

Salesforce.PDI.v2026-06-30.q135

試験コード :	PDI
試験名称 :	Platform Developer I (PDI)
認証ベンダー :	Salesforce
無料問題の数 :	135
バージョン :	v2026-06-30
ページの閲覧量 :	102
問題集の閲覧量 :	1358

<https://www.jpnsiken.com/shiken/Salesforce.PDI.v2026-06-30.q135.html>

質問: 1

以下のコードスニペットは、2,000件以上のリードレコードが存在する環境で、Lightning Webコンポーネントによって実行されます。

```
@AuraEnabled
public void static updateLeads() {
    for(Lead thisLead : [SELECT Origin__c FROM Lead]){
        thisLead.LeadSource = thisLead.Origin__c;
        update thisLead;
    }
}
```

Apexトランザクションにおいて、どのガバナー制限を超える可能性が高いでしょうか？

- A. 発行されたSOQL照会の総数
- B. 発行されたDML明細書の総数
- C. DMLステートメントの結果として処理されたレコードの総数

正解: C (コメントを發表する)

提供されたコードは、2,000 を超えるレコードが存在する環境で allLeadrecords を更新しようとしています。これは、DML ステートメントで処理されるレコードの総数に関するガバナー制限を超える可能性が高く、

1回のトランザクションにつき10,000。これを回避するには、開発者はDatabase.updateをバッチ処理ロジックとともに使用できます。

参考資料 :DML行のガバナー制限

質問: 2

開発者は、Expense__c オブジェクト上で以下のトリガーを特定しました。

```
deleteExpense,
applyDefaultsToExpense,
validateExpenseUpdate;
```

トリガーは、それぞれ削除イベント、挿入イベント、更新イベントの前に処理を実行します。

トリガーのベストプラクティスを確実に遵守するために、開発者はどの2つの手法を導入すべきでしょうか？

2つの回答を選択してください

- A. Unity は、すべてのイベントを含む Expense__c オブジェクト上の単一のトリガーで、3 つのすべてのトリガーを実装します。
- B. before insert と before update のトリガーを統合し、削除アクションには Flow を使用します。
- C. レコードが保存されたときに適切なロジックを実行するヘルパークラスを作成します。
- D. Expense __c オブジェクト上の 3 つのトリガーはすべて維持しますが、Apex ロジックをトリガ一定義から移動します。

正解: ([正解を表示します](#))

A: 3つのトリガーすべてを、すべてのイベントを含むExpense__cオブジェクトの単一のトリガーに統合します。

Salesforceのベストプラクティスでは、冗長性や競合を避けるため、オブジェクトごとにトリガーを1つだけ設定することを推奨しています。

削除前、挿入前、更新前のすべてのイベントを単一のトリガーにまとめることで、開発者はロジックを整理された保守しやすい方法で管理できます。

これによりデバッグが簡素化され、トリガーロジックが予測可能な順序で実行されることが保証されます。

C :レコードが保存されたときに適切なロジックを実行するためのヘルパークラスを作成します。

ヘルパークラスを使用することで、関心の分離を明確に行うことができます。トリガーはディスパッチャーとして機能し、ロジックを専用クラスに委譲します。

例えば、applyDefaultsToExpense()、validateExpenseUpdate()、deleteExpense() などのメソッドをヘルパークラスに作成し、トリガーから呼び出すことができます。

これにより、コードの再利用性、可読性、テスト容易性が向上します。

他の選択肢はどうでしょうか？

B: 挿入前トリガーと更新前トリガーを統合し、削除アクションにFlowを使用する。

Flowは強力なツールですが、レコードの削除やApexトリガー機能の置き換えには最適ではありません。特に、他のイベントに対して既にトリガーが存在する場合はなおさらです。

D: Expense__c オブジェクト上の 3 つのトリガーはすべて維持しますが、Apex ロジックをトリガ一定義の外に移動します。

同じオブジェクトに対して複数のトリガーを維持すると、ロジックをヘルパークラスに移動した場合でも、競合や実行順序の問題が発生する可能性があります。

参考文献：

トリガーのベストプラクティス

トリガー設計パターン

質問: 3

Apexテストクラスで@testSetupアノテーションを使用することに関して、正しい記述は次のうちどれですか？ 2つ選択してください。）

- A. @testSetup メソッドで作成されたレコードは、個々のテストメソッドで更新できません。
- B. @isTest(SeeAllData=True) アノテーションが使用されている場合、@testSetup アノテーションは使用できません。
- C. クラス内のすべてのテストメソッドに対して、テストデータは一度だけ挿入されます。
- D. @testSetup メソッドは、テストクラス内の各テストメソッドが実行される前に自動的に実行されます。

正解: ([正解を表示します](#))

質問: 4

開発者は、Apexクラス内のメソッドをインポートするLightning Webコンポーネントを作成します。検証ボタンが押されると、そのメソッドが実行され、複雑な検証が実行されます。

この実装シナリオでは、MVCアーキテクチャに従ってコントローラーのどの2つのオプションが適切ですか？

2つの回答を選択してください

- A. HTMLファイル
- B. アペックスクラス
- C. JavaScriptファイル
- D. XMLファイル

正解: ([正解を表示します](#))

Lightning Web Component (LWC) の MVC (Model-View-Controller) アーキテクチャにおいては、Apex クラスと JavaScript ファイルがコントローラの役割を処理する責任を負います。

- * Apexクラス (コントローラ) :
- * サーバー側のコントローラとして機能します。
- * シナリオで言及されている複雑な検証など、ビジネスロジックを実行します。
- * @AuraEnabled を使用して LWC が呼び出すメソッドを公開します。

参照: <https://developer.salesforce.com/docs/component-library/documentation/en/lwc/lwc.apex> JavaScript ファイル (コントローラ):

クライアント側のコントローラとして機能します。

HTML (ビュー) と Apex メソッド間の相互作用を管理します。

「検証」ボタンをクリックするなど、ユーザーの操作によってトリガーされる機能を実行します。

参照: https://developer.salesforce.com/docs/component-library/documentation/en/lwc/lwc.js_props_methods 他のオプションを選ばない理由 A. HTML

ファイル: MVC の「ビュー」を表し、UI レンダリングに焦点を当てます。

D: XMLファイル: メタデータ構成に使用され、コントローラ機能の一部ではありません。

質問: 5

アプリケーション開発のメリットは何ですか？

- A. ユニットテストとコードカバレッジのベストプラクティスを徹底する
- B. 定義済みのコンピューティングリソースへのアクセス
- C. ビッグデータ向けに事前設定されたストレージ
- D. 無制限の処理能力とメモリ

正解: B ([コメントを发表する](#))

Salesforceのマルチテナントアーキテクチャは、Salesforceによって事前に定義および管理される共有コンピューティングリソースを提供します。これにより、拡張性、セキュリティ、信頼性が確保されるとともに、顧客がインフラストラクチャを管理する必要がなくなります。

参照 :

Salesforceマルチテナントアーキテクチャ

質問: 6

Universal Containers は、新しい Salesforce アプリケーションの構築に宣言型開発のみを使用することにしました。アプリケーションのデータベース層を構築するために使用すべき3つのオプションはどれですか？3つ選択してください。

- A. トリガー
- B. フロー
- C. ロールアップサマリー
- D. 人間関係
- E. カスタムオブジェクトとフィールド

正解: [\(正解を表示します\)](#)

質問: 7

クイックアクションを使用してレコードを作成する場合、フィードアイテムを投稿する最も簡単な方法は何ですか？

- A. クイックアクションでフィードアイテムの作成を選択します。
- B. 新しいレコードにワークフロールールを追加することで。
- C. 新しいレコードで「ケースフィードの作成」を選択します。
- D. 新しいレコードにトリガーを追加することで。

正解: [\(正解を表示します\)](#)

質問: 8

開発者が、アカウントのステータスが変更されたときに、アカウントに関連付けられたアセットの status__c を更新するトリガーをデプロイしたところ、アカウントを一括更新する夜間の統合処理が制限エラーで失敗するようになりました。

```
01: trigger AccountTrigger on Account(after update) {
02:     List<Asset> assetsToUpdate = new List<Asset>();
03:     for (Account newA : Trigger.new) {
04:         Account oldA = Trigger.oldMap.get(newA.Id);
05:         if (oldA.Status__c != newA.Status__c) {
06:             assetsToUpdate.addAll( AccountHelper.getAssetsToUpdate(newA) );
07:         }
08:     }
09:     update assetsToUpdate;
10: }

25: }
26: }
```

開発者は、コードがすべてのアセットを正しく更新し続けるようにしつつ、この不具合に対処するために、コードのどの部分を変更すべきでしょうか？

- A. すべてのロジックを、アセットをクエリして更新する Queueable クラスに移動し、トリガーからそれを呼び出します。
- B. 14行目に List<asset> assets = [SELECT id, Status_ FROM WHERE AccountId = : acctId] を追加し、15行目の for ループで assets リストを反復処理します。
- C. SOQLクエリの16行目にLIMIT句を追加して、アカウントに対してクエリされる資産の数を制限します。
- D. getAssetsToUpdate メソッドを、すべてのアカウントを1つの呼び出しで処理するように変更し、03行目から始まる for ループの外で呼び出します。

正解: [\(正解を表示します\)](#)

Queueableを選ぶ理由とは？

* キュー対応のApexは非同期処理を可能にし、ガバナー制限に達することなく大量の更新を効率的に処理できます。

* トリガーは、資産の更新を処理するためにQueueableクラスをキューに追加します。

限界値超過を回避する：

* ロジックをQueueableクラスに移動することで、リソースを大量に消費する処理を非同期処理にオフロードできます。

質問: 9

ユニバーサルコンテナーズ社は、顧客とサービス担当者が注文を識別するために注文番号を使用する注文システムを採用しています。注文記録はSalesforceにインポートされます。

Salesforceの注文番号フィールドはどのように定義すべきですか？

A. 直接検索

B. 外部IDと一意性

C. 間接検索

D. ルックアップ

正解: ([正解を表示します](#))

質問: 10

開発者が、queueableインターフェースを実装するクラスの、キューに登録されたジョブのステータスを取得するには、どのような方法が2つありますか？

2つの回答を選択してください

A. Apexステータスページを表示する

B. Apex求人ページを見る

C. AsyncApexJob オブジェクトを照会する

D. Apex Flexキューを表示する

正解: ([正解を表示します](#))

* Apex求人ページ：

* Queueableインターフェースを実装しているジョブを含め、キューに登録されたジョブのステータスと詳細を表示するためのユーザーフレンドリーなインターフェースを提供します。

* AsyncApexJob オブジェクト：

* AsyncApexJob オブジェクトには、キューに登録された Apex ジョブに関する情報が格納されます。

* 開発者はこのオブジェクトを照会して、ステータス、ジョブタイプなどのジョブの詳細を取得できます。

SOQLクエリの例 :SELECT Id, Status, JobType, CreatedDate

AsyncApexJob から

WHERE JobType = 'Queueable'

* 他の選択肢はどうでしょうか？

* A. Apex ステータス ページ: Salesforce の関連機能ではありません。

* D. Apex Flex Queue: バッチ Apex ジョブにのみ適用され、キューイング可能なジョブには適用されません。

* AsyncApexJob: <https://developer.salesforce.com/docs/atlas.en-us.api.meta/api>

/sforce_api_objects_asyncapexjob.htm

参考文献：

質問: 11

Universal Containers は最近、Classic から Lightning Experience に移行しました。同社の業務プロセスの一つでは、商談ページでユーザーが開始したアクションに基づいて、商談オブジェクトから特定の値を HTTP REST コールアウト経由で外部の注文管理システムに送信する必要があります。値の例は次のとおりです。名前、金額、アカウント。開発者は、この業務要件を満たすために、どの2つのメソッドを実装すべきでしょうか？ 2つ選択してください)

- A. @Future(Callout=true) を使用してヘルパーメソッドを呼び出し、HTTP REST コールアウトを実行する Opportunity オブジェクトに更新後トリガーを作成します。
- B. Opportunity オブジェクト上にプロセス ビルダーを作成し、Opportunity が更新されるたびに、HTTP REST コールアウトを実行する Apex 即時アクションを実行します。
- C. HTTP REST コールアウトを実行する Lightning コンポーネントを作成し、Lightning アクションを使用して、そのコンポーネントを商談の詳細ページに表示します。
- D. HTTP REST コールアウトを実行する Visualforce ページを作成し、Visualforce クイックアクションを使用して、商談の詳細ページにコンポーネントを公開します。

正解: **A,C** ([コメントを发表する](#))

質問: 12

本番環境組織にデプロイする際に、必ず発生させる必要がある2つのイベントは何ですか？

2つの回答を選択してください

- A. すべてのカスタムオブジェクトは、開発以外の値に可視性を設定する必要があります。
- B. すべてのApexコードは、少なくとも75%のテストカバレッジを満たす必要があります。
- C. すべてのトリガーには、ある程度のテストカバレッジが必要です
- D. すべてのビジュアルフローは、少なくとも1%のテストカバレッジが必要です。

正解: **B,C** ([コメントを发表する](#))

* B :Salesforceでは、すべてのApexコードについて、デプロイ前に本番環境で少なくとも75%のテストカバレッジを達成することを義務付けています。これにより、システムの信頼性が確保されます。

* C: Apexコード全体の75%の要件の一部として、すべてのトリガーにはテストカバレッジが必要です。

他の選択肢はどうでしょうか？

* A: オブジェクトの表示設定は、デプロイメントの要件ではありません。

* D: ビジュアルフローは、本番環境へのデプロイに際してテストカバレッジを必要としません。

参考文献 :

* Apexテストカバレッジ要件

質問: 13

以下のApexコードを参照してください。

```
Integer x = 0;
do (
    x = 1;
    x++;
) while (x < 1);
System.debug(x);
```

デバッグログに書き込まれる際のxの値は何ですか？

- A. 2
- B. 0
- C. 1
- D. 3

正解: [\(正解を表示します\)](#)

質問: 14

ある会社には、商談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 Sales_Help_Request__cがあります。Sales_Help_Request__cには、Sales_Help_Request__cに費やし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、商談にtotal_Hour2__cというフィールドを作成するよう指示されています。このフィールドは、その商談に関連付けられたSales_Help_Request__cレコードのNumber_of_Hours__c値の合計となるようにする必要があります。

開発者はこれを実装するために何を使用すべきでしょうか？

- A. 商談オブジェクトに対するトリガー
- B. 商談オブジェクトのロールアップ集計項目
- C. Sales_Help_Request__c オブジェクト上のレコードトリガーフロー
- D. 販売Help_Request__cオブジェクトのロールアップ集計フィールド

正解: [\(正解を表示します\)](#)

質問: 15

開発者は、会社名が「Universal Containers」であるすべてのリード、アカウント、および連絡先のIDと名前を取得するために、どのクエリを使用すべきでしょうか？

- A. SELECT lead(id, name), account(id, name), contact(id,name) FROM Lead, Account, Contact WHERE Name = 'Universal Containers'
- B. CompanyName フィールドから「Universal Containers」を検索し、lead(id,name)、account(id,name)、contact(id,name) を返します。
- C. SELECT Lead.id, Lead.Name, Account.id, Account.Name, Contact.Id, Contact.Name FROM Lead, Account, Contact WHERE CompanyName = 'Universal Containers'
- D. IND 'Universal Containers' IN Name Fields RETURNING lead(id, name), account(id,name), contact(id,name)

