**Proiect baze de date**

**Problema 10**

Gavra Alexandru-Andrei

Gr. 3.1

Se considera o aplicatie pentru evidenta cheltuielilor unei firme, care efectueaza plati in valuta (registru de casa). Baza de date Oracle contine urmatoarele informatii:

- curs valutar: data, rata de schimb

- parteneri: nume, CUI, Adresa

- tranzactii: data efectuarii, CUI, suma in valuta, suma in lei

- bilant zilnic: data, sold initial, total intrari, total iesiri, sold final.

Stiind ca rata de schimb este cuprinsa intre 1.00 si 4.00, numele nu depaseste 10 caractere, CUI are exact 6 caractere, suma este negativa daca se plateste partenerului si pozitiva daca se incaseaza de la partener, si ca soldul initial a fost 0, se cere:

1. Să se realizeze proiectarea bazei de date aferente (structura de tabele, structura de coloane a fiecărei tabele, constrângeri).

2. Sa se scrie comenzile SQL pentru tabelele proiectate la punctul anterior.

3. Să se scrie comenzile SQL pentru popularea bazei de date cu 5 tranzactii efectuate considerand 3 parteneri, in 2 zile consecutive.

4. Să se implementeze un mecanism care sa nu permita modificarea cursului valutar pentru zilelele cand sunt efectuate tranzactii, sa se argumenteze alegerea facuta.

5. Sa se scrie o procedura care sa genereze un raport care sa cuprinda numele,CUI, intrari, iesiri pentru partenerul specificat ca si argument prin CUI.

6. Sa se genereze un raport detaliat care sa cuprinda pentru fiecare partener: numele, CUI-ul, si toate tranzactiile efectuate (data tranzactie, tip tranzactie (intrare sau iesire), suma in valuta, suma in lei), ordonat dupa nume si data efectuarii tranzactiilor.

7. Sa se scrie un trigger care la adaugarea unei tranzactii sa calculeze automat suma in lei, iar daca cursul nu este specificat, sa semnaleze eroare si sa nu permita adaugarea tranzactiei.

8. Sa se scrie o functie care sa primeasca ca si parametru CUI si sa returneze rulajul efectuat de acel partener (suma dintre intrari si iesiri luate fara semn).

9. Sa se afiseze partenerii care doar au incasat si care nu au facut plati intr-un an intreg, precizand: nume, CUI, suma totala incasata.

10. Sa se afiseze partenerul care are cele mai multe tranzactii, precizand numele, CUI-ul, numar de tranzactii, rata lui de rentabilitate (cat la suta este profit din suma totala a tranzactiilor).

**Rezolvare:**

**1.Să se realizeze proiectarea bazei de date aferente (structura de tabele, structura de coloane a fiecărei tabele, constrângeri).**

Text

Description automatically generated

**2. Sa se scrie comenzile SQL pentru tabelele proiectate la punctul anterior.**

CREATE TABLE curs(

2 data date PRIMARY KEY,

3 rata NUMERIC(3,2) CHECK (rata BETWEEN 1.00 AND 4.00));

CREATE TABLE parteneri(

2 cui char(6) PRIMARY KEY,

3 nume VARCHAR(10)

4 adresa varchar(10));

CREATE TABLE tranzactii(

2 cui char(6) REFERENCES parteneri (cui),

3 data\_ef date default NULL,

4 partener VARCHAR(10),

5 sumav numeric(8),

6 sumal numeric(8));

CREATE TABLE bilant(

2 data date UNIQUE,

3 soldi numeric(8) default 0,

4 tin numeric(8),

5 tout numeric(8),

6 soldf numeric(8));

Legatura dintre tabela **parteneri** si tabela **tranzactii** se face prin campul “CUI”, legatura de tip **REFERENCES.**

Legatura dintre cele 2 tabele provoaca urmatoarea restrictie : o linie din tabela **parteneri** nu poate fi stearsa pana cand toate liniile  din tabela **tranzactii** care contin CUI-ul liniei care se doreste a fi stearsa nu sunt sterse.

**3. Să se scrie comenzile SQL pentru popularea bazei de date cu 5 tranzactii efectuate considerand 3 parteneri, in 2 zile consecutive.**

insert into curs values('13-06-2020',3);

insert into curs values('20-05-2020',3.5);

insert into curs values('21-05-2020',3);

insert into curs values('25-05-2020',4);

insert into curs values('25-05-2021',3);

insert into parteneri values('123456','Ionescu','Bucuresti');

insert into parteneri values('123457','Popescu','Timisoara');

insert into parteneri values('123458','George','Arad');

insert into parteneri values('123459','Hagi','Constanta');

insert into tranzactii values('123456','21-05-2020','Ionescu',10,35);

insert into tranzactii values('123458','21-05-2020','George',-20,-70);

insert into tranzactii values('123456','22-05-2020','Ionescu',-5,-15);

insert into tranzactii values('123457','22-05-2020','Popescu',15,45);

insert into tranzactii values('123458','22-05-2020','George',30,90);

insert into tranzactii values('123459','13-06-2020','Hagi',20,60);

insert into tranzactii values('123459','13-06-2020','Hagi',-30,90);

insert into tranzactii values('123459','20-05-2020','Hagi',-10,-35);

insert into tranzactii values('123459','21-05-2020','Hagi',-30,-90);

insert into tranzactii values('123459','21-05-2021','Hagi',40,75);

insert into bilant values('20-05-2020',0,10,-20,-10);

insert into bilant values('21-05-2020',-10,45,-5,30);

Text

Description automatically generated

**4. Să se implementeze un mecanism care sa nu permita modificarea cursului valutar pentru zilelele cand sunt efectuate tranzactii, sa se argumenteze alegerea facuta.**

