```
DECLARE
  v categ categorii%ROWTYPE;
  v categ2 categorii%ROWTYPE;
  v categ modific v categ%ROWTYPE;
  v categ null categorii%ROWTYPE;
BEGIN
  v categ.denumire := 'Categorie noua';
  v cateq.nivel :=1;
  SELECT MAX(id categorie) +1 INTO v categ.id categorie
  FROM
         categorii;
  INSERT INTO categorii VALUES v categ;
  SELECT * INTO v categ2
         categorii
  FROM
         id_categorie= v_categ.id_categorie;
  WHERE
  DBMS OUTPUT.PUT LINE ('Ati inserat: '||
    v_categ2.id_categorie || ' ' || v_categ2.denumire ||
' '|| v_categ2.nivel || ' ' ||
    NVL(v categ2.id parinte,0));
 v categ modific := v categ;
 v categ modific.id categorie := v categ.id categorie + 1;
 UPDATE categorii
 SET ROW = v categ modific
 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 := v categ null;
 DELETE FROM categorii
 WHERE id categorie= v categ modific.id categorie
 RETURNING id categorie, denumire, nivel, id parinte
 INTO v categ2;
 DBMS OUTPUT.PUT LINE ('Ati sters linia: '||
   v categ2.id categorie || ' ' || v categ2.denumire ||
   ' '|| v categ2.nivel || ' ' ||
   NVL(v categ2.id parinte,0));
END;
```

```
DECLARE
  TYPE tab ind IS TABLE OF tip plata.descriere%TYPE
       INDEX BY PLS INTEGER;
       tab ind;
BEGIN
  -- atribuire valori
DELETE FROM tip plata
WHERE id tip plata NOT IN (SELECT id tip plata FROM facturi)
RETURNING descriere BULK COLLECT INTO t;
 --parcurgere
 DBMS OUTPUT.PUT LINE('Tabloul are ' || t.COUNT ||
     ' elemente: ');
FOR i IN t.FIRST..t.LAST LOOP
     DBMS OUTPUT.PUT LINE(t(i));
END LOOP;
ROLLBACK;
END;
```

```
DECLARE
  TYPE rec IS RECORD (id tip plata.id tip plata%TYPE,
                      den tip plata.descriere%TYPE);
  TYPE tab ind IS TABLE OF rec
      INDEX BY PLS INTEGER;
  t
      tab ind;
BEGIN
  -- atribuire valori
   DELETE FROM tip plata
   WHERE id tip plata NOT IN (SELECT id tip plata
                               FROM facturi)
  RETURNING id tip plata, descriere BULK COLLECT INTO t;
  --parcurgere
  DBMS OUTPUT.PUT LINE('Tabloul are ' || t.COUNT
        ||' elemente:');
  FOR i IN t.FIRST..t.LAST LOOP
      DBMS OUTPUT.PUT LINE(t(i).id || ' '|| t(i).den);
  END LOOP;
ROLLBACK;
END;
```

```
DECLARE
  TYPE tab ind IS TABLE OF NUMBER INDEX BY VARCHAR2(1);
  t tab ind;
   i varchar2(1);
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
  i := t.FIRST;
  WHILE i IS NOT NULL LOOP
     DBMS OUTPUT.PUT LINE('t('||i ||')='||t(i));
     i := t.NEXT(i);
  END LOOP;
END;
```

```
DECLARE
  TYPE tab imb IS TABLE OF NUMBER;
      tab imb := tab imb(1,2,3,4,5);
  t null tab imb;
BEGIN
  -- atribuire valori
  t.EXTEND(5);
  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;
  t := t null;
  IF t IS NULL THEN
    DBMS OUTPUT.PUT LINE('Colectie atomic null');
  END IF;
END;
```

```
DECLARE
 -- tipul a fost definit la ex4.13
 v_grupe t_imb_categ := t_imb_categ();
 v id categ raion grupe imb.id categorie%TYPE;
 v den
             raion grupe imb.denumire%TYPE;
BEGIN
 SELECT * INTO v id categ, v den, v grupe
 FROM raion grupe imb
 WHERE id_categorie = 1;
 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;
```

```
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
  IF t IS EMPTY THEN
     DBMS OUTPUT.PUT LINE('t nu are elemente');
  END IF;
  -- CARDINALITY
  DBMS OUTPUT.PUT('t1 are '|| CARDINALITY(t1) ||
                  ' 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
  t := SET(t1);
  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
  t := t1 MULTISET EXCEPT t2;
  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: ');
  t := t1 MULTISET UNION DISTINCT t2;
    FOR i IN 1..t.LAST LOOP
      DBMS OUTPUT.PUT(t(i) | | ' ');
  END LOOP;
  DBMS OUTPUT.NEW LINE;
```

```
-- MULTISET INSERSECT
  t := t1 MULTISET INTERSECT DISTINCT t2;
  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
  IF t2=t3 THEN
     DBMS OUTPUT.PUT LINE('t2 = t3');
     DBMS OUTPUT.PUT LINE('t2 <> t3');
 END IF;
  IF t3=t4 THEN
     DBMS OUTPUT.PUT LINE('t3 = t4');
    DBMS OUTPUT.PUT LINE('t3 <> t4');
  END IF;
  -- IN
  IF t4 IN (t1, t2, t3) THEN
    DBMS OUTPUT.PUT LINE('t4 in (t1,t2,t3)');
     DBMS OUTPUT.PUT LINE('t4 not in (t1,t2,t3)');
  END IF;
  -- IS A SET
  IF t4 IS A SET THEN
     DBMS OUTPUT.PUT LINE('t4 este multime');
    DBMS OUTPUT.PUT LINE('t4 nu este multime');
 END IF;
  -- MEMBER OF
  IF 2 MEMBER OF t4 THEN
     DBMS OUTPUT.PUT LINE('2 este in t4');
 ELSE DBMS OUTPUT.PUT LINE('2 nu este in t4');
 END IF;
   -- SUBMULTISET OF
  IF t5 SUBMULTISET OF t4 THEN
    DBMS OUTPUT.PUT LINE('t5 este inclus in t4');
    DBMS OUTPUT.PUT LINE('t5 nu este inclus in t4');
  END IF;
END;
```

```
INSERT INTO CATEGORII(id categorie)
VALUES (9999);
INSERT INTO produse (id produs, denumire, id categorie)
VALUES(9999, 'D9999', 9999);
COMMIT;
SELECT id produs, denumire, id categorie
FROM produse
WHERE id categorie in (800, 900, 9999)
ORDER BY 3;
DECLARE
   TYPE tip_vec IS VARRAY(3) OF NUMBER(4);
   v \text{ tip vec} := tip vec(800, 900, 9999);
   eroare EXCEPTION;
   PRAGMA EXCEPTION INIT (eroare, -24381);
   nr erori NUMBER;
BEGIN
   FORALL i IN 1...3 SAVE EXCEPTIONS
      DELETE FROM produse
     WHERE id categorie = v(i);
  END LOOP;
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;
SELECT id produs, denumire, id categorie
     produse
FROM
WHERE id categorie in (800, 900, 9999)
ORDER BY 3;
```