正解: **D** [\(コメントを發表する\)](#)

質問: 16

外部IDを使用するメリットを2つ教えてください。

2つの回答を選択してください

- A. 外部IDはSalesforce Mobileで使用して、外部データを表示させることができます。
- B. 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。
- C. 外部IDフィールドは、別の外部システムのIDを参照するために使用できます。
- D. 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。

正解: ([正解を表示します](#))

有効的なPDI問題集はJPNTTest.com提供され、PDI試験に合格することに役に立ちます！JPNTTest.comは今最新PDI試験問題集を提供します。JPNTTest.com PDI試験問題集はもう更新されました。ここでPDI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/PDI-mondaishu> 205問、30%ディスカウント、特別な割引コード: **JPNshiken**」

質問: 17

あるソフトウェア会社は、Salesforceのアカウントオブジェクトを使用して、自社ソフトウェアの販売先企業を追跡しています。また、カスタムオブジェクトであるBug__cを使用して、ソフトウェアのバグを追跡するためにもSalesforceを利用しています。業務改善の一環として、どの企業がどのバグを報告したかを報告できるようにしたいと考えています。各企業は複数のバグを報告でき、また、1つのバグが複数の企業によって報告されることもあります。

この報告を可能にするには何が必要ですか？

- A. アカウムの Bug__c のロールアップ集計フィールド
- B. Bug__c のマスター/ディテール フィールドを Account に
- C. Bug__c のルックアップフィールドを Account にリンクします。
- D. Bug__c と Account 間の関数オブジェクト

正解: D ([コメントを发表する](#))

多対多の関係 :

どの企業 (アカウント) がどのバグを報告したかを追跡し、両方に対して複数の関連付けを可能にするには、結合オブジェクトが必要です。

この結合オブジェクトには、Bug__c と Account の 2 つのマスター/ディテール関係があります。

他の選択肢はどうでしょうか？

- A) ロールアップ集計フィールド: 多対多の関係には適していません。
- B) Bug__c の Account へのマスター/ディテール フィールド: 多対多ではなく、一対多の関係を作成します。
- C) Bug__c の Account へのルックアップ フィールド: 依然として一対多の関係が作成され、多対多の関係は作成されません。

参照:多対多の関係:https://developer.salesforce.com/docs/atlas.en-us.object_reference.

[meta/object_reference/relationships_manytomany.htm](https://developer.salesforce.com/docs/atlas.en-us.object_reference/meta/object_reference/relationships_manytomany.htm)

質問: 18

開発者がヘルパークラスを使用して新しいApexトリガーを作成し、新しいApexヘルパークラスの95%のみをテストするテストクラスを作成しました。変更セットを本番環境にデプロイすると、テストカバレッジの警告が表示されて失敗します。

「選択したApexトリガーのテストカバレッジは0%です。少なくとも1%のテストカバレッジが必要です。」開発者は、新しいApexトリガーとヘルパークラスを正常にデプロイするために何をすべきでしょうか？

- A. テストクラスから、落下しているテストメソッドを削除します。
- B. ヘルパークラスのテストクラスカバレッジを向上させる

C. すべてのテストを実行」メソッドを使用してテストを実行します。

D. Apexトリガーを網羅するテストクラスとメソッドを作成します。

正解: ([正解を表示します](#))

質問: 19

Universal Containers社は、新しいSalesforceアプリケーションの構築にあたり、宣言型開発のみを採用することを決定した。

アプリケーションのデータベース層を構築するために使用すべき3つのオプションはどれですか？

3つの回答を選択してください

A. カスタムオブジェクトとフィールド

B. トリガー

C. ロールアップサマリー

D. 人間関係

E. 流れ

正解: ([正解を表示します](#))

* A: カスタムオブジェクトとフィールドは、Salesforceのデータベース層におけるカスタムデータの作成と保存のための基本的な構成要素です。

* C: ロールアップ集計フィールドは、子レコードに対して計算（合計カウントなど）を実行し、その結果を親レコードに表示するために使用される宣言的な機能です。

* D: 関係（例マスター/ディテール、ルックアップ）は、データオブジェクトが互いにどのように関連しているかを定義するために不可欠です。他の選択肢はどうでしょうか？

* B: トリガーはプログラムによる解決策であり、宣言型のみという要件に反します。

* E: フローは自動化とロジックのために使用され、データベース層の構築には使用されません。

:

Salesforceデータモデリング

質問: 20

Lightning WebコンポーネントにカスタムSVGを含めるには、どの3つの手順が必要ですか？3つ選択してください。

A. SVGを静的リソースとしてアップロードします。

B. SVGをコンテンツアセットファイルとしてインポートします。

C. HTMLテンプレートでインポートを参照します。

D. 静的リソースをインポートし、JavaScriptでそのゲッターを提供します。

E. HTMLテンプレートでgetterを参照します。

正解: **A,D,E** ([コメントを發表する](#))

質問: 21

開発者は、単体テスト用のテストデータを作成するヘルパーメソッドを作成するよう依頼される。

```
01: public TestUtils (
02:
03:     public static Account createAccount() {
04:         Account act = new Account();
05:         // ...set some fields on acct...
06:         return act;
07:     }
08:     //...other methods...
09: }
```

TestUtilsクラスのメソッドが単体テストメソッドでのみ使用できるようにするには、TestUtilsクラスのどの部分を変更すればよいでしょうか？

- A. @isTest は 03 行目より上にあります。
- B. 01行目の上に@istestを追加します。
- C. 1行目のpublicをprivateに変更します。
- D. 3行目から静電気を除去します。

正解: **B** ([コメントを发表する](#))

TestUtils クラスとそのメソッドが単体テストメソッドからのみアクセス可能であることを保証するには、クラスに @isTest 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により、クラスとそのメソッドの範囲はテストコンテキストのみに制限されます。

なぜ@isTestなのか？

@isTest の目的:

@isTest アノテーションは、クラスまたはメソッドをテストシナリオ内でのみ使用するようにマークします。これにより、メソッドが本番コードで誤って呼び出されるのを防ぎます。

これにより、テストの分離性が向上し、不正使用が防止されます。

静的テストデータ手法:

テストデータ作成用のヘルパーメソッドに@isTestというアノテーションを付けることで、それらが本番コードに干渉せず、テスト目的のみに使用されることが保証されます。

オプション分析:

オプションA :3行目上の@isTest :

誤りです。createAccount メソッドのみに @isTest を適用すれば機能しますが、クラス内の他のメソッドにはアクセスできてしまいます。これでは、クラス全体を制限するという要件を満たしません。

オプションB :01行目の上に@isTestを追加する :

その通りです。クラスに注釈を付けることで、クラス内のすべてのメソッドがテストのコンテキストでのみアクセス可能になります。

オプションC :1行目のpublicをprivateに変更する。

誤りです。クラスをプライベートに変更するとコンパイルエラーが発生します。Salesforce のトップレベルクラスはプライベートとして宣言できないためです。

オプションD :3行目から静的ノイズを削除する :

誤りです。staticを削除しても、クラスやそのメソッドのテスト範囲には影響しません。staticメソッドは引き続き呼び出すことができ、テストコンテキストへのアクセスも制限されません。

正しい実装 :

修正後のコードは以下のようになります。

```
@isTest
public class TestUtils {
    public static Account createAccount() {
        Account acct = new Account();
        // アカウントにいくつかのフィールドを設定します
        アカウントを返します。
    }
    // その他のメソッド...
}
```

参考文献：

Apexテスト : テストクラスとメソッド

Apexテストに関するTrailheadモジュール

質問: 22

組織内の「アカウント」オブジェクトは、「支店」と呼ばれる子オブジェクトとマスター/ディテール関係にあります。以下の自動化が存在します。

- * 集計集計項目
- * カスタム検証ルール
- * 重複するルール

開発者がAccountオブジェクトにトリガーを作成しました。

トリガーコードをテストする際に、開発者はどのような2つの点を考慮すべきでしょうか？

2つの回答を選択してください

- A. 重複ルールは、すべての DML 操作がデータベースにコミットされた後に実行されます。
- B. 検証ルールにより、トリガーが再度発火します。
- C. ロールアップ集計フィールドは、親レコードの保存処理を引き起こす可能性があります。
- D. トリガーはトランザクション中に複数回発火する可能性があります。

正解: **C,D** ([コメントを發表する](#))

質問: 23

正しく評価される数値式はどれですか？ 2つ選択してください。)

- A. 小数d = 3.14159;
- B. ダブルd = 3.14159;
- C. 長l = 3.14159;
- D. 整数 l = 3.14159;

正解: ([正解を表示します](#))

質問: 24

レコードリスト内の一部のレコードは挿入できるが、他のレコードは挿入できない場合、どのステートメントを使用すればよいですか？

- A. レコードを挿入する
- B. Database.insert(records, false)

C. Database.insert(records, true)

D. 挿入 (レコード、偽)

正解: ([正解を表示します](#))

質問: 25

開発者が、売上記録の一覧を表示するLightning Webコンポーネントを作成しています。

営業担当者ユーザーは、各レコードの手数料欄を表示する必要があります。営業アシスタントユーザーは、手数料欄を除くすべてのレコード欄を表示する必要があります。

コンポーネントが両方のユーザーに対してエラーを表示せずに動作するようにするには、どのようにすればこれを強制できるでしょうか？

A. Lightning Locker Serviceを使用して、共有ルールとフィールドレベルのセキュリティを適用します。

B. Lightning Data Serviceを使用して販売記録のコレクションを取得します。

C. コンポーネントのデータを取得する SOQL で WITH SECURITY_ENFORCED を使用します。

D. Security.stripInaccessible Le を使用して、現在のユーザーがアクセスできないフィールドを削除します。

正解: ([正解を表示します](#))

質問: 26

開発者が親の Lightning Web コンポーネントの中に子 Lightning Web コンポーネントをネストして作成しました。親コンポーネントは子コンポーネントに文字列値を渡す必要があります。

これを実現するには、どのような2つの方法がありますか？

2つの回答を選択してください

A. 親コンポーネントは、Apex コントローラ クラスを使用して子コンポーネントにデータを送信できます。

B. 親コンポーネントは子コンポーネントのメソッドを呼び出すことができます

C. 親コンポーネントはパブリックプロパティを使用してデータを子コンポーネントに渡すことができます。

D. 親コンポーネントはカスタムイベントを使用してデータを子コンポーネントに渡すことができます。

正解: ([正解を表示します](#))

質問: 27

サンドボックス環境で作業中に、テストランナーでApexテストを実行すると失敗します。しかし、匿名実行ウィンドウでApexロジックを実行すると、例外やエラーは発生せず成功します。

なぜこのメソッドはサンドボックステストフレームワークでは失敗したのに、開発者コンソールでは成功したのでしょうか？

A. テストメソッドは、特定のユーザーとして実行するために system.runAs を使用しません。

B. テストメソッドが @future メソッドを呼び出しています。

C. テスト方法はサンドボックス内の既存データに依存します。

D. テストメソッドのコードに構文エラーがあります。

正解: ([正解を表示します](#))

テスト分離の原則：

seeAllData=trueを使用しない限り、Apexテストメソッドは既存の組織データにアクセスできないサンドボックス環境で実行されます。

テストが、テスト自体の中で作成されていないデータに依存している場合、テストは失敗します。

他の選択肢はどうでしょうか？

A:runAsはユーザーコンテキストをシミュレートしますが、テストの分離には影響しません。

B: @future メソッドは、正しく処理されれば障害を引き起こしません。

D: 構文エラーがあるとコードのコンパイルができません。

参考資料 :テストデータ分離 https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_testing_data.htm

質問: 28

検索語句を指定してアカウントのリストを返す Apex メソッド getAccounts は、Lightning Web コンポーネントで使用できます。getAccounts メソッドを使用する Lightning Web コンポーネント プロパティの正しい定義は何ですか？

A. @AuraEnabled(getAccounts, '\$searchTerm')

アカウントリスト;

B. @wire(getAccounts, '\$searchTerm')

アカウントリスト;

C. @AuraEnabled(getAccounts, {searchTerm: '\$searchTerm'})

アカウントリスト;

D. @wire(getAccounts, {searchTerm: '\$searchTerm'})

アカウントリスト;

正解: ([正解を表示します](#))

https://developer.salesforce.com/docs/component-library/documentation/en/48.0/lwc/lwc.data_wire_service_about

質問: 29

時間管理アプリケーションを開発している開発者は、各タイムカードの合計時間をアプリケーションユーザーが利用できるようにしたいと考えています。タイムカードのエントリは、タイムカードに対してマスター/ディテール関係を持っています。

開発者は、これを宣言的に実現するために、どの方法を用いるべきでしょうか？

A. タイムカードオブジェクトのロールアップ集計フィールドで、そのタイムカードのエントリから合計時間を計算します。

B. タイムカードの合計時間数を計算し、ページに表示するVisualforceページ

C. 集計クエリを使用して指定されたタイムカードの時間を計算し、カスタムフィールドに保存するApexトリガー

D. タイムカードエントリが作成されたときにタイムカードのフィールドを更新するプロセスビルダープロセス

正解: ([正解を表示します](#))

質問: 30

注文時に注文の送料を計算し、関連する注文商品の一部に送料の一定割合を適用するために使用できるプロセス自動化の2つのタイプはどれですか？

2つの回答を選択してください

A. フロービルダー

B. 承認プロセス

C. ワークフロールール

D. プロセスビルダー

正解: A,D ([コメントを發表する](#))

質問: 31

METADATA APIを使用してデプロイできるコンポーネントはどれですか？ (2つ選択してください)

- A. ケース設定
- B. Web-to-Lead
- C. リード変換設定
- D. Web-to-Case

正解: ([正解を表示します](#))

有効的なPDI問題集はJPNTTest.com提供され、PDI試験に合格することに役に立ちます！JPNTTest.comは今最新PDI試験問題集を提供します。JPNTTest.com PDI試験問題集はもう更新されました。ここでPDI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/PDI-mondaishu> 205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、アカウントの詳細ページに各アカウントのすべてのタスクを一覧表示したいと考えています。連絡先に対してタスクが作成された場合、開発者は関連するアカウントレコードにそのタスクを表示するために何をする必要がありますか？

- A. タスク情報を表示するアカウント数式項目を作成します。
- B. 特になし。タスクはアカウントページに自動的に表示されます。
- C. なし。タスクはアカウントと連絡先に関連付けることはできません。
- D. タスクを連絡先のアカウントに関連付ける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: 33

Universal Containers (UC) は、注文と注文明細項目の間にマスター/ディテール関係を持つ、既製の注文管理機能を使用しています。UCは各注文明細項目に在庫入荷予定日を保存し、すべての注文明細項目が在庫可能になった場合にのみ注文が出荷されます。注文の出荷予定日を計算するには、どの方法を使用すべきですか？

- A. 各利用可能日フィールドにDAYS関数を使用し、注文にCOUNTロールアップ集計フィールドを使用します。
- B. 最新の利用可能日フィールドにMAXロールアップ集計フィールドを使用します。
- C. 各最新利用可能日フィールドにLATEST関数を使用します。
- D. 最新の利用可能日の各フィールドに CEILING 関数を使用します。

