# 1Z0-148<sup>Q&As</sup>

Oracle Database: Advanced PL/SQL

## Pass Oracle 1Z0-148 Exam with 100% Guarantee

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

**https://www.passapply.com/1z0-148.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**

Which tablespace is used to store the data collected by PL/Scope?

A. UNDOTBS1

B. SYSAUX

C. SYSTEM

D. TEMP

E. USERS

Correct Answer: B

Reference: https://docs.oracle.com/cd/B28359_01/appdev.111/b28424/adfns_plscope.htm#BABDGJAF

**QUESTION 2**

Examine the code in the following PL/SQL block:

DECLARE

TYPE NumList IS TABLE OF INTEGER;

List1 NumList := NumList(11,22,33,44);

BEGIN

List1.DELETE(2);

DBMS_OUTPUT.PUT_LINE

( \\'The last element# in List1 is \\' || List1.LAST ||

\\' and total of elements is \\'||List1.COUNT);

List1.EXTEND(4,3);

END;

/

Which two statements are true about the above code? (Choose two.)

A. LAST and COUNT give different values.

B. LAST and COUNT give the same values.

C. The four new elements that are added contain the value 33.

D. The four new elements that are added contain the value 44.

Correct Answer: AC

**QUESTION 3**

You are connected as SCOTT who has an EMP table with this structure:

```
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)
```

Examine this code:

```
   CREATE FUNCTION emp_policy_fn (v_schema IN VARCHAR2, v_objname IN VARCHAR2) RETURN
VARCHAR2 AS
    con VARCHAR2(200);
  BEGIN
    con := 'deptno = 30';
    RETURN con;
  END emp_policy_fn;
  /
  BEGIN
    DBMS_RLS.ADD_POLICY (
      object_schema => 'scott',
      object_name => 'emp',
      policy_name => 'emp_policy',
      policy_function => 'emp_policy_fn',
      sec_relevant_cols => 'sal, comm' );
  END;
  /
  CREATE EDITION a1;
  ALTER SESSION SET EDITION = a1;
  CREATE OR REPLACE FUNCTION emp_policy_fn (v_schema IN VARCHAR2, v_objname IN
VARCHAR2) RETURN VARCHAR2 AS
    con VARCHAR2(200);
  BEGIN
    con := 'deptno = 20';
    RETURN con;
  END emp_policy_fn;
  /
  SELECT * FROM EMP;
```

Assuming the default edition is ORA$BASE, which is correct?

A. The query will return records pertaining to department 30 with SAL and COMM values displayed as NULL.

B. The query will return records pertaining to department 20.

C. The query will return records pertaining to department 20 with SAL and COMM values displayed as NULL.

D. The query will return records pertaining to department 30.

Correct Answer: A

## QUESTION 4

A procedure is created in the SYS schema to allow users to change the password as follows:

CREATE OR REPLACE

PROCEDURE change_password(p_username VARCHAR2 DEFAULT NULL,

p_new_password VARCHAR2 DEFAULT NULL)

IS

v_sql_stmt VARCHAR2(500);

BEGIN

v_sql_stmt := \\'ALTER USER \\'||p_username ||\\' IDENTIFIED BY \\'

|| p_new_password;

EXECUTE IMMEDIATE v_sql_stmt;

END change_password;

The SYS user has granted EXECUTE privilege on the procedure to the OE user. But OE is able to change

the password for SYS by using this procedure. How would you protect this?

A. by using the procedure as part of a PL/SQL package

B. by using a bind argument with dynamic SQL in the procedure

C. by using AUTHID DEFINER in the procedure to implement the definer\\'s right

D. by using AUTHID CURRENT_USER in the procedure to implement the invoker\\'s right

Correct Answer: D

## QUESTION 5

Which two blocks of code execute successfully?

A. DECLARESUBTYPE new_one IS BINARY_INTERGER RANGE 0..9;my_val new_one;BEGINmy_val :=0;END;

B. DECLARESUBTYPE new_string IS VARCHAR2 (5) NOT NULL;my_str_new_string;BEGINmy_str := `abc\\';END;

C. DECLARESUBTYPE new_one IS NUMBER (2, 1);my_val new_one;BEGINmy_val :=12.5;END;

D. DECLARESUBTYPE new_one IS INTEGER RANGE 1..10 NOT NULL;my_val new_one;BEGINmy_val :=2;END;

E. DECLARESUBTYPE new_one IS NUMBER (1, 0);my_val new_one;BEGINmy_val := -1;END;

Correct Answer: AD