GESTIONARE COMPANIE BANCARA

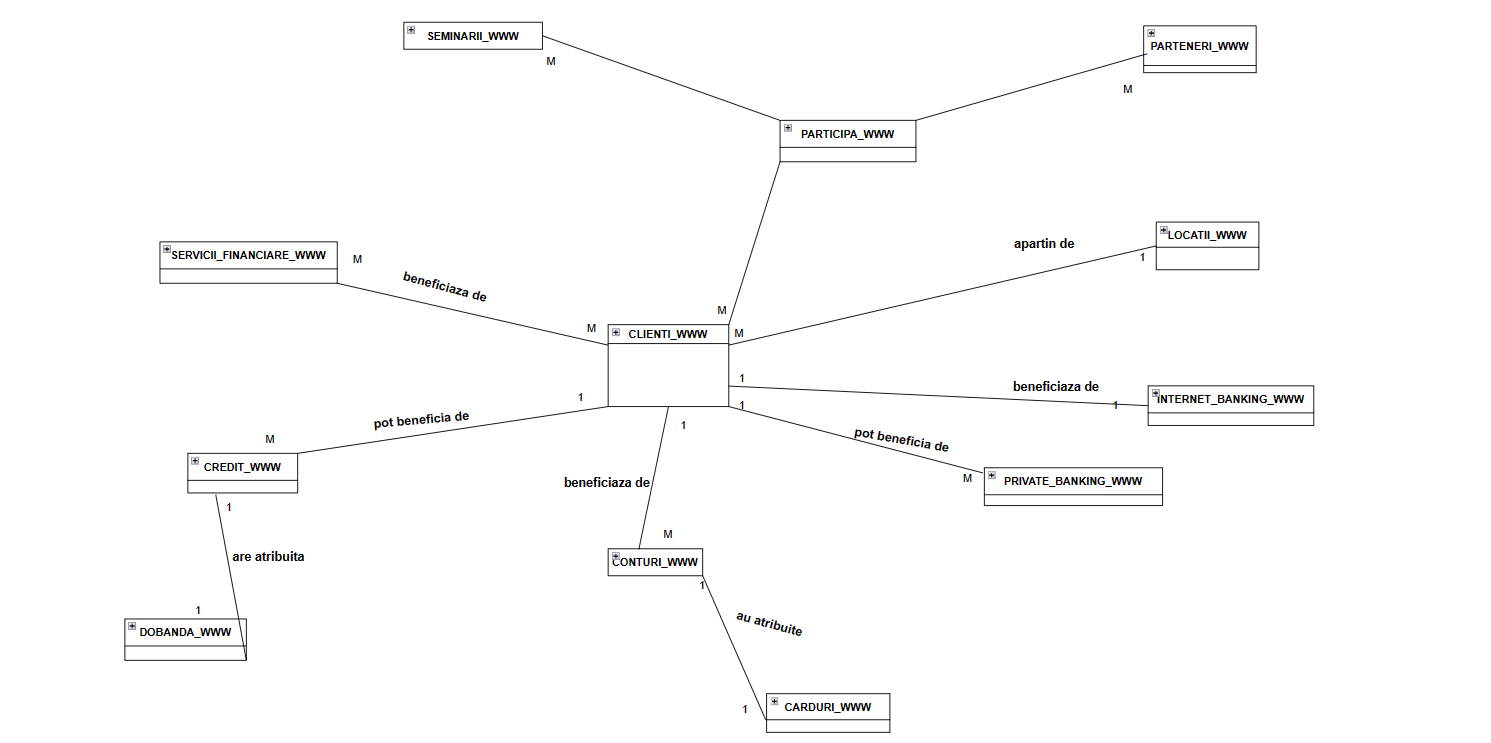
REALIZAT DE : DIACONESCU ALEXANDRA

GRUPA 244

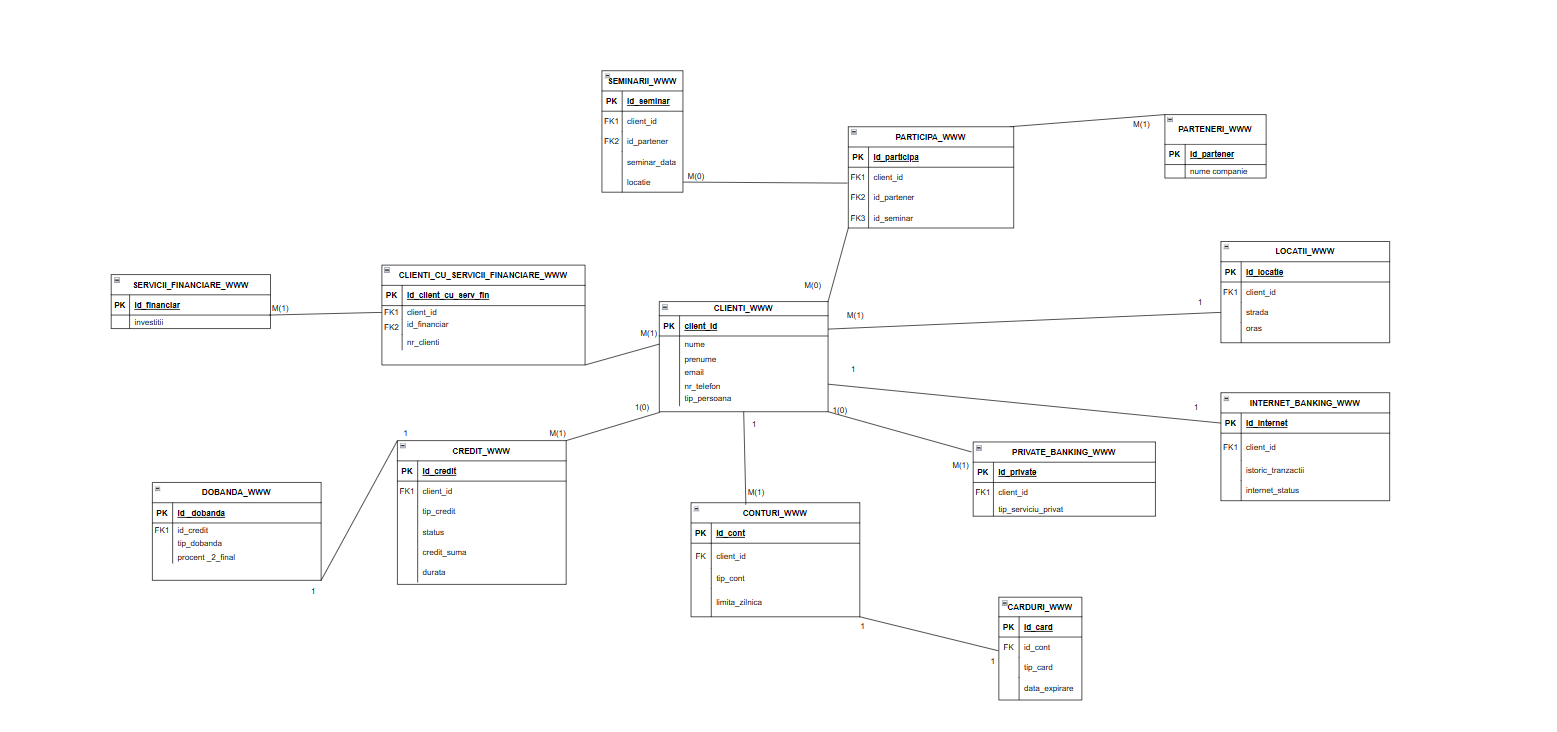
1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.

Proiectul consta in modelarea unei baze de date pentru sistemul de management al unei companii bancare ce ofera servicii bancare si financiare pentru clientii sai. Compania are sedii in diverse locatii si ofera o gama larga de produse si servicii bancare, cum ar fi conturi, credite pentru persoane fizice si juridice si plati internationale. In ceea ce priveste conturile bancare, clientii pot deschide conturi de economii sau conturi curente. In plus, compania ofera servicii de internet banking, prin intermediul carora clientii pot accesa conturile lor si pot efectua plati sau transferuri de bani. Creditele sunt disponibile la dobanzi variabile sau fixe. Clientii pot solicita credite pentru achizitionarea de bunuri, investitii sau pentru nevoi personale. Compania poate organiza si evenimente sau seminarii pentru educatia financiara a clientilor si pentru a le oferi mai multe informatii despre produsele si serviciile oferite. In functie de serviciile pe care le solicita, clientii pot deveni clienti private, si pot primi diverse oferte personalizate.

2.Realizați diagrama entitate-relație (ERD): entitățile, relațiile și atributele trebuie definite în limba română.



3.Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare: entitățile, relațiile și atributele trebuie definite în limba română.



4.Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, definind toate constrângerile de integritate necesare (chei primare, cheile externe etc).

SET FEEDBACK 1

SET LINESIZE 80

SET TRIMSPOOL ON

SET TAB OFF

SET PAGESIZE 100

SET ECHO OFF

--clienti--

CREATE TABLE clienti\_www (

client\_id NUMBER CONSTRAINT client\_id\_n NOT NULL,

nume VARCHAR2(30) CONSTRAINT nume\_n NOT NULL,

prenume VARCHAR2(30) CONSTRAINT prenume\_n NOT NULL,

email VARCHAR2(30) CONSTRAINT email\_n NOT NULL,

numar\_telefon NUMBER(10) CONSTRAINT numar\_telefon\_n NOT NULL,

tip\_persoana VARCHAR2(30) CONSTRAINT tip\_persoana\_n NOT NULL,

CONSTRAINT client\_id\_pk PRIMARY KEY (client\_id)

);

--servicii financiare--

CREATE TABLE servicii\_financiare\_www (

id\_financiar NUMBER,

investitii VARCHAR(50),

CONSTRAINT id\_financiar\_pk PRIMARY KEY(id\_financiar)

);

--clienti\_cu\_servicii\_financiare--

CREATE TABLE clienti\_cu\_servicii\_financiare\_www(

id\_client\_cu\_serv\_fin NUMBER CONSTRAINT id\_client\_cu\_serv\_fin\_n NOT NULL,

client\_id NUMBER CONSTRAINT client\_id\_x NOT NULL,

id\_financiar NUMBER CONSTRAINT id\_financiar\_x NOT NULL,

CONSTRAINT id\_client\_cu\_serv\_fin\_pk PRIMARY KEY (id\_client\_cu\_serv\_fin)

);

ALTER TABLE clienti\_cu\_servicii\_financiare\_www

ADD (CONSTRAINT clienti\_cu\_servicii\_financiare\_fk1 FOREIGN KEY( client\_id ) REFERENCES clienti\_www( client\_id ),

CONSTRAINT clienti\_cu\_servicii\_financiare\_fk2 FOREIGN KEY(id\_financiar ) REFERENCES servicii\_financiare\_www( id\_financiar )

);

--locatii--

CREATE TABLE locatii\_www (

id\_locatie NUMBER CONSTRAINT id\_locatie\_n NOT NULL,

client\_id NUMBER,

strada VARCHAR(20) CONSTRAINT strada\_id\_n NOT NULL,

oras VARCHAR(20) CONSTRAINT oras\_id\_n NOT NULL,

CONSTRAINT id\_locatie\_pk PRIMARY KEY(id\_locatie)

) ORGANIZATION INDEX;

ALTER TABLE locatii\_www

ADD ( CONSTRAINT locatii\_clienti\_fk FOREIGN KEY(client\_id) REFERENCES clienti\_www(client\_id));

--parteneri--

CREATE TABLE parteneri\_www(

id\_partener NUMBER CONSTRAINT id\_partener\_n NOT NULL,

nume\_companie VARCHAR(30),

CONSTRAINT id\_partener\_pk PRIMARY KEY(id\_partener)

);

--seminarii--

CREATE TABLE seminarii\_www(

id\_seminar NUMBER CONSTRAINT id\_seminar\_n NOT NULL,

client\_id NUMBER,

id\_partener NUMBER CONSTRAINT id\_partener\_nn NOT NULL,

seminar\_data VARCHAR(20) CONSTRAINT seminar\_data\_n NOT NULL,

locatie VARCHAR(20) CONSTRAINT locatie\_n NOT NULL,

CONSTRAINT id\_seminar\_pk PRIMARY KEY(id\_seminar)

