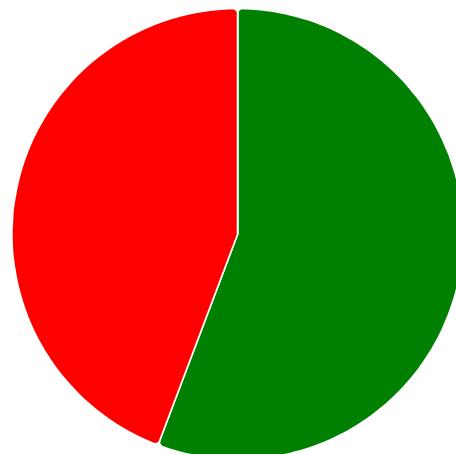


[◀ Return To Review](#)**Attempt: 61 FAIL!****56%** correct (34/61)● Right ● Wrong ● Skipped

Knowledge Area

Q.1. Refer to the code segment below.

```
Line 1 trigger ContactTrigger on Contact (before insert, before update)
Line 2 {
Line 3     Map<Id,Account> accountMap = new Map<Id,Account>();
Line 4     for(Contact c : Trigger.new)
Line 5     {
Line 6         Account a = [SELECT Id, Name, BillingCountry FROM Account WHERE Id := c.AccountId];
Line 7         accountMap.put(a.Id,a);
Line 8     }
Line 9
Line 10    //Do stuff with accountMap
Line 11    for(Contact c : Trigger.new)
Line 12    {
Line 13        Account a = accountMap.get(c.AccountId);
Line 14        if(a != null)
Line 15        {
Line 16            c.BillingCountry = a.BillingCountry;
Line 17        }
Line 18    }
Line 19
Line 20    update Trigger.new;
Line 21 }
```

When following best practices for writing Apex triggers, which two lines are wrong or cause for concern?

Choose 2 answers.

Answered RightAsk Instructor

1. Line 6 (Correct Answer) (Selected Answer)
2. Line 11
3. Line 16
4. Line 20 (Correct Answer) (Selected Answer)

Create Notes

This question is related to the concept of Apex Trigger in Salesforce. Let's get basics to Apex Triggers.

Apex can be invoked by using triggers. Apex triggers enable you to perform custom actions before or after changes to Salesforce records, such as insertions, updates, or deletions.

A trigger is Apex code that executes before or after the following types of operations:

- Insert
- update
- delete
- merge
- upsert
- undelete

There are two types of triggers:

• **Before triggers** are used to update or validate record values before they're saved to the database. The system saves the records that fired the before trigger after the trigger finishes execution. You can modify the records in the trigger without explicitly calling a DML insert or update operation. If you perform DML statements on those records, you get an error.

• **After triggers** are used to access field values that are set by the system (such as a record's Id or LastModifiedDate field), and to affect changes in other records, such as logging into an audit table or firing asynchronous events with a queue. The records that fire the after trigger are read-only.

Now, coming back to the question, the ask is – as per best practices for writing Apex triggers, which two lines are wrong or cause for concern.

Two correct answers are:

“**Line 6**” is correct because as per Apex Trigger best practice, we should Avoid SOQL Queries or DML statements inside FOR Loops.

“**Line 20**” is correct because the code given in question is “before trigger” so we should not explicitly use DML statement.

References:

<https://trailhead.salesforce.com/en/content/learn/modules/success-cloud-coding-conventions/improve-your-apex-code-sc>

https://trailhead.salesforce.com/content/learn/modules/apex_triggers/apex_triggers_intro

Q.2. What is a benefit of JavaScript remoting over Visualforce Remote Objects?

Answered Wrong

Ask Instructor

1. Does not require any JavaScript code.
(Selected Answer)
2. Supports complex server-side application logic.
(Correct Answer)
3. Does not require any Apex code.
4. Allows for specified re-render targets.

This question is related to the concept of “JavaScript Remoting” & “Remote Objects” in Salesforce. Let's look at Salesforce Documentation to check the differences between “JavaScript Remoting” & “Remote Objects”.

Comparing JavaScript Remoting and Remote Objects

JavaScript Remoting and Remote Objects offer similar features, and both are useful tools for creating dynamic, responsive pages. They have some important differences that you should consider before choosing which to use.

In general, Remote Objects is well-suited to pages that need to perform only simple Create-Read-Update-Delete, or “CRUD”, object access. JavaScript Remoting is better suited to pages that access higher-level server actions. Remote Objects lets you get up and running quickly without a lot of ceremony, while JavaScript Remoting is suited for more complex applications that require some up front API-style design work.

Visualforce Remote Objects:

- Makes basic “CRUD” object access easy
- Doesn't require any Apex code
- Supports minimal server-side application logic
- Doesn't provide automatic relationship traversals; you must look up related objects yourself

JavaScript Remoting:

- Requires both JavaScript and Apex code
- **Supports complex server-side application logic** ←
- Handles complex object relationships better
- Uses network connections (even) more efficiently

With this above screenshot from Salesforce Documentation,

the correct answer is “**Supports complex server-side application logic**”.

References:

<https://developer.salesforce.com/docs/atlas.en->

us.pages.meta/pages/pages_js_remoting_compare_remote_objects.htm
https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages/pages_js_remoting_example.htm

Q.3. Universal Containers has an existing automation where a custom record called Account Plan is created upon an Account being marked as a Customer. Recently, a Workflow Rule was added so that whenever an Account is marked as a Customer, a 'Customer Since' date field is updated with today's date.

Now, since the addition of the Workflow Rule, two Account Plan records are created whenever the Account is marked as a Customer.
 What might cause this to happen?

Answered Wrong

Ask Instructor

1. The Apex Trigger responsible for the record creation does not use a static variable to ensure it only fires once.
(Correct Answer)
2. The Process Builder responsible for the record creation fires before and after the Workflow rule.
3. The Workflow Rule responsible for the record creation fires twice because the 'Customer Since' field Update is marked as 'Re-evaluate Workflow Rules After Field Change'.
(Selected Answer)
4. The Apex Trigger responsible for the record creation is not safe and calls insert inside of a for loop.

This question is a scenario-based question and is related to the concept of Trigger Order of Execution in Salesforce.

Let's understand the scenario given in the question:

- Apex trigger on Account that creates an Account Plan record when in Account is marked as a customer.
 - That means, this Apex Trigger is written on after update event. When the Account Type is updated to "Customer", the apex trigger logic fires and creates a record of Account Plan object.
- Recently a workflow rule was added so that whenever an Account is marked as a customer, a 'Customer Since' date field is updated with today's date.
 - That means a workflow rule is created which fires when Account Type field value is updated to "Customer" and the field update action updates the date field "Customer Since" on Account with today's date.

The issue is given in question:

- After addition of workflow rule, two Account Plan records are created whenever the Account is marked as a "Customer".

Let's discuss about debugging:

- As we know, the logic to create record of Account Plan is written in the Apex Trigger which executes and creates single record of Account Plan. If two Account Plan records are created, that means this trigger logic is getting fired two time. **BUT WHY?**

- To understand this, let's understand about Trigger Order of execution in Salesforce.

Order of Execution in Salesforce

A set of rules that describe the path a record takes through all automations and the events that happen from **SAVE to COMMIT**. Salesforce performs the following events in below order.

1. Loads Initial record.
2. If the request came from a standard UI edit page, Salesforce runs **system validation** to check the record for page layout specific rules, field definition, Maximum field length.
3. Executes flows that make before-save update. (New Change in Winter 20)
4. Executes all before triggers.
5. Runs most Custom validation.
6. Executes duplicate rules.
7. Saves the record to the database, but doesn't commit yet.
8. Executes all after triggers.
9. Assignment rules.
10. Executes auto-response rules.
11. Executes workflow rules.
12. If there are workflow field updates, updates the record again.
13. Due to Workflow field updates introduced new duplicate field values, executes duplicate rules again. If the record was updated with workflow field updates, fires before update triggers and after update triggers one more time (and only one more time), in addition to standard validations. Custom validation rules are not run again.
14. Executes processes and flow.
15. Escalation rules.
16. Executes entitlement rules.
17. Executes record-triggered flows that are configured to run after the record is saved.
18. If the record contains a roll-up summary field or is part of a cross-object workflow, performs calculations and updates the roll-up summary field in the parent record. Parent record goes through save procedure.

Create Notes

If you see **Point-13**, it says that if the field was updated with workflow field update, the before and after update triggers are fired one more time (and only one more time). With this point, it's clear that the workflow rule added in this scenario is calling the after-update trigger again and therefore, two Account Plan records are created.

Handle Recursion in Trigger:

To avoid the recursion on a trigger, make sure your trigger is getting executed only one time. You can use Static Boolean variable in apex class and check the variable in Apex Trigger IF it is true then execute your logic and make it false so that trigger cannot execute Again.

That means, the Apex Trigger logic in this scenario does not use a static variable to ensure it only fires once.

With this explanation, the correct answer is:

"The Apex Trigger responsible for the record creation does not use a static variable to ensure it only fires once."

Let's see below screenshot for more understanding. The code given in below screenshot uses static Boolean variable to avoid Recursion.

Screenshot – Apex Class having Boolean Variable

```
TriggerController.apxc
Code Coverage: None API Version: 53
1 public class TriggerController {
2
3     // Boolean variable to stop recursion
4     public static Boolean isFirstCall = true;
5 }
```

Screenshot – Apex Trigger Code using Boolean variable so that Trigger should fire once

```
AccountTrigger.apxt
Code Coverage: None API Version: 53
1 trigger AccountTrigger on Account (after update) {
2
3     if(Trigger.isAfter){
4         if(Trigger.isUpdate){
5
6             if(TriggerController.isFirstCall){
7                 TriggerController.isFirstCall = false;
8
9                 Map<Id,Account> oldMap = new Map<Id,Account>();
10                List<Account_Plan__c> listInsert = new List<Account_Plan__c>();
11                oldMap = Trigger.oldMap;
12                for(Account acct : Trigger.New){
13
14                    if(acct.Type != oldMap.get(acct.Id).Type && acct.Type == 'Customer'){
15                        Account_Plan__c ap = new Account_Plan__c();
16                        ap.put('Name', 'Test');
17                        ap.put('Account__c', acct.Id);
18                        listInsert.add(ap);
19
20
21                }
22
23                System.debug('>>>' +listInsert);
24                insert listInsert;
25            }
26        }
27    }
28}
```

Create Notes

References:

- https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_triggers_order_of_execution.htm
- <https://help.salesforce.com/s/articleView?id=000332407&type=1>

Q.4. Universal Containers wants to use an external Web Service provided by a third-party vendor to validate that shipping and building addresses are correct. The current vendor uses basic password authentication, but Universal Containers might switch to a different vendor who uses OAuth. Which approach follows best practices and allows Universal Containers to switch vendors without updating the code to handle authentication?

Answered Right

Ask Instructor

1. Dynamic Endpoint.
2. Named Credential.
(Correct Answer)(Selected Answer)
3. Custom Setting (List).
4. Custom Metadata.

This is a scenario-based question and is related to the authentication mechanism used in Salesforce. The scenario given in this question is:

- Universal Containers wants to use an external Web Service provided by a third-party vendor to validate that shipping and building addresses are correct.
- The current vendor uses basic **password authentication**.
- Universal Containers might switch to a different vendor who uses **OAuth**.

That means, Universal Containers may need to switch to different vendor who uses OAuth so, which feature we can use in Salesforce which supports different authentication like password authentication, OAuth 2.0, JWT etc.

As per Salesforce documentation, we can use **Named Credential**.

- A named credential authentication protocol **supports basic password authentication, OAuth 2.0, JWT, JWT** Token Exchange, and AWS Signature Version 4.

- You can set up each named credential to use an org-wide named principal or per-user authentication. A named principal applies the same credential or authentication configuration for the entire org, while per-user authentication provides access control at the individual user level.

Screenshot—Named Credential Setup

Named Credential: My Named Credential

Specify the callout endpoint's URL and the authentication settings that are required for Salesforce to make callouts to the remote system.

« Back to Named Credentials

Authentication

Identity Type: Named Principal
Authentication Protocol: Password Authentication
Username: myname

If you use OAuth instead of password authentication, the Apex code remains the same. The authentication settings differ in the named credential, which references an authentication provider that's defined in the org.

Authentication

Identity Type: Named Principal
Authentication Protocol: OAuth 2.0
Authentication Provider: GoogleAuth
Scope: Pending
Authentication Status: Pending

With this explanation,

the correct answer is "**Named Credential**"

Reference:

https://help.salesforce.com/s/articleView?id=sf.named_credentials_about.htm&type=5

Q.5. Which statement is true regarding savepoints?

Answered Wrong

Ask Instructor

- Reference to savepoints can cross trigger invocations.
- Savepoints are not limited by DML statement governor limits.
- Static variables are not reverted during a rollback.
(Correct Answer)
- You can roll back to any savepoint variable created in any order.
(Selected Answer)

This question is related to the concept of Transaction Control in Salesforce.

Let's learn about savepoints and considerations.

Sometimes during the processing of records, your business rules require that partial work (already executed DML statements) be "rolled back" so that the processing can continue in another direction. Apex gives you the ability to generate a savepoint, that is, a point in the request that specifies the state of the database at that time. Any DML statement that occurs after the savepoint can be discarded, and the database can be restored to the same condition it was in at the time you generated the savepoint.

Screenshot—SavePoint Example

In this example, if any error occurs while inserting the Contact "cont" then the entire transaction will be rolled back to savepoint sp1 (as specified in the catch section by Database.rollback method).

```

AccountContactController.apxc
Code Coverage: None API Version: 54
1 public class AccountContactController {
2
3     public static void insertDummyAccountandContact(){
4         Savepoint sp1 = Database.setSavepoint(); ← SavePoint sp1
5
6         Account acct = new Account();
7         acct.Name='AB Texttile';
8         acct.AccountNumber = '0120';
9         insert acct;
10
11        Contact cont = new Contact(AccountId = acct.Id);
12        try
13        {
14            insert cont; ← If any error occurs while inserting contact "cont" then rollback to SavePoint sp1
15        }
16        catch(Exception e)
17        {
18            Database.Rollback(sp1); ←
19        }
20    }
21 }

```

Screenshot – Considerations using savepoint

The following limitations apply to generating savepoint variables and rolling back the database:

- If you set more than one savepoint, then roll back to a savepoint that is not the last savepoint you generated, the later savepoint variables become invalid. For example, if you generated savepoint SP1 first, savepoint SP2 after that, and then you rolled back to SP1, the variable SP2 would no longer be valid. You will receive a runtime error if you try to use it.
- References to savepoints cannot cross trigger invocations because each trigger invocation is a new trigger context. If you declare a savepoint as a static variable then try to use it across trigger contexts, you will receive a run-time error.
- Each savepoint you set counts against the governor limit for DML statements.
- Static variables are not reverted during a rollback. If you try to run the trigger again, the static variables retain the values from the first run.
- Each rollback counts against the governor limit for DML statements. You will receive a runtime error if you try to rollback the database additional times.
- The ID on an sObject inserted after setting a savepoint is not cleared after a rollback. Create an sObject to insert after a rollback. Attempting to insert the sObject using the variable created before the rollback fails because the sObject variable has an ID. Updating or upserting the sObject using the same variable also fails because the sObject is not in the database and, thus, cannot be updated.

With this explanation and above considerations,

the correct answer is "**Static variables are not reverted during a rollback**".

Reference:

https://developer.salesforce.com/docs/atlas.en-us.234.0.apexcode.meta/apexcode/langCon_apex_transaction_control.htm

Q.6. Refer to the following code snippet:

```

public class LeadController {
    public static List<Lead> getFetchLeadList(String searchTerm, Decimal aRevenue) {
        String safeTerm = '%' + searchTerm.escapeSingleQuote() + '%'
        return [
            SELECT Name, Company, AnnualRevenue
            FROM Lead
            WHERE AnnualRevenue >= :aRevenue
            AND Company LIKE :safeTerm;
            LIMIT 20
        ];
    }
}

```

A developer created a JavaScript function as part of a Lightning web component (LWC) that surfaces information about Leads by imperatively calling getFetchLeadList when certain criteria are met.

What are three changes the developer should implement in the Apex class above to ensure the LWC can display data efficiently while preserving security?

Choose 3 answers.

Answered Wrong

Ask Instructor

1. Annotate the Apex method with `@AuraEnabled(Cacheable=true)`.
(Correct Answer)
2. Implement the `without sharing` keyword in the class declaration.
3. Implement the `with sharing` keyword in the class declaration.
(Correct Answer)(Selected Answer)
4. Use the `WITH SECURITY_ENFORCED` clause within the SOQL query.
(Correct Answer)(Selected Answer)
5. Annotate the Apex method with `@AuraEnabled`.
(Selected Answer)

This is a scenario-based question which is related to calling Apex Method from Lightning Web Components. The scenario given in this question is:

- There is an Apex Controller "LeadController" and a method "getFetchLeadList()" which returns List of Lead based on searchTerm.
- There is a Lightning Web Component having a JavaScript function that imperatively calls "getFetchLeadList()" method when certain criteria are met.

The ask is: In the given apex code, what are three changes the developer should implement to ensure LWC can display data efferently while preserving security.

Let's check the given options and find the correct answers.

"Implement the with sharing keyword in the class declaration" is correct because as per apex best practice, Apex without an explicit sharing declaration is insecure by default. We strongly recommend that you always specify a sharing declaration for a class. Use the with sharing keyword when declaring a class to enforce sharing rules of the current user. Explicitly setting this keyword ensures that Apex code runs in the current user context.

"Annotate the Apex method with @AuraEnabled(Cacheable=true)" is correct because:

- We need to use `@AuraEnabled` on Apex class static methods to make them accessible as remote controller actions in your Lightning components.
- To improve runtime performance, set `@AuraEnabled(cacheable=true)` to cache the method results on the client. To set `cacheable=true`, a method must only get data. It can't mutate data.

"Use the WITH SECURITY_ENFORCED clause within the SOQL query" is correct because we should use the `WITH SECURITY_ENFORCED` clause to enable field- and object-level security permissions checking for SOQL SELECT queries in Apex code, including subqueries and cross-object relationships.

Screenshot—Updated Code given in Question

```

LeadController.apxc
Code Coverage: None API Version: 54
1 public with sharing class LeadController {
2
3     @AuraEnabled(cacheable=true)
4     public static List<Lead> getFetchLeadList(String searchTerm, Decimal aRevenue){
5         String safeTerm = '%' + String.escapeSingleQuotes(searchTerm) + '%';
6
7         return [
8             SELECT Name, Company, AnnualRevenue
9                 FROM Lead
10                WHERE AnnualRevenue >= :aRevenue
11                AND Company LIKE :safeTerm
12                WITH SECURITY_ENFORCED
13                LIMIT 20
14        ];
15    }
16 }
17 }
```

References:

- https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_classes_keywords_sharing.htm
- https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/controllers_server_apex_auraenabled_annotation.htm
- https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_classes_with_security_enforced.htm

Q.7. Users complain that a page is very slow to respond. Upon investigation, the query below is found to perform slowly.

SELECT Id, Name FROM Contact WHERE CustomField_c = null;
Which two actions can a developer take to improve performance?
Choose 2 answers.

Answered Right

Ask Instructor

1. Make the `CustomField_c` field an External ID.
(Correct Answer)(Selected Answer)
2. Contact Salesforce customer support to create a custom index to include null values.

3. Add a **LIMIT clause to the query to reduce the number of records returned.**
 4. Make the field **CustomerField_c required because Salesforce field indexes do not include nulls.**
- (Correct Answer)(Selected Answer)

This question is related to the concept of Selective Queries in Salesforce. First of all, let's learn about few concepts.

