

PROIECT SGBD

CLINICĂ
VETERINARĂ

Student: Staicu Octavian-Florin

Grupa 231

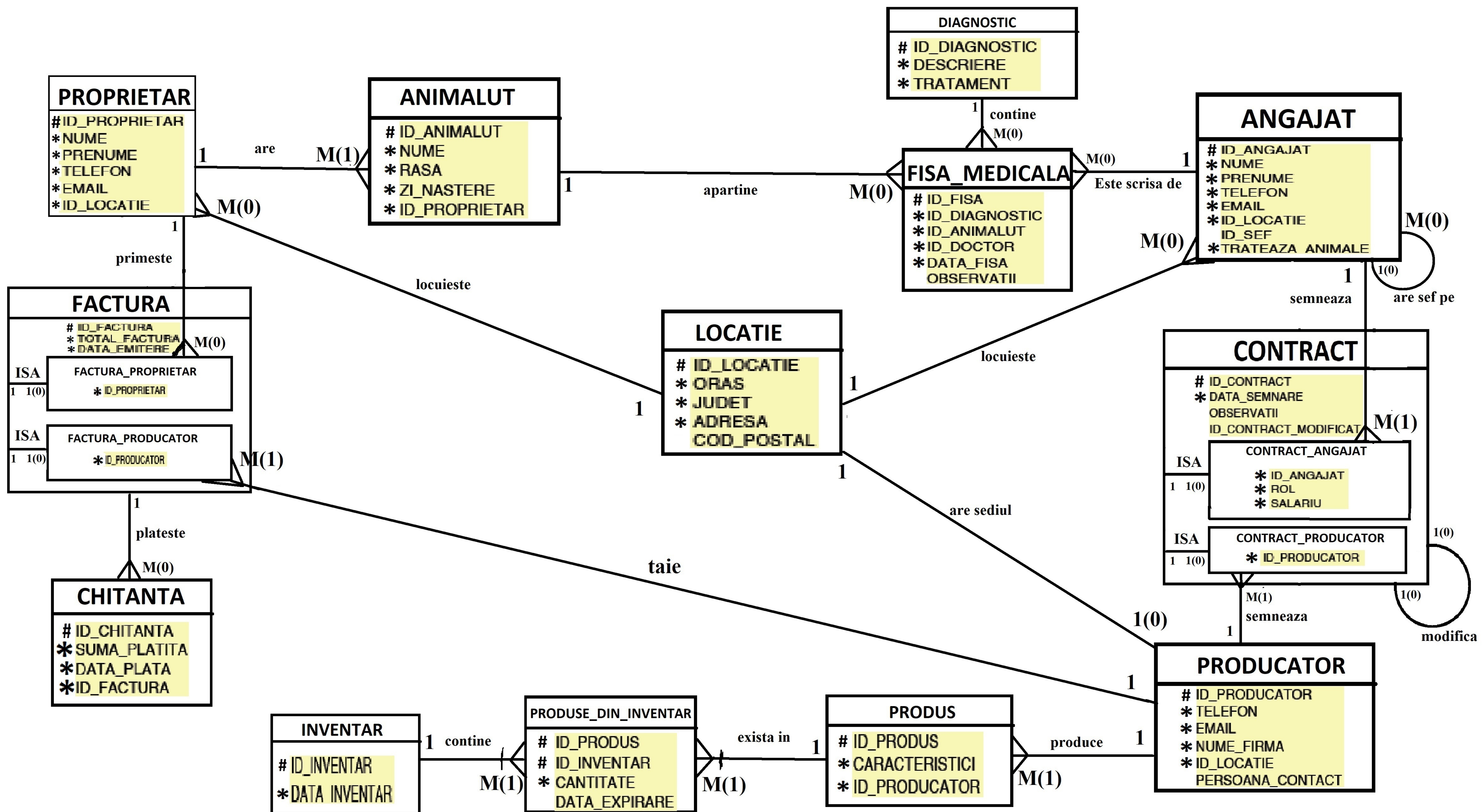
Clinică veterinară

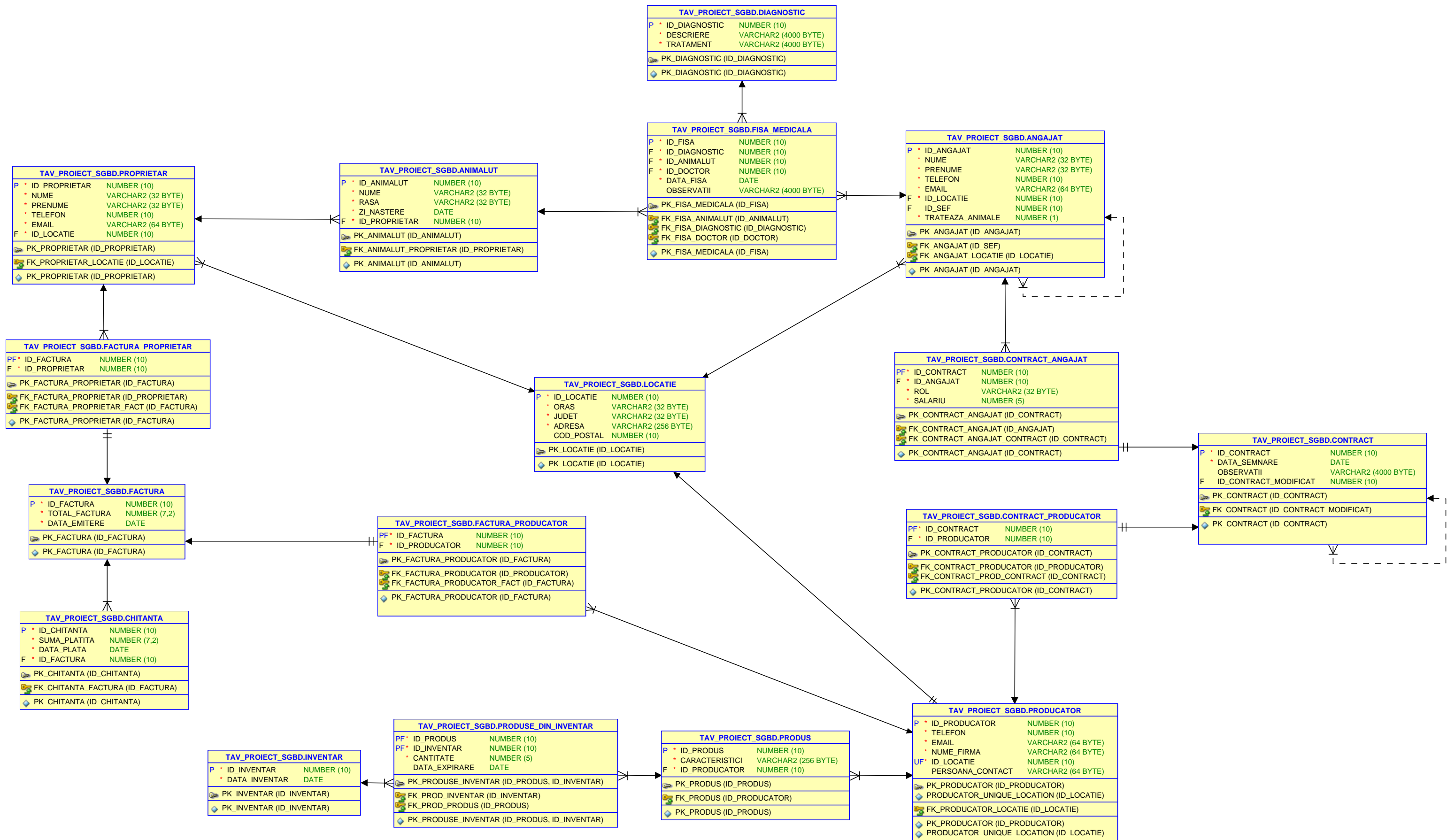
Acest proiect se adresează unei clinici veterinare

