

NowExam

Higher Quality , Better Service !



Q&A

<http://www.nowexam.com>

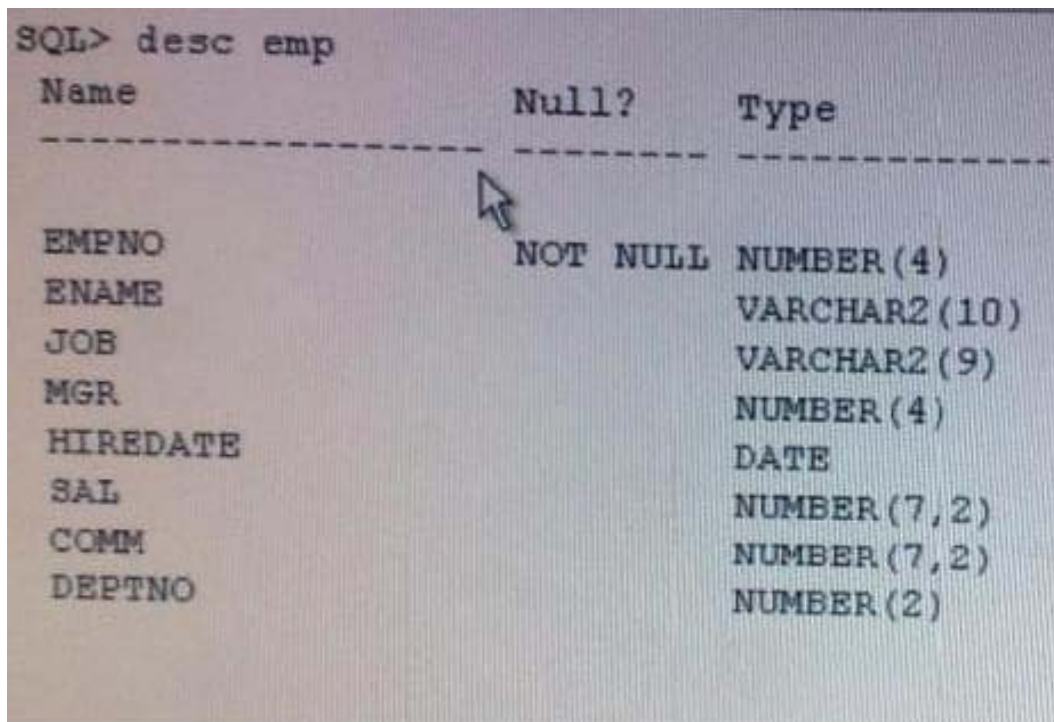
We offer free update service for one year.

Exam : **1Z0-144**

Title : Oracle Database 11g:
Program with PL/SQL

Version : DEMO

1.View the Exhibit to examine the PL/SQL code:



```
SQL> desc emp
Name                Null?              Type
-----
EMPNO               NOT NULL         NUMBER(4)
ENAME
JOB                 VARCHA2(9)
MGR                 NUMBER(4)
HIREDATE            DATE
SAL                 NUMBER(7,2)
COMM                NUMBER(7,2)
DEPTNO              NUMBER(2)
```

SREVIROUPT 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

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

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

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

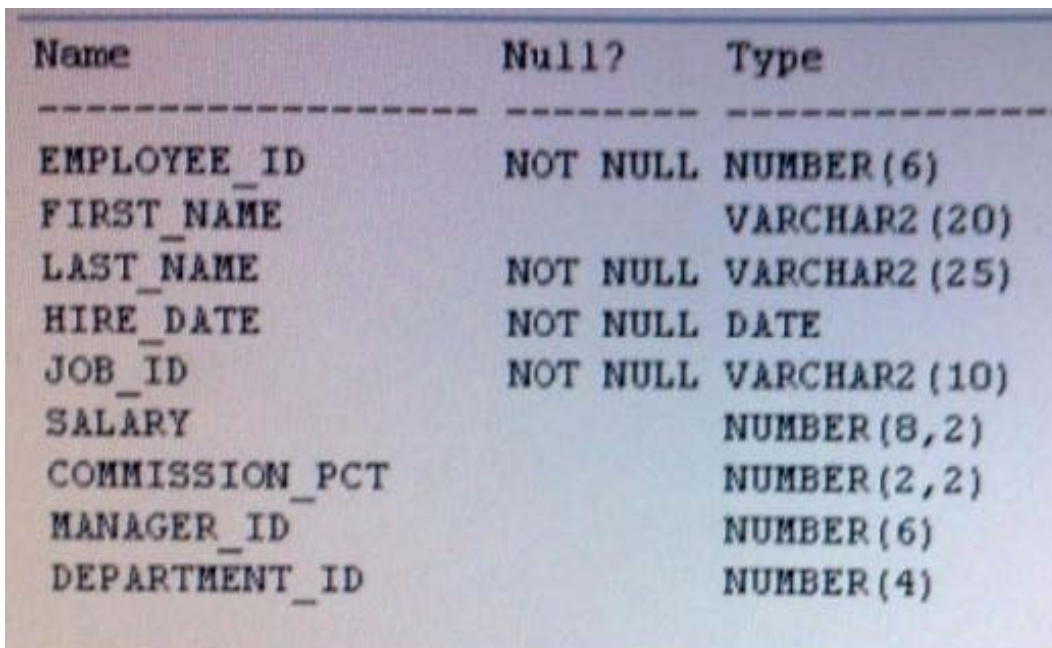
4.Which two tasks should be created as functions instead of as procedures? (Choose two.)

- A. Reference host or bind variables in a PL7SQL blockof 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: Explanation/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

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