

Salesforce.PDII.v2026-03-26.q116

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：	116
バージョン：	v2026-03-26
ページの閲覧量：	105
問題集の閲覧量：	1331

<https://www.jpnsiken.com/shiken/Salesforce.PDII.v2026-03-26.q116.html>

質問: 1

Universal Containers (UC) には、顧客情報を保存する ERP システムがあります。Salesforce でアカウントが作成されると、新規顧客を作成するための ERP システムの REST エンドポイントが、アカウント情報を使用して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ます。UC 組織内のアカウントは、UC 顧客オンボーディング部門のユーザーによって、一度に 1 つずつのみ作成されます。

ERPを呼び出すために開発者は何を実装すべきか
システムの REST エンドポイントですか？

- A. JavaScriptからのREST呼び出し
- 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オブジェクトを呼び出す
- D. Apex継続

正解: ([正解を表示します](#))

アカウント作成時に外部ERPシステムへのコールアウトを実行し、呼び出しが失敗した場合でもアカウント作成を可能にすることが要件であるため、トリガーから呼び出されるQueueableクラスが最適なソリューションです。Queueable Apexを使用すると、futureメソッドと同様に非同期処理のジョブを送信できますが、ジョブのチェーニングやジョブステータスの監視などの追加のメリットもあります。

Queueable を使用すると、アカウントの作成後にコールアウトを実行できるため、Salesforce の実行順序が尊重され、同じトランザクション内の DML 操作後にコールアウトを実行できないという制限を回避できます。また、このアプローチでは、ERP システムへのコールアウトに時間がかかったり失敗したりした場合でも、ユーザーインターフェースやアカウントレコードの作成がブロックされることはありません。

参考文献:

Salesforce ドキュメントの Queueable Apex: Queueable Apex を使用してジョブを実行する
Salesforce 開発者ガイドの非同期 Apex: Queueable Apex

質問: 2

Universal Containers には、アカウントが顧客としてマークされたときにアカウント プラン レコードを作成する、アカウントに対する Apex トリガーがあります。

最近、レコードトリガーフローが追加され、アカウントが顧客としてマークされるたびに、「顧客開始」日付フィールドが今日の日付に更新されます。フローの追加以降、アカウントが顧客としてマークされるたびに、2 つのアカウント プラン レコードが作成されます。

これが起こる原因は何でしょうか？

- A. Apex トリガーは、1 回だけ実行されることを保証するために静的変数を使用しません。
- B. フローは「レコードの更新」要素を使用するように構成されています。
- C. Apex トリガーは一括安全ではなく、for ループ内で insert を呼び出します。
- D. フローは、レコードが作成された時点と編集されるたびに評価するように構成されています。

正解: [\(正解を表示します\)](#)

フローがレコードの作成時と編集時に評価するように設定されており、Apexトリガーも同じ条件で実行される場合、操作が重複する可能性があります。この場合、アカウントが顧客としてマークされた時点でフローとトリガーの両方が実行されるため、アカウントプランレコードが2つ作成されます。

質問: 3

開発者は、10,000件の取引先の売上金額を毎晩更新するバッチApexプロセス

「Batch_Account_Sales」を開発しています。このバッチApexはサンドボックス内では設計どおりに動作しますが、開発者はバッチApexクラスのコードカバレッジを取得できません。

テストクラスは以下のとおりです。

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

コード カバレッジの問題の原因は何ですか？

- A. バッチでは複数のアカウント レコードを作成する必要があります。
- B. バッチプロセスでは、同じセッションで作成された新しいアカウントは認識されません。
- C. アカウントの作成により、状態数は既に 0 に設定されています。
- D. executeBatch は test.startTest() から test.stopTest() までの間に失敗する必要があります。

正解: D [\(コメントを发表する\)](#)

質問: 4

Salesforce ユーザーは、アカウントを保存するときに常に「トリガーの最大深度を超えました」というエラーを受け取ります。

開発者はこのエラーをどのように修正できますか？

- A. トリガー ロジックを 2 つの個別のトリガーに分割します。
- B. isMultiThread=true アノテーションを使用するようにトリガーを変更します。
- C. トリガーを変換して @future アノテーションを使用し、後続のトリガー呼び出しを Account オブジェクトに連鎖します。
- D. ヘルパー クラスを使用して、トリガーが最初に起動されるときにブール値を 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](#))

1

「トリガーの最大深度を超えました」というエラーは、再帰ループが作成された際に発生し、プラットフォームの再帰呼び出し回数の上限である16回に達するまでトリガーが繰り返し実行されます。これは、更新後トリガーがトリガーを起動したレコードと同じレコードに対してDML操作を実行した場合や、2つの異なるオブジェクトに循環的に更新を行うトリガーがある場合によく発生します。

これを解決するには、開発者はヘルパー クラス内で静的ブール変数を使用して実行状態を管理します。

Apex の静的変数は単一のトランザクションの実行中のみ保持されるため、トリガーはロジックを実行する前にこのブール値をチェックできます。トリガーが初めて実行される際、ブール値が FALSE かどうかを確認し、TRUE に設定してから処理を続行します。同じトランザクション内でトリガーが再度呼び出された場合（再帰）ブール値のチェックは失敗し、ロジックはスキップされます。この「再帰ガード」により、ロジックはトランザクションごとに 1 回だけ実行されます。

ロジックを2つのトリガーに分割しても A)、どちらのトリガーも同じ再帰サイクルの一部となるため、効果はありません。isMultiThreadアノテーション B)は存在せず、@future C)は直近の実行チェーンを切断することはできますが、根本的なロジックの欠陥には対処できず、管理不能な非同期オーバーヘッドにつながる可能性があります。静的変数を用いたアプローチは、再帰制御における業界標準の「ベストプラクティス」です。

質問: 5

管理パッケージでは、さまざまな場所で国 ISO コードと国名のリストを参照データとして使用します。管理パッケージ Apex コード。

リストを保存および取得する最適な方法は何ですか？

- A. カスタム メタデータに情報を保存し、SOQL を使用してクエリを実行します。
- B. リストカスタム設定に情報を保存し、SOQL を使用してクエリを実行します。
- C. リストカスタム設定に情報を保存し、getAll() メソッドでアクセスします。
- D. 情報をカスタム メタデータに保存し、getAll() メソッドを使用してアクセスします。

正解: ([正解を表示します](#))

質問: 6

Visualforce ページには業種選択リストが含まれており、業種フィールドに一致する値を持つ取引先のテーブルが表示されます。

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

ユーザーが業界選択リストの値を変更すると、アカウント テーブルが自動的に更新され、選択した業界に関連付けられたアカウントが表示される必要があります。これを実装する最適な方法は何ですか。

- A. <apex:selectOptions> 内に <apex:actionsupport> を追加します。
- B. <apex:selectList> 内に <apex:actionFunction> を追加します。
- C. <apex:selectList> 内に <apex:actionsupport> を追加します。
- D. <apex:selectOptions> 内に <apex:actionFunction> を追加します。

正解: [\(正解を表示します\)](#)

<apex:actionSupport> コンポーネントは、他のコンポーネントと組み合わせて AJAX サポートを提供するように設計されています。業種選択リストの値が変更されると、<apex:actionSupport> はサーバーへの AJAX リクエストをトリガーし、一致する業種の取引先テーブルをページ上で再レンダリングします。このプロセスはページ全体を更新することなく実行されるため、よりスムーズなユーザーエクスペリエンスが実現します。

参考文献:

Visualforce 開発者ガイド

質問: 7

開発者は、商談のステージが何回変更されたかを追跡する既存の機能を使用しています。商談のステージが変更されると、ワークフロールールが起動され、フィールドの値が1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ためのafter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、手動でカウント フィールドを 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。

このようなことが起こる理由は何でしょうか？

- A. フィールドの更新後、トリガー .new は変更されません。
- B. トリガー .old には、カウント フィールドの更新された値が含まれていません。
- C. フィールドの更新後に After トリガーは起動されません。
- D. After トリガーはワークフロー ルールの前に実行されます。

正解: [\(正解を表示します\)](#)

質問: 8

```
<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> 上記のコンポーネントコードを参照してください。情報はモバイルデバイス
では期待どおりに 3行で) 表示されます。しかし、デスクトップやタブレットでは期待どおりに 1
行で) 表示されません。デスクトップとタブレットで正しく表示されるようにコンポーネントが変
更される正しいオプションはどれですか？

```
D. <lightning: layout multipleRows="true">

<lightning:layoutItem gize="12" mediumDeviceSize="6" largeDevic
eSize="4">{!v.account.Name}

</lightning: layoutItem>

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

</lightning: layoutItem>

<lightning:layoutItem size="12" mediumDeviceSize="6" largeDevic
eSize="4">{!v.account.Industry}

</lightning: layoutItem></lightning: layout>
```

A.

```
A. <lightning: layout multipleRows="true">

<lightning:layoutItem gize="12" mediumDeviceSize="4">{!v.account
t.Name}

</lightning: layoutItem>

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

</lightning: layoutItem>

<lightning:layoutItem size="12" mediumDeviceSize="4">{!v.accou
nt.Industry}</lightning: layoutItem>

</lightning: layout>
```

B.


```

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: layout>

```

C.

```

B. <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>

```

D.

正解: ([正解を表示します](#))

質問: 9

ある組織では、取引先には必ず1人の連絡先 (プライマリ)のみを登録する必要があるという要件があります。そのため、1人の連絡先を選択すると、他の連絡先は選択解除されます。クライアント

は、この機能を制御するために、連絡先に「プライマリ」というチェックボックスを追加したいと考えています。

また、クライアントは、すべての連絡先の姓がすべて大文字で保存されるようにしたいと考えています。

これらの要件を実装する最適な方法は何ですか？

- A. 「Is Primary」ロジックの「Contact」に検証ルールを記述し、姓」ロジックの「Contact」に更新前トリガーを記述します。
- B. 「Is Primary」ロジック用に「Contact」に更新後トリガーを記述し、姓ロジック用に「Contact」に別の更新前トリガーを記述します。
- C. 更新後と更新前の両方に対して「Contact」に単一のトリガーを記述し、各ロジック セットを処理するためにヘルパー クラスを呼び出します。
- D. Is Primary ロジック用に「Account」に更新後トリガーを記述し、姓ロジック用に「Contact」に更新前トリガーを記述します。

正解: [\(正解を表示します\)](#)

更新後イベントと更新前イベントの両方を処理するContactの単一トリガーを、各ロジックセットにヘルパークラスを用いて実装するのが、最も最適な方法です。これにより、主要連絡先の維持と姓の正規化のロジックを一元管理しながら、ヘルパークラスによって明確な関心事の分離が可能になります。参考資料：

Apex 開発者ガイド - トリガー

質問: 10

AW Computing (AWC) は、Salesforce で注文を処理し、カスタム オブジェクトの inventory_c フィールドに製品在庫を保存します。

product_c。product_c の注文が行われると、Apex トリガーを使用して、Inventory_c 項目が注文数量だけ減少します。

```
public void reduceInventory(Id prodId, Integer qty) {  
    Integer newInventoryAmt = getNewInventoryAmt(prodId, qty);  
    Product__c p = new Product__c(Id = prodId, Inventory__c = newInventoryAmt);  
    update p;  
  
    // code goes here  
}
```

AWC は、現在計画中の将来のシステムを含む、多くの外部システムに送信される製品の在庫をリアルタイムで削減したいと考えています。

これらの要件を満たすために、開発者はプレースホルダーのコードに何を追加する必要がありますか？

A)

```
InventoryReductionEvent__c ev = new InventoryReductionEvent__c(ProductId__c = prodId, Reduction__c = qty);  
EventBus.publish(ev);
```

B)

```
InventoryReductionEvent__e ev = new InventoryReductionEvent__e(ProductId__c = prodId, Reduction__c = qty);  
insert ev;
```

C)

```
InventoryReductionEvent__e ev = new InventoryReductionEvent__e(ProductId__c = prodId, Reduction__c = qty);  
EventBus.publish(ev);
```

D)

```
InventoryReductionEvent__c ev = new InventoryReductionEvent__c(ProductId__c = prodId, Reduction__c = qty);  
insert ev;
```

A. オプションB

B. オプションD

C. オプションC

D. オプションA

正解: ([正解を表示します](#))

質問: 11

新しい展開の直前に、開発者サンドボックスで新しい注文履行プロセス機能を構成した Salesforce 管理者が突然退職しました。

UAT サイクルの最後の部分では、ユーザーはサンドボックスですべての変更を完全にテストし、承認しました。これにより、注文処理機能を本番環境で稼働させる準備が整いました。

残念ながら、変更セットは開始されましたが、以前の管理者によって完了されていませんでした。展開を完了するために開発者が招聘されます。

本番環境に移行する必要がある構成の変更を識別するために、開発者は何をすべきでしょうか？

A. Salesforce セットアップでは、すべての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るオブジェクトを決定します。

B. メタデータ API とサポートされている開発 IDE を使用して、すべての構成をサンドボックスから本番環境にプッシュし、変更が失われないようにします。

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. 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 メタデータのすべての変更を本番環境メタデータに自動的にマージします。

正解: ([正解を表示します](#))

本番環境に移行する必要がある設定変更を特定する最良の方法は、設定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することです。設定監査証跡は、あなたや他の管理者が組織に対して行った最近の設定変更を追跡する機能です。設定監査証跡では最大180日間の履歴を表示でき、CSVファイルとしてダウンロードできます。設定監査証跡は、開発者が以前の管理者がオブジェクト、項目、ワークフロー、プロセス、フローなどの作成や変更など、どのような設定変更を行ったかを把握するのに役立ちます。開発者はこの情報を使用して変更セットを完成させ、本番環境に展開できます。参考 [設定監査証跡の表示]、[変更セット]

質問: 12

Universal Containers は、ユーザー設定を階層カスタム設定 User_Prefs_c に保存します。この設定にはチェックボックスフィールド Show_Help_c が含まれます。会社レベルのデフォルト設定は組織レベルで保存されますが、ユーザーレベルで上書きできます。ユーザーが設定を上書きしていない場合は、デフォルト設定が使用されます。

現在のユーザーの Show_Help_c 設定をどのように取得すればよいですか？

- A. ブール値 show = User_Prefs__c.getValues(UserInfo.getUserId()).Show_Help__c;
- B. ブール値 show = User_Prefs__c.getInstance().Show_Help__c;
- C. ブール値 show = User_Prefs__c.Show_Help__c;
- D. ブール値 show = User_Prefs__c.getValues().Show_Help__c;

正解: ([正解を表示します](#))

階層カスタム設定から現在のユーザーの Show_Help__c 設定を取得するには、getInstance() メソッドを使用します。このメソッドは、ユーザーのカスタム設定データ、またはユーザー固有の設定が見つからない場合は組織のデフォルト設定を取得します。参考 :Salesforceヘルプ - GetInstance メソッド

質問: 13

開発者には、Case の標準コントローラをオーバーライドする 2 つの拡張機能を含むページがあります。