);

ALTER TABLE seminarii\_www

ADD (CONSTRAINT seminarii\_fk FOREIGN KEY(client\_id) REFERENCES clienti\_www(client\_id),

CONSTRAINT parteneri\_fk FOREIGN KEY(id\_partener) REFERENCES parteneri\_www(id\_partener));

--credit--

CREATE TABLE credit\_www(

id\_credit NUMBER,

client\_id NUMBER,

tip\_credit VARCHAR(30),

status VARCHAR(20) ,

credit\_suma VARCHAR(20),

durata VARCHAR(20),

CONSTRAINT id\_credit\_pk PRIMARY KEY(id\_credit)

);

ALTER TABLE credit\_www

ADD (CONSTRAINT credit\_fk FOREIGN KEY(client\_id) REFERENCES clienti\_www(client\_id));

--dobanda--

CREATE TABLE dobanda\_www(

id\_dobanda NUMBER ,

id\_credit NUMBER ,

tip\_dobanda VARCHAR(30) ,

procent\_2\_final VARCHAR(20),

CONSTRAINT id\_dobanda\_pk PRIMARY KEY(id\_dobanda)

);

ALTER TABLE dobanda\_www

ADD (CONSTRAINT dobanda\_fk FOREIGN KEY( id\_credit ) REFERENCES credit\_www( id\_credit ));

--conturi--

CREATE TABLE conturi\_www(

id\_cont NUMBER CONSTRAINT cont\_n NOT NULL,

client\_id NUMBER CONSTRAINT client\_n NOT NULL,

tip\_cont VARCHAR2(10) CONSTRAINT tip\_cont\_n NOT NULL,

limita\_zilnica NUMBER CONSTRAINT limita\_n NOT NULL,

CONSTRAINT id\_cont\_pk PRIMARY KEY(id\_cont)

);

ALTER TABLE conturi\_www

ADD (CONSTRAINT conturi\_fk FOREIGN KEY( client\_id ) REFERENCES clienti\_www( client\_id ));

--carduri--

CREATE TABLE carduri\_www(

id\_card NUMBER CONSTRAINT card\_n NOT NULL,

id\_cont NUMBER CONSTRAINT cont\_nn NOT NULL,

tip\_card VARCHAR2(10) CONSTRAINT tip\_card\_n NOT NULL,

data\_expirare VARCHAR(20) CONSTRAINT data\_exp\_n NOT NULL,

CONSTRAINT id\_card\_pk PRIMARY KEY(id\_card)

);

ALTER TABLE carduri\_www

ADD (CONSTRAINT carduri\_fk FOREIGN KEY( id\_cont ) REFERENCES conturi\_www(id\_cont));

--private banking--

CREATE TABLE private\_banking\_www(

id\_private NUMBER,

client\_id NUMBER,

tip\_serviciu\_privat VARCHAR(50),

CONSTRAINT id\_private\_pk PRIMARY KEY(id\_private)

);

ALTER TABLE private\_banking\_www

ADD (CONSTRAINT private\_fk FOREIGN KEY( client\_id ) REFERENCES clienti\_www( client\_id ));

--internet banking--

CREATE TABLE internet\_banking\_www(

id\_internet NUMBER CONSTRAINT int\_n NOT NULL,

client\_id NUMBER CONSTRAINT cli\_n NOT NULL,

istoric\_tranzactii VARCHAR(2000),

internet\_status VARCHAR(20) CONSTRAINT int\_status\_n NOT NULL,

CONSTRAINT id\_internet\_pk PRIMARY KEY(id\_internet)

);

ALTER TABLE internet\_banking\_www

ADD (CONSTRAINT internet\_fk FOREIGN KEY( client\_id ) REFERENCES clienti\_www( client\_id ));

--participa--

CREATE TABLE participa\_www(

id\_participa NUMBER,

id\_partener NUMBER,

client\_id NUMBER,

id\_seminar NUMBER,

CONSTRAINT id\_participa\_pk PRIMARY KEY(id\_participa)

);

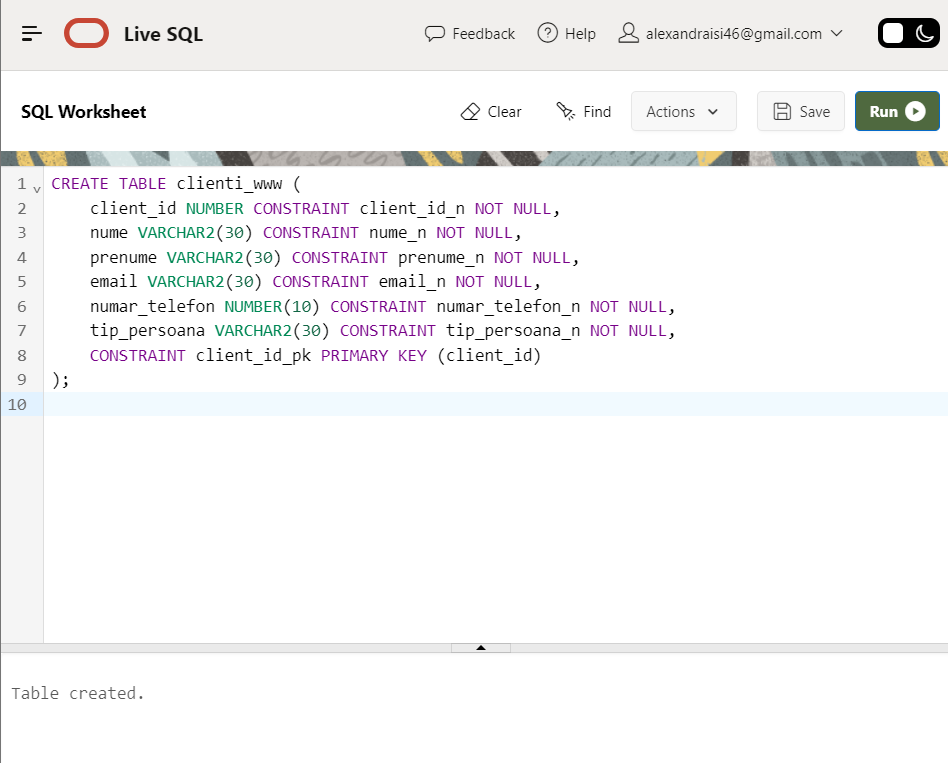
ALTER TABLE participa\_www

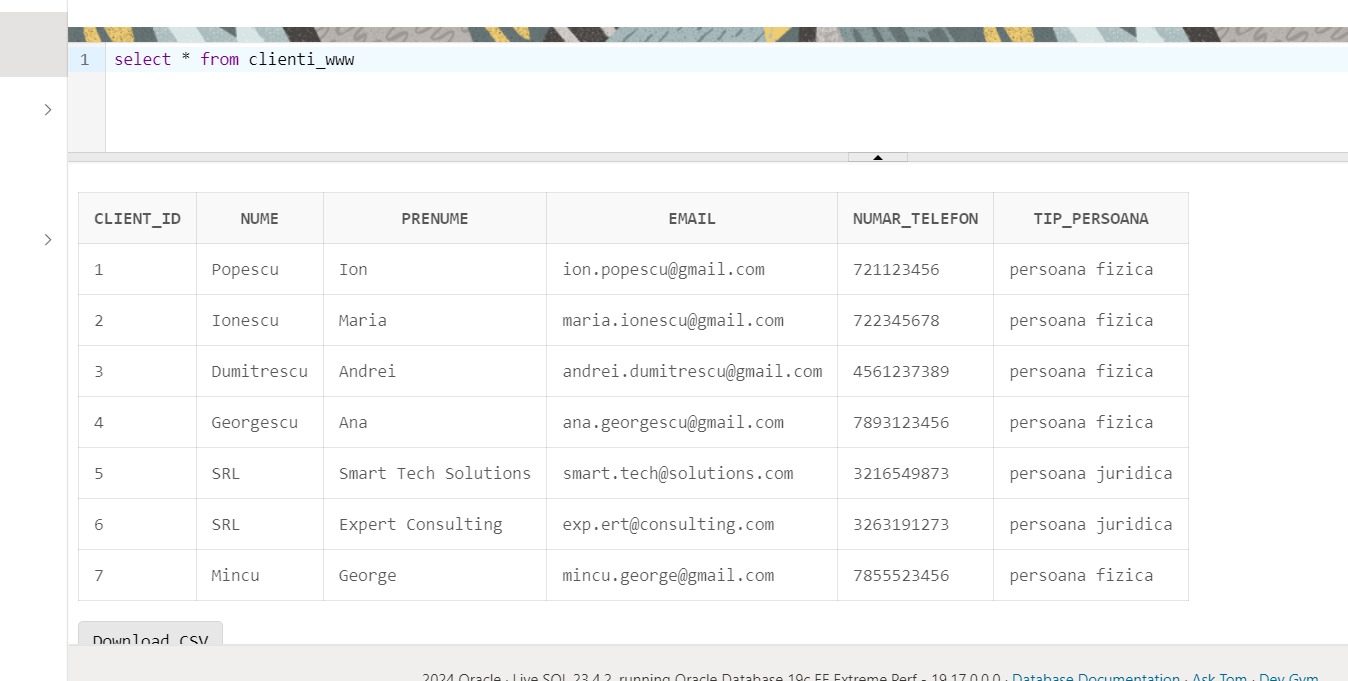
ADD (CONSTRAINT participa\_fk FOREIGN KEY( id\_seminar ) REFERENCES seminarii\_www( id\_seminar ),

CONSTRAINT participa\_fk1 FOREIGN KEY(id\_partener ) REFERENCES parteneri\_www( id\_partener ),

CONSTRAINT participa\_fk2 FOREIGN KEY( client\_id ) REFERENCES clienti\_www( client\_id )

);





5. Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

CREATE SEQUENCE clienti\_seq START WITH 1 INCREMENT BY 1;

INSERT INTO clienti\_www (client\_id, nume, prenume, email, numar\_telefon, tip\_persoana)

VALUES (clienti\_seq.NEXTVAL, 'Popescu', 'Ion', 'ion.popescu@gmail.com', 0721123456, 'persoana fizica');

INSERT INTO clienti\_www (client\_id, nume, prenume, email, numar\_telefon, tip\_persoana)

VALUES (clienti\_seq.NEXTVAL, 'Ionescu', 'Maria', 'maria.ionescu@gmail.com', 0722345678, 'persoana fizica');

INSERT INTO clienti\_www (client\_id, nume, prenume, email, numar\_telefon, tip\_persoana)

VALUES (clienti\_seq.NEXTVAL, 'Dumitrescu', 'Andrei', 'andrei.dumitrescu@gmail.com', 4561237389, 'persoana fizica');

INSERT INTO clienti\_www (client\_id, nume, prenume, email, numar\_telefon, tip\_persoana)

VALUES (clienti\_seq.NEXTVAL, 'Georgescu', 'Ana', 'ana.georgescu@gmail.com', 7893123456, 'persoana fizica');

INSERT INTO clienti\_www (client\_id, nume, prenume, email, numar\_telefon, tip\_persoana)

VALUES (clienti\_seq.NEXTVAL, 'SRL', 'Smart Tech Solutions', 'smart.tech@solutions.com', 3216549873, 'persoana juridica');

INSERT INTO clienti\_www (client\_id, nume, prenume, email, numar\_telefon, tip\_persoana)

VALUES (clienti\_seq.NEXTVAL, 'SRL', 'Expert Consulting', 'exp.ert@consulting.com', 3263191273, 'persoana juridica');

INSERT INTO clienti\_www (client\_id, nume, prenume, email, numar\_telefon, tip\_persoana)

VALUES (clienti\_seq.NEXTVAL, 'Mincu', 'George', 'mincu.george@gmail.com', 7855523456, 'persoana fizica' );