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.
- 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.





```
--4
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)
);

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) NOT NULL,
    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)
);
```

```

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)
);

CREATE TABLE CONTRACT
(
    id_contract      NUMBER(10),
    data_semnare     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)
);

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 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),
    id_inventar    NUMBER(10),
    cantitate      NUMBER(5) NOT NULL,
    data_expirare  DATE,
    CONSTRAINT pk_produce_inventar PRIMARY KEY (id_produs, id_inventar),
    CONSTRAINT fk_prod_inventar FOREIGN KEY (id_inventar) REFERENCES
INVENTAR (id_inventar),
    CONSTRAINT fk_prod_produs FOREIGN KEY (id_produs) REFERENCES PRODUS
(id_produs)
);

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,
    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)
);

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)
);

commit;

```

```

)
[2021-01-09 21:13:56] completed in 30 ms
TAV_PROJECT_SG00> CREATE TABLE FACTURA_PROPRIETAR
(
    id_factura    NUMBER(10),
    id_proprietar NUMBER(10) NOT NULL,
    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)
)
[2021-01-09 21:13:56] completed in 26 ms
TAV_PROJECT_SG00> 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)
)
[2021-01-09 21:13:57] completed in 32 ms
TAV_PROJECT_SG00> 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)
)
[2021-01-09 21:13:57] completed in 31 ms
TAV_PROJECT_SG00> commit
[2021-01-09 21:13:57] completed in 3 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 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 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 DIAGNOSTIC(id_diagnostic, descriere, tratament)
VALUES (1, 'Vaccinare Leptospiroza', 'Vaccin Leptospiroza');
INSERT INTO DIAGNOSTIC(id_diagnostic, descriere, tratament)
VALUES (2, 'Enterocolita', 'Enteroguarg');
INSERT INTO DIAGNOSTIC(id_diagnostic, descriere, tratament)
VALUES (3, 'Rana deschisa', 'Spray Cicatrizant');

INSERT INTO ANGAJAT(id_angajat, nume, prenume, telefon, email, id_locatie,
trateaza_animale)
```

```

VALUES (1, 'Staicu', 'Clara', 0736659440, 'dr.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.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 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 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@aparaturaanimale.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 CONTRACT(id_contract, data_semnaire)
VALUES (1, to_date('02.01.2005', 'dd.mm.yyyy'));
INSERT INTO CONTRACT(id_contract, data_semnaire)
VALUES (2, to_date('13.07.2004', 'dd.mm.yyyy'));
INSERT INTO CONTRACT(id_contract, data_semnaire)
VALUES (3, to_date('09.09.2005', 'dd.mm.yyyy'));
INSERT INTO CONTRACT(id_contract, data_semnaire)
VALUES (4, to_date('01.05.2007', 'dd.mm.yyyy'));
INSERT INTO CONTRACT(id_contract, data_semnaire)
VALUES (5, to_date('02.06.2008', 'dd.mm.yyyy'));
INSERT INTO CONTRACT(id_contract, data_semnaire)
VALUES (6, to_date('02.01.2005', 'dd.mm.yyyy'));
INSERT INTO CONTRACT(id_contract, data_semnaire, observatii,
id_contract_modificat)
VALUES (7, to_date('28.12.2020', 'dd.mm.yyyy'), 'Promovare', 2);

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_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 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 PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate,
data_expirare)
VALUES (1, 1, 10, to_date('05.01.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate,
data_expirare)
VALUES (2, 1, 15, to_date('05.01.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate,
data_expirare)
VALUES (1, 2, 20, to_date('05.01.2016', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate,
data_expirare)
VALUES (2, 2, 25, to_date('05.01.2016', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate,
data_expirare)
VALUES (1, 3, 35, to_date('05.01.2021', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate,
data_expirare)
VALUES (2, 3, 30, to_date('05.01.2021', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate)
VALUES (3, 1, 1);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate)
VALUES (3, 2, 1);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate)
VALUES (3, 3, 2);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate)
VALUES (4, 1, 5);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate)
VALUES (4, 2, 10);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate)
VALUES (4, 3, 15);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate)
VALUES (5, 1, 1);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate)
VALUES (5, 2, 1);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate)
VALUES (5, 3, 2);
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate,
data_expirare)
VALUES (6, 1, 100, to_date('13.07.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate,
data_expirare)
VALUES (7, 1, 150, to_date('13.07.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate,
data_expirare)
VALUES (6, 2, 100, to_date('13.07.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate,
data_expirare)
VALUES (7, 2, 150, to_date('13.07.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate,
data_expirare)
VALUES (6, 3, 234, to_date('13.07.2011', 'dd.mm.yyyy'));
INSERT INTO PRODUSE_DIN_INVENTAR(id_produ, id_inventar, cantitate,
data_expirare)
VALUES (7, 3, 512, to_date('13.07.2011', 'dd.mm.yyyy'));

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, 300, to_date('13.07.2021', 'dd.mm.yyyy'));
insert into FACTURA(id_factura, total_factura, data_emitere)
VALUES (106, 600, to_date('12.05.2021', 'dd.mm.yyyy'));
Insert into FACTURA(id_factura, total_factura, data_emitere)
VALUES (1000, 1000, sysdate);

INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar)
VALUES (1, 1);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar)
VALUES (2, 3);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar)
VALUES (3, 2);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar)
VALUES (4, 1);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar)
VALUES (5, 5);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar)
VALUES (6, 4);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar)
VALUES (7, 4);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar)
VALUES (8, 5);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar)
VALUES (9, 2);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar)
VALUES (10, 1);
insert into FACTURA_PROPRIETAR(id_factura, id_proprietar)
VALUES (104, 1);
insert into FACTURA_PROPRIETAR(id_factura, id_proprietar)
VALUES (105, 1);
insert into FACTURA_PROPRIETAR(id_factura, id_proprietar)
VALUES (106, 2);
INSERT INTO FACTURA_PROPRIETAR(id_factura, id_proprietar)
VALUES (1000, 5);

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_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, 300, 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);

