



Free Questions for PDII by [braindumpscollection](#)

Shared by [Boone](#) on [18-01-2024](#)

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

A company notices that their unit tests in a test class with many methods to create many records for prerequisite reference data are slow.

What can a developer do to address the issue?

A Turn off triggers, flows, and validations when running tests.

Options:

- B) Move the prerequisite reference data setup to a TestDataFactory and call that from each test method,
- C) Move the prerequisite reference data setup to a @testSetup method in the test class.
- D) Move the prerequisite reference data setup to the constructor for the test class.

Answer:

C

Explanation:

To address the issue of slow unit tests in a test class with many methods that create many records for prerequisite reference data, a developer should move the prerequisite reference data setup to a @testSetup method in the test class. A @testSetup method is a special method that runs once before all test methods in a test class and creates common test data for all test methods. This can improve the performance of the test class by reducing the number of DML operations and avoiding code repetition. The test methods can access the test data created by the @testSetup method by using SOQL queries or the Test.loadData() method. Reference: [@testSetup Annotation], [Create Test Data for Apex Tests], [Test.loadData Method]

Question 2

Question Type: MultipleChoice

When the component is rendered, the following message is displayed: 'Error retrieving data'.

Which modification should be implemented to the Apex class to overcome the issue?

Options:

- A) Use the Cacheable=true attribute in the Apex method,
- B) Ensure the OWD for the Opportunity object is Public.

- C) Edit the code to use the w. cut sharing keyword in the Apex class.
- D) Use the Continuation=true attribute in the Apex method.

Answer:

A

Explanation:

The Cacheable=true attribute is used to indicate that the Apex method can be cached and reused on the client side, such as in a Lightning web component. This attribute enables the method to be called imperatively or via @wire in the component, and improves the performance and user experience of the component. Without this attribute, the Apex method can only be called imperatively, and will not work with @wire, which is used to get data from the server and update the component automatically. This can cause an error when the component is rendered, such as "Error retrieving data".Reference: [Call Apex Methods], [Imperative Apex]

Question 3

Question Type: MultipleChoice

When the sales team views an individual customer record, they need to see recent interactions for the customer. These interactions can be sales orders, phone calls, or Cases. The date range for recent interactions will be different for every customer record type.

How can this be accomplished?

A Use batch Apex to query for the most recent interactions when the customer view screen is loaded.

Options:

- B)** Use Lightning Flow to read the customer's record type, and then do a dynamic query for recent interactions and display on the View page.
- C)** Use a Lightning component to query and display interactions based on record type that is passed in using a design:attribute from the Lightning page.
- D)** Use a dynamic form on the customer record page to display recent interactions.

Answer:

C

Explanation:

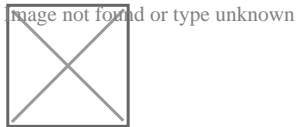
The best way to accomplish this requirement is to use a Lightning component to query and display interactions based on record type that is passed in using a design:attribute from the Lightning page. A Lightning component is a reusable unit of an app that can include markup, JavaScript, CSS, and Apex. A Lightning component can use the Lightning Data Service or Apex to query data from Salesforce and display it on a Lightning page. A design:attribute is a way to define attributes for a Lightning component that can be set in Lightning App Builder or Experience Builder. By using a Lightning component, the developer can create a custom UI element that can query and

display the recent interactions for the customer based on the record type that is passed in as a design:attribute from the Lightning page. This way, the developer can leverage the Lightning framework and the Lightning Data Service or Apex to create a dynamic and interactive component that meets the requirement. Reference: [Lightning Components Basics], [Lightning Data Service Basics], [Design Attributes in Aura Components], [Design Attributes in Lightning Web Components]

Question 4

Question Type: MultipleChoice

A company has code to update a Request and Request Lines and make a callout to their external ERP system's REST endpoint with the updated records.



The callousUtil.makeRestCallout fails with a 'You have uncommitted work pending. Please commit or rollback before calling cut' error.

What should be done to address the problem?

A Change the callousUtil makeRestCallout to an @InvocsblsMethod method.

Options:

- B) Remove the Database. setSavepoint and Database. rollback.
- C) Change the CallousUtil .makeRestCallout to an @future method.

Answer:

C

Explanation:

Changing the CalloutUtil.makeRestCallout to an @future method is a way to address the problem of the "You have uncommitted work pending. Please commit or rollback before calling out" error. This error occurs when a DML operation is performed before a callout to an external service, such as a REST endpoint. This is because the DML operation locks the database records and prevents them from being rolled back if the callout fails. To avoid this error, the developer can use the @future annotation to mark the CalloutUtil.makeRestCallout method as asynchronous, which means it will run in a separate thread after the DML operation is committed. This way, the DML operation and the callout will not interfere with each other, and the error will be resolved. Reference: [Asynchronous Apex], [Making Callouts]

Question 5

Question Type: MultipleChoice

However, when the test runs, no data is returned and the assertion fails.

