

# **PROIECT SGBD ORACLE**

## **-GESTIUNEA UNUI LANT DE BIBLIOTECI-**

Fusneica Florentin-Cristian, grupa 241

### **1. DESCRIEREA PROBLEMEI ECONOMICE ALESE**

#### **OBIECTIVUL PROIECTULUI**

Prin prisma acestui proiect, am ales sa prezint demersul unui lant de biblioteci, amplasate in trei locatii. Astfel, proiectul are ca obiectiv principal analiza activitatii bibliotecilor, gestionand relatia dintre angajati, carti si clienti.

#### **DESCRIEREA BAZEI DE DATE**

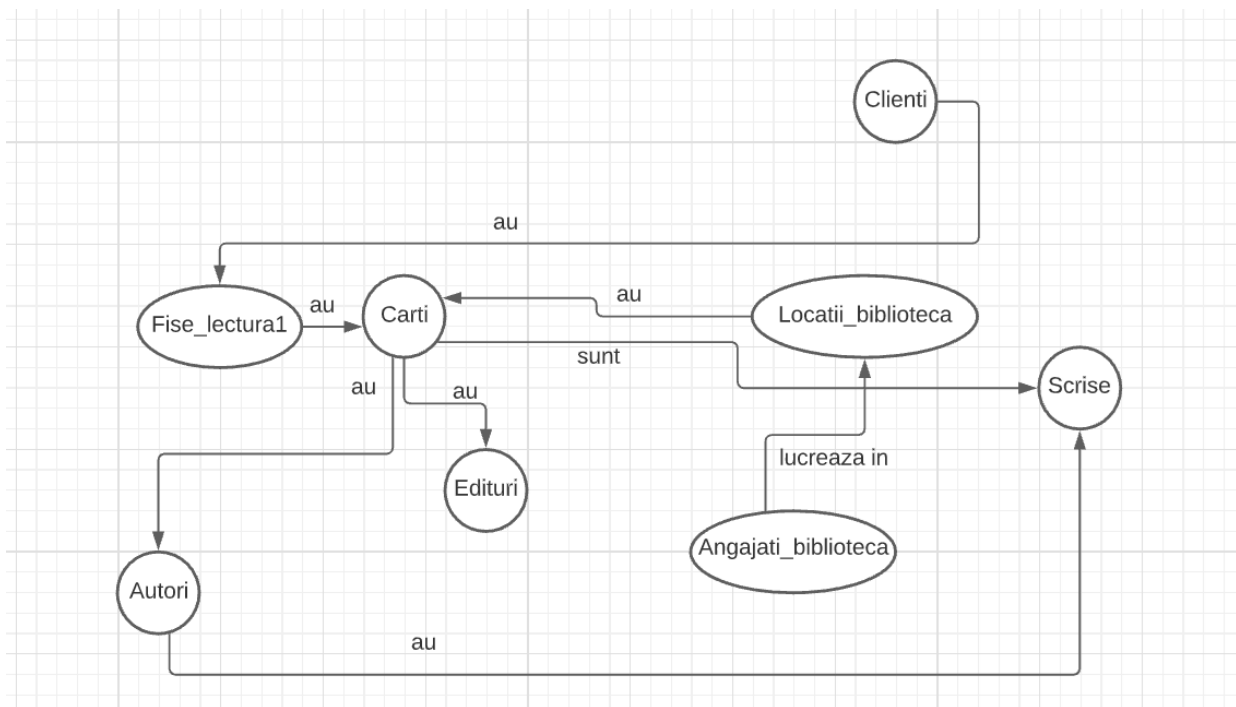
Baza de date consta in opt tabele:

1. CARTI: contine cartile care apar in fiecare biblioteca; tabela ofera informatii despre acestea; fiecare carte are cod, nume,autor si apartine unui domeniu.
2. FISE\_LECTURA1: constituie fisa fiecarui cititor, continand date atat despre cel ce imprumuta cartea, cat si despre carte.
3. LOCATII\_BIBLIOTECA: arata toate locatiile bibliotecilor; unde sunt amplasate acestea
4. CITITORI: tabela contine informatii despre persoana care imprumuta o carte (nume, prenume, adresa,...)
5. ANGAJATI\_BIBLIOTECA:ofera informatii cu legatura la angajatii bibliotecii
6. SCRISE: tabela arata cartile scrise de un anumit autor
7. EDITURI: editurile cartilor de la biblioteca
8. AUTORI: tabela contine informatii referitoare la autorii cartilor (nume, prenume,..)

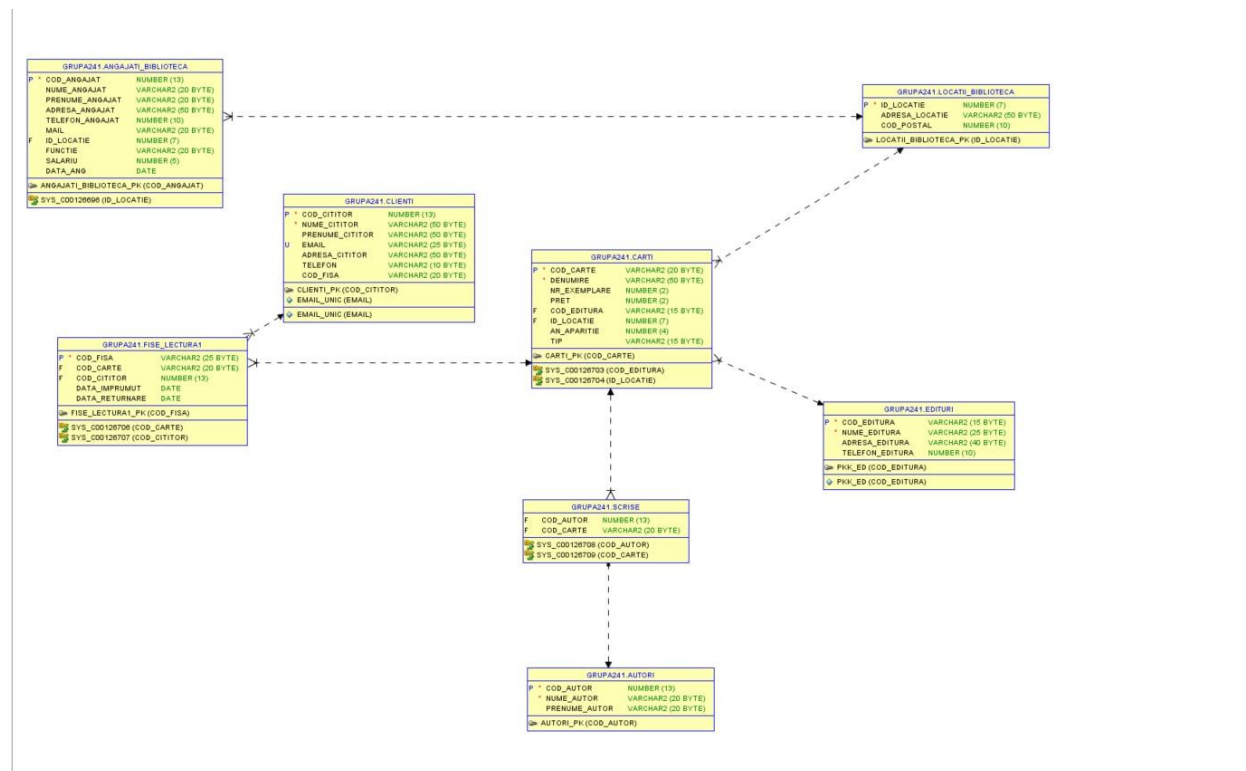
TABELA	ATTRIBUTE	TIP DATA	RESTRICȚII
EDITURI	cod_editura nume_editura adresa_editura telefon_editura	VARCHAR2(25) VARCHAR(15) VARCHAR2(40) NUMBER(10)	PRIMARY KEY  NOT NULL
CARTI	cod_carte  denumire nr_exemplare pret an_aparitie tip cod_editura id_locatie	NUMBER(20)  VARCHAR2(50) NUMBER(2) NUMBER(2) NUMBER(4) VARCHAR2(25) VARCHAR2(25) NUMBER(7)	PRIMARY KEY NOT NULL    FOREIGN KEY FOREIGN KEY
CLIENTI	cod_cititor  nume_cititor prenume_cititor adresa_mail  telefon adresa cod_fisa	NUMBER(13)  VARCHAR2(20) VARCHAR2(20) VARCHAR2(25),  NUMBER(10) VARCHAR2(10) VARCHAR(25)	PRIMARY Key NOT NULL NOT NULL CHECK UNIQUE   FOREIGN KEY

