

Oracle

Exam 1z0-144

Oracle Database 11g: Program with PL/SQL

Version: 8.5

[Total Questions: 103]

Question No : 1

What is the correct definition of the persistent state of a packaged variable?

- A. It is a private variable defined in a procedure or function within a package body whose value is consistent within a user session.
- B. It is a public variable in a package specification whose value is consistent within a user session.
- C. It is a private variable in a package body whose value is consistent across all current active sessions.
- D. It is a public variable in a package specification whose value is always consistent across all current active sessions.

Answer: B

Question No : 2

View Exhibit1 and examine the structure of the employees table.

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

View Exhibit2 and examine the code.

```
CREATE OR REPLACE FUNCTION increase (emp_num NUMBER)
RETURN number IS
inc_amt NUMBER;
sal NUMBER;
BEGIN
SELECT salary INTO sal FROM employees WHERE employee_id = emp_num;
inc_amt := sal * .10;
RETURN inc_amt;
END increase;
/
CREATE OR REPLACE PROCEDURE calc_sal IS
emp_num NUMBER(6) := 120;
amt NUMBER := 0;
PROCEDURE raise_salary (emp_id NUMBER) IS
BEGIN
amt := increase(emp_num);
UPDATE employees SET salary = salary + amt
WHERE employee_id = emp_id;
END raise_salary;
BEGIN
raise_salary(emp_num);
END calc_sal;
/
```

What is the outcome when the code is executed?

- A. Both blocks compile and execute successfully when called.
- B. Both blocks compile successfully but the CALC_SAL procedure gives an error on execution.
- C. The CALC_SAL procedure gives an error on compilation because the amt variable should be declared in the RAISE_SALARY procedure.
- D. The CALC_SAL procedure gives an error on compilation because the RAISE_SALARY procedure cannot call the stand-alone increase function.

Answer: A

Question No : 3

View the Exhibit and examine the structure of the EMP table.

```
SQL> desc emp
```

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

You want to create two procedures using the overloading feature to search for employee details based on either the employee name or employee number.

Which two rules should you apply to ensure that the overloading feature is used successfully? (Choose two.)

- A. The procedures can be either stand-alone or packaged.
- B. The procedures should be created only as packaged subprograms
- C. The procedures should be created only as stand-alone subprograms
- D. Each subprogram's formal parameters should differ in both name and data type.
- E. The formal parameters of each subprogram should differ in data type but can use the same names.

Answer: B,E

Question No : 4

View the exhibit to examine the PL/SQL code.

```
DECLARE
  emp_column VARCHAR2(30) := 'last_name';
  table_name VARCHAR2(30) := 'emp';
  temp_var   VARCHAR2(30);
BEGIN
  temp_var := emp_column;
  SELECT COLUMN_NAME INTO temp_var FROM USER_TAB_COLS
  WHERE TABLE_NAME = 'EMPLOYEES'
  AND COLUMN_NAME = UPPER(emp_column);
  temp_var := table_name;
  SELECT OBJECT_NAME INTO temp_var FROM USER_OBJECTS
  WHERE OBJECT_NAME = UPPER(table_name)
  AND OBJECT_TYPE = 'TABLE';
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE
      ('No Data found for SELECT on ' || temp_var);
END;
```

Which statement is true about the exception handlers in the PL/SQL code?

- A. All the exceptions in the code are trapped by the exception handler.
- B. All the "no data found" errors in the code are trapped by the exception handler.
- C. The PL/SQL program does not execute because an exception is not declared in the declare section.
- D. An exception handler in the code traps the "no data found" error after executing the handler code and the program flow returns to the next line of code.

Answer: B

Question No : 5

You want to maintain an audit of the date and time when each user of the database logs off.

Examine the following code:

```
SQL>CREATE TABLE log_trig_table(  
user_id VARCHAR2(30),  
log_date TIMESTAMP,  
action VARCHAR2(40));  
  
SQL>CREATE OR REPLACE TRIGGER logoff_trig  
  
BEGIN  
INSERT INTO log_trig_table(user_id,log_date,action)  
VALUES (USER, SYSDATE, 'Logging off');  
END;
```

Which two clauses should be used to fill in the blanks and complete the above code?

(Choose two.)

- A. ON SCHEMA
- B. ON QRXABASE
- C. AFTER LOGOFF
- D. BEFORE LOGOFF

Answer: A,D

Question No : 6

Which statements are true about PL/SQL procedures? (Choose all that apply.)

- A. Users with definer's rights who are granted access to a procedure that updates a table must be granted access to the table itself.
- B. Reuse of parsed PL/SQL code that becomes available in the shared SQL area of the server avoids the parsing overhead of SQL statements at run time.
- C. Depending on the number of calls, multiple copies of the procedure are loaded into memory for execution by multiple users to speed up performance.
- D. A PL/SQL procedure executing on the Oracle database can call an external procedure or function that is written in a different programming language, such as C or Java.

Answer: B,D

Question No : 7

Which statements are true about database triggers? (Choose all that apply.)

