

Salesforce.PDII.v2026-07-20.q163

試験コード：	PDII
試験名称：	Platform Developer II
認証ベンダー：	Salesforce
無料問題の数：	163
バージョン：	v2026-07-20
ページの閲覧量：	102
問題集の閲覧量：	1660
https://www.jpnsshiken.com/shiken/Salesforce.PDII.v2026-07-20.q163.html	

質問: 1

開発者はApexテストクラス用のテストデータを必要とします。
開発者はテストメソッドにテストデータを提供するために何を利用できますか？ 2つ選択してください。）

A. List<sObject> ls = Test.loadData (Lead.sObjectType, 'myTestLeads');

B. myDataFactory.createTestRecords (10)

C. Database.createTestRecords (10)

D. List<sObject> ls = Test.loadDat (Lead.sObjectType, \$Resource + 'myTestLeads');

正解: ([正解を表示します](#))

説明／参考資料：

質問: 2

Lightning Webコンポーネントを構築する際に、データ取得のパフォーマンスを最大限に引き出すために、開発者はどの3つのアプローチを採用すべきでしょうか？ 3つ選択してください。

A. アクセス頻度の低いデータには遅延ロードを使用します。

B. layoutTypes: 9'Full') を使用して、一連のフィールドを表示します。

C. Lightning Data Serviceを使用します。

D. getRecordUi を使用してメタデータを取得します。

E. 可能な限り{Cacheable=true}を使用してください。

正解: ([正解を表示します](#))

遅延読み込みは、必要なときにのみデータを読み込むことでパフォーマンスを向上させます。
Lightning Data Service (LDS) は、データ操作とキャッシュを効率的に処理します。
Apex メソッドで {Cacheable=true} を使用すると、Lightning コンポーネント フレームワークがメソッドの戻り値をキャッシュできるようになり、同じデータに対するサーバーとの往復回数が削減されます。

参考文献：

Lightning Web Components 開発者ガイド

質問: 3

20万件のアカウントレコードという大規模なデータセットに使用できる、選択的なSOQLクエリはどれですか？ 2つ選択可能）

2つの回答を選択してください

A. SELECT Id FRM Account WHEP Name IN (List of Names) AND Customer_Number_c= 'ValueA

B. SELECT Id FRM Account WHERE Name != ''

- C. SELECT Id FROM Account WHERE Name LIKE '!'-NULL
D. SELECT Id FROM Account WHERE II IK (アカウントIDのリスト)

正解: ([正解を表示します](#))

質問: 4

メタデータAPI_____。

- A. 組織のカスタマイズを取得、デプロイ、作成、更新、または削除するために使用されます。最も一般的な用途は、サンドボックスまたはテスト組織から本番環境に変更を移行することです。
B. アカウント、リード、カスタムオブジェクトなどのレコードの作成、取得、更新、削除に使用され、パスワードの管理、検索の実行などを行うことができます。
C. RESTの原則に基づいており、大量のデータの読み込みや削除に最適化されています。バッチを送信することで、多数のレコードを非同期的にクエリ、クエリAll、挿入、更新、アップサート、または削除できます。
D. Salesforceとの連携を可能にする、強力で便利かつシンプルなRESTベースのWebサービスインターフェースを提供します。統合と開発が容易な点が利点であり、モバイルアプリケーションやWebプロジェクトで使用するのに最適な技術です。

正解: ([正解を表示します](#))

質問: 5

次のコードスニペットを考えてみましょう。

```
HttpRequest req = new HttpRequest();
req.setEndpoint('https://TestEndpoint.example.com/some_path');
req.setMethod('GET');
Blob headerValue = Blob.valueOf('myUserName' + ':' + 'strongPassword');
String authorizationHeader = 'Basic ' + EncodingUtil.base64Encode(headerValue);
req.setHeader('Authorization', authorizationHeader);
Http http = new Http();
HttpResponse res = http.send(req);
```

2つの回答を選択してください

- A. エンドポイントのURLを、endpointURLという名前のカスタムラベルに保存します。
B. エンドポイントと認証情報を保存するための名前付き認証情報 endpoint_NC を作成します。
C. コールアウトリクエスト内で req.setEndpoint,calloat;eodPolat NC'); を使用します。
D. req.setEndpoint(Label.endPoinrURL); を使用します。

正解: ([正解を表示します](#))

質問: 6

バッチクラスをテストする際、開発者は何をしなければなりませんか？ 2つ選択してください。）

- A. seeAllData=true を使用してください
B. Test.startTestQ と Test.stopTestQ にコードをカプセル化する
C. クラス*の executeメソッドを呼び出す
D. テストするレコード数を200件未満に制限する

正解: ([正解を表示します](#))

説明

seeAllData=trueを使う人は悪い人です

質問: 7

<apex:pageMessage> の正しい定義を選択してください。

- A. 標準的なSalesforceフォーマットで、ページに特定のメッセージを表示します
- B. 書式設定のない単一のメッセージで、ページ上の特定のコンポーネントに関連付けることができます。
- C. 標準の Salesforce フォーマットで、ページ上で発生したすべてのエラーを表示します。

ApexPages.addMessage」関数

- D. 書式設定なし。ページ上のすべてのエラーを表示します。

正解: ([正解を表示します](#))

質問: 8

開発者が、lightning-datatable 内のレコード一覧を表示する Lightning Web コンポーネントを作成しています。データベースに新しいレコードを保存した後、一覧が更新されません。

```
data;
@wire(recordList, ({recordId : 'recordId'}))
records(result){
    if(result.data){
        this.data = result.data;
    } else if(result.error){
        this.showToast(result.error);
    }
}
```

開発者は上記のコードをどのように変更すれば、起こる？

- A. 結果を格納する新しい変数を作成し、@track で注釈を付けます。
- B. データ変数に @track デコレータを追加します。
- C. 結果を格納する変数を作成し、refreshpex() を呼び出します。
- D. this.dazta に対して rafrasnApex() を呼び出します。

正解: ([正解を表示します](#))

質問: 9

<apex:actionStatus> の正しい定義を選択してください。

- A. コントローラーメソッドをJavaScriptから直接呼び出すことができます。<apex:form>タグで囲む必要があります。actionSupportとは異なり、これらの関数<apex:actionPoller>はJavaScriptコードから直接呼び出すことができます。
- B. AJAXリクエスト {actionFunction/actionSupport/actionPoller)に関連付けることができ、リクエストのステータス (進行中/完了)に応じて条件付きでコンテンツを表示します。id」フィールドを使用して名前を指定し、関連コンポーネントの {status}フィールドを使用してそれらを接続します。
- C. AJAXリクエストが生成された際に、サーバーが処理すべきコンポーネントを指定します。

- D. 他のコンポーネントにAJAXサポートを追加します（例onClick、onMouseUp、onFocusなど）。
- E. 指定した時間間隔でAJAXリクエストを送信します。再レンダリングされた場合はリセットされます。
- 正解: B ([コメントを发表する](#))

質問: 10

以下のAuraコンポーネントを参照してください。

Component markup:

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

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

Controller JS:

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

開発者は、コンポーネントの読み込みが遅いという苦情を受ける。

開発者は、コンポーネントのパフォーマンスを向上させるために、どのような変更を実装すればよいでしょうか？

- A. showContactInfo のデフォルト値を "False" に変更します。
- B. contactInfo のタイプを Map に変更します。
- C. showContactInfo の変更イベントハンドラを追加します。
- D. <c: contactInfo の内容をコンポーネントに移動します。

正解: A ([コメントを发表する](#))

質問: 11

次のコードスニペットを考えてみましょう。

開発者は、上記のコードスニペットに対して適切なコードカバレッジを提供するために、以下のテストクラスを作成しました。

しかし、テストを実行してもデータが返されず、アサーションが失敗します。

しかし、テストを実行してもデータが返されず、アサーションが失敗します。
開発者は、テストクラスが正常に実行されるようにするために、どのような編集を行うべきでしょうか？

- A. テストクラスにsetFixedSearchResultsメソッドを実装します。
- B. 検索機能の Apex クラスに、without sharing キーワードを実装します。
- C. @isTest アノテーションに seeAllData=true 属性を実装します。
- D. メソッド呼び出しを Test.startTest() と Test.stopTest() で囲みます。

正解: ([正解を表示します](#))

質問: 12

トランザクションの最大実行時間制限はどれくらいですか？

- A. 10分
- B. 5分
- C. 15分
- D. 制限はありません
- E. 20分

正解: A ([コメントを發表する](#))

質問: 13

開発者が検索機能を持つ Lightning Web コンポーネントを構築しています
連絡先の場合。コンポーネントは検索を伝達する必要があります。
他の無関係な Lightning Web コンポーネントの結果は、
検索が完了すると、異なるDOMツリーになります。
開発者は、この通信機能を実装するために何をすべきでしょうか？

- A. カスタムコンポーネントイベントを発火します。
- B. メッセージチャンネルにメッセージを投稿稿します。
- C. イベントチャンネルにイベントを公開します。
- D. アプリケーションイベントを発火する。

正解: ([正解を表示します](#))

質問: 14

開発者は、レコードページ上ではコンポーネントがデスクトップサイズのフォームファクターでのみレンダリングされるように、Lightning Webコンポーネントの設定ファイルにコードを追加する必要があります。

この要件を満たすために、開発者はコンポーネントのレコードページターゲット設定に何を追加すべきでしょうか？

- A.

```
<design>
  <designAttribute name="formFactor" value="Large"/>
</design>
```
- B.

```
<supportedFormFactors>
  <supportedFormFactor type="Large"/>
</supportedFormFactors>
```


- C.

```
<lightningLayout formFactor="Large">
</lightningLayout>

<property name="formFactor" value="Large">
</property>
```
- D.

正解: [\(正解を表示します\)](#)

Lightning Web コンポーネントを特定のフォーム ファクターでのみレンダリングしたい場合は、<targets> を使用できます。

コンポーネントの .xml 設定ファイル内のタグです。コンポーネントをデスクトップにのみ表示するように指定するには、targets="lightning__RecordPage" 属性を持つ <targetConfig> タグを使用し、その中に <supportedFormFactors> タグをネストします。このタグ内に、type="Large" 属性を持つ <supportedFormFactor> タグを含めます。

オプションBは、レコードページにおいてコンポーネントがデスクトップ版でのみ表示されるようにするための正しいマークアップです。

参考文献：

Lightning Web Components 開発者ガイド - Lightning Experience および Salesforce モバイル アプリ用のコンポーネントの設定

質問: 15

開発者は、アカウント登録に使用されるLightning Webコンポーネントからの入力に基づいてアカウントを更新するApexクラスを作成しました。アカウントの更新は、アカウントがまだ登録されていない場合にのみ行う必要があります。

```
account = [SELECT Id, Is_Salesforce_registered__c FROM Account WHERE Id = :accountId];

if (!account.Is_Registered__c) {
    account.Is_Registered = true;
    // ...set other account fields...
    update account;
}
```

開発者は、ユーザーが同じアカウントに対して同時にアップデートを行った場合に、互いのアップデートを上書きしないようにするために、どのような対策を講じるべきでしょうか？

- A. Add a try/catch block around the update.
- B. Use upsert instead of update.
- C. Use FOR UPDATE in the SOQL query.
- D. Include Is_Registered__c in the query to make sure it wasn't recently updated.

正解: [\(正解を表示します\)](#)

複数のユーザーが同時に同じレコードを更新する場合、互いの変更内容が上書きされるリスクがあります。Salesforceでは、これを防ぐために「レコードロック」と呼ばれる仕組みを提供しています。

選択肢Dが正解です。SOQLクエリにFOR UPDATEを含めることで、トランザクションの期間中、取得したレコードがロックされます。これにより、現在のトランザクションが完了するまで、他のトランザクションがレコードを更新することが防止されます。これは、2人のユーザーが互いの更新を上書きしてしまうような競合状態を回避するために不可欠です。

オプションAは誤りです。更新操作の周囲にtry/catchブロックを追加することは例外処理の良い方法ですが、同時更新による上書きを防ぐことはできません。

オプションBは誤りです。updateではなくupsertを使用しても、同時更新の問題は解決されません。upsert操作は、一致するIDまたは外部IDを持つレコードが既に存在するかどうかに基づいて、新しいレコードを挿入するか、既存のレコードを更新するために使用されます。

オプションCは誤りです。レコードをロックするには、FOR UPDATEをSELECT文自体ではなく、SOQLクエリ内で使用する必要があります。

参考文献：

Salesforce開発者向けドキュメント (ロックステートメントについて) ロックステートメント Salesforce開発者向けブログ Apexでの同時実行処理について) Apexでの同時実行処理

質問: 16

開発者は、Visualforceコントローラーのテストクラスで特定のアカウントレコードがテストされていることをどのように確認すればよいでしょうか？

- A. テストクラスでページ参照をインスタンス化し、テストクラスにアカウントを挿入し、System.setParentRecordId().get() を使用してアカウント ID を設定します。
- B. テストクラスにアカウントを挿入し、テストクラスでページ参照をインスタンス化し、System.currentPageReference().getParameters{}.put() を使用してアカウント ID を設定します。
- C. アカウントを Salesforce に挿入し、テスト クラスでページ参照をインスタンス化してから、System.setParentRecordId().get() を使用してアカウント ID を設定します。
- D. テストクラスでページ参照をインスタンス化し、テストクラスにアカウントを挿入し、seeAHDData=true を使用してアカウントを表示します。

正解: ([正解を表示します](#))

有効的な**PDII**問題集はJPNTTest.com提供され、**PDII**試験に合格することに役に立ちます！JPNTTest.comは今最新**PDII**試験問題集を提供します。JPNTTest.com PDII試験問題集はもう更新されました。ここで**PDII**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/PDII-mondaishu> **163**問、**30%ディスカウント**、特別な割引コード：**JPNshiken**」

質問: 17

開発者が、商談トリガーから呼び出される複雑な手数料計算エンジンをApexで構築しています。品質保証 (QA) 中に、計算結果が間違っているとの報告がありました。

開発者は、開発者サンドボックス内に代表的なテストデータと合格するテストメソッドを用意している。

開発者は、コードを実行し、重要な行で一時停止してさまざまなApex変数の値を視覚的に検査するために、どのようなツールまたはテクニックを3つ使用できますか？

3つの回答を選択してください

- A. Apexインタラクティブデバッガー
- B. 作業台
- C. 開発者コンソール
- D. ブレークポイント
- E. Apex Replay Debugger

正解: ([正解を表示します](#))

手数料計算エンジンのデバッグとトラブルシューティングを行うには、開発者はSalesforceが提供するいくつかのツールを使用して、Apexコードの実行を検査したり一時停止したりできます。Apexインタラクティブデバッガー :Apexコードの実行をリアルタイムでデバッグできます。このツールを使用すると、開発者はブレークポイントを設定したり、コードをステップ実行したり、変数を検査したり、式を評価したりすることができます。

開発者コンソール：開発者コンソールでは対話型のデバッグはできませんが、コードの実行状況を記録するログを表示する機能は備えています。デバッグログを調べることで、実行の流れや、さまざまな時点での変数の値を把握できます。

Apex Replay Debugger: このツールは、Visual Studio Code 用の Salesforce 拡張機能の一部です。開発者は、デバッグ ログをコードを一行ずつ実行しているかのように再生できるため、実行中の特定の時点での変数の状態を調べるのに非常に役立ちます。

参考文献：

Apexインタラクティブデバッガー

開発者コンソールのデバッグ

Apexリプレイデバッガー

質問: 18

組織には、Opportunity オブジェクトへのルックアップ関係を持つカスタムオブジェクト Registration_c があります。

開発者は、商談に関連する Registration__c レコードを表示するスタンドアロンの Visualforce ページを作成するために何を使用すべきでしょうか。

- A. 標準コントローラー
- B. コントローラー拡張機能を備えた標準コントローラー
- C. カスタムコントローラー
- D. コントローラー拡張機能

正解: ([正解を表示します](#))

質問: 19

最近、あるSalesforce組織の連携処理が、24時間以内のAPI呼び出し回数の上限を超えたために失敗しました。この連携処理は、Salesforceへのほぼリアルタイムかつ複雑なデータ挿入を扱っています。

データの流れは以下のとおりです。

- * この連携機能は、指定されたメールアドレスを持つ連絡先レコードを検索し、見つかった場合は、最初に見つかった一致する連絡先にタスクを追加します。
- * 一致するレコードが見つからない場合、統合機能は指定されたメールアドレスを持つリードレコードを検索し、見つかった場合は、最初に見つかった一致するリードにタスクを追加します。
- * 一致するものが見つからない場合、統合機能は、新しく作成されたリードに対してリードとタスクを作成します。

統合処理をほぼリアルタイムに保ちつつ、24時間あたりのAPI呼び出し回数の上限を超えないようにするには、どのような方法がありますか？

- A. REST APIとSOAP APIの両方を使用することで、24時間あたりに許可されるAPI呼び出しを効果的に2倍にすることができます。
- B. Flow を使用して、統合コードが行っていたすべてのロジックを実行できる受信メッセージを作成します。
- C. メールアドレスが与えられたときに、統合コードが行っていたすべてのロジックを実行するカスタムApex Webサービスを作成します。
- D. サードパーティシステムからの呼び出しを受け入れるための複数のApex InboundEmailHandlerを作成し、API制限を回避します。

正解: ([正解を表示します](#))

説明

質問: 20

以下の選択肢にあるクエリと、次の情報を考慮してください。

- * これらのクエリでは、アカウントレコードが20万件以上あることを前提とします。
- * これらのレコードには、ソフト削除されたレコード、つまりごみ箱に残っている削除済みレコードが含まれます。
- * 外部 ID としてマークされているフィールドが 2 つあります。

アカウント。これらのフィールドは customer_Number_c と ERR_Key_s です。

大量のデータ処理に最適化されているクエリはどれですか？ 2つ選択)

2つの回答を選択してください

- A. SELECT Id FROM Account WHERE Name != ' 'AND IsDeleted = false
- B. SELECT Id FROM Account WHERE Name != NULL
- C. SELECT ID FROM Account WHRE id IN :aListVariable
- D. SELECT Id FROM Accounts WHERE Name != ''

AND Customer_Number_c= 'ValueA'

正解: ([正解を表示します](#))

質問: 21

```
グローバル共有クラス MyRemoter {  
public String accountName { get; set; }  
public static Account account { get; set; }  
}
```



```
public AccountRemoter() {}  
@RemoteAction  
global static Account getAccount(String accountName) {  
    アカウント = [SELECT Id, Name, NumberOfEmployees]  
    FROM Account WHERE Name = :accountName];  
    アカウントを返す。  
}  
}
```

上記のApexクラスは、Visualforce検索ページで使用されるRemoteActionを定義しています。

どのコードスニペットを使用すれば、リモートアクションが正しいアカウントを返したことを検証できますか？

Account a = controller.getAccount('TestAccount');

A. System.assertEquals('TestAccount', a.Name);

B. Account a = remote.getAccount('TestAccount');

System.assertEquals('TestAccount', a.Name);

MyRemoter remote = new MyRemoter('TestAccount');

C. Account a = remote.getAccount();

System.assertEquals('TestAccount', a.Name);

Account a = MyRemoter.getAccount('TestAccount');

D. System.assertEquals('TestAccount', a.Name);

MyRemoter remote = new MyRemoter();

正解: ([正解を表示します](#))

質問: **22**

開発者は、ユーザーが名前でアカウントを検索できるVisualforceページを作成したいと考えています。アカウントが見つかった場合は、アカウントの詳細が画面に表示されます。アカウントが見つからない場合は、エラーメッセージがユーザーに表示されます。

これはどのように実現できるのでしょうか？ 2つ選択してください。）

A. エラーメッセージを表示するには、(apex: information) タグを使用します。

B. ApexPages.addMessage() メソッドを使用してエラーメッセージを追加します。

C. エラーメッセージを表示するには、<apex:pageMessages> タグを使用します。

D. account.addError() メソッドを使用してエラーメッセージを追加します

正解: ([正解を表示します](#))

説明／参考資料：

質問: **23**

REST API _____。

A. RESTの原則に基づいており、大量のデータの読み込みや削除に最適化されています。バッチを送信することで、多数のレコードを非同期的にクエリ、クエリオール、挿入、更新、アップサート、または削除できます。

B. Salesforceとの連携を可能にする、強力で便利かつシンプルなRESTベースのWebサービスインターフェースを提供します。統合と開発が容易な点が利点であり、モバイルアプリケーションやWebプロジェクトで使用するのに最適な技術です。

C. アカウント、リード、カスタムオブジェクトなどのレコードの作成、取得、更新、削除に使用され、パスワードの管理、検索の実行などを行うことができます。

D. 組織のカスタマイズを取得、デプロイ、作成、更新、または削除するために使用されます。最も一般的な用途は、サンドボックスまたはテスト組織から本番環境に変更を移行することです。

正解: ([正解を表示します](#))

説明／参考資料：

質問: **24**

ある企業は、名前、ウェブサイト、従業員番号などのアカウントレコード情報を取得する必要のあるウェブページを運営しています。ウェブページはSalesforceレコード1Dを認識しており、JavaScriptを使用してアカウント情報を取得します。

どの統合方法が最適でしょうか？

A. REST API

B. Apex SOAPウェブサービス

C. SOAP API

D. Apex REST Webサービス

正解: **B** ([コメントを發表する](#))