CREATE OR REPLACE TRIGGER zile\_tranzactii

BEFORE UPDATE ON curs

FOR EACH ROW

DECLARE

y NUMERIC(3);

a EXCEPTION;

BEGIN

SELECT count(data\_ef) INTO y from tranzactii WHERE data\_ef=:NEW.data;

IF y<>0 THEN :NEW.rata:=:OLD.rata;

DBMS\_OUTPUT.PUT\_LINE('Nu se poate schimba rata pentru ca s-au efectuat tranzactii in data de '||:new.data||'');

END IF;

END;

/

Text

Description automatically generated

Am ales sa folosesc un **trigger** pe care l-am numit **zile-tranzactii**. Acesta va verifica de fiecare data conditiile impuse inainte de un **UPDATE** la tabela **curs**.

Se cauta in tabela **tranzactii** data in care se doreste sa se schimbe rata, daca se gaseste atunci nu se poate schimba.

**5. Sa se scrie o procedura care sa genereze un raport care sa cuprinda numele,CUI, intrari, iesiri pentru partenerul specificat ca si argument prin CUI.**

create or replace procedure intrari\_iesiri(c varchar)

as

x varchar(10);

y numeric(8) default 0;

z numeric(8) default 0;

begin

select nume into x from parteneri where cui=c;

select abs(sum(sumav)) into y from tranzactii where cui=c and sumav<0;

select sum(sumav) into z from tranzactii where cui=c and sumav>=0;

dbms\_output.put\_line('NUME:'||x);

dbms\_output.put\_line('CUI:'||c);

dbms\_output.put\_line('INTRARI:'||y);

dbms\_output.put\_line('IESIRI:'||z);

end;

/

Text

Description automatically generated

Am creat procedura cu numele **intrari\_iesiri** care are ca parametru variabila **c** de tip varchar. Proceduta are 3 variabile: **x** de tip varchar si **y, z** de tip numeric.

Daca **CUI** este egal cu **c** se selecteaza campul nume in variabila **x**, valoarea absoluta a sumei in valuta(**sumav**) in variabila **y**, daca **sumav<0**, iar daca este **>0**, se selecteaza **sumav** in variabila **z**.

Procedura se executa folsind comanda : **exec intrari\_iesiri(‘123459’);**

**6. Sa se genereze un raport detaliat care sa cuprinda pentru fiecare partener: numele, CUI-ul, si toate tranzactiile efectuate (data tranzactie, tip tranzactie (intrare sau iesire), suma in valuta, suma in lei), ordonat dupa nume si data efectuarii tranzactiilor.**

select partener,cui,data\_ef,sumav,sumal, 'Intrare' as Tip from tranzactii

where sumav<0

group by partener,cui,data\_ef,sumav ,sumal

union

select partener,cui,data\_ef,sumav,sumal, 'Iesire' as Tip from tranzactii

where sumav>=0

group by partener,cui,data\_ef,sumav,sumal

order by partener,data\_ef;

Text

Description automatically generated

**Pasii raportului**: se introduce campul **Tip** care poate contine cuvintele **‘Intrare’** sau **‘Iesire’** in functie valoarea campului **sumav** (suma in valuta). Daca **sumav<0** atunci **Tip=’Intrare’**, daca **sumav>=0** atunci **Tip=’Iesire’**. Am folosit operatorul **UNION** pentru a combina rezultatele celor 2 **SELECT**-uri.

**8. Sa se scrie o functie care sa primeasca ca si parametru CUI si sa returneze rulajul efectuat de acel partener (suma dintre intrari si iesiri luate fara semn).**

create or replace function rulaj(c char)

return numeric as

x numeric;

begin

select sum(abs(sumav)) into x from tranzactii where cui=c;

return x;

end;

/

Text

Description automatically generated

Am creat funnctia **rulaj** care are ca parametru variabila **c** de tip char. In functie avem variabila **x** de tip numeric in care selectam suma valorilor absolute a campului **sumav**(suma in valuta) pentru **CUI-ul** dorit. Functia returneaza valoarea lui **x**.

Functia se apeleaza folosind comanda: **slect rulaj(‘123458’) from dual;**

**10. Sa se afiseze partenerul care are cele mai multe tranzactii, precizand numele, CUI-ul, numar de tranzactii, rata lui de rentabilitate (cat la suta este profit din suma totala a tranzactiilor).**

create or replace function rata\_rent(c char)

return numeric as

a numeric(8);

b numeric(8);

d numeric(8);

begin

select sum(sumav) into b from tranzactii where sumav<0 and c=cui;

select sum(sumav) into d from tranzactii where sumav>0 and c=cui;

a:=(abs(b)\*100)/(abs(b)+d);

return a;

end;

/

select partener,cui,count(cui) as nr\_tranzactii,rata\_rent(cui) as rata from tranzactii

group by partener,cui having count(cui)=(select max(count(cui)) from tranzactii group by partener);

Text

Description automatically generated

Am creat functia **rata\_rent** care are parametrul **c** de tip char. Functia are 3 variabile de tip numeric: **a, b, c**. Pentru **CUI=c** am selectat suma valorilor campului **sumav**(suma in valuta) in variabila **b** daca **sumav<0** si in variabila **d** daca **sumav>0**. Am foloit formula

**a:=(abs(b)\*100)/(abs(b)+d)** pentru calcularea ratei de rentabilitate. Functia returneaza valoarea variabilei **a**.

Functia este apelata folosind campul **cui**.

Am aflat partenerul cu numarul maxim de tranzactii folosid conditia: **having count(cui)=(select max(count(cui))**.