Student : Costi-Daniel DOBRESCU

Anul 2, IS, grupa 2.2

Proiect baze de date

Problema 1

Proiect la Baze de date -1

Se considera un magazin virtual pentru vanzarea de produse electonice. Baza de date Oracle (considerata ca o colectie de tabele) trebuie sa contina informatiile enumerate mai jos.

-NumarCont - sir de exact 8 caractere
-Nume – sir de maxim 30 de caractere
-Prenume – sir de maxim 30 de caractere
-DataNasterii – data calendaristica
-IdProdus - intreg
-Produs – un sir de maxim 80 de caractere
-Garantie – intreg (nu poate sa fie mai mare de 5 ani)
-Stoc – interg (nu pot fi niciodata mai mult de 200 de produse)
-ValoareUnitara – un numar real cu maxim 5 cifre in partea intreaga si 2 zecimale)
-CantiateVanduta - intreg
-DataVanzarii – data calendaristica

a) Să se realizeze proiectarea bazei de date aferente (structura de tabele, structura de coloane a fiecărei tabele, constrângeri – PRIMARY KEY, NOT NULL, CHECK, FOREIGN KEY .. REFERENCES).

b) Să se scrie comenzile SQL pentru tabelele proiectate la punctul anterior.

c) Să se scrie comenzile SQL pentru popularea bazei de date cu urmatoarele produse si cu urmatorii clienti:

IdProdus	Produs	Garantie	Stoc	ValoareUnitara
1	Fujitsu Siemens Amilo Pro	1	10	2000
2	Indesit WLI1000	3	5	900
3	Gorenje RC400	3	4	1500

NumarCont	Nume	Prenume	DataNasterii
11111111	Popescu	Ion	01-Jan-1985
22222222	Georgescu	Andreea	23-Aug-1983
33333333	Ionescu	Robert	08-Mar-1982

Pentru fiecare produs vandut, trebuie sa se salveze in baza de date, informatiile asociate vanzarii (catre cine a fost vandut produsul, care produs a fost vandut, cantitatea vanduta si data in care s-a efectuat vanzarea).

d) Să se scrie un trigger care realizeaza decrementarea stocului atunci cand un produs a fost vandut.

e) Să se scrie o procedură stocata care realizeaza vanzarea unei anumite cantitati a unui produs catre un client.

f) Să se genereze un raport care sa afiseze toate cumparaturile facute de o persoana. Raportul va afişa următoarele coloane:

Nume Prenume
Produs Cantitate
ValoareTotala

g) Să se genereze un raport care să afişeze toate produsele vandute pentru care nu a expirat inca garantia. Raportul va contine urmatoarele coloane:

Produs
DataExpirării

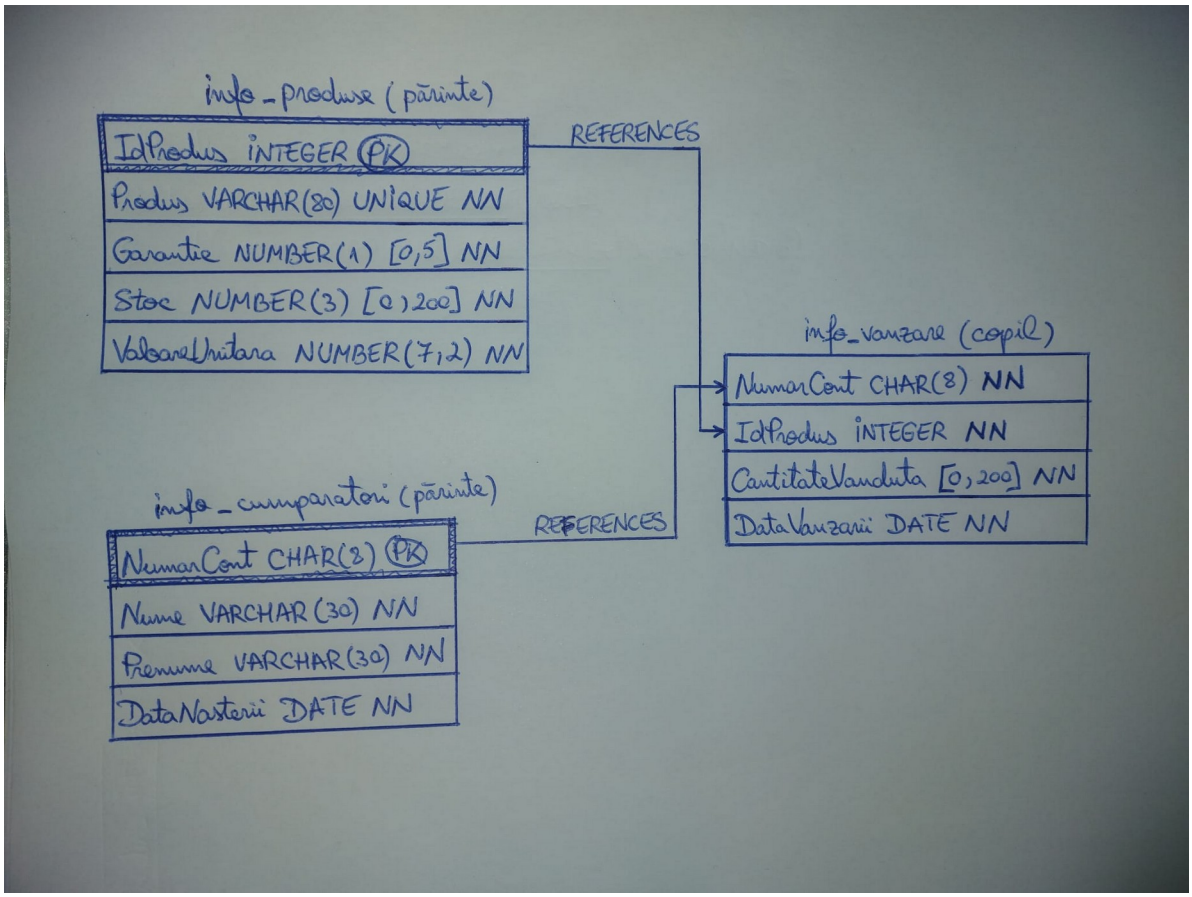
h) Să se afişeze cel mai vandut produs;

i) Să se afişeze data în care au avut loc cele mai multe vanzări;

j) Să se afişeze clientul (nume,prenume) care a cumpărat cele mai multe produse (şi valoarea totală a cumparaturilor).

Rezolvare :

a) Să se realizeze proiectarea bazei de date aferente (structura de tabele, structura de coloane a fiecărei tabele, constrângeri – PRIMARY KEY, NOT NULL, CHECK, FOREIGN KEY .. REFERENCES).



