Proiect la Elemente Avansate

De Baze de Date

Banci

Bancomat

Danci Anca Denisa

Informatica Economica, an III, grupa III

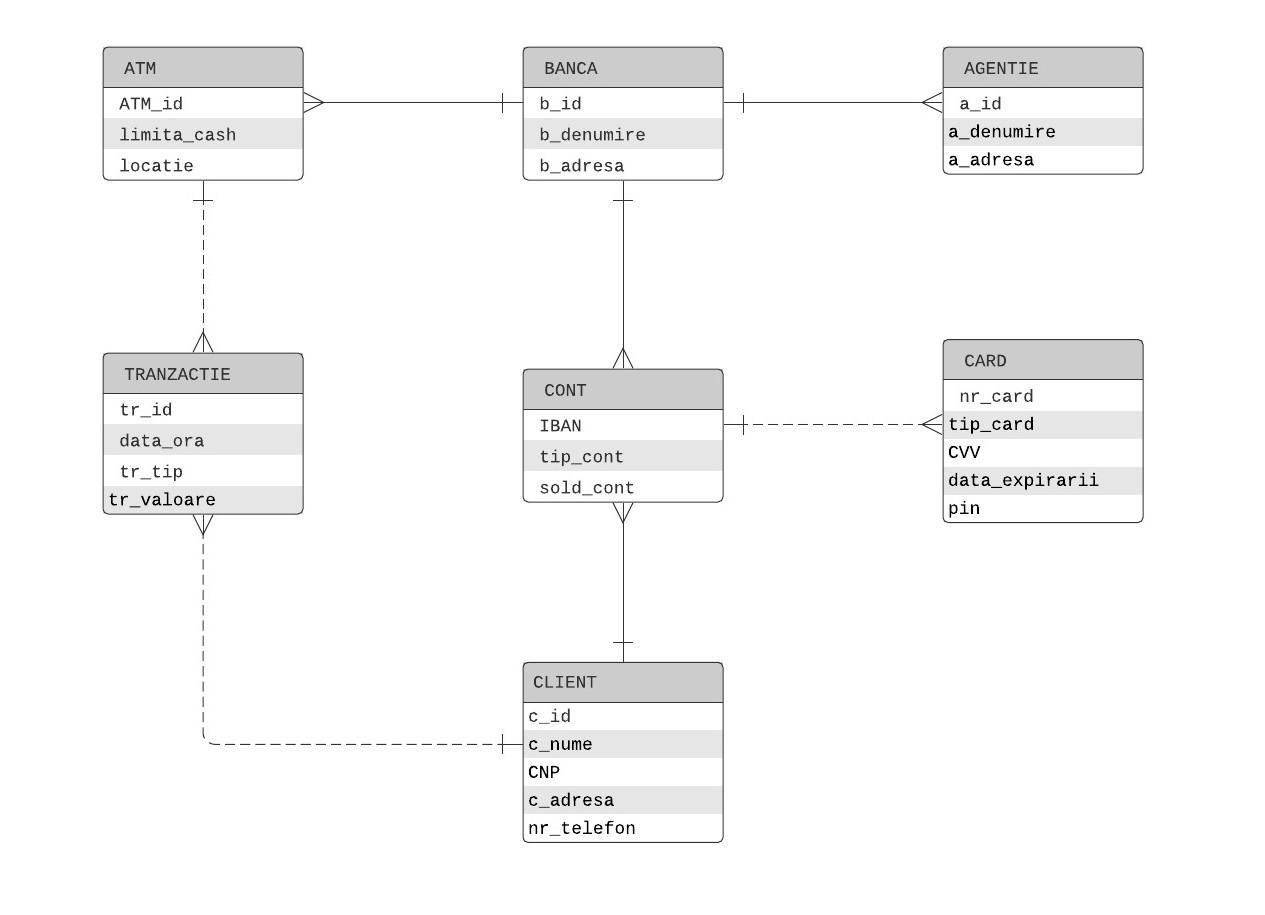
2021-2022

**Scenariu**

In cadrul acestei baze de date este surprinsa activitatea bancilor in legatura cu bancomatele.

Fiecare banca are un sediu central de care apartin una sau mai multe agentii din diverse locatii. Tot de banca, apartin si bancomatele care se pot afla si in afara agentiilor. Acestea au o limita de bani cash care se pot retrage zilnic, dar aceasta difera in functie de politica bancii respective. Un bancomat poate opera mai multi clienti, dar de asemenea si un client poate folosi mai multe bancomate. In baza de date retinem date despre client cum ar fi nume, CNP, adresa si numar de telefon. Un client poate efectua una sau mai multe tranzactii la bancomat. Fiecare client detine unul sau mai multe conturi, fie in lei sau euro, acestea identificandu-se prin codul IBAN. Pentru a putea realiza operatiuni in cadrul unui bancomat si nu numai, fiecare cont poate avea unul sau mai multe carduri, insa nu pot exista doua carduri cu acelasi nume de detinator asignate unui singur cont.

**Diagrama ERD**



**CREARE TABELE**

* **Creare tabela Banca**

CREATE TABLE Banca (

b\_id varchar2(4),

b\_denumire varchar2(50) NOT NULL,

b\_adresa varchar2(50),

constraint Banca\_PK PRIMARY KEY (b\_id)

using INDEX ENABLE

);

* **Creare tabela Agentie**

CREATE TABLE Agentie (

a\_id number,

a\_denumire varchar2(50) NOT NULL,

a\_adresa varchar2(50),

a\_banca varchar2(4) NOT NULL,

constraint Agentie\_PK PRIMARY KEY (a\_id)

using INDEX ENABLE

);

ALTER TABLE Agentie

ADD CONSTRAINT Agentie\_FK

FOREIGN KEY (a\_banca)

REFERENCES Banca (b\_id);

CREATE SEQUENCE Agentie\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE OR REPLACE TRIGGER Agentie\_trg

BEFORE INSERT ON Agentie

FOR EACH ROW

BEGIN

  SELECT Agentie\_seq.NEXTVAL

  INTO   :new.a\_id

  FROM   dual;

END;

* **Creare tabela ATM**

CREATE TABLE ATM (

ATM\_id number,

limita\_cash number,

locatie varchar2(50),

ATM\_banca varchar2(4) NOT NULL,

constraint ATM\_PK PRIMARY KEY (ATM\_id)

using INDEX ENABLE

);

ALTER TABLE ATM

ADD CONSTRAINT ATM\_FK

FOREIGN KEY (ATM\_banca)

REFERENCES Banca (b\_id);

CREATE SEQUENCE ATM\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE OR REPLACE TRIGGER ATM\_trg

BEFORE INSERT ON ATM

FOR EACH ROW

BEGIN

  SELECT ATM\_seq.NEXTVAL

  INTO   :new.ATM\_id

  FROM   dual;

END;

