Popescu Maria-Daniela

Grupa 243

PROIECT SGBD FERMA

1. Aceasta baza de data a fost facuta pe tema unei ferme de animale, de vreme ce idea aceasta mi s-a parut destul de practica. La fel si entitatile folosite am vrut sa fie realiste.

Astfel exista furnizori care procura hrana animalelor care apartin fermei.

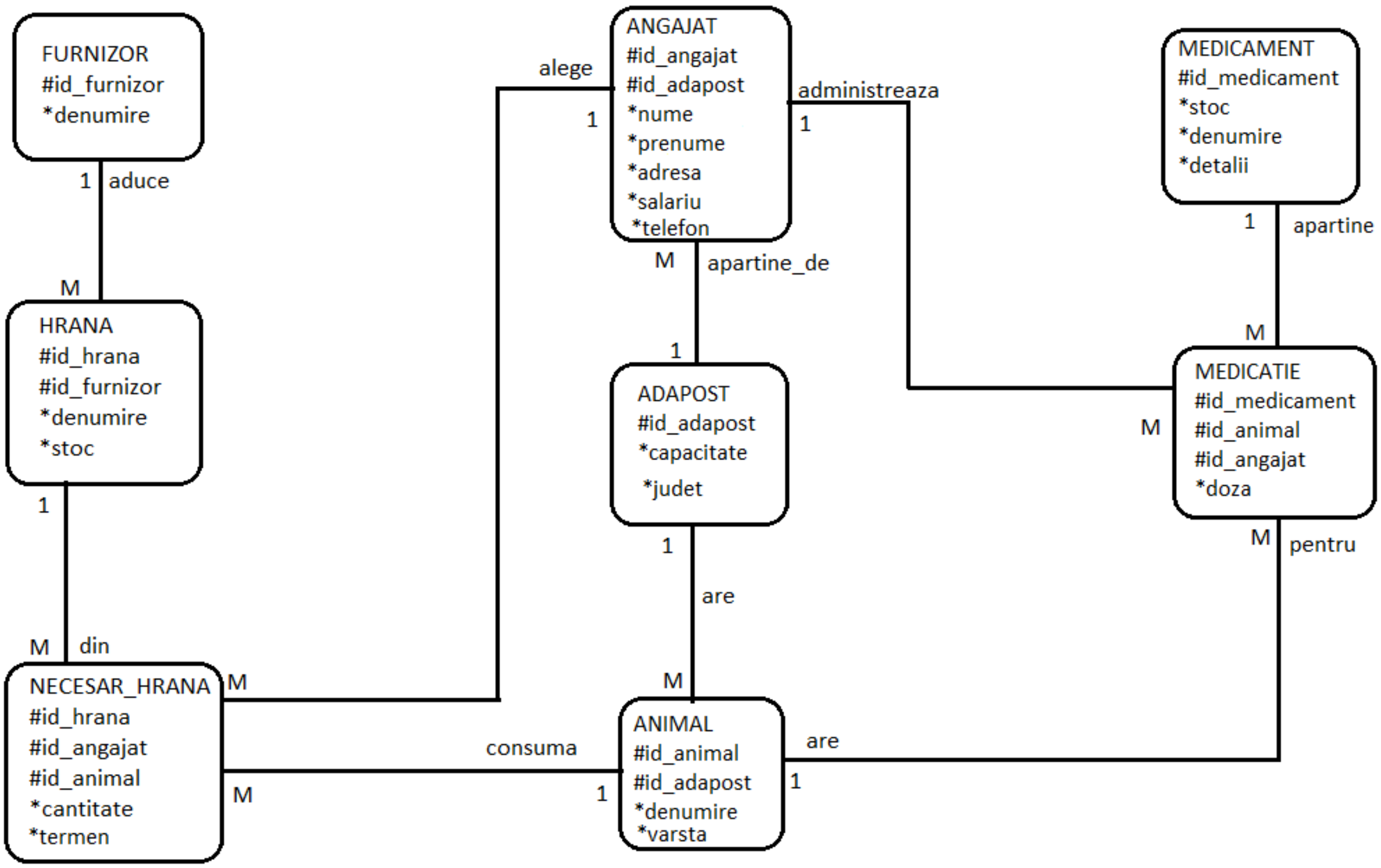
Hrana se gaseste intr-un anume stoc, dar pentru ferma exista o alta cantitate de hrana necesara.

Exista medicamente care trebuie administrate animalelor intr-o anumita doza si de o anumita categorie.

Animalele se gasesc in diferite adaposturi, cu capacitate diferite.

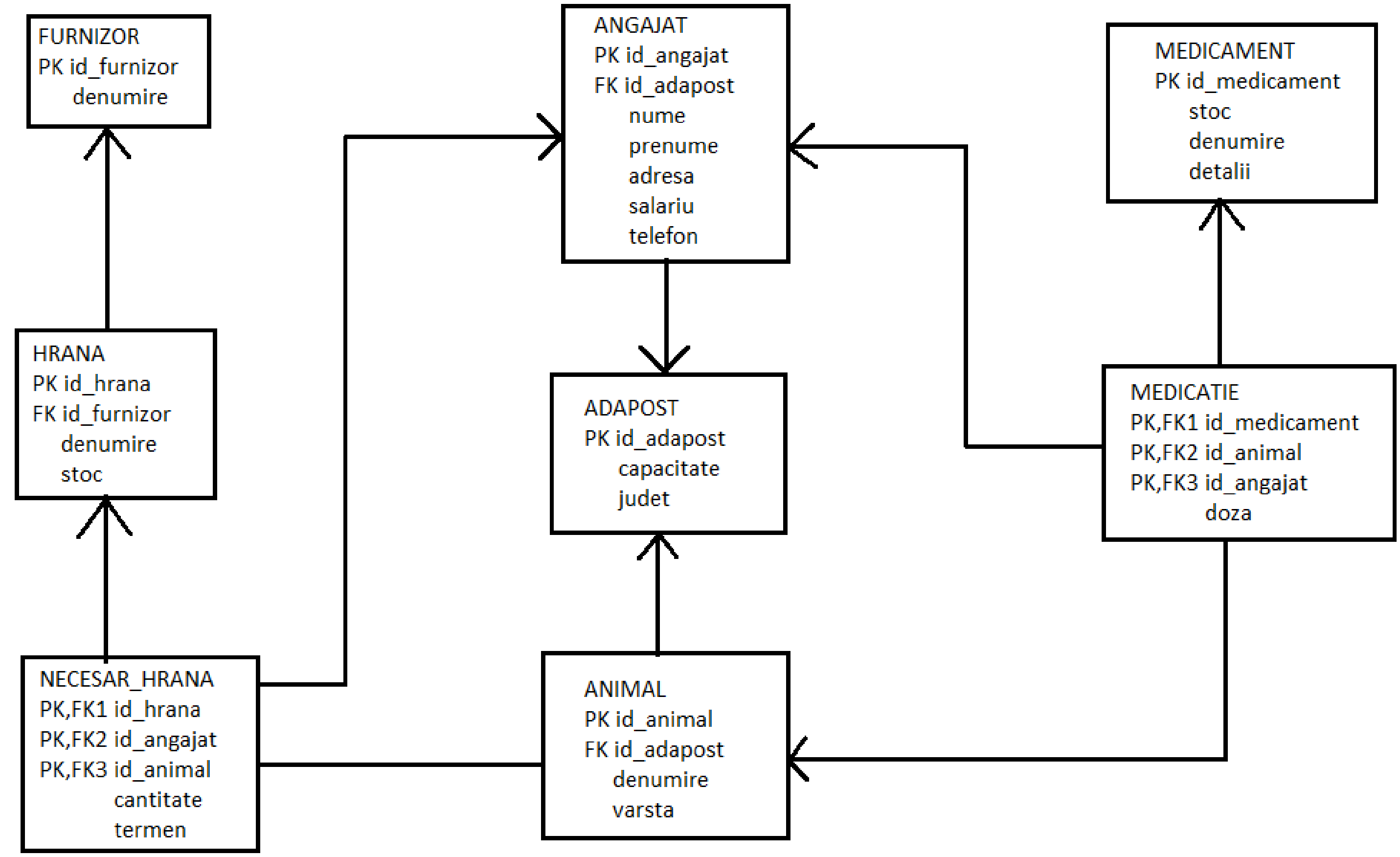
In final, angajatii se asigura ca animalele sunt hranite si ca li se administreaza medicamentele corespunzator.

2. DIAGRAMA ENTITATE-RELATIE



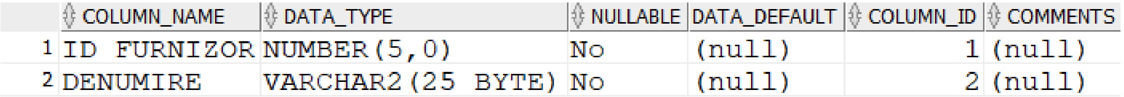
https://zoom.us/j/98865339721?pwd=dUYwaE96MVNacThaWVpGOHBQSFlNdz09

3. DIAGRAMA CONCEPTUALA



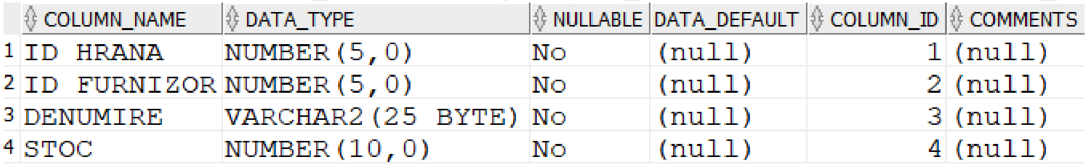
4. Implementare diagram in Oracle

```
create table furnizor (id_furnizor number(5) not null,
                  denumire varchar2(25) not null,
                  constraint furnizor_pk primary key (id_furnizor));
```



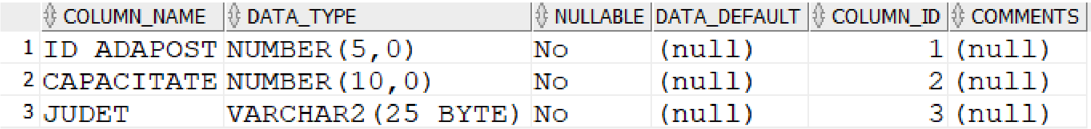
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID FURNIZOR	NUMBER(5,0)	No	(null)	1	(null)
2	DENUMIRE	VARCHAR2(25 BYTE)	No	(null)	2	(null)

```
create table hrana (id_hrana number(5) not null,
               id_furnizor number(5) not null,
               denumire varchar2(25) not null,
               stoc number(10) not null,
               constraint hrana_pk primary key (id_hrana),
               constraint fk_furnizor foreign key (id_furnizor) references furnizor(id_furnizor));
```



