lightning App Builder is a tool for creating Lightning pages, not OmniScripts. While it can embed an OmniScript via a component, it's not a best practice for enhancing the OmniScript itself—it's a deployment mechanism, not a development approach. C . Use Angular OmniScripts: Standard OmniScripts (based on AngularJS) are the older framework and are being phased out in favor of LWC OmniScripts. They are less performant due to AngularJS's heavier footprint and lack the flexibility of LWCs, making them a poor choice for enhancing performance and usability.
D . Use the Classic OmniScript Designer: There's no "Classic OmniScript Designer" in OmniStudio— Classic refers to Salesforce's pre-Lightning UI. The OmniScript Designer is a Lightning-based tool, and reverting to older Salesforce paradigms contradicts performance and usability goals.

Reference:

Salesforce OmniStudio Documentation: LWC OmniScripts – Highlights performance benefits and custom LWC integration.
Salesforce OmniStudio Developer Guide: Best Practices – Recommends LWC OmniScripts for modern development.

Question: 135

A consultant is designing a Flexcard for a client. The client wants the Flexcard to launch a child card with additional information and actions related to the parent card when the agent clicks on a button.
How should the consultant design the Flexcard to meet the client's requirements?

A. Design the parentcard tolaunch a child card when the specific action is taken.
B. Design the parentcard tolaunch a child card through a Flyout Action.
C. Design the parentcard tolaunch a new Data Mapper when the specific action istaken.
D. Design the parentcard tolaunch a new Omniscript when the specific action is taken,

Answer: B

Explanation:

The requirement is for a FlexCard (parent card) to launch a child card with additional information and actions when an agent clicks a button. In Salesforce OmniStudio, the Flyout Action is the specific mechanism designed within FlexCards to achieve this, making B the most precise and correct answer.

Here's why B. Design the parent card to launch a child card through a Flyout Action is the correct answer:

Flyout Action Overview: A Flyout Action in FlexCards is an interactive feature that displays a pop-up panel (flyout) when triggered, typically by a button or link. This flyout can contain a child FlexCard, OmniScript, or custom content, showing additional details or enabling actions related to the parent card's data.
Meeting the Requirement:
Child Card Launch: The Flyout Action can embed a child FlexCard, which displays supplemental information (e.g., related records or details) and includes actions (e.g., buttons to update data). Button Trigger: In the FlexCard Designer, the consultant can add a Button element to the parent card, configure its action type as "Flyout," and link it to a child FlexCard. When the agent clicks the button, the flyout appears with the child card.
Contextual Data: The parent card's data (e.g., a record ID) can be passed to the child card via the Flyout Action's context parameters, ensuring the child card shows relevant information.
Example: If the parent card displays a customer's summary (name, account number), clicking the button could launch a child card in a flyout showing order history and a button to initiate a return, all tied to the parent card's account ID.
Now, let's examine why the other options are incorrect or less optimal:

A . Design the parent card to launch a child card when the specific action is taken: While this is

conceptually correct, it's vague and lacks specificity. "Launch a child card" isn't a defined OmniStudio action—Flyout Action is the actual mechanism to achieve this. Thus, B is the more precise answer, as it names the tool explicitly supported by FlexCards.

C . Design the parent card to launch a new DataRaptor when the specific action is taken: A DataRaptor (e.g., DataRaptor Extract) retrieves or manipulates data, not displays a UI like a child card. While a DataRaptor might fetch data for the child card, it doesn't "launch" anything visible to the agent, making this option irrelevant to the UI requirement.

D . Design the parent card to launch a new OmniScript when the specific action is taken: An OmniScript could be launched via a FlexCard action (e.g., an "OmniScript" action type), but it's a guided process, not a "child card." The requirement specifies a child card (implying another FlexCard), not a multi-step script, so a Flyout Action with a child FlexCard is more appropriate than an OmniScript.

Reference:

Salesforce OmniStudio Documentation: FlexCard Actions – Details the Flyout Action for launching child cards or content.
Salesforce OmniStudio Developer Guide: Flyouts in FlexCards – Explains configuring flyouts with child FlexCards.

Question: 136

A consultant is designing a Flexcard for a client. The client wants to customize the look of each element on the Flexcard. What tool should the consultant use to meet the client's requirements?

A. Use Data Mapper to customize the look of each element.
B. Use Integration Procedures to customize the look of each element.
C. Use the style panel in the Flexcard Designer to customize the look of each element.
D. Use Omniscripts to customize the look of each element.

Answer: C

Explanation:

Below are the formatted questions with 100% verified answers based on official Salesforce OmniStudio documentation, including comprehensive explanations and references.

