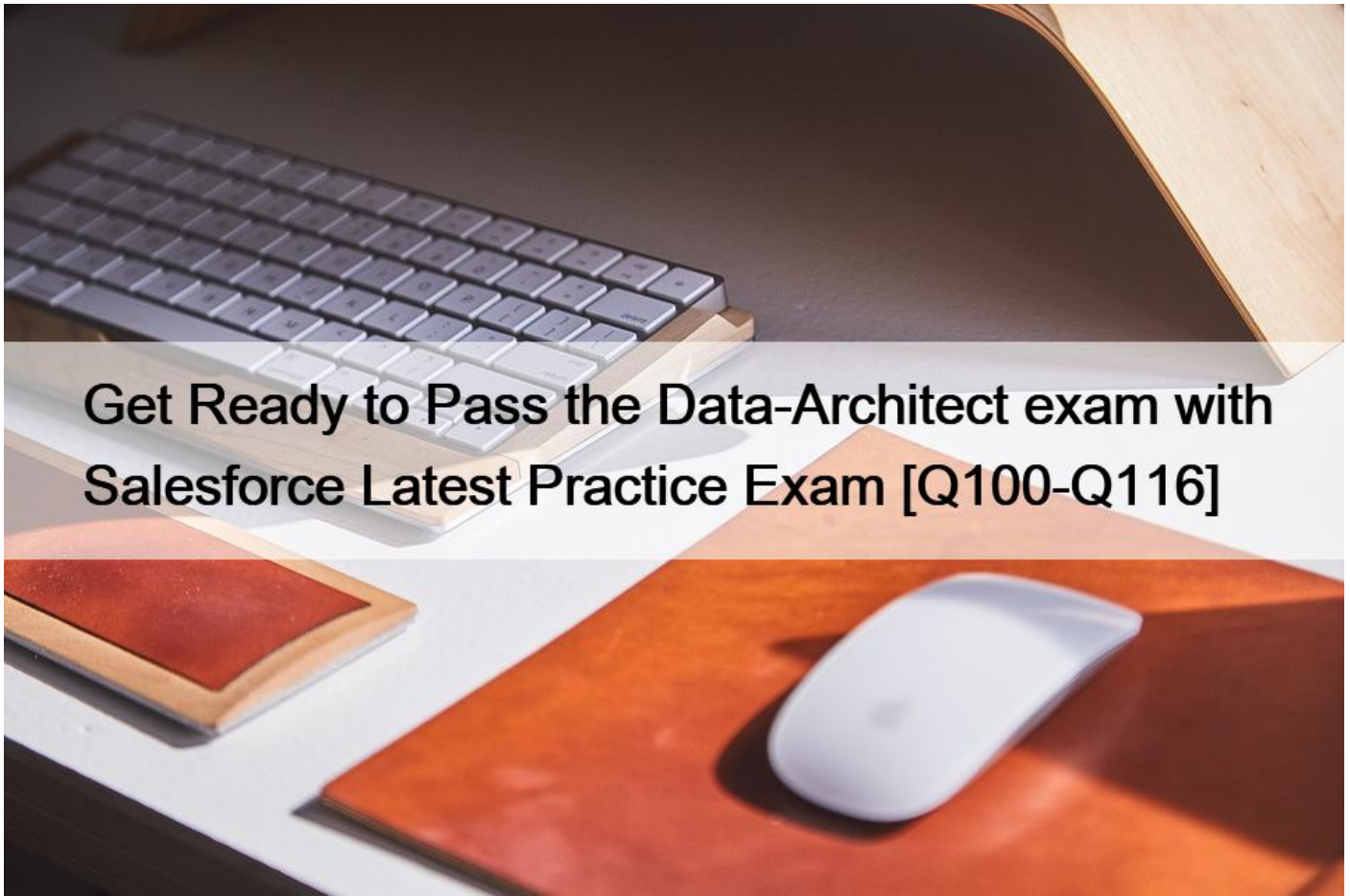


## Get Ready to Pass the Data-Architect exam with Salesforce Latest Practice Exam [Q100-Q116]



### **Get Ready to Pass the Data-Architect exam with Salesforce Latest Practice Exam Get Prepared for Your Data-Architect Exam With Actual Salesforce Study Guide!**

Salesforce Certified Data Architect certification exam is one of the most prestigious certifications in the Salesforce ecosystem. It is intended for professionals who have experience in designing and developing data solutions using Salesforce. Salesforce Certified Data Architect certification exam covers a wide range of topics, including data modeling, data architecture, data integration, data storage, and data governance.