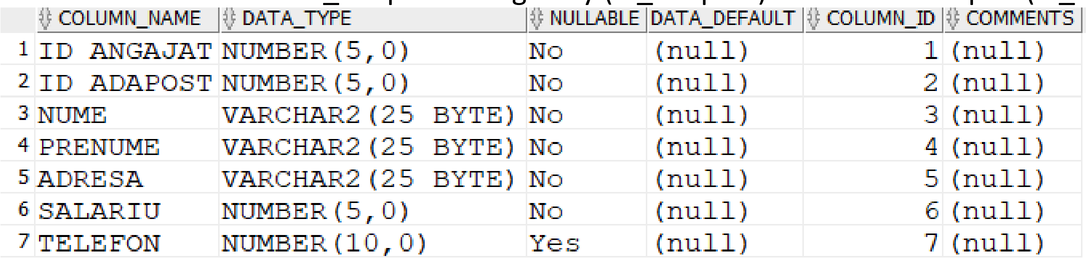
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID HRANA	NUMBER(5,0)	No	(null)	1	(null)
2	ID FURNIZOR	NUMBER(5,0)	No	(null)	2	(null)
3	DENUMIRE	VARCHAR2(25 BYTE)	No	(null)	3	(null)
4	STOC	NUMBER(10,0)	No	(null)	4	(null)

```
create table adapost (id_adapost number(5) not null,
                      capacitate number(10) not null,
                      judet varchar2(25) not null,
                      constraint adapost_pk primary key (id_adapost));
```



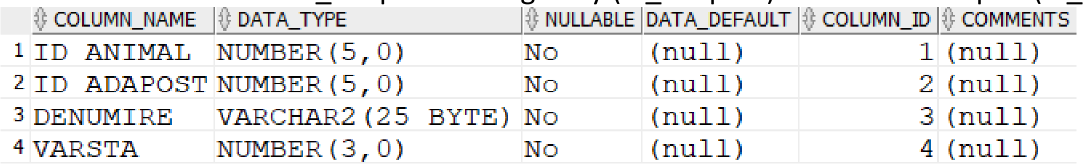
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID ADAPOST	NUMBER(5,0)	No	(null)	1	(null)
2	CAPACITATE	NUMBER(10,0)	No	(null)	2	(null)
3	JUDET	VARCHAR2(25 BYTE)	No	(null)	3	(null)

```
create table angajat (id_angajat number(5) not null,
                      id_adapost number(5) not null,
                      nume varchar2(25) not null,
                      prenume varchar2(25) not null,
                      adresa varchar2(25) not null,
                      salariu number(5) not null,
                      telefon number(10),
                      constraint angajat_pk primary key (id_angajat),
                      constraint fk_adapost foreign key (id_adapost) references adapost(id_adapost));
```



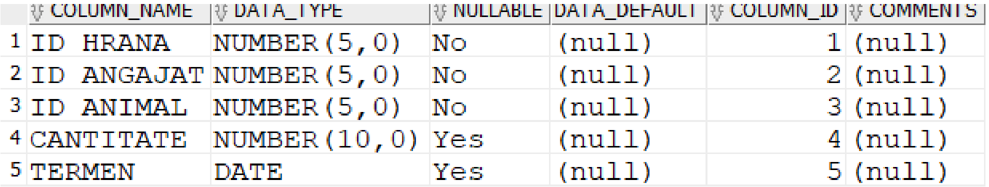
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID ANGAJAT	NUMBER(5,0)	No	(null)	1	(null)
2	ID ADAPOST	NUMBER(5,0)	No	(null)	2	(null)
3	NUME	VARCHAR2(25 BYTE)	No	(null)	3	(null)
4	PRENUME	VARCHAR2(25 BYTE)	No	(null)	4	(null)
5	ADRESA	VARCHAR2(25 BYTE)	No	(null)	5	(null)
6	SALARIU	NUMBER(5,0)	No	(null)	6	(null)
7	TELEFON	NUMBER(10,0)	Yes	(null)	7	(null)

```
create table animal (id_animal number(5) not null,
                     id_adapost number(5) not null,
                     denumire varchar2(25) not null,
                     varsta number(3) not null,
                     constraint animal_pk primary key (id_animal),
                     constraint fk_adapost2 foreign key (id_adapost) references adapost(id_adapost));
```



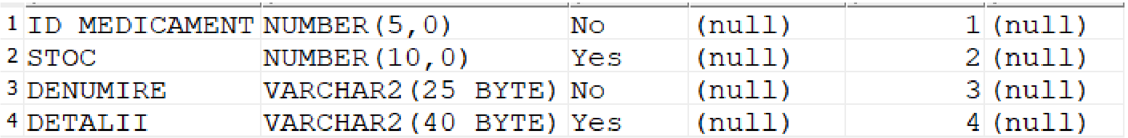
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID ANIMAL	NUMBER(5,0)	No	(null)	1	(null)
2	ID ADAPOST	NUMBER(5,0)	No	(null)	2	(null)
3	DENUMIRE	VARCHAR2(25 BYTE)	No	(null)	3	(null)
4	VARSTA	NUMBER(3,0)	No	(null)	4	(null)

```
create table necesar_hrana (id_hrana number(5) not null,
                            id_angajat number(5) not null,
                            id_animal number(5) not null,
                            cantitate number(10),
                            termen date,
                            constraint fk_hrana foreign key (id_hrana) references hrana(id_hrana),
                            constraint fk_angajat foreign key (id_angajat) references angajat(id_angajat),
                            constraint fk_animal foreign key (id_animal) references animal(id_animal));
```



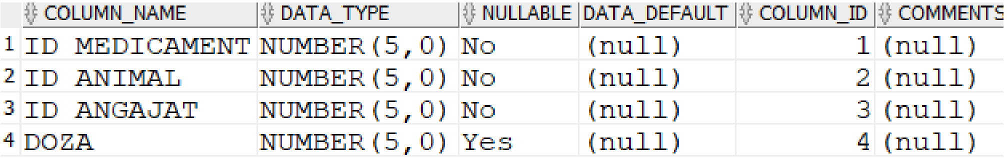
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_HRANA	NUMBER(5,0)	No	(null)	1	(null)
2	ID_ANGAJAT	NUMBER(5,0)	No	(null)	2	(null)
3	ID_ANIMAL	NUMBER(5,0)	No	(null)	3	(null)
4	CANTITATE	NUMBER(10,0)	Yes	(null)	4	(null)
5	TERMEN	DATE	Yes	(null)	5	(null)

```
create table medicament (id_medicament number(5) not null,
                    stoc number(10),
                    denumire varchar2(25) not null,
                    detalii varchar2(40),
                    constraint medicament_pk primary key (id_medicament));
```



1	ID_MEDICAMENT	NUMBER(5,0)	No	(null)	1	(null)
2	STOC	NUMBER(10,0)	Yes	(null)	2	(null)
3	DENUMIRE	VARCHAR2(25 BYTE)	No	(null)	3	(null)
4	DETALII	VARCHAR2(40 BYTE)	Yes	(null)	4	(null)

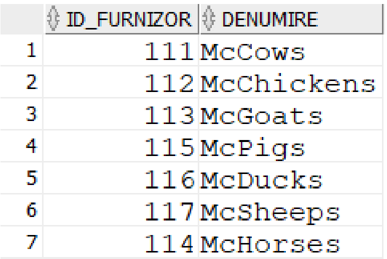
```
create table medicatie (id_medicament number(5) not null,
                    id_animal number(5) not null,
                    id_angajat number(5) not null,
                    doza number(5),
                    constraint fk_angajat2 foreign key (id_angajat) references angajat(id_angajat),
                    constraint fk_animal2 foreign key (id_animal) references animal(id_animal),
                    constraint fk_medicament foreign key (id_medicament) references
medicament(id_medicament) );
```



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_MEDICAMENT	NUMBER(5,0)	No	(null)	1	(null)
2	ID_ANIMAL	NUMBER(5,0)	No	(null)	2	(null)
3	ID_ANGAJAT	NUMBER(5,0)	No	(null)	3	(null)
4	DOZA	NUMBER(5,0)	Yes	(null)	4	(null)

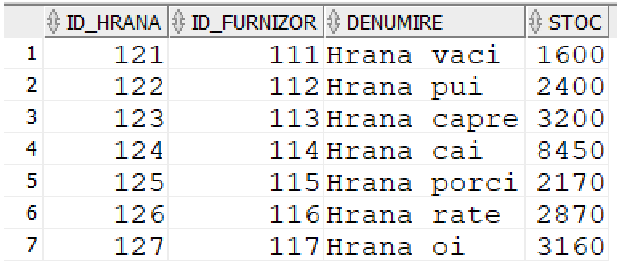
5. Populare tabele

```
--TABELUL FURNIZOR
insert into furnizor(id_furnizor, denumire)
values (111, 'McCows');
insert into furnizor(id_furnizor, denumire)
values (112, 'McChickens');
insert into furnizor(id_furnizor, denumire)
values (113, 'McGoats');
insert into furnizor(id_furnizor, denumire)
values (114, 'McHorses');
insert into furnizor(id_furnizor, denumire)
values (115, 'McPigs');
insert into furnizor(id_furnizor, denumire)
values (116, 'McDucks');
insert into furnizor(id_furnizor, denumire)
values (117, 'McSheeps');
select * from furnizor;
```



	ID_FURNIZOR	DENUMIRE
1	111	McCows
2	112	McChickens
3	113	McGoats
4	115	McPigs
5	116	McDucks
6	117	McSheeps
7	114	McHorses

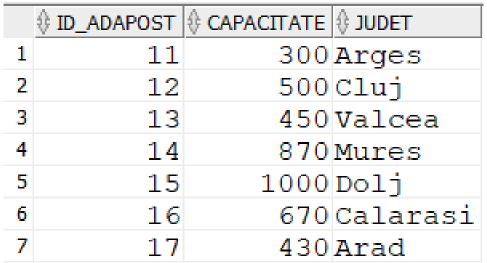
```
--TABELUL HRANA
insert into hrana(id_hrana, id_furnizor, denumire, stoc)
values (121, 111, 'Hrana vaci', 1600);
insert into hrana(id_hrana, id_furnizor, denumire, stoc)
values (122, 112, 'Hrana pui', 2400);
insert into hrana(id_hrana, id_furnizor, denumire, stoc)
values (123, 113, 'Hrana capre', 3200);
insert into hrana(id_hrana, id_furnizor, denumire, stoc)
values (124, 114, 'Hrana cai', 8450);
insert into hrana(id_hrana, id_furnizor, denumire, stoc)
values (125, 115, 'Hrana porci', 2170);
insert into hrana(id_hrana, id_furnizor, denumire, stoc)
values (126, 116, 'Hrana rate', 2870);
insert into hrana(id_hrana, id_furnizor, denumire, stoc)
values (127, 117, 'Hrana oi', 3160);
select * from hrana;
```



