PROHECT CLINICĂ VIETTEIRINAIRĂ

Student: Staicu Octavian-Florin

Grupa 406 Master Inginerie Software

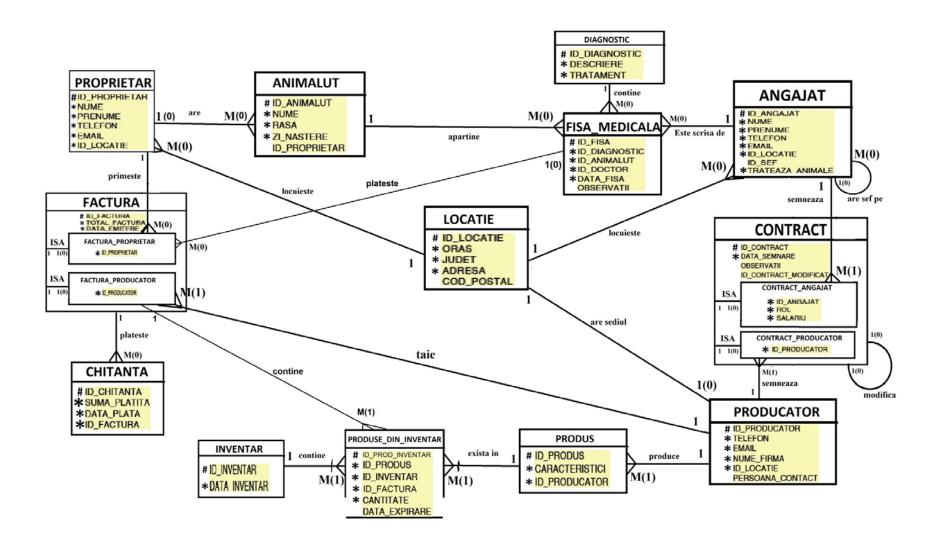
Clinică veterinară

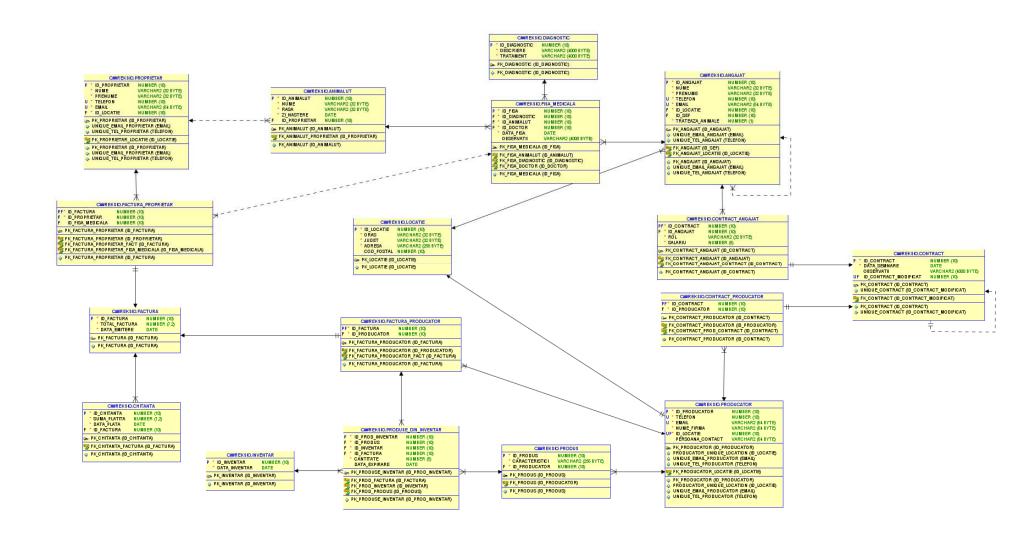
Acest proiect se adresează unei clinici veterinare

Descrierea Regulilor de Bussines

- Un proprietar are mai multe animale. Când un animăluț este tratat la clinică, i se face o nouă fișă medicală (relație de tip 3 între animal, diagnostic și un angajat care are voie să trateze animale). Pe fișa medicală apare un diagnostic, pentru care se știe tratamentul (sau se adaugă în tabela diagnostic). Un animăluț se poate găsi în baza de date fără să aibă o fișă medicală (spre exemplu, dacă doar s-a microcipat)
- Angajații au unul sau mai multe contracte de muncă (în caz că un angajat vrea să negocieze ulterior ore suplimentare sau măriri de salariu etc., se face un nou contract, care, eventual, poate modifica un contract anterior). Doar unii angajați au voie să trateze animale (precizat in atributul din tabela angajat ex. doar doctorii și asistentele etc.; întrucât au mai multe contracte, am ales să includ acest atribut pentru a evita o căutare prin toate contractele semnate pentru a verifica dacă există unul care să îi permită să trateze animăluțe).
- Clinica are mai multe produse: vaccinuri, medicamente, aparatură medicală, mâncare de animale, dezinfectanți etc. Produsele sunt produse de producători, care încheie contracte cu clinica. Produsele se vor găsi în inventarele care se fac o dată la o perioadă de timp îndelungată.
- Clinica emite facturi către proprietarii de animăluțe. Aceștia le pot achita eșalonat, deci există mai multe chitanțe pentru o singură factură.
- Producătorii de produse emit și ei facturi, pe care clinica trebuie să le plătească (la fel, eșalonat).

