**FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ**

**SEMESTRUL I, ANUL UNIVERSITAR 2021-2022**

**PROIECT - SISTEME DE GESTIUNE A BAZELOR DE DATE**

**GESTIUNEA UNEI BIBLIOTECI**

**TOADER VLAD-MARIAN**

**GRUPA 242**

1. **Prezentarea bazei de date**

Baza de date reține informații cu privire la funcționarea unei biblioteci, conținând date atât despre cărți (editură, categorie, exemplare, autori), cât și despre cititori (împrumuturile făcute de aceștia și abonamentele lor).

Astfel, în diagrama entitate-relație (fig. 1), vor apărea următoarele entități: CARTE, EDITURĂ, CATEGORIE, AUTOR, EXEMPLAR, CITITOR, ABONAMENT și LOCAȚIE.

1. **Diagrama entitate-relație (ERD)**

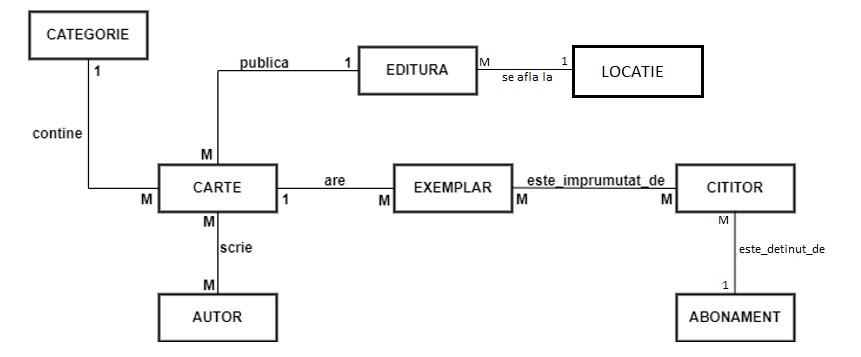
****

Figura 1. Diagrama entitate-relație

1. **Diagrama conceptuală**

Relațiile MANY-TO-MANY dintre entitățile EXEMPLAR și CITITOR, respectiv dintre CARTE și AUTOR vor deveni tabele asociative în diagrama conceptuală (fig. 2).

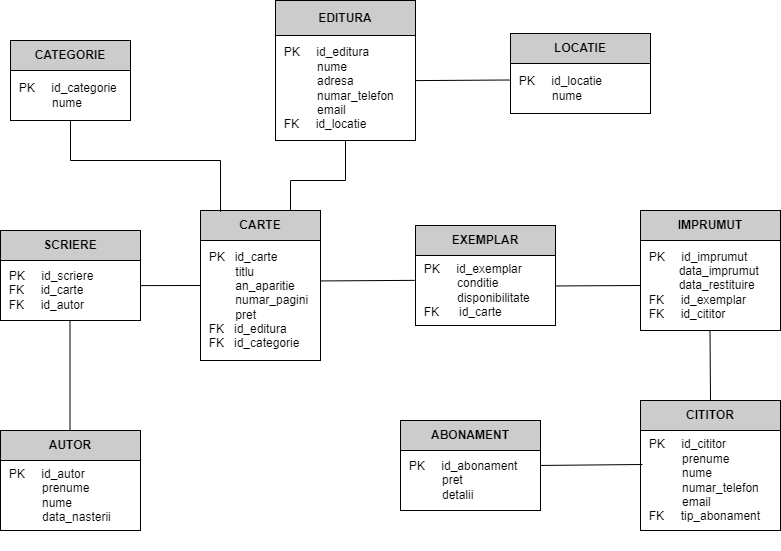
****

Figura 2. Diagrama conceptuală

1. **Implementarea diagramei conceptuale în Oracle**

CREATE TABLE categorie

(id\_categorie number(3) constraint pk\_categorie primary key,

nume varchar2(50));

CREATE TABLE locatie

(id\_locatie number(3) constraint pk\_locatie primary key,

nume varchar2(100));

CREATE TABLE editura

(id\_editura number(3) constraint pk\_editura primary key,

nume varchar2(50),

adresa varchar2(100),

numar\_telefon varchar2(15),

email varchar2(30),

id\_locatie number(3),

constraint fk\_editura\_locatie foreign key (id\_locatie) references locatie (id\_locatie));

CREATE TABLE autor

(id\_autor number(3) constraint pk\_autor primary key,

prenume varchar2(30),

nume varchar2(30),

data\_nasterii date);

CREATE TABLE abonament

(id\_abonament number(3) constraint pk\_abonament primary key,

pret number(2,2),

detalii varchar2(100));

CREATE TABLE carte

(id\_carte number(3) constraint pk\_carte primary key,

titlu varchar2(100),

an\_aparitie number(4),

numar\_pagini number(4),

pret number(3,2),

id\_editura number(3),

id\_categorie number(3),

constraint fk\_carte\_editura foreign key (id\_editura) references editura (id\_editura),

constraint fk\_carte\_categorie foreign key (id\_categorie) references categorie (id\_categorie));

CREATE TABLE exemplar

(id\_exemplar number(3) constraint pk\_exemplar primary key,

conditie varchar2(20),

disponibilitate varchar2(20),

id\_carte number(3),

constraint fk\_exemplar\_carte foreign key (id\_carte) references carte (id\_carte));

CREATE TABLE cititor

(id\_cititor number(3) constraint pk\_cititor primary key,

prenume varchar2(100),

nume varchar2(100),

numar\_telefon varchar2(15),

email varchar2(100),

tip\_abonament number(3),

constraint fk\_cititor\_abonament foreign key (tip\_abonament) references abonament (id\_abonament));

CREATE TABLE scriere

(id\_scriere number(3) constraint pk\_scriere primary key,

id\_autor number(3),

id\_carte number(3),

constraint fk\_scriere\_autor foreign key (id\_autor) references autor (id\_autor),

constraint fk\_scriere\_carte foreign key (id\_carte) references carte (id\_carte));

CREATE TABLE imprumut

(id\_imprumut number(3) constraint pk\_imprumut primary key,

data\_imprumut date,

data\_restituire date,

id\_exemplar number(3),

id\_cititor number(3),

constraint fk\_imprumut\_exemplar foreign key (id\_exemplar) references exemplar (id\_exemplar),

constraint fk\_imprumut\_cititor foreign key (id\_cititor) references cititor (id\_cititor));

