

**BAZA DE DATE A**

**UNEI BIBLIOTECI**

Igescu Rares-Andrei, grupa 232, seria 23



Contents

[1.](#_heading=h.gjdgxs) Descrierea modelului real si a utilitatii acestuia 3

[2.](#_heading=h.30j0zll) Realizați diagrama entitate-relație (ERD) 5

[3. Realizați diagrama conceptuală a modelului propus 6](#_heading=h.1fob9te)

[4. Implementați in Oracle diagrama conceptuală realizată 7](#_heading=h.3znysh7)

[5. Adăugați informații corente în tabelele create 10](#_heading=h.2et92p0)

[6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze cele 3 tipuri de colecții studiate. Apelați subprogramul. 22](#_heading=h.tyjcwt)

[7. Formulați în limbaj natural o problem pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul fiind cursor parametrizat dependent de cursorul anterior. 24](#_heading=h.3dy6vkm)

[8. Formulați în limbaj natural o problem pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele create. Tratați toate excepțiile care pot apărea. 26](#_heading=h.1t3h5sf)

[9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să aibă minim 2 parametri și să utilizeze într-o singură comandă SQL 5 tabele create. Definiți minim 2 excepții proprii. 28](#_heading=h.4d34og8)

[10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul. 31](#_heading=h.2s8eyo1)

[11. Definți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul. 33](#_heading=h.17dp8vu)

[12. Definți un trigger de tip LDD. Declanșați trigger-ul. 36](#_heading=h.3rdcrjn)

[13. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite. 37](#_heading=h.26in1rg)

Igescu Rares-Andrei

Seria 23, Grupa 232

Proiect la materia Sisteme de Gestiune ale Bazelor de Date

**Titlu:** Gestionarea bazei de date a unei biblioteci

# Descrierea modelului real si a utilitatii acestuia

Modelul bazei de date este realizat pentru a aranja eficient datele necesare gestionării resurselor și a activităților zilnice, printre care se numără cititorii, cărțile, evenimentele și multe altele. Ea se caracterizează printr-o multitudine de tabele interconectate prin diferite tipuri de relații cunoscute, stabilite de creator.

Relațiile oferă accesarea și gestionarea rapidă a datelor din bibliotecă. Funcționalitatea bazei de date descrisă creează o structură centralizată și automatizată de gestiune a evenimentelor din bibliotecă. Ea permite angajaților să urmărească eficient resursele și să dezvolte plăcut interacțiunea cititorilor cu baza de date.

Modelul de date va gestiona informații despre o bibliotecă. Cărțile din inventarul bibliotecii vor putea fi împrumutate de un cititor din cadrul bibliotecii. Fiecare împrumut are înregistrat în baza de date ziua în care s-a realizat împrumutul și data la care a fost cartea returnată. Nu există o perioadă maximă pentru un împrumut. La rândul său, fiecare cititor este subordonat unui angajat. Un împrumut se poate realiza doar dacă în bibliotecă există cel puțin un exemplare disponibil din cartea dorită. Disponibilitatea se poate verifica printr-o tabelă specifică care contorizează cărțile disponibile.

În cadrul bibliotecii se pot desfășura mai multe evenimente (club de lectură, prezentare de carte) într-o sală anunțată din timp și în funcție de disponibilitate. Evenimentele sunt organizate de cel puțin unul dintre clienții bibliotecii, care poate fi și el participant în cadrul evenimentului.

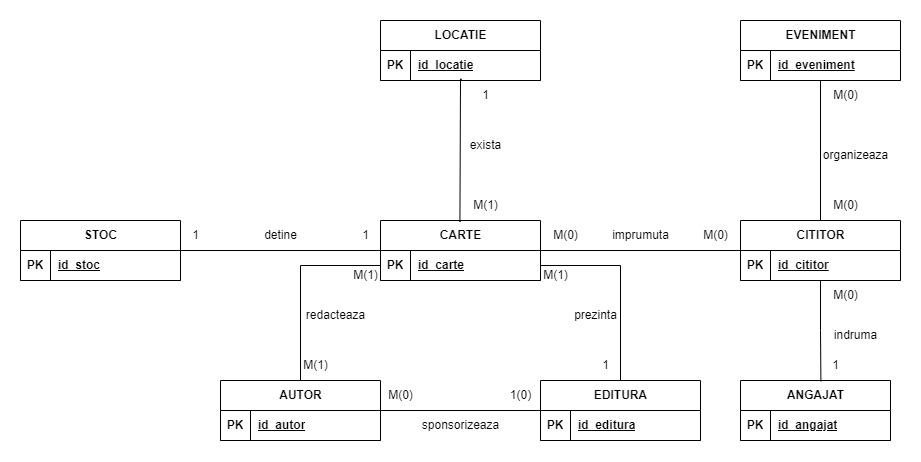
O editură poate sponsoriza, deși nu este obligatoriu, un autor. Totuși asta nu restrânge autorul sub nicio forma, el poate să publice cărți și împreună cu alte edituri.

Când vine vorba de cărți, un autor poate să scrie mai multe cărți, iar o carte poate fi scrisă de mai mulți autori. Ea aparține totuși unui singur gen literar. O carte poate fi publicată de o singură editură. Totodată, o carte și toate exemplarele ei se află în aceeași locație în bibliotecă. Stocul fiecărei cărți este reținut în baza de date și manipulat corespunzător în momentul în care se realizează un împrumut sau o carte este adusă înapoi.

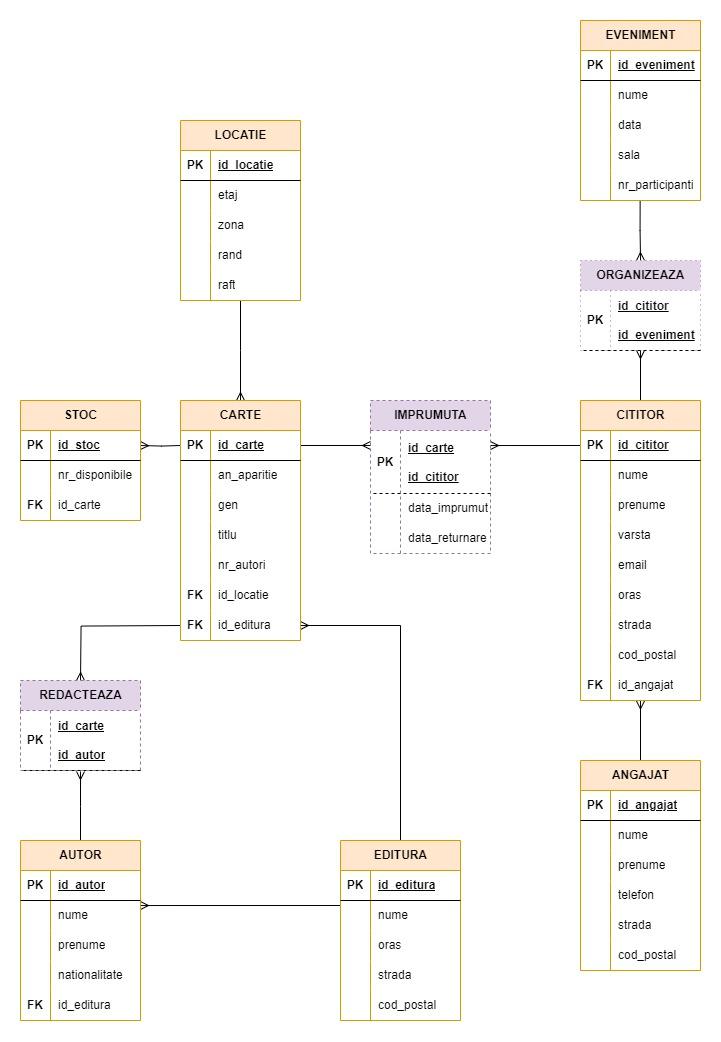
Am realizat proiectul pe Oracle SQL Developer, verisunea 23.1.1.345. Am un total de 2 GB alocați. Proiectul a fost dezvoltat pe laptop-ul personal ce are următoarea configurație:

* Procesor: Intel® Core™ I7-10870H @ 2.20 GHz
* Placă Video: NVIDIA GeForce RTX 2060
* Memorie RAM: 24 GB
* Sistem de Operare: Windows 10 Pro, fără mașină virtuală

# Realizați diagrama entitate-relație (ERD)



# Realizați diagrama conceptuală a modelului propus



# Implementați in Oracle diagrama conceptuală realizată

create table LOCATIE (

id\_locatie number(3) constraint locatie\_pk primary key,

etaj number(1),

zona number(1),

rand number(2),

raft number(2));

create table EDITURA (

id\_editura number(3) constraint editura\_pk primary key,

nume varchar2(30),

oras varchar2(30),

strada varchar2(30),

cod\_postal number(6));

create table AUTOR (

id\_autor number(3) constraint autor\_pk primary key,

id\_editura number(3),

nume varchar2(30),

prenume varchar2(30),

nationalitate varchar2(30),

constraint autor\_editura\_fk foreign key (id\_editura) references EDITURA(id\_editura));

create table CARTE (

id\_carte number(3) constraint carte\_pk primary key,

id\_locatie number(3),

id\_editura number(3),

an\_aparitie number(4),

gen varchar2(20),

titlu varchar2(30),

nr\_autori number(2),

constraint carte\_locatie\_fk foreign key (id\_locatie) references LOCATIE(id\_locatie),

constraint carte\_editura\_fk foreign key (id\_editura) references EDITURA(id\_editura));

create table STOC (

id\_stoc number(3) constraint stoc\_pk primary key,

id\_carte number(3),

nr\_disponibile number(3),

constraint stoc\_carte\_fk foreign key (id\_carte) references CARTE(id\_carte));

create table REDACTEAZA (

id\_carte number(3),

id\_autor number(3),

constraint redacteaza\_pk primary key (id\_carte, id\_autor),

constraint redacteaza\_carte\_fk foreign key (id\_carte) references CARTE(id\_carte),

constraint redacteaza\_autor\_fk foreign key (id\_autor) references AUTOR(id\_autor));

create table ANGAJAT (

id\_angajat number(3) constraint angajat\_pk primary key,

nume varchar2(30),

prenume varchar2(30),

telefon number(9),

strada varchar2(30),

cod\_postal number(6));

create table CITITOR (

id\_cititor number(3) constraint cititor\_pk primary key,

id\_angajat number(3),

nume varchar2(30),

prenume varchar2(30),

varsta varchar2(30),

email varchar2(50),

oras varchar2(30),

strada varchar2(30),

cod\_postal number(6),

constraint cititor\_angajat\_fk foreign key (id\_angajat) references ANGAJAT(id\_angajat));

create table EVENIMENT (

id\_eveniment number(3) constraint eveniment\_pk primary key,

nume varchar2(30),

data date,

sala number(3),

nr\_participanti number(3));

create table ORGANIZEAZA (

id\_cititor number(3),

id\_eveniment number(3),

constraint organizeaza\_pk primary key (id\_cititor, id\_eveniment),

constraint organizeaza\_cititor\_fk foreign key (id\_cititor) references CITITOR(id\_cititor),

constraint organizeaza\_eveniment\_fk foreign key (id\_eveniment) references EVENIMENT(id\_eveniment));

create table IMPRUMUT (

id\_cititor number(3),

id\_carte number(3),

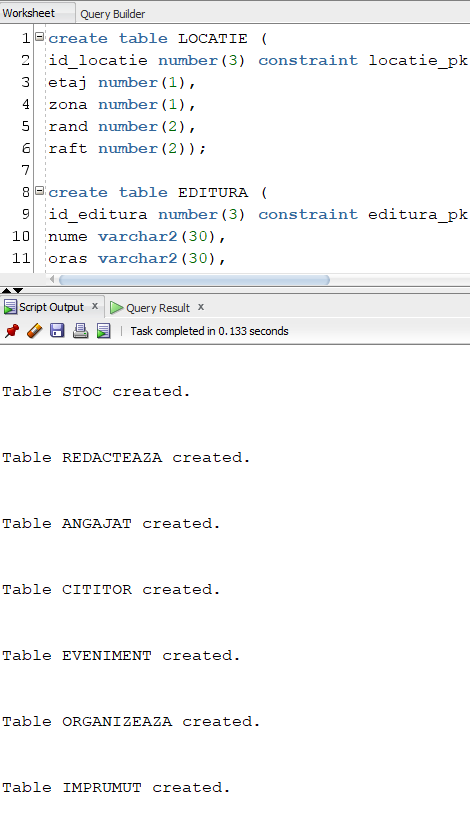
data\_imprumut date,

data\_returnare date,

constraint imprumut\_pk primary key (id\_cititor, id\_carte),

constraint imprumut\_cititor\_fk foreign key (id\_cititor) references CITITOR(id\_cititor),

constraint imprumut\_carte\_fk foreign key (id\_carte) references CARTE(id\_carte));