Pe următoarele două pagini se află diagrama entitate relație și diagrama conceptuală.





```
CREATE TABLE LOCATIE
CREATE TABLE PROPRIETAR
   CONSTRAINT pk_proprietar PRIMARY KEY (id_proprietar),
   CONSTRAINT fk proprietar locatie FOREIGN KEY (id locatie) REFERENCES LOCATIE (id locatie),
   CONSTRAINT unique_tel_proprietar UNIQUE (telefon),
   CONSTRAINT unique_email_proprietar UNIQUE (email)
CREATE TABLE ANIMALUT
```

```
CONSTRAINT pk_animalut PRIMARY KEY (id_animalut),
   CONSTRAINT fk_animalut_proprietar FOREIGN KEY (id_proprietar) REFERENCES PROPRIETAR (id_proprietar)
CREATE TABLE DIAGNOSTIC
   CONSTRAINT pk_diagnostic PRIMARY KEY (id_diagnostic)
CREATE TABLE ANGAJAT
   CONSTRAINT pk_angajat PRIMARY KEY (id_angajat),
   CONSTRAINT fk_angajat_locatie FOREIGN KEY (id_locatie) REFERENCES LOCATIE (id_locatie),
   CONSTRAINT fk_angajat FOREIGN KEY (id_sef) REFERENCES ANGAJAT (id_angajat),
   CONSTRAINT unique_tel_angajat UNIQUE (telefon),
   CONSTRAINT unique_email_angajat UNIQUE (email)
CREATE TABLE FISA_MEDICALA
```

```
CONSTRAINT pk fisa medicala PRIMARY KEY (id fisa),
   CONSTRAINT fk fisa diagnostic FOREIGN KEY (id diagnostic) REFERENCES DIAGNOSTIC (id diagnostic),
   CONSTRAINT fk_fisa_animalut FOREIGN KEY (id_animalut) REFERENCES ANIMALUT (id_animalut),
   CONSTRAINT fk_fisa_doctor FOREIGN KEY (id_doctor) REFERENCES ANGAJAT (id_angajat)
CREATE TABLE PRODUCATOR
   CONSTRAINT pk producator PRIMARY KEY (id producator),
   CONSTRAINT fk_producator_locatie FOREIGN KEY (id_locatie) REFERENCES LOCATIE (id_locatie),
   CONSTRAINT producator_unique_location UNIQUE (id_locatie),
   CONSTRAINT unique_tel_producator UNIQUE (telefon),
   CONSTRAINT unique_email_producator UNIQUE (email)
   CONSTRAINT unique contract UNIQUE (id contract modificat)
CREATE TABLE CONTRACT ANGAJAT
```

```
CONSTRAINT pk_contract_angajat PRIMARY KEY (id_contract),
   CONSTRAINT fk contract angajat contract FOREIGN KEY (id contract) REFERENCES CONTRACT (id contract),
   CONSTRAINT fk contract angajat FOREIGN KEY (id angajat) REFERENCES ANGAJAT (id angajat)
CREATE TABLE CONTRACT_PRODUCATOR
   CONSTRAINT pk_contract_producator PRIMARY KEY (id_contract),
   CONSTRAINT fk contract prod contract FOREIGN KEY (id contract) REFERENCES CONTRACT (id contract),
   CONSTRAINT fk contract producator FOREIGN KEY (id producator) REFERENCES PRODUCATOR (id producator)
CREATE TABLE FACTURA
   CONSTRAINT pk_factura PRIMARY KEY (id_factura)
CREATE TABLE FACTURA_PROPRIETAR
   CONSTRAINT pk factura proprietar PRIMARY KEY (id factura),
   CONSTRAINT fk_factura_proprietar FOREIGN KEY (id_proprietar) REFERENCES PROPRIETAR (id_proprietar),
   CONSTRAINT fk_factura_proprietar_fact FOREIGN KEY (id_factura) REFERENCES FACTURA (id_factura),
   CONSTRAINT fk factura proprietar fisa medicala FOREIGN KEY (id fisa medicala) REFERENCES FISA MEDICALA (id fisa)
CREATE TABLE FACTURA PRODUCATOR
   CONSTRAINT pk factura_producator PRIMARY KEY (id_factura),
   CONSTRAINT fk_factura_producator FOREIGN KEY (id_producator) REFERENCES PRODUCATOR (id_producator),
```

```
CONSTRAINT fk_factura_producator_fact FOREIGN KEY (id_factura) REFERENCES FACTURA (id_factura)
CREATE TABLE CHITANTA
   CONSTRAINT fk chitanta factura FOREIGN KEY (id factura) references FACTURA (id factura)
CREATE TABLE PRODUS
   CONSTRAINT pk_produs PRIMARY KEY (id_produs),
   CONSTRAINT fk produs FOREIGN KEY (id producator) REFERENCES PRODUCATOR (id producator)
CREATE TABLE INVENTAR
   CONSTRAINT pk_inventar PRIMARY KEY (id_inventar)
CREATE TABLE PRODUSE DIN INVENTAR
   CONSTRAINT pk_produse_inventar PRIMARY KEY (id_prod_inventar),
```

```
CONSTRAINT fk_prod_inventar FOREIGN KEY (id_inventar) REFERENCES INVENTAR (id_inventar),

CONSTRAINT fk_prod_produs FOREIGN KEY (id_produs) REFERENCES PRODUS (id_produs),

CONSTRAINT fk_prod_factura FOREIGN KEY (id_factura) REFERENCES FACTURA_PRODUCATOR (id_factura)
);

commit;
```

```
#REKSIO> CREATE TABLE LOCATIE
              id_locatie NUMBER(10),
              oras
             judet VARCHAR2(32) NOT NULL,
              adresa VARCHAR2(256) NOT NULL,
              cod_postal NUMBER(10),
              CONSTRAINT pk_locatie PRIMARY KEY (id_locatie)
[2023-02-06 05:36:21] completed in 26 ms
##REKSIO> CREATE TABLE PROPRIETAR
             id_proprietar NUMBER(10),
             prenume
              telefon NUMBER(10) NOT NULL,
              email
              id_locatie NUMBER(10) NOT NULL,
              CONSTRAINT pk_proprietar PRIMARY KEY (id_proprietar),
              CONSTRAINT fk_proprietar_locatie FOREIGN KEY (id_locatie) REFERENCES LOCATIE (id_locatie),
              CONSTRAINT unique_tel_proprietar UNIQUE (telefon),
              CONSTRAINT unique_email_proprietar UNIQUE (email)
[2023-02-06 05:36:27] completed in 31 ms
              id_animalut NUMBER(10),
             rasa
                           Varchar2(32) NOT NULL,
             zi_nastere DATE
             id_proprietar NUMBER(10),
             CONSTRAINT pk_animalut PRIMARY KEY (id_animalut),
              CONSTRAINT fk_animalut_proprietar FOREIGN KEY (id_proprietar) REFERENCES PROPRIETAR (id_proprietar)
[2023-02-06 05:36:33] completed in 25 ms
```

```
NSERT INTO LOCATIE(id locatie, oras, judet, adresa)
INSERT INTO LOCATIE(id_locatie, oras, judet, adresa, cod_postal)
INSERT INTO LOCATIE(id_locatie, oras, judet, adresa)
INSERT INTO LOCATIE(id locatie, oras, judet, adresa)
INSERT INTO PROPRIETAR(id proprietar, nume, prenume, email, telefon, id locatie)
VALUES (1, 'Staicu', 'Octavian-Florin', 'octavian.staicu@s.unibuc.ro', 0736659442, 1);
INSERT INTO PROPRIETAR(id_proprietar, nume, prenume, email, telefon, id_locatie)
:NSERT INTO PROPRIETAR(id_proprietar, nume, prenume, email, telefon, id_locatie)
VALUES (3, 'Staicu', 'Adrian', 'adi@qmail.com', 0736659441, 1);
INSERT INTO PROPRIETAR(id_proprietar, nume, prenume, email, telefon, id_locatie)
INSERT INTO PROPRIETAR(id_proprietar, nume, prenume, email, telefon, id_locatie)
INSERT INTO PROPRIETAR(id proprietar, nume, prenume, email, telefon, id locatie)
INSERT INTO ANIMALUT(id animalut, nume, rasa, zi nastere, id proprietar)
/ALUES (1, 'Bit', 'Boxer', to_date('05/05/2017', 'DD/MM/YYYY'), 1);
INSERT INTO ANIMALUT(id animalut, nume, rasa, zi nastere, id proprietar)
:NSERT INTO ANIMALUT(id_animalut, nume, rasa, zi_nastere, id_proprietar)
/ALUES (3, 'Kitty', 'Birmaneza', to date('25/12/2015', 'DD/MM/YYYY'), 2);
```

```
VALUES (4. 'Angelina', 'Pisica Tigrata de Hymalaia', to date('06/06/2016', 'DD/MM/YYYY'), 5);
INSERT INTO ANIMALUT(id animalut, nume, rasa, zi nastere, id proprietar)
/ALUES (5, 'John', 'Metis', to_date('05/12/2008', 'DD/MM/YYYY'), 4);
VALUES (6, 'dog', 'Boxer', to_date('09.01.2021', 'dd.mm.yyyy'), 5);
nsert into ANIMALUT(id animalut, nume, rasa, zi nastere, id proprietar)
VALUES (13, 'Thor', 'Golden Retriever', to_date('13.07.2020', 'dd.mm.yyyy'), 1);
nsert into ANIMALUT(id animalut, nume, rasa, zi nastere)
VALUES (11, 'Idk', 'Comuna', to_date('13.01.2023', 'dd.mm.yyyy'));
insert into ANIMALUT(id animalut, nume, rasa, zi nastere)
INSERT INTO DIAGNOSTIC(id_diagnostic, descriere, tratament)
INSERT INTO DIAGNOSTIC(id_diagnostic, descriere, tratament)
INSERT INTO DIAGNOSTIC(id diagnostic, descriere, tratament)
INSERT INTO DIAGNOSTIC(id_diagnostic, descriere, tratament)
INSERT INTO DIAGNOSTIC(id_diagnostic, descriere, tratament)
INSERT INTO ANGAJAT(id angajat, nume, prenume, telefon, email, id sef, id locatie, trateaza animale)
INSERT INTO ANGAJAT(id angajat, nume, prenume, telefon, email, id locatie, trateaza animale)
INSERT INTO FISA MEDICALA(id fisa, id diagnostic, id animalut, id doctor, data fisa, observatii)
VALUES (1, 2, 1, 1, to date('13.07.2020', 'dd.mm.yyyy'), 'Stapanul a dat caineul sa manance tort');
INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
VALUES (2, 1, 2, 1, to_date('13.07.2011', 'dd.mm.yyyy'));
```

```
INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
VALUES (3, 1, 3, 2, to date('13.07.2016', 'dd.mm.yyyy'));
INSERT INTO FISA MEDICALA(id fisa, id diagnostic, id animalut, id doctor, data fisa, observatii)
/ALUES (4, 3, 1, 2, to_date('13.09.2020', 'dd.mm.yyyy'), 'Rana deschisa in zona botului');
:NSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa, observatii)
/ALUES (5, 1, 4, 1, to_date('13.07.2017', 'dd.mm.yyyy'), 'Una din cele mai frumoase specimene');
INSERT INTO FISA MEDICALA(id fisa, id diagnostic, id animalut, id doctor, data fisa)
JALUES (6, 2, 5, 1, to_date('04.05.2009', 'dd.mm.yyyy'));
:NSERT INTO FISA MEDICALA(id fisa, id diagnostic, id animalut, id doctor, data fisa, observatii)
JALUES (7, 1, 5, 2, to_date('14.06.2010', 'dd.mm.yyyy'), 'Vaccinare anuala');
INSERT INTO FISA MEDICALA(id fisa, id diagnostic, id animalut, id doctor, data fisa)
/ALUES (8, 1, 4, 2, to_date('13.07.2020', 'dd.mm.yyyy'));
INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa, observatii)
/ALUES (9, 3, 3, 1, to_date('13.12.2016', 'dd.mm.yyyy'), 'S-a taiat intr-o conserva de peste');
INSERT INTO FISA MEDICALA(id fisa, id diagnostic, id animalut, id doctor, data fisa, observatii)
/ALUES (10, 2, 1, 1, to date('25.12.2020', 'dd.mm.yyyy'), 'Cainele a mancat din pregaritile de Craciun');
:NSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
MALUES (11, 3, 5, 1, to_date('14.08.2015', 'dd.mm.yyyy'));
nsert into FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
/ALUES (13, 1, 13, 1, to_date('13.09.2020', 'dd.mm.yyyy'));
insert into FISA MEDICALA(id fisa, id diagnostic, id animalut, id doctor, data fisa)
VALUES (12, 1, 13, 1, to date('13.07.2021', 'dd.mm.yyyy'));
:NSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa, observatii)
VALUES (14, 3, 11, 1, to date('05.02.2023', 'dd.mm.yyyy'), 'catelus fara stapan');
:NSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa, observatii)
/ALUES (15, 3, 14, 1, to_date('05.02.2023', 'dd.mm.yyyy'), 'catelus fara stapan');
INSERT INTO PRODUCATOR(id producator, nume firma, telefon, email, id locatie)
INSERT INTO PRODUCATOR(id producator, nume firma, telefon, email, persoana contact, id locatie)
INSERT INTO PRODUCATOR(id producator, nume firma, telefon, email, persoana contact, id locatie)
INSERT INTO CONTRACT(id_contract, data_semnare)
```

```
VALUES (1, to_date('02.01.2005', 'dd.mm.yyyy'));
JALUES (2, to_date('13.07.2004', 'dd.mm.yyyy'));
INSERT INTO CONTRACT(id_contract, data_semnare)
/ALUES (3, to date('09.09.2005', 'dd.mm.yyyy'));
VALUES (4, to_date('01.05.2007', 'dd.mm.yyyy'));
INSERT INTO CONTRACT(id_contract, data_semnare)
VALUES (5, to_date('02.06.2008', 'dd.mm.yyyy'));
VALUES (6, to_date('02.01.2005', 'dd.mm.yyyy'));
VALUES (7, to_date('28.12.2020', 'dd.mm.yyyy'), 'Promovare', 2);
INSERT INTO CONTRACT(id_contract, data_semnare)
VALUES (8, to_date('02.01.2020', 'dd.mm.yyyy'));
INSERT INTO CONTRACT(id_contract, data_semnare)
VALUES (9, to_date('02.01.2021', 'dd.mm.yyyy'));
INSERT INTO CONTRACT(id contract, data semnare)
VALUES (10, to_date('02.07.2022', 'dd.mm.yyyy'));
INSERT INTO CONTRACT(id_contract, data_semnare)
VALUES (11, to_date('13.01.2023', 'dd.mm.yyyy'));
INSERT INTO CONTRACT_ANGAJAT(id_contract, id_angajat, rol, salariu)
INSERT INTO CONTRACT_ANGAJAT(id_contract, id_angajat, rol, salariu)
INSERT INTO CONTRACT ANGAJAT(id contract, id angajat, rol, salariu)
INSERT INTO CONTRACT_ANGAJAT(id_contract, id_angajat, rol, salariu)
INSERT INTO CONTRACT ANGAJAT(id contract, id angajat, rol, salariu)
INSERT INTO CONTRACT_ANGAJAT(id_contract, id_angajat, rol, salariu)
INSERT INTO CONTRACT PRODUCATOR(id contract, id producator)
INSERT INTO CONTRACT_PRODUCATOR(id_contract, id_producator)
```

```
INSERT INTO CONTRACT_PRODUCATOR(id_contract, id_producator)
INSERT INTO CONTRACT PRODUCATOR(id contract, id producator)
INSERT INTO CONTRACT_PRODUCATOR(id_contract, id_producator)
INSERT INTO FACTURA(ID FACTURA, TOTAL FACTURA, DATA EMITERE)
/ALUES (1, 300, to date('13.07.2020', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