```

```
commit;
```

```

FAY_PROJECT_8680> INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (15, 4325, to_date('25.10.2009', 'dd.mm.yyyy'), 12)
[2021-01-09 21:16:08] 1 row affected in 4 ms
FAY_PROJECT_8680> INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (16, 3000, to_date('25.07.2009', 'dd.mm.yyyy'), 13)
[2021-01-09 21:16:08] 1 row affected in 4 ms
FAY_PROJECT_8680> INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (17, 3000, to_date('25.09.2009', 'dd.mm.yyyy'), 13)
[2021-01-09 21:16:08] 1 row affected in 5 ms
FAY_PROJECT_8680> INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (18, 1400, to_date('25.10.2014', 'dd.mm.yyyy'), 14)
[2021-01-09 21:16:08] 1 row affected in 4 ms
FAY_PROJECT_8680> INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (19, 4725, to_date('13.05.2013', 'dd.mm.yyyy'), 15)
[2021-01-09 21:16:09] 1 row affected in 4 ms
FAY_PROJECT_8680> INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (20, 3025, to_date('25.09.2014', 'dd.mm.yyyy'), 16)
[2021-01-09 21:16:09] 1 row affected in 5 ms
FAY_PROJECT_8680> INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (21, 3000, to_date('01.11.2014', 'dd.mm.yyyy'), 16)
[2021-01-09 21:16:09] 1 row affected in 3 ms
FAY_PROJECT_8680> INSERT INTO CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (22, 4500, to_date('25.10.2019', 'dd.mm.yyyy'), 18)
[2021-01-09 21:16:09] 1 row affected in 3 ms
FAY_PROJECT_8680> insert into CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (104, 300, to_date('13.07.2021', 'dd.mm.yyyy'), 104)
[2021-01-09 21:16:09] 1 row affected in 17 ms
FAY_PROJECT_8680> insert into CHITANTA(id_chitanta, suma_platita, data_plata, id_factura)
VALUES (106, 600, to_date('13.07.2021', 'dd.mm.yyyy'), 106)
[2021-01-09 21:16:10] 1 row affected in 4 ms
FAY_PROJECT_8680> commit
[2021-01-09 21:16:10] completed in 6 ms

```

```

--6
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;
    cnt          tab_ind;
    index_max    pls_integer;
    ok           binary_integer := 0;
    id           pls_integer;
    num          ANGAJAT.nume%type;
    prenum       ANGAJAT.prenume%type;
    co           pls_integer := 0;
BEGIN
    for i in (select fm.id_doctor
              from FISA_MEDICALA fm
              where fm.id_animalut = id_animal)
    loop
        begin
            cnt(i.id_doctor) := cnt(i.id_doctor) + 1;
            if ok = 1 then
                if cnt(i.id_doctor) > cnt(index_max) then
                    index_max := i.id_doctor;
                end if;
            end if;
        EXCEPTION
            when NO_DATA_FOUND then
                cnt(i.id_doctor) := 1;
                if ok = 0 then
                    ok := 1;
                    index_max := i.id_doctor;
                end if;
            end;
        end loop;
        if ok = 0 then
            DBMS_OUTPUT.PUT_LINE('Acest animalut nu are nicio fisa medicala in
sistem');
        else
            id := cnt.FIRST;
            DBMS_OUTPUT.PUT_LINE('Acest animalut 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;
    /

begin
    afis_doctor_favorit(3);
end;
/

```

```
END;  
[2021-01-09 21:17:28] completed in 33 ms  
TAV_PROJECT_0000> begin  
    afis_doctor_favorit(3);  
end;  
[2021-01-09 21:17:36] completed in 11 ms  
[2021-01-09 21:17:36] Acest animalut a mers de cele mai multe ori la acesti doctori:  
[2021-01-09 21:17:36] 1. Staicu Clara  
[2021-01-09 21:17:36] 2. Popa Adi
```



```

--7
CREATE OR REPLACE PROCEDURE afis_tratamente_animalute
IS
TYPE refcursor IS REF CURSOR;
cursor mycursor is select a.numa,
                           p.numa,
                           p.prenume,
                           cursor (select diag.descriere, fm.data_fisa
                                   from FISA_MEDICALA fm,
                                   DIAGNOSTIC diag
                                   where fm.id_diagnostic =
diag.id_diagnostic
                                   and a.id_animalut = fm.id_animalut
                                   )
                           from ANIMALUT a,
                           PROPRIETAR p
                           where a.id_proprietar = p.id_proprietar;

v_cursor
numa_animalut      ANIMALUT.numa%type;
numa_proprietar    PROPRIETAR.numa%type;
prenume_proprietar PROPRIETAR.prenume%type;
diag              DIAGNOSTIC.descriere%type;
data_diagnostic    FISA_MEDICALA.data_fisa%type;
numa_animalut_max  ANIMALUT.numa%type;
numa_proprietar_max PROPRIETAR.numa%type;
prenume_proprietar_max PROPRIETAR.prenume%type;
co_max            pls_integer := 0;
faliment EXCEPTION;
BEGIN
    open mycursor;
    loop
        fetch mycursor into
numa_animalut,numa_proprietar,prenume_proprietar,v_cursor;
        exit when mycursor%notfound;
        fetch v_cursor into diag, data_diagnostic;
        if v_cursor%notfound then
            DBMS_OUTPUT.PUT_LINE(
                'Animalutul ' || numa_animalut || ' al lui ' ||
numa_proprietar || ' ' || prenume_proprietar ||
                ' nu a avut niciodata parte de o procedura
medicala');
            DBMS_OUTPUT.NEW_LINE();
        else
            DBMS_OUTPUT.PUT_LINE(
                'Animalutul ' || numa_animalut || ' al lui ' ||
numa_proprietar || ' ' || prenume_proprietar ||
                ' a avut urmatoarele interventii medicale:');
            loop
                exit when v_cursor%notfound;
                DBMS_OUTPUT.PUT_LINE(v_cursor%rowcount || ' ' || diag || '
la data de ' || data_diagnostic);
                fetch v_cursor into diag, data_diagnostic;
            end loop;
            if v_cursor%rowcount > co_max then
                co_max := v_cursor%rowcount;
                numa_animalut_max := numa_animalut;
                numa_proprietar_max := numa_proprietar;
                prenume_proprietar_max := prenume_proprietar;
            end if;
            DBMS_OUTPUT.NEW_LINE();
        end if;
    end if;