* **Creare tabela Client**

CREATE TABLE Client (

c\_id number,

c\_nume varchar2(50) NOT NULL,

c\_adresa varchar2(50) NOT NULL,

nr\_telefon varchar2(12) NOT NULL,

CNP varchar2(20) NOT NULL,

constraint Client\_PK PRIMARY KEY (c\_id)

using INDEX ENABLE

);

ALTER TABLE Client

ADD CONSTRAINT cnp\_unique UNIQUE (cnp);

* **Creare tabela Cont**

CREATE TABLE Cont (

IBAN varchar2(40),

tip\_cont varchar2(20) NOT NULL,

sold\_cont number,

client\_cont number NOT NULL,

banca\_cont varchar2(4) NOT NULL,

constraint Cont\_PK PRIMARY KEY (IBAN)

using INDEX ENABLE

);

ALTER TABLE Cont

ADD CONSTRAINT Cont\_FK1

FOREIGN KEY (client\_cont)

REFERENCES Client (c\_id);

ALTER TABLE Cont

ADD CONSTRAINT Cont\_FK2

FOREIGN KEY (banca\_cont)

REFERENCES Banca (b\_id);

* **Creare tabela Card**

CREATE TABLE Card (

nr\_card varchar2(20),

tip\_card varchar2(20),

cvv number(3) NOT NULL,

data\_expirarii date NOT NULL,

pin number(4) NOT NULL,

nume\_detinator clob NOT NULL,

card\_IBAN varchar2(40) NOT NULL,

constraint Card\_PK PRIMARY KEY (nr\_card)

using INDEX ENABLE

);

ALTER TABLE Card

ADD CONSTRAINT Card\_FK

FOREIGN KEY (card\_iban)

REFERENCES Cont (IBAN);

* **Creare tabela Tranzactie**

CREATE TABLE Tranzactie(

tr\_id number,

data\_ora timestamp(0) NOT NULL,

tr\_tip varchar2(20),

tr\_valoare number,

tr\_client number NOT NULL,

tr\_atm number NOT NULL,

constraint Tranzactie\_PK PRIMARY KEY (tr\_id)

using INDEX ENABLE

);

ALTER TABLE Tranzactie

ADD CONSTRAINT Tranzactie\_FK

FOREIGN KEY (tr\_client)

REFERENCES Client (c\_id);

ALTER TABLE Tranzactie

ADD CONSTRAINT Tranzactie\_ATM\_FK

FOREIGN KEY (tr\_ATM)

REFERENCES ATM (ATM\_id);

CREATE SEQUENCE Tranzactie\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE OR REPLACE TRIGGER Tranzactie\_trg

BEFORE INSERT ON Tranzactie

FOR EACH ROW

BEGIN

  SELECT Tranzactie\_seq.NEXTVAL

  INTO   :new.tr\_id

  FROM   dual;

END;

**POPULARE TABELE**

* **Populare tabela Banca**

INSERT INTO Banca

VALUES

('BUCU', 'Alpha Bank', 'Calea Dorobanti, nr. 237B, sector 1, Bucuresti');

INSERT INTO Banca

VALUES

('RNCB', 'BCR', 'Calea Victoriei nr. 15, sector 3, București');

INSERT INTO Banca

VALUES

('BTRL', 'Banca Transilvania', 'str. G. Bariţiu nr.8, Cluj-Napoca');

INSERT INTO Banca

VALUES

('BRDE', 'BRD', 'Bd. Ion Mihalache 1-7, sector 1, Bucuresti');

INSERT INTO Banca

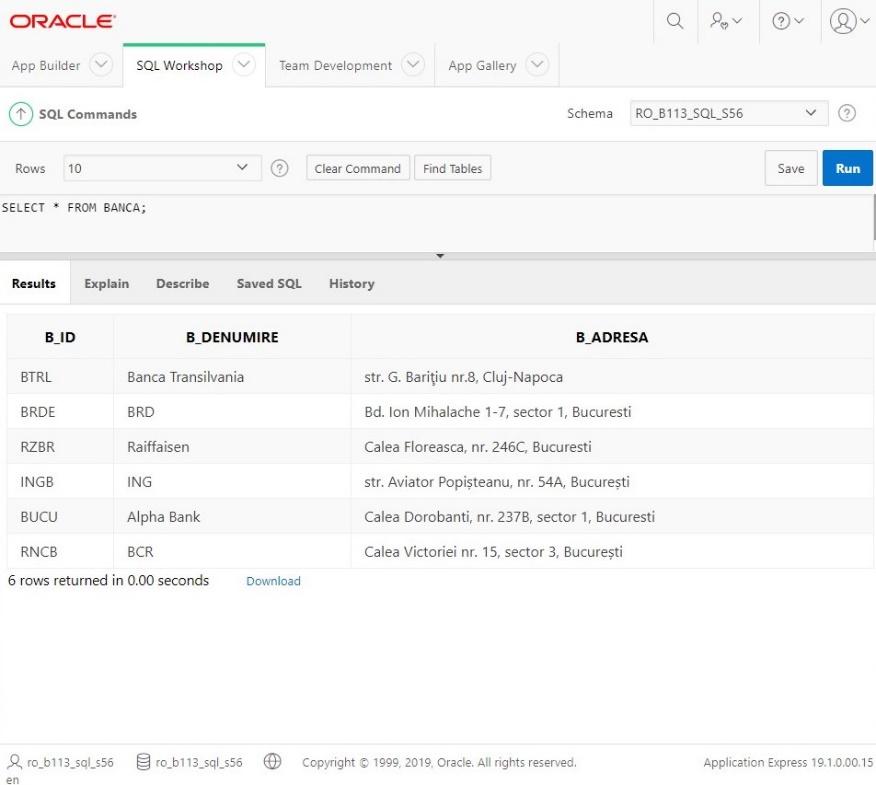
VALUES

('RZBR', 'Raiffaisen', 'Calea Floreasca, nr. 246C, Bucuresti');

INSERT INTO Banca

VALUES

('INGB', 'ING', 'str. Aviator Popișteanu, nr. 54A, București');



* **Populare tabela Agentie**

INSERT INTO Agentie(a\_denumire, a\_adresa, a\_banca)

VALUES

('Alpha Bank Marasti', 'Strada Fabricii nr. 7, Cluj-Napoca', 'BUCU');

INSERT INTO Agentie(a\_denumire, a\_adresa, a\_banca)

VALUES

('Alpha Bank Zorilor', 'Strada Louis Pasteur, nr. 79','BUCU');

INSERT INTO Agentie(a\_denumire, a\_adresa, a\_banca)

VALUES

('BCR', 'Strada 21 Decembrie, Nr. 16, Cluj-Napoca','RNCB');

INSERT INTO Agentie(a\_denumire, a\_adresa, a\_banca)

VALUES

('BCR', 'Strada Fantanele, Nr. 7, Cluj-Napoca','RNCB');

INSERT INTO Agentie(a\_denumire, a\_adresa, a\_banca)

VALUES

('BT Ferdinand', 'Str. Regele Ferdinand nr. 29, Cluj-Napoca','BTRL');

INSERT INTO Agentie(a\_denumire, a\_adresa, a\_banca)