```
<apex:page standardController="Case" extension="CaseExtensionOne,CaseExtensionTwo" showHeader="false"> Each extension has a method called Save. The page has a command button as defined: <apex:commandButton value="Save" action="{!save}"/>
```

ユーザーがコマンド ボタンをクリックすると何が起こりますか？

- A. ケース標準コントローラからの保存が実行されます
- B. CaseExtensionOneからの保存が実行されます
- C. CaseExtensionTwoからの保存が実行されます
- D. 3つのSaveメソッドすべてが実行されます

正解: ([正解を表示します](#))

質問: 14

<apex:message> の正しい定義を選択してください

- A. Salesforce標準のフォーマット。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
- B. 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る、書式設定のない単一のメッセージ
- C. 書式設定なし。ページ上のすべてのエラーを表示します。
- D. 標準の Salesforce フォーマット。ページに特定のメッセージを表示します。

正解: C ([コメントを發表する](#))

質問: 15

開発者は、大量のデータを持つオブジェクトに対してクエリを実行するときに、SOQLクエリの選択性が十分ではありません」という例外を受け取ります。

a. 問題を解決するにはどのような手順を踏む必要がありますか？

A. REST API への呼び出しを介して SOQL クエリを実行します。

B. FOR ループの一部として SOQL クエリを実行します。

C. SOQL クエリを非同期プロセス内に移動します。

D. SOQL クエリの WHERE 句で ID を使用します。

正解: ([正解を表示します](#))

質問: 16

Cloud KicksのSalesforce管理者は、米国とCloudのすべての郵便番号を保存するためにRegion__cというカスタムオブジェクトを作成しました。

郵便番号が属するキックの販売地域。



Cloud Kicks では、リードの郵便番号に基づいて地域を入力するためのリードトリガーを必要としています。

A)

```

// ...
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);
    }
}

```

B)

```

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;
        }
    }
}

```

```

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);
    }
}

```

D)

