EtnoWeb

Arhitectura bazei de date

Dinu Georgiana

Laslau Laurentiu

Rapa Mihaela

Toma Lucian

Cuprins

**Descriere tabele si constrangeri**...............................................3

**Pachete**

* Pachetul full\_info............................................................................4
* Pachetul p\_gallery\_info..................................................................8
* Pachetul price\_filter........................................................................9
* Pachetul count\_results...................................................................13
* Tipurile returnate de functii...........................................................15

**Triggere**.............................................................................................16

**Indecsi**................................................................................................17

**Descriere tabele si constrangeri**

Prima tabela este tabela **muzeu** care contine un id muzeu unic pentru fiecare inregistrare, numele muzeului si pretul de intrare.

La randul sau, fiecare muzeu contine mai multe ansambluri, de aceea avem creata tabela **ansamblu** cu urmatoarele campuri: id-ul muzeului din care face parte, un id ansamblu unic, denumirea ansamblului, denumirea sa in muzeu, denumirea sa la origine, etnia, datarea, descrierea si imprejmuirile.

Avem ,de asemenea ,tabela **locatii**, care contine toate locatiile posibile ale ansamblurilor, formate din zona de provenienta si locatie de provenienta, fiecare pereche avand un id locatie unic.

Tabela **locatie\_ansamblu** este formata dintr-o pereche <id\_ansamblu,id\_locatie>, pentru a evita datele redundante.

Pentru ca vizitatorii nostrii sunt de toate varstele si unii din ei inca sunt in invatamantul statului, am creat tabela **preturi** care are preturi diverse pt fiecare persoana ( angajati, studenti , scolari). Fiecare muzeu are un pret asignat , pret care se reflecta si in ansamblurile constinute de acesta prin chei straine.

Fiecare ansamblu poate avea asociata o bibliografie si o imagine, de aceea am construit tabela **informatii** care sa contine aceste doua campuri si id-ul ansamblului.

ALTER TABLE locatii ADD CONSTRAINT pk\_id\_locatie PRIMARY KEY (id\_locatie);

ALTER TABLE turisti ADD CONSTRAINT ck\_nume\_turist CHECK (nume = INITCAP (nume));

ALTER TABLE turisti ADD CONSTRAINT ck\_prenume CHECK (prenume = INITCAP (prenume));

ALTER TABLE preturi MODIFY (pret\_angajat CONSTRAINT nn\_pret\_angajat NOT NULL);

ALTER TABLE informatii ADD CONSTRAINT fk\_informatii FOREIGN KEY (id\_ansamblu) REFERENCES ansamblu (id\_ansamblu);

ALTER TABLE locatie\_ansamblu ADD CONSTRAINT fk\_locatie\_ans FOREIGN KEY (id\_ansamblu) REFERENCES ansamblu (id\_ansamblu);

ALTER TABLE locatie\_ansamblu ADD CONSTRAINT fk\_locatie\_ans\_id\_loc FOREIGN KEY (id\_locatie) REFERENCES locatii (id\_locatie);

ALTER TABLE preturi ADD CONSTRAINT fk\_preturi FOREIGN KEY (id\_muzeu) REFERENCES muzeu (id\_muzeu);

**Pachete**

-Pachetul full\_info

-functia **getFullInfoByValueAndColumn** are ca parametrii un cuvant cheie dupa care userul doreste sa se efectueze cautarea, coloana in care se cauta cuvantul introdus , si min\_rownum si max\_rownum pe care ii folosim la paginare si afiseaza un anumit numar de rezultate in functie de intervalul dat

-functia **getGalleryInfoByOffset** este folosita pentru a returna un numar exact de informatii pentru galerie

- functia **getDetailsById** primeste un id si returneaza detalii ale ansamblului cu id-ul respectiv

CREATE OR REPLACE PACKAGE full\_info IS

FUNCTION getFullInfoByValueAndColumn (p\_value IN varchar, p\_column IN varchar2, p\_min integer, p\_max integer) RETURN full\_info\_object\_array;

FUNCTION getGalleryInfoByOffset (p\_min integer, p\_max integer)RETURN gallery\_info\_object\_array;

FUNCTION getDetailsById (p\_id varchar) return full\_info\_object\_array;

END full\_info;

/

CREATE OR REPLACE PACKAGE BODY full\_info IS

FUNCTION getFullInfoByValueAndColumn (p\_value IN varchar, p\_column IN varchar2, p\_min integer, p\_max integer)RETURN full\_info\_object\_array AS

v\_id number(4);

v\_denumire\_ansamblu varchar(129);

v\_denumire\_in\_muzeu varchar(129);

v\_denumire\_la\_origine varchar(60);

v\_pret\_bilet number(4,2);

v\_nume\_muzeu varchar2(90);

v\_etnia varchar(24);

v\_datarea varchar(193);

v\_descriere varchar(940);

v\_imprejmuiri varchar(960);

v\_zona\_prov varchar(43);

v\_localitate\_prov varchar(100);

v\_bibliografie varchar(312);

v\_url varchar(86);

v\_rown integer;

v\_select varchar2(1000);

v\_pagination\_s varchar2(2000);

