DUMPSBOSS.

Oracle Database 19c: Program with PL/SQL

Oracle 1z0-149

Version Demo

Total Demo Questions: 10

Total Premium Questions: 65

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

QUESTION NO: 1

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.

ANSWER: B F

QUESTION NO: 2

Which two blocks of code display a numerical zero? (Choose two.)

A)

```
CREATE OR REPLACE PROCEDURE calc_price IS
 price NUMBER := 0;
BEGIN
  DECLARE
    price NUMBER;
  BEGIN
    price := calc_price.price;
    DBMS_OUTPUT.PUT_LINE(price);
  END;
END;
/
BEGIN
  calc_price;
END;
/
```

B)

```
<<outer>>
DECLARE
  price NUMBER := 0;
  PROCEDURE calc_price AS
    BEGIN
      DBMS_OUTPUT.PUT_LINE(price);
  END;
BEGIN
  calc_price;
END;
/
```

C)

```
<<outer>>
<<inner>>
DECLARE
  price NUMBER := 0;
BEGIN
<<inner>>
  DECLARE
    price NUMBER := NULL;
  BEGIN
    price := inner.price;
    DBMS_OUTPUT.PUT_LINE(price);
  END;
END;
/
```

D)

```
<<outer>>
DECLARE
  price NUMBER;
BEGIN
<<inner>
  DECLARE
    price NUMBER;
  BEGIN
    price := 0;
  END;
  DBMS_OUTPUT.PUT_LINE(price);
END;
/
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: A B

QUESTION NO: 3

SERVEROUTPUT is enabled.

Which code block will display the values from 1 to 10 in descending order?

A)

```
BEGIN
FOR i IN REVERSE 10..1 LOOP
DBMS_OUTPUT.PUT_LINE(i);
END LOOP;
END;
```

B)

```
DECLARE
i NUMBER;
BEGIN
i:=10;
FOR i IN 1..10 LOOP
i:=i-1;
DBMS_OUTPUT.PUT_LINE(i);
END LOOP;
END;
/
```

C)

```
BEGIN
FOR i IN 10..1 LOOP
DBMS_OUTPUT.PUT_LINE(i);
END LOOP;
END;
/
```

D)

```
BEGIN
FOR i IN REVERSE 1..10 LOOP
DBMS_OUTPUT.PUT_LINE(i);
END LOOP;
END;
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: D

QUESTION NO: 4

Which block of code displays the error message “Incorrect price value”?

A)

```
DECLARE
  price CONSTANT NUMBER(4) := 10000;
BEGIN
  NULL;
EXCEPTION
  WHEN VALUE_ERROR THEN
    DBMS_OUTPUT.PUT_LINE('Incorrect price value');
END;
/
```

B)

```
BEGIN
DECLARE
  price CONSTANT NUMBER(4) := 50000;
BEGIN
  NULL;
END;
EXCEPTION
  WHEN VALUE_ERROR THEN
    DBMS_OUTPUT.PUT_LINE('Incorrect price value');
END;
/
```

C)

```
BEGIN
DECLARE
  error_detected EXCEPTION;
  PRAGMA EXCEPTION_INIT(error_detected, -2001);
  price CONSTANT NUMBER(4) := 10000;
BEGIN
  NULL;
END;
EXCEPTION
  WHEN error_detected THEN
    DBMS_OUTPUT.PUT_LINE('Incorrect price value');
END;
/
```

D)

```
DECLARE
  price CONSTANT NUMBER(4) := 10000;
BEGIN
  NULL;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Incorrect price value');
END;
/
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: B

QUESTION NO: 5

Which two are true about exception handling? (Choose two.)

A. Internally defined exceptions can be handled only by the OTHERS exception handler.

B. All declared exceptions are raised implicitly by the runtime system.

C. User-defined exceptions can be defined in the declarative part of any PL/SQL anonymous block, subprogram, or package.

D. Only predefined exceptions and user-defined exceptions can have a user-declared name associated with them.

E. Predefined exceptions are globally declared in the standard package.

ANSWER: C E

QUESTION NO: 6

Which is the correct method to implement a local subprogram in an anonymous block?

A)

```
DECLARE
fnam VARCHAR2(10) := 'King';
lnam VARCHAR2(12) := 'Cobra';
BEGIN
FUNCTION full_name ( A VARCHAR2, B VARCHAR2) RETURN VARCHAR2 AS
C VARCHAR2(20);
BEGIN
C := A || ';' || B;
RETURN C;
END full_name;
DBMS_OUTPUT.PUT_LINE(full_name (fnam, lnam));
END;
```

B)

```
BEGIN
DECLARE
fnam VARCHAR2(10) := 'King';
lnam VARCHAR2(12) := 'Cobra';
FUNCTION full_name ( A  VARCHAR2, B  VARCHAR2) RETURN VARCHAR2 AS
C VARCHAR2(20);
BEGIN
C := A || ';' || B;
RETURN C;
END full_name;
BEGIN
DBMS_OUTPUT.PUT_LINE('And the output is...');
END;
DBMS_OUTPUT.PUT_LINE(full_name (fnam, lnam));
END;
```

C)