	ID_HRANA	ID_FURNIZOR	DENUMIRE	STOC
1	121	111	Hrana vaci	1600
2	122	112	Hrana pui	2400
3	123	113	Hrana capre	3200
4	124	114	Hrana cai	8450
5	125	115	Hrana porci	2170
6	126	116	Hrana rate	2870
7	127	117	Hrana oi	3160

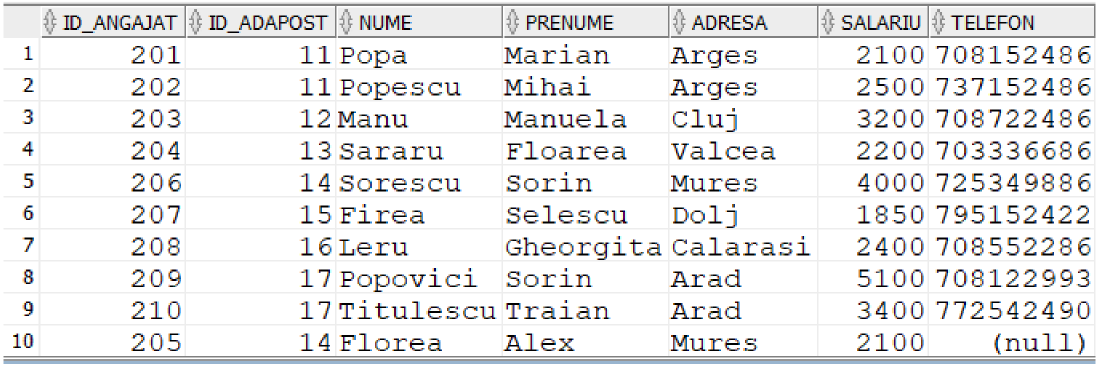
```
--TABELUL ADAPOST
insert into adapost(id_adapost, capacitate, judet)
values (11, 300, 'Arges');
insert into adapost(id_adapost, capacitate, judet)
values (12, 500, 'Cluj');
insert into adapost(id_adapost, capacitate, judet)
values (13, 450, 'Valcea');
insert into adapost(id_adapost, capacitate, judet)
values (14, 870, 'Mures');
insert into adapost(id_adapost, capacitate, judet)
values (15, 1000, 'Dolj');
insert into adapost(id_adapost, capacitate, judet)
values (16, 670, 'Calarasi');
insert into adapost(id_adapost, capacitate, judet)
```

```
values (17, 430, 'Arad');
select * from adapost;
```



	ID_ADAPOST	CAPACITATE	JUDET
1	11	300	Arges
2	12	500	Cluj
3	13	450	Valcea
4	14	870	Mures
5	15	1000	Dolj
6	16	670	Calarasi
7	17	430	Arad

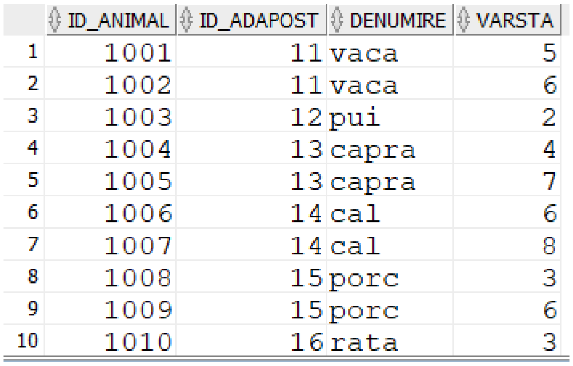
```
--TABELUL ANGAJAT
insert into angajat(id_angajat, id_adapost, nume, prenume, adresa, salariu, telefon)
values (201, 11, 'Popa', 'Marian', 'Arges', 2100, 0708152486);
insert into angajat(id_angajat, id_adapost, nume, prenume, adresa, salariu, telefon)
values (202, 11, 'Popescu', 'Mihai', 'Arges', 2500, 0737152486);
insert into angajat(id_angajat, id_adapost, nume, prenume, adresa, salariu, telefon)
values (203, 12, 'Manu', 'Manuela', 'Cluj', 3200, 0708722486);
insert into angajat(id_angajat, id_adapost, nume, prenume, adresa, salariu, telefon)
values (204, 13, 'Sararu', 'Floarea', 'Valcea', 2200, 0703336686);
insert into angajat(id_angajat, id_adapost, nume, prenume, adresa, salariu, telefon)
values (205, 14, 'Florea', 'Alex', 'Mures', 2100, NULL);
insert into angajat(id_angajat, id_adapost, nume, prenume, adresa, salariu, telefon)
values (206, 14, 'Sorescu', 'Sorin', 'Mures', 4000, 0725349886);
insert into angajat(id_angajat, id_adapost, nume, prenume, adresa, salariu, telefon)
values (207, 15, 'Firea', 'Selescu', 'Dolj', 1850, 0795152422);
insert into angajat(id_angajat, id_adapost, nume, prenume, adresa, salariu, telefon)
values (208, 16, 'Leru', 'Gheorgita', 'Calarasi', 2400, 0708552286);
insert into angajat(id_angajat, id_adapost, nume, prenume, adresa, salariu, telefon)
values (209, 17, 'Popovici', 'Sorin', 'Arad', 5100, 0708122993);
insert into angajat(id_angajat, id_adapost, nume, prenume, adresa, salariu, telefon)
values (210, 17, 'Titulescu', 'Traian', 'Arad', 3400, 0772542490);
select * from angajat;
```



	ID_ANGAJAT	ID_ADAPOST	NUME	PRENUME	ADRESA	SALARIU	TELEFON
1	201	11	Popa	Marian	Arges	2100	708152486
2	202	11	Popescu	Mihai	Arges	2500	737152486
3	203	12	Manu	Manuela	Cluj	3200	708722486
4	204	13	Sararu	Floarea	Valcea	2200	703336686
5	206	14	Sorescu	Sorin	Mures	4000	725349886
6	207	15	Firea	Selescu	Dolj	1850	795152422
7	208	16	Leru	Gheorgita	Calarasi	2400	708552286
8	209	17	Popovici	Sorin	Arad	5100	708122993
9	210	17	Titulescu	Traian	Arad	3400	772542490
10	205	14	Florea	Alex	Mures	2100	(null)

```
--TABELUL ANIMAL
insert into animal(id_animal, id_adapost, denumire, varsta)
values (1001, 11, 'vaca', 5);
insert into animal(id_animal, id_adapost, denumire, varsta)
```

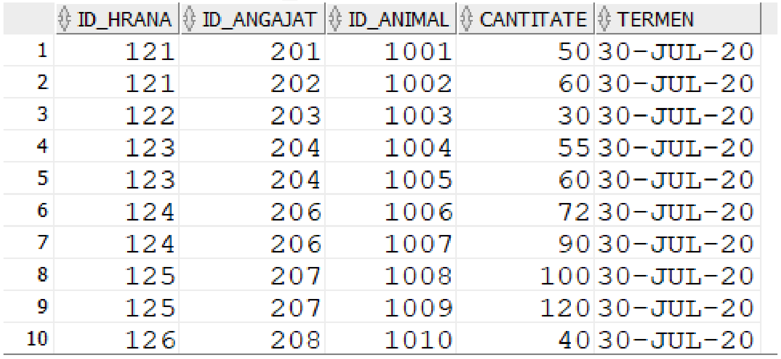
```
values (1002, 11, 'vaca', 6);
insert into animal(id_animal, id_adapost, denumire, varsta)
values (1003, 12, 'pui', 2);
insert into animal(id_animal, id_adapost, denumire, varsta)
values (1004, 13, 'capra', 4);
insert into animal(id_animal, id_adapost, denumire, varsta)
values (1005, 13, 'capra', 7);
insert into animal(id_animal, id_adapost, denumire, varsta)
values (1006, 14, 'cal', 6);
insert into animal(id_animal, id_adapost, denumire, varsta)
values (1007, 14, 'cal', 8);
insert into animal(id_animal, id_adapost, denumire, varsta)
values (1008, 15, 'porc', 3);
insert into animal(id_animal, id_adapost, denumire, varsta)
values (1009, 15, 'porc', 6);
insert into animal(id_animal, id_adapost, denumire, varsta)
values (1010, 16, 'rata', 3);
insert into animal(id_animal, id_adapost, denumire, varsta)
values (1011, 17, 'oaie', 5);
insert into animal(id_animal, id_adapost, denumire, varsta)
values (1012, 17, 'oaie', 7);
select * from animal;
```



	ID_ANIMAL	ID_ADAPOST	DENUMIRE	VARSTA
1	1001	11	vaca	5
2	1002	11	vaca	6
3	1003	12	pui	2
4	1004	13	capra	4
5	1005	13	capra	7
6	1006	14	cal	6
7	1007	14	cal	8
8	1008	15	porc	3
9	1009	15	porc	6
10	1010	16	rata	3

