**Proiect SGBD – Magazin Online de Carti**

1. Prezentarea bazei de date si utilitatea ei:

Baza de date simuleaza procesul unui magazin online, prin functionalitatea de a avea un cont, de a da comenzi (sau de a pune anumite carti in cos), de a lasa review-uri la produse si de a grupa aceste carti pe categorii.

Avem tabelele: book, category, review, basket, book\_in\_basket, usser (cu 2 de s).

Tabelele asociative sunt basket si book. Cardinalitatea bazei de date contine relatii de doua feluri: many-to-many si one-to-many.

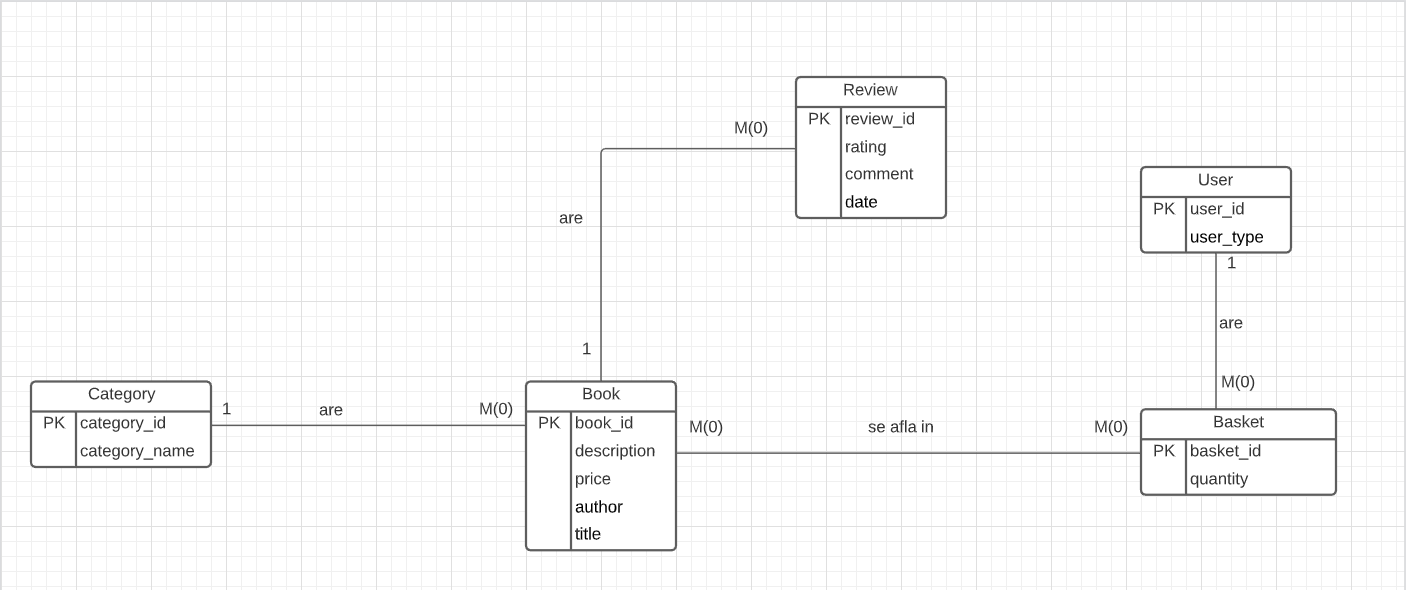
Tabela book contine atributele id-ul cartii, descriere, pret, autor si titlu.

Tabela category contine id-ul categoriei si denumirea ei.

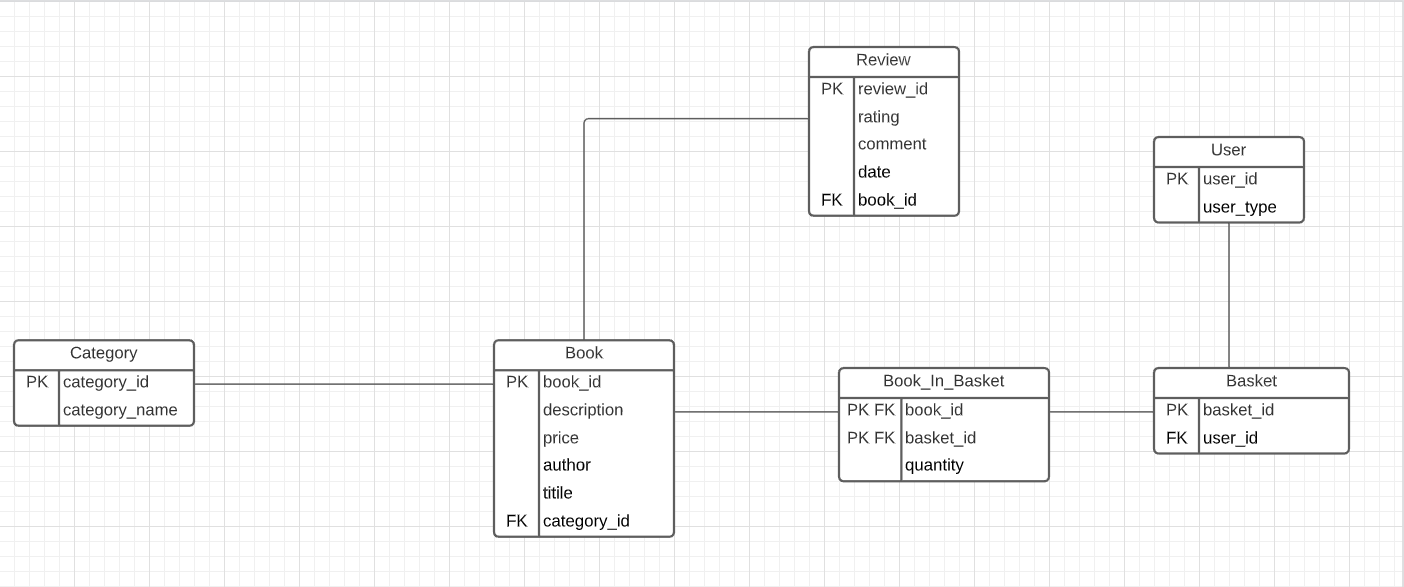
Tabela review contine id-ul review-ului, ratingul, comentariul si data adaugarii acestora.

Tabela user contine id-ul userului si tipul acestuia (Admin, Colaborator, Utilizator Inregistrat, Utilizator Neinregistrat). Tabela basket contine id-ul cosului si cantitatea.