```

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

```

- A. オプションC
- B. オプションD
- C. オプションB
- D. オプションA

正解: D ([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: 17

開発者は、次の要件を満たすコードを作成するように求められています。次の入力を受け取りま
 す。

Map<Id, Project__c>、List<Account> は、潜在的に長時間実行される外部 Web サービスへの呼び
 出しを実行します。プロセスが正常に実行されたことを確認する方法を提供します。どの非同期
 機能を使用する必要がありますか？

- A. Database.AllowsCallouts インターフェース

- B. キュー可能なインターフェース
- C. スケジュール可能なインターフェース
- D. @future (コールアウト=true)

正解: ([正解を表示します](#))

質問: 18

開発者は、トリガー、ワークフロー ルール、Apex クラス、プロセスを含む複雑なアプリケーションを作成しています。

開発者はアプリケーションを開発する際に、実行順序を慎重に検討する必要があります。以下の操作はどのような順序で実行されますか？ 1. トリガーの前 2. トリガーの後 3. メール送信などのコミット後ロジック 4. データベースにコミットされたDML 5. ワークフロールール 6. ロールアップ集計計算

- A. 1、6、5、2、4、3
- B. 1、2、4、5、6、3
- C. 1、2、5、6、4、3
- D. 1、5、6、2、4、3

正解: ([正解を表示します](#))

質問: 19

カスタム開発の一環として、開発者は特定の商談の進捗状況を時間経過とともに表示する Lightningコンポーネントを作成します。このコンポーネントでは、以下のいずれかの項目が変更された際に日付スタンプを表示する必要があります。

* 金額の見込み、ステージ、またはクローズ日

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. [Opportunity FieldHistory から NewValue、Oldvalue を選択 WHERE OpportunityId = :oppld AND Field IN
- B. ('StageName', Probability',Amount /CloseDate)];
- C. [OpportunityField_Historyから新しい値、古い値を選択。Oppportunity = oppid AND Field IN
- D. [Amount CloseDate. StageName, Probability FROM Opportunity_History WHERE OpportunityId - :oppld を選択];
- E. [OpportunityHistory から Amount、CloseDate、StageName、Probability を選択] = :oppld];

正解: ([正解を表示します](#))

質問: 20

ソフトウェア会社は、ソフトウェアの欠陥を追跡するためにカスタムオブジェクトDefect_cを使用しています。Defect__cには組織全体のデフォルトが設定されています。

プライベート 各 Defect__c には Reviewer_c レコードの関連リストがあり、各リストには User への参照フィールドがあり、User が Defect_c をレビューすることを示すために使用されます。

Reviewer_c レコードのユーザーに、Reviewer_c レコードの Defect_c レコードへの読み取り専用アクセス権を付与するには何を使用すればよいですか？

- A. Apex 管理共有
- B. Defect_c のすべてを表示
- C. Lightning Web コンポーネント
- D. 基準に基づく共有

正解: ([正解を表示します](#))

質問: 21

次の例外のうち、キャッチできずにエラーを強制するものはどれですか？ (3 つ選択してください。)

- A. 制限例外
- B. アサート例外
- C. SObjectExceptions
- D. DML例外
- E. ライセンス例外
- F. リスト例外

正解: ([正解を表示します](#))

/参照 :

SObjectException は、クエリされていないフィールドにアクセスしたとき、または間違った dml ステートメント中にフィールドを変更しようとしたとき (つまり、挿入時に編集専用フィールドが設定されているとき) に発生します。カスタム例外は、Exception クラスを拡張する必要があります。

質問: 22

開発者がコード実行時に [LimitException: クエリ行数が多すぎます: 50001] というエラーを受け取りました。開発者コンソールを使用したデバッグ方法の中で、予期しない行数を返している可能性のある特定のコンポーネントを最も迅速かつ正確に特定できる方法は何ですか？

- A. デバッグ ログ内の行制限警告メッセージの数をカウントします。
- B. 実行概要を使用して、実行された各ユニットによって返される行数を確認します。
- C. SOQL の使用状況を追跡するには、コードに System.debug (System.getQueryRows ()) を追加します。
- D. SOQL_EXECUTE_END ステートメントのデバッグ ログをフィルターして、各 SOQL クエリの結果を追跡します。

正解: ([正解を表示します](#))

質問: 23

開発者は、連絡先とカスタムオブジェクト Survey_Response__c に入力されたメールアドレスが、ブラックリストに登録されたドメインのリストに含まれていないことを確認する必要があります。

まず、ブラックリストに登録されたドメインのリストは、簡単に管理できるようにカスタムオブジェクトに保存されます。

ユーザーによるメンテナンス。Survey_Response__c オブジェクトはカスタム Visualforce ページから入力されることに注意してください。

これを実装する最適な方法は何ですか？

- A. 連絡先の検証ルールと Survey_Response__c オブジェクトの検証ルールにロジックを実装します。
- B. 連絡先の Apex トリガーにロジックを実装し、カスタム Visualforce ページ コントローラ内にもロジックを実装します。
- C. 連絡先の Apex トリガーおよびカスタム Visualforce ページ コントローラから呼び出されるヘルパー クラスにロジックを実装します。
- D. カスタム Visualforce ページ コントローラにロジックを実装し、連絡先の Apex トリガーからそのメソッドを呼び出します。

正解: ([正解を表示します](#))

質問: 24

20万件の取引先レコードの大規模データセットに使用できる選択的SOQLクエリはどれですか？
2つの回答を選択してください。

- A. Name = Null AND Customer_Number_c = 'ValueA' の場合、ACCOUNT から id を選択
- B. SELECT id FROM ACCOUNT WHERE id IN (アカウントIDのリスト)
- C. アカUNTから ID を選択、名前 = !: ''
- D. アカUNTからIDを選択 WHERE Name = Null

正解: ([正解を表示します](#))

質問: 25

開発者がLightning Webコンポーネントで問題が発生しています。このコンポーネントは、現在ログインしているユーザーが所有する商談に関する情報を表示する必要があります。

コンポーネントがレンダリングされると、次のメッセージが表示されます: 「データ取得エラー」

この問題を克服するには、Apex クラスにどのような変更を実装する必要がありますか？

- A. Apex メソッドで Continuation=true 属性を使用します。
- B. Opportunity オブジェクトの OWD がパブリックであることを確認します。
- C. Apex クラスで without sharing キーワードを使用するようにコードを編集します。
- D. Apex メソッドで Cacheable=True 属性を使用します。

正解: D ([コメントを公表する](#))

質問: 26

企業は見積 発注プロセスの一環として、OAuth 2.0をサポートするドキュメントストレージシステムのRESTエンドポイントにPOFを送信する必要があります。各Salesforceユーザーは、PDFを送信するにあたり、ドキュメントストレージシステムで個別に認証を受ける必要があります。

開発者が REST エンドポイントへの認証を実装する最適な方法は何ですか？

- A. OAuth トークンカスタムフィールドを使用した階層カスタム設定
- B. パスワード認証による名前付き認証情報
- C. 2つのパスワードカスタムフィールドを持つ階層カスタム設定
- D. OAuth 認証プロバイダによる名前付き認証情報

正解: **D** ([コメントを發表する](#))

質問: 27

Lightning Web コンポーネントがロードされたときにカスタムロジックを実行できる手法はどれですか？

- A. <aura:handler> init イベントを使用して関数を呼び出します。
- B. connectedCallback() メソッドを使用します。
- C. renderedCallback() メソッドを使用します。
- D. \$A.enqueueAction を呼び出し、呼び出すメソッドを渡します。

正解: **B** ([コメントを發表する](#))

Lightning Web Components (LWC) では、コンポーネントのライフサイクルは標準の Web Component ライフサイクルフックによって管理されます。コンポーネントがドキュメントオブジェクトモデル (DOM) に挿入されたときにロジックを 1 回だけ実行するには、開発者は connectedCallback() メソッド (オプション B) を使用する必要があります。

connectedCallback() は、LWC における Aura の init イベントに相当します。以下の処理に最適です。

- * 初期データ取得を実行します (命令型 Apex 経由)。
- * 内部状態または変数を初期化します。
- * Lightning Message Service (LMS) チャンネルをサブスクライブします。
- * 親コンポーネントとの通信を確立します。

オプション A と D は Aura 固有の構文であり、LWC では無効です。オプション C (renderedCallback()) は、コンポーネントのレンダリングサイクルが完了するたびに実行されます。コンポーネントは (リアクティブプロパティが変更されるたびに) 何度再レンダリングされる可能性があるため、renderedCallback に初期化ロジックを配置すると、慎重に処理しないとパフォーマンスの問題や無限ループが発生する可能性があります。したがって、connectedCallback() は、LWC で「ロード時」ロジックを処理するための標準的かつ最も効率的な方法です。

質問: 28

開発者が Chatter 内の特定のユーザー レコードを誰がフォローしているかをプログラムで判断するために使用できるオブジェクトはどれですか。

- A. エンティティサブスクリプション
- B. エンティティフィヒストリー
- C. フォローサブスクリプション
- D. 履歴をフォロー

正解: **A** ([コメントを發表する](#))

質問: 29

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すると、どの構文が Apex コードをテストしますか?

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B
- B. オプションD
- C. オプションA
- D. オプションC

正解: ([正解を表示します](#))

質問: 30

現在、企業には外部の注文管理システム (OMS) から Salesforce に注文を手動でアップロードするプロセスがあります。

IDを取得するには、Salesforceからアカウントをエクスポートする必要があるため、これは非常に手間のかかるプロセスです。注文と対応するアカウントを関連付けるには、アップロードファイルを正しいアカウントIDで更新する必要があります。

このプロセスをより効率的にするには、どの2つの推奨事項が適していますか？

2つの回答を選択してください

- A. データローダーの挿入ウィザードを使用してデータをインポートします。
- B. ファイル内のデータが注文 ID でソートされていることを確認します。
- C. 注文とアカウントの一意のフィールドを識別し、外部 ID として設定します。
- D. データローダーのアップサートウィザードを使用してデータをインポートします。

正解: B,C ([コメントを發表する](#))

質問: 31

Universal Charities (UC) は、Salesforce を使用して、個人や企業からクレジットカード引き落としの形で電子寄付を収集しています。

カスタマー サービス エージェントがクレジットカード情報を入力すると、寄付を処理するためにその情報が8つのサードパーティ決済プロセッサに送信され、UC は個人用に1つの決済プロセッサを使用し、法人用に別の決済プロセッサを使用します。

開発者は、さまざまな支払いプロセッサの支払いプロセッサ設定を保存して、必要に応じて展開後にシステム管理者が設定を変更できるようにするために、何を使用する必要がありますか？

- A. カスタムオブジェクト
- B. カスタムメタデータ
- C. 階層のカスタム設定
- D. カスタム設定をリストする

正解: C ([コメントを發表する](#))

階層カスタム設定は、様々な決済処理業者の設定を保存するための手段です。これにより、システム管理者は導入後に必要に応じて設定を変更できます。階層カスタム設定は、ユーザーやプロファイルごとに異なる設定データを保存できるカスタム設定の一種です。階層カスタム設定は Apexコードからアクセスでき、ユーザーのコンテキストに基づいてコードの動作を制御するために使用できます。階層カスタム設定を使用することで、開発者は個人顧客と法人顧客の決済処理業者のURLと認証情報を保存できるカスタム設定を作成できます。開発者は、Apexコードでこのカスタム設定を使用することで、ユーザーのプロファイルまたはIDに基づいて使用する決済処理業者を決定できます。また、システム管理者は、決済処理業者の設定が変更された場合、コードを変更することなく、ユーザーインターフェースでカスタム設定値を変更することもできます。参照: [階層カスタム設定]、[カスタム設定へのアクセス]

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

開発者は、関連付けられた商談が高価値と判断された際にアカウントの評価を更新する商談トリガーを作成しました。現在、商談が高価値と判断される基準は、金額が1,000,000ドル以上です。ただし、この基準値は時間の経過とともに変更される可能性があります。

Lightning Web コンポーネントに高価値の商談も表示するという新しい要件があります。これらのビジネス要件を満たし、高価値の機会を獲得するビジネス ロジックが複数の場所で繰り返されるのを防ぐために、開発者が実行する必要がある 2 つのアクションはどれですか。

2つの回答を選択してください

- A. カスタム メタデータを使用して高額の金額を保持します。
- B. 効率性を高めるために、ビジネス ロジック コードをトリガー内に残します。
- C. Lightning Web コンポーネントからトリガーを呼び出します。
- D. 高価値な商談を取得するヘルパークラスを作成します。

正解: ([正解を表示します](#))

質問: 33

開発者が外部ウェブ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ます。@future メソッドで呼び出しを行う必要があるのはどのようなシナリオですか？

- A. コールアウトが完了するまでに 60 秒以上かかる可能性があります。
- B. コールアウトは Apex トリガーで実行されます。
- C. 1 回のトランザクションで 10 回以上のコールアウトが行われます。どれがお好みですか？
- D. コールアウトは Apex テストクラスで行われます。

正解: B ([コメントを発表する](#))

質問: 34

ある企業のWebページから、取引先レコードの情報（名前 ウェブサイト、従業員番号など）を取得する必要があります。Salesforceレコード1DはWebページによって認識されており、JavaScriptを使用して取引先情報を取得します。

どの統合方法が最適ですか？

- A. Apex SOAP Webサービス
- B. SOAP API
- C. Apex REST Webサービス
- D. REST API

正解: **A** ([コメントを发表する](#))

このシナリオに最適な統合方法は、Apex SOAP Web サービスです。Apex SOAP Web サービスは、SOAP プロトコルを使用して外部システムから呼び出すことができるメソッドを公開するカスタム Apex クラスです。Apex SOAP Web サービスを使用すると、JavaScript を使用する Web ページに Salesforce のデータと機能を公開できます。ただし、その Web ページが Web サービスを定義する WSDL (Web サービス記述言語) ファイルにアクセスする必要があります。Apex SOAP Web サービスは、ガバナ制限、テスト、デバッグなど、Apex の利点を活用することもできます。SOAP API は、外部システムが SOAP プロトコルを使用して Salesforce のデータとメタデータにアクセスし、操作できるようにする標準の Salesforce API です。SOAP API を使用して JavaScript を使用する Web ページと統合することもできますが、特定のユースケースに合わせてカスタマイズされないため、Apex SOAP Web サービスよりも多くのコーディングと設定が必要になります。Apex REST Web サービスは、RESTful 原則と JSON または XML 形式を使用して外部システムから呼び出すことができるメソッドを公開するカスタム Apex クラスです。Apex REST Web サービスは、JavaScript を使用する Web ページとの統合にも使用できますが、Web ページで JSON または XML レスポンスを解析し、認証と承認を処理する必要があるため、このシナリオには最適ではありません。REST API は、外部システムが RESTful な原則と JSON または XML 形式を使用して Salesforce のデータとメタデータにアクセスし、操作できるようにする標準の Salesforce API です。REST API は、JavaScript を使用する Web ページとの統合にも使用できますが、Web ページで JSON または XML レスポンスを解析し、認証と承認を処理する必要があるため、このシナリオには最適ではありません。参考資料 [\[Apex Web サービスとコールアウト\]](#)、[\[SOAP API 開発者ガイド\]](#)、[\[Apex REST Web サービス\]](#)、[\[REST API 開発者ガイド\]](#)

質問: **35**

ある企業には、注文を行うためのネイティブ iOS アプリがあり、このアプリを Salesforce に接続して、JSON 形式のさまざまなオブジェクトから統合された情報を取得する必要があります。これを Salesforce に実装する最適な方法はどれですか？

- A. Apex REST コールアウト
- B. Apex REST Web サービス
- C. Apex SOAP Web サービス
- D. Apex SOAP コールアウト

正解: ([正解を表示します](#))

質問: **36**

Universal Containers では、バッチ Apex プロセスの終了メソッドからサポート担当者に直接電子メールを送信するのを待つのではなく、処理されない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. `Eventbus.publish()` メソッドを使用してエラー イベントを公開します。
- B. 公開は不要です。外部システムで `BatchapexErrorEvent` をサブスクライブしてください。

C. addError メソッドを使用してエラー イベントを公開します。

D. フローを使用してエラー イベントを公開します。

正解: ([正解を表示します](#))

通知にはBatchApexErrorEventを使用してください。バッチApexで未処理の例外が発生した場合、自動的にトリガーされます。手動での公開は不要です。外部システムからこのイベントをサブスクライブして通知を受け取ることができます。参考: バッチApex

質問: 37

現在、企業には外部 OMS から注文を手動でアップロードするという、手間のかかるプロセスがあります。

注文をリンクするためのIDを取得するには、アカウントをエクスポートする必要があります。このプロセスをより効率的にするには、どの2つの推奨事項が適切でしょうか？

A. データローダーの挿入ウィザードを使用してデータをインポートします。

B. ファイル内のデータが注文 ID でソートされていることを確認します。

C. データローダーのアップサートウィザードを使用してデータをインポートします。

D. 注文とアカウントの一意のフィールドを識別し、外部 ID として設定します。

正解: ([正解を表示します](#))

現在のボトルネックは、注文をインポートする前に、取引先検索でSalesforce内部ID \$001...\$)を解決する必要があることに起因しています。これは、外部ID オプションD)とUpsert操作 オプションC)を使用することで完全に回避できます。

* 外部 ID: OMS の一意のフィールド (OMS_Account_Number__c など) を Salesforce の外部 ID としてマークすることで、開発者は Salesforce が理解できる 外部キー」を作成します。

* Upsertウィザード :データローダのUpsertウィザードを使用するとき、親アカウントと照合するためにどの項目を使用するか尋ねられます。OMS_Account_Number__c外部IDを選択すると、データローダはCSVファイル内のOMS番号を使用して注文をアカウントに直接マッピングできます。

Salesforceはバックグラウンドで正しいアカウントIDを自動的に検出します。これにより、エクスポートとVLOOKUP」の手順が完全に不要になります。オプションA (挿入では依然として内部IDが必要です。オプションB (並べ替え)は、大量のデータ読み込み時のパフォーマンス向上に役立ちますが、手動でのID照合の問題は解決しません。外部IDを使用したUpsertの使用は、定期的なデータ統合を効率化するためのプラットフォーム推奨の方法です。

質問: 38

ユニバーサルコンテナーズは、顧客サポートケースの管理にカスタム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、特定のケース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ケースの可視性とアクセスを制御したいと考えています。この要件を満たすのに役立つSalesforceの機能はどれでしょうか？

A. Apex管理共有23

B. 共有セット24

C. ロール階層25

D. 基準に基づく共有ルール26

正解: ([正解を表示します](#))

28

外部ユーザー (Community/Experience Cloud ユーザー) を扱う場合、ロール階層などの標準の共有メカニズムは適用されないことが多く、特にロール階層をまったく使用しない大量の Customer Community ライセンスの場合は当てはまりません。

共有セット オプションB)は、外部ユーザーに取引先または取引先責任者に関連付けられたレコードへのアクセスを許可するために特別に設計されています。共有セットを使用すると、管理者は簡単なマッピングを定義できます。例えば、

「ケースの取引先がユーザの取引先と一致するケースへのアクセスを許可します。」これは、複雑な Apex コードを使用せずに、外部パートナーに関連するレコードを表示できる最も効率的でスケーラブルな方法です。

Apexによる共有管理 オプションA)は技術的には機能しますが、管理がはるかに複雑で、

Customer Community Plus」または Partner」ライセンスでのみ利用可能です。条件に基づく共有ルール オプションD)は、通常、社内ユーザーや広範なグループに使用され、個々の外部パートナーに対するきめ細かなアカウント固有のアクセス制御には使用されません。共有セットは、外部レコードの可視性を制御するための、高性能で宣言的なソリューションを提供します。

質問: 39

開発者が最近、商談トリガから呼び出されるApexコードで複雑なコミッション計算を実行する機能を本番環境にリリースしました。ユーザーから、コミッションの値が間違っているため、計算が間違っているように見えるという報告がありました。

開発者は開発者サンドボックスに代表的なテスト データを保持しています。

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、開発者が使用する必要がある 3 つのツールまたはテクニックはどれですか。

3つの回答を選択してください

A. ブレークポイント

B. ワークベンチ

C. ビジュアルスタジオコード

D. 開発者コンソール

E. Apex リプレイデバッガー

正解: ([正解を表示します](#))

質問: 40

ある企業では、ユーザーが特定のオブジェクトタイプのレコードを検索できるカスタムコンポーネントを運用しています。Apexコントローラを呼び出すことで、ユーザーの入力に基づいて結果リストが返されます。検索が完了すると、searchCompleteイベントが発行され、結果がイベントの

results属性に格納されます。このコンポーネントは他のコンポーネント内で使用できるように設計されており、1つのページに複数回表示されることがあります。
検索が完了したときにイベントを発生させるために追加する必要がある最適なコードは何ですか？

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

A.

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

B.

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

C.

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

D.

正解: ([正解を表示します](#))

質問: 41

Universal Containers には、アカウントが顧客としてマークされたときにアカウント プラン レコードを作成する、アカウントに対する Apex トリガーがあります。
最近、レコード トリガー フローが追加され、アカウントが顧客としてマークされるたびに、「顧客 開始」日付フィールドが今日の日付で更新されるようになりました。フローの追加以降、アカウントが顧客としてマークされるたびに2つのアカウント プラン レコードが作成されるようになりました。

これが起こる原因は何でしょうか？

- A. フローは、レコードが作成された時点と編集されるたびに評価するように構成されています。
- B. フローは「レコードの更新」要素を使用するように構成されています。
- C. Apex トリガーは、1 回だけ実行されることを保証するために静的変数を使用しません。
- D. Apex トリガーは一括安全ではなく、for ループ内で insert を呼び出します。

正解: ([正解を表示します](#))

質問: 42

Visualforce ページの「get」リクエスト（最初のページ訪問）の正しい実行順序は何ですか？

- A. 1) 存在するカスタムコンポーネントの属性定義のコンストラクタ、拡張機能、および式を評価する
- 2) コントローラと拡張機能のコンストラクタを評価する

3) メインページで式、<apex:page>属性アクション、その他のメソッド呼び出し (ゲッター/セッター) を評価する

4) <apex:form>要素がある場合は、ビューステートを作成します。

5) HTMLをブラウザに送信する

B. 1) コントローラと拡張機能のコンストラクタを評価する

2) 存在するカスタムコンポーネントの属性定義のコンストラクタ、拡張機能、および式を評価する

3) メインページで式、<apex:page>属性アクション、その他のメソッド呼び出し (ゲッター/セッター) を評価する

4) <apex:form>要素がある場合は、ビューステートを作成します。

5) HTMLをブラウザに送信する

C. 1) コントローラと拡張機能のコンストラクタを評価する

2) <apex:form>要素がある場合は、ビューステートを作成します。

3) メインページで式、<apex:page>属性アクション、その他のメソッド呼び出し (ゲッター/セッター) を評価する

4) 存在するカスタムコンポーネントの属性定義のコンストラクタ、拡張機能、および式を評価する

5) HTMLをブラウザに送信する

正解: ([正解を表示します](#))

質問: 43

Universal Containers は、ユーザー設定を階層カスタム設定 User_Prefs__c に保存します。この設定にはチェックボックスフィールド Show_Help__c が含まれます。会社レベルのデフォルト設定は組織レベルで保存されますが、ユーザーレベルで上書きできます。ユーザーが設定を上書きしていない場合は、デフォルト設定が使用されます。

現在のユーザーの Show_Help__c 設定をどのように取得すればよいですか？

A. ブール値 show = User_Prefs__c.getValues(UserInfo.getUserId()).Show_Help__c;

B. ブール値 show = User_Prefs__c.getInstance().Show_Help__c;

C. ブール値 show = User_Prefs__c.Show_Help__c;

D. ブール値 show = User_Prefs__c.getValues().Show_Help__c;

正解: ([正解を表示します](#))

説明/参考資料: <https://developer.salesforce.com/forums/?id=906F00000008ySqlAI>

質問: 44

Universal Containersは、カスタムLightningページを使用して、ステップバイステップのウィザードで取引先を検索できるメカニズムを提供しています。ウィザードのステップの1つは、ユーザーがテキスト項目 (ERF Number__c) にテキストを入力できるようにすることです。入力されたテキストは、一致する取引先を検索するためのクエリで使用されます。