Index Fields in Salesforce

In the modern cloud application with a large data set like Salesforce, indexing is very important to allow an application to search and return results to users at a minimum time and cost.

There are **2 types** of indexes in Salesforce:

- Standard Index
- Custom Index

Salesforce maintains indexes on the following fields for most objects:

- Salesforce record ID
- RecordTypeID
- Division
- CreatedDate
- SystemModstamp (LastModifiedDate)
- Name
- Email (for contacts and leads)
- Lookups and master-detail relationship

Salesforce administrator can define a **custom index field** by making by making **the field as "External ID" or unique field for the following field type:**

- Number
- Auto-number
- Text
- Email

Selective SOQL Queries:

SOQL queries must be selective, particularly for queries inside triggers for the best performance. To avoid long execution times, non-selective SOQL queries may be terminated by the system.

A query is selective when one of the query filters is on an indexed field and the query filter reduces the resulting number of rows below a system-defined threshold.

Now, coming back to the question,

- The Query given is: `SELECT Id, Name FROM Contact WHERE CustomField_c = null;`
- The ask is how to improve the performance to Query.

Two correct answers are:

"Make the CustomField__c field an External ID." is correct because we can define custom index field by making it External Id.

and

"Make the field CustomerField_c required because Salesforce field indexes do not include nulls."

For Reference & More Details:

<https://help.salesforce.com/s/articleView?id=000325247&type=1>

https://developer.salesforce.com/docs/atlas.en-us.228.0.salesforce_large_data_volumes_bp.meta/salesforce_large_data_volumes_bp/lv_deployments_case_studies_indexing_wi

Q.8. A company needs to automatically delete sensitive information after seven years. This could delete almost a million records every day.
How can this be achieved?

Answered Right

Ask Instructor

1. Use aggregate functions to query for records older than seven years, and then delete the *AggregateResult* objects.
 2. Schedule an `@future` process to query records older than seven years, and then recursively invoke itself in 1,000 record batches to delete them.
 3. Perform a SOSL statement to find records older than 7 years, and then delete the entire result set.
 4. Schedule a batch Apex process to run every day that queries and deletes records older than seven years.
- (Correct Answer)(Selected Answer)

This is a scenario-based question. The scenario given in this question is:

- A company needs to automatically delete sensitive information after seven years.
- This could delete almost a million records every day.

Conclusion

- The records deleted will be in high volume (millions of records)
- This scenario does not require any synchronous behavior as the records should be deleted after 7 years.

Solution Approach

- We need to use asynchronous apex feature of Salesforce as the data deletion need to happen in backend without impacting any other logic.
- We need to specifically use "Batch Apex" (Asynchronous Apex) because the requirement is of data cleaning which may be millions of records.

Let's learn more about high-level overview of Asynchronous Apex features with use cases.

| Type | Overview | Common Scenarios |
|----------------|--|---|
| Future Methods | Run in their own thread, and do not start until resources are available. | Web service callout. |
| Batch Apex | Run large jobs that would exceed normal processing limits. | Data cleansing or archiving of records. |
| Queueable Apex | Similar to future methods, but provide additional job chaining and allow more complex data types to be used. | Performing sequential processing operations with external Web services. |
| Scheduled Apex | Schedule Apex to run at a specified time. | Daily or weekly tasks. |

With this explanation,

the correct answer is "**Schedule a batch Apex process to run every day that queries and deletes records older than seven years.**"

Reference:

https://trailhead.salesforce.com/content/learn/modules/asynchronous_apex/async_apex_introduction

Q.9. Universal Containers needs to integrate with a Heroku service that resizes product images submitted by users.

What are two alternatives to implement the integration and protect against malicious calls to the Heroku app's endpoint?

Choose 2 answers.

Answered Wrong

Ask Instructor

1. Create a Workflow Rule with an Outbound Message allowing the Heroku app to automatically store the resized images in Salesforce.
2. Create a Workflow Rule with an Outbound Message and select Send Session ID so that the Heroku app can use it to send the resized images back to Salesforce.
(Correct Answer)
3. Create a trigger that uses an @future Apex HTTP callout passing JSON serialized data and some form of pre-shared secret key, so that the Heroku app can authenticate requests and store the resized images in Salesforce.
(Correct Answer) (Selected Answer)
4. Create a trigger that uses an @future Apex HTTP callout passing JSON serialized data; therefore the Heroku app can automatically reply back to the callout with the resized images in Salesforce.
(Selected Answer)

This is a scenario-based question which is related to "Salesforce & Heroku Integration". The scenario given in this question is:

- Universal Containers needs to integrate with a Heroku service that resizes product images submitted by users.

The ask in this question is:

- What are the ways to implement the integration and protect against malicious calls to the Heroku app's endpoint.

As per Salesforce Documentation, there are two primary methods to call a Heroku app based on an event in Salesforce:

- Workflow outbound messages**

- A workflow outbound message declaratively makes a SOAP call.
- While configuring outbound message, if we select "**Send Session**

Id" the Heroku app can use that token to make REST API calls on the user's behalf. If you don't send the session ID, there's no way to check that the request was valid or protect against malicious calls to your Heroku app's API endpoint.

- Apex HTTP callouts**

- An Apex HTTP callout programmatically makes a REST call to a Heroku app.

- With Apex triggers, you can use some form of pre-shared key** to authenticate requests, avoiding the potential for malicious requests. You can also have the payload include a session ID to let the Heroku app make REST API requests back to Salesforce to fetch or update data.

With this explanation, the correct answers are:

"Create a Workflow Rule with an Outbound Message and select Send Session ID so that the Heroku app can use it to send the resized images back to Salesforce."

and

"Create a trigger that uses an @future Apex HTTP callout passing JSON serialized data and some form of pre-shared secret key, so that the Heroku app can authenticate requests and store the resized images in Salesforce."

For Reference:

https://trailhead.salesforce.com/en/content/learn/modules/salesforce_heroku_integration/callouts_workflow_with_heroku

Q.10. A developer wrote a trigger on Opportunity that will update a custom Last Sold Date field on the Opportunity's Account whenever an Opportunity is closed. In the test class for the trigger, the assertion to validate the Last Sold Date field fails.

What might be causing the failed assertion?

Answered Right

Ask Instructor

1. The test class is not using System.runAs() to run tests as a Salesforce administrator.
2. The test class has not defined an Account owner when inserting the test data.
3. The test class has not re-queried the Account record after updating the Opportunity.
(Correct Answer) (Selected Answer)
4. The test class has not implemented seeAllData = true in the test method.

This is a scenario-based question related to the concept of Apex Trigger and Test Class. The scenario given in this question is:

- There is an Apex Trigger on Opportunity that will update a custom Last Sold Date field on the Opportunity's Account whenever an Opportunity is closed.
- There is an Apex Test class written to cover this scenario.

The ask in this question is:

- In the Apex Test class for the trigger, the assertion to validate the Last Sold Date field fails. What might be causing the failed assertion?

Let's recreate the Scenario.

Screenshot – Apex Trigger

```

OpportunityTrigger.apc
Code Coverage: None • API Version: 53
1 * trigger OpportunityTrigger on Opportunity (after update) {
2
3     Map<Id,Opportunity> oMap = new Map<Id,Opportunity>();
4     oMap = trigger.oldMap;
5     Set<Id> accountIds = new Set<Id>();
6     List<Account> listAccountToUpdate = new List<Account>();
7
8     if(Trigger.isAfter){
9         if(Trigger.isUpdate){
10             for(Opportunity oppt : Trigger.New){
11                 String oldStage = oMap.get(oppt.Id).StageName;
12                 if(oppt.StageName != oldStage && (oppt.StageName == 'Closed Won' || oppt.StageName == 'Closed Lost')){
13                     accountIds.add(oppt.AccountId);
14                 }
15             }
16
17             for(Account acct : [SELECT Id, Last_Sold_Date__c FROM Account WHERE Id IN :accountIds]){
18                 acct.Last_Sold_Date__c = System.today();
19                 listAccountToUpdate.add(acct);
20             }
21             update listAccountToUpdate;
22         }
23     }
}

```

Annotations in the code:

- A red box with an arrow points to the condition in line 12: "if(oppt.StageName != oldStage && (oppt.StageName == 'Closed Won' || oppt.StageName == 'Closed Lost'))". The annotation text is: "Checking if Opportunity is updated to any of the closed stage, collect the corresponding accountId".
- A red box with an arrow points to the assignment in line 18: "acct.Last_Sold_Date__c = System.today();". The annotation text is: "For these Accounts, update the Last Sold Date field to today".
- A red box with an arrow points to the update statement in line 21: "update listAccountToUpdate;".

Screenshot – Apex Trigger Test

```

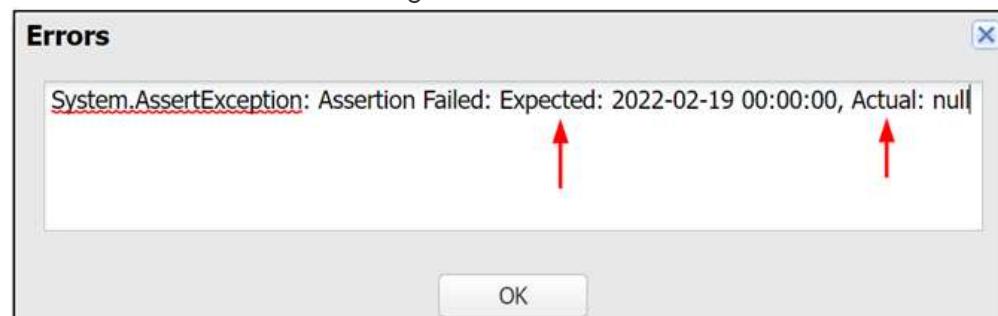
OpportunityTriggerTest.apc
Code Coverage: None • API Version: 54
1 @isTest
2 public class OpportunityTriggerTest {
3
4     @isTest
5     public static void testTriggerLogic(){
6         //create test account record
7         Account acct = new Account(Name='AB Textile Test', Last_Sold_Date__c = null);
8         insert acct;
9
10        //create test opportunity record for Account Record
11        Opportunity oppt = new Opportunity(Name='Opp for AB Textile Test', CloseDate = System.today(),
12                                            StageName = 'Prospecting', AccountId = acct.Id);
13        insert oppt;
14
15        Test.startTest();
16        Opportunity opp1 = [SELECT Id, StageName from Opportunity WHERE Id = :oppt.Id];
17        opp1.StageName = 'Closed Won'; //updating the opportunity to Closed
18        update opp1;
19        Test.stopTest();
20
21        //as per scenario, we know that the opportunity is updated to Closed, the Account.Last_Sold_Date__c should be populated
22        System.assertEquals(System.today(), acct.Last_Sold_Date__c); // After opportunity is updated, we expect the Last Sold Date of Account to be updated to TODAY
23    }
24 }

```

Annotations in the code:

- A red box with an arrow points to the update statement in line 18: "opp1.StageName = 'Closed Won';". The annotation text is: "As per the scenario, updating the Opportunity Stage to Closed".
- A red box with an arrow points to the assertion in line 22: "System.assertEquals(System.today(), acct.Last_Sold_Date__c);". The annotation text is: "After opportunity is updated, we expect the Last Sold Date of Account to be updated to TODAY".

Screenshot – Error while running above Test Class



As per the scenario given in question, the assertion is getting failed with error:
Expected = 2022-02-19 and Actual = null for Last Sold Field on Account.

But as per the Trigger Code, we know that when the Opportunity Stage is updated to "Closed", the Last Sold Date field on Account should be updated to TODAY's date but WHY its null.

Please Note: In the Apex Test Class, we are using Account acct to check the Last Sold Date after Opportunity is updated. But the correct approach is we should re-query the Account record after the Opportunity record is updated.

Screenshot – Correct Apex Test Class

```

1  @isTest
2  public class OpportunityTriggerTest {
3
4      @isTest
5      public static void testTriggerLogic(){
6          //create test account record
7          Account acct = new Account(Name='AB Textile Test', Last_Sold_Date__c = null);
8          insert acct;
9
10         //create test opportunity record for Account Record
11         Opportunity opp = new Opportunity(Name='Opp for AB Textile Test', CloseDate = System.today(),
12                                         StageName = 'Prospecting', AccountId = acct.Id);
13         insert opp;
14
15         Test.startTest();
16         Opportunity opp1 = [SELECT Id, StageName from Opportunity WHERE Id = :opp.Id];
17         opp1.StageName = 'Closed Won'; //updating the opportunity to Closed
18         update opp1;
19         Test.stopTest();
20
21         //as per scenario, we know that the opportunity is updated to Closed, the Account.Last_Sold_Date__c should be populated
22         //fetch the last value of Account
23         Account getAcct = [SELECT Id, Last_Sold_Date__c FROM Account WHERE Id = :acct.Id];
24         System.assertEquals(System.today(), getAcct.Last_Sold_Date__c); //Re-Query the Account after opportunity update and inserting
25     }
26 }

```

With this explanation, the correct answer is

"The test class has not re-queried the Account record after updating the Opportunity."

Q.11. A developer created an Apex class that makes an outbound RESTful callout. The following class was created to all send a fake response in Apex test methods.

```

@isTest
public class TestHttpCalloutMock implements HttpCalloutMock {
    public HttpResponse respond (HTTPRequest request) {
        HttpResponse response = new HttpResponse();
        response.setHeader('Content-Type', 'application/json');
        response.setBody('{"colors": ["red", "blue", "yellow", "green", "pink"]}');
        response.setStatusCode(200);
        return response;
    }
}

```

Which method can be called to return this fake response in the test methods?

Answered Right

Ask Instructor

1. **Test.setMock**
(Correct Answer)(Selected Answer)
2. **testSetup**
3. **Test.setTestData**
4. **Test.createStub**

This question is related to Testing HTTP Callouts by Implementing the HttpCalloutMock interface. The scenario given in the question is:

- The sample Apex Test code is given to create fake response.

The ask in this question is:

- Which method can be called to return this fake response in the test methods?

As per the Salesforce Document, one we have specified the values of the fake response, instruct the Apex runtime to send this fake response by calling

Test.setMock in your test method. For the first argument, pass `HttpCalloutMock.class`, and for the second argument, pass a new instance of your interface implementation of `HttpCalloutMock`, as follows:

```
1  Test.setMock(HttpCalloutMock.class, new YourHttpCalloutMockImpl());
```

With this explanation,

the correct answer is "**Test.setMock**"

Reference:

https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_classes_restful_http_testing_httpcalloutmock.htm

Q.12. A developer created and tested a Visualforce page in their developer sandbox, but now receives reports that user encounter ViewState errors when using it in Production.

What should the developer ensure to correct these errors?

Answered Right**Ask Instructor**

- 1. Ensure queries do not exceed governor limits.**
- 2. Ensure variables are marked as Transient.**
(Correct Answer)(Selected Answer)
- 3. Ensure properties are marked as private.**
- 4. Ensure profiles have access to the Visualforce page.**

This question is related to the performance of visual force page in Salesforce.

The scenario given in this question is:

- A developer created and tested a Visualforce page in their developer sandbox, but now receives reports that user encounter ViewState errors when using it in Production

Visualforce is designed to provide developers with the ability to match the functionality, behavior, and performance of standard Salesforce pages.

There are different Visualforce performance issues and one of them is View State Size.

- The view state size of your Visualforce pages must be under 170KB. By reducing your view state size, your pages can load quicker and stall less often.
- You can monitor view state performance through the View State tab in the development mode footer and take the following actions:
 - Use the transient keyword in your Apex controllers for variables that aren't essential for maintaining state and aren't necessary during page refreshes.
 - If you notice that a large percentage of your view state comes from objects used in controllers or controller extensions, consider **refining your SOQL calls** to return only data that's relevant to the Visualforce page.
 - If your view state is affected by a large component tree, try **reducing the number of components your page depends on**.

With this explanation, the correct answer is "**Ensure variables are marked as Transient.**"

Reference:

https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages_pages_best_practices_performance.htm

Create Notes

Q.13. Given a list of Opportunity records named opportunityList, which code snippet is best for querying all Contacts of the Opportunity's Account?

Answered Right**Ask Instructor**

- ```

List <Contact> contactList = new List <Contact> ();
Set <Id> accountIds = new Set <Id> ();
for (Opportunity o : opportunityList) {
 contactIds.add(o.ContactId);
}
for (Contact c : [SELECT Id FROM Contact WHERE Id IN :contactIds]) {
 contactList.add (c);
}

List <Contact> contactList = new List <Contact> ();
for (Contact c : [SELECT Id FROM Contact WHERE AccountId IN
2. : opportunityList.AccountId]) {
 contactList.add (c);
}

3.
List <Contact> contactList = new List <Contact> ();
Set <Id> accountIds = new Set <Id> ();
for (Opportunity o : opportunityList) {
 accountIds.add(o.AccountId);
}
for (Account a : [SELECT Id, (SELECT Id FROM Contacts) FROM Account WHERE Id IN
:accountIds]) {
 contactList.addAll (a.Contacts);
}
(Correct Answer)(Selected Answer)

List <Contact> contactList = new List <Contact> ();
for (Contact c : [SELECT Id FROM Contact WHERE AccountId IN
4. : opportunityList.AccountId]) {
 contactList.add (c);
}

```

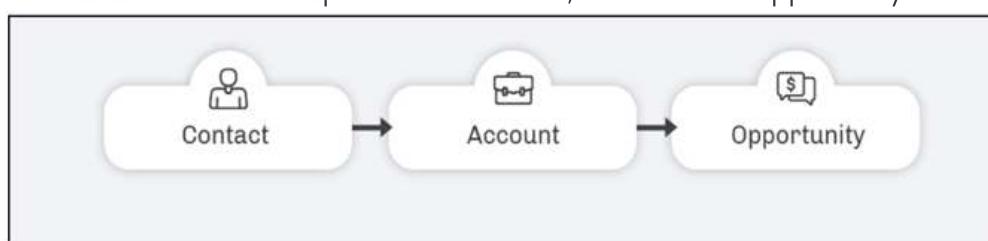
This is a scenario-based question which is related to the SOQL query in Salesforce. The scenario given in this question is:

- Given a list of Opportunity records named opportunityList, which code snippet is best for querying all Contacts of the Opportunity's Account

**Please Note:**

- In Salesforce, contact records do not have a direct relationship with Opportunities.
- Opportunities have a relationship with an Account record, and the Account has a relationship with Contacts – yet, no direct relationship.

**Screenshot – Relationship between Contact, Account and Opportunity**



Let's check given options.

Below Option is NOT CORRECT because there is no direct relation between Contact and Opportunity. Therefore, we can not have ContactId on Opportunity record.

```

List <Contact> contactList = new List <Contact>();
Set <Id> accountIds = new Set <Id> ();
for(Opportunity o : opportunityList) {
 contactIds.add(o.ContactId);
}
for(Contact c : [SELECT Id FROM Contact WHERE Id IN :contactIds]){
 contactList.add(c);
}

```

Below option is NOT CORRECT because opportunityList.AccountId is invalid.

```

List <Contact> contactList = new List <Contact>();
for (Contact c : [SELECT Id FROM Contact WHERE AccountId IN
:opportunityList.AccountId]){
 contactList.add(c);
}

```

Below option is NOT CORRECT because opportunityList.AccountId is invalid.

```

List <Contact> contactList = new List <Contact>();
for (Contact c : [SELECT Id FROM Contact WHERE AccountId IN
:opportunityList.AccountId]){
 contactList.add(c);
}

```

Below option is **CORRECT**

```

List <Contact> contactList = new List <Contact>();
Set <Id> accountIds = new Set <Id> ();
for(Opportunity o : opportunityList){
 accountIds.add(o.AccountId);
}
for(Account a : [SELECT Id, (SELECT Id FROM Contacts) FROM Account WHERE Id IN
:accountIds]){
 contactList.addAll(a.Contacts);
}

```

Because it uses the correct approach by querying **parent-to-child**, which are almost always one-to-many. Specify these relationships using a subquery (enclosed in parentheses), where the initial member of the FROM clause in the subquery is related to the initial member of the outer query FROM clause. **Note that for standard object subqueries, you should specify the plural name of the object as that is the name of the relationship for each object.**

- Collecting AccountId in a collection using Set
- Querying Account with Contact using Set of AccountId
- Adding contacts to a List named as ContactList

**Reference:**

[https://developer.salesforce.com/docs/atlas.en-us.soql\\_sosl.meta/soql\\_sosl/sforce\\_api\\_calls\\_soql\\_relationships\\_query\\_using.htm#force\\_api\\_calls\\_soql\\_relationships\\_query\\_using](https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/sforce_api_calls_soql_relationships_query_using.htm#force_api_calls_soql_relationships_query_using)

Q.14. A developer creates an application event that has triggered an infinite loop.