CREATE SEQUENCE servicii\_financiare\_seq START WITH 10 INCREMENT BY 3;

INSERT INTO servicii\_financiare\_www (id\_financiar, investitii)

VALUES (servicii\_financiare\_seq.NEXTVAL, 'Fonduri mutuale');

INSERT INTO servicii\_financiare\_www (id\_financiar, investitii)

VALUES (servicii\_financiare\_seq.NEXTVAL, 'Investitii NASDAQ');

INSERT INTO servicii\_financiare\_www (id\_financiar, investitii)

VALUES (servicii\_financiare\_seq.NEXTVAL, 'Investitii crypto');

INSERT INTO servicii\_financiare\_www (id\_financiar, investitii)

VALUES (servicii\_financiare\_seq.NEXTVAL, 'Investitii in companii');

INSERT INTO servicii\_financiare\_www (id\_financiar, investitii)

VALUES (servicii\_financiare\_seq.NEXTVAL, 'Investitii imobiliare');

INSERT INTO servicii\_financiare\_www (id\_financiar, investitii)

VALUES (servicii\_financiare\_seq.NEXTVAL, 'Investitii BVB');

CREATE SEQUENCE clienti\_cu\_servicii\_financiare\_seq START WITH 500 INCREMENT BY 1;

INSERT INTO clienti\_cu\_servicii\_financiare\_www (id\_client\_cu\_serv\_fin, client\_id, id\_financiar)

VALUES (clienti\_cu\_servicii\_financiare\_seq.NEXTVAL, 1, 10);

INSERT INTO clienti\_cu\_servicii\_financiare\_www (id\_client\_cu\_serv\_fin, client\_id, id\_financiar)

VALUES (clienti\_cu\_servicii\_financiare\_seq.NEXTVAL, 2, 10);

INSERT INTO clienti\_cu\_servicii\_financiare\_www (id\_client\_cu\_serv\_fin, client\_id, id\_financiar)

VALUES (clienti\_cu\_servicii\_financiare\_seq.NEXTVAL, 3, 13);

INSERT INTO clienti\_cu\_servicii\_financiare\_www (id\_client\_cu\_serv\_fin, client\_id, id\_financiar)

VALUES (clienti\_cu\_servicii\_financiare\_seq.NEXTVAL, 3, 16);

INSERT INTO clienti\_cu\_servicii\_financiare\_www (id\_client\_cu\_serv\_fin, client\_id, id\_financiar)

VALUES (clienti\_cu\_servicii\_financiare\_seq.NEXTVAL, 4, 19);

INSERT INTO clienti\_cu\_servicii\_financiare\_www (id\_client\_cu\_serv\_fin, client\_id, id\_financiar)

VALUES (clienti\_cu\_servicii\_financiare\_seq.NEXTVAL, 5, 22);

INSERT INTO clienti\_cu\_servicii\_financiare\_www (id\_client\_cu\_serv\_fin, client\_id, id\_financiar)

VALUES (clienti\_cu\_servicii\_financiare\_seq.NEXTVAL, 6, 22);

INSERT INTO clienti\_cu\_servicii\_financiare\_www (id\_client\_cu\_serv\_fin, client\_id, id\_financiar)

VALUES (clienti\_cu\_servicii\_financiare\_seq.NEXTVAL, 7, 22);

INSERT INTO clienti\_cu\_servicii\_financiare\_www (id\_client\_cu\_serv\_fin, client\_id, id\_financiar)

VALUES (clienti\_cu\_servicii\_financiare\_seq.NEXTVAL, 2, 22);

INSERT INTO clienti\_cu\_servicii\_financiare\_www (id\_client\_cu\_serv\_fin, client\_id, id\_financiar)

VALUES (clienti\_cu\_servicii\_financiare\_seq.NEXTVAL, 3, 22);

CREATE SEQUENCE parteneri\_seq START WITH 70 INCREMENT BY 1;

INSERT INTO parteneri\_www (id\_partener, nume\_companie)

VALUES (parteneri\_seq.NEXTVAL, 'BCR');

INSERT INTO parteneri\_www (id\_partener, nume\_companie)

VALUES (parteneri\_seq.NEXTVAL, 'BRD');

INSERT INTO parteneri\_www (id\_partener, nume\_companie)

VALUES (parteneri\_seq.NEXTVAL, 'ING');

INSERT INTO parteneri\_www (id\_partener, nume\_companie)

VALUES (parteneri\_seq.NEXTVAL, 'Raiffeisen Bank');

INSERT INTO parteneri\_www (id\_partener, nume\_companie)

VALUES (parteneri\_seq.NEXTVAL, 'Banca Transilvania');

INSERT INTO parteneri\_www (id\_partener, nume\_companie)

VALUES (parteneri\_seq.NEXTVAL, 'UniCredit Bank');

CREATE SEQUENCE seminarii\_seq START WITH 50 INCREMENT BY 1;

INSERT INTO seminarii\_www (id\_seminar, client\_id, id\_partener, seminar\_data, locatie)

VALUES (seminarii\_seq.NEXTVAL, 1, 70,'01-Jan-22', 'Bucuresti');

INSERT INTO seminarii\_www (id\_seminar, client\_id, id\_partener, seminar\_data, locatie)

VALUES (seminarii\_seq.NEXTVAL, 2, 71, '15-Mar-22', 'Cluj-Napoca');

INSERT INTO seminarii\_www (id\_seminar, client\_id, id\_partener, seminar\_data, locatie)

VALUES (seminarii\_seq.NEXTVAL, 3, 72, '10-Apr-22', 'Timisoara');

INSERT INTO seminarii\_www (id\_seminar, client\_id, id\_partener, seminar\_data, locatie)

VALUES (seminarii\_seq.NEXTVAL, 4, 73, '25-Jun-22', 'Iasi');

INSERT INTO seminarii\_www (id\_seminar, client\_id, id\_partener, seminar\_data, locatie)

VALUES (seminarii\_seq.NEXTVAL, 5, 74, '05-Jul-22', 'Constanta');

INSERT INTO seminarii\_www (id\_seminar, client\_id, id\_partener, seminar\_data, locatie)

VALUES (seminarii\_seq.NEXTVAL, 6, 75, '20-Aug-22', 'Brasov');

INSERT INTO seminarii\_www (id\_seminar, client\_id, id\_partener, seminar\_data, locatie)

VALUES (seminarii\_seq.NEXTVAL,1, 75, '15-Nov-22', 'Sibiu');

CREATE SEQUENCE locatii\_seq START WITH 80 INCREMENT BY 1;

INSERT INTO locatii\_www (id\_locatie, client\_id, strada, oras)

VALUES (locatii\_seq.NEXTVAL,1, 'Titulescu', 'Bucuresti');

INSERT INTO locatii\_www (id\_locatie, client\_id, strada, oras)

VALUES (locatii\_seq.NEXTVAL,2, 'Titulescu', 'Bucuresti');

INSERT INTO locatii\_www (id\_locatie, client\_id, strada, oras)

VALUES (locatii\_seq.NEXTVAL,3, 'Avram Iancu', 'Cluj-Napoca');

INSERT INTO locatii\_www (id\_locatie, client\_id, strada, oras)

VALUES (locatii\_seq.NEXTVAL, 4, 'Victoriei', 'Bucuresti');

INSERT INTO locatii\_www (id\_locatie, client\_id, strada, oras)

VALUES (locatii\_seq.NEXTVAL, 5, 'Icoanei', 'Iasi');

INSERT INTO locatii\_www (id\_locatie, client\_id, strada, oras)

VALUES (locatii\_seq.NEXTVAL, 6, 'Eroilor', 'Timisoara');

INSERT INTO locatii\_www (id\_locatie, client\_id, strada, oras)

VALUES (locatii\_seq.NEXTVAL, 7, 'Unirii', 'Bucuresti');

CREATE SEQUENCE participa\_seq START WITH 100 INCREMENT BY 1;

INSERT INTO participa\_www (id\_participa, id\_partener, client\_id, id\_seminar)

VALUES (participa\_seq.NEXTVAL, 70, 1, 50);

INSERT INTO participa\_www (id\_participa, id\_partener, client\_id, id\_seminar)

VALUES(participa\_seq.NEXTVAL, 71, 2, 51);

INSERT INTO participa\_www (id\_participa, id\_partener, client\_id, id\_seminar)

VALUES (participa\_seq.NEXTVAL, 71, 3, 52);

INSERT INTO participa\_www (id\_participa, id\_partener, client\_id, id\_seminar)

VALUES(participa\_seq.NEXTVAL, 74, 4, 53);

INSERT INTO participa\_www (id\_participa, id\_partener, client\_id, id\_seminar)

VALUES(participa\_seq.NEXTVAL, 73, 5, 54);

INSERT INTO participa\_www (id\_participa, id\_partener, client\_id, id\_seminar)

VALUES(participa\_seq.NEXTVAL, 72, 6, 55);

INSERT INTO participa\_www (id\_participa, id\_partener, client\_id, id\_seminar)

VALUES(participa\_seq.NEXTVAL, 75, 1, 56);

INSERT INTO participa\_www (id\_participa, id\_partener, client\_id, id\_seminar)

VALUES(participa\_seq.NEXTVAL, 75, 2, 56);

INSERT INTO participa\_www (id\_participa, id\_partener, client\_id, id\_seminar)

VALUES(participa\_seq.NEXTVAL, 72, 3, 55);

INSERT INTO participa\_www (id\_participa, id\_partener, client\_id, id\_seminar)

VALUES(participa\_seq.NEXTVAL, 73, 4, 55);

INSERT INTO participa\_www (id\_participa, id\_partener, client\_id, id\_seminar)

VALUES(participa\_seq.NEXTVAL, 73, 7, 51);

CREATE SEQUENCE credit\_seq START WITH 170 INCREMENT BY 1;

INSERT INTO credit\_www (id\_credit, client\_id, tip\_credit, status, credit\_suma, durata)

VALUES (credit\_seq.NEXTVAL,1,'investitii','Aprobat','10000', '2 ani');

INSERT INTO credit\_www (id\_credit, client\_id, tip\_credit, status, credit\_suma, durata)

VALUES (credit\_seq.NEXTVAL, 2, 'investitii', 'Aprobat', '500000', '5 ani');

INSERT INTO credit\_www (id\_credit, client\_id, tip\_credit, status, credit\_suma, durata)

VALUES(credit\_seq.NEXTVAL, 3, 'nevoi personale', 'Aprobat', '25000', '5 ani');

INSERT INTO credit\_www (id\_credit, client\_id, tip\_credit, status, credit\_suma, durata)

VALUES(credit\_seq.NEXTVAL, 4, 'over draft', 'In asteptare', '15000', '2 ani');

INSERT INTO credit\_www (id\_credit, client\_id, tip\_credit, status, credit\_suma, durata)

VALUES(credit\_seq.NEXTVAL, 5, 'investitii', 'Aprobat', '750000', '20 ani');

INSERT INTO credit\_www (id\_credit, client\_id, tip\_credit, status, credit\_suma, durata)

VALUES(credit\_seq.NEXTVAL, 6, NULL, NULL,NULL, NULL);

INSERT INTO credit\_www (id\_credit, client\_id, tip\_credit, status, credit\_suma, durata)

VALUES(credit\_seq.NEXTVAL, 7, NULL, NULL,NULL, NULL);

CREATE SEQUENCE dobanda\_seq START WITH 150 INCREMENT BY 1;

INSERT INTO dobanda\_www(id\_dobanda, id\_credit, tip\_dobanda, procent\_2\_final)

VALUES (dobanda\_seq.NEXTVAL,170,'variabila','5%');

INSERT INTO dobanda\_www (id\_dobanda, id\_credit, tip\_dobanda, procent\_2\_final)

VALUES (dobanda\_seq.NEXTVAL, 171, 'fixa', '3.5%');

INSERT INTO dobanda\_www (id\_dobanda, id\_credit, tip\_dobanda, procent\_2\_final)