VALUES

('BT Flora', 'Calea Floresti, Nr. 75, Cluj Napoca','BTRL');

INSERT INTO Agentie(a\_denumire, a\_adresa, a\_banca)

VALUES

('BRD Farmec', 'Strada Henri Barbusse nr. 16','BRDE');

INSERT INTO Agentie(a\_denumire, a\_adresa, a\_banca)

VALUES

('Raiffaisen Brancusi', 'Strada Constantin Brancusi, nr. 149, Cluj Napoca','RZBR');

INSERT INTO Agentie(a\_denumire, a\_adresa, a\_banca)

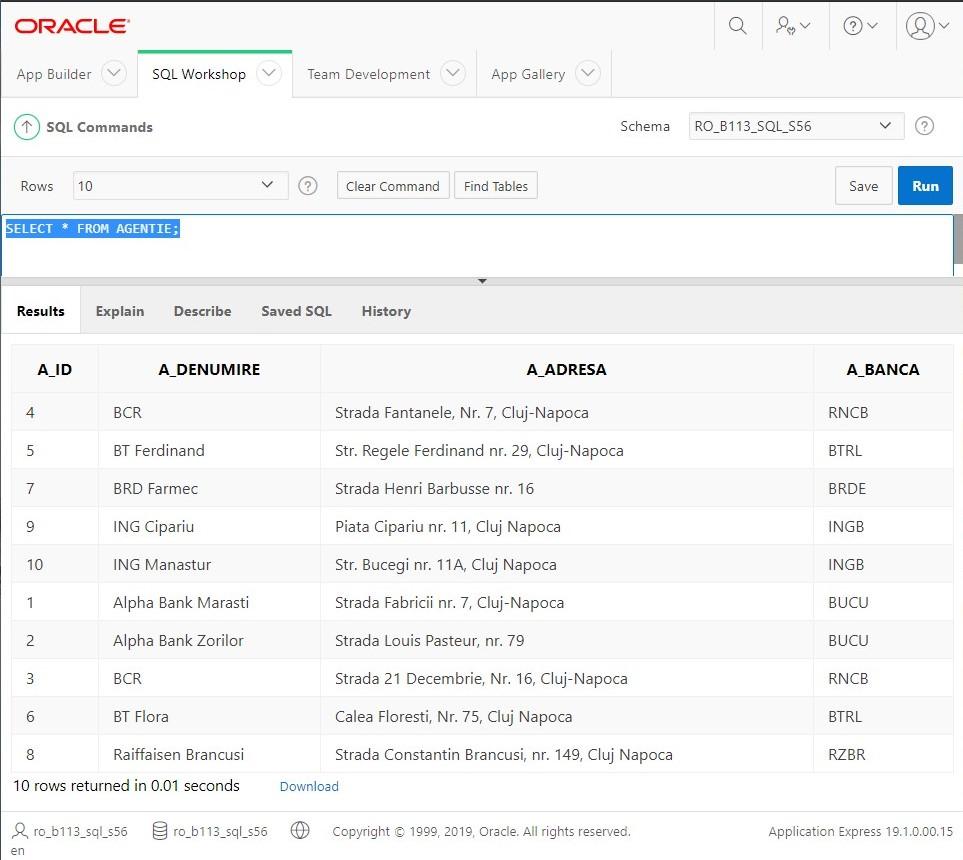
VALUES

('ING Cipariu', 'Piata Cipariu nr. 11, Cluj Napoca','INGB');

INSERT INTO Agentie(a\_denumire, a\_adresa, a\_banca)

VALUES

('ING Manastur', 'Str. Bucegi nr. 11A, Cluj Napoca','INGB');



* **Populare tabela ATM**

INSERT INTO ATM(limita\_cash, locatie, ATM\_banca)

VALUES

(5000, 'Strada Fabricii nr. 7, Cluj-Napoca', 'BUCU');

INSERT INTO ATM(limita\_cash, locatie, ATM\_banca)

VALUES

(3000, 'Strada 21 Decembrie, Nr. 16, Cluj-Napoca', 'RNCB');

INSERT INTO ATM(limita\_cash, locatie, ATM\_banca)

VALUES

(5000, 'Str. Regele Ferdinand nr. 29, Cluj-Napoca', 'BTRL');

INSERT INTO ATM(limita\_cash, locatie, ATM\_banca)

VALUES

(5000, 'Strada Henri Barbusse nr. 16, Cluj-Napoca', 'BRDE');

INSERT INTO ATM(limita\_cash, locatie, ATM\_banca)

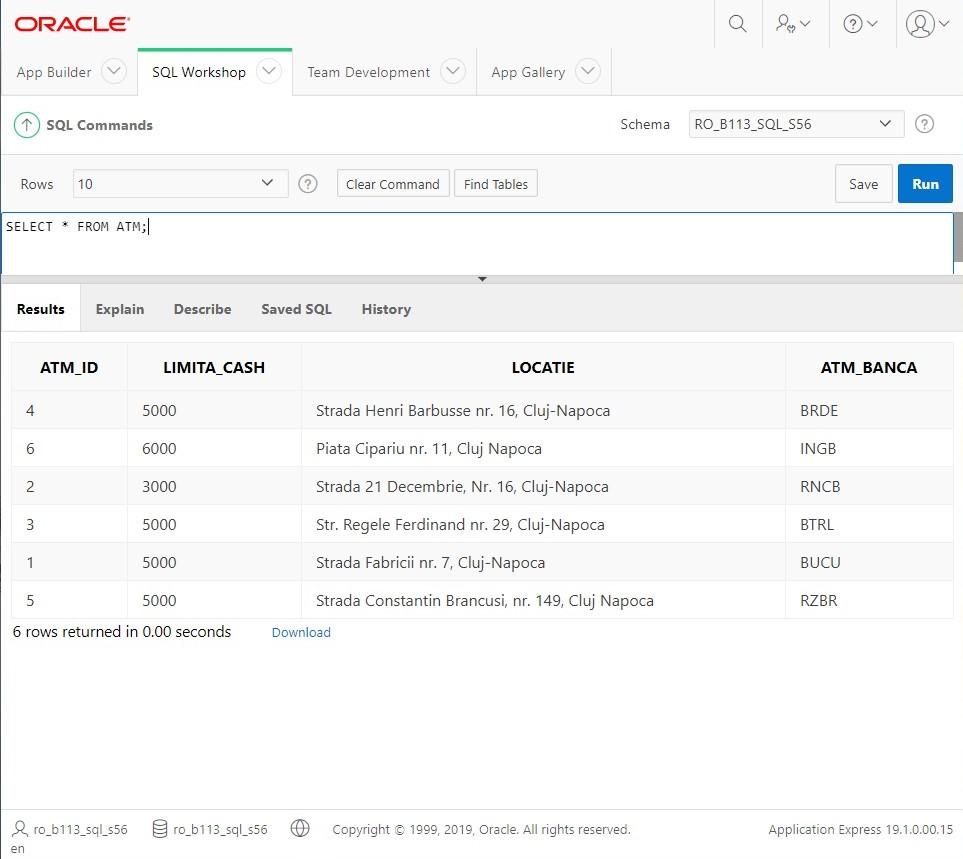
VALUES

(5000, 'Strada Constantin Brancusi, nr. 149, Cluj Napoca', 'RZBR');