--cursori

TYPE cursor\_type IS REF CURSOR;

fullinfo cursor\_type;

one\_record info\_object := info\_object();

all\_records full\_info\_object\_array := new full\_info\_object\_array();

counter integer :=0;

BEGIN

v\_select := 'SELECT ansamblu.id\_ansamblu, denumirea\_ansamblului, denumirea\_in\_muzeu, denumirea\_la\_origine, pret\_angajat, nume, etnia, datarea, descriere, imprejmuiri, zona\_provenienta, localitate\_provenienta, bibliografie, url\_imagine ' ||

'FROM (((((ansamblu JOIN locatie\_ansamblu ON ansamblu.id\_ansamblu = locatie\_ansamblu.id\_ansamblu) ' ||

'JOIN locatii ON locatie\_ansamblu.id\_locatie = locatii.id\_locatie) ' ||

'JOIN informatii ON ansamblu.id\_ansamblu = informatii.id\_ansamblu) ' ||

'JOIN muzeu ON ansamblu.id\_muzeu = muzeu.id\_muzeu) ' ||

'JOIN preturi ON preturi.id\_muzeu = muzeu.id\_muzeu) WHERE lower('|| p\_column || ') like lower(''%' || p\_value ||'%'')';

v\_pagination\_s := 'select \* from (select rownum rown, a.\* from ( ' || v\_select || ' ) a where rownum <= :p\_max ) where rown >= :p\_min ';

OPEN fullinfo FOR v\_pagination\_s USING p\_max, p\_min;

LOOP

FETCH fullinfo INTO v\_rown, v\_id, v\_denumire\_ansamblu, v\_denumire\_in\_muzeu, v\_denumire\_la\_origine, v\_pret\_bilet, v\_nume\_muzeu, v\_etnia, v\_datarea, v\_descriere, v\_imprejmuiri, v\_zona\_prov, v\_localitate\_prov, v\_bibliografie, v\_url;

EXIT WHEN fullinfo%notfound;

IF(v\_bibliografie is null) THEN

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

v\_bibliografie := 'Nu exista blibiografie de afisat';--RAISE null\_bibliografie;

END IF;

one\_record.denumire\_ansamblu := v\_denumire\_ansamblu;

one\_record.id\_ansamblu := v\_id;

one\_record.denumire\_in\_muzeu := v\_denumire\_in\_muzeu;

one\_record.denumire\_la\_origine := v\_denumire\_la\_origine;

one\_record.pret\_bilet := v\_pret\_bilet;

one\_record.nume\_muzeu := v\_nume\_muzeu;

one\_record.etnia := v\_etnia;

one\_record.datarea := v\_datarea;

one\_record.descriere := v\_descriere;

one\_record.imprejmuiri := v\_imprejmuiri;

one\_record.zona\_prov := v\_zona\_prov;

one\_record.localitate\_prov := v\_localitate\_prov;

one\_record.bibliografie := v\_bibliografie;

one\_record.url := v\_url;

counter := counter + 1;

all\_records.extend;

all\_records(counter) := one\_record;

END LOOP;

CLOSE fullinfo;

return all\_records;

END getFullInfoByValueAndColumn;

FUNCTION getGalleryInfoByOffset (p\_min integer, p\_max integer)RETURN gallery\_info\_object\_array AS

v\_id number(4);

v\_denumire\_ansamblu varchar(129);

v\_denumire\_in\_muzeu varchar(129);

v\_denumire\_la\_origine varchar(60);

v\_url varchar(86);

v\_rown integer;

v\_select varchar2(1000);

v\_pagination\_s varchar2(2000);

--cursori

TYPE cursor\_type IS REF CURSOR;

fullinfo cursor\_type;

one\_record gallery\_info\_object := gallery\_info\_object();

all\_records gallery\_info\_object\_array := new gallery\_info\_object\_array();

counter integer :=0;

BEGIN

v\_select := 'SELECT ansamblu.id\_ansamblu, denumirea\_ansamblului, denumirea\_in\_muzeu, denumirea\_la\_origine, url\_imagine ' ||

'FROM (((((ansamblu JOIN locatie\_ansamblu ON ansamblu.id\_ansamblu = locatie\_ansamblu.id\_ansamblu) ' ||

