PROIECTUL NR.2

BAZE DE DATE

Proiect întocmit de: Bociort Dinu Iulian

Facultatea: Automatică și Calculatoare , Seria: IS

Anul II, Semigrupa 1.1



i) Stuctura bazei de date utilizată:

Bilant:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nr\_cont | Descriere | Tip\_cont | Sold\_init | Sold | Suma\_creditoare | Suma\_debitoare | Nr\_tranza\_implicat |

Tranzitii:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nr\_cont | cont\_c | cont\_d | Nr\_tranzactie | Data\_tranzactiei | Suma\_trans | Descriere\_tranz |

Primary Key

Foreign Key

ii)Dimensionare coloane, constrângeri:

* Constrangeri de domeniu, care definesc valorile luate de un atribut:

NOT NULL se aplica pentru campurile: tip\_cont din tabela bilant;

UNIQUE se aplica la campul nr\_tranzactie din tabela tranzactii;

CHECK – se verifica in tabela bilant daca nr\_tranzactiei si suma\_tranz sunt numere pozitive si nu poate fi mai mari decat 10000.

* Constrangeri de integritate a entitatii:

PRIMARY KEY – nr\_cont din tabela bilant;

* Constrangeri de integritate referentiala:

FOREIGN KEY – nr\_cont din tabela tranzactii cu referinta la nr\_cont din tabela bilant cu clauza ON DELETE CASCADE;

1. Crearea tabelelor se face după cum urmează:

CREATE TABLE bilant (

nr\_cont NUMBER(5) PRIMARY KEY,

descriere VARCHAR(50),

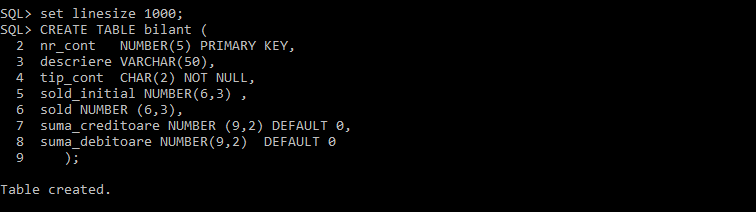
tip\_cont CHAR(2) NOT NULL,

sold\_initial NUMBER(6,3) ,

sold NUMBER (6,3),

suma\_creditoare NUMBER (9,2) DEFAULT 0,

suma\_debitoare NUMBER(9,2) DEFAULT 0

 );

CREATE TABLE tranzactii (

Nr\_cont NUMBER(5),

nr\_tranzactie INTEGER Unique CHECK ((nr\_tranzactie>0) and (nr\_tranzactie<10000)) ,

cont\_d NUMBER(5) NOT NULL,

cont\_c NUMBER(5) NOT NULL,

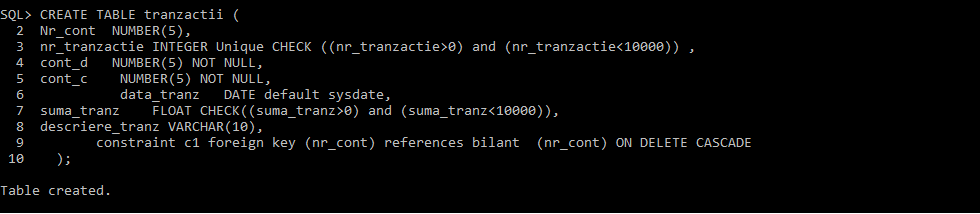
data\_tranz DATE default sysdate,

suma\_tranz FLOAT CHECK((suma\_tranz>0) and (suma\_tranz<10000)),

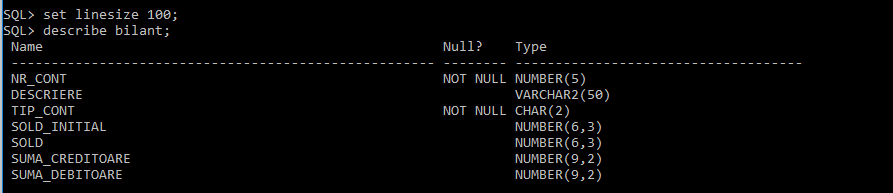
descriere\_tranz VARCHAR(10),

constraint c1 foreign key (nr\_cont) references bilant (nr\_cont) ON DELETE CASCADE

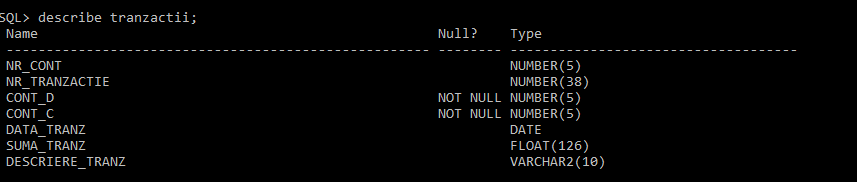
);