1. Diagrama entitate-relatie (ERD):



1. Diagrama conceptuala:



1. Crearea tabelelor:

DROP TABLE book;

CREATE TABLE book(

book\_id INT NOT NULL PRIMARY KEY,

description VARCHAR(256),

price VARCHAR(256),

author VARCHAR(256),

title VARCHAR(256),

category\_id INT NOT NULL

);

ALTER TABLE book

MODIFY price decimal(6,2);

DROP TABLE category;

CREATE TABLE category(

category\_id INT NOT NULL PRIMARY KEY,

category\_name VARCHAR(256)

);

DROP TABLE review;

CREATE TABLE review(

review\_id INT NOT NULL PRIMARY KEY,

rating INT,

coment VARCHAR(256),

data DATE,

book\_id INT NOT NULL

);

DROP TABLE usser;

CREATE TABLE usser(

user\_id INT NOT NULL PRIMARY KEY,

user\_type VARCHAR(256)

);

DROP TABLE basket;

CREATE TABLE basket(

basket\_id INT NOT NULL PRIMARY KEY,

user\_id INT NOT NULL

);

DROP TABLE book\_in\_basket;

CREATE TABLE book\_in\_basket(

book\_id INT,

basket\_id INT,

quantity INT

);

ALTER TABLE book

ADD FOREIGN KEY (category\_id) REFERENCES category(category\_id);

ALTER TABLE review

ADD FOREIGN KEY (book\_id) REFERENCES book(book\_id);

ALTER TABLE basket

ADD FOREIGN KEY (user\_id) REFERENCES usser(user\_id);

ALTER TABLE book\_in\_basket

ADD FOREIGN KEY (book\_id) REFERENCES book(book\_id);

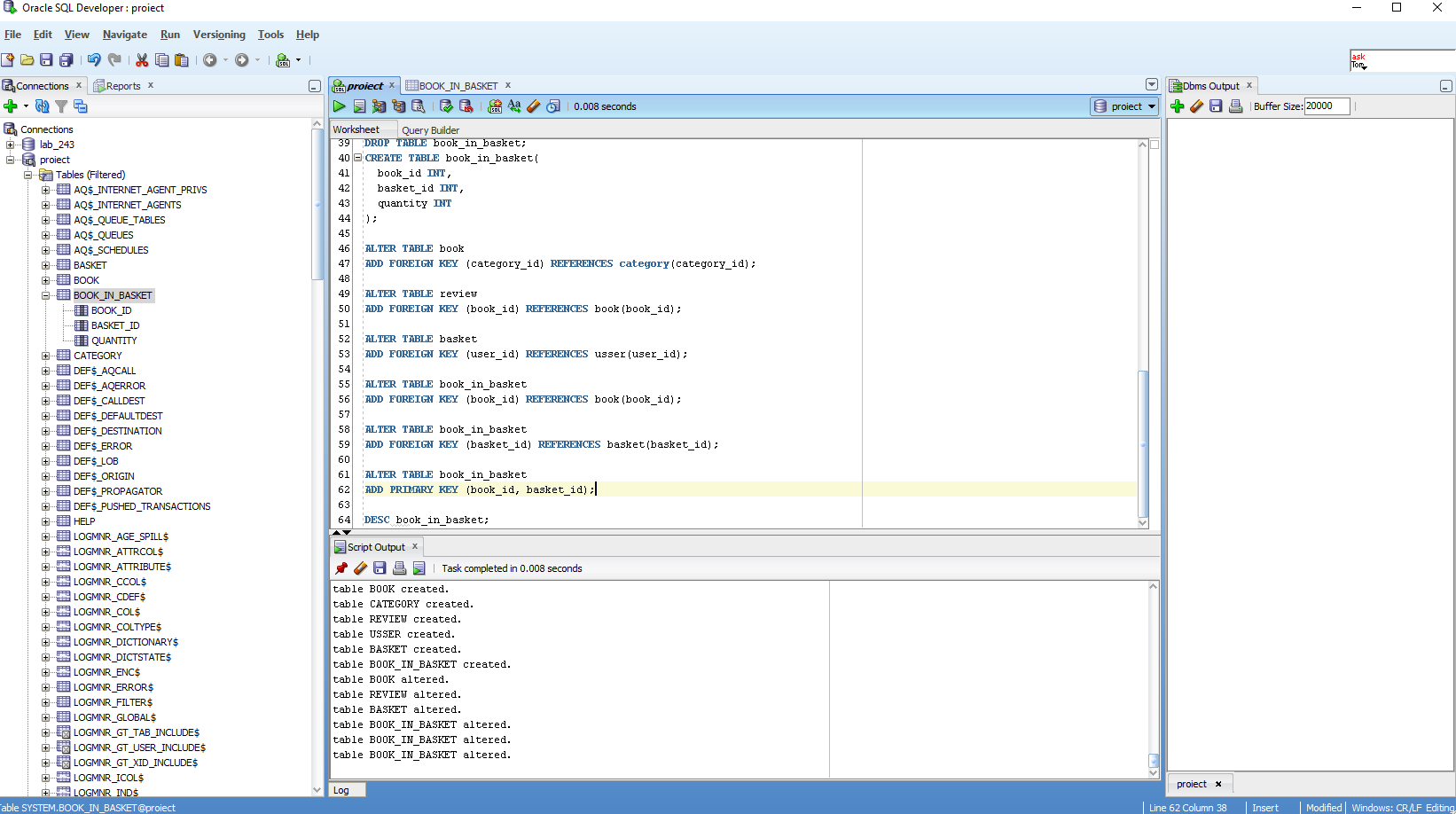
ALTER TABLE book\_in\_basket

ADD FOREIGN KEY (basket\_id) REFERENCES basket(basket\_id);

ALTER TABLE book\_in\_basket

ADD PRIMARY KEY (book\_id, basket\_id);

Print-screen:



1. Inserarea datelor:

INSERT INTO usser VALUES (1,'Admin');

INSERT INTO usser VALUES (2,'Colaborator');

INSERT INTO usser VALUES (3,'Utilizator inregistrat');

INSERT INTO usser VALUES (4,'Utilizator inregistrat');

INSERT INTO usser VALUES (5,'Utilizator neinregistrat');

INSERT INTO category VALUES (1,'Drama');

INSERT INTO category VALUES (2,'Horror');

INSERT INTO category VALUES (3,'Poezie');

INSERT INTO category VALUES (4,'Aventura');

INSERT INTO category VALUES (5,'Fantasy');

INSERT INTO book VALUES (1,'În vâltoarea celui de-al Doilea R?zboi Mondial, cinci tineri americani ideali?ti primesc câte o scrisoare misterioas? de la Oficiul pentru Servicii Strategice, în care li se pune întrebarea dac? sunt dispu?i s? lupte pentru ?ara lor.', 49.90, 'LEILA MEACHAM', 'Zborul libelulei', 1);

INSERT INTO book VALUES (2,'Sub conducerea dictatoriala a generalului Francisco Franco, Spania continua sa traiasca, dar redusa la tacere.', 256.90, 'RUTA SEPETYS', 'Fantanile tacerii', 1);

INSERT INTO book VALUES (3,'Noi, cei care putem citi, nu ne putem imagina cu ce greutati se confrunta acesti oameni în lumea scrisului în care traim cu totii.', 62.00, 'SEBASTIAN FITZEK', 'Cadoul', 2);

INSERT INTO book VALUES (4,'Cand Tommy si Tuppence o viziteaza pe batrana lor matusa intr-un azil de batrani, nu dau atentie prea mare nici motivelor ei de suparare, nici pacientei care se plange ca a fost otravita.', 34.90, 'AGATHA CHRISTIE', 'Tabloul', 2);

INSERT INTO book VALUES (5,'Natura cu a ei desavârsire si maretie, în fiecare zi ne zâmbeste oferind cadre mirifice, sublime, pline de culoare, iar noua nu ne râmâne decât ss le observam si sa-i raspundem tot cu un fermecator zâmbet.', 24.00, 'GEORGETA GANEA', 'Ea, Natura', 3);