What may have caused this problem?

**Answered Wrong**

**Ask Instructor**

1. The event handler calls a trigger.
2. An event is fired 'ontouchend' and is unhandled.
3. The event has multiple handlers registered in the project. **(Selected Answer)**
4. The event is fired from a custom renderer. **(Correct Answer)**

This question is related to an issue while creating Application Event in Aura Component. The issue given in this question while creating Application Event is:

- Application event is triggering an infinite loop.

Let's understand basics about Application Event

Application events follow a traditional publish-subscribe model. An application event is fired from an instance of a component. All components that provide a handler for the event are notified.

As per Salesforce documentation, there are some anti-patterns that we should avoid when using events.

**Don't Fire an Event in a Renderer** ←

Firing an event in a renderer can cause an infinite rendering loop.

**Don't do this!**

```
1. afterRender: function(cmp, helper) {
2. this.superAfterRender();
3. $A.get("e.myNamespace:myEvent").fire();
4. }
```

Instead, use the `init` hook to run a controller action after component construction but before rendering. Add this code to your component:

```
1. <aura:handler name="init" value="{!this}" action=" {!c.doinit} "/>
```

For more details, see [Invoking Actions on Component Initialization](#).

**Don't Use onclick and ontouchend Events**

You can't use different actions for `onclick` and `ontouchend` events in a component. The framework translates touch-tap events into clicks and activates any `onclick` handlers that are present.

With this explanation,

the correct answer is "**The event is fired from a custom renderer**" because it can cause an infinite rendering loop.

#### References:

[https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/events\\_anti\\_patterns.htm](https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/events_anti_patterns.htm)  
[https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/events\\_application.htm](https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/events_application.htm)

**Q.15. A developer wrote an Apex class to make several callouts to an external system.**

If the URLs used in these callouts will change often, which feature should the developer use to minimize changes needed to the Apex class?

**Answered Right**

**Ask Instructor**

1. Connected Apps.
2. Named Credentials. **(Correct Answer)(Selected Answer)**
3. Remote Site Settings.
4. Session Id.

Create Notes

An Apex callout enables you to tightly integrate your Apex with an external service by making a call to an external Web service or sending a HTTP request from Apex code and then receiving the response. Apex provides integration with Web services that utilize SOAP and WSDL, or HTTP services (RESTful services).

**Please Note:**

- Before any Apex callout can call an external site, that site must be registered in the Remote Site Settings page, or the callout fails. Salesforce prevents calls to unauthorized network addresses.
- If the callout specifies a named credential as the endpoint, you don't need to configure remote site settings. A named credential specifies the URL of a callout endpoint and its required authentication parameters in one definition.

**About Named Credentials**

- A named credential specifies the URL of a callout endpoint and its required authentication parameters in one definition. Salesforce manages all authentication for Apex callouts that specify a named credential as the callout endpoint so that your code doesn't have to. You can also skip remote site settings, which are otherwise required for callouts to external sites, for the site defined in the named credential.
- Named Credentials also include an Outbound Network Connection field that you can use to route callouts through a private connection. By separating the endpoint URL and authentication from the callout definition, named credentials make callouts easier to maintain. **For example, if an endpoint URL changes, you update only the named credential. All callouts that reference the named credential simply continue to work.**

With this explanation, the correct answer is "**Named Credentials.**"

**References:**

- [https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_callouts.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_callouts.htm)  
[https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_callouts\\_named\\_credentials.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_callouts_named_credentials.htm)

Q.16. Universal Containers (UC) wants to develop a customer community to help their customers log issues with their Containers. The community needs to function for their German- and Spanish-speaking customers also. UC heard that it's easy to create an international community using Salesforce, and hired a developer to build out the site.

What should the developer use to ensure the site is multilingual?

Answered Right

Ask Instructor

1. Use Custom Labels to ensure custom messages are translated properly. (Correct Answer) (Selected Answer)
2. Use Custom Objects to translate custom picklist values.
3. Use Custom Metadata to translate custom picklist values.
4. Use Custom Settings to ensure custom messages are translated properly.

This is a scenario-based question which is related to customer community.

The scenario given in this question is:

- The community needs to function for their German- and Spanish-speaking customers.
- That means, the same community site should show content in specific language.

The ask is:

- How should the developer use to ensure the site is multilingual

If your Salesforce org has multiple languages enabled, manage translations so that your global users can use Salesforce in their language. One of the ways is to use Custom Labels.

Custom labels enable developers to create multilingual applications by automatically presenting information (for example, help text or error messages) in a user's native language. Custom labels are custom text values that can be accessed from Apex classes, Visualforce pages, Lightning pages, or Lightning components. The values can be translated into any language Salesforce supports.

With this explanation,

the correct answer is "**Use Custom Labels to ensure custom messages are translated properly**".

**References:**

[https://help.salesforce.com/s/articleView?id=sf.cl\\_about.htm&type=5](https://help.salesforce.com/s/articleView?id=sf.cl_about.htm&type=5)  
[https://help.salesforce.com/s/articleView?id=sf.workbench\\_overview.htm&type=5](https://help.salesforce.com/s/articleView?id=sf.workbench_overview.htm&type=5)

Q.17. Users report that a button on a custom Lightning web component is not working. However, there are no other details provided.

What should the developer use to ensure error messages are properly displayed?

Answered Right

Ask Instructor

1. Add JavaScript and HTML to display an error message. (Correct Answer) (Selected Answer)
2. Add a Try/Catch block surrounding the DML statement.
3. Use the Database method with allOrNone set to false.
4. Add the `<apex:messages/>` tag to the component.

This question is related to Lightning Web Component. The scenario given in this question is:

- There is a button on custom Lightning Web Component but it's not working.
- There are no other details provided.

The ask in this question is:

- What should the developer use to ensure that error messages are properly displayed.
- That means, whenever an error occurred, we need show that error to the user.

The correct answer of this question is "**Add JavaScript and HTML to display an error message**" because to display an error message in Lightning Web Component, we can add logic in to the JavaScript file.

That means, the js file can collect the error and display it. There are several ways to achieve this. One of the ways is to use toast component.

Let's see an example.

**Screenshot – LWC .html file**

```
force-app > main > default > lwc > demoShowError > demoShowError.html > ...
1 <template>
2 <lightning-card title="Notification" icon-name="custom:custom19">
3 <lightning-button label="Show Error" onclick={showErrorToast}></lightning-button>
4 </lightning-card>
5 </template>
6
7
```



**Screenshot – LWC .js file**

```
JS demoShowError.js X
force-app > main > default > lwc > demoShowError > JS demoShowError.js > ...
1 import { LightningElement } from 'lwc';
2 import { ShowToastEvent } from 'lightning/platformShowToastEvent'; ←
3 export default class DemoShowError extends LightningElement {
4
5 showErrorToast() {
6 const evt = new ShowToastEvent({ ←
7 title: 'Toast Error',
8 message: 'Some unexpected error',
9 variant: 'error',
10 mode: 'dismissable'
11 });
12 this.dispatchEvent(evt);
13 }
14 }
```

**Reference:**

<https://developer.salesforce.com/docs/component-library/bundle/lightning-platform-show-toast-event/documentation>

Q.18. A company wants to build a custom Aura component that displays a specified Account Field Set and that can only be added to the Account record page.

Which design resource configuration should be used?

Answered Right

Ask Instructor

1. `<design:component label="Account FS Component">
<design:attribute name="fieldSetName" label="Field Set Name" />
<sfdc:objects>
<sfdc:object>Account</ sfdc:object>
</ sfdc:objects>
</design:component>`
- (Correct Answer) (Selected Answer)
2. `<design:component label="Account FS Component">
<aura:attribute name="fieldSetName" label="Field Set Name" />
<sfdc:objects>
<sfdc:object>FieldSet</ sfdc:object>
</ sfdc:objects>
</design:component>`
3. `<design:component label="Account FS Component">
<aura:attribute name="fieldSetName" label="Field Set Name" />
<sfdc:objects>
<sfdc:object>Account</ sfdc:object>
</ sfdc:objects>
</design:component>`
4. `<design:component label="Account FS Component">
<design:attribute name="fieldSetName" label="Field Set Name" />
<sfdc:objects>
<sfdc:object>FieldSet</ sfdc:object>
</ sfdc:objects>
</design:component>`

Create Notes

This question is related to Aura Component Bundle Design Resources.

Let's learn basic about Design Resource:

Use a design resource to control which attributes are exposed to builder tools like the Lightning App Builder, Experience Builder, or Flow Builder. A design resource lives in the same folder as your .cmp resource, and describes the design-time behavior of the Aura component—information that visual tools need to display the component in a page or app.

### Screenshot – Aura Component (.cmp file)



### Screenshot – Aura Component (.design file)



#### design:attribute

To make an Aura component attribute available for admins to edit in tools such as the App Builder, add a design:attribute node for the attribute into the design resource.

#### <sfdc:objects> and <sfdc:object>

Use these tag sets to restrict your component to one or more objects.



Now, coming back to the question it says

- To displays a specified Account Field Set
  - For this, we need to use <design:attribute>
- To restrict the component to Account record page.
  - For this, we need to use <sfdc:object> Account <sfdc:object>

With this explanation,

below is the **correct answer** because it uses <design:attribute> and <sfdc:object> Account <sfdc:object>

```

<design:component label="Account FS Component">
 <design:attribute name="fieldSetName" label="Field Set Name" />
 <sfdc:objects>
 <sfdc:object>Account</sfdc:object>
 </sfdc:objects>
</design:component>

```

#### Reference:

[https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/components\\_config\\_for\\_app\\_builder\\_design\\_files.htm](https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/components_config_for_app_builder_design_files.htm)

Q.19. How can a developer efficiently incorporate multiple JavaScript libraries in a Lightning component?

Answered Wrong

Ask Instructor

1. Use JavaScript remoting and script tags.
2. Implement the libraries in separate helper files. (Selected Answer)
3. Use CDNs with script attributes.

**4. Join multiple assets from a static resource.  
(Correct Answer)**

This question is related to using external JavaScript library in Lightning Component. As per Salesforce documentation, To reference a JavaScript library, upload it as a static resource and use a `<ltng:require>` tag in your .cmp or .app markup.

Therefore, there are **two steps:**

1. Load the external JavaScript library in static resource.
2. Use `<ltng:require>` tag to load the JavaScript library in lightning Component as shown below:

**Screenshot** – Use the join operator to include multiple libraries in Lightning Component

```

1 <aura:component>
2 <!-- For one js library --> <!-- Here filename1 is static resource name -->
3 <ltng:require scripts="{$Resource.filename1}"></ltng:require>
4
5 <!-- For more than one js file --> <!-- Here filename1 and filename2 is static resource name -->
6 <ltng:require scripts="!join(',', $Resource.filename1, $Resource.filename2)"></ltng:require>
7 </aura:component>
8
9

```

With this explanation,

the correct answer is "**Join multiple assets from a static resource**" because we need to use join operator to incorporate multiple JavaScript libraries in a Lightning component.

For Reference & More Detail:

[https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/js\\_libs\\_platform.htm](https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/js_libs_platform.htm)

---

Q.20. A developer is building a Lightning web component to get data from an Apex method called `getdata` that takes parameter, `name`.The data should be retrieved when the user clicks the Load Data button.

```

import { LightningElement } from 'lwc';
import getData from '@salesforce/apex/AccountInfo.getData';
export default class AccountInfo extends LightningElement {
 account;
 name;

 nameChanged(evt) {
 this.name = evt.target.value;
 }
 loadData() {
 }
}

```

**Component markup Snippet:**

```

<lightning-button label="Load Data" onclick={loadData}></lightning-button>
<lightning-input type="text" label="Name" value={name} onchange={nameChanged}></lightning-input>

```

What must be added to get the data?

**Answered Right**

**Ask Instructor**

**1.**

Add `@wire(getData, {name: '$name'})` to the account field and `this.account = getData()`; to the `loadData()` function.

**2. Add `this.account = getData(this.name)` ; to the `loadData()` function.**

**3.**

Add `@wire(getData, {name: '$name'})` to the account field and delete `loadData()` because it is not needed.

**4.**

Add `getData({name: this.name}).then(result => { this.account = result; })`; to the `loadData()` function.

**(Correct Answer)(Selected Answer)**

This is a scenario-based question related to Lightning Web Component and calling Apex Method from Lightning Web Component.

The scenario given in this question is:

- There is a Lightning Web Component named AccountInfo
- There is a Apex Method named as "getData()
- The LWC uses this method "getData()" to get the Account Data
- The data should be retrieved when user clicks the "Load Data" button.

(IMPORTANT)

### About Calling Apex Methods from LWC

Lightning web components can import methods from Apex classes. The imported methods are functions that the component can call either via:

- @wire
- Imperatively

Now, coming back to the question, it says that the data should be retrieved when user clicks the "Load Data" button. Therefore, we can not use @wire to call apex method. We need to call apex method imperatively.

### Call Apex Method Imperatively

To control when the method invocation occurs (for example, in response to clicking a button), call the method imperatively. When you call a method imperatively, you receive only a single response.

In the following scenarios, you must call an Apex method imperatively as opposed to using @wire.

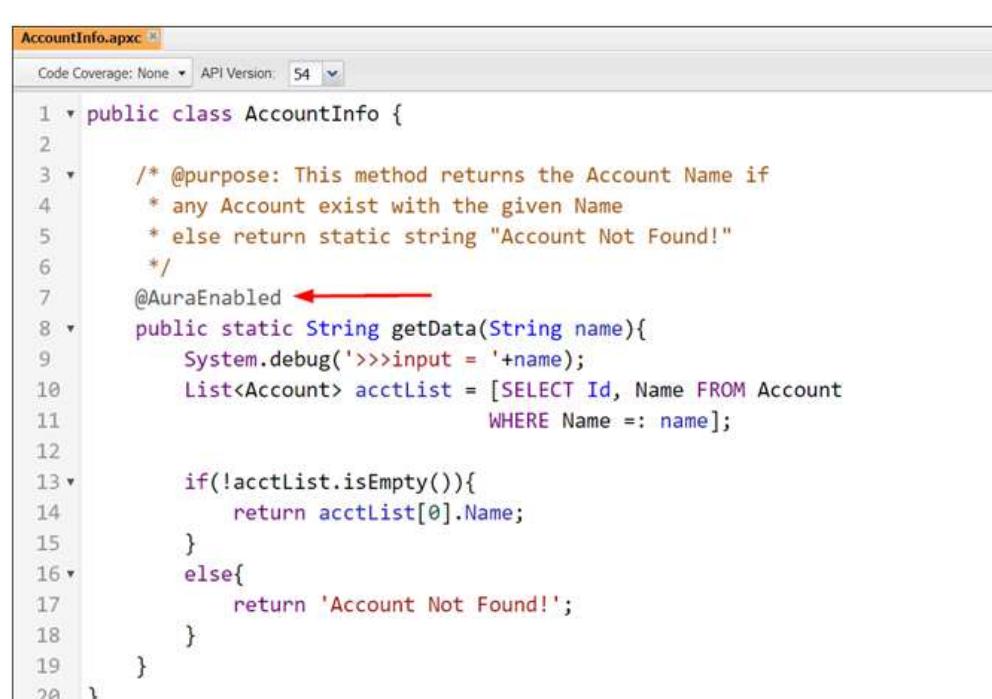
- To call a method that isn't annotated with cacheable=true, which includes any method that inserts, updates, or deletes data.
- To control when the invocation occurs.
- To work with objects that aren't supported by User Interface API, like Task and Event.
- To call a method from an ES6 module that doesn't extend LightningElement.

### Steps to call Apex Method Imperatively from LWC

- Import an Apex method into the component
- The imported function returns a promise. Use logic to assign the result to property.

Let's do some hands-on.

#### Screenshot – Apex Class & Method



```

AccountInfo.apxc
Code Coverage: None API Version: 54
1 * public class AccountInfo {
2
3 /* @purpose: This method returns the Account Name if
4 * any Account exist with the given Name
5 * else return static string "Account Not Found!"
6 */
7 @AuraEnabled ←
8 public static String getData(String name){
9 System.debug('>>>input = '+name);
10 List<Account> acctList = [SELECT Id, Name FROM Account
11 WHERE Name =: name];
12
13 if(!acctList.isEmpty()){
14 return acctList[0].Name;
15 }
16 else{
17 return 'Account Not Found!';
18 }
19 }
20 }

```

#### Screenshot – LWC (accountInfo.html)



```

1 <template>
2 <h1>This is Sample LWC - AccountInfo</h1>
3 <p>Enter the desired name in TEXT field and click button Load Data</p>
4 <lightning-button label="Load Data" onclick={loadData}></lightning-button>
5 <lightning-input type="text" label="Name" value={name} onchange={nameChanged}></lightning-input>
6
7 Account = {account}
8 </template>
9
10

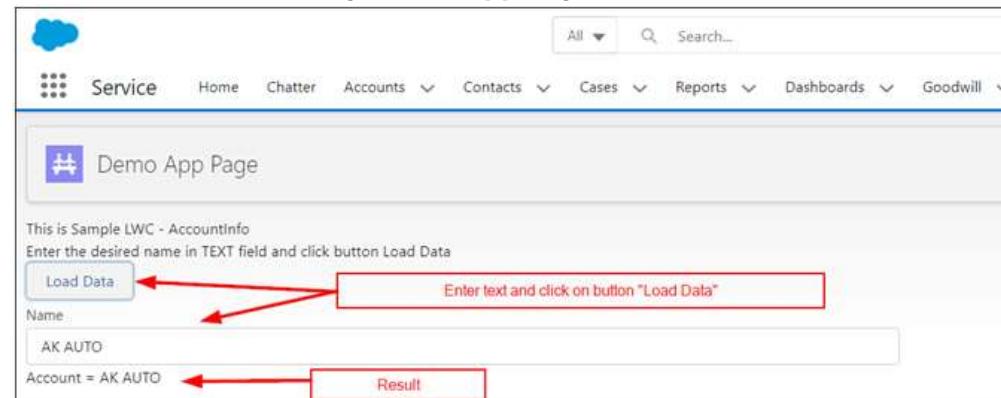
```

**Screenshot – LWC (accountInfo.js)**

```

1 import { LightningElement } from 'lwc';
2 import getData from '@salesforce/apex/AccountInfo.getData'; ← STEP 1: Import Apex Method
3
4 export default class AccountInfo extends LightningElement {
5 account;
6 name;
7
8 nameChanged(evt){
9 this.name = evt.target.value;
10 console.log('>>>'+this.name);
11 }
12
13 loadData(){
14 getData({name : this.name}) ← Use the imported function and get the result
15 .then(result=>{
16 this.account = result
17 });
18 }
19 }
20

```

**Screenshot – Demo (Using LWC as App Page)**

With this explanation, the correct answer is

Add `getData({name: this.name}).then(result => { this.account = result; })` to the `loadData()` function.