Salesforce Data-Architect Certification Exam is a rigorous test that requires a deep understanding of Salesforce data architecture concepts, best practices, and implementation strategies. Data-Architect exam is designed to test the candidate's knowledge of how to design, implement, and manage complex data architectures within the Salesforce platform. Data-Architect exam covers topics such as data modeling, data integration, data migration, data governance, and data security.

**Q100.** NTO has a loyalty program to reward repeat customers. The following conditions exists:

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?

- \* 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.
- \* Provide a button so that the agent can quickly open the point of sales system displaying the customer history.
- \* 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.
- \* 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.

**Q101.** Universal Containers (UC) has several custom Visualforce applications have been developed in which users are able to edit Opportunity records. UC struggles with data completeness on their Opportunity records and has decided to make certain fields required that have not been in the past. The newly required fields are dependent on the Stage of the Opportunity, such that certain fields are only required once an Opportunity advances to later stages. There are two fields. What is the simplest approach to handle this new requirement?

- \* Update the Opportunity page layout to mark these fields as required.
- \* Use a validation rule for each field that takes the Stage into consideration.
- \* Update these Opportunity field definitions in Setup to be required.
- \* Write an Apex trigger that checks each field when records are saved.

Using a validation rule for each field that takes the Stage into consideration is the simplest approach to handle this new requirement. A validation rule can enforce the field requirements based on the logic and criteria that you define, and display an error message when users try to save a record that does not meet the requirements. Updating the Opportunity page layout to mark these fields as required will not work because page layouts do not support conditional field requirements. Updating these Opportunity field definitions in Setup to be required will not work because it will apply to all stages and records. Writing an Apex trigger that checks each field when records are saved is not the simplest approach because it requires coding and testing

**Q102.** NTO processes orders from its website via an order management system (OMS). The OMS stores over 2 million historical records and is currently not integrated with SF. The Sales team at NTO using Sales cloud and would like visibility into related customer orders yet they do not want to persist millions of records directly in Salesforce. NTO has asked the data architect to evaluate SF connect and the concept of data verification. Which 3 considerations are needed prior to a SF Connect implementation?

Choose 3 answers:

- \* Assess whether the external data source is reachable via an ODATA endpoint.
- \* Identify the external tables to sync into external objects
- \* Configure a middleware tool to poll external table data
- \* Create a 2nd system Admin user for authentication to the external source.
- \* Develop an object relationship strategy.

**Q103.** Universal Containers (UC) is implementing a Salesforce project with large volumes of data and daily transactions. The solution includes both real-time web service integrations and Visualforce mash-ups with back-end systems. The Salesforce Full sandbox used by the project integrates with full-scale back-end testing systems. What two types of performance testing are

appropriate for this project?

Choose 2 answers

- \* Post go -live automated page -load testing against the Salesforce Production org.
- \* Pre -go -live unit testing in the Salesforce Full sandbox.
- \* Pre -go -live automated page -load testing against the Salesforce Full sandbox.
- \* Stress testing against the web services hosted by the integration middleware.

**Q104.** Universal Containers' system administrators have been complaining that they are not able to make changes to its users' record, including moving them to new territories without getting unable to lock row' errors. This is causing the system admins to spend hours updating user records every day.

What should the data architect do to prevent the error?

- \* Reduce number of users updated concurrently.
- \* Enable granular locking.
- \* Analyze Splunk query to spot offending records.
- \* Increase CPU for the Salesforce org.

**Q105.** Universal Containers (UC) has a custom discount request object set as a detail object with a custom product object as the master. There is a requirement to allow the creation of generic discount requests without the custom product object as its master record. What solution should an Architect recommend to UC?

