

Oracle.1z1-149.v2023-09-18.q23

□□□□:	1z1-149
□□□□:	Oracle Database 19c: Program with PL/SQL
□□□:	Oracle
□□ □□ □□□:	23
□□:	v2023-09-18
# □□ □:	411
# □□ □□□:	230
https://www.krdump.com/Oracle.1z1-149.v2023-09-18.q23.html	

NEW QUESTION: 1

CBAC(□□ □□ □□□ □□)□ □□□□ □□□ □ □□□ □□□□□? (3□□ □□□□□.)

- A. CBAC □□□ □□□□□□ □□□ □□□ □□ □□□□□ □□ □□□□ □□□ □□ □□□□ □□□□ □□□□ □□□ □□□□□.
- B. CBAC □□□ □□□□□ □□ □□□ □□□□ □□□□ □□□ □□□ □ □□□□□.
- C. □□ □□□ □□□□ □□□□ CBAC□ DELEGATE □□□ □□□ □ □□□□□.
- D. CBAC□□ ADMIN □ DELEGATE □□□ □□□ □□□□□ □□ □□□ □ □□□□□.
- E. CBAC□ □□□□ □□□□□□ □□□ PL/SQL □□ □□ □□□□□□ □□□ □ □□□□□.
- F. □□□□ □□□ □□□□ □□ CBAC□ □□□ □ □□□□□.
- G. CBAC□ □□□□ □□□□□□ □□□ PL/SQL □□, □□□□ □□ □□□□ □□□ □ □□ □□□.

Answer: B,D,G ([LEAVE A REPLY](#))

NEW QUESTION: 2

SH.PRODUCTS □□□□ □□□□□.

DESC products		
Name	Null?	Type
PDT_ID	NOT NULL	NUMBER
PDT_NAME		VARCHAR2(10)
PRICE		NUMBER

PDT_ID = 1□ □□ SH.PRODUCTS□ □□□□□.

□□ □ □□□ SH□ □□ □□□ □□□ □□□□□.

```

CREATE OR REPLACE PROCEDURE price_divide (p_id NUMBER, p_val NUMBER) IS
  v_price NUMBER;
BEGIN
  SELECT price INTO v_price FROM products WHERE pdt_id = p_id;
  BEGIN
    DBMS_OUTPUT.PUT_LINE(v_price/p_val);
  EXCEPTION
    WHEN ZERO_DIVIDE THEN
      DBMS_OUTPUT.PUT_LINE('Error in inner block');
  END;
EXCEPTION
  WHEN ZERO_DIVIDE THEN
    DBMS_OUTPUT.PUT('Error in outer block');
END;
/
Procedure created.

```

□□ □□□□ □□□□□□□□.

```

BEGIN
  price_divide(1,0);
EXCEPTION WHEN ZERO_DIVIDE THEN
  DBMS_OUTPUT.PUT('Error in calling block');
END;
/

```

□□□ SH□ □□□ □ □□□□ □□ □□□□ □□□□□?

- A. □□ □□ □□
- B. □□ □□ □□ □□ □□ □□ □□ □□ □□
- C. □□ □□ □□ □□ □□ □□
- D. □□ □□ □□ □□ □□ □□

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 3

□□□□ □□□ □ □□ □ □□ □□□ □□□□ □□□□□? (3□□ □□□□□.)

- A. □□ □
- B. □□ □□
- C. □□ □
- D. □□
- E. □□

<□>: DDL

- F. □□□ □□

Answer: B,D,E ([LEAVE A REPLY](#))

NEW QUESTION: 4

EXIT □ CONTINUE □□ □□ □□□□ □□ □□?

- A. WHEN □□□ □□□ □□□.
- B. □□ □□□ □□□ □□□ □□□□.
- C. □□ □□□ □□□□ □□□ □ □□□□.
- D. □□□ □□□□ □□□.

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 5

PLSCOPE_SETTINGS □□□□□ □□ □□□□ □□ □□?

- A. □□□□ PL/SQL □□□ □□ □□ □□□ □□□□ □ □□□ □ □□□□.
- B. □□□□□ □□□□ □ □□ □□□□ □□ □□□ □□ □□ □□□ □ □□□□.
- C. □□□□ PL/SQL □□□ □□ □□□□ □□□ □□□□ □ □□□ □ □□□□.
- D. Oracle 12c□□□□ □ □□ □□□□ □□□□.

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 6

□□ □□□ □□□□□□.

```

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;

```

□ □ □ □ □ □ □ □ □ □ ? (□ □ □ □ □ □ □ □ □ □ .)

A. □ .

B. □ .

C. bank_activity_proc □ □ □ □ □ □ □ □ □ □ bank_transaction_proc □ □ □ □ □ □ □ □ □ □ .


```

1 DECLARE
2   v_join_date DATE := SYSDATE - 10;
3   v_flag BOOLEAN NOT NULL DEFAULT TRUE;
4   v_char VARCHAR2 := NULL;
5   v_bonus_pct CONSTANT REAL(2) := 8.25;
6   v_zip_code VARCHAR2(80) := SUBSTR('Oracle Corporation', 24, 0);

```

□□ □ □□□ □□□□ □□ □□□□□?

- A. 6□
- B. 3□
- C. 5□
- D. 2□
- E. 4□

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 10

□□□ □□ □□□□ □□ □□□□ □□ □□ □□□□□?