**Reference:**

[https://developer.salesforce.com/docs/component-library/documentation/en/lwc/lwc.apex\\_call\\_imperative](https://developer.salesforce.com/docs/component-library/documentation/en/lwc/lwc.apex_call_imperative)

Q.21. A large company uses Salesforce across several departments. Each department has its own Salesforce Administrator. It was agreed that each Administrator would have their own sandbox in which to test changes. Recently, users notice that fields that are recently added for one department suddenly disappear without warning. Also, Workflows that once sent emails and created tasks no longer do so. Which two statements are true regarding these issues and resolution? Choose 2 answers.

**Answered Right**

[Ask Instructor](#)

1. A sandbox should be created to use as a unified testing environment instead of deploying Change Sets directly to production.  
**(Correct Answer) (Selected Answer)**
2. The administrators are deploying their own Change Sets over each other, thus replacing entire Page Layouts and Workflows in Production.  
**(Correct Answer) (Selected Answer)**
3. The administrators are deploying their own Change Sets, thus deleting each other's fields from the object in production.
4. Page Layouts should never be deployed via Change Sets, as this causes Workflows and field-level Security to be reset and fields to disappear.

This is a scenario-based question related to the deploying components from Sandbox to Production.

**The scenario given in this question is:**

- In a company, there are different departments and each department has its own Salesforce administrator.
- Each administrator has its own Sandbox to test the changes done.
- There are below few issues they are noticing:
  - Fields that are recently added for one department suddenly disappear without warning.
  - Workflows that once sent emails and created tasks no longer do so.

**The ask is:**

What is the root-cause of above issues?

From the above scenario, we can see that each administrator has its own Sandbox and they can deploy components from Sandbox to Production. In this situation, there are more chances that the same component can be overlapped.

For example: Let's assume.

- Sandbox-1 having Case Page Layout having a field "Reason"
- Sandbox-2 having Case Page Layout does not have field "Reason"
- Now, if the Case Page Layout is deployed from Sandbox-1 to Production.

Therefore, the Case Page Layout in Production will have field "Reason"

- But, after that if Case Page Layout does not have field "Reason" is deployed from Sandbox-2 to Production. It will override the Case Page Layout in Production.

Therefore, as per best practice:

A Sandbox should be created to use as a unified testing environment and all the production deployment should happen from the unified Sandbox.

With this explanation, the correct answers are:

**"A sandbox should be created to use as a unified testing environment instead of deploying Change Sets directly to production."**

and

**"The administrators are deploying their own Change Sets over each other, thus replacing entire Page Layouts and Workflows in Production."**

**Reference:**

<https://help.salesforce.com/s/articleView?id=000338970&type=1>

---

Q.22. Which scenario requires a developer to use an Apex callout instead of Outbound Messaging?

Answered Wrong

Ask Instructor

1. The callout needs to be invoked from a Workflow Rule.
2. The target system uses a SOAP API.
3. The callout needs to be asynchronous.  
**(Selected Answer)**
4. The target system uses a REST API.  
**(Correct Answer)**

This is a scenario-based question which is related to using Apex and Outbound Messaging for callout to external system.

Salesforce provides the following features to implement Integration:

- Apex Callout
- Outbound Messaging (OM)

It is very important that we fully understand the capabilities of these features and then decide the suitable approach.

#### 1. Apex Callout:

- Enable Apex to invoke external web services. This allows you to connect to 3rd party web services such as Google, Amazon, Facebook, and any other external web service.

- Callout requires familiarity with Apex coding
- Transaction failure & retry logic should be explicitly coded
- Supports both SOAP & REST API Invocation
- Sync & Async requests handling is supported

#### 2. Outbound Messages:

- OM can be implemented using Point & Click tools without having to write any code in Salesforce.

- Sends configured fields in the form of XML SOAP message to the specified end-point.

- The destination end point should be setup in a specific way for it to accept the contents of the SOAP message being sent. We won't be able to use existing web services of the Target system.

- We can only send single parent record fields in the SOAP message.

Additional web service transaction should be issued back to Salesforce (from custom solution) using the session id sent in the SOAP message to get additional (related) records.

Now, coming back to the question, it says in which scenario we need to use Apex Callout instead of Outbound Messages.

Option "**The callout needs to be invoked from a Workflow Rule**" is NOT CORRECT because we can use Apex Callout from workflow rule.

Option "**The target system uses a SOAP API**" is NOT CORRECT because if the target system uses SOAP API, we can use both Apex Callout as well as Outbound Message.

Option "**The callout needs to be asynchronous**" is NOT CORRECT because for this scenario also, we can use either Apex or Outbound Message.

Option "**The target system uses a REST API**" is CORRECT because for this situation, we should only use Apex Callout.

---

#### Q.23. Which two relationship queries use the proper syntax?

Choose 2 answers.

Answered Right

Ask Instructor

1.

SELECT Id, Name, Account\_r.Name FROM Contact WHERE Account\_r.Industry = 'Media'

2. SELECT Name, (SELECT LastName FROM Contacts\_r) FROM Account.

3. SELECT Name, (SELECT LastName FROM Contacts) FROM Account.

(Correct Answer)(Selected Answer)

4.

SELECT Id, Name, Account.Name FROM Contact WHERE Account.Industry = 'Media'

(Correct Answer)(Selected Answer)

This question is related to the concept of Relationship Queries in Salesforce. Let's learn some basics about it.

### Relationship Queries

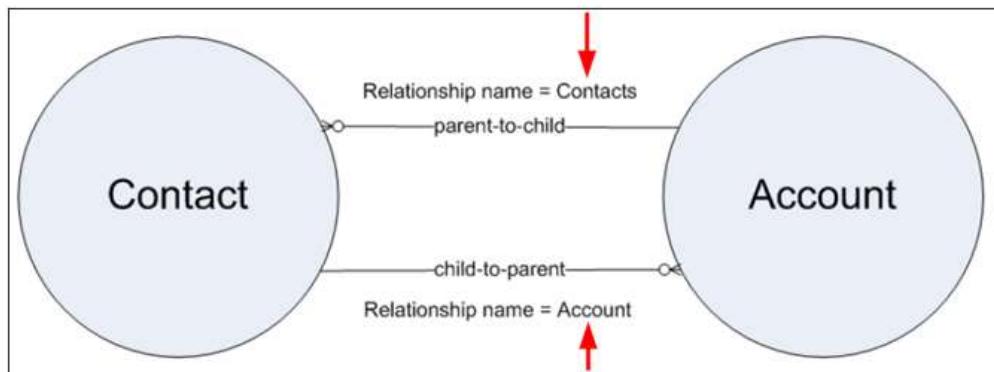
Client applications need to be able to query for more than a single type of object at a time. SOQL provides syntax to support these types of queries, called relationship queries, against standard objects and custom objects.

There are two types of relationship queries:

- Parent-to-child relationships
  - This simply means we query to parent object and also get child records.
- Child-to-parent relationships
  - This means we query to child and get parent fields.

### Understanding Relationship Names

Parent-to-child and child-to-parent relationships exist between many types of objects. For example, Account is a parent of Contact.



**For child-to-parent relationships**, contact child object has a child-to-parent relationship to the Account object, so the value of relationshipName in Contact is Account.

**For parent-to-child relationships**, the parent object has a name for the child relationship that is unique to the parent, the plural of the child object name. For example, Account has child relationshipsContacts and has a relationshipName as Contacts.

#### Syntax – child to parent Query

Specify these relationships directly in the SELECT, FROM, or WHERE clauses using the dot (.) operator.

`SELECT Id, Name, Account.Name FROM Contact`

#### Syntax – parent to child

Specify these relationships using a subquery (enclosed in parentheses), where the initial member of the FROM clause in the subquery is related to the initial member of the outer query FROM clause.

`SELECT Id, (SELECT LastName FROM Contacts) FROM Account`

Now, coming back to the options given.

Below option is **NOT CORRECT** because this is child-to-parent query having child as Contact and Account as parent and the relationshipName in Contact is Account but it uses Account\_r.

`SELECT Id, Name, Account_r.Name FROM Contact WHERE Account_r.Industry = 'Media'`

Below option is **NOT CORRECT** because this is parent-to-child query having child as Contact and Account as parent. Account has child relationshipsContacts and has a relationshipName as Contacts. But it uses Contacts\_r

`SELECT Name, (SELECT LastName FROM Contacts_r) FROM Account`

Below option is **CORRECT** as it uses correct relationshipName and syntax.

`SELECT Name, (SELECT LastName FROM Contacts) FROM Account`

Below option is **CORRECT** as it uses correct relationshipName and syntax.

`SELECT Id, Name, Account.Name FROM Contact WHERE Account.Industry = 'Media'`

#### References:

[https://developer.salesforce.com/docs/atlas.en-us.soql\\_sosl.meta/soql\\_sosl/sforce\\_api\\_calls\\_soql\\_relationships\\_understanding.htm#sforce\\_api\\_calls\\_soql\\_relationships\\_understanding](https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/sforce_api_calls_soql_relationships_understanding.htm#sforce_api_calls_soql_relationships_understanding)  
[https://developer.salesforce.com/docs/atlas.en-us.soql\\_sosl.meta/soql\\_sosl/sforce\\_api\\_calls\\_soql\\_relationships\\_query\\_using.htm#sforce\\_api\\_calls\\_soql\\_relationships\\_query\\_using](https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/sforce_api_calls_soql_relationships_query_using.htm#sforce_api_calls_soql_relationships_query_using)

Q.24. A developer is integrating with legacy on-premise SQL database. What should the developer use to ensure the data being integrated is matched to the right records in Salesforce?

**Answered Right**

**Ask Instructor**

1. Lookup field.
2. Formula field.
3. External Object.

**4. External Id field.**  
**(Correct Answer)(Selected Answer)**

This question is related to integrating legacy system with Salesforce. The ask in this question is "what should the developer use to ensure the data being integrated is matched to the right records in Salesforce"?

That means, this scenario is all ensuring record matching so no duplicate records are inserted / updated.

This can be achieved by using "External Id Field"

The External ID field allows you to store unique record IDs from an external system, typically for integration purposes. For example, if we have a legacy system which need to load/update data in Salesforce may reference these external records in Salesforce using unique IDs from Legacy system records.

To enable the External ID flag for a field:

- Navigate to the object's field list
- In Classic, go to Setup | Create | Objects | <object name> | Fields
- In Lightning Experience, go to Setup | Object Manager | <object name> | Fields & Relationships.

- Then, click Edit next to the field name enable the External ID checkbox.

**Screenshot – To make a field as External Id**

The screenshot shows the 'New Custom Field' setup page for an 'Account' object. The 'Field Label' is set to 'External Id'. The 'Length' field is set to 100. The 'Field Name' is set to 'External Id'. The 'Description' and 'Help Text' fields are empty. Under the 'Required' section, there are two checkboxes: 'Always require a value in this field in order to save a record' (unchecked) and 'Set this field as the unique record identifier from an external system' (checked). There are also two radio buttons: 'Treat "ABC" and "abc" as duplicate values (case insensitive)' (selected) and 'Treat "ABC" and "abc" as different values (case sensitive)'. Under the 'Unique' section, there are two checkboxes: 'Do not allow duplicate values' (unchecked) and 'Set this field as the unique record identifier from an external system' (checked). At the bottom, there are two checkboxes: 'Auto add to custom report type' (unchecked) and 'Add this field to existing custom report types that contain this entity' (checked). A red box highlights the 'Set this field as the unique record identifier from an external system' checkbox under the 'Required' section, and a red arrow points from the text 'Check the checkbox if you want to make the field as External Id' to this highlighted checkbox.

With this explanation, the correct answer is "**External Id field.**"

**Reference:**

[https://help.salesforce.com/s/articleView?id=sf.faq\\_import\\_general\\_what\\_is\\_an\\_external.htm&type=5](https://help.salesforce.com/s/articleView?id=sf.faq_import_general_what_is_an_external.htm&type=5)

**Q.25. Which technique can run custom logic when a Lightning web component is loaded?**

**Answered Right**

**Ask Instructor**

1. Use the `renderedCallback()` method.
2. Use the `connectedCallback()` method.  
**(Correct Answer)(Selected Answer)**
3. Use an aura:handler "init" event to call a function.
4. Call `$A.enqueueAction` passing in the method to call.

This question is related to the concept of "Lifecycle Hooks" in Lightning Web Component. The ask in this question is:

- Which technique can run custom logic when a Lightning web component is loaded. That means we want to execute some custom logic on load of Lightning Web Component.

#### Lifecycle Hooks

Lightning Web Components provides methods that allow you to "hook" your code up to critical events in a component's lifecycle. These events include when a component is:

- Created
- Added to the DOM
- Rendered in the browser
- Encountering errors
- Removed from the DOM

In short, A lifecycle hook is a callback method triggered at a specific phase of a component instance's lifecycle.

#### connectedCallback()

The connectedCallback() lifecycle hook fires when a component is inserted into the DOM. If you have to do something onload of the Lightning web component, use this method.

#### Screenshot – Sample Code using connectedCallback()

```
JS demoLWC.js X
force-app > main > default > lwc > demoLWC > JS demoLWC.js > ...
1 import { LightningElement } from 'lwc';
2
3 export default class LwcComponent extends LightningElement {
4
5 connectedCallback() { ←
6 console.log('LWC Component Loaded Successfully');
7 }
8 }
9
```

With this explanation, the correct answer is **"Use the connectedCallback() method."**

#### References:

- [https://developer.salesforce.com/docs/component-library/documentation/en/lwc/lwc.reference\\_lifecycle\\_hooks](https://developer.salesforce.com/docs/component-library/documentation/en/lwc/lwc.reference_lifecycle_hooks)
- <https://trailhead.salesforce.com/en/content/learn/modules/lightning-web-components-basics/create-lightning-web-components>

Q.26. A developer has a Batch Apex process, Batch\_Account\_Sales, that updates the sales amount for 10,000 Accounts on a nightly basis. The Batch Apex works as designed in the sandbox. However, the developer cannot get code coverage on the Batch Apex class.

The test class is below:

```
@IsTest private Batch_Account_Update_Test() {
 @IsTest static void UnitTest() {
 Account a = new Account(Name='test', Type='Customer', Sales_Amount__c=0);
 insert a;
 Batch_Account_Sales bas = new Batch_Account_Sales();
 ID jobid = database.executebatch(bas);
 }
}
```

What is causing the code coverage problem?

Answered Right

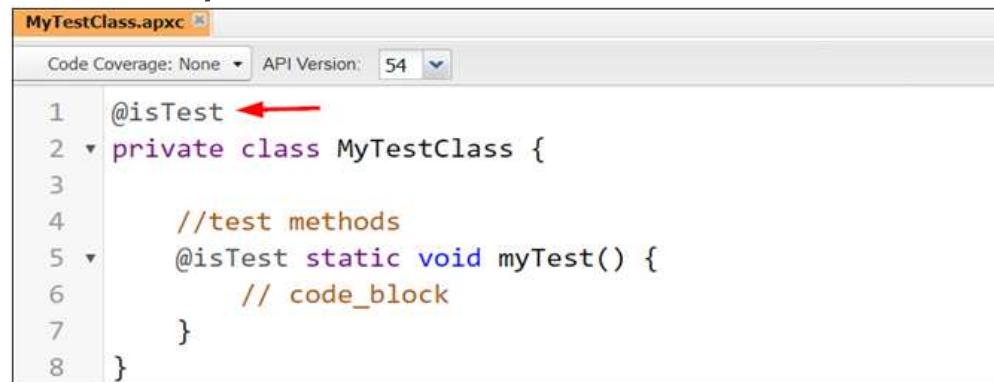
Ask Instructor

1. The executeBatch must fall within test.startTest() and test.stopTest() (Correct Answer) (Selected Answer)
2. The batch process will not recognize new accounts created in the same session.
3. The account creation already sets the sales amount to 0.
4. The batch needs more than one account record created.

This question is related to Apex Unit Test in Salesforce. The Apex testing framework enables you to write and execute tests for your Apex classes and triggers on the Lightning Platform. Apex unit tests ensure high quality for your Apex code and let you meet requirements for deploying Apex.

Test methods must be defined in test classes, which are classes annotated with `isTest`. This sample class shows a definition of a test class with one test method.

#### Screenshot – Syntax of Test Class



```
MyTestClass.apxc
Code Coverage: None API Version: 54
1 @isTest ←
2 private class MyTestClass {
3
4 //test methods
5 @isTest static void myTest() {
6 // code_block
7 }
8 }
```

Now, coming back to the question. The scenario given in this question is:

- There is a Batch Apex Class which works fine in Sandbox.
- There is a Apex Test class written for the Batch Apex but the developer cannot get code coverage on the Batch Apex class.

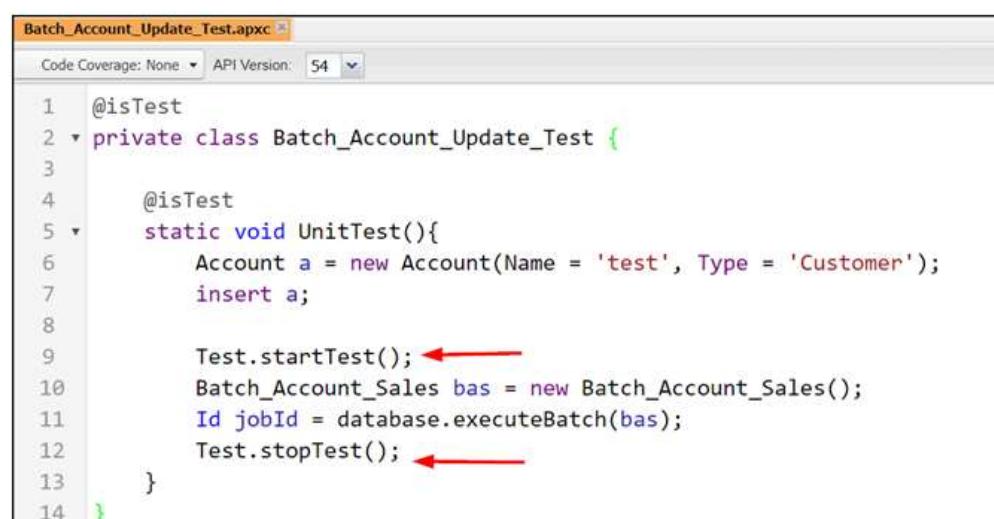
**Please Note:** The ask in this question is to write Apex Test for Batch Class which is Anynchronous in nature. A unit test forms a single transaction, and asynchronous code enqueued within that transaction cannot be executed until the transaction commits successfully.

For this reason, Salesforce has provided a framework to force asynchronous code to execute synchronously for testing: We enclose our test code between `Test.startTest()` and `Test.stopTest()` methods.

The system collects all asynchronous calls made after `startTest()`. When `stopTest()` is executed, these collected asynchronous processes are then run synchronously and complete before control returns to our code.

But as we see, the sample Apex Test code given in this question does not used `Test.startTest()` and `Test.stopTest()` methods.

#### Screenshot – Updated Apex Test Class



```
Batch_Account_Update_Test.apxc
Code Coverage: None API Version: 54
1 @isTest
2 private class Batch_Account_Update_Test {
3
4 @isTest
5 static void UnitTest(){
6 Account a = new Account(Name = 'test', Type = 'Customer');
7 insert a;
8
9 Test.startTest(); ←
10 Batch_Account_Sales bas = new Batch_Account_Sales();
11 Id jobId = database.executeBatch(bas);
12 Test.stopTest(); ←
13 }
14 }
```

With this explanation,

the correct answer is “**The executeBatch must fall within `test.startTest()` and `test.stopTest()`.**”

#### Reference:

[https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_testing\\_tools\\_start\\_stop\\_test.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_testing_tools_start_stop_test.htm)

Q.27. As part of a custom development, a developer creates a Lightning Component to show how a particular opportunity progresses over time. The component must display the date stamp when any of the following fields change:

- Amount, Probability, Stage, or Close Date

What is the most efficient way to query such information?

