

# PROIECT

## -GESTIUNEA GRADINILOR ZOOLOGICE-

Putinelu Andrei Bogdan

Grupa 233

### **1. Prezentări pe scurt baza de date (utilitatea ei).**

Modelul gestioneaza activitatea mai multor gradini zoologice. Se gestioneaza mai multe aspecte ale gradinilor zoologice precum:

- Angajatii gradinilor zoologice si ce pozitii ocupa
- Produsele vandute (de exemplu bilete pentru spectacole, bilete de intrare, bilete cu reduceri speciale, consumabile precum apa, popcorn etc.)
- Hrana pentru animale detinuta de gradini
- Animalele ce apartin gradinilor zoologice, detalii despre acestea, precum si eventuale restrictii in urma unor evenimente neprevazute (de exemplu dupa o vizita medicala un animal trebuie sa stea la repaus 2 zile)
- Partenerii cu care gradinile zoologice fac tranzactii, acestea pot cumpara servicii (de exemplu hartie pentru bilete), acestea pot vinde servicii (de exemplu un spectacol cu animale privat) si pot primi donatii de la acesti parteneri (de exemplu hrana, bani, medicamente pentru animale etc.)
- Tranzactiile facute cu partenerii gradinilor zoologice
- Hranirea animalelor facuta de catre angajati.

Utilitatea modelului consta in gestionarea mai usoara a acestor gradini zoologice si a activitatilor zilnice ce se petrec in acestea. Ingrijirea animalelor se realizeaza mai

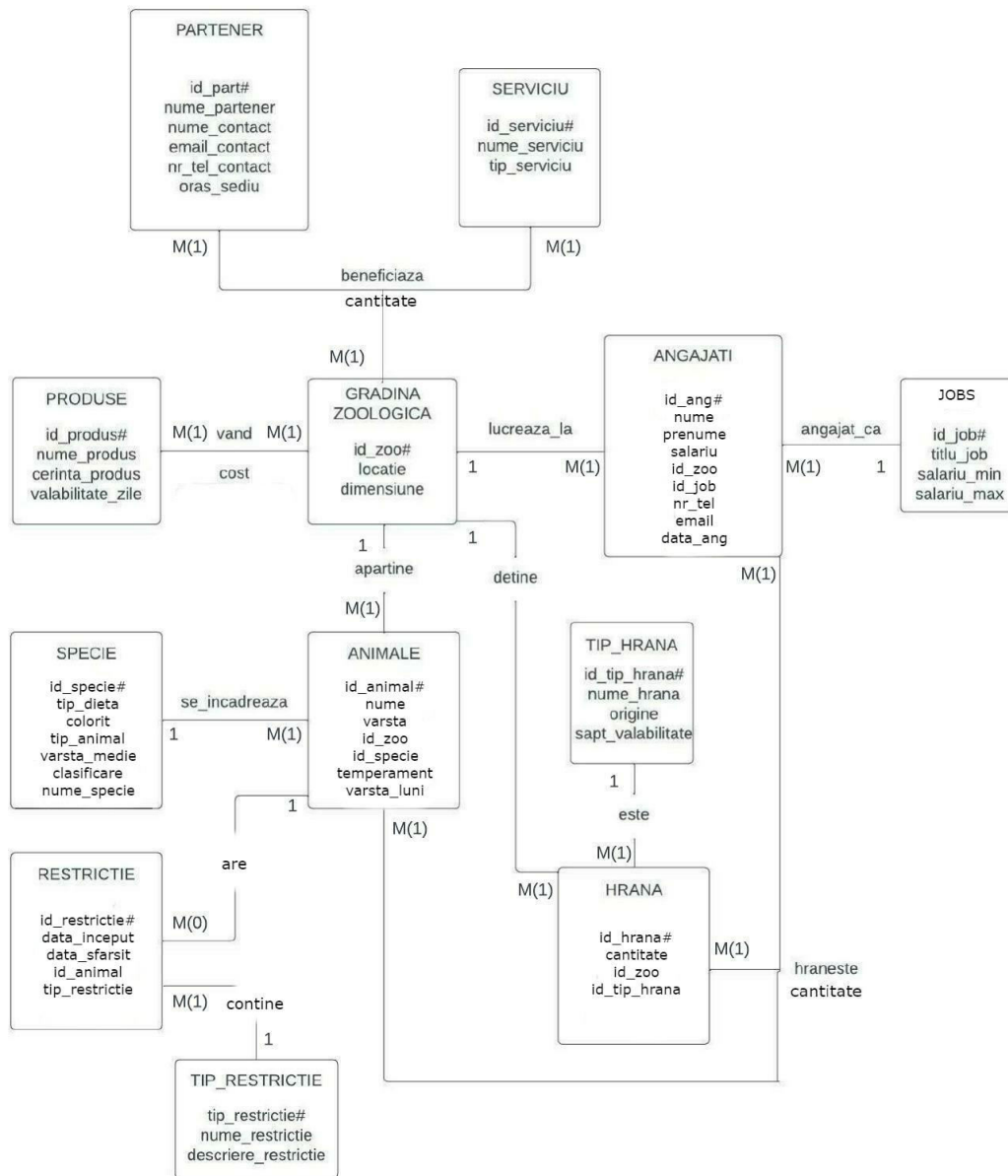
usor, baza de date avand un istoric de hraniri ale acestora care arata ce animale au fost hranite, care a fost angajatul care le-a hranit, precum si hrana pe care au primit-o. De asemenea, exista un tabel cu eventuale restrictii pe care un animal trebuie sa le urmeze, si pe care un angajat trebuie sa le respecte (de exemplu sa nu hraneasca un animal care are restrictie de a nu manca nimic pentru 1 zi). Desigur, utilitatea modelului consta si in gestionarea angajatilor, produselor lor, activitatilor cu alti parteneri (precum a fost descris mai sus).

Constrângeri (restrictii, reguli) impuse asupra modelului:

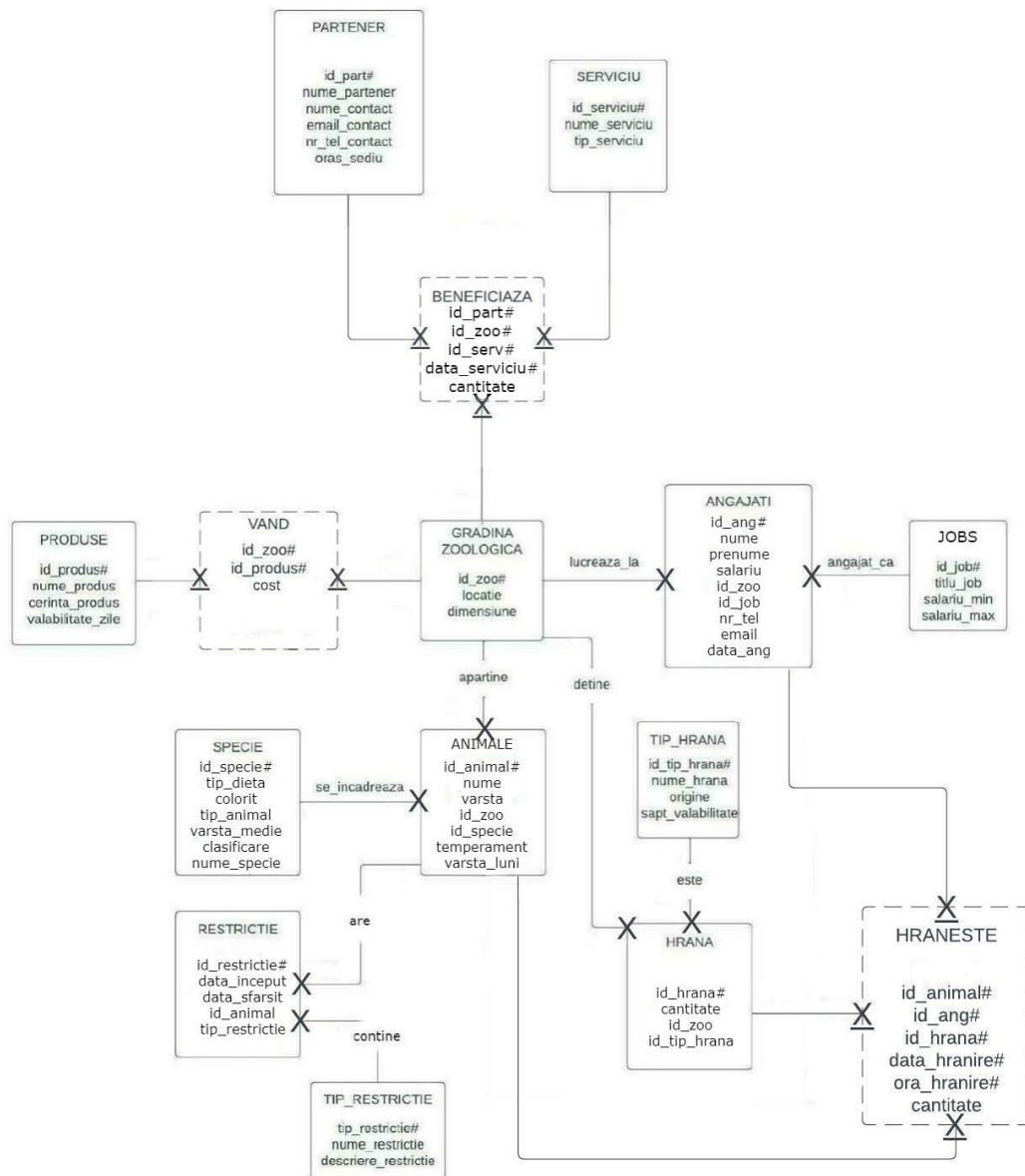
- Gradinile zoologice pot lucra cu mai multi parteneri si pot beneficia de mai multe servicii.
- Un partener poate lucra cu mai multe gradini zoologice si poate oferi mai multe servicii.
- Un serviciu poate fi oferit mai multor gradini zoologice de catre mai multi parteneri.
- Deoarece nicio gradina zoologica nu este autonoma, fiecare gradina zoologica va avea minim un partener si va beneficia de minim un serviciu.
- Mai multe produse pot fi vandute de mai multe gradini zoologice (diferenta constand in pret)
- Un angajat poate fi angajat doar la o gradina zoologica si ocupa o singura pozitie.
- Un animal poate apartine unei singure gradini zoologice, se incadreaza intr-o singura specie.
- Un animal poate sa nu aiba nicio restrictie sau poate sa aiba mai multe restrictii.
- O gradina zoologica are minim un angajat si minim un animal.
- O gradina zoologica poate sa aiba mai multe animale de aceeasi specie.
- O gradina zoologica nu o sa ramana niciodata fara hrana, avand mereu minim un fel de hrana.
- O hrana poate fi de un singur tip de hrana.
- Deoarece se pot intampla situatii neprevazute, orice angajat se poate ocupa de hranirea animalelor, insa, in majoritatea cazurilor aceasta sarcina va fii indeplinita de catre ingrijitori.

- Animalele pot fi hranite de mai multe ori in decursul unei zile, de diferiti angajati, cu mai multe feluri de hrana.
- O gradina zoologica poate avea mai multi angajati cu acelasi rol si trebuie sa aiba minim un angajat cu rol de ingrijitor.
- Un partener are o singura persoana de contact, cu care gradinile zoologice comunica.
- Un produs are o singura cerinta pentru a putea fi achizitionat (de exemplu bilet elev are ca cerinta doar carnetul de elev).
- Valabilitatea unui tip de hrana este unica pentru acel tip de hrana.
- Descrierea unei restrictii este unica pentru restrictia respectiva.
- Numele retinut al persoanei de contact al partenerilor va fi numele de familie.
- O gradina zoologica nu poate beneficia de acelasi serviciu, realizat de catre acelasi partener de mai multe ori in aceeasi zi.
- Un animal nu poate avea mai multe restrictii de acelasi tip in acelasi timp.

## 2. Realizați diagrama entitate-relație (ERD).



3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare.



#### 4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

```
--entitatea PARTENER
CREATE TABLE partener
(
    id_part          NUMBER(4) CONSTRAINT pk_part PRIMARY KEY,
    nume_partener    VARCHAR2(30) CONSTRAINT null_nume_partener NOT NULL,
    nume_contact     VARCHAR2(25) CONSTRAINT null_nume_contact NOT NULL,
    email_contact    VARCHAR2(30) CONSTRAINT null_email NOT NULL CONSTRAINT
    unq_mail
    UNIQUE,
    nr_tel_contact   VARCHAR2(15) CONSTRAINT unq_tel UNIQUE,
    oras_sediu       VARCHAR2(25)
);

--entitatea SERVICIU
CREATE TABLE serviciu
(
    id_serviciu      NUMBER(4) CONSTRAINT pk_serv PRIMARY KEY,
    nume_serviciu    VARCHAR2(60) CONSTRAINT null_serv NOT NULL,
    tip_serviciu     VARCHAR2(15) CONSTRAINT null_tip NOT NULL CONSTRAINT
    check_tip
    CHECK(LOWER(tip_serviciu) IN (
    'donatie', 'achizitionare', 'vanzare'))
);

--entitatea GRADINA_ZOOLOGICA
CREATE TABLE gradina_zoologica
(
    id_zoo           NUMBER(4) CONSTRAINT pk_zoo PRIMARY KEY,
    locatie          VARCHAR2(30) CONSTRAINT null_locatie NOT NULL,
    dimensiune       NUMBER(4)
);

--entitatea BENEFICIAZA
CREATE TABLE beneficiaza
(
    id_part          NUMBER(4) CONSTRAINT fk_part REFERENCES partener(id_part) ON
    DELETE
    CASCADE,
    id_zoo           NUMBER(4) CONSTRAINT fk_zoo REFERENCES gradina_zoologica(
    id_zoo) ON
    DELETE CASCADE,
    id_serviciu      NUMBER(4) CONSTRAINT fk_serv REFERENCES serviciu(id_serviciu)
    ON
    DELETE CASCADE,
    data_serviciu    DATE DEFAULT sysdate,
    cantitate        VARCHAR2(30) CONSTRAINT null_cantitate NOT NULL,
    CONSTRAINT pk_part_zoo_serv_data PRIMARY KEY(id_part, id_zoo, id_serviciu,
    data_serviciu)
);

--entitatea PRODUSE
CREATE TABLE produse
(
    id_produs        NUMBER(4) CONSTRAINT pk_produs PRIMARY KEY,
    nume_produs      VARCHAR2(30) CONSTRAINT null_nume NOT NULL,
    cerinta_produs   VARCHAR2(30),
```

```

        valabilitate_zile NUMBER(4) CONSTRAINT null_valabilitate NOT NULL
        CONSTRAINT
        check_valabilitate CHECK(
        valabilitate_zile>0)
    );

--entitatea VAND
CREATE TABLE vand
(
    id_zoo      NUMBER(4) CONSTRAINT fk_zoo_vand REFERENCES gradina_zoologica(
    id_zoo) ON
    DELETE CASCADE,
    id_produc NUMBER(4) CONSTRAINT fk_produc REFERENCES produse(id_produc) ON
    DELETE
    CASCADE,
    cost        NUMBER(4) CONSTRAINT null_cost NOT NULL CONSTRAINT check_cost
    CHECK(cost>0),
    CONSTRAINT pk_zoo_produc PRIMARY KEY(id_zoo, id_produc)
);

--entitatea JOBS
CREATE TABLE jobs
(
    id_job      VARCHAR2(20) CONSTRAINT pk_job PRIMARY KEY,
    titlu_job   VARCHAR2(35) CONSTRAINT null_titlu NOT NULL,
    salariu_min NUMBER(8) CONSTRAINT null_sal_min NOT NULL CONSTRAINT
    check_sal_min
    CHECK(salariu_min>0),
    salariu_max NUMBER(8) CONSTRAINT null_sal_max NOT NULL,
    CONSTRAINT check_sal_max CHECK(salariu_max>salariu_min)
);

--entitatea ANGAJATI
CREATE TABLE angajati
(
    id_ang      NUMBER(4) CONSTRAINT pk_angajati PRIMARY KEY,
    nume        VARCHAR2(30) CONSTRAINT null_nume_ang NOT NULL,
    prenume     VARCHAR2(30),
    salariu     NUMBER(8) CONSTRAINT null_salariu_ang NOT NULL CONSTRAINT
    check_salariu_ang CHECK(salariu>
    0),
    id_zoo      NUMBER(4) CONSTRAINT null_zoo_ang NOT NULL CONSTRAINT fk_zoo_ang
    REFERENCES
    gradina_zoologica(id_zoo) ON DELETE CASCADE,
    id_job      VARCHAR2(20) CONSTRAINT null_job_ang NOT NULL CONSTRAINT
    fk_job_ang
    REFERENCES jobs(id_job)
    ON DELETE CASCADE,
    nr_tel      VARCHAR2(15) CONSTRAINT unq_tel_ang UNIQUE,
    email       VARCHAR2(30) CONSTRAINT unq_email_ang UNIQUE CONSTRAINT
    null_email_ang NOT
    NULL,
    data_ang    DATE DEFAULT sysdate
);

--entitatea SPECIE
CREATE TABLE specie
(
    id_specie   VARCHAR2(25) CONSTRAINT pk_specie PRIMARY KEY,
    tip_dieta   VARCHAR2(10) CONSTRAINT null_dieta NOT NULL CONSTRAINT
    check_dieta
    CHECK(Lower(tip_dieta) IN
    ('carnivor', 'erbivor', 'omnivor')),

