



**Free Questions for PDII by ebraindumps**

**Shared by Bray on 12-12-2023**

**For More Free Questions and Preparation Resources**

**Check the Links on Last Page**

# Question 1

---

**Question Type:** MultipleChoice

---

Refer to the test method below:

```
Line 1: @isTest
Line 2: static void testMyTrigger()
Line 3: {
Line 4:     //Do a bunch of data setup
Line 5:     DataFactory.setupDataForMyTriggerTest();
Line 6:
Line 7:     List<Account> acctsBefore = [SELECT Is_Customer__c FROM Account WHERE Id IN :DataFactory.accounts];
Line 8:
Line 9:     //Utility to assert all accounts are not customers before the update
Line 10:    AssertUtil.assertNotCustomers(acctsBefore);
Line 11:
Line 12:    //Set accounts to be customers
Line 13:    for(Account a : DataFactory.accounts)
Line 14:    {
Line 15:        a.Is_Customer__c = true;
Line 16:    }
Line 17:
Line 18:    update DataFactory.accounts;
Line 19:
Line 20:    List<Account> acctsAfter = [SELECT Number_Of_Transfers__c FROM Account WHERE Id IN :DataFactory.accounts];
Line 21:
Line 22:    //Utility to assert Number_Of_Transfers__c is correct based on test data
Line 23:    AssertUtil.assertNumberOfTransfers(acctsAfter);
Line 24: }
```

The test method tests an Apex trigger that the developer knows will make a lot of queries when a lot of Accounts are simultaneously updated to be customers.

The test method fails at the Line 20 because of too many SOQL queries.

What is the correct way to fix this?

A.

Use `Limits.getLimitQueries()` to find the total number of queries that can be issued.

B.

Change the `DataFactory` class to create fewer Accounts so that the number of queries in the trigger is reduced.

C.

Add `Test.startTest()` before and add `Test.stopTest()` after both Line 7 and Line 20 of the code.

D.

Add `Test.startTest()` before and add `Test.stopTest()` after Line 18 of the code.

**Options:**

---

A- Option A

B- Option B

C- Option C

D- Option D

E- Option E

**Answer:**

---

D

**Explanation:**

---

The correct way to fix this is to add `Test.startTest()` before and add `Test.stopTest()` after Line 18 of the code. This will reset the governor limits for the code that executes between these two methods, and allow the test to run without hitting the SOQL query limit. The `Test.startTest()` and `Test.stopTest()` methods are used to test code that makes a lot of queries or calls asynchronous methods<sup>12</sup>. By using these methods, the developer can isolate the code that needs more resources and verify that it does not exceed the governor limits.

[Using Limits, startTest, and stopTest | Apex Developer Guide | Salesforce Developers](#)

[Test.StartTest & Test.StopTest - Salesforce Developer Community](#)

[Why would a developer use test startTest and test stopTest in Salesforce? | ForceTalks](#)

## Question 2

---

### Question Type: MultipleChoice

---

The Salesforce admin at Cloud Kicks created a custom object called Region\_c to store all postal zip codes in the United States and the Cloud Kicks sales region the zip code belongs to.

**Object Name:**

Region\_\_c

**Fields:**

zip\_code\_\_c (Text)

Region\_Name\_\_c (Text)

Cloud Kicks wants a trigger on the Lead to populate the Region based on the Lead's zip code.

Which code segment is the most efficient way to fulfill this request?

A)

```
Set<String> zips = new Set<String>();
for(Lead l : Trigger.new) {
    if(l.PostalCode != Null) {
        zips.add(l.PostalCode);
    }
}

for(Lead l : Trigger.new) {
    List<Region__c> regions = [SELECT Zip_Code__c, Region_Name__c FROM Region__c WHERE Zip_Code__c IN
    for(Region__c r :regions) {
        if(l.PostalCode == r.Zip_Code__c) {
            l.Region__c = r.Region_Name__c;
        }
    }
}
}
```

