



Free Questions for PDI by [braindumpscollection](#)

Shared by [Klein](#) on 09-08-2024

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

A developer has the following requirements:

Calculate the total amount on an Order.

Calculate the line amount for each Line Item based on quantity selected and price.

Move Line Items to a different Order if a Line Item is not stock.

Which relationship implementation supports these requirements?

Options:

- A) Line Items has a Master-Detail field to Order and the Master can be re-parented.
- B) Line Item has a Lookup field to Order and there can be many Line Items per Order
- C) Order has a Lookup field to Line Item and there can be many Line Items per Order.
- D) Order has a Master-Detail field to Line Item and there can be many Line Items per Order.

Answer:

A

Question 2

Question Type: MultipleChoice

Universal Containers (UC) decided it will not to send emails to support personnel directly from Salesforce in the event that an unhandled exception occurs. Instead, UC wants an external system be notified of the error. What is the appropriate publish/subscribe logic to meet these requirements?

Options:

- A) Publish the error event using the `addError()` method and have the external system subscribe to the event using CometD.
- B) Publish the error event using the `Eventbus.publish()` method and have the external system subscribe to the event using CometD.
- C) Have the external system subscribe to the `BatchApexError` event, no publishing is necessary.
- D) Publish the error event using the `addError()` method and write a trigger to subscribe to the event and notify the external system.

Answer:

B

Question 3

Question Type: MultipleChoice

While writing a test class that covers an OpportunityLineItem trigger, a Developer is unable to create a standard PriceBook since one already exists in the org.

How should the Developer overcome this problem?

Options:

- A) Use Test.getStandardPricebookId() to get the standard PriceBook ID.
- B) Use @IsTest(SeeAllData=true) and delete the existing standard PriceBook.
- C) Use Test.loadData() and a Static Resource to load a standard Pricebook.
- D) Use @TestVisible to allow the test method to see the standard PriceBook.

Answer:

A

Question 4

Question Type: MultipleChoice

A developer is asked to create a Visualforce page that displays some Account fields as well as fields configured on the page layout for related Contacts.

How should the developer implement this request?

Options:

- A) Use the `apex:page` tag.
- B) Create a controller extension.
- C) Use the `apex:form` tag.
- D) Add a method to the standard controller.

Answer:

A

Question 5

Question Type: MultipleChoice

Given the following block code: `try{ List <Accounts> retrievedRecords = [SELECT Id FROM Account WHERE Website = null]; }catch(Exception e){ //manage exception logic }` What should a developer do to ensure the code execution is disrupted if the `retrievedRecordslist` remains empty after the SOQL query?

Options:

- A) Check the state of the `retrieveRecords` variable and throw a custom exception if the variable is empty.
- B) Check the state of the `retrievedRecords` variable and use `System.assert(false)` if the variable is empty
- C) Check the state of the `retrievedRecords` variable and access the first element of the list if the variable is empty.
- D) Replace the `retrievedRecords` variable declaration from `ftount` to a single `Account`.

Answer:

B

Question 6

Question Type: MultipleChoice

How should a custom user interface be provided when a user edits an Account in Lightning Experience?

Options:

- A) Override the Account's Edit button with Lightning Flow
- B) Override the Account's Edit button with Lightning Action
- C) Override the Account's Edit button with Lightning page.
- D) Override the Account's Edit button with Lightning component.

Answer:

D

Question 7

Question Type: MultipleChoice

A developer created a Visualforce page and custom controller to display the account type field as shown below. Custom controller code:

```
public class customCtrlr{ private Account theAccount; public String actType; public customCtrlr() { theAccount = [SELECT Id, Type FROM Account WHERE Id = :apexPages.currentPage().getParameters().get('id')]; actType = theAccount.Type; } }
```

Visualforce page snippet: The Account Type is {!actType} The value of the account type field is not being displayed correctly on the page. Assuming the custom controller is property referenced on the Visualforce page, what should the developer do to correct the problem?

Options:

- A) Add a getter method for the actType attribute.
- B) Change theAccount attribute to public.
- C) Convert theAccount.Type to a String.
- D) Add with sharing to the custom controller.

Answer:

A

Explanation:

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

To Get Premium Files for PDI Visit

<https://www.p2pexams.com/products/pdi>

For More Free Questions Visit

<https://www.p2pexams.com/salesforce/pdf/pdi>