Answered Right

Ask Instructor

1.

[Select Amount, CloseDate, StageName, Probability FROM Opportunity\_History WHERE OpportunityId = :oppId];

**2.**

[Select NewValue, OldValue FROM OpportunityFieldHistory WHERE OpportunityId = :oppId AND Field IN ('StageName', 'Probability', 'Amount', 'CloseDate' )];

**3.**

[Select Amount, CloseDate, StageName, Probability FROM OpportunityHistory WHERE OpportunityId = :oppId];  
**(Correct Answer)(Selected Answer)**

**4.**

[Select NewValue, OldValue FROM OpportunityField\_History WHERE OpportunityId = :oppId AND Field IN ('StageName', 'Probability', 'Amount', 'CloseDate' )];

This question is related to the concept of "Field History Tracking" in Salesforce. In Salesforce, we can select certain fields to track and display the field history in the History related list of an object.

Salesforce stores an object's tracked field history in an associated object called StandardObjectNameHistory or CustomObjectName\_\_History. For example, OpportunityHistory represents the history of changes to the values of an Opportunity record's fields. Similarly, MyCustomObject\_\_History tracks field history for the MyCustomObject\_\_c custom object.

### OpportunityHistory

Represents the stage history of an Opportunity. This object represents the history of a change to the Amount, Probability, Stage, or Close Date fields of an Opportunity. The OpportunityFieldHistory object represents the history of a change to any of the fields of an Opportunity.

**As per Salesforce Documentation**, to obtain information about how a particular opportunity is progressing, query the OpportunityHistory records associated with a given Opportunity. Please note that if an opportunity's **Amount, Probability, Stage, or Close Date** fields have not changed, nothing will be returned in the OpportunityHistory objects.

#### For Example, Let's say:

- We created an Opportunity named "AC INSTALLATION". At this time, we kept Amount as blank and Stage = "Prospecting"
- Then we updated the Stage as from "Prospecting" to "Qualification".
- Then we updated the Amount from blank to "1000000"

Now, let's Query to OpportunityHistory Object.

#### Screenshot – Query to OpportunityHistory



| Query Results - Total Rows: 3 |            |               |             |
|-------------------------------|------------|---------------|-------------|
| Amount                        | CloseDate  | StageName     | Probability |
| 100000                        | 2022-02-28 | Qualification | 10          |
|                               | 2022-02-28 | Qualification | 10          |
|                               | 2022-02-28 | Prospecting   | 10          |

With this explanation, the correct answer is:

**[Select Amount, CloseDate, StageName, Probability FROM OpportunityHistory WHERE OpportunityId = :oppId];**

#### Reference:

[https://developer.salesforce.com/docs/atlas.en-us.api.meta/api\\_sforce\\_objects\\_opportunityhistory.htm](https://developer.salesforce.com/docs/atlas.en-us.api.meta/api_sforce_objects_opportunityhistory.htm)

Q.28. An Apex trigger and Apex class increment a counter, Edit\_Count\_\_c, any time that the Case is changed.

```
public class CaseTriggerHandler {
 public static void handle(List<Case> cases) {
 for (Case c : cases) {
 c.Edit_Count__c = c.Edit_Count__c + 1;
 }
 }

 trigger on Case(before update) {
 CaseTriggerHandler.handle(Trigger.new);
 }
}
```

A new process on the Case object was just created in production for when a Case is created or updated. Since the process was created, there are reports that the Edit Count is being incremented by more than one on Case edits. Which change in the Apex code will fix the problem?

**Answered Wrong****Ask Instructor**

```
trigger on Case(before update) {
 Boolean firstRun = true;
 If (firstRun) {
 1. CaseTriggerHandler.handle(Trigger.newMap);
 }
 firstRun = false;
}
```

**(Correct Answer)**

```
trigger on Case(before update) {
 if (CaseTriggerHandler.firstRun) {
 CaseTriggerHandler.handle(Trigger.new);
 }
 CaseTriggerHandler.firstRun = false;
}

trigger on Case(before update) {
 CaseTriggerHandler.firstRun = true;
 If (CaseTriggerHandler.firstRun.firstRun) {
 3. CaseTriggerHandler.handle(Trigger.newMap);
 }
 CaseTriggerHandler.firstRun = false;
}
```

**public class CaseTriggerHandler {**

```
 public static Boolean firstRun = true;
 public static void handle(List<Case> cases) {
 for (Case c : cases) {
 4. c.Edit_Count_c = c.Edit Count + 1;
 }
 }
}
```

**(Selected Answer)**

This question is related to the concept of Recursive Trigger in Salesforce. The scenario given in this question is:

- There is a trigger class to increment a counter "Edit\_Count\_\_c", any time that the Case is changed.
- After a new Process is created, it seems Edit Count is being incremented by more than one on Case edits.

### **Handle Recursion in Trigger**

To avoid the recursion on a trigger, make sure your trigger is getting executed only one time. You can use the Static Boolean variable in the apex class and check the variable in Apex Trigger IF it is true then execute your logic and make it false so that trigger cannot execute Again.

#### **Screenshot – Apex Class having Boolean Variable**

```
TriggerController.apxc
Code Coverage: None API Version: 53
1. public class TriggerController {
2.
3. // Boolean variable to stop recursion
4. public static Boolean isFirstCall = true;
5. }
```

Now, we can use this Boolean variable in the Trigger before calling the apex method and then update the Boolean variable value to false.

With this explanation, let's find the correct answer from the options given.

Below Option is CORRECT because it added the static Boolean variable to apex class.

**Create Notes**

```
trigger on Case(before update) {
 Boolean firstRun = true;
 If (firstRun) {
 CaseTriggerHandler.handle(Trigger.newMap);
 }
 firstRun = false;
}
```

The rest of the answers are incorrect.

**Reference:**

<https://help.salesforce.com/s/articleView?id=000332407&type=1>

Q.29. Which three Visualforce components can be used to initiate Ajax behavior to perform partial page updates?  
Choose 3 answers.

Answered Wrong

Ask Instructor

1. <apex:actionSupport>  
(Correct Answer)(Selected Answer)
2. <apex:commandButton>  
(Correct Answer)(Selected Answer)
3. <apex:actionStatus>  
(Selected Answer)
4. <apex:commandLink>  
(Correct Answer)
5. <apex:form>

This question is related to implementing AJAX behavior to perform partial page updates. Let's understand few concepts of Visualforce Page.

**Implementing Partial Page Updates with Command Links and Buttons**

One of the most widely used Ajax behaviors is a partial page update, in which only a specific portion of a page is updated following some user action, rather than a reload of the entire page.

The simplest way to implement a partial page update is to use the **reRender** attribute on an **<apex:commandLink>** or **<apex:commandButton>** tag to identify a component that should be refreshed. When a user clicks the button or link, only the identified component and all of its child components are refreshed.

**Screenshot – VF Page to perform partial page updates**

```
DemoVFPage.vfp
Preview API Version: 54
1 <apex:page standardController="Account">
2 <apex:pageBlock title="Hello {!$User.FirstName}!">
3 You can see Contacts from {!account.name} account.
4 Click a contact's name to view his or her details.
5 </apex:pageBlock>
6 <apex:pageBlock title="Contacts">
7 <apex:form>
8 <apex:dataTable value="{!!account.Contacts}" var="contact" cellPadding="4"
9 border="1">
10 <apex:column>
11 <apex:commandLink rerender="detail">
12 {!contact.Name}
13 <apex:param name="cid" value="{!!contact.id}"/>
14 </apex:commandLink>
15 </apex:column>
16 </apex:dataTable>
17 </apex:form>
18 </apex:pageBlock>
19 <apex:outputPanel id="detail">
20 <apex:detail subject="{!!$CurrentPage.parameters.cid}" relatedList="false"
21 title="false"/>
22 </apex:outputPanel>
23 </apex:page>
```

**Screenshot – Invoking the above VF Page for Account "AB Textile"**

Create Notes

The screenshot shows a Salesforce Lightning page titled 'Hello Test User!'. At the top, it says 'You can see Contacts from AB Textile account. Click a contact's name to view his or her details.' Below this is a 'Contacts' section with two contacts listed: 'Contact One' and 'Contact Two'. A red box highlights these two names, with two red arrows pointing to a callout box containing the text: 'When we click on "Contact One" or "Contact Two" it only refreshes the below details.' Below the contact list is a 'Contact Detail' section with various fields like Name, Account Name, Title, Department, Birthdate, Reports To, Lead Source, Picture, Mailing Address, Created By, and Description. At the bottom of the detail section are buttons for Edit, Delete, Clone, and Submit for Approval.

### Implementing Partial Page Updates using `<apex:actionSupport>`

Using command links and buttons to implement a partial page update is relatively simple, but suppose you want to have the same page update occur just by hovering the mouse over a contact's name?

To do this with the contact list example, remove the `<apex:commandLink>` tag from the data table and wrap the contact name in an `<apex:outputPanel>` tag instead. Within this output panel, add an `<apex:actionSupport>` element as a sibling of the contact's name.

Therefore, in such scenarios we can use `<apex:actionSupport>`

For **sample code** please refer:

[https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages\\_quick\\_start\\_ajax\\_partial\\_page\\_update\\_any\\_component.htm](https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages_quick_start_ajax_partial_page_update_any_component.htm)

With this explanation, the correct answers are:

"`<apex:actionSupport>`"  
and            "`<apex:commandButton>`" and "`<apex:commandLink>`"

### References:

[https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages\\_quick\\_start\\_ajax\\_partial\\_page\\_updates.htm](https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages_quick_start_ajax_partial_page_updates.htm)  
[https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages\\_quick\\_start\\_ajax\\_partial\\_page\\_update\\_any\\_component.htm](https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages_quick_start_ajax_partial_page_update_any_component.htm)

Q.30. A developer is tasked with creating an application-centric feature on which end-users can access and update information. This feature must be available in Lightning Experience while working seamlessly in multiple device form factors, such as desktops, phones, and tablets. Additionally, the feature must support Addressable URL Tabs and interact with the Salesforce Console APIs.

What are two approaches a developer can take to build the application and support the business requirements?  
Choose 2 answers.

Answered Wrong

Ask Instructor

1. Create the application using Lightning Experience Builder.
2. Create the application using Aura Components wrapped in Lightning Web Components.  
**(Selected Answer)**
3. Create the application using Aura Components.  
**(Correct Answer) (Selected Answer)**
4. Create the application using Lightning Web Components wrapped in Aura Components.  
**(Correct Answer)**

This is a scenario-based question related to developing Lightning Components. The scenario/requirement given in this question is:

- The implementation should work in multiple device form factors, such as desktops, phones, and tablets.
- The implementation must support Addressable URL Tabs and interact with the Salesforce Console APIs.

For second point, To enable direct navigation to a Lightning component via URL, add the lightning:isUrlAddressable interface to the component. But this is not supported in Lightning Web Components. Therefore, we can implement this in two ways:

- Either use Aura Component or
- Use Lightning Web Component wrapped inside Aura Component so that the parent Aura Component can be used to achieve Addressable URLfunctionlaity.

With this explanation, the correct answers are:

**"Create the application using Aura Components."**

and

**"Create the application using Lightning Web Components wrapped in Aura Components."**

For Reference & Code:

<https://developer.salesforce.com/docs/component-library/bundle/lightning:isUrlAddressable/documentation>

Q.31. Part of a custom Lightning Component displays the total number of Opportunities in the org, which is in the millions. The Lightning Component uses an Apex Controller to get the data it needs.

What is the optimal way for a developer to get the total number of Opportunities for the Lightning Component?

Answered Right

Ask Instructor

- Apex Batch job that counts the number of Opportunity records.
- COUNT() SOQL aggregate query on the Opportunity object.  
(Correct Answer)(Selected Answer)
- SUM() SOQL aggregate query on the Opportunity object.
- SOQL for loop that counts the number of Opportunities records.

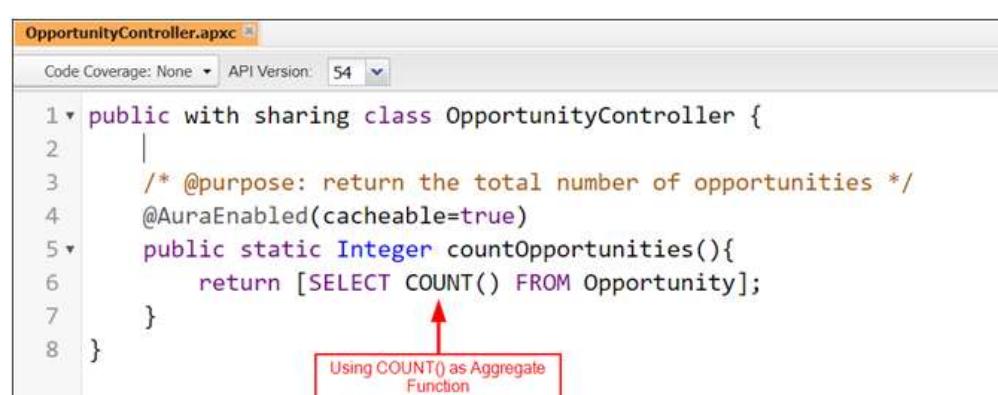
This is a scenario-based question related to Lightning Component. The scenario given in this question is:

- Lightning Component should display the total number of Opportunities in the org.
- The Lightning Component uses an Apex Controller to get the data it needs.

To discover the number of rows that a query returns, use the **aggregate function COUNT()** in a SELECT statement of a SOQL query.

#### Let's do some hands-on

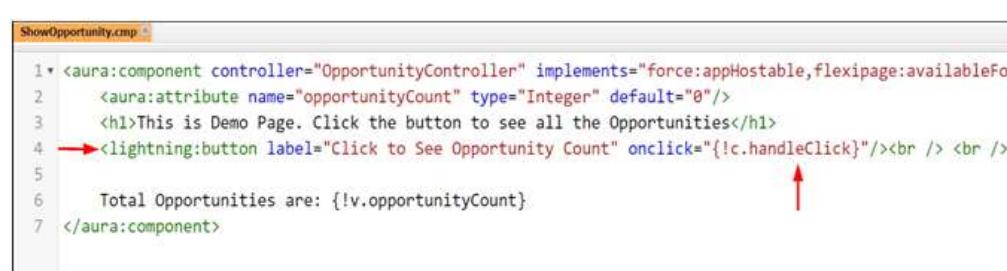
**Screenshot – Apex Controller & Method to return Opportunity Count**



```
OpportunityController.apxc
Code Coverage: None API Version: 54
1 * public with sharing class OpportunityController {
2 |
3 /* @purpose: return the total number of opportunities */
4 @AuraEnabled(cacheable=true)
5 public static Integer countOpportunities(){
6 return [SELECT COUNT() FROM Opportunity];
7 }
8 }
```

Using COUNT() as Aggregate Function

**Screenshot – Aura Component (.cmp file)**



```
ShowOpportunity.cmp
1 * <aura:component controller="OpportunityController" implements="force:appHostable,flexipage:availableForAllPageTypes">
2 <aura:attribute name="opportunityCount" type="Integer" default="0"/>
3 <h1>This is Demo Page. Click the button to see all the Opportunities</h1>
4 <lightning:button label="Click to See Opportunity Count" onclick="{!c.handleClick}"/>

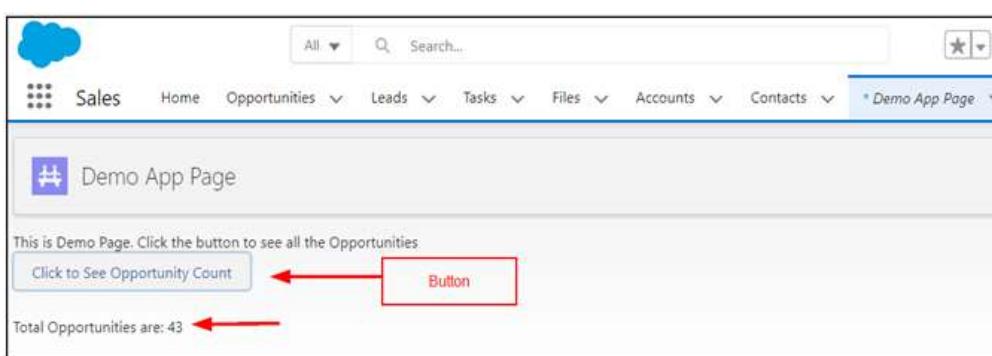
5
6 Total Opportunities are: {!v.opportunityCount}
7 </aura:component>
```

**Screenshot – Aura Component (.js file)**

```

1 * ({{
2 * handleClick : function(cmp, event, helper) {
3 * var action = cmp.get("c.countOpportunities");
4 * action.setCallback(this, function(response) {
5 * var state = response.getState();
6 * if (state === "SUCCESS") {
7 * cmp.set("v.opportunityCount", response.getReturnValue());
8 * }
9 * else if (state === "INCOMPLETE") {
10 * // do something
11 * }
12 * else if (state === "ERROR") {
13 * var errors = response.getError();
14 * if (errors) {
15 * if (errors[0] && errors[0].message) {
16 * console.log("Error message: " +
17 * errors[0].message);
18 * }
19 * }
20 * }
21 * });
22 * $A.enqueueAction(action);
23 * }
24 * });
25 *

```

**Screenshot – Adding Aura Component to App Page**

With this explanation, the correct answer is **"COUNT () SOQL aggregate query on the Opportunity object."**

**Reference:**

[https://developer.salesforce.com/docs/atlas.en-us.soql\\_sosl.meta/soql\\_sosl/sforce\\_api\\_calls\\_soql\\_select\\_count.htm](https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/sforce_api_calls_soql_select_count.htm)

Q.32. What are three benefits of using static resources in Visualforce and Aura components? Choose 3 answers.

Answered Right

Ask Instructor

1. Relative paths can be used in files in static resource archives to refer to other content within the archive.  
(Correct Answer)(Selected Answer)
2. Static resource files are automatically minified.
3. Static resource files can be referenced by using the \$Resource global variable instead of hardcoded.  
(Correct Answer)(Selected Answer)
4. Static resource files do not count against an organization's quota of data storage.
5. Static resource files can be packaged into a collection of related files in a zip or jar archive.  
(Correct Answer)(Selected Answer)

This question is related to the usage of Static Resource in Visualforce and Aura Components.

Static resources allow you to upload content that you can reference in a Visualforce page, including archives (such as .zip and .jar files), images, style sheets, JavaScript, and other files. Static resources can be used only within your Salesforce org, so you can't host content here for other apps or websites.

As per Salesforce documentation, the correct answers of this question are:

Option "**Relative paths can be used in files in static resource archives to refer to other content within the archive.**" is CORRECT.

Option "**Static resource files can be referenced by using the \$Resource global variable instead of hardcoded.**" is CORRECT.

Option "**Static resource files can be packaged into a collection of related files in a zip or jar archive.**" is CORRECT.

**For Reference & Detail:**

[https://help.salesforce.com/s/articleView?id=sf.pages\\_static\\_resources.htm&type=5](https://help.salesforce.com/s/articleView?id=sf.pages_static_resources.htm&type=5)

Q.33. A developer sees test failures in the sandbox but not in production. No code or metadata changes have been actively made to either environment since the sandbox was created.

Which consideration should be checked to resolve the issue?

Answered Right

Ask Instructor

1. Ensure the Apex Classes are on the same API version.
2. Ensure the sandbox is on the same release as production. (Correct Answer) (Selected Answer)
3. Ensure Process Builder processes are inactive.
4. Ensure Workflow Rules are inactive.

