



PDII



Salesforce Certified Platform Developer II



EXAMKILLER

Help Pass Your Exam At First Try

Salesforce

Exam PDII

Salesforce Certified Platform Developer II (SP23)

Version: 27.0

[Total Questions: 175]

Question No : 1

A developer created a Lightning web component that allows users to input a text value that is used to search for Accounts by calling an Apex method. The Apex method returns a list of AccountWrappers and is called imperatively from a JavaScript event handler.

```

01:
02: public class AccountSearcher {
03:
04:     public static List<AccountWrapper> search(String term) {
05:         List<AccountWrapper> wrappers = getMatchingAccountWrappers(term);
06:         return wrappers;
07:     }
08:
09:
10:     public class AccountWrapper {
11:         public Account { get; set; }
12:         public Decimal matchProbability { get; set; }
13:     }
14:     // ...other methods, including getMatchingAccountWrappers implementation...
15: }

```

Which two changes should the developer make so the Apex method functions correctly?

Choose 2 answers

- A. Add @AuraEnabled to line 09.
- B. Add @AuraEnabled to line 03.
- C. Add @AuraEnabled to lines 11 and 12.
- D. Add @AuraEnabled to line 01.

Answer: B,C

Question No : 2

A developer created a JavaScript library that simplifies the development of repetitive tasks and features and uploaded the library as a static resource called jsutils in Salesforce. Another developer is coding a new Lightning web component (LWC) and wants to leverage the library.

Which statement properly loads the static resource within the LWC?

- A. import {jsUtilities} from '@salesforce/resourceUrl/jsUtils';
- B. import jsUtilities from '@salesforce/resourceUrl/jsUtils';
- C. <lightning-require scripts=N { ! SResource. jsUtils}"/>
- D. const jsUtility = SA.get ('SResource.jsUtils');

Answer: B

Question No : 3

Refer to the following code snippets:

MyOpportunities.js

```
import { LightningElement, api, 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

```
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?

- A. Ensure the OWD for the Opportunity object is Public.
- B. Use the Cacheable=true attribute in the Apex method.
- C. Edit the code to use the without sharing keyword in the Apex class.
- D. Use the continuation=true attribute in the Apex method.

Answer: B

Question No : 4

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?

- A. SUM() SOQL aggregate query on the Opportunity object
- B. SOQL for loop that counts the number of Opportunities records
- C. COUNT() SOQL aggregate query on the Opportunity object
- D. Apex Batch job that counts the number of Opportunity records

Answer: C

Question No : 5

Refer to the test method below:

```
@isTest
static void testAccountUpdate() {
    Account acct = new Account(Name = 'Test');
    acct.Integration_Updated__c = false;
    insert acct;

    CalloutUtil.sendAccountUpdate(acct.Id);

    Account acctAfter = [SELECT Id, Integration_Updated__c FROM Account WHERE Id = :acct.Id][0];
    System.assert(true, acctAfter.Integration_Updated__c);
}
```

An external system with Account information and sets the Account's Integration-Updated__c checkbox to True when it completes.

The test fails to execute and exits with an error: "Methods defined as TestMethod do not support Web service callouts."

What is the optimal way to fix this?

- A. Add if ('Test.isRunningTest I)) around calloutUtil.sendAccountUpdate.
- B. Add Test.startTest ()before and Test.stopTest () after CallcutoutUtil. sendAccoutUpdate.
- C. Add Test.startTest before and Test.setMock and Test. stoptest () after CallloutUtil.sendAccountUpdate.
- D. Add Test.startTest() and Test .setMock before and Test.stopTest () after CalloutUtil.sendAccountUpdate.

Answer: D

Question No : 6

Consider the controller code above that is called from a Lightning component and returns data wrapped in a class.

```
public class myServerSideController {
    @AuraEnabled
    public static MyDataWrapper getSomeData( String theType ) {
        Some_Object__c someObj = [
            SELECT ID, Name
            FROM Some_Object__c
            WHERE Type__c = :theType
            LIMIT 1
        ];

        Another_Object__c anotherObj = [
            SELECT ID, Option__c
            FROM Another_Object__c
            WHERE Some_Object__c = :someObj.Name
            LIMIT 1
        ];

        MyDataWrapper theData = new MyDataWrapper();

        theData.Name = someObj.Name;
        theData.Option = anotherObj.Option__c;
        return theData;
    }

    public class MyDataWrapper {
        public String Name { get; set; }
        public String Option { get; set; }

        public MyDataWrapper() {}
    }
}
```

Consider the controller code above that is called from a Lightning component and returns data wrapped in a class.

The developer verified that the Queries return a single record each and there is error handling in the Lightning component, but the component is not getting anything back when calling the controller getSomeData().

What is wrong?