```
erpNumber = erpNumber + '%';  
List<Account> accounts = [SELECT Id, Name FROM Account WHERE  
ERP_Number__c LIKE :erpNumber]
```

salesforce

問題を解決するにはどのような手順を踏む必要がありますか？

- A. for ループの一部として SOQL クエリを実行します。
- B. ERP_Numker = フィールドを外部 ID としてマークします。
- C. ERP_Number__c フィールドを必須としてマークします。
- D. SOQL クエリを非同期プロセス内に移動します。

正解: ([正解を表示します](#))

この問題を解決するには、ERP_Number__c フィールドを外部 ID としてマークする必要があります。外部 ID とは、外部 ID 属性を持つカスタム フィールドです。つまり、Salesforce 外部のシステムからの一意のレコード識別子が含まれます。フィールドが外部 ID としてマークされると、Salesforce は自動的にインデックスを作成し、SOQL クエリの WHERE 句などのフィルターで使えるようになります。これにより、クエリのパフォーマンスが向上し、非選択クエリ例外を回避できます。SOQL クエリを for ループの一部として実行しても、クエリ制限に達して効率が悪くなるため、問題は解決されません。ERP_Number__c フィールドを必須としてマークしても、クエリのパフォーマンスやフィールドのインデックス作成可能性には影響しないため、問題は解決されません。SOQL クエリを非同期プロセス内に移動しても、同じクエリ制限と例外が発生するため、問題は解決されません。参考資料: [カスタム フィールド属性]、[非常に大きな SOQL クエリの操作]

質問: 45

Aura コンポーネントから呼び出され、クラスにラップされたデータを返す以下のコントローラーコードを検討してください。

```

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

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

        MyDataWrapper theData = new MyDataWrapper();

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

```

開発者は、クエリがそれぞれ 1 つのレコードを返すこと、および Aura コンポーネントでエラー処理が行われることを確認しましたが、コントローラ getSomeData を呼び出してもコンポーネントは何も返されません。

なにが問題ですか？

- A. MyDataWrapper などの Apex クラスのインスタンスは、Lightning コンポーネントに返すことができません。
- B. メンバーの名前とオプションには、ゲッターとセッターを設定しないでください。
- C. メンバーの名前とオプションは public として宣言しないでください。
- D. クラス MyDataWrapper のメンバーの Name とオプションにも @AuraEnabled アノテーションを付ける必要があります。

正解: **D** ([コメントを發表する](#))

Aura コンポーネントにデータを渡すには、コンポーネントからアクセスされる Apex クラスのすべてのプロパティに @AuraEnabled アノテーションを付与する必要があります。このアノテーションがないと、Aura フレームワークはプロパティをシリアル化してフロントエンドに送信できません。

参考資料: Aura コンポーネント開発者ガイド - AuraEnabled アノテーション

テストクラスで Vizualforce ページを初期化するためのベスト プラクティスは何ですか？

- A. controller.currentPage.setPage(MyTastfage) を使用します。
- B. Test.setCurrentPage(Page.MyTeatPage); を使用します。
- C. Test.setCurrentPage MyTestPags を使用します。
- D. Test.currantPage.getParamatars を使用します (MyTaestPaga) ;

正解: **B** ([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: **47**

開発者が、アウトバウンドRESTfulコールアウトを実行するApexクラスを作成しました。以下は、Apexテストメソッドで偽のレスポンスを送信するために作成されたものです。

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

テストメソッドでこの偽の応答を返すにはどのメソッドを呼び出すことができますか？

- A. テスト.setTestData
- B. テストセットモック
- C. テストセットアップ
- D. テスト作成スタブ

正解: ([正解を表示します](#))

質問: **48**

企業は、商談のレコードタイプに基づいて異なるロジックを実行したいと考えています。どのコードセグメントがこの要求を処理し、ベスト プラクティスに従っていますか？

```
List<RecordType> recTypes = [SELECT Id, Name FROM
RecordType WHERE SOBJECTTYPE = 'Opportunity' AND
```

A.

```
RecordType recType_New = [SELECT Id, Name FROM  
RecordType WHERE SOBJECTType = 'Opportunity' AND  
IsActive = True And DeveloperName = 'New' LIMIT  
1];
```

```
RecordType recType_Renewal = [SELECT Id, Name FROM  
RecordType WHERE SOBJECTType = 'Opportunity' AND  
IsActive = True And DeveloperName = 'Renewal'  
LIMIT 1];
```

```
for(Opportunity o : Trigger.new) {  
    if(o.RecordTypeId == recType_New ) {  
        // do some logic Record Type 1  
    }  
    else if (o.RecordTypeId == recType_Renewal ) {  
        // do some logic for Record Type 2  
    }  
}
```

B.

salesforce

C.

```
Id recType_New =
Schema.SObjectType.Opportunity.getRecordTypeInfoById('New');

Id recType_Renewal =
Schema.SObjectType.Opportunity.getRecordTypeInfoById('Renewal');

for(Opportunity o : Trigger.new) {
    if(o.RecordTypeId == recType_New ) {
        // do some logic Record Type 1
    }
    else if (o.RecordTypeId == recType_Renewal ) {
        // do some logic for Record Type 2
    }
}
```

salesforce

D.

```
for(Opportunity o : Trigger.new) {
    if(o.RecordType.Name == 'New') {
        // do some logic Record Type 1
    }
}
```

正解: C ([コメントを发表する](#))

質問: 49

開発者は次のテストメソッドを作成しました。

ジャワ

@isTest(すべてのデータを参照 = true)

パブリック静的void testDeleteTrigger(){

アカウント testAccount = 新しいアカウント(name = 'Test1');

testAccount を挿入します。

List<Account> testAccounts = [SELECT Id, Name from Account WHERE Name like 'Test%'];

System.assert(testAccounts.size() > 0); delete testAccounts; testAccounts = [SELECT Id, Name from Account WHERE Name like 'Test%']; System.assert(testAccounts.size() == 0);

}

開発者組織には、名前が「Test」で始まるアカウントが5つあります。開発者は開発者コンソールでこのテストを実行します。

テストコードを実行した後、正しい記述はどれですか。

- A. 名前が「Test」で始まるアカウントが5つあります。
- B. 名前が「Test」で始まるアカウントはありません。
- C. 名前が「Test」で始まるアカウントが6つあります。
- D. テストは失敗します。

正解: ([正解を表示します](#))

すべてのApexテスト実行は、一時的なトランザクションとして扱われます。この実行における最も重要な点の一つは、データベースに加えられた変更（レコードの挿入、変更、削除など）が、テスト実行の終了時に自動的にロールバックされることです。このメカニズムにより、ユニットテストによって「ゴミ」データが残されたり、組織の実際のビジネスデータの状態が変更されたりすることがなくなり、開発環境と本番環境の一貫性が維持されます。1234 このシナリオで

は、@isTest(SeeAllData=true)アノテーションが使用されています。これにより、テストメソッドは組織内の既存データを参照できます。組織には既に「Test」で始まる5つのアカウントが含まれているため、テストメソッドはそれらを参照します。テストコードが「Test1」という名前の新しいアカウントを挿入すると、そのトランザクション内のカウントは6になります。後続のコードは、これらの6つのアカウントをクエリし、削除します。

テストの実行中は削除が成功し、ゼロアカウントのアサーションチェックが成功します。

しかし、テストが終了すると、トランザクション全体がロールバックされます。このロールバックにより、元の5つのアカウントの削除と1つの新規アカウントの作成が実質的に「元に戻される」こととなります。その結果、組織はテスト開始前の状態に戻ります。元の5つのアカウントは、テストが行われなかったかのようにデータベースに残ります。そのため、選択肢Aが正解となります。

質問: 50

クエリのどこで地理位置情報と距離を使用できますか？

- A. 選択句
- B. Order By句
- C. Group By句
- D. フィルター句

正解: ([正解を表示します](#))

質問: 51

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ればよいでしょうか？

- A. テストクラスにすべ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. Salesforce にレコードを挿入し、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 を使用して、アサーションを実行します。

D. テスト クラスにすべてのテスト データを作成し、Test.startTest{} と Test.stopTest {} を呼び出してアサーションを実行します。

正解: **D** ([コメントを發表する](#))

Test.startTest() と Test.stopTest() を使用します。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、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: 52

ある企業では、特定の地理的地域のデフォルト情報と削除動作を表す参照データを複数のカスタム メタデータ レコードに保存しています。

連絡先を挿入するときに、連絡先の住所情報に基づいて、カスタム メタデータ レコードから連絡先にデフォルト情報が設定される必要があります。

さらに、フラグが付けられた地域に属する連絡先をユーザーが削除しようとする、エラー メッセージが表示される必要があります。

会社の人的リソースに応じて、これを自動化する 2 つの方法は何ですか？

2つの回答を選択してください

A. リモートアクション

B. フロービルダー

C. Apexトリガー

D. Apex呼び出し可能メソッド

正解: ([正解を表示します](#))

このプロセスを自動化する方法は 2 つあり、Flow Builder と Apex トリガを使用する方法です。Flow Builder は、開発者がロジックの実行、データの操作、Salesforce データベースとのやり取りができる宣言型ワークフローを作成できるツールです。開発者は Flow Builder を使用して、連絡先が挿入されたときに実行されるレコードトリガフローを作成し、連絡先の住所情報に基づいてカスタムメタデータレコードから連絡先のデフォルト情報を設定できます。また、開発者は Flow Builder を使用して、ユーザーが連絡先を削除しようとしたときに実行される削除前フローを作成し、連絡先がフラグ付き地域に属している場合はエラーメッセージを表示することもできます。Apex トリガは、レコードが挿入、更新、削除、または復元されたときに開発者が Apex ロジックを実行できるコードベースのソリューションです。開発者は Apex トリガを使用して、カスタムメタデータレコードを照会し、連絡先の住所情報に基づいて一致するレコードから連絡先のデフォルト情報を設定する、連絡先の挿入後トリガを作成できます。開発者は、Apexトリガを使用して、連絡先にbefore-deleteトリガを作成することもできます。このトリガは、カスタムメタデータレコードを照会し、連絡先がフラグ付き地域に属している場合は例外をスローします。リモートアクションは有効な回答ではありません。これは、開発者がJavaScriptを使用してVisualforce ページからApexコントローラメソッドを呼び出すことを可能にする方法ですが、プロセスを自動化するものではありません。Apex呼び出し可能メソッドは有効な回答ではありません。これは、開発者がフローやプロセスビルダーなどのプロセスにApexメソッドを公開することを可能にする

方法ですが、それ自体ではプロセスを自動化するものではありません。参考資料 [Flow Builder]、[Apexトリガ]、[Apex開発者ガイド]

質問: 53

Universal Containers では、取引先に Apex トリガがあり、取引先が顧客としてマークされるとアカウントプランレコードが作成されます。最近、日付項目を更新するためのレコードトリガーフローが追加されました。フローの追加以降、2 つのアカウントプランレコードが作成されています。これは何が原因でしょうか？

- A. Apex トリガーは一括安全ではなく、for ループ内で insert を呼び出します。
- B. フローは、「レコードの更新」要素を使用するように構成されています。
- C. フローは、レコードが作成された時点と編集されるたびに評価するように構成されています。
- D. Apex トリガーは、1 回だけ実行されることを保証するために静的変数を使用しません。

正解: ([正解を表示します](#))

これは、Salesforce の実行順序によって発生するトリガーの再帰の典型的な例です。取引先が「顧客」に更新されると、以下の処理が発生します。

- * 元の「Before」トリガーと「After」トリガーが実行されます。Afterトリガーは最初のアカウントプランを作成します。
- * レコードトリガーフローが実行され、「顧客開始」日付フィールドが更新されます。
- * フロー (またはワークフロー) によってフィールドの更新が実行されたため、Salesforce はアカウントオブジェクトで Apex トリガーを再トリガーし、新しい日付フィールドに依存するロジックが確実に実行されるようにします。
- * この 2 回目のパス中に、After トリガーが再度実行され、ステータスがまだ「顧客」であることを確認して、2 番目のアカウントプランが作成されます。

これを防ぐため、開発者はハンドラクラスで静的変数 (オプションD) を使用して「再帰ガード」として機能します。静的変数はトランザクション全体にわたって保持されます。トリガの開始時にこの変数をチェックすることで、コードは既に1回実行されたことを検出し、2回目のパスでレコードの作成をスキップできます。オプションBとCは標準的なフローの動作を記述していますが、トリガロジックの「修正」ではありません。オプションAは制限に関する問題を引き起こしますが、再帰が関与しない限り、必ずしも二重発火を引き起こすわけではありません。

質問: 54

現在、企業には外部の注文管理システム (OMS) から Salesforce に注文を手動でアップロードするプロセスがあります。

IDを取得するには、Salesforceからアカウントをエクスポートする必要があるため、これは非常に手間のかかるプロセスです。注文と対応するアカウントを関連付けるには、アップロードファイルを正しいアカウントIDで更新する必要があります。

このプロセスをより効率的にするには、どの 2 つの推奨事項が適していますか？

2つの回答を選択してください

- A. データ ローダーの挿入ウィザードを使用してデータをインポートします。

- B. 注文とアカウントの一意のフィールドを識別し、外部 ID として設定します。
- C. データローダーのアップサートウィザードを使用してデータをインポートします。
- D. ファイル内のデータが注文 ID でソートされていることを確認します。

正解: ([正解を表示します](#))

質問: 55

Salesforce ユーザーは、m アカウントを保存するときに常に「トリガーの最大深度を超えました」というエラーを受け取ります。

開発者はこのエラーをどのように修正できますか？

- A. ヘルパー クラスを使用して、トリガーが最初に起動されるときにブール値を 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トリガーを変更します。
- B. トリガーを変換して future アノテーションを使用し、後続のトリガー呼び出しを Account オブジェクトに連鎖します。
- C. isMultiThread=true アノテーションを使用するようにトリガーを変更します。
- D. トリガー ロジックを 2 つの個別のトリガーに分割します。

正解: ([正解を表示します](#))

質問: 56

開発者が、単一のカスタムコントローラを使用してデータのクエリと更新を行う複数ページのウィザードを構築しました。ユーザーからページの読み込みが遅いという苦情が寄せられています。パフォーマンスを改善するにはどうすればよいでしょうか？3つの回答を選択してください。

- A. ビューステートを減らします。
- B. Apex ページ属性を cache=true に設定します。
- C. 標準スタイルシートをオフにします。
- D. actionRegion と rerender を使用します。
- E. 選択的クエリの使用。

正解: ([正解を表示します](#))

質問: 57

Lightningコンポーネントでは、ユーザーがボタンをクリックして変更を保存し、別のページにリダイレクトすることができます。現在、ユーザーが「保存」ボタンを押す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つのテクニックはどれですか？1

- A. ブラウザの開発ツールを使用して JavaScript をデバッグします。2
- B. 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。34
- C. JavaScriptでconsole.log()メッセージを使用します。56
- D. 開発者コンソールを使用して7つのチェックポイントを表示します。8
- E. ユーザーの Lightning コンポーネントのデバッグ モードを有効にします。9

正解: ([正解を表示します](#))

Lightning コンポーネントでクライアント側の JavaScript をデバッグするには、サーバー側の Apex をデバッグする場合とは異なるツールが必要です。

まず、ブラウザの開発者ツール オプションA)を使用するのが最も強力な方法です。これにより、開発者はコントローラファイルまたはヘルパーファイルにブレークポイントを設定し、コンポーネント属性の値をリアルタイムで検査し、リダイレクトロジックを1行ずつステップ実行できます。このプロセスを容易にするために、開発者はSalesforceの「設定」でデバッグモード オプションE)を有効化する必要があります。デフォルトでは、Lightningコードはパフォーマンス向上のために「縮小」(圧縮されているため、開発ツールで読み取ることはほぼ不可能です。デバッグモードを有効にすると、圧縮されていないソースコードがブラウザに配信され、スクリプトが人間が読める形式になります。

さらに、`console.log()` メッセージを挿入する オプションC)は、変数の状態や実行マーカをブラウザのコンソールに出力するための標準的な手法です。これにより、`helper.saveAndRedirect` 関数が実際に実行されているかどうか、またリダイレクトパラメータが正しいかどうかを確認できます。

選択肢BとDは不正解です。開発者コンソールのデバッグログとチェックポイントは、サーバーサイドApex実行のみを対象としているためです。JavaScriptエラーやブラウザでのLightningフレームワークの実行フローは記録されません。

質問: 58

セーブポイントに関して正しい記述はどれですか？

- A. セーブポイントへの参照はトリガー呼び出しをまたぐことができます。
- B. セーブポイントはDML文のガバナ制限によって制限されません
- C. 静的変数はロールバック中に元に戻されません。
- 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: 59

開発者が開発者サンドボックスでVisualforceページを作成し、テストしたところ、本番環境で使用する「View State」エラーが発生するという報告を受けました。これらのエラーを修正するには、開発者は何を確認すればよいでしょうか？

- A. プロファイルが Visualforce ページにアクセスできることを確認します。
- B. プロパティがプライベートとしてマークされていることを確認します。
- C. 変数が一時変数としてマークされていることを確認します。12
- D. クエリがガバナ制限を超えないようにします。34

正解: ([正解を表示します](#))

Visualforceのビューステートは、サーバーリクエスト間でページの状態 (コントローラ変数を含む)を維持する非表示のフォームフィールドです。Salesforceは、ビューステートに135KBという厳格な制限を設けています。ページが大量のデータ (大規模なレコードリストや複雑なオブジェクト

トなど)を処理する場合、コントローラ変数のシリアル化サイズがこの制限を簡単に超え、ユーザーにランタイムエラーが発生する可能性があります。

これらのエラーを修正するには、開発者は transient キーワード オプション C)を使用する必要があります。変数を transient としてマークすると、ビューステートにシリアル化されなくなります。これは、「ポストバック」ページ全体またはページの一部を更新するアクション)間で維持する必要のないデータに対する標準的なベストプラクティスです。

transient キーワードの一般的な候補は次のとおりです。

- * 表示専用を取得された大量のレコードのリスト。
- * 単一のリクエスト中の計算に使用される一時変数。
- * 必要に応じて簡単に再クエリできる概要データ。

オプションAはパフォーマンスではなくセキュリティに関係します。オプションB プライベート変数)では、ビューステートから変数は除外されません。除外されるのは静的変数と一時変数のみです。オプションDはSOQLガバナ制限に関係し、ビューステートのサイズとは無関係です。開発者は一時変数を適切に使用することで、ページのデータ量を抑え、実世界のデータ量を扱う本番環境においてスムーズなユーザーエクスペリエンスを実現できます。

質問: 60

ある企業は、前提条件となる参照データ用の多数のレコードを作成するためのメソッドが多数含まれるテストクラスでの単体テストが遅いことに気づきました。

開発者はこの問題に対処するために何ができるでしょうか？

- A. テストを実行するときに、トリガー、フロー、検証をオフにします。
- 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呼び出します。
- C. 前提条件となる参照データのセットアップをテストクラスの @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ーに移動します。

正解: [C \(コメントを發表する\)](#)

@testSetupメソッドはテスト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、冗長なデータ作成を削減できます。参考資料 testSetupメソッドの使用

質問: 61

<apex:messages> の正しい定義を選択してください。

- A. 標準の Salesforce フォーマット。ページに特定のメッセージを表示します。
- B. Salesforce標準のフォーマット。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
- C. 書式設定なし。ページ上のすべてのエラーを表示します。
- D. 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る、書式設定のない単一のメッセージ

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: 62

次の標準フィールドのうち、インデックス化されるのはどれですか。3つ選択してください。

- A. システムモッドスタンプ
- B. レコードタイプ
- C. 名前
- D. 最終更新日
- E. 作成者

正解: ([正解を表示します](#))

質問: 63

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_Count_c を増分します。