```

```

        colorit          VARCHAR2(60),
        tip_animal       VARCHAR2(20) CONSTRAINT null_tip_sp NOT NULL CONSTRAINT
        check_tip_sp
        CHECK(LOWER(tip_animal)
        IN ('nocturn', 'diurn', 'crepuscular')),
        varsta_medie     NUMBER(4) CONSTRAINT null_varsta NOT NULL,
        clasificare      VARCHAR2(20) CONSTRAINT null_clasificare NOT NULL,
        nume_specie      VARCHAR2(35) CONSTRAINT null_nume_sp NOT NULL
    );

--entitatea ANIMALE
CREATE TABLE animale
(
    id_animal            NUMBER(4) CONSTRAINT pk_animal PRIMARY KEY,
    nume                 VARCHAR2(25) DEFAULT 'No Name',
    varsta              NUMBER(4) CONSTRAINT null_varsta_anim NOT NULL CONSTRAINT
    check_varsta
    CHECK(varsta>=0),
    id_zoo              NUMBER(4) CONSTRAINT null_zoo_anim NOT NULL CONSTRAINT
    fk_zoo_anim
    REFERENCES
    gradina_zoologica(id_zoo) ON DELETE CASCADE,
    id_specie           VARCHAR2(25) CONSTRAINT null_specie NOT NULL CONSTRAINT
    fk_specie
    REFERENCES specie(id_specie
    ) ON DELETE CASCADE,
    temperament         VARCHAR2(60),
    varsta_luni         NUMBER(4) CONSTRAINT null_varsta_luni NOT NULL CONSTRAINT
    check_varsta_luni CHECK(
    varsta_luni>=0 AND varsta_luni<12)
);

--entitatea TIP RESTRICTIE
CREATE TABLE tip_restrictie
(
    tip_restrictie      VARCHAR2(20) CONSTRAINT pk_tip_restrictie PRIMARY KEY,
    nume_restrictie     VARCHAR2(30) CONSTRAINT null_nume_restrictie NOT NULL,
    descriere_restrictie VARCHAR2(80) CONSTRAINT null_descriere NOT NULL
);

--entitatea RESTRICTIE
CREATE TABLE restrictie
(
    id_restrictie       NUMBER(4) CONSTRAINT pk_restrictie PRIMARY KEY,
    data_inceput        DATE DEFAULT sysdate CONSTRAINT null_data_inc NOT NULL,
    data_sfarsit        DATE DEFAULT sysdate CONSTRAINT null_data_sf NOT NULL,
    id_animal           NUMBER(4) CONSTRAINT null_animal NOT NULL CONSTRAINT
    fk_animal_rest
    REFERENCES animale(
    id_animal) ON DELETE CASCADE,
    tip_restrictie      VARCHAR2(20) CONSTRAINT null_tip_rest NOT NULL CONSTRAINT
    fk_tip_rest
    REFERENCES
    tip_restrictie(tip_restrictie) ON DELETE CASCADE,
    CONSTRAINT check_data_sf CHECK(data_sfarsit>=data_inceput)
);

--entitatea TIP HRANA
CREATE TABLE tip_hrana
(
    id_tip_hrana        VARCHAR2(20) CONSTRAINT pk_tip_hrana PRIMARY KEY,
    nume_hrana          VARCHAR2(30) CONSTRAINT null_nume_tip NOT NULL,
    origine             VARCHAR2(30) CONSTRAINT null_origine_tip NOT NULL,
    sapt_valabilitate   NUMBER(4) CONSTRAINT null_valabilitate_tip NOT NULL

```



```

        CONSTRAINT
        check_valabilitate_tip
        CHECK(sapt_valabilitate>0)
    );

--entitatea HRANA
CREATE TABLE hrana
(
    id_hrana      NUMBER(4) CONSTRAINT pk_hrana PRIMARY KEY,
    cantitate     NUMBER(4) CONSTRAINT null_cantitate_hrana NOT NULL CONSTRAINT
    check_cantitate CHECK(
    cantitate>0),
    id_zoo        NUMBER(4) CONSTRAINT null_zoo_hrana NOT NULL CONSTRAINT
    fk_zoo_hrana  REFERENCES
    gradina_zoologica(id_zoo) ON DELETE CASCADE,
    id_tip_hrana  VARCHAR2(20) CONSTRAINT null_tip_hrana NOT NULL CONSTRAINT
    fk_tip_hrana  REFERENCES tip_hrana(
    id_tip_hrana) ON DELETE CASCADE
);

--entitatea HRANESTE
CREATE TABLE hraneste
(
    id_animal     NUMBER(4) CONSTRAINT fk_animal_hraneste REFERENCES animale(
    id_animal)
    ON DELETE CASCADE,
    id_ang        NUMBER(4) CONSTRAINT fk_ang_hraneste REFERENCES angajati(
    id_ang) ON
    DELETE CASCADE,
    id_hrana      NUMBER(4) CONSTRAINT fk_hrana_hraneste REFERENCES hrana(
    id_hrana) ON
    DELETE CASCADE,
    data_hranire  DATE,
    ora_hranire   DATE,
    cantitate     NUMBER(4) CONSTRAINT null_cantitate_hraneste NOT NULL
    CONSTRAINT
    check_cantitate_hraneste
    CHECK(cantitate>0),
    CONSTRAINT pk_animal_ang_hrana_data_ora PRIMARY KEY(id_animal, id_ang,
    id_hrana, data_hranire, ora_hranire)
);

--entitatea NUMAR_HRANA pentru exercitiul 10
CREATE TABLE numar_hrana
(
    id_zoo        NUMBER(4) CONSTRAINT pk_numar_hrana PRIMARY KEY,
    nr            NUMBER(4),
    CONSTRAINT fk_zoo_nr_hrana FOREIGN KEY (id_zoo) REFERENCES
    gradina_zoologica(id_zoo)
);

--entitatea EVENIMENTE pentru exercitiul 12
CREATE TABLE evenimente
(
    username      VARCHAR2(50),
    data_eveniment DATE,
    actiune       VARCHAR2(30),
    nume          VARCHAR2(30)
);

```

```
-- 4) Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele,
-- implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

--entitatea PARTENER
create table PARTENER(
    id_part number(4) constraint pk_part primary key,
    nume_partener varchar2(30) constraint null_nume_partener not null,
    nume_contact varchar2(25) constraint null_nume_contact not null,
    email_contact varchar2(30) constraint null_email not null constraint unq_email unique,
    nr_tel_contact varchar2(15) constraint unq_tel unique,
    oras_sediu varchar2(25)
);

--entitatea SERVICIU
create table SERVICIU(
    id_serviciu number(4) constraint pk_serv primary key,
    nume_serviciu varchar2(60) constraint null_serv not null,
    tip_serviciu varchar2(15) constraint null_tip not null constraint check_tip check(lower(tip_serviciu) in ('donatie','achizitionare','vanzare'))
);

--entitatea GRADINA_ZOOLOGICA
create table GRADINA_ZOOLOGICA(
    id_zoo number(4) constraint pk_zoo primary key,
    locatie varchar2(30) constraint null_locatie not null,
    dimensiune number(4)
);

--entitatea BENEFICIAZA
create table BENEFICIAZA(
    id_part number(4) constraint fk_part references PARTENER(id_part) on delete cascade,
```

Query Result x Script Output x

Task completed in 0.414 seconds

Table PARTENER created.

Table SERVICIU created.

Table GRADINA\_ZOOLOGICA created.

Table BENEFICIAZA created.

Table PRODUSE created.

## 5. Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

- Pentru inserarea înregistrărilor am folosit și secvențe.
- De menționat este faptul că o bună parte a comenzilor de insert au fost generate de niște script-uri scrise în python de mine (acestea vor fi incluse la trimiterea proiectului împreună cu un fișier .txt care explică cum se folosesc).
- Deoarece sunt mult prea multe comenzi de insert o să includ în documentație doar câteva.

```

--secventa care va ajuta in inserarea inregistrarilor in GRADINA_ZOOLOGICA
CREATE SEQUENCE secv_zoo
  INCREMENT BY 1
  START WITH 1
  MAXVALUE 9999
  NOCYCLE
  NOCACHE;

--secventa care va ajuta in inserarea inregistrarilor in ANIMALE
CREATE SEQUENCE secv_animale
  INCREMENT BY 1
  START WITH 100
  MAXVALUE 9999
  NOCYCLE
  NOCACHE;

--secventa care va ajuta in inserarea inregistrarilor in PARTENER
CREATE SEQUENCE secv_part
  INCREMENT BY 10
  START WITH 10
  MAXVALUE 9999
  NOCYCLE
  NOCACHE;

--secventa care va ajuta in inserarea inregistrarilor in SERVICIU
CREATE SEQUENCE secv_serv
  INCREMENT BY 5
  START WITH 200
  MAXVALUE 9999
  NOCYCLE
  NOCACHE;

--secventa care va ajuta in inserarea inregistrarilor in PRODUSE
CREATE SEQUENCE secv_produus
  INCREMENT BY 10
  START WITH 1000
  MAXVALUE 9999
  NOCYCLE
  NOCACHE;

--secventa care va ajuta in inserarea inregistrarilor in ANGAJATI
CREATE SEQUENCE secv_angajat
  INCREMENT BY 1
  START WITH 500
  MAXVALUE 9999
  NOCYCLE
  NOCACHE;

--secventa care va ajuta in inserarea inregistrarilor in RESTRICTIE
CREATE SEQUENCE secv_restrictie
  INCREMENT BY 1
  START WITH 300
  MAXVALUE 9999
  NOCYCLE
  NOCACHE;

--secventa care va ajuta in inserarea inregistrarilor in HRANA
CREATE SEQUENCE secv_hrana

```

```

INCREMENT BY 1
START WITH 700
MAXVALUE 9999
NOCYCLE
NOCACHE;

--inseram inregistrari in PARTENER
INSERT INTO partener
    (id_part,nume_partener,nume_contact,email_contact,nr_tel_contact,
    oras_sediu)
VALUES
    (
secv_part.nextval, 'SUFIPAPER', 'Smith', 'jason.smith@gmail.com', '0772561483', 'Sydney');

INSERT INTO partener
    (id_part,nume_partener,nume_contact,email_contact,nr_tel_contact,
    oras_sediu)
VALUES
    (
secv_part.nextval, 'MEROS', 'Jameson', 'ashley.jameson@gmail.com', '0751483301', 'New York');

INSERT INTO partener
    (id_part,nume_partener,nume_contact,email_contact,nr_tel_contact,
    oras_sediu)
VALUES
    (
secv_part.nextval, 'KINGUS', 'Austin', 'bruce.austin@gmail.com', NULL, 'Tokyo');

INSERT INTO partener
    (id_part,nume_partener,nume_contact,email_contact,nr_tel_contact,
    oras_sediu)
VALUES
    (
secv_part.nextval, 'CRONEKER', 'Mikks', 'brad.mikks@gmail.com', '0772561553', NULL);

INSERT INTO partener
    (id_part,nume_partener,nume_contact,email_contact,nr_tel_contact,
    oras_sediu)
VALUES
    (
secv_part.nextval, 'GRAEL', 'Williams', 'judy.williams@gmail.com', '0760161273', 'San Francisco');

INSERT INTO partener
    (id_part,nume_partener,nume_contact,email_contact,nr_tel_contact,
    oras_sediu)
VALUES
    (
secv_part.nextval, 'TANG', 'Brown', 'rene.brown@gmail.com', '0762351104', 'Mumbai');

INSERT INTO partener
    (id_part,nume_partener,nume_contact,email_contact,nr_tel_contact,
    oras_sediu)
VALUES
    (
secv_part.nextval, 'BLICH', 'Walker', 'jason.walker@gmail.com', '0789210345', NULL);

INSERT INTO partener
    (id_part,nume_partener,nume_contact,email_contact,nr_tel_contact,
    oras_sediu)
VALUES
    (
secv_part.nextval, 'ROWNE', 'Singer', 'robert.singer@gmail.com', NULL, 'Rotterdam');

INSERT INTO partener
    (id_part,nume_partener,nume_contact,email_contact,nr_tel_contact,
    oras_sediu)
VALUES
    (
secv_part.nextval, 'BEGAS', 'Marconi', 'julie.marconi@gmail.com', NULL, 'Paris');

INSERT INTO partener
    (id_part,nume_partener,nume_contact,email_contact,nr_tel_contact,
    oras_sediu)
VALUES
    (

```

```

secv_part.nextval, 'FARKIN', 'Roosevelt', 'tom.roosevelt@gmail.com', '0733691024', 'Milan');

INSERT INTO partener
    (id_part, nume_partener, nume_contact, email_contact, nr_tel_contact,
     oras_sediu)
VALUES
    (
secv_part.nextval, 'REPPER', 'Kleinz', 'mauro.kleinz@gmail.com', '0763500792', 'Frankfurt');

INSERT INTO partener
    (id_part, nume_partener, nume_contact, email_contact, nr_tel_contact,
     oras_sediu)
VALUES
    (secv_part.nextval, 'SCHLET', 'Pope', 'rebecca.pope@gmail.com', NULL, NULL);

--inseram inregistrari in SERVICIU
INSERT INTO serviciu
    (id_serviciu, nume_serviciu, tip_serviciu)
VALUES
    (secv_serv.nextval, 'Hartie', 'Achizitionare');

INSERT INTO serviciu
    (id_serviciu, nume_serviciu, tip_serviciu)
VALUES
    (secv_serv.nextval, 'Bani', 'Donatie');

INSERT INTO serviciu
    (id_serviciu, nume_serviciu, tip_serviciu)
VALUES
    (secv_serv.nextval, 'Spectacol', 'Vanzare');

INSERT INTO serviciu
    (id_serviciu, nume_serviciu, tip_serviciu)
VALUES
    (secv_serv.nextval, 'Morcovi', 'Achizitionare');

INSERT INTO serviciu
    (id_serviciu, nume_serviciu, tip_serviciu)
VALUES
    (secv_serv.nextval, 'Suplimente', 'Achizitionare');

INSERT INTO serviciu
    (id_serviciu, nume_serviciu, tip_serviciu)
VALUES
    (secv_serv.nextval, 'Medicamente', 'Achizitionare');

INSERT INTO serviciu
    (id_serviciu, nume_serviciu, tip_serviciu)
VALUES
    (secv_serv.nextval, 'Medicamente', 'Donatie');

INSERT INTO serviciu
    (id_serviciu, nume_serviciu, tip_serviciu)
VALUES
    (secv_serv.nextval, 'Furaje', 'Achizitionare');

INSERT INTO serviciu
    (id_serviciu, nume_serviciu, tip_serviciu)
VALUES
    (secv_serv.nextval, 'Spectacol Acvatic', 'Vanzare');

INSERT INTO serviciu
    (id_serviciu, nume_serviciu, tip_serviciu)
VALUES
    (secv_serv.nextval, 'Eveniment Privat', 'Vanzare');

INSERT INTO serviciu
    (id_serviciu, nume_serviciu, tip_serviciu)
VALUES
    (secv_serv.nextval, 'Seminte', 'Achizitionare');

INSERT INTO serviciu
    (id_serviciu, nume_serviciu, tip_serviciu)
VALUES
    (secv_serv.nextval, 'Carne de vita', 'Achizitionare');

INSERT INTO serviciu
    (id_serviciu, nume_serviciu, tip_serviciu)
VALUES
    (secv_serv.nextval, 'Carne de oaie', 'Achizitionare');