/ALUES (2, 300, to_date('13.07.2011', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID FACTURA, TOTAL FACTURA, DATA EMITERE)
VALUES (3, 300, to_date('13.07.2015', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (4, 300, to_date('13.09.2020', 'dd.mm.yyyy'));
VALUES (5, 300, to_date('13.07.2017', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
/ALUES (6, 300, to_date('04.05.2009', 'dd.mm.yyyy'));
NSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
/ALUES (7, 300, to_date('14.06.2010', 'dd.mm.yyyy'));
NSERT INTO FACTURA(ID FACTURA, TOTAL FACTURA, DATA EMITERE)
JALUES (8, 300, to_date('13.07.2020', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID FACTURA, TOTAL FACTURA, DATA EMITERE)
/ALUES (9, 300, to_date('13.12.2016', 'dd.mm.yyyy'));
/ALUES (10, 300, to_date('25.12.2020', 'dd.mm.yyyy'));
nsert into FACTURA(id_factura, total_factura, data_emitere)
/ALUES (104, 300, to_date('13.07.2021', 'dd.mm.yyyy'));
VALUES (105, 3000, to_date('13.07.2021', 'dd.mm.yyyy'));
/ALUES (106, 3000, to_date('12.05.2021', 'dd.mm.yyyy'));
Insert into FACTURA(id_factura, total_factura, data_emitere)
VALUES (107, 3000, to_date('12.05.2021', 'dd.mm.yyyy'));
Insert into FACTURA(id factura, total factura, data emitere)
VALUES (1000, 100, sysdate);
Insert into FACTURA(id_factura, total_factura, data_emitere)
/ALUES (1001, 100, sysdate);
```

```
Insert into FACTURA(id_factura, total_factura, data_emitere)
VALUES (1002, 499, to_date('06.02.2023', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID FACTURA, TOTAL FACTURA, DATA EMITERE)
/ALUES (11, 1500, to_date('25.10.2009', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
/ALUES (12, 4325, to_date('13.11.2009', 'dd.mm.yyyy'));
/ALUES (13, 6000, to_date('25.07.2009', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (14, 1400, to_date('25.10.2014', 'dd.mm.yyyy'));
NSERT INTO FACTURA(ID FACTURA, TOTAL FACTURA, DATA EMITERE)
VALUES (15, 4725, to_date('13.04.2013', 'dd.mm.yyyy'));
/ALUES (16, 6025, to_date('25.09.2014', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (17, 5000, to_date('13.11.2019', 'dd.mm.yyyy'));
/ALUES (18, 7500, to_date('25.09.2019', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
/ALUES (19, 9256.35, to_date('12.04.2019', 'dd.mm.yyyy'));
INSERT INTO FACTURA PROPRIETAR(id factura, id proprietar, id fisa medicala)
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
INSERT INTO FACTURA PROPRIETAR(id factura, id proprietar, id fisa medicala)
INSERT INTO FACTURA PROPRIETAR(id factura, id proprietar, id fisa medicala)
INSERT INTO FACTURA PROPRIETAR(id factura, id proprietar, id fisa medicala)
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
INSERT INTO FACTURA PROPRIETAR(id factura, id proprietar, id fisa medicala)
INSERT INTO FACTURA PROPRIETAR(id factura, id proprietar, id fisa medicala)
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
```

```
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
INSERT INTO FACTURA PROPRIETAR(id factura, id proprietar, id fisa medicala)
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
INSERT INTO FACTURA PROPRIETAR(id factura, id proprietar, id fisa medicala)
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
INSERT INTO FACTURA PROPRIETAR(id factura, id proprietar)
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
INSERT INTO FACTURA PRODUCATOR(id factura, id producator)
INSERT INTO FACTURA_PRODUCATOR(id_factura, id_producator)
INSERT INTO FACTURA_PRODUCATOR(id_factura, id_producator)
INSERT INTO FACTURA PRODUCATOR(id factura, id producator)
INSERT INTO FACTURA_PRODUCATOR(id_factura, id_producator)
INSERT INTO FACTURA PRODUCATOR(id factura, id producator)
LNSERT INTO FACTURA_PRODUCATOR(id_factura, id_producator)
INSERT INTO FACTURA PRODUCATOR(id factura, id producator)
INSERT INTO FACTURA_PRODUCATOR(id_factura, id_producator)
INSERT INTO CHITANTA(id chitanta, suma platita, data plata, id factura)
VALUES (1, 150, to_date('13.07.2020', 'dd.mm.yyyy'), 1);
INSERT INTO CHITANTA(id chitanta, suma platita, data plata, id factura)
VALUES (2, 300, to_date('13.07.2011', 'dd.mm.yyyy'), 2);
```

```
VALUES (3, 300, to_date('13.08.2015', 'dd.mm.yyyy'), 3);
INSERT INTO CHITANTA(id chitanta, suma platita, data plata, id factura)
JALUES (4, 300, to_date('03.01.2021', 'dd.mm.yyyy'), 4);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
/ALUES (5, 150, to_date('13.07.2017', 'dd.mm.yyyy'), 5);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
/ALUES (8, 100, to_date('13.07.2009', 'dd.mm.yyyy'), 6);
INSERT INTO CHITANTA(id chitanta, suma platita, data plata, id factura)
JALUES (9, 200, to_date('17.08.2009', 'dd.mm.yyyy'), 6);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
/ALUES (10, 300, to_date('13.07.2010', 'dd.mm.yyyy'), 7);
/ALUES (11, 300, to_date('13.09.2020', 'dd.mm.yyyy'), 8);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (12, 300, to_date('13.12.2016', 'dd.mm.yyyy'), 9);
/ALUES (13, 300, to_date('15.01.2021', 'dd.mm.yyyy'), 10);
INSERT INTO CHITANTA(id chitanta, suma platita, data plata, id factura)
VALUES (14, 1500, to_date('25.10.2009', 'dd.mm.yyyy'), 11);
NSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
/ALUES (15, 4325, to_date('25.10.2009', 'dd.mm.yyyy'), 12);
INSERT INTO CHITANTA(id chitanta, suma platita, data plata, id factura)
INSERT INTO CHITANTA(id chitanta, suma platita, data plata, id factura)
INSERT INTO CHITANTA(id chitanta, suma platita, data plata, id factura)
VALUES (18, 1400, to_date('25.10.2014', 'dd.mm.yyyy'), 14);
/ALUES (20, 3025, to date('25.09.2014', 'dd.mm.yyyy'), 16);
/ALUES (21, 3000, to date('01.11.2014', 'dd.mm.yyyy'), 16);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
/ALUES (22, 4500, to_date('25.10.2019', 'dd.mm.yyyy'), 18);
```

```
<mark>insert into CHITANTA</mark>(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (104, 300, to date('13.07.2021', 'dd.mm.yyyy'), 104);
insert into CHITANTA(id chitanta, suma platita, data plata, id factura)
/ALUES (106, 600, to_date('13.07.2021', 'dd.mm.yyyy'), 106);
INSERT INTO PRODUS(id produs, caracteristici, id producator)
INSERT INTO PRODUS(id_produs, caracteristici, id_producator)
INSERT INTO INVENTAR(id_inventar, data_inventar)
JALUES (1, to_date('05.01.2010', 'dd.mm.yyyy'));
INSERT INTO INVENTAR(id inventar, data inventar)
/ALUES (2, to_date('05.01.2015', 'dd.mm.yyyy'));
INSERT INTO INVENTAR(id_inventar, data_inventar)
JALUES (3, to_date('05.01.2020', 'dd.mm.yyyy'));
INSERT INTO INVENTAR(id inventar, data inventar)
VALUES (4, to_date('05.01.2021', 'dd.mm.yyyy'));
INSERT INTO INVENTAR(id_inventar, data_inventar)
JALUES (5, to_date('05.01.2022', 'dd.mm.yyyy'));
<u> INSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate, data_expirare)</u>
/ALUES (1, 1, 1, 11, 10, to_date('05.01.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE DIN INVENTAR(id prod inventar, id produs, id inventar, id factura, cantitate, data expirare)
/ALUES (2, 2, 1, 12, 15, to_date('05.01.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE DIN INVENTAR(id prod inventar, id produs, id inventar, id factura, cantitate, data expirare)
/ALUES (3, 1, 2, 13, 20, to_date('05.01.2016', 'dd.mm.yyyy'));
:NSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate, data_expirare)
```

```
VALUES (4, 2, 2, 14, 25, to_date('05.01.2016', 'dd.mm.yyyy'));
INSERT INTO PRODUSE DIN INVENTAR(id prod inventar, id produs, id inventar, id factura, cantitate, data expirare)
VALUES (5, 1, 3, 15, 35, to date('05.03.2023', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate, data_expirare)
/ALUES (6, 2, 3, 16, 30, to_date('05.01.2021', 'dd.mm.yyyy'));
:NSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate)
INSERT INTO PRODUSE DIN INVENTAR(id prod inventar, id produs, id inventar, id factura, cantitate)
:NSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate)
INSERT INTO PRODUSE DIN INVENTAR(id prod inventar, id produs, id inventar, id factura, cantitate)
:NSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate)
INSERT INTO PRODUSE DIN INVENTAR(id prod inventar, id produs, id inventar, id factura, cantitate)
:NSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate)
:NSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate)
INSERT INTO PRODUSE DIN INVENTAR(id prod inventar, id produs, id inventar, id factura, cantitate)
<u> INSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate, data_expirare)</u>
/ALUES (16, 6, 1, 17, 100, to_date('13.07.2011', 'dd.mm.yyyy'));
<mark>INSERT INTO PRODUSE_DIN_INVENTAR</mark>(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate, data_expirare)
/ALUES (17, 7, 1, 18, 150, to_date('13.07.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE DIN INVENTAR(id prod inventar, id produs, id inventar, id factura, cantitate, data expirare)
/ALUES (18, 6, 2, 19, 100, to_date('13.07.2011', 'dd.mm.yyyy'));
<mark>INSERT INTO PRODUSE_DIN_INVENTAR</mark>(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate, data_expirare)
VALUES (19, 7, 2, 11, 150, to_date('13.07.2011', 'dd.mm.yyyy'));
<mark>INSERT INTO PRODUSE_DIN_INVENTAR</mark>(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate, data_expirare)
/ALUES (20, 6, 3, 12, 234, to_date('13.07.2011', 'dd.mm.yyyy'));
<u> INSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate, data_expirare)</u>
VALUES (21, 7, 3, 13, 512, to_date('13.07.2011', 'dd.mm.yyyy'));
```

```
[2023-02-06 05:40:18] 1 row affected in 12 ms
C##REKSIO> INSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate)
[2023-02-06 05:40:18] 1 row affected in 26 ms
G##REKSIO> INSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate, data_expirare)
          VALUES (16, 6, 1, 17, 100, to_date('13.07.2011', 'dd.mm.yyyy'))
[2023-02-06 05:40:18] 1 row affected in 11 ms
C##REKSIO> INSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate, data_expirare)
          VALUES (17, 7, 1, 18, 150, to_date('13.07.2011', 'dd.mm.yyyy'))
[2023-02-06 05:40:18] 1 row affected in 16 ms
CHARCE(SIO> INSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate, data_expirare)
          VALUES (18, 6, 2, 19, 100, to_date('13.07.2011', 'dd.mm.yyyy'))
[2023-02-06 05:40:18] 1 row affected in 16 ms
CHAREKSIO> INSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate, data_expirare)
          VALUES (19, 7, 2, 11, 150, to_date('13.07.2011', 'dd.mm.yyyy'))
[2023-02-06 05:40:18] 1 row affected in 11 ms
G##REKSIO> INSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate, data_expirare)
          VALUES (20, 6, 3, 12, 234, to_date('13.07.2011', 'dd.mm.yyyy'))
[2023-02-06 05:40:18] 1 row affected in 11 ms
C##REKSIO> INSERT INTO PRODUSE_DIN_INVENTAR(id_prod_inventar, id_produs, id_inventar, id_factura, cantitate, data_expirare)
          VALUES (21, 7, 3, 13, 512, to_date('13.07.2011', 'dd.mm.yyyy'))
[2023-02-06 05:40:18] 1 row affected in 7 ms
[2023-02-06 05:40:18] completed in 13 ms
```

```
from CONTRACT ANGAJAT a,
         CONTRACT C
ONNECT BY PRIOR ctr.id_contract = ctr.id_contract_modificat;
■■ ID_CONTRACT :
                     ■ ID_ANGAJAT ‡
                                    III DEPTH #
                                                ■ PRIORID_CONTRACT ÷
from ANIMALUT a,
    FISA_MEDICALA fm
and TO_CHAR(fm.DATA_FISA, 'YYYYY') like '2020'
     II ID_ANIMALUT :
                     III COUNT(*) ‡
```

```
select a.id proprietar, a.id animalut, sum(f.total factura), count(*)
rom FACTURA PROPRIETAR fp,
    PROPRIETAR p,
    ANIMALUT a
 and p.id proprietar = a.id proprietar
 and fp.id fisa medicala in (select fm.id fisa
                            from FISA MEDICALA fm
■ ID_PROPRIETAR ÷
                       III ID_ANIMALUT :
                                         III SUM(F.TOTAL_FACTURA) :
                                                               III COUNT(*) ÷
         from FACTURA f,
               FACTURA PROPRIETAR fp
         from FACTURA f.
              FACTURA PRODUCATOR fp
```