Acum voi adăuga constrângerile pentru fiecare tabelă.

alter table autor modify nume not null;

alter table autor modify prenume not null;

alter table editura modify nume not null;

alter table editura modify oras not null;

alter table editura modify cod\_postal not null;

alter table carte modify nr\_exemplare not null;

alter table carte modify gen not null;

alter table carte modify titlu not null;

alter table locatie modify etaj not null;

alter table locatie modify zona not null;

alter table locatie modify rand not null;

alter table locatie modify raft not null;

alter table angajat modify nume not null;

alter table angajat modify prenume not null;

alter table angajat modify telefon not null;

alter table cititor modify nume not null;

alter table cititor modify prenume not null;

alter table cititor modify varsta not null;

alter table cititor modify cod\_postal not null;

alter table eveniment modify nume not null;

alter table eveniment modify sala not null;

alter table eveniment modify nr\_participanti not null;

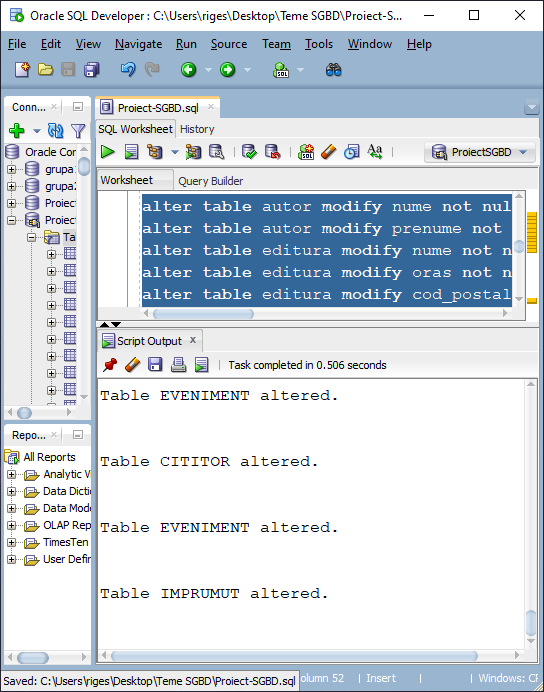
alter table cititor

add constraint check\_varsta check (varsta > 14);

alter table eveniment

add constraint check\_participanti check (nr\_participanti > 10);

alter table imprumut modify data\_imprumut not null;



# Adăugați informații corente în tabelele create

Inserarea de date coerente în tabela LOCATIE

insert all

into locatie values (1, 1, 1, 1, 1)

into locatie values (2, 1, 1, 1, 2)

into locatie values (3, 1, 3, 1, 3)

into locatie values (4, 1, 4, 1, 4)

into locatie values (5, 2, 1, 2, 1)

into locatie values (6, 2, 2, 3, 4)

into locatie values (7, 2, 3, 2, 3)

into locatie values (8, 3, 2, 3, 2)

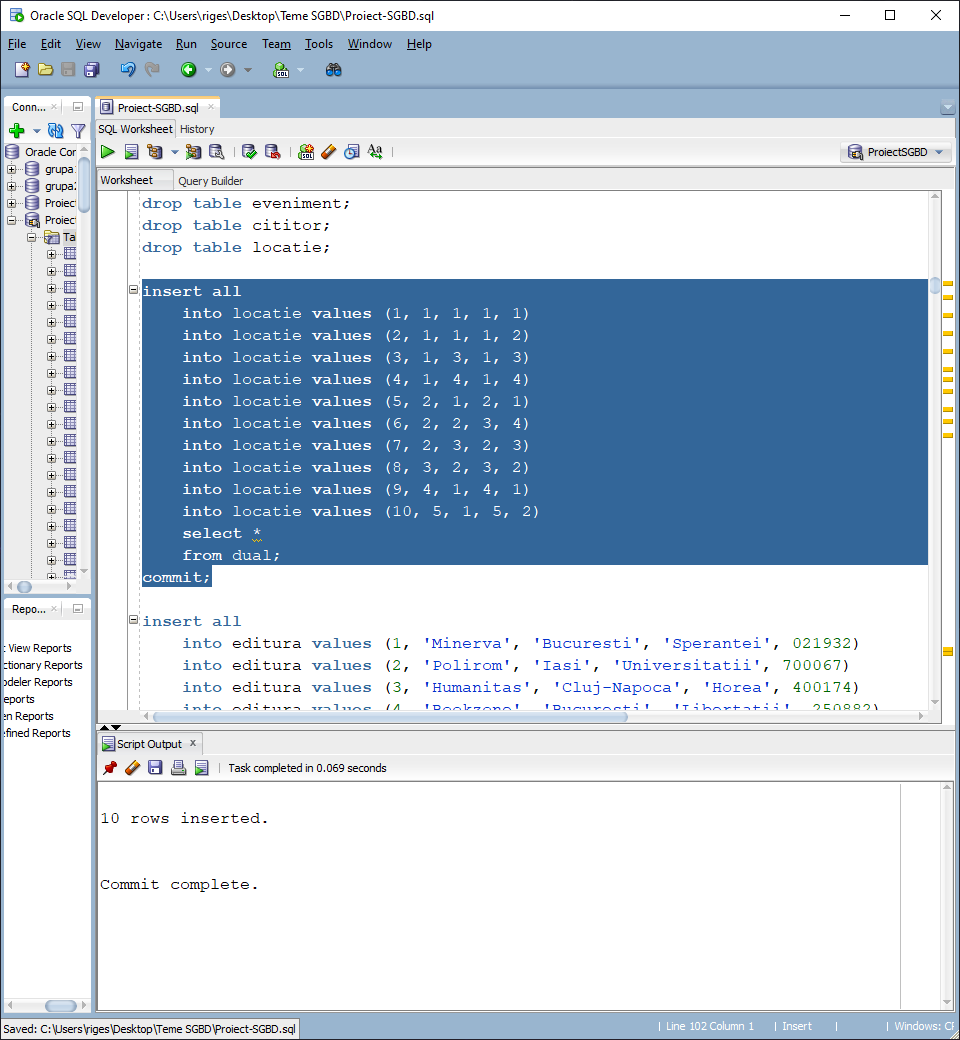
into locatie values (9, 4, 1, 4, 1)

into locatie values (10, 5, 1, 5, 2)

select \*

from dual;

commit;



Inserarea de date corente în tabela EDITURA

insert all

into editura values (1, 'Minerva', 'Bucuresti', 'Sperantei', 021932)

into editura values (2, 'Polirom', 'Iasi', 'Universitatii', 700067)

into editura values (3, 'Humanitas', 'Cluj-Napoca', 'Horea', 400174)

into editura values (4, 'Bookzone', 'Bucuresti', 'Libertatii', 250882)

into editura values (5, 'Paralela 45', 'Bucuresti', 'Magheru', 310224)

into editura values (6, 'Polirom', 'Bucuresti', 'Stirbei Voda', 30205)

into editura values (7, 'Litera', 'Bucuresti', 'Stefan cel Mare', 223004)

into editura values (8, 'Humanitas', 'Timisoara', 'Victoriei', 300101)

into editura values (9, 'Corint', 'Bucuresti', 'Barbu Vacarescu', 203276)

into editura values (10, 'RedRom', 'Vaslui', null, 472490)

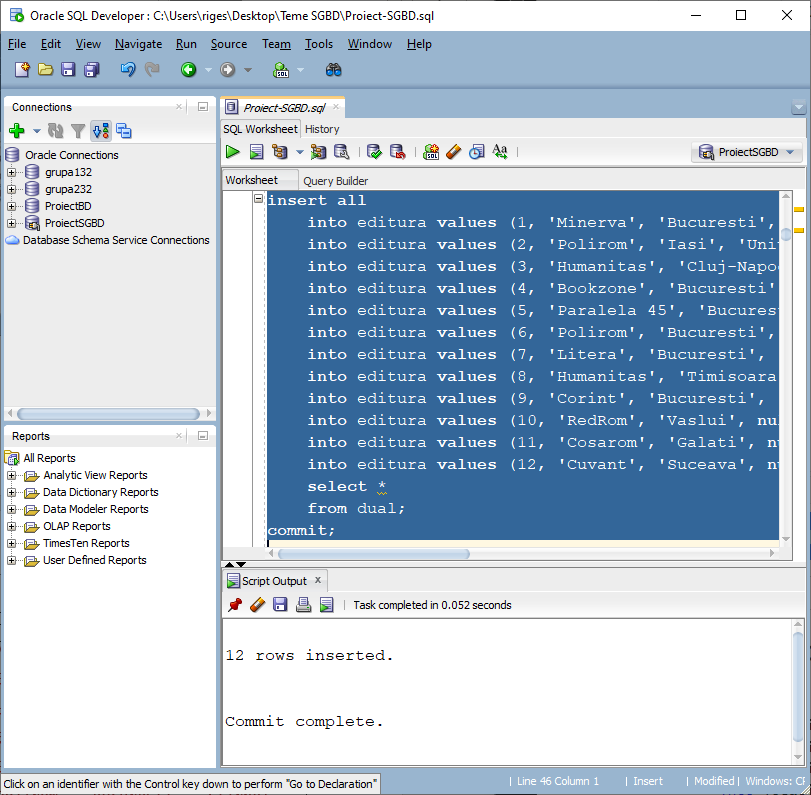
into editura values (11, 'Cosarom', 'Galati', null, 329812)

into editura values (12, 'Cuvant', 'Suceava', null, 432841)

select \*

from dual;

commit;



Inserarea de date în tabela AUTOR

insert all

into autor values(1, null, 'Marin', 'Preda', 'Roman')

into autor values(2, 2, 'Ion', 'Creanga', 'Roman')

into autor values(3, 4, 'Mihai', 'Eminescu', 'Roman')

into autor values(4, null, 'Mircea', 'Eliade', 'Roman')

into autor values(5, null, 'George', 'Cosbuc', 'Roman')

into autor values(6, 7, 'Ioan', 'Slavici', 'Roman')

into autor values(7, 11, 'Liviu', 'Rebreanu', 'Roman')

into autor values(8, 10, 'Ion', 'Luca Caragiale', 'Roman')

into autor values(9, null, 'Matei', 'Visniec', 'Roman')

into autor values(10, 3, 'Octavian', 'Paler', 'Roman')

into autor values(11, 4, 'Eugene', 'Ionesco', 'Francez')

into autor values(12, 9, 'Leo', 'Tolstoi', 'Rus')

into autor values(13, null, 'William', 'Shakespeare', 'Englez')

into autor values(14, null, 'Gabriel', 'Garcia Marquez', 'Columbian')

into autor values(15, 1, 'Haruki', 'Murakami', 'Japonez')

into autor values(16, 2, 'J.K.', 'Rowling', 'Englez')

