

PROIECT  
CLINICĂ  
VETERINARĂ

Student: Staicu Octavian-Florin

Grupa 406 Master Inginerie Software

# *Clinică veterinară*

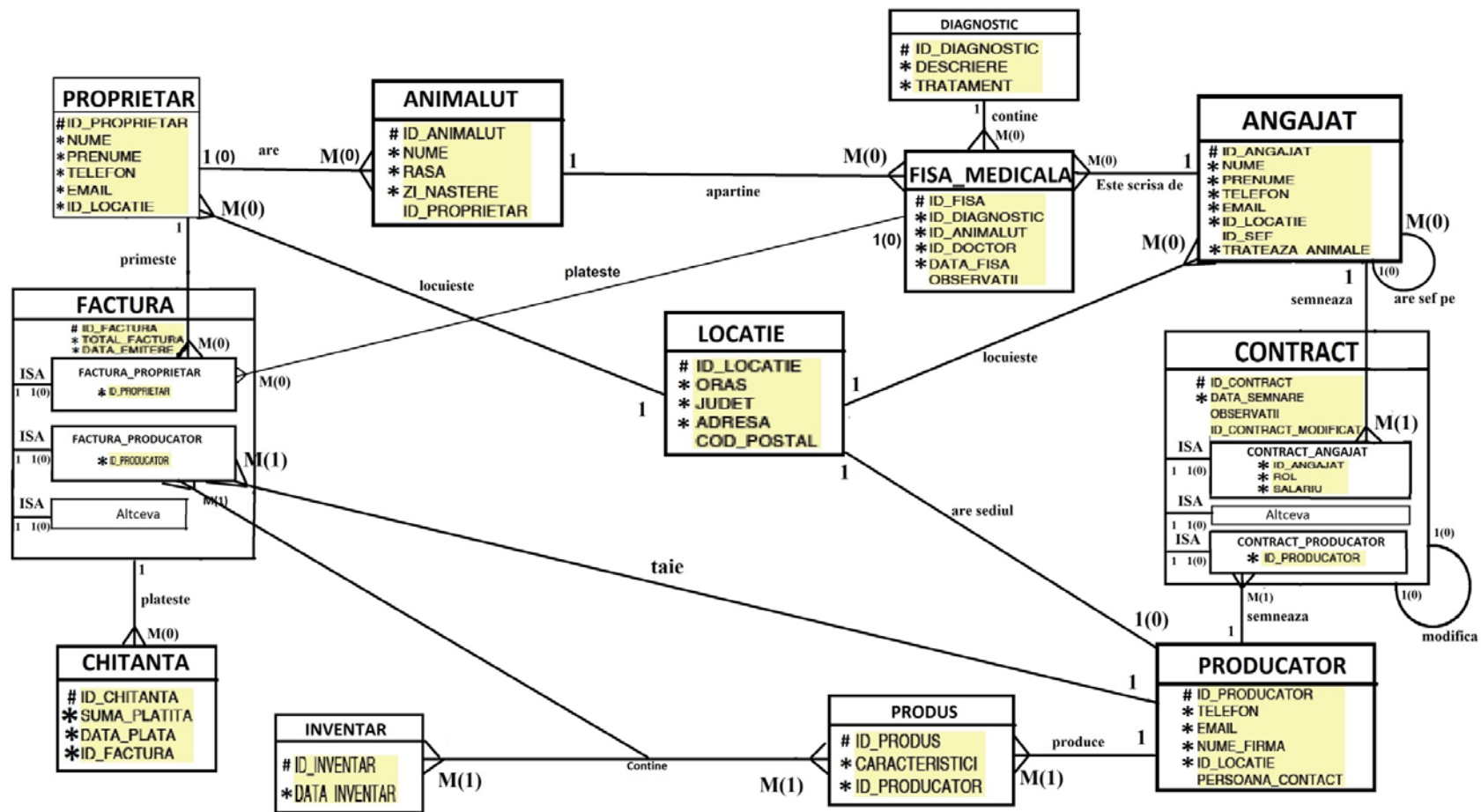
---

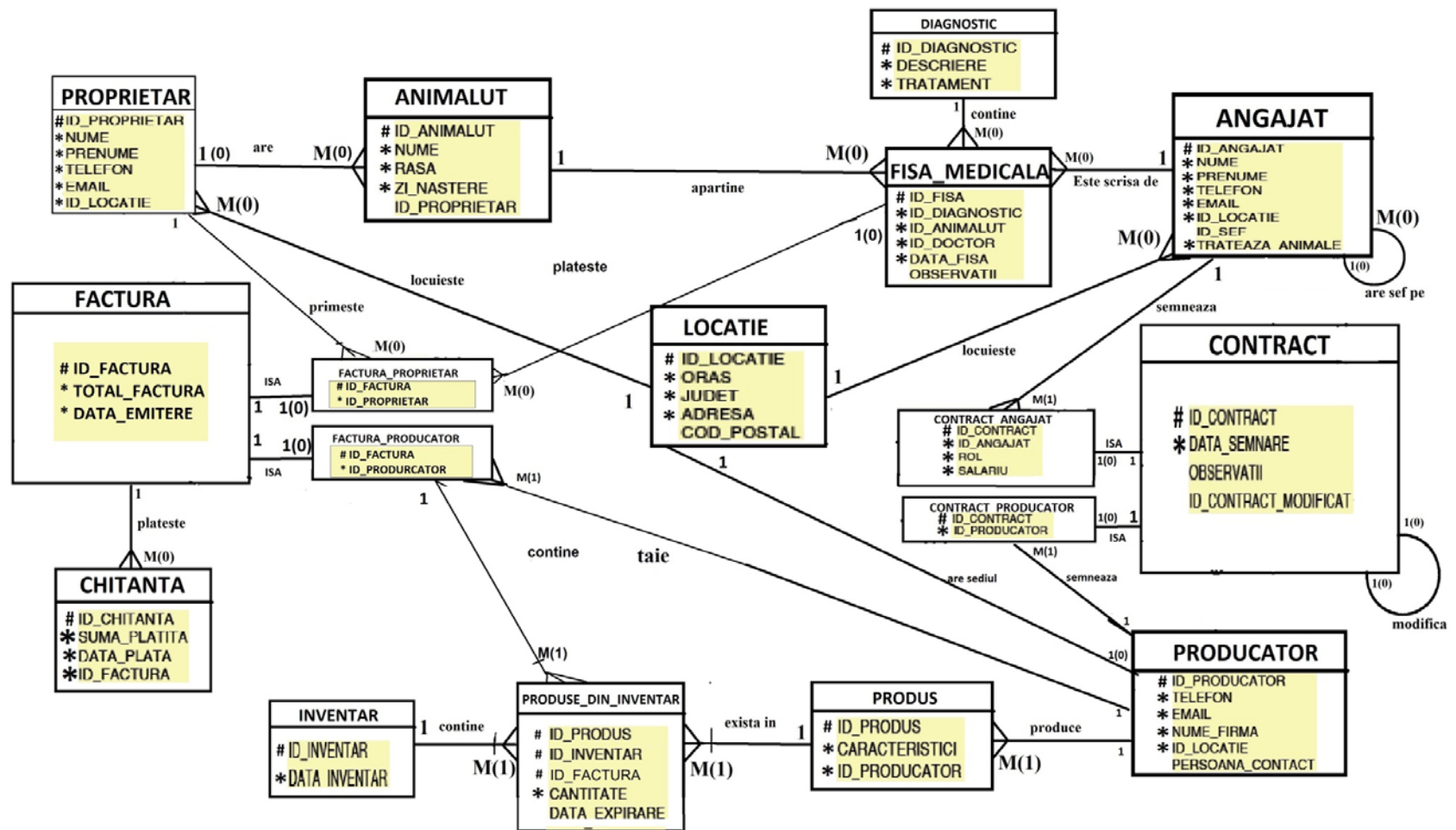
Acest proiect se adresează unei clinici veterinare

## Descrierea Regulilor de Bussines

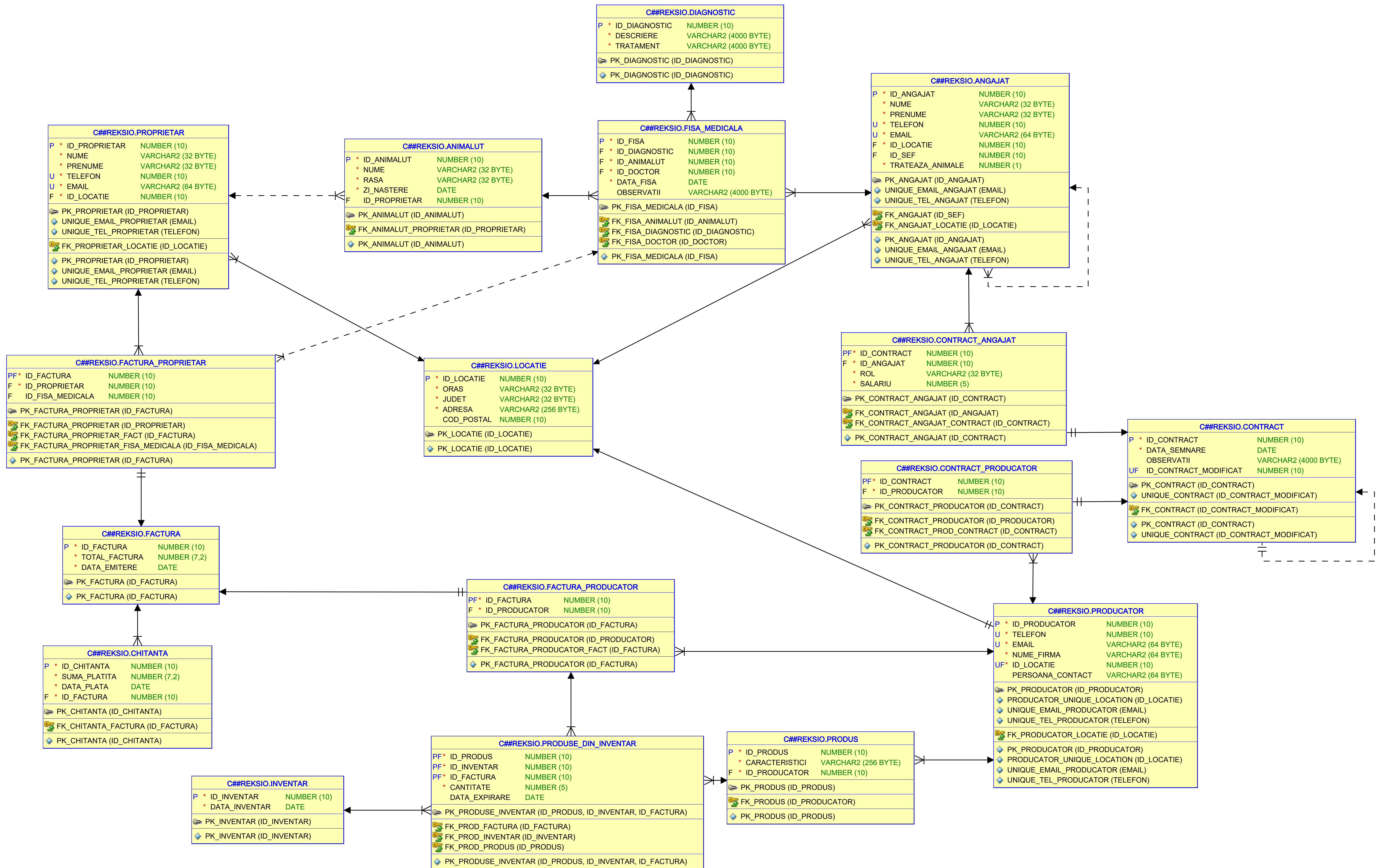
- Un proprietar are mai multe animale. Când un animaluț este tratat la clinică, i se face o nouă fișă medicală (relație de tip 3 între animal, diagnostic și un angajat care are voie să trateze animale). Pe fișa medicală apare un diagnostic, pentru care se știe tratamentul (sau se adaugă în tabela diagnostic). Un animaluț 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 măririi de salariu etc., se face un nou contract, care, eventual, poate modifica un contract anterior). Doar unii angajați au voie să trateze animale (precizat în 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 animale ).
- 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 animale. Aceștia le pot achita eșalonat, deci există mai multe chitanțe pentru o singură factură. Pentru o fișă medicală se pot emite mai multe facturi.
- 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.*









```
-- Ex. 5. Diagramele pe care le-am folosit le-am conceput de la inceput in FN3.
-- Fn1 ar fi fost incalcat daca spre exemplu ar fi existat in tabela proprietar atributul animalute (ar fi existat un atribut cu valori multiple).
-- Fn2 ar fi fost incalcat spre exemplu daca tabela animalut ar fi avut attributele tabelii diagnostic. Astfel, ar fi trebuit sa duplicam intrarile animalutului (nume, rasa, zi_nastere, id_proprietar) ori de cate ori se punea un nou daignostic. Cu alte cuvinte, ar fi existat o asa zisa cheia primara compusa din id_proprietar si id_animalut, iar fn2 spune ca orice atribut care nu face parte din cheia primara trebuie sa depinda de intreaga cheie (spre exemplu, rasa unui catel nu depinda de id_proprietar).
-- Fn3 ar fi fost incalcat spre exemplu daca in tabela proprietar am fi avut si toate attributele tabelii locatie, caci asa am avea dependenta tranzitiva intre acele attribute si id-ul locatiei (pt fiecare proprietar, am avea de fiecare data valori duplicate pt oras, judet, adresa, cod_postal).
```

```
--6
```

```
CREATE TABLE LOCATIE
```

```
(
    id_locatie NUMBER(10),
    oras        VARCHAR2(32) NOT NULL,
    judet       VARCHAR2(32) NOT NULL,
    adresa      VARCHAR2(256) NOT NULL,
    cod_postal  NUMBER(10),
    CONSTRAINT pk_locatie PRIMARY KEY (id_locatie)
);
```

```
CREATE TABLE PROPRIETAR
```

```
(
    id_proprietar NUMBER(10),
    nume           VARCHAR2(32) NOT NULL,
    prenume       VARCHAR2(32) NOT NULL,
    telefon       NUMBER(10) NOT NULL,
    email         VARCHAR2(64) NOT NULL,
    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)
);
```

```
CREATE TABLE ANIMALUT
(
    id_animalut    NUMBER(10),
    nume           VARCHAR2(32) NOT NULL,
    rasa           Varchar2(32) NOT NULL,
    zi_nastere     DATE          NOT NULL,
    id_proprietar  NUMBER(10),
    CONSTRAINT pk_animalut PRIMARY KEY (id_animalut),
    CONSTRAINT fk_animalut_proprietar FOREIGN KEY (id_proprietar) REFERENCES PROPRIETAR (id_proprietar)
);
```

```
CREATE TABLE DIAGNOSTIC
(
    id_diagnostic  NUMBER(10),
    descriere      VARCHAR2(4000) NOT NULL,
    tratament      VARCHAR2(4000) NOT NULL,
    CONSTRAINT pk_diagnostic PRIMARY KEY (id_diagnostic)
);
```

```
CREATE TABLE ANGAJAT
(
    id_angajat     NUMBER(10),
    nume           VARCHAR2(32) NOT NULL,
    prenume        VARCHAR2(32) NOT NULL,
    telefon        NUMBER(10)   NOT NULL,
    email          VARCHAR2(64) NOT NULL,
    id_locatie     NUMBER(10)   NOT NULL,
    id_sef         NUMBER(10),
    trateaza_animale NUMBER(1)   NOT NULL,
    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
(
```

```

    id_fisa          NUMBER(10),
    id_diagnostic    NUMBER(10) NOT NULL,
    id_animalut      NUMBER(10) NOT NULL,
    id_doctor        NUMBER(10) NOT NULL,
    data_fisa        DATE          NOT NULL,
    observatii       VARCHAR2(4000),
    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
(
    id_producator     NUMBER(10),
    telefon           NUMBER(10)   NOT NULL,
    email             VARCHAR2(64) NOT NULL,
    nume_firma        VARCHAR2(64) NOT NULL,
    id_locatie        NUMBER(10)   NOT NULL,
    persoana_contact  VARCHAR2(64),
    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)
);

CREATE TABLE CONTRACT
(
    id_contract       NUMBER(10),
    data_semnaire     DATE NOT NULL,
    observatii        VARCHAR2(4000),
    id_contract_modificat NUMBER(10),
    CONSTRAINT pk_contract PRIMARY KEY (id_contract),
    CONSTRAINT fk_contract FOREIGN KEY (id_contract_modificat) REFERENCES CONTRACT (id_contract),
    CONSTRAINT unique_contract UNIQUE (id_contract_modificat)
);

CREATE TABLE CONTRACT_ANGAJAT