INSERT INTO book VALUES (6,'Dupa ce a izbutit sa înfrânga monstrii care aproape ca i-au distrus lumea deosebit?, Jacob Portman s-a întors în casa parinteasca, în Florida.', 35.20, 'RANSOM RIGGS', 'Harta zilelor', 4);

INSERT INTO book VALUES (7,'Kyllar decide sa renunte la Calea Umbrelor si sa traiasca o viata linistita alaturi de Elene si de fiica lor adoptiva.', 35.20, 'BRENT WEEKS', 'Marginea umbrei', 5);

INSERT INTO book VALUES (8,'Casa lui Piranesi nu e deloc una obisnuita: are camere infinite, coridoare nesfârsite, peretii sunt ticsiti de mii si mii de statui, toate diferite.', 35.99, 'SUSANNA CLARKE', 'Piranesi', 5);

INSERT INTO book VALUES (9,'Ce legatura ar putea exista între o orfana de razboi, care nu-i nici frumoasa si nici educata, si Academia Militara de la Sinegard, cea mai prestigioasa institutie din Imperiul Nikan?', 35.20, 'R. F. KUANG', 'Razboiul macului', 5);

INSERT INTO book VALUES (10,'No description', 100.00, 'Anonymus', 'Us', 1);

INSERT INTO review VALUES (1, 5, 'O carte excelenta!', to\_date('07-DEC-2020','DD-MON-YYYY'), 4);

INSERT INTO review VALUES (2, 1, 'Nu recomand!', to\_date('01-JAN-2021','DD-MON-YYYY'), 1);

INSERT INTO review VALUES (3, 3, 'O carte mediocra.', to\_date('30-NOV-2020','DD-MON-YYYY'), 5);

INSERT INTO review VALUES (4, 3, 'Nici super, nici meh.', to\_date('06-DEC-2020','DD-MON-YYYY'), 8);

INSERT INTO review VALUES (5, 4, 'Aproape perfecta!', to\_date('24-OCT-2020','DD-MON-YYYY'), 7);

INSERT INTO review VALUES (6, 2, 'Nota 2 cu indulgenta!', to\_date('05-JAN-2021','DD-MON-YYYY'), 7);

INSERT INTO review VALUES (7, 5, 'Super!', to\_date('14-FEB-2020','DD-MON-YYYY'), 2);

INSERT INTO basket VALUES (1,1);

INSERT INTO basket VALUES (2,2);

INSERT INTO basket VALUES (3,3);

INSERT INTO basket VALUES (4,3);

INSERT INTO basket VALUES (5,4);

INSERT INTO basket VALUES (6,5);

INSERT INTO basket VALUES (7,4);

INSERT INTO basket VALUES (8,1);

INSERT INTO basket VALUES (9,1);

INSERT INTO basket VALUES (10,2);

INSERT INTO book\_in\_basket VALUES (1,1,3);

INSERT INTO book\_in\_basket VALUES (1,3,2);

INSERT INTO book\_in\_basket VALUES (3,9,32);

INSERT INTO book\_in\_basket VALUES (7,10,9);

INSERT INTO book\_in\_basket VALUES (2,9,100);

INSERT INTO book\_in\_basket VALUES (3,2,77);

INSERT INTO book\_in\_basket VALUES (6,4,6);

INSERT INTO book\_in\_basket VALUES (5,5,43);

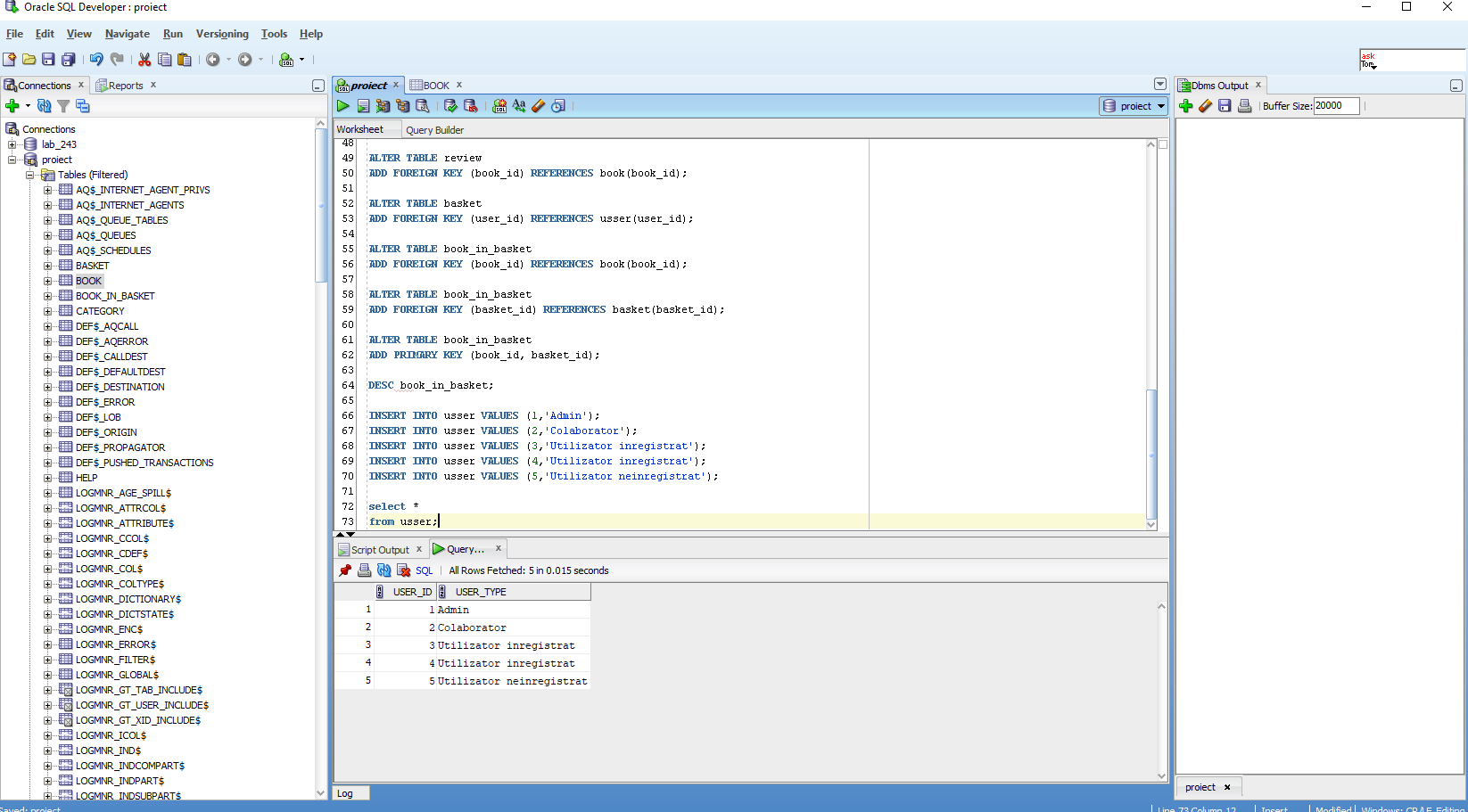
INSERT INTO book\_in\_basket VALUES (4,6,25);

