Exemplul 4.6

```
DECLARE
  v_categ
  v_categ2 categorii%ROWTYPE;
  v_categ_modific
  v_categ_null categorii%ROWTYPE;
BEGIN
  v_categ.denumire := 'Categorie noua';
  v_categ.nivel :=1;
  SELECT MAX(id_categorie)+1 INTO v_categ.id_categorie
  FROM   categorii;

  INSERT INTO categorii



  SELECT *
  FROM   categorii
  WHERE  id_categorie=


 DBMS_OUTPUT.PUT_LINE ('Ati inserat: '||
    v_categ2.id_categorie || ' ' || v_categ2.denumire ||
    '  '|| v_categ2.nivel || ' ' ||



 v_categ_modific

 v_categ_modific.id_categorie := v_categ.id_categorie + 1;

 UPDATE categorii
 SET
 WHERE id_categorie= v_categ.id_categorie;

 SELECT * INTO v_categ2
 FROM   categorii
 WHERE  id_categorie= v_categ_modific.id_categorie;

 DBMS_OUTPUT.PUT_LINE ('Ati modificat in: '||
   v_categ_modific.id_categorie || ' ' ||
   v_categ_modific.denumire || '  '||
   v_categ_modific.nivel || ' ' ||
   NVL(v_categ_modific.id_parinte,0));

 v_categ2

 DELETE FROM categorii
 WHERE  id_categorie= v_categ_modific.id_categorie
 RETURNING id_categorie, denumire, nivel, id_parinte
 INTO

 DBMS_OUTPUT.PUT_LINE ('Ati sters linia: '||
   v_categ2.id_categorie || ' ' || v_categ2.denumire ||
   '  '|| v_categ2.nivel || ' ' ||
   NVL(v_categ2.id_parinte,0));

END;
```

Exemplul 4.9

```
DECLARE
                                         tip_plata.descriere%TYPE




BEGIN
  -- atribuire valori
DELETE FROM tip_plata
WHERE  id_tip_plata NOT IN (SELECT id_tip_plata FROM facturi)



 --parcurgere
 DBMS_OUTPUT.PUT_LINE('Tabloul are ' ||
                     ||' elemente:');

 FOR i IN                         LOOP
      DBMS_OUTPUT.PUT_LINE(        );
 END LOOP;
ROLLBACK;
END;
```

Exemplul 4.10

```
DECLARE
                           (        tip_plata.id_tip_plata%TYPE,
                                    tip_plata.descriere%TYPE);






BEGIN
  -- atribuire valori
   DELETE FROM tip_plata
   WHERE  id_tip_plata NOT IN (SELECT id_tip_plata
                               FROM facturi)






  --parcurgere
  DBMS_OUTPUT.PUT_LINE('Tabloul are ' || t.COUNT
         ||' elemente:');
  FOR i IN t.FIRST..t.LAST LOOP

     DBMS_OUTPUT.PUT_LINE(              ||' '||              );
  END LOOP;
ROLLBACK;
END;
```

Exemplul 4.11

```
DECLARE






BEGIN
   -- initializare
   t('a') := ASCII('a');
   t('A') := ASCII('A');
   t('b') := ASCII('b');
   t('B') := ASCII('B');
   t('x') := ASCII('x');
   t('X') := ASCII('X');

   -- parcurgere










END;
```

Exemplul 4.12_b

```
DECLARE
  TYPE tab_imb IS TABLE OF NUMBER;






BEGIN
  -- atribuire valori


  FOR i IN 6..10 LOOP
    t(i):=i;
  END LOOP;
  --parcurgere
  DBMS_OUTPUT.PUT('Tabloul are ' || t.COUNT ||' elemente: ');
  FOR i IN t.FIRST..t.LAST LOOP
      DBMS_OUTPUT.PUT(t(i) || ' ');
  END LOOP;
  DBMS_OUTPUT.NEW_LINE;






END;
```

Exemplul 4.14

```
DECLARE
  -- tipul a fost definit la ex4.13


  v_id_categ   raion_grupe_imb.id_categorie%TYPE;
  v_den        raion_grupe_imb.denumire%TYPE;
BEGIN










  DBMS_OUTPUT.PUT_LINE(v_id_categ || ' ' || v_den);
  DBMS_OUTPUT.PUT_LINE('---------------------------');
  FOR i IN 1..v_grupe.LAST LOOP
     DBMS_OUTPUT.PUT_LINE(v_grupe(i));
  END LOOP;
END;
```

