



**Free Questions for [Data-Architect](#) by [braindumpscollection](#)**

**Shared by [Newman](#) on [15-04-2024](#)**

**For More Free Questions and Preparation Resources**

**[Check the Links on Last Page](#)**

# Question 1

---

**Question Type:** MultipleChoice

---

UC migrating 100,000 Accounts from an enterprise resource planning (ERP) to salesforce and is concerned about ownership skew and performance.

Which 3 recommendations should a data architect provide to prevent ownership skew?

Choose 3 answers:

## Options:

---

- A- Assigned a default user as owner of accounts, and assign role in hierarchy.
- B- Keep users out of public groups that can be used as the source for sharing rules.
- C- Assign a default user as owner of account and do not assign any role to default user.
- D- Assign "view all" permission on profile to give access to account.
- E- Assign a default user as owner of accounts and assigned top most role in hierarchy.

## Answer:

---

B, C, E

## Explanation:

---

According to the Salesforce documentation<sup>1</sup>, ownership skew occurs when a large number of records (more than 10,000) are owned by a single user or queue. This can cause performance issues and lock contention when multiple users try to access or update those records. To prevent ownership skew, some of the recommended practices are:

Assign a default user as the owner of the records and do not assign any role to the default user (option C). This way, the records will not be visible to other users in the role hierarchy and will not cause sharing recalculations.

Keep users out of public groups that can be used as the source for sharing rules (option B). Sharing rules based on public groups can cause excessive sharing calculations and lock contention when many records are owned by a single user or queue.

Assign a default user as the owner of the records and assign the top most role in the hierarchy to the default user (option E). This way, the records will be visible to all users in the role hierarchy, but will not cause sharing recalculations or lock contention.

Assigning a default user as the owner of the records and assigning a role in the hierarchy (option A) is not a good practice, as it can cause sharing recalculations and lock contention when the role is updated or moved. Assigning "view all" permission on profile to give access to the records (option D) is also not a good practice, as it can bypass the security and sharing model and expose sensitive data to unauthorized users.

## Question 2

---

**Question Type:** MultipleChoice

---

Universal Containers (UC) owns a complex Salesforce org with many Apex classes, triggers, and automated processes that will modify records if available. UC has identified that, in its current development state, UC runs change of encountering race condition on the same record.

What should a data architect recommend to guarantee that records are not being updated at the same time?

### Options:

---

- A- Embed the keywords FOR UPDATE after SOQL statements.
- B- Disable classes or triggers that have the potential to obtain the same record.
- C- Migrate programmatic logic to processes and flows.
- D- Refactor or optimize classes and trigger for maximum CPU performance.

### Answer:

---

A

### Explanation:

---

According to the [How to avoid row lock or race condition in Apex](#) blog post, one of the ways to prevent race condition in Apex is to use the FOR UPDATE keyword in SOQL statements. The blog post states that "We need to lock the records on which we are working such that other batches or threads will not be having any effect on them. How can we lock a record, then? We need to make use of FOR UPDATE keyword in the SOQL query." Therefore, a data architect should recommend this solution to guarantee that records are not being

updated at the same time by different processes.

## Question 3

---

**Question Type:** MultipleChoice

---

A custom pricing engine for a Salesforce customer has to be decided by factors with the following hierarchy:

State in which the customer is located

City in which the customer is located if available

Zip code in which the customer is located if available

Changes to this information should have minimum code changes

What should a data architect recommend to maintain this information for the custom pricing engine that is to be built in Salesforce?

**Options:**

---

**A-** Create a custom object to maintain the pricing criteria.

- B-** Assign the pricing criteria within customer pricing engine.
- C-** Maintain require pricing criteria in custom metadata types.
- D-** Configure the pricing criteria in price books.