ALTER TABLE exemplar

ADD CONSTRAINT check\_disponibilitate

CHECK (disponibilitate in ('DISPONIBIL', 'IMPRUMUTAT'));

ALTER TABLE exemplar

ADD CONSTRAINT check\_conditie

CHECK (conditie in ('NOUA', 'DETERIORATA'));

ALTER TABLE imprumut

ADD CONSTRAINT check\_data

CHECK (data\_restituire >= data\_imprumut);

1. **Adăugarea de date în tabelele create**

INSERT INTO categorie VALUES

(1, 'Thriller');

INSERT INTO categorie VALUES

(2, 'Fantasy');

INSERT INTO categorie VALUES

(3, 'Young adult');

INSERT INTO categorie VALUES

(4, 'Dezvoltare personala');

INSERT INTO categorie VALUES

(5, 'Aventura');

INSERT INTO categorie VALUES

(6, 'SF');

O imagine care conține text

Descriere generată automat

INSERT INTO locatie VALUES

(1, 'Bucuresti');

INSERT INTO locatie VALUES

(2, 'Iasi');

INSERT INTO locatie VALUES

(3, 'Timisoara');

INSERT INTO locatie VALUES

(4, 'Cluj');

INSERT INTO locatie VALUES

(5, 'Brasov');

O imagine care conține text

Descriere generată automat

INSERT INTO editura VALUES

(1, 'Humanitas', 'Piata Presei Libere nr. 1', '0214088355', 'secretariat@humanitas.ro', 1);

INSERT INTO editura VALUES

(2, 'Corint', 'Str. Mihai Eminescu nr. 54A', '0213194797', 'vanzari@edituracorint.ro', 1);

INSERT INTO editura VALUES

(3, 'Litera', 'Str. Moeciu nr. 6', '0374826635', 'contact@litera.ro', 3);

INSERT INTO editura VALUES

(4, 'Nemira', 'Splaiul Unirii nr. 6', ' 0213138978', 'office.bucuresti@polirom.ro', 4);

INSERT INTO editura VALUES

(5, 'Polirom', 'Str. Iani Buzoiani nr. 14', '0721747464', 'office@nemira.ro', 2);

O imagine care conține text

Descriere generată automat

INSERT INTO autor VALUES

(1, 'Stephen', 'King', to\_date('21-09-1947', 'dd-mm-yyyy'));

INSERT INTO autor VALUES

(2, 'Leigh', 'Bardugo', to\_date('06-04-1975', 'dd-mm-yyyy'));

INSERT INTO autor VALUES

(3, 'Adam', 'Silvera', to\_date('07-06-1990', 'dd-mm-yyyy'));

INSERT INTO autor VALUES

(4, 'John', 'Green', to\_date('24-08-1977', 'dd-mm-yyyy'));

INSERT INTO autor VALUES

(5, 'Brian', 'Tracy', to\_date('05-01-1947', 'dd-mm-yyyy'));

INSERT INTO autor VALUES

(6, 'J.R.R.', 'Tolkien', to\_date('03-01-1892', 'dd-mm-yyyy'));

INSERT INTO autor VALUES

(7, 'Jules', 'Verne', to\_date('08-02-1828', 'dd-mm-yyyy'));

O imagine care conține text

Descriere generată automat

ALTER TABLE abonament MODIFY detalii VARCHAR2(500);

ALTER TABLE abonament MODIFY pret NUMBER(2,0);

INSERT INTO abonament VALUES

(1, 50, 'lunar premium - toate zonele');

INSERT INTO abonament VALUES

(2, 40, 'lunar standard - restrictionat la cartile de dezvoltare personala');

INSERT INTO abonament VALUES

(3, 20, 'saptamanal premium - acces in toate zonele');

INSERT INTO abonament VALUES

(4, 10, 'saptamanal standard - restrictionat la cartile de dezvoltare personala');

INSERT INTO abonament VALUES

(5, 5, 'zilnic - acces in toate zonele');

O imagine care conține text

Descriere generată automat

ALTER TABLE carte MODIFY pret NUMBER (5,2);

INSERT INTO carte VALUES

(1, 'Carrie', 1974, 199, 30, 1, 1);

INSERT INTO carte VALUES

(2, 'Mr. Mercedes', 2015, 488, 32.44, 2, 1);

INSERT INTO carte VALUES

(3, 'Regatul umbrelor', 2012, 344, 30, 5, 2);

INSERT INTO carte VALUES

(4, 'Banda celor sase ciori', 2013, 592, 41.65, 5, 2);

INSERT INTO carte VALUES

(5, 'Amandoi mor la sfarsit', 2021, 336, 45, 3, 3);

INSERT INTO carte VALUES

(6, 'Sub aceeasi stea', 2013, 312, 43.50, 4, 3);

INSERT INTO carte VALUES

(7, 'Incepe ce nu-ti place!', 2001, 245, 20.50, 1, 4);

INSERT INTO carte VALUES

(8, 'Succesul in viata', 1993, 198, 25, 3, 4);

INSERT INTO carte VALUES

(9, 'Stapanul inelelor', 1954, 670, 63.50, 4, 5);

INSERT INTO carte VALUES

(10, 'Hobbitul', 1937, 532, 57, 2, 5);