INSERT INTO ATM(limita\_cash, locatie, ATM\_banca)

VALUES

(6000, 'Piata Cipariu nr. 11, Cluj Napoca', 'INGB');



* **Populare tabela Client**

INSERT INTO Client

VALUES

(9876541,'Timis Iulia','Str. Oltului nr 84, Cluj-Napoca', '0756789990','2900908365504');

INSERT INTO Client

VALUES

(1232248,'Stecko Daiana','Str. Crinului nr 7, Cluj-Napoca', '0776843435','2870624015336');

INSERT INTO Client

VALUES

(2436778,'Botezan Vlad','Str. Muresului nr 1, Cluj-Napoca', '0740388093','1941129188640');

INSERT INTO Client

VALUES

(9907211,'Marchis Alexandru','Str. Armoniei nr 5, Cluj-Napoca', '0756786453','5000601519940');

INSERT INTO Client

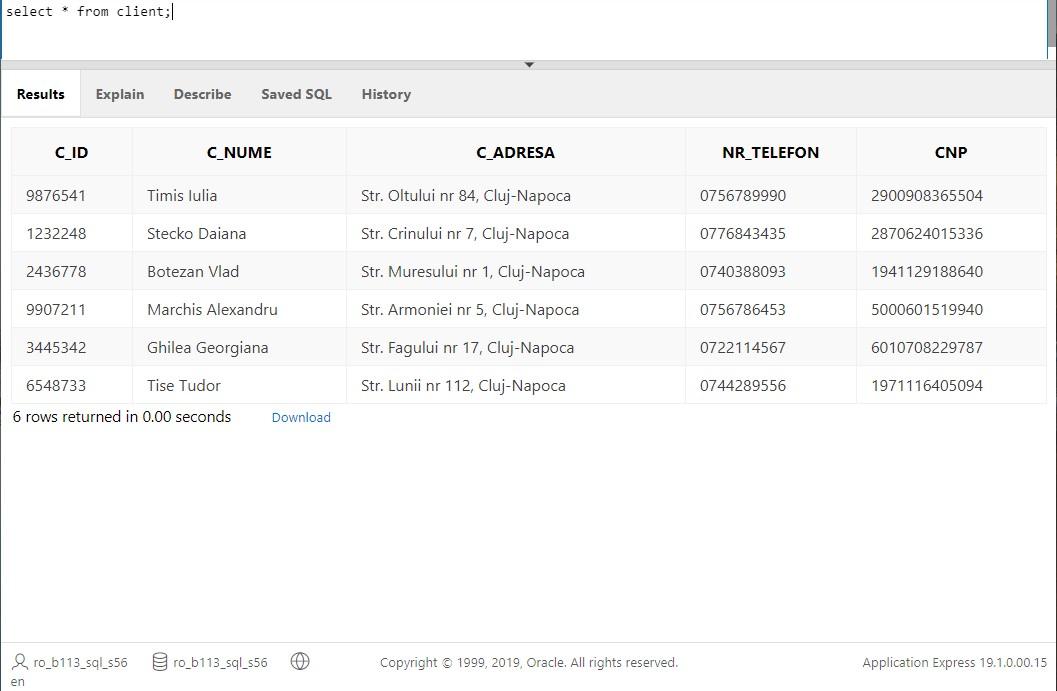
VALUES

(3445342,'Ghilea Georgiana','Str. Fagului nr 17, Cluj-Napoca', '0722114567','6010708229787');

INSERT INTO Client

VALUES

(6548733,'Tise Tudor','Str. Lunii nr 112, Cluj-Napoca', '0744289556','1971116405094');



* **Populare tabela Cont**

INSERT INTO CONT

VALUES

('RO49BUCU1223425687934536','LEI','7000','9876541','BUCU');

INSERT INTO CONT

VALUES

('RO30BTRL8767239861276454','EURO','1000','2436778','BTRL');

INSERT INTO CONT

VALUES

('RO30BTRL8767239861276452','LEI','1200','2436778','BTRL');

INSERT INTO CONT

VALUES

('RO17RNCB7623465987012006','LEI','30000','9907211','RNCB');

INSERT INTO CONT

VALUES

('RO78RZBR0004523477859671','EURO','300','1232248','RZBR');

INSERT INTO CONT

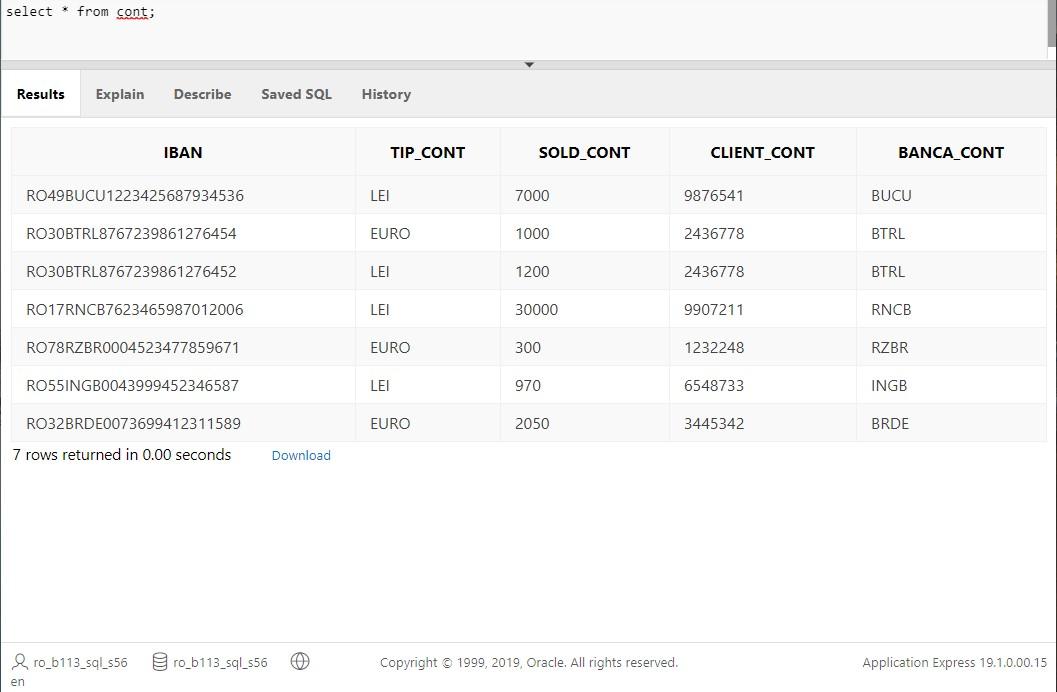
VALUES

('RO55INGB0043999452346587','LEI','970','6548733','INGB');

