



1Z0-148^{Q&As}

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

Examine these two columns:

- | | |
|--------------------------------|--|
| 1. CREATE_TASK | a. A constant defined for task status value |
| 2. CHUNKED | b. One of the views used in conjunction with the package |
| 3. DBA_PARALLEL_EXECUTE_CHUNKS | c. One of the procedures defined in the package |
| 4. CREATE JOB | d. A constant defined for chunk status value |
| 5. PROCESSED | e. System privilege |

Which set of paired items is valid with respect to the DBMS_PARALLEL_EXECUTE package?

- A. 1-c, 2-b, 3-e, 4-a, 5-d
- B. 1-b, 2-d, 3-c, 4-e, 5-a
- C. 1-c, 2-a, 3-b, 4-e, 5-d
- D. 1-c, 2-d, 3-b, 4-e, 5-a
- E. 1-e, 2-a, 3-c, 4-b, 5-d

Correct Answer: D

Reference: https://docs.oracle.com/en/database/oracle/oracle-database/18/arpls/DBMS_PARALLEL_EXECUTE.html#GUID-A31C2E84-91CC-4D35-AAFE-4E7E6609C963

QUESTION 2

Which statements are true about temporary LOBs? (Choose all that apply.)

- A. They can be created only for CLOB and NCLOB data.
- B. They can be accessed only by the user who creates them.
- C. They generate more redo information than persistent LOBs.
- D. They exist for the duration of the session in which they are created.
- E. They are stored temporarily in the default tablespace of the user who creates them.

Correct Answer: BD

QUESTION 3

View the Exhibit and examine the structure of the EMPLOYEES table.



Examine the following PL/SQL block for storing the salary of all sales representatives from the EMPLOYEES table in an associative array:

```
1 DECLARE
2 emp_cv SYS_REFCURSOR;
3 TYPE list IS TABLE OF emp_cv;
4 sals list;
5 BEGIN
6 OPEN emp_cv FOR SELECT salary FROM employees
7 WHERE job_id = 'SA_REP';
8 FETCH emp_cv BULK COLLECT INTO sals;
9 CLOSE emp_cv;
10 END;
```

What should you correct in the above code to ensure that it executes successfully?

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER (6)
FIRST_NAME		VARCHAR2 (20)
LAST_NAME	NOT NULL	VARCHAR2 (25)
JOB_ID	NOT NULL	VARCHAR2 (10)
SALARY	NOT NULL	NUMBER (8, 2)
COMMISSION_PCT		NUMBER (2, 2)
DEPARTMENT_ID		NUMBER (4)

- A. Replace EMP_CV in line 3 with employees.salary%TYPE.
- B. Replace line 2 with TYPE refcur IS REF CURSOR; emp_cv refcur;.
- C. Replace BULK COLLECT in line 8 with the OPEN, FETCH, LOOP, and CLOSE statements.
- D. Replace line 2 with TYPE refcur IS REF CURSOR RETURN employees.salary%TYPE; emp_cv refcur;.

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

Your company's executive team voted to give a one-time special bonus, the amount of which is to be based on number of years of service. The bonus table will be updated using a procedure called

```
update_bonus().
```

Examine this PL/SQL block which shows the logic to calculate the bonus:



```
DECLARE
  emp_rec employees%ROWTYPE;
  bonus1 INTEGER;
  bonus2 INTEGER;
  service_days INTEGER;
  CURSOR c1 IS SELECT * FROM employees;
BEGIN
  OPEN c1;
  LOOP
    FETCH c1 INTO emp_rec;
    EXIT WHEN c1%NOTFOUND;
    service_days := sysdate - emp_rec.hire_date;
    bonus1 := 10000;
    bonus2 := 5000;
    IF (sysdate - emp_rec.hire_date) > 10*365 THEN
      update_bonus (emp_rec.employee_id, bonus1);
    ELSIF (sysdate - emp_rec.hire_date) > 5*365 THEN
      update_bonus (emp_rec.employee_id, bonus2);
    END IF;
  END LOOP;
  CLOSE c1;
END;
/
```

Consider these possible optimization actions:

1.

Remove the service_days calculation statement.

2.

Move bonus1 and bonus2 initializations out of the loop.

3.

Inline the update_bonus() subprogram.

4.

Call update_bonus(), passing the literal bonus amount instead of the bonus1 or bonus2 variables. Which set of optimizations will be performed if this block is compiled with PLSQL_OPTIMIZE_LEVEL = 2?

A. 3 and 4 only

B. 1, 2, 3, and 4

C. 1, 2, and 4 only

D. 1 and 2 only

E. 1 and 3 only



Correct Answer: D

[Latest 1Z0-148 Dumps](#)

[1Z0-148 PDF Dumps](#)

[1Z0-148 VCE Dumps](#)