

Exemplul 5.1

```
BEGIN
  DELETE FROM tip_plata
  WHERE id_tip_plata NOT IN
        (SELECT id_tip_plata FROM facturi);

  -- cursor deschis?

  DBMS_OUTPUT.PUT_LINE('Cursor deschis');

  DBMS_OUTPUT.PUT_LINE('Cursor inchis');

  -- a gasit linii?

  DBMS_OUTPUT.PUT_LINE('A fost gasita cel putin o linie');

  DBMS_OUTPUT.PUT_LINE('Nu a fost gasita nicio linie');

  -- cate linii a gasit
  DBMS_OUTPUT.PUT_LINE('Au fost sterse ' ||
        || ' linii');
END;
```

Exemplul 5.4

```
DECLARE

  SELECT id_categorie, denumire
  FROM   categorii
  WHERE  id_parinte IS NULL;

  v_id_categorie categorii.id_categorie%TYPE;
  v_denumire      categorii.denumire%TYPE;
BEGIN

  DBMS_OUTPUT.PUT_LINE(v_id_categorie || ' ' || v_denumire);

END;
```

Exemplul 5.5

```
DECLARE

    SELECT *
    FROM    categorii
    WHERE   id_parinte IS NULL;

BEGIN

    DBMS_OUTPUT.PUT_LINE(v_categorii.id_categorie || ' ' ||
        v_categorii.denumire);

END;
```

Exemplul 5.6

```
DECLARE
    TYPE tab_imb IS TABLE OF categorii%ROWTYPE;
    v_categorii tab_imb;
    CURSOR c IS
        SELECT * FROM    categorii
        WHERE   id_parinte IS NULL;
BEGIN

    DBMS_OUTPUT.PUT_LINE(
        || v_categorii(i).denumire);

END;
```

Exemplul 5.7

```
--limitarea numarului de linii incarcate
DECLARE
  TYPE tab_imb IS TABLE OF produse.denumire%TYPE;
  v_produce tab_imb;
  v_denumire produse.denumire%TYPE;

  CURSOR c1 IS
    SELECT denumire
    FROM   produse

  CURSOR c2 IS
    SELECT denumire
    FROM   produse;

BEGIN
  OPEN c1;
  LOOP
    FETCH c1 INTO v_denumire;
    EXIT WHEN c1%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(v_denumire);
  END LOOP;
  CLOSE c1;

  DBMS_OUTPUT.PUT_LINE('-----');

  FOR i IN 1..v_produce.LAST LOOP
    DBMS_OUTPUT.PUT_LINE(v_produce(i));
  END LOOP;
END;
```

Exemplul 5.8

```
DECLARE
  CURSOR c IS
    SELECT *
    FROM   categorii
    WHERE  id_parinte IS NULL;

BEGIN

  DBMS_OUTPUT.PUT_LINE(

END;
```

Exemplul 5.9

```

BEGIN

    DBMS_OUTPUT.PUT_LINE(i.id_categorie || ' '
        || i.denumire);

END;
```

Exemplul 5.10

```

DECLARE
    CURSOR categ IS SELECT id_categorie, denumire
                        FROM   categorii
                        WHERE  nivel = 5;

    SELECT MAX(p.denumire), SUM(cantitate)
    FROM   produse p, facturi_produce fp
    WHERE  v_categ = p.id_categorie
    AND    p.id_produc = fp.id_produc
    GROUP BY p.id_produc
    ORDER BY 1,2 desc;
    c_denumire categorii.denumire%TYPE;
    c_id categorii.id_categorie%TYPE;
    p_denumire produse.denumire%TYPE;
    p_cantitate NUMBER;
BEGIN

    DBMS_OUTPUT.PUT_LINE(
    DBMS_OUTPUT.PUT_LINE('-----');

    DBMS_OUTPUT.PUT_LINE(
    );

    DBMS_OUTPUT.PUT_LINE('Nu are produse vandute!');
END IF;

    DBMS_OUTPUT.NEW_LINE;

END;
```