```
public class SalesforceTriggerHandler {  
    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);  
}
```

ケースが作成または更新されるときに、ケース オブジェクトに新しいプロセスが本番環境で作成されました。プロセスが作成されて以来、ケース編集時にカウントが1以上増加しているというレポートがあります。

Apex コードのどの変更によって問題が解決しますか？

```
public class SalesforceTriggerHandler {  
    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;  
        }  
    }  
}  
  
trigger on Case(before update) {  
    if (CaseTriggerHandler.firstRun) {  
        CaseTriggerHandler.handle(trigger.new);  
    }  
}
```

A.

```
public class CaseTriggerHandler {
    Boolean firstRun = true;
    public static void handle(List<Case> cases) {
        if (firstRun) {
```

B.

C.

```
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;
        }
    }
}
```

salesforce

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

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

D.

正解: ([正解を表示します](#))

質問: 64

テストカバレッジの作成にかかる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され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。これを実現するにはどのようなコードを使用できますか？

A. < meta charset="utf-8" />List< sObject> testAccounts = Test . loadData (Account.sObjectType, 'testAccounts');

B. リスト<sObject> testAccounts = Test.loadData(Account.sObjectType, SResource.testAccounts)

C. List<sObject> testAccounts = Test . loadData([StaticResourceからId,Bodyを選択]
=テストアカウント

D. List< sObject> testAccounts = (List< Count>) [StaticResource から Id、Body を選択、Name = testAccounts

正解: ([正解を表示します](#))

質問: 65

開発者は、時々変更される設定データをカスタム設定に保存していました。しかし、最近更新された一部のサンドボックスでテストが失敗するようになりました。

今後この問題を解消するには何をすべきでしょうか？

- A. カスタム設定の設定タイプをリストに設定します。
- B. カスタム設定を静的リソースに置き換えます。
- C. カスタム設定の設定タイプを階層に設定します。
- D. カスタム設定をカスタム メタデータに置き換えます。

正解: ([正解を表示します](#))

カスタム設定をカスタムメタデータ型に置き換えることは、環境間でテストデータの一貫性を確保するための優れたソリューションです。カスタムメタデータ型はデプロイ可能で、そのレコードはデフォルトでテストメソッドで使用できます。一方、リストカスタム設定は更新後に各環境で手動で再作成する必要があります。参考資料 :Salesforceヘルプ - カスタムメタデータ型の概要

質問: 66

開発者は、連絡先を素早く入力できるようにLightning Webコンポーネントを作成しましたが、エラーメッセージが表示されません。

HTML

<テンプレート>

```
<lightning-record-edit-form オブジェクト API 名="連絡先">
<lightning-input-field フィールド名="名"></lightning-input-field>
<lightning-input-field フィールド名="姓"></lightning-input-field>
<lightning-input-field フィールド名="メールアドレス"></lightning-input-field>
<lightning-button type="submit" name="submit" label="連絡先を作成">
</lightning-button>
</lightning-record-edit-form>
</テンプレート>
```

エラー メッセージを表示するために、開発者はどのコンポーネントをフォームに追加する必要がありますか?12

- A. ライトニングメッセージ
- B. 雷エラー
- C. apex:メッセージ
- D. オーラ:メッセージ

正解: ([正解を表示します](#))

lightning-record-edit-form は、レコードの作成と編集を自動化する強力な LWC コンポーネントです。フィールドレベルのセキュリティ、メタデータの取得、データの永続化を自動的に処理します。ただし、完全なユーザーエクスペリエンスを実現するには、フィードバックを処理するための特定のサブコンポーネントが必要です。lightning-input-field はインライン検証（必須フィールドの未入力など）を処理しますが、検証ルールの違反やサーバーから返されたシステムエラーなどの「トップレベル」のエラーメッセージを自動的に表示しません。

これらのエラーを表示するには、開発者は lightning-record-edit-form タグ内に lightning-messages オプションA)コンポーネントを含める必要があります。フォームの送信が失敗すると、lightning-messages コンポーネントは Lightning Data Service から返されたエラーペイロードを自動的に取得し、フォームの上部 またはフォームが配置されている場所)にユーザーフレンドリーな形式で表示します。

オプションBは標準のLightningベースコンポーネント名ではありません。オプションCとDはそれぞれVisualforceフレームワークとAuraフレームワークに属しており、Lightning Webコンポーネントテンプレート内では使用できません。lightning-messagesを含めることは、lightning-record-edit-formを堅牢に実装する上で必須のステップであり、レコードを保存できなかった理由をユーザーに確実に通知します。

質問: 67

開発者が取引先に対して「更新後」トリガを作成しました。ワークフロールールと項目更新により、トリガは取引先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ればよいのでしょうか？

- A. トリガーを非アクティブ化し、ロジックをプロセスまたはフローに移動します。
- B. トリガーが複数回実行されるのを防ぐには、静的変数を使用します。
- 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. トリガーが複数回実行されるのを防ぐには、グローバル変数を使用します。

正解: [\(正解を表示します\)](#)

質問: 68

開発者は、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セットアップが原因で、クラス内のすべてのテスト メソッドの実行に長い時間がかかることに気付きました。

テスト実行を高速化するために開発者は何をすべきでしょうか？

- A. クラス内のテストメソッドの数を減らす
- B. すべてのテストメソッドでテストデータファクトリーが適切に使用されていることを確認する
- C. テストデータを作成するメソッドを定義し、@testSetup でアノテーションを付ける
- D. テストデータを作成するメソッドを定義し、@createData でアノテーションを付ける

正解: C ([コメントを發表する](#))

質問: 69

ビジネスプロセスでは、新しい取引先レコードを外部システムに送信する必要があります。取引先がエラーなく挿入された場合、取引先名、ID、作成日、作成者IDがほぼリアルタイムで外部システムに渡される必要があります。

開発者はこれをどうやって実現すべきでしょうか？

- A. 挿入前トリガーとQueueableクラスを使用する
- B. @InvocableMethod メソッドを呼び出すワークフロー ルールを使用する
- C. @InvocableMethod メソッドを呼び出すプロセスビルダーを使用する

D. 挿入前トリガーと@futureメソッドを使用する

正解: ([正解を表示します](#))

質問: 70

ある会社には、Opportunity と参照関係を持つカスタムオブジェクト Sales_Help_Request__c があります。Sales_Help_Request__c には、Sales_Help_Request__c に費やされた時間を表す数値フィールド Number_of_Hours__c があります。

開発者は、Opportunity に、その Opportunity に関連する Sales_Help_Request__c レコードのすべての Number_of_Hours__c 値の合計となるフィールド Total_Hours__c を作成するように指示されています。

これを実装するには開発者は何を使うべきでしょうか？

- A. 商談オブジェクトのロールアップ集計フィールド**
- B. Sales_Help_Request__c のトリガー**
- C. Sales_Help_Request__c オブジェクトのワークフロールール**
- D. Opportunityオブジェクトのトリガー**

正解: ([正解を表示します](#))

質問: 71

開発者は、バックエンド システムに統合されるカスタム オブジェクトを設計する必要があります。

優れたデータ品質を確保し、データのインポート、統合、検索が適切に実行されるようにするには、開発者は何をする必要がありますか？ (2 つ選択してください。)

- A. カスタムフィールドをSalesforce IDとして設定する**
- B. カスタムフィールドを一意に設定する**
- C. カスタムフィールドをインデックス付きとして設定する**
- D. カスタムフィールドを外部IDとして設定する**

正解: ([正解を表示します](#))

質問: 72

組織内にApexコントローラとVisualforceページがあり、ユーザーが選択した選択リスト値の組み合わせで構成されるカスタムフィルタを使用してレコードを表示しています。一部の入力組み合わせではページが結果を表示するのに時間がかかりすぎ、他の入力選択肢では「ビューステートの最大サイズ制限を超えました」という例外が発生します。開発者はこの問題を解決するためにどのような手順を踏むべきでしょうか？

- A. ピックリスト値はインデックス化されていないため、ピックリスト値でフィルタリングするコードを調整します。**
- B. ビュー ステート エラーを回避するには、Apex コントローラから transient キーワードのインスタンスを削除します。**
- C. レイアウトを分割して、1 つの Visualforce ページでレコードをフィルターし、同じ Apex コントローラを使用して 2 番目のページにレコードのリストを表示します。**

D. Apex コントローラで StandardSetController または SOQL LIMIT を使用して、一度に表示されるレコードの数を制限します。

正解: [\(正解を表示します\)](#)

前述の2つの症状（読み込み時間の遅延と「ビューステートの最大サイズ制限を超えました」）は、どちらもUIで大容量データ（LDV）が適切に処理されていないことが原因です。ユーザーが数千件のレコードを返すフィルタを選択すると、SOQLクエリの実行に時間がかかり、結果のリストはコントローラのメモリに保存されます。Visualforceは、すべての非一時的コントローラ変数を非表示フォーム項目（ビューステート）にシリアル化するため、大きなリストはすぐに135KBの制限を超えてしまいます。

正しい解決策は、ページネーション オプションD)を実装することです。StandardSetControllerを使用すると、開発者は一度に小さなサブセット（例20件）のみをView Stateにロードすることで、大規模なデータセットを容易に管理できます。ユーザーはデータの「ページ」間を移動できます。あるいは、SOQLクエリにLIMITを追加することで、アプリケーションがプラットフォームの制限を超えるデータ処理を試行するのを防ぐことができます。123 オプションAは誤りです。選択リスト項目はSalesforceサポートによってインデックス化されるか、カスタムインデックスの一部である場合にインデックス化される可能性があります。オプションBは必要なことと逆の動作をします。一時的な要素を追加するとView Stateが縮小されます。オプションCは、根本的なデータ量の問題に対処することなく、不必要な複雑さを追加します。StandardSetControllerを使用すると、View Stateを小さくし、ページの応答性を維持しながら、大規模な結果セットを処理するための組み込みの効率的な方法が提供されます。

質問: 73

開発者がApexコールアウトを使用する必要があるシナリオはどれですか
アウトバウンドメッセージングの代わりに？

- A. ターゲット システムは REST API を使用します。
- B. コールアウトは非同期である必要があります。
- C. ターゲット システムは SOAP API を使用します。
- 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: A ([コメントを发表する](#))

質問: 74

次の標準フィールドのうち、インデックスが付けられているのはどれですか? (3 つ選択してください。)

- A. 作成者
- B. 最終更新日
- C. レコードタイプ
- D. 名前
- E. システムモッドスタンプ

正解: [\(正解を表示します\)](#)

質問: 75

展示する。