- A.** The member's Name and Option of the class MyDataWrapper should be annotated with @AuraEnabled also.
- B.** The member's Name and Option should not be declared public.
- C.** The member's Name and Option should not have getter and setter.
- D.** Instances of Apex classes such as MyDataWrapper cannot be returned to a Lightning component.

Answer: A

Question No : 7

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?

- A.** Use Salesforce Connect and external data objects to seamlessly import the invoice data into Salesforce without custom code.
- B.** 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.
- C.** Create a cross-reference table in the custom invoicing system with the Salesforce Account ID of each Customer and insert invoice data nightly.
- D.** Query the Account Object upon each call to insert invoice data to fetch the Salesforce ID corresponding to the Customer Number on the invoice

Answer: B

Question No : 8

Which use case can only be performed by using asynchronous Apex?

- A.** Scheduling a batch process to complete in the future
- B.** Processing high volumes of records
- C.** Updating a record after the completion of an insert
- D.** Calling a web service from an Apex trigger

Answer: D

Question No : 9

What should a developer use to query all Account fields for the Acme account in their sandbox?

- A.** SELECT FROM ACCOUNT WHERE Name = 'Acme' LIMIT 1
- B.** SELECT ALL FROM Account WHERE Name = 'Acme' LIMIT 1
- C.** SELECT FIELDS FROM Account WHERE Name = 'Acme' LIMIT 1
- D.** SELECT FIELDS (ALL) FROM Account WHERE Name = 'Acme' LIMIT 1

Answer: D

Question No : 10

Given the following information regarding Universal Containers (UC):

- UC represents their customers as Accounts in Salesforce
- All customer has a unique Customer-Number- c, that is unique across all of UC's systems.
- UC also has a custom Invoice_ c object, with a Lookup to Account, to represent invoices that are sent out from their external system

UC wants to integrate invoice data back into Salesforce so Sales Reps can see when a customer pays their bills on time.

What is the optimal way to implement this?

A)

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.

B)

Create a cross-reference table in the custom invoicing system with the Salesforce Account ID of each Customer and insert invoice data nightly.

C)

Use Salesforce Connect and external data objects to seamlessly import the invoice data into Salesforce without custom code.

D)

Query the Account Object upon each call to insert invoice data to fetch the Salesforce ID corresponding to the Customer Number on the invoice.

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: B

Question No : 11

A developer is writing a Jest for a Lightning web component that conditionally displays child components based on a user's checkbox selections.

What should the developer do to properly test that the correct components display and hide for each scenario?

- A. Reset the DOM after each test with the after Each method.
- B. Add a teardown block to reset the DOM after each test.
- C. Create a new describe block for each test
- D. Create a new jsdom instance for each test

Answer: C

Question No : 12

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

How can this be accomplished?

- A. Use Lightning Flow to read the customer's record type, and then do a dynamic Query for recent interactions and display on the View page.
- B. Use a Lightning component to query and display interactions based on record type that is passed in using a design: attribute from the Lightning page.
- C. Use batch Apex to query for the most recent interactions when the customer view screen is loaded.
- D. Use a dynamic form on the customer record page to display recent interactions.

Answer: A

Question No : 13

The Contact object in an org is configured with workflow rules that trigger field updates.

The fields are not updating, even though the end user expects them to. The developer creates a debug log to troubleshoot the problem.

What should the developer specify in the debug log to see the values of the workflow rule conditions and debug the problem?

- A. INFO level for the Database log category
- B. ERROR level for the Workflow log category
- C. ERROR level for the Database log category
- D. INFO level for the Workflow log category

Answer: A

Question No : 14

A developer wants to write a generic Apex method that will compare the Salesforce Name field between any two object records, for example to compare the Name field of an Account and an Opportunity; or the name of an Account and a Contact.

How should the developer do this?

- A. Use a String.replace() method to parse the contents of each Name field.
- B. Invoke a Schema.describe() function to compare the values of each Name field.
- C. Cast each object into an sObject and use sObject.get(Name') to compare the Name fields.
- D. Use the salesforce metadata API to extract the value of each object and compare the name fields

Answer: C

Question No : 15

A developer has working business logic code, but sees the following error in the test class:

You have uncommitted work pending. Please commit or rollback before calling out. What is a possible solution?

- A. Rewrite the business logic and test classes with @TestVisible set on the callout.
- B. Set seeAllData to "true" at the top of the test class, since the code does not fail in practice.
- C. Call support for help with the target endpoint, as it is likely an external code error.
- D. Use test.IsRunningTest() before making the callout to bypass it in test execution.

Answer: D

Question No : 16

Universal Containers (UC) has custom Order and Order Line objects that represent orders placed by its customers.

A developer has a new requirement that UC's external enterprise resource planning (ERP) system placed by its customers. integrate with Salesforce to create orders for existing customers in Salesforce *whenever* an ERP system.