b) Să se scrie comenzile SQL pentru tabelele proiectate la punctul anterior.

```
CREATE TABLE info_produse(
IdProdus INTEGER PRIMARY KEY,
Produs VARCHAR(80) UNIQUE NOT NULL,
Garantie NUMBER(1) CHECK(garantie BETWEEN 0 AND 5) NOT NULL,
Stoc NUMBER(3) CHECK(stoc BETWEEN 0 AND 200) NOT NULL,
ValoareUnitara NUMBER(7,2) NOT NULL);

CREATE TABLE info_cumparatori(
NumarCont CHAR(8) PRIMARY KEY,
Nume VARCHAR(30) NOT NULL,
Prenume VARCHAR(30) NOT NULL,
DataNasterii DATE NOT NULL);

CREATE TABLE info_vanzare(
NumarCont CHAR(8) NOT NULL REFERENCES info_cumparatori(NumarCont) ON DELETE
CASCADE,
IdProdus INTEGER NOT NULL REFERENCES info_produse(IdProdus) ON DELETE CASCADE,
CantitateVanduta Number(3) CHECK(CantitateVanduta BETWEEN 0 AND 200) NOT NULL,
DataVanzarii DATE NOT NULL);

DESCRIBE info_produse;
DESCRIBE info_cumparatori;
DESCRIBE info_vanzare;
```

```
SQL> CREATE TABLE info_produse(
  2  IdProdus INTEGER PRIMARY KEY,
  3  Produs VARCHAR(80) UNIQUE NOT NULL,
  4  Garantie NUMBER(1) CHECK(garantie BETWEEN 0 AND 5) NOT NULL,
  5  Stoc NUMBER(3) CHECK(stoc BETWEEN 0 AND 200) NOT NULL,
  6  ValoareUnitara NUMBER(7,2) NOT NULL);

Table created.

SQL> CREATE TABLE info_cumparatori(
  2  NumarCont CHAR(8) PRIMARY KEY,
  3  Nume VARCHAR(30) NOT NULL,
  4  Prenume VARCHAR(30) NOT NULL,
  5  DataNasterii DATE NOT NULL);

Table created.

SQL> CREATE TABLE info_vanzare(
  2  NumarCont CHAR(8) NOT NULL REFERENCES info_cumparatori(NumarCont) ON DELETE CASCADE,
  3  IdProdus INTEGER NOT NULL REFERENCES info_produse(IdProdus) ON DELETE CASCADE,
  4  CantitateVanduta Number(3) CHECK(CantitateVanduta BETWEEN 0 AND 200) NOT NULL,
  5  DataVanzarii DATE NOT NULL);

Table created.
```

```
SQL> DESCRIBE info_produse;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 IDPRODUS                                  NOT NULL NUMBER(38)
 PRODUS                                    NOT NULL VARCHAR2(80)
 GARANTIE                                  NOT NULL NUMBER(1)
 STOC                                      NOT NULL NUMBER(3)
 VALOAREUNITARA                            NOT NULL NUMBER(7,2)

SQL> DESCRIBE info_cumparatori;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NUMARCONT                                 NOT NULL CHAR(8)
 NUME                                      NOT NULL VARCHAR2(30)
 PRENUME                                   NOT NULL VARCHAR2(30)
 DATANASTERII                              NOT NULL DATE

SQL> DESCRIBE info_vanzare;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NUMARCONT                                 NOT NULL CHAR(8)
 IDPRODUS                                  NOT NULL NUMBER(38)
 CANTITATEVANDUTA                          NOT NULL NUMBER(3)
 DATAVANZARII                              NOT NULL DATE
```

c) Să se scrie comenzile SQL pentru popularea bazei de date cu urmatoarele produse si cu urmatorii clienti:

IdProdus	Produs	Garantie	Stoc	ValoareUnitara
1	Fujitsu Siemens Amilo Pro	1	10	2000
2	Indesit WLI1000	3	5	900
3	Gorenje RC400	3	4	1500

NumarCont	Nume	Prenume	DataNasterii
11111111	Popescu	Ion	01-Jan-1985
22222222	Georgescu	Andreea	23-Aug-1983
33333333	Ionescu	Robert	08-Mar-1982