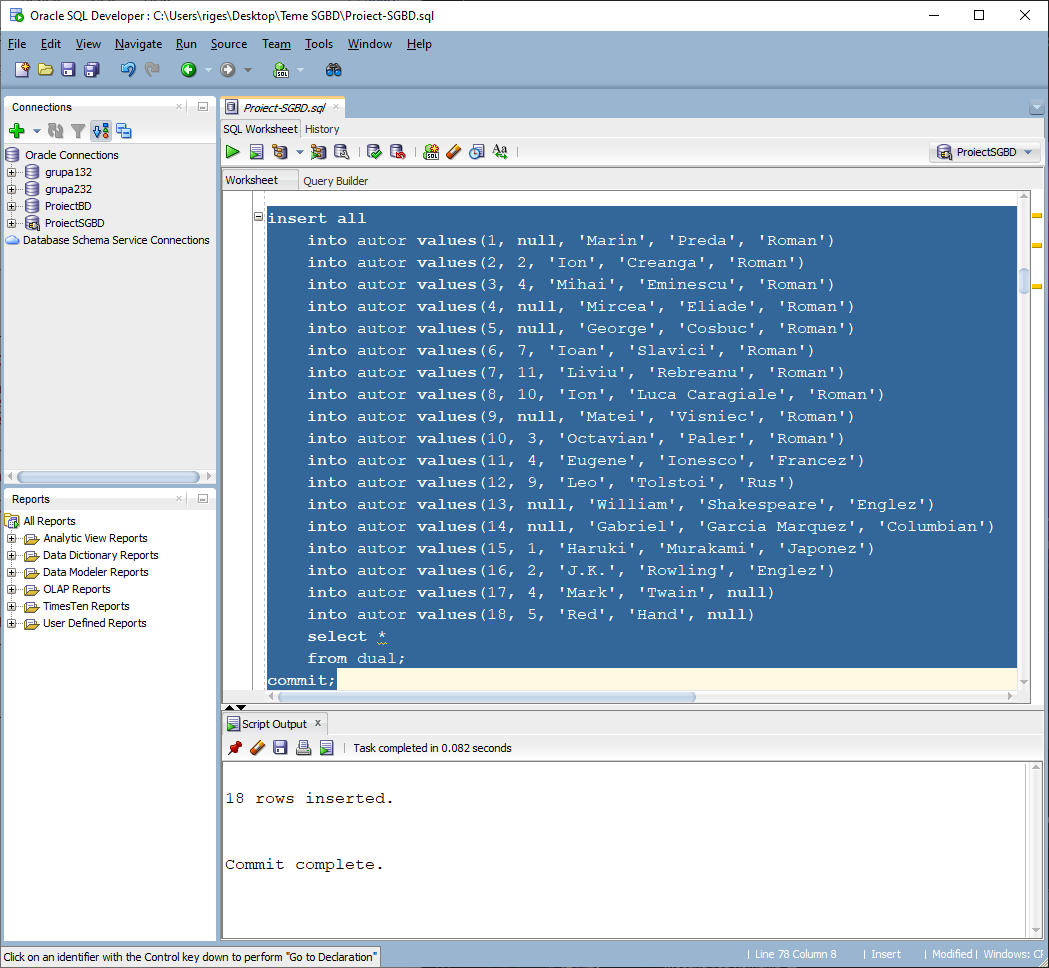
into autor values(17, 4, 'Mark', 'Twain', null)

into autor values(18, 5, 'Red', 'Hand', null)

select \*

from dual;

commit;



Inserarea de date în tabela CARTE

insert all

into carte values (1, 1, 3, 2010, 'Horror', 'Uciderea', 1)

into carte values (2, 3, 7, 1989, 'Istorie', 'Revolutia', 2)

into carte values (3, 10, 12, 2023, 'Fantasy', 'Inapoi in viitor', 3)

into carte values (4, 4, 11, 2008, 'Istorie', 'Batalia', 1)

into carte values (5, 5, 10, 2002, 'Economie', 'Caderea pietei', 1)

into carte values (6, 6 , 9, 2003, 'Aventura', 'Probele insulei', 1)

into carte values (7, 7, 8, 1999, 'Horror', 'IT', 2)

into carte values (8, 8, 7, 1850, 'Geografie', 'Prima harta a pamantului', 2)

into carte values (9, 2, 6, 1998, 'Istorie', 'Primi ani dupa revolutie', 1)

into carte values (10, 3, 2, 2001, 'Fantasy', 'Lupta galaxiilor', 1)

into carte values (11, 9, 4, 2005, 'Geografie', 'Romania: Comoara Ascunsa', 1)

into carte values (12, 10, 1, 2004, 'Economie', 'Blockchain', 1)

into carte values (13, 6, 5, 2003, 'Aventura', 'Urmarirea', 1)

into carte values (14, 3, 4, 1972, 'Istorie', 'Comunismul', 3)

into carte values (15, 1, 6, 2019, 'Dezvoltare personala', 'Cum sa devii mai bun', 1)

into carte values (16, 2, 7, 1945, 'Istorie', 'Ultimele clipe de teroare', 1)

into carte values (17, 4, 8, 2015, 'Fantasy', 'Prietenul secret', 1)

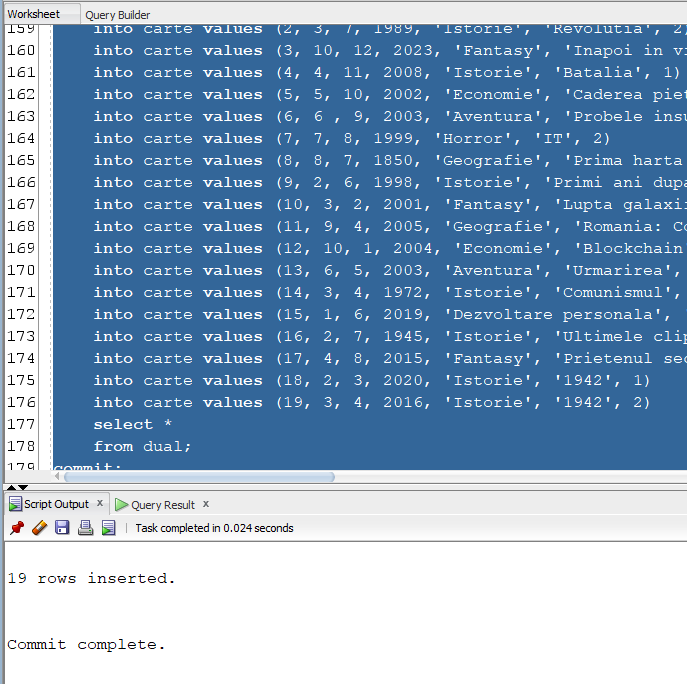
into carte values (18, 2, 3, 2020, 'Istorie', '1942', 1)

into carte values (19, 3, 4, 2016, 'Istorie', '1942', 2)

select \*

from dual;

commit;



Inserarea de date în tabela STOC

insert all

into stoc values (1, 1, 12)

into stoc values (2, 2, 20)

into stoc values (3, 3, 4)

into stoc values (4, 4, 8)

into stoc values (5, 5, 22)

into stoc values (6, 6, 10)

into stoc values (7, 7, 3)

into stoc values (8, 8, 6)

into stoc values (9, 9, 17)

into stoc values (10, 10, 25)

into stoc values (11, 11, 16)

into stoc values (12, 12, 9)

into stoc values (13, 13, 33)

into stoc values (14, 14, 10)

into stoc values (15, 15, 5)

into stoc values (16, 16, 2)

into stoc values (17, 17, 12)

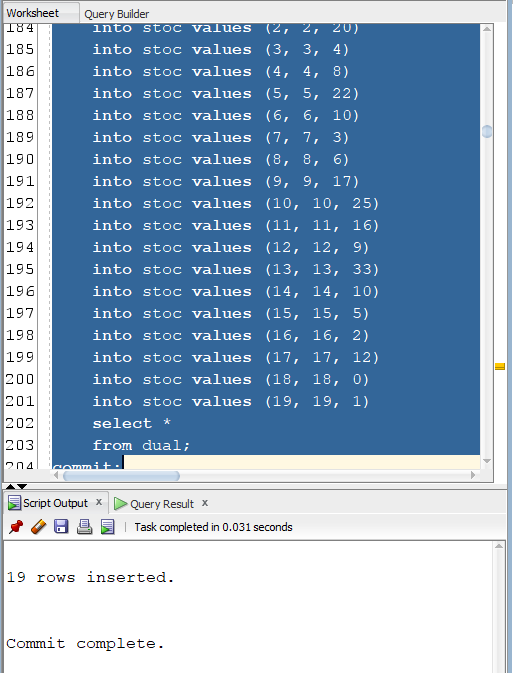
into stoc values (18, 18, 0)

into stoc values (19, 19, 1)

select \*

from dual;

commit;



Inserarea de date în tabela REDACTEAZA

insert all

into redacteaza values (1,1)

into redacteaza values (2,1)

into redacteaza values (2,7)

into redacteaza values (3,1)

into redacteaza values (3,10)

into redacteaza values (3,15)

into redacteaza values (4,5)

into redacteaza values (5,9)

into redacteaza values (6,2)

into redacteaza values (7,11)

into redacteaza values (7,12)

into redacteaza values (8,2)

into redacteaza values (8,17)

into redacteaza values (9,12)

into redacteaza values (10,15)

into redacteaza values (11,16)

into redacteaza values (12,11)

into redacteaza values (13,13)

into redacteaza values (14,9)

into redacteaza values (14,10)

into redacteaza values (14,3)

into redacteaza values (15,9)

into redacteaza values (16,3)

into redacteaza values (17,12)

into redacteaza values (18, 1)

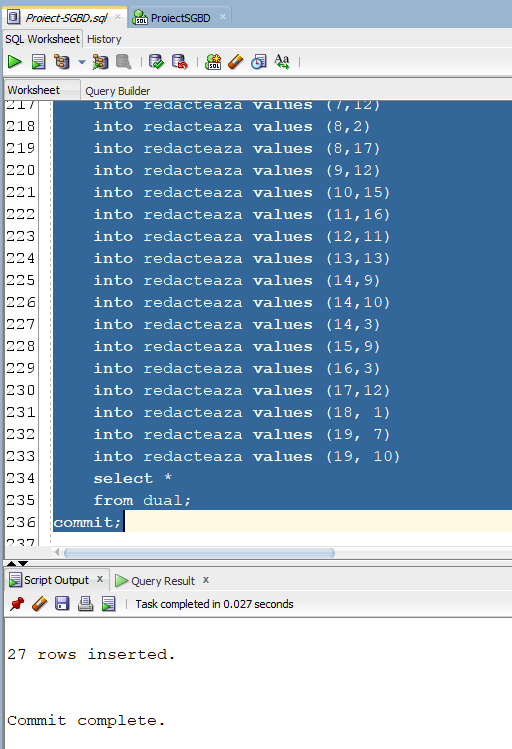
into redacteaza values (19, 7)

into redacteaza values (19, 10)

select \*

from dual;

commit;



Inserarea de date în tabela ANGAJAT

insert all

into angajat values (1, 'Nicolae', 'Stanciu', 0770382921, 'nicostn@gmail.com', 730215)

into angajat values (2, 'Vinicius', 'Junior', 0721567890, 'vincentiu@gmail.com', 032832)

into angajat values (3, 'Jude', 'Belligham', 0732123456, 'judeblg@yahoo.com', 110438)

into angajat values (4, 'Claudiu', 'Petrila', 0710234567, 'petrilacl@gmail.com', 900016)

into angajat values (5, 'Gheorghe', 'Hagi', 0765443210, 'hagimnd@personal.com', 463291)

