



BrainDumps  
Collection

# Oracle

## 1Z0-144 Exam

Oracle Database 11g: Program with PL/SQL

Thank you for Downloading 1Z0-144 exam PDF Demo

You can also try our 1Z0-144 practice exam software

**Download Free Demo**

<https://www.braindumpscollection.com/1Z0-144.html>

DEMO  
VERSION

(LIMITED CONTENT)

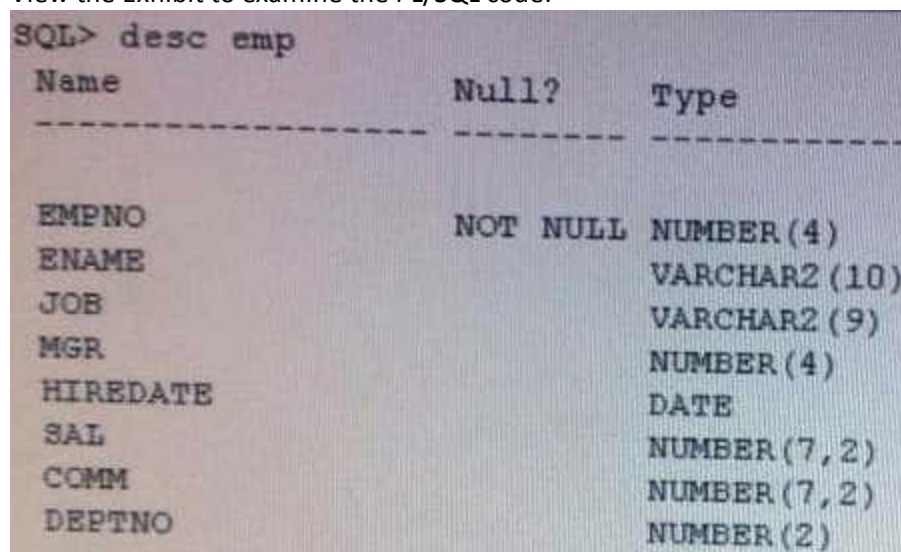
Questions  
& Answers

---

**Question: 1**

---

View the Exhibit to examine the PL/SQL code:



```
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)

SREYROUPUT is on for the session. Which statement is true about the output of the PL/SQL block?

- A. The output is  $x = y$ .
- B. It produces an error.
- C. The output is  $x \neq y$ .
- D. The output is Can't tell if x and y are equal or not.

---

**Answer: A**

---

---

**Question: 2**

---

Examine the following command:

```
SQL> ALTER SESSION  
SET plsql_warnings *  
'enable: severe',  
'enable: performance',  
'ERROR: 05003';
```

What is the implication of the above command?

- A. It issues a warning whenever ERROR: 05003 occur during compilation.
- B. It causes the compilation to fail whenever the warning ERROR.05003 occurs.
- C. It issues warnings whenever the code causes an unexpected action or wrong results performance problems.
- D. It causes the compilation to fail whenever the code gives wrong results or contains statements that are never executed.

---

**Answer: C**

---

---

**Question: 3**

---

View the exhibit and examine the structure of the products table.

Name	Null?	Type
PROD_ID	NOT NULL	NUMBER(4)
PROD_NAME	NOT NULL	VARCHAR2(30)
PROD_LIST_PRICE	NOT NULL	NUMBER(8,2)
PROD_VALID		VARCHAR2(1)

Examine the following code

```
CREATE TABLE debug_output (msg VARCHAR2(100));

CREATE OR REPLACE PROCEDURE debugging (msg VARCHAR2) AS
  PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
  INSERT INTO debug_output VALUES (msg);
  COMMIT;
END debugging;
/