**Answer:**

---

C

**Explanation:**

---

According to the [Get Started with Custom Metadata Types](#) unit on Trailhead, one of the use cases for custom metadata types is to define custom charges for an accounting app. The unit states that "Say that your org uses a standard accounting app. You can create a custom metadata type that defines custom charges, like duties and VAT rates. Then you can write some Apex code that calculates the total amount due for each invoice by using the metadata from your custom metadata type." Therefore, a similar approach can be used to maintain the pricing criteria for a custom pricing engine in Salesforce.

## Question 4

---

**Question Type:** MultipleChoice

---

UC has a legacy client server app that as a relational data base that needs to be migrated to salesforce.

What are the 3 key actions that should be done when data modeling in salesforce?

Choose 3 answers:

**Options:**

---

- A- Identify data elements to be persisted in salesforce.
- B- Map legacy data to salesforce objects.
- C- Map legacy data to salesforce custom objects.
- D- Work with legacy application owner to analysis legacy data model.
- E- Implement legacy data model within salesforce using custom fields.

**Answer:**

---

A, B, E

**Explanation:**

---

According to the [Data Modeling unit on Trailhead](#), some of the key actions that should be done when data modeling in Salesforce are identifying data elements, mapping legacy data, and implementing legacy data model. The unit states that "Before you start creating objects and fields in Salesforce, you need to identify the data elements that you want to store and work with. ... Next, you need to map your legacy data to Salesforce objects and fields. ... Finally, you need to implement your data model in Salesforce by creating custom objects and fields using declarative tools or Metadata API." Therefore, these are the correct actions for migrating a legacy client server

app to Salesforce.

## Question 5

---

**Question Type:** MultipleChoice

---

Universal Container (UC) is replacing a home-grown CRM solution with Salesforce, UC has decided to migrate operational (Open and active) records to Salesforce, while keeping historical records in legacy system, UC would like historical records to be available in Salesforce on an as needed basis.

Which solution should a data architect recommend to meet business requirement?

### Options:

---

- A- Leverage real-time integration to pull records into Salesforce.
- B- Bring all data Salesforce, and delete it after a year.
- C- Leverage mashup to display historical records in Salesforce.
- D- Build a chair solution to go the legacy system and display records.



**Answer:**

---

C

**Explanation:**

---

According to the [Using Mashups](#) article on Salesforce Developers, one of the techniques for deploying large data volumes is to use mashups to display historical records in Salesforce. The article states that "Mashups are a way to display data from an external system within a Salesforce page without copying or synchronizing the data. Mashups use a combination of Visualforce, Apex callouts, and JavaScript code that runs in the browser. Mashups are useful when you want to display large amounts of read-only data that is stored outside of Salesforce." Therefore, a data architect should recommend this solution to meet the business requirement of UC.

## Question 6

---

**Question Type: MultipleChoice**

---

NTO has a loyalty program to reward repeat customers. The following conditions exist:

1. Reward levels are earned based on the amount spent during the previous 12 months.
2. The program will track every item a customer has bought and grant them points for discount.
3. The program generates 100 million records each month.

NTO customer support would like to see a summary of a customer's recent transaction and reward level(s) they have attained.

Which solution should the data architect use to provide the information within the salesforce for the customer support agents?

### Options:

---

- A-** Create a custom object in salesforce to capture and store all reward program. Populate nightly from the point-of-scale system, and present on the customer record.
- B-** Capture the reward program data in an external data store and present the 12 months trailing summary in salesforce using salesforce connect and then external object.
- C-** Provide a button so that the agent can quickly open the point of sales system displaying the customer history.
- D-** Create a custom big object to capture the reward program data and display it on the contact record and update nightly from the point-of-scale system.

### Answer:

---

D

### Explanation:

---

According to the [Get Started with Big Objects](#) unit on Trailhead, one of the use cases for custom big objects is to store and manage loyalty program data for customers. The unit states that "From loyalty programs to transactions, order, and billing information, use a custom big object to keep track of every detail." Therefore, a custom big object can be used to capture the reward program data and

display it on the contact record. Additionally, according to the Big Objects Implementation Guide, big objects can handle massive amounts of data (up to billions of records) and can be updated nightly from external systems using Bulk API or batch Apex. Therefore, a custom big object can meet the requirements of NTO's loyalty program scenario.