Question: 137

A consultant is designing a FlexCard for a client. The client wants the FlexCard to launch a child card with additional information and actions related to the parent card when the agent clicks on a button.
How should the consultant design the FlexCard to meet the client's requirements?

A. Design the parent card to launch a child card when the specific action is taken.
B. Design the parent card to launch a child card through a Flyout Action.

C. Design the parent card to launch a new DataRaptor when the specific action is taken. D. Design the parent card to launch a new OmniScript when the specific action is taken.

Answer: B

Explanation:

Comprehensive and Detailed In-Depth The requirement is for a FlexCard (parent card) to launch a child card with additional information and actions when an agent clicks a button. In Salesforce OmniStudio, the Flyout Action is the specific mechanism designed within FlexCards to achieve this, making B the most precise and correct answer.

Here's why B. Design the parent card to launch a child card through a Flyout Action is the correct answer:

Flyout Action Overview: A Flyout Action in FlexCards is an interactive feature that displays a pop-up panel (flyout) when triggered, typically by a button or link. This flyout can contain a child FlexCard, OmniScript, or custom content, showing additional details or enabling actions related to the parent card's data.
Meeting the Requirement:
Child Card Launch: The Flyout Action can embed a child FlexCard, which displays supplemental information (e.g., related records or details) and includes actions (e.g., buttons to update data). Button Trigger: In the FlexCard Designer, the consultant can add a Button element to the parent card, configure its action type as "Flyout," and link it to a child FlexCard. When the agent clicks the button, the flyout appears with the child card.
Contextual Data: The parent card's data (e.g., a record ID) can be passed to the child card via the Flyout Action's context parameters, ensuring the child card shows relevant information.
Example: If the parent card displays a customer's summary (name, account number), clicking the button could launch a child card in a flyout showing order history and a button to initiate a return, all tied to the parent card's account ID.
Now, let's examine why the other options are incorrect or less optimal:

A . Design the parent card to launch a child card when the specific action is taken: While this is conceptually correct, it's vague and lacks specificity. "Launch a child card" isn't a defined OmniStudio action—Flyout Action is the actual mechanism to achieve this. Thus, B is the more precise answer, as it names the tool explicitly supported by FlexCards.
C . Design the parent card to launch a new DataRaptor when the specific action is taken: A DataRaptor (e.g., DataRaptor Extract) retrieves or manipulates data, not displays a UI like a child card. While a DataRaptor might fetch data for the child card, it doesn't "launch" anything visible to the agent, making this option irrelevant to the UI requirement.
D . Design the parent card to launch a new OmniScript when the specific action is taken: An OmniScript could be launched via a FlexCard action (e.g., an "OmniScript" action type), but it's a guided process, not a "child card." The requirement specifies a child card (implying another FlexCard), not a multi-step script, so a Flyout Action with a child FlexCard is more appropriate than an OmniScript.
Reference:

Salesforce OmniStudio Documentation: FlexCard Actions – Details the Flyout Action for launching child cards or content.
Salesforce OmniStudio Developer Guide: Flyouts in FlexCards – Explains configuring flyouts with child FlexCards.

Question: 138

A consultant is designing a FlexCard for a client. The client wants to customize the look of each element on the FlexCard. What tool should the consultant use to meet the client's requirements?

A. Use DataRaptor to customize the look of each element.

B. Use Integration Procedures to customize the look of each element.
C. Use the style panel in the FlexCard Designer to customize the look of each element.
D. Use OmniScripts to customize the look of each element.

Answer: C

Explanation:

Comprehensive and Detailed In-Depth The requirement is to customize the appearance (look) of each element on a FlexCard. In Salesforce OmniStudio, the Style Panel in the FlexCard Designer is the dedicated tool for this purpose, making C the correct answer.

Here's why C. Use the style panel in the FlexCard Designer to customize the look of each element is the correct answer:

Style Panel Overview: The FlexCard Designer includes a Style Panel (accessed in the properties pane) that allows consultants to customize the visual properties of individual elements (e.g., Text, Image, Button, Datatable) on a FlexCard. This includes settings like:
Font size, color, and weight.
Background color or image.
Borders (style, width, color).
Padding and margins.
Custom CSS for advanced styling.
Meeting the Requirement:
Element-Level Customization: The Style Panel enables granular control over each element. For example, a Text element displaying a customer name can be styled with a bold red font, while a Button can have a blue background and rounded corners.
Ease of Use: It's a no-code solution within the FlexCard Designer, requiring no external tools or coding expertise, aligning with OmniStudio's low-code philosophy.
Preview: Changes made in the Style Panel are instantly previewable in the designer, ensuring the client's vision is met efficiently.
Example: If the FlexCard shows case details, the consultant could use the Style Panel to make the case number bold and blue, add a red border to an "Open" status badge, and adjust the Datatable's row height—all tailored to the client's preferences.
Now, let's examine why the other options are incorrect:
A . Use DataRaptor to customize the look of each element: A DataRaptor (e.g., DataRaptor Extract or Transform) handles data retrieval and manipulation, not visual styling. It provides the data displayed on the FlexCard but has no role in customizing the appearance of elements.
B . Use Integration Procedures to customize the look of each element: Integration Procedures orchestrate backend logic and API calls, not UI styling. They can fetch data or process actions but don't influence the look of FlexCard elements.
D . Use OmniScripts to customize the look of each element: OmniScripts are for guided processes, not for designing or styling FlexCards. While OmniScripts have their own styling options, they're separate from FlexCards and irrelevant to this requirement.
Reference:

Salesforce OmniStudio Documentation: FlexCard Designer – Describes the Style Panel for customizing element appearance.

Salesforce OmniStudio Developer Guide: Styling FlexCards – Details CSS and style options in the designer.

Question: 139

Which Omniscript element enables users to choose from a dropdown list?

A. Select
B. Calculation Action
C. Lookup
D. Data Mapper Extract Action

Answer: A

Explanation:

The requirement is to identify an OmniScript element that enables users to choose from a dropdown list. In Salesforce OmniStudio, the Select element is specifically designed for this purpose, making A the correct answer.

Here's why A. Select is the correct answer:

Select Element Overview: The Select element in OmniScript creates a dropdown list (or similar UI control like a radio button group, depending on settings) that allows users to pick one option from a predefined set. It's a user-facing input element that supports:
Manual Options: Hardcoded values entered in the designer.
SObject Options: Values retrieved from a Salesforce field (e.g., picklist values).
DataRaptor/Custom Options: Dynamic values from a DataRaptor Extract or Apex.
Dropdown Functionality: By default, when configured as a "Dropdown" in the Style settings, the Select element renders as a dropdown menu, enabling users to choose from a list (e.g., selecting a Case Priority like "High," "Medium," "Low").
Meeting the Requirement: The Select element directly fulfills the need for a dropdown list, providing a simple, interactive way for users to make a selection within an OmniScript.

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

B . Calculation Action: A Calculation Action performs backend computations or data manipulations (e.g., multiplying values or setting variables). It's not a UI element and doesn't present a dropdown list for user interaction.
C . Lookup: The Lookup element allows users to search for and select a Salesforce record (e.g., an Account) via a searchable popup. While it involves selection, it's not a dropdown list—it's a dynamic search interface that returns a record, not a predefined list of options.
D . DataRaptor Extract Action: This action (correcting the typo "Data Mapper Extract Action") retrieves Salesforce data using a DataRaptor Extract, but it's a backend process, not a UI element. It can supply data to a Select element for a dropdown, but it doesn't enable user selection itself.

Reference:

Salesforce OmniStudio Documentation: OmniScript Elements Reference – Details the Select element's dropdown capabilities.
Salesforce OmniStudio Developer Guide: Select Element – Explains configuration for dropdown lists.

Question: 140

A consultant is tasked with migrating Calculation Matrices and Procedures as Decision Matrices and Expression Sets respectively.
What is a key consideration when migrating Calculation Matrices and Procedures?

A. Procedures can be migrated without considerations.
B. Matrices can be migrated with minimal considerations.
C. Matrices can be migrated without considerations.
D. Looping procedures cannot be migrated as expression sets.

Answer: D

Explanation:

The task involves migrating legacy Calculation Matrices to Decision Matrices and Calculation Procedures to Expression Sets in OmniStudio. A key consideration arises from the differences in functionality between these tools, particularly with looping logic, making D the correct answer.

Here's why D. Looping procedures cannot be migrated as Expression Sets is the correct answer:

Migration Context:
Calculation Matrices → Decision Matrices: Calculation Matrices (from older Vlocity tools) are lookup tables that map inputs to outputs. They migrate to Decision Matrices in OmniStudio, which serve the same purpose (e.g., returning a discount based on inputs) and are generally straightforward to convert, as both are data-driven tables stored as Salesforce metadata.
Calculation Procedures → Expression Sets: Calculation Procedures are sequential, logic-driven processes that can include formulas, conditions, and loops. They migrate to Expression Sets in OmniStudio, which are collections of reusable expressions (formulas or conditions) executed in a specific order.