VALUES(dobanda\_seq.NEXTVAL, 172, 'variabila', '6%');

INSERT INTO dobanda\_www (id\_dobanda, id\_credit, tip\_dobanda, procent\_2\_final)

VALUES(dobanda\_seq.NEXTVAL, 173, 'fixa', '4.25%');

INSERT INTO dobanda\_www (id\_dobanda, id\_credit, tip\_dobanda, procent\_2\_final)

VALUES(dobanda\_seq.NEXTVAL, 174, 'variabila', '5.75%');

INSERT INTO dobanda\_www (id\_dobanda, id\_credit, tip\_dobanda, procent\_2\_final)

VALUES(dobanda\_seq.NEXTVAL, 175, NULL, NULL);

INSERT INTO dobanda\_www (id\_dobanda, id\_credit, tip\_dobanda, procent\_2\_final)

VALUES(dobanda\_seq.NEXTVAL, 176, NULL, NULL);

CREATE SEQUENCE conturi\_seq START WITH 180 INCREMENT BY 1;

INSERT INTO conturi\_www (id\_cont,client\_id, tip\_cont, limita\_zilnica)

VALUES (conturi\_seq.NEXTVAL,1,'economii','4000');

INSERT INTO conturi\_www (id\_cont, client\_id, tip\_cont, limita\_zilnica)

VALUES (conturi\_seq.NEXTVAL, 2, 'curent', '2000');

INSERT INTO conturi\_www (id\_cont, client\_id, tip\_cont, limita\_zilnica)

VALUES(conturi\_seq.NEXTVAL, 3, 'economii', '5000');

INSERT INTO conturi\_www (id\_cont, client\_id, tip\_cont, limita\_zilnica)

VALUES(conturi\_seq.NEXTVAL, 4, 'curent', '1000');

INSERT INTO conturi\_www (id\_cont, client\_id, tip\_cont, limita\_zilnica)

VALUES(conturi\_seq.NEXTVAL, 5, 'curent', '2000');

INSERT INTO conturi\_www (id\_cont, client\_id, tip\_cont, limita\_zilnica)

VALUES(conturi\_seq.NEXTVAL, 6, 'economii', '8000');

INSERT INTO conturi\_www (id\_cont, client\_id, tip\_cont, limita\_zilnica)

VALUES(conturi\_seq.NEXTVAL, 7, 'economii', '6000');

CREATE SEQUENCE carduri\_seq START WITH 190 INCREMENT BY 1;

INSERT INTO CARDURI\_www (id\_card, id\_cont, tip\_card, data\_expirare)

VALUES (carduri\_seq.NEXTVAL,180,'debit','01-Jun-26');

INSERT INTO CARDURI\_www (id\_card, id\_cont, tip\_card, data\_expirare)

VALUES (carduri\_seq.NEXTVAL, 181, 'credit', '01-Dec-25');

INSERT INTO CARDURI\_www (id\_card, id\_cont, tip\_card, data\_expirare)

VALUES(carduri\_seq.NEXTVAL, 182, 'debit', '01-Sep-27');

INSERT INTO CARDURI\_www (id\_card, id\_cont, tip\_card, data\_expirare)

VALUES(carduri\_seq.NEXTVAL, 183, 'credit', '01-Apr-24');

INSERT INTO CARDURI\_www (id\_card, id\_cont, tip\_card, data\_expirare)

VALUES(carduri\_seq.NEXTVAL, 184, 'credit', '01-Feb-28');

INSERT INTO CARDURI\_www (id\_card, id\_cont, tip\_card, data\_expirare)

VALUES(carduri\_seq.NEXTVAL, 185, 'debit', '01-Nov-26');

INSERT INTO CARDURI\_www (id\_card, id\_cont, tip\_card, data\_expirare)

VALUES(carduri\_seq.NEXTVAL, 186, 'debit', '01-Nov-29');

CREATE SEQUENCE private\_seq START WITH 200 INCREMENT BY 1;

INSERT INTO private\_banking\_www (id\_private, client\_id, tip\_serviciu\_privat)

VALUES (private\_seq.NEXTVAL,1,'Gestionare portofoliu ');

INSERT INTO private\_banking\_www (id\_private, client\_id, tip\_serviciu\_privat)

VALUES (private\_seq.NEXTVAL, 2, 'Consultanta pentru planificarea financiara');

INSERT INTO private\_banking\_www (id\_private, client\_id, tip\_serviciu\_privat)

VALUES (private\_seq.NEXTVAL, 3, 'Servicii de investitii la nivel inalt');

INSERT INTO private\_banking\_www (id\_private, client\_id, tip\_serviciu\_privat)

VALUES (private\_seq.NEXTVAL, 4, NULL);

INSERT INTO private\_banking\_www (id\_private, client\_id, tip\_serviciu\_privat)

VALUES (private\_seq.NEXTVAL, 5, NULL);

INSERT INTO private\_banking\_www (id\_private, client\_id, tip\_serviciu\_privat)

VALUES (private\_seq.NEXTVAL, 6, NULL);

INSERT INTO private\_banking\_www (id\_private, client\_id, tip\_serviciu\_privat)

VALUES (private\_seq.NEXTVAL, 7, NULL);

CREATE SEQUENCE internet\_seq START WITH 220 INCREMENT BY 1;

INSERT INTO internet\_banking\_www (id\_internet, client\_id, istoric\_tranzactii,internet\_status)

VALUES (internet\_seq.NEXTVAL,1,'plata interna','activ');

INSERT INTO internet\_banking\_www (id\_internet, client\_id, istoric\_tranzactii, internet\_status)

VALUES (internet\_seq.NEXTVAL, 2, 'plata interna', 'activ');

INSERT INTO internet\_banking\_www (id\_internet, client\_id, istoric\_tranzactii, internet\_status)

VALUES(internet\_seq.NEXTVAL, 3, 'plata externa', 'activ');

INSERT INTO internet\_banking\_www (id\_internet, client\_id, istoric\_tranzactii, internet\_status)

VALUES(internet\_seq.NEXTVAL, 4, 'plata online', 'activ');

INSERT INTO internet\_banking\_www (id\_internet, client\_id, istoric\_tranzactii, internet\_status)

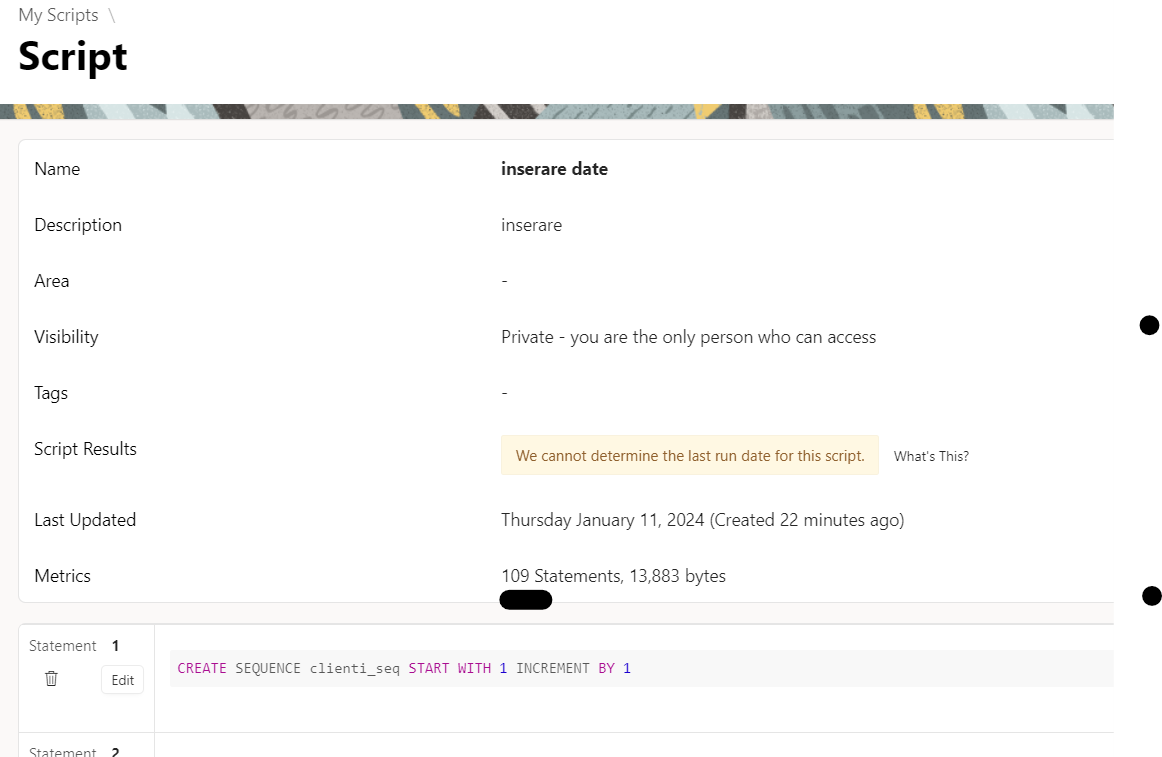
VALUES(internet\_seq.NEXTVAL, 5, 'plata externa', 'activ');

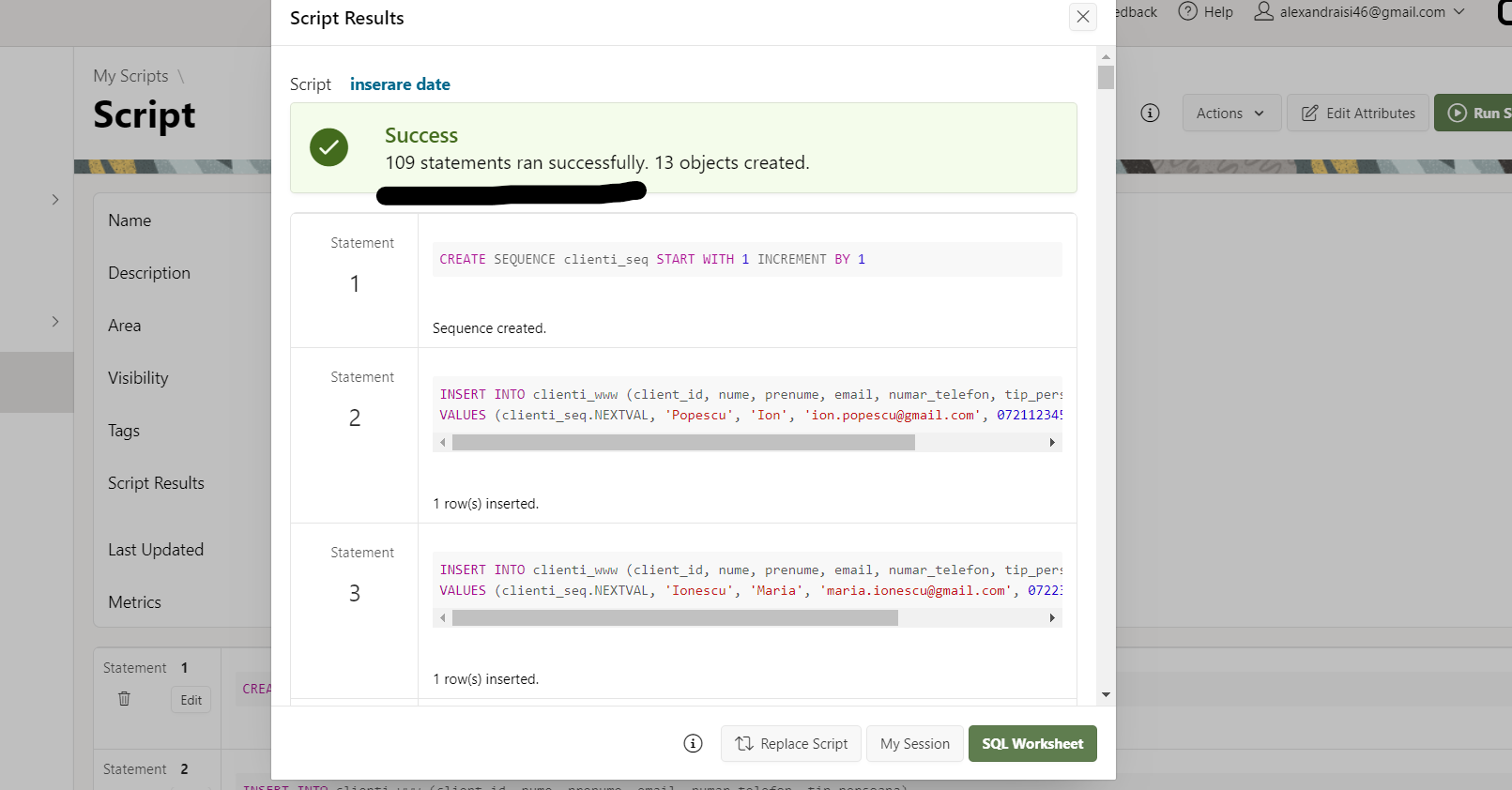
INSERT INTO internet\_banking\_www (id\_internet, client\_id, istoric\_tranzactii, internet\_status)

VALUES(internet\_seq.NEXTVAL, 6, 'plata online', 'activ');

INSERT INTO internet\_banking\_www (id\_internet, client\_id, istoric\_tranzactii, internet\_status)

VALUES(internet\_seq.NEXTVAL, 7, 'plata online', 'activ');





6. Scrie un subprogram stocat independent care creeaza un nou tabel numit clienti\_yyy care retine primii 2 clienti din clienti\_www si afiseaza numarul de telefon al fiecarui client, impreuna cu limita cardului si cu seminariile la care au participat.