```

```

INSERT INTO serviciu
(id_serviciu,nume_serviciu,tip_serviciu)
VALUES
(secv_serv.nextval,'Salata','Achizitionare');

INSERT INTO serviciu
(id_serviciu,nume_serviciu,tip_serviciu)
VALUES
(secv_serv.nextval,'Vitamine','Achizitionare');

INSERT INTO serviciu
(id_serviciu,nume_serviciu,tip_serviciu)
VALUES
(secv_serv.nextval,'Vitamine','Donatie');

INSERT INTO serviciu
(id_serviciu,nume_serviciu,tip_serviciu)
VALUES
(secv_serv.nextval,'Potcoave','Donatie');

--inseram inregistrari in GRADINA_ZOOLOGICA
INSERT INTO gradina_zoologica
(id_zoo,locatie,dimensiune)
VALUES
(secv_zoo.nextval,'Sydney',200);

INSERT INTO gradina_zoologica
(id_zoo,locatie,dimensiune)
VALUES
(secv_zoo.nextval,'Mumbai',200);

INSERT INTO gradina_zoologica
(id_zoo,locatie,dimensiune)
VALUES
(secv_zoo.nextval,'New York',200);

INSERT INTO gradina_zoologica
(id_zoo,locatie,dimensiune)
VALUES
(secv_zoo.nextval,'Tokyo',300);

INSERT INTO gradina_zoologica
(id_zoo,locatie,dimensiune)
VALUES
(secv_zoo.nextval,'San Francisco',350);

INSERT INTO gradina_zoologica
(id_zoo,locatie,dimensiune)
VALUES
(secv_zoo.nextval,'Singapore',360);

INSERT INTO gradina_zoologica
(id_zoo,locatie,dimensiune)
VALUES
(secv_zoo.nextval,'Kansas',195);

INSERT INTO gradina_zoologica
(id_zoo,locatie,dimensiune)
VALUES
(secv_zoo.nextval,'Bordeaux',400);

INSERT INTO gradina_zoologica
(id_zoo,locatie,dimensiune)
VALUES
(secv_zoo.nextval,'Bern',360);

INSERT INTO gradina_zoologica
(id_zoo,locatie,dimensiune)
VALUES
(secv_zoo.nextval,'Jakarta',400);

INSERT INTO gradina_zoologica
(id_zoo,locatie,dimensiune)
VALUES
(secv_zoo.nextval,'Detroit',250);

INSERT INTO gradina_zoologica
(id_zoo,locatie,dimensiune)
VALUES
(secv_zoo.nextval,'Marseille',270);

INSERT INTO gradina_zoologica

```

```

VALUES      (id_zoo,locatie,dimensiune)
VALUES      (secv_zoo.nextval,'Ankara',320);

INSERT INTO gradina_zoologica
VALUES      (id_zoo,locatie,dimensiune)
VALUES      (secv_zoo.nextval,'Boston',350);

--inseram inregistrari in BENEFICIAZA
INSERT INTO beneficiaza
VALUES      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (20,6,205,'11-MAY-2008','1000 euro');

INSERT INTO beneficiaza
VALUES      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (20,7,205,'11-MAY-2008','5000 euro');

INSERT INTO beneficiaza
VALUES      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (20,10,205,'11-MAY-2008','3500 euro');

INSERT INTO beneficiaza
VALUES      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (100,13,225,'23-JUNE-2015','10 flacoane');

INSERT INTO beneficiaza
VALUES      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (30,4,255,'18-NOV-2002','100 kilograme');

INSERT INTO beneficiaza
VALUES      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (30,9,255,'20-DEC-2012','50 kilograme');

INSERT INTO beneficiaza
VALUES      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (30,8,255,'18-JAN-2006','150 kilograme');

INSERT INTO beneficiaza
VALUES      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (110,1,250,'11-AUG-2002','30 kilograme');

INSERT INTO beneficiaza
VALUES      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (110,5,250,'11-JUL-2002','20 kilograme');

INSERT INTO beneficiaza
VALUES      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (110,14,250,'20-AUG-2006','35 kilograme');

INSERT INTO beneficiaza
VALUES      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (70,3,245,'21-APR-2013','1 eveniment');

INSERT INTO beneficiaza
VALUES      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (70,2,265,'14-SEP-2011','30 kilograme');

INSERT INTO beneficiaza
VALUES      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (90,10,270,'17-OCT-2001','20 flacoane');

INSERT INTO beneficiaza
VALUES      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (90,10,270,'17-OCT-2001','20 flacoane');

INSERT INTO beneficiaza
VALUES      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)

```

```

VALUES      (10,10,200,'17-MAR-2021','3500 bucati');

INSERT INTO beneficiaza
      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (10,7,200,'18-MAR-2021','700 bucati');

INSERT INTO beneficiaza
      (id_part,id_zoo,id_serviciu,data_serviciu,cantitate)
VALUES      (10,3,200,'22-MAR-2010','6000 bucati');

--inseram inregistrari in PRODUSE
INSERT INTO produse
      (id_produș,nume_produș,cerinta_produș,valabilitate_zile)
VALUES      (sekv_produș.nextval,'Bilet Elev','Carnet Elev',1);

INSERT INTO produse
      (id_produș,nume_produș,cerinta_produș,valabilitate_zile)
VALUES      (sekv_produș.nextval,'Bilet Student','Legitimatie Student',1);

INSERT INTO produse
      (id_produș,nume_produș,cerinta_produș,valabilitate_zile)
VALUES      (sekv_produș.nextval,'Bilet',NULL,1);

INSERT INTO produse
      (id_produș,nume_produș,cerinta_produș,valabilitate_zile)
VALUES      (sekv_produș.nextval,'Bilet Elev Extins','Carnet Elev',7);

INSERT INTO produse
      (id_produș,nume_produș,cerinta_produș,valabilitate_zile)
VALUES      (sekv_produș.nextval,'Bilet Student Extins','Legitimatie Student',7);

INSERT INTO produse
      (id_produș,nume_produș,cerinta_produș,valabilitate_zile)
VALUES      (sekv_produș.nextval,'Bilet Extins',NULL,7);

INSERT INTO produse
      (id_produș,nume_produș,cerinta_produș,valabilitate_zile)
VALUES      (sekv_produș.nextval,'Bilet Pensionar','Carnet Pensionar',1);

INSERT INTO produse
      (id_produș,nume_produș,cerinta_produș,valabilitate_zile)
VALUES      (sekv_produș.nextval,'Bilet Pensionar Extins','Carnet Pensionar',7);

INSERT INTO produse
      (id_produș,nume_produș,cerinta_produș,valabilitate_zile)
VALUES      (sekv_produș.nextval,'Bilet Lunar',NULL,30);

INSERT INTO produse
      (id_produș,nume_produș,cerinta_produș,valabilitate_zile)
VALUES      (sekv_produș.nextval,'Popcorn',NULL,1);

INSERT INTO produse
      (id_produș,nume_produș,cerinta_produș,valabilitate_zile)
VALUES      (sekv_produș.nextval,'Popcorn cu caramel',NULL,1);

INSERT INTO produse
      (id_produș,nume_produș,cerinta_produș,valabilitate_zile)
VALUES      (sekv_produș.nextval,'Apa',NULL,1);

INSERT INTO produse
      (id_produș,nume_produș,cerinta_produș,valabilitate_zile)
VALUES      (sekv_produș.nextval,'Vata de zahar',NULL,1);

```



```

INSERT INTO produse
(id_produș, nume_produș, cerinta_produș, valabilitate_zile)
VALUES
(secv_produș.nextval, 'Inghetata', NULL, 1);

INSERT INTO produse
(id_produș, nume_produș, cerinta_produș, valabilitate_zile)
VALUES
(secv_produș.nextval, 'Popcorn cu ciocolata', NULL, 1);

INSERT INTO produse
(id_produș, nume_produș, cerinta_produș, valabilitate_zile)
VALUES
(secv_produș.nextval, 'Chipsuri', NULL, 1);

INSERT INTO produse
(id_produș, nume_produș, cerinta_produș, valabilitate_zile)
VALUES
(secv_produș.nextval, 'Slushie Coacaze', NULL, 1);

INSERT INTO produse
(id_produș, nume_produș, cerinta_produș, valabilitate_zile)
VALUES
(secv_produș.nextval, 'Slushie Capsuni', NULL, 1);

INSERT INTO produse
(id_produș, nume_produș, cerinta_produș, valabilitate_zile)
VALUES
(secv_produș.nextval, 'Slushie Portocale', NULL, 1);

INSERT INTO produse
(id_produș, nume_produș, cerinta_produș, valabilitate_zile)
VALUES
(secv_produș.nextval, 'Slushie Cirese', NULL, 1);

--inseram inregistrari in VAND
INSERT INTO vand
(id_zoo, id_produș, cost)
VALUES
(1, 1000, 11);

INSERT INTO vand
(id_zoo, id_produș, cost)
VALUES
(1, 1010, 18);

INSERT INTO vand
(id_zoo, id_produș, cost)
VALUES
(1, 1020, 4);

INSERT INTO vand
(id_zoo, id_produș, cost)
VALUES
(1, 1030, 29);

INSERT INTO vand
(id_zoo, id_produș, cost)
VALUES
(1, 1040, 11);

INSERT INTO vand
(id_zoo, id_produș, cost)
VALUES
(1, 1050, 28);

INSERT INTO vand
(id_zoo, id_produș, cost)
VALUES
(1, 1060, 17);

INSERT INTO vand
(id_zoo, id_produș, cost)
VALUES
(1, 1070, 27);

INSERT INTO vand
(id_zoo, id_produș, cost)
VALUES
(1, 1080, 30);

INSERT INTO vand

```

```

VALUES      (id_zoo,id_produș, cost)
VALUES      (1,1090,28);

INSERT INTO vand
VALUES      (id_zoo,id_produș, cost)
VALUES      (1,1100,24);

INSERT INTO vand
VALUES      (id_zoo,id_produș, cost)
VALUES      (1,1110,1);

INSERT INTO vand
VALUES      (id_zoo,id_produș, cost)
VALUES      (1,1120,27);

INSERT INTO vand
VALUES      (id_zoo,id_produș, cost)
VALUES      (1,1130,25);

INSERT INTO vand
VALUES      (id_zoo,id_produș, cost)
VALUES      (1,1140,18);

INSERT INTO vand
VALUES      (id_zoo,id_produș, cost)
VALUES      (1,1150,13);

INSERT INTO vand
VALUES      (id_zoo,id_produș, cost)
VALUES      (1,1160,29);

INSERT INTO vand
VALUES      (id_zoo,id_produș, cost)
VALUES      (1,1170,11);

INSERT INTO vand
VALUES      (id_zoo,id_produș, cost)
VALUES      (1,1180,19);

INSERT INTO vand
VALUES      (id_zoo,id_produș, cost)
VALUES      (1,1190,22);

--inșeram inregistrari in JOBS
INSERT INTO jobs
VALUES      (id_job,titlu_job,salariu_min,salariu_max)
VALUES      ('VANZ_PROD','Vanzator de produse',1000,5000);

INSERT INTO jobs
VALUES      (id_job,titlu_job,salariu_min,salariu_max)
VALUES      ('INGRJ','Ingrijitor animale',2500,6300);

INSERT INTO jobs
VALUES      (id_job,titlu_job,salariu_min,salariu_max)
VALUES      ('VTRNR','Veterinar',4000,10000);

INSERT INTO jobs
VALUES      (id_job,titlu_job,salariu_min,salariu_max)
VALUES      ('GHD','Ghid',3000,7000);

INSERT INTO jobs
VALUES      (id_job,titlu_job,salariu_min,salariu_max)
VALUES      ('CNTBL','Contabil',2000,5500);

INSERT INTO jobs

```

```

VALUES      (id_job,titlu_job,salariu_min,salariu_max)
VALUES      ('ADMN','Administrator',5000,12000);

INSERT INTO jobs
VALUES      (id_job,titlu_job,salariu_min,salariu_max)
VALUES      ('REL_EXT','Relatii externe',3300,7700);

INSERT INTO jobs
VALUES      (id_job,titlu_job,salariu_min,salariu_max)
VALUES      ('MCNC','Mecanic',3500,6500);

INSERT INTO jobs
VALUES      (id_job,titlu_job,salariu_min,salariu_max)
VALUES      ('DRSR','Dresor',2700,5900);

INSERT INTO jobs
VALUES      (id_job,titlu_job,salariu_min,salariu_max)
VALUES      ('SRVC','Om de serviciu',1800,4500);

--inseram inregistrari in ANGAJATI
INSERT INTO angajati
VALUES      (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email,data_ang)
VALUES      (
secv_angajat.nextval,'Lam','Brit',2700,1,'SRVC','0787513945','brit.lam@gmail.com','19-MAY-2017');

INSERT INTO angajati
VALUES      (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email,data_ang)
VALUES      (
secv_angajat.nextval,'Brown','Rhonda',2800,1,'SRVC','0781349745','rhonda.brown@gmail.com','29-
JUL-2016');

INSERT INTO angajati
VALUES      (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email)
VALUES      (
secv_angajat.nextval,'Kuroki','Patrocinia',2000,1,'SRVC',NULL,'patrocinia.kuroki@gmail.com');

INSERT INTO angajati
VALUES      (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email,data_ang)
VALUES      (
secv_angajat.nextval,'Jones','Valerie',10000,1,'ADMN','0744220556','valerie.jones@gmail.com','09-
APR-2013');

INSERT INTO angajati
VALUES      (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email,data_ang)
VALUES      (
secv_angajat.nextval,'Horvat','Achim',4700,1,'MCNC','0733404408','achim.horvat@gmail.com','12-
DEC-2014');

INSERT INTO angajati
VALUES      (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email,data_ang)
VALUES      (
secv_angajat.nextval,'Hakim','Cadmus',5500,1,'MCNC','0745239526','cadmus.hakim@gmail.com','11-
OCT-2014');

INSERT INTO angajati
VALUES      (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email,data_ang)
VALUES      (
secv_angajat.nextval,'Beck','Eliza',5400,1,'DRSR','0756554712','eliza.beck@gmail.com','22-SEP-
2016');

```

```

INSERT INTO angajati
        (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email,data_ang)
VALUES
(
    secv_angajat.nextval,'Rhodes','Vanesa',5900,1,'DRSR','0733659127','vanesa.rhodes@gmail.com','02-
AUG-2012');

INSERT INTO angajati
        (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email,data_ang)
VALUES
(
    secv_angajat.nextval,'Stanford','Tatiana',7500,1,'REL_EXT','0711957668','tatiana.stanford@gmail.c
om','27-JUN-2013');

INSERT INTO angajati
        (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email,data_ang)
VALUES
(
    secv_angajat.nextval,'Ma','Yusra',4000,1,'CNTBL','0771524366','yusra.ma@gmail.com','24-JAN-
2010');

INSERT INTO angajati
        (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email,data_ang)
VALUES
(
    secv_angajat.nextval,'Milan','Lucas',6000,1,'GHD','0786324912','lucas.milan@gmail.com','22-FEB-
2016');

INSERT INTO angajati
        (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email,data_ang)
VALUES
(
    secv_angajat.nextval,'Kramer','Anubis',5700,1,'GHD','0774552891','anubis.kramer@gmail.com','19-
NOV-2018');

INSERT INTO angajati
        (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email,data_ang)
VALUES
(
    secv_angajat.nextval,'Dwight','Emilia',8000,1,'VTRNR','0752272512','emilia.dwight@gmail.com','20-
OCT-2019');

INSERT INTO angajati
        (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email,data_ang)
VALUES
(
    secv_angajat.nextval,'Hildegard','Kapel',8900,1,'VTRNR','0744811814','kapel.hildegard@gmail.com',
'06-OCT-2017');

INSERT INTO angajati
        (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email,data_ang)
VALUES
(
    secv_angajat.nextval,'Seward','Malte',3500,1,'VANZ_PROD','0744659821','malte.seward@gmail.com','0
6-JAN-2017');

INSERT INTO angajati
        (id_ang,nume,prenume,salariu,id_zoo,id_job,nr_tel,email,data_ang)
VALUES
(
    secv_angajat.nextval,'Kiefer','Zarah',3000,1,'VANZ_PROD','0777153982','zarah.kiefer@gmail.com','1
3-MAY-2010');

--inseram inregistrari in ANIMALE
INSERT INTO animale

```

```

        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Shakeel',0,2,'TES_APA','Calm',11);

INSERT INTO animale
        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Ivy',4,2,'COA','Lenes',4);

INSERT INTO animale
        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Kaylee',2,2,'RAT','Agresiv',7);

INSERT INTO animale
        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Shayla',5,12,'TIG_ALB','Infricosat',8);

INSERT INTO animale
        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Archer',9,5,'SCO_IMP','Infricosat',11);

INSERT INTO animale
        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Alexandros',10,11,'ALP','Agitat',9);

INSERT INTO animale
        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Rhianna',8,2,'ACV_COD','Calm',1);

INSERT INTO animale
        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Emmanuel',11,13,'CAP_EU','Agitat',2);

INSERT INTO animale
        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Ashlea',27,7,'ACV_COD','Lenes',7);

INSERT INTO animale
        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Neave',14,10,'CAP_EU','Agresiv',9);

INSERT INTO animale
        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Yasmin',15,5,'BRSC_D','Lenes',7);

INSERT INTO animale
        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Orlando',17,4,'ALP','Agitat',9);

INSERT INTO animale
        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Lee',22,6,'RAT','Agresiv',8);

INSERT INTO animale
        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Tarik',9,7,'ALP','Prietenos',7);

INSERT INTO animale
        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Ziggy',9,3,'TES_APA','Agitat',7);

INSERT INTO animale
        (id_animal,nume,varsta,id_zoo,id_specie,temperament,varsta_luni)
VALUES      (secv_animale.nextval,'Kristy',12,5,'SCO_IMP','Vesel',7);

--inseram inregistrari in TIP_RESTRICTIE
INSERT INTO tip_restrictie
        (tip_restrictie,nume_restrictie,descriere_restrictie)