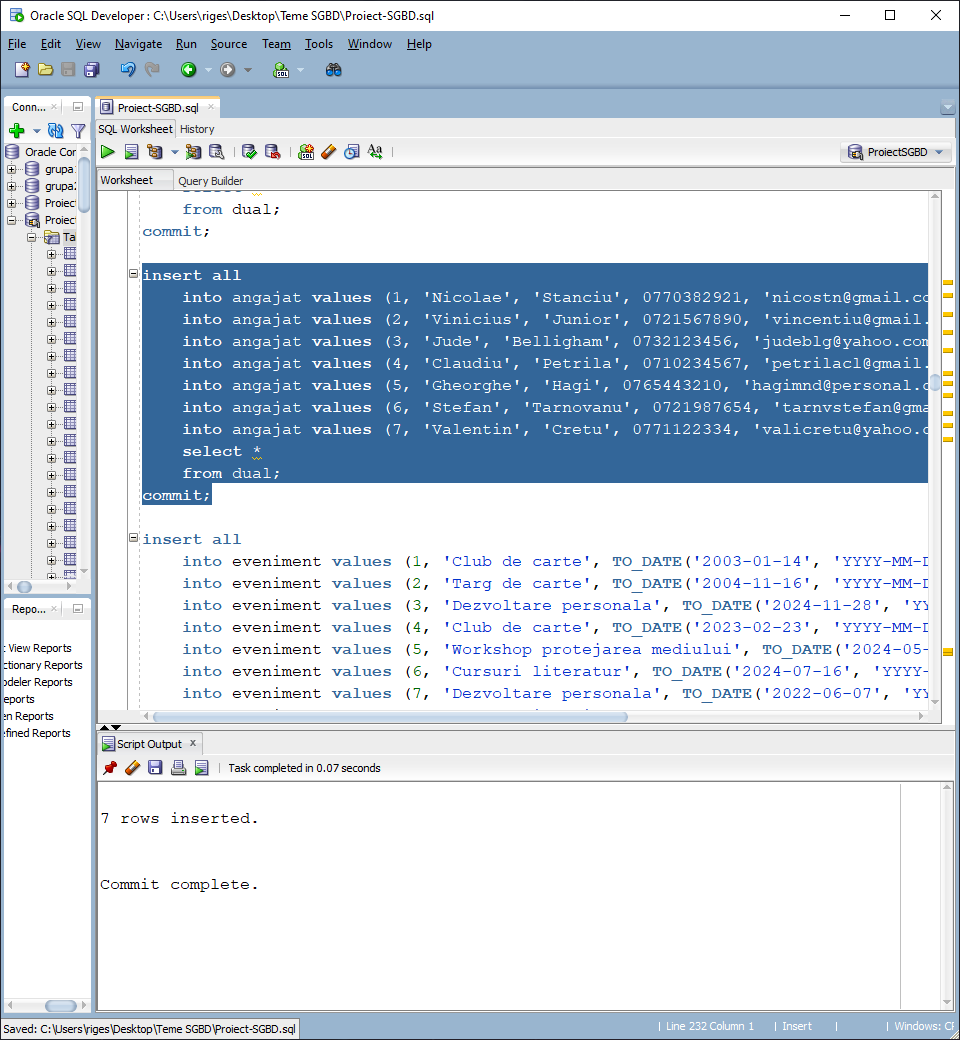
into angajat values (6, 'Stefan', 'Tarnovanu', 0721987654, 'tarnvstefan@gmail.com', 038232)

into angajat values (7, 'Valentin', 'Cretu', 0771122334, 'valicretu@yahoo.com', 573212)

select \*

from dual;

commit;



Inserarea de date în tabela EVENIMENT

insert all

into eveniment values (1, 'Club de carte', TO\_DATE('2003-01-14', 'YYYY-MM-DD'), 2, 32)

into eveniment values (2, 'Targ de carte', TO\_DATE('2004-11-16', 'YYYY-MM-DD'), 10, 71)

into eveniment values (3, 'Dezvoltare personala', TO\_DATE('2024-11-28', 'YYYY-MM-DD'), 8, 62)

into eveniment values (4, 'Club de carte', TO\_DATE('2023-02-23', 'YYYY-MM-DD'), 1, 103)

into eveniment values (5, 'Workshop protejarea mediului', TO\_DATE('2024-05-23', 'YYYY-MM-DD'), 3, 12)

into eveniment values (6, 'Cursuri literatur', TO\_DATE('2024-07-16', 'YYYY-MM-DD'), 7, 21)

into eveniment values (7, 'Dezvoltare personala', TO\_DATE('2022-06-07', 'YYYY-MM-DD'), 10, 43)

into eveniment values (8, 'Podcast istorie', TO\_DATE('2019-09-30', 'YYYY-MM-DD'), 4, 50)

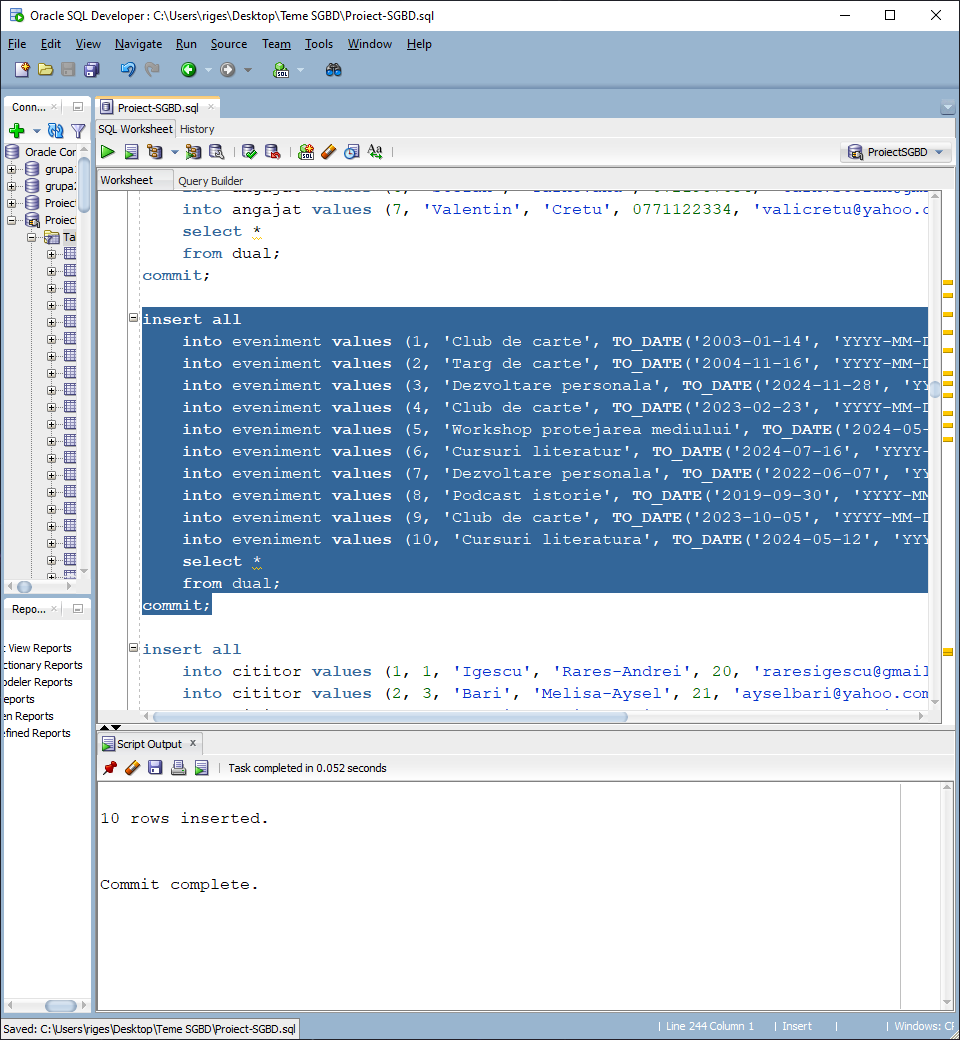
into eveniment values (9, 'Club de carte', TO\_DATE('2023-10-05', 'YYYY-MM-DD'), 10, 27)

into eveniment values (10, 'Cursuri literatura', TO\_DATE('2024-05-12', 'YYYY-MM-DD'), 2, 30)

select \*

from dual;

commit;



Inserarea de date în tabela CITITOR

insert all

into cititor values (1, 1, 'Igescu', 'Rares-Andrei', 20, 'raresigescu@gmail.com', 'Vaslui', 'Pacii', 050882)

into cititor values (2, 3, 'Bari', 'Melisa-Aysel', 21, 'ayselbari@yahoo.com', 'Bucuresti', 'Aviatorilor', 030551)

into cititor values (3, 7, 'Baltic', 'Bianca-Nicoleta', 21, 'mareabaltica@gmail.com', 'Cluj-Napoca', 'Primaverii', 400174)

into cititor values (4, 2, 'Igescu', 'Mircea-Ovidiu', 41, 'elena.dumitrescu@fmi.com', 'Timisoara', 'Traian', 300192)

into cititor values (5, 4, 'Lucescu', 'Mircea', 28, 'mrnrdc@google.com', 'Constanta', 'Mihai Viteazu', 900657)

into cititor values (6, 6, 'Lixandru', 'Maxim', 37, 'andreigeorge@yahoo.com', 'Bucuresti', 'Stefan cel Mare', 020432)

into cititor values (7, 3, 'Dragusin', 'Radu', 33, 'mariaaaaa@evrica.ro', 'Bucuresti', 'Muresenilor', 060723)

into cititor values (8, 6, 'Ratiu', 'Andrei', 29, 'stanciul@gmail.com', 'Oradea', 'Republicii', 410012)

into cititor values (9, 1, 'Nita', 'Florin', 26, 'crisvoicu@personal.com', 'Galati', 'Dunarii', 800052)

into cititor values (10, 3, 'Moldovan', 'Horatiu', 35, 'andreeaflr@google.com', 'Sibiu', 'Victoriei', 550167)

into cititor values (11, 5, 'Manea', 'Cristian', 39, 'mircea-gheorghe@gmail.com', 'Craiova', 'Independentei', 200321)

into cititor values (12, 5, 'Popescu', 'Octavian', 31, 'adinu@gmail.com.com', 'Ploiesti', 'Garii', 100200)

into cititor values (13, 2, 'Mann', 'Denis', 27, 'anamara@gmail.com', 'Targu Mures', 'Cuza Voda', 540039)

into cititor values (14, 7, 'Mihaila', 'Valentin', 30, 'iacobctl@yahoo.com', 'Bucuresti', 'Mihai Eminescu', 040294)

into cititor values (15, 1, 'Marin', 'Razvan', 36, 'lrscns@gmail.com', 'Bucuresti', 'Republicii', 021332)

into cititor values (16, 6, 'Marin', 'Marius', 23, 'barbumrn@gmail.com', 'Braila', 'Traian', 810012)

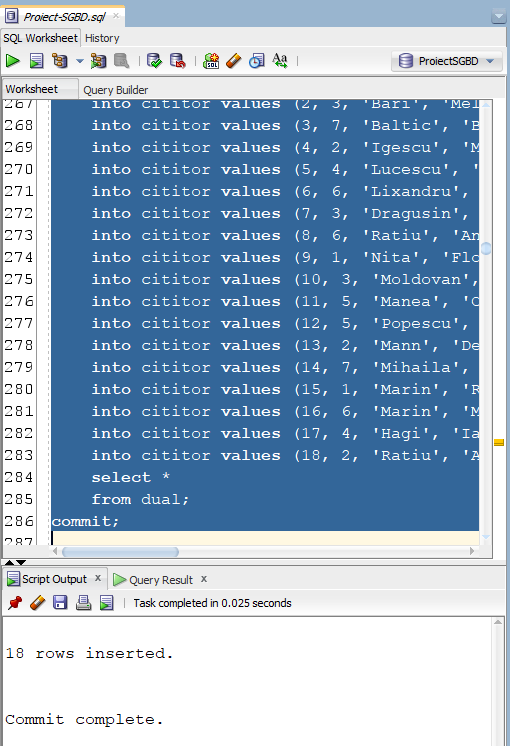
into cititor values (17, 4, 'Hagi', 'Ianis', 40, 'stanelena@yahoo.com', 'Arad', 'Mihai Viteazu', 310021)

into cititor values (18, 2, 'Ratiu', 'Andrei', 28, 'rayvaleccano@gmail.com', 'Barcelona', 'Stadionului', 000100)

select \*

from dual;

commit;