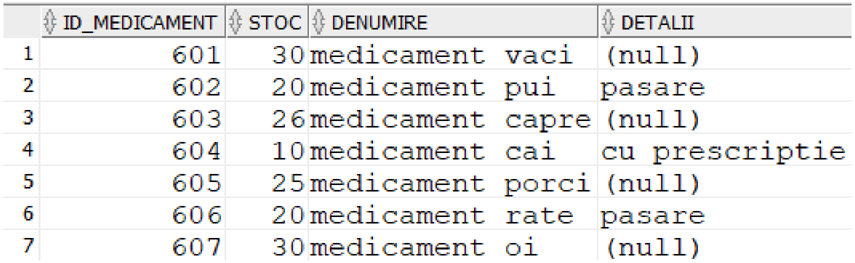
```
--TABELUL NECESAR_HRANA
insert into necesar_hrana (id_hrana, id_angajat, id_animal, cantitate, termen)
values (121, 201, 1001, 50, to_date('20200730','YYYYMMDD'));
insert into necesar_hrana (id_hrana, id_angajat, id_animal, cantitate, termen)
values (121, 202, 1002, 60, to_date('20200730','YYYYMMDD'));
insert into necesar_hrana (id_hrana, id_angajat, id_animal, cantitate, termen)
values (122, 203, 1003, 30, to_date('20200730','YYYYMMDD'));
insert into necesar_hrana (id_hrana, id_angajat, id_animal, cantitate, termen)
values (123, 204, 1004, 55, to_date('20200730','YYYYMMDD'));
insert into necesar_hrana (id_hrana, id_angajat, id_animal, cantitate, termen)
values (123, 204, 1005, 60, to_date('20200730','YYYYMMDD'));
insert into necesar_hrana (id_hrana, id_angajat, id_animal, cantitate, termen)
values (124, 206, 1006, 72, to_date('20200730','YYYYMMDD'));
```

```
insert into necesar_hrana (id_hrana, id_angajat, id_animal, cantitate, termen)
values (124, 206, 1007, 90, to_date('20200730','YYYYMMDD'));
insert into necesar_hrana (id_hrana, id_angajat, id_animal, cantitate, termen)
values (125, 207, 1008, 100, to_date('20200730','YYYYMMDD'));
insert into necesar_hrana (id_hrana, id_angajat, id_animal, cantitate, termen)
values (125, 207, 1009, 120, to_date('20200730','YYYYMMDD'));
insert into necesar_hrana (id_hrana, id_angajat, id_animal, cantitate, termen)
values (126, 208, 1010, 40, to_date('20200730','YYYYMMDD'));
insert into necesar_hrana (id_hrana, id_angajat, id_animal, cantitate, termen)
values (127, 209, 1011, 83, to_date('20200730','YYYYMMDD'));
insert into necesar_hrana (id_hrana, id_angajat, id_animal, cantitate, termen)
values (127, 210, 1012, 80, to_date('20200730','YYYYMMDD'));
select * from necesar_hrana;
```



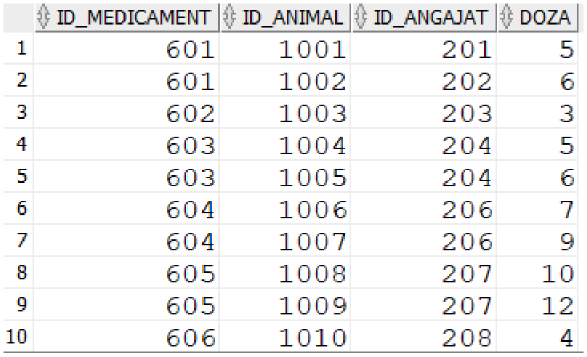
	ID_HRANA	ID_ANGAJAT	ID_ANIMAL	CANTITATE	TERMEN
1	121	201	1001	50	30-JUL-20
2	121	202	1002	60	30-JUL-20
3	122	203	1003	30	30-JUL-20
4	123	204	1004	55	30-JUL-20
5	123	204	1005	60	30-JUL-20
6	124	206	1006	72	30-JUL-20
7	124	206	1007	90	30-JUL-20
8	125	207	1008	100	30-JUL-20
9	125	207	1009	120	30-JUL-20
10	126	208	1010	40	30-JUL-20

```
--TABELUL MEDICAMENT
insert into medicament (id_medicament, stoc, denumire, detalii)
values (601, 30, 'medicament vaci', NULL);
insert into medicament (id_medicament, stoc, denumire, detalii)
values (602, 20, 'medicament pui', 'pasare');
insert into medicament (id_medicament, stoc, denumire, detalii)
values (603, 26, 'medicament capre', NULL);
insert into medicament (id_medicament, stoc, denumire, detalii)
values (604, 10, 'medicament cai', 'cu prescriptie');
insert into medicament (id_medicament, stoc, denumire, detalii)
values (605, 25, 'medicament porci', NULL);
insert into medicament (id_medicament, stoc, denumire, detalii)
values (606, 20, 'medicament rate', 'pasare');
insert into medicament (id_medicament, stoc, denumire, detalii)
values (607, 30, 'medicament oi', NULL);
select * from medicament;
```



	ID_MEDICAMENT	STOC	DENUMIRE	DETALII
1	601	30	medicament vaci	(null)
2	602	20	medicament pui	pasare
3	603	26	medicament capre	(null)
4	604	10	medicament cai	cu prescriptie
5	605	25	medicament porci	(null)
6	606	20	medicament rate	pasare
7	607	30	medicament oi	(null)

```
--TABELUL MEDICATIE
insert into medicatie (id_medicament,id_angajat, id_animal, doza)
values (601, 201, 1001, 5);
insert into medicatie (id_medicament, id_angajat, id_animal, doza)
values (601, 202, 1002, 6);
insert into medicatie (id_medicament, id_angajat, id_animal, doza)
values (602, 203, 1003, 3);
insert into medicatie (id_medicament, id_angajat, id_animal, doza)
values (603, 204, 1004, 5);
insert into medicatie (id_medicament, id_angajat, id_animal, doza)
values (603, 204, 1005, 6);
insert into medicatie (id_medicament, id_angajat, id_animal, doza)
values (604, 206, 1006, 7);
insert into medicatie (id_medicament, id_angajat, id_animal, doza)
values (604, 206, 1007, 9);
insert into medicatie (id_medicament, id_angajat, id_animal, doza)
values (605, 207, 1008, 10);
insert into medicatie (id_medicament, id_angajat, id_animal, doza)
values (605, 207, 1009, 12);
insert into medicatie (id_medicament, id_angajat, id_animal, doza)
values (606, 208, 1010, 4);
insert into medicatie (id_medicament, id_angajat, id_animal, doza)
values (607, 209, 1011, 8);
insert into medicatie (id_medicament, id_angajat, id_animal, doza)
values (607, 210, 1012, 8);
select * from medicatie;
```

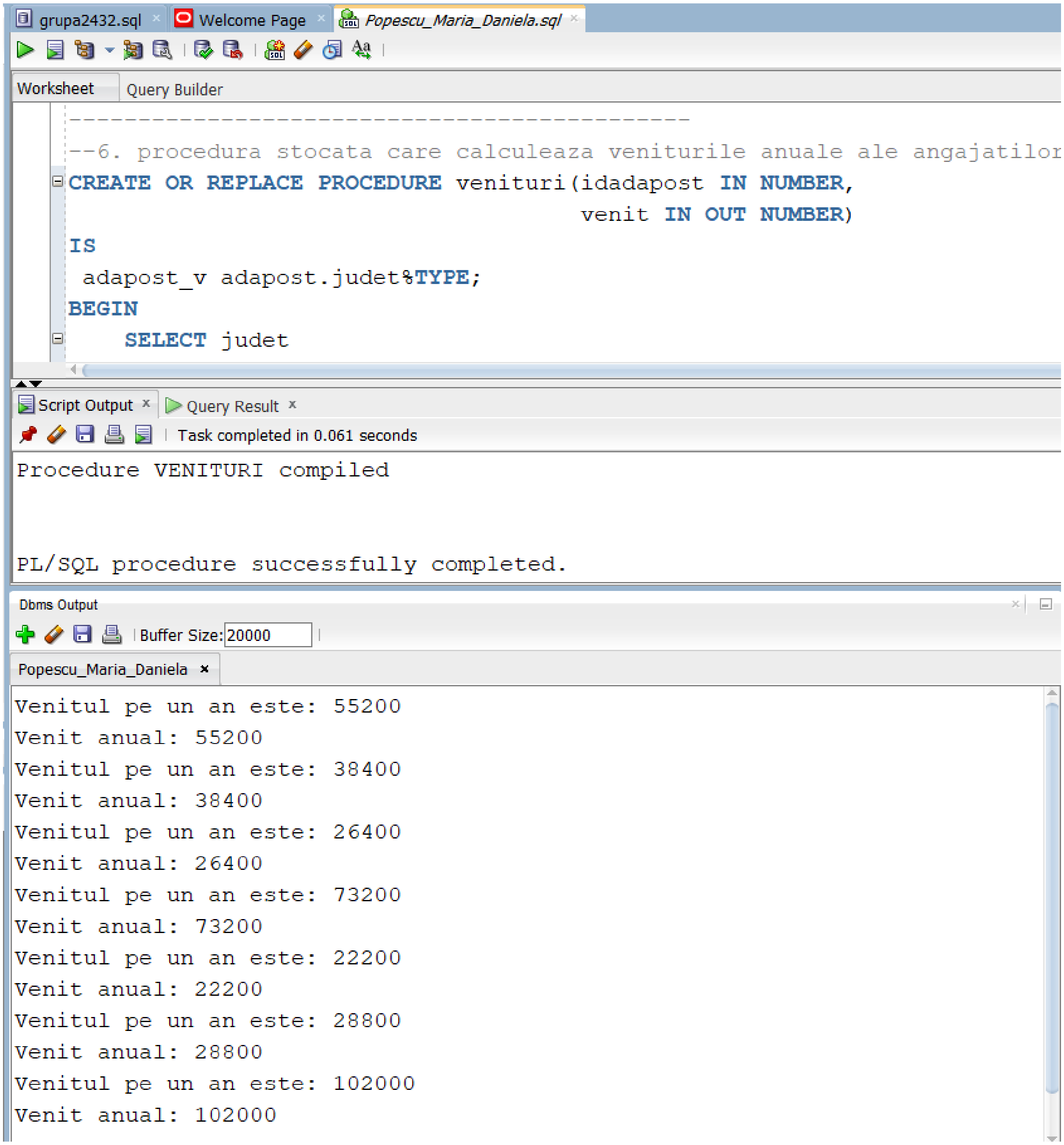


	ID_MEDICAMENT	ID_ANIMAL	ID_ANGAJAT	DOZA
1	601	1001	201	5
2	601	1002	202	6
3	602	1003	203	3
4	603	1004	204	5
5	603	1005	204	6
6	604	1006	206	7
7	604	1007	206	9
8	605	1008	207	10
9	605	1009	207	12
10	606	1010	208	4