## Question 7

---

**Question Type: MultipleChoice**

---

Northern Trail Outfitters (NTO) has the following systems:

Customer master-source of truth for customer information

Service cloud-customer support

Marketing cloud-marketing support

Enterprise data warehouse---business reporting

The customer data is duplicated across all these systems and are not kept in sync. Customers are also complaining that they get repeated marketing emails and have to call into update their information.

NTO is planning to implement master data management (MDM) solution across the enterprise.

Which three data will an MDM tool solve?

Choose 3 answers

**Options:**

---

- A- Data completeness
- B- Data loss and recovery
- C- Data duplication
- D- Data accuracy and quality
- E- Data standardization

**Answer:**

---

C, D, E

**Explanation:**

---

According to the [What is Master Data Management \(MDM\)?](#) article, some of the data challenges that an MDM tool can solve are data duplication, data accuracy and quality, and data standardization. The article states that "MDM solutions comprise a broad range of data cleansing, transformation, and integration practices. As data sources are added to the system, MDM initiates processes to identify, collect, transform, and repair data. Once the data meets the quality thresholds, schemas and taxonomies are created to help maintain a high-quality master reference." Therefore, an MDM tool can help NTO eliminate data duplication across different systems, improve data accuracy and quality by removing errors and inconsistencies, and standardize data formats and definitions for better integration and

analysis.

## Question 8

---

**Question Type:** MultipleChoice

---

Northern trail Outfitters (NTO) runs its entire out of an enterprise data warehouse (EDW), NTD's sales team starting to use Salesforce after a recent implementation, but currently lacks data required to advanced and opportunity to the next stage.

NTO's management has research Salesforce Connect and would like to use It to virtualize and report on data from the EDW within Salesforce. NTO will be running thousands of reports per day across 10 to 15 external objects.

What should a data architect consider before implementing Salesforce Connect for reporting?

### Options:

---

- A- Maximum number for records returned
- B- OData callout limits per day
- C- Maximum page size for server-driven paging

D- Maximum external objects per org

**Answer:**

---

B

**Explanation:**

---

According to the [Salesforce Connect Reporting](#) blog post, one of the considerations for using Salesforce Connect for reporting is the OData callout limits per day. The blog post states that "Salesforce Connect has a limit of 100,000 callouts per day. This limit is shared across all external data sources in your org. If you exceed this limit, you will receive an error message and no more callouts will be allowed until the next day." Therefore, a data architect should consider this limit before implementing Salesforce Connect for reporting.

## Question 9

---

**Question Type:** MultipleChoice

---

Universal Containers (UC) has implemented Salesforce, UC is running out of storage and needs to have an archiving solution, UC would like to maintain two years of data in Salesforce and archive older data out of Salesforce.

Which solution should a data architect recommend as an archiving solution?

## Options:

---

- A- Use a third-party backup solution to backup all data off platform.
- B- Build a batch job to move all records off platform, and delete all records from Salesforce.
- C- Build a batch job to move two-year-old records off platform, and delete records from Salesforce.
- D- Build a batch job to move all records off platform, and delete old records from Salesforce.

## Answer:

---

C

## Explanation:

---

The data architect should recommend building a batch job to move two-year-old records off platform, and delete records from Salesforce as an archiving solution. A batch job is a process that runs in the background and performs operations on large volumes of data in Salesforce. By building a batch job that moves two-year-old records off platform to an external storage system, such as Amazon S3 or Google Cloud Storage, and deletes them from Salesforce, the data architect can reduce the storage consumption and improve the performance of Salesforce org. Option A is incorrect because using a third-party backup solution to backup all data off platform will not free up any storage space in Salesforce, unless the data is also deleted from Salesforce after backup. Option B is incorrect because building a batch job to move all records off platform, and delete all records from Salesforce will result in losing all the current data in Salesforce, which may not be desirable or feasible. Option D is incorrect because building a batch job to move all records off platform, and delete old records from Salesforce does not make sense, as restore implies restoring data back to Salesforce, not moving it off platform.