```

```

end loop;
close mycursor;
if co_max = 0 then
    raise faliment;
end if;
DBMS_OUTPUT.PUT_LINE('Animalutul ' || nume_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;
/

begin
    afis_tratamente_animalute();
end;
/

```

```

END;
[2021-01-09 21:18:48] completed in 30 ms
SQL_PROJECT_S000> begin
    afis_tratamente_animalute();
end;
[2021-01-09 21:18:52] completed in 67 ms
[2021-01-09 21:18:52] Animalutul Bit al lui Staicu Octavian-Florin a avut urmatoarele interventii medicale:
[2021-01-09 21:18:52] 1. Enterocolita la data de 13-07-2020
[2021-01-09 21:18:52] 2. Rana deschisa la data de 13-09-2020
[2021-01-09 21:18:52] 3. Enterocolita la data de 25-12-2020
[2021-01-09 21:18:52] Animalutul Kid al lui Staicu Adrian a avut urmatoarele interventii medicale:
[2021-01-09 21:18:52] 1. Vaccinare Leptospiroza la data de 13-07-2011
[2021-01-09 21:18:52] Animalutul Kitty al lui Bodea George a avut urmatoarele interventii medicale:
[2021-01-09 21:18:52] 1. Vaccinare Leptospiroza la data de 13-07-2016
[2021-01-09 21:18:52] 2. Rana deschisa la data de 13-12-2016
[2021-01-09 21:18:52] Animalutul Angelina al lui Cineva Interesant a avut urmatoarele interventii medicale:
[2021-01-09 21:18:52] 1. Vaccinare Leptospiroza la data de 13-07-2017
[2021-01-09 21:18:52] 2. Vaccinare Leptospiroza la data de 13-07-2020
[2021-01-09 21:18:52] Animalutul John al lui Popescu Ion a avut urmatoarele interventii medicale:
[2021-01-09 21:18:52] 1. Enterocolita la data de 04-05-2009
[2021-01-09 21:18:52] 2. Vaccinare Leptospiroza la data de 14-06-2010
[2021-01-09 21:18:52] 3. Rana deschisa la data de 14-08-2015
[2021-01-09 21:18:52] Animalutul dog al lui Cineva Interesant nu a avut niciodata parte de o procedura medicala
[2021-01-09 21:18:52] Animalutul Thor al lui Staicu Octavian-Florin a avut urmatoarele interventii medicale:
[2021-01-09 21:18:52] 1. Vaccinare Leptospiroza la data de 13-09-2020
[2021-01-09 21:18:52] 2. Vaccinare Leptospiroza la data de 13-07-2021
[2021-01-09 21:18:52] Animalutul Bit al lui Staicu Octavian-Florin este unul dintre cei mai fideli clienti ai nostri: ne-a vizitat

    afis_tratamente_animalute();
end;
[2021-01-09 21:25:20] completed in 10 ms
[2021-01-09 21:25:20] Nu avem niciun animalut care s-a tratat la noi...

```

*in a doua poză s-a rulat procedura fără nicio valoare inserată în tabelele din baza de date, pentru a ilustra excepția

**în cazul în care sunt mai multe animăluțe cu același număr de tratamente, se afișează doar unul

```

--8
--Intoarcem id-ul clientilor cu cea mai mare suma achitata (nu datorata)
--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;
                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_imp;
    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;
/

```

```

[2021-01-09 21:22:13] completed in 26 ms
[2021-01-09 21:22:13] Exista 2 clienti:
[2021-01-09 21:22:13] Staicu Octavian-Florin (id: 1)
[2021-01-09 21:22:13] Bodea George (id: 2)

```



```

--9
--Modelul de bussines nu cere sa se tina ce factura carui tratament
apartine
--(adica nu exista un fk intre factura si fisa medicala)
--acesta e un algoritm probabilistic care gaseste daca stapanul unui
animalut are o factura emisa la data cand acesta
--are o fisa
CREATE OR REPLACE PROCEDURE factura_aferenta_fisa
IS
    id FACTURA_PROPRIETAR.id_factura%type;
BEGIN
    for i in (select p.id_proprietar, p.numa propnum, p.prenume, a.numa
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
            into id
            from FACTURA_PROPRIETAR fp,
                 FACTURA f
            where fp.id_proprietar = i.id_proprietar
                  and fp.id_factura = f.id_factura
                  and i.data_fisa = f.data_emitere;
            DBMS_OUTPUT.PUT_LINE('Proprietarul animalutului ' ||
i.animnume || ' al lui ' ||
                                i.propnum || ' ' || i.prenume || ' are
factura ' || id || ' emisa la data de ' ||
                                i.data_fisa || ', posibil pentru fisa
medicala ' || i.id_fisa);
            EXCEPTION
                when NO_DATA_FOUND then
                    DBMS_OUTPUT.PUT_LINE('Nu s-a emis nicio factura pentru
proprietarul animalutului ' || i.animnume ||
                                ' al lui ' ||
                                i.propnum || ' ' || i.prenume || '
la data de ' || i.data_fisa ||
                                ' desi a fost tratat, cu fisa
medicala cu id ' || i.id_fisa);
                when TOO_MANY_ROWS then
                    DBMS_OUTPUT.PUT_LINE('Proprietarul animalutului ' ||
i.animnume || ' al lui ' ||
                                i.propnum || ' ' || i.prenume || '
are mai multe facturi emise la data de ' ||
                                i.data_fisa);
            END;
        end loop;
    end;
/

BEGIN
    factura_aferenta_fisa();
end;
/