What should the developer use to create the orders in Salesforce?

- A. Connect API
- B. Change Data Capture API
- C. SObject Tree API
- D. Event Monitoring API

Answer: A

Question No : 17

As part of point-to-point integration, a developer must call an external web service which, due to high demand, takes a long time to provide a response. As part of the request, the developer must collect key inputs from the end user before making the callout.

Which two elements should the developer use to implement these business requirements?

Choose 2 answers

- A. Apex method that returns a Continuation object
- B. Screen now
- C. Process Builder
- D. Lightning web component

Answer: A,D

Question No : 18

Which two scenarios require an Apex method to be called imperatively from a Lightning

web component?

Choose 2 answers

- A. Calling a method that makes a web service callout
- B. Calling a method that is not annotated with cacheable=true
- C. Calling a method with the click of a button
- D. Calling a method that is external to the main controller for the Lightning web component

Answer: B,C

Question No : 19

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?

A)

```
Account parentAccount = [SELECT Id FROM Account WHERE Legacy_Id__c = :externalIdentifier].Id;
```

B)

```
Account parentAccount = new Account(Legacy_Id__c = externalIdentifier);
```

C)

```
Account parentAccount = [SELECT Id FROM Account WHERE Legacy_Id__c = :externalIdentifier];
```

D)

```
Account parentAccount = new Account();  
parentAccount.Id = externalIdentifier;
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: A

Question No : 20

Assuming the CreateOneAccount class creates one account and implements the Queueable interface, which syntax tests the Apex code?

A)

```
List<Account> accts;  
System.enqueueJob( new CreateOneAccount() );  
Test.getFlexQueueOrder();  
System.assertEquals( 1, accts.size() );
```

B)

```
List<Account> accts;  
Test.startTest();  
System.enqueueJob( new CreateOneAccount() );  
Test.stopTest();  
accts = [SELECT Id FROM Account];  
System.assertEquals( 1, accts.size() );
```

C)

```
List<Account> accts;  
System.enqueueJob( new CreateOneAccount() );  
Test.startTest();  
System.assertEquals(1, accts.size() );  
Test.stopTest();
```

D)

```
List<Account> accts;  
System.enqueueJob( new CreateOneAccount() );  
accts = [SELECT Id FROM Account];  
System.assertEquals(1, accts.size() );
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: B**Question No : 21**

A Salesforce org has more than 50,000 contacts. A new business process requires a calculation that aggregates data from all of these contact records. This calculation needs to run once a day after business hours.

Which two steps should a developer take to accomplish this?

Choose 2 answers

- A. Use the @future annotation on the method that performs the aggregate calculation.
- B. Implement the schedulable Interface In the class that contains the aggregate calculation method.
- C. Use the areadOnly annotation on the method that performs the aggregate calculation.
- D. Implement the Queueable Interface in the class that contains the aggregate calculation method.

Answer: B,D

Question No : 22

```

@isTest
static void testUpdateSuccess() {
    Account acct = new Account(Name = 'test');
    insert acct;

    // Add code here

    extension.inputValue = 'test';
    PageReference pageRef = extension.update();
    System.assertNotEquals(null, pageRef);
}

```

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

A)

```
AccountControllerExt extension = new AccountControllerExt(acct);
```

B)

```
AccountControllerExt extension = new AccountControllerExt(acct.Id);
```

C)

```
ApexPages.StandardController sc = new ApexPages.StandardController(acct.Id);
AccountControllerExt extension = new AccountControllerExt(sc);
```

D)

```
ApexPages.StandardController sc = new ApexPages.StandardController(acct);
AccountControllerExt extension = new AccountControllerExt(sc);
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: D

Question No : 23

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

- A. Create the application using Aura Components wrapped in Lightning Web Components.
- B. Create the application using Lightning web Components wrapped in Aura Components.
- C. Create application using Aura Components.
- D. Create the application using Lightning Experience Builder.

Answer: B,C

Question No : 24

A developer created a Lightning web component for the Account record page that displays the five most recently contacted Contacts for an Account. The Apex method, `getRecentContacts`, returns a list of Contacts and will be wired to a property in the component.

```
01:
02: public class ContactFetcher {
03:
04:     static List<Contact> getRecentContacts(Id accountId) {
05:         List<Contact> contacts = getFiveMostRecent(accountId);
06:         return contacts;
07:     }
08:
09:     private static List<Contact> getFiveMostRecent(Id accountId) {
10:         //...implementation...
11:     }
12: }
```

Which two lines must change in the above code to make the Apex method able to be wired?

Choose 2 answers

- A. Add `@AuraEnabled(cacheable=true)` to the line 08
- B. Remove `private` from line 09.
- C. Add `@AuraEnabled (cacheable=true)` to line 03.
- D. Add `public` to line 04.

Answer: C,D

Question No : 25

A developer wrote the following method to find all the test accounts in the org:

```
public static Account[] searchTestAccounts() {  
    List<List<SObject>> searchList = [ FIND 'test' IN ALL FIELDS  
                                      RETURNING Account(Name)];  
    return (Account[]) searchList[0];  
}
```

However, the test method below fails.

