
PROJECT

CUPRINS:

- **1.Tema proiectului, schema tabelor**
 - **2.Structuri Fundamentale de Programare**
 - **3.Tratarea Exceptiilor**
 - **4. Functii si proceduri**
 - **5.Pachete**
 - **6. Declansatori**
 - **7. Script**
-

1.TEMA PROIECTULUI:

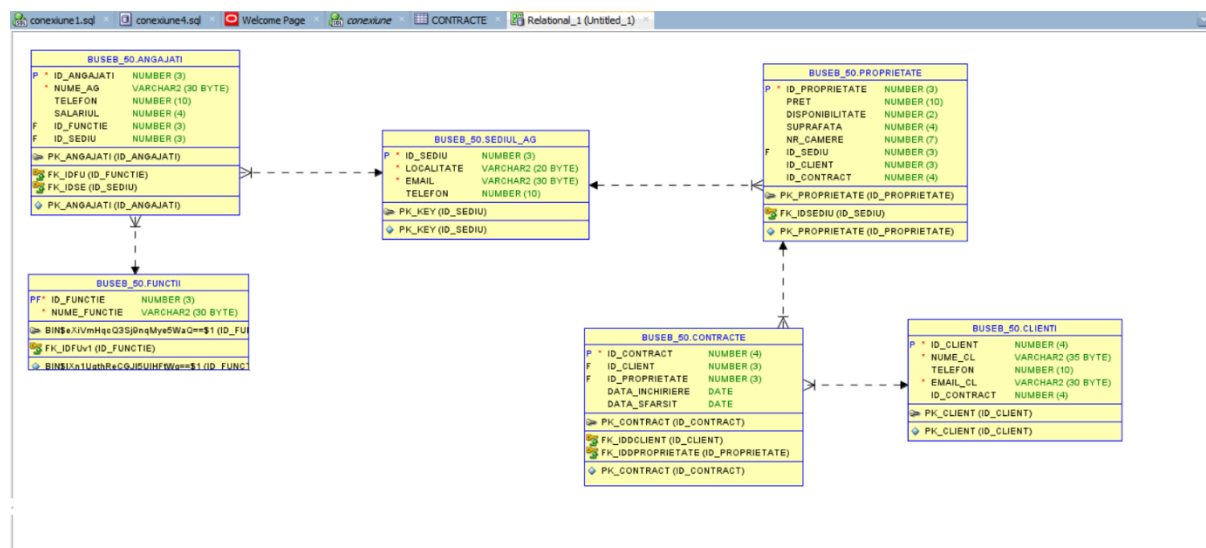
Proiectul prezinta realizarea unei baze de date a unei **Agentii imobiliare**, ce poate avea o utilitate importanta in viata reala.

Datele sunt introduse in sase tabele, intre care sunt realizate legaturi astfel incat sa se evite redundanta.

Baza de date a acestei agentii imobiliare prezinta informatii despre angajati, proprietati si clienti. Agentia are cateva sedii in diferite puncte ale tarii, astfel incat sa se poata ocupa de un numar cat mai mare de clienti si proprietati. Legatura dintre client si proprietate se realizeaza prin tabela de contracte, ce cuprinde ,pe langa id_contract, urmatoarele: id-ul clientului, id-ul proprietatii si data de inchiriere, respectiv data de sfarsit a contractului.

Fiecare tabela are o cheie primara, prin intermediul careia se realizeaza legatura dintre tabele: tabela functii(id_functie -> primary key), angajati(id_angajati PK), sediul_ag(id_sediu PK), proprietate(id_proprietate PK), contracte(id_contract PK), tabela clienti(id_client PK).

Schema tabelelor si a legaturilor dintre acestea:



2. STRUCTURI FUNDAMENTALE DE PROGRAMARE:

●LOOP:

Avem nevoie sa afisam numele si salariul angajatilor care au id-ul >= decat un id introdus de la tastatura:

```
DECLARE
v_id angajati.id_angajati%type:=15;
v_nume angajati.nume_ag%type;
v_salariul angajati.salariul%type;
begin
loop
select nume_ag, salariul into v_nume, v_salariul from angajati
where id_angajati=v_id;
DBMS_OUTPUT.PUT_LINE('Angajatul ' ||v_nume||' cu id-ul: '|| v_id || ' are salariul '|| v_salariul);
v_id:=v_id+1;
exit when v_id>20;
end loop;
end;
```

Script Output x

Task completed in 0.051 seconds

Angajatul Ionescu Alina cu id-ul: 15 are salariul 5500
Angajatul Nica Andreea cu id-ul: 16 are salariul 5500
Angajatul Paun Maria cu id-ul: 17 are salariul 5000
Angajatul Petre Anca cu id-ul: 18 are salariul 5500
Angajatul Marin Sarah cu id-ul: 19 are salariul 6500
Angajatul Popescu Alex cu id-ul: 20 are salariul 5500

PL/SQL procedure successfully completed.

●IF ELSE:

Sa se introduca de la tastatura id-ul unui angajat. Sa se verifice daca salariul angajatului este cuprins intre 4000 si 5000 ron:

```
set serveroutput on
DECLARE
v_id angajati.id_angajati%type:=15;
v_nume angajati.nume_ag%type;
v_salariul angajati.salariul%type;
begin
select nume_ag, salariul into v_nume, v_salariul from angajati
where id_angajati=v_id;
if v_salariul between 4000 and 5000 then
dbms_output.put_line('Angajatul ' ||v_nume||' are salariul cuprins intre 4000 si 5000');
else
dbms_output.put_line('Angajatul ' || v_nume || ' nu are salariul cuprins in intervalul dat');
end if;
end;
```

Script Output x

Task completed in 2.404 seconds

Angajatul Ionescu Alina nu are salariul cuprins in intervalul dat!

PL/SQL procedure successfully completed.

●CASE:

Avem nevoie sa afisam pretul unei proprietati al carei id este data de la tastatura.Dupa afisare, sa se mareasca pretul proprietatii in urmatoarele moduri:

- I. Cu 20% in cazul in care suprafata acesteia este mai mica decat 50mp;
- II. Cu 30% in cazul in care suprafata acesteia este cuprinsa intre 50 si 75 mp;
- III. Cu 10% daca suprafata este mai mare de 75mp;

Sa se afiseze pretul final.

```
declare
v_id proprietate.id_proprietate%type:=10;
v_pret proprietate.pret%type;
v_suprafata proprietate.suprafata%type;
BEGIN
SELECT pret, suprafata into v_pret, v_suprafata from proprietate where id_proprietate=v_id;
dbms_output.put_line ('Pretul initial al proprietatii cu id-ul '|| v_id ||' este: '||v_pret);
v_pret:= CASE WHEN v_suprafata < 50 THEN 1.2* v_pret
WHEN v_suprafata between 50 and 75 THEN 1.3 * v_pret
ELSE 1.1* v_pret END;

dbms_output.put_line('Pretul final este: '||v_pret);
end;
```

Script Output x

Task completed in 4.387 seconds

dbms_output.put_line('Pretul final este: '||v_pret);
end;
Pretul initial al proprietatii cu id-ul 201 este: 400
Pretul final este: 520

PL/SQL procedure successfully completed.

- **FOR:**

Avem nevoie sa afisam id-ul si pretul proprietatilor al caror id este cuprins intre 200 si 205.

Pentru proprietatile cuprinse in intervalul de mai sus, sa se calculeze pretul mediu:

```

DECLARE
type prop is table of proprietate%rowtype index by pls_integer;
v_desc prop;
v_pret proprietate.pret%type:=0;
BEGIN
for i in 200..205 loop
SELECT * into v_desc(i) from proprietate where id_proprietate=i;
end loop;
for i in v_desc.first..v_desc.last loop
dbms_output.put_line('Proprietatea: '||i||' are pretul de: ' ||v_desc(i).pret);
end loop;
SELECT avg(pret) into v_pret from proprietate where id_proprietate between 200 and 205;
dbms_output.put_line('Pretul mediu al celor ' || v_desc.count() ' proprietati din tabloul indexat, este: '|| v_pret );
END;
/

```

Script Output: Task completed in 0.057 seconds

```

Proprietatea: 200 are pretul de: 360
Proprietatea: 201 are pretul de: 400
Proprietatea: 202 are pretul de: 450
Proprietatea: 203 are pretul de: 384
Proprietatea: 204 are pretul de: 500
Proprietatea: 205 are pretul de: 360
Pretul mediu al celor 6 proprietati din tabloul indexat, este: 409

```

PL/SQL procedure successfully completed.

- **WHILE:**

Sa se afiseze salariul mediu al angajatilor.

Sa se afiseze toti angajatii care au id-ul >=10, dar <15, care au salariul mai mare decat salariul mediu:

```

declare
v_sal angajati.salariu%type;
v_sm v_sal%type;
i number(2):=10;
begin
select avg(salariu) into v_sm from angajati;
dbms_output.put_line('Salariul mediu este '||v_sm);
while i<15 loop
select salariu into v_sal from angajati where id_angajati=i;
if v_sal>v_sm then
dbms_output.put_line('Angajatul cu id '|| i||' are salariul ' || v_sal);
end if;
i:=i+1;
end loop;
end;
/

```

Script Output: Task completed in 0.049 seconds

PL/SQL procedure successfully completed.

```

Salariul mediu este 5633
Angajatul cu id 10 are salariul 9500
Angajatul cu id 11 are salariul 6000

```

PL/SQL procedure successfully completed.