```

```
700072704002100077
end;
[2021-01-09 21:23:31] completed in 29 ms
[2021-01-09 21:23:31] Proprietarul animalutului Bit al lui Staicu Octavian-Florin are factura 1 emisa la data de 13-07-2020, posibil pentru fisa medicala 1
[2021-01-09 21:23:31] Proprietarul animalutului Mid al lui Staicu Adrian are factura 2 emisa la data de 13-07-2011, posibil pentru fisa medicala 2
[2021-01-09 21:23:31] Nu s-a emis nicio factura pentru proprietarul animalutului Kitty al lui Bodea George la data de 13-07-2016 desi a fost tratat, cu fisa medicala cu id 3
[2021-01-09 21:23:31] Proprietarul animalutului Bit al lui Staicu Octavian-Florin are factura 4 emisa la data de 13-09-2020, posibil pentru fisa medicala 4
[2021-01-09 21:23:31] Proprietarul animalutului Angelina al lui Cineva Interesant are factura 5 emisa la data de 13-07-2017, posibil pentru fisa medicala 5
[2021-01-09 21:23:31] Proprietarul animalutului John al lui Popescu Ion are factura 6 emisa la data de 04-05-2009, posibil pentru fisa medicala 6
[2021-01-09 21:23:31] Proprietarul animalutului John al lui Popescu Ion are factura 7 emisa la data de 14-06-2010, posibil pentru fisa medicala 7
[2021-01-09 21:23:31] Proprietarul animalutului Angelina al lui Cineva Interesant are factura 8 emisa la data de 13-07-2020, posibil pentru fisa medicala 8
[2021-01-09 21:23:31] Proprietarul animalutului Kitty al lui Bodea George are factura 9 emisa la data de 13-12-2016, posibil pentru fisa medicala 9
[2021-01-09 21:23:31] Proprietarul animalutului Bit al lui Staicu Octavian-Florin are factura 10 emisa la data de 25-12-2020, posibil pentru fisa medicala 10
[2021-01-09 21:23:31] Nu s-a emis nicio factura pentru proprietarul animalutului John al lui Popescu Ion la data de 14-08-2015 desi a fost tratat, cu fisa medicala cu id 11
[2021-01-09 21:23:31] Proprietarul animalutului Thor al lui Staicu Octavian-Florin are factura 4 emisa la data de 13-09-2020, posibil pentru fisa medicala 13
[2021-01-09 21:23:31] Proprietarul animalutului Thor al lui Staicu Octavian-Florin are mai multe facturi emise la data de 13-07-2021
```

```
--10
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;
```

```
TAV_PROIECT_SGBD> DELETE
      from CONTRACT
      where id_contract = 1
[2021-01-09 21:28:27] [72000][20222]
[2021-01-09 21:28:27] ORA-20222: Contractele trebuie sa ramana pe veci in sistem
[2021-01-09 21:28:27] ORA-06512: at "TAV_PROIECT_SGBD.TRIG_CONTRACT", line 2
[2021-01-09 21:28:27] ORA-04088: error during execution of trigger 'TAV_PROIECT_SGBD.TRIG_CONTRACT'
[2021-01-09 21:28:27] Position: 13
TAV_PROIECT_SGBD> DELETE
      from CONTRACT_ANGAJAT
      where id_contract = 1
[2021-01-09 21:28:28] [72000][20222]
[2021-01-09 21:28:28] ORA-20222: Contractele trebuie sa ramana pe veci in sistem
[2021-01-09 21:28:28] ORA-06512: at "TAV_PROIECT_SGBD.TRIG_CONTRACT_ANGAJAT", line 2
[2021-01-09 21:28:28] ORA-04088: error during execution of trigger 'TAV_PROIECT_SGBD.TRIG_CONTRACT_ANGAJAT'
[2021-01-09 21:28:28] Position: 13
TAV_PROIECT_SGBD> DELETE
      from CONTRACT_PRODUCATOR
      where id_contract = 4
[2021-01-09 21:28:30] [72000][20222]
[2021-01-09 21:28:30] ORA-20222: Contractele trebuie sa ramana pe veci in sistem
[2021-01-09 21:28:30] ORA-06512: at "TAV_PROIECT_SGBD.TRIG_CONTRACT_PRODUCATOR", line 2
[2021-01-09 21:28:30] ORA-04088: error during execution of trigger 'TAV_PROIECT_SGBD.TRIG_CONTRACT_PRODUCATOR'
[2021-01-09 21:28:30] Position: 13
```

```

--11
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, 4, sysdate);

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;

```



```

TAV_PROIECT_SGBD> INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
VALUES (60, 1, 1, 3, sysdate)
[2021-01-09 21:29:48] [72000][20001]
[2021-01-09 21:29:48] ORA-20001: angajatul cu id-ul 3 nu poate trata animalute
[2021-01-09 21:29:48] ORA-06512: at "TAV_PROIECT_SGBD.TRIG_TRATAMENT_DOCTOR", line 9
[2021-01-09 21:29:48] ORA-04088: error during execution of trigger 'TAV_PROIECT_SGBD.TRIG_TRATAMENT_DOCTOR'
[2021-01-09 21:29:48] Position: 12
TAV_PROIECT_SGBD> INSERT INTO FISA_MEDICALA(id_fisa, id_diagnostic, id_animalut, id_doctor, data_fisa)
VALUES (60, 1, 1, 4, sysdate)
[2021-01-09 21:29:50] [72000][20000]
[2021-01-09 21:29:50] ORA-20000: Nu exista un asemenea doctor
[2021-01-09 21:29:50] ORA-06512: at "TAV_PROIECT_SGBD.TRIG_TRATAMENT_DOCTOR", line 13
[2021-01-09 21:29:50] ORA-04088: error during execution of trigger 'TAV_PROIECT_SGBD.TRIG_TRATAMENT_DOCTOR'
[2021-01-09 21:29:50] Position: 12
TAV_PROIECT_SGBD> 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'))
[2021-01-09 21:29:58] [72000][20013]
[2021-01-09 21:29:58] ORA-20013: Nu poti trata un animalut inainte ca acesta sa se nasca
[2021-01-09 21:29:58] ORA-06512: at "TAV_PROIECT_SGBD.TRIG_TRATAMENT_ZINASTERE", line 10
[2021-01-09 21:29:58] ORA-04088: error during execution of trigger 'TAV_PROIECT_SGBD.TRIG_TRATAMENT_ZINASTERE'
[2021-01-09 21:29:58] Position: 12
TAV_PROIECT_SGBD> 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'))
[2021-01-09 21:29:59] [72000][20000]
[2021-01-09 21:29:59] ORA-20000: Nu exista un astfel de animalut in sistem
[2021-01-09 21:29:59] ORA-06512: at "TAV_PROIECT_SGBD.TRIG_TRATAMENT_ZINASTERE", line 14
[2021-01-09 21:29:59] ORA-04088: error during execution of trigger 'TAV_PROIECT_SGBD.TRIG_TRATAMENT_ZINASTERE'
[2021-01-09 21:29:59] Position: 12
TAV_PROIECT_SGBD> update FISA_MEDICALA
set data_fisa=to_date('01.01.1991', 'dd.mm.yyyy')
where id_fisa = 1
[2021-01-09 21:30:01] [72000][20013]
[2021-01-09 21:30:01] ORA-20013: Nu poti trata un animalut inainte ca acesta sa se nasca
[2021-01-09 21:30:01] ORA-06512: at "TAV_PROIECT_SGBD.TRIG_TRATAMENT_ZINASTERE", line 10
[2021-01-09 21:30:01] ORA-04088: error during execution of trigger 'TAV_PROIECT_SGBD.TRIG_TRATAMENT_ZINASTERE'
[2021-01-09 21:30:01] Position: 7

