



Free Questions for Data-Architect by vceexamstest

Shared by Glass on 29-01-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

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?

Options:

- A-** Master-detail relationship field on Account, related to Inventory_c
- B-** Master-detail relationship field on Inventory_c, related to Account
- C-** Indirect lookup relationship field on Account, related to Inventory_c
- D-** Lookup relationship fields on Inventory related to Account

Answer:

B

Explanation:

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.

Question 2

Question Type: MultipleChoice

The data architect for UC has written a SOQL query that will return all records from the Task object that do not have a value in the WhatID field:

```
Select id, description, Subject from Task where WhatId!= NULL
```

When the data architect usages the query to select values for a process a time out error occurs.

What does the data architect need to change to make this query more performant?

Options:

- A- Remove description from the requested field set.
- B- Change query to SOSL.
- C- Add limit 100 to the query.
- D- Change the where clause to filter by a deterministic defined value.

Answer:

D

Explanation:

According to the Salesforce documentation, SOQL is a query language that allows querying data from Salesforce objects and fields. SOQL queries have various clauses and operators that can be used to filter and sort the results. However, some clauses and operators can affect the performance of SOQL queries by increasing the cost or complexity of executing them.

To make this query more performant, a data architect should change the where clause to filter by a deterministic defined value (option D). This means using a filter condition that specifies a concrete value or range of values for a field, such as `WhatId = '001xx000003DGg3'` or `WhatId IN ('001xx000003DGg3', '001xx000003DGg4')`. This can improve the performance of the query by reducing the number of records that need to be scanned and returned. A deterministic defined value can also leverage an index on the

field, which can speed up the query execution.

Removing description from the requested field set (option A) is not a good solution, as it can affect the functionality or usability of the query. The description field may contain important or relevant information that is needed for the process. Changing the query to SOSL (option B) is also not a good solution, as SOSL is a different query language that allows searching text fields across multiple objects. SOSL queries have different syntax and limitations than SOQL queries, and may not return the same results or performance. Adding limit 100 to the query (option C) is also not a good solution, as it can affect the completeness or accuracy of the query. The limit clause specifies the maximum number of records that can be returned by the query, which may not include all the records that match the filter condition.

Question 3

Question Type: MultipleChoice

Northern Trail Outfitter has implemented Salesforce for its associates nationwide, Senior management is concerned that the executive dashboard is not reliable for their real-time decision-making. On analysis, the team the following issues with data entered in Salesforce.

Information in certain records is incomplete.

Incorrect entry in certain fields causes records to be excluded in report fitters.

Duplicate entries cause incorrect counts.

Which three steps should a data architect recommend to address the issues?

Options:

- A- Periodically export data to cleanse data and import them back into Salesforce for executive reports.
- B- Build a sales data warehouse with purpose-build data marts for dashboards and senior management reporting.
- C- Explore third-party data providers to enrich and augment information entered in salesforce.
- D- Leverage Salesforce features, such as validate rules, to avoid incomplete and incorrect records.
- E- design and implement data-quality dashboard to monitor and act on records that are incomplete or incorrect

Answer:

B, C, D

Explanation:

According to the Salesforce documentation³, data quality is the measure of how well the data in Salesforce meets the expectations and requirements of the users and stakeholders. Data quality can be affected by various factors, such as data entry errors, data duplication, data inconsistency, data incompleteness, data timeliness, etc. To address the issues with data quality that affect the reliability of executive dashboards, a data architect should recommend:

Building a sales data warehouse with purpose-built data marts for dashboards and senior management reporting (option B). This means creating a separate database or system that stores and organizes sales data from Salesforce and other sources for analytical purposes. A data warehouse can provide a single source of truth for sales data and enable faster and more accurate reporting and analysis. A data

mart is a subset of a data warehouse that focuses on a specific subject or business area, such as sales performance, customer segmentation, product profitability, etc. A data mart can provide tailored and relevant data for different users or groups based on their needs and interests.