Exemplul 5.12

```
DECLARE
  CURSOR c IS

  FROM   produse
  WHERE  id_categorie IN
        (SELECT id_categorie
         FROM   categorii
         WHERE  denumire = 'Placi de retea Wireless')

BEGIN

  UPDATE  produse
  SET     pret_unitar = pret_unitar*0.95

  -- permanentizare si eliberare blocari

END;
```

Exemplul 5.13

```
-- utilizare ROWID in loc de CURRENT OF
DECLARE
  CURSOR c IS

  FROM   produse
  WHERE  id_categorie IN
        (SELECT id_categorie
         FROM   categorii
         WHERE  denumire = 'Placi de retea Wireless')

BEGIN

  UPDATE  produse
  SET     pret_unitar = pret_unitar*0.95

  COMMIT;
END;
```

Exemplul 5.14

```
DECLARE

v_optiune NUMBER(1) := &p_optiune;

BEGIN

SELECT *
FROM produse p
WHERE EXISTS (SELECT 1
              FROM facturi_produce pf, facturi f
              WHERE p.id_produc = pf.id_produc
              AND pf.id_factura = f.id_factura
              AND TO_CHAR(data, 'q') = 1);

SELECT *
FROM produse p
WHERE id_produc IN
      (SELECT id_produc
       FROM facturi_produce pf, facturi f
       WHERE pf.id_factura = f.id_factura
       AND TO_CHAR(data, 'q') = 2);

SELECT DISTINCT p.*
FROM produse p, facturi_produce pf, facturi f
WHERE p.id_produc = pf.id_produc
AND pf.id_factura = f.id_factura
AND TO_CHAR(data, 'q') = 3;

SELECT *
FROM produse p
WHERE id_produc IN (SELECT id_produc
                   FROM facturi_produce);

DBMS_OUTPUT.PUT_LINE(i.id_produc||' ' ||i.denumire);
DBMS_OUTPUT.PUT_LINE('Nr produse vandute: '||
                    c%ROWCOUNT);

END;
```

Exemplul 5.15

```
--SQL DINAMIC
DECLARE

    c tip_cursor;
    v_optiune NUMBER(1) := &p_optiune;
    i produse%ROWTYPE;
BEGIN

    FROM produse p, facturi_produse pf, facturi f
    WHERE p.id_produs = pf.id_produs
    AND    pf.id_factura = f.id_factura

    LOOP
        FETCH c INTO i;
        DBMS_OUTPUT.PUT_LINE(i.id_produs||' ' ||i.denumire);
        EXIT WHEN c%NOTFOUND;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('Nr produse vandute: '||
                        c%ROWCOUNT);

    CLOSE c;
END;
/

OPEN c FOR
'SELECT DISTINCT p.*
FROM produse p, facturi_produse pf, facturi f
WHERE p.id_produs = pf.id_produs
AND    pf.id_factura = f.id_factura
AND    TO_CHAR(data, 'q') =
```

Exemplul 5.16

```
DECLARE

    FROM    produse p, facturi_produce fp
    WHERE   c.id_categorie = p.id_categorie
    AND     p.id_produc = fp.id_produc
    GROUP BY p.id_produc

    FROM
    WHERE   nivel = 5;

    c_denumire categorii.denumire%TYPE;

    TYPE tab_prod IS TABLE OF produse.denumire%TYPE
        INDEX BY BINARY_INTEGER;
    v_prod tab_prod;

BEGIN
    OPEN categ;
    LOOP

        EXIT WHEN categ%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(c_denumire);
        DBMS_OUTPUT.PUT_LINE('-----');

        IF                                THEN
            DBMS_OUTPUT.PUT_LINE('Nu are produse vandute!');
        ELSE
            FOR i IN v_prod.FIRST..v_prod.LAST LOOP
                DBMS_OUTPUT.PUT_LINE(i|| ' ' || v_prod(i));
            END LOOP;
        END IF;
        DBMS_OUTPUT.NEW_LINE;
    END LOOP;

END;
```