'JOIN locatii ON locatie\_ansamblu.id\_locatie = locatii.id\_locatie) ' ||

'JOIN informatii ON ansamblu.id\_ansamblu = informatii.id\_ansamblu) ' ||

'JOIN muzeu ON ansamblu.id\_muzeu = muzeu.id\_muzeu) ' ||

'JOIN preturi ON preturi.id\_muzeu = muzeu.id\_muzeu)';

v\_pagination\_s := 'select \* from (select rownum rown, a.\* from ( ' || v\_select || ' ) a where rownum <= :p\_max ) where rown >= :p\_min ';

OPEN fullinfo FOR v\_pagination\_s USING p\_max, p\_min;

LOOP

FETCH fullinfo INTO v\_rown, v\_id, v\_denumire\_ansamblu, v\_denumire\_in\_muzeu, v\_denumire\_la\_origine, v\_url;

EXIT WHEN fullinfo%notfound;

one\_record.denumire\_ansamblu := v\_denumire\_ansamblu;

one\_record.id\_ansamblu := v\_id;

one\_record.denumire\_in\_muzeu := v\_denumire\_in\_muzeu;

one\_record.denumire\_la\_origine := v\_denumire\_la\_origine;

one\_record.url := v\_url;

counter := counter + 1;

all\_records.extend;

all\_records(counter) := one\_record;

END LOOP;

CLOSE fullinfo;

return all\_records;

END getGalleryInfoByOffset;

FUNCTION getDetailsById (p\_id varchar)RETURN full\_info\_object\_array AS

v\_id number(4);

v\_denumire\_ansamblu varchar(129);

v\_denumire\_in\_muzeu varchar(129);

v\_denumire\_la\_origine varchar(60);

v\_pret\_bilet number(4,2);

v\_nume\_muzeu varchar2(90);

v\_etnia varchar(24);

v\_datarea varchar(193);

v\_descriere varchar(940);

v\_imprejmuiri varchar(960);

v\_zona\_prov varchar(43);

v\_localitate\_prov varchar(100);

v\_bibliografie varchar(312);

v\_url varchar(86);

v\_select varchar2(1000);

TYPE cursor\_type IS REF CURSOR;

fullinfo cursor\_type;

one\_record info\_object := info\_object(); --obiect de tip info\_object (structura ce contine toate campurile returnate pentru un singur ansamblu)

all\_records full\_info\_object\_array := new full\_info\_object\_array(); --array de obiecte care contine informatiile pentru TOATE ansamblurile care corespund cautarii

counter integer :=0;

BEGIN

v\_select := 'SELECT DISTINCT ansamblu.id\_ansamblu, denumirea\_ansamblului, denumirea\_in\_muzeu, denumirea\_la\_origine, pret\_angajat, nume, etnia, datarea, descriere, imprejmuiri, zona\_provenienta, localitate\_provenienta, bibliografie, url\_imagine ' ||

'INTO :id\_a, :denum, :den\_muz, :den\_org, :pret, :nume, :etnie, :datare, :descr, :impr, :zona\_prv, :loc\_prv, :bibl ' ||

'FROM (((((ansamblu JOIN locatie\_ansamblu ON ansamblu.id\_ansamblu = locatie\_ansamblu.id\_ansamblu) ' ||

'JOIN locatii ON locatie\_ansamblu.id\_locatie = locatii.id\_locatie)' ||

'JOIN informatii ON ansamblu.id\_ansamblu = informatii.id\_ansamblu) ' ||

'JOIN muzeu ON ansamblu.id\_muzeu = muzeu.id\_muzeu) '||

'JOIN preturi ON muzeu.id\_muzeu = preturi.id\_muzeu) WHERE ansamblu.id\_ansamblu = '|| p\_id;

EXECUTE IMMEDIATE v\_select INTO v\_id, v\_denumire\_ansamblu, v\_denumire\_in\_muzeu, v\_denumire\_la\_origine, v\_pret\_bilet, v\_nume\_muzeu, v\_etnia, v\_datarea, v\_descriere, v\_imprejmuiri, v\_zona\_prov, v\_localitate\_prov, v\_bibliografie, v\_url;

IF(v\_bibliografie is null) THEN

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

v\_bibliografie := 'Nu exista blibiografie de afisat';

END IF;

one\_record.denumire\_ansamblu := v\_denumire\_ansamblu;

one\_record.id\_ansamblu := v\_id;

one\_record.denumire\_in\_muzeu := v\_denumire\_in\_muzeu;

one\_record.denumire\_la\_origine := v\_denumire\_la\_origine;

one\_record.pret\_bilet := v\_pret\_bilet;

one\_record.nume\_muzeu := v\_nume\_muzeu;

one\_record.etnia := v\_etnia;

one\_record.datarea := v\_datarea;

one\_record.descriere := v\_descriere;

one\_record.imprejmuiri := v\_imprejmuiri;

one\_record.zona\_prov := v\_zona\_prov;

one\_record.localitate\_prov := v\_localitate\_prov;

one\_record.bibliografie := v\_bibliografie;

one\_record.url := v\_url;

counter := counter + 1;

all\_records.extend;

all\_records(counter) := one\_record;

return all\_records;