- A. They can invoke only PL/SQL procedures
- B. They can include SQL and PL/SQL or calls to Java procedures.
- C. They are implicitly fired by an event that must occur within an application
- D. They are implicitly fired when a triggering event occurs, depending on which user is connected

Answer: A,D

Reference: http://docs.oracle.com/cd/A57673_01/DOC/server/doc/SCN73/ch15.htm

Question No : 8

View the Exhibit and examine the package code created by SCOTT. The execute privilege on this package is granted to green.

```
CREATE OR REPLACE PACKAGE pkg1 IS
  PRAGMA SERIALLY_REUSABLE;
  num NUMBER := 0;
  PROCEDURE init_pkg_state(n NUMBER);
  PROCEDURE print_pkg_state;
END pkg1;
/
CREATE OR REPLACE PACKAGE BODY pkg1 IS
  PRAGMA SERIALLY_REUSABLE;
  PROCEDURE init_pkg_state (n NUMBER) IS
  BEGIN
    pkg1.num := n;
    DBMS_OUTPUT.PUT_LINE('Num: ' || pkg1.num);
  END;
  PROCEDURE print_pkg_state IS
  BEGIN
    DBMS_OUTPUT.PUT_LINE('Num: ' || pkg1.num);
  END;
END pkg1;
/
```

Examine the following sequence of commands issued by SCOTT:

```
SQL>SET SERVEROUTPUT ON
SQL>EXEC pkg1.init_pkg_state(5)
SQL>EXEC pkg1.print_pkg_state
GREEN logs in and issues the following commands:
SQL>SET SERVEROUTPUT ON
SQL>EXEC scott.pkg1.print_pkg_state
```

What is the outcome?

- A. SCOTT'S session displays 5, and then 0, green's session displays 0.
- B. SCOTT'S session displays 5, and then 0; green's session displays 5.
- C. SCOTT'S session displays 5, and then 5 again, green's session displays 0.
- D. SCOTT'S session displays 5, and then 5 again; green's session displays 5.

Answer: B

Question No : 9

Examine the following code:

```
SQL> SET SERVEROUTPUT ON
SQL> VARIABLE n1 NUMBER
SQL> VARIABLE n2 NUMBER
SQL>CREATE OR REPLACE PROCEDURE procl
(:n1 IN OUT NUMBER, :n2 IN OUT NUMBER) IS
BEGIN
:n1 := 20;
DBMS_OUTPUT.put_line(:n1);
:n2 := 30;
DBMS_OUTPUT.put_line(:n2);
END;
```

What is the outcome?

- A. The procedure is created successfully and displays the values 20 and 30 when it is called.
- B. The procedure gives errors because the parameters should be in out mode.
- C. The procedure gives errors because the host variables cannot be referenced anywhere in the definition of a PL/SQL stored procedure.
- D. The procedure is created successfully but does not display any values when it is called because the host variables cannot be displayed inside the procedure.

Answer: C

Question No : 10

View the Exhibit and examine the structure of the customer table.

Name	Null?	Type
CUST_ID	NOT NULL	NUMBER
CUST_LAST_NAME	NOT NULL	VARCHAR2 (40)
CUST_CITY	NOT NULL	VARCHAR2 (30)
CUST_CREDIT_LIMIT		NUMBER
CUST_CATEGORY		VARCHAR2 (20)

You create the following trigger to ensure that customers belonging to category "A" or "B" in the customer table can have a credit limit of more than 8000.

```
SQL>CREATE OR REPLACE TRIGGER restrict_credit_limit
  BEFORE INSERT OR UPDATE ON customer
  FOR EACH ROW
  BEGIN
    IF (:NEW.cust_category NOT IN ('A', 'B'))
      AND :NEW.cust_credit_limit > 8000 THEN
      DBMS_OUTPUT.PUT_LINE ('Credit Limit cannot be greater
        than 8000 for this category');
    END IF;
  END;
/
```

You execute the following UPDATE command for CUST_ID 101 existing in the CUSTOMER table.

```
SQL> UPDATE customer SET cust_category = 'C', cust_credit_limit = 9000
  WHERE cust_id = 101;
```

What is the outcome?

- A. The trigger is fired, a message is displayed, and the update is successful
- B. The trigger is fired and a message is displayed, but the update is rolled back.
- C. The trigger is not fired because the when clause should be used to specify the condition, however, the update is successful.
- D. The trigger is not fired because column names must be specified with the update event to identify which columns must be changed to cause the trigger to fire, however, the update is successful.

Answer: D

Question No : 11

Which two statements are correct about PL/SQL package components? (Choose two)

- A. A package must have both specification and body.
- B. A package body can exist without the package specification.
- C. A package specification can exist without the package body.
- D. When a packaged public variable is called for the first time in a session, the entire package is loaded into memory.

Answer: C,D

Question No : 12

Which system events can be used to create triggers that fire both at database and schema levels? (Choose all that apply)

- A. AFTER LOGON
- B. AFTER STARTUP
- C. BEFORE SHUTDOWN
- D. AFTER SERVERERROR

Answer: A,D

Explanation:

http://docs.oracle.com/cd/E11882_01/appdev.112/e25519/create_trigger.htm#LNPLS2064