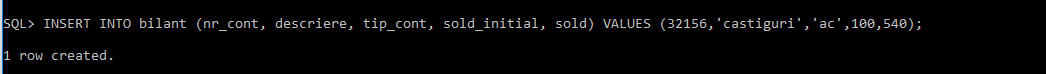
DESCRIBE bilant;

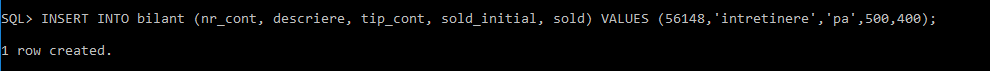


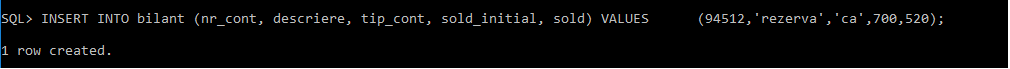
Describe tranzactii;

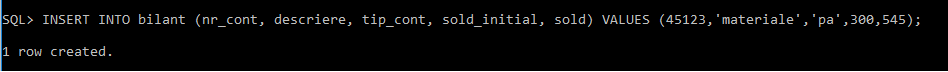


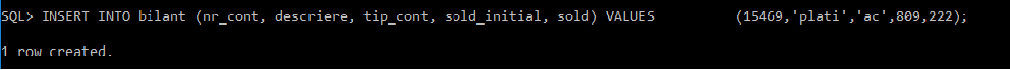
1. Inserarea datelor în tabela bilant:

INSERT INTO bilant (nr\_cont, descriere, tip\_cont, sold\_initial, sold) VALUES (32156,'castiguri','ac',100,540);

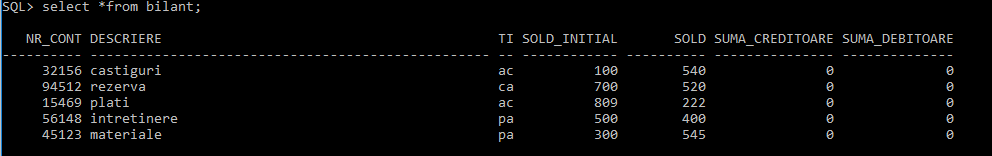
INSERT INTO bilant (nr\_cont, descriere, tip\_cont, sold\_initial, sold) VALUES (56148,'intretinere','pa',500,400);

INSERT INTO bilant (nr\_cont, descriere, tip\_cont, sold\_initial, sold) VALUES (94512,'rezerva','ca',700,520);

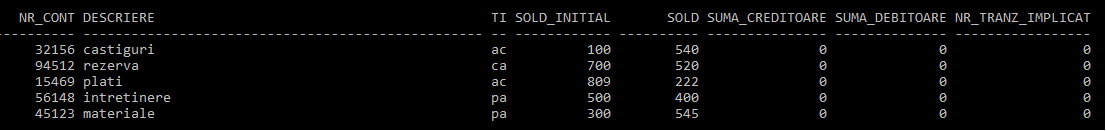
INSERT INTO bilant (nr\_cont, descriere, tip\_cont, sold\_initial, sold) VALUES (45123,'materiale','pa',300,545);

INSERT INTO bilant (nr\_cont, descriere, tip\_cont, sold\_initial, sold) VALUES (15469,'plati','ac',809,222);

SET LINESIZE 1000;

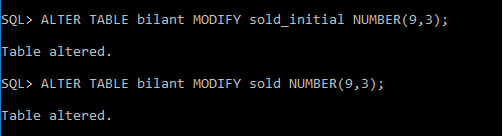
SELECT \* FROM conturi;

ALTER TABLE bilant ADD nr\_tranz\_implicat INTEGER default 0;



ALTER TABLE bilant MODIFY sold\_initial NUMBER(9,3);

ALTER TABLE bilant MODIFY sold NUMBER(9,3);



3. Sa se implementeze o procedura care sa implementeze o tranzactie, stiind ca aceasta presupune 2 conturi – unul debitor (din care pleaca banii) si respectiv unul creditor (in care intra banii), procedura primeste ca si parametrii de intrare numarul tranzactiei, contul debitor, contul credtor, suma tranzactionata precum si o descriere a tranzactiei.