- **FOR+CORSOR:**

Pentru compania noastra, trebuie sa afisam numarul de angajati ce ocupa fiecare functie, in cazul in care functia respectiv e ocupata de mai mult de o persoana. Mai apoi, sa se afiseze rezultatul, in ordine descrescatoare, in functie de numarul de angajati.

```

declare
cursor f is select
f.ume_functie, count(a.id_functie) nr_
from functii f, angajati a
where f.id_functie=a.id_functie and f.ume_functie not in ('CEO')
group by f.ume_functie
having count(a.id_functie)>1
order by nr desc;
begin
for r in f
loop
dbms_output.put_line('Postul de '||r.ume_functie||' are '||r.nr||' angajati!');
end loop;
end;
/

```

Script Output: Task completed in 0.05 seconds

PL/SQL procedure successfully completed.

```

Postul de Agent imobiliar are 6 angajati!
Postul de HR are 3 angajati!
Postul de Programator are 2 angajati!
Postul de Contabil are 2 angajati!

```

PL/SQL procedure successfully completed.

●CURSOR:

Sa se scrie codul cu ajutorulul caruia putem vedea id-ul si salariul primilor 4 angajati cu salariul cel mai mare, din intreaga companie.

```
declare
v_id angajati.id_angajati%type;
v_sal angajati.salariu%type;
cursor c is select id_angajati, salariu
from angajati
order by salariu desc;
begin
open c;
loop
fetch c into v_id, v_sal;
exit when c%rowcount < 4 or c%notfound;
dbms_output.put_line('Angajatul cu id '||v_id||' are sal '||v_sal);
end loop;
close c;
end;
```

Script Output x

Task completed in 0.067 seconds

*Cause: Usually a PL/SQL compilation error.

*Action:

Ang cu id 10 are sal 9500
Ang cu id 19 are sal 6500
Ang cu id 11 are sal 6000
Ang cu id 13 are sal 5500

PL/SQL procedure successfully completed.

●CURSOR CU PARAMETRII:

Se doreste sa se stabileasca o lista cu sediile, respectiv orasele, din care agentia imobiliara obtine un venit de peste 450euro.

```
declare
cursor c_sed(v_venit number) is
select s.id_sediu, s.oras, sum(p.pret) venit
from proprietate p, sediu_ag s
where p.id_sediu=s.id_sediu and p.id_client is not null
group by s.id_sediu, s.oras
having sum(p.pret) > v_venit
order by venit;
v_venit number;
r_sediu c_sed%rowtype;
begin v_venit:=450;
dbms_output.put_line('Lista sediilor in care valoare proprietatilor inchiriate depaseste
if not c_sed%isopen then
open c_sed(v_venit);
end if;
loop fetch c_sed into r_sediu;
exit when c_sed%notfound;
dbms_output.put_line(r_sediu.id_sediu || ' ' || r_sediu.oras || ' ' || r_sediu.venit);
end loop;
end;
```

Script Output x

Task completed in 0.054 seconds

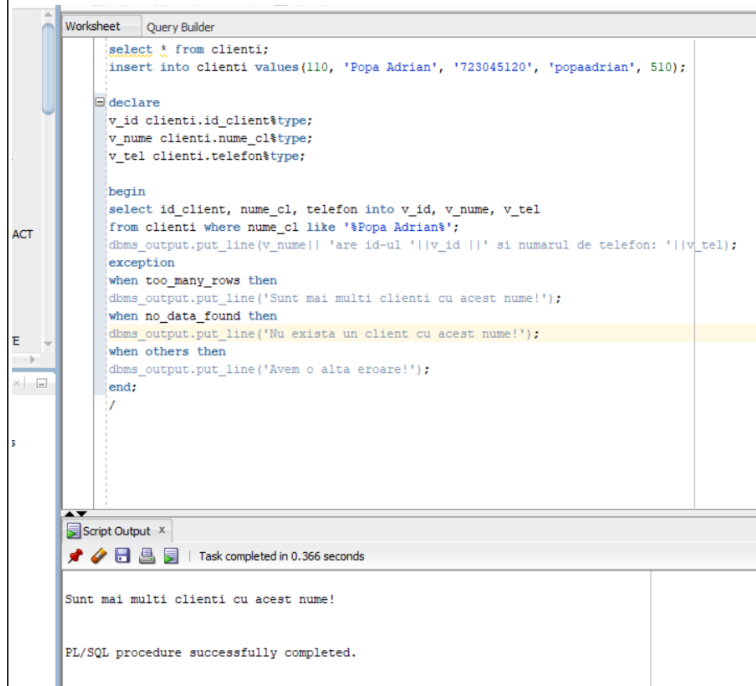
Lista sediilor in care valoare proprietatilor inchiriate depaseste 450
100 Bucuresti 464
102 Brasov 1695

PL/SQL procedure successfully completed.

3. TRATAREA EXCEPTIILOR:

●TOO MANY ROWS:

Avem nevoie sa adaugam un nou client. Din pacate, numele acestui client se regaseste deja in tabela . Sa se creeze un bloc PL/SQL care afiseaza id-ul si numarul de telefon al clientului cu numele adaugat anterior. Sa se trateze exceptia `too_many_rows` prin afisarea unui mesaj. In cazul in care nu exista nici macar un client care sa aiba numele respectiv, sa se afiseze alt mesaj:



```
Worksheet  Query Builder
select * from clienti;
insert into clienti values(110, 'Popa Adrian', '723045120', 'popaadrian', 510);

declare
v_id clienti.id_client%type;
v_nume clienti.nume_cl%type;
v_tel clienti.telefon%type;

begin
select id_client, nume_cl, telefon into v_id, v_nume, v_tel
from clienti where nume_cl like '%Popa Adrian%';
dbms_output.put_line(v_nume|| 'are id-ul '||v_id||' si numarul de telefon: '||v_tel);
exception
when too_many_rows then
dbms_output.put_line('Sunt mai multi clienti cu acest nume!');
when no_data_found then
dbms_output.put_line('Nu exista un client cu acest nume!');
when others then
dbms_output.put_line('Avem o alta eroare!');
end;
/
```

Script Output x | Task completed in 0.366 seconds

Sunt mai multi clienti cu acest nume!

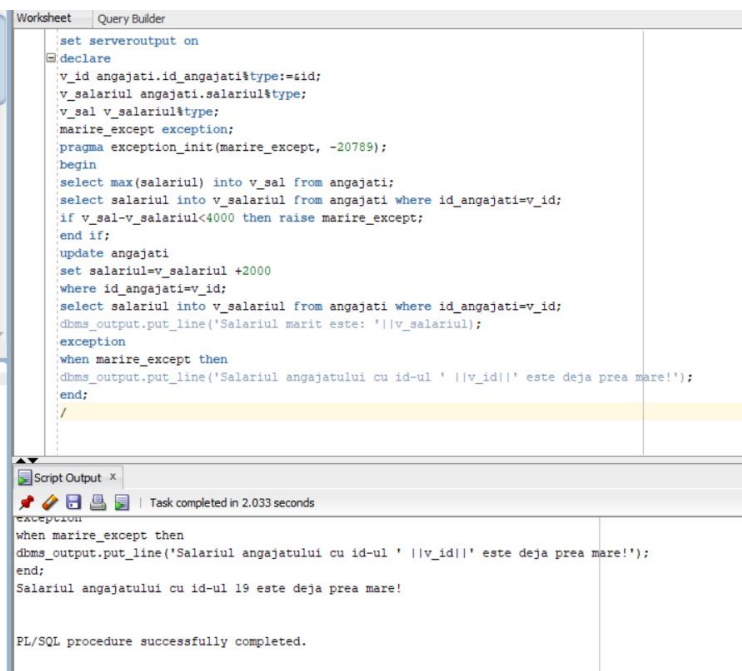
PL/SQL procedure successfully completed.

●Exceptie definita de utilizator:

Vrem sa modificam salariul unui angajat al carui id este introdus de la tastatura.

Sa se afle salariul maxim din tabela ANGAJATI. Daca salariul maxim – salariul actual al angajatului <4000, atunci sa se ridce o exceptie. In caz contrar, se maresta salariul actual cu 2000 ron.

In primul ss am prezentat exceptia "marire_except". In al doilea ss, am prezentat modificarea salariului angajatului cu id-ul 24.



```
Worksheet  Query Builder
set serveroutput on
declare
v_id angajati.id_angajati%type:=&id;
v_salariul angajati.salariul%type;
v_sal v_salariul%type;
marire_except exception;
pragma exception_init(marire_except, -20789);
begin
select max(salariul) into v_sal from angajati;
select salariul into v_salariul from angajati where id_angajati=v_id;
if v_sal-v_salariul<4000 then raise marire_except;
end if;
update angajati
set salariul=v_salariul +2000
where id_angajati=v_id;
select salariul into v_salariul from angajati where id_angajati=v_id;
dbms_output.put_line('Salariul marit este: '||v_salariul);
exception
when marire_except then
dbms_output.put_line('Salariul angajatului cu id-ul '||v_id||' este deja prea mare!');
end;
/
```

Script Output x | Task completed in 2.033 seconds

exception

when marire_except then

dbms_output.put_line('Salariul angajatului cu id-ul '||v_id||' este deja prea mare!');

end;

Salariul angajatului cu id-ul 19 este deja prea mare!