INSERT INTO carte VALUES

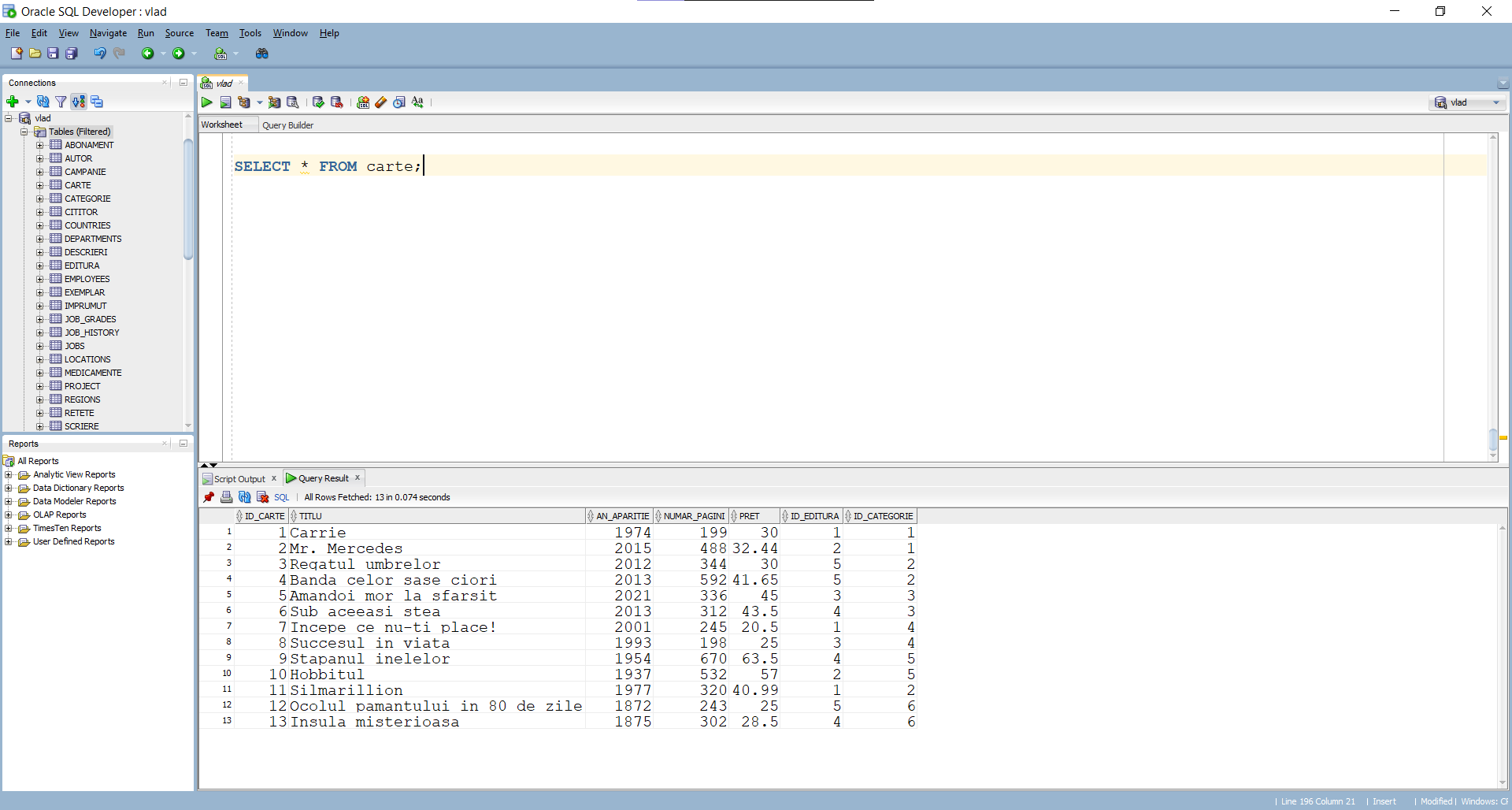
(11, 'Silmarillion', 1977, 320, 40.99, 1, 2);

INSERT INTO carte VALUES

(12, 'Ocolul pamantului in 80 de zile', 1872, 243, 25, 5, 6);

INSERT INTO carte VALUES

(13, 'Insula misterioasa', 1875, 302, 28.5, 4, 6);



INSERT INTO exemplar VALUES

(1, 'NOUA', 'DISPONIBIL', 1);

INSERT INTO exemplar VALUES

(2, 'DETERIORATA', 'DISPONIBIL', 1);

INSERT INTO exemplar VALUES

(3, 'NOUA', 'IMPRUMUTAT', 2);

INSERT INTO exemplar VALUES

(4, 'DETERIORATA', 'IMPRUMUTAT', 2);

INSERT INTO exemplar VALUES

(5, 'NOUA', 'IMPRUMUTAT', 3);

INSERT INTO exemplar VALUES

(6, 'NOUA', 'IMPRUMUTAT', 3);

INSERT INTO exemplar VALUES

(7, 'DETERIORATA', 'IMPRUMUTAT', 3);

INSERT INTO exemplar VALUES

(8, 'NOUA', 'DISPONIBIL', 4);

INSERT INTO exemplar VALUES

(9, 'DETERIORATA', 'IMPRUMUTAT', 5);

INSERT INTO exemplar VALUES

(10, 'DETERIORATA', 'DISPONIBIL', 6);

INSERT INTO exemplar VALUES

(11, 'NOUA', 'IMPRUMUTAT', 6);

INSERT INTO exemplar VALUES

(12, 'NOUA', 'DISPONIBIL', 7);

INSERT INTO exemplar VALUES

(13, 'DETERIORATA', 'DISPONIBIL', 8);

INSERT INTO exemplar VALUES

(14, 'NOUA', 'IMPRUMUTAT', 9);

INSERT INTO exemplar VALUES

(15, 'NOUA', 'IMPRUMUTAT', 9);

INSERT INTO exemplar VALUES

(16, 'DETERIORATA', 'DISPONIBIL', 10);

INSERT INTO exemplar VALUES

(17, 'NOUA', 'IMPRUMUTAT', 11);

INSERT INTO exemplar VALUES

(18, 'DETERIORATA', 'DISPONIBIL', 11);

INSERT INTO exemplar VALUES

(19, 'NOUA', 'DISPONIBIL', 12);

INSERT INTO exemplar VALUES

(20, 'NOUA', 'DISPONIBIL', 12);

INSERT INTO exemplar VALUES

(21, 'DETERIORATA', 'IMPRUMUTAT', 13);

O imagine care conține masă

Descriere generată automat

INSERT INTO cititor VALUES

(1, 'Theodor', 'Popescu', '0752111232', 'theo.p@hotmail.com', 1);

INSERT INTO cititor VALUES

(2, 'Talida', 'Boboc', '0753212343', 'talidaboboc@gmail.com', 4);

INSERT INTO cititor VALUES

(3, 'Andreea', 'Constantinescu', '0721433212', 'andreea.const@yahoo.com', 2);

INSERT INTO cititor VALUES

(4, 'Elena', 'Malancu', '0782232123', 'elena\_malancu\_01@gmail.com', 3);

INSERT INTO cititor VALUES

(5, 'Sofia', 'Vultur', '0742121854', 'sofiavultur@yahoo.com', 5);