END getDetailsById;

END full\_info;

/

-Pachetul p\_gallery\_info

-functia get\_info primeste ca parametrii un cuvant cheie si coloana in care este cautat cuvantul cheie

create or replace type t\_obj as object (v\_url varchar2(500),v\_info1 varchar2(500), v\_info2 varchar2(500),CONSTRUCTOR FUNCTION t\_obj RETURN SELF AS RESULT);

/

create or replace type body t\_obj as

CONSTRUCTOR FUNCTION t\_obj RETURN SELF AS RESULT AS

BEGIN

RETURN;

END;

end ;

/

create or replace type t\_table as table of t\_obj;

/

CREATE OR REPLACE PACKAGE p\_gallery\_info AS

FUNCTION get\_info(v\_word VARCHAR2, v\_column VARCHAR2) RETURN t\_table ;

END p\_gallery\_info;

/

CREATE OR REPLACE PACKAGE BODY p\_gallery\_info IS

nu\_exista\_date exception;

PRAGMA EXCEPTION\_INIT(nu\_exista\_date, -20001);

FUNCTION get\_info(v\_word VARCHAR2, v\_column VARCHAR2)

RETURN t\_table IS

v\_obj t\_obj;

sel varchar2(200);

type curs is ref cursor;

c1 curs;

v\_url varchar2(100);

v\_da varchar2(200);

v\_ta varchar2(100);

v\_dm varchar2(200);

counter integer :=0;

all\_url t\_table := t\_table();

BEGIN

sel := 'SELECT url\_imagine, denumirea\_ansamblului,tipul\_ansamblului,denumirea\_in\_muzeu '||'FROM informatii ' ||'JOIN ansamblu ON ansamblu.id\_ansamblu = informatii.id\_ansamblu ' ||'WHERE '|| v\_column || ' like''%' || v\_word ||'%''';

open c1 for sel;

loop

fetch c1 into v\_url,v\_da,v\_ta,v\_dm;

EXIT WHEN c1%notfound;

v\_obj := t\_obj();

v\_obj.v\_url := v\_url;

if(v\_da is not null) then

v\_obj.v\_info1:=v\_da;

else

v\_obj.v\_info1:=v\_ta;

end if;

v\_obj.v\_info2:=v\_dm;

counter:=counter+1;

all\_url.extend;

all\_url(counter):=v\_obj;

end loop;

close c1;

if(v\_dm is null) then

raise nu\_exista\_date;

else

return all\_url;

end if;

EXCEPTION

WHEN nu\_exista\_date THEN

raise\_application\_error (-20001,'Nu exista date care contin"'||v\_word||'".');

END get\_info;

END p\_gallery\_info;

/

-Pachetul price\_filter

-functia **getAttractionsByPriceRange** este folosita cand userul face cautarea doar dupa pret

-functia **getAttractionsByWordAndPrice** se foloseste cand userul cauta si dupa un cuvant cheie si dupa pret

CREATE OR REPLACE PACKAGE price\_filter IS

FUNCTION getAttractionsByPriceRange (p\_type varchar, p\_min\_price integer, p\_max\_price integer, p\_min\_rownum integer, p\_max\_rownum integer) RETURN full\_info\_object\_array;

FUNCTION getAttractionsByWordAndPrice (p\_value IN varchar, p\_column IN varchar2, p\_min\_price integer, p\_max\_price integer, p\_price\_category varchar, p\_min integer, p\_max integer)RETURN full\_info\_object\_array;

END price\_filter;

/

CREATE OR REPLACE PACKAGE BODY price\_filter IS

FUNCTION getAttractionsByPriceRange (p\_type varchar, p\_min\_price integer, p\_max\_price integer, p\_min\_rownum integer, p\_max\_rownum integer) RETURN full\_info\_object\_array AS

v\_id number(4);

v\_denumire\_ansamblu varchar(129);

v\_denumire\_in\_muzeu varchar(129);

v\_denumire\_la\_origine varchar(60);

v\_pret\_bilet number(4,2);

v\_nume\_muzeu varchar2(90);

v\_etnia varchar(24);

v\_datarea varchar(193);

v\_descriere varchar(940);

v\_imprejmuiri varchar(960);

v\_zona\_prov varchar(43);

v\_localitate\_prov varchar(100);

v\_bibliografie varchar(312);

v\_url varchar(86);

v\_rown integer;

TYPE cursor\_type IS REF CURSOR;

ansambluri cursor\_type;

one\_record info\_object := info\_object();

all\_records full\_info\_object\_array := new full\_info\_object\_array();

counter integer :=0;

v\_select varchar2(2000);

v\_pagination\_s varchar2(3000);