```



```

(
    id_contract NUMBER(10),
    id_angajat  NUMBER(10) NOT NULL,
    rol         VARCHAR2(32) NOT NULL,
    salariu     NUMBER(5) NOT NULL,
    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
(
    id_contract  NUMBER(10),
    id_producator NUMBER(10) NOT NULL,
    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
(
    id_factura  NUMBER(10),
    total_factura NUMBER(7, 2) NOT NULL,
    data_emitere DATE NOT NULL,
    CONSTRAINT pk_factura PRIMARY KEY (id_factura)
);

CREATE TABLE FACTURA_PROPRIETAR
(
    id_factura  NUMBER(10),
    id_proprietar  NUMBER(10) NOT NULL,
    id_fisa_medicala NUMBER(10),
    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

```

```

(
    id_factura    NUMBER(10),
    id_producator NUMBER(10) NOT NULL,
    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
(
    id_chitanta    NUMBER(10),
    suma_platita   NUMBER(7, 2) NOT NULL,
    data_plata     DATE          NOT NULL,
    id_factura     NUMBER(10)    NOT NULL,
    CONSTRAINT pk_chitanta PRIMARY KEY (id_chitanta),
    CONSTRAINT fk_chitanta_factura FOREIGN KEY (id_factura) references FACTURA (id_factura)
);

CREATE TABLE PRODUS
(
    id_produs      NUMBER(10),
    caracteristici VARCHAR2(256) NOT NULL,
    id_producator   NUMBER(10)    NOT NULL,
    CONSTRAINT pk_produs PRIMARY KEY (id_produs),
    CONSTRAINT fk_produs FOREIGN KEY (id_producator) REFERENCES PRODUCATOR (id_producator)
);

CREATE TABLE INVENTAR
(
    id_inventar    NUMBER(10),
    data_inventar  DATE NOT NULL,
    CONSTRAINT pk_inventar PRIMARY KEY (id_inventar)
);

CREATE TABLE PRODUSE_DIN_INVENTAR
(
    id_produs      NUMBER(10) NOT NULL,
    id_inventar    NUMBER(10) NOT NULL,

```

```
id_factura    NUMBER(10) NOT NULL,  
cantitate     NUMBER(5)  NOT NULL,  
data_expirare DATE,  
CONSTRAINT pk_produce_inventar PRIMARY KEY (id_produș, id_inventar, id_factura),  
CONSTRAINT fk_prod_inventar FOREIGN KEY (id_inventar) REFERENCES INVENTAR (id_inventar),  
CONSTRAINT fk_prod_produș FOREIGN KEY (id_produș) REFERENCES PRODUS (id_produș),  
CONSTRAINT fk_prod_factura FOREIGN KEY (id_factura) REFERENCES FACTURA_PRODUCATOR (id_factura)  
);  
  
commit;
```

```
C##REKSI0> CREATE TABLE LOCATIE
(
    id_locatie NUMBER(10),
    oras        VARCHAR2(32) NOT NULL,
    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
C##REKSI0> CREATE TABLE PROPRIETAR
(
    id_proprietar NUMBER(10),
    nume           VARCHAR2(32) NOT NULL,
    prenume        VARCHAR2(32) NOT NULL,
    telefon        NUMBER(10) NOT NULL,
    email          VARCHAR2(64) NOT NULL,
    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
C##REKSI0> CREATE TABLE ANIMALUT
(
    id_animalut  NUMBER(10),
    nume         VARCHAR2(32) NOT NULL,
    rasa         VARCHAR2(32) NOT NULL,
    zi_nastere   DATE NOT NULL,
    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
```

--5

```
INSERT INTO LOCATIE(id_locatie, oras, judet, adresa)
VALUES (1, 'Bucuresti', 'Sector 2', 'soseaua colentina nr 2');
INSERT INTO LOCATIE(id_locatie, oras, judet, adresa, cod_postal)
VALUES (2, 'Bucuresti', 'Sector 1', 'soseaua Aviatorilor nr 1', 12345);
INSERT INTO LOCATIE(id_locatie, oras, judet, adresa)
VALUES (3, 'Pantelimon', 'Ilfov', 'soseaua dn2');
INSERT INTO LOCATIE(id_locatie, oras, judet, adresa)
VALUES (4, 'Cluj-Napoca', 'Cluj', 'soseaua jmekera');
INSERT INTO LOCATIE(id_locatie, oras, judet, adresa)
VALUES (5, 'Iasi', 'Iasi', 'soseaua industriei');
INSERT INTO LOCATIE(id_locatie, oras, judet, adresa)
VALUES (6, 'Timisoara', 'Timis', 'soseaua intreprinderii');
INSERT INTO LOCATIE(id_locatie, oras, judet, adresa)
VALUES (7, 'Constanta', 'Constanta', 'soseaua antreprenorilor');
INSERT INTO LOCATIE(id_locatie, oras, judet, adresa)
VALUES (8, 'Giurgiu', 'Giurgiu', 'soseaua giurgiului');
INSERT INTO LOCATIE(id_locatie, oras, judet, adresa)
VALUES (9, 'Severin', 'Mehedinti', 'traian');

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)
VALUES (2, 'Bodea', 'George', 'george@yahoo.com', 0737392518, 2);
INSERT INTO PROPRIETAR(id_proprietar, nume, prenume, email, telefon, id_locatie)
VALUES (3, 'Staicu', 'Adrian', 'adi@gmail.com', 0736659441, 1);
INSERT INTO PROPRIETAR(id_proprietar, nume, prenume, email, telefon, id_locatie)
VALUES (4, 'Popescu', 'Ion', 'popescu.ion@gmail.com', 0771654789, 3);
INSERT INTO PROPRIETAR(id_proprietar, nume, prenume, email, telefon, id_locatie)
VALUES (5, 'Cineva', 'Interesant', 'jmekerie@my.fmi.unibuc.ro', 0777111999, 4);
INSERT INTO PROPRIETAR(id_proprietar, nume, prenume, email, telefon, id_locatie)
VALUES (6, 'Fara', 'Animalut', 'cumparator@my.fmi.unibuc.ro', 0729123123, 2);

INSERT INTO ANIMALUT(id_animalut, nume, rasa, zi_nastere, id_proprietar)
VALUES (1, 'Bit', 'Boxer', to_date('05/05/2017', 'DD/MM/YYYY'), 1);
INSERT INTO ANIMALUT(id_animalut, nume, rasa, zi_nastere, id_proprietar)
VALUES (2, 'Kid', 'Ciobanesc German', to_date('13/07/2010', 'DD/MM/YYYY'), 3);
INSERT INTO ANIMALUT(id_animalut, nume, rasa, zi_nastere, id_proprietar)
VALUES (3, 'Kitty', 'Birmaneza', to_date('25/12/2015', 'DD/MM/YYYY'), 2);
```

```

INSERT INTO ANIMALUT(id_animalut, nume, rasa, zi_nastere, id_proprietar)
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)
VALUES (5, 'John', 'Metis', to_date('05/12/2008', 'DD/MM/YYYY'), 4);
insert into ANIMALUT(id_animalut, nume, rasa, zi_nastere, id_proprietar)
VALUES (6, 'dog', 'Boxer', to_date('09.01.2021', 'dd.mm.yyyy'), 5);
insert into ANIMALUT(id_animalut, nume, rasa, zi_nastere, id_proprietar)
VALUES (13, 'Thor', 'Golden Retriever', to_date('13.07.2020', 'dd.mm.yyyy'), 1);
insert 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)
VALUES (14, 'Idk2', 'Comuna', to_date('01.02.2023', 'dd.mm.yyyy'));

INSERT INTO DIAGNOSTIC(id_diagnostic, descriere, tratament)
VALUES (1, 'Vaccinare Leptospiroza', 'Vaccin Leptospiroza');
INSERT INTO DIAGNOSTIC(id_diagnostic, descriere, tratament)
VALUES (2, 'Enterocolita', 'Enteroguard');
INSERT INTO DIAGNOSTIC(id_diagnostic, descriere, tratament)
VALUES (3, 'Rana deschisa', 'Spray Cicatrizant');
INSERT INTO DIAGNOSTIC(id_diagnostic, descriere, tratament)
VALUES (4, 'durere abdominala', 'ecografie');
INSERT INTO DIAGNOSTIC(id_diagnostic, descriere, tratament)
VALUES (5, 'capusa', 'eliminare capusa');

INSERT INTO ANGAJAT(id_angajat, nume, prenume, telefon, email, id_locatie, trateaza_animale)
VALUES (1, 'Staicu', 'Clara', 0722012983, 'dr.staicu-clara@clinicatareanimale.ro', 1, 1);
INSERT INTO ANGAJAT(id_angajat, nume, prenume, telefon, email, id_sef, id_locatie, trateaza_animale)
VALUES (2, 'Popa', 'Adi', 0723145867, 'asistent.popa-adi@clinicatareanimale.ro', 1, 2, 1);
INSERT INTO ANGAJAT(id_angajat, nume, prenume, telefon, email, id_locatie, trateaza_animale)
VALUES (3, 'Spalatorul', 'Pop', 0711012070, 'spalatorul.pop@gmail.com', 3, 0);
INSERT INTO ANGAJAT(id_angajat, nume, prenume, telefon, email, id_locatie, trateaza_animale)
VALUES (4, 'Ana', 'Maria', 0723012982, 'dr.ana-maria@clinicatareanimale.ro', 1, 1);
INSERT INTO ANGAJAT(id_angajat, nume, prenume, telefon, email, id_sef, id_locatie, trateaza_animale)
VALUES (5, 'Numinescu', 'George', 0723145868, 'asistent.numinescu-george@clinicatareanimale.ro', 4, 2, 1);

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)
VALUES (4, 3, 1, 2, to_date('13.09.2020', 'dd.mm.yyyy'), 'Rana deschisa in zona botului');
INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa, observatii)
VALUES (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)
VALUES (6, 2, 5, 1, to_date('04.05.2009', 'dd.mm.yyyy'));
INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa, observatii)
VALUES (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)
VALUES (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)
VALUES (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)
VALUES (10, 2, 1, 1, to_date('25.12.2020', 'dd.mm.yyyy'), 'Cainele a mancat din pregaritile de Craciun');
INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
VALUES (11, 3, 5, 1, to_date('14.08.2015', 'dd.mm.yyyy'));
insert into FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
VALUES (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'));
INSERT 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');
INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa, observatii)
VALUES (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)
VALUES (1, 'Mancare Catei SA', 0767582563, 'comenzi@mancarecatei.ro', 5);
INSERT INTO PRODUCATOR(id_producator, nume_firma, telefon, email, persoana_contact, id_locatie)
VALUES (2, 'Aparatura Animale SRL', 0798712365, 'aparatus.ion@aparaturaaanimale.ro', 'Aparatus Ion', 6);
INSERT INTO PRODUCATOR(id_producator, nume_firma, telefon, email, persoana_contact, id_locatie)
VALUES (3, 'Vaccinam cu Iubire', 0712345678, 'bussines@vacciniubire.ro', 'Ana Vacix', 7);
INSERT INTO PRODUCATOR(id_producator, nume_firma, telefon, email, persoana_contact, id_locatie)
VALUES (4, 'Producator 4', 0700000000, 'producator4@bussines.ro', 'Producator 4 contact', 8);
INSERT INTO PRODUCATOR(id_producator, nume_firma, telefon, email, persoana_contact, id_locatie)
VALUES (5, 'Producator 5', 0711111111, 'bussines@producator5.ro', 'contact prod 5', 9);

INSERT INTO CONTRACT(id_contract, data_semnare)