```

```
--12
--blocheaza stergerea sau alterarea tabelelor cu contracte
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;
TAV_PROJECT_SGDD> drop table CONTRACT
[2021-01-09 21:32:00] [60000][604]
[2021-01-09 21:32:00] ORA-00604: error occurred at recursive SQL level 1
[2021-01-09 21:32:00] ORA-20777: Nu aveti voie sa modificati sau sa stegeti tabela cu contracte
[2021-01-09 21:32:00] ORA-06512: at line 4
[2021-01-09 21:32:00] Position: 0
TAV_PROJECT_SGDD> drop table CONTRACT_PRODUCATOR
[2021-01-09 21:32:02] [60000][604]
[2021-01-09 21:32:02] ORA-00604: error occurred at recursive SQL level 1
[2021-01-09 21:32:02] ORA-20777: Nu aveti voie sa modificati sau sa stegeti tabela cu contracte
[2021-01-09 21:32:02] ORA-06512: at line 4
[2021-01-09 21:32:02] Position: 0
TAV_PROJECT_SGDD> drop table CONTRACT_ANGAJAT
[2021-01-09 21:32:03] [60000][604]
[2021-01-09 21:32:03] ORA-00604: error occurred at recursive SQL level 1
[2021-01-09 21:32:03] ORA-20777: Nu aveti voie sa modificati sau sa stegeti tabela cu contracte
[2021-01-09 21:32:03] ORA-06512: at line 4
[2021-01-09 21:32:03] Position: 0
TAV_PROJECT_SGDD> alter table CONTRACT
  drop column observatii
[2021-01-09 21:32:04] [60000][604]
[2021-01-09 21:32:04] ORA-00604: error occurred at recursive SQL level 1
[2021-01-09 21:32:04] ORA-20777: Nu aveti voie sa modificati sau sa stegeti tabela cu contracte
[2021-01-09 21:32:04] ORA-06512: at line 4
[2021-01-09 21:32:04] Position: 0
```

```
--13
CREATE OR REPLACE PACKAGE package_clinica_vet_tav13
AS
    PROCEDURE afis_doctor_favorit(id_animal ANIMALUT.id_animalut%type);
    PROCEDURE afis_tratamente_animalute;
    TYPE tab_imb IS TABLE OF NUMBER(10);
    FUNCTION topclienti(top_clienti out tab_imb) return pls_integer;
    PROCEDURE factura_aferenta_fisa;
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;
        cnt          tab_ind;
        index_max     pls_integer;
        ok            binary_integer := 0;
        id            pls_integer;
        num           ANGAJAT.nume%type;
        prenum        ANGAJAT.prenume%type;
        co            pls_integer := 0;
    BEGIN
        for i in (select fm.id_doctor
                  from FISA_MEDICALA fm
                  where fm.id_animalut = id_animal)
        loop
            begin
                cnt(i.id_doctor) := cnt(i.id_doctor) + 1;
                if ok = 1 then
                    if cnt(i.id_doctor) > cnt(index_max) then
                        index_max := i.id_doctor;
                    end if;
                end if;
            EXCEPTION
                when NO_DATA_FOUND then
                    cnt(i.id_doctor) := 1;
                    if ok = 0 then
                        ok := 1;
                        index_max := i.id_doctor;
                    end if;
            end;
        end loop;
        if ok = 0 then
            DBMS_OUTPUT.PUT_LINE('Acest animalut nu are nicio fisa medicala
in sistem');
        else
            id := cnt.FIRST;
            DBMS_OUTPUT.PUT_LINE('Acest animalut 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;
            end loop;
        end if;
    END afis_doctor_favorit;
END package_clinica_vet_tav13;
/
```

```

        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 is select a.nume,
                             p.nume,
                             p.prenume,
                             cursor (select diag.descriere,
fm.data_fisa
                                     from FISA_MEDICALA fm,
                                     DIAGNOSTIC diag
diag.id_diagnostic
                                     where fm.id_diagnostic =
fm.id_animalut
                                     and a.id_animalut =
                                     )
    from ANIMALUT a,
    PROPRIETAR p
    where a.id_proprietar = p.id_proprietar;
    refcursor;
    ANIMALUT.nume%type;
    PROPRIETAR.nume%type;
    PROPRIETAR.prenume%type;
    DIAGNOSTIC.descriere%type;
    FISA_MEDICALA.data_fisa%type;
    ANIMALUT.nume%type;
    PROPRIETAR.nume%type;
    PROPRIETAR.prenume%type;
    pls_integer := 0;
    faliment EXCEPTION;
BEGIN
    open mycursor;
    loop
        fetch mycursor into
nume_animalut,nume_proprietar,prenume_proprietar,v_cursor;
        exit when mycursor%notfound;
        fetch v_cursor into diag, data_diagnostic;
        if v_cursor%notfound then
            DBMS_OUTPUT.PUT_LINE(
nume_proprietar || ' ' ||
                                'Animalutul ' || nume_animalut || ' al lui ' ||
                                prenume_proprietar ||
                                ' nu a avut niciodata parte de o procedura
medicala');
            DBMS_OUTPUT.NEW_LINE();
        else
            DBMS_OUTPUT.PUT_LINE(
nume_proprietar || ' ' ||
                                'Animalutul ' || nume_animalut || ' al lui ' ||
                                prenume_proprietar ||
                                ' a avut urmatoarele interventii medicale:');
            loop
                exit when v_cursor%notfound;
                DBMS_OUTPUT.PUT_LINE(v_cursor%rowcount || ' ' || diag
|| ' la data de ' || data_diagnostic);
                fetch v_cursor into diag, data_diagnostic;

```

```

        end loop;
        if v_cursor%rowcount > co_max then
            co_max := v_cursor%rowcount;
            nume_animalut_max := nume_animalut;
            nume_proprietar_max := nume_proprietar;
            prenume_proprietar_max := prenume_proprietar;
        end if;
        DBMS_OUTPUT.NEW_LINE();
    end if;
end loop;
close mycursor;
if co_max = 0 then
    raise faliment;
end if;
DBMS_OUTPUT.PUT_LINE('Animalutul ' || nume_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;

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;

PROCEDURE factura_aferenta_fisa
IS
    id FACTURA_PROPRIETAR.id_factura%type;
BEGIN
    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
            into id
            from FACTURA_PROPRIETAR fp,
                 FACTURA f
            where fp.id_proprietar = i.id_proprietar
                  and fp.id_factura = f.id_factura
                  and i.data_fisa = f.data_emitere;
            DBMS_OUTPUT.PUT_LINE('Proprietarul animalutului ' ||
i.animnume || ' al lui ' ||
                                i.propnum || ' ' || i.prenume || '
are factura ' || id ||
                                ' emisa la data de ' ||
                                i.data_fisa || ', posibil pentru
fisa medicala ' || i.id_fisa);
        EXCEPTION
            when NO_DATA_FOUND then
                DBMS_OUTPUT.PUT_LINE(
                    'Nu s-a emis nicio factura pentru
proprietarul animalutului ' || i.animnume ||
                    ' al lui ' ||
                    i.propnum || ' ' || i.prenume || ' la
data de ' || i.data_fisa ||
                    ' desi a fost tratat, cu fisa medicala
cu id ' || i.id_fisa);
            when TOO_MANY_ROWS then
                DBMS_OUTPUT.PUT_LINE('Proprietarul animalutului '