FISE_LECTURA	cod_fisa	VARCHAR(25)	PRIMARY KEY
	cod_carte	NUMBER(20)	FOREIGN KEY
	data_imprumut	DATE	
	data_returnare	DATE	NOT NULL
AUTORI	id_autor	NUMBER(13)	PRIMARY KEY
	nume_autor	VARCHAR2(20)	
	prenume_autor	VARCHAR2(20)	
SCRISE	id_autor	NUMBER(13)	FOREIGN KEYS
	cod_carte	NUMBER(20)	
LOCATII_BIBLIOTECA	id_locatie	NUMBER(7)	PRIMARY KEY
	adresa_locatie	VARCHAR2(50)	
	cod_postal	NUMBER(10)	
ANGAJATI	cod_angajat	NUMBER(13)	PRIMARY KEY
	nume_angajat	VARCHAR2(20)	
	prenume_angajat	VARCHAR2(20)	NOT NULL
	adresa_angajat	VARCHAR2(50)	NOT NULL
	telefon	NUMBER(10)	
	mail_angajat	VARCHAR2(20)	
	salariu	NUMBER(4)	CHECK
	id_locatie		UNIQUE
	functie	NUMBER(3)	CHECK

## 2.DIAGRAMA CONCEPTUALA



### 3.DIAGRAMA ERD



#### 4.IMPLEMENTAREA DIAGramei Conceptuale

- Crearea tabelelor mentionate anterior in care sa fie precizate restrictiile de integritate.

```
CREATE TABLE EDITURI
```

```
(cod_editura VARCHAR2(15) CONSTRAINT pkk_ed PRIMARY KEY,
```

```
nume_editura VARCHAR(25) NOT NULL,
```

```
adresa_editura VARCHAR2(40),
```

```
telefon_editura NUMBER(10));
```

```
ALTER TABLE EDITURI
```

```
ADD CONSTRAINT ED_NUME_NN1 CHECK("NUME_EDITURA" IS NOT NULL);
```

```
CREATE TABLE LOCATII_BIBLIOTECA
```

```
(id_locatie NUMBER(7) PRIMARY KEY,
```

```
adresa_locatie VARCHAR2(50) ,
```

```
cod_postal NUMBER(10));
```

```
CREATE TABLE CARTI
```

```
(cod_carte VARCHAR2(20) primary key,
```

```
denumire VARCHAR2(50) NOT NULL,
```

```
nr_exemplare NUMBER(2),
```

```
pret NUMBER(2),
```

```
cod_editura REFERENCES EDITURI(cod_editura),
```

```
id_locatie references locatii_biblioteca(id_locatie),
```

```
an_aparitiei number(4));
```

```
ALTER TABLE CARTI
```

```
ADD CONSTRAINT CARTE_NUME_N1 CHECK("DENUMIRE" IS NOT NULL);
```

```
ALTER TABLE CARTI ADD TIP VARCHAR(15);
```

CREATE TABLE AUTORI

(cod\_autor NUMBER(13) PRIMARY KEY,  
nume\_autor VARCHAR2(20) ,  
prenume\_autor VARCHAR2(20));

ALTER TABLE AUTORI

ADD CONSTRAINT AUT\_NUME\_N1 CHECK("NUME\_AUTOR" IS NOT NULL);

CREATE TABLE SCRISE

(cod\_autor REFERENCES autori(cod\_autor),  
cod\_carte REFERENCES carti(cod\_carte));

CREATE TABLE ANGAJATI\_BIBLIOTECA

(cod\_angajat NUMBER(13) PRIMARY KEY,  
nume\_angajat VARCHAR2(20) ,  
prenume\_angajat VARCHAR2(20),  
adresa\_angajat VARCHAR2(50),  
telefon\_angajat NUMBER(10),  
mail VARCHAR2(20),  
id\_locatie REFERENCES locatii\_biblioteca(id\_locatie));

alter table angajati\_biblioteca add functie varchar(20);

alter table angajati\_biblioteca add salariu number(5);

alter table angajati\_biblioteca add data\_ang date;

CREATE TABLE CLIENTI

(cod\_cititor NUMBER(13) PRIMARY KEY,

```
nume_cititor VARCHAR2(50) ,
prenume_cititor VARCHAR2(50),
email VARCHAR2(25),
adresa_cititor VARCHAR2(50),
telefon VARCHAR2(10));

ALTER TABLE CLIENTI

ADD CONSTRAINT CIT_NUME_N1 CHECK("NUME_CITITOR" IS NOT NULL);

ALTER TABLE CLIENTI

ADD CONSTRAINT format_email CHECK(email LIKE '%@%.__');

ALTER TABLE CLIENTI

ADD CONSTRAINT email_unic UNIQUE(email);
```

```
CREATE TABLE FISE_LECTURA1
(cod_fisa VARCHAR(25) PRIMARY KEY,
cod_carte REFERENCES CARTI(cod_carte),
cod_cititor references clienti(cod_cititor),
data_imprumut DATE,
data_returnare DATE DEFAULT SYSDATE);
```

```
ALTER TABLE CLIENTI

ADD (COD_FISA VARCHAR2(25)REFERENCES FISE_LECTURA1(COD_FISA));
```

Worksheet Query Builder

```

(cod_autor REFERENCES autori(cod_autor),
cod_carte REFERENCES carti(cod_carte));

CREATE TABLE LOCATII_BIBLIOTECA
(id_locatie NUMBER(7) PRIMARY KEY,
adresa_locatie VARCHAR2(50) ,
cod_postal NUMBER(10),
cod_carte REFERENCES carti(cod_carte));

CREATE TABLE ANGAJATI_BIBLIOTECA
(cod_angajat NUMBER(13) PRIMARY KEY,
nume_angajat VARCHAR2(20) ,
prenume_angajat VARCHAR2(20),
adresa_angajat VARCHAR2(50),
telefon_angajat NUMBER(10),
mail VARCHAR2(20),
id_locatie REFERENCES locatii_biblioteca(id_locatie));

ALTER TABLE CITITORI1
ADD CONSTRAINT format_email CHECK(email LIKE '%@%.__');
ALTER TABLE CITITORI1
ADD CONSTRAINT email_unic UNIQUE(email);

```

Script Output x

Task completed in 0.388 seconds

```

Table CITITORI1 altered.

Table CITITORI dropped.

Table AUTORI created.

Table AUTORI altered.

Table SCRISE created.

Table LOCATII_BIBLIOTECA created.

Table ANGAJATI_BIBLIOTECA created.

```

## 5.ADAUGAREA INFORMATIILOR IN TABELE

```

INSERT INTO AUTORI VALUES(25678543,'Verne','Jules');
INSERT INTO AUTORI VALUES(34747584,'Eliade','Mircea');
INSERT INTO AUTORI VALUES(27583623,'Eminescu','Mihai');
INSERT INTO AUTORI VALUES(37353327,'Cartarescu','Mircea');
INSERT INTO AUTORI VALUES(37387637,'Creanga','Ion');
INSERT INTO AUTORI VALUES(35363627,'Nabokov','Vladimir');
INSERT INTO AUTORI VALUES(37459027,'Proust','Marcel');
INSERT INTO AUTORI VALUES(30593827,'Joyce','James');
INSERT INTO AUTORI VALUES(30483927,'Woolf','Virginia');

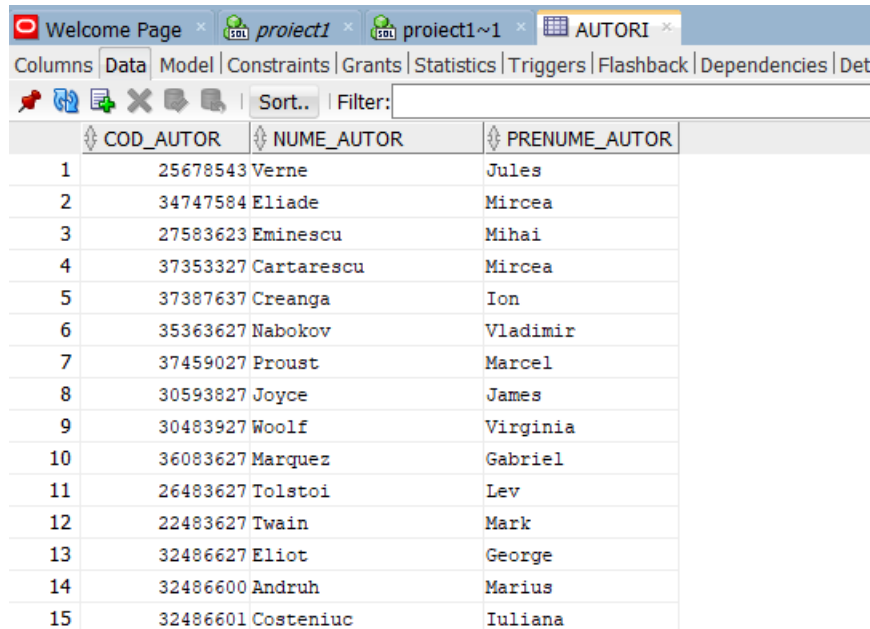
```



```

INSERT INTO AUTORI VALUES(36083627,'Marquez','Gabriel');
INSERT INTO AUTORI VALUES(26483627,'Tolstoi','Lev');
INSERT INTO AUTORI VALUES(22483627,'Twain','Mark');
INSERT INTO AUTORI VALUES(32486627,'Eliot','George');
INSERT INTO AUTORI VALUES(32486600,'Andruh','Marius');
INSERT INTO AUTORI VALUES(32486601,'Costeniuc','Iuliana');