Pentru fiecare produs vandut, trebuie sa se salveze in baza de date, informatiile asociate vanzarii (catre cine a fost vandut produsul, care produs a fost vandut, cantitatea vanduta si data in care s-a efectuat vanzarea).

```
INSERT INTO info_produse VALUES(1,'Fujitsu Siemens Amilo Pro',1,10,2000);
INSERT INTO info_produse VALUES(2,'Indesit WLI1000',3,5,900);
INSERT INTO info_produse VALUES(3,'Gorenje RC400',3,4,1500);

INSERT INTO info_cumparatori VALUES('11111111','Popescu','Ion','01-Jan-1985');
INSERT INTO info_cumparatori VALUES('22222222','Georgescu','Andreea','23-Aug-1983');
INSERT INTO info_cumparatori VALUES('33333333','Ionescu','Robert','08-Mar-1982');

SELECT *FROM info_produse;
SELECT *FROM info_cumparatori;
```

```
SQL> INSERT INTO info_produse VALUES(1,'Fujitsu Siemens Amilo Pro',1,10,2000);

1 row created.

SQL> INSERT INTO info_produse VALUES(2,'Indesit WLI1000',3,5,900);

1 row created.

SQL> INSERT INTO info_produse VALUES(3,'Gorenje RC400',3,4,1500);

1 row created.

SQL> INSERT INTO info_cumparatori VALUES('11111111','Popescu','Ion','01-Jan-1985');

1 row created.

SQL> INSERT INTO info_cumparatori VALUES('22222222','Georgescu','Andreea','23-Aug-1983');

1 row created.

SQL> INSERT INTO info_cumparatori VALUES('33333333','Ionescu','Robert','08-Mar-1982');

1 row created.

SQL> SET LINESIZE 200;
SQL> SET PAGESIZE 30;
SQL> SELECT *FROM info_produse;

  IDPRODUS PRODUS                                                                                GARANTIE       STOC VALOAREUNITARA
---------- ---------------------------------------------------------------------------------- ---------- ---------- ---------------
         1 Fujitsu Siemens Amilo Pro                                                                   1         10            2000
         2 Indesit WLI1000                                                                             3          5             900
         3 Gorenje RC400                                                                               3          4            1500

SQL> SELECT *FROM info_cumparatori;

NUMARCON NUME                           PRENUME                        DATANASTE
-------- ------------------------------ ------------------------------ ---------
11111111 Popescu                        Ion                            01-JAN-85
22222222 Georgescu                      Andreea                        23-AUG-83
33333333 Ionescu                        Robert                         08-MAR-82
```

d) Să se scrie un trigger care realizeaza decrementarea stocului atunci cand un produs a fost vandut.

```
CREATE OR REPLACE TRIGGER decrementare_stoc
AFTER INSERT ON info_vanzare
FOR EACH ROW
DECLARE
stoc_vechi NUMBER(3);
e1 EXCEPTION;
BEGIN
SELECT stoc INTO stoc_vechi FROM info_produse WHERE IdProdus = :New.IdProdus;
IF(stoc_vechi - :New.CantitateVanduta = 0) THEN
UPDATE info_produse  SET stoc = stoc_vechi - :NEW.CantitateVanduta WHERE IdProdus
= :New.IdProdus;
DBMS_OUTPUT.PUT_LINE('S-a vandut!STOC EPUIZAT');
ELSIF(stoc_vechi - :New.CantitateVanduta < 0) THEN RAISE e1;
ELSE
UPDATE info_produse SET stoc = stoc_vechi - :NEW.CantitateVanduta WHERE IdProdus = :New.IdProdus;
DBMS_OUTPUT.PUT_LINE('Vanzare efectuata cu succes !');
END IF;
EXCEPTION
WHEN e1 THEN RAISE_APPLICATION_ERROR(-20000,'Nu s-a vandut!STOC INSUFICIENT');
WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20000,'EROARE !');
END;
```

```
SQL> CREATE OR REPLACE TRIGGER decrementare_stoc
  2  AFTER INSERT ON info_vanzare
  3  FOR EACH ROW
  4  DECLARE
  5  stoc_vechi NUMBER(3);
  6  e1 EXCEPTION;
  7  BEGIN
  8  SELECT stoc INTO stoc_vechi FROM info_produse WHERE IdProdus = :New.IdProdus;
  9  IF(stoc_vechi - :New.CantitateVanduta = 0) THEN
 10  UPDATE info_produse  SET stoc = stoc_vechi - :NEW.CantitateVanduta WHERE IdProdus = :New.IdProdus;
 11  DBMS_OUTPUT.PUT_LINE('S-a vandut!STOC EPUIZAT');
 12  ELSIF(stoc_vechi - :New.CantitateVanduta < 0) THEN RAISE e1;
 13  ELSE
 14  UPDATE info_produse SET stoc = stoc_vechi - :NEW.CantitateVanduta WHERE IdProdus = :New.IdProdus;
 15  DBMS_OUTPUT.PUT_LINE('Vanzare efectuata cu succes !');
 16  END IF;
 17  EXCEPTION
 18  WHEN e1 THEN RAISE_APPLICATION_ERROR(-20000,'Nu s-a vandut!STOC INSUFICIENT');
 19  WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20000,'EROARE !');
 20  END;
 21  /

Trigger created.
```