質問: **25**

Universal Containersは、アカウントに関連付けられた連絡先リストを取得するLightning WebコンポーネントとそのApexコントローラクラスを分析します。コードスニペットは以下のとおりです。

Lightning web component (accountContacts.html):

```
<template>
  <lightning-card title="Account Contacts">
    <lightning-datatable data={contacts} columns={columns} key-
      field="Id"></lightning-datatable>
  </lightning-card>
</template>
```

Apex controller class (AccountContactsController.cls):

```
public with sharing class AccountContactsController {
    @AuraEnabled
    public static List<Contact> getAccountContacts(String accountId) {
        return [SELECT Id, Name, Email, Phone FROM Contact WHERE
AccountId = :accountId];
    }
}
```

コードスニペットに基づいて、Lightning Webコンポーネントで連絡先の郵送先住所を表示するには、どのような変更を加える必要がありますか？

- A. Apex コントローラ クラスに新しいメソッドを追加して、郵送先住所を個別に取得し、Lightning Web コンポーネントを変更してこのメソッドを呼び出すようにします。
- B. Lightning Web コンポーネントの lightning-datatable コンポーネントを拡張して、MailingAddress フィールド用の列を追加します。
- C. getAccountContacts メソッドの SOQL クエリを修正して MailingAddress フィールドを含めるようにします。
- D. getAccountContacts メソッドの SOQL クエリを変更して MailingAddress フィールドを含め、JavaScript ファイルの columns 属性を更新して Mailing address フィールドを追加します。

正解: **C** ([コメントを发表する](#))

コードスニペットに基づくと、Lightning Web コンポーネントで連絡先の郵送先住所を表示するために必要な変更は、getAccountContacts メソッドの SOQL クエリを変更して MailingAddress フィールドを含めることです。こうすることで、Apex コントローラ クラスは、連絡先の名前と電話番号とともに郵送先住所を Lightning Web コンポーネントに返すことができます。Lightning Web コンポーネントは、@wire アノテーションを使用して Apex メソッドからデータを取得し、lightning-datatable コンポーネントに表示できます。郵送先住所を個別に取得する新しいメソッドを Apex コントローラ クラスに追加し、Lightning Web コンポーネントを変更してこのメソッドを呼び出す方法は、郵送先住所は他のフィールドと同じクエリで取得できるため、非効率的で不要です。Lightning Web コンポーネントの lightning-datatable コンポーネントを拡張して MailingAddress フィールドの列を含めることは、その列のデータが Apex メソッドから利用できないため、機能しません。getAccountContacts メソッドの SOQL クエリを変更して MailingAddress フィールドを含め、JavaScript ファイルの columns 属性を更新して Mailing address フィールドを追加する方法は有効ですが、これは最適な解決策ではありません。なぜなら、columns 属性は既に HTML テンプレート ファイルで定義されており、JavaScript ファイルに重複して記述する必要がないからです。参考資料: [Lightning Web コンポーネントから Apex メソッドを呼び出す]、[lightning-datatable]

質問: 26

開発者は、カスタムオブジェクト project_c に、Total_timesheets_c、Total_Approved_timesheet_c、Total_project_Timesheet_c という 3 つのロールアップ集計フィールドを作成しました。次に、開発者は、承認済みタイムシートと未承認タイムシートの比率を表示する新しいフィールドを作成するよう求められています。

- A. 展開には、数式フィールドを検証するテストクラスが必要です。
- B. 数式フィールドは、既存のレコードに対して遡及的に値を計算します。
- C. デプロイ中はテストメソッドは実行されません。
- D. 数式フィールドを使用すると、メンテナンスの手間が軽減されます。

正解: B,D ([コメントを發表する](#))

質問: 27

以下のAuraコンポーネントを参照してください。

HTML

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

開発者は、コンポーネントの読み込みが遅いという苦情を受けました。コンポーネントのパフォーマンスを向上させるために、開発者はどのような変更を実装すればよいのでしょうか？

- A. <c:contactInfo> の内容をコンポーネントに移動します。
- B. contactInfo のタイプを Map」に変更します。
- C. showContactInfo のデフォルト値を false」に変更します。

正解: ([正解を表示します](#))

Auraコンポーネントでは、<aura:if>タグは 遅延評価」です。isTrue式がtrueと評価されるまで、その本体内のコンポーネントはインスタンス化もレンダリングもされません。提供されているコードでは、showContactInfoはデフォルトでtrueに設定されています。

これは、親コンポーネントが初期化を開始すると、子コンポーネント <c:contactInfo> の作成とレンダリングを同時に開始することを意味します。

showContactInfo のデフォルト値を false」 オプション C)に変更することで、開発者は 遅延読み込み」パターンを実装できます。親コンポーネントは子コンポーネントのレンダリングに必要な処理をスキップするため、高速に読み込まれます。その後、初期化ハンドラが実行され、必要なデータが取得され、データが準備できた場合にのみブール値が true に設定されます。これにより、体感パフォーマンス」と初期ページ読み込み時間が向上します。

オプションAは、親コンポーネントの複雑さと重量が増加し、処理速度が低下する可能性があるため、誤りです。オプションB Mapへの変更)は、レンダリングエンジンの速度に根本的な影響を与えません。display属性をfalseに設定することで、ブラウザはページ読み込みの最初の数ミリ秒という重要な時間帯に、必要最低限のものだけをレンダリングするようになります。

質問: 28

Apex関数 PageReference.setRedirect」がFalseに設定されている場合、どのような種類の要求が行われますか？

- A. リクエストを受け取る
- B. PageReferenceが同じコントローラと拡張機能のサブセットを指している場合はポストバックリクエスト、そうでない場合はGETリクエスト
- C. ポストバックリクエスト

正解: [A \(コメントを発表する\)](#)

質問: 29

ユーザーから、カスタムLightning Webコンポーネント (WC) 上のボタンに入力したデータが保存されないという報告が寄せられています。ボタンはクリックされたように見えますが、LWCは何も反応せず、そのままの状態になっています。

開発者は、エラーメッセージが適切に表示されるようにするために、どのような方法を用いるべきでしょうか？

- A. `allowNone` を `false` に設定してデータベースメソッドを使用します。
- B. JavaScriptとHTMLを追加してエラーメッセージを表示します。
- C. コンポーネントに `<apex:messages/>` タグを追加します。
- D. DML ステートメントを `try-catch` ブロックで囲みます。

正解: [\(正解を表示します\)](#)

質問: 30

以下のAuraコンポーネントを参照してください。

Component markup:

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

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

Controller JS:

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

開発者は、コンポーネントの読み込みが遅いという苦情を受ける。

開発者は、コンポーネントのパフォーマンスを向上させるために、どのような変更を実装すればよいでしょうか？

- A. `contactInfo` のタイプを `Map` に変更します。

- B. <c: contactInfo の内容をコンポーネントに移動します。
- C. showContactInfo の変更イベントハンドラを追加します。
- D. showContactInfo のデフォルト値を "False" に変更します。

正解: ([正解を表示します](#))

読み込みが遅いAuraコンポーネントのパフォーマンスを改善するには、開発者は次のような変更を行うことができます。

* D. showContactInfo のデフォルト値を False」に変更する: showContactInfo のデフォルト値を False に設定すると、コンポーネントはデフォルトでは連絡先情報を取得またはレンダリングしないため、初期読み込み時間を短縮できます。情報は、ユーザーの操作やその他のトリガーに基づいて、後で取得して表示できます。

contactInfo のタイプを Map」に変更する オプション A)か、<c:contactInfo> の内容をコンポーネント内に移動する オプション B)ことで、必ずしも読み込み時間が短縮されるとは限りません。showContactInfo の変更イベント ハンドラを追加する オプション C)と、処理が増えるため、効率的に処理されないとコンポーネントの動作が遅くなる可能性があります。

参考文献：

Lightning コンポーネントのパフォーマンスに関するベストプラクティス パフォーマンスを向上させる

質問: 31

開発者は、サンドボックス環境ではテストが失敗するが、本番環境では失敗しないことを確認した。サンドボックスが作成されて以来、どちらの環境にもコードやメタデータの変更は一切行われていない。

この問題を解決するために、どの点を確認すべきでしょうか？

- A. ワークフロールールが非アクティブになっていることを確認してください。
- B. サンドボックスが本番環境と同じリリースであることを確認してください。
- C. Apex クラスが同じ API バージョンであることを確認してください。
- D. Process Builder のプロセスが非アクティブになっていることを確認してください。

正解: ([正解を表示します](#))

有効的な**PDII**問題集はJPNTTest.com提供され、**PDII**試験に合格することに役に立ちます！JPNTTest.comは今最新**PDII**試験問題集を提供します。JPNTTest.com PDII試験問題集はもう更新されました。ここで**PDII**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/PDII-mondaishu> 163問、30%**ディスカウント**、特別な割引コード：**JPNshiken**」

質問: 32

組織には、連絡先とアカウントのアドレスが保存されるたびに、Apexコードによって会社の標準に正規化される必要があるという2つの要件があります。

これを実装する最適な方法は何ですか？

- A. 連絡先とアカウントの Apex トリガーがヘルパー クラスを呼び出し、アドレスを正規化します
- B. 連絡先とアカウントのアドレスを正規化するApexトリガー
- C. アカウントのApexトリガーが連絡先トリガーを呼び出し、住所を正規化する
- D. 取引先責任者のApexトリガーがアカウントトリガーを呼び出し、住所を正規化する

正解: ([正解を表示します](#))

質問: 33

展示する：

```

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

```

上記のテスト方法は、開発者が多数のアカウントが同時に顧客に更新されたときに多数のクエリを実行すると認識しているApexトリガーをテストします。SOQLクエリが多すぎるため、テストメソッドは20行目で失敗します。これを正しく修正するにはどうすればよいでしょうか？

```

Line 18:     update DataFactory.accounts;
Line 19:
Line 20:     List<Account> acctsAfter = [SELECT Number_Of_Transfers__c FROM Account WHERE Id IN
:DataFactory.accounts];
Line 21:
Line 22:     //Utility to assert Number_Of_Transfers__c is correct based on test data
Line 23:     AssertUtil.assertNumberOfTransfers(acctsAfter);
Line 24: }

```

上記のテスト方法は、開発者が多数のアカウントが同時に顧客に更新された際に、大量のクエリを実行することがわかっているApexトリガーをテストします。このテストメソッドは、SOQLクエリが多すぎるため、20行目で失敗します。これを正しく修正するにはどうすればよいでしょうか？

- A. コードの18行目の前にTest.startTest()を追加し、18行目の後にTest.stopTest()を追加してください。
- B. Apex トリガーの大部分を Process Builder プロセスに置き換えて、トリガー内のクエリ数を減らします。
- C. DataFactory クラスを変更してアカウントの作成数を減らし、トリガー内のクエリ数を削減します。
- D. コードの7行目と20行目の両方に、Test.startTest()を前に追加し、Test.stopTest()を後に追加します。

正解: D ([コメントを发表する](#))

質問: 34

開発者が、Salesforceモバイルアプリで使用されるページをApp Builderで作成しています。

開発者は、ページが最適なパフォーマンスで動作するようにするために、どのような2つの方法に従うべきでしょうか？

2つの回答を選択してください

- A. レコード詳細ページには25個のフィールドを制限します。
- B. タブとアコーディオンコンポーネントの数を制限してください。
- C. ページ上に表示されるコンポーネントを5つに制限します。
- D. App Builder のパフォーマンス分析を使用してページを分析します。

正解: A,D ([コメントを发表する](#))

質問: 35

Visualforceのページは、表示するデータ量が多いため、読み込みに時間がかかります。

開発者はパフォーマンスを向上させるためにどのような戦略を用いることができますか？

- A. コントローラのコンストラクタではなく、遅延読み込みを使用して必要に応じてデータをロードします。
- B. ページ内で <apex:actionPoller) を使用して、すべてのデータを非同期で読み込みます。
- C. カスタムコントローラで使用されるリスト変数には、transient キーワードを使用します。
- D. JavaScriptを使用して、データ処理をコントローラーではなくブラウザに移動します。

正解: A ([コメントを發表する](#))

遅延読み込みは、大量のデータを読み込むVisualforceページのパフォーマンスを向上させるための戦略です。

コントローラのコンストラクタではなく、必要に応じてデータを読み込むことで、最初のページ読み込みが高速化され、追加データは必要な場合にのみ読み込まれます。

参考資料 :Visualforce開発者ガイド - パフォーマンスの向上

質問: 36

非同期Apexを使用することによってのみ実行できるユースケースはどれですか？

- A. 数万件のレコードを照会する
- B. 将来完了するバッチ処理をスケジュールするための呼び出しを行う
- C. ApexトリガーからWebサービスを呼び出す
- D. 挿入完了後のレコードの更新1

正解: ([正解を表示します](#))

Salesforceは、データベーストランザクションと外部通信に関して厳格なアーキテクチャ上の境界を設けています。Apexトリガーが実行されると、それは同期データベーストランザクションの一部となります。プラットフォームがトリガーから直接同期コールアウト (Webサービス呼び出し) を許可した場合、すべてのアクティブなレコードロックを含むデータベーストランザクションは、外部システムからの応答を待つ間、開いたままになります。これは深刻なパフォーマンス低下を引き起こし、同時リクエスト制限に達する可能性があります。

したがって、ApexトリガーからWebサービスを呼び出す場合 (オプションC)、非同期処理が必要です。開発者は、呼び出しロジックを@future(callout=true)メソッド、Queueableクラス、またはプラットフォームイベントに渡す必要があります。これにより、トリガーは処理を完了してデータベーストランザクションを即座にコミットできる一方、呼び出しは独自のガバナー制限を持つ別のバックグラウンドプロセスで実行されます。

オプションAは、標準の同期クエリでは最大50,000件のレコードを取得できるため誤りです。オプションBは、System.scheduleBatchがプロセスを開始するための同期呼び出しであり、プロセス自体は後で実行されるため誤りです。オプションDは、レコードの更新は挿入後トリガーのコンテキストで同期的に処理できるため誤りです。

質問: 37

ある企業は、リクエストとリクエストラインを更新し、更新されたレコードを使用して外部ERPシステムのRESTエンドポイントにコールアウトを行うコードを持っています。

```

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);
    }
}
}

```

salesforce

callousUtil.makeRestCallout は 「コミットされていない作業が保留中です。cut を呼び出す前にコミットまたはロールバックしてください」というエラーで失敗します。
この問題に対処するために、どのような対策を講じるべきでしょうか？

- A. データベースを削除します。setSavepoint と Database.rollback。
- B. CallousUitll .makeRestCallout を @future メソッドに変更します。
- C. callousUtil makeRestCallout を @InvocsblsMethod メソッドに変更します。

正解: [B \(コメントを发表する\)](#)

質問: 38

再帰トリガーの深度におけるトランザクション制限はどれくらいですか？

- A. 制限はありません
- B. 10
- C. 16
- D. 17
- E. 3

正解: [\(正解を表示します\)](#)

質問: 39

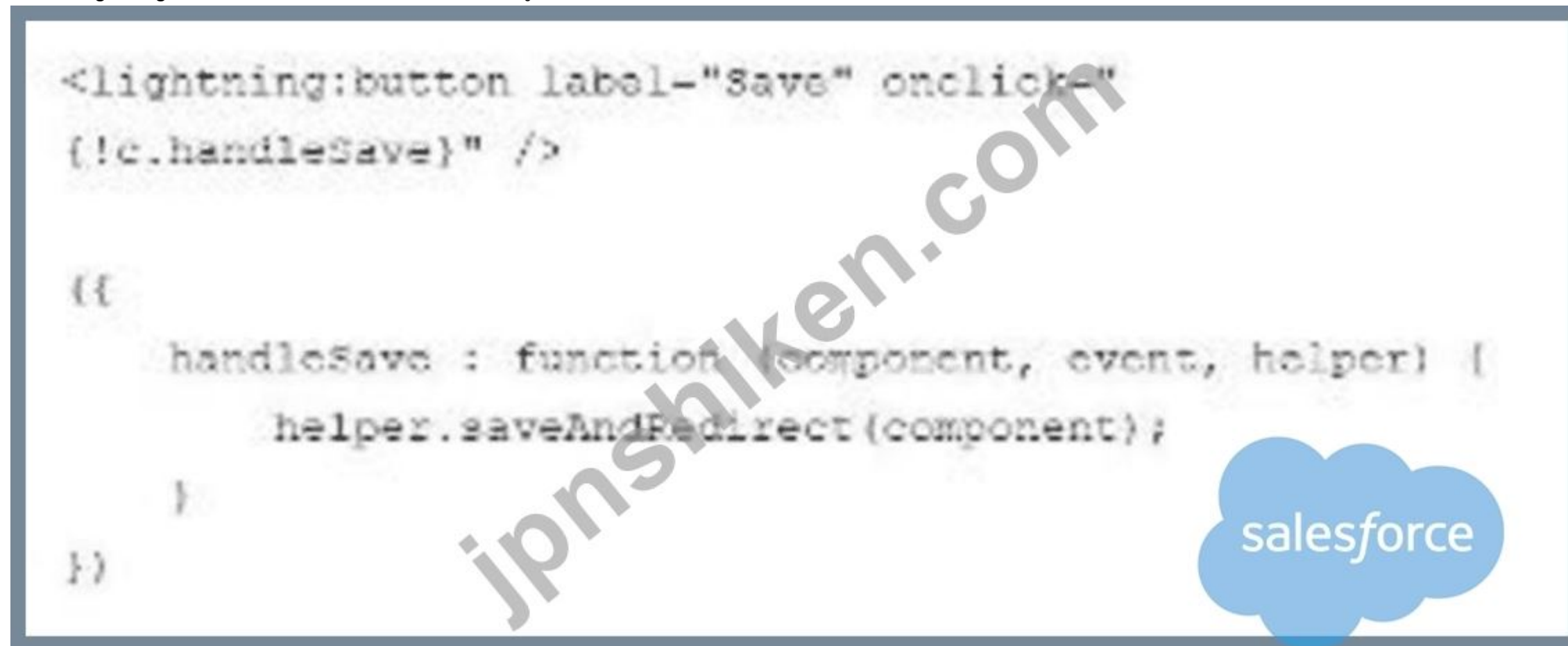
開発者が外部サービスと連携するVisualforceページを作成しています。この機能をテストするために、開発者はどのインターフェイスを実装する必要がありますか？ 2つ選択してください。

- A. HTTPCalloutMock
- B. StaticResourceCalloutMock
- C. HTTPResponseMock
- D. HTTPRequestMock

正解: ([正解を表示します](#))

質問: 40

以下のLightningコンポーネントを参照してください。



Lightningコンポーネントを使用すると、ユーザーはボタンをクリックして変更を保存し、その後別のページにリダイレクトされます。

現在、ユーザーが「保存」ボタンをクリックすると、レコードは保存されますが、リダイレクトされません。

開発者がJavaScriptをデバッグするために使用できる3つのテクニックは何ですか？

3つの回答を選択してください

- A. JavaScript で `console.log()` メッセージを使用します。
- B. ブラウザの開発者ツールを使用してJavaScriptをデバッグします。
- C. ユーザー向けにLightningコンポーネントのデバッグモードを有効にします。
- D. 開発者コンソールを使用してチェックポイントを表示します。
- E. 開発者コンソールを使用してデバッグログを表示します。

正解: ([正解を表示します](#))

LightningコンポーネントでJavaScriptをデバッグする場合、開発者は以下の手法を利用できます。

- * A. JavaScript で console.log() メッセージを使用する: これは、ブラウザのコンソールに情報をログ出力する標準的なデバッグ手法であり、実行フローや出力変数の値を追跡するのに役立ちます。
- * B. ブラウザの開発者ツールを使用して JavaScript をデバッグします。最新のブラウザには、ブレークポイントを設定したり、コードをステップ実行したり、実行時に変数を検査したりできる開発者ツールが付属しています。
- * C. ユーザー向けに Lightning コンポーネントのデバッグ モードを有効にする: ユーザー向けにデバッグ モードを有効にすると、より詳細なエラー メッセージとクライアント側ライブラリの非圧縮バージョンが提供され、問題のデバッグが容易になります。

開発者コンソールを使用してチェックポイント (オプションD)とデバッグログ (オプションE)を表示する方法は、JavaScriptのデバッグよりもApexのデバッグに適しています。

参考文献：

Lightning コンポーネントのデバッグに関する Salesforce ドキュメント: デバッグ

質問: 41

開発者が、サードパーティ製のJavaScriptフレームワークを使用するVisualforceページを作成しました。開発者は、Apexコントローラー用のJavaScriptリモート処理を使用して、JavaScript関数にデータを提供することにしました。

Apexでリモートメソッドを宣言する正しい構文はどれですか？ 2つ選択してください。）

- A. @RemoteAction public static String getTable()
- B. @RemoteAction global String getTable()
- C. @RemoteObject global static String getTableQ
- D. @RemoteAction global static String getTable()

正解: ([正解を表示します](#))

質問: 42

開発者が、フィルター機能を備えたカスタムデータテーブルをLightning Webコンポーネントに実装しました。

しかし、フィルターを変更した際に読み込み時間が長くなるというサポートチケットがユーザーから寄せられています。このコンポーネントは、選択されたフィルターに基づいてレコードをクエリするために呼び出されるApexメソッドを使用しています。