Key Consideration – Looping:
Calculation Procedures: These legacy procedures support looping logic, such as iterating over a list of items (e.g., processing multiple line items in a quote to calculate totals).
Expression Sets: Expression Sets in OmniStudio are designed for linear, non-iterative calculations.
They execute a series of expressions (e.g., Total = Price * Quantity) but do not natively support looping constructs like "for each" or "while" loops. Looping requires an Integration Procedure or custom Apex in OmniStudio, not Expression Sets.
Impact: If a Calculation Procedure includes looping (e.g., summing values across a JSON array), it cannot be directly migrated to an Expression Set without rearchitecting the logic. The consultant must consider alternative tools (e.g., an Integration Procedure with a Loop Block) to handle such cases.
Why It's Key: Failing to account for looping can break the migrated process, as Expression Sets lack this capability. Identifying and addressing looping logic upfront is critical to ensure functional equivalence post-migration.
Now, let's examine why the other options are incorrect:

A . Procedures can be migrated without considerations: This is false. Calculation Procedures often include complex logic (e.g., loops, branching), and Expression Sets have limitations (e.g., no looping), requiring careful analysis and potential redesign during migration.

B . Matrices can be migrated with minimal considerations: While migrating Calculation Matrices to Decision Matrices is relatively straightforward (both are lookup tables), “minimal considerations” understates potential issues like data mapping or deprecated features, though looping isn’t a factor here. This isn’t the key consideration compared to D.
C . Matrices can be migrated without considerations: This is incorrect. Even for matrices, considerations like field mappings, data type compatibility, or version differences exist, though they’re less complex than procedure migrations.
Reference:

Salesforce OmniStudio Documentation: Migration Guide – Discusses migrating Calculation Matrices and Procedures, noting limitations like looping in Expression Sets.
Salesforce OmniStudio Developer Guide: Expression Sets – Confirms Expression Sets lack looping, unlike Integration Procedures.

Question: 141

You want to update 500 Leads through a Data Loader. In the CSV file, you have three columns (ID, Email, and Phone). Also, there are some blank values in the rows (some records do not have email, and some do not have a phone). You do not want null values in the records when updating the values. Which of the following options should you use to do this?

A. Open Data Loader --> Setting --> Uncheck the "Insert Null Values" checkbox.
B. Null values cannot be ignored. If they are present in CSV, they get updated as null.
C. Open Data Loader --> Setting --> Check the "Ignore Null Values" checkbox.
D. Modify the CSV file and remove the records which have null values, and update them manually. E. Null values do not get updated through Data Loader.

Answer: A

Explanation:

The requirement is to update 500 Leads using Data Loader, ensuring that blank (null) values in the CSV file (for Email or Phone) do not overwrite existing values in Salesforce records. The Salesforce Data Loader provides a specific setting to control this behavior, making A the correct answer.

Here’s why A. Open Data Loader --> Setting --> Uncheck the "Insert Null Values" checkbox is the correct answer:

Data Loader Behavior: By default, when you update records via Data Loader, blank values in the CSV file are treated as nulls and will overwrite the corresponding fields in Salesforce with null, replacing any existing data. For example, if a Lead record has Phone = "555-1234" and the CSV has a blank Phone column for that ID, the update will set Phone = null unless configured otherwise.
Insert Null Values Setting: The "Insert Null Values" checkbox in Data Loader’s Settings (found under Settings > Settings) determines whether blank CSV values are treated as nulls.
Checked: Blank values in the CSV overwrite existing field values with null (default behavior). Unchecked: Blank values in the CSV are ignored, and the existing field values in Salesforce are preserved.
Meeting the Requirement: Unchecking "Insert Null Values" ensures that if a row in the CSV has a blank Email or Phone, those fields in the corresponding Lead record remain unchanged (e.g., retaining Email = "john@example.com" instead of setting it to null). Only non-blank values in the CSV (e.g., a new Phone number) will update the records.
Process: Open Data Loader, go to Settings > Settings, uncheck "Insert Null Values," then proceed with the Update operation using the CSV file with ID, Email, and Phone columns.
Now, let’s examine why the other options are incorrect:

B . Null values cannot be ignored. If they are present in CSV, they get updated as null: This is false. Data Loader provides the "Insert Null Values" setting specifically to ignore nulls when unchecked, contradicting this option.
C . Open Data Loader --> Setting --> Check the "Ignore Null Values" checkbox: There's no "Ignore Null Values" checkbox in Data Loader Settings. The relevant option is "Insert Null Values," which must be unchecked (not checked) to ignore nulls, making this option incorrect due to inaccurate terminology and logic.
D . Modify the CSV file and remove the records which have null values, and update them manually: This is a workaround, not a best practice. Manually editing 500 records is inefficient and error-prone when Data Loader's built-in setting can handle this automatically.
E . Null values do not get updated through Data Loader: This is false. By default, null values do update fields unless the "Insert Null Values" setting is unchecked.
Reference:

Salesforce Data Loader Guide: Settings – Details the "Insert Null Values" option and its impact on updates.
Salesforce Help: Updating Records with Data Loader – Explains handling null values in CSV imports/updates.