Exploring third-party data providers to enrich and augment information entered in Salesforce (option C). This means using external services or tools that can validate, correct, update, and enhance the data that is entered or imported into Salesforce. This can help improve data quality and accuracy, and reduce data duplication and incompleteness.

Leveraging Salesforce features, such as validation rules, to avoid incomplete and incorrect records (option D). This means using features that allow defining rules and criteria to validate the data that is entered or updated by the users or integrations. This can help prevent invalid or incorrect data from being saved, and trigger actions or alerts to correct or improve the data.

Periodically exporting data to cleanse data and import them back into Salesforce for executive reports (option A) is not a good solution, as it can be time-consuming, error-prone, and inefficient. It may also cause data inconsistency and synchronization issues between Salesforce and other systems. Designing and implementing data-quality dashboard to monitor and act on records that are incomplete or incorrect (option E) is also not a good solution, as it can be complex, costly, and difficult to maintain. It may also not address the root causes of data quality issues or prevent them from occurring in the first place.

Question 4

Question Type: MultipleChoice

UC developers have created a new lightning component that uses an Apex controller using a SOQL query to populate a custom list view. Users are complaining that the component often fails to load and returns a time-out error.

What tool should a data architect use to identify why the query is taking too long?

Options:

- A- Use Splunk to query the system logs looking for transaction time and CPU usage.
- B- Enable and use the query plan tool in the developer console.
- C- Use salesforce's query optimizer to analyze the query in the developer console.
- D- Open a ticket with salesforce support to retrieve transaction logs to be analyzed for processing time.

Answer:

B

Explanation:

According to the [Salesforce documentation](#)¹, the query plan tool is a tool that can be enabled and used in the developer console to analyze the performance of a SOQL query. The query plan tool shows the cost, cardinality, sObject type, and relative cost of each query plan that Salesforce considers for a query. The relative cost indicates how expensive a query plan is compared to the Force.com query optimizer threshold. A query plan with a relative cost above 1.0 is likely to cause a time-out error.

To identify why the query is taking too long, a data architect should use the query plan tool in the developer console (option B). This way, the data architect can see which query plan is chosen by Salesforce and how it affects the performance of the query. The data architect can also use the query plan tool to optimize the query by adding indexes, filters, or limits to reduce the cost and improve the efficiency of the query.

Using Splunk to query the system logs looking for transaction time and CPU usage (option A) is not a good solution, as it can be complex, costly, and difficult to integrate with Salesforce. It may also not provide enough information or insights to identify and optimize the query performance. Using Salesforce's query optimizer to analyze the query in the developer console (option C) is also not a good solution, as it is not a separate tool that can be used in the developer console. The query optimizer is a feature that runs automatically when a SOQL query is executed and chooses the best query plan based on various factors². Opening a ticket with Salesforce support to retrieve transaction logs to be analyzed for processing time (option D) is also not a good solution, as it can be time-consuming, dependent, and inefficient. It may also not provide enough information or insights to identify and optimize the query performance.

Question 5

Question Type: MultipleChoice

Universal Containers has 30 million case records. The Case object has 80 fields. Agents are reporting reports in the Salesforce org.

Which solution should a data architect recommend to improve reporting performance?

Options:

- A- Create a custom object to store aggregate data and run reports.
- B- Contact Salesforce support to enable skinny table for cases.
- C- Move data off of the platform and run reporting outside Salesforce, and give access to reports.
- D- Build reports using custom Lightning components.

Answer:

C

Explanation:

According to the [Salesforce documentation](#)¹, reporting performance can be affected by various factors, such as the volume and complexity of data, the design and configuration of reports and dashboards, the number and type of users accessing the reports, etc. To improve reporting performance, some of the recommended solutions are:

Move data off of the platform and run reporting outside Salesforce, and give access to reports (option C). This means using an external service or tool that can extract, transform, and load (ETL) data from Salesforce to another system or database, such as a data warehouse or a business intelligence platform. This can improve reporting performance by reducing the load and latency on Salesforce, and enabling faster and more flexible reporting and analysis on the external system. Users can access the reports from the external system using a link or an embedded component in Salesforce.