正解: ([正解を表示します](#))

質問: 34

Visualforceページの内容をPDFとして表示するコードはどれですか？

- A. <apex:page renderAs="pdf">
- B. <apex:page contentType " application/pdf">
- C. <apex:page readerAs= "application/pdf">
- D. <apex:page readeras" application/pdf">

正解: A ([コメントを发表する](#))

質問: 35

カスタムオブジェクト Trainer__c には、別のカスタムオブジェクト Gym__c へのルックアップフィールドがあります。どのSOQLクエリを使えば、ビリディアンシティジムとそのトレーナー全員のレコードを取得できますか？

- A. SELECT Id, (SELECT Id FROM Trainer__c FROM Gym__c WHERE Name = 'Viridian City Gym'
- B. SELECT Id, (SELECT Id FROM Trainers__r) FROM Gym__c WHERE Name = 'Viridian City Gym'
- C. SELECT ID FROM Trainer__c WHERE Gym__r.Name = 'Viridian City Gym'
- D. SELECT Id, (SELECT Id FROM Trainers__c) FROM Gym__c WHERE Name = 'Viridian City Gym'

正解: ([正解を表示します](#))

質問: 36

開発者は、テキストフィールドに無効な値が自動的に入力されているという通知を受け取ります。しかし、これは設定されているカスタム検証ルールによって防止されるはずですが。何が原因でしょうか？

- A. ユーザーは検証ルールを抑制する権限セットに属しています
- B. 保存順序の実行中にDML例外が発生しました
- C. フィールドはbeforeトリガーによって入力されています
- D. このフィールドはワークフローフィールドの更新によって入力されています

正解: ([正解を表示します](#))

質問: 37

Application__c という名前の sObject は、Position__c という名前の別の sObject とルックアップ関係にあります。Application__c と Position__c の両方に、Status__c という名前の選択リスト項目があります。Position__c の Status__c 項目が更新されると、Application__c の Status__c 項目に同じ値が自動的に入力され、Application__c でワークフロールールが実行されるようにします。開発者はこれをどのように実現できますか？

- A. Application__c.Status__c を数式フィールドに変更します。
- B. ワークフローを使用してオブジェクト間のフィールド更新を設定します。
- C. Application__c.Status__c をロールアップ集計フィールドに変更します。
- D. DML操作でApexトリガーを使用します。

正解: ([正解を表示します](#))

質問: 38

開発者は、アカウントレコードページに挿入するための、statuscomponent という名前の Lightning Web 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利用可能にするために、どのような2つのことを行うべきでしょうか？

2つの回答を選択してください

- A. statuscomponent.js ファイルに <target>lightning_Recordpage</target> を追加します。
- B. statusComp.js-meta.xml ファイルに <target>lightning RecordPage</target> を追加します。
- C. statuscomponent.js-meta.xml ファイルで、Exposes を true に設定します。
- D. statusComponent.js-meta.xml ファイルに <mastertabel>Account </masterLabel> を追加します。

正解: ([正解を表示します](#))

レコードページでLWCを使用できるようにするには :

ターゲット構成:

コンポーネントの .js-meta.xml ファイルに <target>lightning__RecordPage</target> を追加して、コンポーネントを使用できる場所を指定します。

コンポーネントを公開する:

.js-meta.xmlファイルでisExposedtotrueを設定すると、コンポーネントが使用可能になります。

Example.js-meta.xmlファイル:<?xml version="1.0"encoding="UTF-8"?>

```
<LightningComponentBundlexmlns="http://soap.sforce.com/2006/04/metadata">
```

```
<apiVersion>57.0</apiVersion>
```

```
<isExposed>true</isExposed>
```

```
<ターゲット>
```

```
<target>lightning__RecordPage</target>
```

```
</targets>
```

```
</LightningComponentBundle>
```

A: 配置が間違っています。<target>タグはJavaScriptファイルではなく、.js-meta.xmlファイルに記述する必要があります。

D: <masterLabel>タグはメタデータのラベル付けに使用され、コンポーネントの公開やターゲット設定には使用されません。

参考資料 LWCメタデータ構成 <https://developer.salesforce.com/docs/component-library>

/documentation/en/lwc/lwc.create_metadata_config_file

質問: 39

Universal Containers社は、Salesforceの実装における特定のユースケースについて、宣言型開発とプログラムによるカスタマイズのどちらが優れているかを評価したいと考えている。

宣言型開発がプログラムによるカスタマイズに比べて持つ2つの特徴は何ですか？

2つの回答を選択してください

- A. 宣言型開発は、設計上の制約とクエリの制約がより大きい。
- B. 宣言型開発では、Apex テスト クラスは必要ありません。
- C. 宣言型開発はセットアップUIを使用して行うことができます。
- D. 宣言型コードロジックは保守やレビューを必要としません。

正解: ([正解を表示します](#))

質問: 40

Universal Containers社は、マーケティング部門向けに新しいLightning Webコンポーネントを開発中です。同社は、このコンポーネントが綿密に調整され、シームレスなユーザーエクスペリエンスを提供できることを確実にしたいと考えています。

Lightning Componentフレームワークを使用するメリットは何ですか？

- A. クライアントサイドレンダリングによりパフォーマンスが向上します
- B. アクセシビリティ規格の自動サポート
- C. すべてのウェブブラウザとの互換性
- D. サードパーティライブラリとの容易な統合

正解: **A,B,D** ([コメントを发表する](#))

* オプションA :Lightningコンポーネントフレームワークは、パフォーマンスと応答性を向上させるためにクライアントサイドレンダリングを使用します。

* オプションB :Lightningコンポーネントはアクセシビリティ標準 (ARIAなど)に自動的に準拠します。

* オプションD :このフレームワークは、JavaScriptを介して外部ライブラリとのシームレスな統合を可能にします。

* 不適切:

* オプションC :Lightningコンポーネントは最新のブラウザと互換性がありますが、すべてのブラウザと互換性があるわけではありません。

質問: 41

Lightning Web コンポーネントのカスタムイベントに当てはまる2つの特徴はどれですか？

2つの回答を選択してください

- A. デフォルトでは、カスタムイベントは直近のコンテナと直近の子コンポーネントにのみ伝播します。
- B. @wireで装飾されたプロパティを使用して、カスタムイベントのペイロードにデータを渡すことができます。
- C. デフォルトでは、カスタムイベントは直近のコンテナにのみ伝播します。
- D. データは、detail と呼ばれるプロパティを使用して、カスタムイベントのペイロードに渡すことができます。

正解: C,D ([コメントを发表する](#))

質問: 42

開発者が作成したこのApexトリガーは、MyClass.myStaticMethodを呼び出します。

```
trigger myTrigger on Contact (before insert)
{
    MyClass.myStaticMethod(trigger.new);
}
```

開発者は、MyClass.myStaticMethod を直接呼び出すテスト メソッドを持つテスト クラスを作成し、結果として全体のコード カバレッジが 81% になります。他のコードが存在しないと仮定して、開発者がトリガーと 2 つのクラスを本番環境にデプロイしようとする、何が起こるでしょうか？

- A. テストメソッド内でアサーションが行われなかったため、デプロイが失敗しました。
- B. Apexコードが必要な75%以上のコードカバレッジを満たしているため、デプロイメントは成功します。
- C. Apexトリガーにコードカバレッジがないため、デプロイが失敗しました。
- D. クラスとトリガーの両方がデプロイメントに含まれているため、デプロイメントは成功します。

正解: C ([コメントを发表する](#))

質問: 43

Lightning Webコンポーネントで使用できるApexメソッドgetAccountsは、検索語を指定してアカウントのリストを返します。

getAccounts メソッドを使用する Lightning Web コンポーネント プロパティの正しい定義は何ですか？

```
A. @wire(getAccounts, { searchTerm: '$searchTerm' })
   accountList;
B. @AuraEnabled(getAccounts, '$searchTerm')
   accountList;
C. @AuraEnabled(getAccounts, { searchTerm: '$searchTerm' })
   accountList;
D. @wire(getAccounts, '$searchTerm')
   accountList;
```

- A. オプションC
- B. 選択肢D
- C. オプションB

D. オプションA

正解: ([正解を表示します](#))

質問: 44

Lightning Webコンポーネントで使用するために、AccountControllerクラスにupdateAccountsという名前のApexメソッドを含めるコードステートメントはどれですか？

- A. import updateAccounts from "AccountControlles";
- B. import updateAccounts from "Salesforce/apex/AccountController:";
- C. import updateAccounts from "Account@ontroller.updateAccounts";
- D. import updateAccounts from "@salesforce/apex/AccountController.updateAccounts";

正解: ([正解を表示します](#))

Lightning Web Component (LWC) を使用する場合、Apex メソッドをインポートするための正しい構文は次のとおりです。

```
import methodName from "@salesforce/apex/ControllerName.methodName";
```

他の選択肢はどうでしょうか？

A、B、C :これらはLWCでApexメソッドをインポートする際に誤った構文を使用しています。

Lightning Web Components ドキュメント

質問: 45

Lightning Web コンポーネントのカスタムイベントに当てはまる2つの特徴はどれですか？

2つの回答を選択してください

- A. デフォルトでは、カスタムイベントは直近のコンテナと直近の子コンポーネントにのみ伝播します。
- B. データは、detail と呼ばれるプロパティを使用して、カスタムイベントのペイロードに渡すことができます。
- C. データは、ワイヤーで装飾されたプロパティを使用して、カスタムイベントのペイロードに渡される場合があります。
- D. デフォルトでは、カスタムイベントは直近のコンテナにのみ伝播します。

正解: ([正解を表示します](#))

質問: 46

開発者が外部システムを呼び出すApexコードを作成しました。テストカバレッジを確保するために、開発者はどのようにテストを記述すべきでしょうか？

- A. HTTPCalloutMock を継承するクラスを作成します。
- B. WebserviceMockを継承するクラスを作成します。
- C. HTTPCalloutMockインターフェースを実装するクラスを作成します。
- D. WebserviceMockインターフェースを実装するクラスを作成します。

正解: C ([コメントを发表する](#))

有効的なPDI問題集はJPNTTest.com提供され、PDI試験に合格することに役に立ちます！JPNTTest.comは今最新PDI試験問題集を提供します。JPNTTest.com PDI試験問題集はもう更新されました。ここでPDI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/PDI-mondaishu> 205問、30%ディスカウント、特別な割引コード: **JPNshiken**」

質問: 47

トリガーが同じオブジェクトとイベントに関連付けられている場合の実行順序を説明しているのは、次のうちどれですか？

- A. トリガーは変更された順序で実行されます。
- B. トリガーの実行順序は保証されません。
- C. トリガーはトリガー名のアルファベット順に実行されます。
- D. トリガーは作成された順序で実行されます。

正解: [\(正解を表示します\)](#)

同じオブジェクトとイベントに複数のトリガーが関連付けられている場合、Salesforce は実行順序を保証しません。そのため、オブジェクトごとにすべてのロジックを単一のトリガーに統合し、ヘルパークラスを使用して実行順序を制御することが推奨されます。

質問: 48

開発者は、カスタムオブジェクト Project_ct に次の3つのロールアップサマリーフィールドを作成しました。

```
Total_Timesheets__c
Total_Approved_Timesheets__c
Total_Rejected_Timesheet__c
```

開発者は、特定のプロジェクトにおける却下されたタイムシートと承認されたタイムシートの比率を示す新しいフィールドを作成するよう求められています。

開発者は、保守コストを最小限に抑えるために、ビジネス要件を実装する際にどちらを使用すべきでしょうか？

- A. エイペックストリガー
- B. プロセスビルダー
- C. レコードトリガーフロー
- D. 数式フィールド

正解: [D \(コメントを发表する\)](#)

質問: 49

ユニバーサルコンテナーズ社は、顧客サービス担当者が注文を識別するために注文番号を使用する注文システムを採用しています。注文記録はSalesforceにインポートされます。

Salesforceで「注文番号」フィールドはどのように定義すべきでしょうか。

- A. 直接検索
- B. 外部ID付き番号
- C. 間接ルックアップ
- D. ルックアップ

正解: [B \(コメントを发表する\)](#)

質問: 50

Lightning Web Component のどのカスタムイベントプロパティ設定を使用すると、イベントが包含階層をバブリングアップして Shadow DOM 境界を越えることができるようになりますか？

- A. バブル: true、構成: false
- B. バブル: true、構成済み: true
- C. bubbles: false, composed: true
- D. バブル: false、構成: false

正解: ([正解を表示します](#))

質問: 51

開発者は、アカウントレコードを保存するためのVisualforceページとカスタムコントローラを持っています。開発者は、検証ルール違反が発生した場合、それをユーザーに表示したいと考えています。検証ルール違反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. メッセージを表示するためのカスタムコントローラ属性を追加します。
- B. Visualforce ページに <apex:messages> を含めます。
- C. Database.upsert() メソッドを使用して DML を実行します。

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

- D. カスタム例外クラスを使用したtry/catchを使用します。

正解: ([正解を表示します](#))

質問: 52

開発者は顧客の購読レコードをSalesforceにインポートし、既存のアカウントレコードに関連付ける必要があります。購読レコードが正しいアカウントレコードに関連付けられるようにするために、開発者はどの2つのアクションを実行する必要がありますか？ 2つ選択してください。

- A. 名前フィールドをインポートしたファイルの列と一致させる
- B. インポートしたファイルの列とIDフィールドを一致させる
- C. 自動採番フィールドをインポートファイルの列に一致させる
- D. 外部IDテキストフィールドをインポートファイルの列に一致させる

正解: ([正解を表示します](#))

質問: 53

Universal Containers社は、ユーザーが複数のレコードを編集できるVisualforceページを表示するリストボタンを実装したいと考えています。

この要件を満たすVisualforceの機能はどれですか？

- A. カスタムリストコントローラ拡張機能を備えた標準コントローラ
- B. recordsetvarページ属性を持つカスタムリストコントローラ
- C. コントローラ拡張機能と<spex:listButton>タグ
- D. 標準コントローラとrecordsetvarページ属性

正解: D ([コメントを發表する](#))

* <apex:page> タグの recordSetVar 属性を使用すると、Visualforce ページで複数のレコードを処理できます。これは、一括編集用のページを表示するリスト ボタンに最適です。

他の選択肢はどうでしょうか？

* A: 標準コントローラは拡張できますが、このシナリオでは recordSetVar を使用するのが最も直接的でサポートされている方法です。

* B: Visualforceには「カスタムリストコントローラ」という概念はありません。

* C: <spex:listButton> は有効な Visualforce タグではありません。

参考文献：

* RecordSetVar を使用した Visualforce 標準コントローラ

質問: 54

以下のApexコードを参照してください。

```
Integer x = 0;
do {
    x = 1;
    x++;
} while (x < 1);
System.debug(x);
```

デバッグログに書き込まれる際のxの値は何ですか？

A. 0

B. 1

C. 2

D. 4

正解: [\(正解を表示します\)](#)

* 実行フロー:

* do-whileループは、条件を評価する前にループ本体を実行します。

* 初期値 x = 0。

* ループ内: x = 1 (再代入) となり、その後 2 にインクリメントされます。

* 最初の反復後、条件 x < 1 が偽となるため、ループが終了します。

デバッグ出力: System.debug(x) は 2 をログに出力します。

質問: 55

質の高いデータを確保するのに役立つ3つの方法は何ですか？

A. Apexでの例外処理

B. 検証ルールを追加する

C. beforeトリガーでフィールドにエラーを追加する

D. ルックアップフィルターを作成する

E. ワークフロールールを使用してメールアラートを送信する

正解: [B,D,E \(コメントを发表する\)](#)

質問: 56

Lightning Webコンポーネントで使用できるApexメソッドgetAccountsは、検索語を指定してアカウントのリストを返します。

getAccounts メソッドを使用する Lightning Web コンポーネント プロパティの正しい定義は何ですか？

```
A. @wire(getAccounts, { searchTerm: '$searchTerm' })
   accountList;
B. @AuraEnabled(getAccounts, '$searchTerm' )
   accountList;
C. @AuraEnabled(getAccounts, { searchTerm: '$searchTerm' })
   accountList;
D. @wire(getAccounts, '$searchTerm')
   accountList;
```

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

正解: ([正解を表示します](#))

質問: 57

Universal Containers社は、新しいSalesforceアプリケーションの構築にあたり、宣言型開発のみを採用することを決定した。アプリケーションのデータベース層を構築するために使用すべき3つのオプションはどれですか？

3つの回答を選択してください

- A. カスタムオブジェクトとフィールド
- B. トリガー
- C. ロールアップサマリー
- D. 人間関係
- E. 流れ

正解: ([正解を表示します](#))

A: カスタムオブジェクトとフィールドは、Salesforceのデータベース層における基本的な構成要素であり、カスタムデータの作成と保存に使用されます。

C: ロールアップ集計フィールドは、子レコードに対して計算（合計カウントなど）を実行し、その結果を親レコードに表示するために使用される宣言型機能です。

D: 関係性（例マスター/ディテール、ルックアップ）は、データオブジェクトが互いにどのように関連しているかを定義するために不可欠です。

他の選択肢はどうでしょうか？

B :トリガーはプログラムによる解決策であり、宣言型のみという要件に反します。

E: フローは自動化とロジックのために使用されるものであり、データベース層の構築に使用されるものではありません。

Salesforceデータモデリング

質問: 58

Universal Hiring は、Salesforce を使用して求人応募を受け付けています。Salesforce 管理者は、マスター オブジェクトとして機能する Job c と、詳細として機能する Job application. __: という 2 つのカスタム オブジェクトを作成しました。

開発者がビジネス要件を満たすために推奨されるツールは何ですか？

- A. レコードトリガーフロー

- B. 数式フィールド
- C. プロセスビルダー
- D. 頂点トリガー

正解: **B** ([コメントを发表する](#))

質問: **59**

開発者が組織内でテストを実行する3つの方法は何ですか？3つ選択してください。

- A. バルクAPI
- B. ツールAPI
- C. セットアップメニュー
- D. Salesforce DX
- E. APIメタデータ。

正解: ([正解を表示します](#))

説明

https://developer.salesforce.com/docs/atlas.en-us.sfdx_dev.meta/sfdx_dev/sfdx_dev_testing.htm

https://developer.salesforce.com/docs/atlas.en-us.api_tooling.meta/api_tooling/tooling_api_objects_apextestsuite

質問: **60**

開発者は、親Lightning Webコンポーネントの中に子Lightning Webコンポーネントをネストして作成しました。親コンポーネントは、子コンポーネントに文字列値を渡す必要があります。

これを実現するには、どのような2つの方法がありますか？

2つの回答を選択してください

- A. 親コンポーネントはパブリックプロパティを使用してデータを子コンポーネントに渡すことができます。
- B. 親コンポーネントは子コンポーネントのパブリックメソッドを呼び出すことができます。
- C. 親コンポーネントはカスタムイベントを使用してデータを子コンポーネントに渡すことができます。
- D. 親コンポーネントは、Apex コントローラ クラスを使用して子コンポーネントにデータを送信できます。

正解: **A,B** ([コメントを发表する](#))

Lightning Web Components (LWC) では、親コンポーネントと子コンポーネント間の通信は、いくつかの方法で実現できます。

オプションA：親コンポーネントは、パブリックプロパティを使用してデータを子コンポーネントに渡すことができます。

正しい方法。

子コンポーネントは、`@api` デコレータを使用してパブリック プロパティを宣言できます。

親コンポーネントは、テンプレート内のこのプロパティを介して子コンポーネントにデータを渡すことができます。

例：

子コンポーネント (childComponent.js)：

```
import { LightningElement, api } from 'lwc';
export default class ChildComponent extends LightningElement {
  @api
  私の所有物;
}
```

親コンポーネントテンプレート (parentComponent.html)：

<テンプレート>

```
<c-child-component my-property={parentValue}></c-child-component>
```

```
</template>
```

子コンポーネントは、@api デコレータを使用してパブリックメソッドを公開できます。

親要素はDOMクエリセレクタを使用してこのメソッドを呼び出すことができます。

例：

子コンポーネント (childComponent.js)：

```
import { LightningElement, api } from 'lwc';
export default class ChildComponent extends LightningElement {
  @api
  myMethod(value) {
    // 値を処理する
  }
}
```

親コンポーネント (parentComponent.js)：

```
import { LightningElement } from 'lwc';
export default class ParentComponent extends LightningElement {
  handleClick() {
    const childComponent = this.template.querySelector('c-child-component');
    childComponent.myMethod('some value');
  }
}
```

LWCのカスタムイベントは、子と親の間のコミュニケーションに使用されます。

子コンポーネントは、親コンポーネントと通信するためにイベントをディスパッチすることができます。

LWCでは、出来事は親から子へと一方的に流れるものではない。

親コンポーネントと子コンポーネント間でデータを渡すためにApexコントローラーを使用することは不要であり、非効率的です。

Apexコントローラーは、コンポーネント間の通信ではなく、サーバー側のデータ操作に使用されます。

参照：

子コンポーネントへのデータの受け渡し

オプションB：親コンポーネントは、子コンポーネントのパブリックメソッドを呼び出すことができます。

正しい方法。

子コンポーネントのメソッド呼び出し

不適切なオプション：

オプションC：親コンポーネントはカスタムイベントを使用して、データを子コンポーネントに渡すことができます。

親子間のコミュニケーションには不適切です。

イベントとのコミュニケーション

オプションD：親コンポーネントはApexコントローラクラスを使用して子コンポーネントにデータを送信できます。

不適切です。

Apexメソッドの呼び出し

結論：


親コンポーネントは、オプションA (パブリックプロパティ)とオプションB (パブリックメソッド)を使用して、文字列値を子コンポーネントに渡すことができます。

これらは、LWC (リビング・ウィメンズ・コミュニティ)における親子間のコミュニケーションの標準的かつ推奨される方法です。

質問: 61

開発者は、インスタンス化できない ShippingCalculator クラスを作成し、サブクラスがオーバーライドできる、動作するデフォルトの calculate メソッドの実装を含める必要があります。

ShippingCalculator クラスの正しい実装方法は何か？

- 
- ```
A. public abstract class ShippingCalculator {
 public abstract calculate() { /*implementation*/ }

B. public abstract class ShippingCalculator {
 public virtual void calculate() { /*implementation*/ }

C. public abstract class ShippingCalculator {
 public void calculate() { /*implementation*/ }

D. public abstract class ShippingCalculator {
 public override calculate() { /*implementation*/ }
```

A. オプションA

B. オプションD

C. オプションB

D. オプションC

正解: C ([コメントを发表する](#))

有効的なPDI問題集はJPNTTest.com提供され、PDI試験に合格することに役に立ちます！JPNTTest.comは今最新PDI試験問題集を提供します。JPNTTest.com PDI試験問題集はもう更新されました。ここでPDI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/PDI-mondaishu> 205問、30%ディスカウント、特別な割引コード: **JPNshiken**」

#### 質問: 62

Universal Containers (UC) は、注文と注文明細項目の間にマスター/ディテール関係を持つ、既製の注文管理機能を使用しています。

UCは各注文明細項目に在庫入荷予定日を保存し、すべての注文明細項目が在庫可能になった場合にのみ注文が出荷されます。

注文の出荷予定日を計算するには、どの方法を使用すべきですか？

A. 各最新利用可能日フィールドに 2 LATEST 関数を使用します。

- B. 最新の利用可能日の各フィールドに CEILING 関数を使用します。
- C. 各利用可能日フィールドに @ DAYS 関数を使用し、注文に COUNT ロールアップ サマリー フィールドを使用します。
- D. 最新の利用可能日フィールドに MAX ロールアップ集計フィールドを使用します。

正解: ([正解を表示します](#))

- \* ロールアップ集計フィールド:
- \* 注文と注文明細項目の関係はマスター/ディテールであるため、注文オブジェクトにロールアップ集計フィールドを使用できます。
- \* ロールアップ集計フィールドのMAX関数は、注文明細項目のすべての利用可能日の中から最新の日付（最大値を計算します。
- \* MAXロールアップサマリーフィールドが機能する理由:
- \* 注文明細項目の中で最も遅い入荷予定日によって、注文全体の出荷可能日が決まります。
- \* 他の選択肢はどうでしょうか？
- \* AとB：日付フィールドには、「LATEST」や「CEILING」と呼ばれる数式は適用できません。
- \* C: COUNT ロールアップ サマリーは日付の計算には関係ありません。
- \* ロールアップ集計項目: [https://developer.salesforce.com/docs/atlas.en-us.object\\_reference.meta/object\\_reference/roll\\_up\\_summary\\_fields.htm](https://developer.salesforce.com/docs/atlas.en-us.object_reference.meta/object_reference/roll_up_summary_fields.htm)

参考文献：

### 質問: 63

開発者は、アカウントレコードページに挿入するための、statuscomponent という名前の Lightning Web 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利用可能にするために、どのような2つのことを行うべきでしょうか？

2つの回答を選択してください

- A. statuscomponent.js ファイルに <target>lightning\_Recordpage</target> を追加します。
- B. statusComp .js-meta.xml ファイルに <target>lightning RecordPage</target> を追加します。
- C. statuscomponent.js-meta.xml ファイルで、Exposes を true に設定します。
- D. statusComponent.js-meta.xml ファイルに <mastertabel>Account </masterLabel> を追加します。

正解: **B,C** ([コメントを发表する](#))

レコードページでLWCを使用できるようにするには：

- \* ターゲット構成:
- \* コンポーネントの .js-meta.xml ファイルに <target>lightning\_\_RecordPage</target> を追加して、コンポーネントを使用できる場所を指定します。
- \* コンポーネントを公開する:
- \* .js-meta.xml ファイルで isExposed を true に設定して、コンポーネントを使用できるようにします。

Example.js-meta.xmlファイル:<?xml version="1.0"encoding="UTF-8"?>

```
<LightningComponentBundlexmlns="http://soap.sforce.com/2006/04/metadata">
```

```
<apiVersion>57.0</apiVersion>
```

```
<isExposed>true</isExposed>
```

```
<ターゲット>
```

```
<target>lightning__RecordPage</target>
```

```
</targets>
```

```
</LightningComponentBundle>
```

- \* A: 配置が間違っています。<target>タグはJavaScriptファイルではなく、.js-meta.xmlファイル内に配置する必要があります。
- \* D: <masterLabel>タグはメタデータのラベル付けに使用され、コンポーネントの公開やターゲット設定には使用されません。

参照: LWC メタデータ構成: [https://developer.salesforce.com/docs/component-library/documentation/en/lwc/lwc.create\\_metadata\\_config\\_file](https://developer.salesforce.com/docs/component-library/documentation/en/lwc/lwc.create_metadata_config_file)

**質問: 64**

開発者が作成したコードブロックには、共有を有効にするか無効にするかを示すステートメントが一切含まれていません。Salesforce組織内でこのコードを実行するユーザーに対して、組織全体のデフォルト設定と共有設定が自動的に適用されるのはどのような場合でしょうか？

- A. HTTPコールアウト
- B. Apexコントローラー
- C. 匿名ブロック
- D. エイペックストリガー

正解: [\(正解を表示します\)](#)

**質問: 65**

Universal Containersの開発者は、自社のSalesforce管理者が利用できる新しいSalesforceアプリケーションを実装する任務を負っている。アプリケーションのビジネスロジック層を構築する際に考慮すべき3つの要素はどれですか？3つ選択してください。

- A. 呼び出し可能なアクション
- B. 検証ルール
- C. ワークフロー
- D. スケジュールされたジョブ
- E. プロセスビルダー

正解: [A,C,E \(コメントを發表する\)](#)

**質問: 66**

Visualforceページには、親オブジェクトへのルックアップ関係を持つオブジェクト用の標準コントローラがあります。開発者は親レコードのデータをページ上にどのように表示できますか？

- A. 親レコードのページに2つ目の標準コントローラーを追加することで。
- B. VisualforceページでSOQLを使用して、親レコードからデータをクエリします。
- C. 子レコードのロールアップ数式フィールドを使用して、親レコードのデータを含める。
- D. マージフィールド構文を使用して親レコードからデータを取得します。

正解: [D \(コメントを發表する\)](#)

**質問: 67**

開発者はどこでマネージドパッケージを構築するのでしょうか？

- A. 開発者版
- B. 部分コピーサンドボックス
- C. 開発者サンドボックス
- D. 無制限版

正解: [A \(コメントを發表する\)](#)

**質問: 68**

開発者は、システムに存在する ContactSearch Apex クラスのセキュリティレビューを実施するよう指示されました。開発者は、そのクラス内で、以下のメソッドをセキュリティ上の脅威として特定しました。

```
List<Contact> performSearch(String lastName){
 return Database.query('SELECT Id, FirstName, LastName FROM Contact WHERE LastName
Like
'+lastName+'%');
}
```

開発者がSOQLインジェクション攻撃を防ぐためにメソッドを更新できる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. パラメータを使用する前に、escapeSingleQuotes メソッドを使用してパラメータをサニタイズします。
- B. パラメータに正規表現を使用して特殊文字を削除します。
- C. クラスに@ReadOnlyアノテーションとwith sharingキーワードを使用します
- D. 変数バインディングを使用して、動的クエリを静的な SOQL に置き換えます。

正解: [\(正解を表示します\)](#)

質問: 69

パッケージ管理機能は、どのタイプの組織で作成できますか？

- A. 開発者版
- B. 部分コピーサンドボックス
- C. 無制限版
- D. 開発者サンドボックス

正解: [\(正解を表示します\)](#)

質問: 70

開発者は、Expense \_\_c オブジェクト上で以下のトリガーを特定しました。

トリガーは、それぞれ削除イベント、挿入イベント、更新イベントの前に処理を実行します。

トリガーのベストプラクティスを確実に遵守するために、開発者はどの2つの手法を導入すべきでしょうか？

```
deleteExpense,
applyDefaultsToExpense,
validateExpenseUpdate;
```

2つの回答を選択してください

- A. 3つのトリガーすべてを、すべてのイベントを含む Expense\_\_c オブジェクト上の単一のトリガーに統合します。
- B. Expense\_\_c オブジェクト上の3つのトリガーはすべて維持しますが、Apex ロジックをトリガ一定義から移動します。
- C. before insert と before update のトリガーを統合し、削除アクションには Flow を使用します。
- D. レコードが保存されたときに適切なロジックを実行するヘルパークラスを作成します。