CREATE OR REPLACE PROCEDURE delete_details(p_id NUMBER) AS
  msg VARCHAR2(100);
BEGIN
  DELETE FROM products WHERE prod_id = p_id;
  COMMIT;
EXCEPTION
  WHEN OTHERS THEN
    msg := SUBSTR(sqlerrm,100);
    debugging (msg);
END delete_details;
/
```

Which statement is true when the procedure DELETE\_DETAILS is invoked?

- A. It executes successfully but no error messages get recorded in the DEBUG\_OUTPUT table
- B. It executes successfully and any error messages get recorded in the DEBUG\_OUTPUT table.
- C. It gives an error because PRAGMA AUTONOMOUS\_TRANSACTION can be used only in packaged procedures.
- D. It gives an error because procedures containing PRAGMA AUTONOMOUS\_TRANSACTION cannot be called from the exception section.

---

**Answer: A**

---

In this case, the debug output will only occur if there is an exception.

---

**Question: 4**

---

Which two tasks should be created as functions instead of as procedures? (Choose two.)

- A. Reference host or bind variables in a PL7SQL block of code
- B. Tasks that compute and return multiple values to the calling environment
- C. Tasks that compute a value that must be returned to the calling environment
- D. Tasks performed in SQL that increase data independence by processing complex data analysis within the Oracle server, rather than by retrieving the data into an application

---

**Answer: A, C**

---

Explanation:

Functions are used to return a value. Functions must return only a single value.

Procedure are used to perform an action.

Both functions and procedures are using to do a special task or action. In functions it is must to return a single value, where as in procedures it's not compulsory

---

**Question: 5**

---

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.

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

What would be the outcome when the code is executed?

- A. It executes successfully.
- B. It gives an error because the SAL variable is not visible in the increase function.
- C. It gives an error because the increase function cannot be called from the RAISE\_SALARY procedure.
- D. It gives an error because the increase function and the RAISE\_SALARY procedure should be declared at the beginning of the declare section before all the other declarations.

---

**Answer: A**

---

---

### Question: 6

---

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: 7**

---

Examine the following block of code:

```
1 DECLARE
2   status          VARCHAR2(10) NOT NULL DEFAULT 'TRUE';
3   net_value       NUMBER := 555;
4   done            BOOLEAN;
5   valid_id        BOOLEAN := TRUE;
6 BEGIN
7   done := (net_value > 100);
8   status := valid_id;
9 END;
```

Which line in the above code would result in errors upon execution?

- A. line 5
- B. line 8
- C. line 2
- D. line 7

---

**Answer: B**

---

---

**Question: 8**

---

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)

Examine the following trigger code:

```
CREATE OR REPLACE TRIGGER max_credit_limit
  BEFORE INSERT OR UPDATE OF cust_category ON customer
  FOR EACH ROW
  WHEN (NEW.cust_category IS NULL)
BEGIN
  IF INSERTING THEN
    :NEW.cust_category := 'C';
    :NEW.cust_credit_limit := 8000;
  ELSIF UPDATING THEN
    :NEW.cust_category := :OLD.cust_category;
    :NEW.cust_credit_limit := :OLD.cust_credit_limit;
  END IF;
END;
```

What is the outcome when the above trigger is compiled?

- A. It compiles successfully.
- B. It gives an error because the when condition is not valid.
- C. It gives an error because when cannot be used for row-level triggers.
- D. It gives an error because the statements under updating are not valid.
- E. It gives an error because the new qualifier in the when clause requires a colon prefix.

---

**Answer: A**

---

---

### Question: 9

---

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: 10

---

The STRING\_TAB table has the following structure:

Name	Null?	Type
-----	-----	-----
STRING1		VARCHAR2 (100)

View the Exhibit and examine the code.

```
SQL>SET SERVEROUTPUT ON
SQL>DECLARE
    in_string VARCHAR2(25) := 'This is my test string.';
    out_string VARCHAR2(25);
    PROCEDURE double (original IN VARCHAR2,
                      new_string OUT VARCHAR2) IS
    BEGIN
        new_string := original || ' + ' || original;
    EXCEPTION
        WHEN VALUE_ERROR THEN
            DBMS_OUTPUT.PUT_LINE('Output buffer not long enough. ');
            COMMIT;
    END;
    BEGIN
        double(in_string, out_string);
        DBMS_OUTPUT.PUT_LINE(in_string || ' - ' || out_string);
    END;
    /
```

What is the outcome on execution?

- A. It displays  
Output buffer not long enough.  
This is my test string.-.
- B. It displays only  
Output buffer not long enough, and exits the anonymous block.
- C. It displays only  
This is my test string. - Because EXCEPTION should have been defined in the anonymous block to get the error message.
- D. It does not display any of the MEMS\_PUTPUT messages and gives an error because a transaction control statement cannot be used in the exception section of a procedure.

---

**Answer: A**

---



---

### Question: 11

---

Identify two situations where the DBMS\_SQL package should be used. (Choose two.)



- A. The SELECT list is not known until run time.
- B. The dynamic SQL statement retrieves rows into records.
- C. You do not know how many columns a select statement will return, or what their data types will.
- D. You must use the %found SQL cursor attribute after issuing a dynamic SQL statement that is an insert or update statement.

---

**Answer: A, C**

---

**Thank You for trying 1Z0-144 PDF Demo**

To try our 1Z0-144 practice exam software visit link below

<https://www.braindumpscollection.com/1Z0-144.html>

## Start Your 1Z0-144 Preparation

Use Coupon "20OFF" for extra 20% discount on the purchase of Practice Test Software. Test your 1Z0-144 preparation with actual exam questions.