コンポーネントのパフォーマンスを向上させるために、開発者は何をすべきでしょうか？

- A. コンポーネントが作成されたときにすべてのレコードをリストに返し、JavaScript で配列をフィルタリングします。
- B. カスタムインデックスを使用した選択的SOQLクエリを使用します。
- C. SOSLを使用して、フィルタ変更時のレコードを照会します。
- D. Apex メソッドで setstorable() を使用して、レスポンスをクライアント側のキャッシュに保存します。

正解: ([正解を表示します](#))

Lightning Web Component (LWC) で SOQL クエリの読み込み時間が原因でパフォーマンスの問題が発生した場合、最適なアプローチはクエリの選択性を向上させることです。これは、カスタムインデックスを使用した選択的な SOQL クエリを使用することで実現できます。Salesforce は、標準のインデックスでは最適化できないクエリのパフォーマンスを向上させるためにカスタム インデックスを作成できます。クエリが選択的である場合、インデックスを使用してデータベースからレコードを効率的に取得できるため、クエリの実行時間が短縮され、フィルタが変更されたときのコンポーネントのパフォーマンスが向上します。その他のオプション (すべてのレコードを返す、SOSL を使用する、またはクライアント側でキャッシュする) は、パフォーマンスの問題の根本原因である、より効率的なデータベース操作の必要性に直接対処するものではありません。

参考文献：

SOQLクエリの選択性を高める

SOQLクエリにおけるインデックスの使用

質問: 43

Lightning WebコンポーネントでJavaScriptファイルを静的リソースとして読み込むために、完了しなければならない3つのアクションはどれですか？3つ選択してください。

- A. <script> タグ内で静的リソースを参照します。
- B. 静的リソースをインポートします。
- C. loadScript を呼び出します。
- D. platformResourceLoader からメソッドをインポートします。
- E. 静的リソースをDOMに追加します。

正解: ([正解を表示します](#))

質問: 44

ある開発者が、外部システムへの複数の呼び出しを行うためのApexクラスを作成しました。

これらのコールアウトで使用する URL が頻繁に変更される場合、開発者は Apex クラスへの変更を最小限に抑えるためにどの機能を使用すべきでしょうか？

- A. リモートサイト設定
- B. セッションID
- C. 名前付き認証情報
- D. 連携アプリ

正解: ([正解を表示します](#))

説明

説明/参考資料 https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_callouts_named_credentials.htm

質問: 45

以下の試験方法を参照してください。

Java

@isTest

```
static void testAccountUpdate() {  
    Account acct = new Account(Name = 'Test');  
    acct.Integration_Updated__c = false;  
    アカウントを挿入します。  
    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);  
}
```

このテストメソッドは、外部システムをアカウント情報で更新するWebサービスを呼び出し、完了時にアカウントのIntegration_Updated__cチェックボックスをTrueに設定します。しかし、テストは実行に失敗し、「TestMethodとして定義されたメソッドはWebサービス呼び出しをサポートしていません」というエラーで終了します。この問題を解決する最適な方法がありますか？

- A. CalloutUtil.sendAccountUpdate の周りに if (!Test.isRunningTest()) を追加します。
- B. CalloutUtil.sendAccountUpdate の前に Test.startTest() を、後に Test.stopTest() を追加します。
- C. CalloutUtil.sendAccountUpdate の前に Test.startTest() を追加し、後に Test.setMock と Test.stopTest() を追加します。
- D. CalloutUtil.sendAccountUpdate の前に Test.startTest() と Test.setMock を追加し、後に Test.stopTest() を追加します。

正解: ([正解を表示します](#))

Salesforce では、ユニット テスト中に実際のネットワーク コールアウトを禁止するという厳格な制限が課せられています。これは、テストが決定論的で高速であり、外部のサードパーティ システムの可用性や状態に依存しないようにするためです。テスト エンジンが System.Http.send() またはモックなしの Web サービス呼び出しを検出すると、TestMethod として定義されたメソッ

ドは Web サービス コールアウトをサポートしていません」というエラーがスローされます。34 この問題を解決するには、開発者はモック実装を提供する必要があります。Test.setMock() (オプション D) を使用すると、開発者は Apex ランタイムに、実際の接続を試みる代わりに、すべてのコールアウトをインターセプトして事前定義された応答を返すように指示します。モック クラス 5 は、HttpCalloutMock インターフェイス (REST 用) または WebServiceMock インターフェイス (SOAP 用) のいずれかを実装する必要があります。

さらに、モックの呼び出しとコールアウト メソッドは Test.startTest() と Test でラップする必要があります。

stopTest().6

* Test.startTest(): ガバナー制限をリセットし、テスト対象の特定のロジックに新たなコンテキストを提供します。7

* Test.stopTest(): 非同期処理 (@future や Queueable などの callouts でよく使用される) を、次のコード行が実行される前に強制的に完了させます。

提供されたコードでは、プラットフォームがどのモックを使用するかを認識できるように、CalloutUtil.sendAccountUpdate の前に Test.setMock を呼び出す必要があります。Test.stopTest() に到達すると、モック応答が処理され、チェックボックスが更新され、後続の SOQL クエリとアサーションで更新されたデータが正しく表示されます。オプション A はロジックを完全にスキップするため、統合ロジックのコードカバレッジが 0% になるため、好ましくない方法です。

質問: 46

ある顧客は、各ユーザーが最大1500件の売上予測を入力し、ピボットされた予測計算結果を即座に表示できるVisualforceページを1つ使用しています。ユーザーからは、ページの読み込みが遅く、ヒープとビュー状態の制限に関するエラーメッセージが表示されるという苦情が寄せられています。ページのパフォーマンスを最適化するための3つの推奨事項は何ですか？

- A. ページネーションを実装し、1ページあたりのレコード数を削減します。
- B. ピボット予測計算を行うための数式項目を作成します。
- C. コントローラーアクションの代わりにJavaScriptリモート処理を使用します。
- D. 売上予測のリストを一時的なものとして指定します。
- E. 計算機能と入力機能を分離する。

正解: B,C,E ([コメントを發表する](#))

有効的な**PDII**問題集はJPNTTest.com提供され、**PDII**試験に合格することに役に立ちます！JPNTTest.comは今最新**PDII**試験問題集を提供します。JPNTTest.com PDII試験問題集はもう更新されました。ここで**PDII**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/PDII-mondaishu> 163問、30%**ディスカウント**、特別な割引コード：**JPNshiken**」

質問: 47

開発者は、送信されたメールを処理し、そのメールの内容をデータとして使用してアカウントと連絡先を作成するサービスを作成する必要があります。開発者はこの要件をどのように満たすことができるでしょうか？

- A. 自動起動フローおよびプロセスビルダーを使用する。
- B. Fuel API をメールデータ拡張機能とともに使用します。
- C. Heroku Data Clipsを使用してメールを処理します。
- D. Apex受信メールハンドラーを使用します。

正解: D ([コメントを發表する](#))

質問: 48

Salesforce組織でプラットフォームイベントが定義された後、イベントはどのメカニズムで公開できますか？

- A. 外部アプリはAPIを使用してイベントメッセージを共同発行します。
- B. 内部アプリは権限プロセスを使用できます。

C. 外部アプリは標準のストリーミングAPIを必要とします。

D. 内部アプリは送信メッセージを使用できます。

正解: (正解を表示します)

Salesforceのプラットフォームイベントは、Salesforce APIを介した外部アプリケーションからのイベントなど、さまざまな方法で公開できます。

選択肢Aが正解です。外部アプリケーションは、REST APIやSOAP APIなどのSalesforce APIを使用して、Salesforceのプラットフォームイベントにイベントを発行できるからです。

オプションB、C、Dは、プラットフォームイベントの公開には使用されないメカニズムについて説明しています。エンタイトルメントプロセスはサービス契約の設定と追跡のためのものであり、標準のストリーミングAPIはイベントの購読のためのもので、公開のためのものではありません。また、アウトバウンドメッセージは、外部サービスに情報を送信するために使用されるワークフローアクションであり、プラットフォームイベントの公開のためのものではありません。

参考文献：

Salesforce開発者向けドキュメント - プラットフォームイベント

質問: 49

opportunityListという名前の商談レコードのリストが与えられた場合、その商談のアカウントのすべての取引先責任者を照会するのに最適なコードスニペットはどれですか？

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

A.

```
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);
}
```

B.

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

C.

```

List<Contact> contactList = new List<Contact>();
Set<Id> contactIds = 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);
}

```

D.

正解: **B** ([コメントを發表する](#))

質問: 50

コントローラーの単体テストを行う際のベストプラクティスは何ですか？ 2つ選択してください。

- A. seeAllData=true を使用してテストデータにアクセスします。
- B. getURL()を使用して正しいページ参照を確認します
- C. getParameters().putを使用してクエリパラメータを設定します
- D. Test.setMock() を利用してユーザー操作をシミュレートします。

正解: ([正解を表示します](#))

質問: 51

非同期Apexを使用することによってのみ実行できるユースケースはどれですか？

- A. ApexトリガーからWebサービス呼び出す
- B. 将来完了するバッチ処理をスケジュールするための呼び出しを行う
- C. 数万件のレコードを照会する
- D. 挿入完了後のレコードの更新

正解: ([正解を表示します](#))

質問: 52

開発者がデータ入力用のVisualforceページを作成し、フィールドレベルでエラーを表示する必要がある。

開発者はどのタグを使用すべきですか？

- A. < apex:pageMessages>
- B. < apex:pageMessage>
- C. < apex:messages>
- D. < apex:message>

正解: **D** ([コメントを發表する](#))

質問: 53

Apex関数 PageReference.setRedirect」がFalseに設定されている場合、どのような種類の要求が行われますか？

- A. リクエストを受け取る
- B. ポストバックリクエスト

C. PageReferenceが同じコントローラと拡張機能のサブセットを指している場合はポストバックリクエスト、そうでない場合はGETリクエスト
正解: **C** ([コメントを發表する](#))

質問: 54

Visualforceのページは、表示するデータ量が多いため、読み込みに時間がかかります。
開発者はパフォーマンスを向上させるためにどのような戦略を用いることができますか？

- A. カスタムコントローラで使用されるリスト変数には、transient キーワードを使用します。
- B. ページ内で <apex:actionPollar) を使用して、すべてのデータを非同期で読み込みます。
- C. コントローラのコンストラクタではなく、遅延読み込みを使用して必要に応じてデータをロードします。
- D. JavaScriptを使用して、データ処理をコントローラではなくブラウザに移動します。

正解: ([正解を表示します](#))

質問: 55

開発者は、カスタムプロファイル 監査担当者」に割り当てられたユーザーが、アカウントオブジェクトに関連するタスクに対して検索を実行できる、履歴タスクレポート機能を実装する必要があります。

開発者は、検索結果が12ヶ月から24ヶ月前のタスクを返すようにする必要があります。削除されたタスクは除外し、アーカイブされているか否かにかかわらず、すべてのタスクを含める必要があります。

以下のコードにおいて、L2から24ヶ月前のタスクを取得する有効な方法として、プレースホルダーに挿入すべきSELECT文はどれでしょうか？

- ```
date startDate = System.today().addMonths(-24);
date endDate = System.today().addMonths(-12);
// Insert SELECT statement here

SELECT AccountId, CreatedDate, Subject
FROM Task
WHERE What.Type = 'Account' AND IsDeleted=FALSE AND CreatedDate >= startDate AND CreatedDate <= endDate);
```
- A.
- ```
( SELECT AccountId, CreatedDate, Subject
FROM Task
WHERE What.Type = 'Account' AND CreatedDate >= startDate AND CreatedDate <= endDate; SELECT ROWS);
```
- B.
- ```
(SELECT AccountId, CreatedDate, Subject
FROM Task
WHERE What.Type = 'Account' AND IsDeleted=FALSE AND CreatedDate >= startDate AND CreatedDate <= endDate; SELECT ROWS);
```
- C.
- ```
(SELECT AccountId, CreatedDate, Subject
FROM Task
WHERE What.Type = 'Account' AND IsDeleted=FALSE AND CreatedDate >= startDate AND CreatedDate <= endDate; SELECT ROWS);
```
- D.

正解: ([正解を表示します](#))

12～24ヶ月前のタスクを取得するには、作成日 (CreatedDate)に基づいてフィルタリングするSOQLクエリを使用する必要があります。このクエリでは、削除されたレコードを除外し、アクティブなタスクとアーカイブされたタスクの両方を含める必要があります。タスクは一定期間（通常は12ヶ月）経過後にアーカイブされる可能性があるため、クエリにはアーカイブされたタスクも含まれるようにする必要があります。

オプションCが正解です。SOQLクエリでALL ROWSを使用しているため、アクティブなレコードとアーカイブ済み (ソフト削除済み)のレコードの両方が含まれます。また、クエリはCreatedDateに範囲指定を正しく使用し、12～24か月前に作成されたタスクを選択できるようにしています。

オプションAは、アーカイブされたタスクを結果に含めるために必要なALL ROWSキーワードが含まれていないため、誤りです。オプションBは、過去12ヶ月以内に作成されたタスクのみを取得するため、12ヶ月から24ヶ月前に作成されたタスクは取得しないため、誤りです。

選択肢Dは、作成日に基づいてタスクを正しくフィルタリングして、12ヶ月から24ヶ月以内のタスクのみを含めることができないため、誤りです。

参考文献：

Salesforce開発者向けドキュメント SOQLおよびSOSLリファレンス SOQLおよびSOSLリファレンス Salesforce開発者向けドキュメント アーカイブ済みレコードの操作 アーカイブ済みレコードの操作

質問: 56

CreateOneAccount クラスが1つのアカウントを作成し、Queueable インターフェースを実装すると仮定した場合、Apex コードをテストする構文はどれですか？

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. 選択肢D

- B. オプションC
- C. オプションB
- D. オプションA

正解: **C** ([コメントを發表する](#))

質問: **57**

開発者が連絡先を検索するLightning Webコンポーネントを作成しています。このコンポーネントは、検索が完了した際に、異なるDOMツリーにある他の関連性のないLightning Webコンポーネントに検索結果を伝える必要があります。

開発者は、この通信機能を実装するために何をすべきでしょうか？

- A. メッセージチャンネルにメッセージを投稿稿します。
- B. イベントチャンネルにイベントを公開します。
- C. カスタムコンポーネントイベントを発火します。
- D. アプリケーションイベントを発火します。

正解: ([正解を表示します](#))

質問: **58**

次のコードスニペットを考えてみましょう。

```
HttpRequest req = new HttpRequest();  
req.setEndpoint('https://TestEndpoint.example.com/salesforce');  
req.setMethod('GET');  
Blob headerValue = Blob.valueOf('myUserName' + ':' + 'strongPassword');  
String authorizationHeader = 'Basic ' + EncodingUtil.base64Encode(headerValue);  
req.setHeader('Authorization', authorizationHeader);  
Http http = new Http();  
HttpResponse res = http.send(req);
```

開発者は、コードを変更することなくエンドポイントと認証情報を変更できる柔軟性を持たせるために、どのような2つの手順を踏むべきでしょうか？

2つの回答を選択してください

- A. コールアウトリクエスト内で req.setEndpoint('callout:endPoint_NC'); を使用します。
- B. req.setEndpoint(Label endPointURL); を使用します。
- C. エンドポイントと認証情報を保存するための名前付き認証情報 endPoint_wc を作成します。
- D. エンドポイントの URL を endPointURL という名前のカスタム ラベルに保存します。

正解: ([正解を表示します](#))

質問: **59**

ある組織は、顧客の注文情報をカスタムオブジェクト order__c に記録しており、このオブジェクトには配送先住所のフィールドが含まれています。開発者は、配送先住所の地域に関連付けられた一定のパーセンテージに基づいて、注文の配送料を計算するコードを追加するよう指示されています。

開発者は、地域ごとの料金を保存するために何を使用すべきでしょうか？そうすれば、変更を本番環境にデプロイした際に、計算が正しく機能するために追加の手順は不要になります。

- A. カスタム階層設定

- B. カスタムメタデータタイプ
- C. カスタムリスト設定
- D. カスタムオブジェクト

正解: B ([コメントを发表する](#))

カスタムメタデータ型は、地域ごとの料金を保存するのに最適なオプションです。これにより、変更を本番環境にデプロイした際に、計算を機能させるために追加の手順は不要になります。カスタムメタデータ型を使用すると、Apexコードで利用できるカスタム構成データを定義できます。カスタムメタデータ型のレコードはメタデータの一部としてデプロイされるため、デプロイ後に本番組織で自動的に利用可能になります。カスタム階層設定は、異なるレベルのユーザーまたはプロファイルのプリファレンスを保存するために使用されますが、このシナリオには適していません。カスタムリスト設定は、Apexコードでアクセスできる単純なデータリストを保存するために使用されますが、カスタムリスト設定レコードはメタデータの一部としてデプロイされないため、デプロイ後に本番組織で手動で作成またはインポートする必要があります。カスタムオブジェクトは、他のオブジェクトに関連付けることができる複雑なデータを保存するために使用されますが、このシナリオでは必要ありません。カスタムオブジェクトレコードもメタデータの一部としてデプロイされないため、デプロイ後に本番組織で手動で作成またはインポートする必要があります。参照: [カスタムメタデータ型]、[カスタム設定]、[カスタムオブジェクト]

質問: 60

Apex トリガーと Apex クラスは、ケースが変更されるたびにカウンター Edit __C をインクリメントします。

```
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);
}

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

A.

```

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

```

B.

```

}

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

```

C.

```

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

```

D.

正解: [A \(コメントを发表する\)](#)

質問: 61

Universal Containers社は、外部のREST APIとの頻繁なやり取りを必要とするSalesforceアプリケーションを開発している。コードの重複を避け、保守性を向上させるために、コードの再利用性を高めるためのAPL統合をどのように実装すべきでしょうか？

- A. APT統合コードを静的リソースとして保存し、各Apexクラスでそれを参照します。
- B. API統合コードを必要とする各Apexクラスに直接含めます。
- C. 各APIエンドポイントごとに個別のApexクラスを使用して統合ロジックをカプセル化します。
- D. AFL統合用の再利用可能なApexクラスを作成し、関連するApexクラスからそれを呼び出します。

正解: [A \(コメントを发表する\)](#)

有効的な**PDII**問題集はJPNTest.com提供され、**PDII**試験に合格することに役に立ちます！JPNTest.comは今最新**PDII**試験問題集を提供します。JPNTest.com PDII試験問題集はもう更新されました。ここで**PDII**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/PDII-mondaishu> **163**問、**30%ディスカウント**、特別な割引コード：**JPNshiken**」

質問: **62**

sendEmail メソッドの呼び出し回数に関するトランザクション制限はいくつですか？

- A. 20
- B. 10
- C. 50
- D. 100
- E. 制限はありません

正解: ([正解を表示します](#))

説明

説明／参考資料：

reserveEmailCapacity メソッドを使用すると、実際に送信する前に送信したいメールの数を宣言できるため、制限エラーを早期に処理できます。

質問: **63**

System.Limit クラス内で、1 つのトランザクションで呼び出すことができる合計制限を取得するには、どのようなメソッドを呼び出しますか？

- A. get[typeOfLimit] --> (例: getDMLStatements())
- B. getLimit[typeOfLimit] --> (例: getLimitDMLStatements())

正解: **B** ([コメントを發表する](#))

質問: **64**

Universal Containers社は、夜間に実行されるApexバッチジョブで未処理の例外が発生した場合に、外部システムに通知したいと考えています。

この要件を満たすための適切なパブリッシュ/サブスクライブロジックは何ですか？

- A. 外部システムが、addError{で発火するカスタムプラットフォームイベントを購読するようにします。
- B. 外部システムが標準規格を購読する

Eventbus.publish(1.

- C. 外部システムが、発生する標準プラットフォームイベントを購読するようにします。
- D. 外部システムにカスタムを購読させる

EventBus.publish(1,

正解: ([正解を表示します](#))

質問: **65**

ある企業は、ERPシステムで顧客からの注文を受け付けており、それらの注文はSalesforceにOrder__cレコードとして統合され、Accountへのルックアップフィールドを持つ必要があります。Accountオブジェクトには、外部IDフィールドであるERP_Customer_ID__cがあります。統合では、正しいAccountに自動的に関連付けられる新しいOrder__cレコードを作成するために、何を使用すべきでしょうか？1234

- A. Order__c オブジェクトに対して Upsert を実行し、アカウント関係に ERP_Customer_ID__c を指定します。5678
- B. アカウントに対してアップサートを実行し、関係9hip.101112に対してERP_Customer_ID__cを指定します。

- C. Order__c オブジェクトでマージし、アカウント関係に ERP_Customer_ID__c を指定します。
- D. Order__c オブジェクトへの挿入に続いて、Order__c オブジェクトの更新が行われます。

正解: [\(正解を表示します\)](#)

Salesforceとの連携において、Upsert操作は非常に強力です。なぜなら、SalesforceレコードID \$001...\$)の代わりに外部IDを使用してレコードを作成または更新し、他のレコードに関連付けることができるからです。

ERPシステムが注文を送信する際、アカウントの18文字のSalesforce IDは分からないかもしれませんが、ERP_Customer_ID__cは分かります。Order__cオブジェクトに対してUpsertを実行することで オプションA)、統合はAPIペイロード内の 外部キー」構文を活用できます。これは基本的にSalesforceに対して、 亡の注文を作成し、そのAccount__cルックアップフィールドを、この特定のERP_Customer_ID__cを持つアカウントにリンクしてください」と指示します。このアプローチは、統合における ボールドスタンダード」です。理由は以下のとおりです。

- * レコードを作成する前に、外部システムがSalesforceでIDを検索するための ルックアップ」クエリを実行する必要がなくなります。
- * API呼び出しの総数を削減し、ガバナー制限を節約します。
- * 冪等性」を処理し、同じ注文が2回送信された場合でも、重複させるのではなく更新することを保証します。

オプションB、C、Dは効率が劣ります。アカウントをアップサートする オプションB)だけでは注文は作成されません。マージする オプションC)のは重複排除のためです。挿入してから更新する オプションD) 方法はアンチパターンであり、必要なAPI呼び出しとトランザクションのオーバーヘッドが倍増します。

質問: 66

以下のコンポーネントコードと要件を参照してください。

HTML