PL/SQL procedure successfully completed.

```

new:declare
v_id angajati.id_angajati%type:=24;
v_salariul angajati.salariul%type;
v_sal v_salariul%type;
marire_except exception;
pragma exception_init(marire_except, -20789);

select salariul into v_salariul from angajati where id_angajati=
dbms_output.put_line('Salariul marit este: '||v_salariul);
exception
when marire_except then
dbms_output.put_line('Salariul angajatului cu id-ul ' ||v_id||'
end;
Salariul marit este: 6500

PL/SQL procedure successfully completed.

```

1.2

●Excepție 'Cursor invalid':

Sa se utilizeze un cursor pentru a extrage din tabela CONTRACTE data de incheiere a contractelor. Sa se afiseze o exceptie pentru cazul in care cursorul este invalid.

```

declare
cursor c is select * from contracte
where extract (month from data_sfarsit) in (09, 10, 07);
r c%rowtype;
begin
open c;
loop
fetch c into r;
exit when c%notfound;
dbms_output.put_line(r.data_sfarsit);
close c;
end loop;
exception
when invalid_cursor then
dbms_output.put_line('Cursorul este inchis!');
end;
/

```

Script Output x

Task completed in 0.062 seconds

PL/SQL procedure successfully completed.

14-SEP-23

Cursorul este inchis!

PL/SQL procedure successfully completed.

●Excepție pentru parola corecta/incorecta:

Vrem sa afisam contractul unui client al carui id este dat de la tastatura. Id-ul contractului este accesibil doar dupa ce a fost introdusa, in mod corect, parola : "client". In cazul in care parola nu este corecta, sa se genereze o exceptie.

In primul ss, avem cazul "parola incorecta".

In al doilea ss, avem cazul "parola corecta"

```

declare v_parola varchar2(20);
v_id clienti.id_client%type;
v_contract clienti.id_contract%type;
begin
v_id:=&v_id;
v_parola:=&v_parola;
if v_parola like 'client' then
select id_contract into v_contract from clienti where id_client=v_id;
dbms_output.put_line('Clientului cu id-ul '|| v_id||' ii corespunde contractul cu nr-ul: '||v_contract);
else
raise_application_error(-20989, 'Nu aveti acces!');
end if;
end;
/

```

Script Output x

Task completed in 7.67 seconds

raise_application_error(-20989, 'Nu aveti acces!');

end if;

end;

Error report -

ORA-20989: Nu aveti acces!

ORA-06512: at line 11

1.1


```
Script Output x
Task completed in 3.974 seconds

end if;
end;
Clientului cu id-ul 100 ii corespunde contractul cu nr-ul: 500

PL/SQL procedure successfully completed.
```

1.2

4. FUNCTII SI PROCEDURI:

●Funcție 1:

Vrem sa aplicam un discount, asupra unei proprietati ce are contractul incheiat intr-o luna de toamna (septembrie, octombrie, noiembrie). Daca, data de incheiere a contractului nu este intr-una din lunile enumerate, se v-a aplica un discount mai mic. Sa se afiseze cat este noul pret pentru proprietatea aleasa.

```
Worksheet Query Builder
0.078 seconds

create or replace function f_discount
(f_id number, f_dis1 number, f_dis2 number)
return number
is
v_data$ contracte.data_sfarsit$type;
v_pret proprietate.pret$type;
begin
select c.data_sfarsit, p.pret into v_data$, v_pret from contracte c, proprietate p where c.id_contract=p.id_contract and c.id_contract=f_id;
if extract(month from v_data$) in (09,10,11) then
return v_pret*(1-f_dis1/100);
else
return v_pret*(1-f_dis2/100);
end if;
end;
/

select c.data_sfarsit, p.pret from contracte c, proprietate p
where c.id_contract=p.id_contract and c.id_contract=102;
begin
dms_output.put_line('Pretul cu discount pentru proprietate este '||f_discount(102, 15, 5));
end;
```

```
Script Output x
Task completed in 0.078 seconds

Function F_DISCOUNT compiled

DATA_SFAR      PRET
-----
14-SEP-23      360

Pretul cu discount pentru proprietate este 306

Messages - Log
```

•Functie 2:

Pentru un angajat ales utilizator dorim marirea salariului.

Daca angajatul respectiv are salariul mai mic decat 5000, salariul acestuia sa fie marit cu 500lei. In cazul in care salariul este cuprins intre 5000-5500, salariul sa fie marit doar cu 250.

Am atasat si ss cu datele din tabela angajati, dupa executarea functiei, pentru a vedea modificarea realizata.

```

create or replace function promovare_ang(p_id angajati.id_angajati%type)
return number is
v_sal angajati.salariu%type;
ang_except Exception;
begin select salariu into v_sal from angajati where id_angajati=p_id;
if v_sal<5000 then
return v_sal+500;
elsif v_sal between 5000 and 5500 then
return v_sal+250;
else
RETURN v_sal;
end if;
exception
when no_data_found then
return 0;
end;
/
declare
v_id angajati.id_angajati%type :=:id;
v_marire number;
begin
v_marire:=promovare_ang(v_id);
if v_id is not null then
update angajati set salariu=v_marire where id_angajati=v_id;
else
dbms_output.put_line('nu exista angajatul cautat!');
end if;
end;

```

Script Output x

Task completed in 11.462 seconds

SQL procedure successfully completed.

7	16 Nica Andreea	741845612	5500	4	100
8	17 Paun Maria	768541236	5000	4	101
9	18 Petre Anca	745123896	5500	4	102
10	19 Marin Sarah	743278945	6500	2	104
11	20 Popescu Alex	754398653	5500	4	105
12	21 Diaconu Marius	745632108	5000	6	104
13	22 Ion Vladimir	745635108	5500	5	105
14	23 Popescu Sorin	754398753	5500	4	104
15	24 Horia Marian	743525999	5000	6	100

_sal;

1

data_found then

:

angajati.id_angajati%type :=:id;

number;

=promovare_ang(v_id);

is not null then

angajati set salariu=v_marire where id_angajati=

ut.put_line('nu exista angajatul cautat!');

Enter Substitution Variable

Enter value for id:

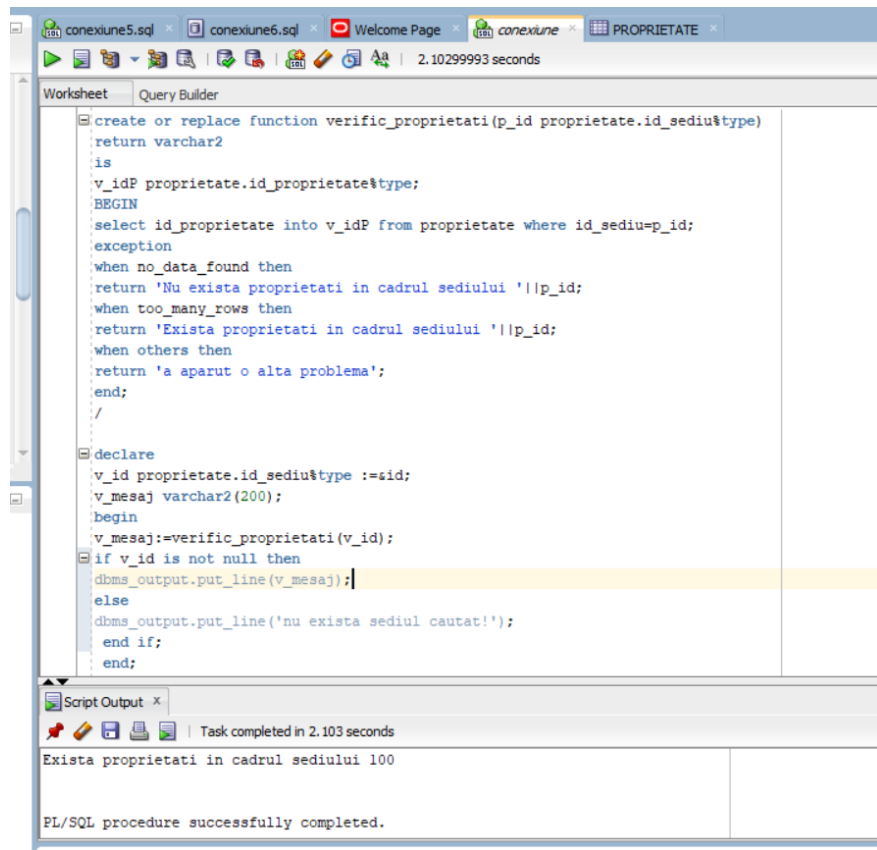
17

OK Cancel

7	16 Nica Andreea	741845612	5500	4	100
8	17 Paun Maria	768541236	5250	4	101
9	18 Petre Anca	745123896	5500	4	102

●Functie 3:

Agentia are cate un sediu in diferite puncte ale tarii. Verifica daca exista sau nu proprietati in cadrul unui anumit sediu. Functia are ca parametru, introdus de la tastatura, id-ul sediului. Pentru fiecare rezultat: exista/nu exista, sa se afiseze un mesaj specific.



```
create or replace function verific_proprietati(p_id proprietate.id_sediu%type)
return varchar2
is
v_idP proprietate.id_proprietate%type;
BEGIN
select id_proprietate into v_idP from proprietate where id_sediu=p_id;
exception
when no_data_found then
return 'Nu exista proprietati in cadrul sediului '||p_id;
when too_many_rows then
return 'Exista proprietati in cadrul sediului '||p_id;
when others then
return 'a aparut o alta problema';
end;
/

declare
v_id proprietate.id_sediu%type :=:id;
v_mesaj varchar2(200);
begin
v_mesaj:=verific_proprietati(v_id);
if v_id is not null then
dbms_output.put_line(v_mesaj);
else
dbms_output.put_line('nu exista sediul cautat!');
end if;
end;
```

Script Output x

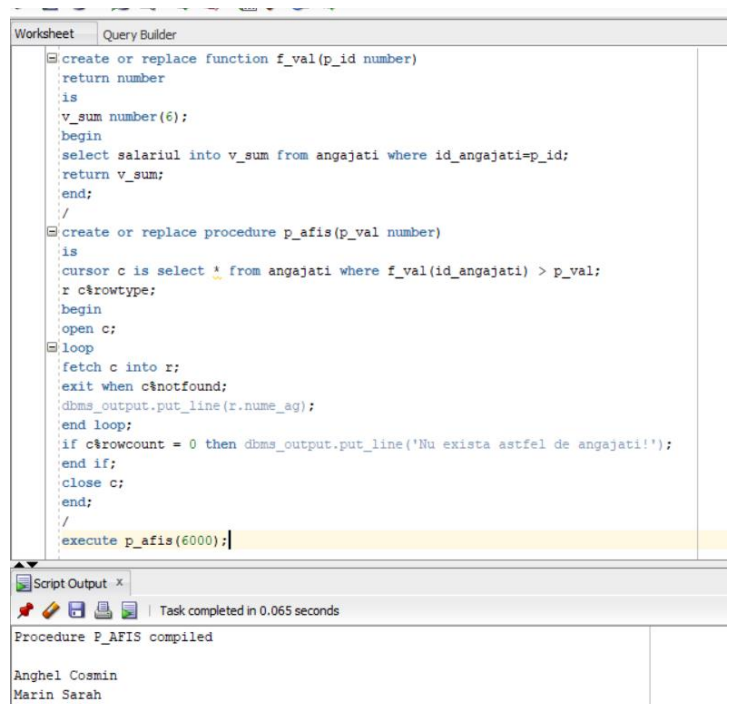
Task completed in 2.103 seconds

Exista proprietati in cadrul sediului 100

PL/SQL procedure successfully completed.

●Procedura 1:

Sa se realizeze o procedura prin intermediul careia se face apelul unei functii. Prin intermediul procedurii, sa se afiseze numele angajatilor care au salariul mai mare decat 6000.



```
create or replace function f_val(p_id number)
return number
is
v_sum number(6);
begin
select salariul into v_sum from angajati where id_angajati=p_id;
return v_sum;
end;
/

create or replace procedure p_afis(p_val number)
is
cursor c is select * from angajati where f_val(id_angajati) > p_val;
r c%rowtype;
begin
open c;
loop
fetch c into r;
exit when c%notfound;
dbms_output.put_line(r.nume_ag);
end loop;
if c%rowcount = 0 then dbms_output.put_line('Nu exista astfel de angajati!');
end if;
close c;
end;
/

execute p_afis(6000);
```

Script Output x

Task completed in 0.065 seconds

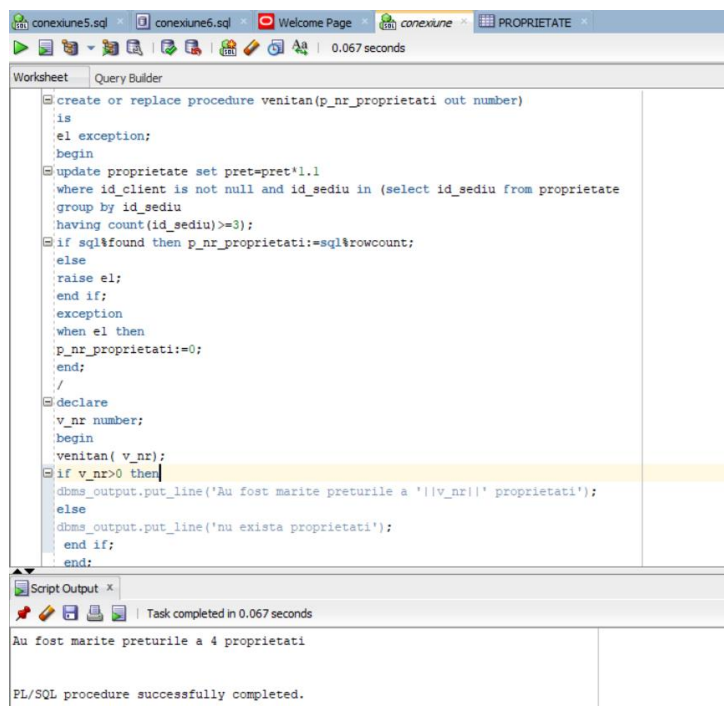
Procedure P_AFIS compiled

Anghel Cosmin

Marin Sarah

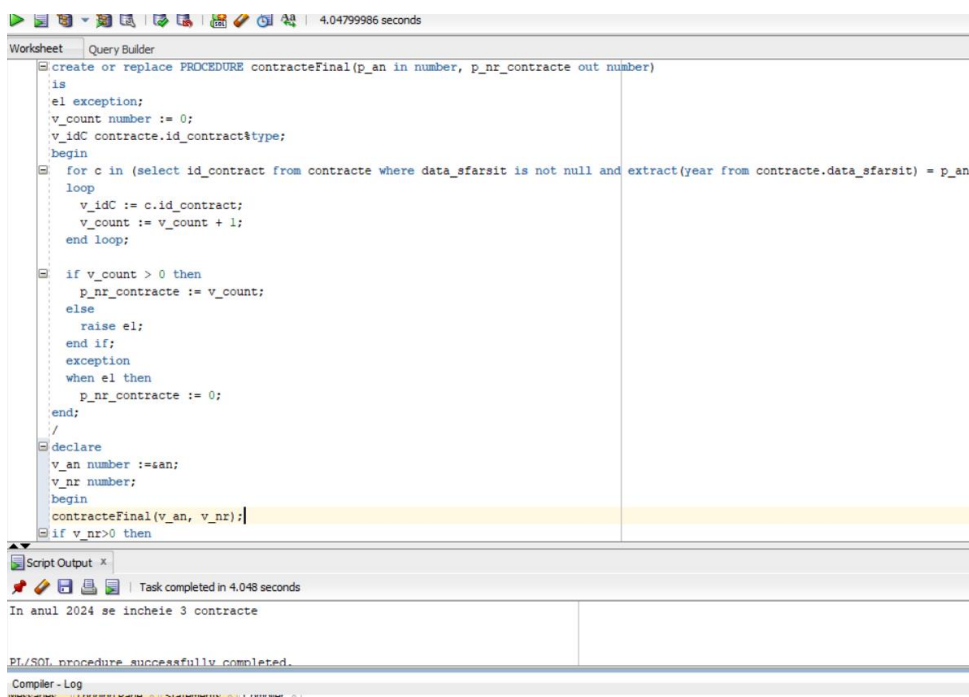
●Procedura 2:

Se doreste marirea pretului asociat proprietatilor care fac parte din sedii cu 3 sau mai multe proprietati si al caror id_client este diferit de null. Sa se returneze si numarul de proprietati pentru care a fost realizata modificarea.



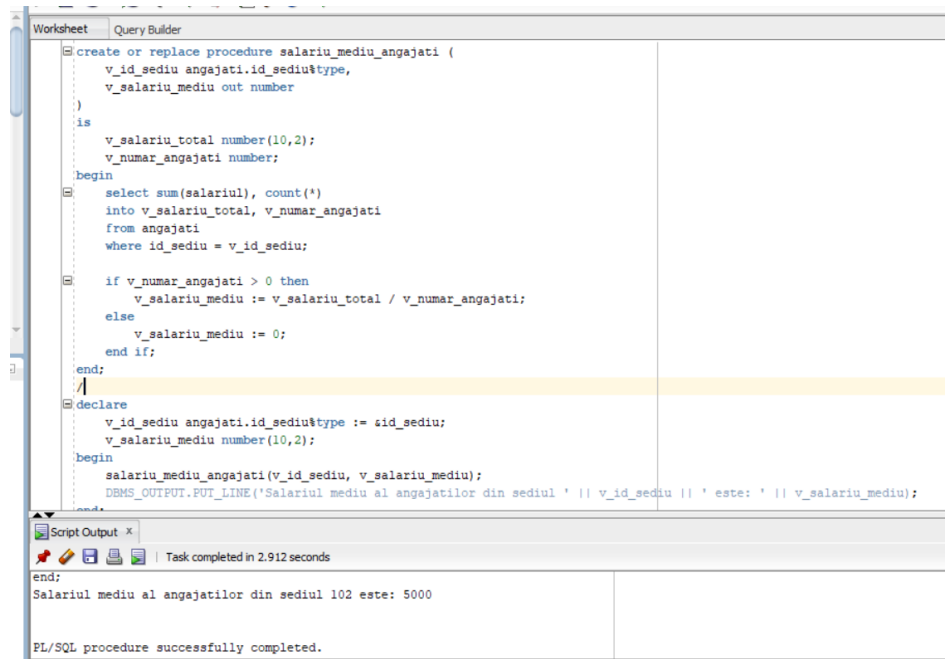
●Procedura 3:

Sa se scrie codul prin intermediul caruia putem afla cate proprietati sunt eliberate/disponibile incepand cu un anumit an ales de utilizator.



