CREATE TABLE PERSOANE

(

"IDPERS" NUMBER(5,0),

"NUME" VARCHAR(50),

"ADRESA" VARCHAR(100),

"GEN" CHAR(1),

"DATA\_NASTERII" DATE,

CONSTRAINT "PERSOANE\_PK" PRIMARY KEY ("IDPERS")

USING INDEX ENABLE

);

CREATE TABLE CONTURI

(

"NRCONT" NUMBER(10,0),

"SOLD" NUMBER(20,0),

"IDPERS" NUMBER(5,0),

CONSTRAINT "CONTURI\_PK" PRIMARY KEY ("NRCONT")

USING INDEX ENABLE

);

CREATE TABLE CARDURI

(

"NRCARD" VARCHAR(20),

"DATA\_DE\_LA" DATE,

"DATA\_LA" DATE,

"LIMITA" NUMBER(4,0),

"NRCONT" NUMBER(10,0),

"TIP" VARCHAR(50),

"CATEGORIE" VARCHAR(50),

CONSTRAINT "CARDURI\_PK" PRIMARY KEY ("NRCARD")

USING INDEX ENABLE

);

CREATE TABLE MISCARI

(

"NRCARD" VARCHAR(20),

"DATA\_ORA" TIMESTAMP,

"VALOARE" NUMBER(4,0),

"SCOP" VARCHAR(100)

--CONSTRAINT "MISCARI\_PK" PRIMARY KEY ("DATA\_ORA")

--USING INDEX ENABLE

);

ALTER TABLE CONTURI ADD CONSTRAINT "CONTURI\_FK" FOREIGN KEY ("IDPERS")

REFERENCES PERSOANE ("IDPERS") ENABLE;

ALTER TABLE CARDURI ADD CONSTRAINT "CARDURI\_FK" FOREIGN KEY ("NRCONT")

REFERENCES CONTURI ("NRCONT") ENABLE;

ALTER TABLE MISCARI ADD CONSTRAINT "MISCARI\_FK" FOREIGN KEY ("NRCARD")

REFERENCES CARDURI ("NRCARD") ENABLE;

INSERT INTO PERSOANE (idpers, nume, adresa, gen, data\_nasterii)

VALUES(1, 'John Doe', 'Str. Principală 123', 'M', TO\_DATE('05-15-1990','mm-dd-yyyy'));

INSERT INTO PERSOANE (idpers, nume, adresa, gen, data\_nasterii)

VALUES(2, 'Jane Smith', 'Str. Centrală 456', 'F', TO\_DATE('09-20-1995','mm-dd-yyyy'));

INSERT INTO PERSOANE (idpers, nume, adresa, gen, data\_nasterii)

VALUES(3, 'Alex Johnson', 'Str. Libertății 789', 'M', TO\_DATE('12-10-1998','mm-dd-yyyy'));

INSERT INTO PERSOANE (idpers, nume, adresa, gen, data\_nasterii)

VALUES(4, 'Emily Brown', 'Str. Victoria 321', 'F', TO\_DATE('07-03-1992','mm-dd-yyyy'));

INSERT INTO PERSOANE (idpers, nume, adresa, gen, data\_nasterii)

VALUES(5, 'Michael Davis', 'Str. Unirii 567', 'M', TO\_DATE('03-28-1997','mm-dd-yyyy'));

INSERT INTO CONTURI(nrcont,sold,idpers)

VALUES(1234567890, 1000, 1);

INSERT INTO CONTURI(nrcont,sold,idpers)

VALUES(1234567899, 2000, 1);

INSERT INTO CONTURI(nrcont,sold,idpers)

VALUES(9876543210, 500, 2);

INSERT INTO CONTURI(nrcont,sold,idpers)

VALUES(4567890123, 1500, 3);

INSERT INTO CONTURI(nrcont,sold,idpers)

VALUES (0123456789, 2000, 4);

INSERT INTO CONTURI(nrcont,sold,idpers)

VALUES (0123456788, 3000, 4);

INSERT INTO CONTURI(nrcont,sold,idpers)

VALUES(5432109876, 800, 5);

INSERT INTO CARDURI(nrcard, data\_de\_la, data\_la, limita, nrcont, tip, categorie)

VALUES('1234-5678-9012-3456',TO\_DATE('01-01-2023','mm-dd-yyyy'), TO\_DATE('01-01-2024','mm-dd-yyyy'), 1000, 1234567890, 'MASTERCARD', 'CREDIT');

INSERT INTO CARDURI(nrcard, data\_de\_la, data\_la, limita, nrcont, tip, categorie)

VALUES('1234-5678-9912-3426',TO\_DATE('01-01-2023','mm-dd-yyyy'), TO\_DATE('01-01-2024','mm-dd-yyyy'), 1300, 1234567890, 'MASTERCARD', 'CREDIT');