```

```

|| i.animnume || ' al lui ' ||
||
|| i.propnum || ' ' || i.prenume
||
|| ' are mai multe facturi emise
la data de ' ||
|| i.data_fisa);

END;
end loop;
end factura_aferenta_fisa;

END package_clinica_vet_tav13;
/

begin
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;

DBMS_OUTPUT.PUT_LINE('.....');
package_clinica_vet_tav13.afis_tratamente_animalute();

DBMS_OUTPUT.PUT_LINE('.....');
package_clinica_vet_tav13.factura_aferenta_fisa();

DBMS_OUTPUT.PUT_LINE('.....');
package_clinica_vet_tav13.afis_doctor_favorit(1);
end;
/

```

```
end;
[2021-01-09 21:33:36] completed in 20 ms
[2021-01-09 21:33:36] Exista 2 clienti:
[2021-01-09 21:33:36] Staicu Octavian-Florin (id: 1)
[2021-01-09 21:33:36] Bodea George (id: 2)
[2021-01-09 21:33:36] .....
[2021-01-09 21:33:36] Animalutul Bit al lui Staicu Octavian-Florin a avut urmatoarele interventii medicale:
[2021-01-09 21:33:36] 1. Enterocolita la data de 13-07-2020
[2021-01-09 21:33:36] 2. Rana deschisa la data de 13-09-2020
[2021-01-09 21:33:36] 3. Enterocolita la data de 25-12-2020
[2021-01-09 21:33:36] Animalutul Kid al lui Staicu Adrian a avut urmatoarele interventii medicale:
[2021-01-09 21:33:36] 1. Vaccinare Leptospiroza la data de 13-07-2011
[2021-01-09 21:33:36] Animalutul Kitty al lui Bodea George a avut urmatoarele interventii medicale:
[2021-01-09 21:33:36] 1. Vaccinare Leptospiroza la data de 13-07-2016
[2021-01-09 21:33:36] 2. Rana deschisa la data de 13-12-2016
[2021-01-09 21:33:36] Animalutul Angelina al lui Cineva Interesant a avut urmatoarele interventii medicale:
[2021-01-09 21:33:36] 1. Vaccinare Leptospiroza la data de 13-07-2017
[2021-01-09 21:33:36] 2. Vaccinare Leptospiroza la data de 13-07-2020
[2021-01-09 21:33:36] Animalutul John al lui Popescu Ion a avut urmatoarele interventii medicale:
[2021-01-09 21:33:36] 1. Enterocolita la data de 04-05-2009
[2021-01-09 21:33:36] 2. Vaccinare Leptospiroza la data de 14-06-2010
[2021-01-09 21:33:36] 3. Rana deschisa la data de 14-08-2015
[2021-01-09 21:33:36] Animalutul dog al lui Cineva Interesant nu a avut niciodata parte de o procedura medicala
[2021-01-09 21:33:36] Animalutul Thor al lui Staicu Octavian-Florin a avut urmatoarele interventii medicale:
[2021-01-09 21:33:36] 1. Vaccinare Leptospiroza la data de 13-09-2020
[2021-01-09 21:33:36] 2. Vaccinare Leptospiroza la data de 13-07-2021
[2021-01-09 21:33:36] Animalutul Bit al lui Staicu Octavian-Florin este unul dintre cei mai fideli clienti ai nostri: ne-a vizitat de 3 ori :)
[2021-01-09 21:33:36] .....
[2021-01-09 21:33:36] Proprietarul animalutului Bit al lui Staicu Octavian-Florin are factura 1 emisa la data de 13-07-2020, posibil pentru fisa medicala 1
[2021-01-09 21:33:36] Proprietarul animalutului Kid al lui Staicu Adrian are factura 2 emisa la data de 13-07-2011, posibil pentru fisa medicala 2
[2021-01-09 21:33:36] Nu s-a emis nicio factura pentru proprietarul animalutului Kitty al lui Bodea George la data de 13-07-2016 desi a fost tratat, cu fisa medicala cu id 3

Problems  Terminal  Profiler  Services
[2021-01-09 21:33:36] Animalutul Bit al lui Staicu Octavian-Florin este unul dintre cei mai fideli clienti ai nostri: ne-a vizitat de 3 ori :)
[2021-01-09 21:33:36] .....
[2021-01-09 21:33:36] Proprietarul animalutului Bit al lui Staicu Octavian-Florin are factura 1 emisa la data de 13-07-2020, posibil pentru fisa medicala 1
[2021-01-09 21:33:36] Proprietarul animalutului Kid al lui Staicu Adrian are factura 2 emisa la data de 13-07-2011, posibil pentru fisa medicala 2
[2021-01-09 21:33:36] Nu s-a emis nicio factura pentru proprietarul animalutului Kitty al lui Bodea George la data de 13-07-2016 desi a fost tratat, cu fisa medicala cu id 3
[2021-01-09 21:33:36] Proprietarul animalutului Bit al lui Staicu Octavian-Florin are factura 4 emisa la data de 13-09-2020, posibil pentru fisa medicala 4
[2021-01-09 21:33:36] Proprietarul animalutului Angelina al lui Cineva Interesant are factura 5 emisa la data de 13-07-2017, posibil pentru fisa medicala 5
[2021-01-09 21:33:36] Proprietarul animalutului John al lui Popescu Ion are factura 6 emisa la data de 04-05-2009, posibil pentru fisa medicala 6
[2021-01-09 21:33:36] Proprietarul animalutului John al lui Popescu Ion are factura 7 emisa la data de 14-06-2010, posibil pentru fisa medicala 7
[2021-01-09 21:33:36] Proprietarul animalutului Angelina al lui Cineva Interesant are factura 8 emisa la data de 13-07-2020, posibil pentru fisa medicala 8
[2021-01-09 21:33:36] Proprietarul animalutului Kitty al lui Bodea George are factura 9 emisa la data de 13-12-2016, posibil pentru fisa medicala 9
[2021-01-09 21:33:36] Proprietarul animalutului Bit al lui Staicu Octavian-Florin are factura 10 emisa la data de 25-12-2020, posibil pentru fisa medicala 10
[2021-01-09 21:33:36] Nu s-a emis nicio factura pentru proprietarul animalutului John al lui Popescu Ion la data de 14-08-2015 desi a fost tratat, cu fisa medicala cu id 11
[2021-01-09 21:33:36] Proprietarul animalutului Thor al lui Staicu Octavian-Florin are factura 4 emisa la data de 13-09-2020, posibil pentru fisa medicala 13
[2021-01-09 21:33:36] Proprietarul animalutului Thor al lui Staicu Octavian-Florin are mai multe facturi emise la data de 13-07-2021
[2021-01-09 21:33:36] .....
[2021-01-09 21:33:36] Acest animalut a mers de cele mai multe ori la acesti doctori:
[2021-01-09 21:33:36] 1. Staicu Clara
```

```

--14
CREATE OR REPLACE PACKAGE package_clinica_vet_tav14 AS
    top_clienti tab_imb := tab_imb();
    FUNCTION topclienti RETURN PLS_INTEGER;
    PROCEDURE afis_topclienti;
    PROCEDURE afis_tratamente_animalute;
    TYPE refcursor IS REF CURSOR;
    type rec is record
        (
            nume_animalut      ANIMALUT.nume%type,
            nume_proprietar    PROPRIETAR.nume%type,
            prenume_proprietar PROPRIETAR.prenume%type
        );
END package_clinica_vet_tav14;
/

CREATE OR REPLACE PACKAGE BODY package_clinica_vet_tav14 AS

    FUNCTION topclienti RETURN PLS_INTEGER
    IS
        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
        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
    END

