1Z0-151 Q&As

Oracle Fusion Middleware 11g: Build Applications with Oracle Forms

Pass Oracle 1Z0-151 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.passapply.com/1z0-151.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Oracle Official Exam Center

- **Instant Download** After Purchase
- **100% Money Back** Guarantee
- **365 Days** Free Update
- **800,000+** Satisfied Customers

QUESTION 1

You must be careful when coding a When-Button-Pressed trigger, because it does not accept restricted built-ins.

A. True

B. False

Correct Answer: B

The When-Button-Pressed trigger:

*

Fires when the operator clicks a button

*

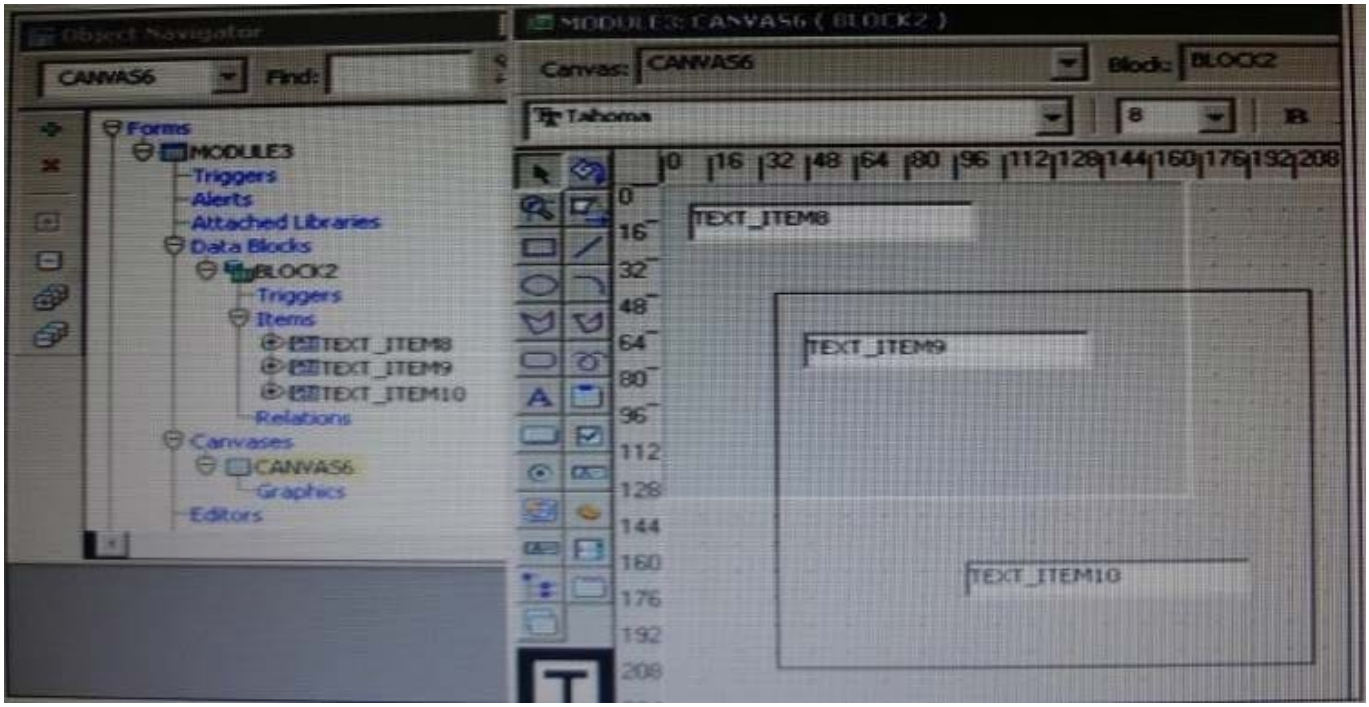
Accepts restricted and unrestricted built-ins

*

Is used to provide convenient navigation, and to display LOVs and many other frequently used functions

QUESTION 2

View the Exhibit.



You have defined the window, canvas, and text items shown in the Exhibit. What happens when click Run Form?

A. The form runs with the cursor initially in TEXT_ITEM8.

B. The form runs with the cursor initially in TEXT_ITEM9.

C. The form does not compile until you move TEXT_ITEM8.

D. The form does not compile until you move TEXT_ITEM10.

Correct Answer: A

QUESTION 3

The Enforce Primary Key property of a data block ensures that the primary key of a new record exists in the database.

