



1Z0-144^{Q&As}

Oracle Database 11g: Program with PL/SQL

Pass Oracle 1Z0-144 Exam with 100% Guarantee

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

<https://www.geekcert.com/1z0-144.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**

View the Exhibit and examine the structure of the departments table in SCOTT's schema.

Name	Null?	Type
DEPARTMENT_ID	NOT NULL	NUMBER (4)
DEPARTMENT_NAME	NOT NULL	VARCHAR2 (30)
MANAGER_ID		NUMBER (6)
LOCATION_ID		NUMBER (4)

Examine the following block of code:

```
CREATE OR REPLACE PROCEDURE add_dept (  
    p_id NUMBER, p_name VARCHAR2) IS  
BEGIN  
    INSERT INTO departments  
    VALUES (p_id, p_name, NULL, NULL);  
END;  
/
```

The above procedure is created by user SCOTT. Another user JONES needs to use the procedure.

Which two statements are true in the above scenario? (Choose two.)

- A. JONES executes the procedure with definer's rights.
- B. JONES executes the procedure with invoker's rights.
- C. SCOTT should grant only the execute privilege for the procedure to JONES.
- D. SCOTT should grant both the execute privilege for the procedure and insert privilege for the table to JONES.

Correct Answer: AC

QUESTION 2



View Exhibit 1 and examine the structure of the EMP table.

```
SQL> desc emp
Name                Null?              Type
-----
EMPNO               NOT NULL          NUMBER(4)
ENAME
JOB                 VARCHAR2(9)
MGR                 NUMBER(4)
HIREDATE            DATE
SAL                 NUMBER(7,2)
COMM                NUMBER(7,2)
DEPTNO              NUMBER(2)
```

View Exhibit 2 and examine the code of the packages that you have created.



```
CREATE OR REPLACE PACKAGE manage_emp IS
  v_empno NUMBER;
  PROCEDURE del_emp (p_empno NUMBER);
END manage_emp;
/
```

```
CREATE OR REPLACE PACKAGE BODY manage_emp IS
  PROCEDURE del_emp (p_empno NUMBER) IS
  BEGIN
    DELETE FROM emp WHERE empno=p_empno;
  END del_emp;
END manage_emp;
/
```

```
CREATE OR REPLACE PACKAGE emp_det IS
  PROCEDURE emp_chk (p_empno NUMBER);
END emp_det;
/
```

```
CREATE OR REPLACE PACKAGE BODY emp_det IS
  PROCEDURE emp_chk (p_empno NUMBER) IS
  BEGIN
    manage_emp.del_emp (p_empno);
  END emp_chk;
END emp_det;
/
```

You issue the following command: SQL> DROP PACKAGE manage_emp; What is the outcome?

- A. It drops both the MANAGE_EMP AND EMP__DET packages because of the cascading effect.
- B. It drops the MANAGE_EMP package and invalidates only the body for the EMP_DET package.
- C. It returns an error and does not drop the MAMAGE_EMP package because of the cascading effect.
- D. It drops the MANAGE_EMP package and invalidates both the specification and body for the EMP_DET package.

Correct Answer: B

QUESTION 3



View the Exhibit to examine the PL/SQL code.

```
DECLARE
    jobid employees.job_id%TYPE;
    empid employees.employee_id%TYPE :=115;
    sal employees.salary%TYPE;
    sal_raise NUMBER(3,2);
BEGIN
    SELECT job_id, salary INTO jobid, sal from employees
    WHERE employee_id = empid;
    CASE
    WHEN jobid = 'PU_CLERK' THEN
        IF sal < 3000 THEN sal_raise := .12;
            ELSE sal_raise := .09;
        END IF;
    WHEN jobid = 'SH_CLERK' THEN
        IF sal < 4000 THEN sal_raise := .11;
            ELSE sal_raise := .08;
        END IF;
    WHEN jobid = 'ST_CLERK' THEN
        IF sal < 3500 THEN sal_raise := .10;
            ELSE sal_raise := .07;
        END IF;
    ELSE
    BEGIN
        DBMS_OUTPUT.PUT_LINE('No raise for this job: ' || jobid);
    END;
    END CASE;
    UPDATE employees SET salary = salary + salary * sal_raise
    WHERE employee_id = empid;
    COMMIT;
END;
```

SERVEROUTPUT is on for the session.

Which statement is true about the execution of the code?

- A. The execution fails because of the misplaced else clause.
- B. The execution is successful even if there is no employee with EMPLOYEE_ID 115.
- C. The execution fails and throws exceptions if no employee with EMPLOYEE_ID 115 is found.
- D. The execution is successful, but it displays an incorrect output if no employee with EMPLOYEE_ID 115 is found.

Correct Answer: C

QUESTION 4



View the Exhibit and examine the code and its outcome on execution:

```
SQL> CREATE PACKAGE my_debug IS
  2   debug CONSTANT BCOLEAN := TRUE;
  3   trace CONSTANT BCOLEAN := TRUE;
  4 END my_debug;
  5 /
```

Package created.

```
SQL> CREATE PROCEDURE my_procl IS
  2 BEGIN
  3   çIF my_debug.debug çTHEN
  4     DBMS_OUTPUT.put_line('Debugging ON');
  5   çELSE
  6     DBMS_OUTPUT.put_line('Debugging OFF');
  7   çEND
  8 END my_procl;
  9 /
```

Procedure created.

```
SQL> CREATE PROCEDURE my_proc2 IS
  2 BEGIN
  3   çIF my_debug.trace çTHEN
  4     DBMS_OUTPUT.put_line('Tracing ON');
  5   çELSE DBMS_OUTPUT.put_line('Tracing OFF');
  6   çEND
  7 END my_proc2;
  8 /
```

Procedure created.

What would be the effect on the two procedures if the value of debug is set to FALSE? (Choose two.)

- A. MY_PROC2 is not recompiled.
- B. MY_PROC1 is recompiled but remains unchanged.
- C. MY_PROC2 is recompiled but remains unchanged.
- D. MY_PROC1 is recompiled without the debugging code.

Correct Answer: CD



QUESTION 5

View Exhibit 1 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)
COMISSION_PCT		NUMBER (2, 2)
MANAGER_ID		NUMBER (6)
DEPARTMENT_ID		NUMBER (4)

View Exhibit 2 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.

Correct Answer: A

[1Z0-144 Study Guide](#)

[1Z0-144 Exam Questions](#)

[1Z0-144 Braindumps](#)