This question is related to Apex Test Class in Sandbox and Production. The scenario given in this question is:

- A developer sees test failures in the sandbox but not in production.
- No code or metadata changes have been actively made to either environment since the sandbox was created.

**The requirement is:**

- Which consideration should be checked to resolve the issue?

If we consider Point-2, it says No code or metadata changes have been actively made.

Therefore, the below options are incorrect.

Option "**Ensure Workflow Rules are inactive**" is not correct because Apex Test should cover the scenarios which is mostly related to Apex Code and also it is said that No code or metadata changes have been actively made.

Option "**Ensure Process Builder processes are inactive**" is not correct because Apex Test should cover the scenarios which is mostly related to Apex Code and also it is said that No code or metadata changes have been actively made.

Option "**Ensure the Apex Classes are on the same API version**" is not correct because it does not impact.

The correct answer is:

**"Ensure the sandbox is on the same release as production"** because once a new release come, the best approach is to refresh the sandbox and test existing customization against release features. So, it might be the reason that the Sandbox is refreshed and having the new release.

**Reference:**

<https://www.infallibletechie.com/2018/10/things-to-do-before-and-after.html>

Q.34. A developer has a Visualforce page that automatically assigns ownership of an Account to a queue upon save. The page appears to correctly assign ownership, but an assertion validating the correct ownership fails.

What can cause this problem?

Answered Wrong

Ask Instructor

1. The test class does not use the seeAllData=true annotation. (Selected Answer)
2. The test class does not retrieve the updated value from the database. (Correct Answer)
3. The test class does not implement the Queueable interface.
4. The test class does not use the Bulk API for loading test data.

This is a scenario-based question related to the concept of Apex Test Class in Salesforce. The scenario given in this question is:

- Visualforce page that automatically assigns ownership of an Account to a queue upon save.
- The page appears to correctly assign ownership, but an assertion validating the correct ownership fails.
- o This means the Apex Test class written is getting failed and the reason is the expected and the actual value for owner is different.

Let's take a simple example & reproduce the same scenario with Case Object.

- There is an Apex Class & method to update the specific Case record to Queue.
- There is an Apex Test Class to cover this scenario.
- At LINE-20 of Apex Test Class "CaseExTest", we expect the Case Owner to be updated to Queue.

#### Screenshot – Apex Class having method "updateCaseOwner"

```

1 * public class CaseEx {
2
3 /* @purpose: update the specific case owner to queue named "US_Queue" */
4 public static void updateCaseOwner(Id caseId){
5 List<Case> listCases = [SELECT Id, OwnerId FROM Case WHERE Id = :caseId];
6 Id QueueId = [SELECT Id FROM Group WHERE Type = 'Queue' AND Name = 'US_Queue'].Id;
7 if(!listCases.isEmpty()){
8 listCases[0].OwnerId = QueueId;
9 update listCases;
10 }
11 }
12 }

```

#### Screenshot – Apex Test Class

```

1 @isTest
2 public class CaseExTest {
3 @isTest static void updateCaseOwnerTest(){
4 //create test data for Case Record
5 Case cases = new Case(Status='New');
6 insert cases;
7 //create test data for Queue
8 Group us_queue = new Group(Name='US_Queue', Type='Queue');
9 insert us_queue;
10 //associate the Queue to Case
11 System.runAs(new User(Id=UserInfo.getUserId())){
12 QueuesObject addCaseToQueue = new QueueSObject(QueueID = us_queue.Id, SObjectType = 'Case');
13 insert addCaseToQueue;
14 }
15 Test.start();
16 //calling the actual apex method to update the Case Owner
17 CaseEx.updateCaseOwner(cases.Id);
18 Test.stopTest();
19 //after update, we expect the Case Owner to be Queue 'US_Queue'
20 System.assertEquals(us_queue.Id, cases.OwnerId); ← At this point, we expect the Case Owner to be Queue "US_Queue" but it fails here
21 }
22 }

```

#### Screenshot – Test Class failed with below Error



**Please Note:** If you see LINE-20 of the above Apex Test Class, it does not retrieve the updated case record. Therefore, the resolution of this issue is we need to fetch the latest version of Apex Test Class and assert it.

#### Screenshot – Updated Test Class

Create Notes

```

1 @isTest
2 * public class CaseExTest {
3 @isTest static void updateCaseOwnerTest(){
4 //create test data for Case Record
5 Case cases = new Case(Status='New');
6 insert cases;
7 //create test data for Queue
8 Group us_queue = new Group(Name='US_Queue', Type='Queue');
9 insert us_queue;
10 //associate the Queue to Case
11 System.runAs(new User(Id=UserInfo.getUserId())){
12 QueuesObject addCaseToQueue = new QueuesObject(QueueID = us_queue.Id, SObjectType = 'Case');
13 insert addCaseToQueue;
14 }
15 Test.startTest();
16 //calling the actual apex method to update the Case Owner
17 CaseEx.updateCaseOwner(cases.Id);
18 Test.stopTest();
19 //after update, we expect the Case Owner to be Queue 'US_Queue'
20 Case updatedCaseRecord = [SELECT Id, OwnerId FROM Case WHERE Id = :cases.Id];
21 System.assertEquals(us_queue.Id, updatedCaseRecord.OwnerId);
22 }
23 }
```

With this explanation, the correct answer is:

**"The test class does not retrieve the updated value from the database."**

Q.35. Consider the following queries. For these queries, assume that there are more than 200,000 Account records. These records include soft-deleted records: that is, deleted records that are still in the Recycle Bin. Note that there are two fields that are marked as External Id on the Account. These fields are Customer\_Number\_\_c and ERP\_Key\_\_c

Which two queries are optimized for large data volumes?

Choose 2 answers.

Answered Wrong

Ask Instructor

1. SELECT Id FROM Account WHERE Name != NULL
2. SELECT Id FROM Account WHERE Id IN :aListVariable  
**(Correct Answer)**
3. SELECT Id FROM Account WHERE Name != '' AND IsDeleted = false  
**(Selected Answer)**
4. SELECT Id FROM Account WHERE Name != '' AND Customer\_Number\_\_c = 'ValueA'  
**(Correct Answer)(Selected Answer)**

Create Notes

This question is related to the concept of Selective Queries in Salesforce. First of all, let's learn about few concepts.

### Index Fields in Salesforce

In the modern cloud application with a large data set like Salesforce, indexing is very important to allow an application to search and return results to users at a minimum time and cost.

There are 2 types of indexes in Salesforce:

- Standard Index
- Custom Index

Salesforce maintains indexes on the following fields for most objects:

- Salesforce record ID
- RecordTypeID
- Division
- CreatedDate
- SystemModstamp (LastModifiedDate)
- Name
- Email (for contacts and leads)
- Lookups and master-detail relationship

Salesforce administrator can define a **custom index field** by making **by making the field as "External ID"** or unique field for the following field type:

- Number
- Auto-number
- Text
- Email

### Selective SOQL Queries:

SOQL queries must be selective, particularly for queries inside triggers for the best performance. To avoid long execution times, non-selective SOQL queries may be terminated by the system.

A query is selective when one of the query filters is on an indexed field and the query filter reduces the resulting number of rows below a system-defined threshold.

Now, coming back to the question, the scenario given is

- Assume that there are more than 200,000 Account records.
- These records include soft-deleted records: that is, deleted records that are still in the Recycle Bin.
- there are two fields that are marked as External Id on the Account.
  - Customer\_Number\_\_c
  - ERP\_Key\_\_c

### The ask is:

- Which two queries are optimized for large data volumes?

With this explanation, the correct answers are:

Below Option is correct because WHERE clause is on an indexed field (Id).

```
SELECT Id FROM Account WHERE Id IN :aListVariable
```

Below Option is correct because the field Customer\_Number\_\_c is selective.

```
SELECT Id FROM Account WHERE Name != '' AND Customer_Number__c = 'ValueA'
```

### Reference & Details:

<https://help.salesforce.com/s/articleView?id=000325247&type=1>

Q.36. A company has a custom object Sales\_Help\_Request\_\_c that has a Lookup relationship to Opportunity. The Sales\_Help\_Request\_\_c has a number field, Number\_of\_Hours\_\_c, that represents the amount of time spent on the Sales\_Help\_Request\_\_c.

A developer is tasked with creating a field, Total\_Hours\_\_c, on Opportunity that should be the sum of all of the Number\_of\_Hours\_\_c values for the Sales\_Help\_Request\_\_c records related to that Opportunity.

What should the developer use to implement this?

Answered Right

Ask Instructor

1. A trigger on the Opportunity object.
2. A trigger on Sales\_Help\_Request\_\_c.  
(Correct Answer)(Selected Answer)
3. A workflow rule on the Sales\_Help\_Request\_\_c object.
4. A roll-up summary field on the Opportunity object.

This is a scenario-based question related to the concept of tracking fields on records. The scenario given in this question is:

- There are two objects. Standard object "Opportunity" and custom object "Sales\_Help\_Request\_\_c".
- Custom object "Sales\_Help\_Request\_\_c" has **lookup relationship** to Standard object "Opportunity"
- There is a field "Number\_of\_Hours\_\_c" on Custom object "Sales\_Help\_Request\_\_c" to represents the amount of time spent on the Sales\_Help\_Request\_\_c.
- There is a field "Total\_Hours\_\_c" on Standard object "Opportunity" that should be the sum of all of the "Number\_of\_Hours\_\_c" values for the Sales\_Help\_Request\_\_c records related to that Opportunity.

**The requirement is:**

- How to populate "Total\_Hours\_\_c" field on Opportunity that should be the sum of all of the "Number\_of\_Hours\_\_c" values for the Sales\_Help\_Request\_\_c records related to that Opportunity.

This requirement is very simple to implement by using "Roll-up Summary Field" if the relationship between two objects is "Master-Detail" relationship.

A roll-up summary field calculates values from related records, such as those in a related list. You can create a roll-up summary field to display a value in a master record based on the values of fields in a detail record.

**But, here we can not use roll-up summary field because the relationship between Opportunity and Sales\_Help\_Request\_\_c is Lookup.**

There are 4 ways one can work around this:

- An AppExchange App
- A Flow
- Apex Triggers
- Reporting

With this explanation, the correct answer is "**A trigger on**

**Sales\_Help\_Request\_\_c**" because when Number\_of\_Hours\_\_c field of any Sales\_Help\_Request\_\_c record is changed, then for the corresponding Opportunity record, the logic need to add all the Number\_of\_Hours\_\_c field values and update "Total\_Hours\_\_c" field on Opportunity record.

**Reference:**

<https://www.salesforceben.com/4-ways-to-create-roll-up-summary-fields-on-lookup-relationships-in-salesforce/>

Q.37. A company has a custom component that allows users to search for records of a certain object type by invoking an Apex Controller that returns a list of results based on the user's input.

When the search is completed, a searchComplete event is fired, with the results put in a results attribute of the event. The component is designed to be used within other components and may appear on a single page more than once.

What is the optimal code that should be added to fire the event when the search has completed?

Answered Wrong

Ask Instructor

```
var evt = $A.get("e.c.searchComplete");
1. evt.setParams({results: results});
 evt.fire();
(Selected Answer)

var evt = component.getEvent("searchComplete");
2. evt.set("v.results", results);
 evt.fire();

var evt = $A.get("e.c.searchComplete");
3. evt.set("v.results", results);
 evt.fire();

var evt = component.getEvent("searchComplete");
4. evt.setParams({results: results});
 evt.fire();
(Correct Answer)
```

This question is related to Component Event and specifically about how to fire an event in Aura Component. The scenario given in this question is:

- When the search is completed, a searchComplete event is fired
- While firing the event, the results put in a results attribute of the event.

The ask is:

- What is the optimal code that should be added to fire the event when the search has been completed?

As per Salesforce documentation, a Component event to communicate data to another component. A component event can be handled by the component that fired the event or by a component in the containment hierarchy that receives the event.

To fire a component event, there are two steps:

### 1. Register an event

A component registers that it may fire an event by using `<aura:registerEvent>` in its markup.

For example:

```
1 <aura:registerEvent name="sampleComponentEvent" type="c:compEvent"/>
```

### 2. Fire an event

To get a reference to a component event in JavaScript, use `cmp.getEvent("evtName")` where evtName matches the name attribute in `<aura:registerEvent>`.

Use `fire()` to fire the event from an instance of a component. For example, in an action function in a client-side controller:

```
1 var compEvent = cmp.getEvent("sampleComponentEvent");
2 // Optional: set some data for the event (also known as event shape)
3 // A parameter's name must match the name attribute
4 // of one of the event's <aura:attribute> tags
5 // compEvent.setParams({ "myParam": myValue });
6 compEvent.fire();
```

**Set Parameter (optional)**

With this explanation, coming back to the options given:

Below is the correct answer because it uses `getEvent()` to get the reference of this component& also it uses the correct syntax to pass parameters to the event.

```
var evt = component.getEvent("searchComplete");
evt.setParams({ results: results });
evt.fire();
```

#### Reference:

[https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/events\\_component\\_fire.htm](https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/events_component_fire.htm)

Create Notes

Q.38. A corporation has many different Salesforce orgs, with some different objects and some common objects, and wants to build a single Java application that can create, retrieve, and update common object records in all of the different orgs.

Which method of integration should the application use?

Answered Wrong

Ask Instructor

1. Metadata API.  
(Selected Answer)

2. SOAP API with the Partner WSDL.  
(Correct Answer)

3. Apex REST Web Service.

4. SOAP API with the Enterprise WSDL.

This is a scenario-based question.

The scenario given in this question is:

- There are many different Salesforce orgs, with some different objects and some common objects, and wants to build a single Java application that can create, retrieve, and update common object records in all of the different orgs.

**The ask is:**

- Which method of integration should the application use?

As per the Salesforce documentation, the SOAP API provides below two WSDLs:

### 1. Enterprise Web Services WSDL

- Used by enterprise developers to build client applications for a single Salesforce organization.

### 2. Partner Web Services WSDL

- Used for client applications that are metadata-driven and dynamic in nature.
- Useful to Salesforce partners who are building client applications for multiple organizations.

As per the scenario given in this question, there are many different Salesforce orgs therefore **Partner Web Services WSDL** can be used.

**Screenshot – Using the Partner WSDL**

## Using the Partner WSDL

The API provides two WSDLs to choose from:

- **Enterprise Web Services WSDL**—Used by enterprise developers to build client applications for a single Salesforce organization. The enterprise WSDL is strongly typed, which means that it contains objects and fields with specific data types, such as `int` and `string`. Customers who use the enterprise WSDL document must download and re-consume it when changes are made to the custom objects or fields in their org or when they want to use a different version of the API. To access the current WSDL for your organization, log in to your Salesforce organization and from Setup, enter API in the Quick Find box. Then, on the API page, select **Generate Enterprise WSDL**.
- **Partner Web Services WSDL**—Used for client applications that are metadata-driven and dynamic in nature. It is particularly—but not exclusively—useful to Salesforce partners who are building client applications for multiple organizations. As a loosely typed representation of the Salesforce data model that works with name-value pairs of field names and values instead of specific data types, it can be used to access data within any organization. This WSDL is most appropriate for developers of clients that can issue a query call to get information about an object before the client acts on the object. The partner WSDL document needs to be downloaded and consumed only once per version of the API. To access the current WSDL for your organization, log in to your Salesforce organization and from Setup, enter API in the Quick Find box. Then, on the API page, select **Generate Partner WSDL**.

With this explanation, the correct answer is:

**"SOAP API with the Partner WSDL."**

**Reference:**

[https://developer.salesforce.com/docs/atlas.en-us.api.meta/api\\_sforce\\_api\\_partner.htm](https://developer.salesforce.com/docs/atlas.en-us.api.meta/api_sforce_api_partner.htm)

Q.39. A company wants to incorporate a third-party web service to set the Address fields when an Account is inserted, if they have not already been set. What is the optimal way to achieve this?

**Answered Right**

**Ask Instructor**

1. Create a Process, execute a Queueable job from it, and make a callout from the Queueable job.
2. Create a Workflow Rule, educate a Queueable job from it, and make a callout from the Queueable job.
3. Create a Before Save flow, execute a Queueable job from it, and make a callout from the Queueable job.
4. Create an Apex trigger, execute a Queueable job from it, and make a callout from the Queueable job. **(Correct Answer) (Selected Answer)**

A newly added scenario and an explanation will be given earlier.

Q.40. Refer to the following code snippets:

### MyOpportunities.js

```
import { LightningElement, wire } from 'lwc';
import getOpportunities from '@salesforce/apex/OpportunityController.findMyOpportunities';

export default class MyOpportunities extends LightningElement {
 @api userId;
 @wire(getOpportunities, {oppOwner: '$userId'})
 opportunities;
}
```

### OpportunityController.cls

Create Notes

```

public with sharing class OpportunityController {
 @AuraEnabled
 public static List<Opportunity> findMyOpportunities(Id oppOwner) {
 return [
 SELECT Id, Name, Amount
 FROM Opportunity
 WHERE OwnerId = :oppOwner
 WITH SECURITY_ENFORCED
 LIMIT 10
];
 }
}

```

A developer is experiencing issues with a Lightning web component. The component must surface information about Opportunities owned by the currently logged-in user. 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?

**Answered Right**

[Ask Instructor](#)

1. Use the `Cacheable=true` attribute in the Apex method. (Correct Answer) (Selected Answer)
2. Ensure the OWD for the Opportunity object is Public.
3. Use the `Continuation=true` attribute in the Apex method.
4. Edit the code to use the `without sharing` keyword in the Apex class.

This question is related to access apex methods from Lightning Web Component. The scenario given in this question is:

- There is an apex class named "OpportunityController"
- There is an apex method named "findMyOpportunities ()" which returns List of Opportunities.
- There is a Lightning Web Component named "MyOpportunities" which calls the apex method to get List of Opportunities.

**The ask is:**

- 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?

Let's learn basics about wire service:

- To read Salesforce data, Lightning web components use a reactive wire service.
- You can `@wire` a property or a function to receive the data.
- To use `@wire` to call an Apex method, annotate the Apex method with `@AuraEnabled(cacheable=true)`.

Now, coming back to the question, the apex method "findMyOpportunities ()" is not annotated with `cacheable=true` but as per the documentation, to use `@wire` to call an apex method, annotate the Apex method with `@AuraEnabled(cacheable=true)`.

#### Screenshot – Apex Method with Cacheable = true

```

force-app > main > default > classes > OpportunityController.cls > ...
1
2 public with sharing class OpportunityController {
3
4 @AuraEnabled(cacheable=true) <-- Using cacheable = true
5 public static List<Opportunity> findMyOpportunities(Id oppOwner){
6 return [
7 SELECT Id, Name, Amount
8 FROM Opportunity
9 WHERE OwnerId =: oppOwner
10 WITH SECURITY_ENFORCED
11 LIMIT 10
12];
13 }
14 }
15

```

With this explanation, the correct answer is:

**"Use the Cacheable=true attribute in the Apex method."**

**Reference:**

[https://developer.salesforce.com/docs/component-library/documentation/en/lwc/lwc.apex\\_wire\\_method](https://developer.salesforce.com/docs/component-library/documentation/en/lwc/lwc.apex_wire_method)

[Create Notes](#)

Q.41. An Aura component has a section that displays some information about an Account and it works well on the desktop, but users have to scroll horizontally to see the description field output on their mobile devices and tablets.

```
<lightning:layout multipleRows="false">
 <lightning:layoutItem size="6">{!v.rec.Name}
 </lightning:layoutItem>

 <lightning:layoutItem size="6">{!v.rec.Description__c}
 </lightning:layoutItem>
</lightning:layout>
```

How should a developer change the component to be responsive for mobile and tablet devices?