Inserarea de date în tabela ORGANIZEAZA

insert all

into organizeaza values (1, 1)

into organizeaza values (2, 1)

into organizeaza values (6, 2)

into organizeaza values (8, 3)

into organizeaza values (10, 4)

into organizeaza values (3, 5)

into organizeaza values (1, 6)

into organizeaza values (3, 7)

into organizeaza values (5, 7)

into organizeaza values (10, 7)

into organizeaza values (12, 8)

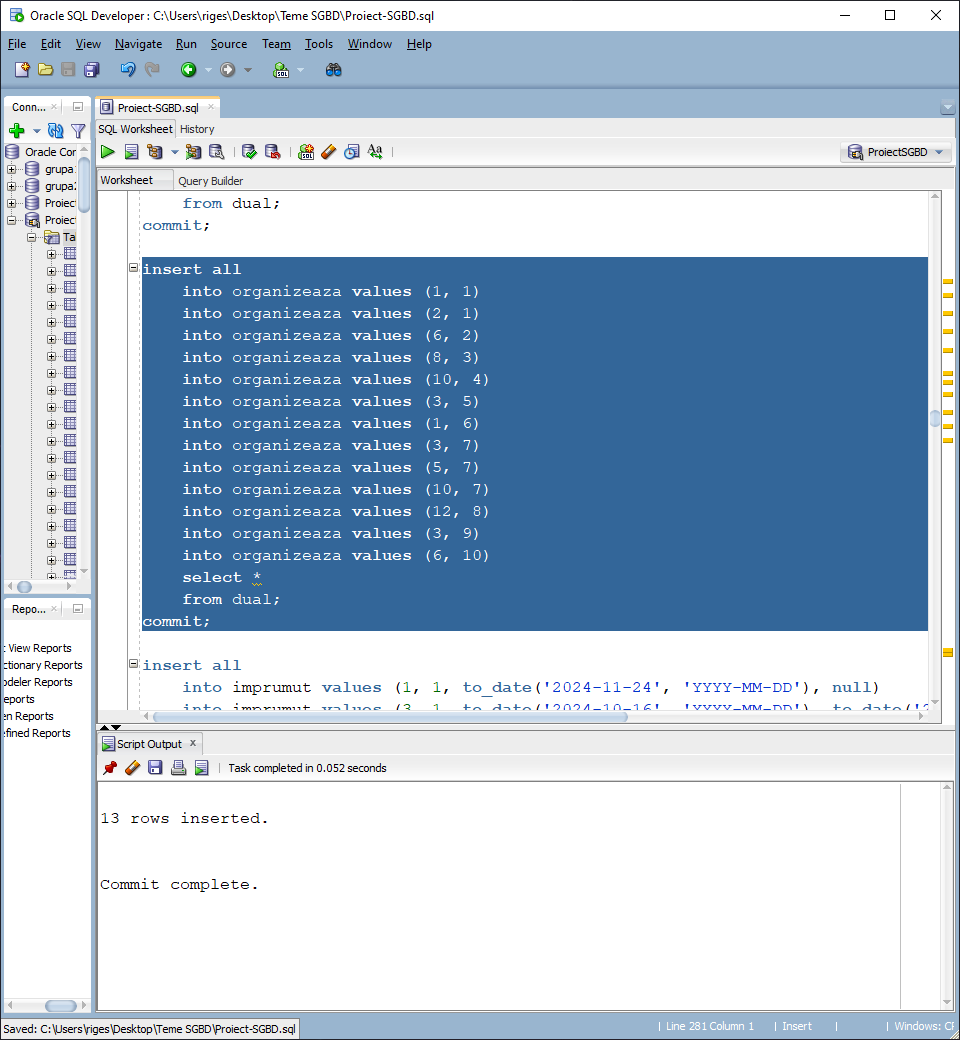
into organizeaza values (3, 9)

into organizeaza values (6, 10)

select \*

from dual;

commit;



Inserarea de date în tabela IMPRUMUT

insert all

into imprumut values (1, 1, to\_date('2024-11-24', 'YYYY-MM-DD'), null)

into imprumut values (3, 1, to\_date('2024-10-16', 'YYYY-MM-DD'), to\_date('2024-10-23', 'YYYY-MM-DD'))

into imprumut values (2, 1, to\_date('2023-09-27', 'YYYY-MM-DD'), to\_date('2023-10-05', 'YYYY-MM-DD'))

into imprumut values (10, 3, to\_date('2024-07-01', 'YYYY-MM-DD'), null)

into imprumut values (16, 3, to\_date('2019-03-10', 'YYYY-MM-DD'), to\_date('2019-03-21', 'YYYY-MM-DD'))

into imprumut values (3, 4, to\_date('2024-12-03', 'YYYY-MM-DD'), to\_date('2024-12-23', 'YYYY-MM-DD'))

into imprumut values (4, 7, to\_date('2023-10-08', 'YYYY-MM-DD'), to\_date('2023-10-16', 'YYYY-MM-DD'))

into imprumut values (5, 10, to\_date('2022-06-09', 'YYYY-MM-DD'), null)

into imprumut values (7, 10, to\_date('2020-02-01', 'YYYY-MM-DD'), to\_date('2020-03-15', 'YYYY-MM-DD'))

into imprumut values (8, 10, to\_date('2019-08-12', 'YYYY-MM-DD'), null)

into imprumut values (12, 13, to\_date('2024-02-06', 'YYYY-MM-DD'), null)

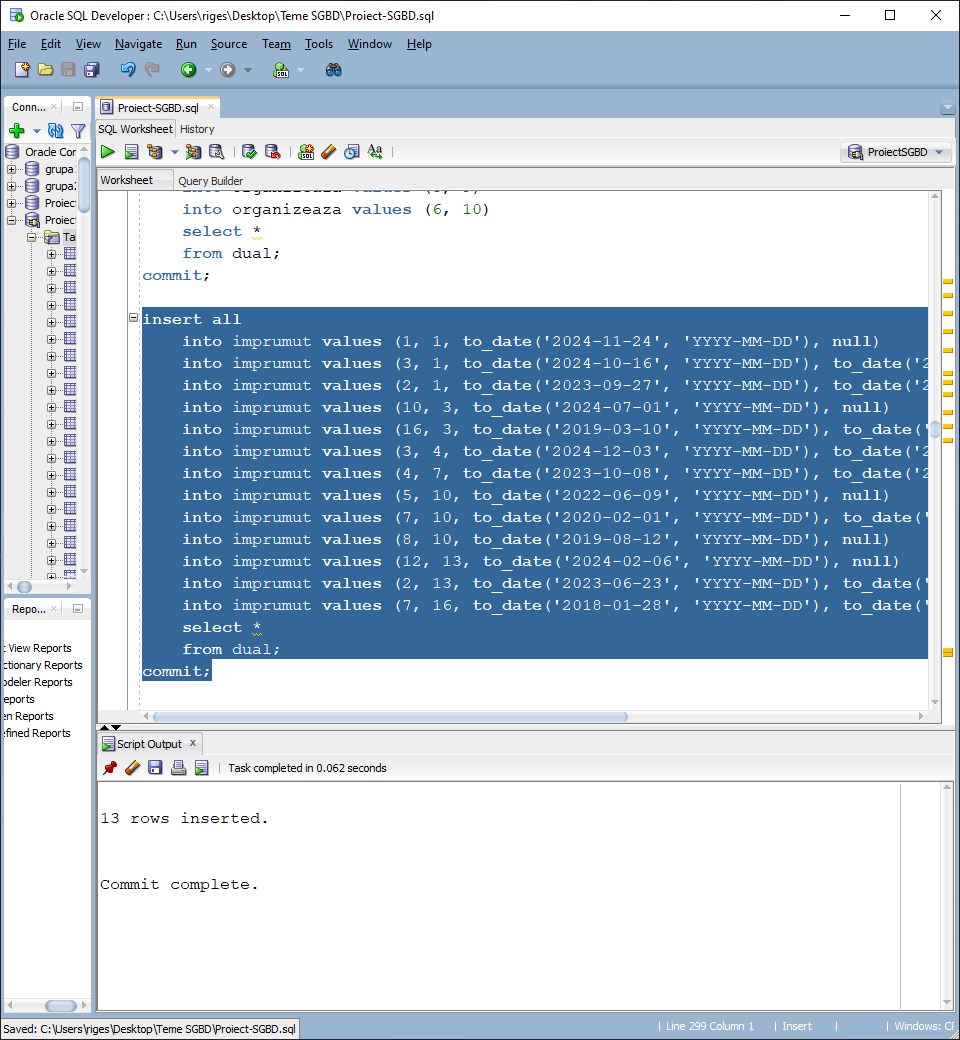
into imprumut values (2, 13, to\_date('2023-06-23', 'YYYY-MM-DD'), to\_date('2023-06-28', 'YYYY-MM-DD'))

into imprumut values (7, 16, to\_date('2018-01-28', 'YYYY-MM-DD'), to\_date('2028-02-10', 'YYYY-MM-DD'))

select \*

from dual;

commit;



# Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze cele 3 tipuri de colecții studiate. Apelați subprogramul.

**Enunț:** Să se implementeze un subprogram stocat independent care să gestioneze lista evenimentelor organizate de toți cititorii care au un împrumut la activ (care au data de returnare egală cu null) grupate după sala în care au avut loc.

create or replace type sala is varray(20) of varchar2(50);

create or replace procedure exercitiul\_6 is

type nested\_tipuri is table of varchar2(50);

type indexat\_tipuri\_sala is table of nested\_tipuri index by varchar2(50);

cursor c is

select c.id\_cititor

from cititor c

join imprumut i on i.id\_cititor = c.id\_cititor

where data\_returnare is null

order by 1 asc;

lista\_sala sala;

lista\_tipuri nested\_tipuri;

lista\_ambele indexat\_tipuri\_sala;

v\_id\_cititor number;

begin

open c;

loop

fetch c into v\_id\_cititor;

exit when c%notfound;

select distinct e.sala

bulk collect into lista\_sala

from eveniment e

join organizeaza o on e.id\_eveniment = o.id\_eveniment

where o.id\_cititor = v\_id\_cititor;

for i in 1..lista\_sala.count loop

select e.nume

bulk collect into lista\_tipuri

from eveniment e

join organizeaza o on e.id\_eveniment = o.id\_eveniment

where o.id\_cititor = v\_id\_cititor

and e.sala = lista\_sala(i);

lista\_ambele(lista\_sala(i)) := lista\_tipuri;

end loop;

if lista\_sala.count = 0 then

dbms\_output.put\_line('Cititorul cu ID-ul ' || v\_id\_cititor || ' nu a organizat niciun eveniment.');

dbms\_output.new\_line;

dbms\_output.put\_line('------------------------------------------------');

dbms\_output.new\_line;

else

dbms\_output.put\_line('Mai jos sunt prezentate evenimentele organizate de cititorul cu ID-ul: ' || v\_id\_cititor);

for i in 1..lista\_sala.count loop

dbms\_output.put\_line('Sala: ' || lista\_sala(i));

if lista\_ambele.exists(lista\_sala(i)) then

for j in 1..lista\_ambele(lista\_sala(i)).count loop

dbms\_output.put\_line(' Nume: ' || lista\_ambele(lista\_sala(i))(j));

end loop;

end if;

end loop;

dbms\_output.new\_line;

dbms\_output.put\_line('------------------------------------------------');

dbms\_output.new\_line;

end if;

end loop;

end;