```

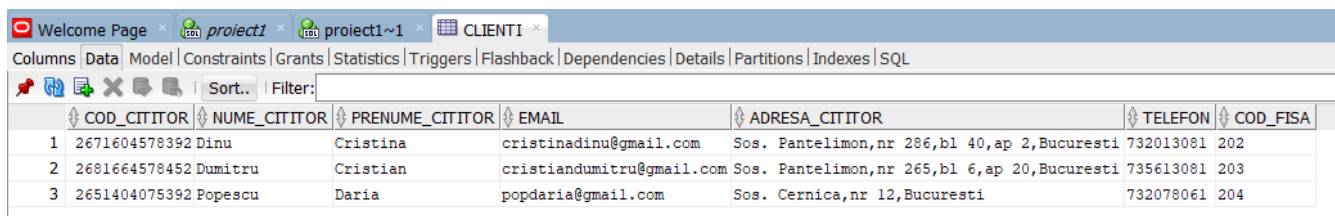


	COD_AUTOR	NUME_AUTOR	PRENUME_AUTOR
1	25678543	Verne	Jules
2	34747584	Eliade	Mircea
3	27583623	Eminescu	Mihai
4	37353327	Cartarescu	Mircea
5	37387637	Creanga	Ion
6	35363627	Nabokov	Vladimir
7	37459027	Proust	Marcel
8	30593827	Joyce	James
9	30483927	Woolf	Virginia
10	36083627	Marquez	Gabriel
11	26483627	Tolstoi	Lev
12	22483627	Twain	Mark
13	32486627	Eliot	George
14	32486600	Andruh	Marius
15	32486601	Costeniuc	Iuliana

```

INSERT INTO CLIENTI VALUES(2671604578392,'Dinu','Cristina','cristinadinu@gmail.com','Sos.
Pantelimon,nr 286,bl 40,ap 2,Bucuresti',0732013081,202);
INSERT INTO CLIENTI
VALUES(2681664578452,'Dumitru','Cristian','cristiandumitru@gmail.com','Sos. Pantelimon,nr 265,bl
6,ap 20,Bucuresti',0735613081,203);
INSERT INTO CLIENTI VALUES(2651404075392,'Popescu','Daria','popdaria@gmail.com','Sos.
Cernica,nr 12,Bucuresti',0732078061,204);

```



	COD_CITITOR	NUME_CITITOR	PRENUME_CITITOR	EMAIL	ADRESA_CITITOR	TELEFON	COD_FISA
1	2671604578392	Dinu	Cristina	cristinadinu@gmail.com	Sos. Pantelimon,nr 286,bl 40,ap 2,Bucuresti	732013081	202
2	2681664578452	Dumitru	Cristian	cristiandumitru@gmail.com	Sos. Pantelimon,nr 265,bl 6,ap 20,Bucuresti	735613081	203
3	2651404075392	Popescu	Daria	popdaria@gmail.com	Sos. Cernica,nr 12,Bucuresti	732078061	204

```

INSERT INTO EDITURI VALUES('101','Accent','Str. Dimitrie Cantemir,nr. 20,Bucuresti',021745634);
INSERT INTO EDITURI VALUES('102','ALL','Str. Alea Lotrului,nr. 25,Bucuresti',031745658);
INSERT INTO EDITURI VALUES('103','Art','Str. Alea Alexandru,nr. 80,Bucuresti',021485694);
INSERT INTO EDITURI VALUES('104','Egmont','Str. Alea Pajurei,nr. 33,Bucuresti',021215634);
INSERT INTO EDITURI VALUES('105','IDEA','Str. Alea Teisani,nr. 10,Bucuresti',021747024);

```

```

INSERT INTO EDITURI VALUES('106','Natiunea','Str.Aleea Faurei,nr. 30,Bucuresti',021745710);
INSERT INTO EDITURI VALUES('107','Adevarul Holding','Str. Priporului,nr.
108,Bucuresti',021745300);
INSERT INTO EDITURI VALUES('108','Salco','Str.Matelotilor,nr. 70,Bucuresti',021745610);
INSERT INTO EDITURI VALUES('109','Unirea','Str. Somesul Rece,nr. 100,Bucuresti',021740048);
INSERT INTO EDITURI VALUES('110','Valdo','Str. Privighetorilor,nr. 29,Bucuresti',021785004);

```

	COD_EDITURA	NUME_EDITURA	ADRESA_EDITURA	TELEFON_EDITURA
1	101	Accent	Str. Dimitrie Cantemir,nr. 20,Bucuresti	21745634
2	102	ALL	Str. Aleea Lotrului,nr. 25,Bucuresti	31745658
3	103	Art	Str. Aleea Alexandru,nr. 80,Bucuresti	21485694
4	104	Egmont	Str. Aleea Pajurei,nr. 33,Bucuresti	21215634
5	105	IDEA	Str. Aleea Teisani,nr. 10,Bucuresti	21747024
6	106	Natiunea	Str.Aleea Faurei,nr. 30,Bucuresti	21745710
7	107	Adevarul Holding	Str. Priporului,nr. 108,Bucuresti	21745300
8	108	Salco	Str.Matelotilor,nr. 70,Bucuresti	21745610
9	109	Unirea	Str. Somesul Rece,nr. 100,Bucuresti	21740048
10	110	Valdo	Str. Privighetorilor,nr. 29,Bucuresti	21785004

```

INSERT INTO CARTI VALUES('ISBN 120-4590','Ocolul Pamantului in 80 de
zile',3,23,107,1873,'FICTIUNE');
INSERT INTO CARTI VALUES('ISBN 103-4570','Copiii Capitanului
Grant',5,28,107,186,'FICTIUNE');
INSERT INTO CARTI VALUES('ISBN 193-4767','Amintiri din
copilarie',2,30,109,1959,'BIOGRAFIE');
INSERT INTO CARTI VALUES('ISBN 193-0567','Maitreyi',5,26,110,1933,'FICTIUNE');
INSERT INTO CARTI VALUES('ISBN 123-9577','Zece Negri Mititei',3,25,105,1939,'POLITIST');
INSERT INTO CARTI VALUES('ISBN 129-4568','Crima din Orient
Express',5,24,105,1934,'POLITIST');
INSERT INTO CARTI VALUES('ISBN 125-9067','Orbitor',2,30,102,2007,'ISTORIC');
INSERT INTO CARTI VALUES('ISBN 123-4576','Nostalgia',6,36,103,1989,'FICTIUNE');
INSERT INTO CARTI VALUES('ISBN 120-4997','Lolita',4,21,108,1955,'FICTIUNE');
INSERT INTO CARTI VALUES('ISBN 123-4550','Razboi si pace',5,36,104,1867,'ISTORIC');
INSERT INTO CARTI VALUES('ISBN 122-4695','Invierea',5,31,109,1899,'FICTIUNE');
INSERT INTO CARTI VALUES('ISBN 102-4986','Sanctuar',5,34,107,1931,'POLITIST');
INSERT INTO CARTI VALUES('ISBN 123-4570','Middlemarch',5,36,109,1871,'ISTORIC');
INSERT INTO CARTI VALUES('ISBN 123-4500','Manual chimie clasa a8-
a',2,40,107,2011,'MANUAL');

```

Welcome Page x proiect1 x proiect1~1 x CARTI x							
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL							
Sort.. Filter:							
	COD_CARTE	DENUMIRE	NR_EXEMPLARE	PRET	COD_EDITURA	AN_APARITIE	TIP
1	ISBN 120-4590	Ocolul Pamantului in 80 de zile	0	28 107		1873	FICTIUNE
2	ISBN 103-4570	Copiii Capitanului Grant	5	28 107		186	FICTIUNE
3	ISBN 193-4767	Amintiri din copilarie	2	30 109		1959	BIOGRAFIE
4	ISBN 193-0567	Maitreyi	5	26 110		1933	FICTIUNE
5	ISBN 123-9577	Zece Negri Mititei	3	25 105		1939	POLITIST
6	ISBN 129-4568	Crima din Orient Express	5	24 105		1934	POLITIST
7	ISBN 125-9067	Orbitor	2	30 102		2007	ISTORIC
8	ISBN 123-4576	Nostalgia	6	36 103		1989	FICTIUNE
9	ISBN 120-4997	Lolita	4	21 108		1955	FICTIUNE
10	ISBN 123-4550	Razboi si pace	5	36 104		1867	ISTORIC
11	ISBN 122-4695	Invierea	5	31 109		1899	FICTIUNE
12	ISBN 102-4986	Sanctuar	5	34 107		1931	POLITIST
13	ISBN 123-4570	Middlemarch	5	36 109		1871	ISTORIC
14	ISBN 123-4500	Manual chimie clasa a8-a	2	40 107		2011	MANUAL

```
INSERT INTO LOCATII_BIBLIOTECA VALUES(500,'Str. Pajurii,nr. 280,Bucuresti',656325);
INSERT INTO LOCATII_BIBLIOTECA VALUES(502,'Str. Uverturii,nr. 27,Bucuresti',656236);
INSERT INTO LOCATII_BIBLIOTECA VALUES(504,'Str. Vatra Dornei,nr. 26,Bucuresti',656200);
```