A. True

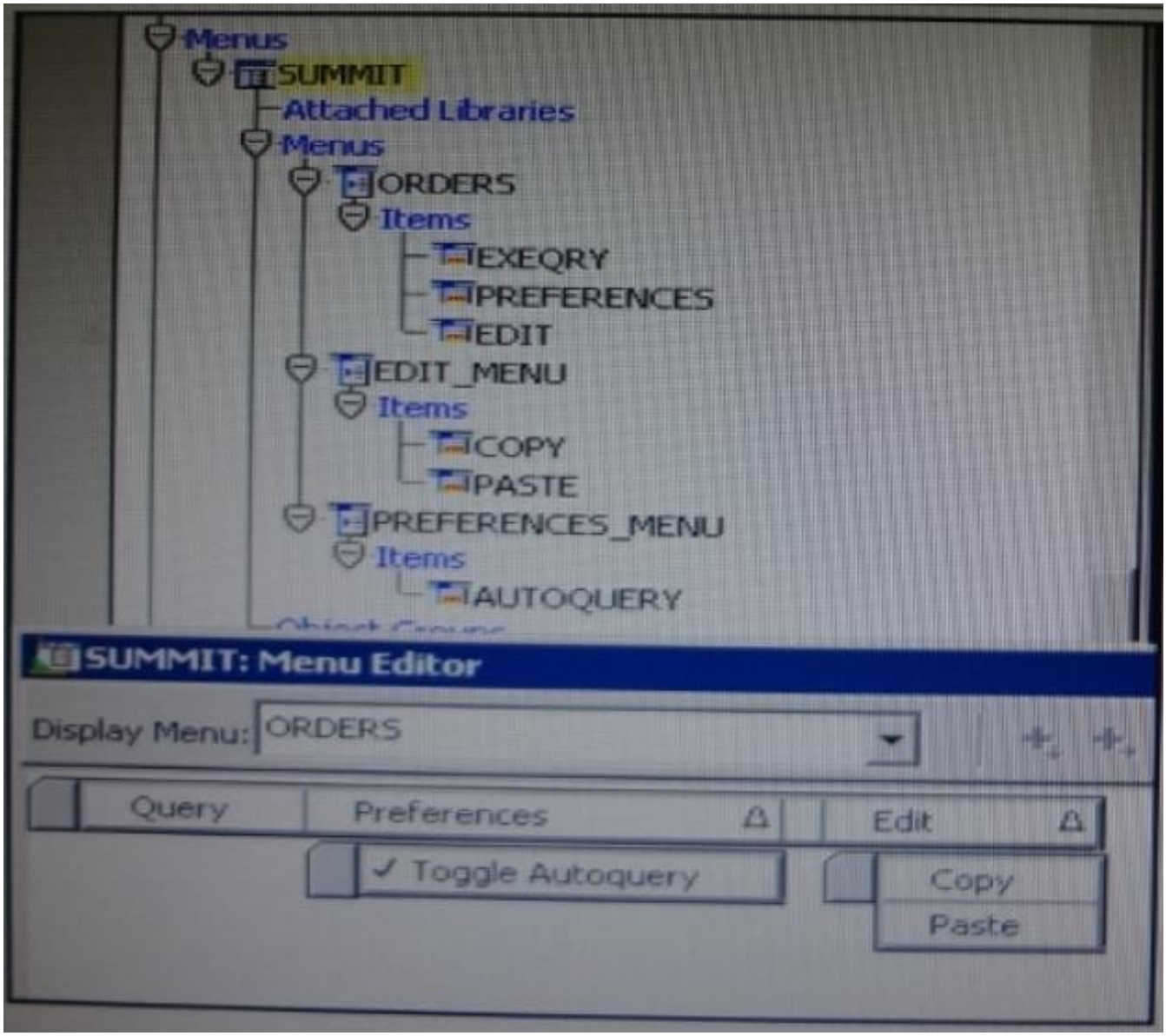
B. False

Correct Answer: A

Starting with Oracle version 8 Oracle has the ability to enforce primary key and unique key constraints using non-unique indexes.

QUESTION 4

View the Exhibit.



The Summit menu is attached to the Orders form. The Toggle Autoquery menu item is a check box that toggles whether a query is automatically performed when the Orders form is first invoked. If the check box is deselected, users must manually query.

In addition to using the menu, users want to be able to toggle the autoquery preference directly from the form. You add a button named Toggle Autoquery with the following When-Button- Pressed trigger:

DECLARE

mi_id MENUITEMS;

BEGIN

mi_id ;=FIND_ITEM (\\'Preferences.AutoQuery\\')

/* Determine the current checked static of the AutoCommit menu checkbox item And toggle the checked

state*/

IF GET_ITEM_PROPERTY (mi_id, CHECKED) = \\'TRUE\\' THEN

SET_ITEM_PROPERTY (mi_id, CHECKED, PROPERTY_FALSE);

ELSE

SET_ITEM_PROPERTY (mi_id, CHECKED, PROPERTY_TRUE);

END IF;

END;

However, the trigger does not compile. What three changes must you make so that the trigger compiles

successfully?