- \* Create a placeholder product record for the generic discount request.
- \* Mandate the selection of a custom product for each discount request.
- \* Remove the master-detail relationship and keep the objects separate.
- \* Change the master-detail relationship to a lookup relationship.

**Q106.** DreamHouse Realty has a Salesforce org that is used to manage Contacts.

What are two things an Architect should consider using to maintain data quality in this situation? (Choose two.)

- \* Use the private sharing model.
- \* Use Salesforce duplicate management.
- \* Use validation rules on new record create and edit.
- \* Use workflow to delete duplicate records.

Using Salesforce duplicate management and using validation rules on new record create and edit are two things that an architect should consider using to maintain data quality for managing Contacts. Salesforce duplicate management allows the architect to create matching rules and duplicate rules to identify, prevent, or allow duplicate records based on various criteria. Validation rules allow the architect to enforce data quality standards and business logic by displaying error messages when users try to save invalid data. The other options are not relevant or helpful for maintaining data quality

**Q107.** Universal Containers (UC) wants to store product data in Salesforce, but the standard Product object does not support the more complex hierarchical structure which is currently being used in the product master system. How can UC modify the standard Product object model to support a hierarchical data structure in order to synchronize product data from the source system to Salesforce?

- \* Create a custom lookup field on the standard Product to reference the child record in the hierarchy.
- \* Create a custom lookup field on the standard Product to reference the parent record in the hierarchy.
- \* Create a custom master-detail field on the standard Product to reference the child record in the hierarchy.
- \* Create an Apex trigger to synchronize the Product Family standard picklist field on the Product object.

Creating a custom lookup field on the standard Product to reference the parent record in the hierarchy is the correct way to modify the standard Product object model to support a hierarchical data structure. This allows UC to create a self-relationship on the Product object and define parent-child relationships among products.

**Q108.** Universal Containers has a large volume of Contact data going into Salesforce.com. There are 100,000 existing contact records. 200,000 new contacts will be loaded. The Contact object has an external ID field that is unique and must be populated for all existing records. What should the architect recommend to reduce data load processing time?

- \* Load Contact records together using the Streaming API via the Upsert operation.
- \* Delete all existing records, and then load all records together via the Insert operation.
- \* Load all records via the Upsert operation to determine new records vs. existing records.
- \* Load new records via the Insert operation and existing records via the Update operation.

Loading new records via the Insert operation and existing records via the Update operation will allow using the external ID field as a unique identifier and avoid any duplication or overwriting of records. This is faster and safer than deleting all existing records or using the Upsert operation, which might cause conflicts or errors

**Q109.** Northern Trail Outfitters (NTO) has recently implemented Salesforce to track opportunities across all their regions. NTO sales teams across all regions have historically managed their sales process in Microsoft Excel. NTO sales teams are complaining that their data from the Excel files were not migrated as part of the implementation and NTO is now facing low Salesforce adoption.

What should a data architect recommend to increase Salesforce adoption?

- \* Use the Excel connector to Salesforce to sync data from individual Excel files.
- \* Define a standard mapping and train sales users to import opportunity data.
- \* Load data in external database and provide access to database to sales users.
- \* Create a chatter group and upload all Excel files to the group.

According to Trailhead2, one of the best practices to increase Salesforce adoption is to migrate existing data from legacy systems or spreadsheets into Salesforce, so that users can access all their data in one place and leverage the features and functionality of Salesforce. Option B is the correct answer because it suggests defining a standard mapping and training sales users to import opportunity data from Excel files into Salesforce, which can help them transition from their old process and increase their confidence and satisfaction with Salesforce. Option A is incorrect because using the Excel connector to Salesforce does not migrate the data into Salesforce, but only syncs it between Excel and Salesforce, which can cause data inconsistency and duplication issues. Option C is incorrect because loading data in an external database and providing access to it does not increase Salesforce adoption, but rather creates another system for users to manage and switch between. Option D is incorrect because creating a chatter group and uploading all Excel files to it does not migrate the data into Salesforce, but only stores it as attachments, which cannot be used for reporting or analysis purposes.