INSERT INTO CARDURI(nrcard, data\_de\_la, data\_la, limita, nrcont, tip, categorie)

VALUES('9876-5432-1098-7654',TO\_DATE('02-01-2023','mm-dd-yyyy'), TO\_DATE('02-01-2024','mm-dd-yyyy'), 500, 9876543210, 'VISA', 'DEBIT');

INSERT INTO CARDURI(nrcard, data\_de\_la, data\_la, limita, nrcont, tip, categorie)

VALUES('4567-8901-2345-6789',TO\_DATE('03-01-2023','mm-dd-yyyy'), TO\_DATE('03-01-2024','mm-dd-yyyy'), 1500, 4567890123, 'MASTERCARD', 'CREDIT');

INSERT INTO CARDURI(nrcard, data\_de\_la, data\_la, limita, nrcont, tip, categorie)

VALUES('1111-2222-3333-4444',TO\_DATE('04-01-2023','mm-dd-yyyy'), TO\_DATE('04-01-2024','mm-dd-yyyy'), 2000, 0123456789, 'VISA', 'DEBIT');

INSERT INTO CARDURI(nrcard, data\_de\_la, data\_la, limita, nrcont, tip, categorie)

VALUES('5432-1098-7654-3210',TO\_DATE('05-01-2023','mm-dd-yyyy'), TO\_DATE('05-01-2024','mm-dd-yyyy'), 800, 5432109876, 'MASTERCARD', 'CREDIT');

INSERT INTO CARDURI(nrcard, data\_de\_la, data\_la, limita, nrcont, tip, categorie)

VALUES('5432-9998-7444-3210',TO\_DATE('05-01-2023','mm-dd-yyyy'), TO\_DATE('05-01-2024','mm-dd-yyyy'), 2000, 0123456788, 'VISA', 'DEBIT');

INSERT INTO MISCARI(nrcard, data\_ora, valoare, scop)

VALUES('1234-5678-9012-3456', TO\_TIMESTAMP('01-01-2022 12:00:20','MM-DD-YYYY HH24:MI:SS'), 100, 'Plata facturii');

INSERT INTO MISCARI(nrcard, data\_ora, valoare, scop)

VALUES('9876-5432-1098-7654', TO\_TIMESTAMP('02-01-2022 13:30:12','MM-DD-YYYY HH24:MI:SS'), 50, 'Cumparare alimente');

INSERT INTO MISCARI(nrcard, data\_ora, valoare, scop)

VALUES('4567-8901-2345-6789', TO\_TIMESTAMP('03-01-2023 15:45:50','MM-DD-YYYY HH24:MI:SS'), 200, 'Transfer bancar');

INSERT INTO MISCARI(nrcard, data\_ora, valoare, scop)

VALUES('1111-2222-3333-4444', TO\_TIMESTAMP('04-01-2022 12:00:20','MM-DD-YYYY HH24:MI:SS'), 150, 'Cumparare mobilier');

INSERT INTO MISCARI(nrcard, data\_ora, valoare, scop)

VALUES('5432-1098-7654-3210', TO\_TIMESTAMP('05-01-2023 16:20:18','MM-DD-YYYY HH24:MI:SS'), 75, 'Plata restaurant');

INSERT INTO MISCARI(nrcard, data\_ora, valoare, scop)

VALUES('5432-9998-7444-3210', TO\_TIMESTAMP('05-01-2023 16:20:18','MM-DD-YYYY HH24:MI:SS'), 80, 'Plata online');

ALTER TABLE Miscari

ADD CONSTRAINT CHK\_ValoareRange CHECK (valoare BETWEEN 1 AND 1000000);

ALTER TABLE Carduri

ADD CONSTRAINT CHK\_TipCategorie CHECK (

(categorie <> 'CREDIT') OR (tip = 'MASTERCARD')

);

Operatii asupra bazei:

12.1

CREATE TABLE PERSOANE

a.

(

"IDPERS" NUMBER(5,0),

"NUME" VARCHAR(50),

"ADRESA" VARCHAR(100),

"GEN" CHAR(1),

"DATA\_NASTERII" DATE,

CONSTRAINT "PERSOANE\_PK" PRIMARY KEY ("IDPERS")

USING INDEX ENABLE

);

b.

CREATE TABLE CONTURI

(

"NRCONT" NUMBER(10,0),

"SOLD" NUMBER(20,0),

"IDPERS" NUMBER(5,0),

CONSTRAINT "CONTURI\_PK" PRIMARY KEY ("NRCONT")

USING INDEX ENABLE

);

c.