CREATE OR REPLACE TYPE telefon\_client AS VARRAY(5) OF VARCHAR2(30);

CREATE OR REPLACE TYPE locatie\_client AS TABLE OF VARCHAR2(20);

CREATE TABLE clienti\_yyy AS

SELECT client\_id, nume

FROM clienti\_www

WHERE ROWNUM <= 2;

ALTER TABLE clienti\_yyy

ADD (telefon VARCHAR2(30), locatie VARCHAR2(20), limita NUMBER);

CREATE OR REPLACE FUNCTION f3(v\_client clienti\_yyy.client\_id%TYPE, v\_nume clienti\_yyy.nume%TYPE)

RETURN VARCHAR2 IS

TYPE indexat\_limita IS TABLE OF NUMBER(10) INDEX BY BINARY\_INTEGER;

v\_limita indexat\_limita;

TYPE limita\_client IS TABLE OF VARCHAR2(20);

v\_telefon telefon\_client := telefon\_client();

v\_locatie locatie\_client := locatie\_client();

BEGIN

SELECT numar\_telefon BULK COLLECT INTO v\_telefon

FROM clienti\_www cl

WHERE cl.client\_id = v\_client;

SELECT limita\_zilnica BULK COLLECT INTO v\_limita

FROM conturi\_www c

WHERE c.client\_id = v\_client;

SELECT oras BULK COLLECT INTO v\_locatie

FROM locatii\_www l

WHERE v\_client = l.client\_id;

FOR i IN 1 .. GREATEST(v\_telefon.COUNT, v\_locatie.COUNT)

LOOP

UPDATE clienti\_yyy c

SET telefon = v\_telefon(i),

locatie = NVL(v\_locatie(i), c.locatie),

limita = NVL(v\_limita(i), c.limita)

WHERE c.client\_id = v\_client;

IF SQL%ROWCOUNT = 1 THEN

COMMIT;

RETURN 'date updatate cu succes';

ELSE

RETURN 'nicio linie updatata.';

END IF;

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'fara date';

WHEN OTHERS THEN

RETURN 'eroare ' || SQLERRM;

END f3;

/

DECLARE

v\_result VARCHAR2(100);

BEGIN

v\_result := f3(2, 'Ionescu');

DBMS\_OUTPUT.PUT\_LINE(v\_result);

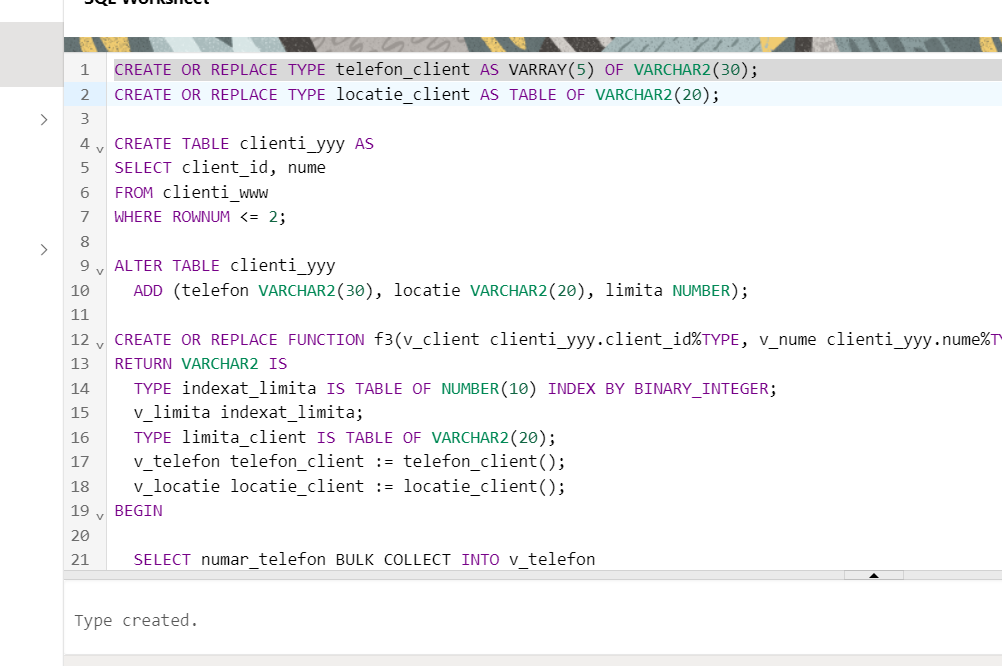
FOR rec IN (SELECT \* FROM clienti\_yyy WHERE client\_id = 2) LOOP

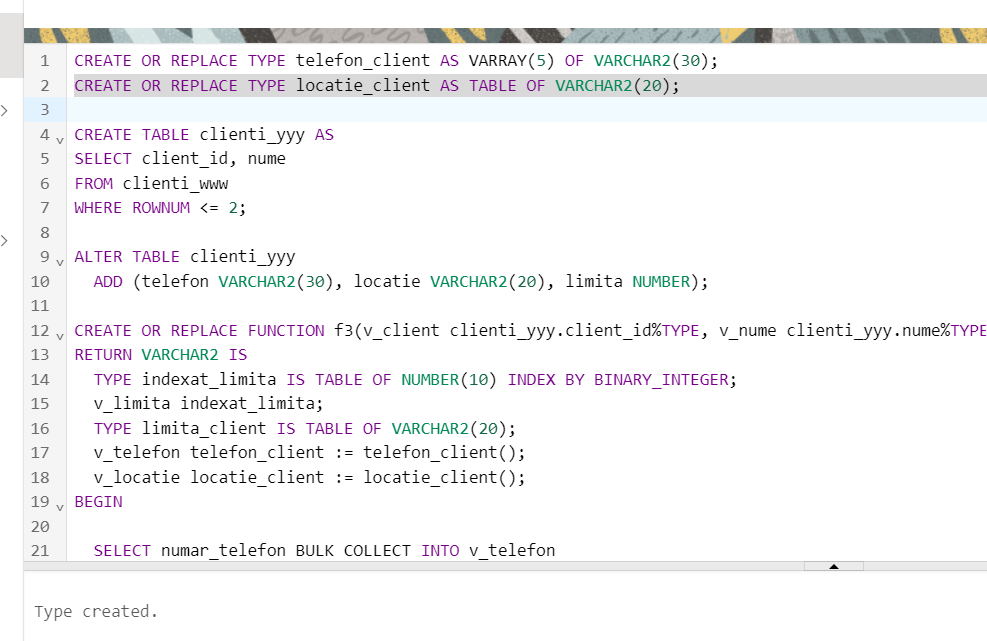
DBMS\_OUTPUT.PUT\_LINE('ID: ' || rec.client\_id || ', telefon: ' || rec.telefon || ', Locatie: ' || rec.locatie || ', Limita: ' || rec.limita);

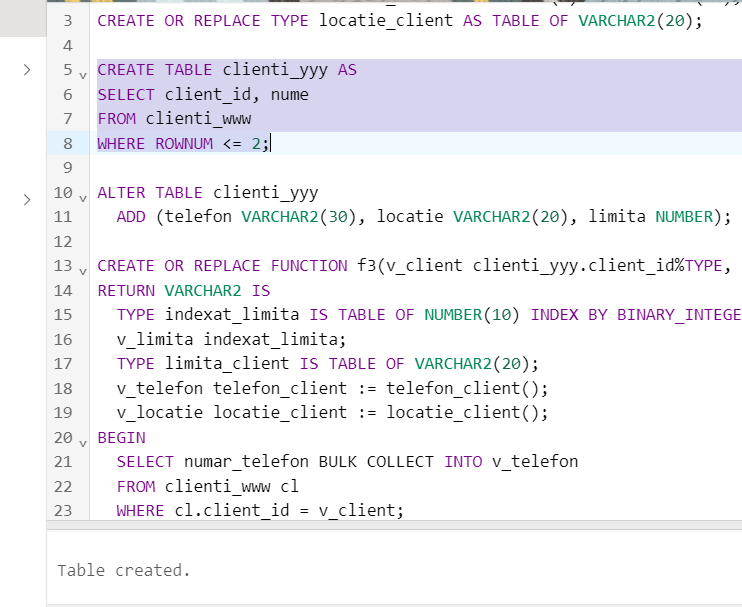
END LOOP;

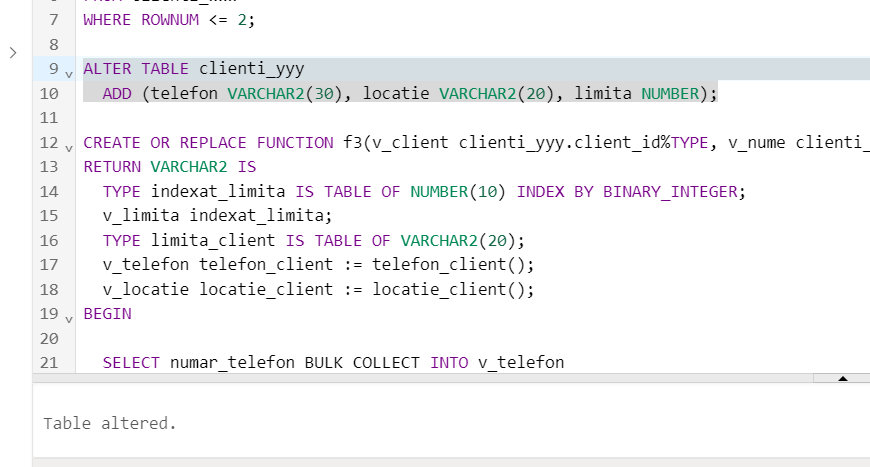
END;