```

```

VALUES      ('REP_US', 'Repaus Usor', 'Repaus usor pentru reducerea oboselii');

INSERT INTO tip_restrictie
            (tip_restrictie, nume_restrictie, descriere_restrictie)
VALUES      ('REP_MED', 'Repaus Medical',
            'Repaus pentru imbunatatirea starii animalului');

INSERT INTO tip_restrictie
            (tip_restrictie, nume_restrictie, descriere_restrictie)
VALUES      ('INT_HR', 'Interzicere Hrana ',
            'Interzicere la hrana pentru imbunatatirea starii animalului');

INSERT INTO tip_restrictie
            (tip_restrictie, nume_restrictie, descriere_restrictie)
VALUES      ('EVI_FUR', 'Evitare Furaje', 'Evitarea hranirii cu furaje');

INSERT INTO tip_restrictie
            (tip_restrictie, nume_restrictie, descriere_restrictie)
VALUES      ('HR_ML', 'Hrana Moale', 'Hranirea se va face cu hrana moale');

INSERT INTO tip_restrictie
            (tip_restrictie, nume_restrictie, descriere_restrictie)
VALUES      ('IZO', 'Izolare', 'Izolare pentru a se recupera');

INSERT INTO tip_restrictie
            (tip_restrictie, nume_restrictie, descriere_restrictie)
VALUES      ('GUL', 'Guler Medical', 'Guler pentru a nu putea ajunge la rani');

INSERT INTO tip_restrictie
            (tip_restrictie, nume_restrictie, descriere_restrictie)
VALUES      ('EVI_RO', 'Evitare Rosie', 'Evitare carne rosie');

INSERT INTO tip_restrictie
            (tip_restrictie, nume_restrictie, descriere_restrictie)
VALUES      ('INT_CUP_FEM', 'Interzicere Cuplare',
            'Interzicere cuplare cu femele in adapost');

INSERT INTO tip_restrictie
            (tip_restrictie, nume_restrictie, descriere_restrictie)
VALUES      ('INT_CUP_MASC', 'Interzicere Cuplare',
            'Interzicere cuplare cu masculi in adapost');

INSERT INTO tip_restrictie
            (tip_restrictie, nume_restrictie, descriere_restrictie)
VALUES      ('INT_CUP_P', 'Interzicere Cuplare',
            'Interzicere cuplare cu pui in adapost');

--inseram inregistrari in RESTRICTIE
INSERT INTO restrictie
            (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '6-MAY-2021', '14-SEP-2023', 275, 'IZO');

INSERT INTO restrictie
            (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '3-FEB-2019', '27-JUL-2019', 308, 'IZO');

INSERT INTO restrictie
            (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '10-APR-2013', '21-JUL-2013', 354, 'INT_HR');

INSERT INTO restrictie
            (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '15-JUN-2008', '23-JUN-2011', 397, 'IZO');

INSERT INTO restrictie
            (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)

```

```

VALUES      (secv_restrictie.nextval, '14-NOV-2016', '20-DEC-2019', 213, 'HR_ML');

INSERT INTO restrictie
      (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '21-JAN-2001', '5-DEC-2001', 241, 'IZO');

INSERT INTO restrictie
      (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '22-MAR-2004', '12-NOV-2006', 305, 'INT_CUP_P');

INSERT INTO restrictie
      (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '28-OCT-2014', '22-DEC-2015', 342, 'HR_ML');

INSERT INTO restrictie
      (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '14-JAN-2019', '15-MAR-2022', 162, 'IZO');

INSERT INTO restrictie
      (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '4-FEB-2013', '21-SEP-2016', 345, 'INT_CUP_FEM');

INSERT INTO restrictie
      (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '26-JUL-2010', '28-JUL-2012', 235, 'HR_ML');

INSERT INTO restrictie
      (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '27-MAR-2005', '9-MAY-2007', 127, 'HR_ML');

INSERT INTO restrictie
      (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (
secv_restrictie.nextval, '20-OCT-2007', '21-OCT-2007', 165, 'INT_CUP_FEM');

INSERT INTO restrictie
      (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (
secv_restrictie.nextval, '16-FEB-2002', '13-NOV-2003', 380, 'INT_CUP_FEM');

INSERT INTO restrictie
      (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '13-DEC-2017', '16-DEC-2020', 375, 'HR_ML');

INSERT INTO restrictie
      (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '11-JAN-2016', '14-FEB-2016', 164, 'IZO');

INSERT INTO restrictie
      (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '27-APR-2012', '25-JUL-2014', 385, 'GUL');

INSERT INTO restrictie
      (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '27-JUN-2011', '18-NOV-2014', 141, 'GUL');

INSERT INTO restrictie
      (id_restrictie, data_inceput, data_sfarsit, id_animal, tip_restrictie)
VALUES      (secv_restrictie.nextval, '23-JUL-2011', '20-NOV-2013', 415, 'INT_CUP_P');

--inseram inregistrari in TIP_HRANA
INSERT INTO tip_hrana
      (id_tip_hrana, nume_hrana, origine, sapt_valabilitate)
VALUES      ('MRCV_FR', 'Morcovi', 'Franta', 1);

INSERT INTO tip_hrana

```

```

        (id_tip_hrana,nume_hrana,origine,sapt_valabilitate)
VALUES      ('MRCV_IT','Morcovi','Italia',2);

INSERT INTO tip_hrana
        (id_tip_hrana,nume_hrana,origine,sapt_valabilitate)
VALUES      ('FUR_GER','Furaje','Germania',5);

INSERT INTO tip_hrana
        (id_tip_hrana,nume_hrana,origine,sapt_valabilitate)
VALUES      ('SAL_AUS','SALATA','Austria',1);

INSERT INTO tip_hrana
        (id_tip_hrana,nume_hrana,origine,sapt_valabilitate)
VALUES      ('SEM_RO','Seminte','Romania',10);

INSERT INTO tip_hrana
        (id_tip_hrana,nume_hrana,origine,sapt_valabilitate)
VALUES      ('CRNV_CH','Carne Vita','China',1);

INSERT INTO tip_hrana
        (id_tip_hrana,nume_hrana,origine,sapt_valabilitate)
VALUES      ('CRNO_US','Carne Oaie','USA',2);

INSERT INTO tip_hrana
        (id_tip_hrana,nume_hrana,origine,sapt_valabilitate)
VALUES      ('INS_IN','Insecte','India',7);

INSERT INTO tip_hrana
        (id_tip_hrana,nume_hrana,origine,sapt_valabilitate)
VALUES      ('PST_BUG','Peste','Bulgaria',3);

INSERT INTO tip_hrana
        (id_tip_hrana,nume_hrana,origine,sapt_valabilitate)
VALUES      ('CER_SP','Cereale','Spania',15);

INSERT INTO tip_hrana
        (id_tip_hrana,nume_hrana,origine,sapt_valabilitate)
VALUES      ('POR_CAN','Porumb','Canada',16);

--inseram inregistrari in HRANA
INSERT INTO hrana
        (id_hrana,cantitate,id_zoo,id_tip_hrana)
VALUES      (secv_hrana.nextval,430,1,'CER_SP');

INSERT INTO hrana
        (id_hrana,cantitate,id_zoo,id_tip_hrana)
VALUES      (secv_hrana.nextval,890,1,'CRNO_US');

INSERT INTO hrana
        (id_hrana,cantitate,id_zoo,id_tip_hrana)
VALUES      (secv_hrana.nextval,20,1,'CRNV_CH');

INSERT INTO hrana
        (id_hrana,cantitate,id_zoo,id_tip_hrana)
VALUES      (secv_hrana.nextval,450,1,'FUR_GER');

INSERT INTO hrana
        (id_hrana,cantitate,id_zoo,id_tip_hrana)
VALUES      (secv_hrana.nextval,580,1,'INS_IN');

INSERT INTO hrana
        (id_hrana,cantitate,id_zoo,id_tip_hrana)
VALUES      (secv_hrana.nextval,440,1,'MRCV_IT');

INSERT INTO hrana
        (id_hrana,cantitate,id_zoo,id_tip_hrana)

```



```

VALUES      (secv_hrana.nextval,260,1,'POR_CAN');

INSERT INTO hrana
      (id_hrana,cantitate,id_zoo,id_tip_hrana)
VALUES      (secv_hrana.nextval,180,1,'PST_BUG');

INSERT INTO hrana
      (id_hrana,cantitate,id_zoo,id_tip_hrana)
VALUES      (secv_hrana.nextval,980,1,'SAL_AUS');

INSERT INTO hrana
      (id_hrana,cantitate,id_zoo,id_tip_hrana)
VALUES      (secv_hrana.nextval,810,1,'SEM_RO');

INSERT INTO hrana
      (id_hrana,cantitate,id_zoo,id_tip_hrana)
VALUES      (secv_hrana.nextval,110,2,'CER_SP');

INSERT INTO hrana
      (id_hrana,cantitate,id_zoo,id_tip_hrana)
VALUES      (secv_hrana.nextval,690,2,'CRNO_US');

INSERT INTO hrana
      (id_hrana,cantitate,id_zoo,id_tip_hrana)
VALUES      (secv_hrana.nextval,730,2,'CRNV_CH');

INSERT INTO hrana
      (id_hrana,cantitate,id_zoo,id_tip_hrana)
VALUES      (secv_hrana.nextval,1000,2,'FUR_GER');

--inseram inregistrari in HRANESTE
INSERT INTO hraneste
      (id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES      (148,517,707,'16-MAY-2022',To_date('16-MAY-2022 17:31',
      'DD-MON-YYYY HH24:MI'),1);

INSERT INTO hraneste
      (id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES      (148,517,707,'16-MAY-2022',To_date('16-MAY-2022 3:8',
      'DD-MON-YYYY HH24:MI'),5);

INSERT INTO hraneste
      (id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES      (148,517,707,'16-MAY-2022',To_date('16-MAY-2022 4:4',
      'DD-MON-YYYY HH24:MI'),2);

INSERT INTO hraneste
      (id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES      (153,518,701,'16-MAY-2022',To_date('16-MAY-2022 9:15',
      'DD-MON-YYYY HH24:MI'),2);

INSERT INTO hraneste
      (id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES      (174,518,704,'16-MAY-2022',To_date('16-MAY-2022 11:6',
      'DD-MON-YYYY HH24:MI'),5);

INSERT INTO hraneste
      (id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES      (192,518,704,'16-MAY-2022',To_date('16-MAY-2022 6:43',
      'DD-MON-YYYY HH24:MI'),4);

INSERT INTO hraneste
      (id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES      (192,500,704,'16-MAY-2022',To_date('16-MAY-2022 23:50',
      'DD-MON-YYYY HH24:MI'),3);

```

```

INSERT INTO hraneste
(id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES
(192,518,704,'16-MAY-2022',To_date('16-MAY-2022 5:36',
'DD-MON-YYYY HH24:MI'),2);

INSERT INTO hraneste
(id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES
(200,518,701,'16-MAY-2022',To_date('16-MAY-2022 4:33',
'DD-MON-YYYY HH24:MI'),4);

INSERT INTO hraneste
(id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES
(210,516,707,'16-MAY-2022',To_date('16-MAY-2022 13:20',
'DD-MON-YYYY HH24:MI'),5);

INSERT INTO hraneste
(id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES
(216,500,702,'16-MAY-2022',To_date('16-MAY-2022 7:23',
'DD-MON-YYYY HH24:MI'),5);

INSERT INTO hraneste
(id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES
(216,518,702,'16-MAY-2022',To_date('16-MAY-2022 0:35',
'DD-MON-YYYY HH24:MI'),2);

INSERT INTO hraneste
(id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES
(216,518,702,'16-MAY-2022',To_date('16-MAY-2022 22:7',
'DD-MON-YYYY HH24:MI'),1);

INSERT INTO hraneste
(id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES
(220,517,706,'16-MAY-2022',To_date('16-MAY-2022 15:47',
'DD-MON-YYYY HH24:MI'),4);

INSERT INTO hraneste
(id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES
(220,516,702,'16-MAY-2022',To_date('16-MAY-2022 5:15',
'DD-MON-YYYY HH24:MI'),3);

INSERT INTO hraneste
(id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES
(220,516,702,'16-MAY-2022',To_date('16-MAY-2022 3:14',
'DD-MON-YYYY HH24:MI'),2);

INSERT INTO hraneste
(id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES
(224,506,702,'16-MAY-2022',To_date('16-MAY-2022 10:41',
'DD-MON-YYYY HH24:MI'),5);

INSERT INTO hraneste
(id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
VALUES
(224,517,701,'16-MAY-2022',To_date('16-MAY-2022 12:30',
'DD-MON-YYYY HH24:MI'),4);

```

```
-- 5)Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare
-- entitate independentă; minim 10 înregistrări pentru tabela asociativă).

--secventa care va ajuta in inserarea inregistrarilor in GRADINA_ZOOLOGICA
create sequence secv_zoo
increment by 1
start with 1
maxvalue 9999
nocycle
nocache;

--secventa care va ajuta in inserarea inregistrarilor in ANIMALE
create sequence secv_animale
increment by 1
start with 100
maxvalue 9999
nocycle
nocache;

--secventa care va ajuta in inserarea inregistrarilor in PARTENER
create sequence secv_part
increment by 10
start with 10
maxvalue 9999
nocycle
nocache;

--secventa care va ajuta in inserarea inregistrarilor in SERVICIU
create sequence secv_serv
increment by 5
```



Sequence SECV\_ZOO created.

Sequence SECV\_ANIMALE created.

Sequence SECV\_PART created.

Sequence SECV\_SERV created.

Sequence SECV\_PRODUS created.

```

insert into PARTENER(id_part,nume_partener,nume_contact,email_contact,nr_tel_contact,oras_sediu)
values(secv_part.nextval,'ROWNE','Singer','robert.singer@gmail.com',null,'Rotterdam');

insert into PARTENER(id_part,nume_partener,nume_contact,email_contact,nr_tel_contact,oras_sediu)
values(secv_part.nextval,'BEGAS','Marconi','julie.marconi@gmail.com',null,'Paris');

insert into PARTENER(id_part,nume_partener,nume_contact,email_contact,nr_tel_contact,oras_sediu)
values(secv_part.nextval,'FARKIN','Roosevelt','tom.roosevelt@gmail.com','0733691024','Milan');

insert into PARTENER(id_part,nume_partener,nume_contact,email_contact,nr_tel_contact,oras_sediu)
values(secv_part.nextval,'REPPER','Kleinz','mauro.kleinz@gmail.com','0763500792','Frankfurt');

insert into PARTENER(id_part,nume_partener,nume_contact,email_contact,nr_tel_contact,oras_sediu)
values(secv_part.nextval,'SCHLET','Pope','rebecca.pope@gmail.com',null,null);

--inseram inregistrari in SERVICIU
insert into SERVICIU(id_serviciu,nume_serviciu,tip_serviciu)
values (secv_serv.nextval,'Hartie','Achizitionare');

insert into SERVICIU(id_serviciu,nume_serviciu,tip_serviciu)
values (secv_serv.nextval,'Bani','Donatie');

insert into SERVICIU(id_serviciu,nume_serviciu,tip_serviciu)
values (secv_serv.nextval,'Spectacol','Vanzare');

insert into SERVICIU(id_serviciu,nume_serviciu,tip_serviciu)
values (secv_serv.nextval,'Morcovi','Achizitionare');

```

Script Output x Query Result x

Task completed in 3.081 seconds

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

--entitatea PARTENER

create table PARTENER(

id\_part number(4) constraint pk\_part primary key,

nume\_partener varchar2(30) constraint null\_nume\_partener not null,

nume\_contact varchar2(25) constraint null\_nume\_contact not null,

email\_contact varchar2(30) constraint null\_email not null constraint unq\_mail unique,

nr\_tel\_contact varchar2(15) constraint unq\_tel unique,

oras\_sediu varchar2(25)

);

Script Output x Query Result x

SQL | All Rows Fetched: 12 in 0.004 seconds

ID_PART	NUME_PARTENER	NUME_CONTACT	EMAIL_CONTACT	NR_TEL_CONTACT	ORAS_SEDIU
1	10 SUFIPAPER	Smith	jason.smith@gmail.com	0772561483	Sydney
2	20 MEROS	Jameson	ashley.jameson@gmail.com	0751483301	New York
3	30 KINGUS	Austin	bruce.austin@gmail.com	(null)	Tokyo
4	40 CRONEKER	Mikks	brad.mikks@gmail.com	0772561553	(null)
5	50 GRAEL	Williams	judy.williams@gmail.com	0760161273	San Francisco
6	60 TANG	Brown	rene.brown@gmail.com	0762351104	Mumbai
7	70 BLICH	Walker	jason.walker@gmail.com	0789210345	(null)
8	80 ROWNE	Singer	robert.singer@gmail.com	(null)	Rotterdam
9	90 BEGAS	Marconi	julie.marconi@gmail.com	(null)	Paris
10	100 FARKIN	Roosevelt	tom.roosevelt@gmail.com	0733691024	Milan
11	110 REPHER	Kleinz	mauro.kleinz@gmail.com	0763500792	Frankfurt
12	120 SCHLET	Pope	rebecca.pope@gmail.com	(null)	(null)