```

```
VALUES (1, to_date('02.01.2005', 'dd.mm.yyyy'));
INSERT INTO CONTRACT(id_contract, data_semnare)
VALUES (2, to_date('13.07.2004', 'dd.mm.yyyy'));
INSERT INTO CONTRACT(id_contract, data_semnare)
VALUES (3, to_date('09.09.2005', 'dd.mm.yyyy'));
INSERT INTO CONTRACT(id_contract, data_semnare)
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'));
INSERT INTO CONTRACT(id_contract, data_semnare)
VALUES (6, to_date('02.01.2005', 'dd.mm.yyyy'));
INSERT INTO CONTRACT(id_contract, data_semnare, observatii, id_contract_modificat)
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)
VALUES (1, 1, 'doctor veterinar', 6000);
INSERT INTO CONTRACT_ANGAJAT(id_contract, id_angajat, rol, salariu)
VALUES (2, 2, 'asistent veterinar', 3000);
INSERT INTO CONTRACT_ANGAJAT(id_contract, id_angajat, rol, salariu)
VALUES (3, 3, 'paznic', 1500);
INSERT INTO CONTRACT_ANGAJAT(id_contract, id_angajat, rol, salariu)
VALUES (7, 2, 'doctor veterinar', 6000);
INSERT INTO CONTRACT_ANGAJAT(id_contract, id_angajat, rol, salariu)
VALUES (8, 4, 'doctor veterinar', 6000);
INSERT INTO CONTRACT_ANGAJAT(id_contract, id_angajat, rol, salariu)
VALUES (9, 5, 'asistent veterinar', 3000);

INSERT INTO CONTRACT_PRODUCATOR(id_contract, id_producator)
VALUES (4, 1);
INSERT INTO CONTRACT_PRODUCATOR(id_contract, id_producator)
VALUES (5, 2);
```

```
INSERT INTO CONTRACT_PRODUCATOR(id_contract, id_producator)
VALUES (6, 3);
INSERT INTO CONTRACT_PRODUCATOR(id_contract, id_producator)
VALUES (10, 4);
INSERT INTO CONTRACT_PRODUCATOR(id_contract, id_producator)
VALUES (11, 5);

INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (1, 300, to_date('13.07.2020', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (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'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (5, 300, to_date('13.07.2017', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (6, 300, to_date('04.05.2009', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (7, 300, to_date('14.06.2010', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (8, 300, to_date('13.07.2020', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (9, 300, to_date('13.12.2016', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (10, 300, to_date('25.12.2020', 'dd.mm.yyyy'));
insert into FACTURA(id_factura, total_factura, data_emitere)
VALUES (104, 300, to_date('13.07.2021', 'dd.mm.yyyy'));
insert into FACTURA(id_factura, total_factura, data_emitere)
VALUES (105, 3000, to_date('13.07.2021', 'dd.mm.yyyy'));
insert into FACTURA(id_factura, total_factura, data_emitere)
VALUES (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)
VALUES (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)
VALUES (11, 1500, to_date('25.10.2009', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (12, 4325, to_date('13.11.2009', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (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'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (15, 4725, to_date('13.04.2013', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (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'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (18, 7500, to_date('25.09.2019', 'dd.mm.yyyy'));
INSERT INTO FACTURA(ID_FACTURA, TOTAL_FACTURA, DATA_EMITERE)
VALUES (19, 9256.35, to_date('12.04.2019', 'dd.mm.yyyy'));

INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
VALUES (1, 1, 1);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
VALUES (2, 3, 2);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
VALUES (3, 2, 3);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
VALUES (4, 1, 4);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
VALUES (5, 5, 5);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
VALUES (6, 4, 6);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
VALUES (7, 4, 7);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
VALUES (8, 5, 8);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar, id_fisa_medicala)
VALUES (9, 2, 9);
```

```
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_propietar, id_fisa_medicala)
VALUES (10, 1, 10);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_propietar, id_fisa_medicala)
VALUES (104, 1, 11);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_propietar, id_fisa_medicala)
VALUES (105, 1, 12);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_propietar, id_fisa_medicala)
VALUES (106, 2, 13);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_propietar, id_fisa_medicala)
VALUES (107, 2, 13);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_propietar)
VALUES (1000, 5);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_propietar, id_fisa_medicala)
VALUES (1002, 5, 15);

INSERT INTO FACTURA_PRODUCATOR(id_factura, id_producator)
VALUES (11, 1);
INSERT INTO FACTURA_PRODUCATOR(id_factura, id_producator)
VALUES (12, 2);
INSERT INTO FACTURA_PRODUCATOR(id_factura, id_producator)
VALUES (13, 3);
INSERT INTO FACTURA_PRODUCATOR(id_factura, id_producator)
VALUES (14, 1);
INSERT INTO FACTURA_PRODUCATOR(id_factura, id_producator)
VALUES (15, 2);
INSERT INTO FACTURA_PRODUCATOR(id_factura, id_producator)
VALUES (16, 3);
INSERT INTO FACTURA_PRODUCATOR(id_factura, id_producator)
VALUES (17, 1);
INSERT INTO FACTURA_PRODUCATOR(id_factura, id_producator)
VALUES (18, 2);
INSERT INTO FACTURA_PRODUCATOR(id_factura, id_producator)
VALUES (19, 3);

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);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
```

```
VALUES (3, 300, to_date('13.08.2015', 'dd.mm.yyyy'), 3);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (4, 300, to_date('03.01.2021', 'dd.mm.yyyy'), 4);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (5, 150, to_date('13.07.2017', 'dd.mm.yyyy'), 5);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (6, 75.25, to_date('13.08.2017', 'dd.mm.yyyy'), 5);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (7, 75.75, to_date('13.09.2017', 'dd.mm.yyyy'), 5);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (8, 100, to_date('13.07.2009', 'dd.mm.yyyy'), 6);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (9, 200, to_date('17.08.2009', 'dd.mm.yyyy'), 6);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (10, 300, to_date('13.07.2010', 'dd.mm.yyyy'), 7);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (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);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (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);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (15, 4325, to_date('25.10.2009', 'dd.mm.yyyy'), 12);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (16, 3000, to_date('25.07.2009', 'dd.mm.yyyy'), 13);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (17, 3000, to_date('25.09.2009', 'dd.mm.yyyy'), 13);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (18, 1400, to_date('25.10.2014', 'dd.mm.yyyy'), 14);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (19, 4725, to_date('13.05.2013', 'dd.mm.yyyy'), 15);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (20, 3025, to_date('25.09.2014', 'dd.mm.yyyy'), 16);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (21, 3000, to_date('01.11.2014', 'dd.mm.yyyy'), 16);
INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (22, 4500, to_date('25.10.2019', 'dd.mm.yyyy'), 18);
```



```
insert into CHITANTA(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)
VALUES (106, 600, to_date('13.07.2021', 'dd.mm.yyyy'), 106);

INSERT INTO PRODUS(id_produs, caracteristici, id_producator)
VALUES (1, 'boabe de caini cu carne de vita si somon', 1);
INSERT INTO PRODUS(id_produs, caracteristici, id_producator)
VALUES (2, 'boabe de pisici cu ulei de peste', 1);
INSERT INTO PRODUS(id_produs, caracteristici, id_producator)
VALUES (3, 'aparat radiografie', 2);
INSERT INTO PRODUS(id_produs, caracteristici, id_producator)
VALUES (4, 'stetoscop', 2);
INSERT INTO PRODUS(id_produs, caracteristici, id_producator)
VALUES (5, 'masa operatii', 2);
INSERT INTO PRODUS(id_produs, caracteristici, id_producator)
VALUES (6, 'vaccin leptospiroza', 3);
INSERT INTO PRODUS(id_produs, caracteristici, id_producator)
VALUES (7, 'vaccin hexavalent', 3);

INSERT INTO INVENTAR(id_inventar, data_inventar)
VALUES (1, to_date('05.01.2010', 'dd.mm.yyyy'));
INSERT INTO INVENTAR(id_inventar, data_inventar)
VALUES (2, to_date('05.01.2015', 'dd.mm.yyyy'));
INSERT INTO INVENTAR(id_inventar, data_inventar)
VALUES (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)
VALUES (5, to_date('05.01.2022', 'dd.mm.yyyy'));

INSERT INTO PRODUSE_DIN_INVENTAR(id_produs, id_inventar, id_factura, cantitate, data_expirare)
VALUES (1, 1, 11, 10, to_date('05.01.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produs, id_inventar, id_factura, cantitate, data_expirare)
VALUES (2, 1, 12, 15, to_date('05.01.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produs, id_inventar, id_factura, cantitate, data_expirare)
VALUES (1, 2, 13, 20, to_date('05.01.2016', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produs, id_inventar, id_factura, cantitate, data_expirare)
```

```
VALUES (2, 2, 14, 25, to_date('05.01.2016', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate, data_expirare)
VALUES (1, 3, 15, 35, to_date('05.03.2023', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate, data_expirare)
VALUES (2, 3, 16, 30, to_date('05.01.2021', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate)
VALUES (3, 1, 17, 1);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate)
VALUES (3, 2, 18, 1);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate)
VALUES (3, 3, 19, 2);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate)
VALUES (4, 1, 11, 5);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate)
VALUES (4, 2, 12, 10);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate)
VALUES (4, 3, 13, 15);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate)
VALUES (5, 1, 14, 1);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate)
VALUES (5, 2, 15, 1);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate)
VALUES (5, 3, 16, 2);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate, data_expirare)
VALUES (6, 1, 17, 100, to_date('13.07.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate, data_expirare)
VALUES (7, 1, 18, 150, to_date('13.07.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate, data_expirare)
VALUES (6, 2, 19, 100, to_date('13.07.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate, data_expirare)
VALUES (7, 2, 11, 150, to_date('13.07.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate, data_expirare)
VALUES (6, 3, 12, 234, to_date('13.07.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, id_factura, cantitate, data_expirare)
VALUES (7, 3, 13, 512, to_date('13.07.2011', 'dd.mm.yyyy'));

commit;
```

Output Result 160 Tx ✓ ↺

[2023-08-28 22:47:47] 1 row affected in 5 ms

```
C##REKSI0> INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa, observatii)
VALUES (10, 2, 1, 1, to_date('25.12.2020', 'dd.mm.yyyy'), 'Cainele a mancat din pregaritile de Craciun')
```

[2023-08-28 22:47:48] 1 row affected in 5 ms

```
C##REKSI0> INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
VALUES (11, 3, 5, 1, to_date('14.08.2015', 'dd.mm.yyyy'))
```

[2023-08-28 22:47:48] 1 row affected in 5 ms

```
C##REKSI0> insert into FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
VALUES (13, 1, 13, 1, to_date('13.09.2020', 'dd.mm.yyyy'))
```

[2023-08-28 22:47:48] 1 row affected in 8 ms

```
C##REKSI0> 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'))
```

[2023-08-28 22:47:48] 1 row affected in 4 ms

```
C##REKSI0> INSERT 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')
```

[2023-08-28 22:47:48] 1 row affected in 8 ms

```
C##REKSI0> INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa, observatii)
VALUES (15, 3, 14, 1, to_date('05.02.2023', 'dd.mm.yyyy'), 'catelus fara stapan')
```