●Procedura 4:

Pentru un anumit sediu, sa se determine salariul mediu al angajatilor ce lucreaza in sediul respectiv.



```
create or replace procedure salariu_mediu_angajati (
    v_id_sediu angajati.id_sediu%type,
    v_salariu_mediu out number
)
is
    v_salariu_total number(10,2);
    v_numar_angajati number;
begin
    select sum(salariul), count(*)
    into v_salariu_total, v_numar_angajati
    from angajati
    where id_sediu = v_id_sediu;

    if v_numar_angajati > 0 then
        v_salariu_mediu := v_salariu_total / v_numar_angajati;
    else
        v_salariu_mediu := 0;
    end if;
end;

declare
    v_id_sediu angajati.id_sediu%type := 102;
    v_salariu_mediu number(10,2);
begin
    salariu_mediu_angajati(v_id_sediu, v_salariu_mediu);
    DBMS_OUTPUT.PUT_LINE('Salariul mediu al angajatilor din sediul ' || v_id_sediu || ' este: ' || v_salariu_mediu);
end;
```

Script Output x

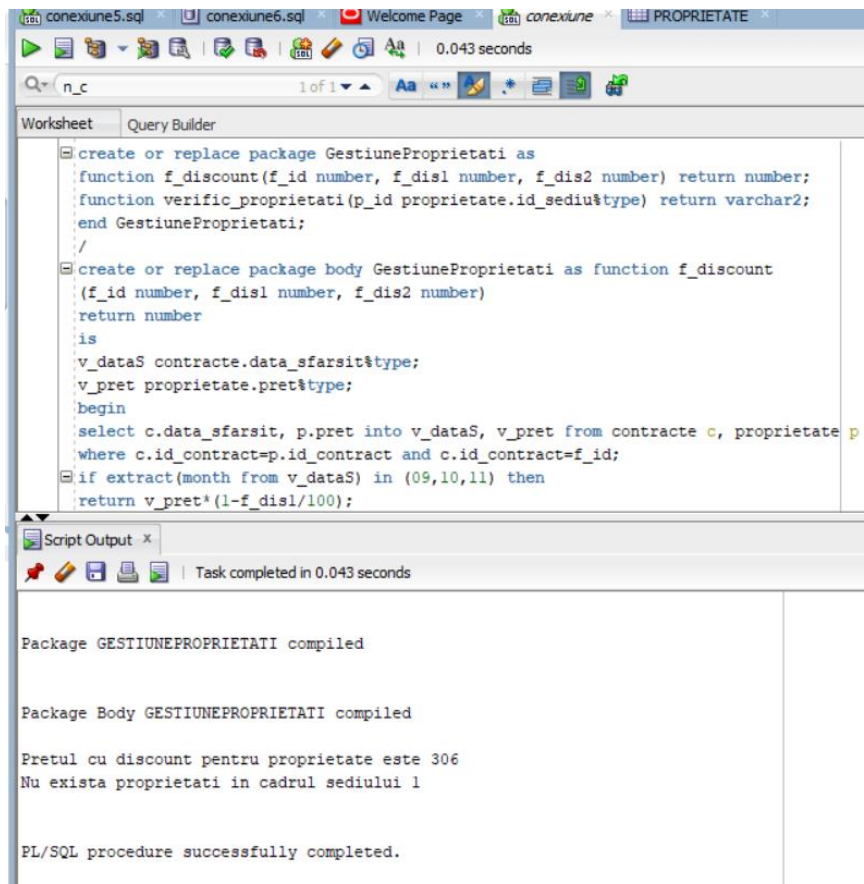
Task completed in 2.912 seconds

end;
Salariul mediu al angajatilor din sediul 102 este: 5000

PL/SQL procedure successfully completed.

5. PACHETE:

In exemplul alaturat am modificat codul de mai sus pentru a adauga functiile f_discount si verific_proprietati In cadrul unui pachet ce se ocupa cu gestiunea proprietatilor.



SCRIPT Pachete:

```
create or replace package GestiuneProprietati as

function f_discount(f_id number, f_dis1 number, f_dis2 number) return number;

function verific_proprietati(p_id proprietate.id_sediu%type) return varchar2;

end GestiuneProprietati;

/

create or replace package body GestiuneProprietati as function f_discount
(f_id number, f_dis1 number, f_dis2 number)
return number
is
--continua cod de mai sus
end f_discount;

function verific_proprietati(p_id proprietate.id_sediu%type)
return varchar2
is
--la fel ca mai sus
end verific_proprietati;
```

--APEL FUNCTII DIN CADRUL PACHETULUI

```
declare

v_id number:=102;

v_sediu number:=1;

v_mesaj varchar2(200);

begin

dbms_output.put_line('Pretul cu discount pentru proprietate este ' || GestiuoneProprietati.f_discount(v_id, 15, 5));

v_mesaj:=GestiuneProprietati.verific_proprietati(v_sediu);

dbms_output.put_line(v_mesaj);

exception

when others then

dbms_output.put_line('A aparut o eroare: ' || SQLERRM);

end;

/
```

●Atunci cand accesam baza de date a angajatilor, vrem sa vedem si denumirea functiei pe care o ocupa, nu numai id-ul acesteia. Prin urmare se va adauga o noua coloana in tabela Angajati si se vor sincroniza datele in functie de id si denumirea functiei. In cazul in care un angajat isi schimba functia in agentie, sa se modifice atat id-ul cat si denumirea.

```
alter table angajati add (denumire_functie varchar2(100));

create or replace package angajatiPkg as
    procedure actualizeaza_denumire_functie;
end angajatiPkg;
/

create or replace package body angajatiPkg as
    cursor angajat_cursor is select id_angajati, id_functie from angajati for update;

    procedure actualizeaza_denumire_functie is
        v_nume_functie functii.nume_functie%type;
    begin
        for angajat in angajat_cursor loop
            select nume_functie into v_nume_functie from functii where id_functie = angajat.id_functie;
            update angajati set denumire_functie = v_nume_functie where id_angajati = angajat.id_angajati;
        end loop;
    end;
end;
/
```

Script Output x

Task completed in 0.084 seconds

Package ANGAJATIPKG compiled

Package Body ANGAJATIPKG compiled

PL/SQL procedure successfully completed.

```

begin
  for angajat in angajat_cursor loop
    select nume_functie into v_nume_functie from functii where id_functie = angajat.id_functie;
    update angajati set denumire_functie = v_nume_functie where current of angajat_cursor;
  end loop;
  commit;
end actualizeaza_denumire_functie;
end angajatiPkg;
/

begin
  angajatiPkg.actualizeaza_denumire_functie;
end;
/

update angajati set id_functie =2 where id_angajati=15;
begin
  angajatiPkg.actualizeaza_denumire_functie;
end;
/

```

Script Output x

Task completed in 0.084 seconds

PL/SQL procedure successfully completed.

1 row updated.

PL/SQL procedure successfully completed.