Contact Salesforce support to enable skinny table for cases (option B). This means requesting Salesforce to create a custom table that contains a subset of fields from the Case object that are frequently used or queried. A skinny table can improve reporting performance by avoiding joins between standard and custom fields, omitting soft-deleted records, and leveraging indexes on the fields².

Create a custom object to store aggregate data and run reports (option A). This means creating a custom object that contains summary or calculated data from the Case object, such as counts, sums, averages, etc. A custom object can improve reporting performance by reducing the number of records and fields that need to be queried and displayed.

Build reports using custom Lightning components (option D). This means creating custom components that use Lightning Web Components or Aura Components frameworks to display report data in Salesforce. A custom component can improve reporting performance by using client-side caching, pagination, lazy loading, or other techniques to optimize data rendering and interaction.

Question 6

Question Type: MultipleChoice

North Trail Outfitters (NTO) operates a majority of its business from a central Salesforce org, NTO also owns several secondary orgs that the service, finance, and marketing teams work out of, At the moment, there is no integration between central and secondary orgs, leading to data-visibility issues.

Moving forward, NTO has identified that a hub-and-spoke model is the proper architect to manage its data, where the central org is the hub and the secondary orgs are the spokes.

Which tool should a data architect use to orchestrate data between the hub org and spoke orgs?

Options:

- A-** A middleware solution that extracts and distributes data across both the hub and spokes.
- B-** Develop custom APIs to poll the hub org for change data and push into the spoke orgs.
- C-** Develop custom APIs to poll the spoke for change data and push into the org.
- D-** A backup and archive solution that extracts and restores data across orgs.

Answer:

A

Explanation:

According to the Salesforce documentation, a hub-and-spoke model is an integration architecture pattern that allows connecting multiple Salesforce orgs using a central org (hub) and one or more secondary orgs (spokes). The hub org acts as the master data source and orchestrates the data flow between the spoke orgs. The spoke orgs act as the consumers or producers of the data and communicate with the hub org.

To orchestrate data between the hub org and spoke orgs, a data architect should use:

A middleware solution that extracts and distributes data across both the hub and spokes (option A). This means using an external service or tool that can connect to multiple Salesforce orgs using APIs or connectors, and perform data extraction, transformation, and distribution operations between the hub and spoke orgs. This can provide a scalable, flexible, and reliable way to orchestrate data across multiple orgs.

Developing custom APIs to poll the hub org for change data and push into the spoke orgs (option B) is not a good solution, as it can be complex, costly, and difficult to maintain. It may also not be able to handle large volumes of data or complex transformations efficiently. Developing custom APIs to poll the spoke orgs for change data and push into the hub org (option C) is also not a good solution, as it can have the same drawbacks as option B. It may also not be able to handle conflicts or errors effectively. Using a backup and archive solution that extracts and restores data across orgs (option D) is also not a good solution, as it can incur additional costs and dependencies. It may also not be able to handle real-time or near-real-time data orchestration requirements.

Question 7

Question Type: MultipleChoice

A large multinational B2C Salesforce customer is looking to implement their distributor management application in Salesforce. The application has the following capabilities:

1. Distributor create sales order in Salesforce
2. Sales order are based on product prices applicable to their region

3. Sales order are closed once they are fulfilled
4. It is decided to maintain the order in opportunity object

How should the data architect model this requirement?

Options:

- A- Create lookup to Custom Price object and share with distributors.
- B- Configure price Books for each region and share with distributors.
- C- Manually update Opportunities with Prices application to distributors.
- D- Add custom fields in Opportunity and use triggers to update prices.