[2023-08-28 22:47:48] 1 row affected in 5 ms

--7 SQL

-- 7.1

-- afisati pt fiecare contract de angajat, al catelea contract aditional este si ce contract modifica

```
SELECT ctr.id_contract, ctr.id_angajat, LEVEL depth, PRIOR id_contract
```

```
FROM (select c.id_contract, c.id_contract_modificat, a.id_angajat
```

```
      from CONTRACT_ANGAJAT a,
```

```
      CONTRACT c
```

```
      where c.id_contract = a.id_contract) ctr
```

```
START WITH ctr.ID_CONTRACT_MODIFICAT IS NULL
```

```
CONNECT BY PRIOR ctr.id_contract = ctr.id_contract_modificat;
```

Output Result 2/3

	ID_CONTRACT	ID_ANGAJAT	DEPTH	PRIORID_CONTRACT
1	1	1	1	<null>
2	2	2	1	<null>
3	7	2	2	2
4	3	3	1	<null>
5	8	4	1	<null>
6	9	5	1	<null>

-- 7.2

-- pt fiecare animalut care a fost tratat in anul 2020, afisati nr de consultatii

```
select fm.id_animalut, count(*)
```

```
from ANIMALUT a,
```

```
      FISA_MEDICALA fm
```

```
where a.id_animalut = fm.id_animalut
```

```
      and TO_CHAR(fm.DATA_FISA, 'YYYY') like '2020'
```

```
group by fm.id_animalut;
```

	ID_ANIMALUT	COUNT(*)
1	1	3
2	4	1
3	13	1

-- 7.3

```
-- sa se afiseze pt fiecare animalut, suma total cheltuita pe consultari
-- a caror factura a fost emisa in aceeasi zi cu consultatia pe numele proprietarului
-- si numarul acestora
select a.id_proprietar, a.id_animalut, sum(f.total_factura), count(*)
from FACTURA_PROPRIETAR fp,
     FACTURA f,
     PROPRIETAR p,
     ANIMALUT a
where fp.id_factura = f.id_factura
     and fp.id_proprietar = p.id_proprietar
     and p.id_proprietar = a.id_proprietar
     and fp.id_fisa_medicala in (select fm.id_fisa
                                from FISA_MEDICALA fm
                                where fm.id_animalut = a.id_animalut
                                   and f.data_emitere = fm.data_fisa)
group by a.id_proprietar, a.id_animalut
order by id_proprietar;
```

	ID_PROPRIETAR	ID_ANIMALUT	SUM(F.TOTAL_FACTURA)	COUNT(*)
1	1	1	900	3
2	1	13	3000	1
3	2	3	300	1
4	3	2	300	1
5	4	5	600	2
6	5	4	600	2

```
-- 7.4
-- afisati fiecare factura ce tip are cu majuscule
-- daca nu are niciun tip(nu e nici producator, nici proprietar), scrie custom
with f as (select f.id_factura, f.data_emitere, f.total_factura, 'proprietar' tip
           from FACTURA f,
           FACTURA_PROPRIETAR fp
           where f.id_factura = fp.id_factura
          union
           select f.id_factura, f.data_emitere, f.total_factura, 'producator' tip
           from FACTURA f,
           FACTURA_PRODUCATOR fp
           where f.id_factura = fp.id_factura)
```

```
select fi.id_factura, fi.data_emitere, fi.total_factura, UPPER(NVL(f.tip, 'custom')) tip1
from f,
     FACTURA fi
where fi.id_factura = f.id_factura(+)
order by tip1, fi.total_factura, fi.data_emitere;
```

	ID_FACTURA	DATA_EMITERE	TOTAL_FACTURA	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	6	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	9	2016-12-13	300.00	PROPRIETAR
17	5	2017-07-13	300.00	PROPRIETAR
18	8	2020-07-13	300.00	PROPRIETAR
19	1	2020-07-13	300.00	PROPRIETAR
20	4	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



```
-- 7.5
-- afisati toti proprietarii si toti cainii acestora
-- inclusiv proprietarii fara caini si cainii fara stapan
select *
from PROPRIETAR p
FULL OUTER JOIN ANIMALUT a ON p.id_proprietar = a.id_proprietar
order by p.id_proprietar;
```

	ID PROPRIETAR	NUME	PRENUME	TELEFON	EMAIL	ID LOCATIE	ID ANIMALUT	NOME	RASA	ZI NASTERE	ID PROPRIETAR
1	1	Staicu	Octavian-Florin	736659442	octavian.staicu@fmi.unibuc.ro	1	1	Bit	Boxer	2017-05-05	1
2	1	Staicu	Octavian-Florin	736659442	octavian.staicu@fmi.unibuc.ro	1	13	Thor	Golden Retriever	2020-07-13	1
3	2	Bodea	George	737392518	george@yahoo.com	2	3	Kitty	Birmaneză	2015-12-25	2
4	3	Staicu	Adrian	736659441	adig@gmail.com	1	2	Kid	Ciobanesc German	2019-07-13	3
5	4	Popescu	Ion	771654789	popescu.ion@gmail.com	3	5	John	Metis	2008-12-05	4
6	5	Cineva	Interesant	777111999	jmeckerie@my.fmi.unibuc.ro	4	6	dog	Boxer	2021-01-09	5
7	5	Cineva	Interesant	777111999	jmeckerie@my.fmi.unibuc.ro	4	4	Angelina	Pisica Tigrata de Himalaia	2016-06-06	5
8	6	Fara	Animalut	729123123	comparator@my.fmi.unibuc.ro	2	<null>	<null>	<null>	<null>	<null>
9	<null>	<null>	<null>	<null>	<null>	<null>	11	Idk	Comuna	2023-01-13	<null>
10	<null>	<null>	<null>	<null>	<null>	<null>	14	Idk2	Comuna	2023-02-01	<null>

```
-- 7.6
-- pt fiecare factura cu suma sub 1000ron, afisati totalul
-- lor, cate chitante au fost eliberate si suma totala a acestora
-- precum si proprietarul
select p.id_proprietar, f.id_factura, total_factura, count(id_chitanta), sum(suma_platita)
from FACTURA f,
CHITANTA c,
FACTURA_PROPRIETAR fp,
PROPRIETAR p
where f.id_factura = c.id_factura
and fp.id_factura = f.id_factura
and p.id_proprietar = fp.id_proprietar
having total_factura < 1000
group by p.id_proprietar, f.id_factura, total_factura;
```

	ID_PROPRIETAR	ID_FACTURA	TOTAL_FACTURA	COUNT(ID_CHITANTA)	SUM(SUMA_PLATITA)
1	1	1	300.00	1	150
2	3	2	300.00	1	300
3	2	3	300.00	1	300
4	1	4	300.00	1	300
5	5	5	300.00	3	301
6	4	6	300.00	2	300
7	4	7	300.00	1	300
8	5	8	300.00	1	300
9	2	9	300.00	1	300
10	1	10	300.00	1	300
11	1	104	300.00	1	300

```
-- 7.7
-- proprietarii care au platite toate facturile (si au minim o factura)
-- operatorul division cu count
select p.id_proprietar
from PROPRIETAR p,
     FACTURA f,
     FACTURA_PROPRIETAR fp
where p.id_proprietar = fp.id_proprietar
     and f.id_factura = fp.id_factura
group by p.id_proprietar
having COUNT(*) = (select count(*)
                  from FACTURA f1,
                  FACTURA_PROPRIETAR fp1
                  where f1.id_factura = fp1.id_factura
                     and fp1.id_proprietar = p.id_proprietar
                     and f1.total_factura <= (select sum(c.suma_platita)
                                              from CHITANTA c
                                              where c.id_factura = f1.id_factura));
```

	ID_PROPRIETAR
1	3
2	4

```
-- 7.8
-- divison: proprietarii pt care nu exista facturi neplatite
-- inclusiv cei care nu au facturi
```

```

select p.id_proprietar
from PROPRIETAR p
where NOT EXISTS(
    select 1
    from FACTURA f,
         FACTURA_PROPRIETAR fp
    where f.id_factura = fp.id_factura
          and fp.id_proprietar = p.id_proprietar
          and not exists(
              select 1
              from FACTURA fl
              where fl.id_factura = f.id_factura
                    and f.total_factura >= (select sum(c.suma_platita) from CHITANTA c where c.id_factura =
fl.id_factura)
              )
    );

```

ID_PROPRIETAR	
1	3
2	4
3	6

```

-- 7.9
-- sa se afiseze pentru fiecare animalut cate luni au trecut intre prima si ultima fise medicala
select a.id_animalut, round(MONTHS_BETWEEN(max(fm.data_fisa), min(fm.data_fisa)))
from ANIMALUT a,
     FISA_MEDICALA fm
where a.id_animalut = fm.id_animalut
group by a.id_animalut;

```



ID_PRODUCATOR
1
3

```
-- 7.12
-- afisati toti angajatii care au consultat animalutul 1
select distinct j.id_angajat, j.nume, j.prenume
from ANGAJAT j,
     FISA_MEDICALA fm,
     ANIMALUT a
where a.id_animalut = fm.id_animalut
     and j.id_angajat = fm.id_doctor
     and a.id_animalut = 1;
```

ID_ANGAJAT	NUME	PRENUME
1	1 Staicu	Clara
2	2 Popa	Adi

```
-- 7.13
-- afisati pt fiecare contract de angajat, al catelea in ierarhia manageriala este
SELECT ctr.id_angajat, LEVEL depth, PRIOR id_angajat
from angajat ctr
START WITH ctr.id_sef IS NULL
CONNECT BY PRIOR ctr.id_angajat = ctr.id_sef;
```

ID_ANGAJAT	DEPTH	PRIORID_ANGAJAT
1	1	<null>
2	2	1
3	3	1
4	4	1
5	5	4

```
-- 7.14 afisati cate tipuri de vaccin au fost vreodata pe stoc
select sum(Decode(INSTR(LOWER(p.caracteristici), 'vaccin'), 0, 0, 1))
```

```
from produs p;
```

```
|| SUM(DECODE(INSTR(LOWER(P.CARACTERISTICI),'VACCIN'),0,0,1)) ÷
```

```
1 2
```

```
-- 7.15 returnati null daca boxer e cea mai consultata rasa, altfel rasa  
-- daca sunt mai multe rase cu acelasi nr de tratamente maxim, sa se afiseze null
```

```
with c as (select rasa, count(*) nr  
            from ANIMALUT  
            group by rasa),  
ans as (select c.rasa from c where c.nr = (select max(c2.nr) from c c2))  
select NULLIF(  
            case (select count(*) from ans)  
              when 1 then (select * from ans)  
              else null  
            end, 'Boxer')
```

```
from dual;
```

```
|| NULLIF(CASE(SELECTCOUNT(*)FROMANS)WHEN1THEN(SELECT*FROMANS)ELSENULLEND,'BOXER')
```

```
1 <null>
```

```
-- 8
```

```
CREATE TABLE MESAJE
```

```
(  
    message_id    NUMBER,  
    message        VARCHAR2(255),  
    message_type   VARCHAR2(1),  
    created_by     VARCHAR2(40) NOT NULL,  
    created_at     DATE          NOT NULL,  
    CONSTRAINT pk_message PRIMARY KEY (message_id),  
    CHECK (message_type = 'E' OR message_type = 'W' OR message_type = 'I')  
);
```

```
CREATE SEQUENCE seq_err START WITH 1;
```