```

```

        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
    cursor mycursor is select a.nume,
                           p.nume,
                           p.prenume,
                           cursor (select diag.descriere,
fm.data_fisa
                                from FISA_MEDICALA fm,
                                DIAGNOSTIC diag
                                where fm.id_diagnostic =
diag.id_diagnostic
                                and a.id_animalut =
fm.id_animalut
                                )
        from ANIMALUT a,
        PROPRIETAR p
        where a.id_proprietar = p.id_proprietar
        and p.id_proprietar in (select * from table
(top_clienti));
    v_cursor          refcursor;
    diag              DIAGNOSTIC.descriere%type;
    data_diagnostic   FISA_MEDICALA.data_fisa%type;
    myrec             rec;
    nume_animalut_max ANIMALUT.nume%type;
    nume_proprietar_max PROPRIETAR.nume%type;
    prenume_proprietar_max PROPRIETAR.prenume%type;
    co_max            pls_integer := 0;
    faliment EXCEPTION;
BEGIN
    open mycursor;
    loop
        fetch mycursor into myrec.nume_animalut, myrec.nume_proprietar,
myrec.prenume_proprietar, v_cursor;
        exit when mycursor%notfound;
        fetch v_cursor into diag, data_diagnostic;
        if v_cursor%notfound then
            DBMS_OUTPUT.PUT_LINE(
                'Animalutul ' || myrec.nume_animalut || ' al
lui ' || myrec.nume_proprietar || ' ' ||
                myrec.prenume_proprietar ||
                ' nu a avut niciodata parte de o procedura
medicala');
            DBMS_OUTPUT.NEW_LINE();
        else
            DBMS_OUTPUT.PUT_LINE(
                'Animalutul ' || myrec.nume_animalut || ' al
lui ' || myrec.nume_proprietar || ' ' ||
                myrec.prenume_proprietar ||
                ' a avut urmatoarele interventii medicale:');
            loop
                exit when v_cursor%notfound;
                DBMS_OUTPUT.PUT_LINE(v_cursor%rowcount || ' ' || diag

```



```

|| ' la data de ' || data_diagnostic);
    fetch v_cursor into diag, data_diagnostic;
end loop;
if v_cursor%rowcount > co_max then
    co_max := v_cursor%rowcount;
    nume_animalut_max := myrec.nume_animalut;
    nume_proprietar_max := myrec.nume_proprietar;
    prenume_proprietar_max := myrec.prenume_proprietar;
end if;
    DBMS_OUTPUT.NEW_LINE();
end if;
end loop;
close mycursor;
if co_max = 0 then
    raise faliment;
end if;
    DBMS_OUTPUT.PUT_LINE('Animalutul ' || nume_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) || ')');
        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;
/

```

```
tav_project_2020: begin
    package_clinica_vet_tav14.afis_topclienti;
end;
[2021-01-09 21:37:02] completed in 13 ms
[2021-01-09 21:37:02] Exista 2 clienti:
[2021-01-09 21:37:02] Staicu Octavian-Florin (id: 1)
[2021-01-09 21:37:02] Bodea George (id: 2)
[2021-01-09 21:37:02] Animalutele acestora au folosit cel mai des aceste tratamente:
[2021-01-09 21:37:02] Animalutul Bit al lui Staicu Octavian-Florin a avut urmatoarele interventii medicale:
[2021-01-09 21:37:02] 1. Enterocolita la data de 13-07-2020
[2021-01-09 21:37:02] 2. Rana deschisa la data de 13-09-2020
[2021-01-09 21:37:02] 3. Enterocolita la data de 25-12-2020
[2021-01-09 21:37:02] Animalutul Kitty al lui Bodea George a avut urmatoarele interventii medicale:
[2021-01-09 21:37:02] 1. Vaccinare Leptospiroza la data de 13-07-2016
[2021-01-09 21:37:02] 2. Rana deschisa la data de 13-12-2016
[2021-01-09 21:37:02] Animalutul Thor al lui Staicu Octavian-Florin a avut urmatoarele interventii medicale:
[2021-01-09 21:37:02] 1. Vaccinare Leptospiroza la data de 13-09-2020
[2021-01-09 21:37:02] 2. Vaccinare Leptospiroza la data de 13-07-2021
[2021-01-09 21:37:02] Animalutul Bit al lui Staicu Octavian-Florin este unul dintre cei mai fideli clienti ai nostri: ne-a vizitat de 3 ori :)
```

```

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;
            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;
                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;
/

```

```

TAV_PROJECT_SGBD> begin
package_clinica_vet_utilitare.afis_datorii;
end;

```

```

[2021-01-09 21:39:34] completed in 6 ms
[2021-01-09 21:39:34] Proprietarul Staicu Octavian-Florin (id:1):
[2021-01-09 21:39:34] La factura 105 mai are de platit: 300
[2021-01-09 21:39:34] Total de plata: 300
[2021-01-09 21:39:34] Proprietarul Bodea George (id:2):
[2021-01-09 21:39:34] Total de plata: 0
[2021-01-09 21:39:34] Proprietarul Staicu Adrian (id:3):
[2021-01-09 21:39:34] Total de plata: 0
[2021-01-09 21:39:34] Proprietarul Popescu Ion (id:4):
[2021-01-09 21:39:34] Total de plata: 0
[2021-01-09 21:39:34] Proprietarul Cineva Interesant (id:5):
[2021-01-09 21:39:34] La factura 1000 mai are de platit: 1000
[2021-01-09 21:39:34] Total de plata: 999

```

```

[2021-01-09 21:39:54] Total de plata: ???

```

```

TAV_PROJECT_SGBD> 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;

```

```

[2021-01-09 21:39:56] completed in 8 ms
[2021-01-09 21:39:56] 1. 5
[2021-01-09 21:39:56] 2. 1
[2021-01-09 21:39:56] 3. 2
[2021-01-09 21:39:56] 4. 4
[2021-01-09 21:39:56] 5. 3

```