**Answered Right**

**Ask Instructor**

**1. First option**

```
<lightning:layout multipleRows="false">
 <lightning:layoutItem smallDeviceSize="12" largeDeviceSize="6">{!v.rec.Name}
 </lightning:layoutItem>

 <lightning:layoutItem smallDeviceSize="12" largeDeviceSize="6">
 {!v.rec.Description__c}
 </lightning:layoutItem>
</lightning:layout>
```

**2.**

**Second option**

```
<lightning:layout multipleRows="true">
 <lightning:layoutItem smallDeviceSize="12" largeDeviceSize="6">{!v.rec.Name}
 </lightning:layoutItem>

 <lightning:layoutItem smallDeviceSize="12" largeDeviceSize="6">
 {!v.rec.Description__c}
 </lightning:layoutItem>
</lightning:layout>
```

**3.**

**Third Option**

```
<lightning:layout multipleRows="true">
 <lightning:layoutItem size="12" largeDeviceSize="6">{!v.rec.Name}
 </lightning:layoutItem>

 <lightning:layoutItem size="12" largeDeviceSize="6">{!v.rec.Description__c}
 </lightning:layoutItem>
</lightning:layout>
```

**(Correct Answer)(Selected Answer)**

**4.**

**Forth Option**

```
<lightning:layout multipleRows="false">
 <lightning:layoutItem size="12" largeDeviceSize="6">{!v.rec.Name}
 </lightning:layoutItem>

 <lightning:layoutItem size="12" largeDeviceSize="6">{!v.rec.Description__c}
 </lightning:layoutItem>
</lightning:layout>
```

A newly added scenario and an explanation will be given earlier.

**Create Notes**

Q.42. As part of their quoting and ordering process, a company needs to send PDFs to their document storage system's REST endpoint that supports OAuth 2.0. Each Salesforce user must be individually authenticated with the document storage system to send the PDF.

What is the optimal way for a developer to implement the authentication to the REST endpoint?

**Answered Right**

**Ask Instructor**

**1. Hierarchy Custom Setting with an OAuth token custom field.**

**2. Named Credential with Password Authentication.**

**3. Hierarchy Custom Setting with a password custom field.**

**4. Named Credential with an OAuth Authentication Provider. (Correct Answer)(Selected Answer)**

This is a scenario-based question related to the authentication while Integrating with external systems in Salesforce. The scenario given in this question is:

- There is a REST endpoint that supports OAuth 2.0
- Each Salesforce user must be individually authenticated with the system

Let's understand few concepts.

### **OAuth 2.0**

OAuth 2.0 is the industry-standard protocol (NOT specific to Salesforce) for providing authorization to web applications.

### **Named Credentials**

A named credential specifies the URL of a callout endpoint and its required authentication parameters in one definition. To simplify the setup of authenticated callouts, specify a named credential as the callout endpoint.

A named credential authentication protocol supports basic password authentication, **OAuth 2.0**, JWT, JWT Token Exchange, and AWS Signature Version 4. You can set up each named credential to use an org-wide named principal or per-user authentication. A named principal applies the same credential or authentication configuration for the entire org, while per-user authentication provides access control at the individual user level.

With this explanation, the correct answer is:

**"Named Credential with an OAuth Authentication Provider."**

For more **reference & Setup**

[https://help.salesforce.com/s/articleView?id=sf.named\\_credentials\\_about.htm&type=5](https://help.salesforce.com/s/articleView?id=sf.named_credentials_about.htm&type=5)

---

Q.43. A page throws an 'Attempt to dereference a null object' error for a Contact.

What change in the controller will fix the error?

**Answered Right**

**Ask Instructor**

1. Change the getter's signature to be static Contact.
2. Change the setter's signature to return a Contact.
3. Declare a static final Contact at the top of the controller.
4. Use a condition in the getter to return a new Contact if it is null. (**Correct Answer**) (**Selected Answer**)

A newly added scenario and an explanation will be given shortly.

---

Q.44. 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.

```
public void updateAndMakeCallout(Map<Id, Request__c> reqs, Map<Id, Request_Line__c> reqLines) {
 Savepoint sp = Database.setSavepoint();

 try {
 insert reqs.values();
 insert reqLines.values();
 HttpResponse response = CalloutUtil.makeRestCallout(reqs.keySet(),
reqLines.keySet());
 } catch (Exception e) {
 Database.rollback(sp);
 System.debug(e);
 }
}
```

The CalloutUtil.makeRestCallout falls with a 'You have uncommitted work pending. Please commit or rollback before calling out' error.

What should be done to address the problem?

**Answered Right**

**Ask Instructor**

1. Change the *CalloutUtil.makeRestCallout* to an *@future* method. (**Correct Answer**) (**Selected Answer**)
2. Remove the *Database.setSavepoint* and *Database.rollback*.
3. Change the *CalloutUtil.makeRestCallout* to an *@InvocableMethod* method.
4. Move the *CalloutUtil.makeRestCallout* method call below the *catch* block.

Create Notes

This is a scenario-based question which is related to performing DML operations and REST callout. The scenario given in this question is:

- There is a 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 CalloutUtil.makeRestCallout fails with a 'You have uncommitted work pending. Please commit or rollback before calling out' error.

**Please Note:**

Callouts are not allowed when there is an uncommitted transaction pending.

The resolution of this problem is to make the callout in a separate context.

We either need to commit the transaction, make the callout prior to any database changes or move your callout to an @future method.

With this explanation, the correct answer is:

**"Change the CalloutUtil.makeRestCallout to an @future method."**

**Reference:**

<https://help.salesforce.com/s/articleView?id=000328873&type=1>

Q.45. A developer is asked to build a solution that will automatically send an email to the customer when an Opportunity stage changes. The solution must scale to allow for 10,000 emails per day. The criteria to send the email should be evaluated after all workflow rules have fired.

What is the optimal way to accomplish this?

Answered Wrong

Ask Instructor

1. Use an Email Alert with Process Builder.  
(Correct Answer)
2. Use a Workflow Email Alert.  
(Selected Answer)
3. Use MassEmailMessage() with an Apex trigger.
4. Use SingleEmailMessage() with an Apex trigger.

This is a scenario-based question that is related to Sending Emails to customers. The scenario given in this question is:

- There is a need to automatically send an email to the customer when an Opportunity stage changes.
- The solution must scale to allow for 10,000 emails per day.
- The criteria to send the email should be evaluated after all workflow rules have been fired.

**Please Note:**

The important part here is "**The criteria to send the email should be evaluated after all workflow rules have fired**". That means this is related to Order of Execution in Salesforce.

**Order of Execution in Salesforce**

A set of rules that describe the path a record takes through all automation and the events that happen from **SAVE to COMMIT**.

Salesforce performs the following **events in the below order**.

Create Notes

1. Loads Initial record.
2. If the request came from a standard UI edit page, Salesforce runs **system validation** to check the record for page layout specific rules, field definition, Maximum field length.
3. Executes flows that make before-save update. (New Change in Winter 20)
4. Executes all **before triggers**.
5. Runs most **Custom validation**.
6. Executes **duplicate rules**.
7. **Saves** the record to the database, but doesn't commit yet.
8. Executes all **after triggers**.
9. **Assignment rules**.
10. Executes **auto-response rules**.
11. Executes **workflow rules**. 
12. If there are **workflow field updates**, updates the record again.
13. Due to Workflow field updates introduced new duplicate field values, executes duplicate rules again. If the record was updated with workflow field updates, fires **before update triggers** and **after update triggers** one more time (and only one more time), in addition to standard validations. Custom validation rules are not run again.
14. Executes processes and flow. 
15. **Escalation rules**.
16. Executes **entitlement rules**.
17. Executes **record-triggered flows** that are configured to run after the record is saved.
18. If the record contains a **roll-up summary field** or is part of a cross-object workflow, performs calculations and updates the roll-up summary field in the parent record. Parent record goes through save procedure.
19. If the parent record is updated, and a **grandparent** record contains a roll-up summary field or is part of a **cross-object workflow**, performs calculations and updates the roll-up summary field in the grandparent record. Grandparent record goes through save procedure.
20. Executes **Criteria Based Sharing evaluation**.
21. Commits all DML operations to the database.
22. Executes all after-commit logic, such as sending email.

Now coming back to the question,

it says that "The criteria to send the email should be evaluated after all workflow rules have fired". If we see the Order of Execution, the Apex Trigger and workflow rule fire before Process.

With this explanation, the correct answer is

**"Use an Email Alert with Process Builder."**

#### Reference:

[https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_triggers\\_order\\_of\\_execution.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_triggers_order_of_execution.htm)

Q.46. A Visualforce page contains an industry select list and displays a table of Accounts that have a matching value in their Industry field.

```
<apex:selectList value="{!!selectedIndustry}">
 <apex:selectOptions values="{!!industries}" />
</apex:selectList>
```

When a user changes the value in the industry select list, the table of Accounts should be automatically updated to show the Accounts associated with the selected industry.

What is the optimal way to implement this?

Answered Wrong

Ask Instructor

1. Add an **<apex:actionFunction>** within the **<apex:selectOptions>** **(Selected Answer)**
2. Add an **<apex:actionSupport>** within the **<apex:selectList>** **(Correct Answer)**
3. Add an **<apex:actionSupport>** within the **<apex:selectOptions>**
4. Add an **<apex:actionFunction>** within the **<apex:selectList>**

This is a scenario-based question which is related to implementing the partial page updates using `<apex:actionSupport>`.

The scenario given in this question is:

- In Visualforce Page, there is an Industry select list.
- When a user changes the value in the industry select list, the table of Accounts should be automatically updated based on the selected industry

To understand this, let's write some code. We have created an Apex Class and an Visualforce Page.

#### Screenshot – Apex Class (controller)

```

1 * public class AccountVFController {
2
3 public String selectedIndustry {get; set;} ← This variable holds the latest value of Industry selected from the UI
4 public Integer totalAccountByIndustry {get; set;}
5
6 //This method is used to show picklist values from Account Industry field
7 public List<SelectOption> getIndustryList(){
8 List<SelectOption> options = new List<SelectOption>();
9 List<Schema.PicklistEntry> fieldResult = Account.Industry.getDescribe().getPicklistValues();
10 options.add(new SelectOption('', '-- select --'));
11 for(Schema.PicklistEntry f : fieldResult) {
12 options.add(new SelectOption(f.getValue(), f.getLabel()));
13 }
14 return options;
15 }
16
17 //This method is called by <apex:actionSupport>
18 public void countAccounts(){
19 totalAccountByIndustry = [SELECT COUNT() FROM Account WHERE Industry =:selectedIndustry];
20 }
21 }
22
```

#### Screenshot – Visualforce Page

```

1 <apex:page controller="AccountVFController">
2 <apex:form>
3 <apex:sectionHeader title="Account" subtitle="Select Industry"/>
4 <apex:pageblock>
5 <apex:pageblocksection>
6 <apex:pageblocksectionitem>
7 <apex:outputlabel value="Account Industry"/>
8 <apex:selectList value="{!selectedIndustry}" size="1"> ← This will be reRender everytime the industry field value is changed
9 <apex:selectOptions value="{!IndustryList}" />
10 <apex:actionSupport event="onchange" reRender="Details" action="!countAccounts"/> ← When Industry value changes, the <apex:actionSupport> calls method countAccounts() and reRender attribute is used to refresh the Panel
11 </apex:selectList>
12 </apex:pageblocksectionitem>
13 </apex:pageblocksection>
14 </apex:pageblock>
15
16 <apex:panelGroup>
17 <apex:outputPanel id="Details"> ← This will be reRender every time the <apex:actionSupport> is triggered
18 <apex:outputLabel value="The Count for Selected Industry is = "/> {!totalAccountByIndustry}
19 </apex:outputPanel>
20 </apex:panelGroup>
21 </apex:form>
22 </apex:page>

```

#### Screenshot – Invoking the Visualforce Page



From the above example,

- We have used `<apex:actionSupport>` to implement partial page update behavior.
- With `<apex:actionSupport>`, we have used "onchange" event because the requirement is to refresh the Account Detail when Industry value changes.
- The "reRender" attribute specifies which part of the page should refresh.
- The `<apex:actionSupport>` is used within `<apex:selectList>`.

With this explanation, the correct answer is:

**"Add an `<apex:actionSupport>` within the `<apex:selectList>`."**

#### References:

[https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages/pages\\_quick\\_start\\_ajax\\_partial\\_page\\_update\\_any\\_component.htm](https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages/pages_quick_start_ajax_partial_page_update_any_component.htm)  
[https://developer.salesforce.com/docs/atlas.en-us.234.0.pages.meta/pages\\_compref\\_selectList.htm](https://developer.salesforce.com/docs/atlas.en-us.234.0.pages.meta/pages_compref_selectList.htm)

Q.47. A lead developer for a Salesforce organization needs to develop a page-centric application that allows the user to interact with multiple objects related to a Contact. The application needs to implement a third-party

JavaScript framework such as Angular, and must be made available in both Classic and Lightning Experience.  
Given these requirements, what is the recommended solution to develop the application?

**Answered Wrong****Ask Instructor**

- 1. Lightning Experience Builder.**
- 2. Lightning Web Components. (Selected Answer)**
- 3. Aura Components.**
- 4. Visualforce. (Correct Answer)**

This is a scenario-based question related to the topic "User Interface Development Considerations" in Salesforce. The scenario given in this question is:

- There is a need to develop a page-centric application that allows the user to interact with multiple objects related to a Contact.
- The application needs to implement a third-party JavaScript framework such as Angular, and must be made available in both Classic and Lightning Experience.

The requirement is to choose the best / recommended solution to develop the application.

As per Salesforce Documentation, Visualforce is conceptually similar to other page-centric technologies like PHP, ASP, JSP, and Ruby on Rails. Salesforce's rich metadata infrastructure makes Visualforce a productive solution.

Visualforce pages are just HTML pages with extra tags resolved by the server. As a result, you can use an empty Visualforce page as a container for a JavaScript application. In this scenario, you don't use Visualforce tags to build your user interface. Instead, you load your JavaScript application in an empty page. Then the user interface is generated on the client-side by the JavaScript application. These applications are generally referred to as single-page applications, or SPAs, and are often built using third-party frameworks like AngularJS or React.

With this explanation, the correct answer is:

**"Visualforce"** because it can be used to develop page-centric application & can be used as a container for your third-party framework, such as AngularJS or React, and your application.

For Reference & More Detail:

[https://trailhead.salesforce.com/en/content/learn/modules/lex\\_dev\\_overview/lex\\_dev\\_overview\\_future](https://trailhead.salesforce.com/en/content/learn/modules/lex_dev_overview/lex_dev_overview_future)

Create Notes

**Q.48.** A company uses their own custom-built enterprise resource planning (ERP) system to handle order management. The company wants Sales Reps to know the status of orders so that if a customer calls to ask about their shipment, the Sales Rep can advise the customer about the order's status and tracking number if it is already shipped.

Which two methods can make this ERP order data visible in Salesforce?

Choose 2 answers.

**Answered Wrong****Ask Instructor**

- 1. Have the ERP system push data into Salesforce using the SOAP API. (Correct Answer)**
- 2. Ensure real-time order data is in Salesforce using the Streaming API. (Selected Answer)**
- 3. Use Salesforce Connect to view real-time Order data in the ERP system. (Correct Answer)(Selected Answer)**
- 4. Write a cron job in Salesforce to poll the ERP system for order updates.**

A newly added scenario and an explanation will be provided earlier.

**Q.49.** Universal Containers wants to use a Customer Community with Customer Community Plus licenses so their customers can track how many containers they are renting and when they are due back. Many of their customers are global companies with complex Account hierarchies, representing various departments within the same organization.

One of the requirements is that certain community users within the same Account hierarchy be able to see several departments' containers, based on a Junction object that relates the Contact to the various Account records that represent the departments.

Which solution solves these requirements?

**Answered Right****Ask Instructor**

1. A Custom List View on the junction object with filters that will show the proper records based on owner.
2. An Apex Trigger that creates Apex Managed Sharing records based on the junction object's relationships.
3. A Lightning web Component on the Community Home Page that uses Lightning Data Services.  
**(Correct Answer) (Selected Answer)**
4. A Visualforce page that uses a Custom Controller that specifies without sharing to expose the records.

A newly added scenario and an explanation will be added earlier.

---

Q.50. A Visualforce page needs to make a callout to get billing information and tax information from two different REST endpoints. The information needs to be displayed to the user at the same time and the return value of the billing information contains the input for the tax information callout. Each endpoint might take up to two minutes to process.  
How should a developer implement the callouts?

Answered Wrong

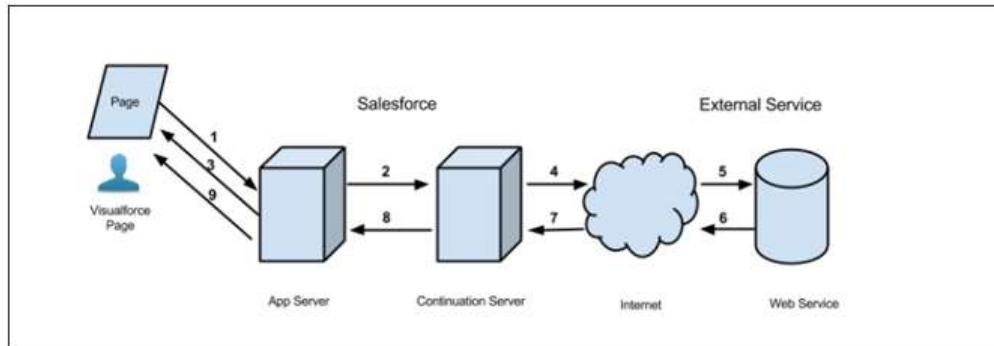
Ask Instructor

1. An HTTP REST callout for both the building callout and the tax callout.
2. A Continuation for both the building callout and the text callout.  
**(Correct Answer)**
3. A Continuation for the building callout and an HTTP REST callout for the text callout.
4. An HTTP REST callout for the billing callout and the Continuation for the tax callout. **(Selected Answer)**

This is a scenario-based question that is related to making long-running callout in Apex. Sometimes we may come across certain business scenarios where we need webservice callouts that will take longer to respond. This is because the other end needs to do a significant amount of work prior to responding back. If the other end processes a huge number of transactions during peak loads, the response will be even slower.

To resolve the above Salesforce limitation, we can use Continuation Class which makes asynchronous callout using REST and SOAP services. Using this class, we can make a long running request from a Visualforce page to external systems, and its response is returned through a callback method.

#### Screenshot – Execution Flow of Long Running Callout



Now, coming back to the question, it says:

- A Visualforce page needs to make a callout to get billing information and tax information from two different REST endpoints.
- Each endpoint might take up to two minutes to process. **(Long Running Callout)**

With this explanation, the correct answer is:

**"A Continuation for both the billing callout and the tax callout."** because both the endpoint might take up to 2 minutes to process which is considered as Long-Running Callout.

#### Reference:

[https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_continuation\\_overview.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_continuation_overview.htm)

Q.51. As part of a custom development, a developer creates a Lightning Component to show how a particular opportunity progresses over time. The component must display the date stamp when any of the following fields change:

- Amount, Probability, Stage, or Close Date

Create Notes

How should the developer access the data that must be displayed?

Answered Wrong

Ask Instructor

1. Export a SOQL query for Amount, Probability, Stage, and Close Date on the OpportunityHistory object.  
**(Selected Answer)**
2. Create custom a custom date field on Opportunity for each field to track the previous date and execute a SOQL query for date fields.  
**(Correct Answer)**
3. Subscribe to the Opportunity Change Data Capture event in the Lightning component.
4. Subscribe to the OpportunityHistory Change Data Capture event in the Lightning component.