6. procedura stocata care calculeaza veniturile anuale ale angajatilor care lucreaza intr-un anume adapost

```
CREATE OR REPLACE PROCEDURE venituri(idadapost IN NUMBER,
                            venit IN OUT NUMBER)
IS
 adapost_v adapost.judet%TYPE;
BEGIN
   SELECT judet
   INTO adapost_v
   FROM adapost
   WHERE id_adapost = idadapost;

   SELECT SUM(salariu * 12)
   INTO venit
   FROM angajat
   WHERE id_adapost = idadapost;
   DBMS_OUTPUT.PUT_LINE('Venitul pe un an este: ' || venit);
EXCEPTION
   WHEN NO_DATA_FOUND THEN
   DBMS_OUTPUT.PUT_LINE('Nu s-a gasit nicio inregistrare');
END venituri;
/

set verify OFF
set serveroutput ON
DECLARE
   venit NUMBER := 0;
BEGIN
   FOR rand IN (SELECT DISTINCT id_adapost
             FROM adapost
             ORDER BY id_adapost) LOOP
      venituri(rand.id_adapost, venit);
      DBMS_OUTPUT.PUT_LINE('Venit anual: '|| venit);
   END LOOP;
END;
```

/



```
----------------------------------------
--6. procedura stocata care calculeaza veniturile anuale ale angajatilor
CREATE OR REPLACE PROCEDURE venituri(idadapost IN NUMBER,
                                     venit IN OUT NUMBER)
IS
 adapost_v adapost.judet%TYPE;
BEGIN
    SELECT judet
```

```
Procedure VENITURI compiled


PL/SQL procedure successfully completed.
```

```
Venitul pe un an este: 55200
Venit anual: 55200
Venitul pe un an este: 38400
Venit anual: 38400
Venitul pe un an este: 26400
Venit anual: 26400
Venitul pe un an este: 73200
Venit anual: 73200
Venitul pe un an este: 22200
Venit anual: 22200
Venitul pe un an este: 28800
Venit anual: 28800
Venitul pe un an este: 102000
Venit anual: 102000
```

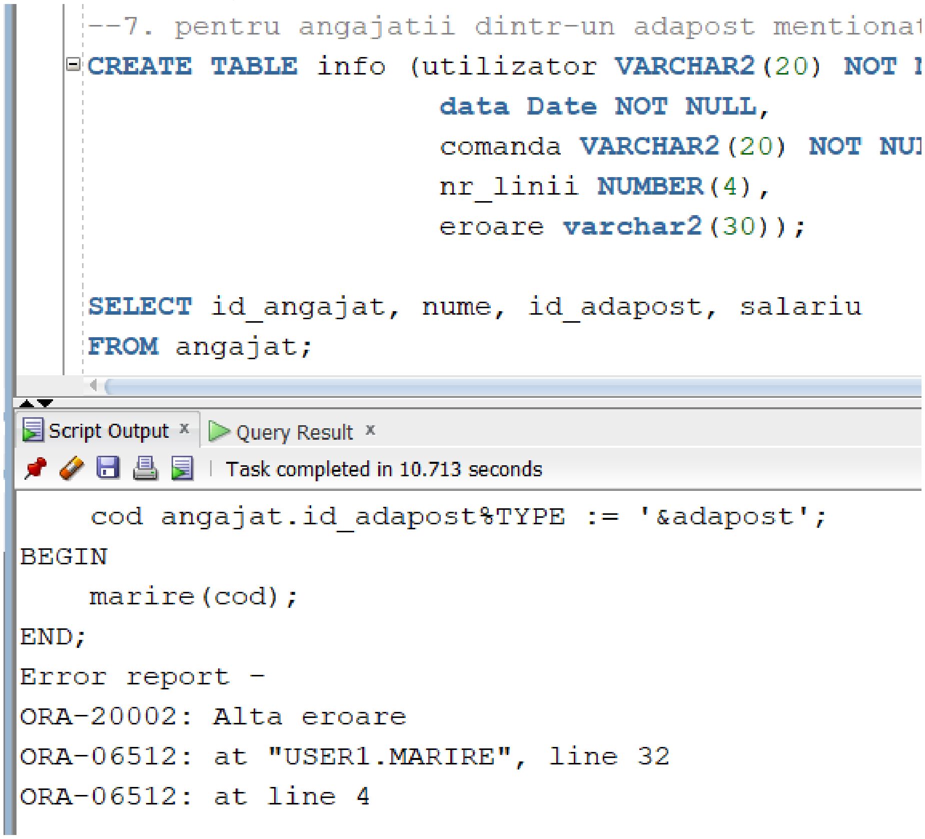
7. pentru angajatii dintr-un adapost mentionat, se scad salariile cu 10%

```
CREATE TABLE info (utilizator VARCHAR2(20) NOT NULL,
            data Date NOT NULL,
            comanda VARCHAR2(20) NOT NULL,
            nr_linii NUMBER(4),
            eroare varchar2(30));
```

```
SELECT id_angajat, nume, id_adapost, salariu
FROM angajat;

CREATE OR REPLACE
   PROCEDURE marire (cod_adapost angajat.id_adapost%TYPE)
IS
   check_error NUMBER;
   CURSOR angajati(cod angajat.id_angajat%TYPE) IS
      SELECT *
      FROM angajat
      START WITH id_adapost = cod
      FOR UPDATE NOWAIT;
BEGIN
   select COUNT(*)
   INTO check_error
   FROM(
         SELECT id_angajat
         FROM angajat
         WHERE id_adapost = cod_adapost
      );
   IF check_error = 0 THEN
      INSERT INTO info VALUES(user, SYSDATE, $$plsql_unit, 0, 'NICIUN_ADAPOST');
      COMMIT;
      RAISE_APPLICATION_ERROR(-20000,'Nu exista adapost cu id-ul acesta');
   END IF;
   FOR angajat_v IN angajati(cod_adapost) LOOP
      UPDATE angajat
      SET salariu = salariu*0.9
      WHERE CURRENT OF angajati;
   END LOOP;
   INSERT INTO info VALUES(user, SYSDATE, $$plsql_unit, CHECK_ERROR, NULL);
EXCEPTION
   WHEN TOO_MANY_ROWS THEN
      RAISE_APPLICATION_ERROR(-20001, 'Exista mai multe adaposturi cu acest id');
   WHEN OTHERS THEN
      RAISE_APPLICATION_ERROR(-20002,'Alta eroare');
END marire;
/

DECLARE
   cod angajat.id_adapost%TYPE := '&adapost';
BEGIN
   marire(cod);
END;
```

/

select * from info;



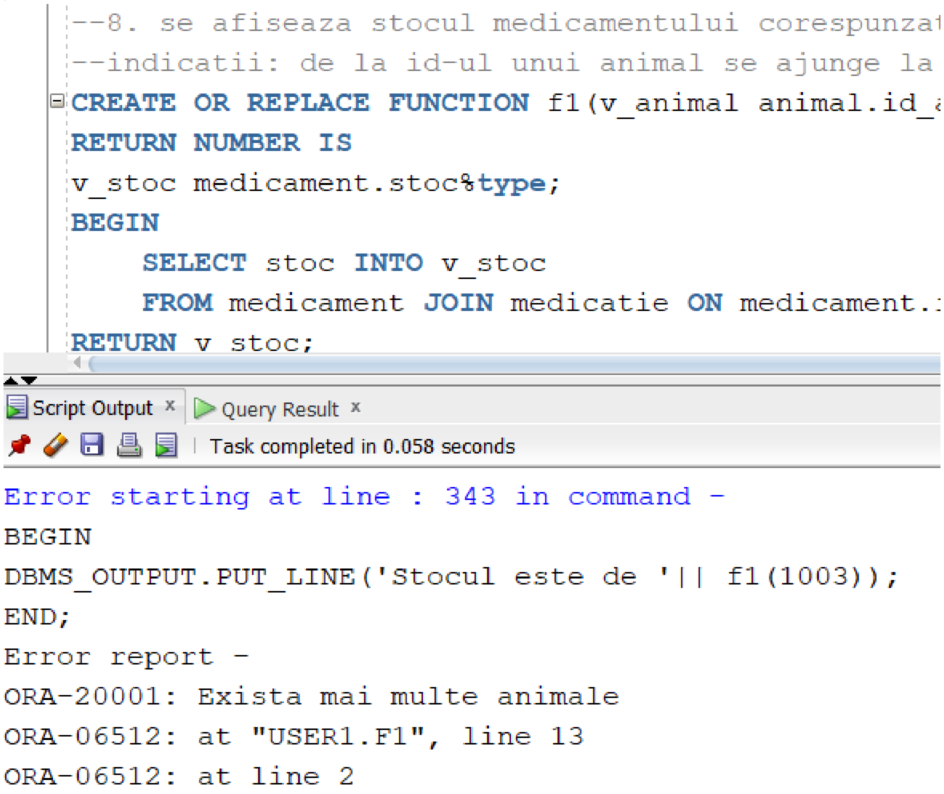
```
--7. pentru angajatii dintr-un adapost mentiona
CREATE TABLE info (utilizator VARCHAR2(20) NOT N
                   data Date NOT NULL,
                   comanda VARCHAR2(20) NOT NU
                   nr_linii NUMBER(4),
                   eroare varchar2(30));

SELECT id_angajat, nume, id_adapost, salariu
FROM angajat;
```

Task completed in 10.713 seconds

```
    cod angajat.id_adapost%TYPE := '&adapost';
BEGIN
    marire(cod);
END;
Error report -
ORA-20002: Alta eroare
ORA-06512: at "USER1.MARIRE", line 32
ORA-06512: at line 4
```

8. se afiseaza stocul medicamentului corespunzator unui animal
--indicatii: de la id-ul unui animal se ajunge la stocul medicamentului pe care il primeste animalul, prin tabelul medicatie care contine id-ul ambelor parti

```
CREATE OR REPLACE FUNCTION f1(v_animal animal.id_animal%TYPE)
RETURN NUMBER IS
v_stoc medicament.stoc%type;
BEGIN
   SELECT stoc INTO v_stoc
   FROM medicament JOIN medicatie ON
medicament.id_medicament=medicatie.id_medicament JOIN animal ON
medicatie.id_animal=animal.id_animal;
RETURN v_stoc;
EXCEPTION
   WHEN NO_DATA_FOUND THEN
   RAISE_APPLICATION_ERROR(-20000,
   'Nu exista medicamentul');
   WHEN TOO_MANY_ROWS THEN
   RAISE_APPLICATION_ERROR(-20001,
    'Exista mai multe animale');
    WHEN OTHERS THEN
    RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
END f1;
/

set verify OFF
set serveroutput ON
BEGIN
DBMS_OUTPUT.PUT_LINE('Stocul este de '|| f1(1003));
END;
/
```



```
--8. se afiseaza stocul medicamentului corespunzat
--indicatii: de la id-ul unui animal se ajunge la
CREATE OR REPLACE FUNCTION f1(v_animal animal.id_
RETURN NUMBER IS
v_stoc medicament.stoc%type;
BEGIN
    SELECT stoc INTO v_stoc
    FROM medicament JOIN medicatie ON medicament.
RETURN v_stoc;
```