A. Change FIND_ITEM to FIND_MENU_ITEM.

B. Change \\'preferences.AutoQuery\\' to \\'orders.preferences.AutoQuery\\'.

C. Change \\'preferences.AutoQuery\\' to \\'AutoQuery\\'.

D. Change \\'preferences.AutoQuery\\' to \\'ORDERS.PREFERENCES>AUTOQUERY\\'.

E. Change \\'preferences.AutoQuery\\' to \\'AUTOQUERY\\'.

F. Change GET_ITEM_PROPERTY to GET_MENU_ITEM_PROPERTY

G. Change SET_ITEM_PROPERTY to SET_MENU_ITEM_PROPERTY

H. Change PROPERTY_FALSE to \\'FALSE\\'.

I. Change PROPERTY_TRUE to \\'TRUE\\'.

Correct Answer: AFG

A: Note: FIND_MENU_ITEM built-in Description Searches the list of menu items and returns a menu item ID when it finds a valid menu item with the given name. You must define an appropriately typed variable to accept the return value. Define the variable with a type of MenuItem.

Note 2:

FIND_ITEM built-in

Description

Searches the list of items in a given block and returns an item ID when it finds a valid item with the given

name. You must define an appropriately typed variable to accept the return value. Define the variable

with a type of Item.

Example (with FIND_MENU_ITEM, GET_MENU_ITEM_PROPERTY,

SET_MENU_ITEM_PROPERTY)

FIND_MENU_ITEM examples

/*

** Built-in: FIND_MENU_ITEM

** Example: Find the id of a menu item before setting

** multiple properties

*/

PROCEDURE Toggle_AutoCommit_Mode IS

mi_id MenuItem;

val VARCHAR2(10);

BEGIN

mi_id := Find_Menu_Item(\\'Preferences.AutoCommit\\');

/*

** Determine the current checked state of the AutoCommit ** menu checkbox item

*/

val := Get_Menu_Item_Property(mi_id,CHECKED);

/*

** Toggle the checked state

*/

IF val = \\'TRUE\\' THEN

Set_Menu_Item_Property(mi_id,CHECKED,PROPERTY_FALSE);

ELSE

Set_Menu_Item_Property(mi_id,CHECKED,PROPERTY_TRUE);

END IF;

END;

QUESTION 5

To troubleshoot a problem with a form, you have added a call to the MESSAGE () built-in at the beginning of the When-Validate-Item trigger of the Customer_Id then in the Orders Block of the Orders form. The message simply states that the trigger is firing.

You run the form, make a change in Customer_Id, and then tab out of the item but the message does not appear. What

are two possible causes for this problem?

A. The form is in Enter-Query mode.

B. The item is using an LOV for validation.

C. The validation unit of the form needs to be changed.

D. The MESSAGE () built-in is not allowed in validation triggers.

E. There is a syntax error in the call to the MESSAGE() built-in.

F. Validation for the Customer_Id item failed.

Correct Answer: AB

A: WHEN-VALIDATE-ITEM does not fire in ENTER-QUERY mode.

B: LOVs in ENTER-QUERY Mode

LOVs in ENTER-QUERY mode should be used sparingly, as Query Find is the preferred method for a user to locate records.

You should only code them where they dramatically improve the usability of ENTER-QUERY mode, and you expect this mode to be used regularly despite Query Find.

An LOV in ENTER-QUERY mode should display all values that the user can query, not just currently valid values.

EXAMPLE: An LOV for vendors in a purchase order form in enter-query mode shows all vendors that could ever be placed on a PO, not just the set of vendors that currently are allowed to be placed on a PO.

Do not reuse the entry LOV in ENTER_QUERY mode unless it provides the correct set of data for both modes.

Important: WHEN-VALIDATE-ITEM does not fire in ENTER-QUERY mode. Therefore, you cannot depend on the WHEN-VALIDATE-ITEM trigger to clear hidden fields when selecting from an ENTER-QUERY LOV.

Note: Validation occurs when you press enter, when you navigate away from the item, or when you save your block to the database.

Note 2: When-Validate-Item trigger Description Fires during the Validate the Item process. Specifically, it fires as the last part of item validation foritems with the New or Changed validation status. Definition Level form, block, or item Legal Commands SELECT statements, unrestricted built-ins Enter Query Mode no Usage Notes

*

Use a When-Validate-Item trigger to supplement Form Builder default item validation processing.

*

It is possible to write a When-Validate-Item trigger that changes the value of an item that FormBuilder is validating. If validation succeeds, Form Builder marks the changed item as Valid anddoes not re-validate it. While this behavior is necessary to avoid validation loops, it does make itpossible for your application to commit an invalid value to the database.

*

The setting you choose for the Defer Required Enforcement property can affect the When- Validate-Item trigger. See

Defer_Required_Enforcement for details. On Failure If fired as part of validation initiated by navigation, navigation fails, and the focus remains on theoriginal item. Fires In Validate the Item