Question: 142

Topic: Salesforce API
You want to display Chatter feeds, users, groups, and followers, especially in mobile applications. Also, you want to provide programmatic access to files, recommendations, topics, notifications, and Data.com purchasing. Which of the following option should you use to do this?

A. Streaming API
B. REST API
C. Tooling API
D. Chatter REST AP!
E. Chatter SOAP API

Answer: D

Explanation:

The requirement involves displaying Chatter-related data (feeds, users, groups, followers) in mobile applications and providing programmatic access to additional features like files, recommendations, topics, notifications, and Data.com purchasing. The Chatter REST API is the most suitable Salesforce API for this purpose, making D the correct answer.

Here's why D. Chatter REST API is the correct answer:

Chatter REST API Overview: The Chatter REST API is a specialized subset of the Salesforce REST API, optimized for social collaboration features. It provides access to Chatter-specific functionality (e.g., feeds, posts, comments, groups, users, followers) and extends to related features like files (Chatter Files), recommendations, topics, notifications, and even Data.com purchasing integrations.
Meeting the Requirements:
Chatter Feeds, Users, Groups, Followers: The Chatter REST API offers endpoints like /chatter/feeds, /chatter/users, /chatter/groups, and /chatter/users/{id}/followers to retrieve and display this data in a mobile app.
Files: Use /chatter/files or /connect/files endpoints to access and manage Chatter Files. Recommendations: Access Chatter recommendations via /connect/recommendations.

Topics: Manage topics with /connect/topics.
Notifications: Handle notifications via /connect/notifications.
Data.com Purchasing: The API supports Data.com-related calls (e.g., purchasing contact data) through specific endpoints like /connect/datacom.
Mobile Optimization: The REST-based architecture of the Chatter REST API is lightweight, uses JSON (or XML), and is ideal for mobile applications due to its stateless, HTTP-based design, aligning with the requirement's emphasis on mobile use.
Programmatic Access: It provides full CRUD (create, read, update, delete) capabilities for Chatter data and related features, meeting the need for programmatic control.
Now, let's examine why the other options are incorrect:

A . Streaming API: The Streaming API is for real-time push notifications (e.g., subscribing to record updates via PushTopics or Platform Events). It's not designed for retrieving Chatter feeds, files, or Data.com data, nor is it suited for general mobile display purposes—it's event-driven, not data-access-focused.
B . REST API: The broader Salesforce REST API (e.g., /services/data/vXX.X/) can access some Chatter data (e.g., via sObjects like FeedItem), files, and notifications, but it's not specialized for Chatter or Data.com purchasing. The Chatter REST API is a tailored extension of this, offering more specific endpoints and better usability for Chatter-related tasks, making it the better fit.
C . Tooling API: The Tooling API is for managing metadata and developer tools (e.g., Apex classes, triggers), not for accessing Chatter data, files, or Data.com features. It's irrelevant to this requirement.
E . Chatter SOAP API: There's no "Chatter SOAP API" in Salesforce—SOAP API exists for general sObject operations, but Chatter-specific features are handled via REST (Chatter REST API). SOAP is less mobile-friendly due to its XML-heavy, stateful nature, making this option incorrect and outdated. Reference:

Salesforce Chatter REST API Developer Guide: Overview – Lists supported features like feeds, files, topics, and Data.com integration.
Salesforce REST API Developer Guide: Chatter Resources – Details Chatter-specific endpoints in the REST framework.

Question: 143

Which of the following is a key difference between Integration Procedures and DataRaptors?
A. DataRaptors can access data from external sources.
B. DataRaptors can invoke multiple actions in a single server call.
C. Integration Procedures can access data from external sources.
D. Integration Procedures can retrieve data from multiple related objects.

Answer: C

Explanation:
The key difference between Integration Procedures and DataRaptors is that Integration Procedures can access data from external sources, while DataRaptors can only access data from Salesforce. Integration Procedures allow the designer to create complex API queries declaratively using HTTP Actions or SOAP Actions. DataRaptors allow the designer to read, transform, and write data between Salesforce objects or fields. Both Integration Procedures and DataRaptors can invoke multiple actions in a single server call, and both can retrieve data from multiple related objects.