WHERE ORDER BY

	MESSAGE_ID	MESSAGE	MESSAGE_TYPE	CREATED_BY	CREATED_AT
1	1	Alerta: raport suma_totala/co prost	I	C##REKSIO	2023-02-06 05:49:30
2	2	Alerta: raport suma_totala/co prost	I	C##REKSIO	2023-02-06 05:49:30
3	3	Alerta: raport suma_totala/co prost	I	C##REKSIO	2023-02-06 05:49:30
4	4	Alerta: raport suma_totala/co prost	I	C##REKSIO	2023-02-06 05:49:30
5	5	Alerta: raport suma_totala/co prost	I	C##REKSIO	2023-02-06 05:49:30
6	6	Alerta: raport suma_totala/co prost	I	C##REKSIO	2023-02-06 05:49:30
7	7	Alerta: raport suma_totala/co prost	I	C##REKSIO	2023-02-06 05:49:30
8	8	Alerta: raport suma_totala/co prost	I	C##REKSIO	2023-02-06 05:49:30
9	9	Alerta: raport suma_totala/co prost	I	C##REKSIO	2023-02-06 05:49:30
10	10	Alerta: raport suma_totala/co prost	I	C##REKSIO	2023-02-06 05:49:30
11	11	Alerta: raport suma_totala/co prost	I	C##REKSIO	2023-02-06 05:49:30
12	12	Proprietarul animalutului Thor al lui Staicu Octavian-Florin are mai mul...	I	C##REKSIO	2023-02-06 05:49:30
13	13	Nu s-a emis nicio factura pentru proprietarul animalutului Idk al lui FA...	I	C##REKSIO	2023-02-06 05:49:30
14	14	Alerta: factura emisa pentru animalut fara stapan	I	C##REKSIO	2023-02-06 05:49:30

```
-- 9 PL/SQL
```

```
-- 9.1
```

```
-- pentru un animalut dat, afisati doctorii la care acesta a mers cel mai des (de cele mai multe ori)
```

```
-- utilizand doua tipuri de colectii invatate
```

```
CREATE OR REPLACE PROCEDURE afis_doctor_favorit(id_animal ANIMALUT.id_animalut%type)
```

```
IS
```

```
TYPE tab_ind IS TABLE OF PLS_INTEGER INDEX BY PLS_INTEGER;
```

```
TYPE tab_imb IS TABLE OF FISA_MEDICALA.id_doctor%type;
```

```
cnt          tab_ind          := tab_ind();
```

```
doctors      tab_imb          := tab_imb();
```

```
index_max    pls_integer;
```

```
id_doctor    pls_integer;
```

```
ok           binary_integer := 0;
```

```
id           pls_integer;
```

```
num          ANGAJAT.ume%type;
```

```
prenum       ANGAJAT.prenume%type;
```

```
co           pls_integer     := 0;
```

```
BEGIN
```

```
select fm.id_doctor bulk collect
```

```

into doctors
from FISA_MEDICALA fm
where fm.id_animalut = id_animal;

id_doctor := doctors.FIRST;
loop
    exit when id_doctor is null;
    begin
        cnt(id_doctor) := cnt(id_doctor) + 1;
        if ok = 1 then
            if cnt(id_doctor) > cnt(index_max) then
                index_max := id_doctor;
            end if;
        end if;
    EXCEPTION
        when NO_DATA_FOUND then
            cnt(id_doctor) := 1;
            if ok = 0 then
                ok := 1;
                index_max := id_doctor;
            end if;
        end;
    id_doctor := doctors.NEXT(id_doctor);
end loop;
if ok = 0 then
    DBMS_OUTPUT.PUT_LINE('Animalutul ' || id_animal || ' nu are nicio fisa medicala in sistem');
else
    id := cnt.FIRST;
    DBMS_OUTPUT.PUT_LINE('Animalutul ' || id_animal || ' a mers de cele mai multe ori la acesti doctori:');
    loop
        exit when id is null;
        if cnt(id) = cnt(index_max) then
            select a.num, a.prenume
            into num, prenum
            from ANGAJAT a
            where a.id_angajat = id;
            co := co + 1;
            DBMS_OUTPUT.PUT_LINE(co || ' ' || num || ' ' || prenum);
        end if;
    end loop;
end if;

```

```

        id := cnt.NEXT(id);
    end loop;
end if;
END;
/

begin
    afis_doctor_favorit(1);
    afis_doctor_favorit(2);
    afis_doctor_favorit(3);
end;
/

C##REKSIO> begin
    afis_doctor_favorit(1);
    afis_doctor_favorit(2);
    afis_doctor_favorit(3);

    end;

[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

-- 9.2
-- pentru fiecare animalut cu proprietar, afisati toate fisele medicale ale acestuia, data la care au fost
diagnosticati
-- precum si facturile corespunzatoare acelor fise
-- utilizand minim 2 tipuri de cursoare invatate, dintre care minim 1 sa fie parametrizat
CREATE OR REPLACE PROCEDURE afis_tratamente_animalute
IS

```

```

TYPE refcursor IS REF CURSOR;
cursor mycursor(id_anim IN FISA_MEDICALA.id_animalut%type) is select diag.descriere,
                                                                    fm.data_fisa,
                                                                    cursor (select fp.id_factura,
f.data_emitere                                                                    from FACTURA_PROPRIETAR fp,
                                                                    FACTURA f
                                                                    where fp.id_factura = f.id_factura
                                                                    and fp.id_fisa_medicala =
fm.id_fisa)                                                                    from FISA_MEDICALA fm,
                                                                    DIAGNOSTIC diag
                                                                    where fm.id_diagnostic = diag.id_diagnostic
                                                                    and fm.id_animalut = id_anim;

v_cursor          refcursor;
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   ANIMALUT.id_animalut%type;
nume_animalut_max ANIMALUT.nume%type;
nume_proprietar_max PROPRIETAR.nume%type;
prenume_proprietar_max PROPRIETAR.prenume%type;
ok_factura        binary_integer;
co_max            pls_integer := 0;
faliment EXCEPTION;
BEGIN
  for i in (select a.id_animalut, a.nume nume_animalut, p.nume nume_proprietar, p.prenume prenume_proprietar
            from ANIMALUT a,
                 PROPRIETAR p
            where a.id_proprietar = p.id_proprietar)
  loop
    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 || ' ' ||
                           i.prenume_proprietar || ' nu a avut niciodata parte de o procedura medicala');
      DBMS_OUTPUT.NEW_LINE();
    else

```

```

        DBMS_OUTPUT.PUT_LINE('Animalutul ' || i.nume_animalut || ' al lui ' || i.nume_proprietar || ' ' ||
                               i.prenume_proprietar || ' a avut urmatoarele interventii medicale:');
    loop
        exit when mycursor%notfound;
        DBMS_OUTPUT.PUT_LINE(mycursor%rowcount || '. ' || diag || ' la data de ' || data_diagnostic ||
                               ', avand emise facturile:');
        ok_factura := 0;
        loop
            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 || '.
id_factura=' ||
                               id_factura || ', emisa la data de ' || data_factura);
        end loop;

        if ok_factura = 0 then
            DBMS_OUTPUT.PUT_LINE(' Nicio factura emisa pentru aceasta fisa medicala');
        end if;

        fetch mycursor into diag, data_diagnostic, v_cursor;
    end loop;
    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;
    end if;
    end if;
    close mycursor;
end loop;
if co_max = 0 then
    raise faliment;
end if;
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;
/

-- pt a da trigger la exceptia faliment
-- drop table FISA_MEDICALA cascade constraints purge
-- apoi re create table FISA_MEDICALA

```

```

C##REKSID> begin
    afis_tratamente_animalute();
end;
[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 :)

```

```

-- 9.3
-- pentru fiecare fisa medicala a fiecarui animalut, afisati factura corespunzatoare si suma acesteia.
-- Pt o suma minima primita ca parametru, returnati cate facturi au minim aceasta suma, iar prin parametru
-- intoarceti suma tuturor facturilor. Daca sunt mai multe facturi pe aceeasi fisa medicala, ignorati-le si
-- afisati un mesaj de alerta. Afisati de asemenea mesaje de alerta daca la fiecare moment in care raportul
-- dintre suma totala a facturilor si nr acestora este mai mica decat suma minima primita ca parametru.
-- Afisati totodata si un mesaj de alerta in caz ca se gasesc facturi emise pentru un animalut fara stapan
-- (probabil ca a fost emisa pentru o persoana care a gasit un catelus pe strada ce avea nevoie de ingrijire,
-- dar care nu a putut sa il adopte),
-- utilizand minim o comanda sql care sa foloseasca 3 din tabelele definite si definind minim 2 exceptii
CREATE OR REPLACE FUNCTION factura_aferinta_fisa(suma_min IN float, suma_totala OUT float) return pls_integer
IS
    id      FACTURA_PROPRIETAR.id_factura%type;
    pret    FACTURA.total_factura%type;
    co      pls_integer := 0;
    co_tot  pls_integer := 0;
    factura_fara_stapan EXCEPTION;
    raport_pret_consultatii_prost EXCEPTION;
BEGIN
    suma_totala := 0;
    for i in (select p.id_proprietar, p.num propnum, p.prenume, a.num animnume, fm.data_fisa, fm.id_fisa
              from FISA_MEDICALA fm,
                   ANIMALUT a,
                   PROPRIETAR p
              where fm.id_animalut = a.id_animalut
                    and a.id_proprietar = p.id_proprietar(+))
    loop
        BEGIN
            select fp.id_factura, f.total_factura
            into id, pret
            from FACTURA_PROPRIETAR fp,
                 FACTURA f
            where fp.id_factura = f.id_factura
                  and i.id_fisa = fp.id_fisa_medicala;
            DBMS_OUTPUT.PUT_LINE('Proprietarul animalutului ' || i.animnume || ' al lui ' ||
                                nvl(i.propnum, 'FARA') || ' ' || nvl(i.prenume, 'PROPRIETAR') || ' are factura
' ||
                                id || ' emisa la data de ' || i.data_fisa || ', pentru fisa medicala ' ||
                                i.id_fisa || ', total_factura=' || pret);
        END
    end loop;
END

```



```

if i.id_proprietar is null then
    raise factura_fara_stapan;
end if;

suma_totala := suma_totala + pret;
co_tot := co_tot + 1;
if pret >= suma_min then
    co := co + 1;
end if;

if suma_totala / co_tot < suma_min then
    raise raport_pret_consultatii_prost;
end if;

EXCEPTION
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')
        || ' la data de ' || i.data_fisa ||
        || ' desi a fost tratat, cu fisa medicala cu id ' || i.id_fisa);
insert into mesaje(message_id, message, message_type, created_by, created_at)
values (seq_err.NEXTVAL,
    'Nu s-a emis nicio factura pentru proprietarul animalutului ' || i.animnume ||
    ' al lui ' || nvl(i.propnum, 'FARA') || ' ' || nvl(i.prenume, 'PROPRIETAR') ||
    ' la data de ' || i.data_fisa ||
    ' 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') ||
        || ' are mai multe facturi emise la data de ' || i.data_fisa);
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');

```

```

        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;
end loop;
commit;
return co;
end;
/

declare
    res          pls_integer;
    suma_min     float := 490;
    suma_totala  float;
BEGIN
    res := factura_aferenta_fisa(suma_min, suma_totala);
    DBMS_OUTPUT.PUT_LINE(res || ' clienti au facturi de minim ' || suma_min);
    DBMS_OUTPUT.PUT_LINE('suma totala din facturi unice: ' || suma_totala);
end;
/

```

```

C##REKSI0> declare
    res          pls_integer;
    suma_min     float := 490;
    suma_totala  float;
BEGIN
    res := factura_aferenta_fisa(suma_min, suma_totala);
    DBMS_OUTPUT.PUT_LINE(res || ' clienti au facturi de minim ' || suma_min);
    DBMS_OUTPUT.PUT_LINE('suma totala din facturi unice: ' || suma_totala);
end;

[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

```

```

--9 suplimentar (folosit in pcahetul 2)
--Intoarcem id-ul clientilor cu cea mai mare suma achitata (nu datorata) prin chitante
--Daca sunt mai multi, ii intoarcem pe toti
CREATE or REPLACE TYPE tab_imb is TABLE OF NUMBER(10);
/

CREATE or REPLACE FUNCTION topclienti(top_clienti out tab_imb) return pls_integer
IS
TYPE refcursor IS REF CURSOR;
cursor mycursor is select f.id_proprietar,
                        cursor (select c.suma_platita
                                from CHITANTA c
                                where c.id_factura = f.id_factura)
                        from FACTURA_PROPRIETAR f,
                        PROPRIETAR p
                        where f.id_proprietar = p.id_proprietar
                        order by p.id_proprietar;

v_cursor    refcursor;
id           FACTURA_PROPRIETAR.id_factura%type := null;
suma        pls_integer                        := 0;
total_suma  pls_integer                        := 0;
max_suma    pls_integer                        := 0;
last_id     PROPRIETAR.id_proprietar%type     := null;
co          pls_integer                        := 0;
BEGIN
top_clienti := tab_imb();
open mycursor;
loop
    fetch mycursor into id, v_cursor;
    if last_id is null then
        last_id := id;
    else
        if mycursor%notfound or id <> last_id then
            if total_suma > max_suma then
                top_clienti.DELETE;
                top_clienti.EXTEND;
                co := 1;
                top_clienti(1) := last_id;
                max_suma := total_suma;
            end if;
        end if;
    end if;
end loop;

```

```

        elsif total_suma = max_suma then
            top_clienti.EXTEND;
            co := co + 1;
            top_clienti(co) := last_id;
        end if;
        last_id := id;
        total_suma := 0;
    end if;
end if;
exit when mycursor%notfound;
loop
    fetch v_cursor into suma;
    exit when v_cursor%notfound;
    total_suma := total_suma + suma;
end loop;
end loop;
close mycursor;
return co;

END;
/
--functia returneaza nr de clienti, iar parametrul intoarce colectia
--vom scrie un bloc anonim mai complex, pt a verifica corectitudinea functiei
declare
    co      pls_integer;
    id      NUMBER(10);
    mytab   tab_imb;
    num     PROPRIETAR.nume%type;
    prenum  PROPRIETAR.prenume%type;
begin
    co := TOPCLIENTI(mytab);
    DBMS_OUTPUT.PUT_LINE('Exista ' || co || ' clienti:');

    id := mytab.FIRST;
    loop
        exit when id is null;
        select nume, prenume
        into num, prenum
        from PROPRIETAR

