Version: 22.0

Question: 1

Cloud Kicks has the following requirements:

• Their Shipment custom object must always relate to a Product, a Sender, and a Receiver (all separate custom objects).

• If a Shipment is currently associated with a Product, Sender, or Receiver, deletion of those records should not be allowed.

• Each custom object must have separate sharing models.

What should an Architect do to fulfill these requirements?

A. Associate the Shipment to each parent record by using a VLOOKUP formula field.

B. Create a required Lookup relationship to each of the three parent records.

C. Create a Master-Detail relationship to each of the three parent records.

D. Create two Master-Detail and one Lookup relationship to the parent records.

Answer: B

E xpl anati on:

A required Lookup relationship ensures that the Shipment record must have a value for each of the three parent records, and also prevents the deletion of those parent records if they are referenced by a Shipment record. A Master-Detail relationship would not allow separate sharing models for each custom object, and a VLOOKUP formula field would not enforce the relationship or prevent deletion

Question: 2

Universal Containers (UC) is planning to move away from legacy CRM to Salesforce. As part of one-time data migration, UC will need to keep the original date when a contact was created in the legacy system. How should an Architect design the data migration solution to meet this requirement?

A. After the data is migrated, perform an update on all records to set the original date in a standard Created Date field.
B. Create a new field on the Contact object to capture the Created Date. Hide the standard Created Date field using Field -Level Security.
C. Enable "Set Audit Fields" and assign the permission to the user loading the data for the duration of the migration.
D. Write an Apex trigger on the Contact object, before insert event to set the original value in a standard Created Date field.

Answer: C

E xpl anati on:

Enabling “Set Audit Fields” allows the user loading the data to set the value of the standard CreatedDate field to match the original date from the legacy system. This is a one-time permission that can be revoked after the migration is completed. The other options would either not work or require additional customization

Question: 3

An architect has been asked to provide error messages when a future date is detected in a custom Birthdate _c field on the Contact object. The client wants the ability to translate the error messages. What are two approaches the architect should use to achieve this solution? Choose 2 answers

A. Implement a third -party validation process with translate functionality.
B. Create a trigger on Contact and add an error to the record with a custom label.
C. Create a workflow field update to set the standard ErrorMessage field.
D. Create a validation rule and translate the error message with translation workbench.

Answer: B, D

E xpl anati on:

Creating a trigger on Contact and adding an error to the record with a custom label allows the architect to use the translation workbench to translate the error message based on the user's language. Creating a validation rule and translating the error message with translation workbench also achieves the same result. The other options would either not provide translation functionality or not display an error message

Question: 4

What is an advantage of using Custom metadata type over Custom setting?

A. Custom metadata records are not copied from production to sandbox.
B. Custom metadata types are available for reporting.
C. Custom metadata records are deployable using packages.
D. Custom metadata records are editable in Apex.

Answer: C

E xpl anati on:

Custom metadata records are deployable using packages, which makes them easier to migrate from one environment to another. Custom settings records are not deployable using packages, and they are copied from production to sandbox. Custom metadata types are not available for reporting, and custom metadata records are not editable in Apex.

Question: 5

Get Cloudy Consulting uses an invoicing system that has specific requirements. One requirement is that attachments associated with the Invoice_c custom object be classified by Types (i.e., ""Purchase Order"", ""Receipt"", etc.) so that reporting can be performed on invoices showing the number of attachments grouped by Type.

What should an Architect do to categorize the attachments to fulfill these requirements?

A. Add additional options to the standard ContentType picklist field for the Attachment object.

B. Add a ContentType picklist field to the Attachment layout and create additional picklist options.

C. Create a custom picklist field for the Type on the standard Attachment object with the values.

D. Create a custom object related to the Invoice object with a picklist field for the Type.

Answer: D

E xpl anati on:

Creating a custom object related to the Invoice object with a picklist field for the Type allows the architect to categorize the attachments and report on them by Type. The standard Attachment object does not have a ContentType picklist field, and adding a custom picklist field to it would not be best practice.