CREATE TABLE CARDURI

(

"NRCARD" VARCHAR(20),

"DATA\_DE\_LA" DATE,

"DATA\_LA" DATE,

"LIMITA" NUMBER(4,0),

"NRCONT" NUMBER(10,0),

"TIP" VARCHAR(50),

"CATEGORIE" VARCHAR(50),

CONSTRAINT "CARDURI\_PK" PRIMARY KEY ("NRCARD")

USING INDEX ENABLE

);

d.

CREATE TABLE MISCARI

(

"NRCARD" VARCHAR(20),

"DATA\_ORA" TIMESTAMP,

"VALOARE" NUMBER(4,0),

"SCOP" VARCHAR(100)

);

e.

ALTER TABLE CONTURI ADD CONSTRAINT "CONTURI\_FK" FOREIGN KEY ("IDPERS")

REFERENCES PERSOANE ("IDPERS") ENABLE;

ALTER TABLE CARDURI ADD CONSTRAINT "CARDURI\_FK" FOREIGN KEY ("NRCONT")

REFERENCES CONTURI ("NRCONT") ENABLE;

ALTER TABLE MISCARI ADD CONSTRAINT "MISCARI\_FK" FOREIGN KEY ("NRCARD")

REFERENCES CARDURI ("NRCARD") ENABLE;

f. tabela Conturi nu contine coloana 'adresa'

ALTER TABLE Persoane MODIFY adresa VARCHAR(150);

12.2.Să se exprime următoarele constrângeri (la nivel atribut sau tuplă):

a) Coloana valoare din tabela Mișcări trebuie să ia valori între 1 și 1000000.

ALTER TABLE Miscari

ADD CONSTRAINT CHK\_ValoareRange CHECK (valoare BETWEEN 1 AND 1000000);

b) Dacă categoria unui card este ‘CREDIT’ atunci tipul este ‘MASTERCARD’.

ALTER TABLE Carduri

ADD CONSTRAINT CHK\_TipCategorie CHECK (

(categorie <> 'CREDIT') OR (tip = 'MASTERCARD')

);

12.03. Să se exprime în SQL următoarele interogări:

a) Să se găsească numărul de cont și soldul pentru conturile cu sold cuprins între

1000 și 2000 ordonat crescător după sold și cont.

SELECT nrcont,sold

FROM conturi

WHERE sold >= 1000 AND sold <= 2000

ORDER BY sold ASC

b) Să se găsească detaliile mișcărilor pentru mișcările ale căror scop conține

’cumpărare’ ordonat descrescător după valoare și crescător după scop.

SELECT \*

FROM miscari

WHERE LOWER(scop) LIKE '%cumparare%'

ORDER BY valoare DESC

SELECT \*

FROM miscari

WHERE LOWER(scop) LIKE '%cumparare%'

ORDER BY scop ASC

12.04. Să se exprime în SQL următoarele interogări folosind operatorul JOIN:

a) Cine sunt titularii cardurilor de tip ’VISA’?

SELECT persoane.nume,carduri.tip

FROM persoane

JOIN conturi ON persoane.idpers = conturi.idpers

JOIN carduri ON conturi.nrcont = carduri.nrcont

WHERE tip = 'VISA';

b) Să se găsească perechi de carduri (nrcard1, nrcard2) care au fost utilizate la

aceeași data\_ora. O pereche este unică în rezultat.

SELECT DISTINCT m1.nrcard AS nrcard1, m2.nrcard AS nrcard2

FROM Miscari m1

JOIN Miscari m2 ON m1.data\_ora = m2.data\_ora

WHERE m1.nrcard <> m2.nrcard

ORDER BY nrcard1, nrcard2;

12.05. Să se exprime în SQL fără funcții de agregare următoarele interogări

folosind cel puţin o interogare imbricată şi operatori de genul EXISTS, IN, ALL,

ANY:

a) Să se găsească numele persoanelor ce dețin un singur cont.

SELECT nume

FROM Persoane

WHERE idpers IN(

SELECT idpers

FROM Conturi

GROUP BY idpers

HAVING COUNT(\*) = 1

);

b) Să se găsească numerele cardurilor care au aceeași limită ca și cardul cu

numărul ’1111-2222-3333-4444’.

SELECT nrcard --,limita

FROM carduri

WHERE limita =(

SELECT limita

FROM carduri

WHERE nrcard = '1111-2222-3333-4444'

);

12.06. Să se exprime în SQL următoarele interogări folosind funcţii de agregare:

a) Să se găsească pentru fiecare persoană de sex masculin câte carduri deține

(nume, număr\_card).

SELECT p.nume, COUNT(c.nrcard) AS număr\_card

FROM Persoane p

JOIN Conturi co ON p.idpers = co.idpers

JOIN Carduri c ON co.nrcont = c.nrcont

WHERE p.gen = 'M'

GROUP BY p.nume;

b) Să se găsească pentru fiecare categorie de card suma valorii cheltuite în anul

2022.