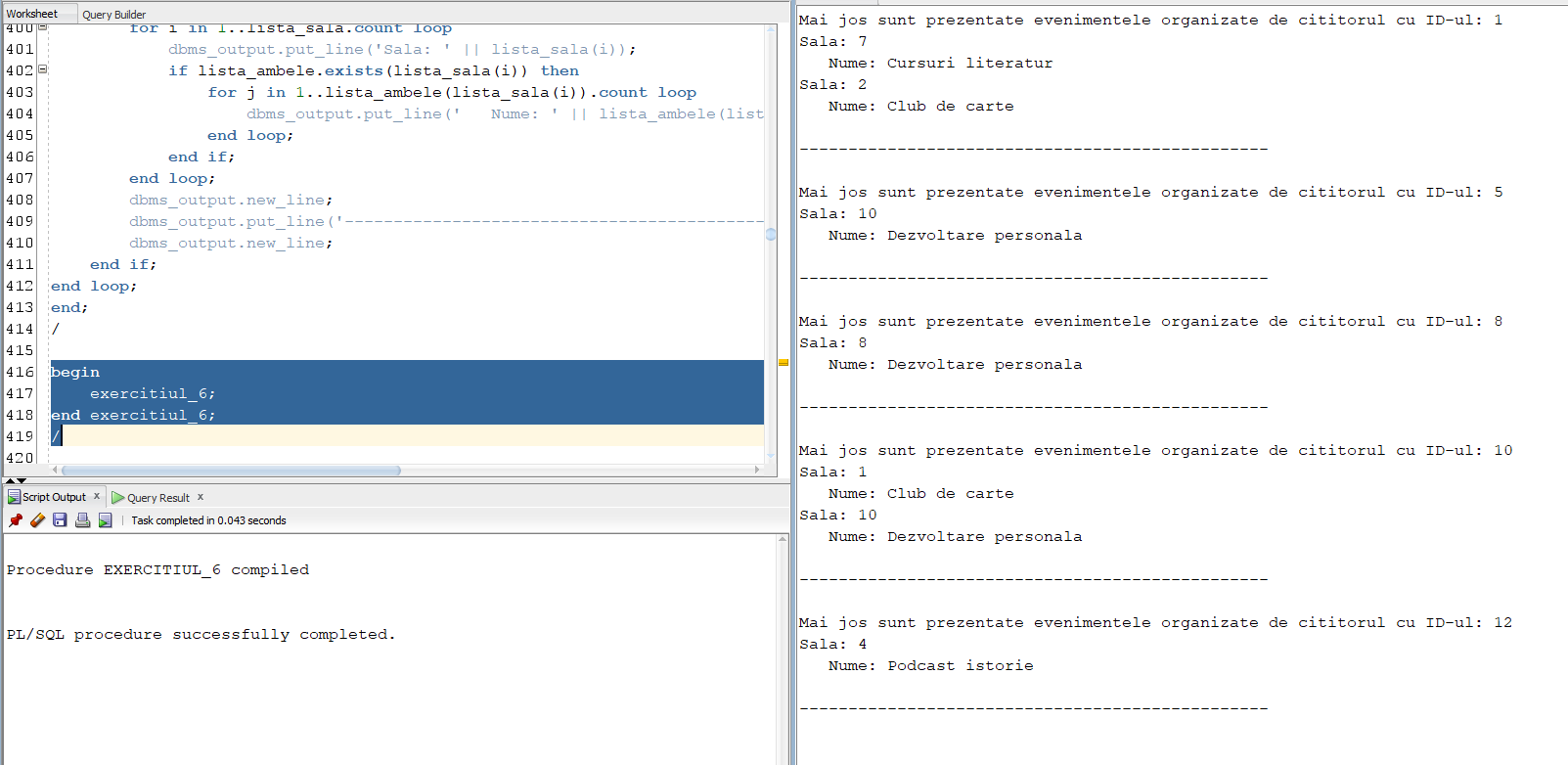
/

begin

exercitiul\_6;

end exercitiul\_6;

/



# Formulați în limbaj natural o problem pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul fiind cursor parametrizat dependent de cursorul anterior.

**Enunt:** Să se afișeze pentru fiecare editură numele autorilor și titlurile cărților scrise de ei care au fost publicate de editura respective după anul 2000 și care au fost împrumutate cel putin o data.

create or replace procedure exercitiul\_7 is

cursor edituri is

select id\_editura, nume

from editura;

cursor carti\_autori(v\_id\_editura number) is

select distinct a.nume, a.prenume, c.titlu

from autor a

join redacteaza r on a.id\_autor = r.id\_autor

join carte c on c.id\_carte = r.id\_carte

join imprumut i on i.id\_carte = c.id\_carte

where c.id\_editura = v\_id\_editura

and c.an\_aparitie > 2000;

v\_editura edituri%rowtype;

v\_carti\_autori carti\_autori%rowtype;

bool number := 0;

begin

open edituri;

loop

fetch edituri into v\_editura;

exit when edituri%notfound;

dbms\_output.put\_line('Editura: ' || v\_editura.nume);

open carti\_autori(v\_editura.id\_editura);

bool := 1;

loop

fetch carti\_autori into v\_carti\_autori;

exit when carti\_autori%notfound;

bool := 0;

dbms\_output.put\_line(' Autor: ' || v\_carti\_autori.nume || ' ' || v\_carti\_autori.prenume);

dbms\_output.put\_line(' Carte: ' || v\_carti\_autori.titlu);

dbms\_output.new\_line;

end loop;

if bool = 1 then

dbms\_output.put\_line(' Nu are nicio carte publicata care sa respecte cerintele.');

end if;

close carti\_autori;

dbms\_output.new\_line;

dbms\_output.put\_line('------------------------------------------');

dbms\_output.new\_line;

end loop;

close edituri;

end exercitiul\_7;

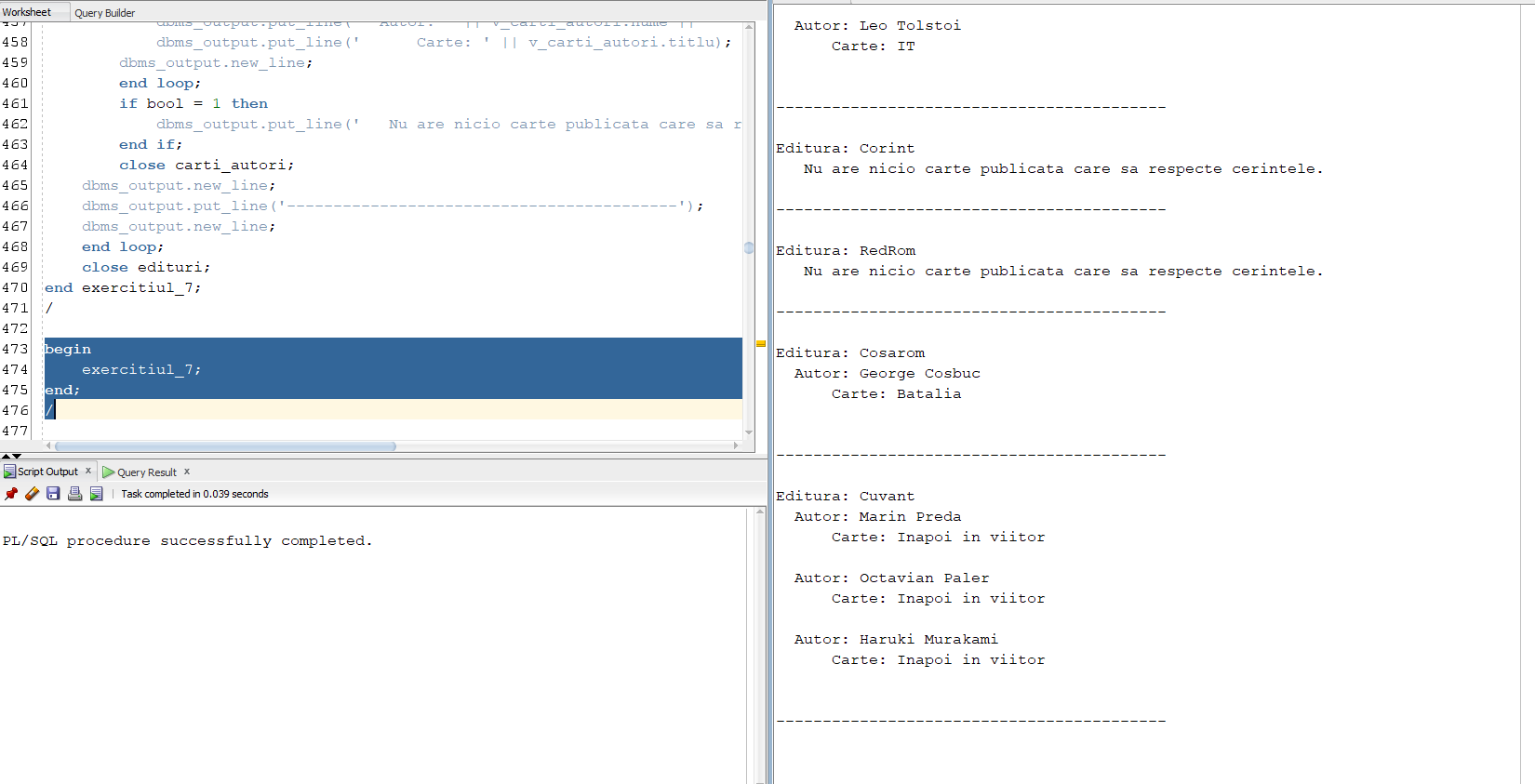
/

begin

exercitiul\_7;

end;

/



# Formulați în limbaj natural o problem pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele create. Tratați toate excepțiile care pot apărea.

**Enunț:** Fomulați o funcție care să primească numele complet al unui cititor ca parametru și returnează cartea cel mai recent împrumutată de acel cititor. Cartea va fi afișată cu toate datele sale la care se adaugă aici și autorul/autorii care au redactat acea carte împreuna cu naționalitatea lor.

create or replace function exercitiul\_8(

nume\_cititor in varchar2) return varchar2 is

v\_id\_carte number;

v\_an\_aparitie number;

v\_gen varchar2(50);

v\_nr\_autori number;

v\_titlu varchar2(50);

v\_nr number;

v\_rezultat varchar2(2000);

begin

select count(\*)

into v\_nr

from cititor

where (nume || ' ' || prenume) = nume\_cititor;

if v\_nr = 0 then

return 'Cititorul nu a fost gasit in baza de date.';

elsif v\_nr > 1 then

raise too\_many\_rows;

end if;

select ca.id\_carte, ca.titlu, ca.an\_aparitie, ca.gen, ca.nr\_autori

into v\_id\_carte, v\_titlu, v\_an\_aparitie, v\_gen, v\_nr\_autori

from carte ca

join imprumut i on i.id\_carte = ca.id\_carte

join cititor c on c.id\_cititor = i.id\_cititor

where (c.nume || ' ' || c.prenume) = nume\_cititor

order by i.data\_imprumut desc

fetch first 1 row only;

v\_rezultat := 'Ultima carte imprumutata: '

|| v\_titlu || chr(10) ||

' Anul aparitiei: ' || v\_an\_aparitie

|| chr(10) || ' Genul cartii: ' || v\_gen ||

chr(10) || ' Numarul de autori: ' || v\_nr\_autori || ' care sunt urmatorii: ' || chr(10);

for i in (select a.nume, a.prenume, a.nationalitate

from autor a

join redacteaza r on r.id\_autor = a.id\_autor

where r.id\_carte = v\_id\_carte) loop

v\_rezultat := v\_rezultat || ' Numele: ' || i.nume ||

' ' || i.prenume || chr(10) ||

' Nationalitate: ' || i.nationalitate || chr(10);

end loop;

return v\_rezultat;

exception

when no\_data\_found then

return 'Cititorul nu a facut niciun imprumut in baza de date';

when too\_many\_rows then

return 'Exista mai multi cititori cu acest nume.';

when others then

return 'Alta eroare.';

end;

/

declare

v\_nume varchar2(50) := '&nume';

begin

dbms\_output.put\_line(exercitiul\_8(v\_nume));

end;