INSERT INTO cititor VALUES

(6, 'Lucas', 'Pelmus', '0765211943', 'lucasss@hotmail.com', 1);

INSERT INTO cititor VALUES

(7, 'Mihail', 'Manea', '0732114666', 'mihail\_seb@gmail.com', 3);

O imagine care conține text

Descriere generată automat

INSERT INTO scriere VALUES

(1, 1, 1);

INSERT INTO scriere VALUES

(2, 1, 2);

INSERT INTO scriere VALUES

(3, 2, 3);

INSERT INTO scriere VALUES

(4, 2, 4);

INSERT INTO scriere VALUES

(5, 3, 5);

INSERT INTO scriere VALUES

(6, 2, 5);

INSERT INTO scriere VALUES

(7, 4, 6);

INSERT INTO scriere VALUES

(8, 5, 7);

INSERT INTO scriere VALUES

(9, 5, 8);

INSERT INTO scriere VALUES

(10, 6, 9);

INSERT INTO scriere VALUES

(11, 7, 9);

INSERT INTO scriere VALUES

(12, 6, 10);

INSERT INTO scriere VALUES

(13, 6, 11);

INSERT INTO scriere VALUES

(14, 7, 12);

INSERT INTO scriere VALUES

(15, 7, 13);

INSERT INTO scriere VALUES

(16, 6, 13);

O imagine care conține text

Descriere generată automat

INSERT INTO imprumut VALUES

(1, to\_date('02-12-2021','dd-mm-yyyy'), to\_date('13-12-2021','dd-mm-yyyy'), 1, 1);

INSERT INTO imprumut VALUES

(2, to\_date('05-12-2021','dd-mm-yyyy'), to\_date('02-01-2022','dd-mm-yyyy'), 2, 1);

INSERT INTO imprumut VALUES

(3, to\_date('30-11-2021','dd-mm-yyyy'), to\_date('03-12-2021','dd-mm-yyyy'), 3, 2);

INSERT INTO imprumut VALUES

(4, to\_date('07-12-2021','dd-mm-yyyy'), to\_date('07-12-2021','dd-mm-yyyy'), 15, 3);

INSERT INTO imprumut VALUES

(5, to\_date('02-12-2021','dd-mm-yyyy'), to\_date('07-12-2021','dd-mm-yyyy'), 12, 4);

INSERT INTO imprumut VALUES

(6, to\_date('25-11-2021','dd-mm-yyyy'), to\_date('28-11-2021','dd-mm-yyyy'), 10, 4);

INSERT INTO imprumut VALUES

(7, to\_date('02-12-2021','dd-mm-yyyy'), to\_date('02-12-2021','dd-mm-yyyy'), 8, 5);

INSERT INTO imprumut VALUES

(8, to\_date('04-01-2022','dd-mm-yyyy'), to\_date('07-01-2022','dd-mm-yyyy'), 4, 1);

INSERT INTO imprumut VALUES

(9, to\_date('11-12-2021','dd-mm-yyyy'), to\_date('13-12-2021','dd-mm-yyyy'), 9, 6);

INSERT INTO imprumut VALUES

(10, to\_date('14-12-2021','dd-mm-yyyy'), to\_date('18-12-2021','dd-mm-yyyy'), 6, 7);

INSERT INTO imprumut VALUES

(11, to\_date('21-11-2021','dd-mm-yyyy'), to\_date('21-12-2021','dd-mm-yyyy'), 21, 2);

O imagine care conține text

Descriere generată automat

1. **Rezolvarea unei probleme folosind două tipuri de colecții studiate**

Cerință: Afișati, pentru fiecare tip de abonament, cititorii care îl dețin și numărul de împrumuturi efectuat de fiecare.

Am folosit două tipuri de colecție, și anume două tablouri imbricate, în care am reținut id-urile, respectiv descrierea abonamentelor, și un tablou indexat de înregistrări, în care rețin pentru fiecare cititor id-ul, prenumele, numele și tipul de abonament.

CREATE OR REPLACE PROCEDURE ex6 IS

TYPE tab\_imb\_id\_abon IS TABLE OF abonament.id\_abonament%TYPE;

t\_id\_abon tab\_imb\_id\_abon := tab\_imb\_id\_abon();

TYPE tab\_imb\_det\_abon IS TABLE OF abonament.detalii%TYPE;

t\_det\_abon tab\_imb\_det\_abon := tab\_imb\_det\_abon();

TYPE cit\_record IS RECORD

(id cititor.id\_cititor%TYPE,

prenume cititor.prenume%TYPE,

nume cititor.nume%TYPE,

abon cititor.tip\_abonament%TYPE);

TYPE tab\_cititori IS TABLE OF cit\_record INDEX BY BINARY\_INTEGER;

t\_cit tab\_cititori;

i INTEGER;

cnt INTEGER;

j INTEGER;

nr\_imprumuturi INTEGER;

BEGIN

SELECT id\_abonament

BULK COLLECT INTO t\_id\_abon

FROM abonament

ORDER BY id\_abonament;

SELECT detalii

BULK COLLECT INTO t\_det\_abon

FROM abonament

ORDER BY id\_abonament;

i := t\_id\_abon.FIRST;

j := t\_det\_abon.FIRST;

WHILE i <= t\_id\_abon.LAST LOOP

DBMS\_OUTPUT.PUT\_LINE('Abonamentul cu id-ul '||t\_id\_abon(i)||' ('||t\_det\_abon(j)||')');

SELECT id\_cititor, prenume, nume, tip\_abonament

BULK COLLECT INTO t\_cit

FROM cititor

WHERE tip\_abonament = t\_id\_abon(i);

IF t\_cit.COUNT = 0 THEN

DBMS\_OUTPUT.PUT\_LINE(' '||'Niciun cititor nu are acest tip de abonament!');

DBMS\_OUTPUT.NEW\_LINE;

ELSE

IF t\_cit.COUNT > 1 THEN

DBMS\_OUTPUT.PUT\_LINE(' '||'Cititorii care au acest tip de abonament:');

DBMS\_OUTPUT.NEW\_LINE;

ELSE

DBMS\_OUTPUT.PUT\_LINE(' '||'Cititorul care are acest tip de abonament:');