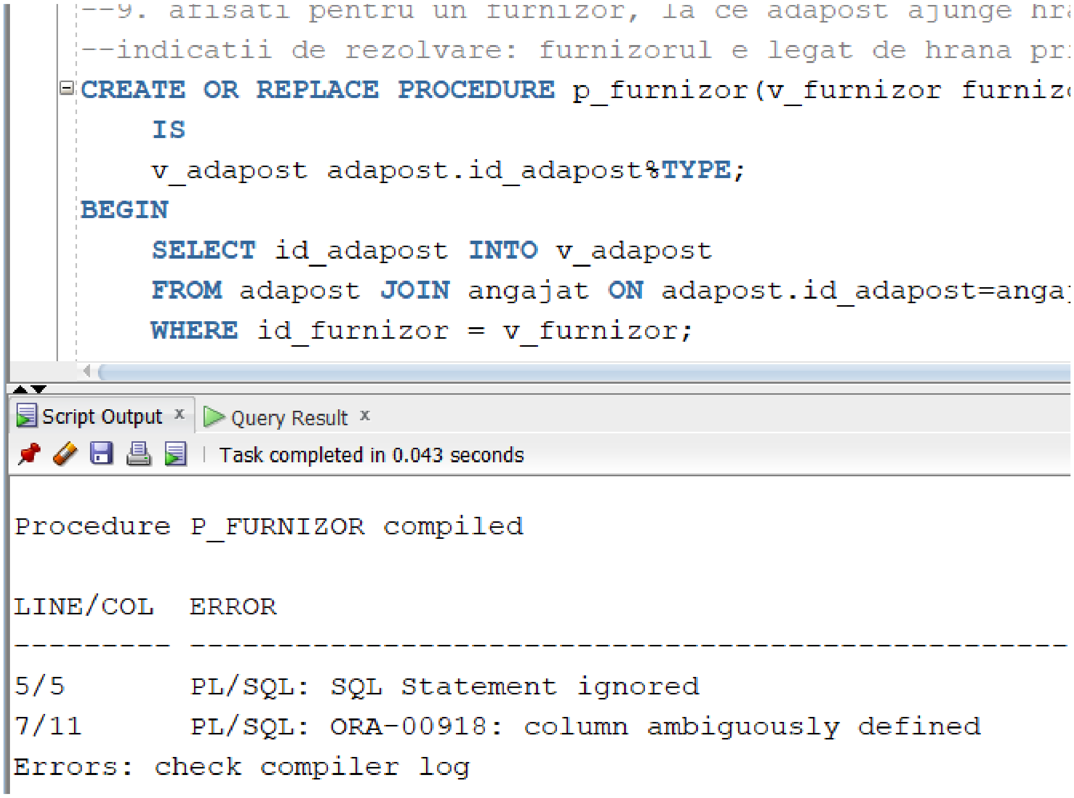
```
Error starting at line : 343 in command -
BEGIN
DBMS_OUTPUT.PUT_LINE('Stocul este de '|| f1(1003));
END;
Error report -
ORA-20001: Exista mai multe animale
ORA-06512: at "USER1.F1", line 13
ORA-06512: at line 2
```

9. afisati pentru un furnizor, la ce adapost ajunge hrana lui

--indicatii de rezolvare: furnizorul e legat de hrana prin id_furnizor, mai departe hrana e legata de angajat prin tabelul necesar_hrana, in final angajat e legat de adapost prin id_adapost

```
CREATE OR REPLACE PROCEDURE p_furnizor(v_furnizor furnizor.id_furnizor%TYPE)
    IS
    v_adapost adapost.id_adapost%TYPE;
BEGIN
    SELECT id_adapost INTO v_adapost
    FROM adapost JOIN angajat ON adapost.id_adapost=angajat.id_adapost JOIN necesar_hrana
ON angajat.id_angajat=necesar_hrana.id_angajat JOIN hrana ON
```

```
necesar_hrana.id_hrana=hrana.id_hrana JOIN furnizor ON
hrana.id_furnizor=furnizor.id_furnizor
   WHERE id_furnizor = v_furnizor;
   DBMS_OUTPUT.PUT_LINE('Adapostul este '|| v_adapost);
EXCEPTION
   WHEN NO_DATA_FOUND THEN
   RAISE_APPLICATION_ERROR(-20000,
   'Nu exista furnizorul dat');
   WHEN TOO_MANY_ROWS THEN
   RAISE_APPLICATION_ERROR(-20001,
   'Exista mai multi furnizori');
   WHEN OTHERS THEN
   RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
END p_furnizor;
/

set verify OFF
set serveroutput ON
BEGIN
   p_furnizor(111);
END;
/
```

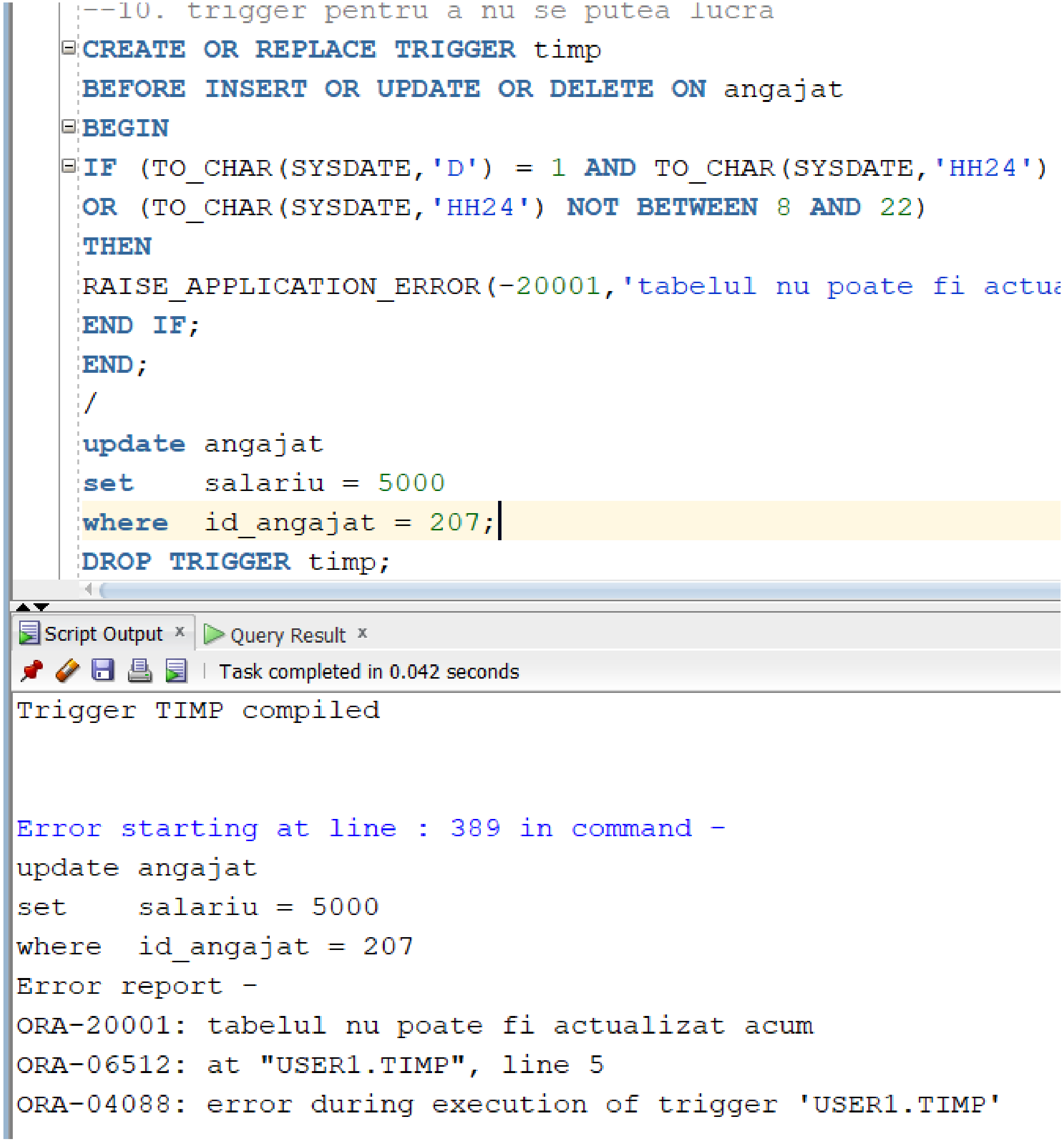


```
--9. afisati pentru un furnizor, la ce adapost ajunge hra
--indicatii de rezolvare: furnizorul e legat de hrana pr
CREATE OR REPLACE PROCEDURE p_furnizor(v_furnizor furniz
    IS
    v_adapost adapost.id_adapost%TYPE;
BEGIN
    SELECT id_adapost INTO v_adapost
    FROM adapost JOIN angajat ON adapost.id_adapost=anga
    WHERE id_furnizor = v_furnizor;
```

```
Procedure P_FURNIZOR compiled

LINE/COL  ERROR
--------- -------------------------------------------------------------
5/5       PL/SQL: SQL Statement ignored
7/11      PL/SQL: ORA-00918: column ambiguously defined
Errors: check compiler log
```

10. trigger pentru a nu se putea lucra

```
CREATE OR REPLACE TRIGGER timp
BEFORE INSERT OR UPDATE OR DELETE ON angajat
BEGIN
IF (TO_CHAR(SYSDATE,'D') = 1 AND TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 12 AND 18)
OR (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 8 AND 22)
THEN
RAISE_APPLICATION_ERROR(-20001,'tabelul nu poate fi actualizat acum');
END IF;
END;
/
DROP TRIGGER timp;
```



