



# 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.geekcert.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

You execute the following command in the user session: `SQL> ALTER SESSION SET PLSQL_DEBUG=true`; Which statement is true about the effect of the command?

- A. All PL/SQL blocks that are executed subsequently in the session are traced.
- B. It enables all PL/SQL blocks that are compiled subsequently in the session for tracing.
- C. Only anonymous PL/SQL blocks that are executed subsequently in the session are traced.
- D. It enables only named PL/SQL blocks that are executed subsequently in the session for tracing.

Correct Answer: B

---

### QUESTION 2

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 3

Examine the following command to create the table EMPLOYEES\_TEMP and the PL/SQL block.

```
CREATE TABLE employees_temp (empid NUMBER(6) NOT NULL,  
deptid NUMBER(6) CONSTRAINT c_emp_deptid CHECK (deptid BETWEEN 100 AND 200),  
salary Number(8),  
deptname VARCHAR2(30) DEFAULT '\\Sales\\')  
  
/  
  
DECLARE  
  
SUBTYPE v_emprec_subtype IS employees_temp%ROWTYPE;  
  
v_emprec v_emprec_subtype;  
  
BEGIN  
  
v_emprec.empid := NULL; v_emprec.salary := 10000.002;  
  
v_emprec.deptid := 50;  
  
DBMS_OUTPUT.PUT_LINE('\\v_emprec.deptname: \\ || v_emprec.deptname);  
  
END;  
  
/
```

Which statements are true about the above PL/SQL block? (Choose two.)

- A. V\_EMPREC.DEPTNAME would display a null value because the default value is not inherited.
- B. Assigning null to V\_EMPREC.EMPID would generate an error because the null constraint is inherited.
- C. Assigning the value 1000.002 to V\_EMPREC.SALARY would generate an error because of the decimal.
- D. Assigning the value 50 to V\_EMPREC.DEPTID would work because the check constraint is not inherited.

Correct Answer: AD

---

### QUESTION 4

Examine this function: Execute the query:



```
CREATE FUNCTION remap_schema RETURN CLOB IS
  h NUMBER;
  th NUMBER;
  doc CLOB;
BEGIN
  h := DBMS_METADATA.OPEN ('TABLE')
  DBMS_METADATA.SET_FILTER (h, 'SCHEMA', 'SCOTT');
  DBMS_METADATA.SET_FILTER (h, 'NAME', 'EMP');
  th := DBMS_METADATA.ADD_TRANSFORM (h, 'MODIFY');
  DBMS_METADATA.SET_REMAP_PARAM (th, 'REMAP_SCHEMA', 'SCOTT', NULL);
  DBMS_METADATA.SET_REMAP_PARAM (th, 'REMAP_TABLESPACE', 'USERS',
'SYSAUX');
  th := DBMS_METADATA.ADD_TRANSFORM (h, 'DDL');
  DBMS_METADATA.SET_TRANSFORM_PARAM (th, 'SEGMENT_ATTRIBUTES',
FALSE);
  doc := DBMS_METADATA.FETCH_CLOB (h);
  DBMS_METADATA.CLOSE (h);
  RETURN doc;
END remap_schema;
```

SELECT remap\_schema FROM dual;

Which is the correct output from the query?

A. CREATE TABLE "EMP" ("EMPNO" NUMBER (4,0), "ENAME" VARCHAR2 (10), "JOB" VARCHAR2 (9), "MGR" NUMBER (4,0), "HIREDATE" DATE, "SAL" NUMBER (7,2) , "COMM" NUMBER (7,2), "DEPTNO" NUMBER (2,0), CONSTRAINT "PK\_EMP" PRIMARY KEY ("EMPNO") USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT) TABLESPACE "USERS" ENABLE, CONSTRAINT "FK\_DEPTNO" FOREIGN KEY ("DEPTNO") REFERENCES "DEPT" ("DEPTNO") ENABLE ) SEGMENT CREATION IMMEDIATE PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT) TABLESPACE "USERS"

