

Salesforce Certified Platform Developer II (Japanese Version)

1. 開発者は、Lightning データテーブル内のレコードのリストを表示する Lightning Web コンポーネントを作成しています。新しいレコードをデータベースに保存した後、リストが更新されません。

A. `this.dazta` で `rafrasnApex()` を呼び出します。

B. 結果を保存する新しい変数を作成し、`@track` アノテーションを付けます。

C. 結果を格納する変数を作成し、`refreshpex()` を呼び出します。

D. `@track` デコレータをデータ変数に追加します。

Answer(s): C

2. 開発者は、Lightning Web コンポーネントのスタイルと動作を制御する変数を保存する必要があります。

A. カスタム設定

B. カスタムオブジェクト

C. カスタム変数

D. カスタムメタデータ

Answer(s): D

3. ある企業には、Salesforce に接続してさまざまなオブジェクトから統合情報を JSON 形式で取得する必要があるネイティブ iOS 注文アプリケーションがあります。

A. Apex REST Web サービス

B. Apex SOAP Web サービス

C. Apex SOAP コールアウト

D. Apex REST コールアウト

Answer(s): A

4. 開発者は、コンポーネントがレコードページ上でデスクトップサイズのフォームファクタに対してのみ表示されるように、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。

A. A.

```
<design>  
  <designAttribute name="formfactor" value="Large"/>  
</design>
```

B. B.

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

C. C.

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

D. D.

```
<property name="formfactor" value="Large">  
</property>
```

Answer(s): B

5. 開発者は、アカウントでトリガーを正常にアサートするテストクラスを作成しました。これは、サンドボックス環境で正しく起動され、データが更新されます。

A. 実稼働環境を有効にするように構成します。「すべてのテストを管理者ユーザーとして実行します。」

B. System.runAs() をテスト クラスに追加して、正しいオブジェクト権限を持つユーザーとしてトリガーを実行します。

C. 運用環境の共有モデル内で動作するように、テスト クラスに seeallData=true を追加します。

D. テストを検証します。startTest() がテスト クラスの For ループ内にない、

Answer(s): B

6. 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。

A. A.

```
<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>
```

B. B.

```
<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>
```

C. C.

```
<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>
```

D. D.

```
<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>
```

Answer(s): C

7. ある企業は、前提条件となる参照データの多数のレコードを作成するための多くのメソッドを含むテストクラスでの単体テストが遅いことに気がきました。

A. テストの実行時にトリガー、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。

Answer(s): C

8. カスタムインターフェースの一部として、開発者チームはさまざまな新しい Lightning Web コンポーネントを作成します。各コンポーネントはトースト メッセージを使用してエラーを処理します。開発が完了すると、すべてのコンポーネントが同じ Lightning ページに追加されます。

A. Lightning Web コンポーネントを使用して、すべてのエラーを集計して表示します

B. `window.alert()` メソッドを使用してエラー メッセージを表示します

C. `<template>` タグを使用して、インプレース エラー メッセージを表示します。

D. 各コンポーネントの `public` プロパティを使用して、エラー メッセージを表示します。

Answer(s): A,D

9. 開発者は、非同期プロセスのトリガーが正常に実行されたことをどのように確認する必要がありますか？

A. テストクラス内にすべてのテストデータを作成し、システムを使用します。{} を実行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` を使用してから、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、残りのクラスで `future` を使用してから、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。

Answer(s): D

10. 複数通貨が有効になっている組織では、開発者は、ログインしているユーザーが最近アクセスした上位 10 件の商談を表示する Lightning コンポーネントを構築するという任務を負っています。開発者は、Amount フィールドと LastModifiedDate フィールドの値がユーザーのロケールに従って表示されていることを確認する必要があります。

A. REGEX 式を使用して、SOQL 経由で取得した値を書式設定します。

B. ラッパークラスを使用して、SOQL 経由で取得した値をフォーマットします。

C. SOQL クエリで FOR VIEW 句を使用します。

D. SOQL クエリで FORMAT () 関数を使用します。

Answer(s): D

11. Universal Containers (LIC) は、顧客がコンテナに関する問題を記録できるようにする顧客コミュニティを開発したいと考えています。コミュニティは、ドイツ語とスペイン語を話す顧客のためにも機能する必要があります。UC は、Salesforce を使用すると国際的なコミュニティを簡単に作成できると聞き、サイトを構築する開発者を雇いました。

A. カスタム ラベルを使用して、カスタム メッセージが適切に翻訳されていることを確認します。

B. カスタム設定を使用して、カスタム メッセージが適切に翻訳されるようにします。

C. カスタム オブジェクトを使用してカスタム選択リスト値を変換します。

D. カスタム メタデータを使用してカスタム選択リスト値を変換します。

Answer(s): A

12. 企業には、データ設定の一部として取引先レコードを作成するさまざまな単体テスト方法があります。新しい必須フィールドがアカウントに追加され、すべての単体テストが失敗するようになりました。