INSERT INTO CONT

VALUES

('RO32BRDE0073699412311589','EURO','2050','3445342','BRDE');



* **Populare tabela Card**

INSERT INTO CARD

VALUES

('4140789065456756','visa','552', to\_date('03/22', 'MM/YY'), 2341, to\_clob('Timis Iulia'),'RO49BUCU1223425687934536');

INSERT INTO CARD

VALUES

('5678563423654532','visa','222', to\_date('03/22', 'MM/YY'), 8642, to\_clob('Timis Maria'),'RO49BUCU1223425687934536');

INSERT INTO CARD

VALUES

('3457678498456998','visa','592', to\_date('01/23', 'MM/YY'), 9167, to\_clob('Botezan Vlad'),'RO30BTRL8767239861276452');

INSERT INTO CARD

VALUES

('1276980956476578','omnipass','983', to\_date('04/24', 'MM/YY'), 5761, to\_clob('Ghilea Georgiana'),'RO32BRDE0073699412311589');

INSERT INTO CARD

VALUES

('7658324258655875','visa','134', to\_date('01/22', 'MM/YY'), 4124, to\_clob('Tise Laura'),'RO55INGB0043999452346587');

INSERT INTO CARD

VALUES

('4778554677612345','visa','441', to\_date('01/22', 'MM/YY'), 1111, to\_clob('Tise Tudor'),'RO55INGB0043999452346587');

INSERT INTO CARD

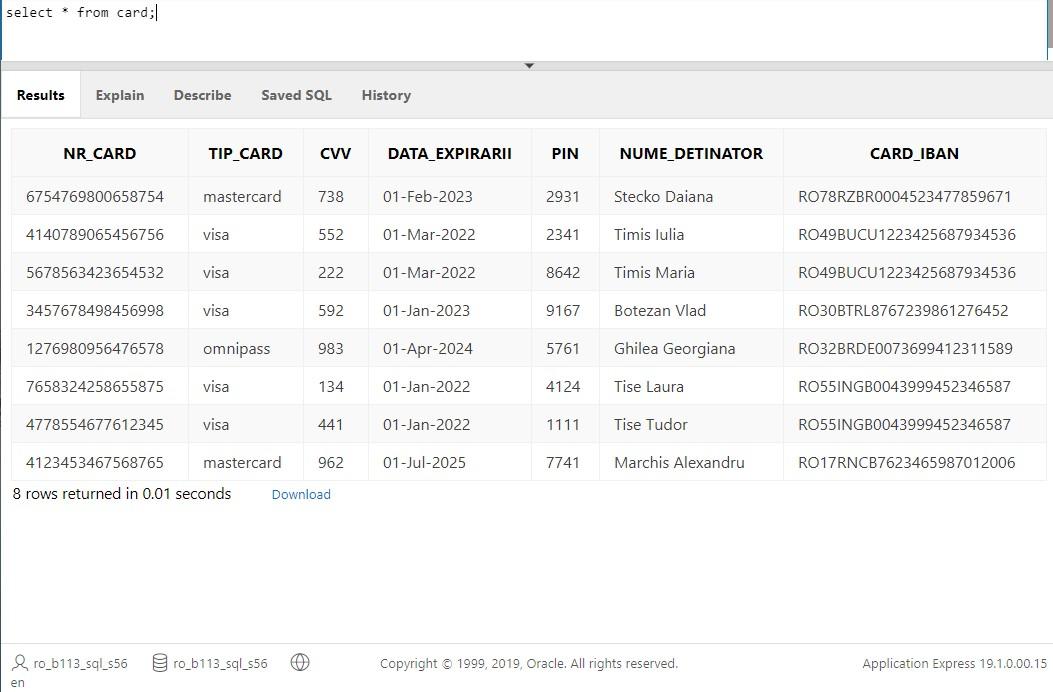
VALUES

('6754769800658754','mastercard','738', to\_date('02/23', 'MM/YY'), 2931, to\_clob('Stecko Daiana'),'RO78RZBR0004523477859671');

INSERT INTO CARD

VALUES

('4123453467568765','mastercard','962', to\_date('07/25', 'MM/YY'), 7741, to\_clob('Marchis Alexandru'),'RO17RNCB7623465987012006');



* **Populare tabela Tranzactie**

INSERT INTO Tranzactie(data\_ora, tr\_tip, tr\_valoare, tr\_client, tr\_ATM)

VALUES

(TIMESTAMP'2021-05-01 17:01:30','retragere numerar', 500,3445342, 1);

INSERT INTO Tranzactie(data\_ora, tr\_tip, tr\_valoare, tr\_client, tr\_ATM)

VALUES

(TIMESTAMP'2021-05-01 19:00:50','sold cont', NULL, 3445342, 4);

INSERT INTO Tranzactie(data\_ora, tr\_tip, tr\_valoare, tr\_client, tr\_ATM)

VALUES

(TIMESTAMP'2021-05-02 11:12:39','depunere numerar', 250, 1232248, 5);

INSERT INTO Tranzactie(data\_ora, tr\_tip, tr\_valoare, tr\_client, tr\_ATM)

VALUES

(TIMESTAMP'2021-05-03 13:54:19','schimbare pin', NULL, 9876541, 1);

INSERT INTO Tranzactie(data\_ora, tr\_tip, tr\_valoare, tr\_client, tr\_ATM)

VALUES

(TIMESTAMP'2021-05-06 08:01:48','schimb valutar', 450, 6548733, 6);

INSERT INTO Tranzactie(data\_ora, tr\_tip, tr\_valoare, tr\_client, tr\_ATM)