```
01 <!-- Lightning Web Component HTML file -->
02
03 <template>
04   <lightning-button label="Call server" onclick={handleClick}></lightning-button>
05 </template>

01 // Lightning Web Component JS file
02
03 import {LightningElement} from 'lwc';
04 import serverEcho from '@salesforce/apex/SimpleServerSideController.serverEcho';
05
06 export default class Helloworld extends LightningElement {
07   firstName = 'world';
08   handleClick() {
09     serverEcho({ firstName: this.firstName })
10       .then((result) => {
11         alert('From server: ' + result);
12       })
13       .catch((error) => {
14         console.error(error);
15       });
16   }
17 }

01 // Apex Controller for Lightning Component
02
03 public with sharing class SimpleServerSideController {
04   @AuraEnabled
05   public static String serverEcho(sObject firstName) {
06     String firstNameStr = (String) firstName.get('firstName');
07     return ('Hello from the server, ' + firstNameStr);
08   }
09 }
```

上記のコードでは、コードを機能させるために Apex コントローラでどのような 2 つの変更を加える必要がありますか？

2つの回答を選択してください

- A. 単一のメソッドではなく、クラス全体に @AuraEnabled としてアノテーションを付けます。
- B. Apex コントローラの行 05 の引数を subject から string に変更します。
- C. メソッドシグネチャを、String に対して public ではなく、global static に変更します。
- D. Apex コントローラから行 06 を削除し、代わりに行 07 の戻り値で firstName を使用します。

正解: ([正解を表示します](#))

提供されたコードに基づいて、必要な変更は次のとおりです。

変更Bが必要なのは、ApexコントローラのserverEchoメソッドの引数が、メソッドを呼び出すJavaScriptが期待するデータ型と一致する必要があるためです。ApexメソッドはfirstNameというオブジェクトを期待し、メソッド内で文字列に変換されます。JavaScriptはfirstNameプロパティを持つオブジェクトを渡しているため、JavaScript側での変更は不要です。

変更Dは正しいです。6行目は冗長で不要です。firstNameは既にString型なので、return文で直接使用できます。get()メソッドは通常、Mapがあり、そのキーで値を取得する必要がある場合に使用されます。firstNameは既にStringとして渡されているため、キャストしたりget()メソッドで取得したりする必要はありません。

参考文献:

Apex 開発者ガイド

質問: 76

開発者が@remoteActionデコレータを使用してグローバルサーバーサイドメソッドを呼び出そうとするとエラーが発生します。開発者はこのエラーをどのように解決すればよいのでしょうか？

- A. サーバー側メソッドを (static=false) で装飾します。
- B. 関数シグネチャを private static に変更します。
- C. サーバー側メソッドを (static=true) で装飾します。
- D. サーバー側メソッド シグネチャに static を追加します。

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: 77

Apex実行のどこでコールアウトを行うことができますか？2つの回答を選択してください

- A. @future コンテキスト内。
- B. SOQL クエリの後。
- C. DML操作後
- D. コンテキストをトリガーします。

正解: A,B ([コメントを发表する](#))

質問: 78

ある企業では、特定の地理的地域のデフォルト情報と削除動作を表す参照データを複数のカスタムメタデータレコードに保存しています。

連絡先を挿入するときに、連絡先の住所情報に基づいて、カスタム メタデータ レコードから連絡先にデフォルト情報が設定される必要があります。

さらに、フラグが付けられた地域に属する連絡先をユーザーが削除しようとする、エラー メッセージが表示される必要があります。

会社の人的リソースに応じて、これを自動化する 2 つの方法は何ですか？

2つの回答を選択してください

- A. リモートアクション
- B. フロービルダー
- C. Apexトリガー
- D. Apex呼び出し可能メソッド

正解: ([正解を表示します](#))

連絡先の住所情報に基づいてプロセスを自動化し、フラグが付けられた地域からの削除を防止するには:

フロー ビルダー: レコードの作成時にレコードのデフォルト情報を設定するために使用できません。

Apexトリガー: 挿入や削除などのデータ操作の前後に複雑なロジックを実行できます。削除を禁止するなどのルールを適用するために使用できます。

参考文献:

フロービルダー: Salesforce ヘルプ記事

Apex トリガー: Apex 開発者ガイド

質問: 79

コードの全体的なコード カバレッジは X% である必要があります。

- A. X = 100
- B. X = 75
- C. X = 65
- D. X = 50

正解: B ([コメントを發表する](#))

説明

Explanation:

すべてのトリガーのコードカバレッジは少なくとも1%である必要があります

質問: 80

開発者は、ユーザーのチェックボックスの選択状況に応じて子コンポーネントを表示する Lightning WebコンポーネントのJestテストを作成しています。各シナリオにおいて適切なコンポーネントの表示と非表示を適切にテストするには、開発者はどのようにすればよいでしょうか？

- A. 各テストの後に DOM をリセットするためのティアダウン ブロックを追加します。9
- B. テストごとに新しい記述ブロックを作成します。10
- C. テストごとに新しいj.sdomインスタンスを作成します。11

D. `afterEach()` メソッドを使用して、各テストの後に DOM をリセットします。12

正解: ([正解を表示します](#))

包括的かつ詳細な150～14250 語の説明：

Lightning Web Components (LWC) の Jest テストは、JSDOM と呼ばれる仮想ブラウザ環境で実行されます。この環境では、`document.body` は単一のテストファイル内の個々のテストケース (またはテストブロック) に渡って保持されます。

開発者が最初のテストでコンポーネントを DOM にマウントして削除しない場合、2 番目のテストの開始時にコンポーネントの状態と DOM 要素がまだ存在し、データが「漏洩」し、テスト結果が信頼できなくなります。

条件付きレンダリングのテストにおいて重要なテストの分離を確保するため、開発者は各テストの後に DOM をクリーンアップする必要があります。標準的なベストプラクティスは、`afterEach()` フック (オプション D) を使用することです。このフック内で、開発者は `document.body` からすべての子要素を削除するループを実行する必要があります。これにより、各テストが完全に「白紙の状態」で開始されます。オプション A は曖昧すぎます。「ティアダウンスブロック」は一般的な用語ですが、`afterEach()` は Jest の具体的な実装です。

オプション B はテストを整理するためのものですが、DOM のクリーンアップは処理しません。LWC テストユーティリティが JSDOM インスタンスを管理するため、オプション C は不要です。`afterEach()` を使用して DOM をリセットすることで、開発者は個々のテストケースで指定された入力に基づいて、コンポーネントが子コンポーネントを正しく表示または非表示にしていることを正確に検証できます。

質問: 81

開発者は、プロセスの最終実行前に 3 つのセーブポイント (SP1、SP2、SP3 (順番に作成)) を必要とする Apex ベースの注文作成プロセスをデバッグしています。

最終実行プロセスにおいて、開発者は特定の条件で SP1 にロールバックするルーチンを用意しています。条件が解決されると、コードは 2 つの SP3 へのロールバックを呼び出し、最終実行を続行します。しかし、SP3 へのロールバックが呼び出されると、`RuntimeError` が発生します。

開発者がランタイム エラーを受け取るのはなぜですか？

- A. SP1 がロールバックされたときに SP3 が無効になりました。
- B. 開発者が 1 つのトリガー セッションで使用したセーブポイントが多すぎます。
- C. 開発者はセーブポイント間に DML ステートメントを多すぎます。
- D. 開発者は SP3 を呼び出す前に SP2 を呼び出す必要があります。

正解: A ([コメントを發表する](#))

質問: 82

ABC 社には、複数の大規模なデータベース操作と Web サービスの呼び出しを実行する Apex プロセスがあります。データベースプロセスと Web サービスの実行には長い時間がかかるため、順番に実行する必要があります。

開発者は、ガバナ制限やシステム制限に違反することなく、この Apex コードをどのように記述すればよいでしょうか？

- A. ガバナー制限に達したら、Limits クラスを使用してプロセス全体を停止します。
- B. 各プロセスとコールアウトに複数の @future メソッドを使用します。
- C. Apex Scheduler を使用して各プロセスをスケジュールします。
- D. Queueable Apex を使用してジョブを連鎖し、順番に実行します。

正解: [A \(コメントを發表する\)](#)

質問: 83

開発者はカレンダーを表示するための 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複数の国で使用されます。ロケールによっては、週の最初の曜日が月曜日、土曜日、または日曜日になることがあります。開発者は、すべてのロケールのユーザーにカレンダーが正しく表示されるようにするために、何をすべきでしょうか？

- A. カスタム メタデータ タイプを使用して、キーと値のペアを保存します。
- B. コンポーネントで UserInfo.getLocale() を使用します。
- C. 現在のユーザーのロケールから FirstDayOfWeek フィールドを照会します。
- D. @salesforce/i18n モジュールをインポートし、firstDayOfWeek 国際化プロパティ。

正解: [\(正解を表示します\)](#)

質問: 84

企業は見積 発注プロセスの一環として、OAuth 2.0 をサポートするドキュメントストレージシステムの REST エンドポイントに POF を送信する必要があります。各 Salesforce ユーザーは、PDF を送信するにあたり、ドキュメントストレージシステムで個別に認証を受ける必要があります。開発者が REST エンドポイントへの認証を実装する最適な方法は何ですか？

- A. OAuth 認証プロバイダによる名前付き認証情報
- B. パスワード認証による名前付き認証情報
- C. OAuth トークンカスタムフィールドを使用した階層カスタム設定
- D. 2つのパスワードカスタムフィールドを持つ階層カスタム設定

正解: [\(正解を表示します\)](#)

OAuth 認証プロバイダを使用した名前付き認証情報は、開発者が REST エンドポイントへの認証を実装するための最適な方法です。名前付き認証情報は、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。OAuth 認証プロバイダを使用すると、開発者は名前付き認証情報の ID プロバイダとして OAuth 2.0 をサポートする外部サービスを使用できます。これにより、開発者はコード内で認証情報やトークンをハードコーディングする必要がなくなり、OAuth 2.0 プロトコルを活用して、ドキュメントストレージシステムで各 Salesforce ユーザーを安全に認証できます³⁴。参考資料 : コールアウトエンドポイントとしての名前付き認証情報、認証プロバイダ、OAuth 2.0、Apex Web サービスおよびコールアウト

質問: 85

テストメソッドでグローバル ピックリスト オプションをプログラムで取得するために開発者が使用すべき最適な手法は何ですか？

- A. メタデータ API への呼び出しを実行します。
- B. 静的リソースを使用する
- C. スキーマ名前空間を使用します。
- D. SOQL クエリを実行します。

正解: C ([コメントを發表する](#))

質問: 86

ページの読み込み時に rendered が false に設定された VisualForce コンポーネントを含む VisualForce ページで、開発者は再レンダリング時にそれが確実に表示されるようにするにはどうすればよいでしょうか。

- A. コンポーネントの rendered 属性を true に設定し、コンポーネントを再レンダリングします。
- B. レンダリングされた要素は更新せずに再レンダリングできないため、ページ全体の更新を実行します。
- C. コンポーネントの re-render 属性を true に設定します。
- D. コンポーネントの rendered 属性を true に設定し、親コンポーネントを再レンダリングします。

正解: D ([コメントを發表する](#))

質問: 87

許可される DML ステートメントの数に対するトランザクション制限は何ですか？

- A. 2,000
- B. 100 (同期)、200 (非同期)
- C. 150
- D. 200 (同期)、100 (非同期)
- E. 20

正解: ([正解を表示します](#))

質問: 88