14	Popa Vlad	756823564	5000	6	103 HR
15	Ionescu Alina	748578945	5500	6	100 HR
16	Nica Andreea	741845612	5500	4	100 Agent imobiliar
14	Popa Vlad	756823564	5000	6	103 HR
15	Ionescu Alina	748578945	5500	2	100 Programator
16	Nica Andreea	741845612	5500	4	100 Agent imobiliar
17	Paun Maria	768541236	5250	4	101 Agent imobiliar

6. DECLANSATORI:

●Trigger 1:

Agentia imobiliara are cate un sediu in diferite orase si fiecarui sediu ii corespunde un numar de proprietati. Se doreste ca id ul sediului asociat fiecarei proprietati sa nu se poata modifica. In cazul in care cineva doreste modificarea acestui id, sa se afiseze un mesaj.

```
create or replace trigger VerificareIdSediu
before update of id_sediu on proprietate
for each row
begin
    if :old.id_sediu <> :new.id_sediu then
        raise_application_error(-20222, 'Nu puteti schimba sediul unei proprietati!');
    end if;
end;
/

UPDATE Proprietate
SET id_sediu = 100
WHERE id_proprietate = 201;
```

Script Output x

Task completed in 0.135 seconds

Trigger VERIFICAREIDSEDIU compiled

Error starting at line : 524 in command -

```
UPDATE Proprietate
SET id_sediu = 100
WHERE id_proprietate = 201
Error report -
ORA-20222: Nu puteti schimba sediul unei proprietati!
ORA-06512: at "BUSEB_50.VERIFICAREIDSEDIU", line 3
```

●Trigger 2:

Pentru compania noastra avem un numar de functii. In cazul in care adaugam un angajat sau modificam functia unui angajat, se doreste ca acesta sa nu aiba o functie pe care agentia sa n-o aiba ca functie.

```
create or replace trigger VerificareFuncctie
before insert or update on angajati
for each row
declare
    v_count number;
begin
    select count(*) into v_count
    from functii
    where id_funcctie = :new.id_funcctie;

    if v_count = 0 then
        raise_application_error(-20001, 'Id_funcctie nu există în tabela functii!');
    end if;
end;
/

update angajati set id_funcctie=10 where id_angajati=10;
```

Script Output x

Task completed in 0.062 seconds

Trigger VERIFICAREFUNCTIE compiled

Error starting at line : 546 in command -

```
update angajati set id_funcctie=10 where id_angajati=10
Error report -
ORA-20001: Id_funcctie nu exista în tabela functii!
ORA-06512: at "BUSEB_50.VERIFICAREFUNCTIE", line 9
ORA-04088: error during execution of trigger 'BUSEB_50.VERIFICAREFUNCTIE'
```

●Trigger 3:

Se doreste aplicarea unei reduceri la nivelul unei proprietati. In cazul in care, noul pret este mai mic decat media de pret din cadrul orasului din care face parte proprietatea, reducerea sa nu se aplice. In caz contrar, aplicam o reducere de 20%.

```
create global temporary table temp_proprietate (  
  id_proprietate number,  
  pret number  
) on commit preserve rows;  
  
create or replace trigger ReduceriPret  
before update of pret  
on proprietate  
for each row  
begin  
  insert into temp_proprietate (id_proprietate, pret)  
  values (:new.id_proprietate, :new.pret);  
end;  
/  
  
create or replace procedure aplicarereduceri is  
  v_media_pret_sediu number;  
begin  
  for rec in (select id_proprietate, pret from temp_proprietate) loop  
    select avg(pret)  
    into v_media_pret_sediu  
    from proprietate
```

Script Output x
Task completed in 0.082 seconds

Trigger REDUCERIPRET compiled

Procedure APLICAREREDUCERI compiled

A aparut o eroare: ORA-20222: Nu puteti reduce pretul acestei proprietati sub media din sediu!

Worksheet Query Builder

```
select avg(pret)  
into v_media_pret_sediu  
from proprietate  
where id_sediu = (  
  select id_sediu  
  from proprietate  
  where id_proprietate = rec.id_proprietate  
);  
  
if v_media_pret_sediu is not null and rec.pret < v_media_pret_sediu then  
  raise_application_error(-20222, 'Nu puteti reduce pretul acestei proprietati sub media din sediu!');  
else  
  update proprietate  
  set pret = rec.pret - 0.2 * rec.pret  
  where id_proprietate = rec.id_proprietate;  
end if;  
end loop;  
  
delete from temp_proprietate;  
  
commit;  
exception  
when others then  
  rollback;
```

Script Output x
Task completed in 0.082 seconds

Procedure APLICAREREDUCERI compiled



A aparut o eroare: ORA-20222: Nu puteti reduce pretul acestei proprietati sub media din sediu!

PL/SQL procedure successfully completed.

```
when others then
    rollback;
    dbms_output.put_line('A apărut o eroare: ' || sqlerrm);
end;
/

DECLARE
    v_id_proprietate proprietate.id_proprietate%TYPE := 200;
BEGIN
    UPDATE proprietate SET pret = 100 WHERE id_proprietate = v_id_proprietate;
    aplicareReduceri;
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('A apărut o eroare: ' || SQLERRM);
END;
/
```

Script Output x

  | Task completed in 0.082 seconds

```
procedure APLICAREREDUCERI compiled
```

7. SCRIPT:

create table functii(

id_functie number(3) constraint pk_functie primary key,

nume_functie varchar2(30) constraint nn_numef check(nume_functie is not null));

create table sediul_ag(

id_sediu number(2) constraint pk_key primary key,

localitate varchar2(20) constraint nn_loc check(localitate is not null),

email varchar2(30) constraint nn_email check(email is not null),

telefon number(10));

create table angajati(

```
id_angajati number(3) constraint pk_angajati primary key,  
nume_ag varchar2(30) constraint nn_nume_ag check( nume_ag is not null),  
telefon number(10),  
salariul number(4),  
id_functie number(3),  
id_sediu number (3),  
constraint fk_idfu foreign key(id_functie) references functii(id_functie),  
constraint fk_idse foreign key(id_sediu) references sediul_ag(id_sediu));
```

```
create table clienti(  
id_client number(3) constraint pk_client primary key,  
nume_cl varchar2(30) constraint nn_nume_cl check(nume_cl is not null),  
telefon number(10),  
email_cl varchar2(30) constraint nn_email_cl check(email_cl is not null),  
id_contract number(4));
```

```
create table proprietate(  
id_proprietate number(3) constraint pk_proprietate primary key,  
pret number(10),  
disponibilitate number(2),  
suprafata number(4),  
nr_camere number(7),  
id_sediu number(3),  
id_client number(3),  
id_contract number(4));
```

```
create table contracte(  
id_contract number(4) constraint pk_contract primary key,  
id_client number(3),  
id_proprietate number(3),  
constraint fk_iddclient foreign key(id_client) references clienti(id_client),  
constraint fk_iddproprietate foreign key(id_proprietate) references  
proprietate(id_proprietate),  
data_inchiriere DATE DEFAULT SYSDATE );
```

```
describe proprietate;
```

```
describe contracte;
```

```
ALTER TABLE proprietate ADD CONSTRAINT fk_idsediu foreign key (id_sediu) references  
sediu_ag(id_sediu);
```

```
ALTER TABLE contracte
```

```
ADD (data_sfarsit date);
```

```
ALTER TABLE sediu_ag modify id_sediu number(3);
```

```
ALTER TABLE clienti modify id_client number(4);
```

```
ALTER TABLE clienti modify nume_cl varchar2(35);
```

```
describe clienti;
```

```
describe functii;
```

```
insert into functii values(1, 'CEO');
```

```
insert into functii values(2, 'Programator');
```

```
insert into functii values(3, 'Contabil');
```

```
insert into functii values(4, 'Agent imobiliar');
```

```
insert into functii values(5, 'Reprezentant vanzari credite');
```

```
insert into functii values(6, 'HR');
```

```
describe sediul_ag;
```

```
insert into sediul_ag values(100, 'Bucuresti' , 'agimobbucuresti' , '0245689562');
```

```
insert into sediul_ag values(101, 'Cluj' , 'agimobcluj' , '0245689563');
```

```
insert into sediul_ag values(102, 'Brasov' , 'agimobbrasov' , '0245689564');
```

```
insert into sediul_ag values(103, 'Timisoara' , 'agimobtimisoara' , '0245689565');
```

```
insert into sediul_ag values(104, 'Constanta' , 'agimobconstantat' , '0245689566');
```

```
insert into sediul_ag values(105, 'Iasi' , 'agimobiaso' , '0245689567');
```

```
insert into sediul_ag values(106, 'Arad' , 'agimobarad' , '0245689568');
```

```
insert into sediul_ag values(107, 'Oradea' , 'agimoboradea' , '0245689569');
```

```
insert into sediul_ag values(108, 'Sibiu' , 'agimobsibiu' , '0245689570');
```

```
insert into sediul_ag values(109, 'Craiova' , 'agimobcraiova' , '0245689571');
```

```
describe angajati;
```

```
insert into angajati values(10, 'Anghel Cosmin' , '0749635108' , 9500, 1, 100);
```

```
insert into angajati values(11, 'Maria Ion' , '0743526599' , 6000, 2,101);
```

```
insert into angajati values(12, 'Popescu Cristian' , '0756526599' ,4500, 3, 102);
```

```
insert into angajati values(13, 'Iancu Andrei' , '0756851234' , 5000, 4, 103);
```

```
insert into angajati values(14, 'Popa Vlad' , '0756823564' , 5000, 6, 103);
```

```
insert into angajati values(15, 'Ionescu Alina' , '0748578945' , 4500,3, 100);
```

```
insert into angajati values(16, 'Nica Andreea' , '0741845612' , 5500, 4, 100);
```

```
insert into angajati values(17, 'Paun Maria' , '0768541236' , 5000, 4,101);
```

```
insert into angajati values(18, 'Petre Anca' , '0745123896' , 5500, 4,102);
```

```
insert into angajati values(19, 'Marin Sarah' , '0743278945' , 6500, 2, 104);  
insert into angajati values(20, 'Popescu Alex', '0754398653', 5500, 4,105);  
insert into angajati values(21, 'Diaconu Marius', '0745632108', 5000, 6, 104);  
insert into angajati values(22, 'Ion Vladimir', '0745635108', 5000, 5, 105);  
insert into angajati values(23, 'Popescu Sorin', '0754398753', 5500, 4,104);  
insert into angajati values(24, 'Horia Marian', '0743525999', 4500, 6,100);
```

```
describe proprietate;
```

```
insert into proprietate values(200, 300, 1, 45 , 2, 100,NULL, 100 );  
insert into proprietate values(201, 400, 1, 60 , 2 ,101, NULL, 101);  
insert into proprietate values(202, 450, 0, 70 , 3, 102,100, 500);  
insert into proprietate values(203, 320, 0, 45 , 2 ,103, 101, 501);  
insert into proprietate values(204, 500, 0, 85 , 3 ,102, 104, 504);  
insert into proprietate values(205, 300, 1, 45 , 2, 108, NULL, 102 );  
insert into proprietate values(206, 350, 1, 60 , 2 ,105,NULL, 103);  
insert into proprietate values(207, 450, 0, 70 , 3, 102,109, 507 );  
insert into proprietate values(208, 320, 0, 45 , 2 ,100, 107, 502);  
insert into proprietate values(209, 450, 0, 85 , 3 ,104, 105, 505);
```

```
describe clienti;
```

```
insert into clienti values(100, 'Mincu Andra', '0765984523' , 'andramincu' ,500);  
insert into clienti values(101, 'Minulescu Andrei' , '0745986523', 'andreiminulescu' , 501);  
insert into clienti values(102, 'Topan Matias', '0755487456', 'matiastopan' , 502);  
insert into clienti values(103, 'Popa Adrian', '0723045120', 'adrianpopa', 503);  
insert into clienti values(104, 'Radu Daniela', '0789960231', 'danielaradu' , 504);  
insert into clienti values(105, 'Paun Andrada', '0765984553' , 'andradapaun' ,505);
```

```
insert into clienti values(106, 'Minulescu Mara' , '0745466523', 'maraminulescu' , 501);
```

```
insert into clienti values(107, 'Cristea Toader', '0758787456', 'toadercristea' , 502);
```

```
insert into clienti values(108, 'Popa Teodora', '0723045567', 'teodorapopa', 506);
```

```
insert into clienti values(109, 'Costea Elena', '0789964523', 'elenacostea' , 507);
```

```
describe contracte;
```

```
insert into contracte values(100, NULL,200, NULL, NULL );
```

```
insert into contracte values(101, NULL,201, NULL, NULL );
```

```
insert into contracte values(102, 100,202, sysdate , TO_DATE('14-SEP-2023', 'DD-MON-YYYY'));
```

```
insert into contracte values(103, 101,203, sysdate , TO_DATE('14-SEP-2024', 'DD-MON-YYYY'));
```

```
insert into contracte values(104, 104,204, sysdate , TO_DATE('24-NOV-2023', 'DD-MON-YYYY'));
```

```
insert into contracte values(105, NULL,205, NULL, NULL );
```

```
insert into contracte values(106, NULL,206, NULL, NULL );
```

```
insert into contracte values(107, 109,207, sysdate , TO_DATE('14-APR-2023', 'DD-MON-YYYY'));
```

```
insert into contracte values(108, 107,208, sysdate , TO_DATE('14-FEB-2024', 'DD-MON-YYYY'));
```

```
insert into contracte values(109, 105,209, sysdate , TO_DATE('29-OCT-2024', 'DD-MON-YYYY'));
```

```
--CONTINUE PL/SQL:
```

```
set serveroutput on
```

```
DECLARE
```

```
v_id angajati.id_angajati%type:=15;
```

```
v_nume angajati.nume_ag%type;
```

```
v_salariul angajati.salariul%type;
```



```

begin

loop

select nume_ag, salariul into v_nume, v_salariul from angajati

where id_angajati=v_id;

DBMS_OUTPUT.PUT_LINE('Angajatul ' || v_nume || ' cu id-ul: ' || v_id || ' are salariul ' || v_salariul);

v_id:=v_id+1;

exit when v_id>20;

end loop;

end;

/