```

```

        where id_proprietar = mytab(id);
        DBMS_OUTPUT.PUT_LINE(num || ' ' || prenum || ' (id: ' || mytab(id) || ')');
        id := mytab.NEXT(id);
    end loop;
end;
/

        end;
[2023-02-06 05:53:35] completed in 34 ms
Exista 1 clienti:
Bodea George (id: 2)

-- 9.4 trigger lmd la nivel de comanda
CREATE OR REPLACE TRIGGER trig_contract
    BEFORE DELETE
    on CONTRACT
BEGIN
    RAISE_APPLICATION_ERROR(-20222, 'Contractele trebuie sa ramana pe veci in sistem');
END;
/

CREATE OR REPLACE TRIGGER trig_contract_angajat
    BEFORE DELETE
    on CONTRACT_ANGAJAT
BEGIN
    RAISE_APPLICATION_ERROR(-20222, 'Contractele trebuie sa ramana pe veci in sistem');
END;
/

CREATE OR REPLACE TRIGGER trig_contract_producator
    BEFORE DELETE
    on CONTRACT_PRODUCATOR
BEGIN
    RAISE_APPLICATION_ERROR(-20222, 'Contractele trebuie sa ramana pe veci in sistem');
END;
/

DELETE
from CONTRACT
where id_contract = 1;

```

```

DELETE
from CONTRACT_ANGAJAT
where id_contract = 1;
DELETE
from CONTRACT_PRODUCATOR
where id_contract = 4;
C##REKSI0> DELETE
      from CONTRACT
      where id_contract = 1
[2023-02-06 05:55:11] [72000][20222]
[2023-02-06 05:55:11]   ORA-20222: Contractele trebuie sa ramana pe veci in sistem
[2023-02-06 05:55:11]   ORA-06512: at "C##REKSI0.TRIG_CONTRACT", line 2
[2023-02-06 05:55:11]   ORA-04088: error during execution of trigger 'C##REKSI0.TRIG_CONTRACT'
[2023-02-06 05:55:11] Position: 13
C##REKSI0> DELETE
      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##REKSI0.TRIG_CONTRACT_ANGAJAT", line 2
[2023-02-06 05:55:14]   ORA-04088: error during execution of trigger 'C##REKSI0.TRIG_CONTRACT_ANGAJAT'
[2023-02-06 05:55:14] Position: 13
C##REKSI0> DELETE
      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]   ORA-06512: at "C##REKSI0.TRIG_CONTRACT_PRODUCATOR", line 2
[2023-02-06 05:55:16]   ORA-04088: error during execution of trigger 'C##REKSI0.TRIG_CONTRACT_PRODUCATOR'
[2023-02-06 05:55:16] Position: 13

-- 9.5 triggeri lmd la nivel de linie
CREATE OR REPLACE TRIGGER trig_tratament_doctor
  BEFORE INSERT OR UPDATE of id_doctor
  on FISA_MEDICALA
  FOR EACH ROW
DECLARE
  ok ANGAJAT.trateaza_animale%type;

```

```

BEGIN
    select trateaza_animale
    into ok
    from ANGAJAT
    where id_angajat = :NEW.id_doctor;
    if ok <> 1 then
        RAISE_APPLICATION_ERROR(-20001, 'angajatul cu id-ul ' || :NEW.id_doctor || ' nu poate trata animalute');
    end if;
EXCEPTION
    when NO_DATA_FOUND then
        RAISE_APPLICATION_ERROR(-20000, 'Nu exista un asemenea doctor');
end;
/

INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
VALUES (60, 1, 1, 3, sysdate);
INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
VALUES (60, 1, 1, 6, sysdate);
C##REKSI0> INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
VALUES (60, 1, 1, 3, sysdate)
[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##REKSI0.TRIG_TRATAMENT_DOCTOR", line 9
[2023-02-06 05:55:59] ORA-04088: error during execution of trigger 'C##REKSI0.TRIG_TRATAMENT_DOCTOR'
[2023-02-06 05:55:59] Position: 12
C##REKSI0> INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
VALUES (60, 1, 1, 6, sysdate)
[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-06512: at "C##REKSI0.TRIG_TRATAMENT_DOCTOR", line 13
[2023-02-06 05:56:01] ORA-04088: error during execution of trigger 'C##REKSI0.TRIG_TRATAMENT_DOCTOR'
[2023-02-06 05:56:01] Position: 12

CREATE OR REPLACE TRIGGER trig_tratament_zinastere
BEFORE INSERT OR UPDATE
on FISA_MEDICALA
FOR EACH ROW
DECLARE
    zinastere ANIMALUT.zi_nastere%type;

```



```
BEGIN
    select zi_nastere
    into zinastere
    from ANIMALUT
    where id_animalut = :NEW.id_animalut;

    if :NEW.data_fisa < zinastere then
        RAISE_APPLICATION_ERROR(-20013, 'Nu poti trata un animalut inainte ca acesta sa se nasca');
    end if;
EXCEPTION
    when NO_DATA_FOUND then
        RAISE_APPLICATION_ERROR(-20000, 'Nu exista un astfel de animalut in sistem');
end;
/

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'));
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'));
update FISA_MEDICALA
set data_fisa=to_date('01.01.1991', 'dd.mm.yyyy')
where id_fisa = 1;
```

```

C##REKSI0> 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##REKSI0.TRIG_TRATAMENT_ZINASTERE", line 10
[2023-02-06 05:56:44] ORA-04088: error during execution of trigger 'C##REKSI0.TRIG_TRATAMENT_ZINASTERE'
[2023-02-06 05:56:44] Position: 12
C##REKSI0> 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-06512: at "C##REKSI0.TRIG_TRATAMENT_ZINASTERE", line 14
[2023-02-06 05:56:45] ORA-04088: error during execution of trigger 'C##REKSI0.TRIG_TRATAMENT_ZINASTERE'
[2023-02-06 05:56:45] Position: 12
C##REKSI0> update FISA_MEDICALA
set data_fisa=to_date('01.01.1991', 'dd.mm.yyyy')
where id_fisa = 1
[2023-02-06 05:56:47] [72000][20013]
[2023-02-06 05:56:47] ORA-20013: Nu poti trata un animalut inainte ca acesta sa se nasca
[2023-02-06 05:56:47] ORA-06512: at "C##REKSI0.TRIG_TRATAMENT_ZINASTERE", line 10
[2023-02-06 05:56:47] ORA-04088: error during execution of trigger 'C##REKSI0.TRIG_TRATAMENT_ZINASTERE'
[2023-02-06 05:56:47] Position: 7

-- compound trigger
CREATE OR REPLACE TRIGGER trig_update_contracte
FOR UPDATE
ON CONTRACT COMPOUND TRIGGER

BEFORE STATEMENT IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('Afisare cotracte actualizate');
END BEFORE STATEMENT;

AFTER EACH ROW IS
BEGIN
    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
    BEGIN
        DBMS_OUTPUT.PUT_LINE('Contractele au fost actualizate');
    END AFTER STATEMENT;
    END trig_update_contracte;
/

```

```

UPDATE CONTRACT
SET OBSERVATII='Obs'
WHERE ID_CONTRACT = 10;

```

```

C##REKSIQ> 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_isa_factura_proprietar
FOR INSERT OR UPDATE
ON FACTURA_PROPRIETAR COMPOUND TRIGGER

BEFORE STATEMENT IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('Verificare unicitate subtip facturi proprietar');
END BEFORE STATEMENT;
BEFORE EACH ROW IS
Begin
    declare
        co pls_integer := 0;

```

```

        begin
            select count(*) into co from FACTURA_PRODUCATOR fp where fp.id_factura = :NEW.id_factura;
            if co > 0 then
                RAISE_APPLICATION_ERROR(-20778, 'O factura nu poate fi simultan si pt proprietar si pt producator');
            end if;
        end;
    end before each row;

    AFTER EACH ROW IS
    BEGIN
        DBMS_OUTPUT.PUT_LINE('ID factura ' || :new.id_factura || ' este acum de tip proprietar');
    END AFTER EACH
    ROW;

    AFTER STATEMENT IS
    BEGIN
        DBMS_OUTPUT.PUT_LINE('Facturile au fost actualizare');
    END AFTER STATEMENT;
END trig_isa_factura_proprietar;
/

insert into FACTURA_PROPRIETAR (id_factura, id_proprietar, id_fisa_medicala)
VALUES (11, 1, 1);
update FACTURA_PROPRIETAR
set id_factura = 11
where id_factura = 1;

```

```

C##REKSIO> insert into FACTURA_PROPRIETAR (id_factura, id_proprietar, id_fisa_medicala)
VALUES (11, 1, 1)
[2023-08-28 22:56:39] [72000][20778]
[2023-08-28 22:56:39] ORA-20778: 0 factura nu poate fi simultan si pt proprietar si pt producator
[2023-08-28 22:56:39] ORA-06512: at "C##REKSIO.TRIG_ISA_FACTURA_PROPRIETAR", line 14
[2023-08-28 22:56:39] ORA-04088: error during execution of trigger 'C##REKSIO.TRIG_ISA_FACTURA_PROPRIETAR'
[2023-08-28 22:56:39] Position: 12
Verificare unicitate subtip facturi proprietar
C##REKSIO> update FACTURA_PROPRIETAR
set id_factura = 11
where id_factura = 1
[2023-08-28 22:56:41] [72000][20778]
[2023-08-28 22:56:41] ORA-20778: 0 factura nu poate fi simultan si pt proprietar si pt producator
[2023-08-28 22:56:41] ORA-06512: at "C##REKSIO.TRIG_ISA_FACTURA_PROPRIETAR", line 14
[2023-08-28 22:56:41] ORA-04088: error during execution of trigger 'C##REKSIO.TRIG_ISA_FACTURA_PROPRIETAR'
[2023-08-28 22:56:41] Position: 7
Verificare unicitate subtip facturi proprietar

CREATE OR REPLACE TRIGGER trig_isa_factura_producator
FOR INSERT OR UPDATE
ON FACTURA_PRODUCATOR COMPOUND TRIGGER

BEFORE STATEMENT IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('Verificare unicitate subtip facturi producator');
END BEFORE STATEMENT;
BEFORE EACH ROW IS
Begin
    declare
        co pls_integer := 0;
    begin
        select count(*) into co from FACTURA_PROPRIETAR fp where fp.id_factura = :NEW.id_factura;
        if co > 0 then
            RAISE_APPLICATION_ERROR(-20778, '0 factura nu poate fi simultan si pt proprietar si pt producator');
        end if;
    end;
end before each row;

AFTER EACH ROW IS

```

```

BEGIN
    DBMS_OUTPUT.PUT_LINE('ID factura ' || :new.id_factura || ' este acum de tip producator');
END AFTER EACH
ROW;

AFTER STATEMENT IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('Facturile au fost actualizare');
END AFTER STATEMENT;
END trig_isa_factura_producator;
/

```

```

insert into FACTURA_PRODUCATOR(id_factura, id_producator)
values (1, 1);
update FACTURA_PRODUCATOR
set id_factura=1
where id_factura = 11;

```

```

C##REKSI0> insert into FACTURA_PRODUCATOR(id_factura, id_producator)
      values (1, 1)
[2023-08-28 22:57:59] [72000][20778]
[2023-08-28 22:57:59]   ORA-20778: 0 factura nu poate fi simultan si pt proprietar si pt producator
[2023-08-28 22:57:59]   ORA-06512: at "C##REKSI0.TRIG_ISA_FACTURA_PRODUCATOR", line 14
[2023-08-28 22:57:59]   ORA-04088: error during execution of trigger 'C##REKSI0.TRIG_ISA_FACTURA_PRODUCATOR'
[2023-08-28 22:57:59] Position: 12
Verificare unicitate subtip facturi producator
C##REKSI0> update FACTURA_PRODUCATOR
      set id_factura=1
      where id_factura = 11
[2023-08-28 22:58:00] [72000][20778]
[2023-08-28 22:58:00]   ORA-20778: 0 factura nu poate fi simultan si pt proprietar si pt producator
[2023-08-28 22:58:00]   ORA-06512: at "C##REKSI0.TRIG_ISA_FACTURA_PRODUCATOR", line 14
[2023-08-28 22:58:00]   ORA-04088: error during execution of trigger 'C##REKSI0.TRIG_ISA_FACTURA_PRODUCATOR'
[2023-08-28 22:58:00] Position: 7
Verificare unicitate subtip facturi producator

```

```

CREATE OR REPLACE TRIGGER trig_isa_contract_producator
    FOR INSERT OR UPDATE
    ON CONTRACT_PRODUCATOR COMPOUND TRIGGER

BEFORE STATEMENT IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('Verificare unicitate subtip contract producator');
END BEFORE STATEMENT;

BEFORE EACH ROW IS
    Begin
        declare
            co pls_integer := 0;
        begin
            select count(*) into co from CONTRACT_ANGAJAT cp where cp.id_contract = :NEW.id_contract;
            if co > 0 then
                RAISE_APPLICATION_ERROR(-20778,
                    'Un contract nu poate fi simultan si pt un angajat si pt un producator');
            end if;
        end;
    end before each row;

AFTER EACH ROW IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('ID contract ' || :new.id_contract || ' este acum de tip producator');
END AFTER EACH
ROW;

AFTER STATEMENT IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('Contractele au fost actualizare');
END AFTER STATEMENT;
END trig_isa_contract_producator;

/

insert into CONTRACT_PRODUCATOR(id_contract, id_producator)
values (1, 1);
update CONTRACT_PRODUCATOR
set id_contract=1
where id_contract = 4;

