

Boost up Your Certification Score

Salesforce Plat-Arch-201

Salesforce Certified Platform Data Architect



For More Information – Visit link below:

<https://www.examsboost.com/>

Product Version

- ✓ Up to Date products, reliable and verified.
- ✓ Questions and Answers in PDF Format.

Visit us at: <https://www.examsboost.com/test/plat-arch-201>

Latest Version: 6.0

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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

Explanation:

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.

Question: 6

Universal Containers has a legacy system that captures Conferences and Venues. These Conferences can occur at any Venue. They create hundreds of thousands of Conferences per year. Historically, they have only used 20 Venues. Which two things should the data architect consider when denormalizing this data model into a single Conference object with a Venue picklist? Choose 2 answers

- A. Limitations on master -detail relationships.
- B. Org data storage limitations.
- C. Bulk API limitations on picklist fields.
- D. Standard list view in -line editing.

Answer: C D

Explanation:

When denormalizing a data model into a single object with a picklist field, the data architect should consider the Bulk API limitations on picklist fields and the standard list view in-line editing. The Bulk API has a limit of 1,000 distinct picklist values per file¹, which could be an issue if there are more than 1,000 venues in the future. The standard list view in-line editing allows users to edit multiple records at once, which could introduce data quality issues if the venue picklist is not validated or restricted². The other options are not relevant to denormalizing a data model.

Question: 7

Universal Container (UC) has around 200,000 Customers (stored in Account object). They get 1 or 2 Orders every month from each Customer. Orders are stored in a custom object called "Order c"; this has about 50 fields. UC is expecting a growth of 10% year-over-year. What are two considerations an architect should consider to improve the performance of SOQL queries that retrieve data from the Order_c object? Choose 2 answers

- A. Use SOQL queries without WHERE conditions.
- B. Work with Salesforce Support to enable Skinny Tables.
- C. Reduce the number of triggers on Order_c object.
- D. Make the queries more selective using indexed fields.

Answer: B D

Explanation:

To improve the performance of SOQL queries that retrieve data from the Order_c object, the data architect should work with Salesforce Support to enable Skinny Tables and make the queries more selective using indexed fields. Skinny Tables are custom tables that contain frequently used fields and are kept in sync with the base tables³. They can improve performance by reducing the number of table joins and using indexes. Making the queries more selective using indexed fields can also improve performance by reducing the query execution time and avoiding query timeouts⁴. The other options are not effective or recommended for improving SOQL performance.

Question: 8

Universal Containers (UC) provides shipping services to its customers. They use Opportunities to track customer shipments. At any given time, shipping status can be one of the 10 values. UC has 200,000 Opportunity records. When creating a new field to track shipping status on opportunity, what should the architect do to improve data quality and avoid data skew?

- A. Create a picklist field, values sorted alphabetically.

- B. Create a Master -Detail to custom object ShippingStatus c.
- C. Create a Lookup to custom object ShippingStatus c.
- D. Create a text field and make it an external ID.

Answer: A

Explanation:

To improve data quality and avoid data skew, the data architect should create a picklist field with values sorted alphabetically for tracking shipping status on opportunity. A picklist field ensures that only valid values are entered and prevents typos or variations in spelling. Sorting the values alphabetically makes it easier for users to find and select the correct value. Data skew occurs when a large number of records are owned by a single user or have a single value for a field. Creating a picklist field with a limited number of values does not cause data skew, as long as the distribution of values is balanced and not skewed towards one value.

Question: 9

Universal Containers (UC) management has identified a total of ten text fields on the Contact object as important to capture any changes made to these fields, such as who made the change, when they made the change, what is the old value, and what is the new value. UC needs to be able to report on these field data changes within Salesforce for the past 3 months. What are two approaches that will meet this requirement? Choose 2 answers

- A. Create a workflow to evaluate the rule when a record is created and use field update actions to store previous values for these ten fields in ten new fields.
- B. Write an Apex trigger on Contact after insert event and after update events and store the old values in another custom object.
- C. Turn on field Contact object history tracking for these ten fields, then create reports on contact history.
- D. Create a Contact report including these ten fields and Salesforce Id, then schedule the report to run once a day and send email to the admin.

Answer: B C

Explanation:

To capture and report on any changes made to ten text fields on the Contact object for the past 3 months, the data architect should write an Apex trigger on Contact after insert and after update events and store the old values in another custom object, or turn on field Contact object history tracking for these ten fields and create reports on contact history. An Apex trigger can capture the old and new values of the fields, as well as the user and time of the change, and store them in a custom object that can be used for reporting. Field history tracking can also track the changes to the fields and store them in a history table that can be used for reporting. However, field history tracking only retains data for up to 18 months or 24 months with an

extension, so it may not be suitable for longer-term reporting needs. The other options are not feasible or effective for capturing and reporting on field data changes.

Question: 10

Universal Containers (UC) has an open sharing model for its Salesforce users to allow all its Salesforce internal users to edit all contacts, regardless of who owns the contact. However, UC management wants to allow only the owner of a contact record to delete that contact. If a user does not own the contact, then the user should not be allowed to delete the record. How should the architect approach the project so that the requirements are met?

- A. Create a "before delete" trigger to check if the current user is not the owner.
- B. Set the Sharing settings as Public Read Only for the Contact object.
- C. Set the profile of the users to remove delete permission from the Contact object.
- D. Create a validation rule on the Contact object to check if the current user is not the owner.

Answer: A

Explanation:

To allow only the owner of a contact record to delete that contact, the data architect should create a "before delete" trigger to check if the current user is not the owner. The trigger can use the `UserInfo.getUserId()` method to get the current user's ID and compare it with the `OwnerId` field of the contact record. If they are not equal, the trigger can add an error to the record and prevent it from being deleted. The other options are not suitable for meeting the requirements, as they would either restrict the edit access or delete access for all users, regardless of ownership.

Thank You for Trying Our Product

For More Information – **Visit link below:**

<https://www.examsboost.com/>

15 USD Discount Coupon Code:

G74JA8UF

FEATURES

- ✓ **90 Days Free Updates**
- ✓ **Money Back Pass Guarantee**
- ✓ **Instant Download or Email Attachment**
- ✓ **24/7 Live Chat Support**
- ✓ **PDF file could be used at any Platform**
- ✓ **50,000 Happy Customer**



Visit us at: <https://www.examsboost.com/test/plat-arch-201>