Answer:

B

Explanation:

According to the Salesforce documentation, an opportunity is a standard object that represents a potential sale or deal with an account or contact. An opportunity can have products and prices associated with it using price books. A price book is a standard object that contains a list of products and their prices for different regions, currencies, segments, etc. A price book can be shared with different users or groups based on their visibility and access settings.

To model the requirement of implementing a distributor management application in Salesforce, where distributors create sales orders based on product prices applicable to their region, and sales orders are closed once they are fulfilled, a data architect should:

Configure price books for each region and share with distributors (option B). This means creating different price books for different regions with the appropriate products and prices, and sharing them with the distributors who belong to those regions. This way, distributors can create sales orders (opportunities) using the price books that are relevant to their region.

Creating a lookup to Custom Price object and sharing with distributors (option A) is not a good solution, as it can introduce unnecessary complexity and redundancy to the data model. It is better to use standard objects and features that are designed for managing products and prices in Salesforce. Manually updating opportunities with prices applicable to distributors (option C) is also not a good solution, as it can be time-consuming, error-prone, and inefficient. It is better to use automation tools or features that can update prices based on predefined criteria or logic. Adding custom fields in opportunity and using triggers to update prices (option D) is also not a good solution, as it can be complex, costly, and difficult to maintain. It is better to use standard fields and features that can handle prices more effectively.

Question 8

Question Type: MultipleChoice

Universal Containers (CU) is in the process of implementing an enterprise data warehouse (EDW). UC needs to extract 100 million records from Salesforce for migration to the EDW.

What data extraction strategy should a data architect use for maximum performance?

Options:

- A- Install a third-party AppExchange tool.
- B- Call the REST API in successive queries.
- C- Utilize PK Chunking with the Bulk API.
- D- Use the Bulk API in parallel mode.

Answer:

C

Explanation:

According to the Salesforce documentation², extracting large amounts of data from Salesforce can be challenging and time-consuming, as it can encounter performance issues, API limits, timeouts, etc. To extract 100 million records from Salesforce for migration to an enterprise data warehouse (EDW), a data extraction strategy that can provide maximum performance is:

Utilize PK Chunking with the Bulk API (option C). This means using a feature that allows splitting a large query into smaller batches based on the record IDs (primary keys) of the queried object. This can improve performance and avoid timeouts by processing each batch asynchronously and in parallel using the Bulk API³.

Installing a third-party AppExchange tool (option A) is not a good solution, as it can incur additional costs and dependencies. It may also not be able to handle such a large volume of data efficiently. Calling the REST API in successive queries (option B) is also not a good

solution, as it can encounter API limits and performance issues when querying such a large volume of data. Using the Bulk API in parallel mode (option D) is also not a good solution, as it can still cause timeouts and errors when querying such a large volume of data without chunking.

Question 9

Question Type: MultipleChoice

Universal Containers (UC) is migrating from a legacy system to Salesforce CRM, UC is concerned about the quality of data being entered by users and through external integrations.

Which two solutions should a data architect recommend to mitigate data quality issues?

Options:

- A-** Leverage picklist and lookup fields where possible
- B-** Leverage Apex to validate the format of data being entered via a mobile device.
- C-** Leverage validation rules and workflows.
- D-** Leverage third-party- AppExchange tools

Answer:

A, C

Explanation:

According to the Salesforce documentation¹, data quality is the measure of how well the data in Salesforce meets the expectations and requirements of the users and stakeholders. Data quality can be affected by various factors, such as data entry errors, data duplication, data inconsistency, data incompleteness, data timeliness, etc. To mitigate data quality issues, some of the recommended solutions are:

Leverage picklist and lookup fields where possible (option A). This means using fields that restrict the values or references that can be entered by the users or integrations. This can help reduce data entry errors, enforce data consistency, and improve data accuracy.

Leverage validation rules and workflows (option C). This means using features that allow defining rules and criteria to validate the data that is entered or updated by the users or integrations. This can help prevent invalid or incorrect data from being saved, and trigger actions or alerts to correct or improve the data.