```
@isTest  
public static void testSearchTestAccounts() {  
    Account a = new Account(name='test');  
    insert a;  
    Account [] accounts = TestAccountFinder.searchTestAccounts();  
    System.assert(accounts.size() == 1 );  
}
```

What should be used to fix this falling test?

- A. Test. loaddata to set up expected data
- B. Test. fixedSearchResults () method to set up expected data
- C. @isTest (See AllData=true) to access org data for the test
- D. @testSetup method to set up expected data

Answer: C

Question No : 26

Just prior to a new deployment, the Salesforce Administrator who configured a new order fulfillment process in a developer sandbox suddenly left the company. The users had fully tested all of the changes in the sandbox and signed off on them.

Unfortunately, although a Change Set was started, it was not complete. A developer is brought in to help finish the deployment.

What should the developer do to identify the configuration changes that need to be moved into production?

- A. Leverage the Setup Audit Trail to review the changes made by the departed Administrator and identify which changes should be added to the Change Set.
- B. Use the Metadata API and a supported development IDE to push all of the configuration from the sandbox into production to ensure no changes are lost.
- C. Set up Continuous Integration and a Git repository to automatically merge all changes from the sandbox metadata with the production metadata.
- D. In Salesforce setup, look at the last modified date for every object to determine which should be added to the Change Set.

Answer: A

Question No : 27

A developer creates a Lightning web component to allow a Contact to be quickly entered- However, error messages are not displayed.

```
<template>
  <lightning-record-edit-form
    object-api-name="Contact">
    <lightning-input-field field-name="FirstName"></lightning-input-field>
    <lightning-input-field field-name="LastName"></lightning-input-field>
    <lightning-input-field field-name="Email"></lightning-input-field>
    <lightning-button type="submit"
      name="submit"
      label="Create Contact">
    </lightning-button>
  </lightning-record-edit-form>
</template>
```

Which component should the developer add to the form to display error messages?

- A. apex:messages
- B. aura:messages
- C. lightning-messages
- D. lightning-error

Answer: C

Question No : 28

A developer is asked to create a Lightning web component that will be invoked via button on a record page. The component must be displayed in a model dialog.

Which three steps should the developer take to achieve this?

Choose 3 answers

- A. Add a targetConfig and set targets to lightning_RecordAction.
- B. In targetConfigs, add Lightning_Apppage as a target.
- C. Set action Type to screenAction.
- D. In target, add lightning_RecordAction as a target.
- E. Set eventType to action.

Answer: A,C,D

Question No : 29

There are user complaints about slow render times of a custom data table within a Visualforce page that loads thousands of Account records at once. What can a developer do to help alleviate such issues?

- A. Upload a third-party data table library as a static resource.
- B. Use JavaScript remoting to query the accounts.
- C. Use the standard Account List controller and implement pagination.
- D. Use the transient keyword in the Apex code when querying the Account records.

Answer: C

Question No : 30

In a previous data audit, It was determined that close to 5 million Opportunity records are stored within the Salesforce environment. The organization-wide default for the object are set to Public Read-Only and most opportunities are related to an external case.

The method is called from a Lightning web component. Some end users do not provide a cased value and experience low performance while running the query.

Which two techniques should the developer implement to avoid low performance queries from executing? Choose 2 answers

- A. Implement a LIMIT clause within the SOQL query to restrict the result set.
- B. Ensure the user-provided input is not null before executing the SOQL query.
- C. Implement the with sharing keyword on the Apex class.
- D. Use SOSL instead of SOQL queries to perform text-based searches.

Answer: A,D

Question No : 31

Refer to the Lightning component below:

The Lightning Component allows users to click a button to save their changes and then redirects them to a different page. Currently when the user hits the Save button, the records are getting saved, but they are not redirected.

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

Choose 3 answers

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

Answer: A,B,D

Question No : 32

A developer is trying to access org data from within a test class. Which sObject type requires the test class to have the (seeAllData=true) annotation?

- A. RecordType
- B. Profile
- C. User
- D. Report

Answer: D

Explanation:

Test (SeeAllData=true) annotation is used to grant test classes and individual test methods access to all data in the organization, including pre-existing data that the test didn't create. This annotation is required for Apex code saved using Salesforce API version 24.0 and later.

Profile is one of the sObject types that require the isTest (SeeAllData=true) annotation to access org data from within a test class. Other sObject types that require this annotation are User, UserRole, Group, GroupMember, QueueSubject, and CronTrigger.