Exemplul 4.18

```
DECLARE
   TYPE t_imb IS TABLE OF NUMBER(2);
   t t_imb := t_imb();
   t1 t_imb := t_imb(1,2,1,3,3);
   t2 t_imb := t_imb(1,2,4,2);
   t3 t_imb := t_imb(1,2,4);
   t4 t_imb := t_imb(1,2,4);
   t5 t_imb := t_imb(1,2);

BEGIN
  -- IS EMPTY



      DBMS_OUTPUT.PUT_LINE('t nu are elemente');
   END IF;

  -- CARDINALITY
  DBMS_OUTPUT.PUT('t1 are '||                         ||
                  ' elemente: ');
  FOR i IN 1..t1.LAST LOOP
      DBMS_OUTPUT.PUT(t1(i)||' ');
  END LOOP;
  DBMS_OUTPUT.NEW_LINE;

  DBMS_OUTPUT.PUT('t2 are '|| CARDINALITY(t2) ||
                  ' elemente: ');
  FOR i IN 1..t2.LAST LOOP
      DBMS_OUTPUT.PUT(t2(i)||' ');
  END LOOP;
  DBMS_OUTPUT.NEW_LINE;

  -- SET



  DBMS_OUTPUT.PUT('t1 fara duplicate: ');
  FOR i IN 1..t.LAST LOOP
      DBMS_OUTPUT.PUT(t(i)||' ');
  END LOOP;
  DBMS_OUTPUT.NEW_LINE;

  -- MULTISET EXCEPT



  DBMS_OUTPUT.PUT('t1 minus t2: ');
    FOR i IN 1..t.LAST LOOP
      DBMS_OUTPUT.PUT(t(i)||' ');
  END LOOP;
  DBMS_OUTPUT.NEW_LINE;

  -- MULTISET UNION
  DBMS_OUTPUT.PUT('t1 union distinct t2: ');
  FOR i IN 1..t.LAST LOOP
      DBMS_OUTPUT.PUT(t(i)||' ');
  END LOOP;
  DBMS_OUTPUT.NEW_LINE;
  -- MULTISET INSERSECT



  DBMS_OUTPUT.PUT('t1 intersect distinct t2 : ');
    FOR i IN 1..t.LAST LOOP
      DBMS_OUTPUT.PUT(t(i)||' ');
  END LOOP;
  DBMS_OUTPUT.NEW_LINE;

  -- test egalitate



     DBMS_OUTPUT.PUT_LINE('t2 = t3');
  ELSE
     DBMS_OUTPUT.PUT_LINE('t2 <> t3');
  END IF;

  IF t3=t4 THEN
     DBMS_OUTPUT.PUT_LINE('t3 = t4');
  ELSE
     DBMS_OUTPUT.PUT_LINE('t3 <> t4');
  END IF;

  -- IN



     DBMS_OUTPUT.PUT_LINE('t4 in (t1,t2,t3)');
  ELSE
     DBMS_OUTPUT.PUT_LINE('t4 not in (t1,t2,t3)');
  END IF;

  -- IS A SET



      DBMS_OUTPUT.PUT_LINE('t4 este multime');
  ELSE
     DBMS_OUTPUT.PUT_LINE('t4 nu este multime');
  END IF;

  -- MEMBER OF



     DBMS_OUTPUT.PUT_LINE('2 este in t4');
  ELSE DBMS_OUTPUT.PUT_LINE('2 nu este in t4');
  END IF;

    -- SUBMULTISET OF



     DBMS_OUTPUT.PUT_LINE('t5 este inclus in t4');
  ELSE
     DBMS_OUTPUT.PUT_LINE('t5 nu este inclus in t4');
  END IF;
END;
```

Exemplul 4.25

```
DECLARE
   TYPE tip_vec IS VARRAY(3) OF NUMBER(4);
   v tip_vec := tip_vec(800, 900, 9999);




BEGIN




EXCEPTION
  WHEN eroare THEN
   nr_erori := SQL%BULK_EXCEPTIONS.COUNT;
   DBMS_OUTPUT.PUT_LINE('Numar comenzi esuate: ' || nr_erori);
   FOR i IN 1..nr_erori LOOP
      DBMS_OUTPUT.PUT_LINE('Eroare ' || i ||
           ' aparuta in timpul iteratiei ' ||
           SQL%BULK_EXCEPTIONS(i).ERROR_INDEX);
      DBMS_OUTPUT.PUT_LINE('Mesajul erorii: ' ||
          SQLERRM(-SQL%BULK_EXCEPTIONS(i).ERROR_CODE));
   END LOOP;
END;
```