ACD200^{Q&As}

Appian Certified Senior Developer

Pass Appian ACD200 Exam with 100% Guarantee

QUESTION 1

You are creating the group structure of a new application.

Which three best practices apply? (Choose three.)

A. Avoid creating custom group types unless there is a strong need/requirement.

B. Only create the groups necessary for task assignment or security.

C. Flat group structures should be avoided.

D. Group names should not include the application prefix.

E. Keep group names unique.

Correct Answer: ABE

QUESTION 2

You have been asked to produce a Tempo report, with the following requirements:

The report should display in Appian.

The data is held in a business database, but performance in Appian is a key requirement. Which three actions can you take to minimize the performance impact of your report? (Choose three.)

A. Display the report as a task report on the Tasks tab.

B. Limit the number of series values and categories in the report.

C. Use local variables in your report interface to avoid redundant database queries.

D. Use the query process analytics function to retrieve the data.

E. Perform aggregation in the database using a view prior to reading the data into Appian.

Correct Answer: BCE

The three actions that can be taken to minimize the performance impact of the Tempo report are: Limit the number of series values and categories in the report. This is because having too many series values and categories can make the report difficult to read and render, as well as increase the amount of data that needs to be transferred from the database to Appian. A good practice is to limit the number of series values to 10 or less, and the number of categories to 20 or less. Use local variables in your report interface to avoid redundant database queries. This is because local variables can store the results of database queries and reuse them in multiple places within the interface, reducing the number of times the database is accessed. A good practice is to use local variables for common filters, aggregations, or calculations that are used in multiple charts or grids. Perform aggregation in the database using a view prior to reading the data into Appian. This is because performing aggregation in the database can reduce the amount of data that needs to be transferred from the database to Appian, as well as leverage the database\\'s optimization capabilities. A good practice is to create a view in the database that performs the aggregation and then query that view using Appian.
References: Report Performance Best Practices

QUESTION 3

An organization has decided to integrate with a third-party to scan incoming documents and capture the details in a table called [appian].[document]. Each document will form a new case in Appian to be displayed on a Record List.

The record needs to show data from both [appian].[document] and [appian].[caseData], which holds additional case information.

What is the most efficient way to achieve this?

A. Create a trigger on the [appian].[document] table to copy all the data across to the [appian].[caseData] table and point the record at [appian].[caseData].

B. Create a SSIS package to run at a regular interval.

C. Create a view between both the [appian].[document] and [appian].[caseData] tables to feed the record.

D. Create a stored procedure to query the data from both the [appian].[document] and [appian].[caseData] tables.

Correct Answer: C

The most efficient way to achieve the integration between the third-party document scanner and the Appian record list is to create a view between both the [appian].[document] and [appian].[caseData] tables to feed the record. A view is a virtual table that is defined by a query and does not store any data. It can join, filter, aggregate, or transform data from one or more tables and present it in a desired format. A view can be used to simplify complex queries, provide security, or enhance performance. By creating a view, the record can show data from both tables without duplicating or copying any data. Therefore, the correct answer is C. References: Relational Database Guidance Views

QUESTION 4

You need to show joined data from 5 tables. Each table contains a large number of rows and could generate a large result set after executing the Joins.

The business is not expecting live data, and a 2-hour refresh is acceptable. Performance is a top priority.

What should you use? (Choose the best answer.)

A. Table

B. View

C. Stored procedure

D. Materialized view

Correct Answer: D

A materialized view is the best option to show joined data from 5 tables that contain a large number of rows and could generate a large result set after executing the joins. A materialized view is a physical table that holds the results of the SQL that a view would normally be constructed from and can be generated periodically. A materialized view can improve performance by reducing the execution time of complex queries that involve multiple joins, aggregations, or calculations. A materialized view can also reduce the load on the database server by storing the query results in advance. A materialized view can be refreshed at regular intervals or on demand to reflect the changes in the underlying tables. References: [Materialized Views], [View Performance]

QUESTION 5

Your organization is in the process of redesigning its user interfaces in order to use space efficiently.

Which layout component is most appropriate for displaying the interface where the meaningful order of components is vertical?

A. a!columnsLayout()

B. a!columnOverlay()

C. a!cardLayout()

D. a!sideBySideLayout()

Correct Answer: A

The requirement is to use space efficiently in a user interface where the meaningful order of components is vertical. A columns layout is the most appropriate layout component for this requirement, as it allows you to arrange components into columns that stack vertically on smaller devices. A columns layout also provides options to control the width, alignment, and visibility of each column. The other options are not suitable for this requirement, as they either do not preserve the vertical order of components or do not use space efficiently. References: Columns Layout