```

```

DECLARE

v_id angajati.id_angajati%type:=&id;

v_nume angajati.nume_ag%type;

v_salariul angajati.salariul%type;

begin

select nume_ag, salariul into v_nume, v_salariul from angajati

where id_angajati=v_id;

if v_salariul between 4000 and 5000 then

    dbms_output.put_line('Angajatul ' || v_nume || ' are salariul cuprins intre 4000 si 5000');

else

    dbms_output.put_line('Angajatul ' || v_nume || ' nu are salariul cuprins in intervalul dat!');

end if;

end;

/

```

```

declare

```

```

v_id proprietate.id_proprietate%type:=&id;

v_pret proprietate.pret%type;

v_suprafata proprietate.suprafata%type;

BEGIN

SELECT pret, suprafata into v_pret, v_suprafata from proprietate where id_proprietate=v_id;

dbms_output.put_line ('Pretul initial al proprietatii cu id-ul ' || v_id || ' este: ' || v_pret);

v_pret:= CASE WHEN v_suprafata < 50 THEN 1.2* v_pret

            WHEN v_suprafata between 50 and 75 THEN 1.3 * v_pret

            ELSE 1.1* v_pret END;

dbms_output.put_line('Pretul final este: ' || v_pret);

end;

/

DECLARE

type prop is table of proprietate%rowtype index by pls_integer;

v_desc prop;

v_pret proprietate.pret%type:=0;

BEGIN

for i in 200..205 loop

SELECT * into v_desc(i) from proprietate where id_proprietate=i;

end loop;

for i in v_desc.first..v_desc.last loop

dbms_output.put_line('Proprietatea: ' || i || ' are pretul de: ' || v_desc(i).pret);

end loop;

SELECT avg(pret) into v_pret from proprietate where id_proprietate between 200 and 205;

dbms_output.put_line('Pretul mediu al celor ' || v_desc.count || ' proprietati din tabloul indexat, este: ' || v_pret

);

END;

```

/

declare

v_sal angajati.salariul%type;

v_sm v_sal%type;

i number(2):=10;

begin

select avg(salariul) into v_sm from angajati;

dbms_output.put_line('Salariul mediu este ' || v_sm);

while i<15 loop

select salariul into v_sal from angajati where id_angajati=i;

if v_sal>v_sm then

dbms_output.put_line('Angajatul cu id ' || i || ' are salariul ' || v_sal);

end if;

i:=i+1;

end loop;

end;

/

declare

cursor f is select

f.ume_functie, count(a.id_functie) nr

from functii f, angajati a

where f.id_functie=a.id_functie and f.ume_functie not in ('CEO')

group by f.ume_functie

having count(a.id_functie)>1

order by nr desc;

begin

```

for r in f

loop

dbms_output.put_line('Postul de ' || r.ume_functie || ' are ' || r.nr || ' angajati!');

end loop;

end;

/

```

```

declare

v_id angajati.id_angajati%type;

v_sal angajati.salariul%type;

cursor c is select id_angajati, salariul

from angajati

order by salariul desc;

begin

open c;

loop

fetch c into v_id, v_sal;

exit when c%rowcount>4 or c%notfound;

dbms_output.put_line('Angajatul cu id ' || v_id || ' are sal ' || v_sal);

end loop;

close c;

end;

/

```

```

ALTER TABLE Sediul_ag RENAME COLUMN localitate TO oras;

```

```

declare

cursor c_sed(v_venit number) is

select s.id_sediu, s.oras, sum(p.pret) venit

from proprietate p, sediul_ag s

where p.id_sediu=s.id_sediu and p.id_client is not null

```

```

group by s.id_sediu, s.oras

having sum(p.pret)> v_venit

order by venit;

v_venit number;

r_sediu c_sed%rowtype;

begin v_venit:=450;

dbms_output.put_line('Lista sediilor in care valoare proprietatilor inchiriate depaseste '||v_venit);

if not c_sed%isopen then

open c_sed(v_venit);

end if;

loop fetch c_sed into r_sediu;

exit when c_sed%notfound;

dbms_output.put_line(r_sediu.id_sediu || ' ' || r_sediu.oras || ' ' || r_sediu.venit);

end loop;

end;

/

```

```

select * from clienti;

insert into clienti values(110, 'Popa Adrian', '723045120', 'popaadrian', 510);

```

```

declare

v_id clienti.id_client%type;

v_nume clienti.nume_cl%type;

v_tel clienti.telefon%type;

begin

select id_client, nume_cl, telefon into v_id, v_nume, v_tel

from clienti where nume_cl like '%Popa Adrian%';

dbms_output.put_line(v_nume|| ' are id-ul ' ||v_id || ' si numarul de telefon: ' ||v_tel);

```

```
exception

when too_many_rows then

dbms_output.put_line('Sunt mai multi clienti cu acest nume!');

when no_data_found then

dbms_output.put_line('Nu exista un client cu acest nume!');

when others then

dbms_output.put_line('Avem o alta eroare!');

end;

/
```

```
declare

cursor c is select * from contracte

where extract (month from data_sfarsit) in (09, 10, 07);

r c%rowtype;

begin

open c;

loop

fetch c into r;

exit when c%notfound;

dbms_output.put_line(r.data_sfarsit);

close c;

end loop;

exception

when invalid_cursor then

dbms_output.put_line('Cursorul este inchis!');

end;

/
```

```

declare v_parola varchar2(20);

v_id clienti.id_client%type;

v_contract clienti.id_contract%type;

begin

v_id:=&v_id;

v_parola:='&v_parola';

if v_parola like 'client' then

select id_contract into v_contract from clienti where id_client=v_id;

dbms_output.put_line('Clientului cu id-ul ' || v_id || ' ii corespunde contractul cu nr-ul: ' || v_contract);

else

raise_application_error(-20989, 'Nu aveti acces!');

end if;

end;

/