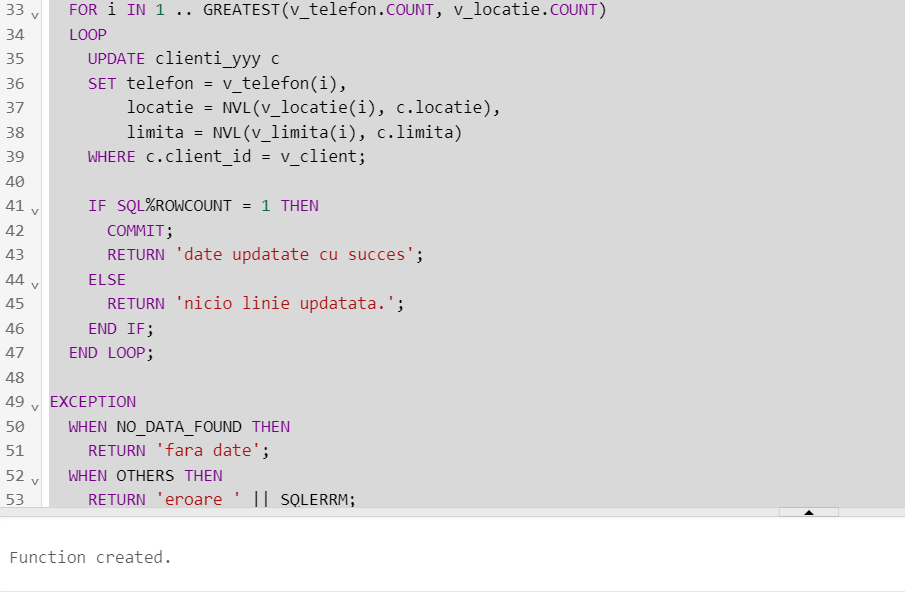
/

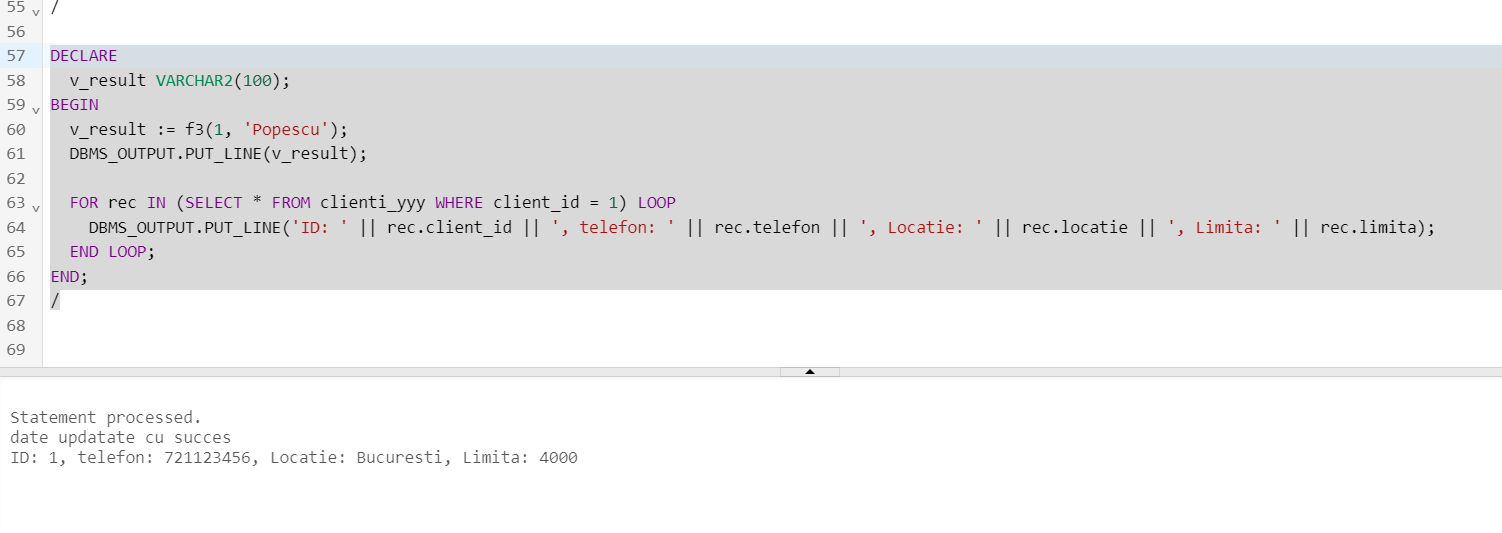














7. Pentru clientii care au participat la seminarul cu id ul 50 , care au un credit aprobat, sa se afiseze tipul creditului pe care il au si suma imprumutata.

CREATE OR REPLACE PROCEDURE suma\_credit\_nume

IS

v\_tip\_credit credit\_www.tip\_credit%TYPE;

v\_suma credit\_www.credit\_suma%TYPE;

CURSOR seminar\_cursor IS

SELECT p.client\_id

FROM participa\_www p

WHERE p.id\_seminar = 50;

CURSOR credit\_cursor(p\_client\_id participa\_www.client\_id%TYPE) IS

SELECT c.tip\_credit, c.credit\_suma

FROM credit\_www c

WHERE c.client\_id = credit\_cursor.p\_client\_id

AND c.status = 'Aprobat';

BEGIN

FOR seminar\_rec IN seminar\_cursor

LOOP

DBMS\_OUTPUT.PUT\_LINE('Clientul ' || seminar\_rec.client\_id || ' a participat la seminarul cu ID-ul 50');

BEGIN

OPEN credit\_cursor(seminar\_rec.client\_id);

LOOP

FETCH credit\_cursor INTO v\_tip\_credit, v\_suma;

EXIT WHEN credit\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(' - Tipul creditului: ' || v\_tip\_credit || ', Suma imprumutata: ' || v\_suma);

END LOOP;

CLOSE credit\_cursor;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('eroare pentru clientul: ' || seminar\_rec.client\_id || ': ' || SQLERRM);

CLOSE credit\_cursor;

END;

END LOOP;

END suma\_credit\_nume;

/

BEGIN

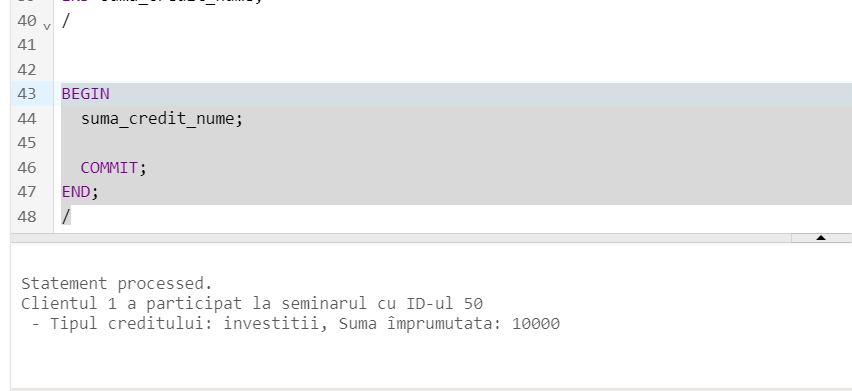
suma\_credit\_nume;

COMMIT;

END;

/





8. Definiti o functie stocata independent prin care sa obtineti numele,tipul serviciului privat si tipul contului clientilor, care beneficiaza de private banking si au limita zilnica mai mica sau egala cu 4000. Sa se trateze toate exceptiile. Apelati subprogramul pentru clientii cu id ul 1,3,4,6,10.

CREATE OR REPLACE FUNCTION f4(v\_id clienti\_www.client\_id%TYPE)

RETURN VARCHAR IS

nume clienti\_www.nume%TYPE;

serviciu private\_banking\_www.tip\_serviciu\_privat%TYPE;

cont conturi\_www.tip\_cont%TYPE;

limita conturi\_www.limita\_zilnica%TYPE;

serviciu\_exceptie EXCEPTION;

limita\_exceptie EXCEPTION;

serv\_limita\_exceptie EXCEPTION;

result VARCHAR2(4000);

BEGIN

SELECT c.nume, p.tip\_serviciu\_privat, co.tip\_cont, co.limita\_zilnica

INTO nume, serviciu, cont, limita

FROM clienti\_www c, private\_banking\_www p, conturi\_www co

WHERE c.client\_id = v\_id

AND c.client\_id = co.client\_id

AND c.client\_id = p.client\_id;

IF limita > 4000 THEN

IF serviciu IS NULL THEN

RAISE serv\_limita\_exceptie;

ELSE

RAISE limita\_exceptie;

END IF;

END IF;

IF serviciu IS NULL THEN

RAISE serviciu\_exceptie;

END IF;

result := 'Clientul ' || nume|| ' cu id ul ' || v\_id || ' are un cont de ' || cont || ' cu limita zilnica de ' || limita || ' si beneficiaza de ' || serviciu;

RETURN result;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'Nu exista clienti cu id ul ' || v\_id;

WHEN limita\_exceptie THEN

RETURN 'Limita clientului cu id ul ' || v\_id || ' depaseste 4000 lei';

WHEN serviciu\_exceptie THEN

RETURN 'Clientul cu id ul ' || v\_id|| ' nu beneficiaza de un serviciu privat';

WHEN serv\_limita\_exceptie THEN

RETURN 'Clientul cu id ul ' || v\_id|| ' nu beneficiaza de un serviciu privat si are limita mai mare de 4000 lei';

END f4;

/

--apelare

DECLARE

result VARCHAR2(4000);

BEGIN

result := f4(1);

DBMS\_OUTPUT.PUT\_LINE(result);

END;

/

DECLARE

result VARCHAR2(4000);

BEGIN

result := f4(3);

DBMS\_OUTPUT.PUT\_LINE(result);

END;

/

DECLARE

result VARCHAR2(4000);

BEGIN

result := f4(4);

DBMS\_OUTPUT.PUT\_LINE(result);

END;

/

DECLARE

result VARCHAR2(4000);

BEGIN

result := f4(6);

DBMS\_OUTPUT.PUT\_LINE(result);

END;

/

DECLARE

result VARCHAR2(4000);