```
<lightning:layout multipleRows="true">
<lightning:layoutItem size="12">{!v.account.Name}</lightning:layoutItem>
<lightning:layoutItem size="12">{!v.account.AccountNumber}</lightning:layoutItem>
<lightning:layoutItem size="12">{!v.account.Industry}</lightning:layoutItem>
</lightning:layout>
```

要件：

- * モバイル端末の場合、情報は3行で表示される必要があります。
- * デスクトップパソコンとタブレット端末では、情報は1行で表示される必要があります。

要件2が期待どおりに表示されません。デスクトップおよびタブレットの要件を満たす正しいコンポーネントコードを持つオプションはどれですか？

A. HTML

```
<lightning:layout multipleRows="true">
<lightning:layoutItem size="12" mediumDeviceSize="6">{!v.account.Name}</lightning:layoutItem>
<lightning:layoutItem size="12" mediumDeviceSize="6">{!v.account.AccountNumber}</lightning:
レイアウトアイテム>
<lightning:layoutItem size="12" mediumDeviceSize="6">{!v.account.Industry}</lightning:layoutItem>
</lightning:layout>
```

B. <lightning:layout multipleRows="true"></lightning:layout>1213

C. 1415

HTML

```
<lightning:layout multipleRows="true">
<lightning:layoutItem size="12" largeDeviceSize="4">{!v.account.Name}</lightning:layoutItem>
<lightning:layoutItem size="12" largeDeviceSize="4">{!v.account.AccountNumber}</lightning:
レイアウトアイテム>
```

```
<lightning:layoutItem size="12" largeDeviceSize="4">{!v.account.Industry}</lightning:layoutItem>
</lightning:layout>
D. HTML
<lightning:layout multipleRows="true">
<lightning:layoutItem size="12" mediumDeviceSize="4" largeDeviceSize="4"> {!v.account.Name} <
/lightning:layoutItem>
<lightning:layoutItem size="12" mediumDeviceSize="4" largeDeviceSize="4"> {!v.account.
アカウント番号}</lightning:layoutItem>
<lightning:layoutItem size="12" mediumDeviceSize="4" largeDeviceSize="4"> {!v.account.Industry}
</lightning:layoutItem>
</lightning:layout>
```

正解: [\(正解を表示します\)](#)

Salesforce Auraコンポーネントでレスポンシブレイアウトを実現するために、lightning:layoutコンポーネントとlightning:layoutItemコンポーネントは12列のグリッドシステムを使用します。size属性はデフォルトの動作を定義し、最小サイズのデバイス (モバイル) を対象とします。元のコードでは、size="12"が3つのアイテムすべてに適用されています。各アイテムは12列すべてを占めるため、3行に縦に積み重ねられます。

要件 2 (タブレットとデスクトップで単一の行に表示) を満たすには、3 つのアイテムが共有する必要があります。

12列の水平方向のスペース。数学的には、 $12 \div 3 = 4$ 。したがって、各アイテムには次のサイズを割り当てる必要があります。

4 より大きな画面のブレイクポイント。

オプションDが正しい実装です。モバイル端末ではサイズを12に維持し (要件)、mediumDeviceSize (タブレット) とlargeDeviceSize (デスクトップ) を4に設定します。これにより、スマートフォンよりも大きな画面では、3つのアイテムが1行に横並びで表示されるようになります。

オプションAは、mediumDeviceSize="6"では1行に2つのアイテムしか表示できないため ($6 + 6 = 12$)、3つ目のアイテムが2行目に強制的に移動してしまうため、不適切です。オプションCは、largeDeviceSizeのみを指定しているため、タブレットが積み重ねられた状態になる可能性があり、不完全です。オプションDは、すべての関連デバイスカテゴリのスパンを明示的に定義することで、一貫したユーザーエクスペリエンスを確保するというベストプラクティスに従っています。

質問: 67

開発者は、非同期プロセスを含むトリガーが正常に実行されたことをどのように確認すべきでしょうか？

- A. テストクラスで全てのテストデータを作成し、system.runs{}を使用してトリガーを呼び出し、アサーションを実行します。
- B. レコードをSalesforceに挿入し、seeAllData=trueを使用して、アサーションを実行します。
- C. すべてのテストデータを作成し、rest クラスで future を使用し、アサーションを実行します。
- D. テストクラスですべてのテストデータを作成し、Test.startTest{} と Test.etopTaata{} を呼び出して、アサーションを実行します。

正解: [\(正解を表示します\)](#)

Test.startTest() と Test.stopTest() を使用してください。これらのメソッドは、テスト実行中に非同期コードが実行されることを保証し、正確なアサーションを可能にします。参考資料: 非同期 Apex のテスト

質問: 68

Universal Containersは、カスタムLightningページを使用して、アカウントを段階的に検索するウィザード機能を提供しています。ウィザードの手順の一つとして、ユーザーがテキストフィールド ERP_Number__cにテキストを入力できるようにし、そのテキストをクエリに使用して一致するアカウントを検索します。

Java

```
erpNumber = erpNumber + '%';
```

```
List<Account> accounts = [SELECT Id, Name FROM Account WHERE ERP_Number__c LIKE :
```

erpNumber];

開発者が「SOQLクエリの選択性が不十分です」という例外を受け取りました。この問題を解決するには、どのような手順を踏むべきでしょうか？

- A. SOQLクエリを非同期プロセス内に移動します。
- B. ERP_Number__c フィールドを必須項目としてマークします。
- C. ERP_Number__c フィールドを外部 ID としてマークします。
- D. クエリをSOQLではなくSOSLステートメントを使用するように変更します。

正解: ([正解を表示します](#))

SOQLクエリの選択性が不十分です」エラーは、大量のレコード（通常0万件以上）を含むオブジェクトに対するクエリで、結果セットを効果的に絞り込むためのフィルタリングされたインデックスが使用されていない場合に発生します。Salesforceでは、パフォーマンスを維持するために、クエリ最適化ツールがインデックスを使用してレコード全体のごく一部をスキャンする必要があります。クエリがインデックスのないフィールドでフィルタリングされると、システムは強制的にフルテーブルスキャンを実行することになり、パフォーマンスの低下を防ぐためにこの例外が発生します。

この問題を解決するには、開発者は WHERE 句で使用されるフィールドにインデックスが付けられていることを確認する必要があります。カスタム フィールドを「外部 ID」または「一意」としてマークすると、そのフィールドにカスタム インデックスが自動的に作成されます。ERP_Number__c を外部 ID としてマークすると、プラットフォームは基となるデータベース インデックスを生成し、クエリ オプティマイザがテーブル全体をスキャンすることなく、一致するレコードをすばやく見つけることができるようになります。末尾にワイルドカード (例: 'Text%') が付いた LIKE 演算子はインデックスを利用できませんが、フィルターによって結果がプラットフォームの選択性しきい値 (通常、最初の 100 万レコードの 10%) を下回る場合にのみ「選択的」とみなされます。

フィールドを外部 ID にすることは、選択性に必要なインデックスを提供する標準的なプログラムによる解決策です。

選択肢Aは誤りです。非選択的なクエリを非同期プロセスに移動しても、それが非選択的であるという事実は変わりません。そのため、クエリは依然として失敗します。

質問: 69

セーブポイントに関して、正しい記述はどれですか？

- A. 任意の順序で作成されたセーブポイント変数にロールバックできます。
- B. ロールバック中に静的変数は元に戻されません。
- C. セーブポイントへの参照はトリガー呼び出しをクロスできます。
- D. セーブポイントは、DML ステートメントのガバナー制限によって制限されません。

正解: ([正解を表示します](#))

セーブポイントまでロールバックすると、そのセーブポイント以降に発生したDML操作はすべて取り消されますが、静的変数は元に戻されません。したがって、ロールバックが発生すると、ロールバック時のすべての静的変数の状態は、ロールバック直前の状態と同じままになります。参照 Apex開発者ガイド - セーブポイントとトランザクション制御

質問: 70

Dynamic Apexに関する以下の記述のうち、誤っているものはどれですか？

- A. .getObjectType() を呼び出すことで、ID から sObject のタイプを取得できます。
- B. getDescribe() を使用すると、特定のオブジェクト/フィールドに関するさまざまな情報を取得できます。
- C. Schema.getGlobalDescribe() は、すべての sObject のマップを取得します
- D. 動的SOQLでは、バインド変数とバインド変数フィールドを使用できます。

正解: ([正解を表示します](#))

動的SOQLでは単純なバインド変数を使用できますが、バインド変数フィールド（例myVariable.field1_c）は使用できません。SOQLインジェクションを防ぐには、escapeSingleQuotesを使用してください。

質問: 71

ある組織では、連絡先と取引先の住所を保存する際に、Apexコードを使用して会社標準に正規化する必要があるという要件があります。これを実装する最適な方法がありますか？

- A. 連絡先とアカウントのApexトリガーで住所を正規化します
- B. アカウントのApexトリガーが連絡先トリガーを呼び出し、住所を正規化する
- C. 取引先責任者のApexトリガーがアカウントトリガーを呼び出し、住所を正規化する
- D. 連絡先とアカウントの Apex トリガーがヘルパー クラスを呼び出し、アドレスを正規化します

正解: [\(正解を表示します\)](#)

ソフトウェアエンジニアリングにおいて、DRY (Don't Repeat Yourself : 繰り返しを避ける) 原則は保守性を確保する上で不可欠です。住所を正規化するロジックがアカウントと連絡先で同じであれば、そのロジックを2つの別々のトリガーファイルに重複して記述すべきではありません。

最適な実装方法 (オプションD)は、単一のトリガーヘルパーまたはユーティリティクラスを作成することです。このクラスには、正規化ロジックを実行する静的メソッド (例 : `AddressService.normalize(List<sObject> records)`)が含まれています。次に、AccountオブジェクトとContactオブジェクトにそれぞれ小さなトリガーを作成し、どちらもこの共有ヘルパーメソッドを呼び出すようにします。

このアプローチにはいくつかの利点があります。

* 保守性: 住所の会社標準が変更された場合 (たとえば、`St.`を `Street`に変更する場合)、開発者はコードを 1箇所だけ更新すれば済みます。

* 再利用性: ヘルパーメソッドは、匿名 Apex スクリプトやバッチジョブなど、他のコンテキストからも呼び出すことができます。

* 可読性: トリガーは「簡素」で読みやすく、複雑なビジネスロジックを含むのではなく、エントリーポイントとしてのみ機能します。

オプションBとCは技術的に不可能です。あるオブジェクトのトリガーから別のオブジェクトのトリガーを直接呼び出すことはできません。オプションAは、コードの重複や、一方のトリガーのロジックが更新されても他方のトリガーで更新が忘れられた場合にバグが発生するリスクが高まるため、最適とは言えません。

質問: 72

ある企業は、顧客への売上を追跡するために商談 (Opportunities)を使用しており、組織内には数百万件の商談が存在します。同社は、関連する収益オブジェクトを通じて、時間の経過に伴う収益の追跡を開始したいと考えています。

初期導入の一環として、複雑なロジックに基づいて、商談の収益レコードを自動的に作成・入力することで、データの初回投入を行いたいと考えている。

彼らの推定では、およそ10万件の商談において収益記録が作成され、入力される見込みだ。

これを自動化する最適な方法は何ですか？

- A. データベースを使用します。=executeBatch() を使用して、データベースのバッチ可能なクラスを呼び出します。
- B. Database.executeBatch() を使用して Queueable クラスを呼び出します。
- C. system、scheduleJob() を使用して、System.schedule クラスのスケジュールを設定します。
- D. システム、enqueueJob() を使用して、バッチ可能なクラスを呼び出します。

正解: [A \(コメントを発表する\)](#)

質問: 73

開発者は、商談オブジェクトにトリガーを作成し、商談が成立するたびに、商談のアカウントにあるカスタムフィールド「最終販売日」を更新するようにしました。しかし、そのトリガーのテストクラスでは、「最終販売日」フィールドを検証するアサーションが失敗します。

アサーションが失敗した原因は何でしょうか？

- A. テストクラスは、テストメソッド seeAllData=true を実装していません。
- B. テストクラスは、商談を更新した後、アカウントレコードを再照会していません。
- C. テストクラスは、テストデータを挿入する際にアカウント所有者を定義していません。
- D. テストクラスは、Salesforce管理者としてテストを実行するためにSystem.runAs()を使用していません。

正解: [\(正解を表示します\)](#)

質問: 74

開発者が作成したアプリケーションイベントが無限ループを引き起こしました。この問題の原因は何でしょうか？

- A. このイベントはカスタムレンダラーから発生します。
- B. イベントが「touchend」で発生し、処理されませんでした。
- C. このイベントには、プロジェクトに登録されている複数のハンドラーがあります。
- D. イベントハンドラがトリガーを呼び出します。

正解: (正解を表示します)

質問: 75

バルクセーフなApexトリガーを作成する際のベストプラクティスと考えられる記述はどれですか？

- A. DML ステートメントの代わりに、allOrNone を False に設定したデータベース メソッドを使用します。
- B. すべてのSOQLステートメントにLIMIT 50000を追加する
- C. レコードをコレクションに追加し、これらのコレクションに対してDML操作を実行します。
- D. Future メソッド内からすべての DML 操作を実行します。

正解: C (コメントを発表する)

質問: 76

開発者は、保存時にアカウントの所有権をキューに自動的に割り当てるVisualforceページを持っています。

ページ上では所有権が正しく割り当てられているように見えますが、所有権の正当性を検証するアサーションが失敗します。この問題の原因は何でしょうか？

- A. テストクラスは Queueable インターフェースを実装していません。
- B. テストクラスはデータベースから更新された値を取得しません。
- C. テストクラスは seeAllData=true アノテーションを使用していません。
- D. テストクラスは、テストデータの読み込みにBulk APIを使用しません。

正解: (正解を表示します)

Apexの単体テストでよくある間違いは、メモリ内のSObject変数が、トリガー、フロー、またはその他のロジックによってデータベースに加えられた変更を自動的に反映しないことを認識していないことです。開発者がテスト4でアカウントレコードを保存すると、データベースが更新されます。たとえば、トリガーまたはコントローラがOwnerIdをキューIDに変更します)。しかし、テストで使用される変数には、最初にインスタンス化または挿入されたときに存在していた元のデータがまだ保持されています。

変更を正しく検証するには、開発者は「更新」クエリを実行する必要があります。これには、SOQL ステートメントを使用して、ID でデータベースからレコードを取得する処理が含まれます。例:

Account updatedAcc = [SELECT OwnerId FROM Account WHERE Id = :acc.Id];。その後、updatedAcc.OwnerId に対して検証を実行する必要があります。

オプションBは、これが根本原因であることを正しく特定しています。オプションAは、Queueableが非同期実行パターンであるため、無関係です。オプションC (seeAllData=true)は不適切な方法であり、レコードを変更するロジックのテストには必要ありません。オプションDは、アサーションロジックとは無関係です。標準的なテスト分離では、プラットフォームの自動化またはコントローラロジックによって引き起こされる副作用を確認するために、データベースを明示的に再クエリする必要があります。

有効的なPDII問題集はJPNTest.com提供され、PDII試験に合格することに役に立ちます！JPNTest.comは今最新PDII試験問題集を提供します。JPNTest.com PDII試験問題集はもう更新されました。ここでPDII問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/PDII-mondaishu> 163問、30%ディスカウント、特別な割引コード:

JPNshiken」

質問: 77

Salesforceユーザーは、アカウントを保存する際に「最大トリガー深度を超えました」というエラーを頻繁に受け取ります。

開発者はこのエラーをどのように修正すればよいでしょうか？

- A. トリガーロジックを2つの別々のトリガーに分割します。
- B. トリガーを 3suzure アノテーションを使用するように変換し、後続のトリガー呼び出しを Account オブジェクトにチェーンします。
- C. トリガーを変更して L=MultiThread=true アノテーションを使用するようにし、
- D. ヘルパークラスを使用して、トリガーが最初に起動したときにブール値をTRUEに設定し、その後、ブール値がFALSEの場合にのみトリガーが起動するように変更します。

正解: (正解を表示します)

トリガーの最大深度を超えました」エラーに対する一般的な解決策は、ヘルパークラスで静的変数を使用することです。この変数はスイッチとして機能し、トリガーロジックがトランザクションごとに一度だけ実行されるようにすることで、トリガーの再帰呼び出しを防ぎます。

参考資料 :Salesforce開発者ブログ - 再帰トリガー呼び出しの回避

質問: 78

Apexトリガーは、営業担当者が商談を獲得するたびにOrder__cレコードを作成します。最近、このトリガーが2つの注文を作成しています。開発者がこの問題をトラブルシューティングする最適な方法は何ですか？

- A. 各営業担当者ごとにデバッグログを設定し、ログを監視してエラーや例外を検出します。
- B. すべてのワークフロー ルールをオフにしてから、1 つずつオンにして、どのルールがエラーの原因になっているかを確認します。
- C. コードに system.debug() ステートメントを追加し、開発者コンソールのログを使用してコードをトレースします。
- D. ApexトリガーのApexテストクラスを実行して、コードに十分なコードカバレッジが確保されていることを確認します。

正解: C (コメントを発表する)

質問: 79

開発者が、外部のWebサービスへの呼び出しを必要とするコードを記述している。

どのシナリオで、コールアウトを非同期メソッドで行う必要がありますか？

- A. 呼び出しの完了には60秒以上かかる場合があります。
- B. コールアウトはApexトリガーで行われます。
- C. コールアウトは REST APL を使用して行われます。
- D. 1回の取引で10件以上の呼び出しが行われます。

正解: (正解を表示します)

質問: 80

展示資料を参照してください

```

opportunity">
    <apex:input label="opportunity name"/>
    <apex:commandButton value="search"
action="{!search}" />
</apex:pageBlockSection>
<apex:pageBlockSection title="Opportunity
List" id="opportunityList">
    <!-- DATA Table -->
</apex:pageBlockSection>
</apex:pageBlock>
</apex:form>
</apex:page>

```

このVisualforceページのユーザーから、検索ボタンが押されるたびにページが完全に更新されるという苦情が寄せられています。

開発者は、部分的な更新が行われ、機会リストに関連付けられたセクションのみが画面上に再描画されるようにするには、どのような手順を踏むべきでしょうか？

- A. DATA テーブルを <apex:actionRegion> タグで囲みます。
- B. <apex:commandButton> タグに render 属性を実装します。
- C. アクションメソッドの検索がnullを返すことを確認します。
- D. <apex:actionfunction> タグを immediate = true で実装します。

正解: ([正解を表示します](#))

開発者は、apex:commandButton タグに render 属性を実装して、部分的な更新が行われるようにし、opportunity List で識別されるセクションのみが画面に再描画されるようにする必要があります。render 属性は、AJAX 更新リクエストの結果がクライアントに返されたときに再描画される 1 つ以上のコンポーネントの ID を指定します。開発者は render 属性を使用して、更新の範囲をデータ テーブルを含む apex:pageBlock コンポーネントに限定し、ページ全体の更新を回避できます。apex:actionRegion タグでデータ テーブルを囲んでも効果はありません。apex:actionRegion タグは、フォーム内のフォームを区切るページのセクションを定義し、開発者がフォーム要素の処理をその領域の内容に限定できるようにするためです。ただし、ページやコンポーネントの更新には影響しません。アクション メソッド search が null を返すようにしても効果はありません。これはページの再読み込みを防ぐだけで、新しい検索結果でデータ テーブルを更新しないためです。apex:actionFunction タグに immediate = true を指定して実装しても効果はありません。apex:actionFunction タグは、コントローラのアクション メソッドを呼び出すことができる JavaScript 関数を定義しており、immediate 属性は、検証ルールを処理せずにアクション メソッドを即座に実行するかどうかを指定するためです。ただし、ページやコンポーネントの更新には影響しません。参照: [apex:commandButton]、[apex:actionRegion]、[apex:actionFunction]、[Visualforce 開発者ガイド]

質問: 81

「sendEmail」メソッドの呼び出し回数に関するトランザクション制限はいくつですか？

- A. 20
- B. 10
- C. 50

D. 100

E. 制限はありません

正解: **B** ([コメントを發表する](#))

参照 :

reserveEmailCapacity メソッドを使用すると、実際にメールを送信する前に送信したいメールの数を宣言できます。

送信することで、制限エラーを早期に処理できるようになります

質問: **82**

Universal Containers は Convention_Attendence__c カスタムオブジェクトにプライベート共有モデルを実装しています。新しい品質保証活動の一環として、同社はオブジェクト上に

Event__Reviewer__c ユーザー検索フィールドを作成しました。経営陣はイベントレビュー担当者に自動的に

割り当てられたすべてのレコードに対する読み取り/書き込みアクセス権を取得する。

担当レビュー担当者がレコードへの読み取り/書き込みアクセス権を確実に取得するための最善策は何ですか？

A. Convention Attendee カスタム オブジェクトに After Insert トリガーを作成し、Apex Sharing Reasons と Apex Managed Sharing を使用します。

B. コンベンション参加者カスタムオブジェクトに基準ベースの共有ルールを作成し、レコードをイベントレビュー担当者のグループと共有します。

C. コンベンション参加者カスタムオブジェクトに基準ベースの共有ルールを作成し、レコードをイベントレビュー担当者と共有します。

D. Convention Attendee カスタム オブジェクトに Before Insert トリガーを作成し、Apex Sharing Reasons と Apex Managed Sharing を使用します。

正解: ([正解を表示します](#))

質問: **83**

開発者が、フィールド履歴追跡に依存する asccuntRisteryManager というクラスを作成しました。このクラスには、Account をパラメータとして受け取り、関連付けられた accountHistory オブ

ジェクトレコードのリストを返す getaccountHistory という静的メソッドがあります。

以下のテストが失敗しました。