Welcome Page x proiect1 x proiect1~1 x LOCATII_BIBLIOTECA x			
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Detail			
Sort.. Filter:			
	ID_LOCATIE	ADRESA_LOCATIE	COD_POSTAL
1	500	Str. Pajurii,nr. 280,Bucuresti	656325
2	502	Str. Uverturii,nr. 27,Bucuresti	656236
3	504	Str. Vatra Dornei,nr. 26,Bucuresti	656200

```
INSERT INTO ANGAJATI_BIBLIOTECA VALUES(002,'Pop','Andreea','Str. Aleea Stanila
23,Bucuresti',0736013028,'poopand@gmail.com',500,'ADMINISTRATOR',2400,to_date('08-09-
2016','dd-mm-yyyy'));
INSERT INTO ANGAJATI_BIBLIOTECA VALUES(003,'Mihai','Andrei','Blv.Iuliu Maniu,nr.
23,bl.41C,ap.4',0722623902,'mihaindrei@gmail.com',502,'AJUTOR
BIBLIOTECAR',2000,to_date('01-01-2017','dd-mm-yyyy'));
INSERT INTO ANGAJATI_BIBLIOTECA VALUES(004,'Chiritescu','Silviu','Str. Mihai Viteazul,nr.
2A,Bucuresti',0766256202,'silviuch@gmail.com',504,'ADMINISTRATOR',3400,to_date('04-05-
2015','dd-mm-yyyy'));
INSERT INTO ANGAJATI_BIBLIOTECA VALUES(005,'Agapie','Miruna','Str. Mihail Mirinescu
nr.26, Bucuresti',0733612025,'agapiemir@gmail.com',500,'BIBLIOTECARA',1500,to_date('08-11-
2018','dd-mm-yyyy'));
INSERT INTO ANGAJATI_BIBLIOTECA VALUES(006,'Apostol','Miruna','Str. Mihai Viteazul nr.6,
Bucuresti',07336574025,'apostolmir@gmail.com',502,'BIBLIOTECARA',1800,to_date('08-01-
2018','dd-mm-yyyy'));
```

Welcome Page   proiect1   proiect1~1   ANGAJATI_BIBLIOTECA									
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies   Details   Partitions   Indexes   SQL									
Sort:   Filter:									
COD_ANGAJAT	NUME_ANGAJAT	PRENUME_ANGAJAT	ADRESA_ANGAJAT	TELEFON_ANGAJAT	MAIL	ID_LOCATIE	FUNCTIE	SALARIU	DATA_ANG
1	2 Pop	Andreea	Str. Aleea Stanila 23,Bucuresti	736013028	poopand@gmail.com	500	ADMINISTRATOR	2400	08-SEP-16
2	3 Mihai	Andrei	Blv.Tuliu Maniu,nr. 23,bl.41C,ap.4	722623902	mihaandrei@gmail.com	502	AJUTOR BIBLIOTECAR	2200	01-JAN-17
3	5 Agapie	Miruna	Str. Mihail Mirinescu nr.26, Bucuresti	733612025	agapiemir@gmail.com	500	BIBLIOTECARA	3800	08-NOV-18
4	6 Apostol	Miruna	Str. Mihai Viteazul nr.6, Bucuresti	7336574025	apostolmir@gmail.com	502	BIBLIOTECARA	3800	08-JAN-18
5	4 Chiritescu	Silviu	Str. Mihai Viteazul,nr. 2A,Bucuresti	766256202	silviuch@gmail.com	504	ADMINISTRATOR	3400	04-MAY-15

```

INSERT INTO SCRISE VALUES(25678543,'ISBN 120-4590');
INSERT INTO SCRISE VALUES(25678543,'ISBN 123-4567');
INSERT INTO SCRISE VALUES(25678543,'ISBN 103-4570');
INSERT INTO SCRISE VALUES(34747584,'ISBN 193-0567');
INSERT INTO SCRISE VALUES(37353327,'ISBN 125-9067');
INSERT INTO SCRISE VALUES(37353327,'ISBN 123-4576');
INSERT INTO SCRISE VALUES(35363627,'ISBN 120-4997');
INSERT INTO SCRISE VALUES(35363627,'ISBN 123-4907');
INSERT INTO SCRISE VALUES(32483657,'ISBN 123-4907');
INSERT INTO SCRISE VALUES(36483627,'ISBN 196-4789');
INSERT INTO SCRISE VALUES(32883627,'ISBN 120-4570');
INSERT INTO SCRISE VALUES(32486600,'ISBN 123-4500');
INSERT INTO SCRISE VALUES(32486601,'ISBN 123-4500');

```

Welcome Page   proiect1   proiect1~1   SCRISE			
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies			
Sort:   Filter:			
COD_AUTOR	COD_CARTE		
1	25678543 ISBN 120-4590		
2	25678543 ISBN 103-4570		
3	34747584 ISBN 193-0567		
4	37353327 ISBN 125-9067		
5	37353327 ISBN 123-4576		
6	35363627 ISBN 120-4997		
7	26483627 ISBN 123-4550		
8	26483627 ISBN 122-4695		
9	37387637 ISBN 193-4767		
10	32486600 ISBN 123-4500		
11	32486601 ISBN 123-4500		

```

INSERT INTO FISE_LECTURA1 VALUES('202','ISBN 122-4695',TO_DATE('12-03-2021','DD-MM-YYYY'),TO_DATE('18-03-2021','DD-MM-YYYY'));
INSERT INTO FISE_LECTURA1 VALUES('203','ISBN 123-4500',TO_DATE('12-04-2021','DD-MM-YYYY'),TO_DATE('18-04-2021','DD-MM-YYYY'));
INSERT INTO FISE_LECTURA1 VALUES('204','ISBN 102-4986',TO_DATE('12-05-2021','DD-MM-YYYY'),TO_DATE('18-05-2021','DD-MM-YYYY'));

```

Welcome Page × project1 × project1~1 × FISE_LLECTURA1 ×				
Columns   Data   Model   Constraints   Grants   Statistics   Triggers   Flashback   Dependencies				
Sort.. Filter:				
	COD_FISA	COD_CARTE	DATA_IMPRUMUT	DATA_RETURNARE
1	203	ISBN 123-4500	12-APR-21	18-APR-21
2	204	ISBN 102-4986	12-MAY-21	18-MAY-21
3	202	ISBN 122-4695	12-MAR-21	18-MAR-21

## 6.SUBPROGRAM CU INDEX-BY TABLE

- Creati un subprogram de tip procedura care sa afiseze toate editurile din cadrul bibliotecilor si sa calculeze numarul lor.

```

set serveroutput on
create or replace procedure nr_ed
as
TYPE ed_tab IS TABLE OF edituri%ROWTYPE INDEX BY
PLS_INTEGER;
v_ed ed_tab;
BEGIN
DBMS_OUTPUT.PUT_LINE('Edituri:');
FOR i IN 101..110 LOOP
SELECT * INTO v_ed(i) FROM edituri WHERE
cod_editura=i;
END LOOP;
FOR i IN v_ed.FIRST..v_ed.LAST LOOP
DBMS_OUTPUT.PUT_LINE(v_ed(i).nume_editura);
END LOOP;
DBMS_OUTPUT.PUT_LINE('Total edituri: '||v_ed.COUNT);
END;
/
execute nr_ed;

```

7.

The screenshot shows the Oracle SQL Developer interface. The top pane displays a PL/SQL procedure named `nr_ed` in the 'Query Builder' tab. The procedure is designed to collect data from the `edituri` table. It includes a loop to iterate through rows and another loop to iterate through the columns of each row, outputting the column names and the total count of rows. The bottom pane shows the 'Script Output' and 'Query Result' tabs. The 'Script Output' tab indicates that the procedure was compiled successfully. The 'Query Result' tab shows the output of the procedure, which lists the names of 10 publishers and the total count of books.

```
--colectie--
set serveroutput on
create or replace procedure nr_ed
as
TYPE ed_tab IS TABLE OF edituri%ROWTYPE INDEX BY
PLS_INTEGER;
v_ed ed_tab;
BEGIN
DBMS_OUTPUT.PUT_LINE('Edituri:');
FOR i IN 101..110 LOOP
SELECT * INTO v_ed(i) FROM edituri WHERE
cod_editura=i;
END LOOP;
FOR i IN v_ed.FIRST..v_ed.LAST LOOP
DBMS_OUTPUT.PUT_LINE(v_ed(i).nume_editura);
END LOOP;
DBMS_OUTPUT.PUT_LINE('Total edituri: '||v_ed.COUNT);
END;
/
execute nr_ed;
```

Script Output x Query Result x  
Task completed in 0.329 seconds

Procedure NR\_ED compiled

Edituri:  
Accent  
ALL  
Art  
Egmont  
IDEA  
Natiunea  
Adevarul Holding  
Salco  
Unirea  
Valdo  
Total edituri: 10

PL/SQL procedure successfully completed.