```
BEGIN
DECLARE
fnam VARCHAR2(10) := 'King';
lnam VARCHAR2(12) := 'Cobra';
BEGIN
FUNCTION full_name ( A  VARCHAR2, B  VARCHAR2) RETURN VARCHAR2 AS
C VARCHAR2(20);
BEGIN
C := A || ';' || B;
RETURN C;
END full_name;
DBMS_OUTPUT.PUT_LINE('And the output is...');
END;
DBMS_OUTPUT.PUT_LINE(full_name (fnam, lnam));
END;
```

D)

```
DECLARE
fnam VARCHAR2(10) := 'King';
lnam VARCHAR2(12) := 'Cobra';
FUNCTION full_name ( A  VARCHAR2, B  VARCHAR2) RETURN VARCHAR2 AS
C VARCHAR2(20);
BEGIN
C := A || ';' || B  ;
RETURN C;
END full_name;
BEGIN
DBMS_OUTPUT.PUT_LINE(full_name (fnam, lnam));
END;
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: A

QUESTION NO: 7

Which is true about counter variables in a FOR loop?

A. It must explicitly be declared.

B. It can be modified in the body of the loop.

C. It cannot be NULL.

D. It is accessible outside the body of the loop.

ANSWER: C

QUESTION NO: 8

Sequence S and table PRODUCTS exist in your schema.

Examine the table description:

```
DESC products
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PDT_ID                                    NOT NULL NUMBER(3)
 PDT_NAME                                           VARCHAR2(25)
 PRICE                                              NUMBER(8,2)
```

Now, examine this block of code:

```
1 CREATE OR REPLACE PROCEDURE report(UPPER(pdt_name_in)IN products.pdt_name%TYPE) IS
2  current_price NUMBER := 155.55;
3  new_price NUMBER(10,2) := ROUND(current_price + (current_price * .05));
4  compare_value  VARCHAR2(20);
5  BEGIN
6    compare_value := DECODE( new_price,  1001,  'Above 1000', 'Below 1000');
7    DBMS_OUTPUT.PUT_LINE(s.NEXTVAL || ' ' || UPPER('New Price')|| ' ' || TO_CHAR(new_price));
8    DBMS_OUTPUT.PUT_LINE(s.CURRVAL+1 || ' ' || UPPER('New Price')|| ' ' || new_price);
9 END;
/
```

Which two lines each result in a compilation error? (Choose two.)

A. line 1

B. line 6

C. line 8

D. line 2

E. line 3

F. line 7

ANSWER: A B

QUESTION NO: 9

Which three are true about DDL triggers? (Choose three.)

A. They cannot include the WHEN clause.

B. They must be created in an enabled state.

C. They can be fired when a table is truncated.

D. They fire only when a DDL statement is executed by the owner of the trigger.

E. They can be fired either before or after a DDL statement executes.

F. They can be fired when a privilege is granted to a user.

G. They must be created in a disabled state.

ANSWER: C D E

QUESTION NO: 10

Examine this table in the SH schema:

```
DESC products

 Name                                 Null?    Type
 ------------------------------------ -------- ----------------------
 PDT_ID                               NOT NULL NUMBER
 PDT_NAME                                      VARCHAR2(10)
 PRICE                                         NUMBER
```

User SH executes this code:

```
DECLARE
  v_price NUMBER := 1000;
  v_pdt_name VARCHAR2(15);
BEGIN
   SELECT pdt_name INTO v_pdt_name
     FROM products
     WHERE price = v_price;

  ---placeholder

END;
/
```

The program must terminate with a user-defined message and no rows displayed if more than one product's price is 1000.

With which option must “---placeholder” be replaced?

A)

```
DBMS_OUTPUT.PUT_LINE ('Product name is :'||v_pdt_name);

    EXCEPTION WHEN others THEN
      DBMS_OUTPUT.PUT_LINE ('More than one row found');
```

B)

```
IF SQL%ROWCOUNT > 1 THEN
    DBMS_OUTPUT.PUT_LINE ('More than one row found');
 ELSE
    DBMS_OUTPUT.PUT_LINE ('Product name is :'||v_pdt_name);
END IF;
```

C)

```
EXCEPTION WHEN too_many_rows THEN
  DBMS_OUTPUT.PUT_LINE ('More than one row found');

DBMS_OUTPUT.PUT_LINE ('Product name is :'||v_pdt_name);
```

D)

```
IF too_many_rows THEN
   DBMS_OUTPUT.PUT_LINE ('More than one row found');
ELSE
   DBMS_OUTPUT.PUT_LINE ('Product name is :'||v_pdt_name);
END IF;
```

E)

```
EXCEPTION WHEN OTHERS THEN
       RAISE too_many_rows;
DBMS_OUTPUT.PUT_LINE ('Product name is :'||v_pdt_name);
```

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

ANSWER: A