BEGIN

v\_select := 'SELECT ansamblu.id\_ansamblu, denumirea\_ansamblului, denumirea\_in\_muzeu, denumirea\_la\_origine, pret\_' || p\_type || ' , nume, etnia, datarea, descriere, imprejmuiri, zona\_provenienta, localitate\_provenienta, bibliografie, url\_imagine ' ||

'FROM (((((ansamblu JOIN locatie\_ansamblu ON ansamblu.id\_ansamblu = locatie\_ansamblu.id\_ansamblu) '||

'JOIN locatii ON locatie\_ansamblu.id\_locatie = locatii.id\_locatie) '||

'JOIN informatii ON ansamblu.id\_ansamblu = informatii.id\_ansamblu) '||

'JOIN muzeu ON ansamblu.id\_muzeu = muzeu.id\_muzeu) ' ||

'JOIN preturi ON muzeu.id\_muzeu = preturi.id\_muzeu) WHERE pret\_' || p\_type || ' > ' || p\_min\_price || ' AND pret\_' || p\_type || ' < ' || p\_max\_price;

v\_pagination\_s := 'select \* from (select rownum rown, a.\* from ( ' || v\_select || ' ) a where rownum <= :p\_max ) where rown >= :p\_min ';

DBMS\_OUTPUT.PUT\_LINE(v\_pagination\_s);

OPEN ansambluri for v\_pagination\_s using p\_max\_rownum, p\_min\_rownum;

LOOP

FETCH ansambluri INTO v\_rown, v\_id, v\_denumire\_ansamblu, v\_denumire\_in\_muzeu, v\_denumire\_la\_origine, v\_pret\_bilet, v\_nume\_muzeu, v\_etnia, v\_datarea, v\_descriere, v\_imprejmuiri, v\_zona\_prov, v\_localitate\_prov, v\_bibliografie, v\_url;

EXIT WHEN ansambluri%notfound;

one\_record.denumire\_ansamblu := v\_denumire\_ansamblu;

one\_record.id\_ansamblu := v\_id;

one\_record.denumire\_in\_muzeu := v\_denumire\_in\_muzeu;

one\_record.denumire\_la\_origine := v\_denumire\_la\_origine;

one\_record.pret\_bilet := v\_pret\_bilet;

one\_record.nume\_muzeu := v\_nume\_muzeu;

one\_record.etnia := v\_etnia;

one\_record.datarea := v\_datarea;

one\_record.descriere := v\_descriere;

one\_record.imprejmuiri := v\_imprejmuiri;

one\_record.zona\_prov := v\_zona\_prov;

one\_record.localitate\_prov := v\_localitate\_prov;

one\_record.bibliografie := v\_bibliografie;

one\_record.url := v\_url;

counter := counter + 1;

all\_records.extend;

all\_records(counter) := one\_record;

END LOOP;

CLOSE ansambluri;

return all\_records;

END getAttractionsByPriceRange;

FUNCTION getAttractionsByWordAndPrice (p\_value IN varchar, p\_column IN varchar2, p\_min\_price integer, p\_max\_price integer, p\_price\_category varchar, p\_min integer, p\_max integer)RETURN full\_info\_object\_array AS

v\_id number(4);

v\_denumire\_ansamblu varchar(129);

v\_denumire\_in\_muzeu varchar(129);

v\_denumire\_la\_origine varchar(60);

v\_pret\_bilet number(4,2);

v\_nume\_muzeu varchar2(90);

v\_etnia varchar(24);

v\_datarea varchar(193);

v\_descriere varchar(940);

v\_imprejmuiri varchar(960);

v\_zona\_prov varchar(43);

v\_localitate\_prov varchar(100);

v\_bibliografie varchar(312);

v\_url varchar(86);

v\_rown integer;

v\_select varchar2(1000);

v\_pagination\_s varchar2(2000);

TYPE cursor\_type IS REF CURSOR;

fullinfo cursor\_type;

one\_record info\_object := info\_object();

all\_records full\_info\_object\_array := new full\_info\_object\_array();

counter integer :=0;

BEGIN

v\_select := 'SELECT ansamblu.id\_ansamblu, denumirea\_ansamblului, denumirea\_in\_muzeu, denumirea\_la\_origine, pret\_' || p\_price\_category || ' , nume, etnia, datarea, descriere, imprejmuiri, zona\_provenienta, localitate\_provenienta, bibliografie, url\_imagine ' ||

'FROM (((((ansamblu JOIN locatie\_ansamblu ON ansamblu.id\_ansamblu = locatie\_ansamblu.id\_ansamblu) ' ||