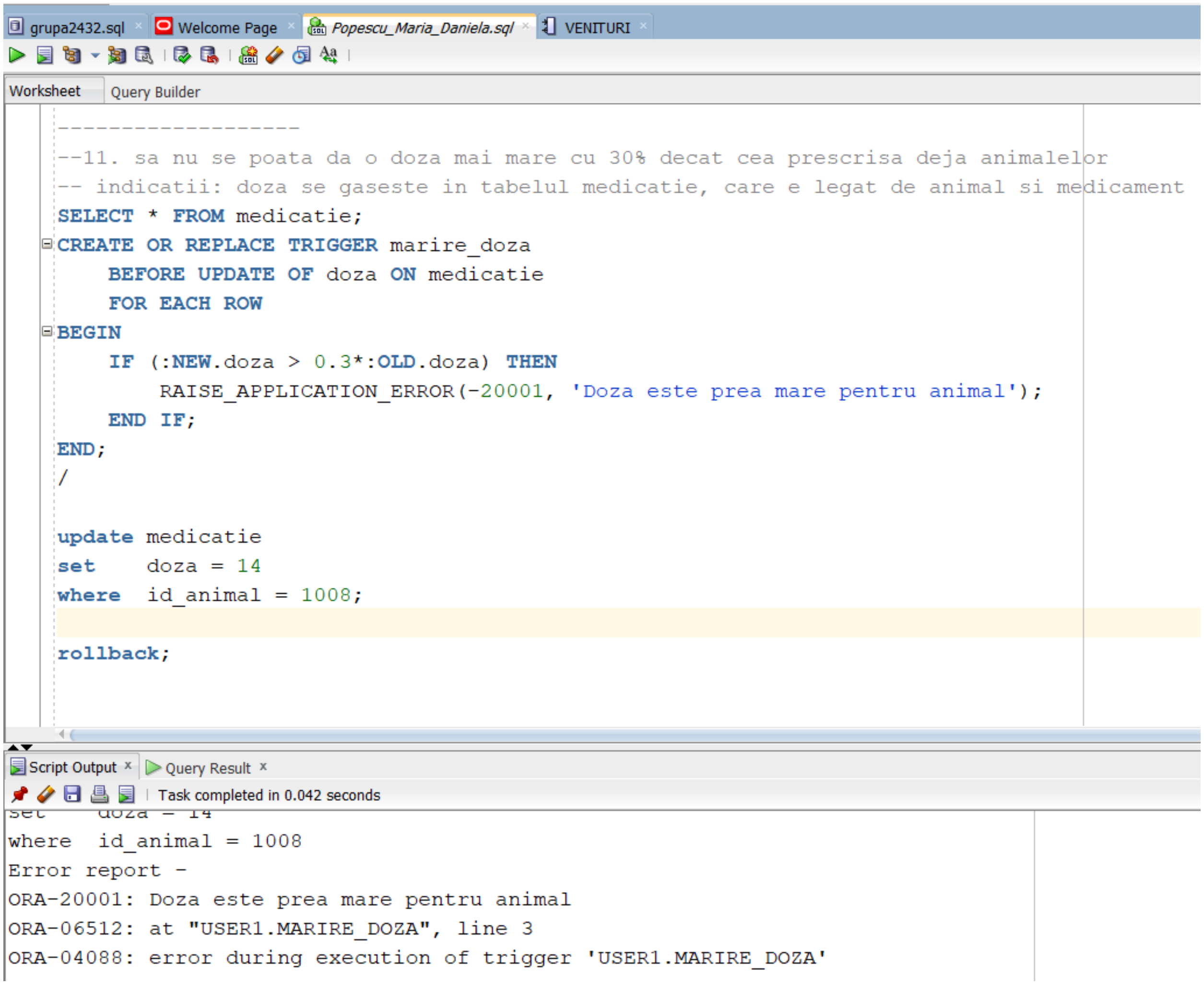
```
--10. trigger pentru a nu se putea lucra
CREATE OR REPLACE TRIGGER timp
BEFORE INSERT OR UPDATE OR DELETE ON angajat
BEGIN
IF  (TO_CHAR(SYSDATE,'D')  = 1  AND  TO_CHAR(SYSDATE,'HH24')
OR  (TO_CHAR(SYSDATE,'HH24')  NOT BETWEEN  8  AND  22)
THEN
RAISE_APPLICATION_ERROR(-20001,'tabelul nu poate fi actua
END IF;
END;
/
update angajat
set    salariu = 5000
where  id_angajat = 207;
DROP TRIGGER timp;
```

Task completed in 0.042 seconds

```
Trigger TIMP compiled


Error starting at line : 389 in command -
update angajat
set    salariu = 5000
where  id_angajat = 207
Error report -
ORA-20001: tabelul nu poate fi actualizat acum
ORA-06512: at "USER1.TIMP", line 5
ORA-04088: error during execution of trigger 'USER1.TIMP'
```

11. trigger pentru a nu se putea da o doza mai mare cu 30% decat cea prescrisa deja animalelor
-- indicatii: doza se gaseste in tabelul medicatie, care e legat de animal si medicament

```
SELECT * FROM medicatie;
CREATE OR REPLACE TRIGGER marire_doza
   BEFORE UPDATE OF doza ON medicatie
   FOR EACH ROW
BEGIN
   IF (:NEW.doza > 0.3*:OLD.doza) THEN
      RAISE_APPLICATION_ERROR(-20001, 'Doza este prea mare pentru animal');
   END IF;
END;
/

update medicatie
set    doza = 14
where  id_animal = 1008;

DROP TRIGGER marire_doza;
```

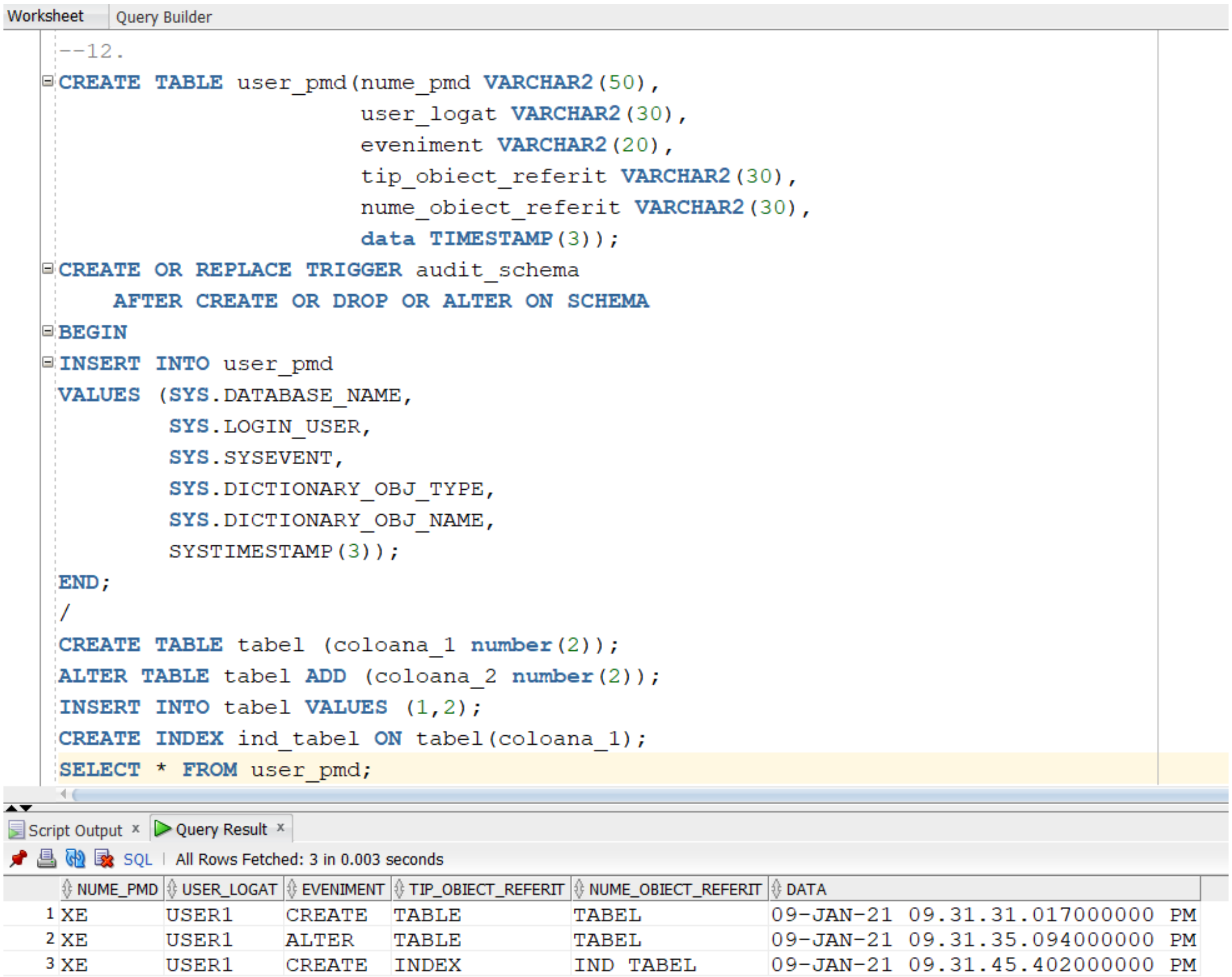


```
--------------------
--11. sa nu se poata da o doza mai mare cu 30% decat cea prescrisa deja animalelor
-- indicatii: doza se gaseste in tabelul medicatie, care e legat de animal si medicament
SELECT * FROM medicatie;
CREATE OR REPLACE TRIGGER marire_doza
    BEFORE UPDATE OF doza ON medicatie
    FOR EACH ROW
BEGIN
    IF (:NEW.doza > 0.3*:OLD.doza) THEN
        RAISE_APPLICATION_ERROR(-20001, 'Doza este prea mare pentru animal');
    END IF;
END;
/

update medicatie
set    doza = 14
where  id_animal = 1008;

rollback;
```

Task completed in 0.042 seconds

```
set    doza = 14
where  id_animal = 1008
Error report -
ORA-20001: Doza este prea mare pentru animal
ORA-06512: at "USER1.MARIRE_DOZA", line 3
ORA-04088: error during execution of trigger 'USER1.MARIRE_DOZA'
```

```
12. CREATE TABLE user_pmd(nume_pmd VARCHAR2(50),
                 user_logat VARCHAR2(30),
                 eveniment VARCHAR2(20),
                 tip_obiect_referit VARCHAR2(30),
                 nume_obiect_referit VARCHAR2(30),
                 data TIMESTAMP(3));
CREATE OR REPLACE TRIGGER audit_schema
   AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
INSERT INTO user_pmd
VALUES (SYS.DATABASE_NAME,
      SYS.LOGIN_USER,
      SYS.SYSEVENT,
      SYS.DICTIONARY_OBJ_TYPE,
      SYS.DICTIONARY_OBJ_NAME,
      SYSTIMESTAMP(3));
END;
/
CREATE TABLE tabel (coloana_1 number(2));
ALTER TABLE tabel ADD (coloana_2 number(2));
INSERT INTO tabel VALUES (1,2);
CREATE INDEX ind_tabel ON tabel(coloana_1);
SELECT * FROM user_pmd;
```



```
--12.
CREATE TABLE user_pmd(nume_pmd VARCHAR2(50),
                      user_logat VARCHAR2(30),
                      eveniment VARCHAR2(20),
                      tip_obiect_referit VARCHAR2(30),
                      nume_obiect_referit VARCHAR2(30),
                      data TIMESTAMP(3));
CREATE OR REPLACE TRIGGER audit_schema
    AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
INSERT INTO user_pmd
VALUES (SYS.DATABASE_NAME,
        SYS.LOGIN_USER,
        SYS.SYSEVENT,
        SYS.DICTIONARY_OBJ_TYPE,
        SYS.DICTIONARY_OBJ_NAME,
        SYSTIMESTAMP(3));
END;
/
CREATE TABLE tabel  (coloana_1 number(2));
ALTER TABLE tabel ADD  (coloana_2 number(2));
INSERT INTO tabel VALUES  (1,2);
CREATE INDEX ind_tabel ON tabel(coloana_1);
SELECT * FROM user_pmd;
```