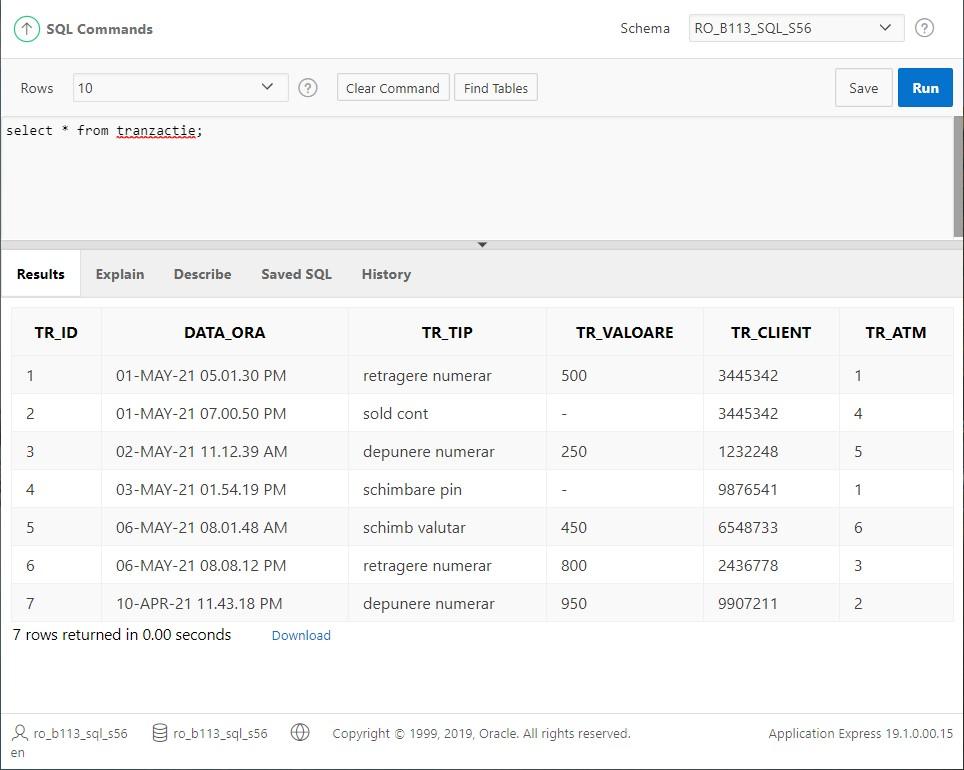
VALUES

(TIMESTAMP'2021-05-06 20:08:12','retragere numerar', 800, 2436778, 3);

INSERT INTO Tranzactie(data\_ora, tr\_tip, tr\_valoare, tr\_client, tr\_ATM)

VALUES

(TIMESTAMP'2021-04-10 23:43:18','depunere numerar', 950, 9907211, 2);



**PACKAGE**

CREATE OR REPLACE package pack\_atm as

function nr\_tranzactii(p\_tr IN client.c\_id%TYPE) return int;

function total\_sold(p\_id cont.client\_cont%type) return number;

procedure sterge\_tranzactie(p\_tr in tranzactie.tr\_id%type);

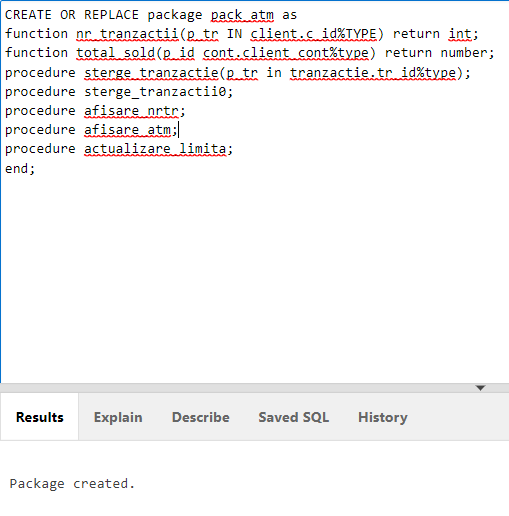
procedure sterge\_tranzactii0;

procedure afisare\_nrtr;

procedure afisare\_atm;

procedure actualizare\_limita;

end;

****

* **functie care returneaza nr de tranzactii efectuate de un client**

function nr\_tranzactii(p\_tr in client.c\_id%type) return int is

v\_tr INT :=0;

cursor nr\_tr is select tr\_id from tranzactie where tranzactie.tr\_client=p\_tr;

r\_tr nr\_tr%ROWTYPE;

begin

open nr\_tr;

loop

fetch nr\_tr into r\_tr;

exit when nr\_tr%NOTFOUND;

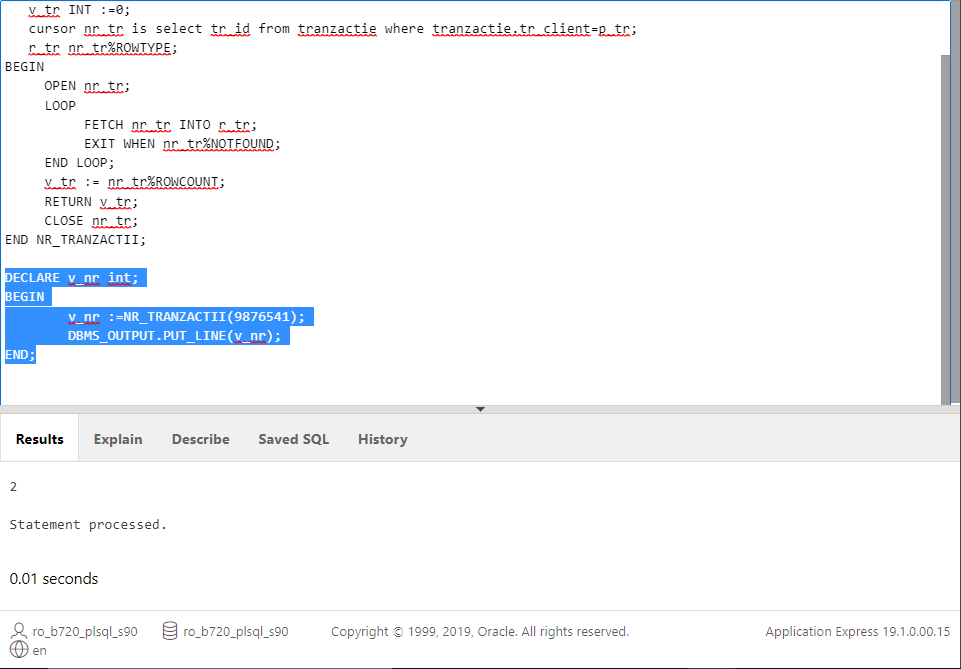
end loop;

v\_tr := nr\_tr%ROWCOUNT;

return v\_tr;

close nr\_tr;

end nr\_tranzactii;

****

* **functie care returneaza soldul conturilor detinute de acelasi client**

function total\_sold(p\_id cont.client\_cont%type) return number is

v\_total number :=0;

cursor c\_conturi is select sold\_cont from cont where client\_cont=p\_id;

begin

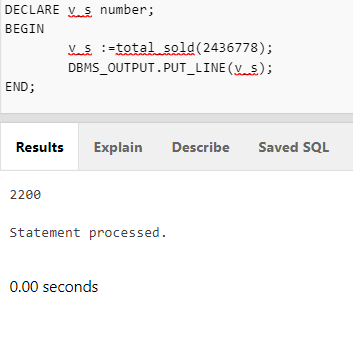
for v\_sold in c\_conturi loop

v\_total:=v\_total + v\_sold.sold\_cont;

end loop;

return v\_total;

end total\_sold;



* **procedura care sterge o tranzactie**

procedure sterge\_tranzactie(p\_tr in tranzactie.tr\_id%type) is

v\_tr tranzactie.tr\_id%type;

begin

select tr\_id

into v\_tr

from tranzactie

where tr\_id = p\_tr;

delete from tranzactie where tr\_id=p\_tr;