DBMS\_OUTPUT.NEW\_LINE;

END IF;

cnt := 0;

FOR k IN t\_cit.FIRST..t\_cit.LAST LOOP

cnt := cnt+1;

DBMS\_OUTPUT.PUT(' '||cnt||'. '||t\_cit(k).prenume||' '||t\_cit(k).nume);

SELECT COUNT(\*)

INTO nr\_imprumuturi

FROM IMPRUMUT

WHERE id\_cititor = t\_cit(k).id;

DBMS\_OUTPUT.PUT\_LINE(' | numar de imprumuturi: '||nr\_imprumuturi);

END LOOP;

END IF;

i:=i+1;

j:=j+1;

DBMS\_OUTPUT.NEW\_LINE;

DBMS\_OUTPUT.NEW\_LINE;

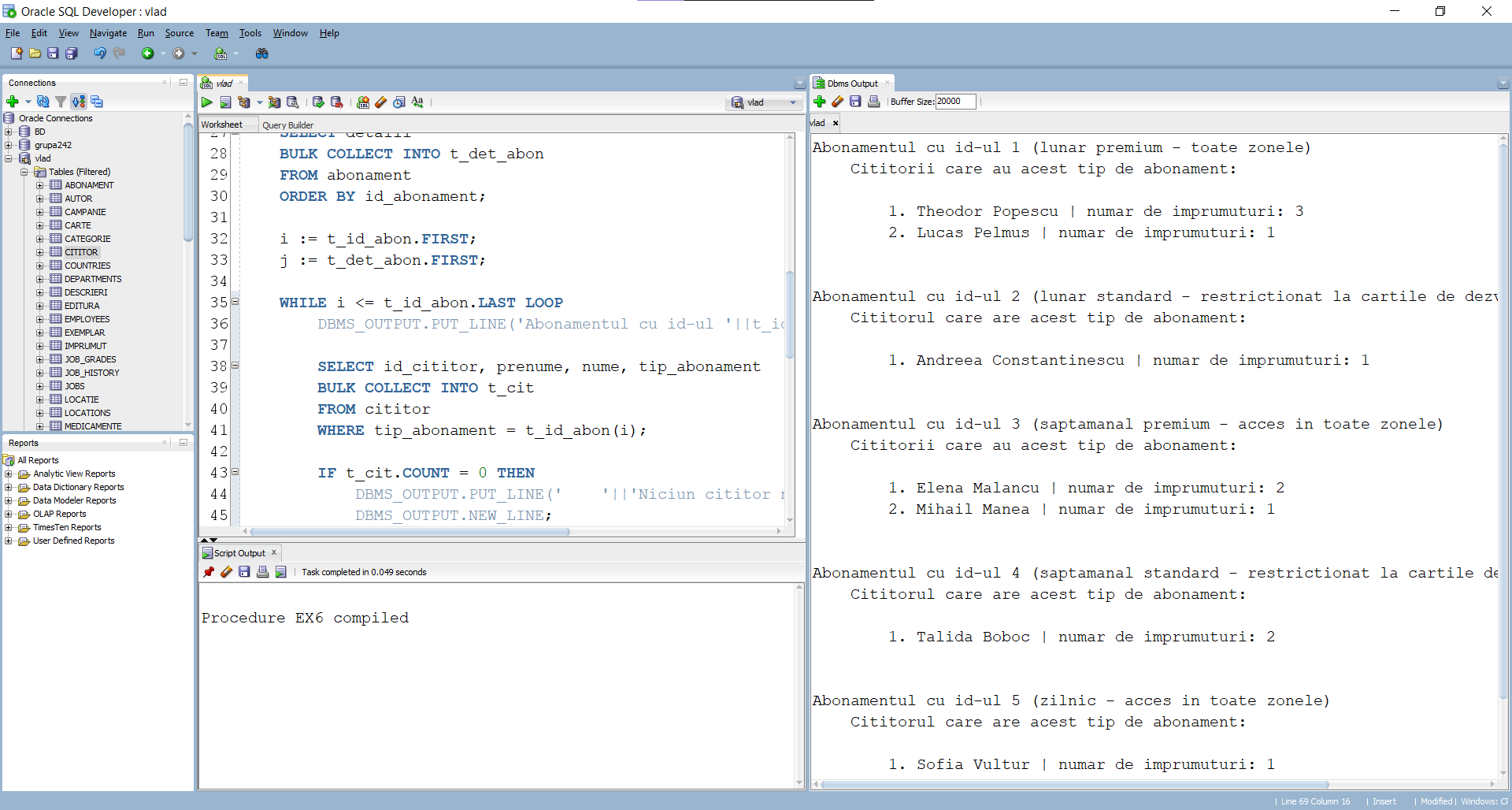
END LOOP;

END ex6;

BEGIN

ex6;

END;



1. **Rezolvarea unei probleme folosind două tipuri de colecții studiate**

Cerință: Afișați, pentru fiecare carte publicată după anul 2000, autorii acesteia și vârsta lor la momentul publicării cărții.

Am folosit un cursor în care am selectat informațiile despre cărți, iar apoi pentru fiecare carte am selectat autorii din tabelul asociativ SCRIERE și am afișat informațiile cerute.

CREATE OR REPLACE PROCEDURE ex7 IS

TYPE tab\_imb\_id\_aut IS TABLE OF autor.id\_autor%TYPE;

t\_id\_aut tab\_imb\_id\_aut := tab\_imb\_id\_aut();

v\_id\_carte carte.id\_carte%TYPE;

v\_titlu carte.titlu%TYPE;

v\_an carte.an\_aparitie%TYPE;

v\_prenume autor.prenume%TYPE;

v\_nume autor.nume%TYPE;

v\_ani NUMBER;

CURSOR c IS

SELECT id\_carte, titlu, an\_aparitie

FROM carte

WHERE an\_aparitie >= 2000;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Carti publicate dupa anul 2000');

DBMS\_OUTPUT.NEW\_LINE;

OPEN c;

LOOP

FETCH c INTO v\_id\_carte, v\_titlu, v\_an;

EXIT WHEN c%NOTFOUND;

SELECT id\_autor

BULK COLLECT INTO t\_id\_aut

FROM scriere

WHERE id\_carte = v\_id\_carte;