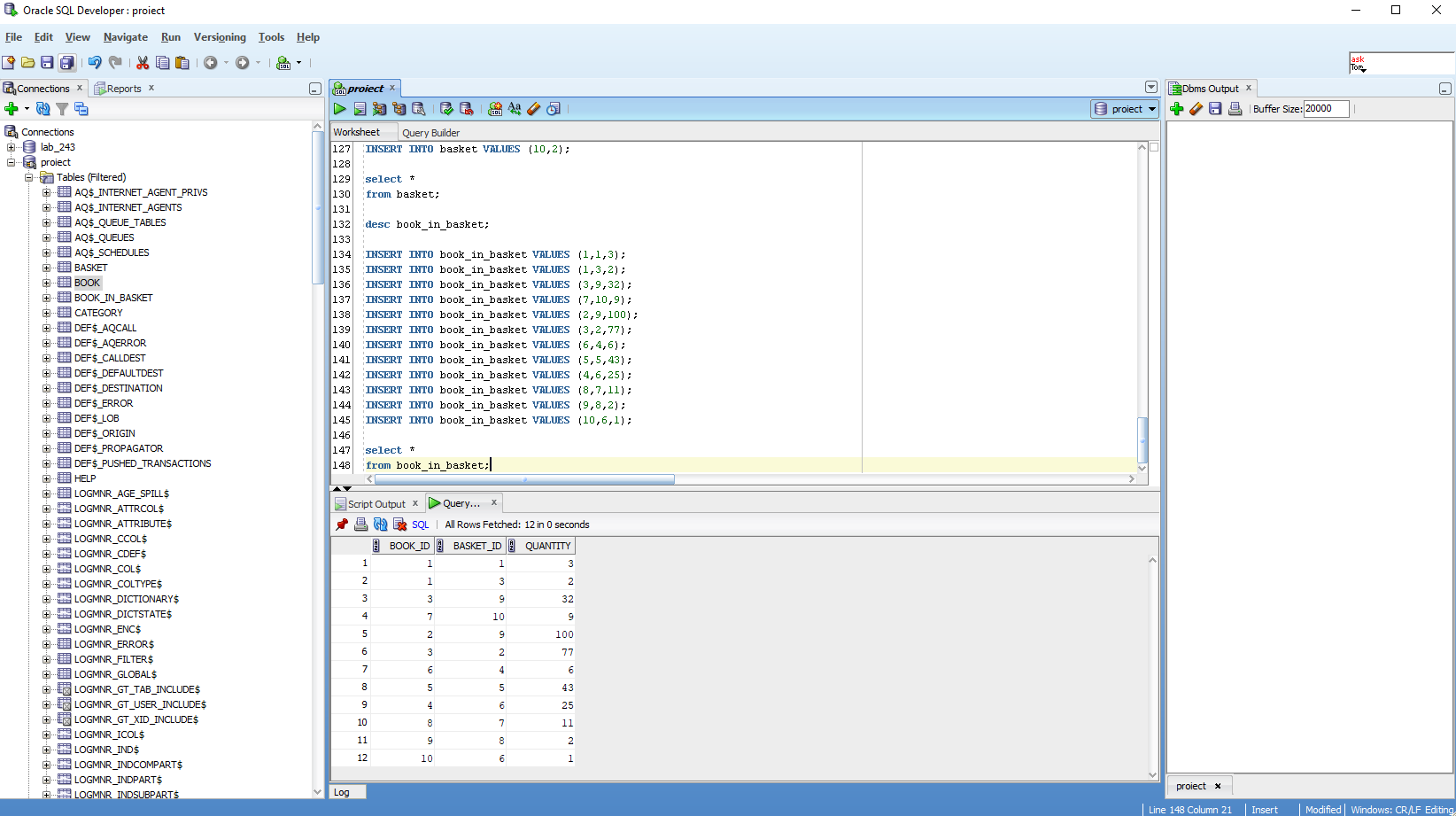
INSERT INTO book\_in\_basket VALUES (8,7,11);

INSERT INTO book\_in\_basket VALUES (9,8,2);

INSERT INTO book\_in\_basket VALUES (10,6,1);

Print-screen cu datele adaugate in prima si ultima tabela:





1. Definirea unui subprogram stocat care sa utilizeze un tip

de coletie studiat.

Enuntul meu: Afisati pe linii diferite top 3 autori pe anul pe anul 2020, luna decembrie, noiembrie si octombrie. Adica 3 autori random care au cartile in magazinul online facut.

Codul:

CREATE OR REPLACE PROCEDURE p6\_paa IS

TYPE Autor IS VARRAY(3) OF VARCHAR2(20);

top Autor := Autor('GEORGETA GANEA', 'RUTA SEPETYS', 'SUSANNA CLARKE');

--am initializat vectorul

PROCEDURE print\_top (luna VARCHAR) IS

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Pentru luna ' || luna || ':');

FOR I IN 1..3 LOOP --afisarea in loop

DBMS\_OUTPUT.PUT\_LINE('LOCUL ' || i || ' -> ' || top(i));

END LOOP;

DBMS\_OUTPUT.PUT\_LINE(' ');

END;

BEGIN

print\_top('Decembrie');--afisez pe decembrie

top := Autor('RUTA SEPETYS', 'R. F. KUANG', 'BRENT WEEKS'); --schimb toate valorile

print\_top('Noiembrie'); --afisez pe noiembrie

top(1) := 'Anonymus'; --schimb 1 valoare

print\_top('Octombrie'); --afisez pe octombrie

END p6\_paa;

/

--verificarea

BEGIN

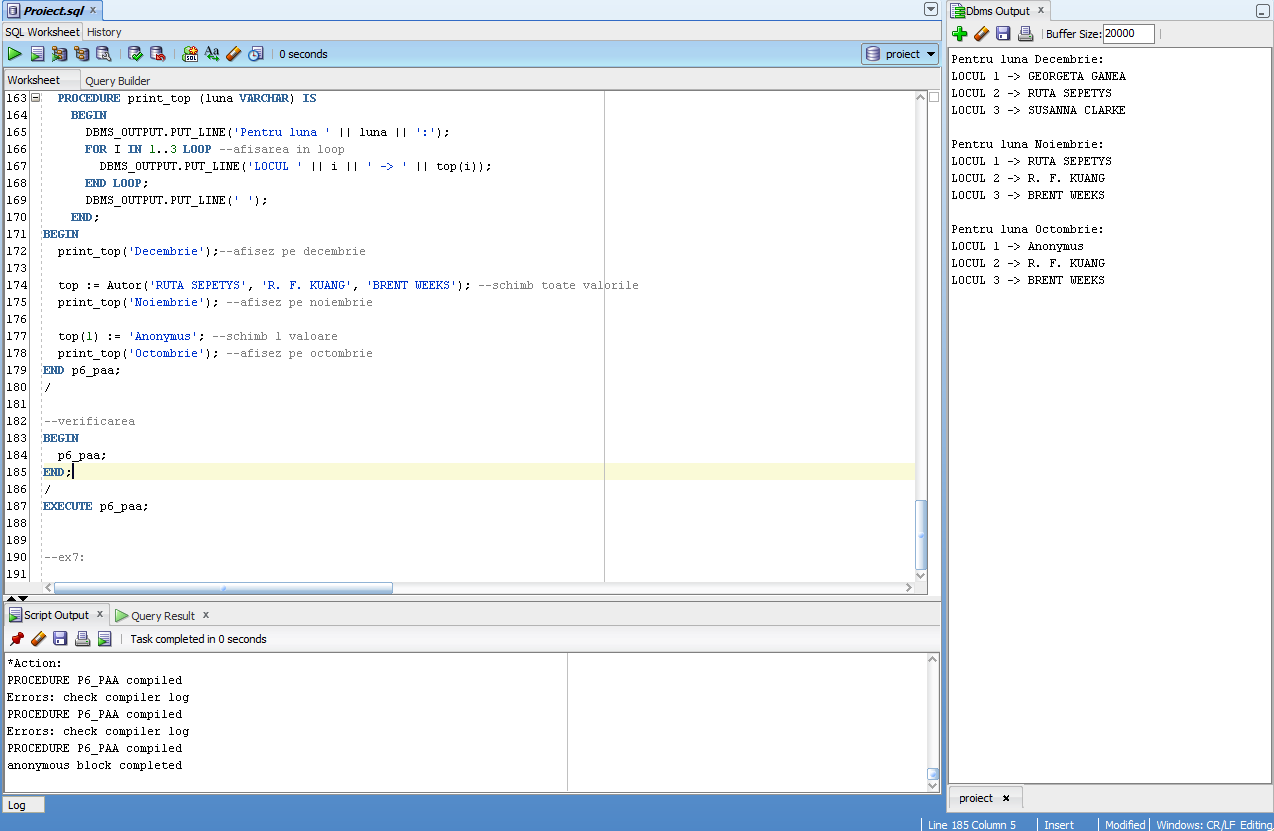
p6\_paa;