All Rows Fetched: 3 in 0.003 seconds

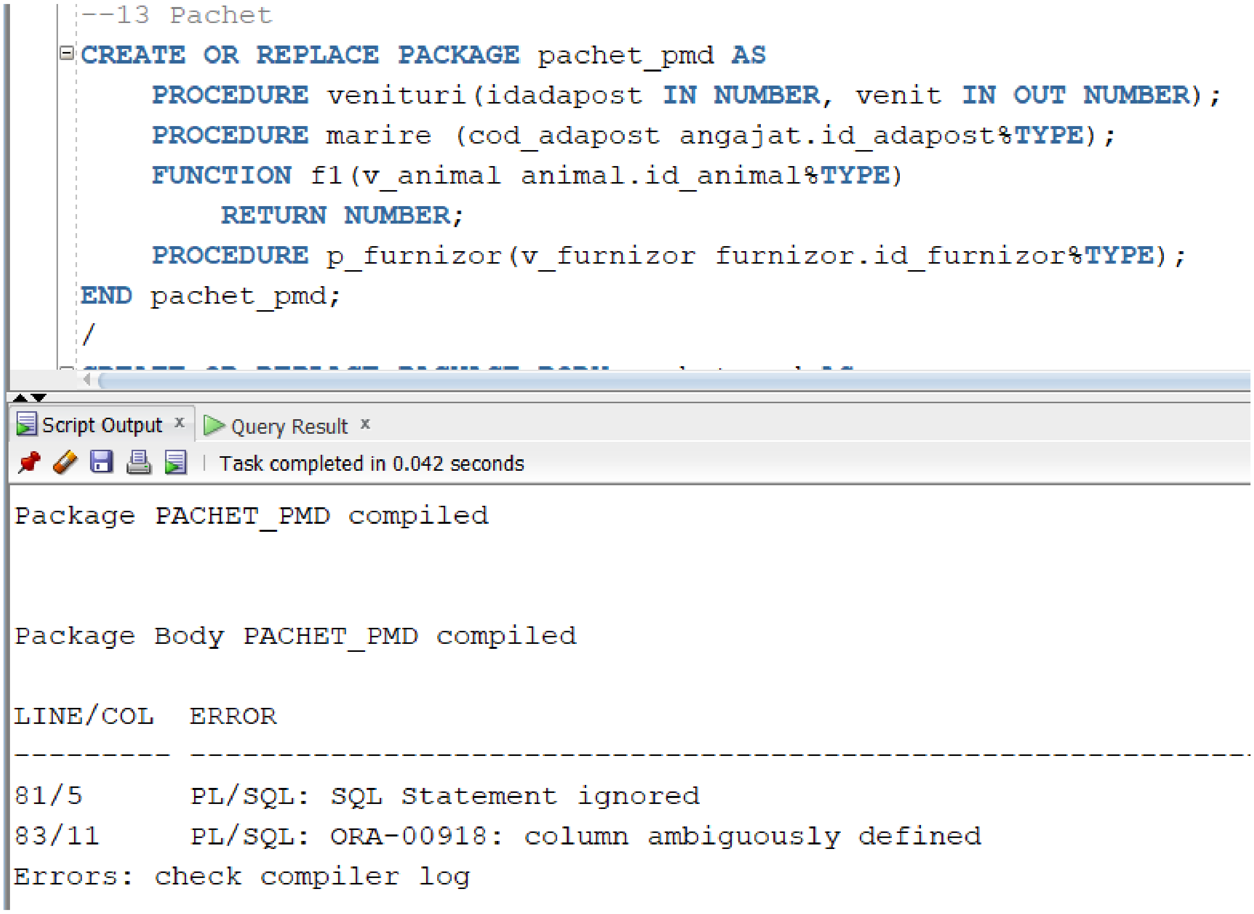
	NUME_PMD	USER_LOGAT	EVENIMENT	TIP_OBIECT_REFERIT	NUME_OBIECT_REFERIT	DATA
1	XE	USER1	CREATE	TABLE	TABEL	09-JAN-21 09.31.31.017000000 PM
2	XE	USER1	ALTER	TABLE	TABEL	09-JAN-21 09.31.35.094000000 PM
3	XE	USER1	CREATE	INDEX	IND_TABEL	09-JAN-21 09.31.45.402000000 PM

13. Pachet

```
CREATE OR REPLACE PACKAGE pachet_pmd AS
   PROCEDURE venituri(idadapost IN NUMBER, venit IN OUT NUMBER);
   PROCEDURE marire (cod_adapost angajat.id_adapost%TYPE);
   FUNCTION f1(v_animal animal.id_animal%TYPE)
      RETURN NUMBER;
   PROCEDURE p_furnizor(v_furnizor furnizor.id_furnizor%TYPE);
END pachet_pmd;
/
CREATE OR REPLACE PACKAGE BODY pachet_pmd AS
--6
PROCEDURE venituri(idadapost IN NUMBER,
                            venit IN OUT NUMBER)
IS
 adapost_v adapost.judet%TYPE;
BEGIN
   SELECT judet
   INTO adapost_v
   FROM adapost
   WHERE id_adapost = idadapost;

   SELECT SUM(salariu * 12)
   INTO venit
   FROM angajat
   WHERE id_adapost = idadapost;
   DBMS_OUTPUT.PUT_LINE('Venitul pe un an este: ' || venit);
EXCEPTION
   WHEN NO_DATA_FOUND THEN
   DBMS_OUTPUT.PUT_LINE('Nu s-a gasit nicio inregistrare');
END venituri;
--7
PROCEDURE marire (cod_adapost angajat.id_adapost%TYPE)
IS
   check_error NUMBER;
   CURSOR angajati(cod angajat.id_angajat%TYPE) IS
      SELECT *
      FROM angajat
      START WITH id_adapost = cod
      FOR UPDATE NOWAIT;
BEGIN
   select COUNT(*)
   INTO check_error
   FROM(
         SELECT id_angajat
         FROM angajat
         WHERE id_adapost = cod_adapost
      );
   IF check_error = 0 THEN
      INSERT INTO info VALUES(user, SYSDATE, $$plsql_unit, 0, 'NICIUN_ADAPOST');
      COMMIT;
      RAISE_APPLICATION_ERROR(-20000,'Nu exista adapost cu id-ul acesta');
   END IF;
   FOR angajat_v IN angajati(cod_adapost) LOOP
      UPDATE angajat
      SET salariu = salariu*0.9
      WHERE CURRENT OF angajati;
   END LOOP;
   INSERT INTO info VALUES(user, SYSDATE, $$plsql_unit, CHECK_ERROR, NULL);
EXCEPTION
   WHEN TOO_MANY_ROWS THEN
      RAISE_APPLICATION_ERROR(-20001, 'Exista mai multe adaposturi cu acest id');
   WHEN OTHERS THEN
```

```
      RAISE_APPLICATION_ERROR(-20002,'Alta eroare');
END marire;
--8.
FUNCTION f1(v_animal animal.id_animal%TYPE)
RETURN NUMBER IS
v_stoc medicament.stoc%type;
BEGIN
   SELECT stoc INTO v_stoc
   FROM medicament JOIN medicatie ON
medicament.id_medicament=medicatie.id_medicament JOIN animal ON
medicatie.id_animal=animal.id_animal;
RETURN v_stoc;
EXCEPTION
   WHEN NO_DATA_FOUND THEN
   RAISE_APPLICATION_ERROR(-20000,
   'Nu exista medicamentul');
   WHEN TOO_MANY_ROWS THEN
   RAISE_APPLICATION_ERROR(-20001,
   'Exista mai multe animale');
   WHEN OTHERS THEN
   RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
END f1;
--9
PROCEDURE p_furnizor(v_furnizor furnizor.id_furnizor%TYPE)
   IS
   v_adapost adapost.id_adapost%TYPE;
BEGIN
   SELECT id_adapost INTO v_adapost
   FROM adapost JOIN angajat ON adapost.id_adapost=angajat.id_adapost JOIN necesar_hrana
ON angajat.id_angajat=necesar_hrana.id_angajat JOIN hrana ON
necesar_hrana.id_hrana=hrana.id_hrana JOIN furnizor ON
hrana.id_furnizor=furnizor.id_furnizor
   WHERE id_furnizor = v_furnizor;
   DBMS_OUTPUT.PUT_LINE('Adapostul este '|| v_adapost);
EXCEPTION
   WHEN NO_DATA_FOUND THEN
   RAISE_APPLICATION_ERROR(-20000,
   'Nu exista furnizorul dat');
   WHEN TOO_MANY_ROWS THEN
   RAISE_APPLICATION_ERROR(-20001,
   'Exista mai multi furnizori');
   WHEN OTHERS THEN
   RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
END p_furnizor;
```

END pachet_pmd;
/



```
--13 Pachet
CREATE OR REPLACE PACKAGE pachet_pmd AS
    PROCEDURE venituri(idadapost IN NUMBER, venit IN OUT NUMBER);
    PROCEDURE marire (cod_adapost angajat.id_adapost%TYPE);
    FUNCTION f1(v_animal animal.id_animal%TYPE)
        RETURN NUMBER;
    PROCEDURE p_furnizor(v_furnizor furnizor.id_furnizor%TYPE);
END pachet_pmd;
/
```

Task completed in 0.042 seconds

```
Package PACHET_PMD compiled


Package Body PACHET_PMD compiled

LINE/COL  ERROR
--------- -------------------------------------------------------------
81/5      PL/SQL: SQL Statement ignored
83/11     PL/SQL: ORA-00918: column ambiguously defined
Errors: check compiler log
```