```

@isTest
public static void testAccountHistory() {
    Account a = new Account(name = 'test');
    insert a;
    a.name = a.name + '1';
    update a;
    List<AccountHistory> ahList =
AccountHistoryManager.getAccountHistory(a);
    Assert.assertEquals(ahList.size(), 1);
}

```

このテストに合格するにはどうすれば良いでしょうか？

- A. テスト設定でアカウント履歴レコードを手動で作成し、それらを取得するクエリを作成します。
- B. @isTest (SeeAllData=true) を使用して組織の履歴データを表示し、AccountHistory レコードをクエリします。
- C. getAccountHistory() で Test.isRunningTest() を使用して、条件付きで偽の AccountHistory レコードを返します。
- D. このコードはテストできないため、タクトメソッドは削除する必要があります

正解: [\(正解を表示します\)](#)

テストをパスさせるには、@isTest(SeeAllData=true) を使用して、テストが組織の履歴データにアクセスできるようにする必要があります。これは、フィールド履歴追跡データがテストコンテキストにコピーされないため、テストがライブデータにアクセスして検証できるようにする必要があります。参照: Apex 開発者ガイド - IsTest アノテーション

質問: 84

カスタム開発の一環として、開発者は特定の商談が時間の経過とともにどのように進展するかを示す Lightning コンポーネントを作成します。このコンポーネントは、以下のいずれかのフィールドが変更されたときに日付スタンプを表示する必要があります。

* 金額、確率、ステージ、または完了日

このような情報を照会する最も効率的な方法は何ですか？

- A. [Select Amount CloseDate, StageName, Probability FROM Opportunity_History WHERE OpportunityId = :oppId];
- B. [Select Newvalue, Oldvalue From OpportunityField_History Where Opportunity = oppid AND Field IN
- C. [Select NewValue, Oldvalue FROM Opportunity FieldHistory WHERE OpportunityId = :oppId AND Field IN
- D. ('ステージ名', 確率', 金額 / 終了日)];
- E. [Select Amount, CloseDate, StageName, Probability FROM OpportunityHistory WHERE OpportunityId = :oppId];

正解: [\(正解を表示します\)](#)

質問: 85

開発者が子コンポーネントを含む Lightning Web コンポーネントを作成しています。プロパティ stage は親コンポーネントから子コンポーネントに渡されます。パブリック プロパティは変更されますが、setOppList 関数が呼び出されません。

```
@api stage;
opp;
connectedCallback() {
  this.opp = this.setOppList(this.stage);
}
```

これを実現するために、開発者は何を改ざんすべきでしょうか？

- A. ロジックをゲッター/セッターのペアに移動します。
- B. 親コンポーネントからカスタムイベントを作成してプロパティを設定します。
- C. connectedCallback() のロジックを constructor() に移動します。
- D. connectedCallback() のロジックを renderedCallback() に移動します。

正解: [\(正解を表示します\)](#)

質問: 86

Cloud KicksのSalesforce管理者は、米国のすべての郵便番号と、その郵便番号が属するCloud Kicksの販売地域を格納するために、Region_cというカスタムオブジェクトを作成しました。

Object Name:

salesforce

Region__c

Fields:

zip_code__c (Text)

Region_Name__c (Text)

Cloud Kicksは、リードの郵便番号に基づいて地域情報を入力するトリガーをリードに設定したいと考えています。この要求を満たす最も効率的な方法は、どのコードセグメントを使用するか？

```

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

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

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

```

A.

```

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

```

B.

```

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

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

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

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

```

salesforce

C.



salesforce

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

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

D.

正解: ([正解を表示します](#))

質問: 87

開発者は、カスタムプロファイルに割り当てられたユーザーが、

「監査担当者」は、Account オブジェクトの履歴レコードに対して検索を実行します。開発者は、検索によって 6 ヶ月から 12 ヶ月前の履歴レコードが返されるようにする必要があります。以下のコードの場合、Account 履歴レコードを取得する有効な方法として、どの select ステートメントを挿入する必要がありますか?

```
4445 Java Date initialDate = System.Today().addMonths(-12);
Date endDate = System.Today().addMonths(-6);
```

// ここにSELECT文を挿入

- A. [SELECT AccountId, CreatedDate, Field, NewValue, OldValue FROM Account_History WHERE CreatedDate <= :initialDate AND CreatedDate <= :endDate];
- B. [SELECT AccountId, CreatedDate, Field, NewValue, OldValue FROM Account_History WHERE CreatedDate >= :endDate AND CreatedDate <= :initialDate];
- C. [SELECT AccountId, CreatedDate, Field, NewValue, OldValue FROM AccountHistory WHERE CreatedDate BETWEEN :initialDate AND :endDate];
- D. [SELECT AccountId, CreatedDate, Field, NewValue, OldValue FROM AccountHistory WHERE CreatedDate >= :initialDate AND CreatedDate <= :endDate];

正解: ([正解を表示します](#))

Accountのような標準オブジェクトの履歴レコードを照会するには、開発者は正しいAPI名であるAccountHistory (Account_Historyではない)を使用する必要があります。日付範囲でフィルタリングする場合、SOQLにはBETWEEN演算子が用意されており、クエリをより簡潔で読みやすくすることができます。オプションCは、正しいオブジェクト名とBETWEEN構文の両方を使用しているため、最も効率的なソリューションです。

SOQLでは、CreatedDate BETWEEN :initialDate AND :endDateは、CreatedDate >= :initialDate AND CreatedDate <= :endDateの省略形です。提供されたコードでは、initialDateは12か月前（古い日付）、endDateは6か月前（新しい日付）です。したがって、この範囲は、これら2つのタイムスタンプ間のレコードを正しく取得します。

オプションAとBは、誤ったオブジェクト名 Account_History)を使用しています。オプションDは、誤った比較演算子=>を使用しています（正しい演算子は>=です）。さらに、オプションCは、標準のAccountHistoryオブジェクトにバインド変数を使用することで、Apexにおける日付範囲クエリのベストプラクティスに従っており、監査担当者」プロファイルがプラットフォームの監査およびセキュリティ制約内で必要な履歴データを取得できるようにしています。

質問: 88

ある企業は、前提となる参照データのために多数のレコードを作成するメソッドを多数含むテストクラス内の単体テストの実行速度が遅いことに気づいた。開発者はこの問題にどう対処すればよいでしょうか？

- A. テスト実行時は、トリガー、フロー、検証をオフにしてください。
- B. 前提条件となる参照データの設定を TestDataFactory に移動し、各テストメソッドからそれ呼び出します。
- C. 前提条件となる参照データの設定をテストクラスの @testSetup メソッドに移動します。
- D. 前提条件となる参照データの設定をテストクラスのコンストラクタに移動します。

正解: ([正解を表示します](#))

@testSetup メソッドはテストのパフォーマンスを向上させます。クラス内のテストメソッド間でデータを共有できるため、冗長なデータ生成を削減できます。

参考資料 testSetupメソッドの使用

質問: 89

<apex:actionRegion> の正しい定義を選択してください。

- A. 指定した時間間隔でAJAXリクエストを送信します。再レンダリングされた場合はリセットされます。
- B. コントローラーメソッドをJavaScriptから直接呼び出すことができます。<apex:form>タグで囲む必要があります。actionSupportとは異なり、これらの関数<apex:actionPoller>はJavaScriptコードから直接呼び出すことができます。
- C. 他のコンポーネントにAJAXサポートを追加します（例onClick、onMouseUp、onFocusなど）。
- D. AJAXリクエストが生成された際に、サーバーが処理すべきコンポーネントを示します。
- E. AJAXリクエスト {actionFunction/actionSupport/actionPoller)に関連付けることができ、リクエストの状態（進行中完了)に応じて条件付きでコンテンツを表示します。Id」フィールドを使用して名前を指定し、関連コンポーネントの {status」フィールドを使用してそれらを接続します。

正解: D ([コメントを發表する](#))

質問: 90

Universal Containers (UC) は翻訳ワークベンチを有効にし、選択リストの値を翻訳しました。UC は、アカウントオブジェクトにカスタムの複数選択選択リストフィールド Products__z を用意しており、営業担当者はアカウントが既に所有している UC 製品を指定できます。開発者は、product_c フィールドを含むアカウントレコードを取得する Apex メソッドを作成するよう指示されています。

開発者は、Products__c の値が現在のユーザーの言語であることを保証するために、どのような対策を講じるべきでしょうか？

- A. SOQL 結果リストの各レコードに対して translate() メソッドを呼び出します。
- B. SOQLクエリでロケール句を使用します。
- C. SOQLクエリのフィールドリストでtoLabel1 Products_cを使用します。
- D. SOQL結果リストの各レコードにロケールを設定します。

正解: ([正解を表示します](#))

質問: 91

テストカバレッジの作成に必要な時間を短縮するため、開発者は静的リソースとしてアップロードしたスプレッドシートを使用して、テストメソッドにテストデータを提供したいと考えています。開発者はどのようなコードを使用すればよいでしょうか？

- A. List< sObject> testAccounts = Test . loadData([Select Id,Body from StaticResource Where Name =テストアカウント
- B. < meta charset="utf-8" />List< sObject> testAccounts = Test . loadData (Account.sObjectType, ' testAccounts');
- C. List< sObject> testAccounts = (List< Count>) [Select Id, Body from StaticResource Where Name = testAccounts
- D. List< sObject> testAccounts = Test . loadData (Account .sObjectType, SResource. testAccounts)

正解: B ([コメントを发表する](#))

有効的なPDII問題集はJPNTTest.com提供され、PDII試験に合格することに役に立ちます！JPNTTest.comは今最新PDII試験問題集を提供します。JPNTTest.com PDII試験問題集はもう更新されました。ここでPDII問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/PDII-mondaishu> 163問、30%ディスカウント、特別な割引コード:

JPNshiken」

質問: 92

開発者は、ケースがクローズされたときに、そのケースに対して単一の調査レコードが作成されるように、次のApexトリガーを作成しました。問題は、ケースごとに複数のSurvey_cレコードが作成されてしまうことです。

```
trigger CaseTrigger on Case (after insert, after update){ List<Survey_c>
createSurveys = new List<Survey_c>(); for (Case c : trigger.new){ if
(c.IsClosed && (trigger.isInsert II trigger.isUpdate && trigger.oldMap.get
(c.Id).IsClosed == false)){ createSurveys.add(new Survey_c(Case_c =
c.Id)); } } insert createSurveys; }
```

この問題の原因は何でしょうか？

- A. ワークフロールールがタスク作成アクションで起動しています
- B. ユーザーがレコードを複数回編集している
- C. ユーザーがレコードをクローズ済みとして作成しています
- D. フィールド更新アクションでワークフロールールが実行されます。

正解: ([正解を表示します](#))

質問: 93

開発者は、コンポーネントの読み込みが遅いという苦情を受ける。

開発者は、コンポーネントのパフォーマンスを向上させるために、どのような変更を実装すればよいでしょうか？

- A. 連絡先情報の種類を 地図」に変更します。
- B. <c: concacclInfo の内容をコンポーネントに移動します。
- C. showContacrInfo のデフォルト値を false」に変更します。
- D. showContactinfo の変更イベントハンドラを追加します

正解: [\(正解を表示します\)](#)

質問: 94

以下のRESTメソッドに対する有効なリクエストはどれですか？ 2つ選択してください。）

A. 弦"

il" : 123, "S1" : "私の最初の文字列", "S2" : "私の2番目の文字列", "bl" : false

B. <bl>>false</bl> </request>

sl" : "私の最初の文字列", 11" : "123", "b1" : "false", "S2" : "私の2番目

C. <b1>>false</bl> </request>

<request> <sl>"私の最初の文字列"</sl> <il>123</il> <sZ>"私の2番目の文字列"</32>

D. @HttpPost global static void myPostMethod(String sl, Integer il, Boolean bl, String 52)

<リクエスト> <sl>最初の文字列</sl> <11>123</sl> <32>2番目の文字列</32>

正解: [\(正解を表示します\)](#)

質問: 95

Lightningコンポーネントがプレビューモードで動作しており、Lightningアプリケーションビルダーページ内で使用する必要があるにもかかわらず、利用できません。このコンポーネントにどのような変更を加えるべきでしょうか？

A. コンポーネント、メタデータ、およびApexコントローラーを削除して再作成します。

B. JavaScriptコントローラーのロジックにエラーがないか確認してください。

C. implements属性とaccess属性を使用してマークアップ内で公開します。

D. サンドボックスを更新し、最新のAPIバージョンにアップグレードしてください。

正解: [\(正解を表示します\)](#)

質問: 96

下記の試験方法を参照してください。

salesforce

```
@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);
}
```

このテストメソッドは、外部システムをアカウント情報で更新するWebサービス呼び出し、完了時に「アカウント統合_更新__c」チェックボックスをTrueに設定します。テストは実行に失敗し、「TestMethodとして定義されたメソッドはWebサービス呼び出しをサポートしていません」というエラーで終了します。

- A. Add if (!Test.isRunningTest()) around CalloutUtil.sendAccountUpdate.
- B. Add Test.startTest() and Test.setMock before and Test.stopTest() after CalloutUtil.sendAccountUpdate.
- C. Add Test.startTest() before and Test.setMock and Test.stopTest() after CalloutUtil.sendAccountUpdate.
- D. Add Test.startTest() before and Test.stopTest() after CalloutUtil.sendAccountUpdate.

正解: [\(正解を表示します\)](#)

コールアウトを含むテストにおける正しい実行順序は、Test.startTest()、Test.setMock()、メソッド呼び出し、そして最後にTest.stopTest()です。これにより、モックコールアウトが確実に使用されます。

参考資料: Apex開発者ガイド - HTTPコールアウトのテスト

開発者は、商談がクローズされるたびに商談アカウントのカスタムフィールド「最終販売日」を更新するトリガーを商談オブジェクトに作成しました。しかし、そのトリガーのテストクラスでは、「最終販売日」フィールドを検証するアサーションが失敗します。

アサーションが失敗した原因は何でしょうか？

- A. テストデータを挿入する際に、テストクラスにアカウント所有者が定義されていません。
- B. テストクラスは、Salesforce管理者としてテストを実行するためにSystem.runAs()を使用していません。
- C. テストクラスのテストメソッドにseeAllData=trueが実装されていません
- D. テストクラスは、商談を更新した後、アカウントレコードを再照会していません

正解: [\(正解を表示します\)](#)

質問: 98

しかし、テストを実行してもデータが返されず、アサーションが失敗します。

開発者は、テストクラスが正常に実行されるようにするために、どのような編集を行うべきでしょうか？

- A. searchFeatures Apex クラスに without sharing キーワードを実装します。
- B. @1sTest アノテーションに seeAllData=true 属性を実装します。
- C. メソッド呼び出しを Test.startTest() と @TestSetup で囲みます。
- D. テストクラスに setFixedSearchResult= メソッドを実装します。

正解: [\(正解を表示します\)](#)

テスト クラスで setFixedSearchResult= メソッドを実装することで、テスト クラスが正常に実行されることを保証できます。setFixedSearchResult= メソッドは、テスト クラスで SOSL クエリによって返されるレコードを指定するために使用できるメソッドです。setFixedSearchResult= メソッドは、レコード ID のリストをパラメーターとして受け取り、SOSL クエリの実行時にそれらのレコードを返します。setFixedSearchResult= メソッドを使用することで、開発者は SOSL クエリによって返されるデータを制御し、組織内のデータへの依存を回避できます。開発者は、Test.loadData メソッドを使用して静的リソースからテスト データをロードし、レコード ID を setFixedSearchResult= メソッドに渡すこともできます。たとえば、開発者は次のコードを使用して、テスト クラスで setFixedSearchResult= メソッドを実装できます。

```
// 静的リソースからテストデータをロードします List<sObject> testRecords = Test.loadData(Account.sObjectType, 'testData');
// レコードIDを取得します List<Id> recordIds = new List<Id>(); for (sObject record : testRecords) { recordIds.add(record.Id); }
// 固定検索結果を設定します Test.setFixedSearchResults(recordIds);
// SOSLクエリを実行するメソッドを呼び出す List<Account> results = searchFeatures.search('Acme');
// 期待される結果を検証する System.assertEquals(testRecords.size(), results.size());
```

質問: 99

開発者は、クラス内のすべてのテストメソッドの実行に時間がかかることに気づきました。これは、テストを実行するために必要なすべてのテストデータの初期設定に時間がかかるためです。テストの実行速度を上げるには、開発者は何をすべきでしょうか？

- A. クラス内のテスト方法の数を減らす。
- B. すべてのテスト方法において、テストデータファクトリが適切に使用されていることを確認してください。
- C. テストデータを作成するメソッドを定義し、@testSetup で注釈を付けます。
- D. テストデータを作成するメソッドを定義し、@createData で注釈を付けます。

正解: **C** ([コメントを發表する](#))

質問: 100

企業は、機密情報を7年後に自動的に削除する必要がある。

これは毎日約100万件の記録を削除する可能性がある。

これはどのように実現できるのでしょうか？

A. 7年以上前のレコードを照会する3fuzureプロセスをスケジュールし、その後、1,000レコードのバッチで再帰的に呼び出してそれらを削除します。

B. 集計関数を使用して 7 年以上前のレコードをクエリし、その後、aggregateResults オブジェクトを削除します。

{civ [+] 7 年より前のレコードを照会して削除するバッチ Apex プロセスを毎日実行するようにスケジュールします。

C. SOSLステートメントを実行して7年以上前のレコードを検索し、結果セット全体を削除します。

正解: ([正解を表示します](#))

Batch Apexは、トランザクションをより小さなバッチに分割することで、大規模なデータ処理を効率的に行うように設計されています。

バッチ処理を毎日実行するようにスケジュール設定することで、システムはガバナー制限に達することなく、大量のレコードを効率的に削除できます。これは、定期的に大量削除を実行するためのベストプラクティスです。

参考文献：

バッチApexの使用

質問: 101

開発者は、異なるテキストを表示するためにどのタグを使用すべきでしょうか。

<apex:commandButton> はアクションを処理していますか？

A. <apex:actionSupports

B. <apex:actionPaller>

C. <apex:action>

D. <apex:pageMes sagas

正解: ([正解を表示します](#))

apex:actionSupport は、apex:commandButton がアクションを処理している間に異なるテキストを表示するために使用するタグです。apex:actionSupport は、ボタン、リンク、入力フィールドなどの他のコンポーネントに AJAX サポートを追加するために使用できるコンポーネントです。apex:actionSupport では、実行するアクション、アクションをトリガーするイベント、およびページの一部を更新するための rerender 属性を指定できます。apex:commandButton 内で apex:actionSupport を使用すると、開発者はボタンがアクションを処理している間に 処理中...」や しばらくお待ちください...」などの異なるテキストを表示できます。たとえば、開発者は次のコードを使用して、ボタンがアクションを処理している間に異なるテキストを表示できます。

```
<apex:commandButton value="Submit" action="{!submit}"> <apex:actionSupport event="onclick" rerender="buttonText"/> <apex:facet name="caption"> <apex:outputPanel id="buttonText">
```

```
<apex:outputText value="Processing..." rendered="{!processing}"/> <apex:outputText value="Please wait..." rendered="{!NOT(processing)}"/> </apex:outputPanel> </apex:facet>
```

```
</apex:commandButton>
```

質問: 102

ある企業がREST Webサービスを公開し、SalesforceとREST Webサービス間で双方向SSL接続を確立したいと考えています。適切な認証局によって署名された証明書が開発者に提供されています。

Salesforce側で必要な変更点は何ですか？ 2つ選択してください。）

A. 提供された証明書を使用して二要素認証を設定します。

B. 証明書と鍵の管理で証明書のエントリを作成します。

C. コードを更新して HttpRequest.setClientCertificateName() を使用するようにします。

D. コードを更新して、HttpRequest.setHeaderQ を使用して Authorization ヘッダーを設定します。

正解: ([正解を表示します](#))

質問: 103

以下のコンポーネントコードと要件を参照してください。

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

  <lightning:layoutItem size="12">{!v.account.AccountNumber}
</lightning:layoutItem>

  <lightning:layoutItem size="12">{!v.account.Industry}
</lightning:layoutItem>
</lightning:layout>
```

```
<lightning:layout multipleRows="true">
  <lightning:layoutItem size="12" mediumDeviceSize="6" largeDeviceSize="4">{!v.account.Name}
</lightning:layoutItem>

  <lightning:layoutItem size="12" mediumDeviceSize="6" largeDeviceSize="4">{!v.account.AccountNumber}
</lightning:layoutItem>

  <lightning:layoutItem size="12" mediumDeviceSize="6" largeDeviceSize="4">{!v.account.Industry}
</lightning:layoutItem>
</lightning:layout>
```

A.

```
<lightning:layout multipleRows="true">
  <lightning:layoutItem size="12" mediumDeviceSize="4">{!v.account.Name}
</lightning:layoutItem>

  <lightning:layoutItem size="12" mediumDeviceSize="4">{!v.account.AccountNumber}
</lightning:layoutItem>

  <lightning:layoutItem size="12" mediumDeviceSize="4">{!v.account.Industry}
</lightning:layoutItem>
</lightning:layout>
```

B. </lightning:layout>


```

<lightning:layout multipleRows="true">
  <lightning:layoutItem size="12" largeDeviceSize="4">{!v.account.Name}
</lightning:layoutItem>

  <lightning:layoutItem size="12" largeDeviceSize="4">{!v.account.AccountNumber}
</lightning:layoutItem>

  <lightning:layoutItem size="12" largeDeviceSize="4">{!v.account.Industry}
</lightning:layoutItem>
</lightning:layout>

```

C.

```

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

  <lightning:layoutItem size="12" mediumDeviceSize="6">{!v.account.AccountNumber}
</lightning:layoutItem>

  <lightning:layoutItem size="12" mediumDeviceSize="6">{!v.account.Industry}
</lightning:layoutItem>

  <lightning:layoutItem size="12" mediumDeviceSize="6">{!v.account.AccountNumber}
</lightning:layoutItem>

  <lightning:layoutItem size="12" mediumDeviceSize="6">{!v.account.Industry}
</lightning:layoutItem>
</lightning:layout>

```

D.