'JOIN locatii ON locatie\_ansamblu.id\_locatie = locatii.id\_locatie) ' ||

'JOIN informatii ON ansamblu.id\_ansamblu = informatii.id\_ansamblu) ' ||

'JOIN muzeu ON ansamblu.id\_muzeu = muzeu.id\_muzeu) ' ||

'JOIN preturi ON preturi.id\_muzeu = muzeu.id\_muzeu) WHERE lower('|| p\_column || ') like lower(''%' || p\_value ||'%'') AND pret\_' || p\_price\_category || ' < ' || p\_max\_price || ' AND pret\_' || p\_price\_category || ' > ' || p\_min\_price;

v\_pagination\_s := 'select \* from (select rownum rown, a.\* from ( ' || v\_select || ' ) a where rownum <= :p\_max ) where rown >= :p\_min ';

OPEN fullinfo FOR v\_pagination\_s USING p\_max, p\_min;

LOOP

FETCH fullinfo INTO v\_rown, v\_id, v\_denumire\_ansamblu, v\_denumire\_in\_muzeu, v\_denumire\_la\_origine, v\_pret\_bilet, v\_nume\_muzeu, v\_etnia, v\_datarea, v\_descriere, v\_imprejmuiri, v\_zona\_prov, v\_localitate\_prov, v\_bibliografie, v\_url;

EXIT WHEN fullinfo%notfound;

IF(v\_bibliografie is null) THEN

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

v\_bibliografie := 'Nu exista blibiografie de afisat';

END IF;

one\_record.denumire\_ansamblu := v\_denumire\_ansamblu;

one\_record.id\_ansamblu := v\_id;

one\_record.denumire\_in\_muzeu := v\_denumire\_in\_muzeu;

one\_record.denumire\_la\_origine := v\_denumire\_la\_origine;

one\_record.pret\_bilet := v\_pret\_bilet;

one\_record.nume\_muzeu := v\_nume\_muzeu;

one\_record.etnia := v\_etnia;

one\_record.datarea := v\_datarea;

one\_record.descriere := v\_descriere;

one\_record.imprejmuiri := v\_imprejmuiri;

one\_record.zona\_prov := v\_zona\_prov;

one\_record.localitate\_prov := v\_localitate\_prov;

one\_record.bibliografie := v\_bibliografie;

one\_record.url := v\_url;

counter := counter + 1;

all\_records.extend;

all\_records(counter) := one\_record;

END LOOP;

CLOSE fullinfo;

return all\_records;

END getAttractionsByWordAndPrice;

END price\_filter;

-Pachetul count\_results

-functia **countResultsByValueAndColumn** returneaza numarul de cautari dupa un cuvant cheie intr- anumita coloana

-functia **countResultsForAllInGallery** returneaza numarul de rezultate returnate de functia getFullInfoByValueAndColumn din pachetul get\_full\_info

-functia **countAttractionsByPriceRange** returneaza numarul de rezultate ale cautarii dupa pret

-functia **countAttractionsByWordAndPrice** returneaza numarul de rezultate ale cautari dupa un cuvant cheie si dupa pret

CREATE OR REPLACE PACKAGE count\_results IS

FUNCTION countResultsByValueAndColumn (p\_value IN varchar, p\_column IN varchar2) RETURN integer;

FUNCTION countResultsForAllInGallery RETURN integer;

FUNCTION countAttractionsByPriceRange (p\_type varchar, p\_min\_price integer, p\_max\_price integer) RETURN integer;

FUNCTION countAttractionsByWordAndPrice (p\_value IN varchar, p\_column IN varchar2, p\_min\_price integer, p\_max\_price integer, p\_price\_category varchar)RETURN integer;

END count\_results;

CREATE OR REPLACE PACKAGE BODY count\_results IS

FUNCTION countResultsByValueAndColumn (p\_value IN varchar, p\_column IN varchar2) RETURN integer AS

v\_count integer;

v\_select varchar2(2000) := 'SELECT count(id\_ansamblu) INTO :v\_count ' ||

'FROM (((((ansamblu JOIN locatie\_ansamblu ON ansamblu.id\_ansamblu = locatie\_ansamblu.id\_ansamblu) ' ||

'JOIN locatii ON locatie\_ansamblu.id\_locatie = locatii.id\_locatie) ' ||

'JOIN informatii ON ansamblu.id\_ansamblu = informatii.id\_ansamblu) ' ||

'JOIN muzeu ON ansamblu.id\_muzeu = muzeu.id\_muzeu) ' ||

'JOIN preturi ON preturi.id\_muzeu = muzeu.id\_muzeu) WHERE lower('|| p\_column || ') like lower(''%' || p\_value ||'%'')';

BEGIN

EXECUTE IMMEDIATE v\_select INTO v\_count;

return v\_count;

END countResultsByValueAndColumn;

FUNCTION countResultsForAllInGallery RETURN integer AS

v\_count integer;

v\_select varchar2(2000) := 'Select count (id\_ansamblu) INTO :v\_count from ansamblu';

BEGIN

EXECUTE IMMEDIATE v\_select INTO v\_count;

return v\_count;