END;

/

EXECUTE p6\_paa;

Print-screen:



1. Definirea unui subprogram stocat care sa utilizeze un tip de cursor studiat.

Enuntul meu: Sa se afiseze pretul unei carti, dat fiind titlul acesteia.

Codul:

CREATE OR REPLACE FUNCTION p7\_paa (nume\_carte IN VARCHAR)

RETURN INT IS pret NUMBER;

CURSOR crs1 IS

SELECT price

FROM book

WHERE title = nume\_carte;

BEGIN

OPEN crs1;

FETCH crs1 INTO pret;

IF crs1%notfound THEN pret := 0;

END IF;

CLOSE crs1;

RETURN pret;

END;

/

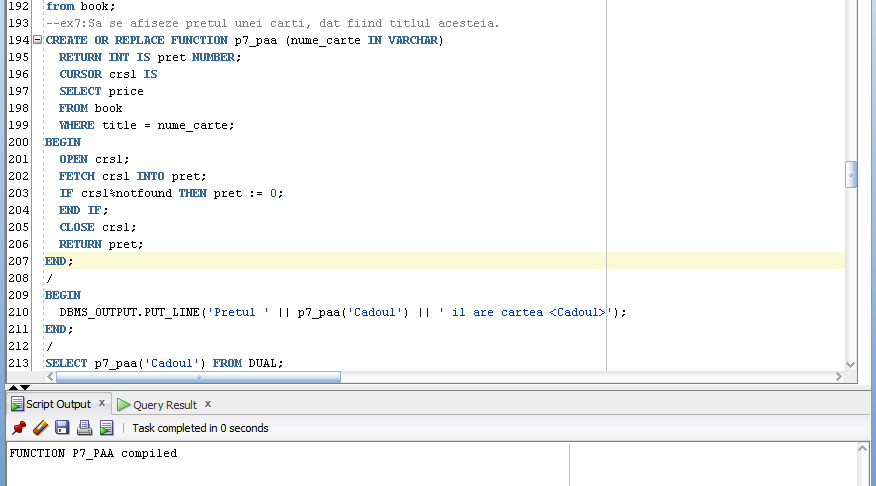
BEGIN

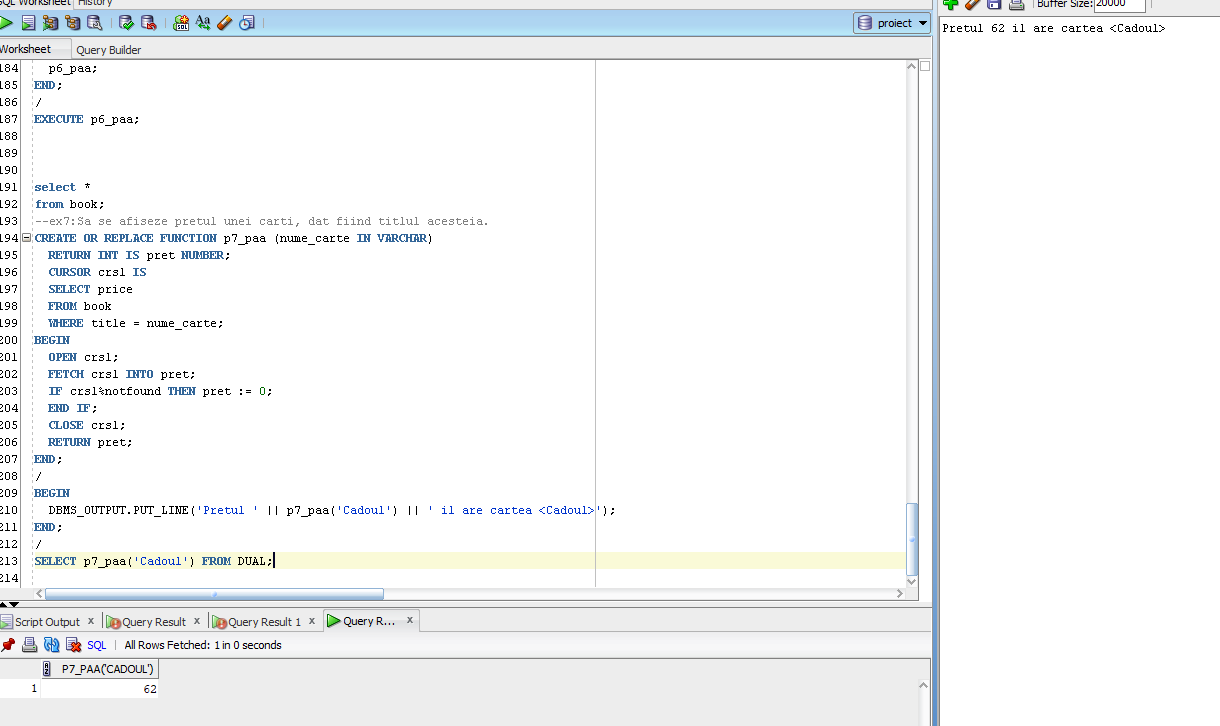
DBMS\_OUTPUT.PUT\_LINE('Pretul ' || p7\_paa('Cadoul') || ' il are cartea <Cadoul>');

END;

/

SELECT p7\_paa('Cadoul') FROM DUAL;

Print-screen: 



1. Definiti un subprogram stocat de tip functie care sa utilizeze 3 dintre tabelele definite. Tratati toate exceptiile.

Enuntul meu: Aflam cate categorii au carti cu rating cel putin 2. (dat fiind numele categoriei ca parametru)

DRAMA = 1; HORROR = 1; POEZIE = 1; AVENTURA = 0; FANTASY = 3; (acestea sunt rezultatele)