正解: ([正解を表示します](#))

質問: 104

Universal Containers社は、顧客サポート案件の管理に独自のSalesforceアプリケーションを使用しています。サポートチームは、特定の案件を解決するために外部パートナーと連携する必要があります。しかし、外部パートナーと共有する案件の可視性とアクセス権限を制御したいと考えています。この要件を満たすために役立つSalesforceの機能は何でしょうか？

- A. 役割階層
- B. 基準に基づく共有ルール
- C. Apex マネージド共有
- D. シェアセット

正解: ([正解を表示します](#))

この要件を満たすために最適な機能は、共有セットです。共有セットを使用すると、開発者は共通のアカウントまたは連絡先に基づいて、外部ユーザーにレコードへのアクセス権を付与できます。この場合、開発者はサポートチームと同じアカウントを持つ外部パートナーにケースへのアクセス権を付与する共有セットを作成できます。ロール階層、条件ベースの共有ルール、およびApex管理共有は、Customer Community Plusライセンスを持つ外部ユーザーには利用できないため、このシナリオには適していません。参照 [共有セット]、[共有アーキテクチャ]、[Trailhead：共有と可視性の基本]

質問: **105**

リアルタイム通知が不要な場合、開発者がユーザーがSalesforceに最後にログインした日時を確認できるSalesforceの機能はどれですか？

- A. カレンダーイベント
- B. 非同期データキャプチャイベント
- C. イベント監視ログ
- D. 開発者ログ

正解: ([正解を表示します](#))

イベント監視ログを使用すると、Salesforce 内のユーザーログインを含むさまざまな種類のイベントを追跡できます。これらのログを使用すると、リアルタイム通知を必要とせずにユーザーの最終ログイン時刻を特定できます。参照: Salesforce ヘルプ - イベント監視

質問: **106**

適切な構文を使用しているリレーションシップクエリはどれですか？ 2つ選択してください。）

SELECT Id, Name, Account__r.Name FROM Contact WHERE Account__r.Industry

- A. SELECT Name, (SELECT LastName FROM Contacts) FROM Account
- B. SELECT Name, (SELECT LastName FROM Contacts__r) FROM Account
- C. 「メディア」
- D. SELECT Id, Name, Account.Name FROM Contact WHERE Account.Industry = 'Media'

正解: ([正解を表示します](#))

有効的な**PDII**問題集はJPNTTest.com提供され、**PDII**試験に合格することに役に立ちます！JPNTTest.comは今最新**PDII**試験問題集を提供します。JPNTTest.com PDII試験問題集はもう更新されました。ここで**PDII**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/PDII-mondaishu> **163**問、**30%ディスカウント**、特別な割引コード：**JPNshiken**」

質問: **107**

以下のコードを参照してください。

Lightning Web コンポーネントのJSファイル

JavaScript

```
import {LightningElement} from 'lwc';
```

```
import serverEcho from '@salesforce/apex/SimpleServerSideController.serverEcho'; export default class Helloworld extends LightningElement { firstName = 'world'; handleClick()
```

```
{ serverEcho({ firstName: this.firstName
```

```
})
```

```
.then((result) => {
```

```
alert('サーバーから: ' + result);
```

```
})
```

```
.catch((error) => {
```

```
console.error(error);
```

```
});
```

```
}
```

```
}
Apexコントローラー
Java
public with sharing class SimpleServerSideController {
    @AuraEnabled
    public static String serverEcho(sObject firstName) {
        String firstNameStr = (String)firstName.get('firstName');
        return ('サーバーからのご挨拶です。' + firstNameStr);
    }
}
```

上記のコードが動作するようにするには、Apexコントローラーでどの2つの変更を行う必要がありますか？

- A. 単一のメソッドだけでなく、クラス全体に `@AuraEnabled` というアノテーションを付けます。
- B. Apexコントローラーの5行目の引数を `sObject` から `String` に変更します。
- C. Apexコントローラーから6行目を削除し、代わりに7行目の `return` で `firstName` を使用します。
- D. メソッドのシグネチャを `public static` ではなく `global static` に変更します。

正解: ([正解を表示します](#))

現在のApexコントローラの実装では、JavaScriptコントローラから送信されるデータとApexメソッドが想定するデータ型との間に型の不一致があります。JavaScriptファイルでは、`this.firstName` が文字列「World」として初期化されています。`serverEcho`メソッドが呼び出されると、この文字列値がキー`firstName`の下に渡されます。

Apex コントローラーでは、パラメーターが `sObject` として誤って定義されています。JavaScript が単純な文字列を渡しているため、Apex メソッドのシグネチャを更新して文字列を受け入れるようにする必要があります (オプション B)。したがって、行

06 (`get()` メソッド `$Object` に固有のメソッド) を使用しようとするもの) はもはや有効ではないため、削除する必要があります。代わりに、開発者は `return` ステートメントで `firstName` 変数を直接使用する必要があります (オプション C)。

オプション A は、`@AuraEnabled` がクラス レベルのアノテーションではなくメソッド レベルのアノテーションであるため、誤りです。オプション D は、`public static` が Lightning コンポーネントに公開される Apex メソッドの十分かつ標準的なアクセス修飾子であるため、誤りです。`global` は、コンポーネントが別の名前空間 (管理パッケージ内など) に存在する場合にのみ必要です。2 つのレイヤー間でデータ型を整合させることで、シリアル化プロセスは JavaScript 文字列を Apex 文字列変数に正しくマッピングできます。

質問: 108

開発者がテストクラスを作成し、モック呼び出しを行う前にテストデータを作成するようにしたところ、`未コミットの作業が保留中です。呼び出しを行う前にコミットまたはロールバックしてください`というエラーが表示されるようになりました。

エラーを解決するには、どの手順を踏むべきですか？

- A. 挿入とモック呼び出しの両方が `l==L` の後に発生するようにします。`stopTest_()`。
- B. レコードが `Test.startTest()` ステートメントの前に挿入され、モック呼び出しが `@testSetup` で注釈されたメソッド内で発生することを確認してください。
- C. `Test.startTest()` の後に挿入とモック呼び出しの両方が行われるようにします。
- D. レコードが `Test.startTest()` ステートメントの前に挿入され、モック呼び出しが `Test.startTest()` の後に行われることを確認してください。

正解: ([正解を表示します](#))

コールアウトを含む Apex テストを実行する際には、`未コミットの作業が保留中`エラーを回避することが重要です。このエラーは通常、コールアウトが試行された時点で DML 操作 (挿入など) が保留中である場合に発生します。Salesforce では、トランザクション内の DML 操作の前にすべてのコールアウトを実行する必要があります。これを回避するには、`Test.startTest()` を呼び出す前にレコードを挿入し、その後にコールアウトを実行するという方法が有効です。

Test.startTest() は、実際のテストが開始されるポイントを示し、ガバナー制限をリセットします。このポイントの後に呼び出しを行うことで、DML 操作がコミットされ、ガバナー制限に関する新しい実行コンテキスト内に入ることが保証されます。

参考文献：

HttpCalloutMockインターフェースを実装してHTTPコールアウトをテストする
実行コンテキストの理解

質問: 109

ページが連絡先に対して nullオブジェクトへの参照解除を試みました」というエラーをスローします。コントローラーのどの変更を加えるとエラーが修正されますか？

- A. getter に条件を使用して、null の場合は新しい Contact を返します。
- B. コントローラの前頭で静的 final Contact を宣言します。
- C. セッターのシグネチャを変更して、Contact を返すようにします。
- D. ゲッターのシグネチャを static Contact に変更します。

正解: ([正解を表示します](#))

質問: 110

どのシナリオで開発者はApexコールアウトを使用する必要があるか
発信メッセージの代わりに？

- A. 対象システムは REST APL を使用します。
- B. 対象システムはSCAP APIを使用します。
- C. コールアウトは非同期である必要があります。
- D. コールアウトはフローから呼び出す必要があります。

正解: ([正解を表示します](#))

開発者がアウトバウンド メッセージングの代わりに Apex コールアウトを使用する必要があるシナリオは、ターゲット システムが REST API を使用している場合です。Apex コールアウトは、Apex コードから外部エンドポイントに HTTP リクエストを送信し、HTTP レスポンスを受信する方法です。Apex コールアウトは、GET、POST、PUT、DELETE などの任意の HTTP メソッドを使用でき、JSON、XML、プレーン テキストなどの任意の形式でデータを送受信できます。Apex コールアウトは、認証、ヘッダー、パラメーター、エラー処理も処理できます。REST API は、REST (Representational State Transfer) の原則に従う Web サービスであり、HTTP メソッドと URI を使用してリソースに対する操作を実行します。REST API は通常、データ形式として JSON を使用し、OAuth や基本認証などのさまざまな認証スキームをサポートします。Apex コールアウトは、適切なメソッド、URI、ヘッダー、パラメーター、およびボディを使用して HTTP リクエストを送信し、ステータス コード、ヘッダー、およびボディを使用して HTTP レスポンスを解析することにより、REST API とやり取りできます。アウトバウンドメッセージングは、Salesforce から外部エンドポイントに SOAP メッセージを送信し、SOAP 確認応答を受信する方法です。アウトバウンドメッセージングはワークフロールールによってトリガーされ、1 つのメッセージに最大 100 個の通知を含めることができます。アウトバウンドメッセージングは、notifications() 呼び出しと WSDL (Web サービス記述言語) を使用して、メッセージ構造とエンドポイント URL を定義します。アウトバウンドメッセージングは、POST 以外の HTTP メソッドをサポートしておらず、XML 以外のデータ形式もサポートしていません。アウトバウンドメッセージングは基本認証のみを処理でき、その他の認証方式はサポートしていません。アウトバウンドメッセージングは REST API と連携できません。REST API は SOAP メッセージ、WSDL、または XML を使用せず、異なる HTTP メソッド、データ形式、および認証方式を必要とする可能性があるためです。したがって、ターゲットシステムが REST API を使用している場合は、開発者はアウトバウンドメッセージングの代わりに Apex コールアウトを使用する必要があります。参照: [Apex Web サービスとコールアウト]、[REST API 開発者ガイド]、[アウトバウンドメッセージング実装ガイド]

質問: 111

部分的なページ更新を実行するためにAjax動作を開始する際に使用できるVisualforceコンポーネントはどれですか？ (3つ選択可能)

3つの回答を選択してください

- A. <apex:commondLink>
- B. <apex:commandButton>

- C. <Apex:form>
- D. <apex:actionsStatus>
- E. <apex:actionSupport>

正解: **A,B,E** ([コメントを发表する](#))

<apex:commandLink> および <apex:commandButton> コンポーネントは、フォームデータをサーバーに送信することができ、ページの特定の部分を更新するように構成できます。

<apex:actionSupport> は、他のコンポーネントに AJAX サポートを追加し、onclick、onchange などのイベントに基づいて部分的なページ更新をトリガーできます。

参考文献：

Visualforceガイド

質問: 112

次のコードスニペットを考えてみましょう。

```
<c-selected-order>
  <template for:each={orders.data} for:item="order">
    <c-order orderId={order.Id}></c-order>
  </template>
</c-selected-order>
```

<e-orders> コンポーネントは、ユーザーが注文を選択したことを <c-selected-orders コンポーネントにどのように伝えるべきですか？

- A. コンポーネントイベントを作成して発火します。
- B. アプリケーションイベントを作成して発火します。
- C. 標準のDOMイベントを作成して発火します。
- D. カスタムイベントを作成して送信します。

正解: ([正解を表示します](#))

カスタムイベントを作成してディスパッチします。このアプローチにより、LWCコンポーネントは効果的に通信でき、コンポーネントのカプセル化と再利用性を維持できます。参考資料：

Lightning Webコンポーネント間の通信

質問: 113

System.Limit クラス内で、トランザクションで行われた呼び出し回数を取得するには、どのようなメソッドを呼び出しますか？

- A. [typeOfLimit] を取得 -> (例: getDXLStaterr.ents())
- B. getLimit [typeOf Limit] -> (例: getLirr.it DXLSt at err.ents ())

正解: **A** ([コメントを发表する](#))

質問: 114

LightningプロセスからApexクラスの機能呼び出すにはどうすればよいですか？ 2つ選択してください)

- A. Process.Plugin インターフェースを実装します。
- B. ProcessInvocable 基底クラスを拡張します。
- C. クラスをカスタム REST API として公開します。

D. @InvocableMethod アノテーションを使用します。

正解: A,D ([コメントを發表する](#))

質問: 115

アーカイブされたタスクを含め、削除されたタスクを除外し、12ヶ月から24ヶ月前のタスクを取得するには、どのSELECT文を挿入すればよいですか？

Java

```
Date initialDate = System.Today().addMonths(-24);
```

```
Date endDate = System.Today().addMonths(-12);
```

A. [SELECT ... FROM Task WHERE What.Type = 'Account' AND isArchived=true AND CreatedDate
=> :initialDate ...]

B. [SELECT ... FROM Task WHERE What.Type = 'Account' AND CreatedDate => :initialDate ... ALL ROWS]

C. [SELECT ... FROM Task WHERE What.Type = 'Account' AND isDeleted=false AND CreatedDate =>
:initialDate ... すべての行]

D. [SELECT ... FROM Task WHERE What.Type = 'Account' AND isArchived=true AND CreatedDate
=> :initialDate ... すべての行]

正解: ([正解を表示します](#))

このクエリを実行するには、Salesforceが古いレコードをどのように処理するかについて、具体的な理解が必要です。Salesforceはパフォーマンスを維持するために、365日以上前のタスクを「アーカイブ」します。これらのアーカイブされたレコードは、標準のSOQLクエリでは返されません。

クエリにアーカイブされたレコードを含めるには、開発者はクエリの最後にALL ROWSキーワードを使用する必要があります。しかし、ALL ROWSはごみ箱（削除済みレコード）にあるレコードも返します。「アーカイブされたレコードを含める」が「削除済みレコードを除外する」という要件を満たすには、開発者はALL ROWSの結果にも標準のWHERE句フィルタが適用されるという事実を利用するだけで済みます。

オプションBが最も正確です。SOQLでは、削除されたレコードを明示的に検索しない限り、isDeleted=falseは暗黙的に指定されます。重要なのは、アーカイブされたタスクも対象に含めるALL ROWSキーワードです。

オプションAとDは、isArchived=trueではアーカイブ済みのタスクのみが返され、アーカイブされていないタスクは除外されるため、誤りです。オプションCは、isDeleted=falseがデフォルトの動作であるため、冗長です。プロンプトオプションの構文=>は、厳密には>=のタイプミスです。オプションBは、ごみ箱を無視して「アクティブ」と「アーカイブ済み」の境界を越えてクエリを実行する標準的な方法を表しています。

質問: 116

Universal Containers (UC) は、顧客情報を保存する ERP システムを使用しています。Salesforce でアカウントが作成されると、新規顧客作成用の ERP システムの REST エンドポイントがアカウント情報とともに自動的に呼び出される必要があります。ERP システムへの呼び出しが失敗した場合でも、アカウントは作成される必要があります。UC 組織のアカウントは、UC の顧客オンボーディング部門のユーザーによってのみ、一度に 1 つずつ作成されます。

開発者はERPへの呼び出しを行うために何を実装すべきでしょうか？

システムのRESTエンドポイント？

A. JavaScriptからのRESTコール

B. ヘッドレスクイックアクション

C. トリガーからキュー可能なオブジェクトを呼び出す

D. 頂点継続

正解: ([正解を表示します](#))

開発者は、トリガーから Queueable を呼び出して、ERP システムの REST エンドポイントを呼び出す必要があります。Queueable は Queueable インターフェイスを実装するクラスで、外部サービスへの呼び出しを行うことができる非同期ジョブを実行するために使用できます。開発者は Database.AllowsCallouts インターフェイスを使用して Queueable クラスを呼び出し可能として

マークし、System.enqueueJob メソッドを使用して Queueable クラスをキューに追加できます。その後、開発者はアカウントのトリガーから Queueable クラスを呼び出し、アカウント情報をパラメーターとして渡すことができます。このようにして、開発者は ERP システムの REST エンドポイントを呼び出し、応答またはエラーを処理できます。ERP システムへの呼び出しが失敗した場合でも、Queueable クラスはトリガーとは別のランザクションで実行されるため、アカウントは作成されます。JavaScript から REST 呼び出しを行うことはできません。JavaScript コードを実行するには Lightning Web コンポーネントまたは Visualforce ページが必要であり、アカウントの作成によってトリガーされないためです。ヘッドレスクイックアクションを使用しても機能しません。アクションを呼び出すにはユーザー操作が必要であり、アカウントの作成ではトリガーされないためです。Apex継続を使用しても機能しません。長時間実行されるコールアウトを処理するためにVisualforceページが必要であり、アカウントの作成ではトリガーされないためです。参照 [キュー可能なApex]、[Apex開発者ガイド]

質問: 117

開発者が外部システムへの複数の呼び出しを行うApexクラスを作成しました。これらの呼び出しで使用されるURLが頻繁に変更される場合、開発者はApexクラスへの変更を最小限に抑えるためにどの機能を使用すべきでしょうか？

- A. 名前付き認証情報
- B. 連携アプリ
- C. リモートサイト設定
- D. セッションID

正解: ([正解を表示します](#))

エンドポイント (URL) や認証情報が変更される可能性のある統合を管理する場合、名前付き認証情報 (オプションA) が最も保守しやすく安全なソリューションとなります。名前付き認証情報では、コールアウトエンドポイントのURLと必要な認証パラメータを1つの定義で指定します。Apexコード内で名前付き認証情報を参照することで (例callout:MyExternalSystem/path)、開発者はコード自体からURLを抽象化できます。

URLが変更された場合でも、管理者は設定メニューで名前付き資格情報レコードを更新するだけで済み、コードのデプロイやApexクラスの変更は不要です。さらに、名前付き資格情報はリモートサイト設定の呼び出しオーバーヘッドを自動的に処理するため、名前付き資格情報を使用する場合は、リモートサイト設定に個別のエントリを保持する必要はありません。コネクテッドアプリ (オプションB) は、外部アプリケーションがSalesforceに接続できるようにするために使用されるものであり、アウトバウンドコールアウトを行うためのものではありません。セッションID (オプションD) は認証トークンであり、エンドポイントURLを管理するものではありません。名前付き認証情報を使用することで、開発者は統合ロジックを環境固有の設定から切り離し、サンドボックス環境と本番環境間の移行をスムーズに行うことができます。

質問: 118

開発者がアカウントに更新後トリガーを作成しました。ワークフロールールとフィールドの更新により、このトリガーがアカウントレコードを繰り返し更新します。開発者は再帰トリガーをどのように処理すべきでしょうか？

- A. グローバル変数を使用して、トリガーが複数回実行されるのを防ぎます。
- B. 静的変数を使用して、トリガーが複数回実行されるのを防ぎます。
- C. フィールド更新が実行されないようにワークフロールールを無効化します
- D. トリガーを無効化し、ロジックをプロセスまたはフローに移動します。

正解: ([正解を表示します](#))

質問: 119

Cloud KicksのSalesforce管理者は、米国のすべての郵便番号と、その郵便番号が属するCloud Kicksの販売地域を格納するために、Region_cというカスタムオブジェクトを作成しました。

Object Name:	salesforce
Region__c	
Fields:	
zip_code__c (Text)	
Region_Name__c (Text)	

Cloud Kicksは、リードの郵便番号に基づいて地域情報を入力するトリガーをリードに設定したいと考えています。
この要求を満たす最も効率的な方法は、どのコードセグメントを使用するか？

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

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

A.


```

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

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

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

```

B.

```

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

```

C.

salesforce

```

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

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

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

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

```

salesforce

D.