--entitatea SERVICIU

create table SERVICIU(

id\_serviciu number(4) constraint pk\_serv primary key,

nume\_serviciu varchar2(60) constraint null\_serv not null,

tip\_serviciu varchar2(15) constraint null\_tip not null constraint check\_tip check(lower(tip\_serviciu) in ('donatie','achizitionare','vanzare'))

);

Script Output x Query Result x

SQL | All Rows Fetched: 17 in 0.006 seconds

ID_SERVICIU	NUME_SERVICIU	TIP_SERVICIU
1	200 Hartie	Achizitionare
2	205 Bani	Donatie
3	210 Spectacol	Vanzare
4	215 Morcovi	Achizitionare
5	220 Suplimente	Achizitionare
6	225 Medicamente	Achizitionare
7	230 Medicamente	Donatie
8	235 Furaje	Achizitionare
9	240 Spectacol Acvatic	Vanzare
10	245 Eveniment Privat	Vanzare
11	250 Seminte	Achizitionare
12	255 Carne de vita	Achizitionare
13	260 Carne de oaie	Achizitionare
14	265 Salata	Achizitionare
15	270 Vitamine	Achizitionare
16	275 Vitamine	Donatie
17	280 Potcoave	Donatie

```
--entitatea GRADINA_ZOOLOGICA
create table GRADINA_ZOOLOGICA(
    id_zoo number(4) constraint pk_zoo primary key,
    locatie varchar2(30) constraint null_locatie not null,
    dimensiune number(4)
);
```

ID_ZOO	LOCATIE	DIMENSIUNE
1	1 Sydney	200
2	2 Mumbai	200
3	3 New York	200
4	4 Tokyo	300
5	5 San Francisco	350
6	6 Singapore	360
7	7 Kansas	195
8	8 Bordeaux	400
9	9 Bern	360
10	10 Jakarta	400
11	11 Detroit	250
12	12 Marseille	270
13	13 Ankara	320
14	14 Boston	350

```
--entitatea BENEFICIAZA
create table BENEFICIAZA(
    id_part number(4) constraint fk_part references PARTENER(id_part) on delete cascade,
    id_zoo number(4) constraint fk_zoo references GRADINA_ZOOLOGICA(id_zoo) on delete cascade,
    id_serviciu number(4) constraint fk_serv references SERVICIU(id_serviciu) on delete cascade,
    data_serviciu date default sysdate,
    cantitate varchar2(30) constraint null_cantitate not null,
    constraint pk_part_zoo_serv_data primary key(id_part,id_zoo,id_serviciu,data_serviciu)
);
```

ID_PART	ID_ZOO	ID_SERVICIU	DATA_SERVICIU	CANTITATE
1	20	6	205 11-MAY-08	1000 euro
2	20	7	205 11-MAY-08	5000 euro
3	20	10	205 11-MAY-08	3500 euro
4	100	13	225 23-JUN-15	10 flacoane
5	30	4	255 18-NOV-02	100 kilograme
6	30	9	255 20-DEC-12	50 kilograme
7	30	8	255 18-JAN-06	150 kilograme
8	110	1	250 11-AUG-02	30 kilograme
9	110	5	250 11-JUL-02	20 kilograme
10	110	14	250 20-AUG-06	35 kilograme
11	70	3	245 21-APR-13	1 eveniment
12	70	2	265 14-SEP-11	30 kilograme
13	90	10	270 17-OCT-01	20 flacoane
14	10	10	200 17-MAR-21	3500 bucati
15	10	7	200 18-MAR-21	700 bucati
16	10	3	200 22-MAR-10	6000 bucati
17	10	12	200 02-MAR-21	1600 bucati
18	10	14	200 23-MAR-21	4250 bucati

```
--entitatea PRODUSE
create table PRODUSE(
    id_produș number(4) constraint pk_produș primary key,
    nume_produș varchar2(30) constraint null_nume not null,
    cerinta_produș varchar2(30),
    valabilitate_zile number(4) constraint null_valabilitate not null constraint check_valabilitate check(valabilitate_zile>0)
);
```

Script Output x Query Result x

SQL | All Rows Fetched: 20 in 0.006 seconds

ID_PRODUS	NUME_PRODUS	CERINTA_PRODUS	VALABILITATE_ZILE
1	1000 Bilet Elev	Carnet Elev	1
2	1010 Bilet Student	Legitimatie Student	1
3	1020 Bilet	(null)	1
4	1030 Bilet Elev Extins	Carnet Elev	7
5	1040 Bilet Student Extins	Legitimatie Student	7
6	1050 Bilet Extins	(null)	7
7	1060 Bilet Pensionar	Carnet Pensionar	1
8	1070 Bilet Pensionar Extins	Carnet Pensionar	7
9	1080 Bilet Lunar	(null)	30
10	1090 Popcorn	(null)	1
11	1100 Popcorn cu caramel	(null)	1
12	1110 Apa	(null)	1
13	1120 Vata de zahar	(null)	1
14	1130 Inghetata	(null)	1
15	1140 Popcorn cu ciocolata	(null)	1
16	1150 Chipsuri	(null)	1
17	1160 Slushie Coacaze	(null)	1
18	1170 Slushie Capsuni	(null)	1

```
--entitatea VAND
create table VAND(
    id_zoo number(4) constraint fk_zoo_vand references GRADINA_ZOOLOGICA(id_zoo) on delete cascade,
    id_produș number(4) constraint fk_produș references PRODUSE(id_produș) on delete cascade,
    cost number(4) constraint null_cost not null constraint check_cost check(cost>0),
    constraint pk_zoo_produș primary key(id_zoo,id_produș)
);
```

Script Output x Query Result x

SQL | Fetched 50 rows in 0.005 seconds

ID_ZOO	ID_PRODUS	COST
1	7	1000
2	7	1010
3	7	1020
4	7	1030
5	7	1040
6	7	1050
7	7	1060
8	7	1070
9	7	1080
10	7	1090
11	7	1100
12	7	1110
13	7	1120
14	7	1130
15	7	1140
16	7	1150
17	7	1160
18	7	1170

--entitatea JOBS

```
create table JOBS(  
    id_job varchar2(20) constraint pk_job primary key,  
    titlu_job varchar2(35) constraint null_titlu not null,  
    salariu_min number(8) constraint null_sal_min not null constraint check_sal_min check(salariu_min>0),  
    salariu_max number(8) constraint null_sal_max not null,  
    constraint check_sal_max check(salariu_max>salariu_min)  
);
```

Script Output x Query Result x

SQL | All Rows Fetched: 10 in 0.007 seconds

ID_JOB	TITLU_JOB	SALARIU_MIN	SALARIU_MAX
1	VANZ_PROD Vanzator de produse	1000	5000
2	INGRJ Ingrijitor animale	2500	6300
3	VTRNR Veterinar	4000	10000
4	GHD Ghid	3000	7000
5	CNTBL Contabil	2000	5500
6	ADMN Administrator	5000	12000
7	REL_EXT Relatii externe	3300	7700
8	MCNC Mecanic	3500	6500
9	DRSR Dresor	2700	5900
10	SRVC Om de serviciu	1800	4500

--entitatea ANGAJATI

```
create table ANGAJATI(  
    id_ang number(4) constraint pk_angajati primary key,  
    nume varchar2(30) constraint null_nume_ang not null,  
    prenume varchar2(30),  
    salariu number(8) constraint null_salariu_ang not null constraint check_salariu_ang check(salariu>0),  
    id_zoo number(4) constraint null_zoo_ang not null constraint fk_zoo_ang references GRADINA_ZOOLOGICA(id_zoo) on delete cascade,  
    id_job varchar2(20) constraint null_job_ang not null constraint fk_job_ang references JOBS(id_job) on delete cascade,  
    nr_tel varchar2(15) constraint unq_tel_ang unique,  
    email varchar2(30) constraint unq_email_ang unique constraint null_email_ang not null,  
    data_ang date default sysdate  
);
```

Script Output x Query Result x

SQL | Fetched 50 rows in 0.009 seconds

ID_ANG	NUME	PRENUME	SALARIU	ID_ZOO	ID_JOB	NR_TEL	EMAIL	DATA_ANG
1	500 Lam	Brit	2700	1	SRVC	0787513945	brit.lam@gmail.com	19-MAY-17
2	501 Brown	Rhonda	2800	1	SRVC	0781349745	rhonda.brown@gmail.com	29-JUL-16
3	502 Kuroki	Patrocinia	2000	1	SRVC	(null)	patrocinia.kuroki@gmail.com	15-MAY-22
4	503 Jones	Valerie	10000	1	ADMN	0744220556	valerie.jones@gmail.com	09-APR-13
5	504 Horvat	Achim	4700	1	MCNC	0733404408	achim.horvat@gmail.com	12-DEC-14
6	505 Hakim	Cadmus	5500	1	MCNC	0745239526	cadmus.hakim@gmail.com	11-OCT-14
7	506 Beck	Eliza	5400	1	DRSR	0756554712	eliza.beck@gmail.com	22-SEP-16
8	507 Rhodes	Vanesa	5900	1	DRSR	0733659127	vanesa.rhodes@gmail.com	02-AUG-12
9	508 Stanford	Tatiana	7500	1	REL_EXT	0711957668	tatiana.stanford@gmail.com	27-JUN-13
10	509 Ma	Yusra	4000	1	CNTBL	0771524366	yusra.ma@gmail.com	24-JAN-10
11	510 Milan	Lucas	6000	1	GHD	0786324912	lucas.milan@gmail.com	22-FEB-16
12	511 Kramer	Anubis	5700	1	GHD	0774552891	anubis.kramer@gmail.com	19-NOV-18
13	512 Dwight	Emilia	8000	1	VTRNR	0752272512	emilia.dwight@gmail.com	20-OCT-19
14	513 Hildegard	Kapel	8900	1	VTRNR	0744811814	kapel.hildegard@gmail.com	06-OCT-17
15	514 Seward	Malte	3500	1	VANZ_PROD	0744659821	malte.seward@gmail.com	06-JAN-17
16	515 Kiefer	Zarah	3000	1	VANZ_PROD	0777153982	zarah.kiefer@gmail.com	13-MAY-10
17	516 White	Goda	4000	1	INGRJ	0744194562	goda.white@gmail.com	01-JUN-11
18	517 Nq	Fenne	5000	1	INGRJ	0726718448	fenne.nq@gmail.com	23-MAY-17



```
--entitatea SPECIE
create table SPECIE(
    id_specie varchar2(25) constraint pk_specie primary key,
    tip_dieta varchar2(10) constraint null_dieta not null constraint check_dieta check(lower(tip_dieta) in ('carnivor','erbivor','omnivor')),
    colorit varchar2(60),
    tip_animal varchar2(20) constraint null_tip_sp not null constraint check_tip_sp check(lower(tip_animal) in ('nocturn','diurn','crepuscular')),
    varsta_medie number(4) constraint null_varsta not null,
    clasificare varchar2(20) constraint null_clasificare not null,
    nume_specie varchar2(35) constraint null_nume_sp not null
);
```

Script Output x Query Result x

SQL | All Rows Fetched: 12 in 0.007 seconds

ID_SPECIE	TIP_DIETA	COLORIT	TIP_ANIMAL	VARSTA_MEDIE	CLASIFICARE	NUME_SPECIE
1 TIG_ALB	Carnivor	Alb cu dungi negre	Crepuscular	11	Mamifer	Tigru Bengalez Alb
2 ACV_COD	Carnivor	Maro cu gat alb	Diurn	42	Pasare	Acvila Codalb
3 ALP	Erbivor	Maro inchis	Diurn	20	Mamifer	Alpaca
4 ANA_V	Carnivor	Verde	Nocturn	30	Reptila	Anaconda Verde
5 BRSC_D	Carnivor	Galben cu negru	Diurn	15	Amfiban	Broasca Dungata Galbena
6 URS_BR	Omnivor	Maro	Crepuscular	35	Mamifer	Urs Brun
7 CAP_EU	Erbivor	Maro deschis	Diurn	15	Mamifer	Caprior European
8 TES_APA	Carnivor	Carapace neagra	Diurn	26	Reptila	Testoasa de Apa
9 COA	Omnivor	Portocaliu inchis	Diurn	17	Mamifer	Coati
10 LEU	Carnivor	Portocaliu deschis	Nocturn	30	Mamifer	Leu African
11 RAT	Carnivor	Gri	Nocturn	20	Mamifer	Raton
12 SCO_IMP	Carnivor	Negru	Nocturn	8	Arahnida	Scorpion Imperial

```
--entitatea ANIMALE
create table ANIMALE(
    id_animal number(4) constraint pk_animal primary key,
    nume varchar2(25) default 'No Name',
    varsta number(4) constraint null_varsta_anim not null constraint check_varsta check(varsta>=0),
    id_zoo number(4) constraint null_zoo_anim not null constraint fk_zoo_anim references GRADINA_ZOOLOGICA(id_zoo) on delete cascade,
    id_specie varchar2(25) constraint null_specie not null constraint fk_specie references SPECIE(id_specie) on delete cascade,
    temperament varchar2(60),
    varsta_luni number(4) constraint null_varsta_luni not null constraint check_varsta_luni check(varsta_luni>=0 and varsta_luni<12)
);
```

Script Output x Query Result x

SQL | Fetched 50 rows in 0.007 seconds

ID_ANIMAL	NUME	VARSTA	ID_ZOO	ID_SPECIE	TEMPERAMENT	VARSTA_LUNI
1	100 Shakeel	0	2	TES_APA	Calm	11
2	101 Ivy	4	2	COA	Lenes	4
3	102 Kaylee	2	2	RAT	Agresiv	7
4	103 Shayla	5	12	TIG_ALB	Infricosat	8
5	104 Archer	9	5	SCO_IMP	Infricosat	11
6	105 Alexandros	10	11	ALP	Agitat	9
7	106 Rhianna	8	2	ACV_COD	Calm	1
8	107 Emmanuel	11	13	CAP_EU	Agitat	2
9	108 Ashlea	27	7	ACV_COD	Lenes	7
10	109 Neave	14	10	CAP_EU	Agresiv	9
11	110 Yasmin	15	5	BRSC_D	Lenes	7
12	111 Orlando	17	4	ALP	Agitat	9
13	112 Lee	22	6	RAT	Agresiv	8
14	113 Tarik	9	7	ALP	Prietenos	7
15	114 Ziggy	9	3	TES_APA	Agitat	7
16	115 Kristy	12	5	SCO_IMP	Vesel	7
17	116 Szymon	11	13	CAP_EU	Infricosat	4
18	117 Sameeha	3	9	BRSC_D	Infricosat	4

```
--entitatea TIP_RESTRICTIE
create table TIP_RESTRICTIE(
    tip_restriction varchar2(20) constraint pk_tip_restriction primary key,
    nume_restriction varchar2(30) constraint null_nume_restriction not null,
    descriere_restriction varchar2(80) constraint null_descriere not null
);
```

Script Output x Query Result x

SQL | All Rows Fetched: 10 in 0.006 seconds

TIP_RESTRICTIE	NUME_RESTRICTIE	DESCRIERE_RESTRICTIE
1 REP_MED	Repaus Medical	Repaus pentru imbunatatirea starii animalului
2 INT_HR	Interzicere Hrana	Interzicere la hrana pentru imbunatatirea starii animalului
3 EVI_FUR	Evitare Furaje	Evitarea hranirii cu furaje
4 HR_ML	Hrana Moale	Hranirea se va face cu hrana moale
5 IZO	Izolare	Izolare pentru a se recupera
6 GUL	Guler Medical	Guler pentru a nu putea ajunge la rani
7 EVI_RO	Evitare Rosie	Evitare carne rosie
8 INT_CUP_FEM	Interzicere Cuplare	Interzicere cuplare cu femele in adapost
9 INT_CUP_MASC	Interzicere Cuplare	Interzicere cuplare cu masculi in adapost
10 INT_CUP_P	Interzicere Cuplare	Interzicere cuplare cu pui in adapost

```
--entitatea RESTRICTIE
create table RESTRICTIE(
    id_restriction number(4) constraint pk_restriction primary key,
    data_inceput date default sysdate constraint null_data_inc not null,
    data_sfarsit date default sysdate constraint null_data_sf not null,
    id_animal number(4) constraint null_animal not null constraint fk_animal_rest references ANIMALE(id_animal) on delete cascade,
    tip_restriction varchar2(20) constraint null_tip_rest not null constraint fk_tip_rest references TIP_RESTRICTIE(tip_restriction) on delete cascade,
    constraint check_data_sf check(data_sfarsit>=data_inceput)
);
```