## SUBPROGRAM CARE SA UTILIZEZE UN CURSOR

→ Creati un subprogram de tip procedura care sa afiseze codul si denumirea cartilor de tipul *ISTORIC*, utilizand un cursor explicit.

```
CREATE OR REPLACE PROCEDURE CARTI_IST
AS
```

```
cursor c is select cod_carte,denumire from carti
where tip='ISTORIC';
```

```
v_cod carti.cod_carte%type;
```

```
v_den carti.denumire%type;
```

```

begin
open c;
loop
fetch c into v_cod,v_den;
exit when c%notfound;
dbms_output.put_line('Denumire: ' ||v_den||';Cod: '||v_cod);
end loop;
close c;
end;
/
execute carti_ist;

```

The screenshot shows the Oracle SQL Developer interface. The top window is titled 'project1' and contains a 'Query Builder' tab. The SQL text in the editor is as follows:

```

execute MARIRE_SALARIU(10,003);

CREATE OR REPLACE PROCEDURE CARTI_IST
AS
cursor c is select cod_carte,denumire from carti
where tip='ISTORIC';
v_cod carti.cod_carte%type;
v_den carti.denumire%type;
begin
open c;
loop
fetch c into v_cod,v_den;
exit when c%notfound;
dbms_output.put_line('Denumire: ' ||v_den||';Cod: '||v_cod);
end loop;
close c;
end;
/
execute carti_ist;

```

Below the editor, the 'Script Output' window shows the results of the execution:

```

Task completed in 0.256 seconds

Procedure CARTI_IST compiled

Procedure CARTI_IST compiled

Denumire: Orbitor;Cod: ISBN 125-9067
Denumire: Razboi si pace;Cod: ISBN 123-4550
Denumire: Middlemarch;Cod: ISBN 123-4570

PL/SQL procedure successfully completed.

```

## 8.SUBPROGRAM STOCAT DE TIP FUNCTIE CARE SA UTILIZEZE 3 TABELE

- Creati o functie care sa afiseze codul, denumirea, tipul, informatii despre autor si valoarea totala (pret\*nr\_exemplare) pentru o carte data prin parametru.

```
set serveroutput on
create or replace function val_tot(nrc in varchar) return number
as
ex exception;
nre number;
v_codc scrie.cod_carte%type;
v_coda scrie.cod_autor%type;
v_den carti.denumire%type;
v_aut autori.num_e_autor%type;
v_autp autori.prenume_autor%type;
v_tip carti.tip%type;
begin
select
s.cod_carte,c.denumire,s.cod_autor,a.num_e_autor,a.prenume_autor,c.tip,sum(c.pret*c.nr_exemplare)
into v_codc,v_den,v_coda,v_aut,v_autp,v_tip,nre from carti c,autori a, scrie s where
s.cod_carte=c.cod_carte and s.cod_autor=a.cod_autor and c.cod_carte=nrc
group by s.cod_carte,c.denumire,s.cod_autor,a.num_e_autor,a.prenume_autor,c.tip ;
dbms_output.put_line('Carte:'||v_den||';valoare totala'||nre||';nume autor:'||v_aut||';prenume
autor:'||v_autp);
return nre;
if(nre=0)
then raise ex;
end if;
exception
when ex then dbms_output.put_line ('Nu sunt exemplare');
end;
/
```

Apel1:

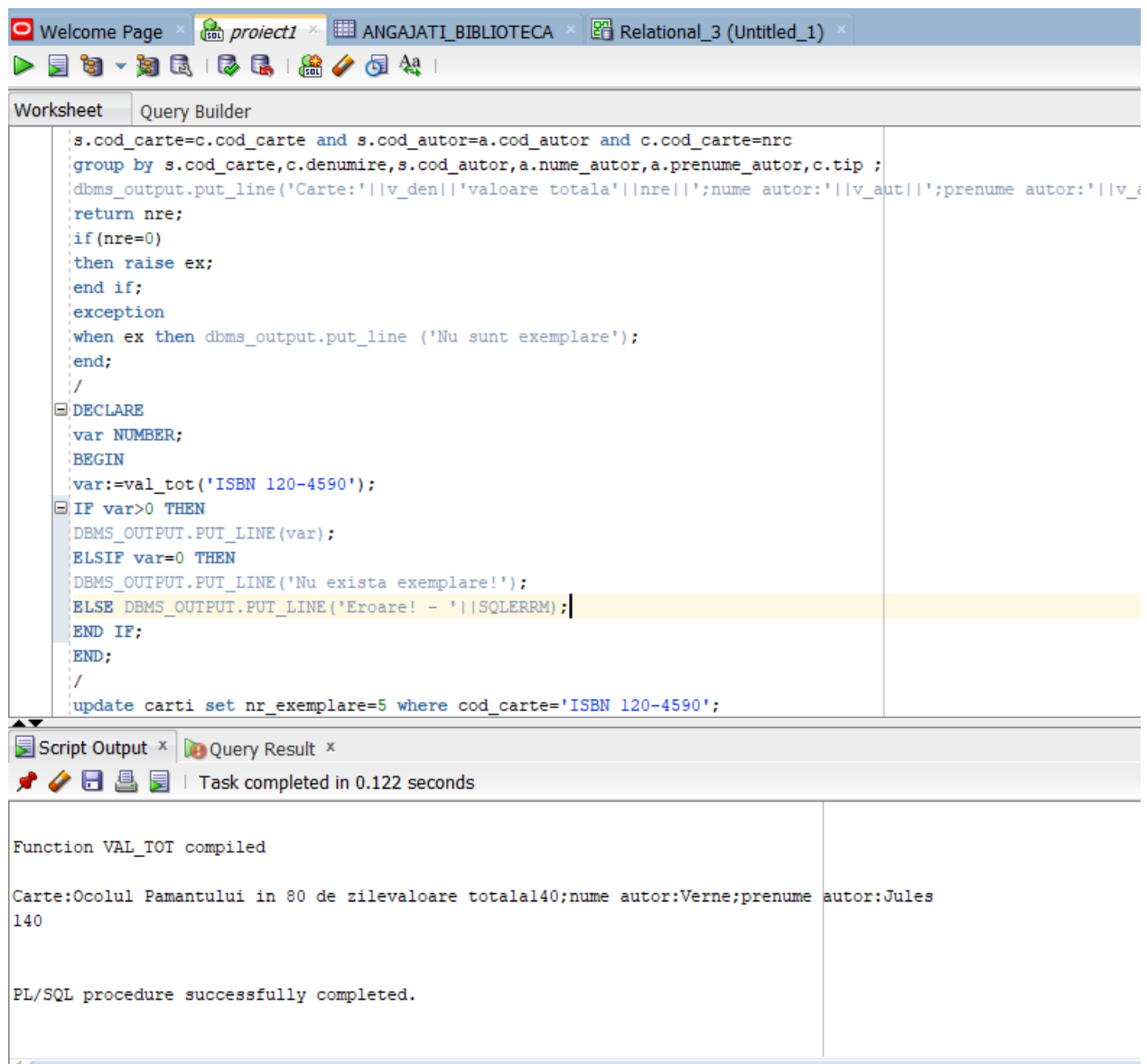
```
DECLARE
var NUMBER;
BEGIN
var:=val_tot('ISBN 120-4590');
IF var>0 THEN
DBMS_OUTPUT.PUT_LINE(var);
ELSIF var=0 THEN
DBMS_OUTPUT.PUT_LINE('Nu exista exemplare!');
```



```

ELSE DBMS_OUTPUT.PUT_LINE('Eroare! - '||SQLERRM);
END IF;
END;
/