B)

```
Set<String> zips = new Set<String>();  
for(Lead l : Trigger.new) {  
    if(l.PostalCode != Null) {  
        zips.add(l.PostalCode);  
    }  
}
```

```
List<Region__c> regions = [SELECT Zip_Code__c, Region_Name__c FROM Region__c WHERE Zip_Code__c IN :zips
```

```
for(Lead l : Trigger.new) {  
    for(Region__c r :regions) {  
        if(l.PostalCode == r.Zip_Code__c) {  
            l.Region__c = r.Region_Name__c;  
        }  
    }  
}
```

C)

```
for(Lead l : Trigger.new) {  
    Region__c reg = [SELECT Region_Name__c FROM Region__c WHERE Zip_Code__c = :l.PostalCode];  
    l.Region__c = reg.Region_Name__c;  
}
```



D)

```
Set<String> zips = new Set<String>();
for(Lead l : Trigger.new) {
    if(l.PostalCode != Null) {
        zips.add(l.PostalCode);
    }
}

List<Region__c> regions = [SELECT Zip_Code__c, Region_Name__c FROM Region__c WHERE Zip_Code__c IN :zips];

Map<String, String> zipMap = new Map<String, String>();
for(Region__c r :regions) {
    zipMap.put(r.Zip_Code__c, r.Region_Name__c);
}

for(Lead l : Trigger.new) {
    if(l.PostalCode != Null) {
        l.Region__c = zipMap.get(l.PostalCode);
    }
}
```

**Options:**

---

A- Option A

B- Option B

C- Option C

D- Option D

**Answer:**

---

D

## Question 3

---

**Question Type:** MultipleChoice

---

A developer created 2 class that implements the Queueable Interface, as follows:

```
public class without sharing OrderQueueableJob implements Queueable {
    public void execute(QueueableContext context) {
        // implementation logic
        System.enqueueJob(new FollowUpJob());
    }
}
```

As part of the deployment process, the developer is asked to create a corresponding test class.

Which two actions should the developer take to successfully execute the test class?

Choose 2 answers

### Options:

---

- A-** Implement `seeAllData=True` to ensure the Queueable job is able to run in bulk mode.
- B-** Ensure the running user of the test class has, at I the View All permission on the Order object.
- C-** Enclose `System.enqueueJob(new OrderQueueableJob ())` within `Tess.` and `Test .stopTest (1.`
- D-** Implement `Test. isRunningTest()` to prevent chaining jobs during test execution.

### Answer:

---

B, C

## Question 4

---

### Question Type: MultipleChoice

---

The Account object has a field, Audit\_Code\_c, that is used to specify what type of auditing the Account needs and a Lookup to User, zudzisar\_c, that is the assigned auditor. When an Account is initially created, the user specifies the Audit\_Code c. Each User in the org has a unique text field, Audit\_Code\_e, that is used to automatically assign the correct user to the Account's Auditor\_c field.

```
trigger AccountTrigger on Account(before insert) {
    AuditAssigner.setAuditor(Trigger.new);
}

public class AuditAssigner {
    public static void setAuditor(List<Account> accounts) {
        for (User u : [SELECT Id, Audit_Code__c FROM User]) {
            for (Account a : accounts) {
                if (u.Audit_Code__c == a.Audit_Code__c) {
                    a.Auditor__c = u.Id;
                }
            }
        }
    }
}
```

What should be changed to most optimize the code's efficiency?

Choose 2 answers

**Options:**

---

- A-** Add an initial SOQL query to get all distinct audit codes.
- B-** Build a Map<string, List<Account>> of audit code to accounts.
- C-** Build a Map<Id, List<string>>of Account Id to audit codes.
- D-** Add a WHERE clause to the SOQL query to filter on audit codes.

**Answer:**

---

C, D

## Question 5

---

**Question Type: MultipleChoice**

---

Universal Containers (UC) has an ERP system that stores customer Information,

When an Account is created in Salesforce, the ERP system's REST endpoint for creating new customers must automatically be called with the Account information. If the call to the ERP

system fails, the Account should still be created. Accounts in the UC org are only created, one at a time, by users in the UC customer onboarding department.

What should a developer implement to make the call to the ERP

system's REST endpoint?

**Options:**

---

- A- REST call from JavaScript
- B- Headless Quick Action
- C- Call a Queueable from a Trigger
- D- Apex Continuation

**Answer:**

---

A

## Question 6

---

**Question Type: MultipleChoice**

---

A developer wrote a test class that successfully asserts a trigger on Account, It fires and updates data correctly In a sandbox environment.