A. すべての単体テストのデータ設定に必須フィールドを追加します。

B. アカウントに挿入前トリガーを追加して、必須フィールドの値を設定します。

C. 必須フィールドを、システム管理者プロファイルを除外する検証ルールに変更します。

D. 単体テスト用のアカウントを作成する単一の場所として機能する TestDataFactory クラスを作成し、そこに必要なフィールドを設定します。

Answer(s): D

13. Universal Containers は、Convention Attendee co カスタム オブジェクトのプライベート共有モデルを実装します。新しい品質保証の取り組みの一環として、同社はオブジェクトに Event_Reviewer_c ユーザー検索フィールドを作成しました。

A. 大会出席者カスタムオブジェクトに挿入前トリガーを作成し、Apex 共有理由と Apex 管理共有を使用します。

B. 大会出席者カスタムオブジェクトに挿入後トリガーを作成し、Apex 共有理由と Apex 管理共有を使用します。

C. 大会出席者カスタム オブジェクトに基準ベースの共有ルールを作成して、イベント審査担当者とレコードを共有します。

D. 大会出席者カスタム オブジェクトに基準ベースの共有ルールを作成して、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。

Answer(s): B

14. 以下のテスト方法を参照してください。

A. A.

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

B. B.

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

C. C.

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

D. D.

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

Answer(s): D

15. 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。

A. A.

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

```
RecordType recType_Renewal = [SELECT Id, Name FROM  
RecordType WHERE SubjectType = '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. B.

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

C. C.

```
Id recType_New =
Schema.SObjectType.Opportunity.getRecordTypeInfoByDe

Id recType_Renewal =
Schema.SObjectType.Opportunity.getRecordTypeInfoByDe

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
    }
}
```

D. D.

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

Answer(s): C

16. Universal Containers は、Customer Community Plus ライセンスを持つ Customer Community を使用して、顧客がレンタルしたコンテナの数と返却期限を追跡できるようにしたいと考えています。Universal Containers は、外部ユーザー向けにプライベート共有モデルを使用しています。顧客の多くは、複雑なアカウント階層を持つ多国籍企業です。階層上の各取引先は、同じビジネス内の部門を表します。要件の 1 つは、取引先担当者をさまざまな取引先レコードに関連付けるカスタム ジャンクション オブジェクトに基づいて、同じ取引先階層内の特定のコミュニティ ユーザーが複数の部門のコンテナを表示できるようにすることです。部門を代表するものです。

A. 共有せずに指定するカスタムコントローラを使用する Visualforce ページ [0 レコードを公開する]

B. ジャンクションオブジェクトの関係に基づいて Apex 管理共有レコードを作成する Apex トリガ

C. Lightning Data Services を使用するコミュニティホームページ上の Lightning Web コンポーネント。

D. 所有者に基づいて適切なレコードを表示するフィルターを備えたジャンクション オブジェクトのカスタム リスト ビュー

Answer(s): B

17. 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

A. ターゲット システムは REST API を使用します。

B. ターゲット システムは SOAP 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。

Answer(s): A

18. Universal Containers は、取引先に関連付けられた取引先責任者のリストを取得する 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。コードの抜粋は次のとおりです。

A. Apex コントローラクラスに新しいメソッドを追加して、メールアドレスを個別に取得し、このメソッドを呼び出すように Lightning Web コンポーネントを変更します。

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 項目の列を含めます。

C. getAccountContacts メソッドの SOQL クエリを変更して、MailingAddress 項目を含めます。

D. getAccountContacts メソッドの SOQL クエリを変更して MailingAddress 項目を含め、JavaScript ファイルの columns 属性を更新して Mailing address 項目を追加します。

Answer(s): D

19. 一括安全な Apex トリガを作成するためのベストプラクティスと考えられるステートメントはどれですか？

A. 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。

B. すべての SOQL ステートメントに LIMIT 50000 を追加します。

C. すべての DML 操作を future メソッド内から実行します。

D. DML ステートメントの代わりに、allorNone を false に設定してデータベース メソッドを使用します。

Answer(s): A

20. 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。Opportunity のステージが変更されると、フィールドの値を 1 つ増やすワークフロー ルールが起動されます。開発者は、フィールドが 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。

A. Trigger.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. ワークフロー ルールの前にトリガーが起動された後。

C. Trigger.new はフィールド更新後に変更されません。

D. フィールド更新後に After トリガーは起動されません。

Answer(s): A