**Q110.** Universal Containers (UC) is in the process of migrating legacy inventory data from an enterprise resources planning (ERP) system into Sales Cloud with the following requirements:

Legacy inventory data will be stored in a custom child object called Inventory\_c.

Inventory data should be related to the standard Account object.

The Inventory object should inherit the same sharing rules as the Account object.

Anytime an Account record is deleted in Salesforce, the related Inventory\_c record(s) should be deleted as well.

What type of relationship field should a data architect recommend in this scenario?

- \* Master-detail relationship field on Account, related to Inventory\_c
- \* Master-detail relationship field on Inventory\_c, related to Account
- \* Indirect lookup relationship field on Account, related to Inventory\_c
- \* Lookup relationship fields on Inventory related to Account

According to the Salesforce documentation, a relationship field is a field that allows linking one object to another object in Salesforce. There are different types of relationship fields that have different characteristics and behaviors, such as master-detail, lookup, indirect lookup, external lookup, etc.

To recommend a type of relationship field for this scenario, where legacy inventory data will be stored in a custom child object called Inventory\_\_c, inventory data should be related to the standard Account object, the Inventory\_\_c object should inherit the same sharing rules as the Account object, and anytime an Account record is deleted in Salesforce, the related Inventory\_\_c record(s) should be deleted as well, a data architect should recommend:

Master-detail relationship field on Inventory\_\_c, related to Account (option B). This means creating a field on the Inventory\_\_c object that references the Account object as its parent. A master-detail relationship field establishes a parent-child relationship between two objects, where the parent object controls certain behaviors of the child object. For example, a master-detail relationship field can:

Inherit the sharing and security settings from the parent object to the child object. This means that the users who can access and edit the parent record can also access and edit the related child records.

Cascade delete from the parent object to the child object. This means that when a parent record is deleted, all the related child records are also deleted.

Roll up summary fields from the child object to the parent object. This means that the parent object can display aggregated information from the child records, such as count, sum, min, max, or average.

Master-detail relationship field on Account, related to Inventory\_\_c (option A) is not a good solution, as it reverses the direction of the relationship. This means creating a field on the Account object that references the Inventory\_\_c object as its parent. This is not possible, as a standard object cannot be on the detail side of a master-detail relationship. Indirect lookup relationship field on Account, related to Inventory\_\_c (option C) is also not a good solution, as it is a special type of relationship field that allows linking a custom object to a standard object on an external system using an indirect reference. This is not applicable for this scenario, as both objects are in Salesforce and do not need an external reference. Lookup relationship field on Inventory\_\_c related to Account (option D) is also not a good solution, as it establishes a looser relationship between two objects than a master-detail relationship. A lookup relationship field does not inherit sharing and security settings from the parent object to the child object, does not cascade delete from the parent object to the child object, and does not roll up summary fields from the child object to the parent object.

**Q111.** NTO uses salesforce to manage relationships and track sales opportunities. It has 10 million customers and 100 million opportunities. The CEO has been complaining 10 minutes to run and sometimes failed to load, throwing a time out error.

Which 3 options should help improve the dashboard performance?

Choose 3 answers:

- \* Reduce the amount of data queried by archiving unused opportunity records.
- \* De-normalize the data by reducing the number of joins.
- \* Use selective queries to reduce the amount of data being returned.
- \* Remove widgets from the dashboard to reduce the number of graphics loaded.
- \* Run the dashboard for CEO and send it via email.

**Q112.** Universal Containers (UC) has lead assignment rules to assign leads to owners. Leads not routed by assignment rules are assigned to a dummy user. Sales rep are complaining of high load times and issues with accessing leads assigned to the dummy user.

What should a data architect recommend to solve these performance issues?