END countResultsForAllInGallery;

FUNCTION countAttractionsByPriceRange (p\_type varchar, p\_min\_price integer, p\_max\_price integer) RETURN integer AS

v\_count integer;

v\_select varchar2(2000) := 'SELECT count(ansamblu.id\_ansamblu) INTO :v\_count '||

'FROM ((ansamblu JOIN muzeu ON ansamblu.id\_muzeu = muzeu.id\_muzeu) ' ||

'JOIN preturi ON muzeu.id\_muzeu = preturi.id\_muzeu) WHERE pret\_' || p\_type || ' > ' || p\_min\_price || ' AND pret\_' || p\_type || ' < ' || p\_max\_price;

BEGIN

EXECUTE IMMEDIATE v\_select INTO v\_count;

return v\_count;

END countAttractionsByPriceRange;

FUNCTION countAttractionsByWordAndPrice (p\_value IN varchar, p\_column IN varchar2, p\_min\_price integer, p\_max\_price integer, p\_price\_category varchar)RETURN integer AS

v\_count integer;

v\_select varchar2(2000) := 'SELECT count(ansamblu.id\_ansamblu) INTO :v\_count ' ||

'FROM (((((ansamblu JOIN locatie\_ansamblu ON ansamblu.id\_ansamblu = locatie\_ansamblu.id\_ansamblu) ' ||

'JOIN locatii ON locatie\_ansamblu.id\_locatie = locatii.id\_locatie) ' ||

'JOIN informatii ON ansamblu.id\_ansamblu = informatii.id\_ansamblu) ' ||

'JOIN muzeu ON ansamblu.id\_muzeu = muzeu.id\_muzeu) ' ||

'JOIN preturi ON preturi.id\_muzeu = muzeu.id\_muzeu) WHERE lower('|| p\_column || ') like lower(''%' || p\_value ||'%'') AND pret\_' || p\_price\_category || ' < ' || p\_max\_price || ' AND pret\_' || p\_price\_category || ' > ' || p\_min\_price;

BEGIN

EXECUTE IMMEDIATE v\_select INTO v\_count;

return v\_count;

END countAttractionsByWordAndPrice;

END count\_results;

declare

v\_a integer;

v\_b integer;

v\_c integer;

v\_d integer;

begin

v\_a := count\_results.countResultsByValueAndColumn('Moara', 'denumirea\_ansamblului');

v\_b := count\_results.countResultsForAllInGallery();

v\_c := count\_results.countAttractionsByPriceRange ('student', 0, 10);

v\_d := count\_results.countAttractionsByWordAndPrice ('moara', 'denumirea\_ansamblului', 0, 20, 'student');

DBMS\_OUTPUT.PUT\_LINE(v\_a);

DBMS\_OUTPUT.PUT\_LINE(v\_b);

DBMS\_OUTPUT.PUT\_LINE(v\_c);

DBMS\_OUTPUT.PUT\_LINE(v\_d);

end;

-Tipurile returnate de functiile de mai sus

CREATE OR REPLACE TYPE info\_object AS OBJECT(

id\_ansamblu number(4),

denumire\_ansamblu varchar(129),

denumire\_in\_muzeu varchar(129),

denumire\_la\_origine varchar(60),

nume\_muzeu varchar2(90),

pret\_bilet number(4,2),

etnia varchar(24),

datarea varchar(193),

descriere varchar(940),

imprejmuiri varchar(960),

zona\_prov varchar(43),

localitate\_prov varchar(100),

bibliografie varchar(312),

url varchar(86),

CONSTRUCTOR FUNCTION info\_object RETURN SELF AS RESULT

);

CREATE OR REPLACE TYPE gallery\_info\_object AS OBJECT(

id\_ansamblu number(4),

denumire\_ansamblu varchar(129),

denumire\_in\_muzeu varchar(129),

denumire\_la\_origine varchar(60),

url varchar(86),

CONSTRUCTOR FUNCTION gallery\_info\_object RETURN SELF AS RESULT

);

CREATE OR REPLACE TYPE BODY gallery\_info\_object

AS

CONSTRUCTOR FUNCTION gallery\_info\_object RETURN SELF AS RESULT

AS

BEGIN

RETURN;

END;

END;

CREATE OR REPLACE TYPE BODY info\_object

AS

CONSTRUCTOR FUNCTION info\_object RETURN SELF AS RESULT

AS

BEGIN

RETURN;

END;

END;

CREATE OR REPLACE type full\_info\_object\_array is table of info\_object;

CREATE OR REPLACE type gallery\_info\_object\_array is table of gallery\_info\_object;

**Triggere**

-trigger-ul **StergeMuzeu** sterge un muzeu si toate ansamblurile si preturile asignate acestuia