Which edit should the developer make to ensure the test class runs successfully?

Options:

- A) Implement the without sharing keyword in the searchFeatures Apex class.
- B) Implement the seeAllData=true attribute in the @IsTest annotation.
- C) Enclose the method call within Test.startTest() and @Test.StopTest().
- D) Implement the setFixedSearchResult= method in the test class.

Answer:

D

Explanation:

Implementing the setFixedSearchResult= method in the test class is a way to ensure that the test class runs successfully. The setFixedSearchResult= method is a method that can be used to specify the records to be returned by the SOSL query in the test class. The setFixedSearchResult= method can accept a list of record IDs as a parameter, and return those records when the SOSL query is executed. By using the setFixedSearchResult= method, the developer can control the data returned by the SOSL query, and avoid the dependency on the data in the org. The developer can also use the Test.loadData method to load the test data from a static resource,

and pass the record IDs to the `setFixedSearchResult=` method. For example, the developer can use the following code to implement the `setFixedSearchResult=` method in the test class:

```
// Load test data from a static resource List<sObject> testRecords = Test.loadData(Account.sObjectType, 'testData');  
  
// Get the record IDs List<Id> recordIds = new List<Id>(); for (sObject record : testRecords) { recordIds.add(record.Id); }  
  
// Set the fixed search result Test.setFixedSearchResults(recordIds);  
  
// Call the method that performs the SOSL query List<Account> results = searchFeatures.search('Acme');  
  
// Assert the expected results System.assertEquals(testRecords.size(), results.size());
```

Question 6

Question Type: MultipleChoice

Which statement is true regarding savepoints?

A You can roll back to any savepoint variable created in any order.

Options:

- B)** Static variables are not reverted during a rollback.
- C)** Reference to savepoints can cross trigger invocations.
- D)** Savepoints are not limited by DML statement governor limits.

Answer:

B

Explanation:

The true statement regarding savepoints is that static variables are not reverted during a rollback. Savepoints are used to mark a point in the transaction that can be rolled back to if an error occurs. However, static variables are not affected by the rollback, as they retain their values throughout the transaction. The other statements are false, as savepoints must be rolled back in the reverse order of their creation, savepoints cannot cross trigger invocations, and savepoints are counted as DML statements and are subject to the governor limits. Reference: [Savepoints and Rollbacks], [Apex Developer Guide]

Question 7

Question Type: MultipleChoice

Refer to exhibit.

Image not found or type unknown



What should be added to the setup, in the location indicated, for the unit test above to create the controller extension for the test?

A)

Image not found or type unknown



B)

Image not found or type unknown



C)

Image not found or type unknown



Options:

A) Option A

B) Option B

C) Option C

Answer:

A

Explanation:

The code snippet that should be added to the setup for the unit test to create the controller extension for the test is option A. This code snippet creates a PageReference object that points to the Visualforce page that uses the controller extension. Then, it sets the current page reference to the PageReference object and passes the Account ID as a URL parameter. Finally, it creates an instance of the controller extension by passing the standard controller as an argument. The code snippet in option B will not work because it does not set the current page reference or pass the Account ID as a URL parameter. The code snippet in option C will not work because it does not create a PageReference object or set the current page reference. Reference: [Testing Custom Controllers and Controller Extensions], [PageReference Class], [ApexPages Class]

Question 8

Question Type: MultipleChoice

Refer to exhibit.

Image not found or type unknown



Consider the above trigger intended to assign the Account to the manager of the Account's region,

Which two changes should a developer make in this trigger to adhere to best practices?

Options:

- A) Use a Map to cache the results of the Region._c query by Id.
- B) Move the Region__c query to outside the loop.
- C) Use a Map accountMap instead of List accountList.
- D) Remove the last line updating accountList as It Is not needed.

Answer:

B, D

Explanation:

The two changes that the developer should make in this trigger to adhere to best practices are to move the Region__c query to outside the loop and to remove the last line updating accountList as it is not needed. Moving the Region__c query to outside the loop avoids the SOQL query limit, as it reduces the number of queries from one per Account record to one for all Account records. The developer can use a Map to store the Region__c records by Name, and then use the Map to assign the OwnerId for each Account record. Removing the last line updating accountList as it is not needed avoids the DML statement limit, as it is redundant to update the records that are already in the trigger context. The developer can use the Trigger.new list to modify the Account records, and the changes will be saved automatically after the trigger execution. Using a Map to cache the results of the Region__c query by Id is not a valid answer, as it does not move the query outside the loop, and it does not match the Name field of the Account record. Using a Map accountMap instead of List accountList is not a valid answer, as it does not affect the query or the DML limits, and it does not change the logic of the trigger. Reference: [Apex Triggers], [Apex Best Practices], [Apex Developer Guide]

To Get Premium Files for PDII Visit

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

For More Free Questions Visit

<https://www.p2pexams.com/salesforce/pdf/pdii>