SET AUTOCOMMIT ON;
SET SERVEROUTPUT ON;

INSERT INTO info_vanzare VALUES('11111111',1,2,'01-Jan-2020');
INSERT INTO info_vanzare VALUES('22222222',2,3,'01-Jan-2020');
INSERT INTO info_vanzare VALUES('33333333',3,1,'01-Jan-2020');

INSERT INTO info_vanzare VALUES('33333333',1,3,'02-Jan-2020');
INSERT INTO info_vanzare VALUES('22222222',2,1,'03-Jan-2020');
INSERT INTO info_vanzare VALUES('11111111',3,1,'03-Jan-2020');

SELECT *FROM info_vanzare ORDER BY NumarCont;

```
SQL> SELECT *FROM info_vanzare ORDER BY NumarCont;

NUMARCON   IDPRODUS CANTITATEVANDUTA DATAVANZA
-------- ---------- ---------------- ---------
11111111          3                1 03-JAN-20
11111111          1                2 01-JAN-20
22222222          2                1 03-JAN-20
22222222          2                3 01-JAN-20
33333333          1                3 02-JAN-20
33333333          3                1 01-JAN-20

6 rows selected.
```

SELECT *FROM info_produse;

```
SQL> SELECT *FROM info_produse;

  IDPRODUS PRODUS                                                                          GARANTIE       STOC VALOAREUNITARA
---------- ---------------------------------------------------------------------------- ---------- ---------- --------------
         1 Fujitsu Siemens Amilo Pro                                                             1          5           2000
         2 Indesit WLI1000                                                                       3          1            900
         3 Gorenje RC400                                                                         3          2           1500
```

-- Observam ca stocul s-a decrementat in urma inserarii

SELECT *FROM info_produse;
INSERT INTO info_vanzare VALUES('11111111',2,1,'05-Jan-2020');
--Cand stocul devine 0 afiseaza STOC EPUIZAT
INSERT INTO info_vanzare VALUES('11111111',2,1,'05-Jan-2020');
--Eroare deoarece nu mai sunt produse in stoc

```
SQL> SELECT *FROM info_produse;

  IDPRODUS PRODUS                                                                          GARANTIE       STOC VALOAREUNITARA
---------- ---------------------------------------------------------------------------- ---------- ---------- --------------
         1 Fujitsu Siemens Amilo Pro                                                             1          5           2000
         2 Indesit WLI1000                                                                       3          1            900
         3 Gorenje RC400                                                                         3          2           1500

SQL> INSERT INTO info_vanzare VALUES('11111111',2,1,'05-Jan-2020');
S-a vandut!STOC EPUIZAT

1 row created.

Commit complete.
SQL> SELECT *FROM info_produse;

  IDPRODUS PRODUS                                                                          GARANTIE       STOC VALOAREUNITARA
---------- ---------------------------------------------------------------------------- ---------- ---------- --------------
         1 Fujitsu Siemens Amilo Pro                                                             1          5           2000
         2 Indesit WLI1000                                                                       3          0            900
         3 Gorenje RC400                                                                         3          2           1500

SQL> INSERT INTO info_vanzare VALUES('11111111',2,1,'05-Jan-2020');
INSERT INTO info_vanzare VALUES('11111111',2,1,'05-Jan-2020')
*
ERROR at line 1:
ORA-20000: Nu s-a vandut!STOC INSUFICIENT
ORA-06512: at "DANI.DECREMENTARE_STOC", line 15
ORA-04088: error during execution of trigger 'DANI.DECREMENTARE_STOC'
```

e) Să se scrie o procedură stocata care realizeaza vanzarea unei anumite cantitati a unui produs catre un client.