Codul:

CREATE OR REPLACE FUNCTION p8\_paa (nume\_categ category.category\_name%TYPE)

RETURN NUMBER IS produse book.book\_id%TYPE;

verif\_existenta category.category\_name%TYPE;

BEGIN

SELECT category\_name

INTO verif\_existenta

FROM category

WHERE category\_name=nume\_categ;

SELECT COUNT(\*)

INTO produse

FROM book

JOIN review on book.book\_id = review.book\_id

WHERE rating >= 2 and book.book\_id IN (SELECT book.book\_id

FROM book

RIGHT OUTER JOIN category on category.category\_id=book.category\_id

WHERE category.category\_name = nume\_categ);

RETURN produse;

EXCEPTION

WHEN no\_data\_found THEN

raise\_application\_error(-20000, 'Categoria nu exista');

WHEN OTHERS THEN

raise\_application\_error(-20001, 'Eroare');

END p8\_paa;

/

DECLARE

nume\_categorie category.category\_name%TYPE := 'Fantasy';

BEGIN

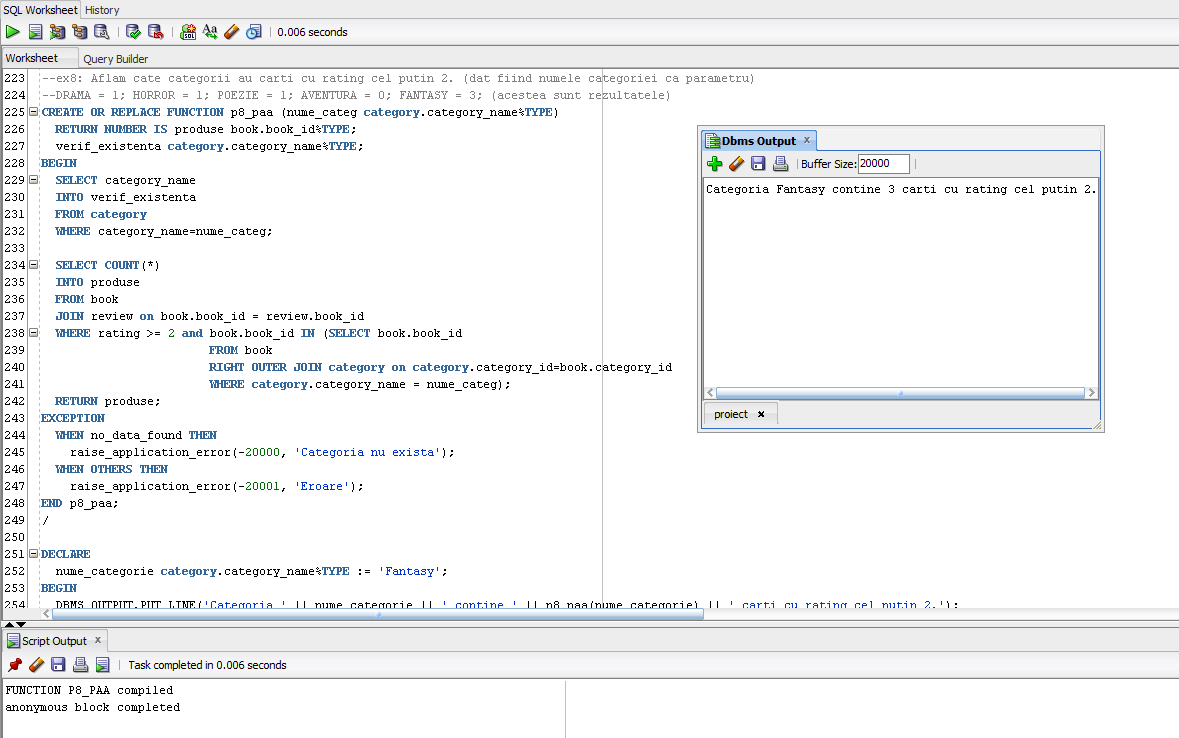
DBMS\_OUTPUT.PUT\_LINE('Categoria ' || nume\_categorie || ' contine ' || p8\_paa(nume\_categorie) || ' carti cu rating cel putin 2.');

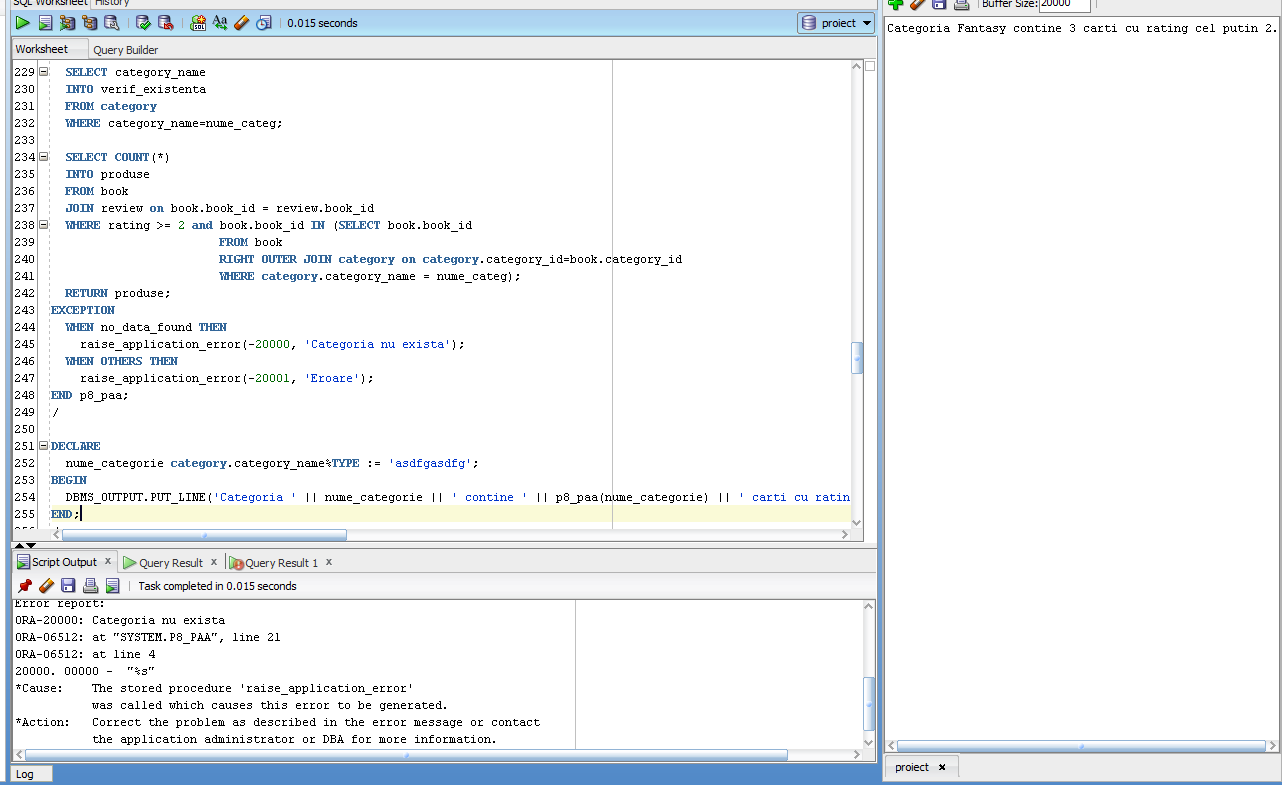
END;

/

SELECT p8\_paa('Horror') FROM DUAL;

Print-screen:





1. Definit un subprogram stocat de tip procedura care sa utilizeze 5 dintre tabelele definite. Tratati toate exceptiile.

Enuntul meu: Userul afla numele categoriei unui produs comandat, daca are o singura comanda.