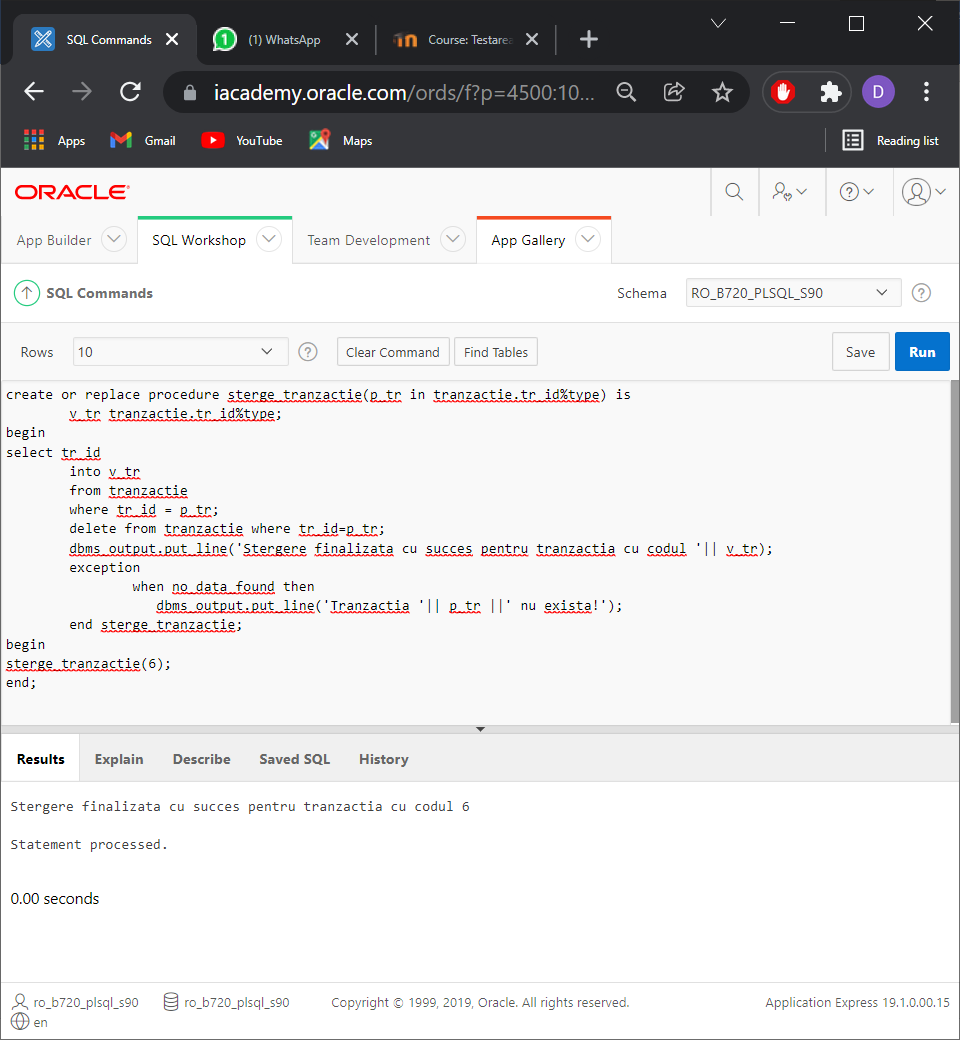
dbms\_output.put\_line('Stergere finalizata cu succes pentru tranzactia cu codul '|| v\_tr);

exception

when no\_data\_found then

dbms\_output.put\_line('Tranzactia '|| p\_tr ||' nu exista!');

end sterge\_tranzactie;



* **procedura care sterge tranzactiile care au valoare nula (ex: interogare sold) folosindu-ne de procedura sterge\_tranzactie**

procedure sterge\_tranzactii0 is

v\_tr tranzactie.tr\_id%type;

cursor c\_tr0 is select tr\_id from tranzactie where tr\_valoare is null;

begin

open c\_tr0;

loop

fetch c\_tr0 into v\_tr;

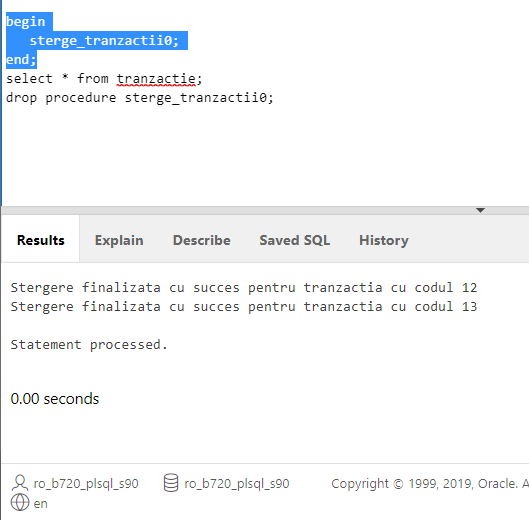
exit when c\_tr0%notfound;

sterge\_tranzactie(v\_tr);

end loop;

close c\_tr0;

end sterge\_tranzactii0;

****

* **procedura care afiseaza numele si nr de tranzactii efectuate de clientii din Cluj-Napoca folosindu-ne de functia nr\_tranzactii**

procedure afisare\_nrtr is

cursor c\_clienti is select c\_id, c\_nume, nr\_telefon

from client WHERE REGEXP\_LIKE (c\_adresa, 'Cluj Napoca') OR REGEXP\_LIKE (c\_adresa, 'Cluj-Napoca');

v\_nume client.c\_nume%type;

v\_nrtel client.nr\_telefon%type;

v\_nrtr int:=0;

v\_id client.c\_id%type;

begin

open c\_clienti;

loop

fetch c\_clienti into v\_id, v\_nume, v\_nrtel;

if c\_clienti%FOUND then

v\_nrtr := nr\_tranzactii(v\_id);

DBMS\_OUTPUT.PUT\_LINE(v\_nume || ' ' || v\_nrtel || ' ' || v\_nrtr);