CREATE OR REPLACE PROCEDURE vanzare(NumarCard_p NUMBER,IdProdus_p INTEGER,CantitateVanduta_p NUMBER) AS
DataVanzarii_p DATE;
BEGIN
SELECT sysdate INTO DataVanzarii_p FROM DUAL;
INSERT INTO info_vanzare VALUES(NumarCard_p,IdProdus_p,CantitateVanduta_p,DataVanzarii_p);
END;

```
SQL> CREATE OR REPLACE PROCEDURE vanzare(NumarCard_p NUMBER,IdProdus_p INTEGER,CantitateVanduta_p NUMBER) AS
  2  DataVanzarii_p DATE;
  3  BEGIN
  4  SELECT sysdate INTO DataVanzarii_p FROM DUAL;
  5  INSERT INTO info_vanzare VALUES(NumarCard_p,IdProdus_p,CantitateVanduta_p,DataVanzarii_p);
  6  END;
  7  /

Procedure created.
```

exec vanzare('11111111',3,1);
exec vanzare('22222222',3,1);
exec vanzare('33333333',3,1);

```
SQL> exec vanzare('11111111',3,1);
Vanzare efectuata cu succes !

PL/SQL procedure successfully completed.

Commit complete.
SQL> exec vanzare('22222222',3,1);
S-a vandut!STOC EPUIZAT

PL/SQL procedure successfully completed.

Commit complete.
SQL> exec vanzare('33333333',3,1);
BEGIN vanzare('33333333',3,1); END;

*
ERROR at line 1:
ORA-20000: Nu s-a vandut!STOC INSUFICIENT
ORA-06512: at "DANI.DECREMENTARE_STOC", line 15
ORA-04088: error during execution of trigger 'DANI.DECREMENTARE_STOC'
ORA-06512: at "DANI.VANZARE", line 5
ORA-06512: at line 1
```

SELECT *FROM info_produse;
SELECT *FROM info_vanzare ORDER BY NumarCont;

```
SQL> SELECT *FROM info_produse;

  IDPRODUS PRODUS                                                                          GARANTIE       STOC VALOAREUNITARA
---------- ---------------------------------------------------------------------------- ---------- ---------- --------------
         1 Fujitsu Siemens Amilo Pro                                                             1          5           2000
         2 Indesit WLI1000                                                                       3          0            900
         3 Gorenje RC400                                                                         3          0           1500

SQL> SELECT *FROM info_vanzare ORDER BY NumarCont;

NUMARCON   IDPRODUS CANTITATEVANDUTA DATAVANZA
-------- ---------- ---------------- ---------
11111111          3                1 03-JAN-20
11111111          1                2 01-JAN-20
11111111          3                1 12-MAY-22
11111111          2                1 05-JAN-20
22222222          3                1 12-MAY-22
22222222          2                3 01-JAN-20
22222222          2                1 03-JAN-20
33333333          3                1 01-JAN-20
33333333          1                3 02-JAN-20

9 rows selected.
```

f) Să se genereze un raport care sa afiseze toate cumparaturile facute de o persoana. Raportul va afişa următoarele coloane:

Nume
Prenume
Produs
Cantitate
ValoareTotala

```
SELECT b.nume,b.prenume,a.produs,c.CantitateVanduta,c.CantitateVanduta*a.ValoareUnitara AS
ValoareTotala
 FROM info_produse a,info_cumparatori b,info_vanzare c
 WHERE c.NumarCont=b.NumarCont AND c.IdProdus=a.IdProdus AND b.nume='Georgescu'
 ORDER BY ValoareTotala DESC;
```

```
SQL> SELECT b.nume,b.prenume,a.produs,c.CantitateVanduta,c.CantitateVanduta*a.ValoareUnitara AS ValoareTotala
  2  FROM info_produse a,info_cumparatori b,info_vanzare c
  3  WHERE c.NumarCont=b.NumarCont AND c.IdProdus=a.IdProdus AND b.nume='Georgescu'
  4  ORDER BY ValoareTotala DESC;

NUME                           PRENUME                        PRODUS                                                                                              CANTITATEVANDUTA VALOARETOTALA
------------------------------ ------------------------------ --------------------------------------------------------------------------------------------------- ---------------- ------------
Georgescu                      Andreea                        Indesit WLI1000                                                                                                    3         2700
Georgescu                      Andreea                        Gorenje RC400                                                                                                      1         1500
Georgescu                      Andreea                        Indesit WLI1000                                                                                                    1          900
```

g) Să se genereze un raport care să afişeze toate produsele vandute pentru care nu a expirat inca garantia. Raportul va contine urmatoarele coloane:
Produs
DataExpirării

```
SELECT a.produs,TRUNC(a.garantie*365.25 + c.DataVanzarii) AS DataExpirarii
FROM info_produse a,info_vanzare c
WHERE c.IdProdus=a.IdProdus AND TRUNC(a.garantie*365.25 + c.DataVanzarii) > sysdate
ORDER BY DataExpirarii
```

```
SQL> SELECT a.produs,TRUNC(a.garantie*365.25 + c.DataVanzarii) AS DataExpirarii
  2  FROM info_produse a,info_vanzare c
  3  WHERE c.IdProdus=a.IdProdus AND TRUNC(a.garantie*365.25 + c.DataVanzarii) > sysdate
  4  ORDER BY DataExpirarii;

PRODUS                                                                                  DATAEXPIR
--------------------------------------------------------------------------------------- ---------
Indesit WLI1000                                                                         31-DEC-22
Gorenje RC400                                                                           31-DEC-22
Indesit WLI1000                                                                         02-JAN-23
Gorenje RC400                                                                           02-JAN-23
Indesit WLI1000                                                                         04-JAN-23
Gorenje RC400                                                                           12-MAY-25
Gorenje RC400                                                                           12-MAY-25

7 rows selected.
```

h) Să se afişeze cel mai vandut produs

```
SQL> SELECT IdProdus FROM info_vanzare
  2  GROUP BY IdProdus
  3  HAVING SUM(CantitateVanduta)=(SELECT MAX(SUM(CantitateVanduta)) FROM info_vanzare GROUP BY IdProdus)
  4  ORDER BY IdProdus;

  IDPRODUS
----------
         1
         2
```

VERIFICARE :

```
SELECT IdProdus,SUM(CantitateVanduta) AS TotalCantitate
FROM info_vanzare
GROUP BY IdProdus
ORDER BY TotalCantitate DESC;
```

```
SQL> SELECT IdProdus,SUM(CantitateVanduta) AS TotalCantitate
  2  FROM info_vanzare
  3  GROUP BY IdProdus
  4  ORDER BY TotalCantitate DESC;

  IDPRODUS TOTALCANTITATE
---------- --------------
         1              5
         2              5
         3              4
```

i) Să se afişeze data în care au avut loc cele mai multe vanzări

SELECT DataVanzarii FROM info_vanzare
GROUP BY DataVanzarii
HAVING SUM(CantitateVanduta)=(SELECT MAX(SUM(CantitateVanduta)) FROM info_vanzare
GROUP BY DataVanzarii);

```
SQL> SELECT DataVanzarii FROM info_vanzare
  2  GROUP BY DataVanzarii
  3  HAVING SUM(CantitateVanduta)=(SELECT MAX(SUM(CantitateVanduta)) FROM info_vanzare GROUP BY DataVanzarii);

DATAVANZA
---------
01-JAN-20
```