K	< 26 rows ∨ > > S	Q *		
	III ID_FACTURA ÷	■ DATA_EMITERE ÷	■ TOTAL_FACTURA :	I≣ TIP1 ÷
1	1001	2023-02-06 05:40:13	100.00	CUSTOM
2	14	2014-10-25	1400.00	PRODUCATOR
3	11	2009-10-25	1500.00	PRODUCATOR
4	12	2009-11-13	4325.00	PRODUCATOR
5	15	2013-04-13	4725.00	PRODUCATOR
6	17	2019-11-13	5000.00	PRODUCATOR
7	13	2009-07-25	6000.00	PRODUCATOR
8	16	2014-09-25	6025.00	PRODUCATOR
9	18	2019-09-25	7500.00	PRODUCATOR
10	19	2019-04-12	9256.35	PRODUCATOR
11	1000	2023-02-06 05:40:13	100.00	PROPRIETAR
12		2009-05-04	300.00	PROPRIETAR
13	7.	2010-06-14	300.00	PROPRIETAR
14	2	2011-07-13	300.00	PROPRIETAR
15	3	2015-07-13	300.00	PROPRIETAR
16		2016-12-13	300.00	PROPRIETAR
17		2017-07-13	300.00	PROPRIETAR
18	8	2020-07-13	300.00	PROPRIETAR
19		2020-07-13	300.00	PROPRIETAR
20		2020-09-13	300.00	PROPRIETAR
21	10	2020-12-25	300.00	PROPRIETAR
22	104	2021-07-13	300.00	PROPRIETAR
23	1002	2023-02-06	499.00	PROPRIETAR
24	107	2021-05-12	3000.00	PROPRIETAR
25	106	2021-05-12	3000.00	PROPRIETAR
26	105	2021-07-13	3000.00	PROPRIETAR

```
rom PROPRIETAR p
             3 Q #
     面 p.IO_PROPRIETAR : 面 p.NUME
                          * $11 PRENUME
                                        : BT TELEFON : BT EMAIL
                                                                                            顧ID ANIMALUT : 顧 a.NUME : 顧 RASA
                                                                                                                                      : BH ZI NASTERE
                                                                                                                 Golden Retriever
                                              771654789 popescu.ion@gmail.com
                                              729123123 cumparator@my.fmi.unibuc.ro
rom FACTURA f,
     CHITANTA C,
     FACTURA PROPRIETAR fp,
     PROPRIETAR p
 and p.id proprietar = fp.id proprietar
```

```
11 rows > > > 5 Q | #
     II ID_PROPRIETAR ÷
                     III ID_FACTURA $
                                   III TOTAL_FACTURA $
                                                    順面 COUNT(ID_CHITANTA) ÷
                                                                        III SUM(SUMA_PLATITA) ÷
                                           300.00
                                           300.00
                                           300.00
                                           300.00
                                           300.00
                                           300.00
                                           300.00
                                           300.00
from PROPRIETAR p,
    FACTURA f,
    FACTURA PROPRIETAR fp
naving COUNT(*) = (select count(*)
                    from FACTURA f1,
                          FACTURA PROPRIETAR fp1
                    where f1.id_factura = fp1.id_factura
                                                   from CHITANTA c
                                                   where c.id factura = f1.id factura));
II ID_PROPRIETAR :
```

```
select p.id_proprietar
from PROPRIETAR p
where NOT EXISTS(
            FACTURA_PROPRIETAR fp
              from FACTURA f1
              where f1.id_factura = f.id_factura
f1.id_factura)
 III ID_PROPRIETAR ‡
select a.id_animalut, round(MONTHS_BETWEEN(max(fm.data_fisa), min(fm.data_fisa)))
from ANIMALUT a,
    FISA_MEDICALA fm
group by a.id animalut;
```

```
III ID_ANIMALUT :
                                                                                                                                    III ROUND(MONTHS_BETWEEN(MAX(FM.DATA_FISA),MIN(FM.DATA_FISA))) :
rom PRODUSE_DIN_INVENTAR pi,
                  PRODUS p
   and TO_CHAR(Add_months(sysdate, 1), 'yyyy-mm') like TO_CHAR(pi.DATA_EXPIRARE, 'yyyy-mm');
                  順 ID_PROD_INVENTAR: 順 ID_PRODUS: 順 ID_INVENTAR: 順 ID_FACTURA: 順 ID_FACTURA: 順 IDATA_EXPIRARE : 順 DATA_EXPIRARE : [ ] DATA_EXPIRARE
                                                                                                                                                                                                                              35 2023-03-05
                                                                                                                                                                                                                                                                                                                 1 boabe de caini cu carne de vita si somon
from PRODUCATOR p
                                                                                                          from FACTURA_PRODUCATOR fp,
                                                                                                                                FACTURA f
                                                                                                          having sum(f.TOTAL_FACTURA) = (select max(s)
                                                                                                                                                                                                                                                       from (select sum(f1.total_factura) s
                                                                                                                                                                                                                                                                                  from factura f1,
                                                                                                                                                                                                                                                                                                         FACTURA PRODUCATOR fp1
                                                                                                                                                                                                                                                                                  where f1.id factura = fp1.id factura
```

```
IN ID_PRODUCATOR +
from ANGAJAT j,
    FISA MEDICALA fm,
    ANIMALUT a
| ⟨ ⟨ 2 rows ∨ ⟩ ⟩ | 😘 Q | | | + - 5 @ ↑ | Txc Manus
      IN ID_ANGAJAT : III NUME : III PRENUME :
                1 Staicu
                             Clara
                 2 Popa
                             Adi
from angajat ctr
| ⟨ ⟨ 5 rows ∨ ⟩ ⟩ | □ ♥ 🐧 🔘 | | ★
      IN ID_ANGAJAT # III DEPTH # III PRIORID_ANGAJAT #
select sum(Decode(INSTR(LOWER(p.caracteristici), 'vaccin'), 0, 0, 1))
```

```
rom produs p;
              III SUM(DECODE(INSTR(LOWER(P.CARACTERISTICI), 'VACCIN'),0,0,1)) :
          from ANIMALUT
select NULLIF(
                   when 1 then (select * from ans)
              IB NULLIF(CASE(SELECTCOUNT(*)FROMANS)WHEN1THEN(SELECT*FROMANS)ELSENULLEND, 'BOXER')
CREATE TABLE MESAJE
   CONSTRAINT pk_message PRIMARY KEY (message_id),
CREATE SEQUENCE seq_err START WITH 1;
```

```
F- ORDER BY
    MESSAGE_ID : MESSAGE
                                                                                     ÷ ■ MESSAGE_TYPE ÷ ■ CREATED_BY ÷ ■ CREATED_AT
                1 Alerta: raport suma_totala/co prost
                                                                                                         C##REKSIO
                                                                                                                        2023-02-06 05:49:30
                2 Alerta: raport suma_totala/co prost
                                                                                                         C##REKSIO
                                                                                                                        2023-02-06 05:49:30
                3 Alerta: raport suma_totala/co prost
                                                                                                        C##REKSIO
                                                                                                                        2023-02-06 05:49:30
                4 Alerta: raport suma_totala/co prost
                                                                                                        C##REKSIO
                                                                                                                        2023-02-06 05:49:30
                5 Alerta: raport suma_totala/co prost
                                                                                                        C##REKSIO
                                                                                                                        2023-02-06 05:49:30
                6 Alerta: raport suma_totala/co prost
                                                                                                        C##REKSIO
                                                                                                                        2023-02-06 05:49:30
                7 Alerta: raport suma_totala/co prost
                                                                                                        C##REKSIO
                                                                                                                        2023-02-06 05:49:30
                                                                                                                        2023-02-06 05:49:30
                8 Alerta: raport suma_totala/co prost
                                                                                                        C##REKSIO
                9 Alerta: raport suma_totala/co prost
                                                                                                        C##REKSIO
                                                                                                                        2023-02-06 05:49:30
               10 Alerta: raport suma_totala/co prost
                                                                                                                        2023-02-06 05:49:30
               11 Alerta: raport suma_totala/co prost
                                                                                                        C##REKSIO
                                                                                                                        2023-02-06 05:49:30
               12 Proprietarul animalutului Thor al lui Staicu Octavian-Florin are mai mul... I
                                                                                                        C##REKSIO
                                                                                                                        2023-02-06 05:49:30
