

1Z0-149^{Q&As}

Oracle Database 19c: Program with PL/SQL

Pass Oracle 1Z0-149 Exam with 100% Guarantee

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

<https://www.leads4pass.com/1z0-149.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 statements: Which is true?

```
Drop procedure calling_proc;

CREATE OR REPLACE PROCEDURE protected_proc
  ACCESSIBLE BY (calling_proc)
AS
BEGIN
  DBMS_OUTPUT.put_line('TEST1 : protected_proc');
END;
```

- A. It will result in a compilation error for protected_proc because calling_proc does not exist.
- B. It will result in a compilation error for protected_proc because calling_proc must be prefixed with the schema name.
- C. It will result in a successful compilation because objects referenced in an ACCESSIBLE BY clause are not checked at compile time.
- D. With adequate privileges, PROTECTED_PROC procedure can be called by other programs apart from CALLING_PROC.

Correct Answer: A

QUESTION 2

Examine these statements:

```
CREATE TABLE bank_activity
(transaction_id NUMBER,
 transaction_date TIMESTAMP);

CREATE TABLE bank_transaction
(transaction_id NUMBER,
 withdrawal_amount NUMBER);

CREATE OR REPLACE PROCEDURE bank_activity_proc
(p_id NUMBER)
AS
pragma autonomous_transaction;
BEGIN
    INSERT INTO bank_activity VALUES
    (p_id, sysdate);
    COMMIT;
END;
/

CREATE OR REPLACE PROCEDURE bank_transaction_proc
(p_id NUMBER,
 p_amount NUMBER)
AS
BEGIN
    INSERT INTO bank_transaction VALUES
    (p_id, p_amount);
    bank_activity_proc(p_id);
END;
/

EXECUTE bank_transaction_proc(200, 500);
ROLLBACK;
```

Which two are true? (Choose two.)

- A. Neither table will have a row inserted and committed.
- B. Both tables will have a row inserted and committed.
- C. The transaction for the bank_activity_proc is independent of the bank_transaction_proc.
- D. The bank_activity_proc will not compile because of the commit.
- E. Only one table will have a row inserted and committed.

Correct Answer: CE

QUESTION 3

Which two are true about collections and RECORD types? (Choose two.)

- A. A variable of RECORD type can contain fields of another RECORD type or any collection type.
- B. Only associative arrays and nested tables can have elements of RECORD type.
- C. All collections and RECORD types can be defined in PL/SQL blocks, packages, or at the schema level.
- D. Collections and RECORD types are always dense.
- E. All collections and RECORD types can be stored in table columns.
- F. VARRAYS, nested tables and each field in %ROWTYPE type variables have a default value of null.

Correct Answer: BF

QUESTION 4

Examine the structure of the ora1.depts table:

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

Now, examine these statements issued by user ora1 which execute successfully:

Create or replace view dep_vu as select * from depts;

Alter table depts add dep_email varchar2(20);

Finally, examine this block of code executed by user ora1:

```
set serveroutput on

declare
x number;
begin
SELECT count(*)
into x colCount
FROM all_tab_columns
WHERE table_name = 'DEP_VU' and
      owner='ORA1';
dbms_output.put_line(x);
end;
/
```

Which is true?

- A. DEP_VU must be manually recompiled to successfully run this code.
- B. It will run successfully producing a result of 4.
- C. It will result in an error because table depts has been altered.
- D. It will run successfully producing a result of 5.

Correct Answer: B

QUESTION 5

Examine these statements which execute successfully:

```
CREATE TABLE t (a INT, b INT, c INT INVISIBLE);
INSERT INTO t (a, b, c) VALUES (1, 2, 3);
COMMIT;
```

Which anonymous block executes successfully?

A.

```
DECLARE
  t_rec t%ROWTYPE;
BEGIN
  t_rec.c := t_rec.a;
  SELECT * INTO t_rec FROM t WHERE ROWNUM < 2;
  DBMS_OUTPUT.PUT_LINE('c = ' || t_rec.c);
END;
```

B.

```
DECLARE
  t_rec t%ROWTYPE;
BEGIN
  t_rec.a := t_rec.b;
  SELECT * INTO t_rec FROM t WHERE ROWNUM < 2;
  DBMS_OUTPUT.PUT_LINE('a = ' || t_rec.a);
END;
```

C.

```
DECLARE
  t_rec t%ROWTYPE;
BEGIN
  t_rec.b := t_rec.c;
  SELECT * INTO t_rec FROM t WHERE ROWNUM < 2;
  DBMS_OUTPUT.PUT_LINE('b = ' || t_rec.b);
END;
```

D.

```
DECLARE
  t_rec t%ROWTYPE;
BEGIN
  t_rec.c := NULL;
  SELECT * INTO t_rec FROM t WHERE ROWNUM < 2;
  DBMS_OUTPUT.PUT_LINE('c = ' || t_rec.c);
END;
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: B