Script Output x Query Result x

SQL | Fetched 50 rows in 0.007 seconds

ID_RESTRICTIE	DATA_INCEPUT	DATA_SFARSIT	ID_ANIMAL	TIP_RESTRICTIE
1	300 03-FEB-19	27-JUL-19	308 IZO	
2	301 10-APR-13	21-JUL-13	354 INT_HR	
3	302 15-JUN-08	23-JUN-11	397 IZO	
4	303 14-NOV-16	20-DEC-19	213 HR_ML	
5	304 21-JAN-01	05-DEC-01	241 IZO	
6	305 22-MAR-04	12-NOV-06	305 INT_CUP_P	
7	306 28-OCT-14	22-DEC-15	342 HR_ML	
8	307 14-JAN-19	15-MAR-22	162 IZO	
9	308 04-FEB-13	21-SEP-16	345 INT_CUP_FEM	
10	309 26-JUL-10	28-JUL-12	235 HR_ML	
11	310 27-MAR-05	09-MAY-07	127 HR_ML	
12	311 20-OCT-07	21-OCT-07	165 INT_CUP_FEM	
13	312 16-FEB-02	13-NOV-03	380 INT_CUP_FEM	
14	313 13-DEC-17	16-DEC-20	375 HR_ML	
15	314 11-JAN-16	14-FEB-16	164 IZO	
16	315 27-APR-12	25-JUL-14	385 GUL	
17	316 27-JUN-11	18-NOV-14	141 GUL	
18	317 23-JUL-11	20-NOV-13	415 INT_CUP_P	

```
--entitatea TIP_HRANA
create table TIP_HRANA(
    id_tip_hrana varchar2(20) constraint pk_tip_hrana primary key,
    nume_hrana varchar2(30) constraint null_nume_tip not null,
    origine varchar2(30) constraint null_origine_tip not null,
    sapt_valabilitate number(4) constraint null_valabilitate_tip not null constraint check_valabilitate_tip check(sapt_valabilitate>0)
);
```

Script Output x Query Result x

SQL | All Rows Fetched: 10 in 0.007 seconds

ID_TIP_HRANA	NUME_HRANA	ORIGINE	SAPT_VALABILITATE
1 MRCV_IT	Morcovi	Italia	2
2 FUR_GER	Furaje	Germania	5
3 SAL_AUS	Salata	Austria	1
4 SEM_RO	Seminte	Romania	10
5 CRNV_CH	Carne Vita	China	1
6 CRNO_US	Carne Oaie	USA	2
7 INS_IN	Insecte	India	7
8 PST_BUG	Peste	Bulgaria	3
9 CER_SP	Cereale	Spania	15
10 POR_CAN	Porumb	Canada	16

```
--entitatea HRANA
create table HRANA(
    id_hrana number(4) constraint pk_hrana primary key,
    cantitate number(4) constraint null_cantitate_hrana not null constraint check_cantitate check(cantitate>0),
    id_zoo number(4) constraint null_zoo_hrana not null constraint fk_zoo_hrana references GRADINA_ZOOLOGICA(id_zoo) on delete cascade,
    id_tip_hrana varchar2(20) constraint null_tip_hrana not null constraint fk_tip_hrana references TIP_HRANA(id_tip_hrana) on delete cascade
);
```

Script Output x Query Result x

SQL | Fetched 50 rows in 0.007 seconds

ID_HRANA	CANTITATE	ID_ZOO	ID_TIP_HRANA
1	700	430	1 CER_SP
2	701	890	1 CRNO_US
3	702	20	1 CRNV_CH
4	703	450	1 FUR_GER
5	704	580	1 INS_IN
6	705	440	1 MRCV_IT
7	706	260	1 POR_CAN
8	707	180	1 PST_BUG
9	708	980	1 SAL_AUS
10	709	810	1 SEM_RO
11	710	110	2 CER_SP
12	711	690	2 CRNO_US
13	712	730	2 CRNV_CH
14	713	1000	2 FUR_GER
15	714	560	2 INS_IN
16	715	950	2 MRCV_IT
17	716	820	2 POR_CAN
18	717	110	2 PST_BUG

```

--entitatea HRANESTE
create table HRANESTE(
    id_animal number(4) constraint fk_animal_hraneste references ANIMALE(id_animal) on delete cascade,
    id_ang number(4) constraint fk_ang_hraneste references ANGAJATI(id_ang) on delete cascade,
    id_hrana number(4) constraint fk_hrana_hraneste references HRANA(id_hrana) on delete cascade,
    data_hranire date,
    ora_hranire date,
    cantitate number(4) constraint null_cantitate_hraneste not null constraint check_cantitate_hraneste check(cantitate>0),
    constraint pk_animal_ang_hrana_data_ora primary key(id_animal,id_ang,id_hrana,data_hranire,ora_hranire)
);

```

Script Output x Query Result x

SQL | Fetched 50 rows in 0.003 seconds

ID_ANIMAL	ID_ANG	ID_HRANA	DATA_HRANIRE	ORA_HRANIRE	CANTITATE
1	430	726	814 16-MAY-22	16-MAY-22	2
2	435	797	817 16-MAY-22	16-MAY-22	3
3	435	724	817 16-MAY-22	16-MAY-22	1
4	107	798	826 16-MAY-22	16-MAY-22	1
5	107	745	823 16-MAY-22	16-MAY-22	1
6	107	781	826 16-MAY-22	16-MAY-22	2
7	116	736	825 16-MAY-22	16-MAY-22	4
8	125	744	827 16-MAY-22	16-MAY-22	3
9	125	745	821 16-MAY-22	16-MAY-22	1
10	128	744	820 16-MAY-22	16-MAY-22	1
11	128	743	828 16-MAY-22	16-MAY-22	1
12	149	781	821 16-MAY-22	16-MAY-22	2
13	149	745	821 16-MAY-22	16-MAY-22	5
14	149	781	822 16-MAY-22	16-MAY-22	2
15	157	745	822 16-MAY-22	16-MAY-22	4
16	157	781	821 16-MAY-22	16-MAY-22	1
17	157	743	827 16-MAY-22	16-MAY-22	3
18	170	798	825 16-MAY-22	16-MAY-22	4

**6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze două tipuri diferite de colecții studiate. Apelați subprogramul.**

*Enunt:*

Specificandu-se un partener (id\_part) afisati pentru fiecare gradina zoologica cu care acest partener a interactionat, lista angajatilor (id, nume, prenume, ocupatie) care au hranit animale, dar nu sunt ingrijitori.

*Explicatie:*

Pentru rezolvarea problemei am utilizat un record care contine date despre angajat, si ca tipuri de colectii un tablou imbricat de record-uri (lista de angajati), un tablou indexat cu pls\_integer care retine listele de angajati.

Practic logica este : tablou(id\_zoo) = lista\_de\_angajati, unde id\_zoo este un numar intreg, iar lista\_de\_angajati corespunde angajatilor din gradina zoologica respective.

Am utilizat si un tablou indexat cu varchar2 ce tine elemente tot de varchar2, pentru a converti de la id\_job -ul angajatului la titlu\_job corespunzator id\_job -ului. (ex. tablou('INGRJ') = 'Ingrijitor').

Am utilizat 3 cursoare, unul care sa ma ajute la construirea tabloului de job, unul pentru initializarea tablourilor imbricate pentru gradinile zoologice cu care partenerul interactioneaza si unul care imi returneaza toti angajatii care se potrivesc cerintei.

Astfel, se va afisa, pentru fiecare gradina zoo cu care partenerul respectiv interactioneaza, lista de angajati sau un mesaj corespunzator daca nu exista angajati care se potrivesc cerintei. De asemenea, au fost tratate exceptiile cand partenerul nu exista, sau cand partenerul nu interactioneaza cu nici o gradina zoologica.

### *Codul:*

```
CREATE OR replace PROCEDURE Angajati_hranesc(id_partener IN
partener.id_part%TYPE)
IS
    TYPE ang IS RECORD (
        id_ang angajati.id_ang%TYPE,
        nume angajati.nume%TYPE,
        prenume angajati.prenume%TYPE,
        titlu_job jobs.titlu_job%TYPE );
    TYPE lista_ang
        IS TABLE OF ANG;
    TYPE gradzoo
        IS TABLE OF LISTA_ANG INDEX BY PLS_INTEGER;
    TYPE nume_job
        IS TABLE OF jobs.titlu_job%TYPE INDEX BY VARCHAR2(10);
    no_partner_found EXCEPTION;
    no_zoo_found EXCEPTION;
    verificare          NUMBER(3);
    angajat_job         NUME_JOB;
    CURSOR joburi IS
        SELECT id_job,
               titlu_job
        FROM   jobs;
    CURSOR gradini_zoo(
        id_p partener.id_part%TYPE) IS
        SELECT DISTINCT id_zoo
        FROM   beneficiaza
        WHERE  id_part = id_p;
```

```

CURSOR cursor_angajati(
    id_p partener.id_part%TYPE) IS
    SELECT DISTINCT a.id_ang,
                   a.nume,
                   a.prenume,
                   a.id_job,
                   a.id_zoo

    FROM    angajati a,
           hraneste h
    WHERE   a.id_zoo IN (SELECT DISTINCT id_zoo
                        FROM    beneficiaza
                        WHERE   id_part = id_p)
           AND h.id_ang = a.id_ang
           AND a.id_job != 'INGRJ';
lista_angajati GRADZOO;
aux            ANG;
i              PLS_INTEGER;
nr_ordine      NATURAL;
BEGIN
    SELECT Count(*)
    INTO  verificare
    FROM  partener
    WHERE id_part = id_partener;

    IF verificare = 0 THEN
        RAISE no_partner_found;
    END IF;

    SELECT Count(*)
    INTO  verificare
    FROM  beneficiaza
    WHERE id_part = id_partener;

    IF verificare = 0 THEN
        RAISE no_zoo_found;
    END IF;

    FOR elem IN joburi LOOP
        Angajat_job(elem.id_job) := elem.titlu_job;
    END LOOP;

    FOR elem IN gradini_zoo(id_partener) LOOP
        Lista_angajati(elem.id_zoo) := Lista_ang();
    END LOOP;

    FOR elem IN cursor_angajati(id_partener) LOOP
        aux.id_ang := elem.id_ang;

        aux.nume := elem.nume;

        aux.prenume := elem.prenume;

        aux.titlu_job := Angajat_job(elem.id_job);

        Lista_angajati(elem.id_zoo).extend;

        lista_angajati(elem.id_zoo)(Lista_angajati(elem.id_zoo).last) := aux;
    END LOOP;

```

```

dbms_output.Put_line('Gradinile zoologice cu care a interactionat partenerul cu id '
                    || id_partener
                    || ' sunt:');

dbms_output.new_line;

dbms_output.new_line;

i := lista_angajati.first;

WHILE i <= lista_angajati.last LOOP
    dbms_output.Put_line('-----');

    dbms_output.Put_line('Gradina zoologica '
                        || i);

    dbms_output.Put_line('-----');

    IF Lista_angajati(i).count = 0 THEN
dbms_output.Put_line('Nu exista angajati care au hranit animalele si sa nu fie ingriji
tori.');
```

```

ELSE
    nr_ordine := 1;

    FOR j IN Lista_angajati(i).first..Lista_angajati(i).last LOOP
        dbms_output.Put_line(nr_ordine
                            || ') '
                            || lista_angajati(i)(j).id_ang
                            || ' '
                            || lista_angajati(i)(j).prenume
                            || ' '
                            || lista_angajati(i)(j).nume
                            || ' '
                            || lista_angajati(i)(j).titlu_job);

        nr_ordine := nr_ordine + 1;
    END LOOP;
END IF;

dbms_output.new_line;

dbms_output.new_line;

i := lista_angajati.NEXT(i);
END LOOP;
EXCEPTION
    WHEN no_partner_found THEN
        Raise_application_error(-20001,
            'Nu exista un partener cu acest id.')
```

```

; WHEN no_zoo_found THEN
        Raise_application_error(-20002,
            'Acest partener nu interactioneaza cu nicio gradina zoologica.');
```

```

END;

/
DECLARE
    id_partener partener.id_part%TYPE := &v_partener;
```

```
BEGIN
    Angajati_hranesc(id_partener);
END;
```

/

Pentru id\_partener = 130

```
exception
    when no_partner_found then
        raise_application_error(-20001, 'Nu exista un partener cu acest id.');
```

```
    when no_zoo_found then
        raise_application_error(-20002, 'Acest partener nu interactioneaza cu nicio gradina zoologica.');
```

```
end;
```

```
/
```

```
declare
    id_partener partener.id_part&type := &v_partener;
```

```
begin
```

```
    angajati_hranesc(id_partener);
```

```
end;
```

```
/
```

Script Output x Query Result x

Task completed in 3.853 seconds

Procedure ANGAJATI\_HRANESC compiled

Error starting at line : 7,047 in command -

```
declare
    id_partener partener.id_part&type := &v_partener;
```

```
begin
    angajati_hranesc(id_partener);
```

```
end;
```

Error report -

ORA-20002: Acest partener nu interactioneaza cu nicio gradina zoologica.

ORA-06512: at "C##USERBOGDAN.ANGAJATI\_HRANESC", line 100

ORA-06512: at line 4

Pentru id\_partener = 140

```
exception
    when no_partner_found then
        raise_application_error(-20001, 'Nu exista un partener cu acest id.');
```

```
    when no_zoo_found then
        raise_application_error(-20002, 'Acest partener nu interactioneaza cu nicio gradina zoologica.');
```

```
end;
```

```
/
```

```
declare
    id_partener partener.id_part&type := &v_partener;
```

```
begin
```

```
    angajati_hranesc(id_partener);
```

```
end;
```

```
/
```

Script Output x Query Result x

Task completed in 2.303 seconds

Error starting at line : 7,047 in command -

```
declare
    id_partener partener.id_part&type := &v_partener;
```

```
begin
    angajati_hranesc(id_partener);
```

```
end;
```

Error report -

ORA-20001: Nu exista un partener cu acest id.

ORA-06512: at "C##USERBOGDAN.ANGAJATI\_HRANESC", line 98

ORA-06512: at line 4



Pentru id\_partener = 10

Exception  
when no partner found then

Script Output x Query Result x  
Task completed in 2.09 seconds

PL/SQL procedure successfully completed.

Dbms Output x  
Buffer Size: 20000

bogdan x

Gradinile zoologice cu care a interactionat partenerul cu id 10 sunt:

-----  
Gradina zoologica 1  
-----  
1) 506 Eliza Beck Dresor  
2) 800 Mamie Ryder Vanzator de produse  
3) 500 Brit Lam Om de serviciu  
4) 786 Joy Neville Om de serviciu  
-----  
Gradina zoologica 2  
-----  
Nu exista angajati care au hranit animalele si sa nu fie ingrijitori.  
-----  
Gradina zoologica 3  
-----  
1) 550 Fox Galvan Veterinar  
2) 551 Ishika Cash Veterinar  
-----  
Gradina zoologica 4  
-----  
1) 571 Arabella Mitchell Vanzator de produse  
2) 572 Rumaisa Lyon Vanzator de produse  
3) 790 Salim Boyd Mecanic  
-----  
Gradina zoologica 5  
-----

-----  
Gradina zoologica 5  
-----

1) 775    Susanna Moreno    Contabil

-----  
Gradina zoologica 6  
-----

1) 610    Beatrix Martinez    Vanzator de produse  
2) 781    Jonah Fowler    Vanzator de produse  
3) 595    Vivaan Ireland    Om de serviciu  
4) 596    Elena Butt    Om de serviciu

-----  
Gradina zoologica 7  
-----

1) 627    Ines Nolan    Veterinar  
2) 629    Suhail Squires    Vanzator de produse  
3) 626    Dominykas Nairn    Veterinar  
4) 788    Misha Larson    Administrator

-----  
Gradina zoologica 8  
-----

1) 648    Haya Simmonds    Vanzator de produse  
2) 647    Sarina Simpson    Vanzator de produse  
3) 777    Cerys Callaghan    Ghid  
4) 637    Isla-Rae Branch    Mecanic

-----  
Gradina zoologica 9  
-----

1) 667    Whitney Farrow    Vanzator de produse  
2) 769    Emanuel Ryan    Administrator  
3) 665    Sayed Coles    Veterinar  
4) 654    Essa Guerra    Om de serviciu  
5) 666    Alishia Boyce    Vanzator de produse

-----  
Gradina zoologica 10  
-----

1) 679    Demi-Leigh Pollard    Relatii externe  
2) 684    Renesmae Blackburn    Veterinar

---

-----  
Gradina zoologica 10  
-----

1)	679	Demi-Leigh Pollard	Relatii externe
2)	684	Renesmae Blackburn	Veterinar
3)	671	Azra Foster	Om de serviciu
4)	685	Eduardo Cannon	Vanzator de produse
5)	686	Lincoln Alcock	Vanzator de produse
6)	796	Evalyn Whitney	Ghid

-----  
Gradina zoologica 12  
-----

1)	724	Isla Cortez	Vanzator de produse
2)	797	Sia Portillo	Veterinar
3)	798	Loren Zhang	Veterinar

-----  
Gradina zoologica 13  
-----

1)	743	Amanda Watson	Vanzator de produse
2)	736	Kavan Mcfarland	Relatii externe

-----  
Gradina zoologica 14  
-----

1)	762	Andre Clarkson	Vanzator de produse
2)	761	Brooklyn Rangel	Vanzator de produse
3)	749	Mahira Gay	Om de serviciu

**7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat. Apelați subprogramul.**

*Enunt:*

Dandu-se un produs, sa se afle primele 5 gradini zoo care vand cele mai multe produse si sa se afiseze pentru fiecare, numarul de produse pe care il vand cat si vanzatorii de produse.