```



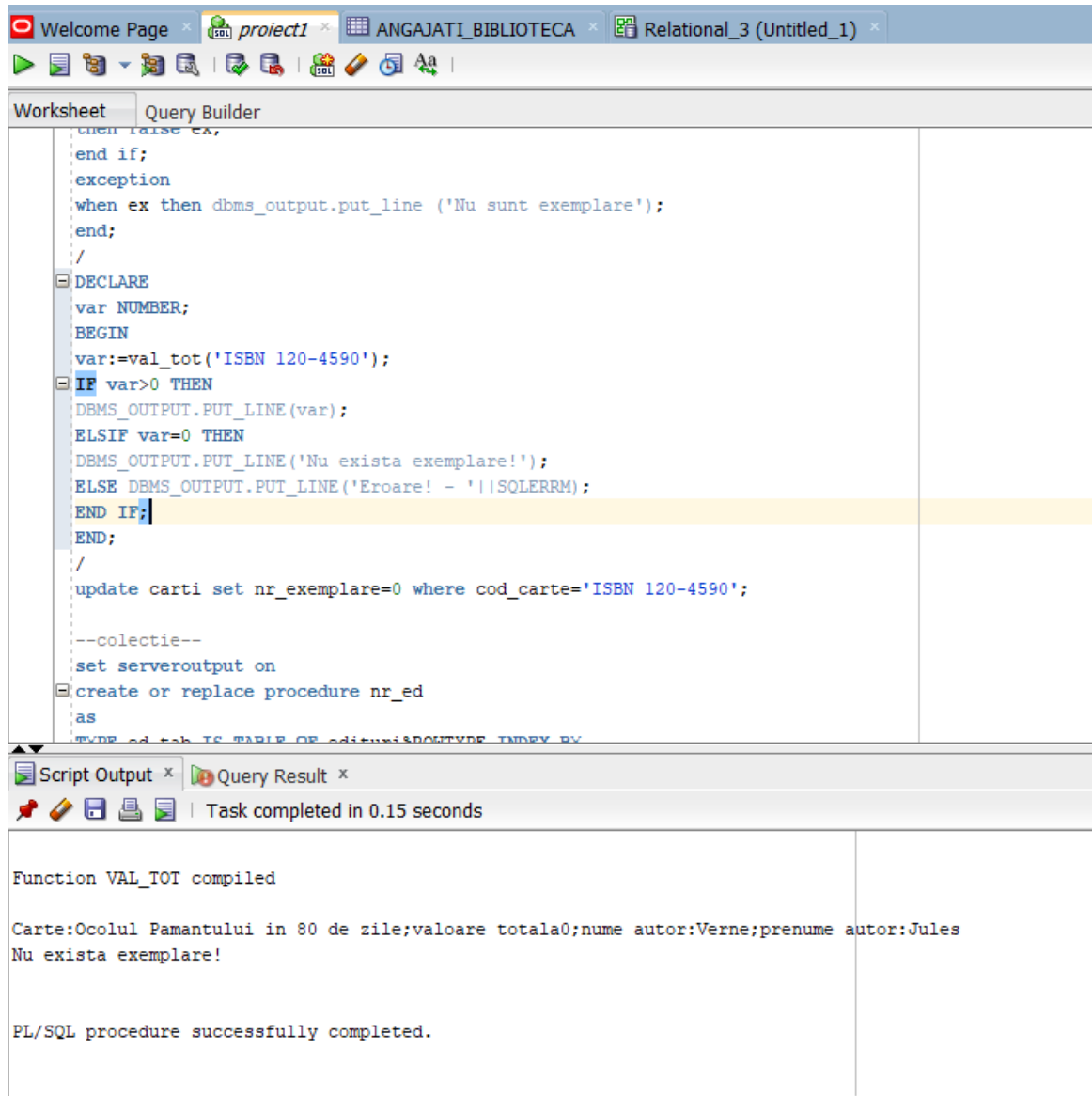
Apel2 (numarul de exemplare pentru cartea data prin parametru este setat la 0):

```

update carti set nr_exemplare=0 where cod_carte='ISBN 120-4590';
DECLARE
var NUMBER;
BEGIN
var:=val_tot('ISBN 120-4590');
IF var>0 THEN
DBMS_OUTPUT.PUT_LINE(var);
ELSIF var=0 THEN

```

```
DBMS_OUTPUT.PUT_LINE('Nu exista exemplare!');
ELSE DBMS_OUTPUT.PUT_LINE('Eroare! - '||SQLERRM);
END IF;
END;
/
```



## 9.SUBPROGRAM DE TIP PROCEDURA CARE SA UTILIZEZE 5 TABELE

- Creați o procedură care să afișeze codul, denumirea, tipul, informații despre autor și editura pentru o carte dată prin parametru. Să se modifice prețul cu un procent dat dacă valoarea totală ( $\text{pret} \times \text{nr exemplare}$ ) a cărții date prin parametru depășește 150 de lei.

```
set serveroutput on
```

```

create or replace procedure modif_pret (nrc in varchar,proc in number)
as
ex exception;
nre number;
v_cod carti.cod_carte%type;
v_coda scrise.cod_autor%type;
v_den carti.denumire%type;
v_aut autori.ume_autor%type;
v_autp autori.prenume_autor%type;
v_tip carti.tip%type;
v_ed edituri.ume_editura%type;
begin
select
s.cod_carte,c.denumire,s.cod_autor,a.ume_autor,a.prenume_autor,c.tip,e.ume_editura,sum(c.pret*c.n
r_exemplare) into v_cod,v_den,v_coda,v_aut,v_autp,v_tip,v_ed,nre from carti c,autori a, scrise
s,edituri e where
s.cod_carte=c.cod_carte and s.cod_autor=a.cod_autor and c.cod_editura=e.cod_editura and
c.cod_carte=v_cod
group by s.cod_carte,c.denumire,s.cod_autor,a.ume_autor,a.prenume_autor,c.tip,e.ume_editura ;
update carti
set pret=pret*(1+proc/100)
where cod_carte=v_cod and pret*nr_exemplare>150;
dbms_output.put_line('Carte:'||v_den||';valoare totala'||nre||';ume autor:'||v_aut||';prenume
autor:'||v_autp||';editura:'||v_ed);
if(nre=0)
then raise ex;
end if;
exception
when ex then dbms_output.put_line ('Nu exista exemplare');
end;
/

execute modif_pret('ISBN 123-4567',10);

```

10.

```
set serveroutput on
create or replace procedure modif_pret (nrc in varchar,proc in number)
as
ex exception;
nre number;
v_cod carti.cod_carte%type;
v_coda scrise.cod_autor%type;
v_den carti.denumire%type;
v_aut autori.numa_autor%type;
v_autp autori.prenume_autor%type;
v_tip carti.tip%type;
v_ed edituri.numa_editura%type;
begin
```

Script Output x

Task completed in 0.114 seconds

Procedure MODIF\_PRET compiled

## TRIGGER LMD LA NIVEL DE COMANDA

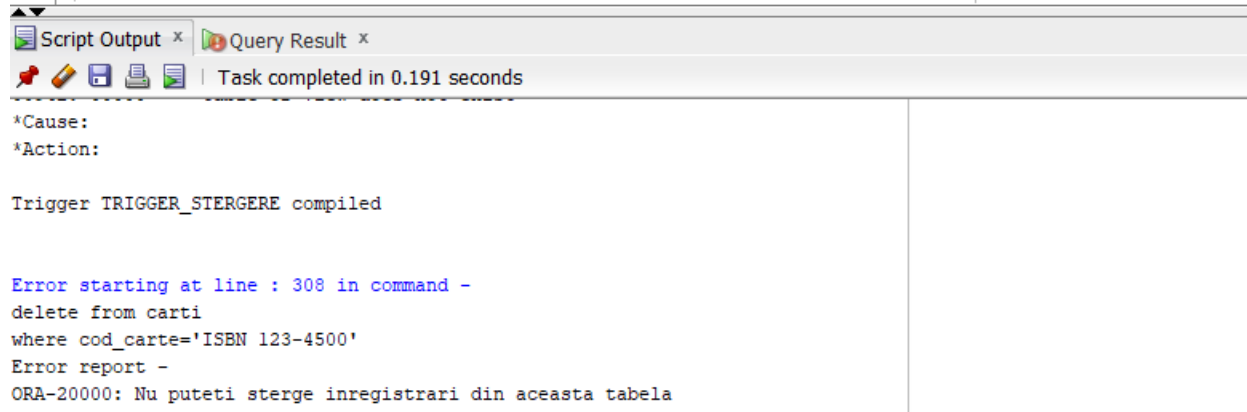
→ Creati un trigger care sa nu permita stergerea din tabela *CARTI*.

```
set serveroutput on
create or replace trigger trigger_stergere
before delete on carti
begin
raise_application_error(-20000,'Nu puteti sterge inregistrari din aceasta tabela');
end;
/
delete from carti
where cod_carte='ISBN 123-4500';
```

```

set serveroutput on
create or replace trigger trigger_stergere
before delete on carti
begin
raise_application_error(-20000,'Nu puteti sterge inregistrari din aceasta tabela');
end;
/
delete from carti
where cod_carte='ISBN 123-4500';

```

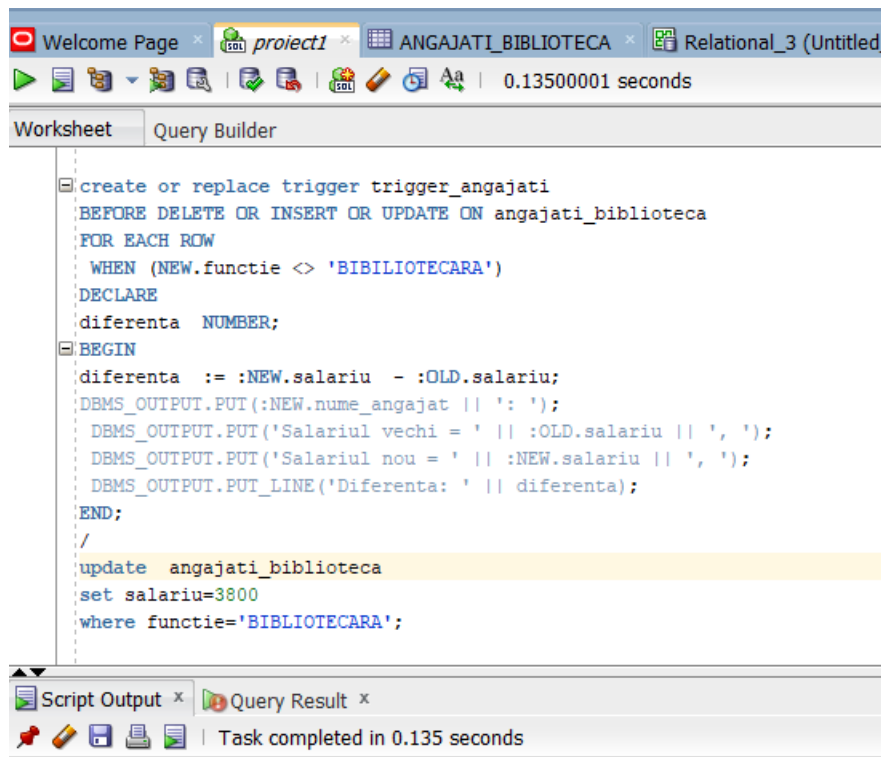


- Creati un trigger care sa afiseze modificarile aparute la un update pe tabela ANGAJATI\_BIBLIOTECA referitor la salariul angajatilor cu functia 'BIBLIOTECARA'.

```

create or replace trigger trigger_angajati
BEFORE DELETE OR INSERT OR UPDATE ON angajati_biblioteca
FOR EACH ROW
WHEN (NEW.functie <> 'BIBLIOTECARA')
DECLARE
diferenta NUMBER;
BEGIN
diferenta := :NEW.salariu - :OLD.salariu;
DBMS_OUTPUT.PUT(:NEW.num_e_angajat || ': ');
DBMS_OUTPUT.PUT('Salariul vechi = ' || :OLD.salariu || ', ');
DBMS_OUTPUT.PUT('Salariul nou = ' || :NEW.salariu || ', ');
DBMS_OUTPUT.PUT_LINE('Diferenta: ' || diferenta);
END;
/
update angajati_biblioteca
set salariu=3800
where functie='BIBLIOTECARA';