```

```

C##REKSI0> insert into CONTRACT_PRODUCATOR(id_contract, id_producator)
      values (1, 1)
[2023-08-28 22:59:08] [72000][20778]
[2023-08-28 22:59:08]   ORA-20778: Un contract nu poate fi simultan si pt un angajat si pt un producator
[2023-08-28 22:59:08]   ORA-06512: at "C##REKSI0.TRIG_ISA_CONTRACT_PRODUCATOR", line 14
[2023-08-28 22:59:08]   ORA-04088: error during execution of trigger 'C##REKSI0.TRIG_ISA_CONTRACT_PRODUCATOR'
[2023-08-28 22:59:08] Position: 12
Verificare unicitate subtip contract producator
C##REKSI0> update CONTRACT_PRODUCATOR
      set id_contract=1
      where id_contract = 4
[2023-08-28 22:59:09] [72000][20778]
[2023-08-28 22:59:09]   ORA-20778: Un contract nu poate fi simultan si pt un angajat si pt un producator
[2023-08-28 22:59:09]   ORA-06512: at "C##REKSI0.TRIG_ISA_CONTRACT_PRODUCATOR", line 14
[2023-08-28 22:59:09]   ORA-04088: error during execution of trigger 'C##REKSI0.TRIG_ISA_CONTRACT_PRODUCATOR'
[2023-08-28 22:59:09] Position: 7
Verificare unicitate subtip contract producator

CREATE OR REPLACE TRIGGER trig_isa_contract_angajat
FOR INSERT OR UPDATE
ON CONTRACT_ANGAJAT COMPOUND TRIGGER

BEFORE STATEMENT IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('Verificare unicitate subtip contract angajat');
END BEFORE STATEMENT;
BEFORE EACH ROW IS
Begin
    declare
        co pls_integer := 0;
    begin
        select count(*) into co from CONTRACT_PRODUCATOR cp where cp.id_contract = :NEW.id_contract;
        if co > 0 then
            RAISE_APPLICATION_ERROR(-20778,
                                    'Un contract nu poate fi simultan si pt un angajat si pt un producator');
        end if;
    end;
end before each row;

```



```

AFTER EACH ROW IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('ID contract ' || :new.id_contract || ' este acum de tip angajat');
END AFTER EACH
ROW;

AFTER STATEMENT IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('Contractele au fost actualizare');
END AFTER STATEMENT;
END trig_isa_contract_angajat;
/

```

```

insert into CONTRACT_ANGAJAT(id_contract, id_angajat, rol, salariu)
values (4, 1, 'rol', 1);
update CONTRACT_ANGAJAT
set id_contract = 4
where id_contract = 1;

```

```

C##REKSI0> insert into CONTRACT_ANGAJAT(id_contract, id_angajat, rol, salariu)
        values (4, 1, 'rol', 1)
[2023-08-28 22:59:48] [72000][20778]
[2023-08-28 22:59:48]   ORA-20778: Un contract nu poate fi simultan si pt un angajat si pt un producator
[2023-08-28 22:59:48]   ORA-06512: at "C##REKSI0.TRIG_ISA_CONTRACT_ANGAJAT", line 14
[2023-08-28 22:59:48]   ORA-04088: error during execution of trigger 'C##REKSI0.TRIG_ISA_CONTRACT_ANGAJAT'
[2023-08-28 22:59:48] Position: 12
Verificare unicitate subtip contract angajat
C##REKSI0> update CONTRACT_ANGAJAT
        set id_contract = 4
        where id_contract = 1
[2023-08-28 22:59:49] [72000][20778]
[2023-08-28 22:59:49]   ORA-20778: Un contract nu poate fi simultan si pt un angajat si pt un producator
[2023-08-28 22:59:49]   ORA-06512: at "C##REKSI0.TRIG_ISA_CONTRACT_ANGAJAT", line 14
[2023-08-28 22:59:49]   ORA-04088: error during execution of trigger 'C##REKSI0.TRIG_ISA_CONTRACT_ANGAJAT'
[2023-08-28 22:59:49] Position: 7
Verificare unicitate subtip contract angajat

```

```
-- 9.6 trigger ldd
CREATE OR REPLACE TRIGGER trig_ldd_contract
    BEFORE DROP OR ALTER
    ON schema
begin
    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');
    end if;
end;
/

drop table CONTRACT;
drop table CONTRACT_PRODUCATOR;
drop table CONTRACT_ANGAJAT;
alter table CONTRACT
    drop column observatii;
```

```
C##REKSI0> 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##REKSI0.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-20777: Nu aveti voie sa modificati sau sa stegeti tabela cu contracte
[2023-02-06 05:57:57] ORA-06512: at line 4
[2023-02-06 05:57:57] Position: 0
C##REKSI0> 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##REKSI0.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-20777: Nu aveti voie sa modificati sau sa stegeti tabela cu contracte
[2023-02-06 05:58:00] ORA-06512: at line 4
[2023-02-06 05:58:00] Position: 0
C##REKSI0> drop table CONTRACT_ANGAJAT
[2023-02-06 05:58:04] [42000][4088]
[2023-02-06 05:58:04] ORA-04088: error during execution of trigger 'C##REKSI0.TRIG_LDD_CONTRACT'
[2023-02-06 05:58:04] ORA-00604: error occurred at recursive SQL level 1
[2023-02-06 05:58:04] ORA-20777: Nu aveti voie sa modificati sau sa stegeti tabela cu contracte
[2023-02-06 05:58:04] ORA-06512: at line 4
[2023-02-06 05:58:04] Position: 0
C##REKSI0> 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##REKSI0.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
```

```

-- 9.7 pachet
CREATE OR REPLACE PACKAGE package_clinica_vet_tav13
AS
    PROCEDURE afis_doctor_favorit(id_animal ANIMALUT.id_animalut%type);
    PROCEDURE afis_tratamente_animalute;
    FUNCTION factura_aferinta_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
AS
    PROCEDURE afis_doctor_favorit(id_animal ANIMALUT.id_animalut%type)
    IS
        TYPE tab_ind IS TABLE OF PLS_INTEGER INDEX BY PLS_INTEGER;
        TYPE tab_imb IS TABLE OF FISA_MEDICALA.id_doctor%type;
        cnt          tab_ind          := tab_ind();
        doctors      tab_imb          := tab_imb();
        index_max    pls_integer;
        id_doctor    pls_integer;
        ok           binary_integer := 0;
        id           pls_integer;
        num          ANGAJAT.nume%type;
        prenum       ANGAJAT.prenume%type;
        co           pls_integer      := 0;
    BEGIN
        select fm.id_doctor bulk collect
        into doctors
        from FISA_MEDICALA fm
        where fm.id_animalut = id_animal;

        id_doctor := doctors.FIRST;
        loop
            exit when id_doctor is null;
            begin
                cnt(id_doctor) := cnt(id_doctor) + 1;
                if ok = 1 then
                    if cnt(id_doctor) > cnt(index_max) then

```

```
        index_max := id_doctor;
    end if;
end if;
EXCEPTION
when NO_DATA_FOUND then
    cnt(id_doctor) := 1;
    if ok = 0 then
        ok := 1;
        index_max := id_doctor;
    end if;
end;
id_doctor := doctors.NEXT(id_doctor);
end loop;
if ok = 0 then
    DBMS_OUTPUT.PUT_LINE('Animalutul ' || id_animal || ' nu are nicio fisa medicala in sistem');
else
    id := cnt.FIRST;
    DBMS_OUTPUT.PUT_LINE('Animalutul ' || id_animal || ' a mers de cele mai multe ori la acesti doctori:');
loop
    exit when id is null;
    if cnt(id) = cnt(index_max) then
        select a.nume, a.prenume
            into num, prenum
          from ANGAJAT a
         where a.id_angajat = id;
        co := co + 1;
        DBMS_OUTPUT.PUT_LINE(co || '. ' || num || ' ' || prenum);
    end if;
    id := cnt.NEXT(id);
end loop;
end if;
END afis_doctor_favorit;
```

PROCEDURE afis\_tratamente\_animalute

```
IS
TYPE refcursor IS REF CURSOR;
cursor mycursor(id_anim IN FISA_MEDICALA.id_animalut%type) is select diag.descriere,
```

```

f.data_emitere
                                from FACTURA_PROPRIETAR fp,
                                FACTURA f
                                where fp.id_factura =

f.id_factura
                                and fp.id_fisa_medicala =

fm.id_fisa)
                                from FISA_MEDICALA fm,
                                DIAGNOSTIC diag
                                where fm.id_diagnostic = diag.id_diagnostic
                                and fm.id_animalut = id_anim;

v_cursor      refcursor;
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 ANIMALUT.id_animalut%type;
nume_animalut_max ANIMALUT.nume%type;
nume_proprietar_max PROPRIETAR.nume%type;
prenume_proprietar_max PROPRIETAR.prenume%type;
ok_factura    binary_integer;
co_max        pls_integer := 0;
faliment EXCEPTION;
BEGIN
  for i in (select a.id_animalut, a.nume nume_animalut, p.nume nume_proprietar, p.prenume prenume_proprietar
            from ANIMALUT a,
                 PROPRIETAR p
            where a.id_proprietar = p.id_proprietar)
  loop
    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 || ' '
                            ||
                            i.prenume_proprietar || ' nu a avut niciodata parte de o procedura
medicala');
      DBMS_OUTPUT.NEW_LINE();
    else
      DBMS_OUTPUT.PUT_LINE('Animalutul ' || i.nume_animalut || ' al lui ' || i.nume_proprietar || ' '
                            ||
                            i.prenume_proprietar || ' a avut o procedura medicala');
    end if;
  end loop;
END;

```

```

                                i.prenume_proprietar || ' a avut urmatoarele interventii medicale:');
loop
    exit when mycursor%notfound;
    DBMS_OUTPUT.PUT_LINE(mycursor%rowcount || '. ' || diag || ' la data de ' || data_diagnostic

                                ', avand emise facturile:');
    ok_factura := 0;
    loop
        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 ||

                                '. id_factura=' ||

                                id_factura || ', emisa la data de ' || data_factura);

    end loop;

    if ok_factura = 0 then
        DBMS_OUTPUT.PUT_LINE(' Nicio factura emisa pentru aceasta fisa medicala');
    end if;

    fetch mycursor into diag, data_diagnostic, v_cursor;
end loop;
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;
end if;
end if;
close mycursor;
end loop;
if co_max = 0 then
    raise faliment;
end if;
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...');
        commit;
END afis_tratamente_animalute;

FUNCTION factura_aferinta_fisa(suma_min IN float, suma_totala OUT float) return pls_integer
IS
    id      FACTURA_PROPRIETAR.id_factura%type;
    pret    FACTURA.total_factura%type;
    co      pls_integer := 0;
    co_tot  pls_integer := 0;
    factura_fara_stapan EXCEPTION;
    raport_pret_consultatii_prost EXCEPTION;
BEGIN
    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,
                   ANIMALUT a,
                   PROPRIETAR p
              where fm.id_animalut = a.id_animalut
                    and a.id_proprietar = p.id_proprietar(+))
    loop
        BEGIN
            select fp.id_factura, f.total_factura
            into id, pret
            from FACTURA_PROPRIETAR fp,
                 FACTURA f
            where fp.id_factura = f.id_factura
                  and i.id_fisa = fp.id_fisa_medicala;
            DBMS_OUTPUT.PUT_LINE('Proprietarul animalutului ' || i.animnume || ' al lui ' ||
                                nvl(i.propnum, 'FARA') || ' ' || nvl(i.prenume, 'PROPRIETAR') ||
                                ' are factura ' ||
                                id || ' emisa la data de ' || i.data_fisa || ', pentru fisa medicala ' ||
                                i.id_fisa || ', total_factura=' || pret);
        END
    end loop;
END;

```



```

        if i.id_proprietar is null then
            raise factura_fara_stapan;
        end if;

        suma_totala := suma_totala + pret;
        co_tot := co_tot + 1;
        if pret >= suma_min then
            co := co + 1;
        end if;

        if suma_totala / co_tot < suma_min then
            raise raport_pret_consultatii_prost;
        end if;
    EXCEPTION
        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') ||
                                ' la data de ' || i.data_fisa ||
                                ' desi a fost tratat, cu fisa medicala cu id ' || i.id_fisa);
            insert into mesaje(message_id, message, message_type, created_by, created_at)
            values (seq_err.NEXTVAL,
                    'Nu s-a emis nicio factura pentru proprietarul animalutului ' || i.animnume ||
                    ' al lui ' || nvl(i.propnum, 'FARA') || ' ' || nvl(i.prenume, 'PROPRIETAR') ||
                    ' la data de ' || i.data_fisa ||
                    ' 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') ||
                                ' are mai multe facturi emise la data de ' || i.data_fisa);
            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');
        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;
end loop;
commit;
return co;
end factura_aferenta_fisa;

FUNCTION topclienti(top_clienti out tab_imb) return pls_integer
IS
TYPE refcursor IS REF CURSOR;
cursor mycursor is select f.id_proprietar,
                        cursor (select c.suma_platita
                                from CHITANTA c
                                where c.id_factura = f.id_factura)
                        from FACTURA_PROPRIETAR f,
                        PROPRIETAR p
                        where f.id_proprietar = p.id_proprietar
                        order by p.id_proprietar;

v_cursor    refcursor;
id           FACTURA_PROPRIETAR.id_factura%type := null;
suma        pls_integer                        := 0;
total_suma   pls_integer                        := 0;
max_suma     pls_integer                        := 0;
last_id      PROPRIETAR.id_proprietar%type     := null;
co           pls_integer                        := 0;
BEGIN
    top_clienti := tab_imb();
    open mycursor;
    loop
        fetch mycursor into id, v_cursor;
        if last_id is null then
            last_id := id;