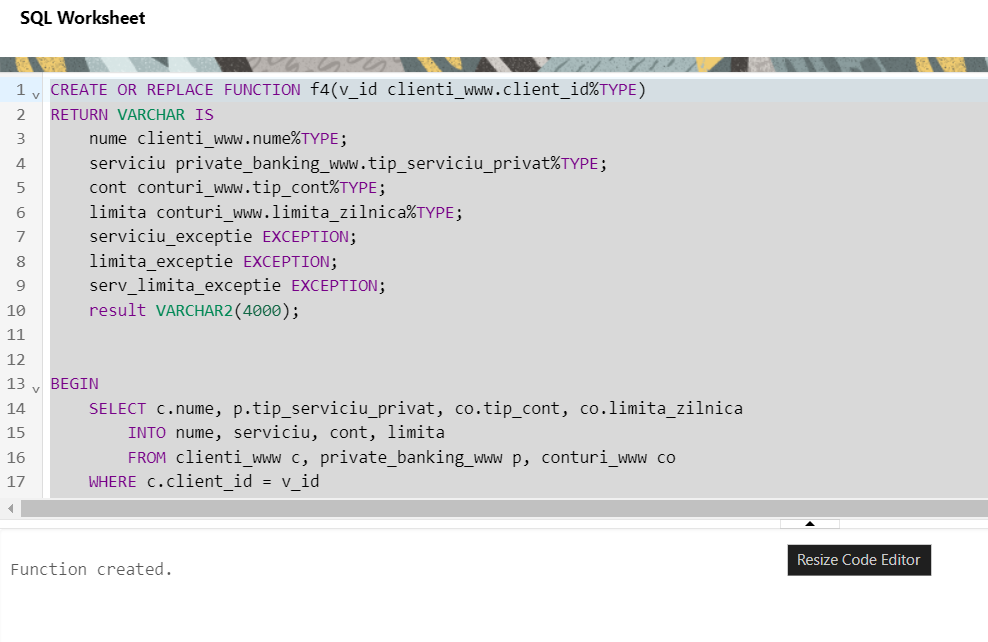
BEGIN

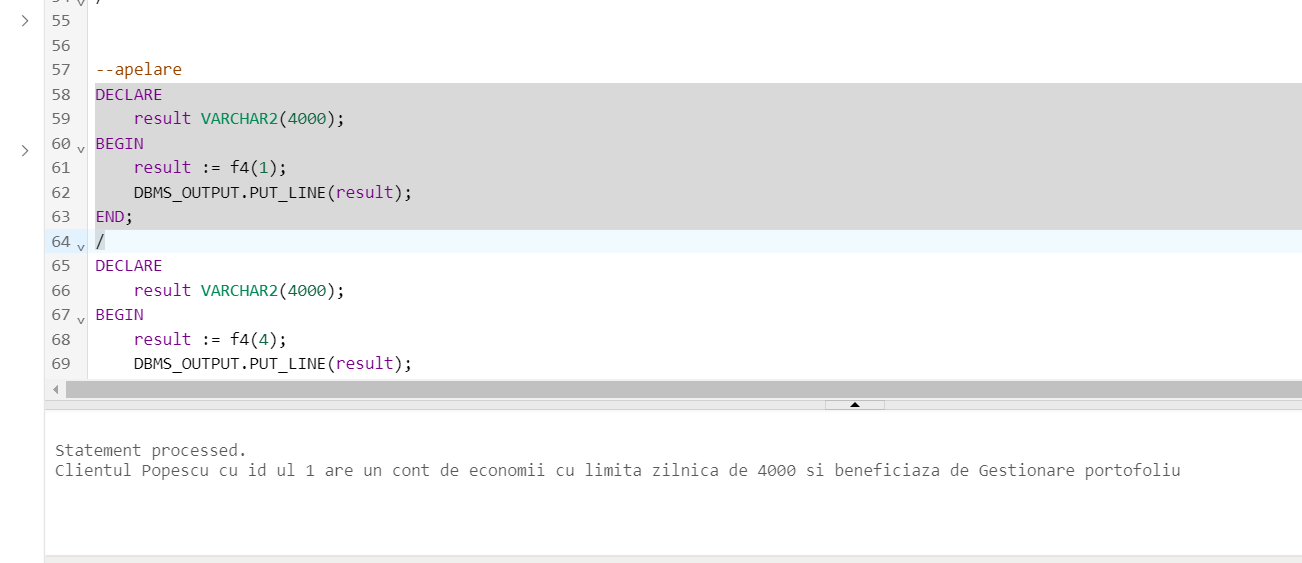
result := f4(10);

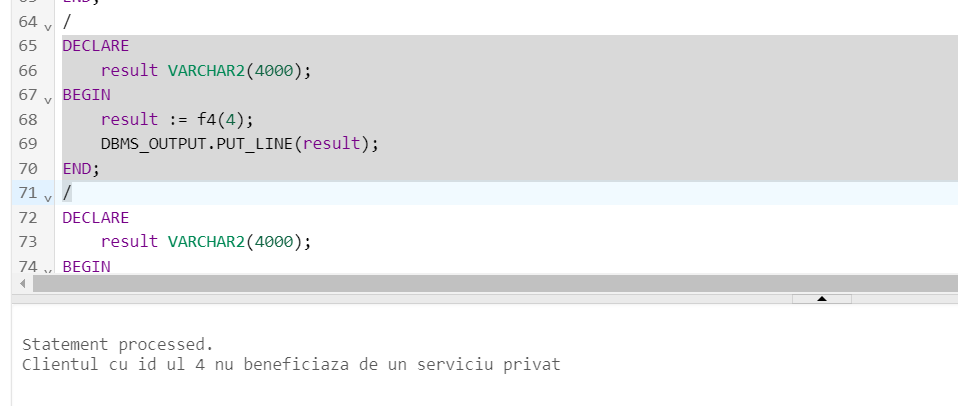
DBMS\_OUTPUT.PUT\_LINE(result);

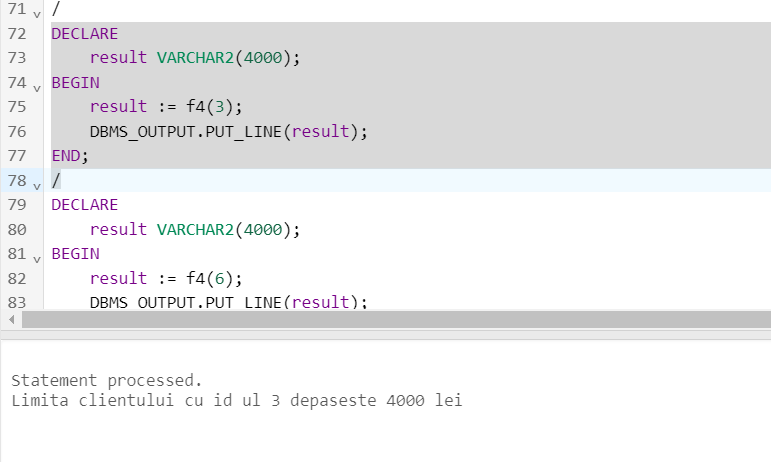
END;

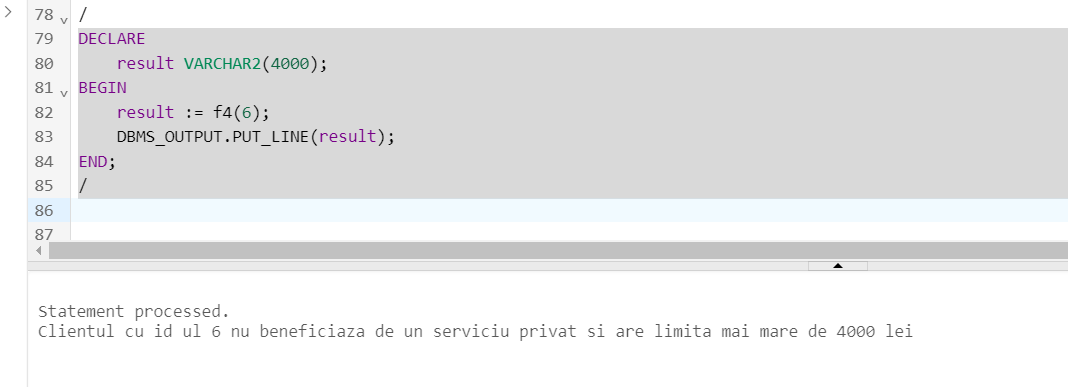
/

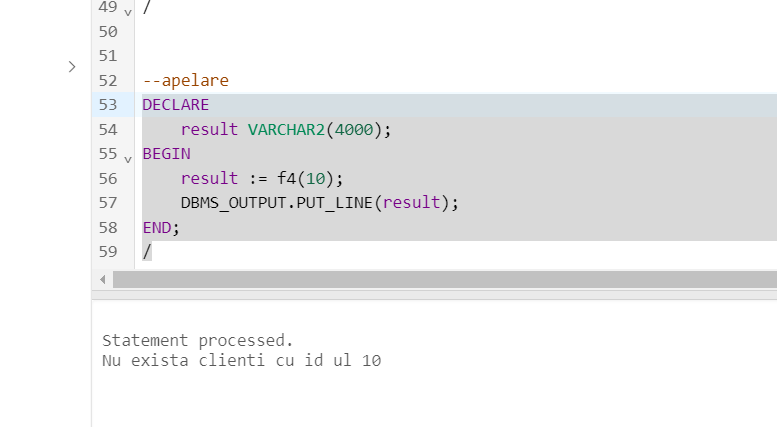












9.Definiti o procedura stocata independent care afiseaza numele clientilor care beneficiaza de un singur serviciu financiar, de servicii de tip private banking si care au un credit aprobat. Sa se testeze pentru clientii cu id ul 1, 4, 5,2,10 .

CREATE OR REPLACE PROCEDURE p1 (v\_id clienti\_www.client\_id%type)

IS

v\_nume clienti\_www.nume%type;

exceptie\_credit\_privat EXCEPTION;

v\_status\_credit credit\_www.status%type;

v\_privat private\_banking\_www.tip\_serviciu\_privat%type;

exceptie\_credit EXCEPTION;

exceptie\_privat EXCEPTION;

BEGIN

SELECT c.nume, cr.status, p.tip\_serviciu\_privat

INTO v\_nume, v\_status\_credit, v\_privat

FROM clienti\_www c, clienti\_cu\_servicii\_financiare\_www csf, servicii\_financiare\_www s, private\_banking\_www p, credit\_www cr

WHERE c.client\_id = csf.client\_id

AND csf.id\_financiar = s.id\_financiar

AND c.client\_id = cr.client\_id

AND c.client\_id = p.client\_id

AND c.client\_id = v\_id

AND ROWNUM = 1;

IF v\_status\_credit IS NULL OR v\_status\_credit = 'In asteptare' THEN

IF v\_privat IS NULL THEN

RAISE exceptie\_credit\_privat;

ELSE

RAISE exceptie\_credit;

END IF;

END IF;

IF v\_privat IS NULL THEN

RAISE exceptie\_privat;

END IF;

DBMS\_OUTPUT.PUT\_LINE('Clientul cu id ul ' ||v\_id|| ' se numeste: ' || v\_nume);

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('clientul cu id ul ' || v\_id || ' beneficiaza de mai multe servicii financiare');

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista clientul cu id ul ' || v\_id );

WHEN exceptie\_credit\_privat THEN

DBMS\_OUTPUT.PUT\_LINE('Clientul cu id ul ' || v\_id || ' nu are un credit aprobat si nu beneficiaza de un serviciu privat');

WHEN exceptie\_credit THEN

DBMS\_OUTPUT.PUT\_LINE('Clientul cu id ul ' || v\_id || ' nu are un credit aprobat');

WHEN exceptie\_privat THEN

DBMS\_OUTPUT.PUT\_LINE('Clientul cu id ul ' || v\_id || ' nu beneficiaza de un serviciu privat');

END p1;

/

BEGIN

p1(1);

END;

/

BEGIN

p1(4);

END;

/

BEGIN

p1(5);

END;

/

BEGIN

p1(2);

END;

/

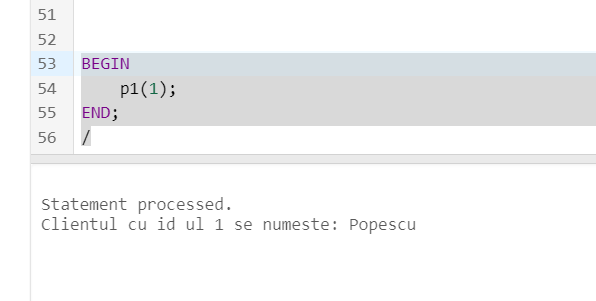
BEGIN

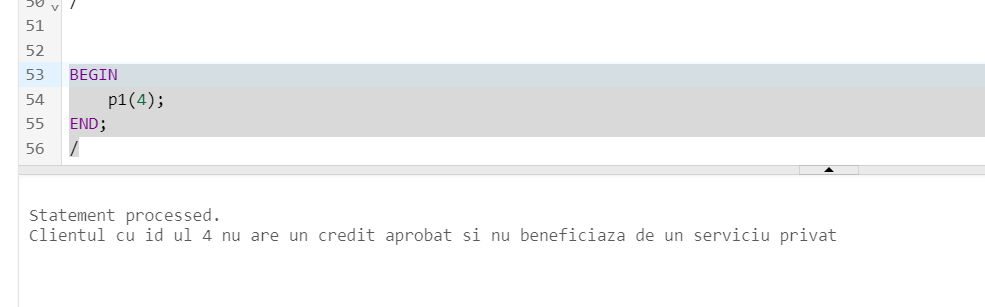
p1(10);

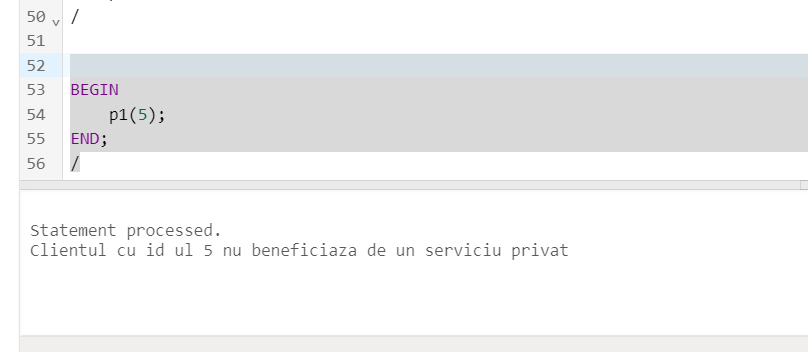
END;