               13 Nu s-a emis nicio factura pentru proprietarul animalutului Idk al lui FA... I
                                                                                                                        2023-02-06 05:49:30
               14 Alerta: factura emisa pentru animalut fara stapan
                                                                                                        C##REKSIO
                                                                                                                        2023-02-06 05:49:30
CREATE OR REPLACE PROCEDURE afis_doctor_favorit(id_animal ANIMALUT.id_animalut%type)
    TYPE tab_ind IS TABLE OF PLS_INTEGER INDEX BY PLS_INTEGER;
    TYPE tab_imb IS TABLE OF FISA_MEDICALA.id_doctor%type;
                tab_ind
                                   := tab_ind();
                tab imb
    index_max pls_integer;
    id doctor pls integer;
                ANGAJAT.nume%type;
   prenum
                ANGAJAT.prenume%type;
```

```
from FISA MEDICALA fm
where fm.id animalut = id animal;
id_doctor := doctors.FIRST;
    exit when id_doctor is null;
        cnt(id_doctor) := cnt(id_doctor) + 1;
            if cnt(id_doctor) > cnt(index_max) then
                index_max := id_doctor;
        when NO DATA FOUND then
            cnt(id_doctor) := 1;
                index_max := id_doctor;
    id_doctor := doctors.NEXT(id_doctor);
   DBMS_OUTPUT.PUT_LINE('Animalutul' | | id_animal | | ' nu are nicio fisa medicala in sistem');
   DBMS_OUTPUT.PUT_LINE('Animalutul ' | | id_animal | | ' a mers de cele mai multe ori la acesti doctori:');
        if cnt(id) = cnt(index_max) then
            into num, prenum
            from ANGAJAT a
```

```
id := cnt.NEXT(id);
    afis doctor favorit(1);
    afis_doctor_favorit(2);
    afis_doctor_favorit(3);
              afis_doctor_favorit(1);
              afis_doctor_favorit(2);
              afis_doctor_favorit(3);
 [2023-02-06 05:50:33] completed in 17 ms
 Animalutul 1 a mers de cele mai multe ori la acesti doctori:
 1. Staicu Clara
2. Popa Adi
3. Spalatorul Pop
 Animalutul 2 a mers de cele mai multe ori la acesti doctori:
1. Staicu Clara
 Animalutul 3 a mers de cele mai multe ori la acesti doctori:
1. Staicu Clara
2. Popa Adi
diagnosticati
CREATE OR REPLACE PROCEDURE afis_tratamente_animalute
```

```
cursor mycursor(id anim IN FISA MEDICALA.id animalut%type) is select diag.descriere,
                                                                          from FACTURA_PROPRIETAR fp,
                                                                               FACTURA f
fm.id_fisa)
                                                            from FISA MEDICALA fm,
                                                                 DIAGNOSTIC diag
                                                              and fm.id animalut = id anim;
   v cursor
   diag
                        DIAGNOSTIC.descriere%type;
   data_diagnostic
                        FISA_MEDICALA.data_fisa%type;
   id factura
                        FACTURA_PROPRIETAR.id_factura%type;
   data factura
                        FACTURA.data_emitere%type;
   id animalut max
   nume animalut max
                        ANIMALUT.nume%type;
   nume proprietar max
                        PROPRIETAR.nume%type;
   prenume_proprietar_max PROPRIETAR.prenume%type;
   co max
   faliment EXCEPTION;
   for i in (select a.id_animalut, a.nume nume_animalut, p.nume nume_proprietar, p.prenume prenume_proprietar
            from ANIMALUT a,
                 PROPRIETAR p
           open mycursor(i.id_animalut);
           fetch mycursor into diag, data_diagnostic, v_cursor;
           if mycursor%notfound then
              DBMS_OUTPUT.PUT_LINE('Animalutul' || i.nume_animalut || 'al lui' || i.nume_proprietar || ''|
                                  DBMS_OUTPUT.NEW_LINE();
```

```
DBMS_OUTPUT.PUT_LINE('Animalutul ' | i.nume_animalut | | al lui ' | i.nume_proprietar | |
                                  exit when mycursor%notfound;
                 DBMS_OUTPUT.PUT_LINE(mycursor%rowcount | | '. ' | | diag | | ' la data de ' | | data_diagnostic | |
                     fetch v_cursor into id_factura, data_factura;
                     exit when v_cursor%notfound;
                     DBMS_OUTPUT.PUT_LINE(' ' || mycursor%rowcount || '.' || v_cursor%rowcount || '.
ld_factura=' ||
                                         id_factura || ', emisa la data de ' || data_factura);
                     DBMS OUTPUT.PUT LINE(' Nicio factura emisa pentru aceasta fisa medicala');
                  fetch mycursor into diag, data diagnostic, v cursor;
              DBMS_OUTPUT.NEW_LINE();
              if mycursor%rowcount > co_max then
                  co_max := mycursor%rowcount;
                  id animalut max := i.id animalut;
                  nume animalut max := i.nume animalut;
                  nume_proprietar_max := i.nume_proprietar;
                  prenume_proprietar_max := i.prenume_proprietar;
          close mycursor;
   if co max = 0 then
      raise faliment;
  DBMS_OUTPUT.PUT_LINE('Animalutul' | nume_animalut_max | | ' [id_animalut=' | id_animalut_max | | '] al lui' | |
                      nume_proprietar_max || ' ' || prenume_proprietar_max ||
```

```
' este unul dintre cei mai fideli clienti ai nostri: ne-a vizitat de ' || co_max || ' ori
:)');

EXCEPTION

when faliment then

DBMS_OUTPUT.PUT_LINE('Nu avem niciun animalut care s-a tratat la noi...');

insert into mesaje(message_id, message, message_type, created_by, created_at)

values (seq_err.NEXTVAL, 'Nu avem niciun animalut care s-a tratat la noi...', 'I', USER, sysdate);

commit;

END;

begin

afis_tratamente_animalute();
end;

/
```

```
afis_tratamente_animalute();
[2023-02-06 05:51:10] completed in 37 ms
Animalutul Bit al lui Staicu Octavian-Florin a avut urmatoarele interventii medicale:
1. Enterocolita la data de 13-JUL-20, avand emise facturile:
 1.1. id_factura=1, emisa la data de 13-JUL-20
2. Rana deschisa la data de 13-SEP-20, avand emise facturile:
 2.1. id_factura=4, emisa la data de 13-SEP-20
3. Enterocolita la data de 25-DEC-20, avand emise facturile:
 3.1. id_factura=10, emisa la data de 25-DEC-20
Animalutul Kid al lui Staicu Adrian a avut urmatoarele interventii medicale:
1. Vaccinare Leptospiroza la data de 13-JUL-11, avand emise facturile:
 1.1. id_factura=2, emisa la data de 13-JUL-11
Animalutul Kitty al lui Bodea George a avut urmatoarele interventii medicale:
1. Vaccinare Leptospiroza la data de 13-JUL-16, avand emise facturile:
 1.1. id_factura=3, emisa la data de 13-JUL-15
2. Rana deschisa la data de 13-DEC-16, avand emise facturile:
 2.1. id_factura=9, emisa la data de 13-DEC-16
Animalutul Angelina al lui Cineva Interesant a avut urmatoarele interventii medicale:
1. Vaccinare Leptospiroza la data de 13-JUL-17, avand emise facturile:
 1.1. id_factura=5, emisa la data de 13-JUL-17
2. Vaccinare Leptospiroza la data de 13-JUL-20, avand emise facturile:
 2.1. id_factura=8, emisa la data de 13-JUL-20
Animalutul John al lui Popescu Ion a avut urmatoarele interventii medicale:
1. Enterocolita la data de 04-MAY-09, avand emise facturile:
 1.1. id_factura=6, emisa la data de 04-MAY-09
2. Vaccinare Leptospiroza la data de 14-JUN-10, avand emise facturile:
 2.1. id_factura=7, emisa la data de 14-JUN-10
3. Rana deschisa la data de 14-AUG-15, avand emise facturile:
 3.1. id_factura=104, emisa la data de 13-JUL-21
Animalutul dog al lui Cineva Interesant nu a avut niciodata parte de o procedura medicala
Animalutul Thor al lui Staicu Octavian-Florin a avut urmatoarele interventii medicale:
1. Vaccinare Leptospiroza la data de 13-SEP-20, avand emise facturile:
 1.1. id_factura=106, emisa la data de 12-MAY-21
 1.2. id_factura=107, emisa la data de 12-MAY-21
2. Vaccinare Leptospiroza la data de 13-JUL-21, avand emise facturile:
 2.1. id_factura=105, emisa la data de 13-JUL-21
Animalutul Bit [id_animalut=1] al lui Staicu Octavian-Florin este unul dintre cei mai fideli clienti ai nostri: ne-a vizitat de 3 ori :)
```

```
REATE OR REPLACE FUNCTION factura aferenta fisa(suma min IN float, suma totala OUT float) return pls integer
        FACTURA PROPRIETAR.id factura%type;
  pret
        FACTURA.total_factura%type;
  factura fara stapan EXCEPTION;
  raport pret consultatii prost EXCEPTION;
  suma totala := 0;
  for i in (select p.id proprietar, p.nume propnum, p.prenume, a.nume animnume, fm.data fisa, fm.id fisa
          from FISA MEDICALA fm,
              PROPRIETAR p
            into id, pret
            from FACTURA PROPRIETAR fp,
                FACTURA f
            DBMS_OUTPUT.PUT_LINE('Proprietarul animalutului ' || i.animnume || ' al lui ' ||
```

```
raise factura fara stapan;
suma_totala := suma_totala + pret;
if pret >= suma min then
if suma totala / co tot < suma min then</pre>
   raise raport pret consultatii prost;
when NO DATA FOUND then
    DBMS_OUTPUT.PUT_LINE('Nu s-a emis nicio factura pentru proprietarul animalutului ' | i.animnume
                        'al lui ' | | nvl(i.propnum, 'FARA') | | ' ' | | nvl(i.prenume, 'PROPRIETAR')
    insert into mesaje(message_id, message, message_type, created_by, created_at)
    values (seg err.NEXTVAL,
           ' al lui ' || nvl(i.propnum, 'FARA') || ' ' || nvl(i.prenume, 'PROPRIETAR') ||
            ' desi a fost tratat, cu fisa medicala cu id ' | | i.id_fisa, 'I', USER, sysdate);
when TOO MANY ROWS then
    DBMS_OUTPUT.PUT_LINE('Proprietarul animalutului ' | | i.animnume | | ' al lui ' | |
                         nvl(i.propnum, 'FARA') |  ' ' |  nvl(i.prenume, 'PROPRIETAR') | |
    insert into mesaje(message_id, message, message_type, created_by, created_at)
    values (seq_err.NEXTVAL, 'Proprietarul animalutului ' | i.animnume | | ' al lui ' | |
                             nvl(i.propnum, 'FARA') | | ' ' | | nvl(i.prenume, 'PROPRIETAR') | |
                             ' are mai multe facturi emise la data de ' | | i.data_fisa, 'I', USER,
            sysdate);
when factura_fara_stapan then
    DBMS_OUTPUT.PUT_LINE('Alerta: factura emisa pentru animalut fara stapan');
```

```
suma_min float := 490;
              suma totala float:
              res := factura_aferenta_fisa(suma_min, suma_totala);
              DBMS_OUTPUT.PUT_LINE('suma totala din facturi unice: ' || suma_totala);
[2023-02-06 05:52:20] completed in 20 ms
Proprietarul animalutului Bit al lui Staicu Octavian-Florin are factura 1 emisa la data de 13-JUL-20, pentru fisa medicala 1, total_factura=300
Alerta: raport suma_totala/co prost
Proprietarul animalutului Kid al lui Staicu Adrian are factura 2 emisa la data de 13-JUL-11, pentru fisa medicala 2, total_factura=300
Alerta: raport suma_totala/co prost