CREATE OR REPLACE TRIGGER StergeMuzeu

BEFORE DELETE ON muzeu

FOR EACH ROW

BEGIN

DELETE FROM ansamblu where id\_muzeu = :Old.id\_muzeu;

DELETE FROM preturi where id\_muzeu = :Old.id\_muzeu;

END;

-trigger-ul **StergeAnsamblu** sterge un ansamblu si informatiile si id-ul sau din tabela locatie\_ansamblu (tabela locatii nu este afectata deoarece o locatie poate fi folosita de mai multe ansambluri)

CREATE OR REPLACE TRIGGER StergeAnsamblu

BEFORE DELETE ON ansamblu

FOR EACH ROW

BEGIN

DELETE FROM informatii where id\_ansamblu = :Old.id\_ansamblu;

DELETE FROM locatie\_ansamblu where id\_ansamblu = :Old.id\_ansamblu;

END;

**Indecsi**

-Am creat indecsii urmatori pentru a oferi acces accelerat la inregistrarile din tabele pe baza coloanelor :

-denumirea\_ansamblului , datarea , etnia din tabela ansamblu;

-nume din tabela muzeu;

-localitate\_provenienta si zona\_provenienta din tabela locatii;

-pret\_student, pret\_elev si pret\_angajat din tabela preturi

CREATE INDEX indx\_nume\_ansamblu ON ansamblu(denumirea\_ansamblului);

CREATE INDEX indx\_datare ON ansamblu(datarea);

CREATE INDEX indx\_etinie ON ansamblu(etnia);

CREATE INDEX unique\_nume\_muzeu ON muzeu(nume);

CREATE INDEX indx\_localitate ON locatii(localitate\_provenienta);

CREATE INDEX indx\_zona ON locatii(zona\_provenienta);

CREATE INDEX indx\_pret\_student ON preturi(pret\_student);

CREATE INDEX indx\_pret\_angajat ON preturi(pret\_angajat);

CREATE INDEX indx\_pret\_elev ON preturi(pret\_elev);

Exemple de select-uri care folosesc indecsii de mai sus

SELECT BY WORD AND PRICE

SELECT ansamblu.id\_ansamblu, denumirea\_ansamblului, denumirea\_in\_muzeu, denumirea\_la\_origine, pret\_student , nume, etnia, datarea, descriere, imprejmuiri, zona\_provenienta, localitate\_provenienta, bibliografie

FROM (((((ansamblu JOIN locatie\_ansamblu ON ansamblu.id\_ansamblu = locatie\_ansamblu.id\_ansamblu)

JOIN locatii ON locatie\_ansamblu.id\_locatie = locatii.id\_locatie)

JOIN informatii ON ansamblu.id\_ansamblu = informatii.id\_ansamblu)

JOIN muzeu ON ansamblu.id\_muzeu = muzeu.id\_muzeu)

JOIN preturi ON preturi.id\_muzeu = muzeu.id\_muzeu) WHERE lower(etnia) like lower('%romani%') AND pret\_student < 10 AND pret\_student > 1;

SLEECT BY KEYWORD

SELECT ansamblu.id\_ansamblu, denumirea\_ansamblului, denumirea\_in\_muzeu, denumirea\_la\_origine, pret\_angajat, nume, etnia, datarea, descriere, imprejmuiri, zona\_provenienta, localitate\_provenienta, bibliografie

FROM (((((ansamblu JOIN locatie\_ansamblu ON ansamblu.id\_ansamblu = locatie\_ansamblu.id\_ansamblu)

JOIN locatii ON locatie\_ansamblu.id\_locatie = locatii.id\_locatie)

JOIN informatii ON ansamblu.id\_ansamblu = informatii.id\_ansamblu)

JOIN muzeu ON ansamblu.id\_muzeu = muzeu.id\_muzeu)

JOIN preturi ON preturi.id\_muzeu = muzeu.id\_muzeu) WHERE denumirea\_ansamblului like '%Moara%' ORDER BY denumirea\_ansamblului;

SELECT DETAILS BY ID

SELECT DISTINCT ansamblu.id\_ansamblu, denumirea\_ansamblului, denumirea\_in\_muzeu, denumirea\_la\_origine, pret\_angajat, nume, etnia, datarea, descriere, imprejmuiri, zona\_provenienta, localitate\_provenienta, bibliografie

FROM (((((ansamblu JOIN locatie\_ansamblu ON ansamblu.id\_ansamblu = locatie\_ansamblu.id\_ansamblu)

JOIN locatii ON locatie\_ansamblu.id\_locatie = locatii.id\_locatie)

JOIN informatii ON ansamblu.id\_ansamblu = informatii.id\_ansamblu)

JOIN muzeu ON ansamblu.id\_muzeu = muzeu.id\_muzeu)

JOIN preturi ON muzeu.id\_muzeu = preturi.id\_muzeu) WHERE ansamblu.id\_ansamblu = '1077';