Codul:

CREATE OR REPLACE PROCEDURE p9\_paa (

c\_id IN usser.user\_id%TYPE)

AS

categ category.category\_name%TYPE;

BEGIN

SELECT category.category\_name

INTO categ

FROM category

WHERE category\_id = (SELECT book.category\_id

FROM book

WHERE book.book\_id = (SELECT book\_in\_basket.book\_id

FROM book\_in\_basket

WHERE book\_in\_basket.basket\_id = (SELECT basket.basket\_id

FROM basket

WHERE basket.user\_id = (SELECT usser.user\_id

FROM usser

WHERE usser.user\_id = c\_id

)

)

)

);

DBMS\_OUTPUT.PUT\_LINE(categ);

EXCEPTION

WHEN no\_data\_found THEN

raise\_application\_error(-20000, 'Nu exista userul/categoria in baza de date');

WHEN OTHERS THEN

raise\_application\_error(-20002, 'Are mai multe cosuri de cumparaturi');

END p9\_paa;

/

--Apelare 9

ALTER PROCEDURE p9\_paa compile;

/

EXEC p9\_paa(5);

/--Totul e ok

/

EXEC p9\_paa(2);

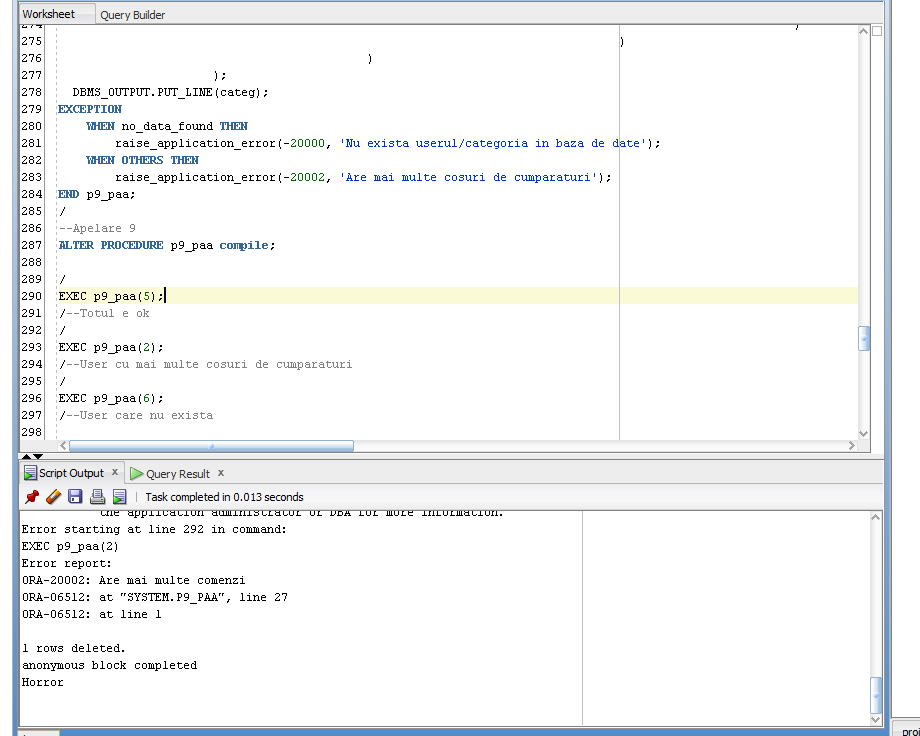
/--User cu mai multe cosuri de cumparaturi

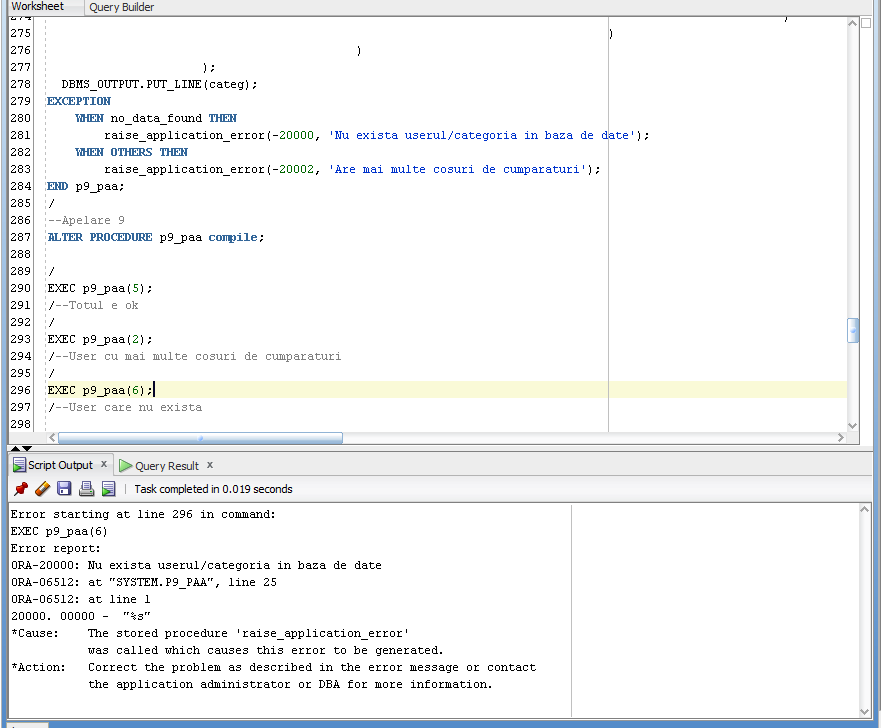
/

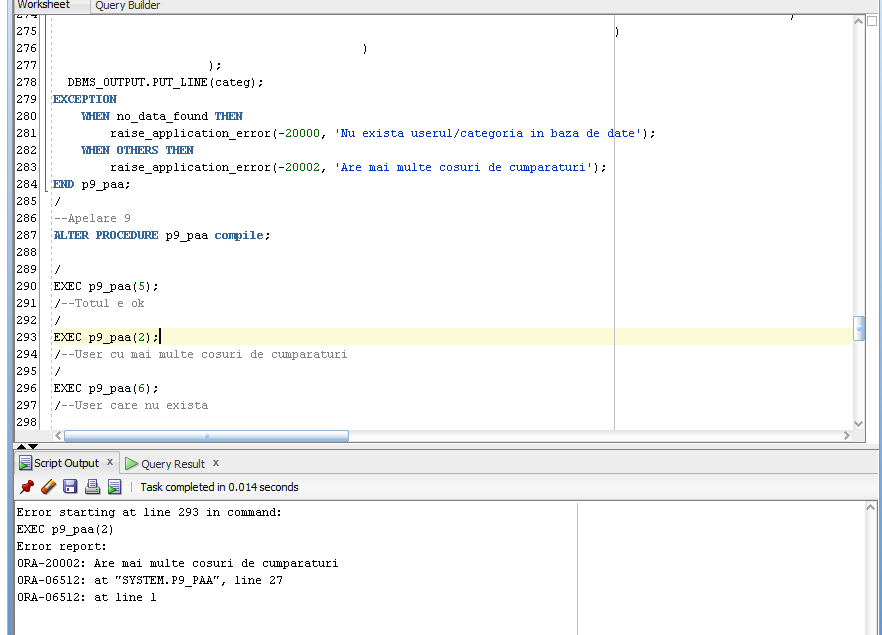
EXEC p9\_paa(6);

/--User care nu exista

Print-screen:







1. Definiti un trigger de tip LMD la nivel de comanda.

Enuntul meu: Inserez book cu permisiune.

