

Test Salesforce PDII-JPN Dumps Demo & Valid PDII-JPN Study Materials

EXAMFORSURE

PDII Dumps Demo Questions:

Q1. A developer has requirement to query three fields (id, name, Type) from an Account and first and last names for all Contacts associated with the Account.

. Which option is the preferred optimized method to achieve this for the Account named 'Ozone Electronics'?

A. Account a = (SELECT ID, Name, Type from Account where name= Ozone Electronics;) list 1contacts = (SELECT first name, last name from Contacts where accountid=: a-ID0);

B. Account a = (SELECT ID, Name, Type, (select contat,first Name, Contact,Last Name from Account, Contacts) from Account where name; Ozone Electronic' Limit 1);

C List 1Accounts = (Select ID, Name, Type from Account Join (Select ID, first name, last name form Contact where contact account , name 'ozone electronics));

D List 1Contacts = new list ();

for(Contact c ; 1Select first name, last name Account, Name Account,ID Account, Type from Contact where Account: Name=' electronics')) { iContacts.add(c);

Answer: B

[Download Full Exam Questions Answer](#)

False PDII-JPN practice materials deprive you of valuable possibilities of getting success. As professional model company in this line, success of the PDII-JPN training guide will be a foreseeable outcome. Even some nit-picking customers cannot stop practicing their high quality and accuracy. We are intransigent to the quality issue and you can totally be confident about their proficiency sternly. Choosing our PDII-JPN Exam Questions is equal to choosing success.

The test software used in our products is a perfect match for Windows' PDII-JPN learning material, which enables you to enjoy the best learning style on your computer. Our PDII-JPN study materials also use the latest science and technology to meet the new requirements of authoritative research material network learning. Unlike the traditional way of learning, the great benefit of our PDII-JPN Study Materials are that when the user finishes the exercise, he can get feedback in the fastest time.

>> Test Salesforce PDII-JPN Dumps Demo <<

Valid Salesforce PDII-JPN Study Materials | Examcollection PDII-JPN Dumps Torrent

ExamcollectionPass PDII-JPN latest exam dumps are the reliable and valid study material with latest & guaranteed questions & answers for your preparation. We promise you the easiest way to success and offer you the most prestigious and updated PDII-JPN Exam Training practice which carry 100% money return policy. Come on, and use Salesforce PDII-JPN pdf download torrent, you can pass your PDII-JPN actual test at first attempt.

Salesforce Sample Questions (Q137-Q142):

NEW QUESTION # 137

商談リストという名前の商談レコードのリストがある場合、商談のアカウントのすべての連絡先を照会するの

に最適なコード スニペットはどれですか。

- **A. Java**

```
List<Contact> contactList = new List<Contact>();  
Set<Id> accountIds = new Set<Id>();  
for(Opportunity o : opportunityList){  
    accountIds.add(o.AccountId);  
}  
for(Account a : [SELECT Id, (SELECT Id FROM Contacts) FROM Account WHERE Id IN :  
accountIds]){  
    contactList.addAll(a.Contacts);  
}
```
- **B. 20**
Java

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

Answer: A

Explanation:

22

In Apex, "bulkification" is the practice of ensuring code can handle multiple records efficiently without hitting governor limits. Snippet A demonstrates the correct bulkified approach for this requirement. It first iterates through the opportunityList to collect all unique AccountId values into a Set. Then, it performs a single SOQL query to retrieve all relevant Accounts and their child Contacts using a subquery (Inner Join).

This ensures that the code only consumes one SOQL query regardless of how many opportunities are in the input list.

Snippet B is syntactically incorrect and will fail to compile. In Apex, you cannot use dot-notation (like opportunityList.AccountId) on a List collection to retrieve a set of IDs from its elements. To access the AccountId of records within a list, you must iterate through the list or use a map. Even if corrected to use a proper ID collection, Snippet A is often preferred when you need the relationship context between the Account and its Contacts. Most importantly, Snippet A correctly identifies the need to extract IDs into a separate collection before querying, which is a fundamental requirement for writing scalable Apex. It avoids the "Query in a loop" anti-pattern and adheres to the platform's execution model.