SELECT cr.categorie, SUM(m.valoare) AS suma\_valoare

FROM Carduri cr

JOIN Miscari m ON cr.nrcard = m.nrcard

WHERE EXTRACT(YEAR FROM m.data\_ora) = 2022

GROUP BY cr.categorie;

12.07. Să se scrie instrucţiunile pentru actualizarea BD:

a) Să se adauge faptul că s-a deschis contul 12345678 pentru persoana cu

identificator 100 și s-a alocat cardul cu numărul 101101 de tip

‘MASTERCARD’ cu limita 5000, valabilă între 01 ianuarie 2022 și 31

decembrie 2022.

Inainte de a insera in tabela conturi un nou obiect va trebui sa introducem in tabela persoane, o persoana cu idpers = 100, altfel vom primi o eroare de integritate:

INSERT INTO PERSOANE (idpers, nume, adresa, gen, data\_nasterii)

VALUES(100, 'Alex Brown', 'Str. Prahovei 57', 'M', TO\_DATE('03-11-1992','mm-dd-yyyy'));

-- Inserarea în tabela Conturi

INSERT INTO Conturi (nrcont, sold, idpers)

VALUES ('12345678', 0, 100);

-- Inserarea în tabela Carduri

INSERT INTO Carduri (nrcard, data\_de\_la, data\_la, limita, nrcont, tip, categorie)

VALUES ('101101', TO\_DATE('01-01-2022', 'MM-DD-YYYY'), TO\_DATE('12-31-2022', 'MM-DD-YYYY'), 5000, '12345678', 'MASTERCARD', NULL);

b) Să se şteargă persoanele care nu au cont.

DELETE FROM Persoane

WHERE idpers NOT IN (

SELECT idpers

FROM Conturi

);

INSERT INTO PERSOANE (idpers, nume, adresa, gen, data\_nasterii)

VALUES(6, 'Alex Davis', 'Str. Decebal 57', 'M', TO\_DATE('04-17-1997','mm-dd-yyyy'));

c) Să se modifice limita pentru cardurile care au limita egală cu soldul contului

adunând 500.

UPDATE Carduri c

SET limita = limita + 500

WHERE limita = (

SELECT sold

FROM Conturi co

WHERE c.nrcont = co.nrcont

);

12.08. Să se definească o procedură stocată care va introduce în tabela Excepții

acele linii din tabela Carduri cu condiția soldul contului este mai mic decât

valoarea unei mișcări pentru acel card sau există mișcare în aceeași lună cu data

expirării cardului. Tabela Excepții va avea aceleași coloane ca și tabela Carduri

plus o coloană ce indică natura excepției.

CREATE OR REPLACE PROCEDURE IntroducereExceptii AS

BEGIN

INSERT INTO Exceptii (nrcard, data\_de\_la, data\_la, limita, nrcont, tip, categorie, natura\_exceptiei)

SELECT c.nrcard, c.data\_de\_la, c.data\_la, c.limita, c.nrcont, c.tip, c.categorie, 'SOLD' AS natura\_exceptiei

FROM Carduri c

JOIN Conturi co ON c.nrcont = co.nrcont

JOIN Miscari m ON c.nrcard = m.nrcard

WHERE co.sold < m.valoare

OR (EXTRACT(MONTH FROM m.data\_ora) = EXTRACT(MONTH FROM c.data\_la) AND EXTRACT(YEAR FROM m.data\_ora) = EXTRACT(YEAR FROM c.data\_la));

END;

12.09. Să se definească triggere pentru:

a) A asigura că la adăugarea unei mișcări, soldul contului pentru care este alocat cardul se micșorează cu valoarea mișcării.

CREATE OR REPLACE TRIGGER ActualizareSold

AFTER INSERT ON Miscari

FOR EACH ROW

BEGIN

UPDATE Conturi

SET sold = sold - :NEW.valoare

WHERE nrcont = (

SELECT nrcont

FROM Carduri

WHERE nrcard = :NEW.nrcard

);

END;

b) Presupunând vederea:

CREATE VIEW CărțiMastercard AS

SELECT idpers, nume, adresa, gen, data\_nasterii, nrcont, sold, nrcard,

data\_de\_la, data\_la, limita

FROM Carduri NATURAL JOIN Conturi NATURAL JOIN Persoane

WHERE tip = ‘MASTERCARD’

CREATE OR REPLACE TRIGGER ActualizareSoldLimita

AFTER UPDATE OF limita ON Carduri

FOR EACH ROW

BEGIN

UPDATE Conturi

SET sold = sold + (:NEW.limita - :OLD.limita)

WHERE nrcont = :NEW.nrcont;

END;