- A. □□ □□□ □ □□□ □□□□ □□□.
- B. □□ □□ □□□□ □ □□ □□□□ □□□.
- C. □□ □□□ □□□ □ □□□ □□□□ □□□.
- D. □□□□ □□□ □ □□□□.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 11

□□ □□□□ □□ □□□□ □□□□□□.

CURSOR c1 IS

```

SELECT * FROM products
FOR UPDATE OF price;

```

UPDATE products

```

SET price = price * 1.05
WHERE CURRENT OF c1;

```

□ □ □□ □□ □□□□□? (□ □□□ □□□□□.)

- A. FOR UPDATE OF □□ WHERE CURRENT OF □□ □□□ □□ □□□□ □□□□ □□□□ □□□□ □□□□.

- B. WHERE CURRENT OF emp_emp_id FOR UPDATE emp_emp_id emp_emp_id emp_emp_id.
- C. OPEN c1 emp_emp_id emp_emp_id COMMIT emp_emp_id ROLLBACK emp_emp_id emp_emp_id emp_emp_id.
- D. PRODUCTS emp_emp_id PRICE emp_emp_id emp_emp_id emp_emp_id.
- E. OPEN c1 emp_emp_id PRODUCTS emp_emp_id PRICE emp_emp_id emp_emp_id emp_emp_id.

Answer: B,C ([LEAVE A REPLY](#))

NEW QUESTION: 12

emp_emp_id emp_emp_id HR.EMPLOYEES emp_emp_id FIRST_NAME emp_emp_id EMPLOYEE_ID emp_emp_id emp_emp_id emp_emp_id.
 EMPLOYEE_ID emp_emp_id 201 emp_emp_id emp_emp_id emp_emp_id.
 emp_emp_id HR emp_emp_id emp_emp_id emp_emp_id emp_emp_id.

```

CREATE PACKAGE helper AUTHID DEFINER ACCESSIBLE BY (api) IS
  PROCEDURE h1;
END;
/

CREATE PACKAGE BODY helper IS
  PROCEDURE h1 IS
    emp_name employees.first_name%TYPE;
  BEGIN
    SELECT first_name INTO emp_name FROM hr.employees WHERE employee_id=201;
    DBMS_OUTPUT.PUT_LINE(emp_name);
  END;
END;
/

CREATE PACKAGE api AUTHID CURRENT_USER IS
  PROCEDURE p1;
END;
/

CREATE PACKAGE BODY api IS
  PROCEDURE p1 IS
  BEGIN
    helper.h1;
  END;
END;
/

```

- EXECUTE emp_emp_id HR.API emp_emp_id HR.HELPER emp_emp_id emp_emp_id SH emp_emp_id emp_emp_id emp_emp_id.
 emp_emp_id emp_emp_id emp_emp_id emp_emp_id emp_emp_id? (emp_emp_id emp_emp_id emp_emp_id.)
- A. HR emp_emp_id emp_emp_id HR.HELPER.H1 emp_emp_id emp_emp_id.
 - B. SH emp_emp_id emp_emp_id HR.API.P1 emp_emp_id emp_emp_id.
 - C. SH emp_emp_id emp_emp_id HR.HELPER.H1 emp_emp_id emp_emp_id.
 - D. HR emp_emp_id emp_emp_id HR.API.P1 emp_emp_id emp_emp_id.

E. SH HR.HELPER.H1 API.H1 .

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 13

PL/SQL ? (3 .)

- A. _#
- B.
- C. _
- D. v_f
- E. #_
- F. Number_of_days_between_March_and_April
- G. 1to7

Answer: A,B,D ([LEAVE A REPLY](#))

NEW QUESTION: 14

? (3 .)

- A. .
- B. .
- C. .
- D. .
- E. .
- F. .
- G. .

Answer: C,E,F ([LEAVE A REPLY](#))

NEW QUESTION: 15

EMPLOYEES .

EMP_ID	NUMBER(10)
FIRST_NAME	VARCHAR2(30)
LAST_NAME	VARCHAR2(30)
SALARY	NUMBER(8,2)

.

F. `CREATE PACKAGE BODY` `PACKAGE BODY`.

G. `min_bal` `min_bal` `min_bal` `min_bal`.

Answer: A,B,C ([LEAVE A REPLY](#))

NEW QUESTION: 20

"`DECLARE` `price` `CONSTANT` `NUMBER`(4) `:=` 10000;" `DECLARE` `price` `CONSTANT` `NUMBER`(4) `:=` 10000;

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
    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;
/
```


A.

```
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;
```

B.

```
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;
```

C.

```
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;
```



D.

```
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;
```



Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 22

□□□ □□ □□□ □□ □□□ □ □□□ □□□□□? (3□□ □□□□□.)

- A. CONNECT BY □ START WITH □□□ □□□ □ □□□□.
- B. □□ □□□ □□□ □□□□.
- C. □□ □□□ □□□□ □□□ □ □□□□.

- D. ORDER BY GROUP BY .
- E. SELECT .
- F. .
- G. .

Answer: A,D,E ([LEAVE A REPLY](#))

NEW QUESTION: 23

OR REPLACE
? (.)

- A. .
- B. .
- C. .
- D. .
- E. .

Answer: A,C ([LEAVE A REPLY](#))

1z1-149 DumpTop 1z1-149 !

DumpTop **1z1-149** , DumpTop 1z1-149 . DumpTop 1z1-149 . <https://www.dumptop.com/Oracle/1z1-149-dump.html> (67 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)