- \* Assign dummy user last role in role hierarchy

- \* Create multiple dummy user and assign leads to them
- \* Assign dummy user to highest role in role hierarchy
- \* Periodically delete leads to reduce number of leads

According to the official Salesforce guide<sup>1</sup>, assigning leads to a single dummy user can cause performance issues and data skew, especially if the dummy user owns more than 10,000 records. Data skew occurs when a single user or a small number of users own a disproportionately large number of records, which can affect query performance and sharing calculations. Option B is the correct answer because it suggests creating multiple dummy users and assigning leads to them, which can distribute the load and reduce data skew. Option A is incorrect because assigning the dummy user to the last role in the role hierarchy does not affect the performance or data skew issues. Option C is incorrect because assigning the dummy user to the highest role in the role hierarchy can worsen the performance and data skew issues, as it will grant access to more users and records. Option D is incorrect because periodically deleting leads can cause data loss and does not address the root cause of the problem.

**Q113.** UC is preparing to implement sales cloud and would like to its users to have read only access to an account record if they have access to its child opportunity record. How would a data architect implement this sharing requirement between objects?

- \* Create a criteria-based sharing rule.
- \* Implicit sharing will automatically handle with standard functionality.
- \* Add appropriate users to the account team.
- \* Create an owner-based sharing rule.

Implicit sharing will automatically handle this sharing requirement with standard functionality, as it grants read-only access to parent accounts when users have access to child opportunities. This is also known as account-opportunity sharing<sup>3</sup>. Creating a criteria-based sharing rule (option A) or an owner-based sharing rule (option D) is not necessary, as they are used to grant additional access based on record criteria or ownership. Adding appropriate users to the account team (option C) is also not required, as it is used to grant access to specific users or groups for individual accounts.

**Q114.** A shipping and logistics company has created a large number of reports within Sales Cloud since Salesforce was introduced. Some of these reports analyze large amounts of data regarding the whereabouts of the company's containers, and they are starting to time out when users are trying to run the reports. What is a recommended approach to avoid these time-out issues?

- \* Improve reporting performance by replacing the existing reports in Sales Cloud with new reports based on Analytics Cloud.
- \* Improve reporting performance by creating a custom Visualforce report that is using a cache of the records in the report.
- \* Improve reporting performance by creating a dashboard that is scheduled to run the reports only once per day.
- \* Improve reporting performance by creating an Apex trigger for the Report object that will pre-fetch data before the report is run.

**Q115.** NTO has decided that it is going to build a channel sales portal with the following requirements:

1. External resellers are able to authenticate to the portal with a login.
2. Lead data, opportunity data and order data are available to authenticated users.
3. Authenticated users many need to run reports and dashboards.
4. There is no need for more than 10 custom objects or additional file storage.

Which community cloud license type should a data architect recommend to meet the portal requirements?

- \* Customer community plus.
- \* Lightning external apps starter.
- \* Customer community.
- \* Partner community.

**Q116.** A casino is implementing salesforce and is planning to build a customer 360 view for a customer who visits its resorts. The casino currently maintained the following systems that records customer activity:

1. Point of sales system: All purchases for a customer.
2. Salesforce: All customer service activity and sales activity for a customer.
3. Mobile app: All bookings, preferences and browser activity for a customer.
4. Marketing: All email, SMS and social campaigns for a customer.

Customer service agents using salesforce would like to view the activities from all system to provide supports to customers. The information has to be current and real time.

What strategy should the data architect implement to satisfy this requirement?

- \* Explore external data sources in salesforce to build 360 view of customer.
- \* Use a customer data mart to view the 360 view of customer.
- \* Migrate customer activities from all 4 systems into salesforce.
- \* Periodically upload summary information in salesforce to build 360 view.

Exploring external data sources in Salesforce to build 360 view of customer is the best strategy to satisfy the requirement, as it allows real-time access to data from other systems without storing it in Salesforce. Using a customer data mart may not provide real-time information or may require additional integration efforts.

Migrating customer activities from all 4 systems into Salesforce may exceed the storage limits or cause data quality issues. Periodically uploading summary information in Salesforce may not provide current or detailed information.

**Pass Your Next Data-Architect Certification Exam Easily & Hassle Free:**

<https://www.actualtests4sure.com/Data-Architect-test-questions.html>