```

--functii

create or replace function f_discount

(f_id number, f_dis1 number, f_dis2 number)

return number

is

v_dataS contracte.data_sfarsit%type;

v_pret proprietate.pret%type;

begin

select c.data_sfarsit, p.pret into v_dataS, v_pret from contracte c, proprietate p

where c.id_contract=p.id_contract and c.id_contract=f_id;

```
if extract(month from v_data$) in (09,10,11) then
```

```
return v_pret*(1-f_dis1/100);
```

```
else
```

```
return v_pret*(1-f_dis2/100);
```

```
end if;
```

```
end;
```

```
/
```

```
select c.data_sfarsit, p.pret from contracte c, proprietate p
```

```
where c.id_contract=p.id_contract and c.id_contract=102;
```

```
begin
```

```
dbms_output.put_line('Pretul cu discount pentru proprietate este ' || f_discount(102, 15, 5));
```

```
end;
```

```
/
```

```
create or replace function promovare_ang(p_id angajati.id_angajati%type)
```

```
return number
```

```
is
```

```
v_sal angajati.salariul%type;
```

```
ang_except Exception;
```

```
begin
```

```
select salariul into v_sal from angajati where id_angajati=p_id;
```

```
if v_sal<5000 then
```

```
return v_sal+500;
```

```
elsif v_sal between 5000 and 5500 then
```

```
return v_sal+250;
```

```
else
```

```
RETURN v_sal;
```



```
end if;

exception

when no_data_found then

return 0;

end;

/
```

```
declare

v_id angajati.id_angajati%type :=&id;

v_marire number;

begin

v_marire:=promovare_ang(v_id);

if v_id is not null then

update angajati set salariul=v_marire where id_angajati=v_id;

else

dbms_output.put_line('nu exista angajatul cautat!');

end if;

end;

/
```

```
create or replace function verific_proprietati(p_id proprietate.id_sediu%type)

return varchar2

is

v_idP proprietate.id_proprietate%type;

BEGIN

select id_proprietate into v_idP from proprietate where id_sediu=p_id;

exception

when no_data_found then

return 'Nu exista proprietati in cadrul sediului '||p_id;
```

```
when too_many_rows then

return 'Exista proprietati in cadrul sediului '||p_id;

when others then

return 'a aparut o alta problema';

end;

/
```

```
declare

v_id proprietate.id_sediu%type :=&id;

v_mesaj varchar2(200);

begin

v_mesaj:=verific_proprietati(v_id);

if v_id is not null then

dbms_output.put_line(v_mesaj);

else

dbms_output.put_line('nu exista sediul cautat!');

end if;

end;

/
```

```
--proceduri

create or replace procedure venitan(p_nr_proprietati out number)

is

e1 exception;

begin

update proprietate set pret=pret*1.1

where id_client is not null and id_sediu in (select id_sediu from proprietate

group by id_sediu

having count(id_sediu)>=3);
```

```

if sql%found then p_nr_proprietati:=sql%rowcount;

else

raise e1;

end if;

exception

when e1 then

p_nr_proprietati:=0;

end;

/

declare

v_nr number;

begin

venitan( v_nr);

if v_nr>0 then

dbms_output.put_line('Au fost marite preturile a ' || v_nr || ' proprietati');

else

dbms_output.put_line('nu exista proprietati');

end if;

end;

/

create or replace PROCEDURE contracteFinal(p_an in number, p_nr_contracte out number)

is

e1 exception;

v_count number := 0;

v_idC contracte.id_contract%type;

begin

for c in (select id_contract from contracte where data_sfarsit is not null and extract(year from contracte.data_sfarsit) = p_an)

loop

```

```

    v_idC := c.id_contract;

    v_count := v_count + 1;

end loop;


if v_count > 0 then

    p_nr_contracte := v_count;

else

    raise e1;

end if;

exception

when e1 then

    p_nr_contracte := 0;

end;

/

declare

v_an number :=&an;

v_nr number;

begin

contracteFinal(v_an, v_nr);

if v_nr>0 then

dbms_output.put_line('In anul ' || v_an || ' se incheie ' || v_nr || ' contracte');

else

dbms_output.put_line('Nu exista contracte incheiate in anul' || v_an);

end if;

end;

/

```

create or replace procedure salariu_mediu_angajati (

```

    v_id_sediu angajati.id_sediu%type,

```

```

    v_salariu_mediu out number
)
is
    v_salariu_total number(10,2);
    v_numar_angajati number;
begin
    select sum(salariul), count(*)
    into v_salariu_total, v_numar_angajati
    from angajati
    where id_sediu = v_id_sediu;

    if v_numar_angajati > 0 then
        v_salariu_mediu := v_salariu_total / v_numar_angajati;
    else
        v_salariu_mediu := 0;
    end if;
end;
/

declare
    v_id_sediu angajati.id_sediu%type := &id_sediu;
    v_salariu_mediu number(10,2);
begin
    salariu_mediu_angajati(v_id_sediu, v_salariu_mediu);

    DBMS_OUTPUT.PUT_LINE('Salariul mediu al angajatilor din sediul ' || v_id_sediu || ' este: ' ||
v_salariu_mediu);
end;
/

alter table angajati add (denumire_functie varchar2(100));

```

```
create or replace package angajatiPkg as

    procedure actualizeaza_denumire_functie;

end angajatiPkg;

/
```

```
create or replace package body angajatiPkg as
```

```
    cursor angajat_cursor is select id_angajati, id_functie from angajati for update;
```

```
    procedure actualizeaza_denumire_functie is
```

```
        v_nume_functie functii.nume_functie%type;
```

```
    begin
```

```
        for angajat in angajat_cursor loop
```

```
            select nume_functie into v_nume_functie from functii where id_functie = angajat.id_functie;
```

```
            update angajati set denumire_functie = v_nume_functie where current of angajat_cursor;
```

```
        end loop;
```

```
        commit;
```

```
    end actualizeaza_denumire_functie;
```

```
end angajatiPkg;
```

```
/
```

```
begin
```

```
    angajatiPkg.actualizeaza_denumire_functie;
```

```
end;
```

```
/
```

```
update angajati set id_functie =2 where id_angajati=15;
```

```
begin
```

```
    angajatiPkg.actualizeaza_denumire_functie;  
end;  
  
/
```

```
--triggeri  
  
create or replace trigger VerificareIdSediu  
before update of id_sediu on proprietate  
for each row  
begin  
    if :old.id_sediu <> :new.id_sediu then  
        raise_application_error(-20222, 'Nu puteti schimba sediul unei proprietati!');  
    end if;  
end;  
  
/
```

```
UPDATE Proprietate  
SET id_sediu = 100  
WHERE id_proprietate = 201;
```

```
create or replace trigger VerificareFunctie  
before insert or update on angajati  
for each row  
declare  
    v_count number;  
begin  
    select count(*) into v_count  
    from functii
```

```
where id_functie = :new.id_functie;
```

```
if v_count = 0 then
```

```
    raise_application_error(-20001, 'Id_functie nu există în tabela functii!');
```

```
end if;
```

```
end;
```

```
/
```

```
update angajati set id_functie=10 where id_angajati=10;
```

```
create global temporary table temp_proprietate (
```

```
    id_proprietate number,
```

```
    pret number
```

```
) on commit preserve rows;
```

```
create or replace trigger ReduceriPret
```

```
before update of pret
```

```
on proprietate
```

```
for each row
```

```
begin
```

```
    insert into temp_proprietate (id_proprietate, pret)
```

```
    values (:new.id_proprietate, :new.pret);
```

```
end;
```

```
/
```

```
create or replace procedure aplicarereduceri is
```

```
    v_media_pret_sediu number;
```

```
begin
```



```

for rec in (select id_proprietate, pret from temp_proprietate) loop

    select avg(pret)

    into v_media_pret_sediu

    from proprietate

    where id_sediu = (

        select id_sediu

        from proprietate

        where id_proprietate = rec.id_proprietate

    );

    if v_media_pret_sediu is not null and rec.pret < v_media_pret_sediu then

        raise_application_error(-20222, 'Nu puteti reduce pretul acestei proprietati sub media din sediu!');

    else

        update proprietate

        set pret = rec.pret - 0.2 * rec.pret

        where id_proprietate = rec.id_proprietate;

    end if;

end loop;

delete from temp_proprietate;

commit;

exception

when others then

    rollback;

    dbms_output.put_line('A apărut o eroare: ' || sqlerrm);

end;

/

```

DECLARE

v_id_proprietate proprietate.id_proprietate%TYPE := 200;

BEGIN

UPDATE proprietate SET pret = 100 WHERE id_proprietate = v_id_proprietate;

aplicareReduceri;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS_OUTPUT.PUT_LINE('A apărut o eroare: ' || SQLERRM);

END;

/