DBMS\_OUTPUT.PUT\_LINE(v\_titlu||' ('||v\_an||') - scrisa de:');

FOR i IN t\_id\_aut.FIRST..t\_id\_aut.LAST LOOP

SELECT prenume, nume, (v\_an - EXTRACT (YEAR FROM data\_nasterii))

INTO v\_prenume, v\_nume, v\_ani

FROM autor

WHERE id\_autor = t\_id\_aut(i);

DBMS\_OUTPUT.PUT\_LINE(' '||v\_prenume||' '||v\_nume||' ('||v\_ani||' ani)');

END LOOP;

DBMS\_OUTPUT.NEW\_LINE;

DBMS\_OUTPUT.NEW\_LINE;

END LOOP;

CLOSE c;

END ex7;

BEGIN

ex7;

END;

**O imagine care conține text

Descriere generată automat**

1. **Rezolvarea unei probleme utilizând într-o comandă 3 tabele**

Cerință: Afișați, folosind o funcție stocată, numărul de exemplare care trebuie înlocuite (sunt deteriorate) ale cărților scrise de un autor al cărui nume este dat ca parametru.

Am selectat numărul de exemplare care îndeplinesc condiția, folosind operația de JOIN pe 3 tabele (SCRIERE, CARTE și EXEMPLAR).

--Pentru a exemplifica tratarea exceptiilor, am mai inserat

--urmatoarea linie in tabelul AUTOR

INSERT INTO autor VALUES

(8, 'John', 'King', to\_date('26-10-1956', 'dd-mm-yyyy'));

CREATE OR REPLACE FUNCTION ex8 (nume\_aut autor.nume%TYPE) RETURN NUMBER IS

id\_aut autor.id\_autor%TYPE;

numar NUMBER;

BEGIN

SELECT id\_autor

INTO id\_aut

FROM autor

WHERE UPPER(nume) = UPPER(nume\_aut);

SELECT COUNT(conditie)

INTO numar

FROM exemplar e, carte c, scriere s

WHERE c.id\_carte = s.id\_carte AND c.id\_carte = e.id\_carte

AND s.id\_autor = id\_aut AND

UPPER(conditie) = 'DETERIORATA';

RETURN numar;

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Au fost gasiti mai multi autori cu numele introdus!');

RETURN -1;

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu a fost gasit autorul cu numele dat!');

RETURN -1;

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('A aparut o eroare!');

RETURN -1;

END ex8;

DECLARE

nume\_aut autor.nume%TYPE := '&nume';

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Numarul de exemplare ale cartilor scrise de '||UPPER(nume\_aut)||' care trebuie inlocuite: '||ex8(nume\_aut));

END;

--am introdus de la tastatura numele Tolkien (functionare corecta)

O imagine care conține text

Descriere generată automat

--am introdus de la tastatura numele King (se intra pe exceptia TOO\_MANY\_ROWS)

O imagine care conține text

Descriere generată automat

--am introdus de la tastatura numele Popescu (se intra pe exceptia NO\_DATA\_FOUND)

O imagine care conține text

Descriere generată automat

1. **Rezolvarea unei probleme utilizând într-o comandă 5 tabele**

Cerință: Afișați, folosind o procedură stocată, pentru cititorii care au împrumutat cărți scrise doar de un autor, cărțile împrumutate, precum și categoria din care face parte fiecare.

Am folosit un tablou imbricat de înregistrări, în care am reținut cărțile, categoria și autorul pentru cărțile împrumutate de fiecare cititor, făcând astfel operația de JOIN pe 6 tabele (CITITOR, ÎMPRUMUT, EXEMPLAR, CARTE, SCRIERE, AUTOR).

--Pentru a exemplifica tratarea exceptiilor, am mai inserat

--urmatoarele linii in tabelul CITITOR, respectiv IMPRUMUT

INSERT INTO cititor VALUES

(8, 'Andrei', 'Popescu', '0752421243', 'andrei2001.p@hotmail.com', 1);

INSERT INTO cititor VALUES

(9, 'Cosmin', 'Stefan', '0752433122', 'cosminstefann@gmail.com', 2);

INSERT INTO imprumut VALUES

(12, to\_date('14-12-2021','dd-mm-yyyy'), to\_date('18-12-2021','dd-mm-yyyy'), 6, 8);

CREATE OR REPLACE PROCEDURE ex9 (nume\_cit cititor.nume%TYPE) IS

TYPE tab\_imb\_id\_aut IS TABLE OF autor.id\_autor%TYPE;

TYPE carte\_categ IS RECORD

(id carte.id\_carte%TYPE,

titlu carte.titlu%TYPE,

categ categorie.nume%TYPE,

id\_aut autor.id\_autor %TYPE);

TYPE tab\_carti IS TABLE OF carte\_categ;

t\_carti tab\_carti := tab\_carti();

id\_cit cititor.id\_cititor%TYPE;

id\_aut autor.id\_autor%TYPE;

exc1 exception;

nume\_aut autor.nume%TYPE;

prenume\_aut autor.prenume%TYPE;

BEGIN

SELECT id\_cititor

INTO id\_cit

FROM cititor

WHERE nume = nume\_cit;

SELECT DISTINCT(c.id\_carte), c.titlu, cat.nume, a.id\_autor

BULK COLLECT INTO t\_carti

FROM categorie cat, carte c, exemplar e, imprumut i, scriere s, autor a

WHERE cat.id\_categorie = c.id\_categorie AND

c.id\_carte = e.id\_carte AND e.id\_exemplar = i.id\_exemplar AND

c.id\_carte = s.id\_carte AND s.id\_autor = a.id\_autor AND

c.id\_carte IN (SELECT MAX(id\_carte)

FROM scriere

GROUP BY id\_carte

HAVING COUNT(id\_carte) = 1) AND

i.id\_cititor = id\_cit;

IF t\_carti.COUNT = 0 THEN

RAISE exc1;

END IF;

DBMS\_OUTPUT.PUT\_LINE('Cititorul cu numele de '||nume\_cit||' a imprumutat:');

FOR i IN t\_carti.FIRST..t\_carti.LAST LOOP

DBMS\_OUTPUT.PUT(' '||t\_carti(i).titlu||' ('||t\_carti(i).categ||') - scrisa de:');

SELECT nume, prenume

INTO nume\_aut, prenume\_aut

FROM autor

WHERE id\_autor = t\_carti(i).id\_aut;