正解: B,D [\(コメントを發表する\)](#)

質問: 71

Universal Containers社は、個人に関する情報を格納するApplicantオブジェクトと、求人情報を表すJobオブジェクトを備えた採用アプリを開発しています。各応募者は複数の求人に応募できます。

応募者が求人に応募したことを示すために、開発者はどのような機能を実装すべきでしょうか？

- A. 応募者と求人をつなぐオブジェクト
- B. 応募者から求人へのルックアップフィールド
- C. 応募者から求人へのマスター/ディテールフィールド
- D. 応募者に関するジョブを参照する数式フィールド

正解: ([正解を表示します](#))

\* A. 応募者と求人との間の結合オブジェクト:

\* 応募者は複数の求人に応募でき、各求人には複数の応募者がいる可能性があるため、これは多対多の関係です。

\* Salesforceでは、多対多の関係を表すために結合オブジェクトが必要です。結合オブジェクトには、ApplicantオブジェクトとJobオブジェクトへの2つのマスター/ディテール関係が存在します。

\* ジャンクションオブジェクトには、JobApplication\_\_cのような名前を付けることができ、これは特定の求人に応募する応募者の具体的なインスタンスを表します。

\* なぜこれが正しいアプローチなのか？

\* ジャンクションオブジェクトを使用すると、多対多の関係において、堅牢なデータ管理およびレポート機能を実現できます。

\* この設計により、応募者と求人の組み合わせごとに、JobApplication\_\_c ジャンクションオブジェクトに一意のレコードとして記録されることが保証されます。

\* また、申請日、ステータス、フィードバックなど、申請に関する追加情報を保存することもできます。

他の選択肢はどうでしょうか？

\* B. 応募者から求人への検索フィールド:

\* ルックアップフィールドは、1対多の関係を作成します。応募者は1つの求人を参照できますが、このシナリオで必要とされる多対多の関係はサポートしていません。

\* C. 応募者から求人へのマスター/ディテールフィールド:

マスター/ディテール関係は1対多の関係であり、多対多の関係には適していません。また、応募者と求人を直接関連付けるために、単一のオブジェクトに2つのマスター/ディテールフィールドを設定することはできません。

\* D. 応募者に関するジョブを参照する数式フィールド:

数式フィールドは、レコード間の関係を確立したり、多対多の関係を表したりすることはできません。数式フィールドは計算フィールド専用です。

参考文献:

\* ジャンクションオブジェクトを使用した多対多リレーションシップの作成

\* マスター/ディテール関係とルックアップ関係の詳細

質問: 72

以下のコードスニペットを実行すると、どのような結果が得られますか？

```
public void doWork(Account salesforce) {
 for (Integer i = 0; i <= 200; i++){
 insert acct;
 }
}
```

- A. アカウントが挿入されました。
- B. 200アカウントが挿入されます。
- C. 201のアカウントが挿入されました。
- D. アカウントが挿入されました。

正解: ([正解を表示します](#))

#### 質問: 73

カスタムの成功メッセージを表示するように設定するには、どのアクションタイプを選択すればよいですか？

- A. 記録を更新する。
- B. 事件を終結させる。
- C. フィードアイテムを投稿します。
- D. レコードを削除します。

正解: ([正解を表示します](#))

#### 質問: 74

Salesforce管理者はFlow Builderを使用して「accountOnboarding」という名前のフローを作成しました。このフローはAuraコンポーネント内で使用する必要があります。

コンポーネント内でフローを表示するには、開発者はどのタグを使用すべきでしょうか？

- A. 雷 : 流れ
- B. 超低速
- C. シュアフロー
- D. オーラ : 流れ

正解: ([正解を表示します](#))

Auraコンポーネントにフローを埋め込むには、lightning:flowtagを使用します。これにより、開発者はAuraコンポーネント内でフローを宣言的に操作できるようになります。

参照:lightning:flow ドキュメント

誤った選択肢 :

B、C、D :これらは無効なタグ、または存在しないタグです。

#### 質問: 75

Visualforceにおけるゲッターメソッドとセッターメソッドについて、正しい記述は次のうちどれですか？ (2つ選択)

- A. ゲッターメソッドは、コントローラからページに値を渡します。
- B. ゲッターメソッドが呼び出される順序は保証されません。

- C. 各ゲッターメソッドには、対応するセッターメソッドが必要です。
- D. セッターメソッドは常にグローバルに宣言する必要があります。

正解: **A,C** ([コメントを发表する](#))

質問: 76

開発者はどのようにしてすべての商談レコードタイプのラベルを取得してリストコレクションに入力できますか？ 2つ選択してください

- A. Opportunity オブジェクトの説明結果を取得します。
- B. RecordType オブジェクトを反復処理する SOQL for ループを作成します。
- C. Opportunity.RecordType.Name フィールドから値を抽出する for ループを作成します。
- D. グローバル変数 \$RecordType を使用して、マップからリストを抽出します。

正解: **A,B** ([コメントを发表する](#))

有効的なPDI問題集はJPNTTest.com提供され、PDI試験に合格することに役に立ちます！JPNTTest.comは今最新PDI試験問題集を提供します。JPNTTest.com PDI試験問題集はもう更新されました。ここでPDI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/PDI-mondaishu> 205問、30%ディスカウント、特別な割引コード: **JPNshiken**」

質問: 77

Apexクラスを作成する際、開発者は開発中のすべての機能が要件どおりに処理されることを確認したいと考えます。開発者は、Apexクラスが仕様どおりに動作していることを確認するために、どの方法を用いるべきでしょうか？

- A. ビジネスロジックを実行するテストクラスを作成し、開発者コンソールでテストを実行します。
- B. 開発者コンソールの匿名実行ブロックでコードを実行します。
- C. セーブポイントと Database.rollback() を含めます。
- D. Apex クラスに try/catch ブロックを追加します。

正解: **A** ([コメントを发表する](#))

Apexクラスが指定された要件に従って機能していることを確認するために、開発者はビジネスロジックを実行し、期待どおりに動作することを確認するテストクラスを作成する必要があります。

オプションA :ビジネスロジックを実行するテストクラスを作成し、開発者コンソールでテストを実行します。

正しいアプローチ。

テストクラスを作成することで、開発者はApexクラスの機能を検証する単体テストを作成できます。

テスト手法を用いることで、クラスが様々な条件下で意図どおりに動作することを確認し、要件への準拠を保証することができる。

開発者コンソールでテストを実行すると、開発者は結果を確認し、必要に応じてデバッグを行うことができます。

利点 :

自動テスト :テストの再現性と一貫性を確保します。

コードカバレッジ :デプロイメントに必要なコードカバレッジの達成を支援します。

回帰テスト :既存の機能を損なう可能性のある将来の変更を検出しやすくします。

匿名実行でコードを実行すると迅速なテストは可能になりますが、徹底的なテストには適していません。

出力が期待される結果と一致することを確認するためのアサーションは提供されていません。

匿名実行」で実行されたテストは保存されないため、回帰テストのために簡単に再実行することはできません。

セーブポイントとロールバックを使用すると、データベースに変更をコミットすることなく、DML操作をテストするのに役立ちます。しかし、これだけではクラスが機能要件を満たしていることを保証するものではありません。try/catchブロックを追加することで例外処理は可能になりますが、クラスが仕様どおりに機能するかどうかは検証されません。例外処理は堅牢なコードの一部ではあるが、テストの必要性をなくすものではない。

参照：

Apexのテスト

単体テストのベストプラクティス

オプションB：開発者コンソールの「匿名実行」ブロックでコードを実行します。

効果が低い。

匿名ブロック

オプションC：セーブポイントとDatabase.rollback()を含める。

不十分です。

セーブポイントとロールバックの使用

オプションD：Apexクラスにtry/catchブロックを追加する。

不十分です。

Apexにおける例外処理

結論：

テストクラスを作成することは、Apexクラスが仕様どおりに動作することを保証する最良の方法です。

テストクラスを使用すると、アサーションを用いた包括的なテストが可能になり、コードが要件を満たしているという確信が得られます。

質問: 78

開発者は次のようなコードを書きます。

```
List<Account> acc = [SELECT Id FROM Account LIMIT 10];
Delete acc;
Database.emptyRecycleBin(acc);

System.Debug(Limits.getDMLStatements() + ', ' + Limits.getLimitDMLStatements());
```

デバッグ文を実行した結果はどうなりますか？

A. 2,150

B. 1,150

C. 2,200

D. 1, 100

正解: ([正解を表示します](#))

質問: 79

開発者はSalesforceプラットフォーム上で新しいアプリケーションを設計しており、それが複数のテナントを効果的にサポートできることを確認したいと考えています。

開発者は、拡張性と保守性を確保するために、どの設計フレームワークを検討すべきでしょうか？

A. アジャイル開発

B. モデル・ビュー・コントローラー (MVC)

C. フラックス (ビュー、アクション、ディスパッチャー、ストア)

#### D. 滝モデル

正解: ([正解を表示します](#))

Salesforceプラットフォーム上で、複数のテナント (つまり、複数の顧客またはユーザー) を効果的にサポートする必要がある新しいアプリケーションを設計する場合、開発者はモデル・ビュー・コントローラー (MVC) フレームワークを検討する必要があります。

オプションB :モデル・ビュー・コントローラー (MVC)

正しい設計フレームワーク。

MVCは、ユーザーインターフェースを実装するためのソフトウェアアーキテクチャパターンです。

モデル :データとビジネスロジックを表します。

ビュー :プレゼンテーション層を表します。

コントローラー :ユーザーの入力と操作を処理します。

SalesforceのプラットフォームはMVCパターンを採用しており、関心の分離、拡張性、保守性の確保に役立っています。

アジャイルはプロジェクト管理のための手法であり、設計フレームワークではありません。

これは、拡張性や保守性については直接的に対処していません。

オプションC :Flux (ビュー、アクション、ディスパッチャー、ストア)

Salesforceではあまり使用されていません。

Fluxは、FacebookがクライアントサイドのWebアプリケーションを構築するために使用しているアプリケーションアーキテクチャです。

これはSalesforce開発における標準的な設計パターンではありません。

オプションD :ウォーターフォールモデル

開発手法であって、設計フレームワークではない。

ウォーターフォールは、ソフトウェア開発における直線的かつ順次的なアプローチである。

これは、マルチテナント環境における拡張性や保守性の向上に特に役立つものではありません。

結論 :

開発者は、オプションBであるモデル・ビュー・コントローラー (MVC) フレームワークを検討すべきです。

参照 :

Salesforce MVCアーキテクチャ

Salesforceアーキテクチャの理解

誤った選択肢 :

オプションA :アジャイル開発

開発手法であって、設計フレームワークではない。

質問: 80

```
public static void insertAccounts(List<Account> theseAccounts) {
 for(Account thisAccount : theseAccounts) {
 if(thisAccount.website == null)
 thisAccount.website = 'https://www.demo.com';
 }
 update theseAccounts;
}
```



コードが実行されると、DML例外がスローされます。

例外が適切に処理されるようにするには、開発者はどのようにコードを修正すべきでしょうか？

- A. アップセットDMLステートメントを実装します。
- B. 変更データキャプチャを実装する。
- C. DML に対して try/catch ブロックを実装します。
- D. アカウントのリストからnull項目を削除します。

正解: **C** ([コメントを發表する](#))

\* try/catchブロックが必要な理由：

\* Salesforceでは、挿入、更新、削除、アップサートなどのDML操作は、検証ルール違反、フィールド制約、共有ルール制限などの問題により例外をスローすることがあります。

\* try/catch ブロックを使用することで、これらの例外が確実に捕捉され、適切に処理されるため、トランザクション全体が失敗するのを防ぐことができます。

\* コードの変更方法：

\* コード内の更新ステートメントは、発生する例外を捕捉して処理するために、try/catch ブロックで囲むことができます。例:

頂点

コードをコピー

```
public static void insertAccounts(List<Account> theseAccounts) {
```

```
 試す {
```

```
 for (Account thisAccount : theseAccounts) {
```

```
 if (thisAccount.website == null) {
```

```
 thisAccount.website = 'https://www.demo.com';
```

```
 }
```

```
 }
```

```
 これらのアカウントを更新する。
```

```
 } catch (DmlException e) {
```

```
 System.debug('DML例外: ' + e.getMessage());
```

```
 }
```

```
}
```

\* 他の選択肢を選ばなかったのはなぜですか？

\* A. upsert DML ステートメントを実装します。

\* upsert は、外部 ID または主キーに基づいてレコードを挿入または更新するために使用されます。例外処理は自動的にには行われません。

\* B. 変更データキャプチャの実装：

\* 変更データキャプチャ (CDC)は、データの変更をリアルタイムで追跡するために使用され、DML例外の処理とは関係ありません。

\* D. アカウントリストからnull項目を削除します。

入力データのクリーニングは良い習慣ではありますが、DML操作中の例外処理の必要性には対応していません。

参考文献：

Apexのエラー処理

ApexにおけるDML操作

質問: 81

Universal Containers には、ユーザーがカスタムオブジェクト Engineering Support c を使用してエンジニアリングチームにサポートをリクエストできるサポートプロセスがあります。

ユーザーは、複数のエンジニアリングサポート\_\_cレコードを単一の商談レコードに関連付けることができる必要があります。

さらに、集計情報

エンジニアリングサポート\_\_cレコードについては、商談レコードに表示される必要があります。

これらの要件をサポートするために、どのリレーションシップフィールドを実装すべきでしょうか？

- A. 商談からエンジニアリング\_サポート\_\_cへのルックアップフィールド
- B. エンジニアリングサポート\_\_c から商談へのマスター/ディテールフィールド
- C. 商談からエンジニアリングサポートへのマスター/ディテールフィールド\_\_c
- D. エンジニアリングサポート\_\_cから商談へのルックアップフィールド

正解: ([正解を表示します](#))

Engineering\_Support\_\_cからOpportunityへのマスター/ディテール関係により、複数の子Engineering\_Support\_\_cレコードを単一の親Opportunityに関連付けることができます。また、ロールアップ集計フィールドを使用して、子レコードのデータを親Opportunityに集約することも可能です。

参考資料 :マスター・ディテール関係

誤った選択肢 :

AとD:ルックアップリレーションシップは、ロールアップ集計フィールドをサポートしていません。

C:このユースケースでは、マスター詳細を逆方向（商談からエンジニアリングサポート\_\_c)に移動することはできません。

#### 質問: 82

開発者は、ユーザーが取引先責任者を編集する際に、カスタムユーザーインターフェイスを提供する必要があります。ユーザーは、Salesforce Classicと Lightning Experienceの両方でそのインターフェイスを使用できる必要があります。

開発者はカスタムユーザーインターフェイスを提供するために何をすべきでしょうか？

- A. Salesforce Classic では Visualforce ページ、Lightning Experience では Lightning ページで、取引先の編集ボタンをオーバーライドします。
- B. Salesforce Classic では Visualforce ページ、Lightning Experience では Lightning コンポーネントを使用して、取引先の編集ボタンをオーバーライドします。
- C. Salesforce Classic の Lightning コンポーネントと Lightning Experience の Lightning コンポーネントを使用して、取引先の編集ボタンをオーバーライドします。
- D. Salesforce Classic では Lightning ページ、Lightning Experience では Visualforce ページを使用して、取引先の編集ボタンをオーバーライドします。