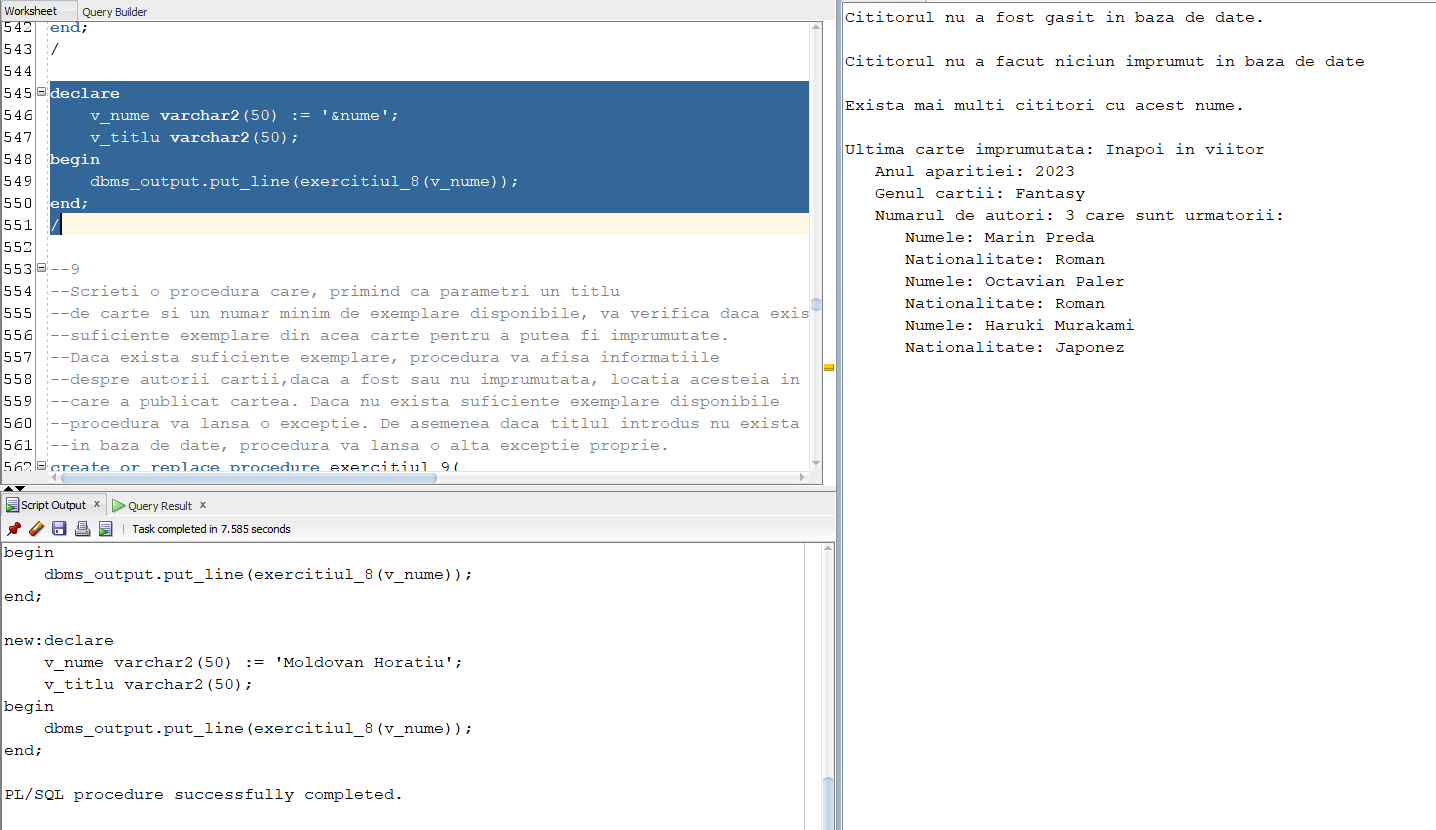
/

Dacă încercăm să introducem un nume care nu există în baza de date, de exemplu ”test”, se va afișa primul mesaj din output.

Dacă introducem un nume valid, de exemplu “Ianis Hagi”, dar care nu are niciun împrumut la activ, pe ecran va apărea al doilea mesaj din output.

Dacă introducem numele „Ratiu Andrei”, se va afișa un mesaj care ne anunță ca există mai mulți cititori cu acest nume.

La introducerea numelui “Moldovan Horatiu”, vom avea afișarea normală.



# Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să aibă minim 2 parametri și să utilizeze într-o singură comandă SQL 5 tabele create. Definiți minim 2 excepții proprii.

**Enunt:** Scrieți o procedură care primește ca parametri titlul complet al unei cărți și un număr reprezentând minimul de exemplare pe care un cititor vrea să le împrumute. Subprogramul va verifica dacă există destule copii ale cărții specificate disponibile în bibliotecă, iar dacă da, va afișa informații cu privire la autorii cărții, daca a fost sau nu înprumutată, locația acesteia în bibliotecă și editura care a publicat cartea. Dacă nu există suficiente exemplare, subprogramul va arunca o eroare proprie. Totodată, dacă numărul de exemplare introdus de la tastură de utilizator este negativ, se va arunca o altă eroare proprie. Sunt tratate și erorile de bază TOO\_MANY\_ROWS și NO\_DATA\_FOUND.

create or replace procedure exercitiul\_9(

v\_titlu in varchar2,

v\_min\_exe in number)

is

numar\_invalid exception;

exemplare\_insuficiente exception;

v\_exemplare number;

v\_id\_carte number;

v\_id\_editura number;

v\_locatie varchar2(100);

v\_nume\_editura varchar2(100);

v\_numar\_afisat number;

v\_stare varchar2(50);

begin

if v\_min\_exe < 0 then

raise numar\_invalid;

end if;

select id\_carte, id\_editura

into v\_id\_carte, v\_id\_editura

from carte

where titlu = v\_titlu;

select distinct (l.etaj || ' - ' || l.zona || ' - Raftul ' || l.raft), e.nume, s.nr\_disponibile, decode(i.data\_imprumut,

null, 'Nu a fost imprumutata',

'A fost imprumutata') as stare

into v\_locatie, v\_nume\_editura, v\_exemplare, v\_stare

from locatie l

join carte c on c.id\_locatie = l.id\_locatie

join editura e on c.id\_editura = e.id\_editura

join stoc s on c.id\_carte = s.id\_carte

left join imprumut i on i.id\_carte = c.id\_carte

where c.id\_carte = v\_id\_carte;

if v\_exemplare < v\_min\_exe then

raise exemplare\_insuficiente;

end if;

dbms\_output.put\_line('Autori pentru cartea "' || v\_titlu || '":');

for autor\_rec in (

select a.nume, a.prenume

from autor a

join redacteaza r on a.id\_autor = r.id\_autor

where r.id\_carte = v\_id\_carte) loop

dbms\_output.put\_line(' - ' || autor\_rec.nume || ' ' || autor\_rec.prenume);

end loop;

dbms\_output.put\_line('Locatia cartii: ' || v\_locatie);

dbms\_output.put\_line('Editura cartii: ' || v\_nume\_editura);

dbms\_output.put\_line('Stare: ' || v\_stare);

exception

when no\_data\_found then

dbms\_output.put\_line('Eroare: Titlul nu poate fi gasit in baza de date.');

when numar\_invalid then

dbms\_output.put\_line('Eroare: Numarul introdus nu este valid' || chr(10) || 'incercati un numar natural cel putin mai mare sau egal cu 1.');

when exemplare\_insuficiente then

dbms\_output.put\_line('Eroare: Nu exista suficiente exemplare pentru cartea: "' || v\_titlu || '".');

when too\_many\_rows then

dbms\_output.put\_line('Eroare: Mai multe carti au acelasi nume.');

when others then

dbms\_output.put\_line('Eroare neprevazuta.');

end;

/

declare

v\_titlu varchar2(100) := '&carte';

v\_nr number :=&numar;

begin

exercitiul\_9(v\_titlu, v\_nr);

end;

/

Ca înainte, voi trata fiecare eroare în parte.

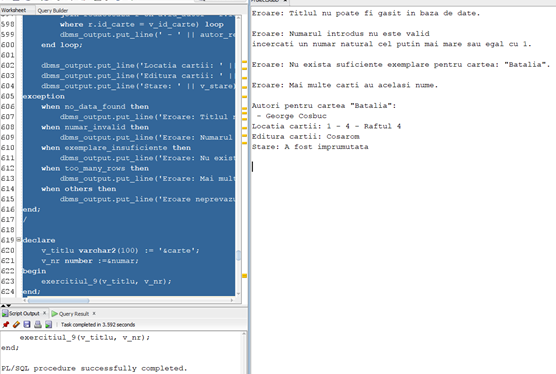
Dacă introducem de la tastatură un titlu, “test”, vom primi un mesaj care ne anunță că el nu poate fi găsit în baza de date.

Dacă tastăm un numar invalid (mai mic sau egal cu 0), programul ne va anunța sa ne schimbăm cererea, cu un număr cel puțin egal cu 1.

Dacă nu există destule exemplare pentru o carte introdusă, se va afișa un mesaj corespunzător cererii.

Dacă există mai multe cărți cu același nume, vom fi anunțați corespunzător.

În orice alt caz corect, vom primi răspunsul normal al programului.



# Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

**Enunț:** Definiți un trigger care monitorizează numărul total de autori al unei cărți. De fiecare dată când un nou autor este adăugat pentru o anumită carte (este inserat în tabela redactează) atunci se va actualiza automat coloana nr\_autori din tabela carte, mărindu-se cu 1.

create or replace trigger actualizare\_autori

after insert on redacteaza

begin

update carte c

set nr\_autori = (

select count(\*)

from redacteaza r

where r.id\_carte = c.id\_carte)

where exists (

select 1

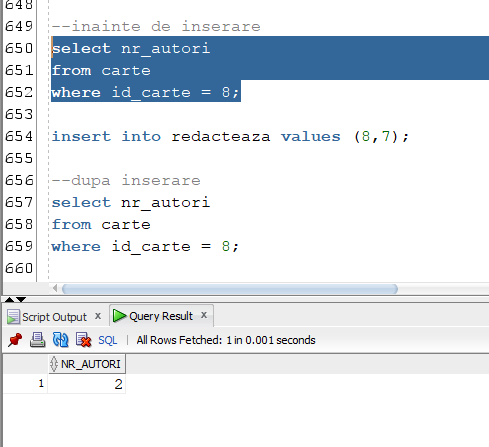
from redacteaza r

where r.id\_carte = c.id\_carte);

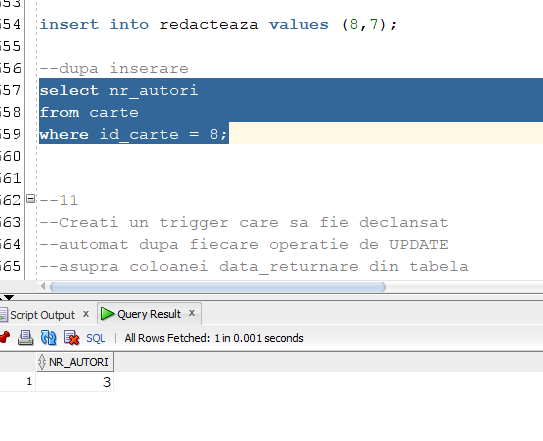
end;

/

Dacă luăm de exemplu cartea cu ID-ul 8, putem vedea ca inițial ea are 2 autori.



Însă, dacă inserăm valoarea (8, 7) în tabela REDACTEAZA, se va actualiza automat și daca rulăm același set de instrucțiuni putem vedea că numărul de autori s-a mărit.