*Explicatie:*

Am folosit 2 cursoare parametrizate, unul clasic unde retin primele 5 gradini zoologice conform cerintei, primind ca parametru id\_produs si un ciclu cursor care primeste ca parametru id\_zoo si retine vanzatorii de produse din acea gradina zoologica.

Am tratat si exceptiile cand nu exista produsul respectiv sau nu este vandut produsul respectiv in nicio gradina zoologica, precum si cazul in care produsul nu este vandut in minim 5 gradini zoologice.

*Codul:*

```
CREATE OR replace PROCEDURE Numar_vanzatori(id_prod IN produse.id_produs%TYPE)
IS
    CURSOR numar_produce(
        id_p produse.id_produs%TYPE) IS
        SELECT *
        FROM (SELECT Count(*) nr_prod,
                        id_zoo
                FROM vand
                WHERE id_zoo IN (SELECT v.id_zoo
                                FROM vand v
                                WHERE v.id_produs = id_p)
                GROUP BY id_zoo
                ORDER BY 1 DESC)
        WHERE ROWNUM <= 5;
    CURSOR ang(
        id_grad gradina_zoologica.id_zoo%TYPE) IS
        SELECT prenume,
               nume
        FROM angajati
        WHERE id_zoo = id_grad
              AND id_job = 'VANZ_PROD';
    id_z      gradina_zoologica.id_zoo%TYPE;
    nr_p      NUMBER(3);
    no_product_found EXCEPTION;
    no_product_sold EXCEPTION;
    verificare NUMBER(3);
    nr_ordine NATURAL;
    nr_zoo    NATURAL := 0;
BEGIN
    SELECT Count(*)
    INTO verificare
    FROM produse
    WHERE id_produs = id_prod;

    IF verificare = 0 THEN
        RAISE no_product_found;
    END IF;

    SELECT Count(*)
    INTO verificare
    FROM vand
    WHERE id_produs = id_prod;

    IF verificare = 0 THEN
        RAISE no_product_sold;
    END IF;

    OPEN numar_produce(id_prod);

    LOOP
        FETCH numar_produce INTO nr_p, id_z;
```

```

        exit WHEN numar_produce%NOTFOUND;

dbms_output.Put_line('-----');
dbms_output.Put_line('-----');

dbms_output.Put_line('Gradina zoologica '
|| id_z
|| ' vinde '
|| nr_p
|| ' tipuri de produse. Vanzatorii de produse sunt:');

dbms_output.Put_line('-----');
dbms_output.Put_line('-----');

nr_ordine := 1;

nr_zoo := nr_zoo + 1;

FOR elem IN ang(id_z) LOOP
    dbms_output.Put_line(nr_ordine
|| ') '
|| elem.prenume
|| ' '
|| elem.numa);

    nr_ordine := nr_ordine + 1;
END LOOP;

dbms_output.new_line;

dbms_output.new_line;
END LOOP;

CLOSE numar_produce;

IF nr_zoo < 5 THEN
    dbms_output.Put_line('Produsul se afla doar in '
|| nr_zoo
|| ' gradini zoologice.');
```

```

END IF;
EXCEPTION
    WHEN no_product_found THEN
        Raise_application_error(-20001, 'Nu exista un produs cu acest id.')
```

```

; WHEN no_product_sold THEN
    Raise_application_error(-20002,
        'Acest produs nu este vandut de nicio gradina zoologica.');
```

```

END;

/
DECLARE
    id_produs produse.id_produs%TYPE := &v_produs;
BEGIN
    Numar_vanzatori(id_produs);
END;

/
```

## Pentru id\_produs = 1

```
exception
when no_product_found then
    raise_application_error(-20001, 'Nu exista un produs cu acest id.');
```

```
when no_product_sold then
    raise_application_error(-20002, 'Acest produs nu este vandut de nicio gradina zoologica.');
```

```
end;
```

```
/
```

```
declare
    id_produs produse.id_produs$type := &v_produs;
```

```
begin
```

```
    numar_vanzatori(id_produs);
```

```
end;
```

```
/
```

Script Output x Query Result x

Task completed in 1.447 seconds

Procedure NUMAR\_VANZATORI compiled

Error starting at line : 7,144 in command -

```
declare
    id_produs produse.id_produs$type := &v_produs;
```

```
begin
```

```
    numar_vanzatori(id_produs);
```

```
end;
```

Error report -

ORA-20001: Nu exista un produs cu acest id.

ORA-06512: at "C##USERBOGDAN.NUMAR\_VANZATORI", line 65

ORA-06512: at line 4

## Pentru id\_produs = 1200

```
exception
when no_product_found then
    raise_application_error(-20001, 'Nu exista un produs cu acest id.');
```

```
when no_product_sold then
    raise_application_error(-20002, 'Acest produs nu este vandut de nicio gradina zoologica.');
```

```
end;
```

```
/
```

```
declare
    id_produs produse.id_produs$type := &v_produs;
```

```
begin
```

```
    numar_vanzatori(id_produs);
```

```
end;
```

```
/
```

Script Output x Query Result x

Task completed in 2.217 seconds

Error starting at line : 7,144 in command -

```
declare
    id_produs produse.id_produs$type := &v_produs;
```

```
begin
```

```
    numar_vanzatori(id_produs);
```

```
end;
```

Error report -

ORA-20002: Acest produs nu este vandut de nicio gradina zoologica.

ORA-06512: at "C##USERBOGDAN.NUMAR\_VANZATORI", line 67

ORA-06512: at line 4

Pentru id\_produs = 1210

```

    end if,
exception
    when no_product_found then
        raise_application_error(-20001, 'Nu exista un produs cu acest id.');
```

```

    when no_product_sold then
        raise_application_error(-20002, 'Acest produs nu este vandut de nicio gradina zoologica.');
```

```

end;
/

declare
    id_produs produse.id_produs%type := &v_produs;
begin
    numar_vanzatori(id_produs);
end;
/
```

Script Output x Query Result x

Task completed in 2.661 seconds

PL/SQL procedure successfully completed.

Dbms Output x

Buffer Size: 20000

ogdan x

```

-----
Gradina zoologica 2 vinde 18 tipuri de produse. Vanzatorii de produse sunt:
-----
1) Wyatt Knott
2) Saima Pike

-----
Gradina zoologica 5 vinde 17 tipuri de produse. Vanzatorii de produse sunt:
-----
1) Daniel Woodward
2) Hamza Cohen

Produsul se afla doar in 2 gradini zoologice.
```

Pentru id\_produs = 1200

```
when no_product_found then
    raise_application_error(-20001, 'Nu exista un produs cu acest id.');
```

Script Output x Query Result x

Task completed in 2.345 seconds

PL/SQL procedure successfully completed.

Dbms Output x

Buffer Size: 20000

bogdan x

-----

Gradina zoologica 8 vinde 20 tipuri de produse. Vanzatorii de produse sunt:

-----

1) Sarina Simpson  
2) Haya Simmonds

-----

Gradina zoologica 9 vinde 20 tipuri de produse. Vanzatorii de produse sunt:

-----

1) Awais Ellison  
2) Alishia Boyce  
3) Whitney Farrow

-----

Gradina zoologica 12 vinde 19 tipuri de produse. Vanzatorii de produse sunt:

-----

1) Nina Krueger  
2) Isla Cortez

-----

Gradina zoologica 2 vinde 18 tipuri de produse. Vanzatorii de produse sunt:

-----

1) Wyatt Knott  
2) Saima Pike

-----

Gradina zoologica 5 vinde 17 tipuri de produse. Vanzatorii de produse sunt:

-----

1) Daniel Woodward  
2) Hamza Cohen

**8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Definiți minim 2 excepții. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.**



### *Enunt:*

Se da o gradina zoologica (id\_zoo). Sa se returneze numele partenerului care a facut cele mai multe donatii catre aceasta gradina zoologica. Sa se trateze exceptiile cand gradina zoo data nu exista, cand exista mai multi parteneri cu numar maxim de donatii si cand nu exista nicio donatie catre gradina zoo.

### *Explicatie:*

Am folosit tabelele **BENEFICIAZA** (pentru detaliile tranzactiilor dintre partenerii si gradinile zoo), **SERVICIU** (pentru a vedea care dintre tranzactii sunt donatii) si **PARTENER** (pentru a afla numele partenerului).

### *Codul:*

```
CREATE OR replace FUNCTION Donator_maxim(id_z gradina_zoologica.id_zoo%TYPE)
RETURN partener.nume_partener%TYPE
IS
    nume          partener.nume_partener%TYPE;
    no_zoo_found  EXCEPTION;
    verificare     NUMBER(1);
BEGIN
    SELECT Count(*)
    INTO   verificare
    FROM   gradina_zoologica
    WHERE  id_zoo = id_z;

    IF verificare = 0 THEN
        RAISE no_zoo_found;
    END IF;

    SELECT p.nume_partener
    INTO   nume
    FROM   beneficiaza b,
          partener p,
          serviciu s
    WHERE  b.id_zoo = id_z
          AND b.id_part = p.id_part
          AND b.id_serviciu = s.id_serviciu
          AND s.tip_serviciu = 'Donatie'
    GROUP BY p.nume_partener
    HAVING Count(*) = (SELECT Max(Count(*))
                      FROM   beneficiaza b,
                              serviciu s
                      WHERE  b.id_zoo = id_z
                            AND b.id_serviciu = s.id_serviciu
                            AND s.tip_serviciu = 'Donatie'
                      GROUP BY b.id_part);

    RETURN nume;
EXCEPTION
    WHEN no_zoo_found THEN
        Raise_application_error(-20001,
                                'Nu exista o gradina zoologica cu acest id.');
```

```

WHEN no_data_found THEN
    Raise_application_error(-20002,
        'Aceasta gradina zoologica nu a primit nicio donatie.');
```

too\_many\_rows THEN

```

    Raise_application_error(-20003,
        'Prea multi parteneri cu donatii maxime.');
```

END;

/

DECLARE

```

    id_z gradina_zoologica.id_zoo%TYPE := &v_zoo;
    nume partener.nume_partener%TYPE;
```

BEGIN

```

    nume := Donator_maxim(id_z);

    dbms_output.Put_line(nume);
```

END;

/

Pentru id\_zoo = 15

The screenshot shows the Oracle SQL Developer interface. The main window displays a PL/SQL script with the following content:

```

return nume;
exception
when no_zoo_found then
    raise_application_error(-20001, 'Nu exista o gradina zoologica cu acest id.');
```

when no\_data\_found then

```

    raise_application_error(-20002, 'Aceasta gradina zoologica nu a primit nicio donatie.');
```

when too\_many\_rows then

```

    raise_application_error(-20003, 'Prea multi parteneri cu donatii maxime.');
```

end;

/

declare

```

    id_z gradina_zoologica.id_zoo%type := &v_zoo;
    nume partener.nume_partener%type;
```

begin

```

    nume := donator_maxim(id_z);
    dbms_output.put_line(nume);
```

end;

/

The bottom panel shows the 'Script Output' tab with the message: 'Task completed in 4.287 seconds'.

Function DONATOR\_MAXIM compiled

Error starting at line : 7,211 in command -

```

declare
    id_z gradina_zoologica.id_zoo%type := &v_zoo;
    nume partener.nume_partener%type;
```

begin

```

    nume := donator_maxim(id_z);
    dbms_output.put_line(nume);
```

end;

Error report -

```

ORA-20002: Aceasta gradina zoologica nu a primit nicio donatie.
ORA-06512: at "C##USERBOGDAN.DONATOR_MAXIM", line 43
ORA-06512: at line 5
```

Pentru id\_zoo = 1

```
exception
when no_zoo_found then
    raise_application_error(-20001, 'Nu exista o gradina zoologica cu acest id.');
```

```
when no_data_found then
    raise_application_error(-20002, 'Aceasta gradina zoologica nu a primit nicio donatie.');
```

```
when too_many_rows then
    raise_application_error(-20003, 'Prea multi parteneri cu donatii maxime.');
```

```
end;
```

```
/
```

```
declare
    id_z gradina_zoologica.id_zoo%type := &v_zoo;
    nume partener.nume_partener%type;
```

```
begin
    nume := donator_maxim(id_z);
    dbms_output.put_line(nume);
```

```
end;
```

```
/
```

Script Output x Query Result x

Task completed in 1.275 seconds

PL/SQL procedure successfully completed.

Dbms Output x

Buffer Size: 20000

bogdan x

BEGAS

Pentru id\_zoo = 3

```
exception
when no_zoo_found then
    raise_application_error(-20001, 'Nu exista o gradina zoologica cu acest id.');
```

```
when no_data_found then
    raise_application_error(-20002, 'Aceasta gradina zoologica nu a primit nicio donatie.');
```

```
when too_many_rows then
    raise_application_error(-20003, 'Prea multi parteneri cu donatii maxime.');
```

```
end;
```

```
/
```

```
declare
    id_z gradina_zoologica.id_zoo%type := &v_zoo;
    nume partener.nume_partener%type;
```

```
begin
    nume := donator_maxim(id_z);
    dbms_output.put_line(nume);
```

```
end;
```

```
/
```

Script Output x Query Result x

Task completed in 2.097 seconds

Error starting at line : 7,204 in command -

```
declare
    id_z gradina_zoologica.id_zoo%type := &v_zoo;
    nume partener.nume_partener%type;
```

```
begin
    nume := donator_maxim(id_z);
    dbms_output.put_line(nume);
```

```
end;
```

Error report -

ORA-20003: Prea multi parteneri cu donatii maxime.

ORA-06512: at "C##USERBOGDAN.DONATOR\_MAXIM", line 38

ORA-06512: at line 5

Pentru id\_zoo = 22

```
exception
when no_zoo_found then
raise_application_error(-20001, 'Nu exista o gradina zoologica cu acest id.');
```

```
when no_data_found then
raise_application_error(-20002, 'Aceasta gradina zoologica nu a primit nicio donatie.');
```

```
when too_many_rows then
raise_application_error(-20003, 'Prea multi parteneri cu donatii maxime.');
```

```
end;
```

```
/
```

```
declare
id_z gradina_zoologica.id_zoo%type := &v_zoo;
```

```
nume partener.nume_partener%type;
```

```
begin
```

```
nume := donator_maxim(id_z);
```

```
dbms_output.put_line(nume);
```

```
end;
```

```
/
```

Script Output x Query Result x

Task completed in 0.965 seconds

Error starting at line : 7,204 in command -

```
declare
id_z gradina_zoologica.id_zoo%type := &v_zoo;
```

```
nume partener.nume_partener%type;
```

```
begin
```

```
nume := donator_maxim(id_z);
```

```
dbms_output.put_line(nume);
```

```
end;
```

Error report -

ORA-20001: Nu exista o gradina zoologica cu acest id.

ORA-06512: at "C##USERBOGDAN.DONATOR\_MAXIM", line 34

ORA-06512: at line 5

**9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO\_DATA\_FOUND și TOO\_MANY\_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.**

*Enunt:*

Dandu-se o gradina zoologica (id\_zoo), sa se afle cea mai comuna specie de animale din aceasta. Apoi sa se afiseze animalele care fac parte din aceasta specie (nu si din aceasta gradina zoologica) care , in ziua de 16 mai 2022, au fost hranite in afara intervalului orar 08:00 - 14:00 cu un tip de hrana ce este valabil 2 saptamani de la cumparare, de catre un angajat care castiga mai putin decat salariul mediu al job-ului sau, adica media dintre salariul minim si maxim al job-ului. Sa se trateze cazurile cand id\_zoo nu exista, cand in gradina zoo sunt mai multe specii comune sau cand gradina zoo inca nu adaposteste animale.

### *Explicatie:*

Verificam mai intai cazurile de exceptie. Apoi salvam intr-un tablou indexat de recorduri animalele conform cerintei si le afisam.