## Question 10

---

### Question Type: MultipleChoice

---

A large Automobile company has implemented SF for its Sales Associates. Leads flow from its website to SF using a batch integration in SF. The Batch job connects the leads to Accounts in SF. Customer visiting their retail stores are also created in SF as Accounts.

The company has noticed a large number of duplicate accounts in SF. On analysis, it was found that certain customers could interact with its website and also visit the store. The Sales associates use Global Search to search for customers in Salesforce before they create the customers.

Which scalable option should a data Architect choose to implement to avoid duplicates?

### Options:

---

- A-** Create duplicate rules in SF to validate duplicates during the account creation process
- B-** Implement a MDM solution to validate the customer information before creating Accounts in SF.
- C-** Build Custom search based on fields on Accounts which can be matched with customer when they visit the store
- D-** Customize Account creation process to search if customer exists before creating an Account.



## Answer:

---

A

## Explanation:

---

The data architect should choose to implement duplicate rules in SF (Salesforce) to validate duplicates during the account creation process. Duplicate rules are a feature in Salesforce that allow users to define criteria and actions for detecting and preventing duplicate records. By creating duplicate rules for accounts, the data architect can ensure that any leads from the website or customers from the retail stores that match existing accounts in Salesforce are flagged or blocked before they are created as new accounts. This will help avoid duplicate accounts in Salesforce and maintain data quality. Option B is incorrect because implementing a MDM (Master Data Management) solution to validate the customer information before creating accounts in SF will require additional infrastructure cost and maintenance effort. Option C is incorrect because building custom search based on fields on accounts which can be matched with customer when they visit the store will require additional development effort and may not be accurate or user-friendly. Option D is incorrect because customizing account creation process to search if customer exists before creating an account will require additional configuration effort and may not be consistent or scalable.

## Question 11

---

**Question Type:** MultipleChoice

---

Northern Trail Outfitters (NTO) wants to implement backup and restore for Salesforce data. Currently, it has data backup processes that runs weekly, which back up all Salesforce data to an enterprise data warehouse (EDW). NTO wants to move to daily backups and provide restore capability to avoid any data loss in case of outage.

What should a data architect recommend for a daily backup and restore solution?

### Options:

---

- A- Use AppExchange package for backup and restore.
- B- Use ETL for backup and restore from EDW.
- C- Use Bulk API to extract data on daily basis to EDW and REST API for restore.
- D- Change weekly backup process to daily backup, and implement a custom restore solution.

### Answer:

---

A

### Explanation:

---

The data architect should recommend using AppExchange package for backup and restore. AppExchange is a marketplace for Salesforce apps and solutions that can be installed and configured in Salesforce orgs. There are several AppExchange packages that provide backup and restore functionality for Salesforce data, such as OwnBackup, Odaseva, or Spanning. These packages can perform daily backups of Salesforce data to a secure cloud storage, and provide restore capability to avoid any data loss in case of outage.

Option B is incorrect because using ETL (Extract, Transform, Load) for backup and restore from EDW (Enterprise Data Warehouse) will require additional development effort and may not be reliable or secure. Option C is incorrect because using Bulk API to extract data on daily basis to EDW and REST API for restore will require additional integration effort and may not be scalable or performant. Option D is incorrect because changing weekly backup process to daily backup, and implementing a custom restore solution will require additional configuration effort and may not be robust or compliant.

**To Get Premium Files for Data-Architect Visit**

<https://www.p2pexams.com/products/data-architect>

**For More Free Questions Visit**

<https://www.p2pexams.com/salesforce/pdf/data-architect>

