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.passapply.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

Which codes executes successfully?

A. CREATE PACKAGE pkg AS TYPE rec_typ IS RECORD (price NUMBER, inc_pct NUMBER); PROCEDURE calc_price (price_rec IN OUT rec_typ); END pkg; / CREATE PACAKGE BODY pkg AS PROCEDURE calc_price (price_rec IN OUT rec_typ) AS BEGIN price_rec.price := price_rec.price + (price_rec.price * price_rec.inc_pct)/100; END calc_price; END pkg; / DECLARE 1_rec pkg. rec_typ; BEGIN 1_rec.price :=100; 1_rec.inc_pct :=50; EXECUTE IMMEDIATE 'BEGIN pkg. calc_price (:rec); END;' USING IN OUT 1_rec; END;

B. CREATE PACKAGE pkg AS TYPE rec_typ IS RECORD (price NUMBER, inc_pct NUMBER); END pkg; / CREATE PROCEDURE calc_price (price_rec IN OUT pkg. rec_typ) AS BEGIN price_rec.price := price_rec.price + (price_rec.price * price_rec.inc_pct)/100; END / DECLARE 1_rec pkg.rec_typ; BEGIN EXECUTE IMMEDIATE 'BEGIN calc_price (:rec); END;' USING IN OUT 1_rec (100, 50); END;

C. CREATE PACKAGE pkg AS TYPE rec_typ IS RECORD (price NUMBER, inc_pct NUMBER); END pkg; / CREATE PROCEDURE calc_price (price_rec IN OUT pkg. rec_typ) AS BEGIN price_rec.price := price_rec.price + (price_rec.price * price_rec.inc_pct)/100; END ; / DECLARE 1_rec pkg. rec_typ; BEGIN 1_rec.price :=100; 1_rec.inc_pct :=50; EXECUTE IMMEDIATE 'BEGIN calc_price (1_rec); END;'; END;

D. DECLARE TYPE rec_typ IS RECORD (price NUMBER, inc_pct NUMBER); 1_rec rec-typ; PROCEDURE calc_price (price_rec IN OUT rec_typ) AS BEGIN price_rec.price := price-rec.price+ (price_rec.price * price_rec.inc_pct)/100; END; BEGIN 1_rec.price :=100; 1_rec.inc_pct :=50; EXECUTE IMMEDIATE 'BEGIN calc_price (:rec); END;' USING IN OUT 1_rec; END;

Correct Answer: B

QUESTION 2

Select the correct statement regarding BEQUEATH CURRENT_USER.

A. If a view references a PL/SQL function then BEQUEATH CURRENT_USER allows the function to

B. execute with DBA privileges, regardless of the invoking user\\'s privileges.

C. The BEQUEATH CURRENT_USER clause allows invoker\\'s rights functions referenced in a view to execute with the privileges of the invoking user.

D. Any view calling a PL/SQL function with BEQUEATH CURRENT_USER in effect will execute with the privileges of the function owner.

E. With the BEQUEATH CURRENT_USER clause, a definer\\'s rights function referenced in a view executes with the privileges of the view owner, not the function owner.

Correct Answer: B

Reference: https://docs.oracle.com/database/121/DBSEG/dr_ir.htm#DBSEG558

QUESTION 3

Examine this code executed as SYS:

```
CREATE USER spider IDETIFIED BY spider DEFAULT TABLESPACE users QUOTA
UNLIMITED ON users;
CREATE ROLE dynamic_table_role;
GRANT CREATE TABLE TO dynamic_table_role;
GRANT CREATE SESSION, CREATE PROCEDURE TO spider;
GRANT dynamic_table_role TO spider WITH ADMIN OPTION;
ALTER USER spider DEFAULT ROLE ALL EXCEPT dynamic_table_role;
```

Examine this code executed as SPIDER and the error message received upon execution:

```
CREATE PROCEDURE dproc AS
BEGIN
    EXECUTE IMMEDIATE 'CREATE TABLE demo (id INTEGER)';
END;
/
SET ROLE dynamic_table_role;
EXEC dproc;

ERROR at line 1:
ORA-01031: insufficient privileges
ORA-06512: at "SPIDER.DPROC", line 4
ORA-06512: at line 1
```

What is the reason for this error?

A. The procedure needs to be granted the DYNAMIC_TABLE_ROLE role.

B. The EXECUTE IMMEDIATE clause is not supported with roles.

C. Privileges granted through roles are never in effect when running definer\'s rights procedures.

D. The user SPIDER needs to be granted the CREATE TABLE privilege and the procedure needs to be granted the DYNAMIC_TABLE_ROLE.

Correct Answer: C

QUESTION 4

You issue the following command to create the PRINT_MEDIA table.

CREATE TABLE print_media

(product_id NUMBER(3),

ad_sourcetext CLOB,

ad_photo BLOB);

Evaluate the following INSERT statements:

INSERT INTO print_media VALUES (1, empty_clob(),empty_blob());

INSERT INTO print_media VALUES (2,\\'This is a One Line Story\\',null);

INSERT INTO print_media VALUES (3,\\'This is another One Line Story\\',empty_blob());

INSERT INTO print_media VALUES (4,empty_clob(),to_blob(\\'This is new Story\\'));

Which of the above INSERT statements are valid?

A. Only the first statement is valid.

B. All the statements are valid.

C. Only the first and fourth statements are valid.

D. Only the first and second statements are valid.

E. Only the first, second and third statements are valid.

Correct Answer: E

QUESTION 5

Which guidelines should be considered when designing and using cursors in a PL/SQL block? ? (Choose all that apply.)

A. When fetching from a cursor, fetch into a record.

B. Use parameters with cursors so that the result set for the cursor is not tied to a specific variable in a program.

C. Use the %NOTFOUND attribute in combination with the SELECT INTO statement to check for non existent values.

D. Whenever possible, explicitly declare the cursor and use the OPEN, FETCH and CLOSE statements to manipulate the cursor instead of using cursor FOR loop.

E. When using data manipulation language statements, (DML) reference a SQL cursor attribute immediately after the DML statement executes in the same block.

Correct Answer: ABE