ACD300 Q&As

Appian Certified Lead Developer

Pass Appian ACD300 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.geekcert.com/acd300.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Appian Official Exam Center

- **Instant Download** After Purchase
- **100% Money Back** Guarantee
- **365 Days** Free Update
- **800,000+** Satisfied Customers

QUESTION 1

What are two advantages of having High Availability (HA) for Applan Cloud applications?

A. An Applan Cloud HA instance is composed of multiple active nodes running in different availability zones in differentregions.

B. Data andtransactions are continuouslyreplicated across the active nodes to achieve redundancy and avoid single points offailure.

C. A typical Appian Cloud HA instance is composed of two active nodes.

D. In the event of a system failure, your Appian instance will fie restored and available to your users in less than 15 minutes.having lost no more than the last 1minute worth of data.

Correct Answer: BD

The two advantages of having High Availability (HA) for Appian Cloud applications are:

B. Data and transactions are continuously replicated across the active nodes to achieve redundancy and avoid single points of failure. This is an advantage of having HA, as it ensures that there is always a backup copy of data and transactions in case one of the nodes fails or becomes unavailable. This also improves data integrity and consistency across the nodes, as any changes made to one node are automatically propagated to the other node. D. In the event of a system failure, your Appian instance will be restored and available to your users in less than 15 minutes, having lost no more than the last 1 minute worth of data. This is an advantage of having HA, as it guarantees a high level of service availability and reliability for your Appian instance. If one of the nodes fails or becomes unavailable, the other node will take over and continue to serve requests without any noticeable downtime or data loss for your users. The other options are incorrect for the following reasons:

A. An Appian Cloud HA instance is composed of multiple active nodes running in different availability zones in different regions. This is not an advantage of having HA, but rather a description of how HA works in Appian Cloud. An Appian Cloud HA instanceconsists of two active nodes running in different availability zones within the same region, not different regions.

C. A typical Appian Cloud HA instance is composed of two active nodes. This is not an advantage of having HA, but rather a description of how HA works in Appian Cloud. A typical Appian Cloud HA instance consists of two active nodes running in different availability zones within the same region, but this does not necessarily provide any benefit over having one active node. Verified References: Appian Documentation, section "High Availability".

QUESTION 2

Your application contains a process model that Is scheduled to run daily at a certain time, which kicks off a user input task to a specified user on the 1ST time zone for morning data collection The time zone is set to the (default) pm!timezone.

In this situation, what does the pm!tinezone reflect?

A. The time zone of the server where Applan is intuited

B. The line zone of the user who most recently published the process model

C. The default time zone for the environment as specified in the Administration Console

D. The time zone of the user who is completing the input task.

Correct Answer: C

In this situation, pm!timezone reflects the default time zone for the environment as specified in the Administration Console. pm!timezone is a process variable that returns the time zone of the process. If the time zone is not explicitly set in the process model, then pm!timezone returns the default time zone for the environment, which can be configured in the Administration Console. In this case, the time zone is set to the (default) pm!timezone, which means that the process model does not have a specific time zone, and therefore uses the default time zone for the environment. The other options are not correct. Option A, the time zone of the server where Appian is installed, is not what pm!timezone reflects, as the server time zone may not be the same as the default time zone for the environment. Option B, the time zone of the user who most recently published the process model, is not what pm!timezone reflects, as the user\\'s time zone may not be the same as the default time zone for the environment. Option D, the time zone of the user who is completing the input task, is not what pm!timezone reflects, as the user\\'s time zone may not be the same as the default time zone for the environment.

QUESTION 3

You are asked to design a case management system for a client in addition to storing some basic metadata about a case, one of the client s requirements Is the ability for users to update a case The client would like any user in their organization of 500 people to be able to make these updates The users are all based in the company\\'s headquarters, and there will be frequent cases where users are attempting to edit the same case. The client wants to ensure no information Is lost when these edits occur and does not want the solution to burden their process administrators with any additional effort.

Which data locking approach should you recommend?

A. Allow edits without locking the case GDI

B. Use the database lo implement tow lewd pessimistic locking.

C. Add an ?ersion annotation to the case COT to manage Vie locking

D. Design a process report and query to determine who opened the edit form first

Correct Answer: C

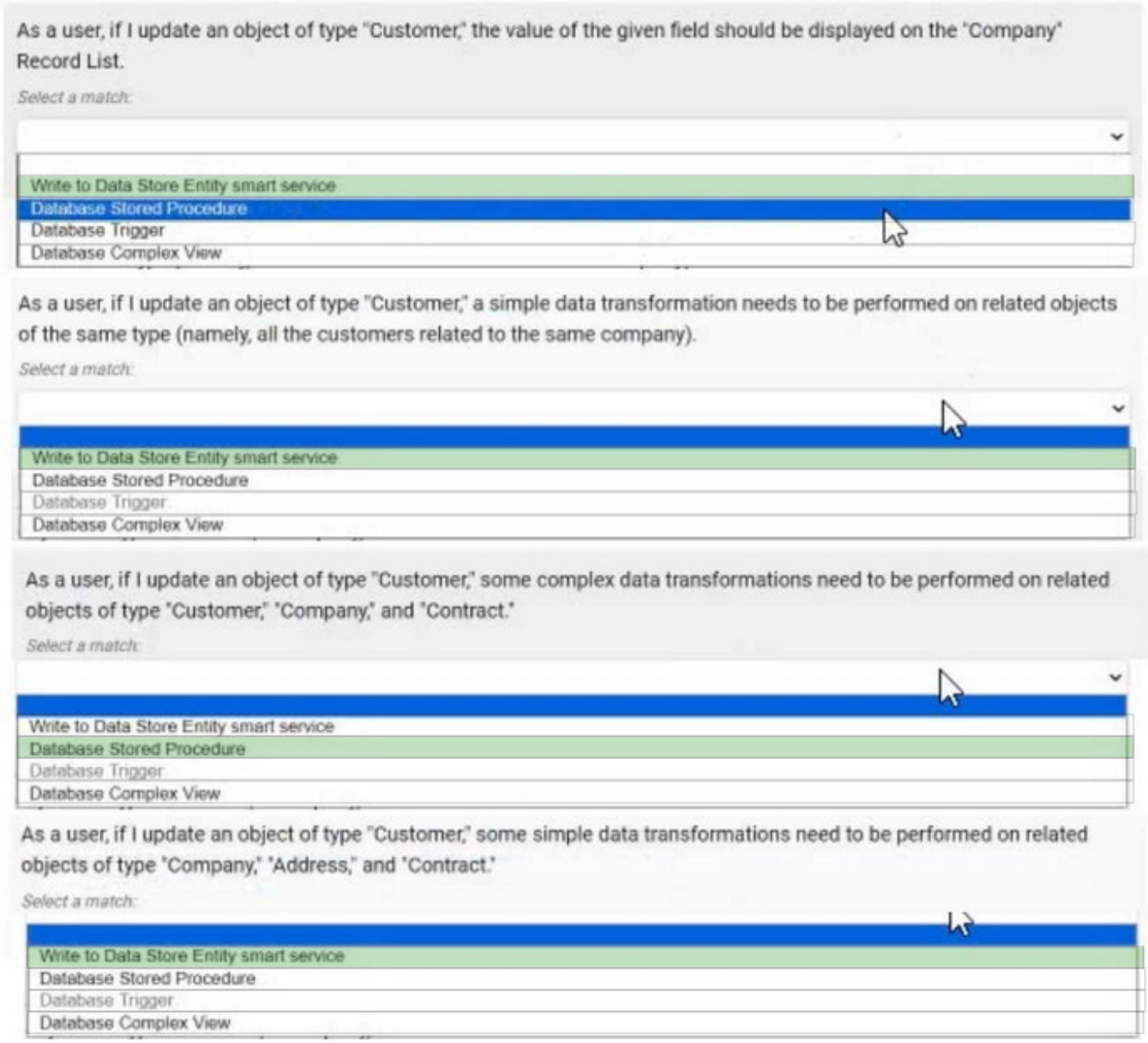
The @Version annotation is a feature of Appian that allows for optimistic locking of CDTs. Optimistic locking assumes that concurrent updates to the same data are rare and does not lock the data until it is saved. If two users try to save changes to the same data, the user who saves first will succeed, while the user who saves second will get an error message and will have to resolve the conflict manually. This approach is suitable for the client\\'s requirement, as itallows any user to update a case without locking it, ensures no information is lost when concurrent edits occur, and does not require any additional effort from the process administrators. Verified References: [Appian Documentation], section "Optimistic Locking".

QUESTION 4

HOTSPOT For each scenario outlined, match the best tool to use to meet expectations. Each tool will be used once Note: To change your responses, you may deselected your response by clicking the blank space at the top of the selection list.

Hot Area:

Correct Answer:



QUESTION 5

You add an index on the searched field of a MySQL table with many rows (>100k).

The field would benefit greatly from the Index in which three scenarios?

A. The field contains a textual shot Business code.

B. The field contains long unstructured text such as a hash

C. The field contains many datetimes, covering a large range

D. The Add contains Dig integers, above and below 0.

E. The field contains a structured JSON.

Correct Answer: ACD

The field would benefit greatly from the index in the following scenarios:

A. The field contains a textual short Business code. This is a scenario where an index can improve the performance of queries that search for exact matches or ranges of values in the field. A textual short Business code is likely to have high

cardinality, meaning that it has many distinct values and low duplication. This makes the index more selective and efficient, as it can quickly narrow down the results based on the search criteria.

C. The field contains many datetimes, covering a large range. This is a scenario where an index can improve the performance of queries that search for exact matches or ranges of values in the field. A datetime field is likely to have high

cardinality, meaning that it has many distinct values and low duplication. This makes the index more selective and efficient, as it can quickly narrow down the results based on the search criteria.

D. The field contains big integers, above and below 0. This is a scenario where an index can improve the performance of queries that search for exact matches or ranges of values in the field. A big integer field is likely to have high cardinality,

meaning that it has many distinct values and low duplication. This makes the index more selective and efficient, as it can quickly narrow down the results based on the search criteria.

The other options are incorrect for the following reasons:

B. The field contains long unstructured text such as a hash. This is a scenario where an index might not improve the performance of queries that search for exact matches or ranges of values in the field. A long unstructured text field is likely to

have low cardinality, meaning that it has few distinct values and high duplication. This makes the index less selective and efficient, as it cannot quickly narrow down the results based on the search criteria. Moreover, indexing a long

unstructured text field could increase thestorage space and maintenance cost for the database, which could affect the overall performance.

E. The field contains a structured JSON. This is a scenario where an index might not improve the performance of queries that search for exact matches or ranges of values in the field. A structured JSON field is not a native data type in

MySQL, and it requires special functions or operators to access or manipulate its elements. Indexing a structured JSON field could increase the complexity and overhead for the database, which could affect the overall performance. Verified

References: Appian Documentation, section "Query Optimization".