```
1 連絡先 con = 新しい連絡先 ( 姓 = fSmith'、部門 = fAdminT)
2 con を挿入します。
3 連絡先 insertedContact=[id=icon.Id の連絡先から名前を選択];
4 セーブポイント sp_admin = Database.setSavepoint();
5 con.Department = fHRf;
6 更新 con;
7 データベース.ロールバック(sp_admin);
8 System.debug(Limits.getDmlStatements 0);
```

次のコードでは、行番号 8 でログにどのような値が出力されますか？

- A. 3
- B. 5
- C. 4
- D. 2

正解: ([正解を表示します](#))

質問: 89

開発者は、取引先登録に使用される Lightning Web コンポーネントからの入力に基づいて取引先を更新する Apex クラスを作成しました。取引先の更新は、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_Registered FROM Salesforce Account WHERE Id = :accountId];  
  
if (!account.Is_Registered__c) {  
    account.Is_Registered = true;  
    // ...set other account fields...  
    update account;  
}
```

複数のユーザーが同時に同じアカウントを更新した場合に、互いの更新が上書きされないようにするには、開発者は何をすべきでしょうか？

- 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 LastModifiedDate in the query to make sure it wasn't recently updated.

正解: D ([コメントを发表する](#))

複数のユーザーが同時に同じレコードを更新すると、互いの変更が上書きされてしまうリスクがあります。Salesforce では、これを防ぐために「レコードロック」と呼ばれるメカニズムを提供しています。

* 選択肢Dは正解です。SOQLクエリにFOR UPDATEを含めると、トランザクションの実行中は取得され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によるレコードの更新が防止されます。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競合状態を回避するために不可欠です。

* オプション A は不正解です。更新操作の周囲に try/catch ブロックを追加するのは例外処理には有効ですが、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は不正解です。updateではなくupsertを使用すると、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 は不正解です。レコードをロックするには、SELECT ステートメント自体ではなく、SOQL クエリで FOR UPDATE を使用する必要があるためです。

参考文献:

Salesforce 開発者向けドキュメント「ステートメントのロック」: ステートメントのロック Apex
での同時実行の処理に関する Salesforce 開発者ブログ: Apex での同時実行の処理

質問: 90

次のうち有効な REST アノテーションはどれですか？

- A. @httpCreate
- B. @httpUpsert
- C. @httpDelete
- D. @httpPatch

正解: ([正解を表示します](#))

質問: 91

Visualforce ページは、表示され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使用できますか？

- A. コントローラーのコンストラクターではなく、遅延読み込みを使用して、要求に応じてデータを読み込みます。
- B. ページ内で <apex:actionPoller> を使用して、すべてのデータを非同期的に読み込みます。
- C. カスタム コントローラー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戦略です。

コントローラーのコンストラクターではなく、オンデマンドでデータをロードすることで、初期ページのロードが高速化され、追加データは必要な場合にのみロードされます。参考資料: Visualforce 開発者ガイド - パフォーマンスの向上

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: 92

ユニバーサルコンテナーズは、コンベンション参加者 (Convention Attendee) 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。新たな品質保証活動の一環として、同社はこのオブジェクトにEvent_Reviewer_cユーザー参照フィールドを作成しました。

経営陣は、イベント レビュー担当者に割り当てられている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自動的に取得することを望んでいます。

割り当てられた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取得できるようにする最適な方法は何ですか？

- A. 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。
- B. コンベンション参加者カスタム オブジェクトに条件に基づく共有ルールを作成し、イベントレビュー担当者のグループとレコードを共有します。
- C. 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。
- D. コンベンション参加者カスタムオブジェクトに条件に基づく共有ルールを作成し、イベントレビュー担当者とレコードを共有します。

正解: **C** ([コメントを發表する](#))

質問: 93

バッチ クラスをテストする場合、開発者は何をする必要がありますか？ (2 つ選択してください。)

- A. seeAllData=true を使用する
- B. Test.startTest() と Test.stopTest() 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 未満に制限します

正解: ([正解を表示します](#))

説明/参照:

Explanation:

「seeAllData=true」を使う人は悪い人です

質問: 94

環境には、取引先の更新後トリガーと取引先責任者の更新後トリガーの 2 つの Apex トリガーがあります。

取引先の更新後トリガーは、取引先の住所が更新されるたびに起動し、その住所に関連付けられているすべての取引先責任者を更新します。取引先責任者の更新後トリガーは、編集が行われるたびに起動し、取引先責任者に関連するすべてのキャンペーンメンバーレコードを、取引先責任者の状態に合わせて更新します。

以下のケースを考えてみましょう。200件の取引先レコードの住所を一括更新します。各取引先には50人の取引先責任者がいます。各取引先責任者には1人のキャンペーンメンバーがいます。つまり、取引先全体で10,000件の取引先責任者レコードがあり、取引先責任者全体で10,000件のキャンペーンメンバーレコードがあることになります。

大量アップデートが行われると何が起こりますか？

- A. DML ステートメントによって処理されるレコード数の制限を超えなかったため、エラーは発生せず、すべての更新が成功します。
- B. アカун ト アドレスの一括更新は成功しますが、DML ステートメントによって処理されるレコード数が超過するため、連絡先アドレスの更新は失敗します。

- C. 各トリガーは独自のコンテキスト内で起動し、各トリガーは DML ステートメントによって処理されるレコード数の制限を超えないため、エラーは発生しません。
- D. 2 つのトリガーが同じコンテキストで起動し、DML ステートメントによって処理されるレコードの数を超えるため、一括更新は失敗します。

正解: ([正解を表示します](#))

このシナリオでは、更新が関連オブジェクト全体にカスケードされ、処理されるレコードの数が大幅に増加するため、DML ステートメントのガバナ制限 (トランザクションあたり 150 の DML 操作と 10,000 の DML 行) を超えてしまいます。

参考文献:

Apex 開発者ガイド

質問: 95

Salesforce 上のイベントは、実行されるプロセスの規模や種類によっては、外部システムで処理する必要がある場合があります。Salesforce のユーザーが、顧客と電話でやり取りしながら Salesforce で作成中の注文の価格を確認したいというユースケースを考えてみましょう。価格設定ロジックは既にサードパーティシステムに存在します。このロジックを Salesforce で再作成するのではなく、サードパーティシステムにリクエストを送信することで活用します。レスポンス (この場合は価格) は Salesforce に返され、保存されます。最適なソリューションは何でしょうか？

- A. リアルタイムの Apex コールアウトを実行して価格を表示し、Salesforce に保存できる Visualforce ページ
- B. 注文を保存するとリアルタイムの Apex 呼び出しを実行し、価格を Salesforce に保存する Apex トリガー
- C. 保存時に価格を取得し、Salesforce に価格を保存するためのプロセスビルダープロセスとアウトバウンドメッセージ
- D. 数分ごとに新しく保存された注文のバッチを処理し、価格を Salesforce に保存する ETL ツール
- 正解: ([正解を表示します](#))

質問: 96

顧客は、アカウントの請求先住所フィールドが更新されると、関連するすべての連絡先レコードの住所フィールドに同じ更新が反映されることを要求しています。

最小限のカスタマイズでこの要件を満たすにはどうすればよいでしょうか？

- A. 関連するアカウント レコードに基づいてすべての連絡先アドレス フィールドを更新するスケジュールされたバッチ ジョブを作成します。
- B. アカウントに After トリガーを作成して、更新時に関連する連絡先レコードを更新します。
- C. 取引先に Lightning プロセスを作成して、関連する子の取引先責任者レコードを更新します。
- D. 関連する子連絡先レコードを更新するためのアカウントの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: 97

Visualforce ページのケースにリンクを追加するための最適な構文は何ですか? (2 つ選択してください。)

<apex:outputLink 値="{!\$URLFOR(\$Action.Case.Open, case)}" ターゲット="空白">

A. {!case.Name} </apex:outputLink>

<apex:outputLink value="{!case.Id}" target="blank"> {!case.Name}

B. apex:outputLink>

<apex:outputLink 値="{!\$URLFOR(\$Action.Case.View,case.Id)}" ターゲット="空白">

C. (!case.Name) </apex:outputLink>

<apex:outputLink value="{!viewCase(case.Id)}" target="_blank">{!case.Name} </

D. apex:outputLink>

正解: ([正解を表示します](#))

説明/参照:

質問: 98

ポイントツーポイント統合の一環として、開発者は外部Webサービスを呼び出す必要がありますが、需要が高いため、応答に時間がかかります。開発者は、呼び出しを行う前に、リクエストの一部としてエンドユーザーから主要な入力情報を収集する必要があります。

これらのビジネス要件を実装するために開発者が使用する必要がある 2 つの要素はどれですか?

2つの回答を選択してください

A. プロセスビルダー

B. Lightning Web コンポーネント

C. 今すぐスクリーン

D. Continuationオブジェクトを返すApexメソッド

正解: ([正解を表示します](#))

質問: 99

開発者は、組織内のすべてのカスタム オブジェクトのドロップダウン リストを表示する Lightning Web コンポーネントを作成します。

Apexメソッドはデータを準備し、コンポーネントに返します。開発者はレスポンスに含めるオブジェクトを決定するために何をすべきでしょうか?

A. @salesforce/schema からすべてのカスタム オブジェクトのリストをインポートします。

B. sObject の記述結果で、getObjectType() の値が 'Custom' か 'Standard' かを確認します。

C. sObject の記述結果の isCustom() 値を確認します。

D. Schema クラスの getCustomObjects() メソッドを使用します。

正解: C ([コメントを公表する](#))

包括的かつ詳細な説明 :

Apex でカスタムオブジェクトをプログラマ的に識別するために、開発者は Schema Describe 情報を利用します。Schema.getGlobalDescribe() メソッドは、組織内のすべてのオブジェクトトークンのマップを返します。これらのトークンを反復処理し、それぞれに対して getDescribe() メ

ソッドを呼び出すことで、開発者はオブジェクトのメタデータのさまざまなプロパティにアクセスできます。

標準オブジェクトとカスタムオブジェクトを区別するための特定のプロパティは、isCustom() メソッド (オプション C) です。このブール型メソッドは、オブジェクトがカスタムオブジェクト (カスタム設定とカスタムメタデータ型を含む) の場合は true を返し、標準プラットフォームオブジェクトの場合は false を返します。

選択肢Aは不正解です。LWCでは@salesforce/schemaは特定の既知のオブジェクト/項目への参照をインポートするために使用され、組織全体のメタデータを動的に検出するためには使用されません。選択肢Bは不正解です。getObjectType()は「Custom」のような文字列値ではな

く、SObjectTypeトークンを返すためです。選択肢Dは不正解です。標準スキーマクラスにはgetCustomObjects()メソッドがないため、検出はグローバル記述マップを介して行う必要があります。

質問: 100

開発者はビジネス ロジック コードを実行できますが、テスト クラスで次のエラーが発生します。コミットされていない作業が保留中です。コールアウトする前にコミットまたはロールバックしてください。どのような解決策がありますか？

- A. 実際にはコードは失敗しないので、テスト クラスの先頭で seeAllData を "true" に設定します。
- B. コールアウトに @TestVisible を設定してビジネス ロジックとテスト クラスを書き換えます。
- C. テスト実行時にそれをバイパスするには、コールアウトを行う前に test.IsRunningTest() を使用します。
- D. 外部コード エラーの可能性が高いため、ターゲット エンドポイントに関するヘルプについてはサポートにお問い合わせください。

正解: ([正解を表示します](#))

質問: 101

上記のテスト メソッドは、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ことを開発者が認識している 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。

これを修正する正しい方法は何ですか？

```
Line 17:
Line 18:   update DataFactory.accounts;
Line 19:
Line 20:   List<Account> acctsAfter = [SELECT Number_Of_Transfers__c FROM Account WHERE Id IN
Line 21:                               :DataFactory.accounts];
Line 22:   //Utility to assert Number_Of_Transfers__c is correct based on test data
Line 23:   AssertUtil.assertNumberOfTransfers(acctsAfter);
Line 24: }
```

上記のテストメソッドは、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ことを開発者が知っている 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。

これを修正する正しい方法は何ですか？

- A. コードの 7 行目と 20 行目の前に Test.startTest() を追加し、その後に Test.stopTest() を追加します。
- B. Apex トリガーの大部分をプロセスビルダープロセスに置き換えて、トリガー内のクエリの数減らします。
- C. コードの 18 行目前に Test.startTest() を追加し、コードの 18 行目後に Test.stopTest() を追加します。
- D. DataFactory クラスを変更して、作成するアカウントの数を減らし、トリガー内のクエリの数減らします。