B. CREATE TABLE "EMP" ("EMPNO" NUMBER (4, 0), "ENAME" VARCHAR2 (10), "JOB" VARCHAR2 (9), "MGR" NUMBER (4, 0), "HIREDATE" DATE, "SAL" NUMBER (7, 2), "COMM" NUMBER (7, 2), "DEPTNO" NUMBER (2, 0), CONSTRAINT "PK\_EMP" PRIMARY KEY ("EMPNO") USING INDEX ENABLE, CONSTRAINT "FK\_DEPTNO" FOREIGN KEY ("DEPTNO") REFERENCES "DEPT" ("DEPTNO") ENABLE)

C. CREATE TABLE "SCOTT"."EMP" ("EMPNO" NUMBER (4, 0), "ENAME" VARCHAR2 (10), "JOB" VARCHAR2 (9), "MGR" NUMBER (4, 0), "HIREDATE" DATE, "SAL" NUMBER (7, 2), "COMM" NUMBER (7, 2), "DEPTNO" NUMBER (2, 0), CONSTRAINT "PK\_EMP" PRIMARY KEY ("EMPNO")

USING INDEX ENABLE,

CONSTRAINT "FK\_DEPTNO" FOREIGN KEY ("DEPTNO")

REFERENCES "DEPT" ("DEPTNO") ENABLE)



D. CREATE TABLE "EMP" ("EMPNO" NUMBER (4,0), "ENAME" VARCHAR2 (10), "JOB" VARCHAR2 (9), "MGR" NUMBER (4,0), "HIREDATE" DATE, "SAL" NUMBER (7, 2), "COMM" NUMBER (7, 2), "DEPTNO" NUMBER (2,0), CONSTRAINT "PK\_EMP" PRIMARY KEY ("EMPNO") USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT) TABLESPACE "SYSAUX" ENABLE, CONSTRAINT "FK\_DEPTNO" FOREIGN KEY ("DEPTNO") REFERENCES "DEPT" ("DEPTNO") ENABLE ) SEGMENT CREATION IMMEDIATE PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT) TABLESPACE "SYSAUX"

Correct Answer: B

## QUESTION 5

Examine the incomplete code:

```
CREATE TYPE numlist IS TABLE OF NUMBER;
/
CREATE PROCEDURE list_sal (dept_id NUMBER)
IS
    sql_stmt  VARCHAR2 (200);
    ret       INTEGER;
    empids    numlist;
    sal       numlist;
BEGIN
    curid := DBMS_SQL.OPEN_CURSOR;
    sql_stmt := 'SELECT employee_id, salary FROM employees WHERE department_id = :a1';
    DBMS_SQL.PARSE (curid, sql_stmt, DBMS_SQL.NATIVE);
    DBMS_SQL.BIND_VARIABLE (curid, 'a1', 'dept_id');
    ret := DBMS_SQL.EXECUTE (curid);
    FETCH src_cur BULK COLLECT INTO empids, sal;
    IF empids.COUNT > 0 THEN
        FOR i IN 1 .. empids.COUNT LOOP
            DBMS_OUTPUT.PUT_LINE (empids (i) || ' ' || sal (i));
        END LOOP;
    END IF;
    CLOSE src_cur;
END;
```

Which three lines of code must be added for it to successfully compile?

- A. curid := DBMS\_SQL.TO\_CURSOR\_NUMBER (src\_cur);
- B. src\_cur := DBMS\_SQL.TO\_REFCURSOR (curid);



C. src\_cur= NUMBER;

D. curid NUMBER;

E. curid SYS\_FEF\_CURSOR;

F. src\_cur SYS\_REFCURSOR;

Correct Answer: BDF

[1Z0-148 PDF Dumps](#)

[1Z0-148 Practice Test](#)

[1Z0-148 Braindumps](#)