# Definți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.

**Enunț:** Creați un trigger care să fie declanșat automat după fiecare operație de UPDATE sau INSERT asupra tabelei împrumut. Trigger-ul va actualiza tabela STOC incrementând sau decrementând numărul de exemplare disponibile pentru cartea returnată/împrumutată.

create or replace trigger actualizare\_stoc

after insert or update

on imprumut

for each row

begin

if updating then

if :OLD.data\_returnare is null and :NEW.data\_returnare is not null then

update stoc

set nr\_disponibile = nr\_disponibile + 1

where id\_carte = :NEW.id\_carte;

end if;

end if;

if inserting then

declare

v\_nr\_disponibile number;

begin

select nr\_disponibile

into v\_nr\_disponibile

from stoc

where id\_carte = :NEW.id\_carte;

if v\_nr\_disponibile <= 0 then

raise\_application\_error(-20001, 'Nu mai sunt exemplare in biblioteca.');

end if;

update stoc

set nr\_disponibile = nr\_disponibile - 1

where id\_carte = :NEW.id\_carte;

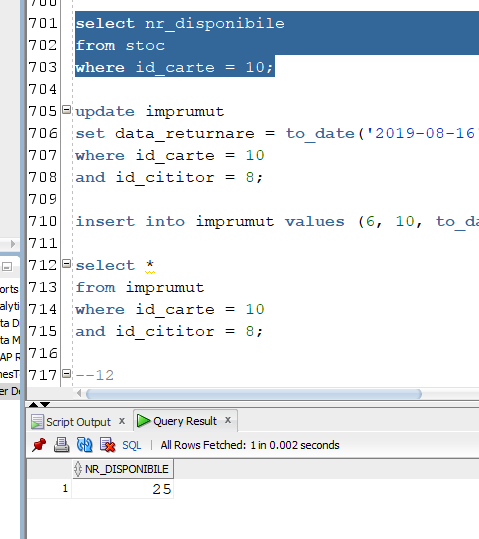
end;

end if;

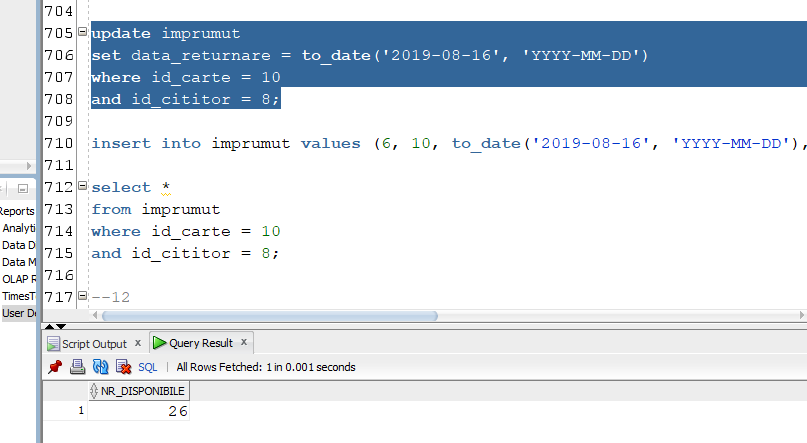
end;

/

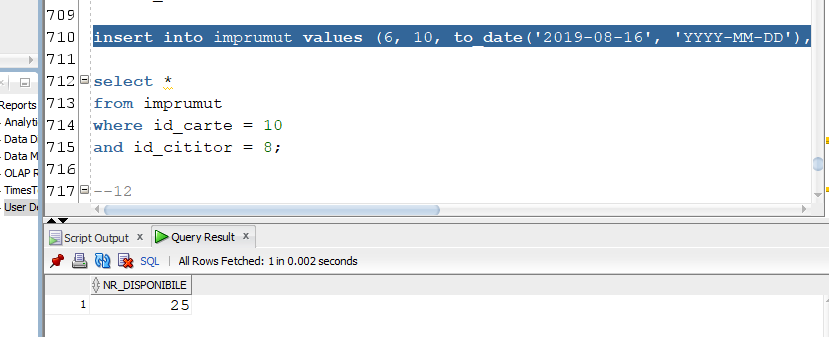
Putem observa că inițial pentru cartea cu ID-ul 10, avem 25 de exemplare disponibile.



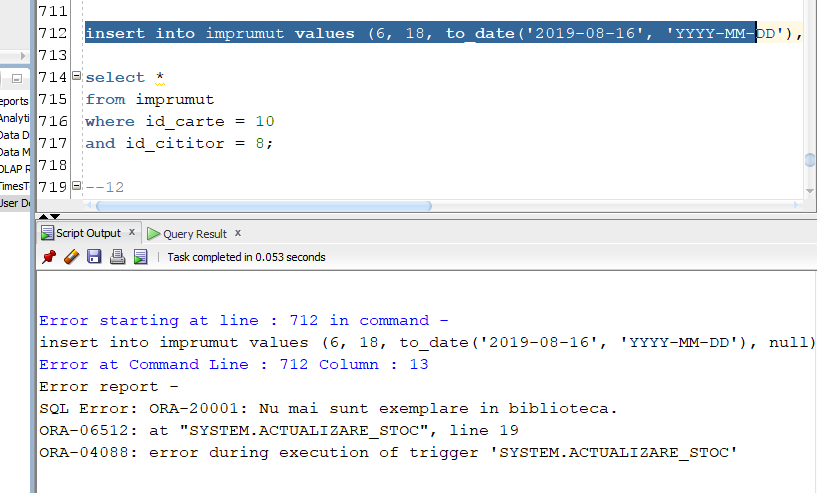
Dacă actualizăm un împrumut pentru cartea cu același ID, putem observa ca numărul se mărește automat.



Totodată, dacă inserăm un nou împrumut pentru cartea cu ID-ul 10, numărul de exemplare va scădea automat.



Dacă încercăm să inserăm un nou împrumut pentru o carte ce nu mai are exemplare disponibile vom avea o eroare. În exemplul meu voi folosi cartea cu ID-ul 18.



# Definți un trigger de tip LDD. Declanșați trigger-ul.

**Enunț:** Definiți un trigger de tip LDD care să stocheze într-o tabelă de audit când o altă tabelă este create modificată sau ștearsă.

create table log1 (

moment\_timp timestamp,

utilizator varchar2(50),

tip varchar2(50),

nume\_obiect varchar2(50),

tip\_obiect varchar2(50));

create or replace trigger exercitiul\_12

after create or alter or drop

on schema

begin

insert into log1 (

moment\_timp,

utilizator,

tip,

nume\_obiect,

tip\_obiect)

values (

systimestamp,

user,

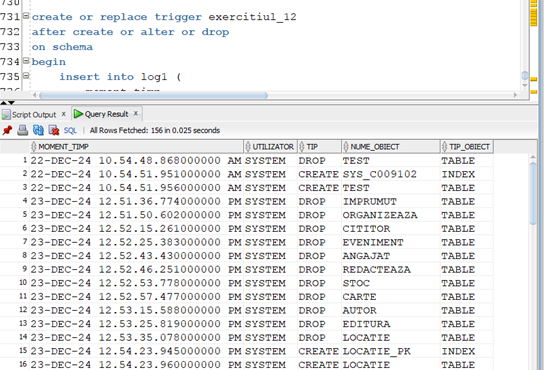
ora\_sysevent,

ora\_dict\_obj\_name,

ora\_dict\_obj\_type);

end;

/



# Formulați în limbaj natural o problemă pe care să o rezolvați folosind un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite.

**Enunț:** Pentru toți cititorii care au realizat împrumuturi, biblioteca vrea să le recomande alte cărți pe baza genurilor cărților pe care deja le-au citit. Cărțile vor avea un scor de popularitate și se vor recomanda primele 3 cele mai populare cărți din fiecare gen. Scorul de popularitate este dat de numărul de împrumuturi pe care acea carte le are înregistrate în bibliotecă. Cărțile recomandate pentru fiecare cititor vor fi stocate mai apoi în tabelul CARTI\_RECOMANDATE.

Mai întâi vom crea tabela CARTI\_RECOMANDATE

create table carti\_recomandate(

id\_recomandare number(3) generated by default as identity primary key,

id\_cititor number(3),

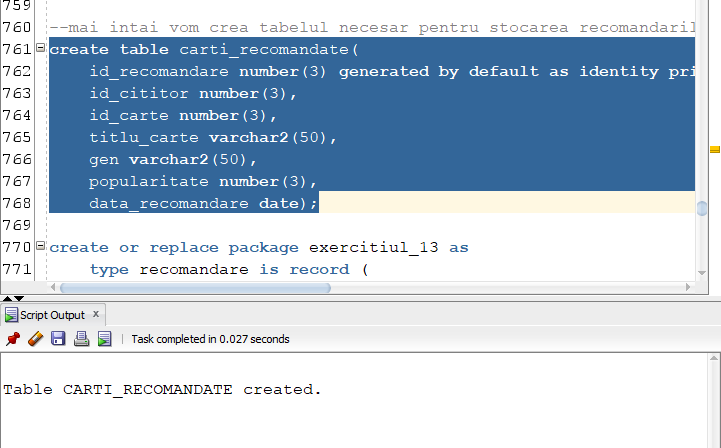
id\_carte number(3),

titlu\_carte varchar2(50),

gen varchar2(50),

popularitate number(3),

data\_recomandare date);



Apoi voi crea pachetul dorit

create or replace package exercitiul\_13 as

type recomandare is record (

id\_carte number(3),

titlu varchar2(50),

gen varchar2(50),

popularitate number);

type vector\_de\_recomandari is table of recomandare;

type vector\_de\_genuri is table of varchar2(50);

function calc\_popularitate(id\_carte\_imprumutata number) return number;

function genuri\_imprumutate(v\_id\_cititor number) return vector\_de\_genuri;

procedure de\_recomandat(v\_id\_cititor number);

procedure commit\_to\_database(v\_id\_cititor number, recomandari vector\_de\_recomandari);

end exercitiul\_13;

/

Și în final voi defini fiecare funcție și procedură

create or replace package body exercitiul\_13 as

function calc\_popularitate(id\_carte\_imprumutata number) return number is

popularitate number;

begin

select count(\*)

into popularitate

from imprumut

where id\_carte = id\_carte\_imprumutata;

return popularitate;

end calc\_popularitate;

function genuri\_imprumutate(v\_id\_cititor number) return vector\_de\_genuri is

genuri vector\_de\_genuri := vector\_de\_genuri();

begin

select distinct ca.gen

bulk collect into genuri

from carte ca

join imprumut i on i.id\_carte = ca.id\_carte

where i.id\_cititor = v\_id\_cititor;

return genuri;

end genuri\_imprumutate;

procedure commit\_to\_database(v\_id\_cititor number, recomandari vector\_de\_recomandari) is

begin

for i in 1..recomandari.count loop

insert into carti\_recomandate (id\_cititor, id\_carte, titlu\_carte, gen, popularitate, data\_recomandare)

values (v\_id\_cititor, recomandari(i).id\_carte, recomandari(i).titlu, recomandari(i).gen, recomandari(i).popularitate, sysdate);

end loop;

end commit\_to\_database;

procedure de\_recomandat(v\_id\_cititor number) is

genuri vector\_de\_genuri;

lista vector\_de\_recomandari := vector\_de\_recomandari();

v\_recomandare recomandare;

begin

genuri := genuri\_imprumutate(v\_id\_cititor);

for i in 1..genuri.count loop

for carte in (select ca.id\_carte, ca.titlu, ca.gen, calc\_popularitate(ca.id\_carte) as popularitate

from carte ca

where ca.gen = genuri(i)

order by popularitate desc

fetch first 3 rows only) loop

v\_recomandare.id\_carte := carte.id\_carte;

v\_recomandare.titlu := carte.titlu;

v\_recomandare.popularitate := carte.popularitate;

v\_recomandare.gen := carte.gen;

lista.extend;

lista(lista.count) := v\_recomandare;

end loop;

end loop;

commit\_to\_database(v\_id\_cititor, lista);

end de\_recomandat;

end exercitiul\_13;

/

În momentul în care apelez procedura “de\_recomandat” pentru cititorul ce are ID-ul 3, se vor adăuga în mod automat în tabelul CARTI\_RECOMANDATE, cărțile care se potrivesc acestuia.