A Salesforce admin with a custom profile attempts to deploy this trigger via a change set into the production environment, but the test class falls with an Insufficient privileges error.

What should a developer do to fix the problem?

**Options:**

---

- A-** Configure the production environment to enable 'Run All Tests as Admin User.'
- B-** Add System.runAs() to the test class to execute the trigger as a user with the correct object permissions.
- C-** Add seeallData=true to the test class to work within the sharing model for the production environment.
- D-** Verify that Test.startTest() Is not Inside a For loop in the test class,

**Answer:**

---

B

## Question 7

---

**Question Type:** MultipleChoice

---

Refer to the Aura component below:



### Component markup:

```
<aura:component>
  <aura:attribute name="contactInfo" type="Object"/>
  <aura:attribute name="showContactInfo" type="boolean" default="true"/>
  <aura:handler name="init" value="{!this}" action="{!c.init}"/>

  <!-- ... other code .... -->
  <aura:if isTrue="{!v.showContactInfo}">
    <c:contactInfo value="{!v.contactInfo}"/>
  </aura:if>
</aura:component>
```

### Controller JS:

```
{
  init: function(cmp, helper) {
    // ...other code ...
    var show = helper.getShowContactInfo();
    cmp.set("v.showContactInfo", show);
  },
  // other code...
}
```

A developer receives complaints that the component loads slowly.

Which change can the developer implement to make the component perform faster?

### Options:

---

- A- Change the type of contactInfo to 'Map'.
- B- Move the contents of <c: contactInfo into the component.
- C- Add a change event handler for showContactInfo.
- D- Change the default for showContactInfo to "False."

### Answer:

---

D

## Question 8

---

**Question Type: MultipleChoice**

---

Refer to the Lightning component below:

```
<lightning:button label="Save" onclick="{!c.handleSave}" />
```

```
{{  
  handleSave : function (component, event, helper) {  
    helper.saveAndRedirect (component);  
  }  
}}
```

The Lightning Component allows users to click a button to save their changes and then redirects them to a different page.

Currently when the user hits the Save button, the records are getting saved, but they are not redirected.

Which three techniques can a developer use to debug the JavaScript?

Choose 3 answers

### Options:

---

- A- Use console.log () messages in the JavaScript.
- B- Use the browser's dev tools to debug the JavaScript.
- C- Enable Debug Mode for Lightning components for the user.
- D- Use Developer Console to view checkpoints.
- E- Use Developer Console to view the debug log.

### Answer:

---

A, B, C

## Question 9

---

### Question Type: MultipleChoice

---

A developer created and tested a Visualforce page in their developer sandbox, but now receives reports that user encounter view state errors when using it in production.

What should the developer ensure to correct these errors?

**Options:**

---

- A- Ensure queries do not exceed governor limits,
- B- Ensure properties are marked as private,
- C- Ensure variables are marked as transient.
- D- Ensure profiles have access to the Visualforce page.

**Answer:**

---

C

## Question 10

---

**Question Type: MultipleChoice**

---

Which three actions must be completed in a Lightning web component for a JavaScript file in a static resource to be loaded? Choose 3 answers

**Options:**

---

- A- Import the static resource.

- B-** Reference the static resource in a <script> tag.
- C-** Call loadScript.
- D-** Import a method from the platformResourceLoader.
- E-** Append the static resource to the DOM.

**Answer:**

---

A, C, D

## Question 11

---

**Question Type:** MultipleChoice

---

A company's support process dictates that any time a case is closed with a status of "Could not fix," an Engineering Review custom object record should be created and populated with information from the case, the contact, and any of the products associated with the case.

What is the correct way to automate this using an Apex trigger?

**Options:**

---

- A- An after update trigger on Case that creates the Engineering Review record and inserts it
- B- An after upset trigger on Case that creates the Engineering Review record and inserts It .
- C- A before update trigger on Case that creates the Engineering Review record and Inserts It
- D- A before upset trigger on Case that creates the Engineering Review record and Inserts it

**Answer:**

---

A

**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>