VERIFICARE :

SELECT DataVanzarii,SUM(CantitateVanduta) AS total
FROM info_vanzare
GROUP BY DataVanzarii
ORDER BY DataVanzarii;

```
SQL> SELECT DataVanzarii,SUM(CantitateVanduta) AS total
  2  FROM info_vanzare
  3  GROUP BY DataVanzarii
  4  ORDER BY DataVanzarii;

DATAVANZA      TOTAL
--------- ----------
01-JAN-20          6
02-JAN-20          3
03-JAN-20          2
05-JAN-20          1
12-MAY-22          1
12-MAY-22          1

6 rows selected.
```

j) Să se afişeze clientul (nume,prenume) care a cumpărat cele mai multe produse (şi valoarea totală a cumparaturilor).

```
SELECT b.nume,b.prenume,SUM(c.CantitateVanduta*a.ValoareUnitara) AS ValoareTotala
FROM info_produse a,info_cumparatori b,info_vanzare c
WHERE c.NumarCont=b.NumarCont AND c.IdProdus=a.IdProdus
GROUP BY b.nume,b.prenume
HAVING SUM(CantitateVanduta)=(SELECT MAX(SUM(CantitateVanduta)) FROM info_vanzare
GROUP BY NumarCont)
ORDER BY ValoareTotala DESC;
```

```
SQL> SELECT b.nume,b.prenume,SUM(c.CantitateVanduta*a.ValoareUnitara) AS ValoareTotala
  2  FROM info_produse a,info_cumparatori b,info_vanzare c
  3  WHERE c.NumarCont=b.NumarCont AND c.IdProdus=a.IdProdus
  4  GROUP BY b.nume,b.prenume
  5  HAVING SUM(CantitateVanduta)=(SELECT MAX(SUM(CantitateVanduta)) FROM info_vanzare GROUP BY NumarCont)
  6  ORDER BY ValoareTotala DESC;

NUME                           PRENUME                        VALOARETOTALA
------------------------------ ------------------------------ -------------
Popescu                        Ion                                     7900
Georgescu                      Andreea                                 5100
```

VERIFICARE :

```
SELECT b.nume,b.prenume,a.produs,c.CantitateVanduta,c.CantitateVanduta*a.ValoareUnitara AS
ValoareTotala
FROM info_produse a,info_cumparatori b,info_vanzare c
WHERE c.NumarCont=b.NumarCont AND c.IdProdus=a.IdProdus
ORDER BY b.nume,b.prenume;
```

```
SQL> SELECT b.nume,b.prenume,a.produs,c.CantitateVanduta,c.CantitateVanduta*a.ValoareUnitara AS ValoareTotala
  2  FROM info_produse a,info_cumparatori b,info_vanzare c
  3  WHERE c.NumarCont=b.NumarCont AND c.IdProdus=a.IdProdus
  4  ORDER BY b.nume,b.prenume;

NUME                           PRENUME                        PRODUS                                                                                           CANTITATEVANDUTA VALOARETOTALA
------------------------------ ------------------------------ ------------------------------------------------------------------------------------------------ ---------------- -------------
Georgescu                      Andreea                        Gorenje RC400                                                                                               1          1500
Georgescu                      Andreea                        Indesit WLI1000                                                                                             3          2700
Georgescu                      Andreea                        Indesit WLI1000                                                                                             1           900
Ionescu                        Robert                         Gorenje RC400                                                                                               1          1500
Ionescu                        Robert                         Fujitsu Siemens Amilo Pro                                                                                   3          6000
Popescu                        Ion                            Fujitsu Siemens Amilo Pro                                                                                   2          4000
Popescu                        Ion                            Gorenje RC400                                                                                               1          1500
Popescu                        Ion                            Indesit WLI1000                                                                                             1           900
Popescu                        Ion                            Gorenje RC400                                                                                               1          1500

9 rows selected.
```