DBMS\_OUTPUT.PUT\_LINE(' '||prenume\_aut||' '||nume\_aut);

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20000, 'Nu a fost gasit cititorul cu numele '||nume\_cit||'!');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Au fost gasiti mai multi cititori cu numele '||nume\_cit||'!');

WHEN exc1 THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Cititorul cu numele '||nume\_cit||' nu a imprumutat nicio carte!');

END ex9;

BEGIN --functioneaza

ex9('Malancu');

END;

O imagine care conține text

Descriere generată automat

BEGIN --intra pe exceptia de TOO\_MANY\_ROWS

ex9('Popescu');

END;

O imagine care conține text

Descriere generată automat

BEGIN --intra pe exceptia predefinita pentru cazul in care utilizatorul

ex9('Stefan'); --nu a imprumutat nicio carte

END;

O imagine care conține text

Descriere generată automat

BEGIN --intra pe exceptia de NO\_DATA\_FOUND

ex9('John');

END;

O imagine care conține text

Descriere generată automat

1. **Trigger LMD la nivel de comandă**

CREATE OR REPLACE TRIGGER trig\_ex10

BEFORE INSERT OR DELETE OR UPDATE ON imprumut

BEGIN

IF (TO\_CHAR(SYSDATE,'D') = 1) OR (TO\_CHAR(SYSDATE,'D') = 7)

OR (TO\_CHAR(SYSDATE,'HH24') NOT BETWEEN 8 AND 16)

THEN

IF INSERTING THEN

RAISE\_APPLICATION\_ERROR(-20001,'Inserarea in tabelul IMPRUMUT

este permisa doar in timpul programului de lucru!');

ELSIF DELETING THEN

RAISE\_APPLICATION\_ERROR(-20002,'Stergerea din tabelul IMPRUMUT

este permisa doar in timpul programului de lucru!');

ELSE

RAISE\_APPLICATION\_ERROR(-20003,'Actualizarea in tabelul IMPRUMUT

este permisa doar in timpul programului de lucru!');

END IF;

END IF;

END;

O imagine care conține text

Descriere generată automat

1. **Trigger LMD la nivel de linie**

CREATE OR REPLACE PROCEDURE proc\_trig\_ex11

IS

BEGIN

RAISE\_APPLICATION\_ERROR (-20000, 'Nu puteti modifica descriere abonamentelor!');

END;

CREATE OR REPLACE TRIGGER trig\_ex11

BEFORE UPDATE OF detalii ON abonament

FOR EACH ROW

WHEN (NEW.detalii <> OLD.detalii)

BEGIN

proc\_trig\_ex11;

END;

O imagine care conține text

Descriere generată automat

1. **Trigger LDD**

CREATE OR REPLACE TRIGGER trig\_ex12

AFTER CREATE OR DROP OR ALTER ON DATABASE

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Baza de date: '||SYS.DATABASE\_NAME);

DBMS\_OUTPUT.PUT\_LINE('User: '||SYS.LOGIN\_USER);

DBMS\_OUTPUT.PUT\_LINE('Eveniment: '||SYS.SYSEVENT);

DBMS\_OUTPUT.PUT\_LINE('Tip obiect referit: '||SYS.DICTIONARY\_OBJ\_TYPE);

DBMS\_OUTPUT.PUT\_LINE('Nume obiect referit: '||SYS.DICTIONARY\_OBJ\_NAME);

DBMS\_OUTPUT.PUT\_LINE('Data: '||SYSTIMESTAMP);

END;

/

O imagine care conține text

Descriere generată automat

1. **Pachet care conține toate obiectele definite**

CREATE OR REPLACE PACKAGE proiect AS

PROCEDURE ex6;

PROCEDURE ex7;

FUNCTION ex8 (nume\_aut autor.nume%TYPE) RETURN NUMBER;

PROCEDURE ex9 (nume\_cit cititor.nume%TYPE);

END proiect;

CREATE OR REPLACE PACKAGE BODY proiect AS

PROCEDURE ex6 IS

TYPE tab\_imb\_id\_abon IS TABLE OF abonament.id\_abonament%TYPE;

t\_id\_abon tab\_imb\_id\_abon := tab\_imb\_id\_abon();

TYPE tab\_imb\_det\_abon IS TABLE OF abonament.detalii%TYPE;

t\_det\_abon tab\_imb\_det\_abon := tab\_imb\_det\_abon();

TYPE cit\_record IS RECORD

(id cititor.id\_cititor%TYPE,

prenume cititor.prenume%TYPE,

nume cititor.nume%TYPE,

abon cititor.tip\_abonament%TYPE);

TYPE tab\_cititori IS TABLE OF cit\_record INDEX BY BINARY\_INTEGER;

t\_cit tab\_cititori;

i INTEGER;

cnt INTEGER;

j INTEGER;

nr\_imprumuturi INTEGER;

BEGIN

SELECT id\_abonament

BULK COLLECT INTO t\_id\_abon

FROM abonament

ORDER BY id\_abonament;

SELECT detalii

BULK COLLECT INTO t\_det\_abon

FROM abonament

ORDER BY id\_abonament;

i := t\_id\_abon.FIRST;

j := t\_det\_abon.FIRST;

WHILE i <= t\_id\_abon.LAST LOOP

DBMS\_OUTPUT.PUT\_LINE('Abonamentul cu id-ul '||t\_id\_abon(i)||' ('||t\_det\_abon(j)||')');

SELECT id\_cititor, prenume, nume, tip\_abonament

BULK COLLECT INTO t\_cit

FROM cititor

WHERE tip\_abonament = t\_id\_abon(i);

IF t\_cit.COUNT = 0 THEN

DBMS\_OUTPUT.PUT\_LINE(' '||'Niciun cititor nu are acest tip de abonament!');

DBMS\_OUTPUT.NEW\_LINE;

ELSE

IF t\_cit.COUNT > 1 THEN

DBMS\_OUTPUT.PUT\_LINE(' '||'Cititorii care au acest tip de abonament:');

DBMS\_OUTPUT.NEW\_LINE;

ELSE

DBMS\_OUTPUT.PUT\_LINE(' '||'Cititorul care are acest tip de abonament:');

DBMS\_OUTPUT.NEW\_LINE;