Leveraging Apex to validate the format of data being entered via a mobile device (option B) is not a good solution, as it can be complex, costly, and difficult to maintain. It is better to use standard features or declarative tools that can handle data validation more effectively. Leveraging third-party AppExchange tools (option D) is also not a good solution, as it can incur additional costs and dependencies. It is better to use native Salesforce features or custom solutions that can handle data quality more efficiently.

Question 10

Question Type: MultipleChoice

Universal Containers (UC) uses the following Salesforce products:

Sales Cloud for customer management.

Marketing Cloud for marketing.

Einstein Analytics for business reporting.

UC occasionally gets a list of prospects from third-party source as comma-separated values (CSV) files for marketing purposes. Historically, UC would load contact Lead object in Salesforce and sync to Marketing Cloud to send marketing communications. The number of records in the Lead object has grown over time and has been consuming large amounts of storage in Sales Cloud, UC is looking for recommendations to reduce the storage and advice on how to optimize the marketing Cloud to send marketing communications. The number of records in the Lead object has grown over time and has been consuming large amounts of storage in Sales Cloud, UC is looking for recommendations to reduce the storage and advice on how to optimize the marketing process.

What should a data architect recommend to UC in order to immediately avoid storage issues in the future?

Options:

- A-** Load the CSV files in Einstein Analytics and sync with Marketing Cloud prior to sending marketing communications;
- B-** Load the CSV files in an external database and sync with Marketing Cloud prior to sending marketing communications.
- C-** Load the contacts directly to Marketing Cloud and have a reconciliation process to track prospects that are converted to customers.

D- Continue to use the existing process to use Lead object to sync with Marketing Cloud and delete Lead records from Sales after the sync is complete.

Answer:

C

Explanation:

According to the Salesforce documentation⁴, Marketing Cloud is a platform that allows creating and managing marketing campaigns across multiple channels, such as email, mobile, social media, web, etc. Marketing Cloud can integrate with Sales Cloud and other Salesforce products to share data and insights. One of the ways to integrate Marketing Cloud with Sales Cloud is using Marketing Cloud Connect⁵, which allows syncing data between the two platforms using synchronized data sources.

However, if UC occasionally gets a list of prospects from third-party sources as CSV files for marketing purposes, it may not be necessary or efficient to load them into Sales Cloud first and then sync them with Marketing Cloud. This can consume large amounts of storage in Sales Cloud, which has a limit based on the license type⁶. It can also cause data quality issues, such as duplicates or outdated information.

A better option for UC is to load the contacts directly to Marketing Cloud using Import Definition, which allows importing data from external files or databases into Marketing Cloud data extensions. Data extensions are custom tables that store marketing data in Marketing Cloud. This way, UC can avoid storage issues in Sales Cloud and optimize the marketing process by sending marketing communications directly from Marketing Cloud.

To track prospects that are converted to customers, UC can have a reconciliation process that compares the contacts in Marketing Cloud with the accounts or contacts in Sales Cloud. This can be done using SQL queries or API calls to access and compare data from both platforms. Alternatively, UC can use Marketing Cloud Connect to sync the converted contacts from Sales Cloud to Marketing Cloud using synchronized data sources.

Loading the CSV files in Einstein Analytics and syncing with Marketing Cloud prior to sending marketing communications (option A) is not a good option, as it can add unnecessary complexity and latency to the process. Einstein Analytics is a platform that allows creating and analyzing data using interactive dashboards and reports. It is not designed for importing and exporting data for marketing purposes.

Loading the CSV files in an external database and syncing with Marketing Cloud prior to sending marketing communications (option B) is also not a good option, as it can incur additional costs and maintenance for the external database. It can also introduce data security and privacy risks, as the data may not be encrypted or protected by Salesforce.