Proprietarul animalutului Kitty al lui Bodea George are factura 3 emisa la data de 13-JUL-16, pentru fisa medicala 3, total_factura=300
Alerta: raport suma_totala/co prost
Proprietarul animalutului Bit al lui Staicu Octavian-Florin are factura 4 emisa la data de 13-SEP-20, pentru fisa medicala 4, total_factura=300
Alerta: raport suma_totala/co prost
Proprietarul animalutului Angelina al lui Cineva Interesant are factura 5 emisa la data de 13-JUL-17, pentru fisa medicala 5, total_factura=300
Alerta: raport suma_totala/co prost
Proprietarul animalutului John al lui Popescu Ion are factura 6 emisa la data de 04-MAY-09, pentru fisa medicala 6, total_factura=300
Alerta: raport suma_totala/co prost
Proprietarul animalutului John al lui Popescu Ion are factura 7 emisa la data de 14-JUN-10, pentru fisa medicala 7, total_factura=300
Alerta: raport suma_totala/co prost
Proprietarul animalutului Angelina al lui Cineva Interesant are factura 8 emisa la data de 13-JUL-20, pentru fisa medicala 8, total_factura=300
Alerta: raport suma_totala/co prost
Proprietarul animalutului Kitty al lui Bodea George are factura 9 emisa la data de 13-DEC-16, pentru fisa medicala 9, total_factura=300
Alerta: raport suma_totala/co prost
Proprietarul animalutului Bit al lui Staicu Octavian-Florin are factura 10 emisa la data de 25-DEC-20, pentru fisa medicala 10, total_factura=300
Alerta: raport suma_totala/co prost
Proprietarul animalutului John al lui Popescu Ion are factura 104 emisa la data de 14-AUG-15, pentru fisa medicala 11, total_factura=300
Alerta: raport suma_totala/co prost
Proprietarul animalutului Thor al lui Staicu Octavian-Florin are mai multe facturi emise la data de 13-SEP-20
Proprietarul animalutului Thor al lui Staicu Octavian-Florin are factura 105 emisa la data de 13-JUL-21, pentru fisa medicala 12, total_factura=3000
Nu s-a emis nicio factura pentru proprietarul animalutului Idk al lui FARA PROPRIETAR la data de 05-FEB-23 desi a fost tratat, cu fisa medicala cu id 14
Proprietarul animalutului Idk2 al lui FARA PROPRIETAR are factura 1002 emisa la data de 05-FEB-23, pentru fisa medicala 15, total_factura=499
Alerta: factura emisa pentru animalut fara stapan
1 clienti au facturi de minim 490
suma totala din facturi unice: 6300
```

```
CREATE or REPLACE TYPE tab imb is TABLE OF NUMBER(10);
CREATE or REPLACE FUNCTION topclienti(top_clienti out tab_imb) return pls_integer,
   cursor mycursor is select f.id_proprietar,
                                     from CHITANTA c
                      from FACTURA_PROPRIETAR f,
                           PROPRIETAR p
   v cursor refcursor;
              FACTURA_PROPRIETAR.id_factura%type := null;
   total_suma pls_integer
   max suma pls integer
             PROPRIETAR.id_proprietar%type
   top_clienti := tab_imb();
   open mycursor;
       fetch mycursor into id, v_cursor;
       if last id is null then
           if mycursor%notfound or id <> last_id then
               if total suma > max suma then
                   top_clienti.DELETE;
                   top_clienti.EXTEND;
                  max_suma := total_suma;
```

```
elsif total_suma = max_suma then
                top_clienti.EXTEND;
    exit when mycursor%notfound;
        fetch v_cursor into suma;
       exit when v_cursor%notfound;
close mycursor;
mytab tab_imb;
      PROPRIETAR.nume%type;
prenum PROPRIETAR.prenume%type;
co := TOPCLIENTI(mytab);
DBMS_OUTPUT.PUT_LINE('Exista ' | co | clienti:');
id := mytab.FIRST;
    from PROPRIETAR
```

```
where id_proprietar = mytab(id);
       id := mytab.NEXT(id);
 [2023-02-06 05:53:35] completed in 34 ms
 Exista 1 clienti:
 Bodea George (id: 2)
CREATE OR REPLACE TRIGGER trig_contract
   BEFORE DELETE
   RAISE APPLICATION ERROR(-20222, 'Contractele trebuie sa ramana pe veci in sistem');
CREATE OR REPLACE TRIGGER trig_contract_angajat
   BEFORE DELETE
   on CONTRACT ANGAJAT
   RAISE_APPLICATION_ERROR(-20222, 'Contractele trebuie sa ramana pe veci in sistem');
CREATE OR REPLACE TRIGGER trig_contract_producator
   on CONTRACT_PRODUCATOR
   RAISE_APPLICATION_ERROR(-20222, 'Contractele trebuie sa ramana pe veci in sistem');
from CONTRACT
```

```
from CONTRACT ANGAJAT
from CONTRACT PRODUCATOR
          from CONTRACT
          where id_contract = 1
[2023-02-06 05:55:11] [72000][20222]
[2023-02-06 05:55:11] ORA-04088: error during execution of trigger 'C##REKSIO.TRIG_CONTRACT'
[2023-02-06 05:55:11] Position: 13
          from CONTRACT ANGAJAT
          where id contract = 1
[2023-02-06 05:55:14] [72000][20222]
[2023-02-06 05:55:14] ORA-20222: Contractele trebuie sa ramana pe veci in sistem
[2023-02-06 05:55:14] ORA-06512: at "C##REKSIO.TRIG_CONTRACT_ANGAJAT", line 2
[2023-02-06 05:55:14] ORA-04088: error during execution of trigger 'C##REKSIO.TRIG_CONTRACT_ANGAJAT'
[2023-02-06 05:55:14] Position: 13
          from CONTRACT_PRODUCATOR
          where id_contract = 4
[2023-02-06 05:55:16] [72000][20222]
[2023-02-06 05:55:16] ORA-20222: Contractele trebuie sa ramana pe veci in sistem
[2023-02-06 05:55:16] Position: 13
CREATE OR REPLACE TRIGGER trig_tratament_doctor
    on FISA MEDICALA
```

```
from ANGAJAT
   where id angajat = :NEW.id doctor;
        RAISE_APPLICATION_ERROR(-20001, 'angajatul cu id-ul ' | : NEW.id_doctor | | ' nu poate trata animalute');
   when NO_DATA_FOUND then
        RAISE APPLICATION ERROR(-20000, 'Nu exista un asemenea doctor');
INSERT INTO FISA MEDICALA(id fisa, id diagnostic, id animalut, id doctor, data fisa)
VALUES (60, 1, 1, 3, sysdate);
.<mark>NSERT INTO FISA_MEDICALA</mark>(id_fisa, id_diagnostic, id_animalut, <u>id_doctor, data_fisa)</u>
MALUES (60, 1, 1, 6, sysdate);
CHARREKSIO> INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
[2023-02-06 05:55:59] [72000][20001]
[2023-02-06 05:55:59] ORA-20001: angajatul cu id-ul 3 nu poate trata animalute
[2023-02-06 05:55:59] ORA-06512: at "C##REKSIO.TRIG_TRATAMENT_DOCTOR", line 9
[2023-02-06 05:55:59] ORA-04088: error during execution of trigger 'C##REKSIO.TRIG_TRATAMENT_DOCTOR'
[2023-02-06 05:55:59] Position: 12
C##REKSIO> INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
[2023-02-06 05:56:01] [72000][20000]
[2023-02-06 05:56:01] ORA-20000: Nu exista un asemenea doctor
[2023-02-06 05:56:01] ORA-04088: error during execution of trigger 'C##REKSIO.TRIG_TRATAMENT_DOCTOR'
[2023-02-06 05:56:01] Position: 12
CREATE OR REPLACE TRIGGER trig_tratament_zinastere
   on FISA MEDICALA
```

```
##REKSIO> INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
          VALUES (100, 1, 1, 1, to_date('01.01.1991', 'dd.mm.yyyy'))
[2023-02-06 05:56:44] [72000][20013]
[2023-02-06 05:56:44] ORA-20013: Nu poti trata un animalut inainte ca acesta sa se nasca
[2023-02-06 05:56:44] ORA-06512: at "C##REKSIO.TRIG_TRATAMENT_ZINASTERE", line 10
[2023-02-06 05:56:44] ORA-04088: error during execution of trigger 'C##REKSIO.TRIG_TRATAMENT_ZINASTERE'
[2023-02-06 05:56:44] Position: 12
C##REKSIO> INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
          VALUES (100, 1, 100, 1, to_date('01.01.1991', 'dd.mm.yyyy'))
[2023-02-06 05:56:45] [72000][20000]
[2023-02-06 05:56:45] ORA-20000: Nu exista un astfel de animalut in sistem
[2023-02-06 05:56:45] ORA-04088: error during execution of trigger 'C##REKSIO.TRIG_TRATAMENT_ZINASTERE'
[2023-02-06 05:56:45] Position: 12
C##REKSIO> update FISA MEDICALA
          set data_fisa=to_date('01.01.1991', 'dd.mm.yyyy')
         where id fisa = 1
[2023-02-06 05:56:47] ORA-04088: error during execution of trigger 'C##REKSIO.TRIG_TRATAMENT_ZINASTERE'
[2023-02-06 05:56:47] Position: 7
CREATE OR REPLACE TRIGGER trig_update_contracte
   DBMS_OUTPUT.PUT_LINE('Afisare cotracte actualizate');
        DBMS_OUTPUT.PUT_LINE('ID contract ' | :new.id_contract);
        DBMS_OUTPUT.PUT_LINE('Data semnare' | : new.data_semnare);
```

```
DBMS_OUTPUT.PUT_LINE('Observatii ' | :new.observatii);
       DBMS_OUTPUT.PUT_LINE('ID contract modificat' | :new.id_contract_modificat);
   END AFTER EACH ROW;
   AFTER STATEMENT IS
       DBMS_OUTPUT.PUT_LINE('Contractele au fost actualizate');
   END AFTER STATEMENT;
   END trig_update_contracte;
JPDATE CONTRACT
C##REKSIO> UPDATE CONTRACT
         SET OBSERVATII='Obs'
         WHERE ID_CONTRACT = 10
[2023-02-06 05:57:13] 1 row affected in 10 ms
Afisare cotracte actualizate
ID contract 10
Data semnare 02-JUL-22
Observatii Obs
ID contract modificat
Contractele au fost actualizate
CREATE OR REPLACE TRIGGER trig_ldd_contract
   if lower(ORA_DICT_OBJ_NAME) = lower('Contract') or lower(ORA_DICT_OBJ_NAME) = lower('Contract_angajat') or
      lower(ORA_DICT_OBJ_NAME) = lower('Contract_producator') then
       RAISE_APPLICATION_ERROR(-20777, 'Nu aveti voie sa modificati sau sa stegeti tabela cu contracte');
```

```
drop table CONTRACT;
drop table CONTRACT PRODUCATOR;
drop table CONTRACT ANGAJAT;
alter table CONTRACT
C##REKSIO> drop table CONTRACT
[2023-02-06 05:57:57] [42000][4088]
[2023-02-06 05:57:57] ORA-04088: error during execution of trigger 'C##REKSIO.TRIG_LDD_CONTRACT'
[2023-02-06 05:57:57] ORA-00604: error occurred at recursive SQL level 1
[2023-02-06 05:57:57] ORA-06512: at line 4
[2023-02-06 05:57:57] Position: 0
 C##REKSIC> drop table CONTRACT_PRODUCATOR
[2023-02-06 05:58:00] [42000][4088]
[2023-02-06 05:58:00] ORA-04088: error during execution of trigger 'C##REKSIO.TRIG_LDD_CONTRACT'
[2023-02-06 05:58:00] ORA-00604: error occurred at recursive SQL level 1
[2023-02-06 05:58:00] ORA-06512: at line 4
[2023-02-06 05:58:00] Position: 0
 C##REKSID> drop table CONTRACT_ANGAJAT
[2023-02-06 05:58:04] [42000][4088]
[2023-02-06 05:58:04] ORA-06512: at line 4
[2023-02-06 05:58:04] Position: 0
 CHUREKSIO > alter table CONTRACT
              drop column observatii
[2023-02-06 05:58:07] [42000][4088]
[2023-02-06 05:58:07] ORA-04088: error during execution of trigger 'C##REKSIO.TRIG_LDD_CONTRACT'
[2023-02-06 05:58:07] ORA-00604: error occurred at recursive SQL level 1
[2023-02-06 05:58:07] ORA-20777: Nu aveti voie sa modificati sau sa stegeti tabela cu contracte
[2023-02-06 05:58:07] ORA-06512: at line 4
[2023-02-06 05:58:07] Position: 0
```

```
CREATE OR REPLACE PACKAGE package clinica vet tav13
   PROCEDURE afis_doctor_favorit(id_animal ANIMALUT.id_animalut%type);
   PROCEDURE afis_tratamente_animalute;
   FUNCTION factura_aferenta_fisa(suma_min IN float, suma_totala OUT float) return pls_integer;
   TYPE tab imb is TABLE OF NUMBER(10);
   FUNCTION topclienti(top_clienti out tab_imb) return pls_integer;
END package_clinica_vet_tav13;
CREATE OR REPLACE PACKAGE BODY package clinica vet tav13
   PROCEDURE afis_doctor_favorit(id_animal ANIMALUT.id_animalut%type)
       TYPE tab ind IS TABLE OF PLS INTEGER INDEX BY PLS INTEGER;
       TYPE tab_imb IS TABLE OF FISA_MEDICALA.id_doctor%type;
                 tab ind
                                := tab ind();
       doctors tab_imb