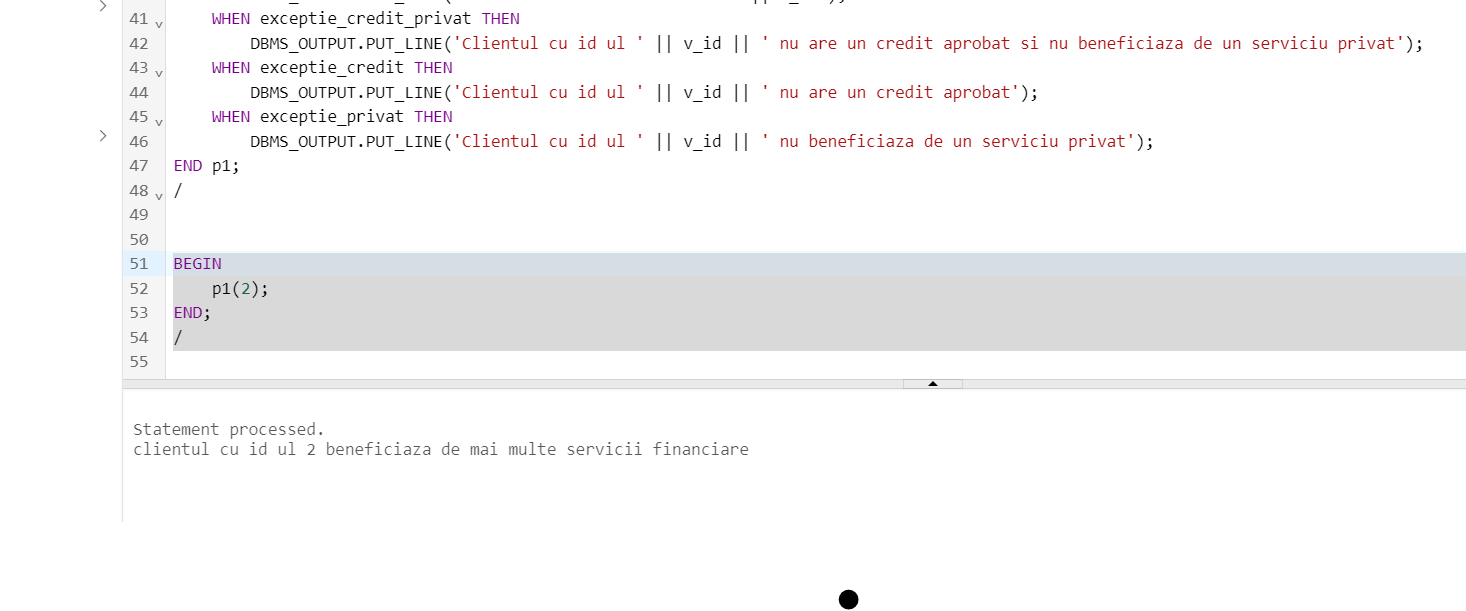
/

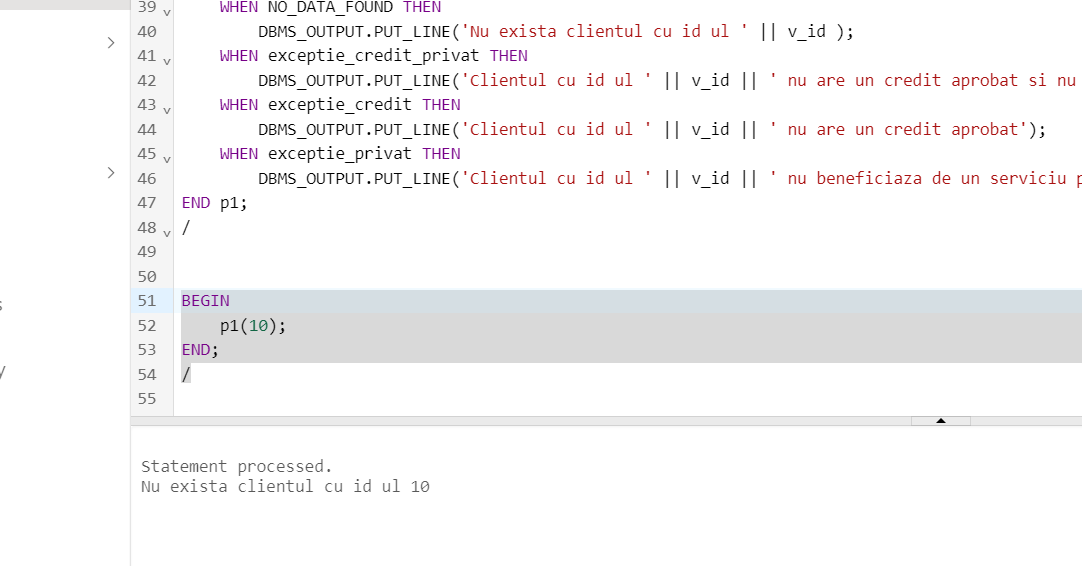












10.Creeaza un tabel "cl" care sa contina clientul cu numele Ionescu. Defineste un trigger care sa permita stergerea asupra tabelului cl doar de luni pana vineri. Declanseaza triggerul.

create table cl as

select nume

from clienti\_www

where nume like '%Ionescu%';

CREATE OR REPLACE TRIGGER trg1

BEFORE DELETE ON cl

BEGIN

IF TO\_CHAR(SYSDATE, 'DY') IN ('SAT', 'SUN') THEN

RAISE\_APPLICATION\_ERROR(-20001, 'tabelul nu poate fi sters');

END IF;

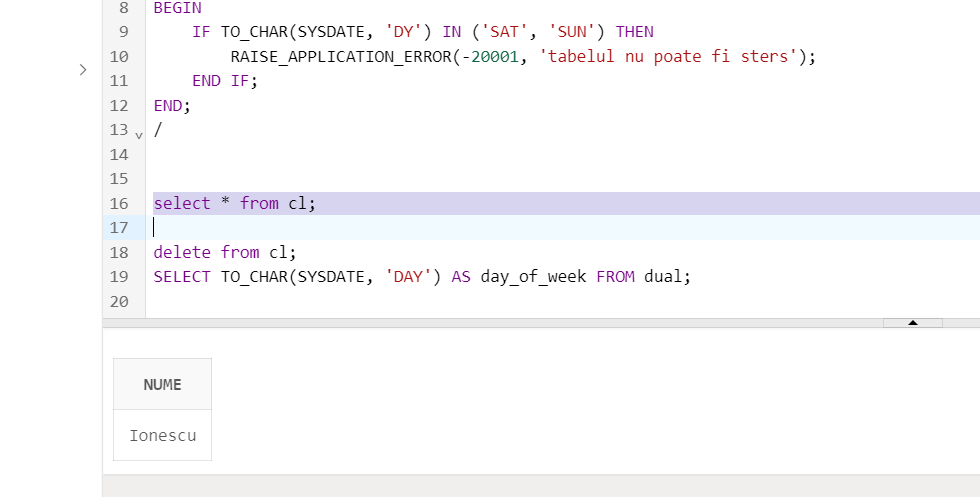
END;

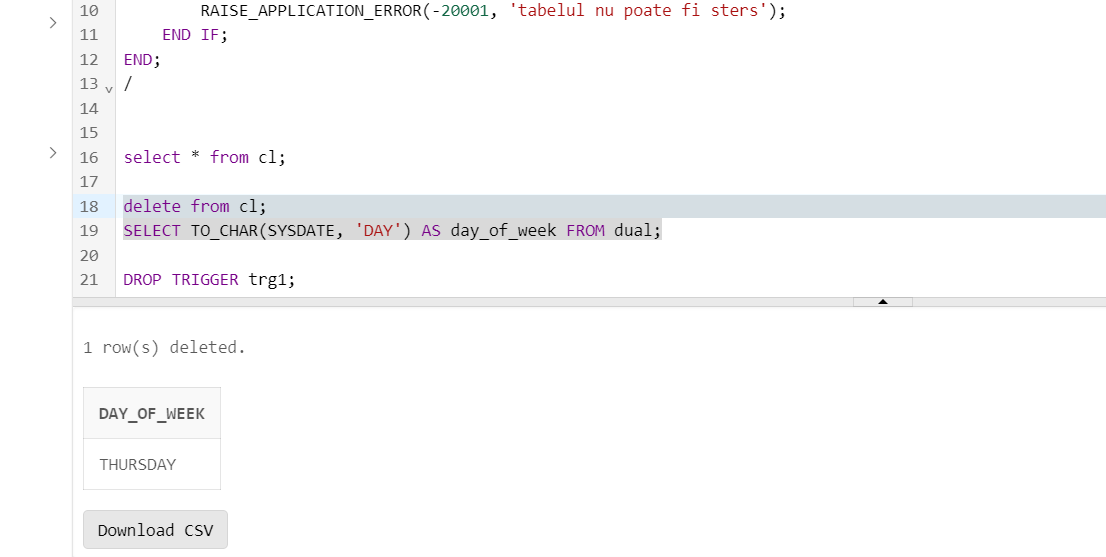
/

select \* from cl;

delete from cl;







11. Definiti un trigger care sa nu permita marirea limitei unui cont din tabelul conturi\_www.

create or replace trigger trg2

before update of limita\_zilnica on conturi\_www

for each row

begin

if(:new.limita\_zilnica > :old.limita\_zilnica) then

raise\_application\_error( - 20001, 'limita zilnica nu poate fi marita');

end if;

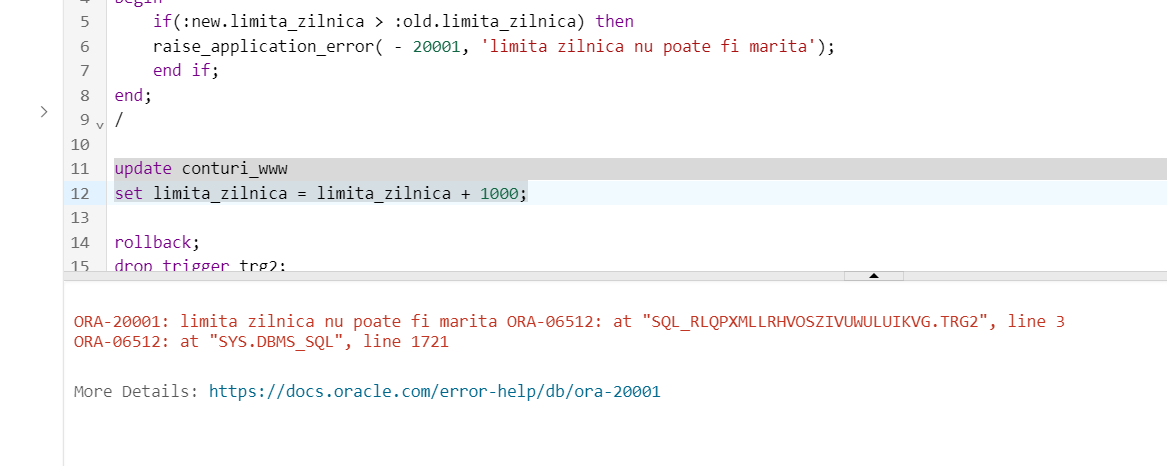
end;

/

update conturi\_www

set limita\_zilnica = limita\_zilnica + 1000;





12. Creeaza tabelul "evidenta" cu urmatoarele campuri: eroare si data(data producerii evenimentului). Defineste un declansator care sa introduca date in tabelul "evidenta" dupa ce o eroare a aparut.

CREATE TABLE evidenta

( eroare varchar2(2000),

data Date);

CREATE OR REPLACE TRIGGER trg3

AFTER SERVERERROR ON SCHEMA

BEGIN

INSERT INTO evidenta

VALUES (DBMS\_UTILITY.FORMAT\_ERROR\_STACK, SYSDATE);

END;

/

select 10/0 from dual;

drop table evidenta;

drop trigger trg3;

SELECT \* FROM evidenta;