NEW QUESTION # 138

Universal Containersは、未処理の例外が発生した場合に、Apexを使用してカスタムイベントをパブリッシュし、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、CometD を使用して外部システムがイベントをサブスクライブするようにします。**
- **B. addError() メソッドを使用してエラーイベントを公開し、CometD を使用して外部システムがイベントをサブスクライブするようにします。46**
- **C. 外部システムにイベントチャンネルをサブスクライブさせます。パブリッシュは不要です。**
- **D. addError() メソッドを使用してエラー イベントを公開し、イベントをサブスクライブして外部システムに通知するトリガーを記述します。**

Answer: A

Explanation:

48

Platform Events provide a powerful way to integrate Salesforce with external systems using an event-driven architecture. To meet the requirement of notifying an external system about an exception, the developer must first define a Custom Platform Event (e.g., Error_Event__e). In the Apex code, specifically within a catch block, the developer should instantiate this event and publish it using the EventBus.publish() method (Option A).

Once published, the event is placed on the event bus. External systems can listen for these events by subscribing to the event channel. The standard protocol for external clients to subscribe to Salesforce Platform Events is CometD, an implementation of the Bayeux protocol that allows for long-polling and real-time push notifications.

Option B and C are incorrect because addError() is a method used in triggers or Visualforce to display a validation error message to the UI and roll back the transaction; it is not used for publishing Platform Events.

Option D is incorrect because an external system cannot receive an event unless the Salesforce application explicitly publishes it to the bus. Using `EventBus.publish()` with a CometD subscriber provides a robust, decoupled integration pattern for real-time error reporting.

NEW QUESTION # 139

Universal Containers は、カスタム Salesforce アプリケーションを使用してカスタマーサポートケースを管理しています。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協力する必要があります。ただし、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。この要件の達成に役立つ Salesforce 機能はどれですか？

- A. 基準ベースの共有ルール
- B. セットの共有
- C. Apex 管理による共有
- D. 役割階層

Answer: B

Explanation:

The best feature to use to achieve this requirement is sharing sets. Sharing sets allow the developer to grant access to records for external users based on a common account or contact. In this case, the developer can create a sharing set that grants access to the cases for the external partners who have the same account as the support team. Role hierarchy, criteria-based sharing rules, and Apex managed sharing are not suitable for this scenario, as they are not available for external users with Customer Community Plus licenses. Reference: [Sharing Sets], [Sharing Architecture], [Trailhead: Sharing and Visibility Basics]

NEW QUESTION # 140

Visualforce リモート オブジェクトに対する JavaScript リモート処理の利点は何ですか？

- A. Apex コードは必要ありません。
- B. 複雑なサーバー側アプリケーション ロジックをサポート
- C. JavaScript コードは必要ありません。
- D. 指定された再レンダリング ターゲットを許可します。

Answer: B

NEW QUESTION # 141

Ajax 動作を開始して部分的なページ更新を実行するために使用できる 3 つの Visualforce コンポーネントはどれですか。

- A. `<apex:form>`
- B. `<apex:actionStatus>`
- C. `<apex:actionSupport>`
- D. `<apex:commandLink>`
- E. `<apex:コマンドボタン>`

Answer: C,D,E

Explanation:

Comprehensive and Detailed Explanation:

In Visualforce, partial page updates (AJAX) are achieved using the `reRender` attribute. This attribute allows a component to refresh only a specific part of the page identified by an ID, rather than performing a full browser reload.

* `<apex:commandButton>` (Option C) and `<apex:commandLink>` (Option E) are standard action components. When their `reRender` attribute is populated, they perform an asynchronous postback.

* `<apex:actionSupport>` (Option D) is an auxiliary component that adds AJAX functionality to other components that do not natively support it. For example, you can place an `actionSupport` inside an `inputText` to trigger a partial refresh on the `onchange` event.

Option A (`<apex:form>`) is a container and does not initiate AJAX behavior itself. Option B (`<apex:actionStatus>`) is used to display the status of an AJAX request (e.g., a loading spinner) but does not initiate the request.