Continuing to use the existing process to use Lead object to sync with Marketing Cloud and delete Lead records from Sales after the sync is complete (option D) is not a good option, as it can cause performance issues and data loss. Deleting Lead records from Sales can affect reporting and auditing, as well as trigger workflows and validations that may not be intended. It can also cause data inconsistency and synchronization errors between Sales Cloud and Marketing Cloud.

Question 11

Question Type: MultipleChoice

A large insurance provider is looking to implement Salesforce. The following exist.

1. Multiple channel for lead acquisition
2. Duplication leads across channels
3. Poor customer experience and higher costs

On analysis, it found that there are duplicate leads that are resulting to mitigate the issues?

Options:

- A-** Build process is manually search and merge duplicates.
- B-** Standard lead information across all channels.
- C-** Build a custom solution to identify and merge duplicate leads.
- D-** Implement third-party solution to clean and event lead data.
- E-** Implement de-duplication strategy to prevent duplicate leads

Answer:

B, D, E

Explanation:

According to the Salesforce documentation², duplicate leads are leads that have the same or similar information as other leads in Salesforce. Duplicate leads can cause poor customer experience, higher costs, and inaccurate reporting. To mitigate the issues caused by duplicate leads, some of the recommended practices are:

Standardize lead information across all channels (option B). This means using consistent formats, values, and fields for capturing lead data from different sources, such as web forms, email campaigns, or third-party vendors. This can help reduce data quality issues and make it easier to identify and prevent duplicate leads.

Implement a third-party solution to clean and enrich lead data (option D). This means using an external service or tool that can validate, correct, update, and enhance lead data before or after importing it into Salesforce. This can help improve data quality and accuracy, and reduce duplicate leads.

Implement a de-duplication strategy to prevent duplicate leads (option E). This means using Salesforce features or custom solutions that can detect and block duplicate leads from being created or imported into Salesforce. For example, using Data.com Duplicate Management³, which allows defining matching rules and duplicate rules for leads and other objects.

Building a process to manually search and merge duplicates (option A) is not a good practice, as it can be time-consuming, error-prone, and inefficient. Building a custom solution to identify and merge duplicate leads (option C) is also not a good practice, as it can be complex, costly, and difficult to maintain. It is better to use existing Salesforce features or third-party solutions that can handle duplicate leads more effectively.

Question 12

Question Type: MultipleChoice

Universal Containers has a rollup summary field on account to calculate the number of contacts associated with an account. During the account load, Salesforce is throwing an "UNABLE _TO_LOCK_ROW" error.

Which solution should a data architect recommend to resolve the error?

Options:

- A- Defer rollup summary field calculation during data migration.
- B- Perform a batch job in serial mode and reduce the batch size.
- C- Perform a batch job in parallel mode and reduce the batch size.
- D- Leverage Data Loader's platform API to load data.

Answer:

B

Explanation:

According to the [Salesforce documentation](#)¹, the "UNABLE _TO_LOCK_ROW" error occurs when a record is being updated or created, and another operation tries to access or update the same record at the same time. This can cause lock contention and timeout issues. To resolve the error, some of the recommended solutions are:

Perform a batch job in serial mode and reduce the batch size (option B). This means running the batch job one at a time and processing fewer records per batch. This can reduce the chances of concurrent updates and lock contention on the same records.

Use the FOR UPDATE keyword to lock records in Apex code or API calls. This means explicitly locking the records that are being accessed or updated by a transaction, and preventing other transactions from modifying them until the lock is released. This can avoid conflicts and errors between concurrent operations on the same records².

Defer rollup summary field calculation during data migration (option A). This means disabling the automatic calculation of rollup summary fields on the parent object when child records are inserted or updated. This can improve performance and avoid locking issues on the parent records. However, this option is only available for custom objects, not standard objects³.

Performing a batch job in parallel mode and reducing the batch size (option C) is not a good solution, as it can still cause lock contention and errors if multiple batches try to access or update the same records at the same time. Leveraging Data Loader's platform API to load data (option D) is also not a good solution, as it can still encounter locking issues if other operations are modifying the same records at the same time.

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>

