

# EtnoWeb

Arhitectura bazei de date

Dinu Georgiana  
Laslau Laurentiu  
Rapa Mihaela  
Toma Lucian

# Cuprins

<b>Descriere tabele si constrangeri.....</b>	<b>3</b>
<b>Pachete</b>	
• Pachetul full_info.....	4
• Pachetul p_gallery_info.....	8
• Pachetul price_filter.....	9
• Pachetul count_results.....	13
• Tipurile returnate de functii.....	15
<b>Triggere.....</b>	<b>16</b>
<b>Indecsi.....</b>	<b>17</b>

# 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 imprejuririle.

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 constituite 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_num_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 cuvânt cheie după care userul dorește să se efectueze căutarea, coloana în care se caută cuvântul introdus , și min\_rownum și max\_rownum pe care îi folosim la paginare și afișează un anumit număr de rezultate în funcție de intervalul dat
- functia **getGalleryInfoByOffset** este folosită pentru a returna un număr exact de informații pentru galerie
- functia **getDetailsById** primește un id și returnează 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_numu_muzeu varchar2(90);
        v_etnia varchar(24);
        v_datarea varchar(193);
        v_descriere varchar(940);
        v_imprejmuiuri 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, imprejmui, 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_imprejmui, 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 bibliografie 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.imprejmui := v_imprejmui;
        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_num_ze_muzeu varchar2(90);
v_etnia varchar(24);
v_datarea varchar(193);
v_descriere varchar(940);
v_imprejmuiiri 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, imprejmuiiri, 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_num_ze_muzeu, v_etnia, v_datarea, v_descriere, v_imprejmuiiri, 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 bibliografie 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.numa_muzeu := v_numa_muzeu;
one_record.etnia := v_etnia;
one_record.datarea := v_datarea;
one_record.descriere := v_descriere;
one_record.imprejmuiuri := v_imprejmuiuri;
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 cuvint 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_imprejmuiuri 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, imprejmuiuri, 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_imprejmuiuri, 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.numa_muzeu := v_numa_muzeu;
one_record.etnia := v_etnia;
one_record.datarea := v_datarea;
one_record.descriere := v_descriere;
one_record.imprejmuiuri := v_imprejmuiuri;
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_numa_muzeu varchar2(90);
v_etnia varchar(24);
v_datarea varchar(193);
v_descriere varchar(940);
v_imprejmuiuri 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, imprejmui, 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_imprejmui, 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 bibliografie 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.imprejmui := v_imprejmui;
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),  
    imprejmuiuri 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 urmatoari pentru a oferi acces accelerat la inregistrările 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_num_ansamblu ON ansamblu(denumirea_ansamblului);
CREATE INDEX indx_datare ON ansamblu(datarea);
CREATE INDEX indx_etnie ON ansamblu(etnia);
CREATE INDEX unique_num_muzeu ON muzeu(num);
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, imprejmui,
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, imprejmui,
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, imprejmui,
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';

```