                                := tab_imb();
       index_max pls_integer;
       id doctor pls integer;
                ANGAJAT.nume%type;
      prenum
                ANGAJAT.prenume%type;
       from FISA MEDICALA fm
       where fm.id animalut = id animal;
       id doctor := doctors.FIRST;
           exit when id doctor is null;
               cnt(id doctor) := cnt(id doctor) + 1;
                  if cnt(id_doctor) > cnt(index_max) then
```

```
index_max := id_doctor;
        EXCEPTION
            when NO_DATA_FOUND then
                cnt(id_doctor) := 1;
                    index_max := id_doctor;
        id doctor := doctors.NEXT(id doctor);
       DBMS_OUTPUT.PUT_LINE('Animalutul ' | | id_animal | | ' nu are nicio fisa medicala in sistem');
        DBMS_OUTPUT.PUT_LINE('Animalutul ' | | id_animal | | ' a mers de cele mai multe ori la acesti doctori:');
            if cnt(id) = cnt(index_max) then
                into num, prenum
                from ANGAJAT a
                DBMS_OUTPUT.PUT_LINE(co || '. ' || num || ' ' || prenum);
            id := cnt.NEXT(id);
END afis_doctor_favorit;
PROCEDURE afis_tratamente_animalute
    cursor mycursor(id anim IN FISA MEDICALA.id animalut%type) is select diag.descriere,
```

```
from FACTURA PROPRIETAR fp,
                                                                                    FACTURA f
fm.id_fisa)
                                                                 from FISA MEDICALA fm,
                                                                      DIAGNOSTIC diag
                                                                   and fm.id animalut = id anim;
       v_cursor
       diag
                             DIAGNOSTIC.descriere%type;
                            FISA_MEDICALA.data_fisa%type;
       data_diagnostic
                            FACTURA PROPRIETAR.id factura%type;
       id factura
       data_factura
                            FACTURA.data_emitere%type;
       id animalut max
       nume_animalut_max
                            ANIMALUT.nume%type;
                            PROPRIETAR.nume%type;
       nume_proprietar_max
       prenume_proprietar_max PROPRIETAR.prenume%type;
       ok_factura
       for i in (select a.id_animalut, a.nume nume_animalut, p.nume nume_proprietar, p.prenume prenume_proprietar
                from ANIMALUT a,
                     PROPRIETAR p
               open mycursor(i.id animalut);
               fetch mycursor into diag, data diagnostic, v cursor;
               if mycursor%notfound then
                  DBMS_OUTPUT.PUT_LINE('Animalutul' || i.nume_animalut || 'al lui' || i.nume_proprietar || ''
                                      DBMS_OUTPUT.NEW_LINE();
                  DBMS_OUTPUT.PUT_LINE('Animalutul ' | | i.nume_animalut | | ' al lui ' | | i.nume_proprietar | | ' '
```

```
exit when mycursor%notfound;
               DBMS_OUTPUT.PUT_LINE(mycursor%rowcount || '. ' || diag || ' la data de ' || data_diagnostic
                  fetch v_cursor into id_factura, data_factura;
                  exit when v_cursor%notfound;
                  DBMS_OUTPUT.PUT_LINE(' ' | mycursor%rowcount | | '.' | v_cursor%rowcount | |
                                      id_factura || ', emisa la data de ' || data_factura);
               if ok_factura = 0 then
                  DBMS_OUTPUT.PUT_LINE(' Nicio factura emisa pentru aceasta fisa medicala');
               fetch mycursor into diag, data diagnostic, v cursor;
           DBMS_OUTPUT.NEW_LINE();
           if mycursor%rowcount > co max then
               co max := mycursor%rowcount;
               id animalut max := i.id animalut;
              nume_animalut_max := i.nume_animalut;
              nume proprietar max := i.nume proprietar;
               prenume proprietar max := i.prenume proprietar;
       close mycursor;
   raise faliment;
DBMS_OUTPUT.PUT_LINE('Animalutul ' | nume_animalut_max | | ' [id_animalut=' | id_animalut_max | | '] al lui
```

```
nume_proprietar_max || ' ' || prenume_proprietar_max ||
   when faliment then
       DBMS_OUTPUT.PUT_LINE('Nu avem nicium animalut care s-a tratat la noi...');
END afis tratamente animalute;
FUNCTION factura aferenta fisa(suma min IN float, suma totala OUT float) return pls integer
          FACTURA_PROPRIETAR.id_factura%type;
   pret
         FACTURA.total_factura%type;
   factura fara stapan EXCEPTION;
   raport_pret_consultatii_prost EXCEPTION;
   suma totala := 0;
   for i in (select p.id proprietar, p.nume propnum, p.prenume, a.nume animnume, fm.data fisa, fm.id fisa
            from FISA MEDICALA fm,
                 PROPRIETAR p
              into id, pret
              from FACTURA PROPRIETAR fp,
                   FACTURA f
              DBMS_OUTPUT.PUT_LINE('Proprietarul animalutului ' | | i.animnume | | ' al lui ' | |
```

```
raise factura fara stapan;
suma_totala := suma_totala + pret;
if pret >= suma min then
if suma totala / co tot < suma min then
    raise raport pret consultatii prost;
when NO_DATA_FOUND then
    DBMS OUTPUT.PUT LINE('Nu s-a emis nicio factura pentru proprietarul animalutului ' | |
                         ' al lui ' || nvl(i.propnum, 'FARA') || ' ' ||
    insert into mesaje(message_id, message, message_type, created_by, created_at)
    values (seg err.NEXTVAL,
           ' al lui ' || nvl(i.propnum, 'FARA') || ' ' || nvl(i.prenume, 'PROPRIETAR') ||
            ' desi a fost tratat, cu fisa medicala cu id ' || i.id_fisa, 'I', USER, sysdate);
when TOO MANY ROWS then
    DBMS_OUTPUT.PUT_LINE('Proprietarul animalutului ' | | i.animnume | | ' al lui ' | |
                         nvl(i.propnum, 'FARA') | | ' ' | | nvl(i.prenume, 'PROPRIETAR') | |
    insert into mesaje(message_id, message, message_type, created_by, created_at)
    values (seq_err.NEXTVAL, 'Proprietarul animalutului ' | | i.animnume | | ' al lui ' | |
                             nvl(i.propnum, 'FARA') | ' ' | nvl(i.prenume, 'PROPRIETAR') | |
            sysdate);
when factura_fara_stapan then
```

```
DBMS_OUTPUT.PUT_LINE('Alerta: factura emisa pentru animalut fara stapan');
                    insert into mesaje (message id, message, message type, created by, created at)
                    values (seq err.NEXTVAL, 'Alerta: factura emisa pentru animalut fara stapan', 'I', USER,
                            sysdate);
               when raport_pret_consultatii_prost then
                    DBMS_OUTPUT.PUT_LINE('Alerta: raport suma_totala/co prost');
                    insert into mesaje(message id, message, message type, created by, created at)
                    values (seq err.NEXTVAL, 'Alerta: raport suma totala/co prost', 'I', USER, sysdate);
end factura_aferenta_fisa;
FUNCTION topclienti(top_clienti out tab_imb) return pls_integer
   cursor mycursor is select f.id_proprietar,
                                     from CHITANTA c
                       from FACTURA PROPRIETAR f,
                            PROPRIETAR p
   v cursor refcursor;
              FACTURA PROPRIETAR.id factura type := null;
   max suma pls integer
             PROPRIETAR.id_proprietar%type
    top_clienti := tab_imb();
   open mycursor;
       fetch mycursor into id, v_cursor;
```

```
if mycursor%notfound or id <> last_id then
                if total suma > max suma then
                    top_clienti(1) := last id;
                    max_suma := total_suma;
                elsif total_suma = max_suma then
                    top clienti(co) := last id;
                total suma := 0;
          exit when mycursor%notfound;
             exit when v_cursor%notfound;
      close mycursor;
   END topclienti;
   package_clinica_vet_tav13.afis_doctor_favorit(1);
DBMS_OUTPUT.PUT_LINE(' .....');
   package_clinica_vet_tav13.afis_tratamente_animalute();
```

```
DBMS_OUTPUT_PUT_LINE('.....');
      sum tot float;
      co := package clinica vet tav13. factura aferenta fisa(490, sum tot);
      DBMS_OUTPUT.PUT_LINE(co |  ' clienti au facturi de minim ' |  490);
      DBMS_OUTPUT.PUT_LINE('suma totala din facturi unice: ' | sum_tot);
DBMS_OUTPUT.PUT_LINE(' .....');
      mytab package_clinica_vet_tav13.tab_imb;
      num PROPRIETAR.nume%type;
      prenum PROPRIETAR.prenume%type;
      co := package clinica vet tav13.TOPCLIENTI(mytab);
      DBMS_OUTPUT.PUT_LINE('Exista ' | co | clienti:');
      id := mytab.FIRST;
         into num, prenum
         from PROPRIETAR
         where id_proprietar = mytab(id);
         DBMS_OUTPUT.PUT_LINE(num | | ' ' | | prenum | | ' (id: ' | | mytab(id) | | ')');
         id := mytab.NEXT(id);
[2023-02-06 05:58:56] completed in 52 ms
Animalutul 1 a mers de cele mai multe ori la acesti doctori:
```

```
1. Staicu Clara
2. Popa Adi
3. Spalatorul Pop
Animalutul Bit al lui Staicu Octavian-Florin a avut urmatoarele interventii medicale:
1. Enterocolita la data de 13-JUL-20, avand emise facturile:
 1.1. id factura=1, emisa la data de 13-JUL-20
2. Rana deschisa la data de 13-SEP-20, avand emise facturile:
 2.1. id factura=4, emisa la data de 13-SEP-20
3. Enterocolita la data de 25-DEC-20, avand emise facturile:
 3.1. id factura=10, emisa la data de 25-DEC-20
Animalutul Kid al lui Staicu Adrian a avut urmatoarele interventii medicale:
1. Vaccinare Leptospiroza la data de 13-JUL-11, avand emise facturile:
 1.1. id factura=2, emisa la data de 13-JUL-11
Animalutul Kitty al lui Bodea George a avut urmatoarele interventii medicale:
1. Vaccinare Leptospiroza la data de 13-JUL-16, avand emise facturile:
 1.1. id factura=3, emisa la data de 13-JUL-15
2. Rana deschisa la data de 13-DEC-16, avand emise facturile:
 2.1. id_factura=9, emisa la data de 13-DEC-16
Animalutul Angelina al lui Cineva Interesant a avut urmatoarele interventii medicale:
1. Vaccinare Leptospiroza la data de 13-JUL-17, avand emise facturile:
 1.1. id factura=5, emisa la data de 13-JUL-17
2. Vaccinare Leptospiroza la data de 13-JUL-20, avand emise facturile:
 2.1. id factura=8, emisa la data de 13-JUL-20
Animalutul John al lui Popescu Ion a avut urmatoarele interventii medicale:
1. Enterocolita la data de 04-MAY-09, avand emise facturile:
 1.1. id factura=6, emisa la data de 04-MAY-09
2. Vaccinare Leptospiroza la data de 14-JUN-10, avand emise facturile:
 2.1. id factura=7, emisa la data de 14-JUN-10
3. Rana deschisa la data de 14-AUG-15, avand emise facturile:
 3.1. id factura=104, emisa la data de 13-JUL-21
Animalutul doq al lui Cineva Interesant nu a avut niciodata parte de o procedura medicala
Animalutul Thor al lui Staicu Octavian-Florin a avut urmatoarele interventii medicale:
1. Vaccinare Leptospiroza la data de 13-SEP-20, avand emise facturile:
 1.1. id_factura=106, emisa la data de 12-MAY-21
 1.2. id factura=107, emisa la data de 12-MAY-21
2. Vaccinare Leptospiroza la data de 13-JUL-21, avand emise facturile:
```