正解: ([正解を表示します](#))

#### 質問: 83

Auraのコンポーネントおよびアプリケーションイベントの処理に関して、ベストプラクティスとなるのはどれとどれですか？

2つの回答を選択してください

- A. コンポーネントイベントではなく、アプリケーションイベントを使用するようにしてください。
- B. コンポーネントバンドル内のイベントロジックを再利用するには、ロジックをヘルパーに配置します。
- C. コンポーネントイベントを使用して、アプリケーションレベルで処理する必要のあるアクションを伝達します。
- D. イベントハンドラで低レベルのイベントを処理し、高レベルのイベントとして再発火します。

正解: ([正解を表示します](#))

\* オプションB :ヘルパー内でイベントロジックを再利用することで、よりクリーンで再利用可能なコードが実現します。

\* オプション D: ハンドラで低レベルのイベントを処理し、それを高レベルのイベントとして再発火することで、より優れた抽象化とモジュール性を実現できます。

**質問: 84**

Batchableインターフェースを実装する際に、Apexクラスのexecute()メソッド内でApexコードが実行される場合、ガバナ制限に関して正しい記述は次のうちどれですか？

2つの回答を選択してください

- A. Apex ガバナ制限は、execute() メソッドの各反復ごとにリセットされます。
- B. トランザクションの非同期性のため、Apex ガバナの制限を超えることはできません。
- C. Apex ガバナ制限は非同期制限レベルを使用します。
- D. Apex クラスのコンストラクタを呼び出す際に、Apex ガバナーの制限は省略されます。

正解: ([正解を表示します](#))

\* オプション A: 各バッチは個別のトランザクションとして扱われるため、バッチ ジョブの execute() メソッドが実行されるたびにガバナー制限がリセットされます。

\* オプション C: バッチ ジョブは非同期ガバナ制限を使用します。これは同期制限よりも高い値です。

:Apex ガバナー制限に関するドキュメント

**質問: 85**

開発者は、発行される SOQL クエリ数がプラットフォーム ガバナーの制限を超えないように、次のコード スニペットを変更する必要があります。

```
public class without sharing OpportunityService(public static List<OpportunityLineItem> getOpportunityProducts(Set<Id> opportunityIds)
{ List<OpportunityLineItem> oppLineItems = new List<OpportunityLineItem>(); for(Id thisOppld : opportunityIds){ oppLineItems.addAll([Select Id FROM
OpportunityLineItems WHERE OpportunityId = :thisOppld]); } return oppLineItems; } }
```

上記のメソッドは、Lightning コンポーネントを介してトリガーの実行中に呼び出される可能性があります。ガバナーの制限に達しないようにするには、どの手法を実装する必要がありますか？

- A. System.Limits.getQueries() メソッドを使用して、クエリ数が 100 未満であることを確認します。
- B. 上記のコードを修正して、opportunityIds のセットに 100 未満の ID が含まれている場合にのみ SOQL クエリを実行するようにします。
- C. System.Limits.getLimitQueries() メソッドを使用して、クエリ数が 100 未満であることを確認します。
- D. 上記のコードをリファクタリングして、機会IDのセットでフィルタリングするSOQLクエリを1つだけ実行します。

正解: ([正解を表示します](#))

**質問: 86**

開発者は、システムに存在する ContactSearch Apex クラスのセキュリティレビューを実施するよう指示されました。クラス内で、開発者は次のメソッドをセキュリティ上の脅威として特定しました。List<Contact> performSearch(String lastName){ return Database.query('Select Id, FirstName, LastName FROM Contact WHERE LastName Like %'+lastName+'%'); } 開発者は、SOQL インジェクション攻撃を防ぐために、このメソッドをどのように更新できますか？ 2つの回答を選択してください。

- A. クラスに @ReadOnly アノテーションと with sharing キーワードを使用します。
- B. パラメータを使用する前に、escapeSingleQuote メソッドを使用してパラメータをサニタイズします。
- C. 変数バインディングを使用して、動的クエリを静的な SOQL に置き換えます。
- D. パラメータに対して正規表現を使用して特殊文字を削除します。

正解: ([正解を表示します](#))

**質問: 87**

多数の開発者からなるチームは、それぞれが本番環境の組織と同じ構成を持つ個別の組織で作業を行います。  
このシナリオに最適な組織形態はどれですか？

- A. フルサンドボックス
- B. 開発者版
- C. パートナー開発者版
- D. 開発者サンドボックス

正解: **A** ([コメントを发表する](#))

**質問: 88**

開発者チームは、さまざまな組織構成で独立して作業できるソースコード駆動型のプロジェクトに取り組んでいます。  
開発にはどのタイプのSalesforce組織を使用すべきでしょうか？

- A. 完全コピーサンドボックス
- B. 開発者組織
- C. 開発者サンドボックス
- D. Scratch組織

正解: **D** ([コメントを发表する](#))

スクラッチ組織は、迅速に作成でき、さまざまな機能や設定を構成できる使い捨てのSalesforce組織です。ソースコード主導型開発に最適で、開発者が独立して作業を進めることができます。

オプションD :スクラッチ組織

正解です。

スクラッチ組織はソースコード駆動型開発をサポートしており、スクリプトとSalesforce CLIを使用して完全に構成できます。

開発者は必要に応じてスクラッチ組織を作成および削除できるため、隔離された環境を確保できます。

参照：

該当しないオプション：

オプションA：完全コピーサンドボックス

ステージングやパフォーマンステストには適していますが、個々の開発者環境には適していません。

オプションBとC：開発者組織と開発者サンドボックス

ソースコード主導型開発には柔軟性が低い。

開発者サンドボックスは、スクラッチ組織ほど簡単に再構成できるものではありません。

結論：

独立した構成によるソース駆動型開発の場合、チームはScratch組織を使用する必要があります。これはオプションDです。

**質問: 89**

Lightning Componentフレームワークを使えば、どのようなものが簡単に開発できるのでしょうか？

- A. Salesforce Classicのユーザーインターフェースページ
- B. ライトニングページ
- C. カスタマイズ可能なJavaScriptボタン
- D. Salesforceとの連携

正解: **B** ([コメントを发表する](#))

質問: 90

開発者は、Visualforceウィザードと連携して複数のsObjectの作成と編集をサポートする単一のカスタムコントローラクラスを持っています。このウィザードは、複数のVisualforceページからのユーザー入力と、初期URLのパラメータからデータを受け取ります。

カスタムコントローラを効果的にテストするために、単体テスト内で役立つ3つのステートメントはどれですか？

3つの回答を選択してください

- A. `String nextPage = controller.save().getUrl();`
- B. `ApexPages.currentPage().getParameters().put('input', 'TeatValue');`
- C. ページを挿入します。
- D. `public Extendedcontroller (ApexPages.Standardcontroller entrl) {}`
- E. `Test.setCurrentPage(pageRef);`

正解: **A,B,D** ([コメントを发表する](#))

A: カスタムコントローラの `save()` メソッドは、意図どおりに動作し、正しいページにリダイレクトされることを確認するためにテストする必要があります。

B: `ApexPages.currentPage().getParameters()` を介してパラメータを設定することは、ユーザー入力や URL パラメータがロジックを制御するシナリオをテストするために不可欠です。

D: 単体テストでページコンテキストをシミュレートするには、`Test.setCurrentPage(pageRef)` の使用が不可欠です。

他の選択肢はどうでしょうか？

C: この記述はカスタムコントローラのテストには該当しません。

B 2回目の出現) : 質問文書式が間違っているようです。

Apexコントローラのテスト

質問: 91

継続的インテグレーションの一環として、開発者はデプロイメントと単体テストの実行をスクリプト化するために何を使用すべきでしょうか？

- A. 開発者コンソール
- B. Salesforce CLI
- C. 匿名実行
- D. VS Code

正解: **A** ([コメントを发表する](#))

有効的なPDI問題集はJPNTTest.com提供され、PDI試験に合格することに役に立ちます！JPNTTest.comは今最新PDI試験問題集を提供します。JPNTTest.com PDI試験問題集はもう更新されました。ここでPDI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/PDI-mondaishu> 205問、30%ディスカウント、特別な割引コード: **JPNshiken**」

質問: 92

Salesforceへのデータのインポートおよびエクスポートに関して、正しい記述は次のうちどれですか？

2つの回答を選択してください

- A. バルクAPLを使用すると、ストレージ制限を回避することなく、開発環境で大量のデータをインポートできます。

- B. バルクAPIを使用すると、開発環境で大量のデータをインポートする際にストレージ制限を回避できます。
- C. データインポートウィザードは、お使いのコンピュータにインストールされるアプリケーションです。
- D. 開発者用サンドボックスと開発者プロ用サンドボックスには、ストレージ制限が異なります。

正解: ([正解を表示します](#))

#### 質問: 93

以下のコードを実行すると、どのような結果になりますか？

```
Account a = new Account();
```

```
Database.insert(a, false);
```

- A. レコードは作成されず、例外がスローされます。
- B. 記録が作成され、エラーは報告されません。
- C. レコードが作成され、デバッグログにメッセージが記録されます。
- D. 記録は作成されず、エラーも報告されません。

正解: ([正解を表示します](#))

#### 質問: 94

Visualforceページやコントローラ内のどのコードにセキュリティ上の脆弱性が存在する可能性がありますか？

```
<apex:outputText value="{!$CurrentPage.parameters.userInput}" />
<apex:outputText escape="false" value="{!$CurrentPage.parameters.userInput}" />
<apex:outputField value="{!ctrl.userInput}" />
<apex:outputField value="{!ctrl.userInput}" rendered="{!isEditable}" />
```

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

正解: ([正解を表示します](#))

オプションAには、ユーザー入力を不適切に処理するコードが含まれており、SOQLインジェクションなどのセキュリティ脆弱性につながる可能性があります。これを回避するには、クエリを作成する際に必ずバインド変数を使用してください。

#### 質問: 95

匿名実行ブロック内で実行される以下のコードを参照してください。

```
for(List<Lead> theseLeads : [SELECT LastName, Company, Email FROM Lead LIMIT 20000]){
 for(Lead thisLead : theseLeads){
 if(thisLead.Email == null)
 thisLead.Email = assignGenericEmail(thisLead.LastName, thisLead.Company);
 }
 Database.Update(theseLeads, false);
}
```

結果セット全体が返される環境において、このコードを実行するとどのような結果が考えられますか？

- A. DMLステートメントの総数が上限を超えます。
- B. DMLステートメントの結果として処理されたレコードの総数が超過します。
- C. トランザクションは成功し、最初の1万件のレコードがデータベースにコミットされます。

D. DMLステートメントの結果として処理されたレコードの総数が上限を超えます

正解: [B \(コメントを公表する\)](#)

質問: 96

捕捉できない例外の種類はどれですか？

- A. カスタム例外
- B. LinkException
- C. NoAccessException
- D. CelloutException

正解: [\(正解を表示します\)](#)

\* LimitException は、ガバナーの制限を超えた場合にスローされるため、try-catch ブロックでキャッチすることはできません。Salesforce は、このような場合にそれ以上の処理を許可しません。

\* カスタム例外、NoAccessException、CalloutException などのその他の例外も捕捉して処理できます。

Apexにおける例外

質問: 97

Lightning Web Componentsでは、検索語を指定してアカウントのリストを返すApexメソッドgetAccountsが利用可能です。getAccounts メソッドを使用する Lightning Web コンポーネントのプロパティの正しい定義は何ですか？

```
@wire(getAccounts, '$searchTerm')
```

A. `accountList;`

```
@AuraEnabled(getAccounts, { searchTerm: '$searchTerm' })
```

B. `accountList;`

```
@AuraEnabled(getAccounts, '$searchTerm')
```

C. `accountList;`

```
@wire(getAccounts, { searchTerm: '$searchTerm' })
```

```
accountList;
```

D.

正解: [D \(コメントを公表する\)](#)

質問: 98

ユニバーサル コンテナズ (UC)は、営業チーム向けに、新規顧客を獲得するたびにすべての営業担当者が一連の手順を踏むことを義務付けるプロセスを開発している。

情報収集の最初のステップとして、UCのERPシステムをRESTエンドポイント経由で確認し、顧客が存在するかどうかを調べます。顧客が存在する場合は、そのデータをSalesforce上で営業担当者に提示する必要があります。

開発者は要件を満たすために、どの2つを実装すべきでしょうか？

2つの回答を選択してください

A. 流れ

B. 呼び出し可能なメソッド

C. 未来法

D. トリガー

正解: [\(正解を表示します\)](#)

\* A: フローは、営業担当者を一連のスク립ト化された手順に沿って誘導し、Apexアクションを呼び出して外部のRESTエンドポイントとやり取りすることができます。

\* B: 呼び出し可能なメソッドは、Apex メソッドをフローから呼び出し可能にするために使用されます。このメソッドは REST エンドポイントを呼び出し、表示用のデータを返すことができます。

他の選択肢はどうでしょうか？

\* C: Future メソッドは非同期であり、リアルタイムでユーザーにデータを返すことはできません。

\* D: トリガーはデータベースイベントに使用されるものであり、スク립ト化されたプロセスを通じてユーザーを誘導したり、REST コールを実行したりするために使用されるものではありません。

:

フローにおける呼び出し可能なメソッド

質問: 99

Auraアプリケーションのイベントは従来のパブリッシュ/サブスクライブモデルに従うため、イベントを発生させるにはどの方法が使用されますか？

A. fire()

B. SegdetesEvent {}

C. FireEvent()

D. emit()

正解: [A \(コメントを公表する\)](#)

\* オプション A: fire() メソッドは、従来のパブリッシュ/サブスクライブ モデルで Aura アプリケーション イベントを発行するために使用されます。

\* 不適切:

\* オプション B: sendEvent() は Aura イベントの有効なメソッドではありません。

\* オプション C :FireEvent()は存在しません。

\* オプション D: emit() は Aura イベント フレームワークの一部ではありません。

質問: 100

開発者は、外部のWebアプリケーションで使用されるカスタムSOAP Webサービスを実装する必要があります。開発者は、Webサービスクラスの実装に、Webアプリケーションでは使用されないヘルパーメソッドを含めることを選択しました。

どのコードセグメントが、クラスとメソッドの正しい宣言を示していますか？

```
global class WebServiceClass {
 private static Boolean helperMethod() { /* implementation ... */ }
 global String updateRecords() { /* implementation ... */ }
}
```

A. }

```
webservice class WebServiceClass {
 private static Boolean helperMethod() { /* implementation ... */ }
 global static String updateRecords() { /* implementation ... */ }
}
```

B. }

```
}
global class WebserviceClass {
 private static Boolean helperMethod() { /* implementation ... */ }
```

C.

```
webservice class WebserviceClass {
 private static Boolean helperMethod() { /* implementation ... */ }
 webservice static String updateRecords() { /* implementation ... */ }
```

D. }

正解: ([正解を表示します](#))

質問: 101

開発者は、アカウントオブジェクトに「更新後」トリガーを使用して、アカウントに関連付けられているすべての連絡先を更新します。以下のトリガーコードがランダムに失敗します。

```
List<Contacts> theContacts = new List<Contacts>();
for(Account a : Trigger.new){
 for(Contact c : [SELECT Id, Account_Date__c FROM Contact
WHERE AccountId = :a.Id]){
 c.Account_Date__c = Date.today();
 theContacts.add(c);
 }
}
```