```

```

        else
            if mycursor%notfound or id <> last_id then
                if total_suma > max_suma then
                    top_clienti.DELETE;
                    top_clienti.EXTEND;
                    co := 1;
                    top_clienti(1) := last_id;
                    max_suma := total_suma;
                elsif total_suma = max_suma then
                    top_clienti.EXTEND;
                    co := co + 1;
                    top_clienti(co) := last_id;
                end if;
                last_id := id;
                total_suma := 0;
            end if;
        end if;
        exit when mycursor%notfound;
    loop
        fetch v_cursor into suma;
        exit when v_cursor%notfound;
        total_suma := total_suma + suma;
    end loop;
end loop;
close mycursor;
return co;

END topclienti;

END package_clinica_vet_tav13;
/

begin
    package_clinica_vet_tav13.afis_doctor_favorit(1);

DBMS_OUTPUT.PUT_LINE('.....');

    package_clinica_vet_tav13.afis_tratamente_animalute();

```

```
DBMS_OUTPUT.PUT_LINE('.....');
```

```
declare
```

```
co      pls_integer;
```

```
sum_tot float;
```

```
begin
```

```
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);
```

```
end;
```

```
DBMS_OUTPUT.PUT_LINE('.....');
```

```
declare
```

```
co      pls_integer;
```

```
id      NUMBER(10);
```

```
mytab   package_clinica_vet_tav13.tab_imb;
```

```
num     PROPRIETAR.nume%type;
```

```
prenum  PROPRIETAR.prenume%type;
```

```
begin
```

```
co := package_clinica_vet_tav13.TOPCLIENTI(mytab);
```

```
DBMS_OUTPUT.PUT_LINE('Exista ' || co || ' clienti:');
```

```
id := mytab.FIRST;
```

```
loop
```

```
    exit when id is null;
```

```
    select nume, prenume
```

```
    into num, prenum
```

```
    from PROPRIETAR
```

```
    where id_proprietar = mytab(id);
```

```
    DBMS_OUTPUT.PUT_LINE(num || ' ' || prenum || ' (id: ' || mytab(id) || ')');
```

```
    id := mytab.NEXT(id);
```

```
end loop;
```

```
end;
```

```
end;
```

```
/
```

[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 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 :)

.....

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



```

        PROPRIETAR p
        where f.id_proprietar = p.id_proprietar
        order by p.id_proprietar;

v_cursor    refcursor;
id           FACTURA_PROPRIETAR.id_factura%type := null;
suma        pls_integer                        := 0;
total_suma   pls_integer                        := 0;
max_suma     pls_integer                        := 0;
last_id      PROPRIETAR.id_proprietar%type     := null;
co           pls_integer                        := 0;

BEGIN
  open mycursor;
  loop
    fetch mycursor into id, v_cursor;
    if last_id is null then
      last_id := id;
    else
      if id <> last_id or mycursor%notfound then
        if total_suma > max_suma then
          top_clienti.DELETE;
          top_clienti.EXTEND;
          co := 1;
          top_clienti(1) := last_id;
          max_suma := total_suma;
        elsif total_suma = max_suma then
          top_clienti.EXTEND;
          co := co + 1;
          top_clienti(co) := last_id;
        end if;
        last_id := id;
        total_suma := 0;
      end if;
    end if;
    exit when mycursor%notfound;
  loop
    fetch v_cursor into suma;
    exit when v_cursor%notfound;
    total_suma := total_suma + suma;
  end loop;

```



```

        end loop;
        close mycursor;
        return co;

END topclienti;

PROCEDURE afis_tratamente_animalute
IS
    TYPE refcursor IS REF CURSOR;
    cursor mycursor(id_anim IN FISA_MEDICALA.id_animalut%type) is select diag.descriere,
                                                                    fm.data_fisa,
                                                                    cursor (select fp.id_factura,
f.data_emitere                                                                    from FACTURA_PROPRIETAR fp,
                                                                    FACTURA f
f.id_factura                                                                    where fp.id_factura =
fm.id_fisa)                                                                    and fp.id_fisa_medicala =
                                                                    from FISA_MEDICALA fm,
                                                                    DIAGNOSTIC diag
                                                                    where fm.id_diagnostic = diag.id_diagnostic
                                                                    and fm.id_animalut = id_anim;

    v_cursor          refcursor;
    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    ANIMALUT.id_animalut%type;
    nume_animalut_max  ANIMALUT.nume%type;
    nume_proprietar_max PROPRIETAR.nume%type;
    prenume_proprietar_max PROPRIETAR.prenume%type;
    ok_factura         binary_integer;
    co_max             pls_integer := 0;
    faliment EXCEPTION;
BEGIN
    for i in (select a.id_animalut, a.nume nume_animalut, p.nume nume_proprietar, p.prenume prenume_proprietar
              from ANIMALUT a,
              PROPRIETAR p

```

```

        where a.id_proprietar = p.id_proprietar
        and p.id_proprietar in (select * from table (top_clienti)) )
loop
    open mycursor(i.id_animalut);
    fetch mycursor into diag, data_diagnostic, v_cursor;
    if mycursor%notfound then
        DBMS_OUTPUT.PUT_LINE('Animalutul ' || i.ume_animalut || ' al lui ' || i.ume_proprietar || ' '
||
||
||
        i.prenume_proprietar || ' nu a avut niciodata parte de o procedura
medicala');
        DBMS_OUTPUT.NEW_LINE();
    else
        DBMS_OUTPUT.PUT_LINE('Animalutul ' || i.ume_animalut || ' al lui ' || i.ume_proprietar || ' '
||
||
||
        i.prenume_proprietar || ' a avut urmatoarele interventii medicale:');
        loop
            exit when mycursor%notfound;
            DBMS_OUTPUT.PUT_LINE(mycursor%rowcount || '. ' || diag || ' la data de ' || data_diagnostic
||
||
||
            ', avand emise facturile:');
            ok_factura := 0;
            loop
                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 ||
||
||
||
                '. id_factura=' ||
                id_factura || ', emisa la data de ' || data_factura);
            end loop;

            if ok_factura = 0 then
                DBMS_OUTPUT.PUT_LINE(' Nicio factura emisa pentru aceasta fisa medicala');
            end if;

            fetch mycursor into diag, data_diagnostic, v_cursor;
        end loop;
        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;
    end if;
end if;
close mycursor;
end loop;
if co_max = 0 then
    raise faliment;
end if;
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...');
END afis_tratamente_animalute;

PROCEDURE afis_topclienti
IS
    co    pls_integer;
    id    NUMBER(10);
    num    PROPRIETAR.nume%type;
    prenum PROPRIETAR.prenume%type;
begin
    co := package_clinica_vet_tav14.TOPCLIENTI;
    DBMS_OUTPUT.PUT_LINE('Exista ' || co || ' clienti:');
    id := top_clienti.FIRST;
    loop
        exit when id is null;
        select nume, prenume
        into num, prenum
        from PROPRIETAR
        where id_proprietar = top_clienti(id);
        DBMS_OUTPUT.PUT_LINE(num || ' ' || prenum || ' (id: ' || top_clienti(id) || ')');
    end loop;
end;

```

```

        id := top_clienti.NEXT(id);
    end loop;
    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;
/

begin
    package_clinica_vet_tav14.afis_topclienti;
end;
/

C##REKSI0> begin
    package_clinica_vet_tav14.afis_topclienti;
end;

[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:
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 Kitty [id_animalut=3] al lui Bodea George este unul dintre cei mai fideli clienti ai nostri: ne-a vizitat de 2 ori :)

-- Inca un pachet suplimentar, cu doua utilitare folositoare
-- Primul subprogram din pachet ne permit sa ordonam pacientii la primirea in cabinet astfel incat sa reducem timpul
-- mediu de asteptare. Al doilea subprogram afiseaza datoriile ramase de achitat ale clientilor
CREATE OR REPLACE PACKAGE package_clinica_vet_utilitare
IS

    type t_idx is table of pls_integer index by pls_integer;

```

```

type array is VARRAY(1000) of pls_integer;
type myrec_datorii is record
(
    id_prop      PROPRIETAR.id_proprietar%type,
    nume_prop    PROPRIETAR.nume%type,
    prenume_prop PROPRIETAR.prenume%type,
    id_fac       FACTURA_PROPRIETAR.id_factura%type,
    total_fac    FACTURA.total_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
IS

FUNCTION timp_minim_de_asteptare(v in out array) RETURN t_idx
    Is --Primind un vector cu timpul in care dureaza fiecare procedura medicala pt oamenii din acea zi
    --se stabileste in ce ordine vor intra in cabinet, ai timpul mediu de asteptare sa fie minim
    poz_min pls_integer;
    sol      t_idx;
    co       pls_integer := 0;
BEGIN
    if v.FIRST is null then
        raise NO_DATA_FOUND;
    end if;
    loop
        poz_min := 1;
        for i in v.FIRST..v.LAST
            loop
                if v(i) < v(poz_min) then
                    poz_min := i;
                end if;
            end loop;
        exit when v(poz_min) = 1001;
    end loop;

```

```

        co := co + 1;
        sol(poz_min) := co;
        v(poz_min) := 1001;
    end loop;
    return sol;
EXCEPTION
    when NO_DATA_FOUND then
        return sol;
END timp_minim_de_asteptare;

PROCEDURE afis_datorii is
    my_t      datorii_imb;
    s          pls_integer;
    s_dat      pls_integer          := 0;
    idx        pls_integer;
    last_prop  PROPRIETAR.id_proprietar%type := null;
Begin
    select p.id_proprietar,
           p.nume,
           p.prenume,
           fp.id_factura,
           f.total_factura,
           sum(c.suma_platita) bulk collect
    into my_t
    from PROPRIETAR p,
         FACTURA_PROPRIETAR fp,
         FACTURA f,
         CHITANTA c
    where p.id_proprietar = fp.id_proprietar
          and fp.id_factura = f.id_factura
          and f.id_factura = c.id_factura(+)
    group by p.id_proprietar, p.nume, p.prenume, fp.id_factura, f.total_factura
    order by p.id_proprietar;
    idx := my_t.FIRST;
    loop
        if last_prop is null then
            if idx is not null then
                last_prop := my_t(idx).id_prop;
            end if;
        end if;
    end loop;

```

```

        DBMS_OUTPUT.PUT_LINE(
            'Proprietarul ' || my_t(idx).nume_prop || ' ' || my_t(idx).prenume_prop || ' (id:'
||
            my_t(idx).id_prop || '):');
    end if;
else
    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;
        if idx is not null then
            last_prop := my_t(idx).id_prop;
            DBMS_OUTPUT.PUT_LINE(
                'Proprietarul ' || my_t(idx).nume_prop || ' ' || my_t(idx).prenume_prop ||
                ' (id:' ||
                my_t(idx).id_prop || '):');
        end if;
    end if;
end if;
exit when idx is null;
s := my_t(idx).total_fac - nvl(my_t(idx).suma_platita, 0);
s_dat := s_dat + s;
if s > 0 then
    DBMS_OUTPUT.PUT_LINE('La factura ' || my_t(idx).id_fac || ' mai are de platit: ' || s);
end if;
idx := my_t.NEXT(idx);
end loop;

end afis_datorii;

END package_clinica_vet_utilitare;
/

begin
    package_clinica_vet_utilitare.afis_datorii;
end;
/

DECLARE

```

```
v    package_clinica_vet_utilitare.array := package_clinica_vet_utilitare.array(7, 1, 2, 5, 3);
sol package_clinica_vet_utilitare.t_idx;
BEGIN

sol := package_clinica_vet_utilitare.timp_minim_de_asteptare(v);
if sol.FIRST is not null then
    for i in sol.FIRST..sol.LAST
        loop
            DBMS_OUTPUT.PUT_LINE(i || ' ' || sol(i));
        end loop;
    else
        DBMS_OUTPUT.PUT_LINE('Niciun Client Programat');
    end if;
end;
/
```



```

SQL> begin
    package_clinica_vet_utilitare.afis_datorii;
end;
[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
SQL> DECLARE
    v package_clinica_vet_utilitare.array := package_clinica_vet_utilitare.array(7, 1, 2, 5, 3);
    sol package_clinica_vet_utilitare.t_idx;
BEGIN

    sol := package_clinica_vet_utilitare.timp_minim_de_asteptare(v);
    if sol.FIRST is not null then
        for i in sol.FIRST..sol.LAST
            loop
                DBMS_OUTPUT.PUT_LINE(i || '. ' || sol(i));
            end loop;
        else
            DBMS_OUTPUT.PUT_LINE('Niciun Client Programat');
        end if;
    end;
[2023-02-06 06:02:18] completed in 7 ms
1. 5
2. 1
3. 2
4. 4
5. 3

```