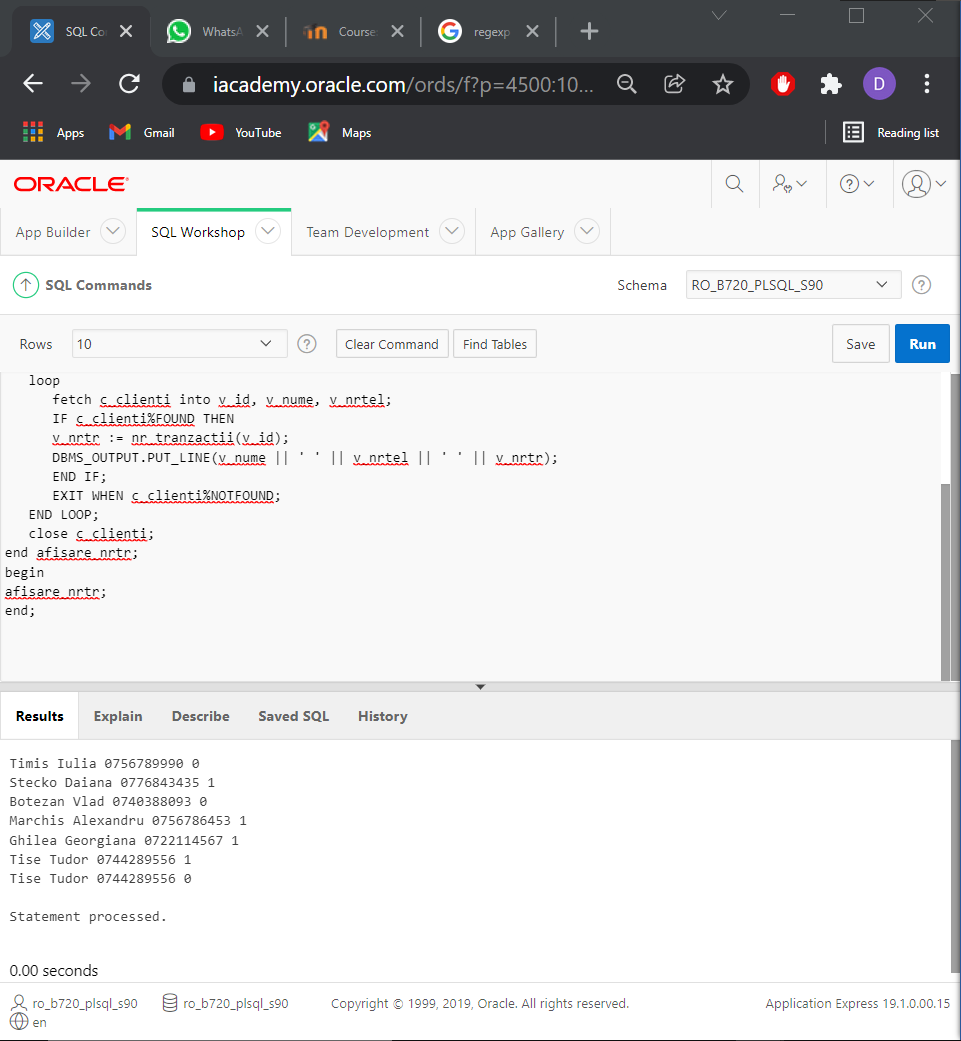
end if;

exit when c\_clienti%NOTFOUND;

end loop;

close c\_clienti;

end afisare\_nrtr;

****

* **procedura care afiseaza locatia si limita de cash a atm-urilor in functie de banca**

procedure afisare\_atm is

cursor c\_banci is select distinct atm\_banca from atm;

cursor c\_atm(p\_banci atm.atm\_banca%TYPE) is

select \* from atm where atm\_banca=p\_banci;

v\_contor int := 0;

v\_contor\_global int := 0;

begin

for v\_banci in c\_banci loop

DBMS\_OUTPUT.PUT\_LINE('======' || v\_banci.atm\_banca || '======');

for v\_atm in c\_atm(v\_banci.atm\_banca) loop

v\_contor\_global := v\_contor\_global + 1;

v\_contor := v\_contor + 1;

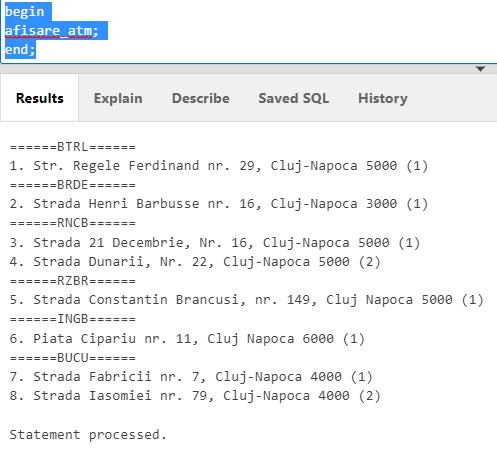
DBMS\_OUTPUT.PUT\_LINE(v\_contor\_global || '. ' || v\_atm.locatie || ' ' || v\_atm.limita\_cash || ' ' || '(' || v\_contor || ')');

end loop;

v\_contor := 0;

end loop;

end afisare\_atm;

****

* **procedura care actualizeazeaza limita de cash sub 4000 a bancomatelor, adaugand inca 1000**

procedure actualizare\_limita is

cursor limita\_cursor (max\_limit number) is

select atm\_id, limita\_cash

from atm where limita\_cash < max\_limit for update of limita\_cash;

v\_id atm.atm\_id%type;

v\_limita atm.limita\_cash%type;

begin

open limita\_cursor(4000);

loop

fetch limita\_cursor into v\_id, v\_limita;

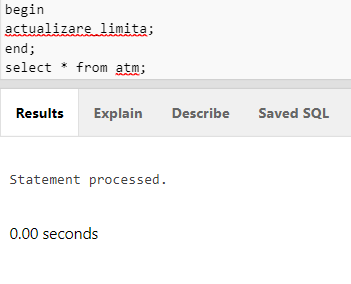
exit when limita\_cursor%NOTFOUND;

update atm set limita\_cash=v\_limita+1000 where v\_id=atm\_id;

end loop;

close limita\_cursor;

end;

****

**Declare**

**v\_nr int;**

**v\_s number;**

**Begin**

**v\_nr :=nr\_tranzactii(9876541);**

**DBMS\_OUTPUT.PUT\_LINE(v\_nr);**

**v\_s :=total\_sold(2436778);**

**DBMS\_OUTPUT.PUT\_LINE(v\_s);**

**sterge\_tranzactii0;**

**afisare\_nrtr;**

**afisare\_atm;**

**actualizare\_limita;**

**End;**

**SECVENTE**

CREATE SEQUENCE Agentie\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE ATM\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE Tranzactie\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

**DECLANSATORI**

* **declansatori utilizati pentru inserarea automata a valorilor de id pentru tabelele Agentie, ATM si Tranzactie**

CREATE OR REPLACE TRIGGER Agentie\_trg

BEFORE INSERT ON Agentie

FOR EACH ROW

BEGIN

  SELECT Agentie\_seq.NEXTVAL

  INTO   :new.a\_id

  FROM   dual;

END;

CREATE OR REPLACE TRIGGER ATM\_trg

BEFORE INSERT ON ATM

FOR EACH ROW

BEGIN

  SELECT ATM\_seq.NEXTVAL

  INTO   :new.ATM\_id

  FROM   dual;

END;

CREATE OR REPLACE TRIGGER Tranzactie\_trg

BEFORE INSERT ON Tranzactie

FOR EACH ROW

BEGIN

  SELECT Tranzactie\_seq.NEXTVAL

  INTO   :new.tr\_id

  FROM   dual;

END;

* **declansator utilizat pentru verificare valoarii introduse sau actualizate a campului limita\_cash pentru a fi in intervalul [2000, 10000]**

CREATE OR REPLACE TRIGGER limita\_cash

BEFORE INSERT OR UPDATE OF limita\_cash ON ATM

FOR EACH ROW

BEGIN

IF :NEW.limita\_cash < 2000 THEN

RAISE\_APPLICATION\_ERROR (-20202,'Limita minima admisa este 2000 lei!');

ELSIF :NEW.limita\_cash > 10000 THEN

RAISE\_APPLICATION\_ERROR (-20203,'Limita maxima admisa este 10000 lei!');

END IF;

END limita\_cash;