}連絡先を更新します。

コードブロックのエラーの原因となっているコード行はどれですか？

- A. Account\_Date\_\_c が null の場合、例外がスローされます。
- B. トリガーは for ループで 200 を超えるレコードを処理します。
- C. Contactsが空の場合、例外がスローされます
- D. SOQLクエリがforループの中にあります。

正解: ([正解を表示します](#))

質問: 102

コードではなく宣言型カスタマイズを使用する2つの利点は何ですか？ 2つの回答を選択してください。コードではなく宣言型カスタマイズを使用する2つの利点は何ですか？

- A. 宣言的なカスタマイズにより、テストクラスが自動的に生成されます。
- 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、Salesforce の各リリースで自動的に更新されます。
- D. 宣言的なカスタマイズにより、テストクラスが自動的に生成されます。

正解: **A,C** ([コメントを公表する](#))

質問: 103

捕捉できない例外の種類はどれですか？

- A. カスタム例外
- B. LinkException
- C. NoAccessException
- D. CelloutException

正解: ([正解を表示します](#))

\* LimitException はガバナー制限を超えた場合にスローされるため、try-catch ブロックでキャッチすることはできません。Salesforce は、このような場合にそれ以上の処理を許可しません。

\* カスタム例外、NoAccessException、CalloutExceptionなどのその他の例外も捕捉して処理できます。

#### 質問: 104

以下のコードブロックによって、いくつかのアカウントが挿入されますか？

```
for(Integer i = 0 ; i < 500; i++){
 Account a = new Account (Name='New Account ' + i);
 insert a;
}
```

A. 100

B. 0

C. 150

D. 500

正解: **D** ([コメントを發表する](#))

実行フロー :

ループは500回実行されます (0から499までの範囲)。

各反復処理において、新しいアカウントレコードが作成され、挿入されます。

ガバナー制限:

このコードは、各挿入処理が個別に実行されるため、トランザクションあたりのDMLステートメント数を150に制限するというガバナーの制限に違反しています。

しかし、このコードは論理的には500個のアカウントを挿入しようとします。

重要な考慮事項 :

このコードはDML操作の処理が非効率的であり、実際の運用環境では失敗するだろう。

参照: ガバナー制限: [https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_gov\\_limits.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_gov_limits.htm)

#### 質問: 105

Salesforce ClassicまたはLightning Experienceのいずれにおいても、ユーザーが取引先責任者を編集する際には、開発者はカスタムユーザーインターフェイスを提供する必要があります。

開発者は、連絡先の編集ボタンをオーバーライドしてこの機能を提供するには、何を使用すればよいでしょうか？

A. Salesforce Classic の Lightning コンポーネントと Lightning Experience の Lightning コンポーネント

B. Salesforce Classic の Visualforce ページと Lightning Experience の Lightning コンポーネント

C. Salesforce Classic の Visualforce ページと Lightning Experience の Lightning ページ

D. Salesforce Classic の Lightning ページと Lightning Experience の Visualforce ページ

正解: **B** ([コメントを發表する](#))

#### 質問: 106

開発者が、売上記録の一覧を表示するLightning Webコンポーネントを作成しています。

営業担当者ユーザーは、各レコードの手数料欄を表示できる必要があります。営業アシスタントユーザーは、手数料欄を除くすべてのレコード欄を表示できる必要があります。

コンポーネントが両方のユーザーに対してエラーを表示せずに動作するようにするには、どのようにすればこれを強制できるでしょうか？

- A. コンポーネントのデータを取得する SOQL で WITH SECURITY\_ENFORCED を使用します。
- B. Security.stripInaccessible Le を使用して、現在のユーザーがアクセスできないフィールドを削除します。
- C. Lightning Locker Serviceを使用して、共有ルールとフィールドレベルのセキュリティを適用します。
- D. Lightning Data Serviceを使用して販売記録のコレクションを取得します。

正解: ([正解を表示します](#))

Security.stripInaccessible を使用すると、フィールドレベルセキュリティ (FLS) に準拠し、ユーザーが表示権限のないフィールドにアクセスできないようにすることができます。このシナリオでは、次のようになります。

- \* 販売アシスタントは手数料欄にアクセスできません。Security.stripInaccessible を使用すると、結果セットからこの欄が自動的に削除されます。
- \* 適切なアクセス権限を持たないユーザーが制限されたフィールドを取得しようとした際に発生する実行時エラーを防ぎます。

有効的なPDI問題集はJPNTTest.com提供され、PDI試験に合格することに役に立ちます！JPNTTest.comは今最新PDI試験問題集を提供します。JPNTTest.com PDI試験問題集はもう更新されました。ここでPDI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/PDI-mondaishu> 205問、30%ディスカウント、特別な割引コード: **JPNshiken**」

質問: 107

開発者が作成したこのApexトリガーは、MyClass、myStartmethodを呼び出します。

```
trigger myTrigger on Contact(before insert)
{
 MyClass.myStaticMethod(trigger,new);
}
```

開発者は、MyClass,myStartmethodを直接呼び出すテストメソッドを作成し、結果として全体のコードカバレッジが81%になった。

開発者が...と2つのクラスを本番環境にデプロイしようとする、他にコードが存在しないと仮定した場合、何が起こりますか？

- A. Apr-MgQM にコードカバレッジがないため、デプロイが失敗します。
- B. デプロイメントは、Apexコードが要求される75%のコードカバレッジを達成したため合格しました。
- C. クラスとトリガーの両方がデプロイメントに含まれていたため、デプロイメントは成功しました。
- D. lest メソッドでアサーションが行われなかったため、デプロイメントが終了します。

正解: ([正解を表示します](#))

質問: 108

ISV (独立系ソフトウェアベンダー)でありSalesforceパートナーでもあるCloud Kicks Fitnessは、マネージドパッケージアプリケーションを開発しています。このアプリケーションのモジュールの一つでは、ApexクラスBodyfatとそのメソッドcalculateBodyFat()を使用して体脂肪を計算できます。プロダクトオーナーは、ISVのパッケージネームスペース外でカスタマイズを開発する際に、アプリケーションの利用者がこのメソッドにアクセスできるようにしたいと考えています。

開発者は、calculateBodyFat() がパッケージ名前空間外からアクセスできるようにするには、どのようなアプローチを取るべきでしょうか？

- A. グローバルアクセス修飾子を使用してクラスとメソッドを宣言します。
- B. public アクセス修飾子を使用してクラスとメソッドを宣言します。
- C. クラスを public として宣言し、メソッドにグローバル アクセス修飾子を使用します。

D. クラスをグローバルとして宣言し、メソッドにパブリックアクセス修飾子を使用します。

正解: ([正解を表示します](#))

\* グローバルアクセス修飾子:

\* 管理パッケージを構築する際、パッケージの名前空間外からアクセスする必要があるメソッドとクラスは、グローバルとして宣言する必要があります。

\* 公開設定では、名前空間内でのアクセスが制限されます。

\* 他の選択肢はどうでしょうか？

\* B: パブリックは名前空間内でのアクセスを制限します。

\* CとD: パブリックとグローバルを誤って混在させると、外部アクセスの要件を満たしません。

\* 管理パッケージとグローバルアクセス: [https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_classes\\_access\\_modifiers.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_classes_access_modifiers.htm)

参考文献 :

質問: 109

開発者がVisualforceページをLightning Webコンポーネントに移行している。

Visualforceページには、単一のレコードに関する情報が表示されます。開発者は、レコードデータにアクセスするためにLightning Data Serviceを使用することにしました。

開発者はどのようなセキュリティ上の考慮事項に留意すべきでしょうか？

A. Lightning Data Service はフィールドレベルのセキュリティを無視します。

B. 共有ルールを適用するには、with sharing キーワードを使用する必要があります。

C. Lightning Data Service は、共有ルールとフィールドレベルのセキュリティを処理します。

D. フィールドレベルのアクセスチェックには、isAccessible() メソッドを使用する必要があります。

正解: ([正解を表示します](#))

\* Lightning Data Serviceを選ぶ理由とは？

\* Salesforceのセキュリティモデルを適用します。これには以下が含まれます。

\* 共有に関するルール。

\* フィールドレベルセキュリティ (FLS)。

\* これにより、追加の手動セキュリティチェックが不要になります。

\* 他の選択肢はどうでしょうか？

\* A: 間違いです。LDSはFLSを強制します。

\* B: 共有機能はApexに適用され、Lightning Data Serviceには適用されません。

\* D: LDSでは既にFLSが強制されているため、isAccessible()は不要です。

\* Lightning Data Service: [https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/data\\_service.htm](https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/data_service.htm)

参考文献 :

質問: 110

Universal Containers は、新しい Salesforce アプリケーションの構築に宣言型開発のみを使用することにしました。アプリケーションのデータベース層を構築するために使用すべき3つのオプションはどれですか？3つ選択してください。

A. カスタムオブジェクトとフィールド

- B. プロセスビルダー
- C. ロールアップサマリー
- D. トリガー
- E. 人間関係

正解: **B,C,E** ([コメントを發表する](#))

質問: 111

開発者が連絡先リストを更新するApexメソッドを作成し、それをLightning Webコンポーネントで使用できるようにしたいと考えています。これを実現するために、開発者はApexメソッドにどの注釈を追加すればよいでしょうか？

- A. `@AuraEnabled(cacheable=true)`
- B. `@AuraEnabled`
- C. `@RemoteAction(cacheable=true)`
- D. `@RemoteAction`

正解: **B** ([コメントを發表する](#))

質問: 112

Lightning Experience 7 のアカウントの標準編集ボタンをオーバーライドするには何をえばよいですか？

- A. 電光石火のアクション
- B. ライトニングページ
- C. ライトニングコンポーネント
- D. 稲妻の流れ

正解: **C** ([コメントを發表する](#))

質問: 113

新機能開発の一環として、開発者は、ステートフルなクライアント上で実行される、タッチイベントに応答できるレスポンシブなアプリケーションを構築するよう求められます。

ビジネス要件を完全にサポートするフレームワークに基づいて構築されている技術は、どれとどれですか？

2つの回答を選択してください

- A. オーラコンポーネント
- B. ビジュアルフォースページ
- C. Lightning Web Components
- D. Visualforceコンポーネント

正解: ([正解を表示します](#))

タッチイベントに응答し、ステートフルなクライアント上で実行されるレスポンシブなアプリケーションを構築するには、開発者は最新のWeb標準、レスポンシブデザイン、およびタッチ操作をサポートするテクノロジーを使用する必要があります。これらの要件を満たす2つのテクノロジーは、Aura ComponentsとLightning Web Componentsです。

オプションA :オーラコンポーネント

参照 :

Aura Componentsのプログラミングモデルは、モバイルデバイスおよびデスクトップデバイス向けのレスポンシブアプリケーションの開発をサポートするように設計されています。」

- Lightning Aura コンポーネント開発者ガイド

オプションC :Lightning Web Components

Lightning Webコンポーネントは、HTMLと最新のJavaScriptを使用して構築されたカスタムHTML要素です。」

Lightning WebコンポーネントはWeb標準に基づいているため、本質的に効率的で、優れたパフォーマンスを発揮します。」

- Lightning Web Components 開発者ガイド

他の選択肢が間違っている理由：

オプションB :Visualforceページ

Visualforceはサーバーサイドフレームワークであり、レスポンシブデザインやタッチイベントをネイティブに完全にはサポートしていません。カスタムコードを使用すればVisualforceページをレスポンシブにすることは可能ですが、Aura ComponentsやLWCほど効果的にタッチイベントをサポートする機能は備えていません。

Visualforceは、Force.comプラットフォーム向けのコンポーネントベースのユーザーインターフェースフレームワークです。」

- Visualforce開発者ガイド

オプションD :Visualforceコンポーネント

Visualforceコンポーネントは、Visualforceフレームワーク内のカスタムコンポーネントです。レスポンシブデザインとタッチ操作のサポートに関しては、Visualforceページと同様の制限を受け継ぎます。

Visualforceコンポーネントとは、Visualforceフレームワークによって提供される標準コンポーネントか、ユーザーが作成するカスタムコンポーネントのいずれかです。」

- Visualforce開発者ガイド

#### 質問: 114

次のコードスニペットを考えてみましょう。

Salesforceプラットフォームのマルチテナントアーキテクチャを考慮すると、開発者がメソッドの成功的な実行を確実にするために実装すべきベストプラクティスは何でしょうか？

A. クエリフィルタとして変数を使用することは避けてください。

B. 空のレコードリストを返さないようにする。

C. forループ内でクエリを実行することは避けてください。

D. 制限句のないクエリの実行は避けてください。

正解: ([正解を表示します](#))

ループ内でクエリを実行すると、単一のトランザクション内でSOQLクエリが多すぎるためにガバナー制限例外が発生する可能性があります。代わりに、クエリはループの外で実行してください。

ガバナー制限を回避するためのベストプラクティス

#### 質問: 115

開発者は、Visualforceウィザードと連携して複数のsObjectの作成と編集をサポートする単一のカスタムコントローラクラスを持っています。このウィザードは、複数のVisualforceページからのユーザー入力と、初期URLのパラメータからデータを受け取ります。

カスタムコントローラを効果的にテストするために、単体テスト内で役立つ3つのステートメントはどれですか？

3つの回答を選択してください

- A. String nextPage = controller.save().getUrl();
- B. ページを挿入します。
- C. ApexPages.currentPage().getParameters().put('input', 'TestValue');
- D. Test.setCurrentPage(pageRef);
- E. public Extendedcontroller (ApexPages.Standardcontroller entrl) { }

正解: A,C,D ([コメントを发表する](#))

質問: 116

下記のコンポーネントコード9fdの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>

```

要件

- \* モバイル端末の場合、情報は3行で表示されます。
- \* 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ていません。

デスクトップ端末とタブレット端末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？

```

<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>

```

A. </lightning:layout>

```

<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 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.