2.1. id factura=105, emisa la data de 13-JUL-21

Animalutul Bit [id_animalut=1] al lui Staicu Octavian-Florin este unul dintre cei mai fideli clienti ai nostri: ne-a vizitat de 3 ori :)

Proprietarul animalutului Bit al lui Staicu Octavian-Florin are factura 1 emisa la data de 13-JUL-20, pentru fisa medicala 1, total factura=300

Alerta: raport suma_totala/co prost

Proprietarul animalutului Kid al lui Staicu Adrian are factura 2 emisa la data de 13-JUL-11, pentru fisa medicala 2, total factura=300

Alerta: raport suma totala/co prost

Proprietarul animalutului Kitty al lui Bodea George are factura 3 emisa la data de 13-JUL-16, pentru fisa medicala 3. total factura=300

Alerta: raport suma totala/co prost

Proprietarul animalutului Bit al lui Staicu Octavian-Florin are factura 4 emisa la data de 13-SEP-20, pentru fisa medicala 4, total factura=300

Alerta: raport suma totala/co prost

Proprietarul animalutului Angelina al lui Cineva Interesant are factura 5 emisa la data de 13-JUL-17, pentru fisa medicala 5, total factura=300

Alerta: raport suma totala/co prost

Proprietarul animalutului John al lui Popescu Ion are factura 6 emisa la data de 04-MAY-09, pentru fisa medicala 6, total factura=300

Alerta: raport suma totala/co prost

Proprietarul animalutului John al lui Popescu Ion are factura 7 emisa la data de 14-JUN-10, pentru fisa medicala 7, total factura=300

Alerta: raport suma totala/co prost

Proprietarul animalutului Angelina al lui Cineva Interesant are factura 8 emisa la data de 13-JUL-20, pentru fisa medicala 8, total factura=300

Alerta: raport suma totala/co prost

Proprietarul animalutului Kitty al lui Bodea George are factura 9 emisa la data de 13-DEC-16, pentru fisa medicala 9, total_factura=300

Alerta: raport suma_totala/co prost

Proprietarul animalutului Bit al lui Staicu Octavian-Florin are factura 10 emisa la data de 25-DEC-20, pentru fisa medicala 10, total factura=300

Alerta: raport suma totala/co prost

Proprietarul animalutului John al lui Popescu Ion are factura 104 emisa la data de 14-AUG-15, pentru fisa medicala 11, total_factura=300

Alerta: raport suma totala/co prost

Proprietarul animalutului Thor al lui Staicu Octavian-Florin are mai multe facturi emise la data de 13-SEP-20 Proprietarul animalutului Thor al lui Staicu Octavian-Florin are factura 105 emisa la data de 13-JUL-21, pentru fisa medicala 12, total factura=3000

```
Nu s-a emis nicio factura pentru proprietarul animalutului Idk al lui FARA PROPRIETAR la data de 05-FEB-23 desi a
fost tratat, cu fisa medicala cu id 14
Proprietarul animalutului Idk2 al lui FARA PROPRIETAR are factura 1002 emisa la data de 05-FEB-23, pentru fisa
medicala 15, total factura=499
Alerta: factura emisa pentru animalut fara stapan
1 clienti au facturi de minim 490
suma totala din facturi unice: 6300
Exista 1 clienti:
Bodea George (id: 2)
CREATE OR REPLACE PACKAGE package_clinica_vet_tav14 AS
   TYPE tab imb is TABLE OF NUMBER(10);
   top clienti tab imb := tab imb();
   FUNCTION topclienti RETURN PLS INTEGER;
   PROCEDURE afis_topclienti;
   PROCEDURE afis tratamente animalute;
                                      ANIMALUT.nume%type,
                   nume_proprietar PROPRIETAR.nume%type,
                   prenume_proprietar PROPRIETAR.prenume%type
END package clinica vet tav14;
CREATE OR REPLACE PACKAGE BODY package clinica vet tav14 AS
   FUNCTION topclienti RETURN PLS_INTEGER
        cursor mycursor is select f.id proprietar,
                                          from CHITANTA c
                           from FACTURA_PROPRIETAR f,
```

```
PROPRIETAR p
v cursor refcursor;
           FACTURA_PROPRIETAR.id_factura%type := null;
max_suma pls_integer
          PROPRIETAR.id_proprietar%type
open mycursor;
    fetch mycursor into id, v_cursor;
        if id <> last_id or mycursor%notfound then
            if total_suma > max_suma then
                top_clienti.DELETE;
               max suma := total suma;
            elsif total_suma = max_suma then
                top_clienti.EXTEND;
    exit when mycursor%notfound;
        exit when v_cursor*notfound;
```

```
close mycursor;
    END topclienti;
    PROCEDURE afis tratamente animalute
       cursor mycursor(id_anim IN FISA_MEDICALA.id_animalut%type) is select diag.descriere,
f.data emitere
                                                                                     from FACTURA_PROPRIETAR fp,
                                                                                          FACTURA f
fm.id_fisa)
                                                                      from FISA_MEDICALA fm,
                                                                        and fm.id_animalut = id_anim;
       v cursor
                               DIAGNOSTIC.descriere%type;
       diaq
       data diagnostic
                              FISA_MEDICALA.data_fisa%type;
       id factura
                              FACTURA_PROPRIETAR.id_factura%type;
       data factura
                              FACTURA.data emitere%type;
       id_animalut_max
       nume animalut max
                              ANIMALUT.nume%type;
       nume proprietar max
                              PROPRIETAR.nume%type;
       prenume_proprietar_max PROPRIETAR.prenume%type;
       co max
       faliment EXCEPTION;
        for i in (select a.id animalut, a.nume nume animalut, p.nume nume proprietar, p.prenume prenume proprietar
                      PROPRIETAR p
```

```
open mycursor(i.id animalut);
fetch mycursor into diag, data_diagnostic, v_cursor;
if mycursor%notfound then
   DBMS_OUTPUT.PUT_LINE('Animalutul ' | i.nume_animalut | | ' al lui ' | | i.nume_proprietar | | ' '
                    DBMS_OUTPUT.NEW_LINE();
   DBMS_OUTPUT.PUT_LINE('Animalutul' | i.nume_animalut | | al lui' | i.nume_proprietar | | ' '
                    exit when mycursor%notfound;
      DBMS_OUTPUT.PUT_LINE(mycursor%rowcount | | '. ' | | diag | | ' la data de ' | | data_diagnostic
         fetch v_cursor into id_factura, data_factura;
         exit when v_cursor%notfound;
         ok_factura := 1;
         DBMS_OUTPUT.PUT_LINE(' ' | mycursor%rowcount | | '.' | v_cursor%rowcount | |
                          DBMS_OUTPUT.PUT_LINE(' Nicio factura emisa pentru aceasta fisa medicala');
      fetch mycursor into diag, data diagnostic, v cursor;
   DBMS_OUTPUT.NEW_LINE();
   if mycursor%rowcount > co_max then
```

```
co_max := mycursor%rowcount;
                    id animalut max := i.id animalut;
                    nume animalut max := i.nume animalut;
                   nume_proprietar_max := i.nume_proprietar;
                    prenume_proprietar_max := i.prenume_proprietar;
            close mycursor;
    if co_max = 0 then
       raise faliment;
   DBMS_OUTPUT.PUT_LINE('Animalutul ' | nume_animalut_max | | ' [id_animalut=' | id_animalut_max | | '] al lui
                         nume_proprietar_max || ' ' || prenume_proprietar_max ||
    when faliment then
        DBMS_OUTPUT.PUT_LINE('Nu avem nicium animalut care s-a tratat la noi...');
END afis tratamente animalute;
PROCEDURE afis_topclienti
          PROPRIETAR.nume%type;
   prenum PROPRIETAR.prenume%type;
    co := package clinica vet tav14. TOPCLIENTI;
   DBMS_OUTPUT.PUT_LINE('Exista ' | co | clienti:');
   id := top_clienti.FIRST;
        into num, prenum
        from PROPRIETAR
       DBMS_OUTPUT.PUT_LINE(num | | ' ' | | prenum | | ' (id: ' | | top_clienti(id) | | ')');
```

```
id := top_clienti.NEXT(id);
        DBMS_OUTPUT.PUT_LINE('Animalutele acestora au folosit cel mai des aceste tratamente:');
       package clinica vet tav14.AFIS TRATAMENTE ANIMALUTE();
   end afis topclienti;
END package clinica vet tav14;
   package clinica vet tav14.afis topclienti;
               package_clinica_vet_tav14.afis_topclienti;
[2023-02-06 06:00:43] completed in 24 ms
Exista 1 clienti:
Bodea George (id: 2)
Animalutele acestora au folosit cel mai des aceste tratamente:
Animalutul Kitty al lui Bodea George a avut urmatoarele interventii medicale:

    Vaccinare Leptospiroza la data de 13-JUL-16, avand emise facturile:

  1.1. id_factura=3, emisa la data de 13-JUL-15
2. Rana deschisa la data de 13-DEC-16, avand emise facturile:
  2.1. id_factura=9, emisa la data de 13-DEC-16
Animalutul Kitty [id_animalut=3] al lui Bodea George este unul dintre cei mai fideli clienti ai nostri: ne-a vizitat de 2 ori :)
```

```
type myrec datorii is record
                                          PROPRIETAR.id proprietar%type,
                                          PROPRIETAR.nume%type,
                             prenume_prop PROPRIETAR.prenume%type,
                                          FACTURA PROPRIETAR.id factura%type,
                             suma_platita CHITANTA.suma_platita%type
   type datorii imb is table of myrec datorii;
   FUNCTION timp minim de asteptare(v in out array) RETURN t idx;
   PROCEDURE afis datorii;
END package_clinica_vet_utilitare;
CREATE OR REPLACE PACKAGE BODY package_clinica_vet_utilitare
   FUNCTION timp minim de asteptare(v in out array) RETURN t idx
       poz_min pls_integer;
           raise NO_DATA_FOUND;
           poz min := 1;
                   if v(i) < v(poz min) then
                       poz min := i;
           exit when v(poz_min) = 1001;
```

```
sol(poz min) := co;
       v(poz min) := 1001;
   when NO DATA FOUND then
END timp_minim_de_asteptare;
PROCEDURE afis datorii is
             datorii_imb;
   s dat
   last prop PROPRIETAR.id proprietar%type := null;
    from PROPRIETAR p,
        FACTURA_PROPRIETAR fp,
        FACTURA f,
        CHITANTA C
       if last_prop is null then
               last_prop := my_t(idx).id_prop;
```

```
DBMS_OUTPUT.PUT_LINE(
               if idx is null or my_t(idx).id_prop <> last_prop then
                  DBMS_OUTPUT.PUT_LINE('Total de plata: ' | s_dat);
                  DBMS_OUTPUT.NEW_LINE();
                  s_dat := 0;
                      last_prop := my_t(idx).id_prop;
                      DBMS_OUTPUT.PUT_LINE(
                                 s := my_t(idx).total_fac - nv1(my_t(idx).suma_platita, 0);
          s dat := s dat + s;
              DBMS_OUTPUT.PUT_LINE('La factura ' | | my_t(idx).id_fac | | ' mai are de platit: ' | | s);
          idx := my_t.NEXT(idx);
   end afis_datorii;
END package clinica vet utilitare;
   package clinica vet utilitare.afis datorii;
```

```
package_clinica_vet_utilitare.afis_datorii;
[2023-02-06 06:02:08] completed in 7 ms
Proprietarul Staicu Octavian-Florin (id:1):
La factura 1 mai are de platit: 150
La factura 105 mai are de platit: 3000
Total de plata: 3150
Proprietarul Bodea George (id:2):
La factura 106 mai are de platit: 2400
La factura 107 mai are de platit: 3000
Total de plata: 5400
Proprietarul Staicu Adrian (id:3):
Total de plata: 0
Proprietarul Popescu Ion (id:4):
Total de plata: 0
Proprietarul Cineva Interesant (id:5):
La factura 1000 mai are de platit: 100
La factura 1002 mai are de platit: 499
Total de plata: 598
               v package_clinica_vet_utilitare.array := package_clinica_vet_utilitare.array(7, 1, 2, 5, 3);
               sol package_clinica_vet_utilitare.t_idx;
               sol := package_clinica_vet_utilitare.timp_minim_de_asteptare(v);
              if sol.FIRST is not null then
                   for i in sol.FIRST..sol.LAST
                          DBMS_OUTPUT.PUT_LINE(i || '. ' || sol(i));
                  DBMS_OUTPUT.PUT_LINE('Nicium Client Programat');
[2023-02-06 06:02:18] completed in 7 ms
4. 4
```