```



## 12.TRIGGER LDD

→ Creați un trigger care să introducă date în tabela *Useri* după ce utilizatorul a folosit o comandă LDD.

```

CREATE TABLE Useri
(user VARCHAR2(20),
nume_bd VARCHAR2(50),
eveniment VARCHAR2(20),
data DATE);
CREATE OR REPLACE TRIGGER trigger_ldd
AFTER CREATE OR DROP OR ALTER ON SCHEMA

```

```

BEGIN
INSERT INTO useri
VALUES (SYS.LOGIN_USER, SYS.DATABASE_NAME, SYS.SYSEVENT,
SYS.DICTIONARY_OBJ_NAME, SYSDATE);
END;
/
SELECT * FROM useri;

```

Trigger TRIGGER\_LDD compiled

insert into useri values ('andrei','carti','eveniment',TO\_DATE('12-05-2021','DD-MM-YYYY'));

	USERNAME	NUME_BD	EVENIMENT	DATA
1	andrei	carti	eveniment	12-MAY-21

### 13.PACHET CARE SA INCLUDA TOATE TABELELE

→ Creati un pachet care sa contina urmatoarele functii/proceduri:

- o procedura de inserare in tabela CARTI
- o functie care modifica numarul de exemplare al fiecărei carti cu 2 exemplare mai mult
- o functie de modificare a pretului unei carti cu 5%.
- o procedura de marire a salariului pentru angajatii care lucreaza intr-o locatie din Bucuresti
- o procedura care afiseaza informatii despre clienti si fisele de lectura
- o functie care sa afiseze codul, denumirea, tipul, informatii despre autor si valoarea totala (pret\*nr\_exemplare) pentru o carte data prin parametru

Procedurile/ functiile trateaza si exceptiile ce pot aparea.

```

create or replace package carti_up is
procedure ins_carti
(cod_c in carti.cod_carte%type,
nume out carti.denumire%type,
pr out carti.pret%type);

```

```

function ex_cod(cod_c carti.cod_carte%type)
return boolean;

```

```
function exp (a_nr_exemplare in number) return number;
```

```
function pretul(a_pret in number) return number;
```

```
function val_tot(nrc in varchar) return number;
```

```
PROCEDURE MARIRE_SALARIU(MAR IN NUMBER, ID_ANG IN NUMBER);
```

```
procedure inf_clienti;
```

```
exc exception;
```

```
end;
```

```
create or replace package body carti_up is
```

```
procedure ins_carti
```

```
(cod_c in carti.cod_carte%type,
```

```
nume out carti.denumire%type,
```

```
pr out carti.pret%type)
```

```
is
```

```
begin
```

```
if ex_cod(cod_c) then
```

```
raise exc;
```

```
else
```

```
insert into carti(cod_carte,denumire,pret) values (cod_c,nume,pr);
```

```
end if;
```

```
exception
```

```
when exc then
```

```
dbms_output.put_line('Nu exista cartea');
```

```
end;
```

```
function ex_cod
```

```
(cod_c carti.cod_carte%type)
```

```
return boolean
```

```
is
```

```
v_adv number;
```

```
begin
```

```
select 1 into v_adv
```

```
from carti
```

```
where cod_carte=cod_c;
```

```
return true;
```

```
exception
```

```
when no_data_found then
```

```
return false;
```

```
end;
```



```

function exp (a_nr_exemplare in number) return number
is
begin
return(a_nr_exemplare+2);
end;

```

```

function pretul(a_pret in number) return number
is
begin
return (a_pret+a_pret*0.05);
end;

```

```

function val_tot(nrc in varchar) return number
as
ex exception;
nre number;
v_codc scrie.cod_carte%type;
v_coda scrie.cod_autor%type;
v_den carti.denumire%type;
v_aut autori.numa_autor%type;
v_autp autori.prenume_autor%type;
v_tip carti.tip%type;
begin
select
s.cod_carte,c.denumire,s.cod_autor,a.numa_autor,a.prenume_autor,c.tip,sum(c.pret*c.nr_exemplare)
into v_codc,v_den,v_coda,v_aut,v_autp,v_tip,nre from carti c,autori a, scrie s where
s.cod_carte=c.cod_carte and s.cod_autor=a.cod_autor and c.cod_carte=nrc
group by s.cod_carte,c.denumire,s.cod_autor,a.numa_autor,a.prenume_autor,c.tip ;
dbms_output.put_line('Carte:'||v_den||';valoare totala'||nre||';numa autor:'||v_aut||';prenume
autor:'||v_autp);
return nre;
if(nre=0)
then raise ex;
end if;
exception
when ex then dbms_output.put_line ('Nu sunt exemplare');
end;

```

```

PROCEDURE MARIRE_SALARIU(MAR IN NUMBER, ID_ANG IN NUMBER)
AS
EXC EXCEPTION;
BEGIN

```

```

UPDATE ANGAJATI_BIBLIOTECA
SET SALARIU= SALARIU*(1+MAR*SALARIU)
WHERE id_locatie=(select id_locatie from locatii_biblioteca where adresa_locatie='%Bucuresti') and
COD_ANGAJAT= ID_ANG;
IF(SQL%ROWCOUNT=0)
THEN RAISE EXC;
ELSE
DBMS_OUTPUT.PUT_LINE('S-a modificat SALARIUL. ');
END IF;
EXCEPTION WHEN EXC
THEN DBMS_OUTPUT.PUT_LINE('Nu exista angajatul');
when others then dbms_output.put_line(sqlerrm);
end MARIRE_SALARIU;

```

```

procedure INF_CLIENTI as
tel clienti.telefon%type;
nume clienti.nume_cititor%type;
fis clienti.cod_fisa%type;
id_c clienti.cod_cititor%type:=&cod_cititor;
begin
begin
select nume_cititor into nume
from clienti
where cod_cititor=id_c;
dbms_output.put_line ('Cititorul'||id_c);
exception
when no_data_found then
dbms_output.put_line ('Nu sunt informatii');
end;
begin
select cod_fisa into fis
from fise_lectura_1
where cod_cititor=id_c;
dbms_output.put_line ('Cititorul'||nume||'are fisa: '||fis);
end;
exception
when no_data_found then
dbms_output.put_line ('Nu exista informatii despre fisa de lectura a acestui cititor');
end;
end;
/

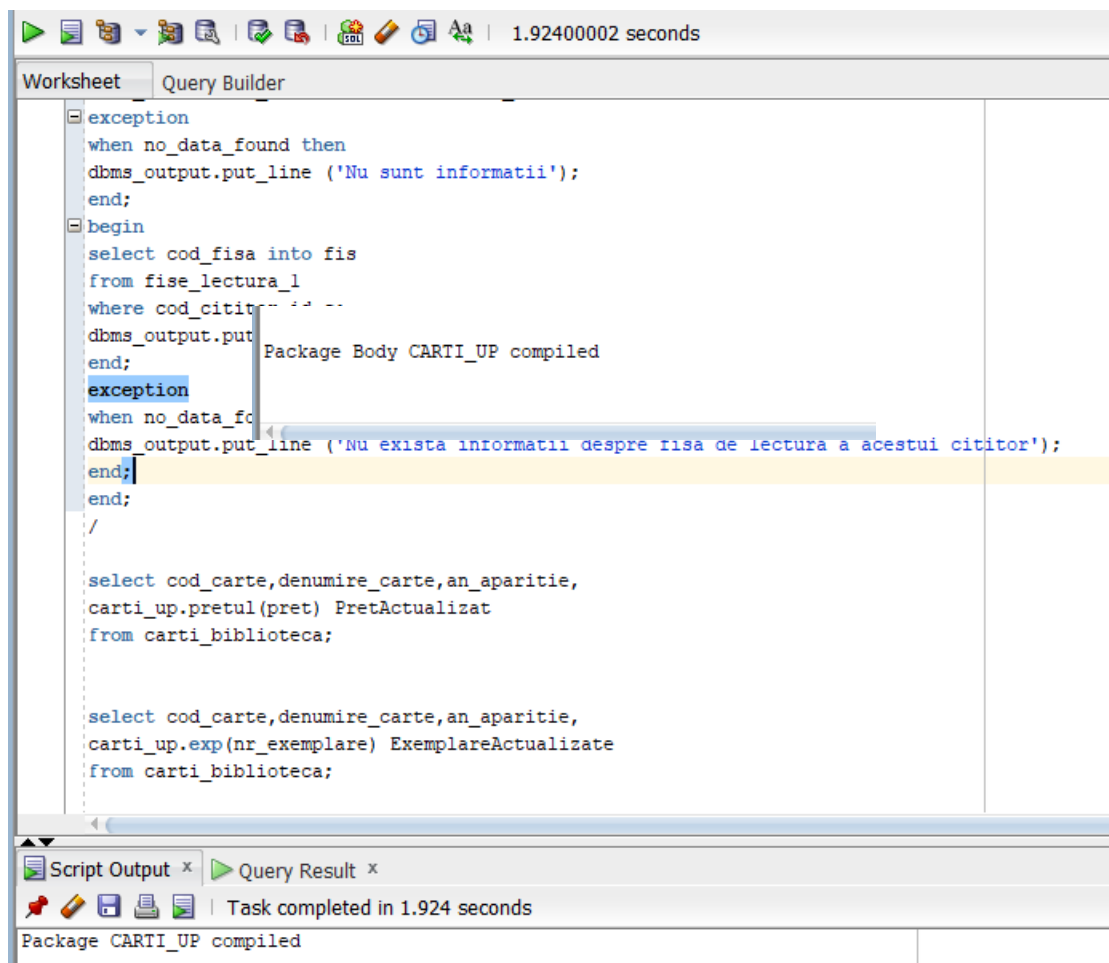
```

