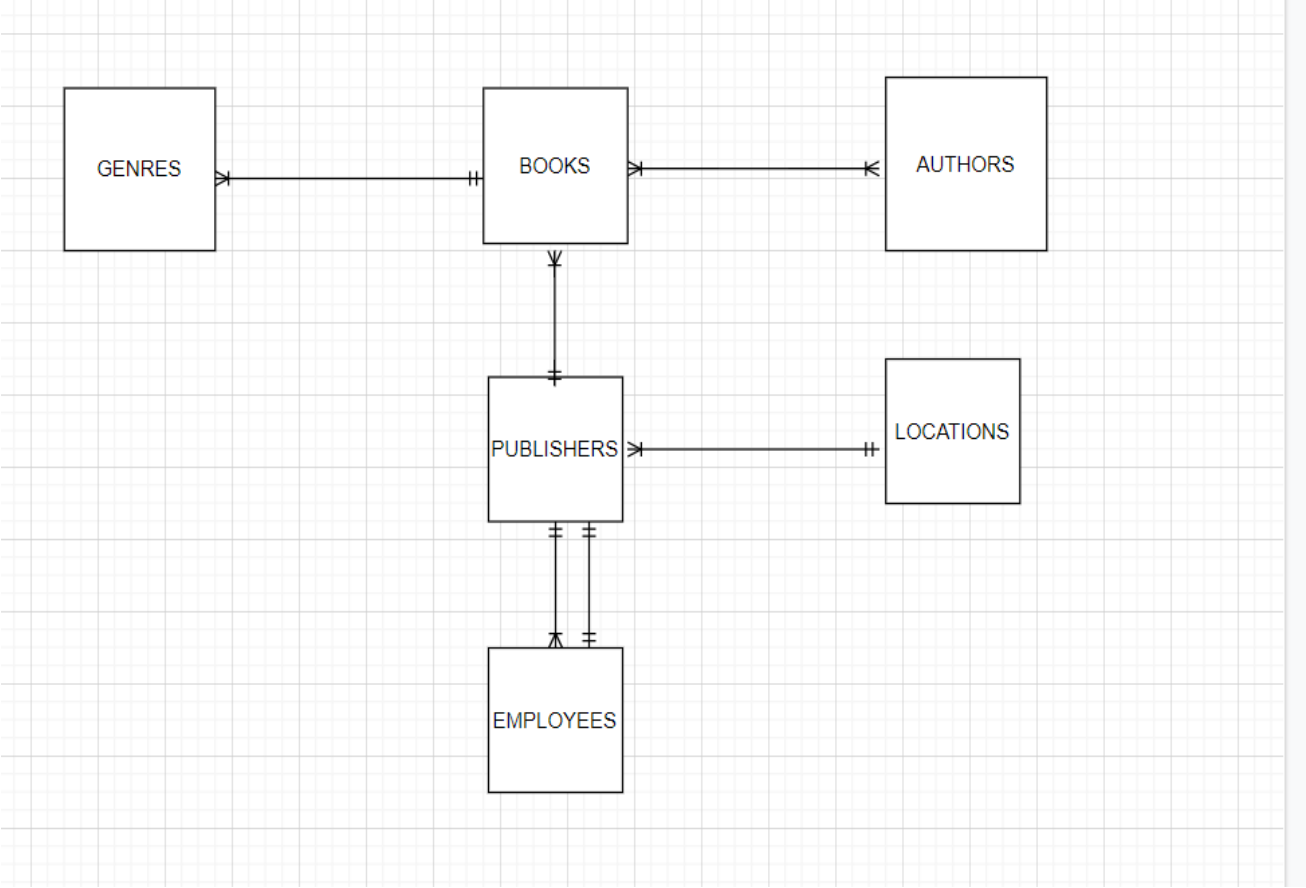
Proiect SGBD

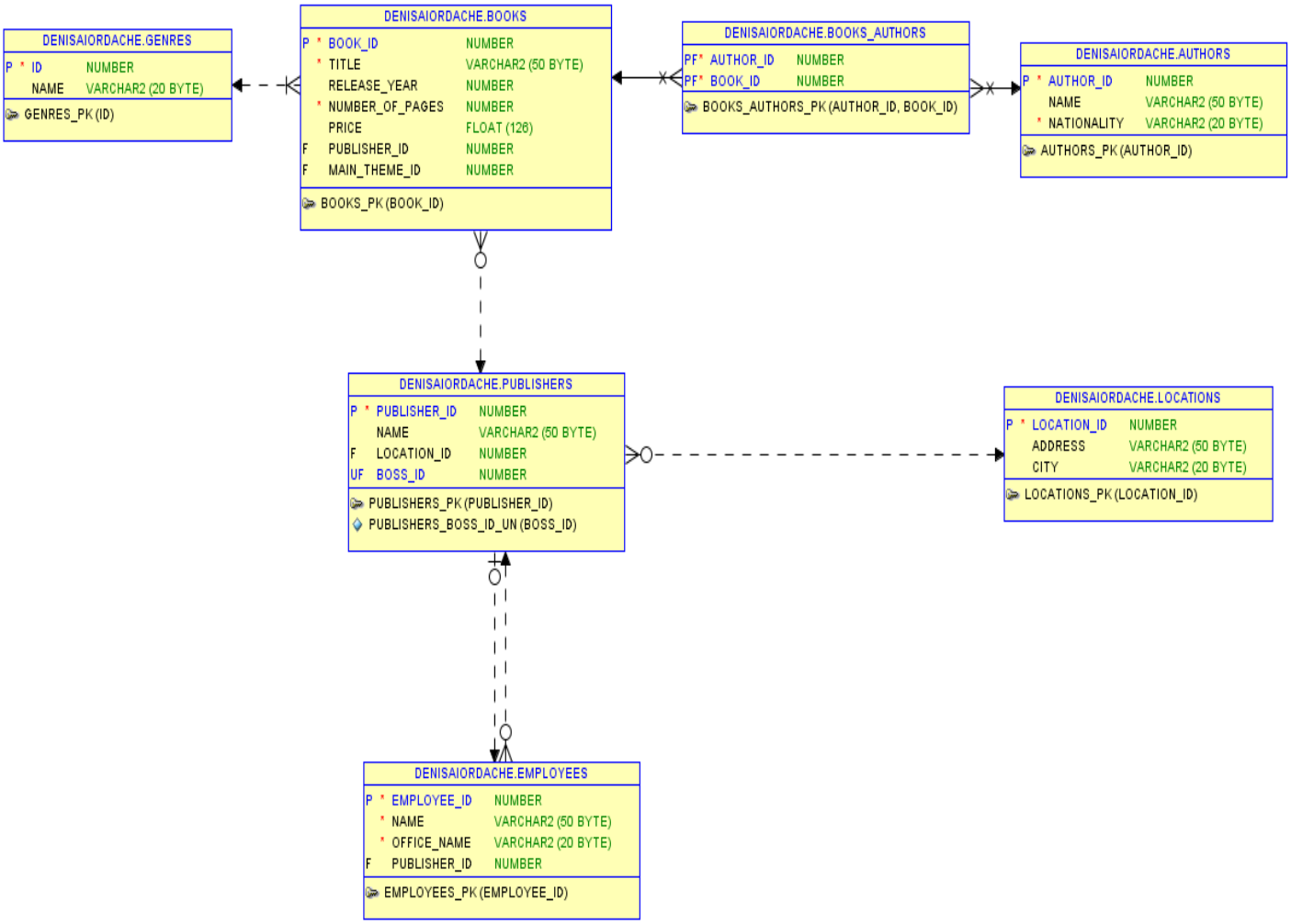
1. Prezentați pe scurt baza de date (utilitatea ei).

Tema proiectului meu este reprezentată de gestionarea resurselor unui anticariat: informații despre cărțile deținute (autor, gen, preț etc), cât și despre proveniența acestora (editura unde au fost publicate, locația acesteia și angajații din cadrul ei).

2. Realizați diagrama entitate-relație (ERD).



3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare.



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

```
CREATE TABLE LOCATIONS (
location_id INT NOT NULL,
address VARCHAR(50),
city VARCHAR(20),
PRIMARY KEY (location_id)
);
```

```
   CREATE TABLE BOOKS (
  book_id INT NOT NULL,
  title VARCHAR(50) NOT NULL,
  release_year INT,  --in curs de scriere
  number_of_pages INT NOT NULL,
  price float,
  publisher_id INT null,
  PRIMARY KEY (book_id)
);



CREATE TABLE AUTHORS (
  author_id INT NOT NULL,
  name VARCHAR(50),
  nationality VARCHAR(20) NOT NULL,
  PRIMARY KEY(author_id)
);

CREATE TABLE BOOKS_AUTHORS (
   author_id INT NOT NULL,
   book_id  INT NOT NULL,
   PRIMARY KEY (author_id,book_id),
    CONSTRAINT FK_Author FOREIGN KEY (author_id) REFERENCES
AUTHORS(author_id) ON DELETE CASCADE,
   CONSTRAINT FK_Book FOREIGN KEY (book_id) REFERENCES BOOKS(book_id)
ON DELETE CASCADE
);

CREATE TABLE PUBLISHERS (
  publisher_id INT NOT NULL,
  name VARCHAR(50),
```

```
  location_id INT NULL,
  boss_id INT NULL,
  PRIMARY KEY (publisher_id),
  CONSTRAINT FK_PublishersLocations FOREIGN KEY (location_id) REFERENCES
Locations(location_id) ON DELETE SET NULL
);

CREATE TABLE EMPLOYEES (
  employee_id INT NOT NULL,
  name VARCHAR(50) NOT NULL,
  office_name VARCHAR(20) NOT NULL,
  publisher_id INT NULL,
  PRIMARY KEY (employee_id),
  CONSTRAINT FK_Publishers FOREIGN KEY (publisher_id) REFERENCES
PUBLISHERS (publisher_id) ON DELETE SET NULL
  );

CREATE TABLE GENRES (
  id INT NOT NULL,
  name VARCHAR(20),
  PRIMARY KEY (id)
  );

ALTER TABLE BOOKS
ADD main_theme_id INT NULL;

ALTER TABLE BOOKS
ADD CONSTRAINT FK_BooksPublishers FOREIGN KEY(publisher_id) references
PUBLISHERS (publisher_id) ON DELETE SET NULL;

ALTER TABLE PUBLISHERS
```

ADD CONSTRAINT FK_PublishersEmp FOREIGN KEY (boss_id) REFERENCES EMPLOYEES (employee_id) ON DELETE SET NULL;

alter table publishers

add unique (boss_id);

5. Adăugați informații coerente în tabelele create (minim 3-5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

INSERT ALL

INTO AUTHORS(author_id, name, nationality) VALUES(1,'Lev Tolstoi', 'Russian')

INTO AUTHORS(author_id, name, nationality) VALUES (2,'Camilla Lackberg', 'Swedish')

INTO AUTHORS(author_id, name, nationality) VALUES (3,'Vintila Corbul','Romanian')

INTO AUTHORS(author_id, name, nationality) VALUES (4,'Eugen Burada','Romanian')

INTO AUTHORS(author_id, name, nationality) VALUES (5,'Neil Gaiman','British')

INTO AUTHORS(author_id, name, nationality) VALUES (6,'Terry Pratchett','British')

INTO AUTHORS(author_id, name, nationality) VALUES (7,'Hector Malot','French')

INTO AUTHORS(author_id, name, nationality) VALUES(8,'Stanley B. Lippman', 'British')

INTO AUTHORS(author_id, name, nationality) VALUES(9,'Josee Lajoie','Canadian')

INTO AUTHORS(author_id, name, nationality) VALUES(10,'Barbara E. Moo','American')

INTO AUTHORS(author_id, name, nationality) VALUES(11,'Simon Montefiore','Romanian')

INTO AUTHORS(author_id, name, nationality) VALUES(12,'Mircea Cartarescu','Romanian')

INTO AUTHORS(author_id, name, nationality) VALUES(13,'Chris Riddell','British')

SELECT * FROM DUAL;

INSERT ALL

INTO BOOKS(book_id, title, release_year,number_of_pages,price) VALUES (1,'The Stonecutter',2005,480,49.99)

INTO BOOKS(book_id, title, release_year,number_of_pages,price) VALUES (2,'Uragan asupra Europei',1979,828,20.00)

INTO BOOKS(book_id, title, release_year,number_of_pages,price) VALUES (3,'Good Omens',1990, 288, 42.00)

```
INTO BOOKS(book_id, title, release_year,number_of_pages,price) VALUES (4, 'Sans famille', 1878, 384, 19.00)

INTO BOOKS(book_id, title, release_year,number_of_pages,price) VALUES (5, 'Anna Karenina' ,1875, 864, 32.29)

INTO BOOKS(book_id, title, release_year,number_of_pages,price) VALUES (6,'C++ Primer 5th Edition',2012,976,150.22)

INTO BOOKS(book_id, title, release_year,number_of_pages,price) VALUES (7,'Romanovii',2016,856,94.99)

INTO BOOKS(book_id, title, release_year,number_of_pages,price) VALUES (8,'Solenoid',2015,840,36.66)

INTO BOOKS(book_id, title, release_year,number_of_pages,price) VALUES (9,'Art matters',2018,112,45.00)

SELECT * FROM DUAL;


INSERT ALL

INTO BOOKS_AUTHORS(author_id,book_id) VALUES (1,5)

INTO BOOKS_AUTHORS(author_id,book_id) VALUES (2,1)

INTO BOOKS_AUTHORS(author_id,book_id) VALUES (3,2)

INTO BOOKS_AUTHORS(author_id,book_id) VALUES (4,2)

INTO BOOKS_AUTHORS(author_id,book_id) VALUES (5,3)

INTO BOOKS_AUTHORS(author_id,book_id) VALUES (6,3)

INTO BOOKS_AUTHORS(author_id,book_id) VALUES (7,4)

INTO BOOKS_AUTHORS(author_id,book_id) VALUES (8,6)

INTO BOOKS_AUTHORS(author_id,book_id) VALUES (9,6)

INTO BOOKS_AUTHORS(author_id,book_id) VALUES (10,6)

INTO BOOKS_AUTHORS(author_id,book_id) VALUES (11,7)

INTO BOOKS_AUTHORS(author_id,book_id) VALUES (12,8)

INTO BOOKS_AUTHORS(author_id,book_id) VALUES (5,9)

INTO BOOKS_AUTHORS(author_id,book_id) VALUES (13,9)

SELECT * FROM dual;


INSERT INTO LOCATIONS

VALUES(1,'Str. Sf. Constantin, 9', 'Bucuresti'); --Trei
```

```
INSERT INTO LOCATIONS
VALUES(2,'Str. Mihai Eminescu, 54A', 'Bucuresti'); --Corint

INSERT INTO LOCATIONS
VALUES(3,'Str. Moeciu, 7A', 'Bucuresti');  --Litera

INSERT INTO LOCATIONS
VALUES(4,'Str. Barbu Vacarescu 164A-B', 'Bucuresti'); --Adevarul

INSERT INTO LOCATIONS
values (5,'20 Vauxhall Bridge Rd', 'Londra'); --Transworld

INSERT ALL
INTO LOCATIONS(location_id,address,city) VALUES (6,'75 Arlington Street Suite 300','Boston')
INTO LOCATIONS(location_id,address,city) VALUES (7,'Piata Presei Libere, 1','Bucuresti')
INTO LOCATIONS(location_id,address,city) VALUES (8,'HarperCollins Publishers, 10 East 53rd','New York')
SELECT * FROM DUAL;

INSERT ALL
INTO PUBLISHERS(publisher_id, name, location_id) VALUES (1,'Trei',1)
INTO PUBLISHERS(publisher_id, name, location_id) VALUES (2,'Adevarul', 4)
INTO PUBLISHERS(publisher_id, name, location_id) VALUES (3, 'Corint',2)
INTO PUBLISHERS(publisher_id, name, location_id) VALUES (4,'Litera', 3)
INTO PUBLISHERS(publisher_id, name, location_id) VALUES (5,'Transword Publisher',5)
INTO PUBLISHERS(publisher_id, name, location_id) VALUES (6,'Addison Wesley',6)
INTO PUBLISHERS(publisher_id, name, location_id) VALUES (7,'William Morrow',8)
INTO PUBLISHERS(publisher_id, name, location_id) VALUES (8,'Humanitas',7)
select * from dual;
```

```
INSERT ALL
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (1,'Popescu Raluca','Grafica',1)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (2,'Cernea Monica','Tehnoredactare',1)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (3,'Negru Andrei','Productie',1)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (4,'Simionescu Elena','Grafica',1)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (5,'Cristescu Stefan','Marketing',2)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES(6,'Rosu Laurentiu','Productie',2)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (7,'Marcu Ionut','Drepturi de editare',2)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (8,'Marcu Simona','Tehnoredactare',2)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (9,'Obreja Lorena','Tehnoredactare',2)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (10,'Caraiman Florin','Promovare',3)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (11,'Begu Paul','Drepturi de editare',3)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (12,'Lupu Andreea','Grafica',4)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (13,'Iordache Denisa','Tehnoredactare',4)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (14, 'Melcea Gabriel','Productie',4)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (15,'Corbu Omar','Drepturi de editare',4)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (16,'Zamfir Loredana','Marketing',4)
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (17,'Gary Reynolds','Marketing',5)
```

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (18,'Russel Maddox','Tehnoredactare',5)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (19,'Elaine Woods','Tehnoredactare',5)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (20,'Laura Clery','Productie',6)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (21,'Michael Gordon','Grafica',6)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (22,'Denise Russel','Tehnoredactare',6)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (23,'Liam Lodge','Marketing',6)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (24,'Veronica Hastings','Marketing',6)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (25,'Andrew Fitterman','Drepturi de autor',6)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (26,'Craig Newman','Productie',6)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (27,'Serena Larson','Grafica',6)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (28,'Lucas Sloane','Contabilitate',7)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (29,'Marcus King','Grafica',7)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (30,'Monica Miller','Tehnoredactare',7)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (31,'Christina Gray','Drepturi de editare',7)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (32,'Stanescu Theodor','Marketing',8)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (33,'Mungiu Lucas','Grafica',8)

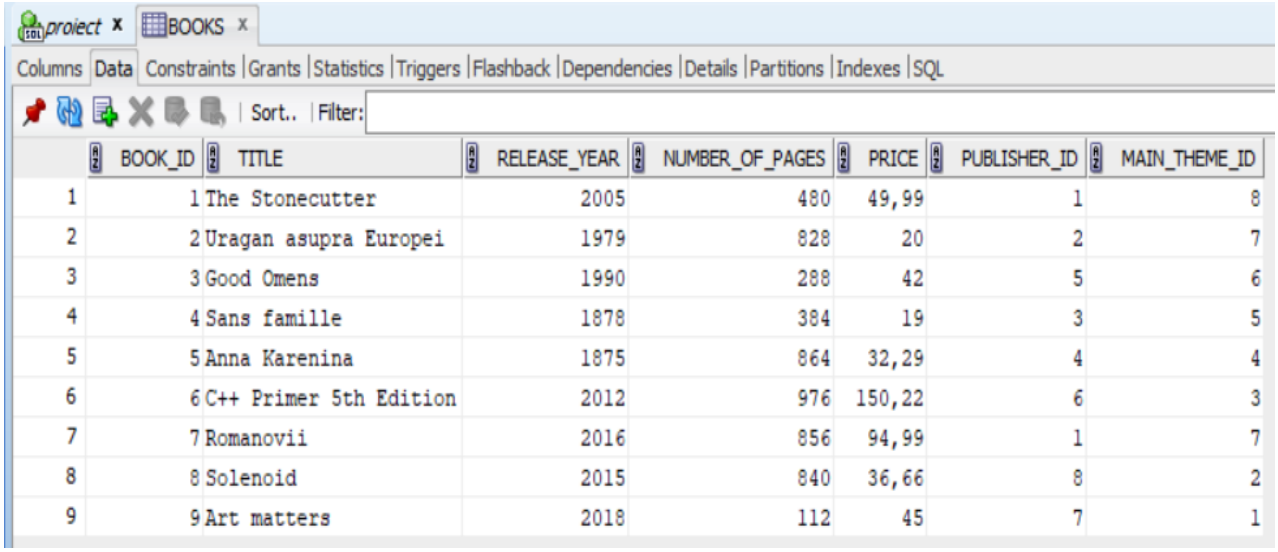
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (34,'Lucian Perju','Tehnoredactare',8)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (35,'Daniela Maxim','Tehnoredactare',8)

```
INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (36,'Cosmina Cretu','Drepturi de editare',8)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (37,'Dorina Anghel','Marketing',8)

INTO EMPLOYEES(employee_id,name,office_name, publisher_id) VALUES (38,'Cristina Arsene','Contabilitate',8)

SELECT * FROM DUAL;

commit;


update books

set publisher_id =7

where book_id = 9;


update publishers

set boss_id = 34

where publisher_id =8;


commit;


INSERT ALL

INTO GENRES(id,name) VALUES(1,'Dezvoltare personala')

INTO GENRES(id,name) VALUES(2,'Autobiografie')

INTO GENRES(id,name) VALUES(3,'Stiinta')

INTO GENRES(id,name) VALUES(4,'Romance')

INTO GENRES(id,name) VALUES(5,'Aventura')

INTO GENRES(id,name) VALUES(6,'Fantasy')

INTO GENRES(id,name) VALUES(7,'Istoric')

INTO GENRES(id,name) VALUES(8,'Thriller')

SELECT * FROM DUAL;


UPDATE BOOKS

SET main_theme_id= 1
```

WHERE book_id = 9;

commit;



	BOOK_ID	TITLE	RELEASE_YEAR	NUMBER_OF_PAGES	PRICE	PUBLISHER_ID	MAIN_THEME_ID
1	1	The Stonecutter	2005	480	49,99	1	8
2	2	Uragan asupra Europei	1979	828	20	2	7
3	3	Good Omens	1990	288	42	5	6
4	4	Sans famille	1878	384	19	3	5
5	5	Anna Karenina	1875	864	32,29	4	4
6	6	C++ Primer 5th Edition	2012	976	150,22	6	3
7	7	Romanovii	2016	856	94,99	1	7
8	8	Solenoid	2015	840	36,66	8	2
9	9	Art matters	2018	112	45	7	1

proiect × AUTHORS ×

Columns | Data | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	AUTHOR_ID	NAME	NATIONALITY
1	1	Lev Tolstoi	Russian
2	2	Camilla Lackberg	Swedish
3	3	Vintila Corbul	Romanian
4	4	Eugen Burada	Romanian
5	5	Neil Gaiman	British
6	6	Terry Pratchett	British
7	7	Hector Malot	French
8	8	Stanley B. Lippman	British
9	9	Josee Lajoie	Canadian
10	10	Barbara E. Moo	American
11	11	Simon Montefiore	Romanian
12	12	Mircea Cartarescu	Romanian
13	13	Chris Riddell	British
14	14	Hector Malot	Belgian
15	15	Charles Dickens	British

proiect × BOOKS_AUTHORS ×

Columns | Data | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	AUTHOR_ID	BOOK_ID
1	1	5
2	2	1
3	3	2
4	4	2
5	5	3
6	6	3
7	7	4
8	8	6
9	9	6
10	10	6
11	11	7
12	12	8
13	5	9
14	13	9

proiect × GENRES ×

Columns | Data | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	ID	NAME
1	1	Dezvoltare personala
2	2	Autobiografie
3	3	Stiinta
4	4	Romance
5	5	Aventura
6	6	Fantasy
7	7	Istoric
8	8	Thriller

proiect × LOCATIONS ×

Columns | Data | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	LOCATION_ID	ADDRESS	CITY
1	1	Str. Sf. Constantin, 9	Bucuresti
2	2	Str. Mihai Eminescu, 54A	Bucuresti
3	3	Str. Moeciu, 7A	Bucuresti
4	4	Str. Barbu Vacarescu 164A-B	Bucuresti
5	5	20 Vauxhall Bridge Rd	Londra
6	6	75 Arlington Street Suite 300	Boston
7	7	Piata Presei Libere, 1	Bucuresti
8	8	HarperCollins Publishers, 10 East 53rd	New York
9	9	Str Amaradiei, nr 4	Craiova
10	10	Str Preot Vasile Lucaciu, 15	Bucuresti

proiect × EMPLOYEES ×

Columns | Data | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	EMPLOYEE_ID	NAME	OFFICE_NAME	PUBLISHER_ID
1	1	Popescu Raluca	Grafica	1
2	2	Cernea Monica	Tehnoredactare	1
3	3	Negru Andrei	Productie	1
4	4	Simionescu Elena	Grafica	1
5	5	Cristescu Stefan	Marketing	2
6	6	Rosu Laurentiu	Productie	2
7	7	Marcu Ionut	Drepturi de editare	2
8	8	Marcu Simona	Tehnoredactare	2
9	9	Obreja Lorena	Tehnoredactare	2
10	10	Caraiman Florin	Promovare	3
11	11	Begu Paul	Drepturi de editare	3
12	12	Lupu Andreea	Grafica	4
13	13	Iordache Denisa	Tehnoredactare	4
14	14	Melcea Gabriel	Productie	4
15	15	Corbu Omar	Drepturi de editare	4
16	16	Zamfir Loredana	Marketing	4
17	17	Gary Reynolds	Marketing	5
18	18	Russel Maddox	Tehnoredactare	5
19	19	Elaine Woods	Tehnoredactare	5
20	20	Laura Clery	Productie	6
21	21	Michael Gordon	Grafica	6
22	22	Denise Russel	Tehnoredactare	6
23	23	Liam Lodge	Marketing	6
24	24	Veronica Hastings	Marketing	6
25	25	Andrew Fitterman	Drepturi de autor	6
26	26	Craig Newman	Productie	6
27	27	Serena Larson	Grafica	6
28	28	Lucas Sloane	Contabilitate	7
29	29	Marcus King	Grafica	7
30	30	Monica Miller	Tehnoredactare	7

proiect × PUBLISHERS ×

Columns | Data | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

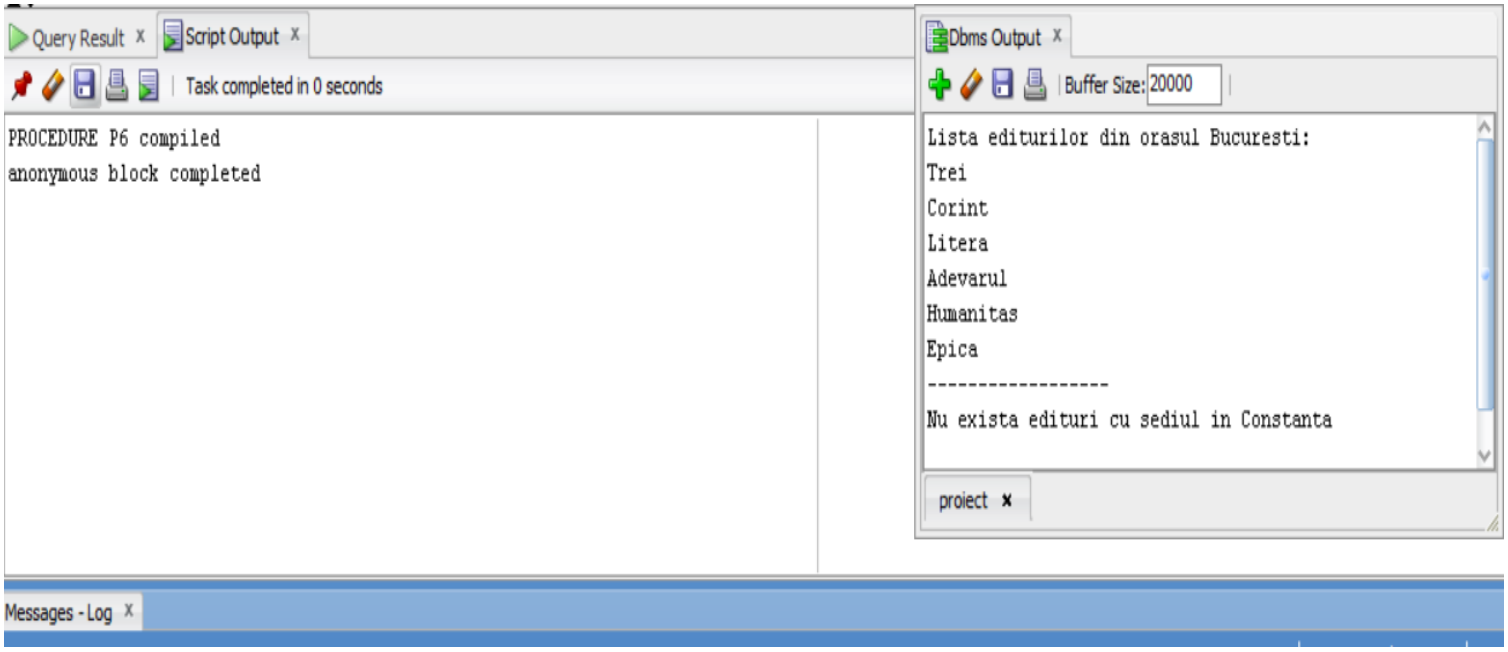
	PUBLISHER_ID	NAME	LOCATION_ID	BOSS_ID
1	1	Trei	1	4
2	2	Adevarul	4	8
3	3	Corint	2	11
4	4	Litera	3	16
5	5	Transword Publisher	5	18
6	6	Addison Wesley	6	25
7	7	William Morrow	8	30
8	8	Humanitas	7	34
9	9	Epica	10	(null)

6. Definiți un subprogram stocat care să utilizeze un tip de colecție studiat. Apelați subprogramul.

Cerinta: Afisati toate editurile ce au sediul intr-un oras dat ca parametru. Rezultatul va fi stocat intr-un vector.

```
CREATE OR REPLACE TYPE vector IS VARRAY(100) OF NUMBER;

/

CREATE OR REPLACE PROCEDURE p6 (v_oras locations.city%type default 'Bucuresti')

IS

v_coduri vector := vector();

v_denumire publishers.name%type;

exc exception;


BEGIN

SELECT location_id

BULK COLLECT INTO v_coduri

FROM locations

WHERE city = v_oras;


IF v_coduri.count = 0 THEN

  raise exc;

END IF;
```

```
dbms_output.put_line('Lista editurilor din orasul '||v_oras||':');
FOR i IN 1..v_coduri.count LOOP
  select name
  into v_denumire
  from publishers
  where location_id = v_coduri(i);
  dbms_output.put_line(v_denumire);
END LOOP;
dbms_output.put_line('------------------');



exception
WHEN exc THEN
  DBMS_OUTPUT.PUT_LINE('Nu exista edituri cu sediul in ' || v_oras);
END p6;
/

BEGIN
p6();
p6('Constanta');
END;
/
```



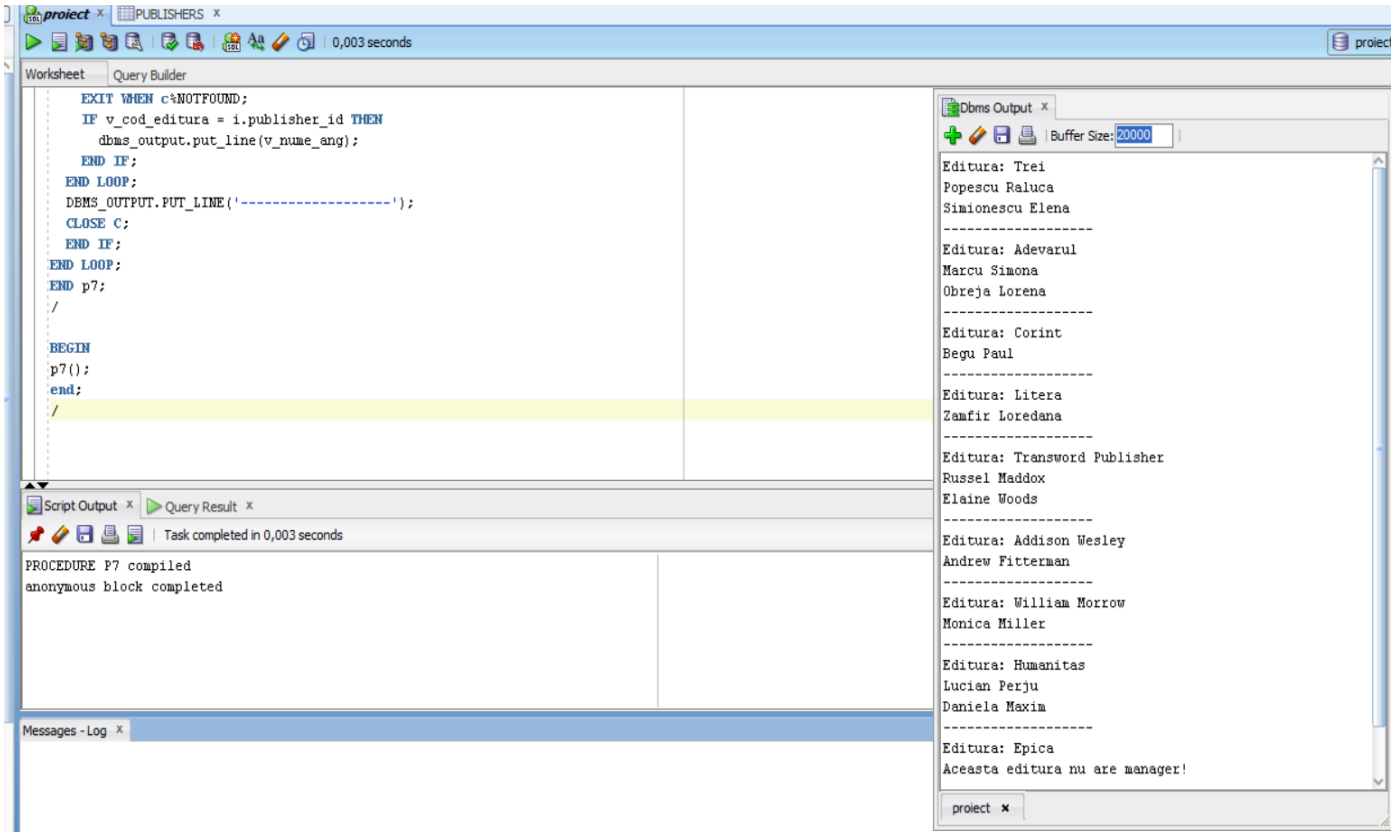
7. Definiți un subprogram stocat care să utilizeze un tip de cursor studiat. Apelați subprogramul.

Cerinta: Pentru fiecare editura, afisati angajatii care lucreaza in acelasi birou ca si managerul (inclusiv numele acestuia)

```
CREATE OR REPLACE PROCEDURE p7
IS
CURSOR c (v_office employees.office_name%type)
IS SELECT name,publisher_id
   FROM employees
   WHERE office_name = v_office;
v_nume_ang employees.name%type;
v_cod_editura INT;
v_cod_sef INT;
v_office employees.office_name%type;
v_manager INT;
```

```
BEGIN

FOR i IN (SELECT publisher_id, name, boss_id FROM publishers) LOOP

  SELECT COUNT(boss_id)
  INTO v_manager
  FROM publishers
  WHERE publisher_id = i.publisher_id;



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

  IF v_manager = 0 THEN
    dbms_output.put_line('Aceasta editura nu are manager!');
    DBMS_OUTPUT.PUT_LINE('--------------------');

  ELSE

  SELECT office_name
  INTO v_office
  FROM employees
  WHERE employee_id = i.boss_id;

  OPEN c(v_office);
  LOOP
    FETCH c INTO v_nume_ang, v_cod_editura;
    EXIT WHEN c%NOTFOUND;
    IF v_cod_editura = i.publisher_id THEN
      dbms_output.put_line(v_nume_ang);
    END IF;
  END LOOP;
```

```
  DBMS_OUTPUT.PUT_LINE('-------------------');
  CLOSE C;
  END IF;
END LOOP;
END p7;
/

BEGIN
p7();
end;
/
```



8. Definiti un subprogram stocat de tip functie care să utilizeze 3 dintre tabelele definite.

Tratati toate exceptiile care pot apărea. Apelati subprogramul astfel încât să evidentiati toate cazurile tratate.

--functie ce returneaza cartea cea mai scumpa a unui autor dat ca parametru

```
CREATE OR REPLACE FUNCTION p8 (v_author authors.name%type)
RETURN books.title%type IS
max_price float := 0;
book_title VARCHAR(20);
v_cod INT;
v_title VARCHAR(20);
v_pret float;

no_authors exception;
no_books exception;
too_many_authors exception;

v_nr_autori INT;
v_nr_carti INT;

BEGIN

SELECT count(name)
INTO v_nr_autori
FROM authors
WHERE UPPER(name) = UPPER(v_author);

IF v_nr_autori = 0 THEN
  RAISE no_authors;
END IF;

IF v_nr_autori > 1 THEN
```

```
  RAISE too_many_authors;
END IF;

SELECT author_id
INTO v_cod
FROM authors
WHERE UPPER(name) = UPPER (v_author);

SELECT count(book_id)
INTO v_nr_carti
FROM BOOKS_AUTHORS
WHERE author_id = v_cod;

IF v_nr_carti = 0 THEN
  RAISE no_books;
END IF;

FOR i IN (SELECT book_id FROM BOOKS_AUTHORS WHERE author_id = v_cod)
LOOP
    SELECT title, price
    INTO v_title, v_pret
    FROM BOOKS
    WHERE book_id = i.book_id;

    IF v_pret > max_price THEN
      max_price := v_pret;
      book_title := v_title;
    END IF;
END LOOP;
-- dbms_output.put_line(v_author||' '||book_title||' '||max_price);
  return book_title;
```

```
EXCEPTION
  WHEN no_authors THEN
    return 'Nu exista autori cu numele dat';
  WHEN too_many_authors THEN
    return 'Exista mai mult de 2 autori cu acest nume!';

  WHEN no_books THEN
    return 'Autorul introdus nu are nicio carte!';
END p8;
/

begin
dbms_output.put_line(p8('Neil Gaiman'));
dbms_output.put_line(p8('Hector Malot'));
dbms_output.put_line(p8('Charles Dickens'));
end;
/
```

```
EXCEPTION
  WHEN no_authors THEN
    return 'Nu exista autori cu numele dat';
  WHEN too_many_authors THEN
    return 'Exista mai mult de 2 autori cu acest nume!';

  WHEN no_books THEN
    return 'Autorul introdus nu are nicio carte!';
END p8;
/

begin
dbms_output.put_line(p8('Neil Gaiman')); -- 2 carti: art matters
dbms_output.put_line(p8('Hector Malot')); --2 autori cu acelasi nume
dbms_output.put_line(p8('Charles Dickens')); --nicio carte pentru autor
end;
/
```

Script Output | Query Result | Task completed in 0 seconds

```
FUNCTION P8 compiled
anonymous block completed
```

Messages - Log

Dbms Output | Buffer Size: 20000

```
Art matters
Exista mai mult de 2 autori cu acest nume!
Autorul introdus nu are nicio carte!
```

proiect

9. Definiți un subprogram stocat de tip procedură care să utilizeze 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

--Afisati codul,titlul, nr de pagini, tema principala, nr de autori pentru cartile publicate la editurile cu sediul intr-un oras dat ca parametru.

--BOOKS, BOOKS_AUTHORS, AUTHORS, GENRES,PUBLISHERS,LOCATIONS

CREATE OR REPLACE PROCEDURE p9 (nume_oras locations.city%type default 'Bucuresti')

IS

CURSOR edituri IS --cursor ce returneaza codurile editurilor din Bucuresti (LOCATIONS,PUBLISHERS)

SELECT publisher_id, name

FROM publishers p, locations l

WHERE p.location_id = l.location_id

AND upper(l.city) = upper(nume_oras);

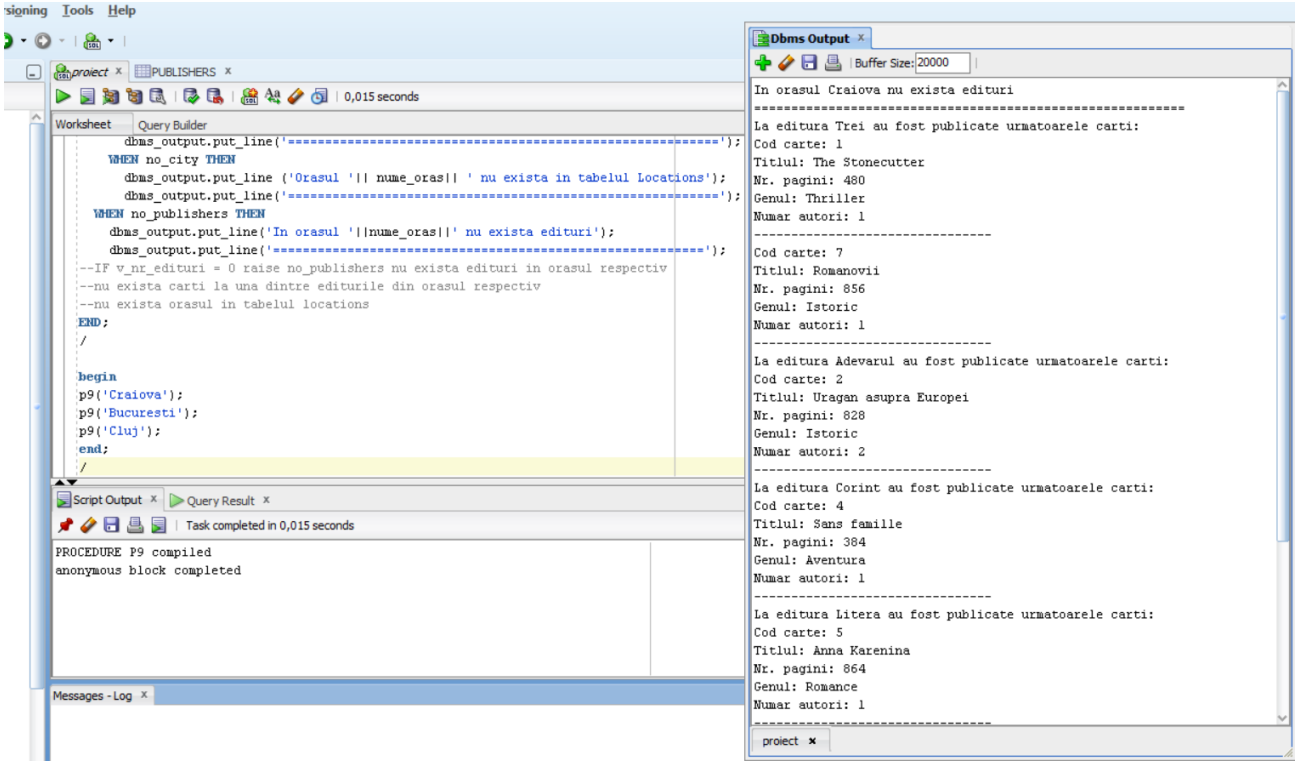
CURSOR info_books(v_cod_editura publishers.publisher_id%type) IS --cursor ce intoarce informatii despre carti (BOOKS,GENRES)

```
SELECT b.book_id,b.title, b.number_of_pages, g.name
FROM books b, genres g
WHERE b.main_theme_id = g.id
AND b.publisher_id = v_cod_editura;

v_cod_editura publishers.publisher_id%type;
v_nume_editura publishers.name%type;
v_nr_edituri INT :=0;
v_cod_carte books.book_id%type;
v_titlu books.title%type;
v_no_pages INT;
v_count_books INT;
v_theme genres.name%type;
v_nr_carti_editura INT; --numarul de carti dintr-o editura
exista_oras INT;

no_publishers exception;
no_city exception;
no_books exception;



BEGIN

  SELECT count(location_id)
  INTO exista_oras
  FROM locations
  WHERE upper(city) = upper(nume_oras);

  IF exista_oras = 0 THEN
    RAISE no_city;
```

```
END IF;

SELECT count(publisher_id)
INTO v_nr_edituri
FROM publishers p, locations l
WHERE p.location_id = l.location_id
AND upper(city) = upper(nume_oras);

IF v_nr_edituri = 0 THEN
  RAISE no_publishers;
END IF;


OPEN edituri;
LOOP
FETCH edituri INTO v_cod_editura, v_nume_editura;
EXIT WHEN edituri%notfound;


  SELECT count(book_id)
  INTO v_nr_carti_editura
  FROM books
  WHERE publisher_id = v_cod_editura;

  IF v_nr_carti_editura = 0 THEN
    RAISE no_books;
  END IF;

  dbms_output.put_line('La editura '||v_nume_editura||' au fost publicate urmatoarele carti:');
  OPEN info_books(v_cod_editura);
  LOOP
```

```
    FETCH info_books INTO v_cod_carte, v_titlu, v_no_pages, v_theme;
    exit when info_books%notfound;

    SELECT count(book_id)
    INTO v_count_books
    FROM books_authors
    WHERE book_id = v_cod_carte;

    dbms_output.put_line('Cod carte: '||v_cod_carte);
    dbms_output.put_line('Titlul: '|| v_titlu);
    dbms_output.put_line('Nr. pagini: '||v_no_pages);
    dbms_output.put_line('Genul: '|| v_theme);
    dbms_output.put_line('Numar autori: '|| v_count_books);
    dbms_output.put_line('---------------------------------');

    END LOOP;
    CLOSE info_books;



  END LOOP;
  CLOSE edituri;



  EXCEPTION
    WHEN no_books THEN
      dbms_output.put_line('La editura '||v_nume_editura|| ' nu au fost publicate carti');

dbms_output.put_line('==================================================
========');
    WHEN no_city THEN
      dbms_output.put_line ('Orasul '|| nume_oras|| ' nu exista in tabelul Locations');
```

```
dbms_output.put_line('==================================================
========');
  WHEN no_publishers THEN
    dbms_output.put_line('In orasul '||nume_oras||' nu exista edituri');

dbms_output.put_line('==================================================
========');
--IF v_nr_edituri = 0 raise no_publishers nu exista edituri in orasul respectiv
--nu exista carti la una dintre editurile din orasul respectiv
--nu exista orasul in tabelul locations
END p9;
/

begin
p9('Craiova');
p9('Bucuresti');
p9('Cluj');
end;
/
```

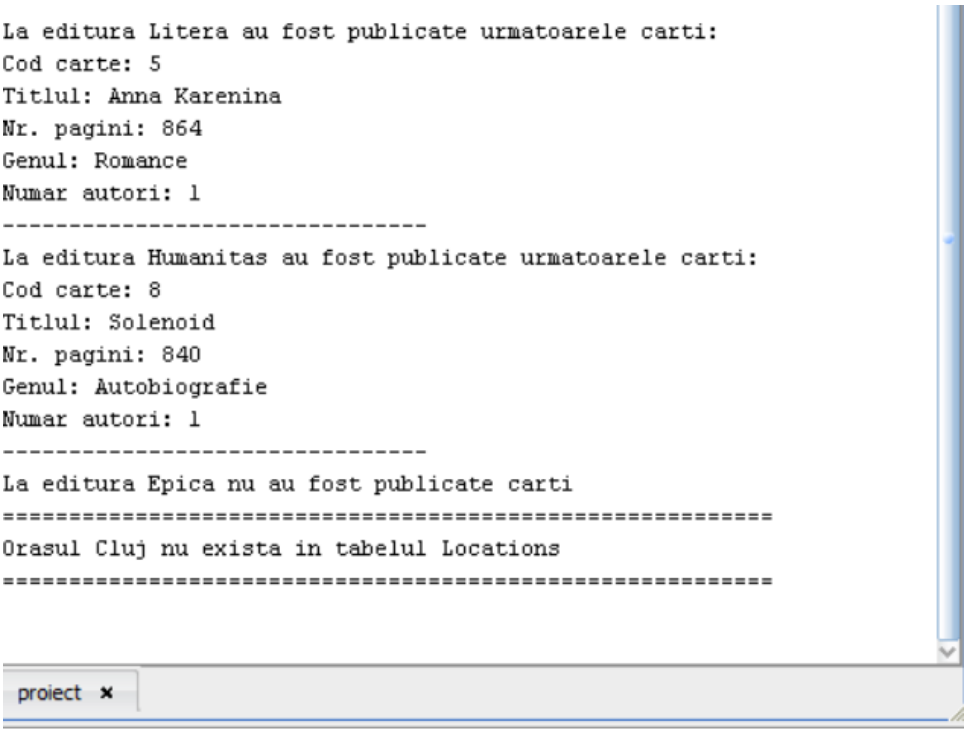

signing Tools Help
proiect
PUBLISHERS
0,015 seconds
Worksheet
Query Builder
dbms_output.put_line('==========');
WHEN no_city THEN
dbms_output.put_line ('Orasul '|| nume_oras|| ' nu exista in tabelul Locations');
dbms_output.put_line('==========');
WHEN no_publishers THEN
dbms_output.put_line('In orasul '||nume_oras||' nu exista edituri');
dbms_output.put_line('==========');
--IF v_nr_edituri = 0 raise no_publishers nu exista edituri in orasul respectiv
--nu exista carti la una dintre edituriile din orasul respectiv
--nu exista orasul in tabelul locations
END;
/
begin
p9('Craiova');
p9('Bucuresti');
p9('Cluj');
end;
/
Script Output
Query Result
Task completed in 0,015 seconds
PROCEDURE P9 compiled
anonymous block completed
Messages - Log
Dbms Output
Buffer Size: 20000
In orasul Craiova nu exista edituri
==========
La editura Trei au fost publicate urmatoarele carti:
Cod carte: 1
Titlul: The Stonecutter
Nr. pagini: 480
Genul: Thriller
Numar autori: 1

Cod carte: 7
Titlul: Romanovii
Nr. pagini: 856
Genul: Istoric
Numar autori: 1

La editura Adevarul au fost publicate urmatoarele carti:
Cod carte: 2
Titlul: Uragan asupra Europei
Nr. pagini: 828
Genul: Istoric
Numar autori: 2

La editura Corint au fost publicate urmatoarele carti:
Cod carte: 4
Titlul: Sans famille
Nr. pagini: 384
Genul: Aventura
Numar autori: 1

La editura Litera au fost publicate urmatoarele carti:
Cod carte: 5
Titlul: Anna Karenina
Nr. pagini: 864
Genul: Romance
Numar autori: 1
---
proiect


La editura Litera au fost publicate urmatoarele carti:
Cod carte: 5
Titlul: Anna Karenina
Nr. pagini: 864
Genul: Romance
Numar autori: 1

La editura Humanitas au fost publicate urmatoarele carti:
Cod carte: 8
Titlul: Solenoid
Nr. pagini: 840
Genul: Autobiografie
Numar autori: 1

La editura Epica nu au fost publicate carti
==========
Orasul Cluj nu exista in tabelul Locations
==========
proiect

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

--Definiti un trigger care sa permita modificari asupra tabelei employees doar in ziua destinata administrarii bazei de date (ultima sambata din luna curenta)

```
CREATE OR REPLACE TRIGGER trig_9
BEFORE UPDATE OR INSERT OR DELETE ON employees
DECLARE
v_date VARCHAR2(11);
begin
SELECT TO_CHAR(NEXT_DAY(LAST_DAY(SYSDATE) - INTERVAL '7' DAY,6),'DD-MM-YYYY')
INTO v_date
FROM   DUAL;

IF (TO_CHAR(SYSDATE,'DD-MM-YYYY') != v_date) then
raise_application_error(-20001,'Nu puteti modifica tabelul in aceasta data!');
end if;
end;
/

insert into employees(employee_id,name,office_name,publisher_id) values(39,'Dragnea Luminita','Grafica',2);
delete from employees where employee_id = 1;
```

```
DECLARE
v_date VARCHAR2(11);
begin
SELECT TO_CHAR(NEXT_DAY(LAST_DAY(SYSDATE) - INTERVAL '7' DAY,6),'DD-MM-YYYY')
INTO v_date
FROM   DUAL;

IF (TO_CHAR(SYSDATE,'DD-MM-YYYY') != v_date) then
raise_application_error(-20001,'Nu puteti modifica tabelul in aceasta data!');
end if;
end;
/

insert into employees(employee_id,name,office_name,publisher_id) values(39,'Dragnea Luminita','Grafica',2);
delete from employees where employee_id = 1;
```

Script Output | Query Result

Task completed in 0,02 seconds

```
TRIGGER TRIG_9 compiled
Error starting at line 432 in command:
insert into employees(employee_id,name,office_name,publisher_id) values(39,'Dragnea Luminita','Grafica',2)
Error report:
SQL Error: ORA-20001: Nu puteti modifica tabelul in aceasta data!
ORA-06512: at "DENISAIORDACHE.TRIG_9", line 9
ORA-04088: error during execution of trigger 'DENISAIORDACHE.TRIG_9'

Error starting at line 433 in command:
delete from employees where employee_id = 1
Error report:
SQL Error: ORA-20001: Nu puteti modifica tabelul in aceasta data!
ORA-06512: at "DENISAIORDACHE.TRIG_9", line 9
ORA-04088: error during execution of trigger 'DENISAIORDACHE.TRIG_9'
```

11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.

--Definiti un declansator la nivel de linie care sa nu permita micsorarea pretului cartilor din anticariat cu un procent care sa depaseasca valoarea 5.

```
CREATE OR REPLACE TRIGGER trig_10

BEFORE UPDATE OF price ON BOOKS

FOR EACH ROW

BEGIN

IF (:NEW.price < :OLD.price*0.95) THEN

RAISE_APPLICATION_ERROR(-20002,'Pretul cartilor nu poate fi micsorat mai mult de 5%');

END IF;

END;

/


UPDATE BOOKS

SET price = price - price*0.1;
```

```
--Definiti un declansator la nivel de linie care sa nu permita micsorarea pretului cartilor din anticariat cu un procen
CREATE OR REPLACE TRIGGER trig_10
BEFORE UPDATE OF price ON BOOKS
FOR EACH ROW
BEGIN
IF (:NEW.price < :OLD.price*0.95) THEN
RAISE_APPLICATION_ERROR(-20002,'Pretul cartilor nu poate fi micsorat mai mult de 5%');
END IF;
END;
/

UPDATE BOOKS
SET price = price - price*0.1;
```

Script Output × | Query Result ×

Task completed in 0,007 seconds

```
TRIGGER TRIG_10 compiled
Error starting at line 446 in command:
UPDATE BOOKS
SET price = price - price*0.1
Error report:
SQL Error: ORA-20002: Pretul cartilor nu poate fi micsorat mai mult de 5%
ORA-06512: at "DENISAIORDACHE.TRIG_10", line 3
ORA-04088: error during execution of trigger 'DENISAIORDACHE.TRIG_10'
```

12. Definiți un trigger de tip LDD. Declanșați trigger-ul.

--Definiti un trigger care sa introduca informatii legate de actiunile LDD produse asupra bazei de date intr-un tabel.

CREATE TABLE INFO_DIO (

utilizator VARCHAR2(30),

nume_bd VARCHAR2(50),

actiune VARCHAR2(20),

nume_obiect VARCHAR2(30),

data DATE);

CREATE OR REPLACE TRIGGER trig_11

AFTER CREATE OR DROP OR ALTER ON SCHEMA

BEGIN

INSERT INTO INFO_DIO VALUES (SYS.LOGIN_USER, SYS.DATABASE_NAME, SYS.SYSEVENT, SYS.DICTIONARY_OBJ_NAME, SYSDATE);

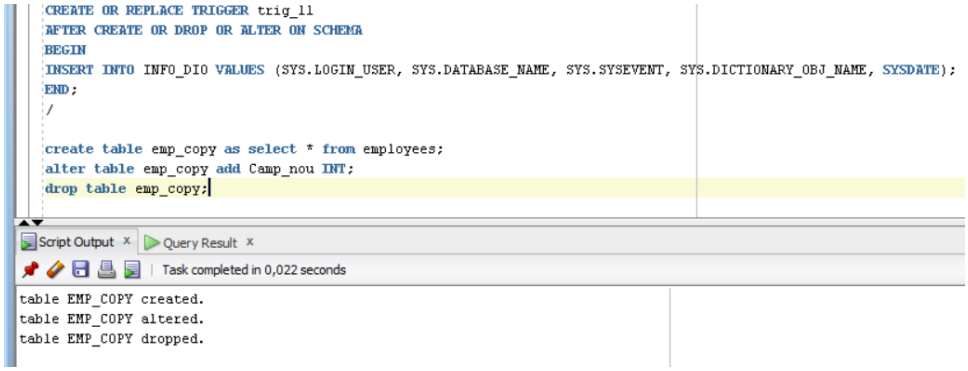
END;

/

create table emp_copy as select * from employees;

alter table emp_copy add Camp_nou INT;

drop table emp_copy;



proiect × INFO_DIO ×

Columns | Data | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partit

Sort.. | Filter:

	UTILIZATOR	NUME_BD	ACTIUNE	NUME_OBIECT	DATA
15	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
16	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
17	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
18	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
19	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
20	DENISAIORDACHE	XE	ALTER	TRIG_9	08-01-2021
21	DENISAIORDACHE	XE	ALTER	TRIG_9	08-01-2021
22	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
23	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
24	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
25	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
26	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
27	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
28	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
29	DENISAIORDACHE	XE	ALTER	TRIG_9	08-01-2021
30	DENISAIORDACHE	XE	ALTER	TRIG_9	08-01-2021
31	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
32	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
33	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
34	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
35	DENISAIORDACHE	XE	CREATE	P7	08-01-2021
36	DENISAIORDACHE	XE	CREATE	P8	08-01-2021
37	DENISAIORDACHE	XE	CREATE	P8	08-01-2021
38	DENISAIORDACHE	XE	CREATE	P8	08-01-2021
39	DENISAIORDACHE	XE	CREATE	P9	08-01-2021
40	DENISAIORDACHE	XE	CREATE	TRIG_9	08-01-2021
41	DENISAIORDACHE	XE	CREATE	TRIG_10	08-01-2021
42	DENISAIORDACHE	XE	CREATE	EMP_COPY	08-01-2021
43	DENISAIORDACHE	XE	ALTER	EMP_COPY	08-01-2021
44	DENISAIORDACHE	XE	DROP	EMP_COPY	08-01-2021

13. Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.

```
CREATE or REPLACE PACKAGE pachet_13 AS
type vector is varray(100) of number;
PROCEDURE p6 (v_oras locations.city%type default 'Bucuresti');
PROCEDURE p7  ;
FUNCTION p8 (v_author authors.name%type)
RETURN books.title%type;
PROCEDURE p9 (nume_oras locations.city%type default 'Bucuresti');
END pachet_13;
/
```

```
CREATE OR REPLACE PACKAGE BODY pachet_13 AS
PROCEDURE p6 (v_oras locations.city%type default 'Bucuresti')
IS
v_coduri vector := vector();
v_denumire publishers.name%type;
exc exception;

BEGIN

SELECT location_id
BULK COLLECT INTO v_coduri
FROM locations
WHERE city = v_oras;

IF v_coduri.count = 0 THEN
  raise exc;

END IF;

dbms_output.put_line('Lista editurilor din orasul '||v_oras||':');

FOR i IN 1..v_coduri.count LOOP
  select name
  into v_denumire
  from publishers
  where location_id = v_coduri(i);
  dbms_output.put_line(v_denumire);

END LOOP;
dbms_output.put_line('------------------');
```

```
exception

WHEN exc THEN
  DBMS_OUTPUT.PUT_LINE('Nu exista edituri cu sediul in ' || v_oras);
END p6;

PROCEDURE p7
IS

CURSOR c (v_office employees.office_name%type)
IS SELECT name,publisher_id
    FROM employees
    WHERE office_name = v_office;

v_nume_ang employees.name%type;
v_cod_editura INT;
v_cod_sef INT;
v_office employees.office_name%type;
v_manager INT;


BEGIN

FOR i IN (SELECT publisher_id, name, boss_id FROM publishers) LOOP

  SELECT COUNT(boss_id)
  INTO v_manager
  FROM publishers
  WHERE publisher_id = i.publisher_id;

  DBMS_OUTPUT.PUT_LINE('Editura: '|| i.name);
```

```
  IF v_manager = 0 THEN

    dbms_output.put_line('Aceasta editura nu are manager!');
    DBMS_OUTPUT.PUT_LINE('-------------------');

  ELSE

  SELECT office_name
  INTO v_office
  FROM employees
  WHERE employee_id = i.boss_id;

  OPEN c(v_office);
  LOOP
    FETCH c INTO v_nume_ang, v_cod_editura;
    EXIT WHEN c%NOTFOUND;
    IF v_cod_editura = i.publisher_id THEN
      dbms_output.put_line(v_nume_ang);
    END IF;
  END LOOP;
  DBMS_OUTPUT.PUT_LINE('-------------------');
  CLOSE C;
  END IF;
END LOOP;
END p7;

FUNCTION p8 (v_author authors.name%type)
RETURN books.title%type IS

max_price float := 0;
```

```
book_title VARCHAR(20);
v_cod INT;
v_title VARCHAR(20);
v_pret float;
no_authors exception;
no_books exception;
too_many_authors exception;
v_nr_autori INT;
v_nr_carti INT;


BEGIN



SELECT count(name)
INTO v_nr_autori
FROM authors
WHERE UPPER(name) = UPPER(v_author);

IF v_nr_autori = 0 THEN
  RAISE no_authors;
END IF;

IF v_nr_autori > 1 THEN
  RAISE too_many_authors;
END IF;

SELECT author_id
INTO v_cod
FROM authors
```

```
WHERE UPPER(name) = UPPER (v_author);

SELECT count(book_id)
INTO v_nr_carti
FROM BOOKS_AUTHORS
WHERE author_id = v_cod;


IF v_nr_carti = 0 THEN
  RAISE no_books;
END IF;



FOR i IN (SELECT book_id FROM BOOKS_AUTHORS WHERE author_id = v_cod)
LOOP
    SELECT title, price
    INTO v_title, v_pret
    FROM BOOKS
    WHERE book_id = i.book_id;

    IF v_pret > max_price THEN
      max_price := v_pret;
      book_title := v_title;
    END IF;
END LOOP;

return book_title;

EXCEPTION
```

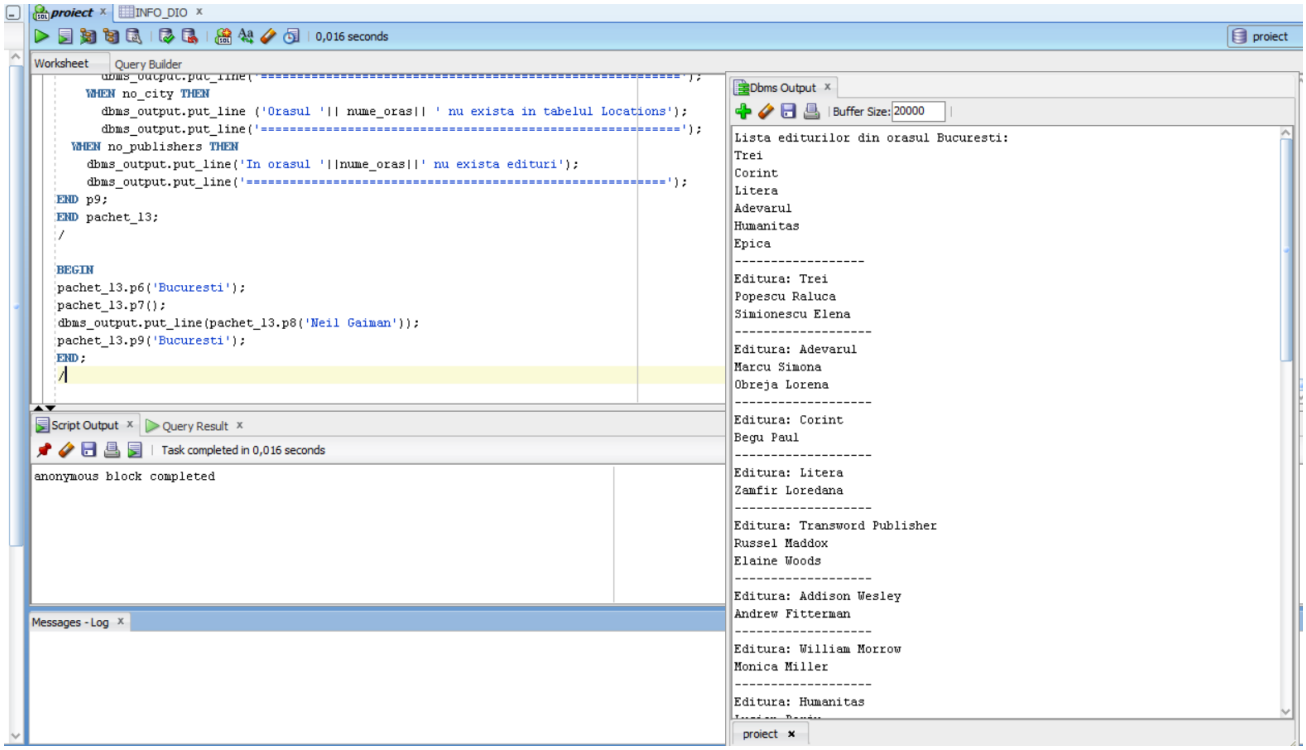
```
  WHEN no_authors THEN
    return 'Nu exista autori cu numele dat';
  WHEN too_many_authors THEN
    return 'Exista mai mult de 2 autori cu acest nume!';

  WHEN no_books THEN
    return 'Autorul introdus nu are nicio carte!';
END p8;

PROCEDURE p9 (nume_oras locations.city%type default 'Bucuresti')
IS
CURSOR edituri IS  --cursor ce returneaza codurile editurilor din Bucuresti
(LOCATIONS,PUBLISHERS)
SELECT publisher_id, name
FROM publishers p, locations l
WHERE p.location_id = l.location_id
AND upper(l.city) = upper(nume_oras);

CURSOR info_books(v_cod_editura publishers.publisher_id%type) IS
SELECT b.book_id,b.title, b.number_of_pages, g.name
FROM books b, genres g
WHERE b.main_theme_id = g.id
AND b.publisher_id = v_cod_editura;

v_cod_editura publishers.publisher_id%type;
v_nume_editura publishers.name%type;
v_nr_edituri INT :=0;
v_cod_carte books.book_id%type;
v_titlu books.title%type;
v_no_pages INT;
v_count_books INT;
```

```
v_theme genres.name%type;
v_nr_carti_editura INT;
exista_oras INT;

no_publishers exception;
no_city exception;
no_books exception;



BEGIN

  SELECT count(location_id)
  INTO exista_oras
  FROM locations
  WHERE upper(city) = upper(nume_oras);

  IF exista_oras = 0 THEN
    RAISE no_city;
  END IF;

  SELECT count(publisher_id)
  INTO v_nr_edituri
  FROM publishers p, locations l
  WHERE p.location_id = l.location_id
  AND upper(city) = upper(nume_oras);

  IF v_nr_edituri = 0 THEN
    RAISE no_publishers;
  END IF;
```

```
OPEN edituri;
LOOP
FETCH edituri INTO v_cod_editura, v_nume_editura;
EXIT WHEN edituri%notfound;


  SELECT count(book_id)
  INTO v_nr_carti_editura
  FROM books
  WHERE publisher_id = v_cod_editura;

  IF v_nr_carti_editura = 0 THEN
    RAISE no_books;
  END IF;

  dbms_output.put_line('La editura '||v_nume_editura||' au fost publicate urmatoarele carti:');
  OPEN info_books(v_cod_editura);
  LOOP
  FETCH info_books INTO v_cod_carte, v_titlu, v_no_pages, v_theme;
  exit when info_books%notfound;

  SELECT count(book_id)
  INTO v_count_books
  FROM books_authors
  WHERE book_id = v_cod_carte;

  dbms_output.put_line('Cod carte: '||v_cod_carte);
  dbms_output.put_line('Titlul: '|| v_titlu);
  dbms_output.put_line('Nr. pagini: '||v_no_pages);
  dbms_output.put_line('Genul: '|| v_theme);
```

```
    dbms_output.put_line('Numar autori: '|| v_count_books);
    dbms_output.put_line('--------------------------------');

    END LOOP;
    CLOSE info_books;



  END LOOP;
  CLOSE edituri;



  EXCEPTION
    WHEN no_books THEN
      dbms_output.put_line('La editura '||v_nume_editura|| ' nu au fost publicate carti');

dbms_output.put_line('==================================================
========');
    WHEN no_city THEN
      dbms_output.put_line ('Orasul '|| nume_oras|| ' nu exista in tabelul Locations');

dbms_output.put_line('==================================================
========');
  WHEN no_publishers THEN
    dbms_output.put_line('In orasul '||nume_oras||' nu exista edituri');

dbms_output.put_line('==================================================
========');
END p9;
END pachet_13;
/


BEGIN
pachet_13.p6('Bucuresti');
```

```
pachet_13.p7();
dbms_output.put_line(pachet_13.p8('Neil Gaiman'));
pachet_13.p9('Bucuresti');
END;
/
```



Dbms Output

Buffer Size: 20000

```
Editura: Humanitas
Lucian Perju
Daniela Maxim
-------------------
Editura: Epica
Aceasta editura nu are manager!
-------------------
Art matters
La editura Trei au fost publicate urmatoarele carti:
Cod carte: 1
Titlul: The Stonecutter
Nr. pagini: 480
Genul: Thriller
Numar autori: 1
--------------------------------
Cod carte: 7
Titlul: Romanovii
Nr. pagini: 856
Genul: Istoric
Numar autori: 1
--------------------------------
La editura Adevarul au fost publicate urmatoarele carti:
Cod carte: 2
Titlul: Uragan asupra Europei
Nr. pagini: 828
Genul: Istoric
Numar autori: 2
--------------------------------
La editura Corint au fost publicate urmatoarele carti:
Cod carte: 4
Titlul: Sans famille
Nr. pagini: 384
Genul: Aventura
```

proiect

Dbms Output

Buffer Size: 20000

```
Numar autori: 1
--------------------------------
La editura Adevarul au fost publicate urmatoarele carti:
Cod carte: 2
Titlul: Uragan asupra Europei
Nr. pagini: 828
Genul: Istoric
Numar autori: 2
--------------------------------
La editura Corint au fost publicate urmatoarele carti:
Cod carte: 4
Titlul: Sans famille
Nr. pagini: 384
Genul: Aventura
Numar autori: 1
--------------------------------
La editura Litera au fost publicate urmatoarele carti:
Cod carte: 5
Titlul: Anna Karenina
Nr. pagini: 864
Genul: Romance
Numar autori: 1
--------------------------------
La editura Humanitas au fost publicate urmatoarele carti:
Cod carte: 8
Titlul: Solenoid
Nr. pagini: 840
Genul: Autobiografie
Numar autori: 1
--------------------------------
La editura Epica nu au fost publicate carti
============================================================
```

proiect