A newly added scenario and an explanation will be given shortly.

Q.52. Universal Containers needs to integrate with their own, existing, internal custom web application. The application accepts JSON payloads, resizes product images, and sends the resized back to Salesforce. What should the developer use to implement this integration?

Answered Wrong

Ask Instructor

1. A workflow rule with an outbound message that contains a session ID.
2. An Apex Trigger that calls an `@future` method that allows callouts.  
**(Selected Answer)**
3. A platform event that makes a callout to the web application.  
**(Correct Answer)**
4. A flow that calls an `@future` method that allows callouts.

A newly added scenario and an explanation will be added earlier.

Q.53. A company accepts orders for customers in their enterprise resource planning (ERP) system that must be integrated into Salesforce as Order\_c records with a lookup field to Account. The Account object has an external ID field, ERP\_Customer\_ID\_c.

What should the integration use to create new Order\_c records that will automatically be related to the correct Account?

Answered Wrong

Ask Instructor

1. Merge on the Order\_c object and specify the `ERP_Customer_ID_c` for the Account relationship.
2. Upsert on the Order\_c object and specify the `ERP_Customer_ID_c` for the Account relationship.  
**(Selected Answer)**
3. Insert on the Order\_c object followed by an update on the Order\_c object.  
**(Correct Answer)**
4. Upsert on the Account and specify the `ERP_Customer_ID_c` for the relationship.

A newly added scenario and an explanation will be added earlier.

Q.54. An org has a requirement that the Shipping Address on the Account must be validated by a third-party web service before the Account is allowed to be inserted. This validation must happen in real-time before the account is inserted into the system. Additionally, the developer wants to prevent the consumption of unnecessary SME statements.

What is the optimal way to meet this requirement?

Answered Right

Ask Instructor

1. Make a callout to the web service from a `before insert` trigger.
2. Make a callout to the web service from a custom Visualforce controller.  
**(Correct Answer) (Selected Answer)**
3. Make a callout to the web service from an `after insert` trigger.
4. Make a callout to the web service from a standard Visualforce controller.

Create Notes

This is a scenario-based question.

The scenario given in this question is:

- An org has a requirement that the Shipping Address on the Account must be validated by a third-party web service before the Account is allowed to be inserted.
- This validation must happen in real-time before the account is inserted into the system.

With these above points, below are the observations:

- The Shipping Address validation must be Synchronous because the address must be validated in real-time before the account is inserted.

Let's go through each option and draw some conclusion:

Option "**Make a callout to the web service from a before insert trigger.**" is **not correct** because we cannot make synchronous callout from trigger.

Option "**Make a callout to the web service from an after-insert trigger.**" is **not correct** because we cannot make synchronous callout from trigger.

Option "**Make a callout to the web service from a standard Visualforce controller.**" is **not correct** because we cannot make a callout from standard Visualforce controller.

Option "**Make a callout to the web service from a custom Visualforce controller.**" is **correct** because we can use custom Visualforce controller to make synchronous callout.

---

Q.55. Refer to the code snippet below:

```
01 public void createChildRecord(String externalIdentifier){
02 Account parentAccount;
03
04 Contact newContact = new contact();
05 newContact.Account = parentAccount;
06
07 insert(newContact);
08 }
```

As part of an integration development effort, a developer is tasked to create an Apex method that solely relies on the use of foreign identifiers in order to relate new contact records to existing Accounts in Salesforce. The account object contains a field marked as an external ID, the API Name of this field is Legacy\_Id\_c.

What is the most efficient way to instantiate the parentAccount variable on line 02 to ensure the newly created contact is properly related to the Account?

Answered Wrong

Ask Instructor

1. *Account parentAccount = new Account (Legacy\_Id\_c = externalIdentifier);*  
**(Correct Answer)**

2. *Account parentAccount = [SELECT Id FROM Account WHERE Legacy\_Id\_c = :externalIdentifier];*

3. *Account parentAccount = new Account = new Account ();*  
*parentAccount.Id = externalIdentifier;*

4. *Account parentAccount = [ SELECT Id FROM Account WHERE Legacy\_Id\_c = :externalIdentifier ].Id;*  
**(Selected Answer)**

This is scenario-based question which is related to the concept of **External Id** in Salesforce.

### External Id in Salesforce

An External Id is a custom field that has the External ID attribute, meaning that it contains unique record identifiers from a system outside of Salesforce.

Now, let's understand the scenario given in this question:

- The Account object in Salesforce contains a field marked as External Id.

The API name of this field is Legacy\_Id\_\_c.

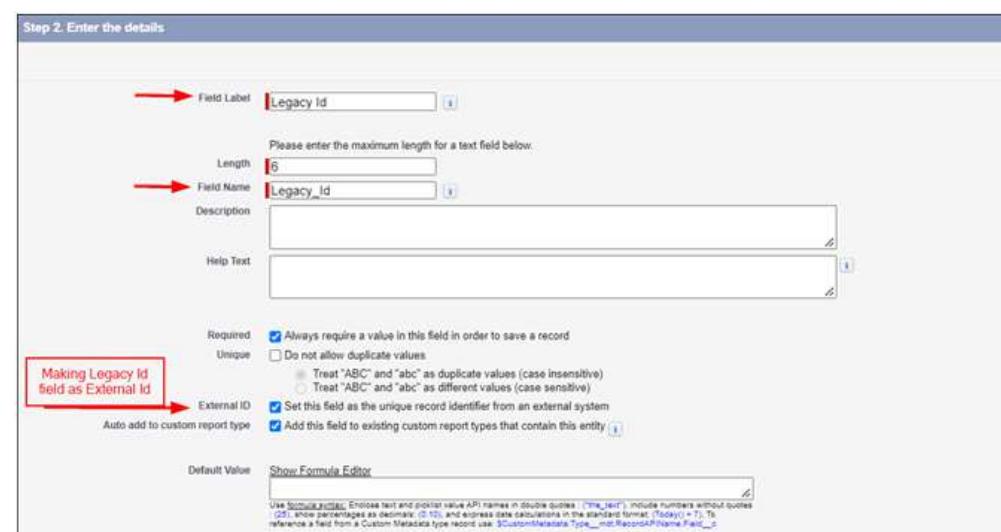
- There is a need to write Apex Method that relates new Contact records to existing Accounts in Salesforce. The sample code given in this question is:

```
01 public void createChildRecord(String externalIdentifier) {
02 Account parentAccount;
03
04 Contact newContact = new contact();
05 newContact.Account = parentAccount;
06
07 insert(newContact);
08 }
```

- The ask is:** what is the most efficient way to instantiate the parentAccount variable on line 02 to ensure the newly created contact is properly related to the Account.

Let's understand this by an example. We created a field on Account Object called "Legacy\_Id\_\_c" and wrote an Apex Method to create new Contact and associate the Contact with Account.

#### Screenshot – Created External Id field on Account Object



#### Screenshot – Created Apex Class

```
ExternalIdApex.apxc
Code Coverage: None | API Version: 54
1 * public class ExternalIdApex {
2
3 //This method will create contact and associate the Contact to Account
4 public static void createContactForAccount(String externalIdentifier){
5
6 //Instantiate new Account based on externalIdentifier
7 Account parentAccount = new Account (Legacy_Id__c = externalIdentifier);
8
9 //create new Contact
10 Contact newContact = new Contact(LastName = 'Test Contact One');
11 newContact.Account = parentAccount;
12
13 //insert new Contact
14 insert newContact;
15 }
16 }
```

With the above Apex Code, we can see that:

- The Account is instantiated based on Legacy\_Id\_\_c field which is an external identifier.

With this explanation, the correct answer is:

**"Account parentAccount = new Account (Legacy\_Id\_\_c = externalIdentifier);"**

Q.56. A developer needs to send Account records to an external system for backup purposes. The process must take a snapshot of Accounts as they are saved and then make a callout to a REST web service. The web service can only receive, at most, one record per call.

Which feature should be used to implement these requirements?

Answered Wrong

Ask Instructor

1. **@future.**
2. Process Builder.  
**(Correct Answer)**
3. Queueable.  
**(Selected Answer)**
4. Workflow.

This is a scenario-based question.

The scenario given in this question is:

- A developer needs to send Account records to an external system for backup purposes.
- Once the Account is saved, the process must take a snapshot of Accounts and then make a callout to a REST web service.

**The ask is:**

- Which feature should be used to implement these requirements?

The important point to consider here is:

- Callout should be made once an Account is saved.

To achieve this, we should use Process Builder (on record creation) which can call invokable apex method to make a callout to external system.

Therefore, with this explanation, the correct answer is:

**"Process Builder."**

**References:**

[https://help.salesforce.com/s/articleView?id=process\\_overview.htm&type=5&language=en\\_US](https://help.salesforce.com/s/articleView?id=process_overview.htm&type=5&language=en_US)  
[https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_classes\\_annotation\\_InvocableMethod.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_classes_annotation_InvocableMethod.htm)

Create Notes

Q.57. A company represents their customers as Accounts in Salesforce. All customers have a unique customer\_number\_c that is unique across all of the company's systems. They also have a custom Invoice\_c object, with a Lookup to Account, to represent invoices that are sent out from their external system.

This company wants to integrate invoice data back into Salesforce so Sales Reps can see when a customer is paying their bills on time.

What is the optimal way to implement this?

Answered Wrong

Ask Instructor

1. Query the Account Object upon each call to insert invoice data to fetch the Salesforce ID corresponding to the Customer Number on the invoice.
2. Create a cross-reference table in the custom invoicing system with the Salesforce Account ID of each Customer and insert invoice data nightly.
3. Use Salesforce Connect and external data objects to seamlessly import the invoice data into Salesforce without custom code.  
**(Selected Answer)**
4. Ensure Customer\_Number\_c is an External ID and that a custom field Invoice\_Number\_c is an External ID and Upsert invoice data nightly.  
**(Correct Answer)**

A newly added scenario and an explanation will be provided shortly.

Q.58. An Apex class does not achieve expected code coverage. The testSetup method explicitly calls a method in the Apex class.  
How can the developer generate code coverage?

Answered Right

Ask Instructor

1. Use `System.assert()` in `testSetup` to verify the values are being returned.
2. Add `@testVisible` to the method in the class the developer is testing.
3. Call the Apex class method from a `testMethod` instead of the `testSetup` method.  
**(Correct Answer)(Selected Answer)**
4. Verify the user has permissions passing a user into `System.runAs()`.

A newly added scenario and an explanation will be added shortly.

Q.59. Which use case can be performed **only** by using asynchronous Apex?

**Answered Wrong**

**Ask Instructor**

1. Scheduling a batch process to complete in the future.  
**(Selected Answer)**
2. Updating a record after the completion of an insert.
3. Calling a web service from an Apex trigger.  
**(Correct Answer)**
4. Processing high volumes of records.

This is a scenario-based question that is related to using Asynchronous Apex in Salesforce. Among the four options given, the ask is:

- "which use case can only be performed by using asynchronous Apex"?

#### Asynchronous Apex:

An asynchronous process is a process or function that executes a task "in the background" without the user having to wait for the task to finish.

Asynchronous Apex comes in a number of different flavors. We'll get into more detail for each one shortly, but here's a high-level overview.

| Type           | Overview                                                                                                     | Common Scenarios                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| Future Methods | Run in their own thread, and do not start until resources are available.                                     | Web service callout.                                                    |
| Batch Apex     | Run large jobs that would exceed normal processing limits.                                                   | Data cleansing or archiving of records.                                 |
| Queueable Apex | Similar to future methods, but provide additional job chaining and allow more complex data types to be used. | Performing sequential processing operations with external Web services. |
| Scheduled Apex | Schedule Apex to run at a specified time.                                                                    | Daily or weekly tasks.                                                  |

The following three reasons are usually behind choosing asynchronous programming.

- **Processing a very large number of records:** This reason is unique to the multi-tenanted world of the Lightning Platform where limits rule. The limits associated with asynchronous processes are higher than those with synchronous processes. Therefore, if you need to process thousands or even millions of records, asynchronous processing is your best bet.
- **Making callouts to external web services:** Callouts can take a long time to process, but in the Lightning Platform, triggers can't make callouts directly.
- **Creating a better and faster user experience:** by offloading some processing to asynchronous calls. Why does everything at once? If it can wait, let it.

With this explanation, the correct answer is:

**"Calling a web service from an Apex trigger."** because we cannot call external web services synchronously from triggers, because calling a web service synchronously from triggers will hold up the database transaction until the callout is completed. In this scenario, we can invoke callouts from triggers by encapsulating the callouts in @future methods. (Asynchronous Apex)

#### Reference:

<https://help.salesforce.com/s/articleView?id=000330691&type=1>

Q.60. Universal Containers implements a private sharing model for the Convention\_Attendee\_\_c custom object. As part of a new quality assurance effort, the company created an Event\_Reviewer\_\_c user lookup field on the object. Management wants the event reviewer to automatically gain Read/Write access to every record they are assigned to. What is the best approach to ensure the assigned reviewer obtains Read/Write access to the record?

**Answered Right**

**Ask Instructor**

1. Create a Before Insert trigger on the Convention Attendee custom object, and use Apex Sharing Reasons and Apex Managed Sharing.
2. Create an After Insert trigger on the Convention Attendee custom object, and use Apex Sharing Reasons and Apex Managed Sharing. **(Correct Answer) (Selected Answer)**
3. Create a criteria-based sharing rule on the Convention Attendee custom object to share the records with a group of Event Reviewers.
4. Create criteria-based sharing rules on the Convention Attendee custom object to share the records with the Event Reviewers.

Create Notes

This is a scenario-based question that is related to Sharing a record using Apex. The scenario given in this question is:

- There is a private sharing model for the Convention\_Attendee\_\_c custom object.
  - Please Note: Since the sharing model is private. The Non Owner should not be able to view the Convention\_Attendee\_\_c custom object record.
- There is a field Event\_Reviewer\_\_c on the custom object having lookup to the User.

### Requirement

Event Reviewer to automatically gain Read/Write access to every record they are assigned to. That means, when the record of Convention Attendee custom object is created, the associated user should also get the Read/Write access.

### Criteria-Based Sharing Rules

A criteria-based sharing rule determines with whom to share records based on field values. But in the scenario given in this question, the criteria is not universal so we can not use it. Therefore we need to use Apex.

### Sharing a Record Using Apex

To access sharing programmatically, you must use the share object associated with the standard or custom object for which you want to share. All custom object sharing objects are named as follows:

- Name of custom object = "MyCustomObject"
- Name of share object = "MyCustomObject\_\_Share"

Every share object has the following properties:

|                                |                                                                                                                                                                                             |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ParentId                       | The Id of the record being shared. This field cannot be updated.                                                                                                                            |
| UserOrGroupId                  | The Id of the User to whom you are granting access. May also be a Public Group Id, Role Id, or Territory Id. This field cannot be updated.                                                  |
| AccessLevel                    | The level of access that the specified User or Group has been granted.<br><br>Valid values for Apex managed sharing are: Edit, Read.                                                        |
|                                | This field must be set to an access level that is higher than the organization's default access level for the parent object. For more information, see <a href="#">Access Levels</a> .      |
| RowCause (aka Sharing Reasons) | The reason why the user or group is being granted access. The reason determines the type of sharing, which in turn controls who can alter the sharing record. This field cannot be updated. |

Now, coming back to the question:

- Since the custom object name is Convention\_Attendee\_\_c, the share object will be created as "Convention\_Attendee\_\_share" because the Organization-Wide Default sharing setting was set to private.
- From the above table, we know that to create a record for "Convention\_Attendee\_\_share" object, we need to assign "ParentId", which is the Id of the record being shared. (Id of Convention\_Attendee\_\_c record in this scenario)

Now, coming back to the options given,

Option "**Create an After-Insert trigger on the Convention Attendee custom object, and use Apex Sharing Reasons and Apex Managed Sharing**" is correct because we need to Set the ID of record being shared which we can only have in After Trigger Context.

Option "**Create a Before Insert trigger on the Convention Attendee custom object, and use Apex Sharing Reasons and Apex Managed Sharing**" is not correct in Before Trigger, we do not have the Id of record being shared.

For more **reference & code example**:

[https://developer.salesforce.com/docs/atlas.en-us.204.0.apexcode.meta/apexcode/apex\\_bulk\\_sharing\\_creating\\_with\\_apex.htm](https://developer.salesforce.com/docs/atlas.en-us.204.0.apexcode.meta/apexcode/apex_bulk_sharing_creating_with_apex.htm)

Q.61. A developer is debugging an Apex-based order creation process that has a requirement to have three savepoints, SP1, SP2, and SP3 (created in order), before the final execution of the process. During the final execution process, the developer has a routine to roll back to SP1 for a given condition. Once the condition is fixed, the code then calls a roll back to SP3 to continue with final execution. However, when the roll back to SP3 is called, a runtime error occurs.

Why does the developer receive a runtime error?

Answered Right

Ask Instructor

1. The developer should have called SP2 before calling SP3.
2. The developer has too many DML statements between the savepoints.

3. The developer used too many savepoints in one trigger session.
4. SP3 became invalid when SP1 was rolled back. (Correct Answer) (Selected Answer)

This is a scenario-based question related to the concept of Transaction Control in Salesforce. The scenario given in this question is:

- There is an Apex Class having three savepoints SP1, SP2 and SP3
- There three savepoints are created in sequence/order.
- For a given condition, there is a logic to roll back to SP1. The code then calls roll back to SP3 to continue with final execution.
- The problem is: When the code rolls back to SP3, the runtime error occurs.

#### Let's understand a few concepts about **Transaction Control in Apex**:

While implementing using Apex, there are certain business rules which require partial work (already executed DML statements) to be "rolled back" so that the processing can continue in another direction. Apex gives you the ability to generate a savepoint, that is, a point in the request that specifies the state of the database at that time. Any DML statement that occurs after the savepoint can be discarded, and the database can be restored to the same condition it was in at the time you generated the savepoint.

As per the Salesforce Documentation, there are the following limits while using savepoint variables and rolling back to the database.

The following limitations apply to generating savepoint variables and rolling back the database:

- If you set more than one savepoint, then roll back to a savepoint that is not the last savepoint you generated, the later savepoint variables become invalid. For example, if you generated savepoint SP1 first, savepoint SP2 after that, and then you rolled back to SP1, the variable SP2 would no longer be valid. You will receive a runtime error if you try to use it.
- References to savepoints cannot cross trigger invocations because each trigger invocation is a new trigger context. If you declare a savepoint as a static variable then try to use it across trigger contexts, you will receive a run-time error.
- Each savepoint you set counts against the governor limit for DML statements.
- Static variables are not reverted during a rollback. If you try to run the trigger again, the static variables retain the values from the first run.
- Each rollback counts against the governor limit for DML statements. You will receive a runtime error if you try to rollback the database additional times.
- The ID on an sObject inserted after setting a savepoint is not cleared after a rollback. Create an sObject to insert after a rollback. Attempting to insert the sObject using the variable created before the rollback fails because the sObject variable has an ID. Updating or upserting the sObject using the same variable also fails because the sObject is not in the database and, thus, cannot be updated.

Now, coming back to the question, the same scenario is given in the question as yellow highlighted.

- For a given condition, there is a logic to roll back to SP1. The code then calls roll back to SP3 to continue with the final execution.

With this explanation, the correct answer is:

**"SP3 became invalid when SP1 was rolled back."** because savepoints SP1, SP2, and SP3 are in sequence, and when the code rollbacks to SP1, the SP3 becomes invalid.

#### Reference:

[https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/langCon\\_apex\\_transaction\\_control.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/langCon_apex_transaction_control.htm)

Copyright ©2024 ReviewNPrep LLC. All rights reserved.

[Privacy Policy](#) [Terms & Conditions](#)

Create Notes