CREATE OR REPLACE PROCEDURE tranz ( nr\_tranzactiei INTEGER, cont\_d NUMBER, cont\_c NUMBER, data\_tranz DATE, suma\_tranz FLOAT, descriere VARCHAR )

AS

BEGIN

INSERT INTO tranzactii VALUES(nr\_tranzactiei, cont\_d, cont\_c, data\_tranz, suma\_tranz, descriere);

UPDATE bilant SET nr\_tranz\_implicat=nr\_tranz\_implicat+1 WHERE nr\_cont=cont\_d OR nr\_cont=cont\_c;

END;

/

EXECUTE tranz(1,32156,15469,'02-03-2010',259.5,'plata');

EXECUTE tranz(2,32156,56148,'10-05-2010',542.6,'cheltuieli');

EXECUTE tranz(3,94512,45123,'24-04-2015',879.7,'stoc');

EXECUTE tranz(4,32156,94512,'06-04-2017',1546,'achizitii');

EXECUTE tranz(5,45123,15469,'02-05-2016',485.5,'taxe');

4. Sa se implementeze un trigger care atunci cand este introdusa o tranzactie automat sa calculeze soldul conturilor implicate in tranzactie.

CREATE OR REPLACE TRIGGER tri

AFTER INSERT ON tranzactii

FOR EACH ROW

DECLARE

c1 NUMBER;

c2 NUMBER;

s FLOAT;

sold FLOAT;

BEGIN

c1:=:NEW.cont\_d;

c2:=:NEW.cont\_c;

s:=:NEW.suma\_tranz;

SELECT sold INTO sold FROM bilant WHERE nr\_cont=c1;

IF sold < s THEN

RAISE\_APPLICATION\_ERROR(-20000,'Suma prea mica in cont!');

ELSE

UPDATE bilant SET sold=sold-s WHERE nr\_cont=c1;

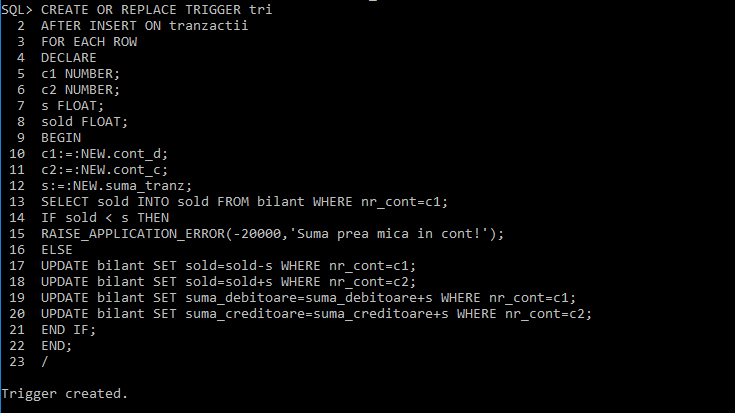
UPDATE bilant SET sold=sold+s WHERE nr\_cont=c2;

UPDATE bilant SET suma\_debitoare=suma\_debitoare+s WHERE nr\_cont=c1;

UPDATE bilant SET suma\_creditoare=suma\_creditoare+s WHERE nr\_cont=c2;

END IF;

END;

/

5) Să se afişeze toate tranzacţiile în care este folosit un anumit cont

SELECT \* FROM tranzactii WHERE cont\_d=32156 OR cont\_c=32156;

6) Să se afişeze toate tranzactiile care au fost introduse in perioada (01.01.2010-01.06.2010)

SELECT \* FROM tranzactii WHERE data\_tranz > '01-01-2010' AND data\_tranz < '01-06-2010';

7) Să se calculeze şi să se afişeze bilantul initial, suma totala creditoare, respectiv debitoare, pentru toate conturile

SELECT sold\_initial AS bilantul\_initial, sum(suma\_creditoare),sum( suma\_debitoare) FROM bilant;

8) Să se afişeze toate tranzacţiile care implică conturi de un anumit tip

SELECT nr\_tranz, cont\_d, cont\_c, data\_tranz, suma\_tranz FROM bilant, tranzactii WHERE (nr\_cont=cont\_d OR nr\_cont=cont\_c) AND tip\_cont='ac group by nr\_cont';

9) Să se steargă un cont daca nu exista tranzactii pentru el

DELETE FROM conturi WHERE nr\_tranz\_implicat = 0;

10) . Să se afiseze contul care apare in cele mai multe tranzactii precum si numarul de tranzactii in care el apare

select nr\_cont, count(nr\_cont) from tranzactii having count(nr\_cont)=(select max(count(nr\_cont)) from tranzactii group by nr\_cont) group by nr\_cont;

DROP TABLE bilant;

DROP TABLE tranzactii;