END IF;

cnt := 0;

FOR k IN t\_cit.FIRST..t\_cit.LAST LOOP

cnt := cnt+1;

DBMS\_OUTPUT.PUT(' '||cnt||'. '||t\_cit(k).prenume||' '||t\_cit(k).nume);

SELECT COUNT(\*)

INTO nr\_imprumuturi

FROM IMPRUMUT

WHERE id\_cititor = t\_cit(k).id;

DBMS\_OUTPUT.PUT\_LINE(' | numar de imprumuturi: '||nr\_imprumuturi);

END LOOP;

END IF;

i:=i+1;

j:=j+1;

DBMS\_OUTPUT.NEW\_LINE;

DBMS\_OUTPUT.NEW\_LINE;

END LOOP;

END ex6;

PROCEDURE ex7 IS

TYPE tab\_imb\_id\_aut IS TABLE OF autor.id\_autor%TYPE;

t\_id\_aut tab\_imb\_id\_aut := tab\_imb\_id\_aut();

v\_id\_carte carte.id\_carte%TYPE;

v\_titlu carte.titlu%TYPE;

v\_an carte.an\_aparitie%TYPE;

v\_prenume autor.prenume%TYPE;

v\_nume autor.nume%TYPE;

v\_ani NUMBER;

CURSOR c IS

SELECT id\_carte, titlu, an\_aparitie

FROM carte

WHERE an\_aparitie >= 2000;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Carti publicate dupa anul 2000');

DBMS\_OUTPUT.NEW\_LINE;

OPEN c;

LOOP

FETCH c INTO v\_id\_carte, v\_titlu, v\_an;

EXIT WHEN c%NOTFOUND;

SELECT id\_autor

BULK COLLECT INTO t\_id\_aut

FROM scriere

WHERE id\_carte = v\_id\_carte;

DBMS\_OUTPUT.PUT\_LINE(v\_titlu||' ('||v\_an||') - scrisa de:');

FOR i IN t\_id\_aut.FIRST..t\_id\_aut.LAST LOOP

SELECT prenume, nume, (v\_an - EXTRACT (YEAR FROM data\_nasterii))

INTO v\_prenume, v\_nume, v\_ani

FROM autor

WHERE id\_autor = t\_id\_aut(i);

DBMS\_OUTPUT.PUT\_LINE(' '||v\_prenume||' '||v\_nume||' ('||v\_ani||' ani)');

END LOOP;

DBMS\_OUTPUT.NEW\_LINE;

DBMS\_OUTPUT.NEW\_LINE;

END LOOP;

CLOSE c;

END ex7;

FUNCTION ex8 (nume\_aut autor.nume%TYPE) RETURN NUMBER IS

id\_aut autor.id\_autor%TYPE;

numar NUMBER;

BEGIN

SELECT id\_autor

INTO id\_aut

FROM autor

WHERE UPPER(nume) = UPPER(nume\_aut);

SELECT COUNT(conditie)

INTO numar

FROM exemplar e, carte c, scriere s

WHERE c.id\_carte = s.id\_carte AND c.id\_carte = e.id\_carte

AND s.id\_autor = id\_aut AND

UPPER(conditie) = 'DETERIORATA';

RETURN numar;

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Au fost gasiti mai multi autori cu numele introdus!');

RETURN -1;

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu a fost gasit autorul cu numele dat!');

RETURN -1;

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('A aparut o eroare!');

RETURN -1;

END ex8;

PROCEDURE ex9 (nume\_cit cititor.nume%TYPE) IS

TYPE tab\_imb\_id\_aut IS TABLE OF autor.id\_autor%TYPE;

TYPE carte\_categ IS RECORD

(id carte.id\_carte%TYPE,

titlu carte.titlu%TYPE,

categ categorie.nume%TYPE,

id\_aut autor.id\_autor %TYPE);

TYPE tab\_carti IS TABLE OF carte\_categ;

t\_carti tab\_carti := tab\_carti();

id\_cit cititor.id\_cititor%TYPE;

id\_aut autor.id\_autor%TYPE;

exc1 exception;

nume\_aut autor.nume%TYPE;

prenume\_aut autor.prenume%TYPE;

BEGIN

SELECT id\_cititor

INTO id\_cit

FROM cititor

WHERE nume = nume\_cit;

SELECT DISTINCT(c.id\_carte), c.titlu, cat.nume, a.id\_autor

BULK COLLECT INTO t\_carti

FROM categorie cat, carte c, exemplar e, imprumut i, scriere s, autor a

WHERE cat.id\_categorie = c.id\_categorie AND

c.id\_carte = e.id\_carte AND e.id\_exemplar = i.id\_exemplar AND

c.id\_carte = s.id\_carte AND s.id\_autor = a.id\_autor AND

c.id\_carte IN (SELECT MAX(id\_carte)

FROM scriere

GROUP BY id\_carte

HAVING COUNT(id\_carte) = 1) AND

i.id\_cititor = id\_cit;

IF t\_carti.COUNT = 0 THEN

RAISE exc1;

END IF;

DBMS\_OUTPUT.PUT\_LINE('Cititorul cu numele de '||nume\_cit||' a imprumutat:');

FOR i IN t\_carti.FIRST..t\_carti.LAST LOOP

DBMS\_OUTPUT.PUT(' '||t\_carti(i).titlu||' ('||t\_carti(i).categ||') - scrisa de:');

SELECT nume, prenume

INTO nume\_aut, prenume\_aut

FROM autor

WHERE id\_autor = t\_carti(i).id\_aut;

DBMS\_OUTPUT.PUT\_LINE(' '||prenume\_aut||' '||nume\_aut);

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20000, 'Nu a fost gasit cititorul cu numele '||nume\_cit||'!');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Au fost gasiti mai multi cititori cu numele '||nume\_cit||'!');

WHEN exc1 THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Cititorul cu numele '||nume\_cit||' nu a imprumutat nicio carte!');

END ex9;

END proiect;

BEGIN

proiect.ex6;

proiect.ex7;

DBMS\_OUTPUT.PUT\_LINE('Numarul de exemplare ale cartilor scrise de Tolkien care trebuie inlocuite: '||proiect.ex8('Tolkien'));

proiect.ex9('Malancu');

END;

O imagine care conține text

Descriere generată automat