正解: ([正解を表示します](#))

郵便番号に基づいてリードの Region フィールドを効率的に設定するには、Map を使用して Zip_Code__c フィールドをキーとして Region__c レコードを格納する方法が最適です。この方法であれば、開発者は各リード レコードに対して Region__c オブジェクトをクエリする必要がなくなり、SOQL の制限例外を回避できます。代わりに、開発者は Region__c オブジェクトを一度クエリして結果を Map に格納します。その後、開発者はリード レコードをループ処理し、Map を使用して Zip_Code__c フィールドに基づいて対応する Region__c レコードを取得します。この方法により、SOQL クエリの実行回数が減り、トリガーのパフォーマンスが向上します。

質問: 120

以下のコードスニペットを参照してください。

```

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
        ];
    }
}

```



開発者は、特定の条件が満たされたときに getFetchLeadList を呼び出すことでリードに関する情報を表示する、Lightning Web コンポーネント (LWC) の一部として JavaScript 関数を作成しました。

開発者は、セキュリティを維持しながらLWCがデータを効率的に表示できるようにするために、上記のApexクラスにどのような3つの変更を加えるべきでしょうか？

3つの回答を選択してください

- A. Apex メソッドに @AuraEnabled を注釈として追加します。
- B. クラス宣言で with sharing キーワードを実装します。
- C. クラス宣言で with キーワードを実装します。
- D. SOQLクエリ内でWHERE句を使用します。
- E. Apex メソッドに @AuraEnabled(cacheable=True) という注釈を付けます。

正解: [\(正解を表示します\)](#)

Apex クラスのメソッドを LWC に公開するには、そのメソッドに @AuraEnabled アノテーションを付ける必要があります。With sharing」キーワードは、ユーザーの権限に基づいてデータアクセスが強制されることを保証します。@AuraEnabled (cacheable=true) は、クライアント側のキャッシュを有効にしてパフォーマンスを向上させます。

参考文献：

@AuraEnabled アノテーション: Apex 開発者ガイド

共有ルールの適用 : Apex開発者ガイド

質問: 121

開発者コンソールのどのツールを使用して、SOQLクエリのカーディナリティとコストを監視できますか？

- A. クエリアクティビティツール
- B. 監査ログ
- C. クエリプランツール
- D. 状態タブを表示

正解: [C \(コメントを發表する\)](#)

有効的な**PDII**問題集はJPNTTest.com提供され、**PDII**試験に合格することに役に立ちます！JPNTTest.comは今最新**PDII**試験問題集を提供します。JPNTTest.com PDII試験問題集はもう更新されました。ここで**PDII**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/PDII-mondaishu> **163**問、**30%ディスカウント**、特別な割引コード：**JPNshiken**」

質問: **122**

開発者がテストクラス内から組織データにアクセスしようとしています。テストクラスに (seeAllData=true) アノテーションが必要な sObject 型はどれですか？

- A. ユーザー
- B. レポート
- C. レコードタイプ
- D. プロフィール

正解: ([正解を表示します](#))

質問: **123**

外部IDを使用するメリットを2つ挙げてください。

- A. 外部IDはインデックス化されており、SOQLクエリのパフォーマンスを向上させることができます。
- B. 外部IDフィールドは、別の外部システムからの一意のIDを参照するために使用できます。
- C. 外部IDはSalesforce Mobileでを使用して、外部データを表示できます。
- D. 外部IDは、Salesforceの2つのフィールドから一意のキーを作成するのに役立つ数式フィールドです。

正解: **A,B** ([コメントを発表する](#))

質問: **124**

以下の変数のうち、ビュー状態では送信されないものはどれですか？ 2つ選択してください。)

- A. プライベート
- B. 過渡的
- C. 公開
- D. 静的

正解: ([正解を表示します](#))

質問: **125**

以下のコードにはどのような設計上の問題がありますか？ trigger accountTrigger on Account (before update) { Boolean processOpportunity = false; List<opportunity> opptysClosedLost = new List<opportunity>(); List<opportunity> lstAllOpp = [select StageName from Opportunity where accountId IN :Trigger.newMap.keySet()]; if(!lstAllOpp.isEmpty()) processOpportunity = true; while(processOpportunity) { for(opportunity o : lstAllOpp) { if(o.StageName == 'Closed - Lost') opptysClosedLost.add(o); processOpportunity = false; } } if(!opptysClosedLost.isEmpty() delete opptysClosedLost;

- A. 関連する商談からアカウントのStageNameの数式フィールドを作成することで、SOQLを回避できます。
- B. このコードを実行すると、System.DmlException:Entity_is_Deleted エラーが発生します。
- C. このコードを実行すると、System.LimitException: スクリプトステートメントが多すぎますというエラーが発生します。
- D. このコードを実行すると、System.LimitException: Apex CPU 時間制限を超えましたというエラーが発生します。

正解: ([正解を表示します](#))

質問: 126

開発者は次のようなコードを書きます。

```
public with sharing class OrderController() {  
    public PageReference sendOrder()  
    {  
        Order_c order = new Order_c(); insert order; ExternalOrder externalOrder = new  
        ExternalOrder(order); Http h = new Http(); HttpRequest req = new HttpRequest  
        (); req.setEndpoint('https://www.example.org/v1/orders'); req.setMethod  
        ('POST'); req.setBody(JSON.serialize(externalOrder)); HttpResponse res =  
        h.send(req); order = (Order_c) JSON.deserialize(res.getBody  
        (), ExternalOrder.class);  
    }  
}
```

コードのテスト中に、開発者は次のエラーメッセージを受け取りました。System.CalloutException : コミットされていない作業が保留中です。開発者は何をすべきでしょうか？ 2つ選択してください。

- A. Database.insert(order, true)を使用して、データベースの変更を即座にコミットします。
- B. DMLステートメントを実行する前に、すべてのコールアウトが完了していることを確認してください。
- C. HTTPクラスのasyncSend()メソッドを使用して、非同期コンテキストでリクエストを送信します。
- D. Webサービス呼び出しを将来のメソッドに移動する

正解: [\(正解を表示します\)](#)

質問: 127

企業のサポートプロセスでは、「ケースが 解決できませんでした」というステータスでクローズされた場合、エンジニアリングレビューのカスタムオブジェクトレコードを作成し、ケース、連絡先、およびケースに関連付けられている製品の情報を入力することが義務付けられています。Apexトリガーを使用してこれを自動化する正しい方法がありますか？

- A. エンジニアリングレビューレコードを作成して挿入する更新前トリガー
- B. エンジニアリングレビューレコードを作成して挿入する更新前トリガー
- C. アップサート後にエンジニアリングレビューレコードを作成して挿入するトリガー
- D. 更新後にエンジニアリングレビューレコードを作成して挿入するトリガー

正解: [\(正解を表示します\)](#)

質問: 128

開発者がHTMLテーブルを表示するためのカスタムコンポーネントを作成しました。開発者は、このコンポーネントを複数のVisualforceページで使用し、テーブルのヘッダーテキストをそれぞれ異なるものに指定できるようにしたいと考えています。

開発者はコンポーネント内でどのタグを使用すべきでしょうか？

- A. <apex:attribute>
- B. <apex:param>
- C. <apex:define>
- D. <apex:variable>

正解: A ([コメントを発表する](#))

質問: 129

どの注釈によって、ApexクラスがRESTfulなwebサービスとして公開されますか？

- A. @RestRtsoorct
- B. @HttpInvocabte

C. @RemoteAction

D. @AuraEnabled

正解: ([正解を表示します](#))

質問: 130

以下のコードスニペットを参照してください。

```
public static void updateCreditMemo(String customerId, Decimal newAmount){  
    List<Credit_Memo__c> toUpdate = new List<Credit_Memo__c>();  
    for(Credit_Memo__c creditMemo : [Select Id, Name, Amount__c FROM Credit_Memo__c WHERE Customer_Id__c = :customerId LIMIT  
50]){  
        creditMemo.Amount__c = newAmount;  
        toUpdate.add(creditMemo);  
    }  
    Database.update(toUpdate, false);  
}
```

Salesforce環境には、Credit_Memo__cというカスタムオブジェクトが存在します。このタイプのレコードを取得および操作する新機能の開発の一環として、開発者は、Apexトランザクション内で一連のレコードが変更される際に、競合状態が発生しないようにする必要があります。

前述のApexコードにおいて、開発者はトランザクション内での競合状態を防ぐために、SOQLの機能を使用するようにクエリステートメントをどのように変更すればよいのでしょうか？

A. [Select Id, Name, Amount__c FROM Credit_Memo__c WHERE Customer_Id__c = :customerId LIMIT 50 FOR UPDATE]

B. [Select Id, Name, Amount__c FROM Credit_Memo__c WHERE Customer_Id__c = :customerId LIMIT 50 FOR VIEW]

C. [Select Id, Name, Amount__c FROM Credit_Memo__c WHERE Customer_Id__c = :customerId USING SCOPE LIMIT 50]

D. [Select Id, Name, Amount__c FROM Credit_Memo__c WHERE Customer_Id__c = :customerId LIMIT 50 FOR REFERENCE]

正解: ([正解を表示します](#))

Apexトランザクション中に競合状態が発生するのを防ぐには、SOQLクエリに「FOR UPDATE」キーワードを追加することで対応できます。このキーワードはクエリによって返されるレコードをロックし、現在のトランザクションが完了するまで他のトランザクションがレコードを変更できないようにします。これにより、現在のトランザクションが完了する前に他のプロセスによって取得されたレコードが変更されることがなくなり、競合状態を防ぐことができます。提供されているコードスニペットで使用する正しいオプションは次のとおりです。

[SELECT Id, Name, Amount__c FROM Credit_Memo__c WHERE Customer_Id__c = :customerId LIMIT

[更新情報50]

これにより、指定された Customer_Id__c を持つ Credit_Memo__c のレコードが最大 50 件まで、トランザクションの期間中ロックされます。

参考文献：

更新情報

レコード更新の競合を防止する

質問: 131

開発者が、Salesforceからデータを取得し、それをレコードのプロパティに割り当てるLightning Webコンポーネントを構築しています。

```
import { LightningElement, api, wire } from 'lwc';
import { getRecord } from 'lightning/uiRecordApi';

export default class Record extends LightningElement {
  @api fields;
  @api recordId;
  record;
}
```

Salesforceからデータを取得するために、コンポーネント内でどのような処理を行う必要がありますか？

A)

Add the following code above record;

```
@api(getRecord, { recordId: '$recordId' })
```

Get the fields in renderedCallback() and assign them to record.

B)

Add the following code above record;

```
@wire(getRecord, { recordId: '$recordId', fields: '$fields' })
```

C)

Add the following code above record;

```
@wire(getRecord, { recordId: '$recordId' })
```

Get the fields in renderedCallback() and assign them to record.

D)

Add the following code above record;

```
@api(getRecord, { recordId: '$recordId', fields: '$fields' })
```

A. オプションB

B. オプションC

C. 選択肢D

D. オプションA

正解: ([正解を表示します](#))

質問: 132

開発者は、商談のステージが変更された回数を追跡する既存の機能に取り組んでいます。商談のステージが変更されると、ワークフロールールが実行され、フィールドの値が1つ増加します。開発者は、フィールドの値が4から5に変更されたときに子レコードを作成するアフタートリガーを作成しました。

ユーザーが商談のステージを変更し、カウントフィールドを手動で4に設定します。カウントフィールドは5に更新されますが、子レコードは作成されません。

なぜこのようなことが起こっているのでしょうか？

A. トリガー .old には、count フィールドの更新された値が含まれていません。

B. トリガー .new はフィールドの更新後も変更されません。

C. トリガーがワークフロールールより前に発火します。

D. フィールド更新後にトリガーが発火しない。

正解: ([正解を表示します](#))

質問: 133

Apex クラスが期待されるコード カバレッジを達成していません。testsetup メソッドが、Apex クラス内のメソッドを明示的に呼び出しています。

開発者はどのようにしてコードカバレッジを生成できるのでしょうか？

- A. testSetup で system.assert() を使用して、値が返されていることを確認します。
- B. ユーザーが System.runs() に渡される権限を持っていることを確認します。
- C. テストセットアップメソッドの代わりに、テストセットアップからApexクラスのメソッドを呼び出します。
- D. 開発者がテストしているクラスのメソッドに @testvisible を追加します。

正解: ([正解を表示します](#))

Apex クラスが期待されるコード カバレッジを達成しない場合、テスト メソッドがクラスの一部のコードを実行していないことが原因と考えられます。@testvisible アノテーションを使用すると、テスト メソッドが別のクラスのプライベート メンバーまたはプロテクト メンバーにアクセスできるようにすることで、コード カバレッジを向上させることができます。これにより、テスト メソッドは直接アクセスできないメソッドを呼び出すことができますが、これはコード カバレッジを直接生成する手段ではありません。コード カバレッジは、テスト メソッドがこれらのメソッド内のコードを実行したときに実際に達成されます。したがって、コード カバレッジを適切に生成するには、開発者はテスト メソッドがカバレッジ対象とするコードの関連部分を呼び出すようにする必要があります。@testvisible を使用すると、通常はアクセスできないコードをテストに利用できるようにすることで、このプロセスを容易にすることができます。

参考文献：

Apex開発者ガイド - テストのベストプラクティス

Apex開発者ガイド - @testVisibleアノテーション

質問: 134

コードではなく宣言型カスタマイズを使用する3つの利点は何ですか？3つ選択してください。

- A. 宣言型カスタマイズは、Salesforce の各リリースで自動的に更新されます。
- B. 宣言型カスタマイズは一般的にメンテナンスの手間が少なくて済みます。
- C. 宣言型カスタマイズはガバナー制限の対象外です。
- D. 宣言的なカスタマイズでは実行時エラーは発生しません。
- E. 宣言型カスタマイズではユーザーテストは不要です。

正解: ([正解を表示します](#))

質問: 135

Universal Containersは、未処理の例外が発生した場合に、Apexを使用してカスタムイベントを発行することで、外部システムに通知したいと考えています。

この要件を満たすための適切なパブリッシュ/サブスクライブロジックは何ですか？

- A. addError() メソッドを使用してエラー イベントを発行し、外部システムが CometD を使用してイベントを購読します。
- B. addError() メソッドを使用してエラー イベントを発行し、イベントを購読して外部システムに通知するトリガーを作成します。
- C. Eventrus.publish() メソッドを使用してエラー イベントを発行し、CometD を使用して外部システムがイベントを購読します。
- D. 外部システムにイベントチャネルを購読させます。公開は不要です。

正解: ([正解を表示します](#))

質問: 136

以下のコードが与えられた場合：


```

for ( Contact c : [SELECT Id, LastName FROM Contact WHERE CreatedDate =
TODAY] )
{
    Account a = [SELECT Id, Name FROM Account WHERE CreatedDate = TODAY
LIMIT 5];
    c.AccountId = a.Id;
    update c;
}

```

本日、10件の連絡先と5件のアカウントが作成されたと仮定した場合、どのような結果が予想されますか？

- A. System.LimitException: Contact に対する SOQL クエリが多すぎます
- B. System.LimitException: アカウントに対するSOQLクエリが多すぎます
- C. System.QueryException: アカウントの割り当てに対してリストに複数の行があります
- D. System.QueryException: Contact で DML ステートメントのエラーが多すぎます

正解: C ([コメントを发表する](#))

有効的なPDII問題集はJPNTTest.com提供され、PDII試験に合格することに役に立ちます！JPNTTest.comは今最新PDII試験問題集を提供します。JPNTTest.com PDII試験問題集はもう更新されました。ここでPDII問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/PDII-mondaishu> 163問、30%ディスカウント、特別な割引コード:

JPNshiken」

質問: 137

以下のコードには、どのような設計上の問題点があるでしょうか？

```

trigger accountTrigger on Account (before update){ Boolean processOpportunity
= false; List<opportunity> opptyClosedLost = new List<opportunity>()
List<opportunity> IstAllOpp = [select StageName from Opportunity where
accountId IN :Trigger.newMap.keySet()]; if(!IstAllOpp.isEmpty())
processOpportunity = true; while(processOpportunity){ for(opportunity o :
IstAllOpp) if(o.StageName == 'Closed - Lost') opptyClosedLost.add(o);
processOpportunity = false; if(!opptyClosedLost.isEmpty()) delete
opptyClosedLost;
}
}

```

- A. このコードを実行すると、System.DmException:Entity_is_Deleted エラーが発生します。
- B. このコードを実行すると、System.LimitException: スクリプトステートメントが多すぎますというエラーが発生します。
- C. 関連する商談からアカウントのStageNameの数式項目を作成することでSOQLを回避できます。
- D. このコードを実行すると、System.LimitException: Apex CPU time limit exceeded エラーが発生します。

正解: **D** ([コメントを發表する](#))

質問: **138**

部分的なページ更新を実行するためにAjax動作を開始するために使用できるVisualforceコンポーネントはどれですか？3つ選択してください。

A. オプション

`<apex:form>`

B. オプション

`<apex:actionStatus>`

C. オプション

`<apex:actionSupport>`

D. オプション

`<apex:commandLink>`

E. オプション

`<apex:commandButton>`

正解: ([正解を表示します](#))

質問: **139**

Lightning Web コンポーネントを開発する際、携帯電話などの小型デバイスでは lightninglayout-items を 1 列で表示し、タブレットサイズやデスクトップサイズの画面では 2 列で表示するには、どの設定を使用しますか？

- A. セットサイズ="12" タブレット端末サイズ="6"
- B. セットサイズ="12" 中型デバイスサイズ="6"
- C. セットサイズ="6" モバイルデバイスサイズ="12"
- D. セットサイズ="6" 小型デバイスサイズ="12"

正解: ([正解を表示します](#))

小型デバイスでは lightning-layout-item を 1 列に、タブレットサイズおよびデスクトップサイズの画面では 2 列に表示するための正しい設定は、size="6" small-device-size="12" と設定することで。size 属性は、lightning-layout 内の lightning-layout-item の相対的な幅を指定します。値の範囲は 1 ～ 12 で、12 は親コンテナの全幅になります。small-device-size 属性は、携帯電話などの小型デバイスでの lightning-layout-item の相対的な幅を指定します。値の範囲も 1 ～ 12 です。size="6" small-device-size="12" と設定することで、開発者は lightning-layout-item をタブレットサイズおよびデスクトップサイズの画面では親コンテナの幅の半分に、小型デバイスでは親コンテナの全幅にすることができます。参照: [lightning-layout-item コンポーネント]、[lightning-layout コンポーネント]

質問: **140**

現在のユーザーのロケールで日付を文字列に変換するには、どの方法を使用すればよいですか？

- A. 日付形式
- B. 文字列形式
- C. Date.parse

D. String.valueOf

正解: **D** ([コメントを發表する](#))

質問: **141**

次のコードスニペットを考えてみましょう。

このVisualforceページのユーザーから、検索ボタンを押すたびにページが完全に更新されるという苦情が寄せられています。

開発者は、部分的な更新を行い、商談リストに関連付けられたセクションのみが画面上に再描画されるようにするには、どのような手順を踏むべきでしょうか？

- A. <apex:actionFunction> タグを immediate = true で実装します。
- B. <apex:commandButton>タグにrendered属性を実装します。
- C. ローションメソッドの検索がnullを返すことを確認してください。
- D. DATA テーブルを <apex:actionRegion> タグで囲みます。

正解: **D** ([コメントを發表する](#))

質問: **142**

開発者は、単一のトランザクション内で複数のレコードリストを挿入、更新、削除しており、エラーが発生した場合はすべての実行が阻止されるようにしたいと考えています。

開発者は、このような事態に対処するために、コード内でどのようにエラー例外処理を実装すべきでしょうか？

- A. Database メソッドを使用して Database.SaveResult のリストを取得します。
- B. try-catch文を使用し、catch文でDMLクリーンアップを処理します。
- C. try-catch ステートメントで Database.setSavepoint {} と Database.rollback を使用します。
- D. try-catch を使用し、失敗した場合は sObject.addError() を使用します。

正解: **C** ([コメントを發表する](#))

この要件に対応するために、コード内でエラー例外処理を実装する最良の方法は、try-catch ステートメントとともに Database.setSavepoint() と Database.rollback() を使用することです。Database.setSavepoint() は、リクエスト内のその時点のデータベースの状態を指定するポイントであるセーブポイントを作成するメソッドです。Database.rollback() は、指定されたセーブポイントで保存された状態にデータベースをロールバックするメソッドです。try-catch ステートメントは、コードの実行中に発生するエラーである例外を処理する方法です。Database.setSavepoint() と Database.rollback() を try-catch ステートメントとともに使用することで、開発者はレコードのリストを挿入、更新、または削除する前にセーブポイントを作成し、エラーが発生した場合はセーブポイントまでロールバックできます。このようにして、開発者はエラーが発生した場合でもすべての実行を阻止し、データベースの整合性を維持できます。参照: [セーブポイントとトランザクション]、[Database.setSavepoint メソッド]、[Database.rollback メソッド]、[例外処理]