Codul:

SET SERVEROUTPUT ON;

CREATE OR REPLACE TRIGGER new\_book

BEFORE INSERT ON book

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Ati adaugat o carte noua in baza de date!');

END;

/

ALTER TRIGGER new\_book ENABLE;

/

ALTER TRIGGER new\_book DISABLE;

/

INSERT INTO book(book\_id, description, price,author,title,category\_id)

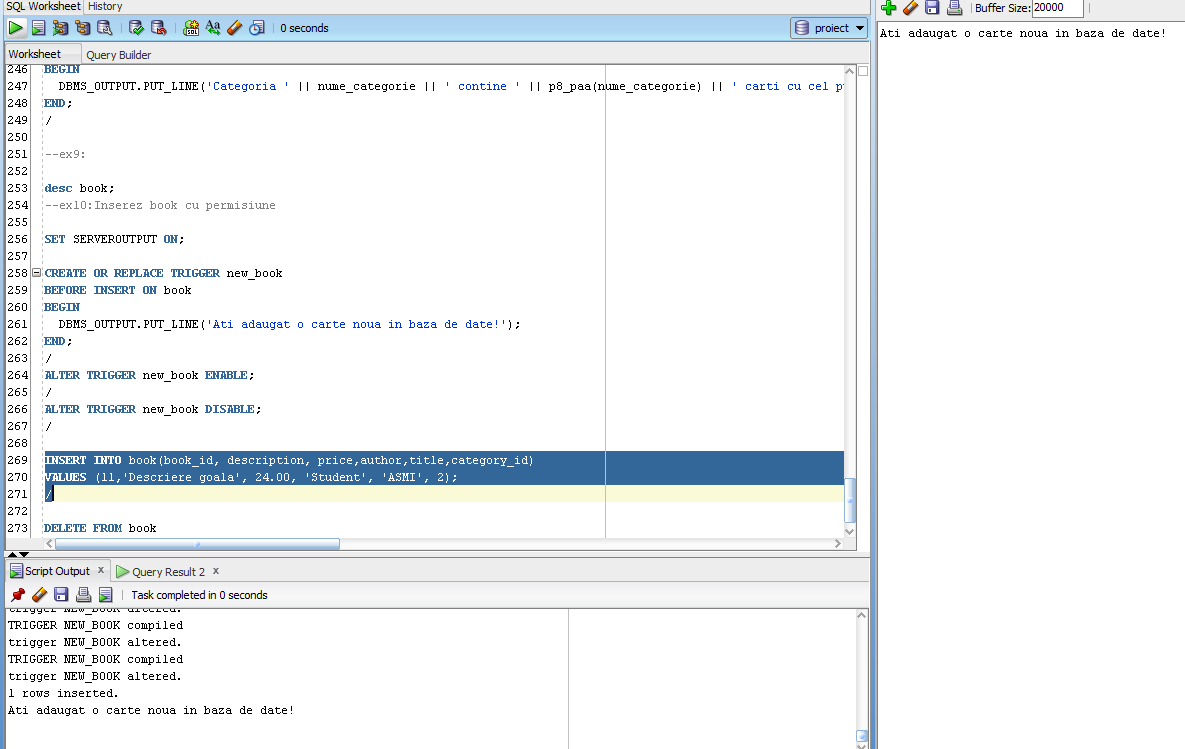
VALUES (11,'Descriere goala', 24.00, 'Student', 'ASMI', 2);

/

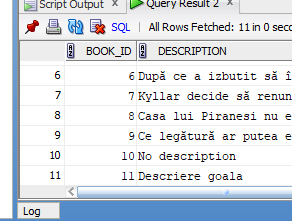
DELETE FROM book

WHERE book\_id = 11;

/

Print-screen:

Dovada ca a mers:



1. Definiti un trigger de tip LMD la nivel de linie.

Enuntul meu: Select o carte dupa book\_id si incerc sa ii modific pretul sa fie peste 500.

Codul:

CREATE OR REPLACE TRIGGER max\_price BEFORE

UPDATE OF price ON book

FOR EACH ROW

BEGIN

IF( :new.price >= 500 ) THEN

raise\_application\_error(-20002, 'Nu se poate creste pretul cartii peste 500');

END IF;

END;

/

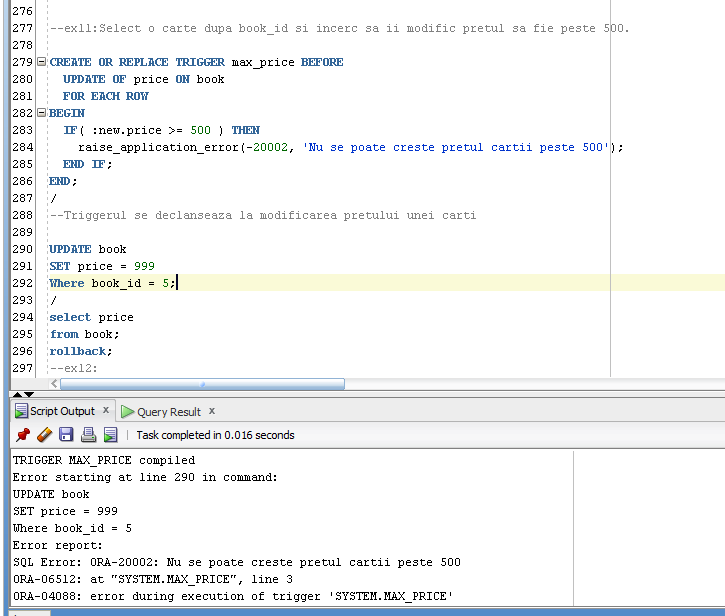
--Triggerul se declanseaza la modificarea pretului unei carti

UPDATE book

SET price = 999

Where book\_id = 5;

Print-screen:



1. Definiti un trigger de tip LDD.

Enuntul meu: Creez o tabela audit\_proiect si un trigger care sa refuze stergerea unei tabele din db. Apoi am dezactivat triggerul si am sters tabela.

Codul:

CREATE OR REPLACE TRIGGER deny

BEFORE DROP ON DATABASE

BEGIN

raise\_application\_error(-20002, 'Nu aveti permisiunea de a sterge tabele!');

END;

/

ALTER TRIGGER deny disable;

--Creez tabela audit\_proiect pt a testa

CREATE TABLE audit\_proiect(

dt DATE,

ceva\_random VARCHAR(256)

);

/

--acum incerc sa sterg si declansez triggerul

DROP TABLE audit\_proiect;

/

Print-screen:

