

Librarie

Munteanu Vlad Paul - 242

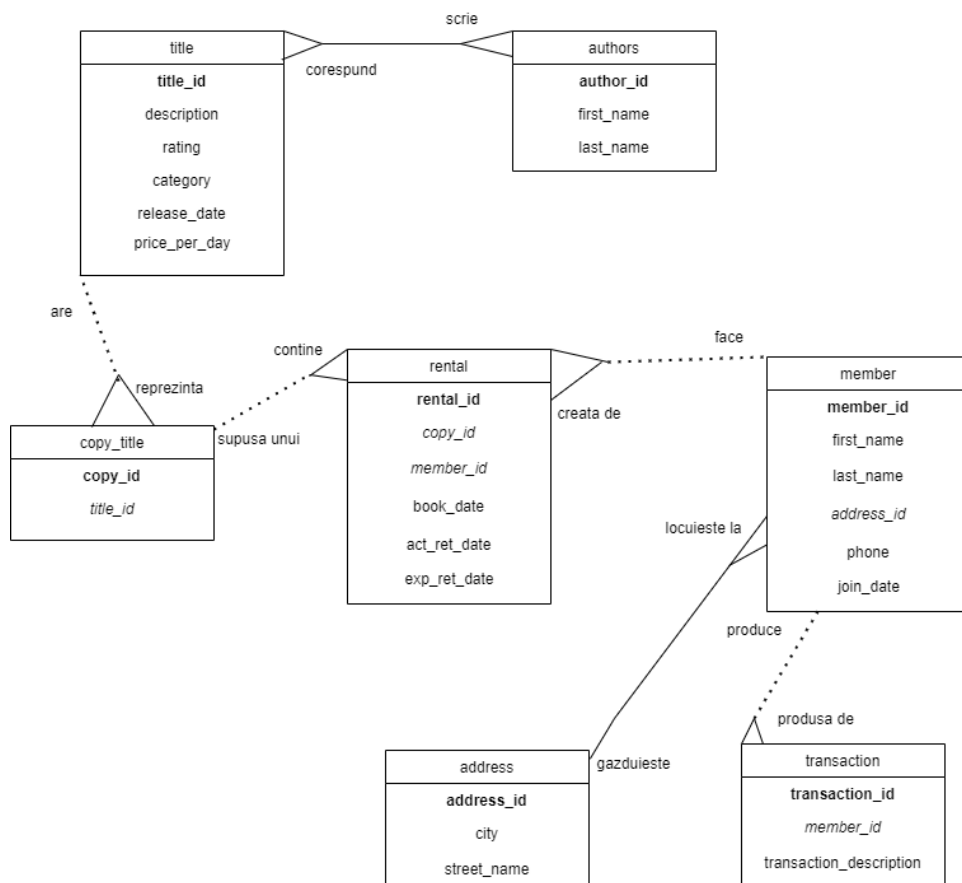
(Nota 5-6)

Proiect SGBD – Librarie

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

Baza de date a unei librării care le permite clienților să devină membri și să împrumute cărți pe o durată de zile aleasă de ei. Pe lângă cerințele proiectului, mai există funcții precum adăugarea copiilor, înscrierea membrilor, adăugarea balanțelor... pe care le voi numi la final care poartă un rol critic în funcționalitatea bazei de date.

2. Realizați diagrama entitate-relație (ERD).



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


```
CREATE TABLE author (  
    author_id NUMBER(10) CONSTRAINT author_pk PRIMARY KEY,  
    first_name VARCHAR2(20) CONSTRAINT author_first_name_nn NOT  
NULL,  
    last_name VARCHAR2(20) CONSTRAINT author_last_name_nn NOT  
NULL  
);
```

```
CREATE TABLE title_author(  
    title_id number(10) constraint title_author_title_fk  
REFERENCES title(title_id) ON DELETE CASCADE,  
    author_id number(10) constraint title_author_author_fk  
REFERENCES author(author_id) ON DELETE CASCADE,  
    CONSTRAINT title_author_pk PRIMARY KEY(title_id,author_id)  
);
```

```
CREATE TABLE member(  
    member_id number(10) constraint member_pk PRIMARY KEY,  
    first_name varchar2(20) constraint member_first_name_nn NOT  
NULL,  
    last_name varchar2(20) constraint member_last_name_nn NOT  
NULL,  
    address_id number(10) constraint member_address_fk  
REFERENCES address(address_id) ON DELETE CASCADE,  
    phone_number varchar2(20) constraint member_phone_number_nn  
NOT NULL,  
    join_date date constraint member_join_date_nn NOT NULL,  
    balance number(10) default 0  
);
```

```
CREATE TABLE address(  
    address_id number(10) constraint address_pk PRIMARY KEY,  
    city varchar2(20) constraint address_city_nn not null,  
    street_name varchar2(40)  
);
```

```
CREATE TABLE copy_title(  
    copy_id number(10),
```

```

    title_id number(10) constraint copy_title_title_fk
REFERENCES title(title_id) ON DELETE CASCADE,

    CONSTRAINT copy_title_pk PRIMARY KEY(copy_id)
);

CREATE TABLE rental(

    rental_id number(10),

    book_date date,

    copy_id constraint rental_copy_id_fk REFERENCES
copy_title(copy_id) ON DELETE CASCADE,

    member_id constraint rental_member_id_fk REFERENCES
member(member_id) ON DELETE CASCADE,

    act_ret_date date default null,

    exp_ret_date date,

    CONSTRAINT rental_pk PRIMARY KEY (rental_id)
);

CREATE TABLE transaction(

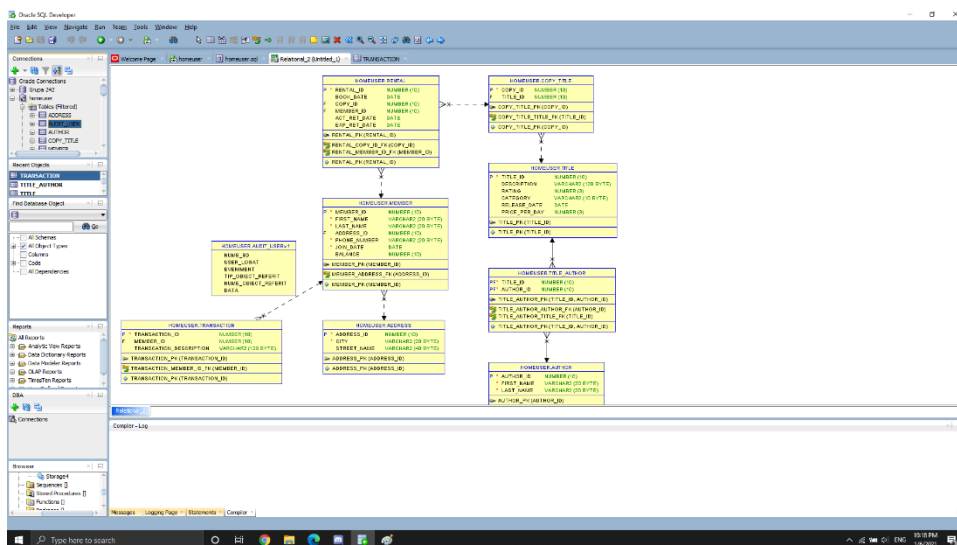
    transaction_id number(10),

    member_id constraint transaction_member_id_fk references
member(member_id) on delete cascade,

    transcation_description varchar2(128)
);

```

Oracle:



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

Text:

```
INSERT INTO author VALUES
(seq_author.nextval, 'Vlad', 'Munteanu');

INSERT INTO author VALUES
(seq_author.nextval, 'Ion', 'Creanga');

INSERT INTO author VALUES
(seq_author.nextval, 'Mihai', 'Eminescu');

INSERT INTO author VALUES (seq_author.nextval, 'Bill', 'Gates');

INSERT INTO author VALUES
(seq_author.nextval, 'Ion', 'Creanga');

INSERT INTO author VALUES
(seq_author.nextval, 'Mircea', 'Eliade');


INSERT INTO title VALUES (seq_title.nextval, 'Hunting
Bears', 65, 'Hunting', TO_DATE('2020-03-20', 'YYYY-MM-DD'), 2);

INSERT INTO title VALUES (seq_title.nextval, 'Cooking Vegan
Food', 89, 'Cooking', TO_DATE('2015-01-25', 'YYYY-MM-DD'), 7);

INSERT INTO title VALUES (seq_title.nextval, 'Using javascript
in programming', 65, 'IT', TO_DATE('2010-04-02', 'YYYY-MM-DD'), 9);

INSERT INTO title VALUES (seq_title.nextval, 'Using c++ in
programming', 40, 'IT', TO_DATE('2005-01-01', 'YYYY-MM-DD'), 9);

INSERT INTO title VALUES (seq_title.nextval, 'Using java in
programming', 50, 'IT', TO_DATE('2016-01-03', 'YYYY-MM-DD'), 9);

INSERT INTO title VALUES (seq_title.nextval, 'Using python in
programming', 75, 'IT', TO_DATE('2019-04-02', 'YYYY-MM-DD'), 9);

INSERT INTO title VALUES (seq_title.nextval, 'Philosphy Starter
Pack', 80, 'Philosophy', TO_DATE('2017-12-15', 'YYYY-MM-DD'), 9);

INSERT INTO title VALUES (seq_title.nextval, 'Cooking
Desserts', 50, 'Cooking', TO_DATE('2012-01-25', 'YYYY-MM-DD'), 8);

INSERT INTO title VALUES (seq_title.nextval, 'Surviving on an
island', 65, 'Survival', TO_DATE('2020-01-12', 'YYYY-MM-DD'), 3);

INSERT INTO title VALUES (seq_title.nextval, 'Poetry in Mihai
Eminescu Style', 89, 'Poetry', TO_DATE('2000-02-10', 'YYYY-MM-
DD'), 2);
```

```
INSERT INTO member
(member_id,first_name,last_name,address_id,phone_number,join_d
ate) VALUES
(seq_member.nextval,'Cezara','Grigoras',1,'0727587574',SYSDATE
);
```

```
INSERT INTO member
(member_id,first_name,last_name,address_id,phone_number,join_d
ate) VALUES
(seq_member.nextval,'Gabi','Flavius',2,'0721341296',TO_DATE('2
010-04-02','YYYY-MM-DD'));
```

```
INSERT INTO member
(member_id,first_name,last_name,address_id,phone_number,join_d
ate) VALUES
(seq_member.nextval,'Edi','Sujon',2,'0761413122',TO_DATE('2015
-06-15','YYYY-MM-DD'));
```

```
INSERT INTO member
(member_id,first_name,last_name,address_id,phone_number,join_d
ate) VALUES
(seq_member.nextval,'Mario','Marian',3,'02313133',SYSDATE);
```

```
INSERT INTO copy_title values (seq_copy_title.nextval,1);
INSERT INTO copy_title values (seq_copy_title.nextval,2);
INSERT INTO copy_title values (seq_copy_title.nextval,3);
INSERT INTO copy_title values (seq_copy_title.nextval,2);
INSERT INTO copy_title values (seq_copy_title.nextval,3);
INSERT INTO copy_title values (seq_copy_title.nextval,4);
INSERT INTO copy_title values (seq_copy_title.nextval,5);
INSERT INTO copy_title values (seq_copy_title.nextval,6);
INSERT INTO copy_title values (seq_copy_title.nextval,4);
INSERT INTO copy_title values (seq_copy_title.nextval,5);
INSERT INTO copy_title values (seq_copy_title.nextval,10);
INSERT INTO copy_title values (seq_copy_title.nextval,10);
```

```
INSERT INTO title_author values (1,1);
INSERT INTO title_author values (2,2);
INSERT INTO title_author values (3,3);
INSERT INTO title_author values (4,4);
```

```
INSERT INTO title_author values (5,5);
INSERT INTO title_author values (6,3);
INSERT INTO title_author values (7,4);
INSERT INTO title_author values (8,4);
INSERT INTO title_author values (9,4);
INSERT INTO title_author values (10,5);
```

```
INSERT INTO rental values
(seq_rental.nextval,SYSDATE,1,8,TO_DATE('2020-11-15','YYYY-MM-DD'),TO_DATE('2020-11-10','YYYY-MM-DD'));
```

```
INSERT INTO rental values (seq_rental.nextval,TO_DATE('2019-05-02','YYYY-MM-DD'),5,8,SYSDATE,TO_DATE('2019-12-15','YYYY-MM-DD'));
```

```
INSERT INTO rental values (seq_rental.nextval,TO_DATE('2020-01-05','YYYY-MM-DD'),6,5,TO_DATE('2020-12-10','YYYY-MM-DD'),SYSDATE);
```

```
INSERT INTO rental
(rental_id,book_date,copy_id,member_id,exp_ret_date) values
(seq_rental.nextval,SYSDATE,4,7,TO_DATE('2021-01-10','YYYY-MM-DD'));
```

```
INSERT INTO address values
(seq_address.nextval,'Bucharest','Ion Mihalache');
```

```
INSERT INTO address (address_id,city) values
(seq_address.nextval,'Bucharest');
```

```
INSERT INTO address values
(seq_address.nextval,'Cluj','Nicolae Grigorescu');
```

```
INSERT INTO address values
(seq_address.nextval,'Constanta','Ion Mihalache');
```

```
INSERT INTO address (address_id,city) values
(seq_address.nextval,'Brasov');
```

```
insert into transaction values
(seq_transaction.nextval,5,'Member rented copy 1 of title 10');
```

```
insert into transaction values
(seq_transaction.nextval,6,'Member rented copy 1 of title 10');
```



```
insert into transaction values
(seq_transaction.nextval,7,'Member rented copy 1 of title
10');
```

Oracle:

Audit_user:

```

491 select * from audit_user;
492 --13
493 create or replace package projectSGBD as
494     type copies is table of number;
495
496     function showRented return copies;
497     function return_carti IT (id author in author_id$type) return nume_carti:

```

NUME_BD	USER_LOGAT	EVENIMENT	TIP_OBJECT_REFERIT	NUME_OBJECT_REFERIT	DATA
1 XE	HOMEUSER	CREATE	TABLE	PO	06-JAN-21 10.05.30.871000000 PM
2 XE	HOMEUSER	DROP	TABLE	PO	06-JAN-21 10.11.08.981000000 PM
3 XE	HOMEUSER	CREATE	TABLE	title	06-JAN-21 10.09.23.258000000 PM
4 XE	HOMEUSER	CREATE	TABLE	title_author	06-JAN-21 10.10.00.481000000 PM
5 XE	HOMEUSER	CREATE	TABLE	author	06-JAN-21 10.10.17.849000000 PM
6 XE	HOMEUSER	CREATE	TABLE	transaction	06-JAN-21 10.10.36.254000000 PM
7 XE	HOMEUSER	CREATE	TABLE	member	06-JAN-21 10.10.45.015000000 PM
8 XE	HOMEUSER	DROP	TRIGGER	EX_12	06-JAN-21 10.11.05.321000000 PM
9 XE	HOMEUSER	CREATE	TABLE	address	06-JAN-21 10.11.26.445000000 PM
10 XE	HOMEUSER	CREATE	TABLE	copy_title	06-JAN-21 10.11.37.578000000 PM
11 XE	HOMEUSER	CREATE	TABLE	rental	06-JAN-21 10.11.48.497000000 PM

Title:

```

138 INSERT INTO author VALUES (seq_author.nextval,'Vlad','Munteanu');
139 INSERT INTO author VALUES (seq_author.nextval,'Ion','Creanga');
140 INSERT INTO author VALUES (seq_author.nextval,'Mihai','Eminescu');
141 INSERT INTO author VALUES (seq_author.nextval,'Bill','Gates');
142 INSERT INTO author VALUES (seq_author.nextval,'Ion','Creanga');
143 INSERT INTO author VALUES (seq_author.nextval,'Mircea','Eliade');
144
145 INSERT INTO title VALUES (seq_title.nextval,'Hunting Bears',65,'Hunting',TO_DATE('2020-03-20','YYYY-MM-DD'),2);
146 INSERT INTO title VALUES (seq_title.nextval,'Cooking Vegan Food',89,'Cooking',TO_DATE('2015-01-25','YYYY-MM-DD'),7);
147 INSERT INTO title VALUES (seq_title.nextval,'Using javascript in programming',65,'IT',TO_DATE('2010-04-02','YYYY-MM-DD'),9);
148 INSERT INTO title VALUES (seq_title.nextval,'Using c++ in programming',40,'IT',TO_DATE('2005-01-01','YYYY-MM-DD'),9);
149 INSERT INTO title VALUES (seq_title.nextval,'Using java in programming',50,'IT',TO_DATE('2016-01-03','YYYY-MM-DD'),9);
150 INSERT INTO title VALUES (seq_title.nextval,'Using python in programming',75,'IT',TO_DATE('2019-04-02','YYYY-MM-DD'),9);
151 INSERT INTO title VALUES (seq_title.nextval,'Philosophy Starter Pack',80,'Philosophy',TO_DATE('2017-12-15','YYYY-MM-DD'),9);
152 INSERT INTO title VALUES (seq_title.nextval,'Cooking Desserts',50,'Cooking',TO_DATE('2012-01-25','YYYY-MM-DD'),8);
153 INSERT INTO title VALUES (seq_title.nextval,'Surviving on an island',65,'Survival',TO_DATE('2020-01-12','YYYY-MM-DD'),3);
154 INSERT INTO title VALUES (seq_title.nextval,'Poetry in Mihai Eminescu Style',89,'Poetry',TO_DATE('2000-02-10','YYYY-MM-DD'),2);
155 select * from title;
156 INSERT INTO member (member_id,first_name,last_name,address_id,phone_number,join_date) VALUES (seq_member.nextval,'Cezara','Grig');
157 INSERT INTO member (member_id,first_name,last_name,address_id,phone_number,join_date) VALUES (seq_member.nextval,'Gabi','Flaviu');

```

TITLE_ID	DESCRIPTION	RATING	CATEGORY	RELEASE_DATE	PRICE_PER_DAY
1	21 Hunting Bears	65	Hunting	20-MAR-20	2
2	22 Cooking Vegan Food	89	Cooking	25-JAN-15	7
3	23 Using javascript in programming	65	IT	02-APR-10	9
4	24 Using c++ in programming	40	IT	01-JAN-05	9
5	25 Using java in programming	50	IT	03-JAN-16	9
6	26 Using python in programming	75	IT	02-APR-19	9
7	27 Philosophy Starter Pack	80	Philosophy	15-DEC-17	9
8	28 Cooking Desserts	50	Cooking	25-JAN-12	8
9	29 Surviving on an island	65	Survival	12-JAN-20	3
10	30 Poetry in Mihai Eminescu Style	89	Poetry	10-FEB-00	2

Author:

```

189
190 INSERT INTO address values (seq_address.nextval,'Bucharest','Ion Mihalache');
191 INSERT INTO address (address_id,city) values (seq_address.nextval,'Bucharest');
192 INSERT INTO address values (seq_address.nextval,'Cluj','Nicolae Grigorescu');
193 INSERT INTO address values (seq_address.nextval,'Constanta','Ion Mihalache');
194 INSERT INTO address (address_id,city) values (seq_address.nextval,'Brasov');
195
196 insert into transaction values (seq_transaction.nextval,5,'Member rented copy 1 of title 10');
197 insert into transaction values (seq_transaction.nextval,6,'Member rented copy 1 of title 10');
198 insert into transaction values (seq_transaction.nextval,7,'Member rented copy 1 of title 10');
199
200 select *
201 from author;
202
203 -- 6.

```

AUTHOR_ID	FIRST_NAME	LAST_NAME
1	Vlad	Munteanu
2	Ion	Creanga
3	Mihai	Eminescu
4	Bill	Gates
5	Ion	Creanga
6	Mircea	Eliade

Title_Author:

```

189
190 INSERT INTO address values (seq_address.nextval,'Bucharest','Ion Mihalache');
191 INSERT INTO address (address_id,city) values (seq_address.nextval,'Bucharest');
192 INSERT INTO address values (seq_address.nextval,'Cluj','Nicolae Grigorescu');
193 INSERT INTO address values (seq_address.nextval,'Constanta','Ion Mihalache');
194 INSERT INTO address (address_id,city) values (seq_address.nextval,'Brasov');
195
196 insert into transaction values (seq_transaction.nextval,5,'Member rented copy 1 of title 10');
197 insert into transaction values (seq_transaction.nextval,6,'Member rented copy 1 of title 10');
198 insert into transaction values (seq_transaction.nextval,7,'Member rented copy 1 of title 10');
199
200 select *
201 from title_author;
202
203 -- 6.

```

TITLE_ID	AUTHOR_ID
1	1
2	2
3	3
4	4
5	5
6	3
7	4
8	4
9	4
10	5

Copy_Title:

```

189
190 INSERT INTO address values (seq_address.nextval,'Bucharest','Ion Mihalache');
191 INSERT INTO address (address_id,city) values (seq_address.nextval,'Bucharest');
192 INSERT INTO address values (seq_address.nextval,'Cluj','Nicolae Grigorescu');
193 INSERT INTO address values (seq_address.nextval,'Constanta','Ion Mihalache');
194 INSERT INTO address (address_id,city) values (seq_address.nextval,'Brasov');
195
196 insert into transaction values (seq_transaction.nextval,5,'Member rented copy 1 of title 10');
197 insert into transaction values (seq_transaction.nextval,6,'Member rented copy 1 of title 10');
198 insert into transaction values (seq_transaction.nextval,7,'Member rented copy 1 of title 10');
199
200 select *
201 from copy_title;
202
203 -- 6.

```

	COPY_ID	TITLE_ID
1	1	1
2	2	2
3	3	3
4	4	2
5	5	3
6	6	4
7	7	5
8	8	6
9	9	4
10	10	5
11	11	10
12	12	10

Rental:

```

189
190 INSERT INTO address values (seq_address.nextval,'Bucharest','Ion Mihalache');
191 INSERT INTO address (address_id,city) values (seq_address.nextval,'Bucharest');
192 INSERT INTO address values (seq_address.nextval,'Cluj','Nicolae Grigorescu');
193 INSERT INTO address values (seq_address.nextval,'Constanta','Ion Mihalache');
194 INSERT INTO address (address_id,city) values (seq_address.nextval,'Brasov');
195
196 insert into transaction values (seq_transaction.nextval,5,'Member rented copy 1 of title 10')
197 insert into transaction values (seq_transaction.nextval,6,'Member rented copy 1 of title 10')
198 insert into transaction values (seq_transaction.nextval,7,'Member rented copy 1 of title 10')
199
200 select *
201 from rental;
202
203 -- 6.

```

	RENTAL_ID	BOOK_DATE	COPY_ID	MEMBER_ID	ACT_RET_DATE	EXP_RET_DATE
1	9	04-JAN-21	1	8	15-NOV-20	10-NOV-20
2	10	02-MAY-19	5	8	04-JAN-21	15-DEC-19
3	11	05-JAN-20	6	5	10-DEC-20	04-JAN-21
4	12	04-JAN-21	4	7	(null)	10-JAN-21

Member:

```

189
190 INSERT INTO address values (seq_address.nextval,'Bucharest','Ion Mihalache');
191 INSERT INTO address (address_id,city) values (seq_address.nextval,'Bucharest');
192 INSERT INTO address values (seq_address.nextval,'Cluj','Nicolae Grigorescu');
193 INSERT INTO address values (seq_address.nextval,'Constanta','Ion Mihalache');
194 INSERT INTO address (address_id,city) values (seq_address.nextval,'Brasov');
195
196 insert into transaction values (seq_transaction.nextval,5,'Member rented copy 1 of title 10');
197 insert into transaction values (seq_transaction.nextval,6,'Member rented copy 1 of title 10');
198 insert into transaction values (seq_transaction.nextval,7,'Member rented copy 1 of title 10');
199
200 select *
201 from member;
202
203 -- 6.
204

```

MEMBER_ID	FIRST_NAME	LAST_NAME	ADDRESS_ID	PHONE_NUMBER	JOIN_DATE	BALANCE
1	5 Cezara	Grigoras	1	0727587574	04-JAN-21	0
2	6 Gabi	Flavius	2	0721341296	02-APR-10	0
3	7 Edi	Sujon	2	0761413122	15-JUN-15	0
4	8 Mario	Marian	3	02313133	04-JAN-21	0

Address:

```

192 INSERT INTO address values (seq_address.nextval,'Cluj','Nicolae Grigorescu');
193 INSERT INTO address values (seq_address.nextval,'Constanta','Ion Mihalache');
194 INSERT INTO address (address_id,city) values (seq_address.nextval,'Brasov');
195
196 insert into transaction values (seq_transaction.nextval,5,'Member rented copy 1 of title 10');
197 insert into transaction values (seq_transaction.nextval,6,'Member rented copy 1 of title 10');
198 insert into transaction values (seq_transaction.nextval,7,'Member rented copy 1 of title 10');
199
200 select *
201 from address;
202
203 -- 6.
204 -- functie care returneaza id-ul copililor care sunt in momentul asta inchiriate
205 create or replace type copies is table of number;
206

```

ADDRESS_ID	CITY	STREET_NAME
1	1 Bucharest	Ion Mihalache
2	2 Bucharest	(null)
3	3 Cluj	Nicolae Grigorescu
4	4 Constanta	Ion Mihalache
5	5 Brasov	(null)

Transaction:

```

192 INSERT INTO address values (seq_address.nextval,'Cluj','Nicolae Grigorescu');
193 INSERT INTO address values (seq_address.nextval,'Constanta','Ion Mihalache');
194 INSERT INTO address (address_id,city) values (seq_address.nextval,'Brasov');
195
196 insert into transaction values (seq_transaction.nextval,5,'Member rented copy 1 of title 10');
197 insert into transaction values (seq_transaction.nextval,6,'Member rented copy 1 of title 10');
198 insert into transaction values (seq_transaction.nextval,7,'Member rented copy 1 of title 10');
199
200 select *
201 from transaction;
202
203 -- 6.
204 -- functie care returneaza id-ul copiilor care sunt in momentul asta inchiriate
205 create or replace type copies is table of number;
206
207 create or replace function showRented

```

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.004 seconds

TRANSACTION_ID	MEMBER_ID	TRANSACTION_DESCRIPTION
1	4	5 Member rented copy 1 of title 10
2	5	6 Member rented copy 1 of title 10
3	6	7 Member rented copy 1 of title 10

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

Functie care returneaza id-ul copiilor care sunt in acest moment imprumutate(act_ret_date null, folosesc tabel de number). Am tratat cazul in care nu exista nicio copie imprumutata.

Text:

```
create or replace type copies is table of number;
```

```
create or replace function showRented
```

```
return copies
```

```
is
```

```
    v_returned copies :=copies();
```

```
    v_numarCopii number:=0;
```

```
begin
```

```
    select count(*)
```

```
    into v_numarCopii
```

```
    from rental
```

```
    where act_ret_date is null;
```

```

        if (v_numarCopii=0) then
            raise_application_error(-20001,'Nu exista copii
imprumutate.');
```

```

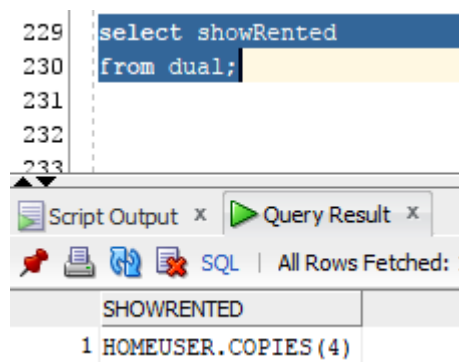
        else

            select copy_id
            bulk collect into v_returned
            from rental
            where act_ret_date is null;
            return v_returned;

        end if;
end;
```

Oracle:

Succes:



(Copia 4 este singura imprumutata)

Exceptie:

```
229 select showRented
230 from dual;
231
232
233
234
235
236 --7 cartile scrise de un autor care sunt in categoria IT
237 create or replace function return_carti_IT (id_author in author.author_id%type)
238 return nume_carti
239 is
240     return_nume_carti nume_carti := nume_carti();
241     v_nrOfRows number;
242     cursor c is
```

Script Output x Query Result x

SQL | Executing:select showRentedfrom dual in 0 seconds

ORA-20001: Nu exista copii imprumutate.
ORA-06512: at "HOMEUSER.SHOWRENTED", line 12

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

Funcție care returnează numele cartilor scrise de un anumit autor_id dat ca parametru din categoria IT. (folosesc cursor explicit cu care fac query-ul de sql). Tratez cazul în care id-ul dat nu este valid.

Text:

```
create or replace function return_carti_IT (id_author in
author.author_id%type)
return nume_carti
is
    return_nume_carti nume_carti := nume_carti();
    v_nrOfRows number;
    cursor c is

        select t.description
        from title t, title_author ta
        where id_author=ta.author_id
        and ta.title_id=t.title_id
        and t.category = 'IT';
begin
```

```

select count(*)
into v_nrOfRows
from author
where author_id=id_author;

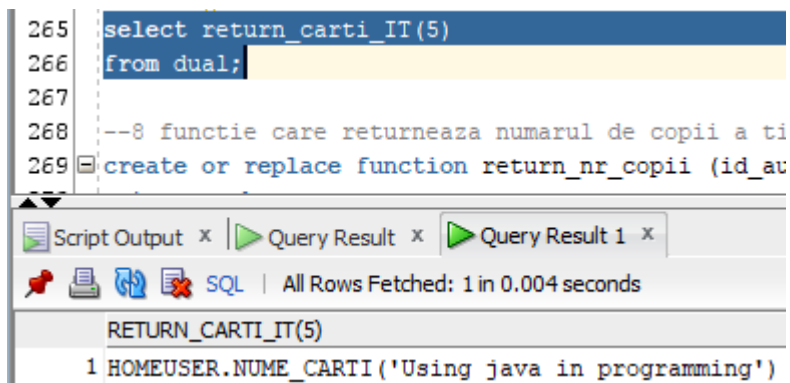
if (v_nrOfRows = 0) then
    raise_application_error(-20001,'Nu exista autorul.');
```

```

else
    OPEN c;
    FETCH c BULK COLLECT INTO return_nume_carti;
    CLOSE c;
    return return_nume_carti;
end if;
end;
```

Oracle:

Succes:



Exceptie:


```

265 select return_carti_IT(11)
266 from dual;
267
268 --8 functie care returneaza numarul c
269 create or replace function return_nr_
270 return number
271 is

```

Script Output x Query Result x

SQL | Executing:select return_carti_IT(1

ORA-20001: Nu exista autorul.
ORA-06512: at "HOMEUSER.RETURN_CARTI_IT", line 20

8. Definiți un subprogram stocat de tip funcție care să utilizeze 3 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Funcție pentru returnat carti care, in cazul in care totul a fost introdus corect, returneaza balance-ul membrului care a adus cartea. Tratez cazurile in care id-urile date pt titlu sau membru sunt gresite si cel in care membrul nu imprumutase nicio copie a titlului dat ca parametru.

Text:

```

create or replace function returnBook (id_title in
title.title_id%type,id_member in member.member_id%type)
return number
is
    v_nrofRows number;
    v_returnPermis number:=1;
    v_nrCopie number :=-1;
    v_balance number;

    v_exp_date rental.exp_ret_date%type;
    v_book_date rental.book_date%type;
    v_price_per_day title.price_per_day%type;
begin
    --verificam titlul
    select count(*)

```

```

into v_nrOfRows
from title
where title_id=id_title;
if (v_nrOfRows !=1) then
    v_returnPermis:=0;
end if;
--verificam ca membrul sa fie corect
select count(*)
into v_nrOfRows
from member
where member_id=id_member;
if (v_nrOfRows !=1) then
    v_returnPermis:=0;
end if;
if (v_returnPermis=0) then
    raise_application_error(-20001,'Id titlu sau membru
gresit.');
```

```

else
    select exp_ret_date,book_date,copy_id
    into v_exp_date,v_book_date,v_nrCopie
    from rental
    where member_id=id_member
    and act_ret_date is null
    and copy_id in (select copy_id
                     from copy_title
                     where title_id=id_title);
    if (v_nrCopie=-1) then
        raise_application_error(-20001,'Nicio copie
imprumutata');
```

```

    else
        select price_per_day
        into v_price_per_day
        from title
```

```

        where title_id=id_title;

        update rental
        set act_ret_date=sysdate
        where member_id=id_member
        and act_ret_date is null
        and copy_id in (select copy_id
                        from copy_title
                        where title_id=id_title);

        update member
            set balance = balance+
v_price_per_day*(v_exp_date-v_book_date);
        if (sysdate>v_exp_date) then
            update member
                set balance = balance -
v_price_per_day*(sysdate-v_exp_date);
        end if;

        select balance
        into v_balance
        from member
        where member_id=id_member;
        return v_balance;
    end if;
end if;
end;
```

Oracle:

Succes:

```
357 begin
358     DBMS_OUTPUT.put_line(returnBook(2,6));
359 end;
360 select * from rental;
361
362
```

Script Output x Query Result x Query Result 1 x

Task completed in 0.037 seconds

Dbms Output

Buffer Size: 20000

homeuser x

0

New Balance is : 0

Exceptii:

Id gresit:

```
346 select returnBook(1,100)
347 from dual;
348
349
350
351 -- EX 9 title,copy title, rental
352 create or replace procedure rent
353 is
```

Script Output x Query Result x Query Result 1 x

SQL | Executing:select returnBook

ORA-20001: Id titlu sau membru gresit.
ORA-06512: at "HOMEUSER.RETURNBOOK", line 30

Nicio copie imprumutata din titlul respectiv:

```
357 begin
358     DBMS_OUTPUT.put_line(returnBook(2,6));
359 end;
360 select * from rental;
361
362
```

Script Output x Query Result x Query Result 1 x

Task completed in 0.037 seconds

Error starting at line : 357 in command -
begin
 DBMS_OUTPUT.put_line(returnBook(2,6));
end;
Error report -
ORA-20001: Nicio copie imprumutata
ORA-06512: at "HOMEUSER.RETURNBOOK", line 42
ORA-06512: at line 2

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.

Funcție pentru a face un imprumut : primește title_id, member_id și nr de zile care dorește membrul să imprumute cartea. Funcția caută ce copie îi poate oferi membrului, îi scade din balance price_per_day * numărul de zile dat de membru și salvează în transaction copia, titlul și pretul tranzacției. (tabele folosite: title, copy_title, rental, member, transaction).

Cazurile tratate sunt : id titlu sau membru gresite și cazul în care nu e valabilă nicio copie.

Text:

```
create or replace procedure rent (id_title in
title.title_id%type, id_member in
member.member_id%type, exp_days_rented in int)
is
    v_price title.price_per_day%type;
    type copies is table of copy_title.copy_id%type;
```

```

t_copies copies;
v_copy_to_give number :=-1;
v_copy_ordered number :=1;

v_nrOfRows number;
v_rentPermis number:=1;

begin
    --verificam titlul
    select count(*)
    into v_nrOfRows
    from title
    where title_id=id_title;
    if (v_nrOfRows !=1) then
        v_rentPermis:=0;
    end if;
    --verificam ca membrul sa fie corect
    select count(*)
    into v_nrOfRows
    from member
    where member_id=id_member;
    if (v_nrOfRows !=1) then
        v_rentPermis:=0;
    end if;
    if (v_rentPermis=1) then

        select copy_id
        bulk collect into t_copies
        from copy_title
        where title_id=id_title;

        for i in t_copies.first..t_copies.last

```

```

loop
    v_copy_ordered:=1;
    select count(*)
    into v_copy_ordered
    from rental
    where
        copy_id=t_copies(i)
        and
        act_ret_date is null;
    if (v_copy_ordered = 0) then
        v_copy_to_give:=t_copies(i);
    end if;
end loop;

if (v_copy_to_give=-1) then
    raise_application_error(-20001,'Nu este valabila
nicio copie');
else

    insert into rental
    (rental_id,book_date,copy_id,member_id,exp_ret_date) values
    (seq_rental.nextval,sysdate,v_copy_to_give,id_member,sysdate+e
xp_days_rented);

    select price_per_day
    into v_price
    from title
    where title_id=id_title;

    update member
    set balance = balance - exp_days_rented*v_price
    where member_id = id_member;

    insert into transaction values
    (seq_transaction.nextval,id_member,'Member rented copy '

```

```

||v_copy_to_give || ', copy of title with id : ' || id_title
|| '->' || ' -' || exp_days_rented*v_price);

    end if;

    else

        raise_application_error(-20001,'Id membru sau titlu
gresit.');
```

```

    end if;

end;
```

Oracle:

Succes:

```

437 BEGIN
438 rent(3,6,20);
439 end;
440 select * from rental
441 where member_id=6;
442
443
444 --10 trigger la nivel de instructiune : nu se pot adauga copii decat in
445
446 create or replace trigger ex_10
447     before insert on copy_title
448 BEGIN
449     if (TO_CHAR(SYSDATE,'D') =1)
450         OR (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 8 AND 18)
```

Script Output x Query Result x Query Result 1 x

SQL | All Rows Fetched: 4 in 0.002 seconds

	RENTAL_ID	BOOK_DATE	COPY_ID	MEMBER_ID	ACT_RET_DATE	EXP_RET_DATE
1	15	04-JAN-21	2	6	04-JAN-21	03-FEB-21
2	16	04-JAN-21	2	6	04-JAN-21	03-FEB-21
3	17	04-JAN-21	5	6	(null)	24-JAN-21
4	14	04-JAN-21	2	6	04-JAN-21	03-FEB-21

Exceptii:

Id titlu sau membru gresit :


```

437 BEGIN
438 rent(1,100,20);
439 end;
440
441
442 --10 trigger la nivel de instructiune : 1
443
444 create or replace trigger ex_10
445     before insert on copy_title
446 BEGIN
447     if (TO_CHAR(SYSDATE,'D') =1)
448         OR (TO_CHAR(SYSDATE,'HH24') NOT 1
449     THEN
450         raise application_error(-20001,'Nu po

```

Script Output x Query Result x Query Result 1 x
 Task completed in 0.037 seconds

```

rent(1,100,20);
end;
Error report -
ORA-20001: Id membru sau titlu gresit.
ORA-06512: at "HOMEUSER.RENT", line 68
ORA-06512: at line 2

```

Nicio copie valabila:

```

437 BEGIN
438 rent(1,6,20);
439 end;
440
441
442 --10 trigger la nivel de instructiun
443
444 create or replace trigger ex_10
445     before insert on copy_title
446 BEGIN
447     if (TO_CHAR(SYSDATE,'D') =1)
448         OR (TO_CHAR(SYSDATE,'HH24')
449     THEN
450         raise application_error(-20001,'

```

Script Output x Query Result x Query Result
 Task completed in 0.045 seconds

```

rent(1,6,20);
end;
Error report -
ORA-20001: Nu este valabila nicio copie
ORA-06512: at "HOMEUSER.RENT", line 51
ORA-06512: at line 2

```

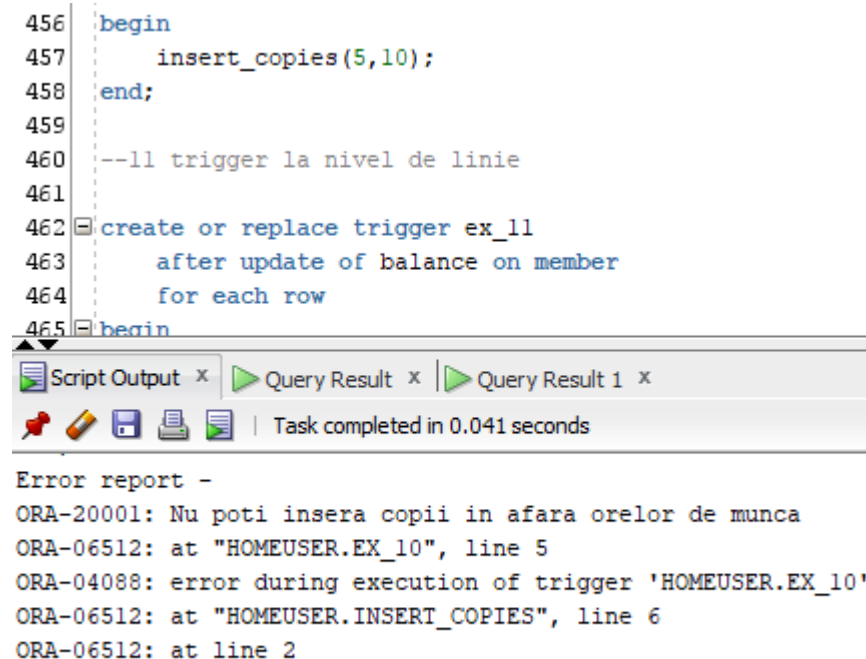
10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

Trigger care nu lasa pe nimeni sa introduca copii in afara programului de lucru 8-18.

Text:

```
create or replace trigger ex_10
  before insert on copy_title
BEGIN
  if (TO_CHAR(SYSDATE, 'D') =1)
    OR (TO_CHAR(SYSDATE, 'HH24') NOT BETWEEN 8 AND 17)
  THEN
    raise_application_error(-20001, 'Nu poti insera copii in
afara orelor de munca');
  end if;
END;
```

Oracle:



The screenshot shows a SQL Developer window with a script editor and a results pane. The script editor contains the following SQL code:

```
456 begin
457     insert_copies(5,10);
458 end;
459
460 --ll trigger la nivel de linie
461
462 create or replace trigger ex_11
463     after update of balance on member
464     for each row
465 begin
```

The results pane shows the following error report:

```
Error report -
ORA-20001: Nu poti insera copii in afara orelor de munca
ORA-06512: at "HOMEUSER.EX_10", line 5
ORA-04088: error during execution of trigger 'HOMEUSER.EX_10'
ORA-06512: at "HOMEUSER.INSERT_COPIES", line 6
ORA-06512: at line 2
```

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

Trigger care se declanseaza cand un membru a ajuns la balance de sub –500 lei.

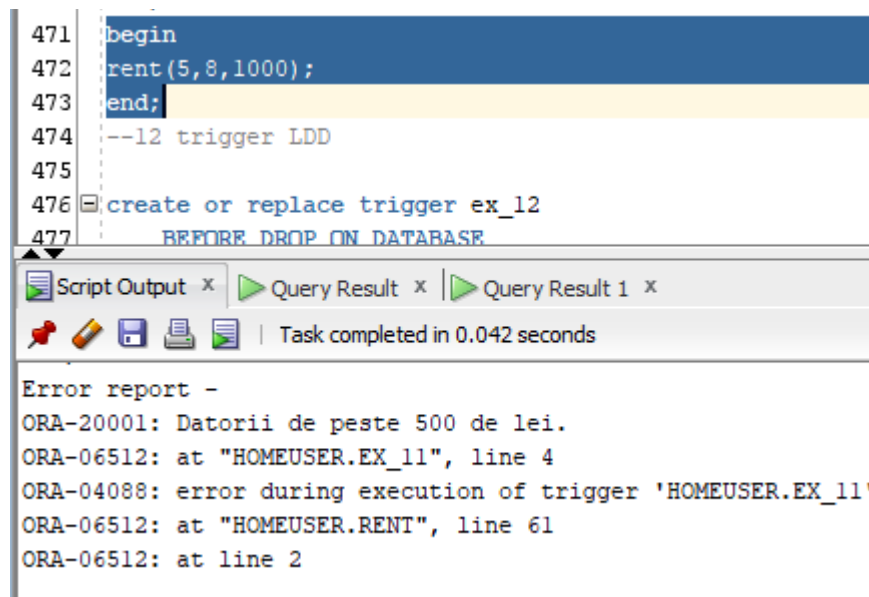
Text:

```
create or replace trigger ex_11
  after update of balance on member
  for each row
begin
  if (:NEW.balance < -500)
  then
    raise_application_error(-20001,'Datorii de peste 500 de
lei.');
```

end if;

end;

Oracle:



```
471 begin
472 rent(5,8,1000);
473 end;
474 --12 trigger LDD
475
476 create or replace trigger ex_12
477 BEFORE DROP ON DATABASE
```

Script Output x Query Result x Query Result 1 x

Task completed in 0.042 seconds

Error report -

```
ORA-20001: Datorii de peste 500 de lei.
ORA-06512: at "HOMEUSER.EX_11", line 4
ORA-04088: error during execution of trigger 'HOMEUSER.EX_11'
ORA-06512: at "HOMEUSER.RENT", line 61
ORA-06512: at line 2
```

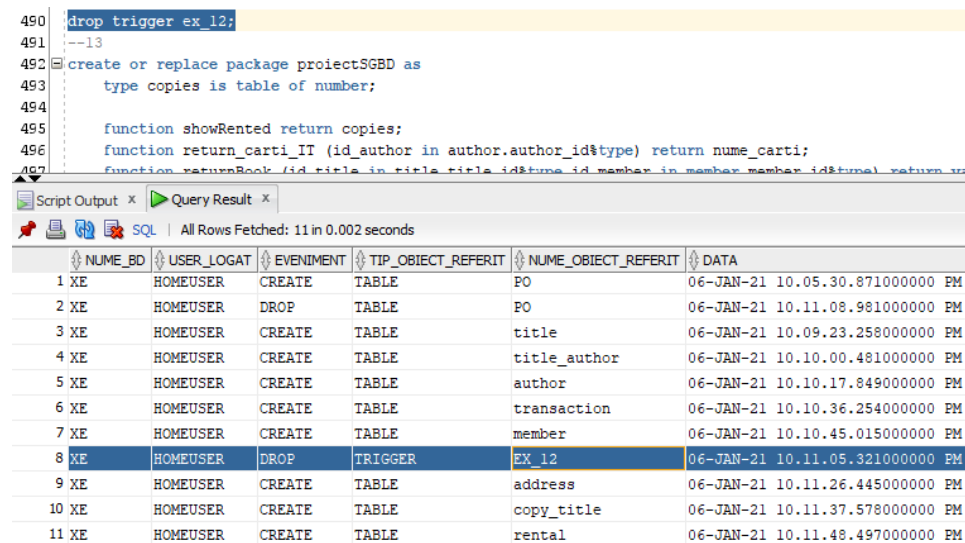
12. Definiți un trigger de tip LDD. Declanșați trigger-ul.

Trigger care nu permite executia comenzii drop asupra unui tabel.

Text:

```
create or replace trigger audit_schema
    AFTER CREATE OR DROP OR ALTER ON SCHEMA
    BEGIN
        INSERT INTO audit_user VALUES (SYS.DATABASE_NAME,
        SYS.LOGIN_USER, SYS.SYSEVENT,
        SYS.DICTIONARY_OBJ_TYPE, SYS.DICTIONARY_OBJ_NAME,
        SYSTIMESTAMP(3));
    END;
```

Oracle:



The screenshot shows an Oracle SQL Developer window with a script editor and a query result grid. The script editor contains the following SQL code:

```
490 drop trigger ex_12;
491 --13
492 create or replace package proiectSGBD as
493     type copies is table of number;
494
495     function showRented return copies;
496     function return_carti_IT (id_author in author.author_id%type) return nume_carti;
497     function returnBook (id_title in title.title_id%type, id_member in member.member_id%type) return nume_book;
```

The query result grid shows the execution of the script, with 11 rows fetched in 0.002 seconds. The columns are: NUME_BD, USER_LOGAT, EVENIMENT, TIP_OBJECT_REFERIT, NUME_OBJECT_REFERIT, and DATA. The data shows the creation and dropping of various database objects, including tables, triggers, and packages.

	NUME_BD	USER_LOGAT	EVENIMENT	TIP_OBJECT_REFERIT	NUME_OBJECT_REFERIT	DATA
1	XE	HOMEUSER	CREATE	TABLE	PO	06-JAN-21 10.05.30.871000000 PM
2	XE	HOMEUSER	DROP	TABLE	PO	06-JAN-21 10.11.08.981000000 PM
3	XE	HOMEUSER	CREATE	TABLE	title	06-JAN-21 10.09.23.258000000 PM
4	XE	HOMEUSER	CREATE	TABLE	title_author	06-JAN-21 10.10.00.481000000 PM
5	XE	HOMEUSER	CREATE	TABLE	author	06-JAN-21 10.10.17.849000000 PM
6	XE	HOMEUSER	CREATE	TABLE	transaction	06-JAN-21 10.10.36.254000000 PM
7	XE	HOMEUSER	CREATE	TABLE	member	06-JAN-21 10.10.45.015000000 PM
8	XE	HOMEUSER	DROP	TRIGGER	EX_12	06-JAN-21 10.11.05.321000000 PM
9	XE	HOMEUSER	CREATE	TABLE	address	06-JAN-21 10.11.26.445000000 PM
10	XE	HOMEUSER	CREATE	TABLE	copy_title	06-JAN-21 10.11.37.578000000 PM
11	XE	HOMEUSER	CREATE	TABLE	rental	06-JAN-21 10.11.48.497000000 PM

(Ex 12 era fostul trigger de la exercitiul 12 insa aveam probleme cu el)

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

Text:

```
create or replace package proiectSGBD as
    type copies is table of number;
    type grup_autori is table of number;
    type nume_carti is table of varchar2(128);

    function showRented return copies;
    function return_carti_IT (id_author in
    author.author_id%type) return nume_carti;
```

```
function returnBook (id_title in  
title.title_id%type,id_member in member.member_id%type) return  
varchar;
```

```
procedure rent (id_title in title.title_id%type,id_member  
in member.member_id%type,exp_days_rented in int);
```

```
procedure insert_copies (copy_nr in number,id_title in  
title.title_id%type);
```

```
procedure addBalance (id_member in  
member.member_id%type,value_added in number);
```

```
function viewBalance (id_member in member.member_id%type)  
return number;
```

```
procedure createMember (name_first in varchar2,name_last  
in varchar2,phone in varchar2,p_city in varchar2,name_street  
in varchar2);
```

```
function memberExists(name_first in varchar2,name_last in  
varchar2,phone in varchar2,p_city in varchar2,name_street in  
varchar2) return int;
```

```
function getMemberId(name_first in varchar2,name_last in  
varchar2,phone in varchar2,p_city in varchar2,name_street in  
varchar2) return int;
```

```
function getTitlesOfAuthor(id_author in number) return  
nume_carti;
```

```
function getAuthorId(name_first in varchar2,name_last in  
varchar2) return grup_autori;
```

```
end proiectSGBD;
```

```
create or replace package body proiectSGBD as
```

```
function showRented
```

```
return copies
```

```
is
```

```
    v_returned copies :=copies();
```

```
    v_numarCopii number:=0;
```

```
begin
```

```
    select count(*)
```

```
    into v_numarCopii
```

```

from rental
where act_ret_date is null;
if (v_numarCopii=0) then
    raise_application_error(-20001,'Nu exista copii
imprumutate.');
```

```

else

    select copy_id
    bulk collect into v_returned
    from rental
    where act_ret_date is null;
    return v_returned;
end if;
end;
```

```

function return_carti_IT (id_author in author.author_id%type)
return nume_carti
is
    return_nume_carti nume_carti := nume_carti();
    v_nrOfRows number;
    cursor c is

    select t.description
    from title t, title_author ta
    where id_author=ta.author_id
    and ta.title_id=t.title_id
    and t.category ='IT';
begin
    select count(*)
    into v_nrOfRows
    from author
    where author_id=id_author;
```

```

    if (v_nrOfRows = 0) then
        raise_application_error(-20001,'Nu exista autorul.');
```

else

```

        OPEN c;
        FETCH c BULK COLLECT INTO return_nume_carti;
        CLOSE c;
        return return_nume_carti;
    end if;
end;
```

function returnBook (id_title in title.title_id%type,id_member
in member.member_id%type)

return varchar

is

```

    v_nrOfRows number;
    v_returnPermis number:=1;
    v_nrCopie number :=-1;
    v_balance number;
```

```

    v_exp_date rental.exp_ret_date%type;
    v_book_date rental.book_date%type;
    v_price_per_day title.price_per_day%type;
```

begin

```

    --verificam titlul
    select count(*)
    into v_nrOfRows
    from title
    where title_id=id_title;
    if (v_nrOfRows !=1) then
        v_returnPermis:=0;
    end if;
    --verificam ca membrul sa fie corect
    select count(*)
```

```

into v_nrOfRows
from member
where member_id=id_member;
if (v_nrOfRows !=1) then
    v_returnPermis:=0;
end if;
if (v_returnPermis=0) then
    raise_application_error(-20001,'Id titlu sau membru
gresit.');
```

```

else
    select count(*)
    into v_nrOfRows
    from rental
    where act_ret_date is null
    and member_id=id_member
    and copy_id in (select copy_id
                    from copy_title
                    where title_id=id_title);

    if (v_nrOfRows=0) then
        raise_application_error(-20001,'Nicio copie
imprumutata');
```

```

    else

        select exp_ret_date,book_date,copy_id
        into v_exp_date,v_book_date,v_nrCopie
        from rental
        where member_id=id_member
        and act_ret_date is null
        and copy_id in (select copy_id
                        from copy_title
                        where title_id=id_title);

```



```

        select price_per_day
        into v_price_per_day
        from title
        where title_id=id_title;

        update rental
        set act_ret_date=sysdate
        where member_id=id_member
        and act_ret_date is null
        and copy_id in (select copy_id
                        from copy_title
                        where title_id=id_title);

        update member
            set balance = balance+
v_price_per_day*(v_exp_date-v_book_date);
            if (sysdate>v_exp_date) then
                update member
                    set balance = balance -
v_price_per_day*(sysdate-v_exp_date);
            end if;
            select balance
            into v_balance
            from member
            where member_id=id_member;
            return 'New Balance is : '|| v_balance;
        end if;
    end if;
end;

procedure rent (id_title in title.title_id%type,id_member in
member.member_id%type,exp_days_rented in int)
is

```

```

v_price title.price_per_day%type;
type copies is table of copy_title.copy_id%type;
t_copies copies;
v_copy_to_give number :=-1;
v_copy_ordered number :=1;

v_nrOfRows number;
v_rentPermis number:=1;

begin
    --verificam titlul
    select count(*)
    into v_nrOfRows
    from title
    where title_id=id_title;
    if (v_nrOfRows !=1) then
        v_rentPermis:=0;
    end if;
    --verificam ca membrul sa fie corect
    select count(*)
    into v_nrOfRows
    from member
    where member_id=id_member;
    if (v_nrOfRows !=1) then
        v_rentPermis:=0;
    end if;
    if (v_rentPermis=1) then

        select copy_id
        bulk collect into t_copies
        from copy_title
        where title_id=id_title;

```

```

for i in t_copies.first..t_copies.last
loop
    v_copy_ordered:=1;
    select count(*)
    into v_copy_ordered
    from rental
    where
    copy_id=t_copies(i)
    and
    act_ret_date is null;
    if (v_copy_ordered = 0) then
        v_copy_to_give:=t_copies(i);
    end if;
end loop;
if (v_copy_to_give=-1) then
    raise_application_error(-20001,'Nu este valabila
nicio copie');
else

    insert into rental
(rental_id,book_date,copy_id,member_id,exp_ret_date) values
(seq_rental.nextval,sysdate,v_copy_to_give,id_member,sysdate+e
xp_days_rented);

    select price_per_day
    into v_price
    from title
    where title_id=id_title;

    update member
    set balance = balance - exp_days_rented*v_price
    where member_id = id_member;

```

```

            insert into transaction values
(seq_transaction.nextval,id_member,'Member rented copy '
||v_copy_to_give || ', copy of title with id : ' || id_title
|| '->' || ' -' || exp_days_rented*v_price);

        end if;

    else

        raise_application_error(-20001,'Id membru sau titlu
gresit.');
```

end if;

end;

```

procedure insert_copies (copy_nr in number,id_title in
title.title_id%type)
is
begin
    for i in 1..copy_nr
    loop
        INSERT INTO copy_title values
(seq_copy_title.nextval,id_title);
    end loop;
end;
```

```

procedure addBalance (id_member in
member.member_id%type,value_added in number)
is
begin
    update member
    set balance=balance+value_added
    where member_id=id_member;
end;
```

```

function viewBalance (id_member in member.member_id%type)
return number
is
    v_balance number;
```

```

begin
    select balance
    into v_balance
    from member
    where member_id=id_member;
    return v_balance;
end;

```

```

procedure createMember (name_first in varchar2,name_last in
varchar2,phone in varchar2,p_city in varchar2,name_street in
varchar2)

```

```

is

```

```

    v_address_id number :=-1;

```

```

begin

```

```

    if
(memberExists(name_first,name_last,phone,p_city,name_street)=0
) then

```

```

        select address_id
        into v_address_id
        from address
        where city=p_city
        and street_name=name_street;

```

```

        if (v_address_id = -1) then

```

```

            v_address_id:=seq_address.nextval;

```

```

            insert into address values
(v_address_id,p_city,name_street);

```

```

            insert into member
(member_id,first_name,last_name,address_id,phone_number,join_d
ate) values
(seq_member.nextval,name_first,name_last,v_address_id,phone,Sy
sdate);

```

```

        else

```

```

            insert into member
(member_id,first_name,last_name,address_id,phone_number,join_d
ate) values
(seq_member.nextval,name_first,name_last,v_address_id,phone,Sy
sdate);

```

```

        end if;
        DBMS_OUTPUT.put_line('Member added');
    else
        DBMS_OUTPUT.put_line('Member already exists');
    end if;
end;

function memberExists(name_first in varchar2,name_last in
varchar2,phone in varchar2,p_city in varchar2,name_street in
varchar2)
return int
is
    v_exists int;
    v_address_id number :=-1;
    v_member_id number :=-1;
begin
    select count(*)
    into v_address_id
    from address
    where city=p_city
    and street_name=name_street;
    if (v_address_id =-1) then
        return 0;
    else
        select count(*)
        into v_member_id
        from member
        where first_name=name_first
        and last_name=name_last
        and phone_number=phone;
        if (v_member_id=-1) then
            return 0;
        else

```

```

        return v_member_id;
    end if;
end if;
end;

function getMemberId(name_first in varchar2,name_last in
varchar2,phone in varchar2,p_city in varchar2,name_street in
varchar2)
return int
is
    v_member_id number ;
    v_address_id number;
begin
    if
(memberExists(name_first,name_last,phone,p_city,name_street)=1
) then

        select address_id
        into v_address_id
        from address
        where city=p_city
        and street_name=name_street;
        select member_id
        into v_member_id
        from member
        where first_name=name_first
        and last_name=name_last
        and address_id=v_address_id
        and phone_number=phone;
        return v_member_id;
    else
        return -1;
    end if;
end;

```

```

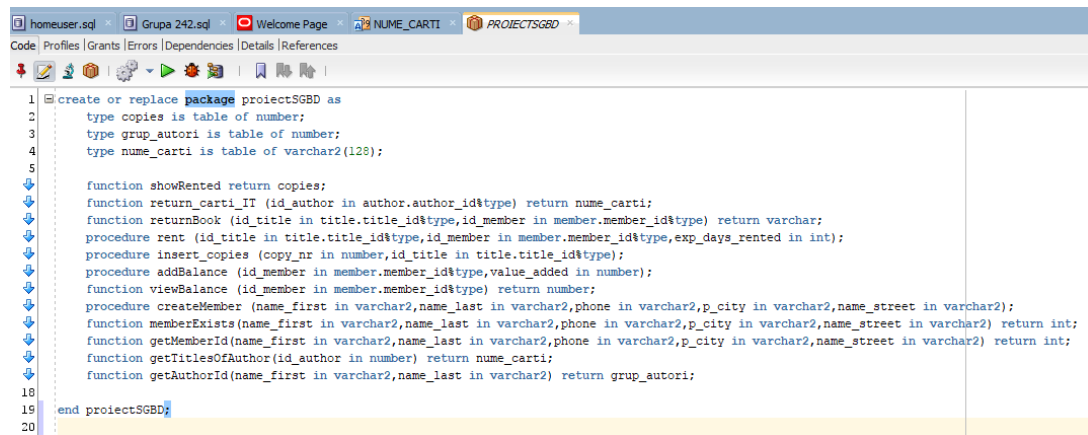
function getAuthorId(name_first in varchar2,name_last in
varchar2)
return grup_autori
is
    v_autori grup_autori := grup_autori();
begin
    select author_id
    bulk collect into v_autori
    from author
    where first_name=name_first
    and last_name=name_last;
    return v_autori;
end;

function getTitlesOfAuthor(id_author in number)
return nume_carti
is
    v_return nume_carti := nume_carti();
begin
    select description
    bulk collect into v_return
    from title t,title_author ta
    where t.title_id=ta.title_id
    and ta.author_id=id_author;
    return v_return;
end;

end proiectSGBD;

```

Oracle:



```
1 create or replace package projectSGBD as
2     type copies is table of number;
3     type grup_autori is table of number;
4     type nume_carti is table of varchar2(128);
5
6     function showRented return copies;
7     function return_carti_IT (id_author in author.author_id%type) return nume_carti;
8     function returnBook (id_title in title.title_id%type, id_member in member.member_id%type) return varchar;
9     procedure rent (id_title in title.title_id%type, id_member in member.member_id%type, exp_days_rented in int);
10    procedure insert_copies (copy_nr in number, id_title in title.title_id%type);
11    procedure addBalance (id_member in member.member_id%type, value_added in number);
12    function viewBalance (id_member in member.member_id%type) return number;
13    procedure createMember (name_first in varchar2, name_last in varchar2, phone in varchar2, p_city in varchar2, name_street in varchar2);
14    function memberExists (name_first in varchar2, name_last in varchar2, phone in varchar2, p_city in varchar2, name_street in varchar2) return int;
15    function getMemberId (name_first in varchar2, name_last in varchar2, phone in varchar2, p_city in varchar2, name_street in varchar2) return int;
16    function getTitlesOfAuthor (id_author in number) return nume_carti;
17    function getAuthorId (name_first in varchar2, name_last in varchar2) return grup_autori;
18
19 end projectSGBD;
```

Funcții pentru utilitatea bibliotecii:

- inserat x copii pt titlu y
- adauga balance
- view balance
- adaugare membrii
- verifica daca exista membru
- get memberid
- get authorid
- get titlurile autorilor