質問: **143**

Salesforce開発者が多国籍企業に雇われ、従業員の福利厚生と経年的な報酬額を表示するカスタムLightningアプリケーションを構築することになった。このアプリケーションは、日付、時刻、数値、通貨、通貨記号に関して、ユーザーの地域設定を認識し尊重する必要がある。

Auraコンポーネントを使用する場合、Lightningアプリケーションがユーザーのロケールに準拠するようにするには、開発者はどのようなアプローチを採用すべきでしょうか？

- A. \$User グローバル変数を使用してユーザー設定を取得します。
- B. \$Locale 値プロバイダを使用してユーザー設定を取得します。
- C. \$Label グローバル値プロバイダーを使用します。
- D. ユーザー設定を保存するための階層型カスタム設定を作成します。

正解: ([正解を表示します](#))

質問: **144**

開発者がクラス内にDebugメソッドを作成し、それが数百回呼び出されるとします。開発者コンソールでそのメソッドの呼び出し回数をカウントする最適な機能は何でしょうか？

- A. スタックツリーパネルの下にある「実行ツリー」タブ。
- B. 「実行スタック」パネル。
- C. 実行概要パネルの「実行済みユニット」タブ。
- D. 「実行ログ」パネル。

正解: ([正解を表示します](#))

質問: 145

開発者がテストクラスを作成し、モック呼び出しを行う前にテストデータを作成するようにしたところ、「**毎コミットの作業が保留中です。呼び出しを行う前にコミットまたはロールバックしてください**」というエラーが表示されるようになりました。

エラーを解決するには、どの手順を踏むべきですか？

- A. 挿入とモック呼び出しの両方が l==L の後に発生するようにします。stopTest_()。
- B. レコードが Tezt.startTest() ステートメントの前に挿入され、モック呼び出しが @testSetup で注釈されたメソッド内で発生することを確認してください。
- C. Test.startTest() の後に挿入とモック呼び出しの両方が行われるようにします。
- D. レコードが Test.startTess() ステートメントの前に挿入され、モック呼び出しが Test.Starttest() の後に行われることを確認してください。

正解: **D** ([コメントを公表する](#))

コールアウトを含む Apex テストを実行する際には、「コミットされていない作業が保留中」エラーを回避することが重要です。このエラーは通常、コールアウトが試行された時点で DML 操作 (挿入など) が保留中であるために発生します。Salesforce では、トランザクション内の DML 操作の前にすべてのコールアウトを実行する必要があります。これを回避するには、Test.startTest() を呼び出す前にレコードを挿入し、その後にコールアウトを実行するという方法があります。Test.startTest() は実際のテストの開始点を示し、ガバナ制限をリセットします。この後にコールアウトを実行することで、DML 操作がコミットされ、ガバナ制限に関する新しい実行コンテキスト内に入ることができます。

参考文献：

HttpCalloutMockインターフェースを実装してHTTPコールアウトをテストする

実行コンテキストの理解

質問: 146

新しいシステム導入の直前、開発者サンドボックスで新しい注文処理プロセスを設定したSalesforce管理者が突然退職した。ユーザーはサンドボックス内で変更内容をすべてテストし、承認済みだった。

残念ながら、変更セットは開始されたものの、完了には至っていませんでした。そこで、デプロイメントを完了させるために開発者が招集されました。

開発者は、本番環境に移行する必要がある構成変更を特定するために、どのような手順を踏むべきでしょうか？

- A. 継続的インテグレーションとGitリポジトリを設定し、サンドボックスのメタデータからのすべての変更を本番環境のメタデータに自動的にマージします。
- B. Salesforceの設定で、各オブジェクトの最終更新日を確認し、変更セットに追加すべきオブジェクトを決定します。
- C. セットアップ監査証跡を活用して、退職した管理者が行った変更を確認し、変更セットに追加する必要がある変更を特定します。
- D. メタデータ API とサポートされている開発 IDE を使用して、サンドボックスから本番環境にすべての設定をプッシュし、変更が失われないようにします。

正解: ([正解を表示します](#))

質問: 147

Apexトリガーは、商談レコードが「クローズ済み」および「受注済み」とマークされるたびに契約レコードを作成します。このトリガーは正常に動作していますが、(最近の買収により) 過去の商談レコードをSalesforceインスタンスにロードする必要があります。

テスト用のレコードバッチがロードされると、Apexトリガーによって契約レコードが作成されます。

開発者は、商談を一括ロードする際に契約レコードが作成されないようにする任務を負っているが、日常的に使用するユーザーは契約レコードを作成する必要がある。これを実現するために、Apexトリガーを更新する最も拡張性の高い方法は何ですか？

- A. データをロードするユーザーによる契約の作成を防止するために、契約に検証ルールを追加します。
- B. 階層カスタム設定を使用して、データをロードするユーザーに対してトリガー内のロジックの実行をスキップします。
- C. リストカスタム設定を使用して、データを読み込むユーザーのトリガーを無効にします。
- D. データをロードするユーザーのプロファイル ID をトリガーに追加します。これにより、このユーザーに対してトリガーが起動しなくなります。

正解: ([正解を表示します](#))

質問: 148

セーブポイントに関して、正しい記述はどれですか？

- A. セーブポイントはDMLステートメントのガバナー制限の影響を受けません
- B. セーブポイントへの参照は、トリガー呼び出しをクロスさせることができます。
- C. ロールバック中に静的変数は元に戻されません。
- D. 任意の順序で作成されたセーブポイント変数にロールバックできます

正解: ([正解を表示します](#))

質問: 149

開発者は、どのプロファイルとユーザーがどの秘密情報にアクセスできるかを指定できる機能を備えた秘密データを保存する方法を見つけるよう依頼されました。このデータを保存するために何を使用すべきでしょうか？

- A. 静的リソース
- B. カスタムメタデータ
- C. カスタム設定
- D. System.Cookie クラス

正解: ([正解を表示します](#))

構成データや「秘密」データなど、特定のユーザーやプロファイルに応じて変更する必要があるデータを保存する必要がある要件がある場合、階層カスタム設定 オプションC)が適切な選択肢となります。ユーザーに関係なく組織全体で利用できるカスタムメタデータ オプションB)とは異なり、階層カスタム設定は組み込みのロジックを使用して、定義された「最も具体的な」レベルに基づいて値を提供します。

階層構造は、ユーザー > プロファイル > 組織という特定の順序に従います。ユーザーレベルで秘密鍵が定義されている場合、その特定のユーザーに対してその値が返されます。定義されていない場合は、システムはユーザーのプロファイルレベルで定義されている値を探し、最終的に組織全体のデフォルト値にフォールバックします。これにより、開発者は機密性の高い鍵やフラグを保存し、現在のユーザーのIDに基づいて動的に制限または変更することができます。

カスタムメタデータは、パッケージを介してデプロイする必要があるアプリケーション全体の構成に適していますが、階層型カスタム設定に固有の、ユーザーごと/プロファイルごとの詳細な上書き機能は備えていません。静的リソース オプションA)は、リソースにアクセスできるすべての人に公開され、Cookie オプションD)はクライアント側に保存されるため、「秘密」データには安全ではありません。したがって、このユースケースでは、カスタム設定がセキュリティと階層的な柔軟性の最適なバランスを提供します。

質問: 150

上記のコードを考えてみてください。

```

<apex:page standardController="Account">
  <apex:pageBlock title="Hello {!$User.FirstName}!">
    You are displaying contacts from the {!account.name} account. Click a contact's name to view his or her
    details.
  </apex:pageBlock>
  <apex:pageBlock title="Contacts">
    <apex:form>
      <apex:dataTable value="{!account.Contacts}" var="contact" cellPadding="4" border="1">
        <apex:column>
          <apex:commandLink rerender="detail">
            {!contact.Name}
            <apex:param name="cid" value="{!contact.id}"/>
          </apex:commandLink>
        </apex:column>
      </apex:dataTable>
    </apex:form>
  </apex:pageBlock>

  <apex:outputPanel id="detail">
    <apex:detail subject="{!$CurrentPage.parameters.cid}" relatedList="false" title="false"/>
  </apex:outputPanel>
</apex:page>

```

ユーザーが連絡先の名前のリンクをクリックすると、何が起こるか？

- A. outputPanelが更新され、連絡先の詳細が表示されます。
- B. 何も起こりません: commandLink に action 属性がありません。
- C. 新しいページが開き、連絡先の詳細が表示されます。
- D. ページが更新され、連絡先の詳細が表示されます。

正解: A ([コメントを发表する](#))

質問: 151

CreateOneAccount クラスが 1 つのアカウントを作成し、Queueable インターフェースを実装すると仮定した場合、Apex コードをテストする構文はどれですか？

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. オプションD

B. オプションA

C. オプションC

D. オプションB

正解: ([正解を表示します](#))

有効的な**PDII**問題集はJPNTTest.com提供され、**PDII**試験に合格することに役に立ちます！JPNTTest.comは今最新**PDII**試験問題集を提供します。JPNTTest.com PDII試験問題集はもう更新されました。ここで**PDII**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/PDII-mondaishu> **163**問、**30%ディスカウント**、特別な割引コード：**JPNshiken**」

質問: **152**

Universal Containers (UC) は現在、フルコピーのサンドボックス環境での開発は行っていません。

最近、UCでは複数の開発者が同時に開発を行うプロジェクトがいくつかある。UCは、開発者が行った変更によって他の開発者の作業にエラーが発生するという問題に直面している。

さらに、デプロイ準備が整った段階で、多くの単体テストが失敗し、デプロイが妨げられる。

これらの問題を解消するために、UCにはどのような3種類の組織を推奨すべきでしょうか？

3つの回答を選択してください

A. システムインテグレーション組織

B. 開発組織

C. 継続的インテグレーション (CI) 組織

D. データ移行組織

E. ステージング組織

正解: ([正解を表示します](#))

質問: 153

Universal Containersでは、顧客がSalesforceコミュニティにログインし、カスタムVisualforceページを通じて注文を更新できるようにしています。Universal Containersの営業担当者も、同じVisualforceページで注文を編集できます。

開発者は、Apexテストクラスで、Visualforceページでレコード共有が正しく適用されていることをテストするために、何を使用すべきでしょうか？

- A. System.profiles() を使用して、営業担当者およびコミュニティユーザーとしてテストします。
- B. System.profiles() を使用して、管理者およびコミュニティ ユーザーとしてテストします。
- C. System.profiles1h() を使用して、営業担当者およびコミュニティユーザーとしてテストします。
- D. System.runsAs() を使用して、管理者およびコミュニティ ユーザーとしてテストします。

正解: C ([コメントを發表する](#))

Visualforce ページでレコード共有が強制されていることをテストするには、開発者は System.runAs() を使用して、営業担当者およびコミュニティ ユーザーとしてテストする必要があります。System.runAs() メソッドを使用すると、開発者はテスト メソッドを指定したユーザーとして実行し、そのユーザーがレコードに対して適切な権限とアクセス権を持っていることを確認できます。開発者はこのメソッドを使用して、営業担当者ユーザーとコミュニティ ユーザーを作成し、各ユーザーとしてテスト メソッドを実行して、レコード共有ルールが Visualforce ページに正しく適用されていることを確認できます。開発者は System.assert() を使用して、期待される結果を検証することもできます。Apex にはそのようなメソッドがないため、System.profiles() を使用して営業担当者およびコミュニティ ユーザーとしてテストしても機能しません。System.profiles() を使用して管理者およびコミュニティ ユーザーとしてテストしても機能しません。管理者ユーザーはすべてのレコードにアクセスでき、レコード共有ルールをテストできないためです。System.runAs() を使用して管理者およびコミュニティ ユーザーとしてテストしても機能しません。管理者ユーザーはすべてのレコードにアクセスでき、レコード共有ルールをテストできないためです。参考資料 [runAsメソッドを使用したテスト]、[カスタムコントローラとコントローラ拡張機能のテスト]

質問: 154

開発者は、Lightning Webコンポーネントを修正して、スマートフォンでは1列で表示され、タブレット、デスクトップ、および大型デバイスでは2列で表示されるようにするよう依頼された。以下のコードが与えられます。

```
<lightning-layout multiple-rows="true">
  <lightning-layout-item>
    <div>Example 1</div>
  </lightning-layout-item>
  <lightning-layout-item>
    <div>Example 2</div>
  </lightning-layout-item>
</lightning-layout>
```

開発者は要件を満たすために、コードに何を追加すべきでしょうか？

- A. Add small-device-size="12" to the <lightning-layout-item> elements
- B. Add size="6" small-device-size="12" to the <lightning-layout-item> elements
- C. Add size="12" medium-device-size="6" to the <lightning-layout-item> elements
- D. Add medium-device-size="6" to the <lightning-layout-item> elements

正解: ([正解を表示します](#))

Lightning Webコンポーネントは、スマートフォンでは1列表示、大型デバイスでは2列表示にする必要があります。これを実現するには、Lightningデザインシステムが提供するデザイントークンを使用して、デバイスごとにサイズ属性を指定する必要があります。

オプションBが正解です。size="6"を指定すると、各lightning-layout-itemが利用可能なスペースの半分を占めるようになり (グリッドシステムは12列構造に基づいているため)、タブレット、デスクトップ、および大型デバイスでは2列になります。small-device-size="12"を指定すると、スマートフォンなどの小型デバイスでは各アイテムが全幅を占め、結果として1列になります。

オプションA、C、Dは、小型デバイスのサイズを正しく設定していないか、小型デバイスと大型デバイスの両方のサイズを適切に指定していないため、要件を正しく満たしていません。

参考文献：

Lightning Design System Grid ドキュメント: Lightning Grid

質問: 155

Lightning レコードページのボタンをクリックしたときにモーダルダイアログに表示されるようにするには、Lightning コンポーネントにどのインターフェイスを実装する必要がありますか？

A. lightning:editAction

B. force:lightningEditAction

C. ライトニング:クイックアクション

D. force:lightningQuickAction

正解: (正解を表示します)

質問: 156

Visualforceページは、表示するデータ量が多いため、読み込みが遅くなります。開発者は、パフォーマンスを向上させるためにどのような戦略を用いるべきでしょうか？

A. カスタムコントローラで使用されるリスト変数には、transientキーワードを使用してください。

B. コントローラのコンストラクタではなく、必要に応じてデータをロードするために遅延ロードを使用します。

C. JavaScriptを使用して、データ処理をコントローラーではなくブラウザに移動します。

D. ページ内で <apex:actionPoller> を使用して、すべてのデータを非同期で読み込みます。

正解: B (コメントを発表する)

Visualforceページが大量のデータセットを一度に読み込もうとして遅くなる場合、ボトルネックとなるのは、ページがレンダリングを開始する前にコントローラのコンストラクターがレコードを照会して処理するのにかかる時間であることが多い。遅延読み込み オプションB)は、開発者がページの初期初期化時にすべてのデータを読み込むことを避ける戦略である。コンストラクターで 10,000 件のレコードをクエリする代わりに、開発者はまずページの構造要素のみを読み込むことができます。その後、アクション メソッド (<apex:actionFunction> または window.onload スクリプトによってトリガーされる) を使用して、データはバックグラウンドで取得されます。これにより、ユーザーはデータが読み込まれる前にページ レイアウトをすぐに確認できます。

すぐに 読み込み」が行われます。これにより、最初のバイトまでの時間 (TTFB)とページの体感パフォーマンスが大幅に向上します。

オプションA (一時的は、ビュー状態のサイズを縮小し、ポストバック ボタンクリック)を高速化するのに優れていますが、大規模データセットの初期読み込み時間を本質的に高速化するものではありません。オプションC (JavaScript処理)は役立ちますが、データ取得のボトルネックを解消するものではありません。オプションD (actionPoller)は定期的な更新に使用され、大規模データセットを初期に読み込む効率的な方法ではありません。遅延読み込みにより、重い処理」が初期ページ読み込みから切り離されるため、より応答性の高いエクスペリエンスが実現します。

質問: 157

Universal Containersは、ユーザーが送信した商品画像をリサイズするHerokuサービスと連携する必要がある。

Herokuアプリのエンドポイントへの悪意のある呼び出しから保護しつつ、統合を実装するための2つの代替案は何ですか？2つ選択してください。

A. JSONシリアル化されたデータを渡す@future Apex HTTPコールアウトを使用するトリガーを作成します。これにより、HerokuアプリはSalesforceでサイズ変更された画像でコールアウトに自動的に応答できます。

B. アウトバウンドメッセージを含むワークフロールールを作成し、セッションIDを送信」を選択して、Herokuアプリがそれを使用して再画像をSalesforceに送信できるようにします。

C. JSONシリアル化データと事前共有シークレットキーを渡す@future Apex HTTPコールアウトを使用するトリガーを作成し、Herokuアプリがリクエストを認証して、サイズ変更された画像をSalesforceに保存できるようにします。

D. Herokuアプリがサイズ変更された画像をSalesforceに自動的に保存できるように、アウトバウンドメッセージを含むワークフロールールを作成します。

正解: (正解を表示します)

質問: 158

QueryLocatorを使用した場合、トランザクションで処理できるレコード数の上限はいくつですか？

- A. 50,000
- B. 50,000,000
- C. 100,000
- D. 5,000,000
- E. 制限はありません

正解: ([正解を表示します](#))

説明

説明／参考資料：

executeBatchの Scopeパラメータは、最大2,000レコードまで設定できます。

質問: 159

次のコードスニペットを考えてみましょう。

```
import { LightningElement } from 'lwc';
import getOrders from '@salesforce/apex/OrderController.getAvailableOrders';

export default class OrderManagement extends LightningElement {
  orders;
  error;

  @wire(getOrders)
  wiredOrders({ error, data }) {
    if (data) {
      this.orders = data;
      this.error = undefined;
    } else if (error) {
      this.error = error;
      this.orders = undefined;
    }
  }
}
```

2つの回答を選択してください

- A. import getOrders from ,@salesforce/apex/OrderController.getAvailableOrders';
- B. import { LightningElement, wire } from 'lwc';
- C. import (LightningElement-apt) from 'lwc*.-
- D. import getOrders from @salesforc/apex/c.OrderController.getAvailablOrders';

正解: ([正解を表示します](#))

質問: 160

有効なApex RESTアノテーションはどれですか？ 2つ選択してください。)

- A. @HttpUpsert
- B. @HttpAction

C. @HttpDelete

D. @HttpPatch

正解: ([正解を表示します](#))

質問: 161

以下のコードを参照してください。

```
List<Opportunity> opportunities = [SELECT Id, Amount from Opportunity ];  
for(Opportunity opp: opportunities){  
    // perform operation here  
}
```

コードを実行すると、システム制限例外が発生し、「Apex ヒープ サイズが大きすぎます」というエラーメッセージが表示されます。

このエラーを修正するにはどうすればよいでしょうか？

A. Limits.getLimitHeapSize() を使用します。

B. SOQLのforループを使用してデータを処理します。

C. Lis: を 5tc に変換します。

D. try/catch ブロックを使用してエラーを捕捉します。

正解: ([正解を表示します](#))

質問: 162

メッセージがページに表示されています。メッセージを表示するには、Visualforceページにどのコンポーネントを追加すればよいですか？

A. `<apex:pageMessage severity="info" />`

B. `<apex:facet name="messages" />`

C. `<apex:message for="info"/>`

D. `<apex:pageMessages />`

正解: ([正解を表示します](#))

質問: 163

以下の試験方法を参照してください。

```

@isTest
static void testIncrement() {
    Account acct = new Account(Name = 'Test');
    acct.Number_Of_Times_Viewed__c = 0;
    insert acct;

    AuditUtil.incrementViewed(acct.Id);
    Account acctAfter = (SELECT Number_Of_Times_Viewed__c FROM Account
    WHERE Id = :acct.Id)[0];

    System.assertEquals(1, acctAfter.Number_Of_Times_Viewed__c);
}

```

テストメソッドは、@future メソッドを呼び出し、
Number_of_Times_Viewed__c 値。アサーションが失敗しているのは、
視聴回数_cは0です。

この問題を解決するための最適な方法は何ですか？

- A. アサーションを system.assertEquals (0, acctAfter Number _Cf_Timeea_Viewed__c) に変更します。
- B. insert= acct の前に Test.startTest() を、後に Test.stopTest() を追加します。
- C. AuditUtil.incrementViewed の前に test.startTest() を、後に Test.stopTest() を追加します。
- D. 初期化を acct. Number_Of_Times_Viewed_c = 1 に変更します。

正解: C ([コメントを发表する](#))

テストメソッドがDML操作を実行する@futureメソッドを呼び出すと、そのメソッドによって行われた変更は、
@future メソッドは、Test.stopTest() が呼び出されるまで表示されません。このメソッドは、テストの実行が継続される前にすべての非同期処理が完了していることを保証します。したがって、Test.startTest() は @future メソッドの呼び出しの前に、Test.stopTest() は直後に呼び出す必要があります。

参考資料 Apex開発者ガイド - 非同期Apexのテスト

有効的なPDII問題集はJPNTTest.com提供され、PDII試験に合格することに役に立ちます！JPNTTest.comは今最新PDII試験問題集を提供します。JPNTTest.com PDII試験問題集はもう更新されました。ここでPDII問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/PDII-mondaishu> 163問、30%ディスカウント、特別な割引コード:

JPNshiken」