正解: [\(正解を表示します\)](#)

質問: 117

非同期Apexトランザクションにおいて、ガバナ制限値が高いトランザクションごとの制限値はどれですか？ 3つ挙げてください

- A. SOQLクエリの合計数
- B. 最大実行時間
- C. SOQLによって返されたレコード
- D. 最大ヒープサイズ
- E. 最大CPU時間

正解: [A,C,E \(コメントを發表する\)](#)

質問: 118

ユニバーサルコンテナ (UC)は、配送プロセスを効率化しつつ、配送コストを削減したいと考えています。配送担当者は、複数のアカウントでデフォルトの集荷先住所を共有できるように、グローバルアドレスを導入することをUCに提案しました。開発者は、このビジネス要件をサポートするオブジェクトとリレーションシップを作成するよう指示され、設定メニューを使用して「グローバルアドレス」というカスタムオブジェクトを作成しました。開発者は、ビジネスニーズをサポートする最も効率的なモデルを作成するために、どのフィールドを追加すべきでしょうか？

- A. グローバルアドレスオブジェクトにマスター/ディテールフィールドを追加して、アカウントオブジェクトに追加します。
- B. グローバルアドレスオブジェクトに、アカウントオブジェクトへのルックアップフィールドを追加します。
- C. アカウントオブジェクトにグローバルアドレスオブジェクトへのルックアップフィールドを追加します。
- D. アカウントオブジェクトのマスター/ディテールフィールドをグローバルアドレスオブジェクトに追加する

正解: [\(正解を表示します\)](#)

質問: 119

開発者は、単体テスト用のテストデータを作成するヘルパーメソッドを作成するよう依頼される。

```
01: public TestUtils {
02:
03: public static Account createAccount() {
04: Account act = new Account();
05: // ...set some fields on acct
06: return act;
07: }
08: //...other methods...
09: }
```

Testutils クラスのメソッドが単体テストのメソッドでのみ使用できるようにするには、Testutils クラスのどこを変更すればよいでしょうか？

- A. 3行目から静電気を除去します。
- B. 3行目の上に@IsTestを追加してください。
- C. 01行目の上に@IsTestを追加します。
- D. 1行目のpublicをprivateに変更します。

正解: ([正解を表示します](#))

質問: 120

Salesforceでレコードが保存される際の処理順序はどうなっていますか？

- A. ワークフロー、トリガー、プロセスフロー、コミット
- B. ワークフロー、プロセスフロー、トリガー、コミット
- C. プロセスフロー、トリガー、ワークフロー、コミット
- D. トリガー、ワークフロー、プロセスフロー、コミット

正解: ([正解を表示します](#))

質問: 121

開発者が、複数のLightning Webコンポーネントを含むアプリケーションを作成している。

子コンポーネントの1つはナビゲーションに使用されます。ユーザーが子コンポーネント内の「次へ」ボタンをクリックすると、親コンポーネントに通知が届き、次のページへ移動できるようになります。

これはどのように実現すべきでしょうか？

- A. 親のプロパティを更新します。
- B. Apexコントローラ内のメソッドを呼び出します。
- C. 通知を発信する。
- D. カスタムイベントを作成します。

正解: ([正解を表示します](#))

子コンポーネントの「次へ」ボタンがクリックされたときに親コンポーネントに通知するには、子コンポーネントがカスタムイベントを発行します。親コンポーネントはこのイベントをリッスンし、それに応じてナビゲーションを処理します。

参考資料 LWCのカスタムイベント

誤った選択肢：

A: LWCのプロパティは一方向のデータバインドであり、更新しても保護者には通知されません。

B: 親子間の通信にはApexメソッドの呼び出しは不要です。

C: コンポーネント間の通信において、LWCには通知機能は搭載されていません。

有効的なPDI問題集はJPNTTest.com提供され、PDI試験に合格することに役に立ちます！JPNTTest.comは今最新PDI試験問題集を提供します。JPNTTest.com PDI試験問題集はもう更新されました。ここでPDI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/PDI-mondaishu> 205問、30%ディスカウント、特別な割引コード: **JPNshiken**」

質問: 122

Universal Containersは、ユーザーがアカウントに関する情報を入力できるカスタムユーザーインターフェースコンポーネントを作成する必要があります。

コンポーネントは、情報をデータベースに保存する前に、ユーザー入力を検証する必要があります。

この部品を製造するのに最適な技術は何ですか？

A. 流れ

B. Lightning Web Components

C. ビジュアルフォース

D. VUE JavaScriptフレームワーク

正解: (正解を表示します)

Lightning Web Components (LWC) は、Salesforce でカスタム UI コンポーネントを作成するための最適な選択肢です。

これらは、イベント処理やサーバー側通信といった堅牢な機能を提供し、データを保存する前にユーザー入力を検証することを可能にする。

質問: 123

開発者は、自動起動フローをチェンジセットにデプロイする前に、そのテストカバレッジをどのように確認できますか？

A. ApextestResult クラスを使用します。

B. SOQLとTooling APIを使用します。

C. コードカバレッジ設定ページを使用します。

D. フロープロパティページを使用します。

正解: B (コメントを公表する)

質問: 124

Lightning Webコンポーネントのプロパティに接続できるようにするには、開発者はApexメソッドにどの注釈を使用すればよいでしょうか？

A. @AuraEnabled(cacheable=true)

B. @RemoteAction (cacheable=true)

C. @RemoteAction

**D. @AuraEnabled**

正解: ([正解を表示します](#))

@AuraEnabled(cacheable=true)アノテーションを使用すると、Apex メソッドを Lightning Web コンポーネントで使用できるようになり、プロパティに接続できるだけでなく、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ることができます。

参考資料 :Lightning Web コンポーネント用の Apex メソッド

誤った選択肢 :

B、C:@RemoteAction は Lightning Web Components では使用されません。

D:@AuraEnabledwithoutcacheable=trueは配線できません。

**質問: 125**

VisualforceページがApexコントローラーを呼び出し、そのコントローラーが別のApexクラスを呼び出し、その結果ガバナー制限に達した場合、どのような結果になりますか？

- A. すべての変更は最初のApexクラスに保存されます。
- B. エラーが発生するまでの変更はすべて保存されます。
- C. エラーが発生するまでの変更はすべてロールバックされます。
- D. セーブポイントより前の変更はすべて保存されます。

正解: ([正解を表示します](#))

**質問: 126**

Account オブジェクトには、小数点以下 2 桁の数式 (数値) として定義されたカスタム数式フィールド Level\_\_c があります。有効な割り当ては次の 3 つのうちどれですか。3 つ選択してください。

- A. Double myLevel = acct.Level\_\_c;
- B. オブジェクト myLevel = acct.Level\_\_c;
- C. Decimal myLevel = acct.Level\_\_c;
- D. Integer myLevel = acct.Level\_\_c;
- E. Long myLevel = acct.Level\_\_c;

正解: **A,B,C** ([コメントを發表する](#))

**質問: 127**

部分コピーサンドボックスと完全コピーサンドボックスの2つの特徴は何ですか？ 2つ選択してください。

- A. より多くのデータ記録ストレージを提供します
- B. サンドボックステンプレートが必要です
- C. メタデータのサブセットが含まれます
- D. より頻繁な更新に対応

正解: **C,D** ([コメントを發表する](#))

**質問: 128**

Universal Containersは、Accountオブジェクトにプライベート共有モデルを導入しました。営業担当者が指定した複数の条件に一致するアカウントを検索できるように、Apexを使用してカスタムアカウント検索ツールが開発されました。しかし、リリース後、このツールのユーザーから、自分が所有していないアカウントが表示されるという報告が寄せられています。

開発者は、カスタム検索ツールを使用する際に、現在ログインしているユーザーの共有権限を強制するために、どのような方法を用いるべきでしょうか？

- A. クラス宣言で with sharing キーワードを使用します。
- B. クラス宣言で without sharing キーワードを使用します。
- C. UserInfo Apex クラスを使用して、SOQL クエリをフィルタリングし、ログインしているユーザーが所有するレコードを返します。
- D. スキーマ記述呼び出しを使用して、ログインしているユーザーが Account オブジェクトにアクセスできるかどうかを判断します。

正解: [A \(コメントを发表する\)](#)

with sharing キーワードを使用すると、Apex クラスは現在のユーザーの共有ルール (プライベート共有モデルを含む) を尊重するようになります。つまり、ログインしているユーザーは、自分が所有しているレコード、または自分と共有されているレコードのみを表示できます。

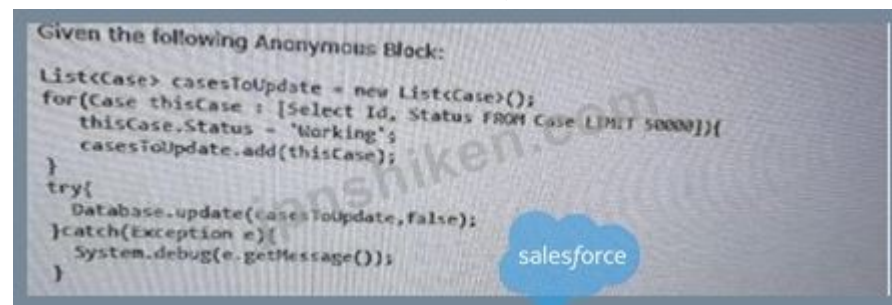
参考資料: Apex クラス共有ルール

誤った選択肢:

- B: 共有なしは共有ルールを無視し、ユーザーがアクセス権限を持たないデータを公開します。
- C: UserInfo を使用したフィルタリングは、拡張性や保守性に優れたソリューションではありません。
- D: Schema describe 呼び出しは共有権限を強制せず、オブジェクトレベルのアクセスのみをチェックします。

質問: 129

以下の匿名ブロックが与えられた場合:



```
Given the following Anonymous Block:
List<Case> casesToUpdate = new List<Case>();
for(Case thisCase : [Select Id, Status FROM Case LIMIT 50000]){
 thisCase.Status = 'Working';
 casesToUpdate.add(thisCase);
}
try{
 Database.update(casesToUpdate, false);
}catch(Exception e){
 System.debug(e.getMessage());
}
```

どちらがお好みですか？

1万件以上のケースレコードが存在する環境において、開発者はどのような点を考慮すべきでしょうか？

- A. ガバナーの制限を超えたため、取引は失敗します。
- B. 取引は成功し、変更がコミットされます。
- C. try/catch ブロックは、スローされた DML 例外を処理します。
- D. try/catch ブロックは、ガバナー制限によってスローされた例外を処理します。

正解: [\(正解を表示します\)](#)

質問: 130

開発者は、カスタムのサードパーティ製 JavaScript フレームワークを使用して構築され、Visualforce ページでレンダリングされる複数のアプリケーションに、Lightning Experience のルックアンドフィールを適用する必要があります。どのオプションがこれを実現しますか？

- A. セットアップメニューでユーザーインターフェイスオプションを構成して、Visualforce のレガシーモードを有効にします。
- B. 定義で属性 enablelightning を "true" に設定します。
- C. Salesforce Lightning Design System の CSS スタイルシートを JavaScript アプリケーションに組み込む
- D. サードパーティの JavaScript ライブラリをネイティブの Visualforce タグに置き換える

正解: [\(正解を表示します\)](#)

質問: 131

開発者は、システムに特定の条件が存在する場合に、Request\_\_c レコードが作成されないようにする必要があります。条件をチェックする RequestLogic クラスが存在します。

正しい実装方法は？

```
trigger RequestTrigger on Request__c (before insert) {
 if (RequestLogic.isValid(Request__c))
 Request.addError('Your request cannot be created at this time.');
```

A. }

```
trigger RequestTrigger on Request__c (before insert) {
 RequestLogic.validateRecords(trigger.new);
```

B. }

```
trigger RequestTrigger on Request__c (after insert) {
 if (RequestLogic.isValid(Request__c))
 Request.addError('Your request cannot be created at this time.');
```

C. }

```
trigger RequestTrigger on Request__c (after insert) {
 RequestLogic.validateRecords(trigger.new);
```

D. }

正解: [\(正解を表示します\)](#)

質問: 132

naze が Visualforce ページの <apex:inputText> タグによって取得された 8 つの文字列であると仮定すると、実行される 2 つの SOQL クエリのうち、SOQL インジェクションから安全なのはどれですか？

2つの回答を選択してください

```
String query = 'SELECT Id FROM Account WHERE Name LIKE \'' + name + '%\'';
```

A. List<Account> results = Database.query(query);

```
String query = 'SELECT Id FROM Account WHERE Name LIKE \'' + String.escapeSingleQuotes(name) + '%\'';
```

B. List<Account> results = Database.query(query);

```
String query = '%' + name + '%';
```

```
List<Account> results = [SELECT Id FROM Account WHERE Name LIKE :query];
```

C.

```
String query = 'SELECT Id FROM Account WHERE Name LIKE \'' + name.noQuotes() + '%\'';
```

D. List<Account> results = Database.query(query);

正解: [\(正解を表示します\)](#)

質問: 133

開発者は、ユーザーが取引先責任者レコード情報を入力して取引先責任者レコードを作成できるようにする Lightning コンポーネントを作成する必要があります。これには、給与 c カスタム フィールドが含まれます。

開発者は、lightning-record-edit-form と併せて何を使用すれば、Salary\_\_c フィールドが通貨入力として機能し、Salary\_c に対する適切なフィールドレベルの権限を持つユーザーのみが表示および編集できるようになるのでしょうか？

```
<lightning-input-currency value="Salary__c">
```

A. </lightning-input-currency>

```
<lightning-formatted-number value="Salary__c" format-style="currency">
```

B. 

```
</lightning-formatted-number>
```

```
<lightning-input-field field-name="Salary__c">
```

C. 

```
</lightning-input-field>
```

```
<lightning-input type="number" value="Salary__c" formatter="currency">
```

D. 

```
</lightning-input>
```

正解: [\(正解を表示します\)](#)

\* この要件では、Salary\_\_c フィールドが通貨入力として機能し、フィールドレベルのセキュリティを尊重する必要があると規定されています。

\* lightning-input-field コンポーネント (オプション C) は、Salesforce で設定されたフィールドレベルの権限を自動的に尊重し、Salary\_\_c フィールドが承認されたユーザーのみによって編集および表示できるようにします。

\* その他のオプションでは、フィールドレベルのセキュリティへの準拠が保証されません。

\* オプション A: <lightning-input-currency> はフィールドレベルのセキュリティを適用しません。

\* オプション B: <lightning-formatted-number> は読み取り専用であり、編集には使用できません。

\* オプション D: <lightning-input> では、セキュリティを適用するためのカスタムロジックが必要です。

参考文献 :

\* Lightning Web Components のドキュメント

\* フィールドレベルのセキュリティに関するベストプラクティス

質問: 134

Visualforceページに外部JavaScriptファイルを読み込む有効な方法はどれですか？ (2つ選択してください)

A. (リンク)\*タグを使用します。

B. (apex:includeScript)\* タグを使用します。

C. (script)\*タグを使用します。

D. (apex:define)\* タグを使用します。

正解: **B,C** ([コメントを發表する](#))

質問: 135

継続的インテグレーションの一環として、開発者はデプロイメントと単体テストの実行をスクリプト化するために何を使用すべきでしょうか？

A. VS Code

B. 匿名実行

C. Salesforce CLI

D. 開発者コンソール

正解: [\(正解を表示します\)](#)

Salesforce CLIを使用すると、開発者は継続的インテグレーション (CI)パイプラインの一環として、デプロイメントの自動化や単体テストの実行を行うことができます。これは、これらのプロセスをスクリプト化するための最も効率的なツールです。

参考資料 :Salesforce CLI

誤った選択肢 :

A: VS CodeはIDEであり、スクリプト作成にはCLIとの統合が必要です。

B:匿名実行は、コードスニペットを手動で実行する場合に使用され、CIには使用されません。

D:開発者コンソールは、手動テストとデバッグのためのものであり、CI自動化のためのものではありません。

以下は、指定された形式と要件に従って作成された、提示された質問に対する回答です。

この回答は、Salesforce Platform Developer I の公式ドキュメントに基づき、データモデリングと管理のトピックに焦点を当てています。回答には包括的な説明、潜在的な誤りの修正、および関連するSalesforceドキュメントへの参照が含まれています。

有効的な**PDI**問題集はJPNTTest.com提供され、**PDI**試験に合格することに役に立ちます！JPNTTest.comは今最新**PDI**試験問題集を提供します。JPNTTest.com **PDI**試験問題集はもう更新されました。ここで**PDI**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/PDI-mondaishu> **205**問、**30%**ディスカウント、特別な割引コード: **JPNshiken**」