正解: ([正解を表示します](#))

質問: 102

ある企業では、注文用のネイティブiOSアプリをSalesforceに接続し、様々なオブジェクトからJSON形式で統合された情報を取得する必要があります。Salesforceでこれを実装する最適な方法はどれでしょうか？

- A. Apex SOAP Web サービス
- B. Apex REST Web サービス
- C. Apex SOAP コールアウト
- D. Apex REST コールアウト

正解: B ([コメントを发表する](#))

質問: 103

Salesforce 組織でプラットフォームイベントが定義された後、どの 2 つのメカニズムを介してイベントを公開できますか？

2つの回答を選択してください

- A. 外部アプリは標準のストリーミング API を使用できます。
- B. 内部アプリはプロセス ビルダーを使用できます。
- C. 内部アプリは送信メッセージを使用できる
- D. 外部アプリにはカスタム Apex Web サービスが必要です。

正解: ([正解を表示します](#))

質問: 104

開発者は、商談トリガから呼び出される複雑なコミッション計算エンジンをApexで構築しています。QA中に、計算結果に誤りがあると報告されました。開発者は、開発者サンドボックスに代表的なテストデータと合格したテストメソッドを保有しています。コードを実行し、重要な行で一時停止して様々なApex変数の値を視覚的に確認するために、開発者が使用できるツールまたはテクニックを3つ挙げてください。

- A. Apex インタラクティブデバッガー
- B. Apex リプレイデバッガー
- C. ワークベンチ
- D. 開発者コンソール

E. ブレークポイント

正解: [\(正解を表示します\)](#)

Salesforce で従来の「ステップスルー」デバッグを実行するには(実行を一時停止して、リアルタイムまたはほぼリアルタイムで変数の状態を検査できます)、開発者は VS Code 用の Salesforce 拡張機能を使用します。

Apex Replay Debugger (オプションB)は、開発者がデバッグログを使用してトランザクションを「再生」できる無料ツールです。コードにブレークポイント (オプションE)を設定し、ロジックを実行 (またはテストを実行)することで、開発者はログファイルを生成できます。Replay Debuggerはこのログを解析し、開発者はVS Codeでコードを1行ずつ、まるでライブで実行されているかのようにステップ実行できます1.2。Apex Interactive Debugger (オプションA)は、サンドボックスまたはスクラッチ組織でリアルタイムのライブデバッグを可能にする専用ツール (通常サンドボックス用の有料アドオンが必要)です。Replay Debuggerとは異なり、Salesforceサーバ上での実際の実行を一時停止するため、開発者は環境のライブビューを確認できます。

開発者コンソール (オプションD)とワークベンチ (オプションC)はログの表示とクエリ機能を提供しますが、従来のIDEデバッガのように特定の行で実行を「一時停止」して視覚的に確認する機能はサポートしていません。そのため、このデバッグパターンでは、デバッグツールとブレークポイントの組み合わせが標準的な要件となります。

質問: 105

Universal Containersは、Salesforceにおいてソース駆動開発アプローチを採用する開発チームを率いています。継続的インテグレーションおよびデリバリー (CI/CD) プロセスの一環として、サンドボックス環境や本番環境を含む複数の環境への変更の自動デプロイが必要です。ソース駆動開発におけるCI/CDパイプラインを最も効果的にサポートするメカニズムまたはツールはどれでしょうか？

- A. Salesforce DX を使用した Salesforce CLI
- B. 変更セット
- C. Visual Studio Code 向け Salesforce 拡張機能
- D. Ant 移行ツール

正解: [\(正解を表示します\)](#)

ソース駆動開発では、「信頼できる情報源」をSalesforce組織からバージョン管理システム (Gitなど)に移行します。Salesforce DX (Developer Experience)は、Salesforce CLI (コマンドラインインターフェース)を導入することで、このパラダイムをサポートするように特別に設計されました。Salesforce CLIは、シンプルなコマンドライン命令を使用して、スクリプトによるスクラッチ組織の作成、自動テスト、そしてあらゆる環境 (サンドボックス、スクラッチ組織、本番環境)へのソースコードのデプロイを可能にするため、最新のCI/CDパイプラインの主要エンジンとなっています3。

従来のツールとは異なり、Salesforce CLI は「ソース」形式をサポートしています。これは、Ant 移行ツールで使用される従来のメタデータ形式よりも詳細で、バージョン管理が容易です。変更セット (オプション B) は、CI/CD パイプラインで自動化できず、組織間でのみ使用される手動の

UI 駆動型ツールです。VS Code 向け Salesforce 拡張機能 (オプション C) は、開発者に強力な IDE インターフェースを提供しますが、拡張機能自体は、基盤となる Salesforce CLI を使用してデプロイメントを実行します。パイプライン自体で使われる自動化ツールではありません。Ant 移行ツール (オプション D) は、メタデータ API を使用する古いテクノロジーであり、より柔軟で強力な Salesforce DX コマンドと比較すると、一般的にレガシーと見なされています。したがって、堅牢で自動化されたソース駆動型のデプロイメント戦略には、Salesforce CLI と Salesforce DX の組み合わせが最適です。

質問: 106

```
共有クラス MyRemoter を持つグローバル {  
パブリック文字列アカウント名 { 取得; 設定; }  
パブリック静的アカウントアカウント { 取得; 設定; }  
パブリックアカウントリモート() {}  
@リモートアクション  
グローバル静的アカウント getAccount(String accountName) {  
アカウント = [SELECT ID, 名前, 従業員数  
FROM アカウント WHERE 名前 = :アカウント名];  
アカウントを返す;  
}  
}
```

Visualforce 検索ページで使われる RemoteAction を定義する上記の Apex クラスを検討してください。

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アサートするコード スニペットはどれですか。

```
アカウント a = controller.getAccount('TestAccount');  
A. System.assertEquals( 'TestAccount', a.Name );  
B. System.assertEquals( 'TestAccount', a.Name );  
MyRemoter リモート = 新しい MyRemoter();  
C. アカウント a = remote.getAccount('TestAccount');  
System.assertEquals( 'TestAccount', a.Name );  
MyRemoter リモート = 新しい MyRemoter('TestAccount');  
D. アカウント a = remote.getAccount ();  
System.assertEquals( 'TestAccount', a.Name );  
アカウント a = MyRemoter.getAccount('TestAccount');  
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: 107

Visualforce ページには業種選択リストが含まれており、業種フィールドに一致する値を持つ取引先のテーブルが表示されます。

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

ユーザーが業界選択リストの値を変更すると、アカウント テーブルが自動的に更新され、選択した業界に関連付けられているアカウントが表示されます。

これを実装する最適な方法は何ですか？

- A. <apex: selectOptions> 内に <apex: actionSupport> を追加します。
- B. <apex: selectOptions> 内に <apex: actionFunction> を追加します。
- C. <apex: select List > 内に <apex: actionFunction> を追加します。
- D. <apex:selectList> 内に <apex: actionSupport> を追加します。

正解: ([正解を表示します](#))

質問: 108

開発者には、Case の標準コントローラをオーバーライドする 2 つの拡張機能を含むページがあります。

<apex:page standardController="Case" extension="CaseExtensionOne,CaseExtension Two" showHeader="false"> 各拡張機能には Save」というメソッドがあります。ページには次のように定義されたコマンドボタンがあります: <apex:commandButton value="Save" action="{!save}"/> ユーザーがこのコマンドボタンをクリックすると何が起こるでしょうか？

- A. CaseExtensionOneからの保存が実行されます
- B. Case標準コントローラからの保存が実行されます
- C. 3つのSaveメソッドすべてが実行されます
- D. CaseExtensionTwoからの保存が実行されます

正解: ([正解を表示します](#))

質問: 109

ユニバーサル・チャリティーズ (UC) は、Salesforceを使用して、個人および法人からクレジットカードによる電子寄付を徴収しています。カスタマーサービス担当者がクレジットカード情報を入力すると、寄付を処理するために、その情報はサードパーティの決済代行業者に送信する必要があります。UCは、個人向けと法人向けにそれぞれ異なる決済代行業者を使用しています。システム管理者が導入後に必要に応じて設定を変更できるよう、開発者は各決済代行業者の設定をどのような方法で保存すればよいのでしょうか？

- A. 階層のカスタム設定
- B. カスタムラベル
- C. カスタムメタデータ
- D. カスタム設定をリストする

正解: ([正解を表示します](#))

管理者が容易に変更し、複数の環境に展開する必要があるアプリケーション構成と統合設定を保存するには、カスタムメタデータタイプが最適なソリューションです。カスタム設定 (オプション A および D) とは異なり、カスタムメタデータタイプ内のレコードはデータではなくメタデータとして扱われます。これは開発ライフサイクルにおいて重要な違いです。なぜなら、これらのレコードは変更セットまたは展開パッケージに含めることができるからです。これにより、サンドボックス展開後に本番環境で構成レコードを再作成する際の手作業によるオーバーヘッドとリスクが排除されます。

このシナリオでは、UCは2つの異なる決済処理業者の設定を管理する必要があります。APIエンドポイント、加盟店ID、セキュリティキーなどのフィールドを含むカスタムメタデータタイプを作成できます。管理者は設定メニューから「個人」および「法人」決済処理業者の特定のレコードを直接作成・編集できます。

さらに、カスタムメタデータクエリは効率的であり、多くの状況において標準のSOQLガバナ制限にカウントされません。カスタムラベル (オプション B) はテキストの翻訳には便利ですが、複雑で構造化された設定データには適していません。階層カスタム設定はユーザー固有の上書きを目的として設計されているため、ここでは適用できません。したがって、カスタムメタデータは、外部サービス設定を管理するための最も堅牢で導入しやすく、管理者にとって使いやすい方法を提供します。

質問: 110

Visualforce ページの「ポストバック」リクエスト (ユーザー操作でページの更新が必要な場合) の正しい実行順序は何ですか？

- A. 1) ビューステートをデコードする
- 2) メインページとカスタムコンポーネントの式とメソッド呼び出しを評価する
- 3) 正常に完了したら、データを更新し、ユーザーをリダイレクト/ビューステートを更新します。
- 4) 正常に完了したら、ポストバックをトリガーしたアクションを評価します。
- 5) ブラウザに送信されるHTML
- B. 1) メインページとカスタムコンポーネントの式とメソッド呼び出しを評価する
- 2) ビューステートをデコードする
- 3) 正常に完了したら、データを更新し、ユーザーをリダイレクト/ビューステートを更新します。
- 4) 正常に完了したら、ポストバックをトリガーしたアクションを評価します。
- 5) ブラウザに送信されるHTML
- C. 1) ビューステートをデコードする
- 2) メインページとカスタムコンポーネントの式とメソッド呼び出しを評価する
- 3) 正常に完了したら、ポストバックをトリガーしたアクションを評価します。

- 4) 正常に完了したら、データを更新し、ユーザーをリダイレクト/ビューステートを更新します。
5) ブラウザに送信されるHTML
正解: **C** ([コメントを發表する](#))

質問: 111

開発者は、カスタム オブジェクトにあるすべてのチェックボックス フィールドのリストを表示する Visualforce ページを作成しています。

これを実現するために開発者が使用すべき推奨メカニズムは何ですか？

- A. スキーマビルダー
- B. メタデータAPI
- C. Apex API
- D. スキーマクラス

正解: ([正解を表示します](#))

質問: 112

バッチ クラスをテストする場合、開発者は何をする必要がありますか？ (2 つ選択してください。)

- A. seeAllData=true を使用します
- 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 未満に制限します

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: 113

Universal Containersは、Convention_Attendance_cカスタムオブジェクトに非公開共有モデルを実装しています。新たな品質保証活動の一環として、同社はこのオブジェクトに

Event__Reviewer__cユーザー参照項目を作成しました。経営陣は、イベントレビュー担当者に、担当するすべてのレコードへの読み取り/書き込みアクセス権を自動的に付与したいと考えています。

割り当てられた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取得できるようにする最適な方法は何ですか？

- A. 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。
- B. コンベンション参加者カスタム オブジェクトに条件に基づく共有ルールを作成し、イベントレビュー担当者とレコードを共有します。
- C. 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。
- D. コンベンション参加者カスタム オブジェクトに条件に基づく共有ルールを作成し、イベントレビュー担当者のグループとレコードを共有します。

正解: ([正解を表示します](#))

質問: 114

<apex:actionPoller> 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>タグでカプセル化する必要があります。actionSupportとは異なり、これらのfunction<apex:actionPoller>はJavaScriptコードから直接呼び出すことができます。
- C. 別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 (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。名前を指定するには「id」フィールドを使用し、関連コンポーネントを接続するには「status」フィールドを使用します。
- E. AJAXリクエストが生成されたときにサーバーが処理するコンポーネントを示します。

正解: ([正解を表示します](#))

質問: 115

コールアウト回数の取引制限はいくらですか？

- A. 制限はありません
- B. 150
- C. 100
- D. 200
- E. 50

正解: ([正解を表示します](#))

質問: 116

次のコード スニペットを検討してください。

```

import { LightningElement } from 'lwc';
import getOrders from '@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;
    }
  }
}

```



コンポーネントがデプロイされると、エラーが報告されます。

コンポーネントが正常にデプロイされるようにするには、開発者がコードに実装する必要がある2つの変更はどれですか。

2つの回答を選択してください

A)

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

B)

```
import getOrders from '@salesforce/apex/OrderController.getAvailableOrders';
```

C)

```
import getOrders from '@salesforce/apex/c.OrderController.getAvailableOrders';
```

D)

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

A. オプションB

B. オプションC

C. オプションD

D. オプションA

正解: A,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**」