Apel:

```
select cod_carte,denumire,an_aparitie,  
carti_up.pretul(pret) PretActualizat  
from carti;
```

```
select cod_carte,denumire,an_aparitie,  
carti_up.exp(nr_exemplare) ExemplareActualizate  
from carti;
```

```
--apel prin bloc anonim--  
begin  
carti_up.INF_CLIENTI;  
end;  
/
```



Worksheet Query Builder

```

exception
when no_data_found then
dbms_output.put_line ('Nu sunt informatii');
end;
begin
select cod_fisa into fis
from fise_lectura_1
where cod_cititor=id_c;
dbms_output.put_line ('Cititorul'||nume||'are fisa:'||fis);
end;
exception
when no_data_found then
dbms_output.put_line ('Nu exista informatii despre fisa de lectura a acestui cititor');
end;
end;
/

select cod_carte,denumire,an_aparitie,
carti_up.pretul(pret) PretActualizat
from carti;

select cod_carte,denumire_carte,an_aparitie,
carti_up.exp(nr_exemplare) ExemplareActualizate
from carti_biblioteca;

```

Script Output x Query Result x Query Result 1 x Query Result 2 x

SQL | All Rows Fetched: 14 in 0.034 seconds

	COD_CARTE	DENUMIRE	AN_APARTIE	PRETACTUALIZAT
1	ISBN 120-4590	Ocolul Pamantului in 80 de zile	1873	29.4
2	ISBN 103-4570	Copiii Capitanului Grant	186	29.4
3	ISBN 193-4767	Amintiri din copilarie	1959	31.5
4	ISBN 193-0567	Maitreyi	1933	27.3
5	ISBN 123-9577	Zece Negri Mititei	1939	26.25
6	ISBN 129-4568	Crima din Orient Express	1934	25.2
7	ISBN 125-9067	Orbitor	2007	31.5
8	ISBN 123-4576	Nostalgia	1989	37.8
9	ISBN 120-4997	Lolita	1955	22.05
10	ISBN 123-4550	Razboi si pace	1867	37.8
11	ISBN 122-4695	Invierea	1899	32.55

The screenshot shows the Oracle SQL Developer interface. The 'Script Output' window displays the following text:

```

select cod_carte,denumire,an_aparitie,
carti_up.exp(nr_exemplare) ExemplareActualizate
from carti;

begin
carti_up.INF_CLIENTI;
end;
/

Task completed in 0.146 seconds

when no_data_found then
dbms_output.put_line ('Nu exista informatii despre fisa de lectura a acestui cititor');
end;
end;

Package Body CARTI_UP compiled

Cititorul2671604578392
Nu exista informatii despre fisa de lectura a acestui cititor

PL/SQL procedure successfully completed.

```

The 'Query Result' window shows a table with the following data:

ExemplareActualizate
2
7
4
7
5
7
4
8
6
7
7

## 14.PACHET

- Construiti un pachet care sa contina functii si proceduri referitoare la editurile cartilor din biblioteca:
- un cursor LISTA\_ED care sa parcurga editurile
  - o functie care sa gaseasca o editura al carei cod este dat prin parametru
  - o procedura care afiseaza editura
  - o procedura care afiseaza informatii despre editura data prin parametru si despre cartile care au editura respectiva.

```

CREATE SEQUENCE sec_cod_editura START WITH 101;
CREATE OR REPLACE PACKAGE pachet_edituri
IS
  CURSOR lista_ed(
    v_cod edituri.cod_editura%type)
  IS
    SELECT * FROM edituri
    WHERE cod_editura = v_cod
;

```

```

FUNCTION gaseste_editura(
v_cod edituri.cod_editura%type)
RETURN BOOLEAN;

PROCEDURE afiseaza_editura(
v_cod edituri.cod_editura%type);

PROCEDURE carti_cu_editura_data(
v_cod in out edituri.cod_editura%type);

END pachet_edituri;
/
CREATE OR REPLACE PACKAGE BODY pachet_edituri
IS

```

```

FUNCTION gaseste_editura(
v_cod edituri.cod_editura%type)
RETURN BOOLEAN
IS
rez NUMBER;
BEGIN
SELECT COUNT(*) INTO rez
FROM edituri
WHERE cod_editura=v_cod;

IF rez>0 THEN RETURN TRUE;
ELSE RETURN FALSE;
END IF;
END gaseste_editura;

```

```

PROCEDURE afiseaza_editura(
v_cod edituri.cod_editura%type)
IS

BEGIN
FOR i IN lista_ed(v_cod) LOOP
IF lista_ed%ROWCOUNT=1 THEN

DBMS_OUTPUT.PUT_LINE('Editura: '||
i.num_e_editura || ' cu codul '||i.cod_editura );

```

```

ELSE
DBMS_OUTPUT.PUT_LINE('Nu s-a gasit editura');
END IF;
END LOOP;
END afiseaza_editura;

```

```

PROCEDURE carti_cu_editura_data(
v_cod in out edituri.cod_editura%type)
IS
v_cod_carte carti.cod_carte%type;
v_den carti.denumire%type;
BEGIN
IF gaseste_editura(v_cod)
THEN
DBMS_OUTPUT.PUT_LINE('Editura gasita!');
afiseaza_editura(v_cod);
select c.cod_carte, c.denumire, e.ume_editura into v_cod_carte,v_den,v_cod from carti c, edituri e
where c.cod_editura=e.cod_editura and c.cod_editura=v_cod;
ELSE

```

```

DBMS_OUTPUT.PUT_LINE('Nu s-a gasit editura');
END IF;
END carti_cu_editura_data;
END pachet_edituri;

```

```

set serveroutput on

```

```

begin
pachet_edituri.carti_cu_editura_data(101);
end;
/

```

```

begin
pachet_edituri.afiseaza_editura(101);
end;
/

```

0.38299999 seconds

Worksheet Query Builder

```
END IF;
END gaseste_editura;

PROCEDURE afiseaza_editura(
  v_cod edituri.cod_editura%type)
IS
BEGIN
FOR i IN lista_ed(v_cod) LOOP
IF lista_ed%ROWCOUNT=1 THEN

DBMS_OUTPUT.PUT_LINE('Editura: '||
i.num_e_editura );

ELSE
DBMS_OUTPUT.PUT_LINE('Nu s-a gasit editura');
END IF;
END LOOP;
END afiseaza_editura;

PROCEDURE carti_cu_editura_data(
v_cod out edituri.cod_editura%type)
IS
v_cod_carte carti.cod_carte%type;
v_den carti.denumire%type;
BEGIN
IF gaseste_editura(v_cod)
THEN
DBMS_OUTPUT.PUT_LINE('Editura gasita!');
afiseaza_editura(v_cod);
select c.cod_carte, c.denumire, e.num_e_editura into v_cod_carte,v_den,v_cod from carti c, edituri e
where c.cod_editura=e.cod_editura and c.cod_editura=v_cod;
ELSE

DBMS_OUTPUT.PUT_LINE('Nu s-a gasit editura');
END IF;
END carti_cu_editura_data;
END pachet_edituri;
```

Script Output x

Task completed in 0.383 seconds

Package PACHET\_EDITURI compiled

Package Body PACHET\_EDITURI compiled

```
/
begin
pachet_edituri.afiseaza_editura(101);
end;
/
```

Script Output x

Task completed in 0.125 seconds

Package PACHET\_EDITURI compiled

Package Body PACHET\_EDITURI compiled

Editura: Accent cu codul 101

PL/SQL procedure successfully completed.