



Appian ACD201 Practice Questions

Shared by Randolph on 08-10-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page



Question 1

Question Type: MultipleChoice

The synced record Task has a self-referential relationship defined in the column parentTaskId. There is a many-to- one record relationship between the id and parentTaskId called parentTask.

For a given task ID, you need to return the task name and the parent task name.

What should you do?

Options:

- A- Use a!queryRecordType() With a filter on the task id, with fields specified to return recordType!Task.name and recordType!Task.parentTask.name.
- B- Create a sync-time custom record field on the Task record called parentName. Specify this field to return in the query field selection.
- C- Use a!queryRecordType() filtered on the task id once to return the task name and parent task ID. Query the record again to return the parent task name.

Answer:

A

Explanation:

A single a!queryRecordType() call filtered by the task's id and selecting both recordType!Task.name and recordType!Task.parentTask.name leverages the existing self-referential relationship, returning the task name and its parent's name in one efficient query without additional custom fields or extra queries.

Question 2

Question Type: MultipleChoice

An insurance company's product details are currently distributed across numerous database tables, encompassing over 60 to 70 fields.

You need to implement a product comparison tool for business users to compare up to three insurance products simultaneously, with all the product details in a tabular format.

Refer to the following sample format:

Attribute	PRODUCT 1	PRODUCT 2	PRODUCT 3
Name			
Type			
Premium			
Average			
Claim time			

Which object should be used to transform the data into the required format?

Options:

- A- Stored procedure
- B- Record type with relationships
- C- VCCViews

Answer:

C

Explanation:

Database views are ideal for transforming complex, distributed product data into a single, flattened structure tailored for reporting or comparison. In this case, a view can pivot the 60--70 fields and organize them by attribute (e.g., Name, Type, Premium) across multiple product columns, enabling a tabular comparison layout like the one shown. Views also optimize performance and reduce transformation overhead in the interface layer.

Question 3

Question Type: MultipleChoice

You're designing a file upload process that integrates with an external document management system.

Part of this process is to develop a start form with a file upload component, so that the user can upload multiple files.

The business has requested that the maximum file size that can be uploaded through this process is 1GB. Additionally, you're aware that the integration's API can accept files with a maximum file size of 3GB, but recommends no larger than 1GB.

Can the process be more performant without straining the application or integration?

Options:

A- Yes. Request the business allows for these files to be uploaded as an overnight process by storing large files in a staging table, so that adequate resources are available.

B- No. The file upload component is capable of accepting 1GB files and the integration can handle this.

C- Yes. Request the business reconsiders its requirements and find manual workarounds as it is not recommended to upload files of this size in both Appian and the integration.

Answer:

A

Explanation:

While Appian and the integration technically support 1 GB file uploads, handling such large files during interactive sessions can negatively impact application performance, user experience, and resource utilization.

A better approach is to request that the business allow these files to be uploaded asynchronously (e.g., overnight batch process), by temporarily storing them (e.g., in a staging table or folder) and processing them when system load is lower. This makes the process more performant and reliable without straining Appian or the external system.

Question 4

Question Type: MultipleChoice

You need to update an existing Appian application.

The application needs to be responsive and compatible with a variety of screen widths. For example: phone/tablet browser, Appian mobile app, desktop monitor.

Which two functions/parameters should you consider when updating the interfaces? (Choose two.)

Options:

A- stackWhen in Columns Layout and Side By Side Layout

B- a!isNativeMobile()

C- Width of a page in the site object

D- a!isPageWidth()

Answer:

A, B

Explanation:

The stackWhen parameter in layout components helps interfaces adapt to different screen sizes by controlling when columns or side-by-side layouts stack vertically.

The a!isNativeMobile() function detects if the interface is being viewed on a native mobile app, allowing you to tailor the interface for mobile responsiveness.

Question 5

Question Type: MultipleChoice

The synced record type Customer has a one-to-many relationship with the Case record type.

You need to calculate the number of open cases for each customer.

What are two valid methods to accomplish this? (Choose two.)

Options:

A- Within the Case record type, edit the relationship to the Customer record and select the 'Open' status as the grouping filter.

B- Within the Customer record type, create a custom field that evaluates in real-time to aggregate related record fields and count the open cases.

C- Create a query expression for the Customer record type. Utilize the a!measure() function within the aggregation fields to count the related cases in the "Open" status.

D- Create a sync-time custom record field within the Customer record. Utilize the a!relatedRecordData() function to filter open cases and count the number of entries.

Answer:

B, D

Explanation:

A real-time custom field in the Customer record type can aggregate and count related open cases.

A sync-time custom record field using `a!relatedRecordData()` can filter and count open cases, making the count available as a synced field.

Question 6

Question Type: MultipleChoice

You need to relate two entities, Employee and Skill, in the data structure. You want to adhere to Appian best practices.

Employees can have multiple skills, and a single skill can relate to multiple employees.

What type of relationship do these entities represent, and What is the minimum number of tables required to implement the design?

Options:

- A- Many-to-many; 3 tables
- B- Many-to-many; 2 tables
- C- One-to-many; 2 tables

Answer:

A

Explanation:

The relationship between Employee and Skill is many-to-many - an employee can have multiple skills, and each skill can belong to multiple employees.

To model this properly and according to Appian best practices, you need 3 tables:

Question 7

Question Type: MultipleChoice

You need to configure Appian user authentication against an external directory server. The requirement is to use the same corporate logon information without single sign-on (SSO).

Which authentication type should you use?

Options:

- A- PIEE user authentication
- B- OpenID Connect authentication
- C- LDAP authentication

Answer:

C

Explanation:

LDAP authentication allows users to log in with their corporate credentials by authenticating against an external directory server, without requiring single sign-on (SSO).

Question 8

Question Type: MultipleChoice

What are three ways to optimize the memory usage of a process model? (Select three.)

Options:

- A- Pass data directly to activity nodes and smart services.
- B- Minimize the use of subprocesses in the model.
- C- Decrease the number of process variables.
- D- Configure process variables as parameters.
- E- Divide large nodes into multiple smaller nodes.
- F- Avoid storing large blocks of text in process variables.

Answer:

A, D, F

Explanation:

Passing data directly to activity nodes and smart services reduces unnecessary use of process variables, optimizing memory.

Configuring process variables as parameters limits their lifespan and persistence, saving memory.

Avoiding large blocks of text in process variables minimizes memory consumption within the process model.

Question 9

Question Type: MultipleChoice

You run a report on different employee transactions by using a View.

You encounter the following error: "a!queryEntity: An error occurred while retrieving the data."

What is the most likely root cause of this error?

Options:

- A- The view doesn't have a column mapped as a primary key in its corresponding CDT.
- B- The view contains a large number of rows, requiring more time to fetch the data.
- C- The rule contains a missing syntax.

Answer:

A

Explanation:

The most likely root cause is that the view doesn't have a column mapped as a primary key in its corresponding CDT. Appian requires a primary key mapping for querying views to ensure data integrity and identification.

Question 10

Question Type: MultipleChoice

What are three locations to call an Integration that changes the source data? (Choose three.)

Options:

- A- Rule
- B- Web API (GET)
- C- Process model (Call Integration Smart Service]
- D- Web API (POST, PUT, DELETE)
- E- Interface component saveInto parameter

Answer:

A, C, D

Explanation:

Integrations can be called from rules, allowing for source data changes when used in the proper context.

The Call Integration Smart Service in a process model can execute integrations that modify source data.

Web APIs using POST, PUT, or DELETE methods are typically used to change source data via integrations.

Question 11

Question Type: MultipleChoice

You're designing an integration object with JSON (application/json) as the content type.

Which two data types can be automatically converted to JSON in Appian? (Choose two.)

Options:

- A- Appian Custom data type

- B- Base64 documents inline with JSON
- C- Primitive data types
- D- HTML file type

Answer:

A, C

Explanation:

Appian Custom data types (CDTs) can be automatically converted to JSON for integration.

Primitive data types such as text, number, and boolean are also automatically converted to JSON.

Question 12

Question Type: MultipleChoice

You're creating a synced record type which will store read-only data from an external web service. Data in the external system is updated nightly, and the latest information must be available to users in the morning. Currently there are approximately 150,000 entries in the external system, and this number is not expected to change significantly.

There is an existing integration object and associated expression rule, which when called returns the complete dataset from the external system.

What are two configurations that you must apply to ensure that the latest and complete record set is available? (Select two.)

Options:

- A- Within the record type, schedule a full sync to occur nightly after the data is refreshed in the external system.
- B- Modify the integration object and associated expression rule to return batches of 1000 based on a batch number passed in from the record type.
- C- Within the Source Filters settings on the record type, disable the row limit option so that all entries can be retrieved from the external system.
- D- Set all record fields to evaluate in real-time so that the latest information is always retrieved from the web service.

Answer:

A, C

Explanation:

Scheduling a nightly full sync ensures the record type is updated with the latest data after the external system refreshes.

Disabling the row limit in Source Filters allows the complete dataset to be retrieved and synced, ensuring all entries are available.



To Get Premium Files for ACD201 Visit

<https://www.p2pexams.com/products/acd201>

For More Free Questions Visit

<https://www.p2pexams.com/appian/pdf/acd201>

20%
DISCOUNT

P2P
exams