### *Codul:*

```
CREATE OR replace PROCEDURE Animale_specie(id_z IN
gradina_zoologica.id_zoo%TYPE)
IS
    TYPE animal IS RECORD (
        id_animal animale.id_animal%TYPE,
        nume animale.nume%TYPE,
        varsta animale.varsta%TYPE,
        id_zoo gradina_zoologica.id_zoo%TYPE,
        temperament animale.temperament%TYPE );
    TYPE lista_animale
    IS TABLE OF ANIMAL INDEX BY PLS_INTEGER;
    lista_animali lista_animale;
    sp            specie.id_specie%TYPE;
    no_zoo_found EXCEPTION;
    verificare NUMBER(1);
    nr_ordine NATURAL := 1;
BEGIN
    BEGIN
        SELECT Count(*)
        INTO   verificare
        FROM   gradina_zoologica
        WHERE  id_zoo = id_z;

        IF verificare = 0 THEN
            RAISE no_zoo_found;
        END IF;

        SELECT id_specie
        INTO   sp
        FROM   animale
        WHERE  id_zoo = id_z
        GROUP BY id_specie
        HAVING Count(*) = (SELECT Max(Count(*))
                           FROM   animale
                           WHERE  id_zoo = id_z
                           GROUP BY id_specie);
    EXCEPTION
        WHEN no_zoo_found THEN
            Raise_application_error(-20001,
                'Nu exista o gradina zoologica cu acest id. ');
        WHEN no_data_found THEN
            Raise_application_error(-20002,
                'Aceasta gradina zoologica nu adaposteste inca niciun animal. ');
        WHEN too_many_rows THEN
            Raise_application_error(-20003, 'Prea multe specii comune. ');
    END;

    SELECT ani.id_animal,
           ani.nume,
           ani.varsta,
```

```

        ani.id_zoo,
        ani.temperament
bulk    collect INTO lista_ani
FROM    animale ani,
        hrana hr,
        angajati ang,
        hraneste hran,
        tip_hrana t
WHERE   ani.id_animal = hran.id_animal
        AND ang.id_ang = hran.id_ang
        AND hr.id_hrana = hran.id_hrana
        AND hr.id_tip_hrana = t.id_tip_hrana
        AND hran.data_hranire = '16-MAY-22'
        AND t.sapt_valabilitate = 2
        AND ani.id_specie = sp
        AND ( To_char(hran.ora_hranire, 'HH24:MI') > '14:00'
              OR To_char(hran.ora_hranire, 'HH24:MI') < '08:00' )
        AND ang.id_ang IN (SELECT a.id_ang
                           FROM   angajati a
                           WHERE  a.salariu <= (SELECT ( salariu_max +
                                                         salariu_min ) / 2
                                                FROM   jobs
                                                WHERE  a.id_job = id_job));

dbms_output.Put_line('Specia comuna este '
                    || sp);

FOR i IN lista_ani.first..lista_ani.last LOOP
    dbms_output.Put_line(nr_ordine
                        || ') ID '
                        || Lista_ani(i).id_animal
                        || ', '
                        || Lista_ani(i).nume
                        || ', are '
                        || Lista_ani(i).varsta
                        || ' ani, se afla la gradina zoo '
                        || Lista_ani(i).id_zoo
                        || ' si este '
                        || Lista_ani(i).temperament);

    nr_ordine := nr_ordine + 1;
END LOOP;
END;

/
DECLARE
    id_z gradina_zoologica.id_zoo%TYPE := &v_zoo;
BEGIN
    Animale_specie(id_z);
END;

/

```

Pentru id\_zoo = 22

```
for i in lista_ani.first..lista_ani.last loop
    dbms_output.put_line(nr_ordine || ' ID ' || lista_ani(i).id_animal || ', ' ||
        nr_ordine := nr_ordine +1;
end loop;
end;
/

declare
    id_z gradina_zoologica.id_zoo%type := &v_zoo;
begin
    animale_specie(id_z);
end;
/
```

Script Output x Query Result x

Task completed in 4.697 seconds

Procedure ANIMALE\_SPECIE compiled

Error starting at line : 7,296 in command -

```
declare
    id_z gradina_zoologica.id_zoo%type := &v_zoo;
begin
    animale_specie(id_z);
end;
```

Error report -

ORA-20001: Nu exista o gradina zoologica cu acest id.  
ORA-06512: at "C##USERBOGDAN.ANIMALE\_SPECIE", line 38  
ORA-06512: at line 4

Pentru id\_zoo = 1

```
for i in lista_ani.first..lista_ani.last loop
    dbms_output.put_line(nr_ordine || ' ID ' || lista_ani(i).id_animal ||
        nr_ordine := nr_ordine +1;
end loop;
end;
/

declare
    id_z gradina_zoologica.id_zoo%type := &v_zoo;
begin
    animale_specie(id_z);
end;
/
```

Script Output x Query Result x

Task completed in 1.109 seconds

Error starting at line : 7,296 in command -

```
declare
    id_z gradina_zoologica.id_zoo%type := &v_zoo;
begin
    animale_specie(id_z);
end;
```

Error report -

ORA-20003: Prea multe specii comune.  
ORA-06512: at "C##USERBOGDAN.ANIMALE\_SPECIE", line 42  
ORA-06512: at line 4

Pentru id\_zoo = 15

```
for i in lista_ani.first..lista_ani.last loop
    dbms_output.put_line(nr_ordine || ' ID ' || lista_ani(i).id_animal || ', ' || lista_ani(i).nume || ' are ' || lista_ani(i).varsta || ' ani');
    nr_ordine := nr_ordine + 1;
end loop;
end;
/

declare
    id_z gradina_zoologica.id_zoo%type := &v_zoo;
begin
    animale_specie(id_z);
end;
/
```

Script Output x Query Result x

Task completed in 0.909 seconds

Error starting at line : 7,296 in command -  
declare  
 id\_z gradina\_zoologica.id\_zoo%type := &v\_zoo;  
begin  
 animale\_specie(id\_z);  
end;  
Error report -  
ORA-20002: Aceasta gradina zoologica nu adaposteste inca niciun animal.  
ORA-06512: at "C#USERBOGDAN.ANIMALE\_SPECIE", line 40  
ORA-06512: at line 4

Pentru id\_zoo = 2

```
for i in lista_ani.first..lista_ani.last loop
    dbms_output.put_line(nr_ordine || ' ID ' || lista_ani(i).id_animal || ', ' || lista_ani(i).nume || ' are ' || lista_ani(i).varsta || ' ani');
    nr_ordine := nr_ordine + 1;
end loop;
end;
/

declare
    id_z gradina_zoologica.id_zoo%type := &v_zoo;
begin
    animale_specie(id_z);
end;
/
```

Script Output x Query Result x

Task completed in 4.058 seconds

PL/SQL procedure successfully completed.

Dbms Output x

Buffer Size: 20000

bogdan x

Specia comuna este CAP\_EU

- 1) ID 238, Danniella, are 16 ani, se afla la gradina zoo 9 si este Vesel
- 2) ID 307, Siya, are 11 ani, se afla la gradina zoo 2 si este Calm
- 3) ID 367, Rylan, are 8 ani, se afla la gradina zoo 6 si este Agresiv
- 4) ID 136, Isa, are 14 ani, se afla la gradina zoo 10 si este Prietenos
- 5) ID 116, Szymon, are 11 ani, se afla la gradina zoo 13 si este Infricosat
- 6) ID 345, Callum, are 12 ani, se afla la gradina zoo 14 si este Prietenos



## 10. Definiți un *trigger* de tip LMD la nivel de comandă. Declanșați *trigger*-ul.

*Enunt:*

Sa se creeze un trigger care la nivel de comanda care actualizeaza tabelul numar\_hrana, ce contine informatii despre cate tipuri de hrana are fiecare gradina zoologica.

*Explicatie:*

Am creat trigger-ul dupa insert sau delete pe tabela hrana care updateaza tabela in numar\_hrana pentru fiecare gradina zoologica, numarul de tipuri de hrana.

*Codul:*

```
CREATE OR replace TRIGGER update_hrana
  AFTER INSERT OR DELETE ON hrana
BEGIN
  FOR elem IN (SELECT id_zoo,
                     Count(*) nr
                FROM   hrana
                GROUP BY id_zoo) LOOP
    UPDATE numar_hrana
    SET    nr = elem.nr
    WHERE  id_zoo = elem.id_zoo;

    IF SQL%NOTFOUND THEN
      INSERT INTO numar_hrana
        (id_zoo,
         nr)
      VALUES (elem.id_zoo,
              elem.nr);
    END IF;
  END LOOP;
END;

/
DELETE FROM hrana
WHERE  id_hrana = 700
AND id_zoo = 1;

INSERT INTO hrana
  (id_hrana,
   cantitate,
   id_zoo,
   id_tip_hrana)
VALUES (700,
        430,
        1,
        'CER_SP');
```

```

SELECT *
FROM hrana
WHERE id_zoo = 1;

SELECT *
FROM numar_hrana;

```

Exemplul este pentru id\_zoo = 1

Inregistrarile initiale din tabela

```

select *
from hrana
where id_zoo=1;

```

ID_HRANA	CANTITATE	ID_ZOO	ID_TIP_HRANA
1	700	430	1 CER_SP
2	701	890	1 CRNO_US
3	702	20	1 CRNV_CH
4	703	450	1 FUR_GER
5	704	580	1 INS_IN
6	705	440	1 MRCV_IT
7	706	260	1 POR_CAN
8	707	180	1 PST_BUG
9	708	980	1 SAL_AUS
10	709	810	1 SEM_RO

Stergem o inregistrare

```

delete from hrana
where id_hrana = 700
and id_zoo = 1;

select *
from hrana
where id_zoo=1;

```

ID_HRANA	CANTITATE	ID_ZOO	ID_TIP_HRANA
1	701	890	1 CRNO_US
2	702	20	1 CRNV_CH
3	703	450	1 FUR_GER
4	704	580	1 INS_IN
5	705	440	1 MRCV_IT
6	706	260	1 POR_CAN
7	707	180	1 PST_BUG
8	708	980	1 SAL_AUS
9	709	810	1 SEM_RO

## Trigger-ul s-a declansat

```
select *  
from numar_hrana;
```

Script Output x Query Result x

SQL | All Rows Fetched: 14 in 0

ID_ZOO	NR
1	9
2	10
3	10
4	10
5	10
6	10
7	10
8	10
9	10
10	10
11	10
12	10
13	10
14	10

## Inseram inapoi inregistrarea

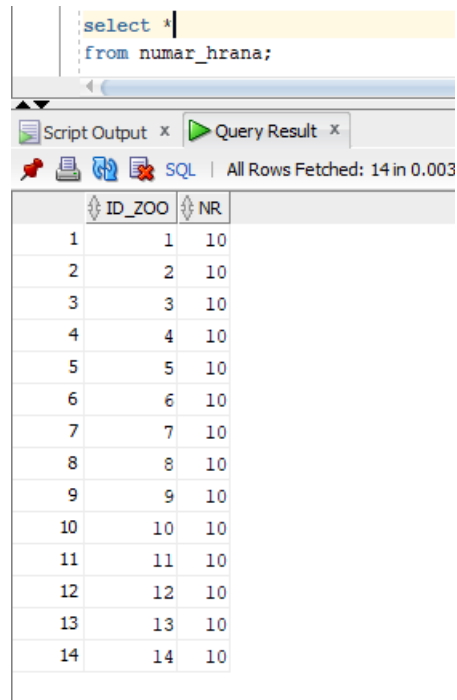
```
insert into HRANA(id_hrana,cantitate,id_zoo,id_tip_hrana)  
values (700,430,1,'CER_SP');  
  
select *  
from hrana  
where id_zoo=1;
```

Script Output x Query Result x

SQL | All Rows Fetched: 10 in 0.003 seconds

ID_HRANA	CANTITATE	ID_ZOO	ID_TIP_HRANA
1	701	890	1 CRNO_US
2	702	20	1 CRNV_CH
3	703	450	1 FUR_GER
4	704	580	1 INS_IN
5	705	440	1 MRCV_IT
6	706	260	1 POR_CAN
7	707	180	1 PST_BUG
8	708	980	1 SAL_AUS
9	709	810	1 SEM_RO
10	700	430	1 CER_SP

Trigger-ul se declanseaza din nou



ID_ZOO	NR
1	10
2	10
3	10
4	10
5	10
6	10
7	10
8	10
9	10
10	10
11	10
12	10
13	10
14	10

## 11. Definiți un *trigger* de tip LMD la nivel de linie. Declanșați *trigger*-ul.

*Enunt:*

Sa se creeze un trigger care nu permite inserarea unei inregistrari de hranire a unui animal, daca acesta are restrictie de interzicere hrana, in interiorul perioadei de restrictie.

*Explicatie:*

Verificam inainte de insert daca animalul poate fi hranit.

*Codul:*

```
CREATE OR replace TRIGGER hraneste_animal
  BEFORE INSERT ON hraneste
  FOR EACH ROW
DECLARE
  verificare NUMBER(1);
BEGIN
  SELECT Count(*)
  INTO  verificare
```

```

FROM    restrictie
WHERE    tip_restrictie = 'INT_HR'
        AND id_animal = :new.id_animal
        AND :new.data_hranire BETWEEN data_inceput AND data_sfarsit;

IF verificare != 0 THEN
    Raise_application_error(-20001,
        'Animalul are restrictie de interzicere a hranii.');
```

END IF;

END;

/

```

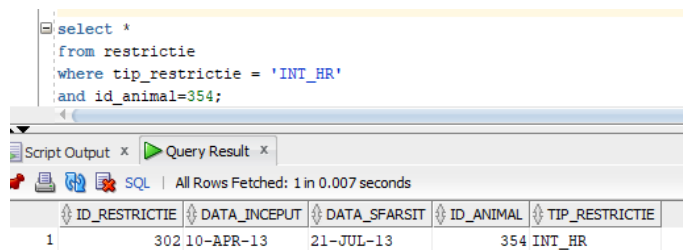
INSERT INTO hraneste
        (id_animal,
         id_ang,
         id_hrana,
         data_hranire,
         ora_hranire,
         cantitate)
VALUES    (354,
          783,
          707,
          '16-APR-2013',
          To_date('16-APR-2013 17:31', 'DD-MON-YYYY HH24:MI'),
          1);

INSERT INTO hraneste
        (id_animal,
         id_ang,
         id_hrana,
         data_hranire,
         ora_hranire,
         cantitate)
VALUES    (354,
          783,
          707,
          '16-APR-2014',
          To_date('16-APR-2014 17:31', 'DD-MON-YYYY HH24:MI'),
          1);

SELECT *
FROM    restrictie
WHERE    tip_restrictie = 'INT_HR'
        AND id_animal = 354;
```

Pentru animalul cu id\_animal = 354

Avem restrictie intre 10 Aprilie 2013 – 21 Iulie 2013



```

select *
from restrictie
where tip_restrictie = 'INT_HR'
and id_animal=354;
```

ID_RESTRICTIE	DATA_INCEPUT	DATA_SFARSIT	ID_ANIMAL	TIP_RESTRICTIE
1	302 10-APR-13	21-JUL-13	354	INT_HR

Nu merge sa hranim in 16 Aprilie 2013

Se afla in interiorul perioadei

```
into verificare
from restrictie
where tip_restrictie = 'INT_HR'
and id_animal = :new.id_animal
and :new.data_hranire BETWEEN data_inceput and data_sfarsit;

if verificare != 0 then
    raise_application_error(-20001, 'Animalul are restrictie de interzicere a hranii.');
```

```
end if;
end;
/

insert into HRANESTE(id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
values(354,783,707,'16-APR-2013',TO_DATE('16-APR-2013 17:31','DD-MON-YYYY HH24:MI'),1);
```

Script Output x Query Result x

Task completed in 0.039 seconds

Trigger HRANESTE\_ANIMAL compiled

Error starting at line : 7,373 in command -

```
insert into HRANESTE(id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
values(354,783,707,'16-APR-2013',TO_DATE('16-APR-2013 17:31','DD-MON-YYYY HH24:MI'),1)
```

Error report -

ORA-20001: Animalul are restrictie de interzicere a hranii.

ORA-06512: at "C##USERBOGDAN.HRANESTE\_ANIMAL", line 12

ORA-04088: error during execution of trigger 'C##USERBOGDAN.HRANESTE\_ANIMAL'

Pentru 16 Aprilie 2014 merge

Nu se afla in interiorul perioadei

```
insert into HRANESTE(id_animal,id_ang,id_hrana,data_hranire,ora_hranire,cantitate)
values(354,783,707,'16-APR-2014',TO_DATE('16-APR-2014 17:31','DD-MON-YYYY HH24:MI'),1);
```

Script Output x Query Result x

Task completed in 0.043 seconds

1 row inserted.

## 12. Definiți un *trigger* de tip LDD. Declanșați *trigger*-ul.

*Enunt:*

Sa se creeze un trigger care introduce in tabela evenimente detalii despre comanda LDD executata.

*Codul:*

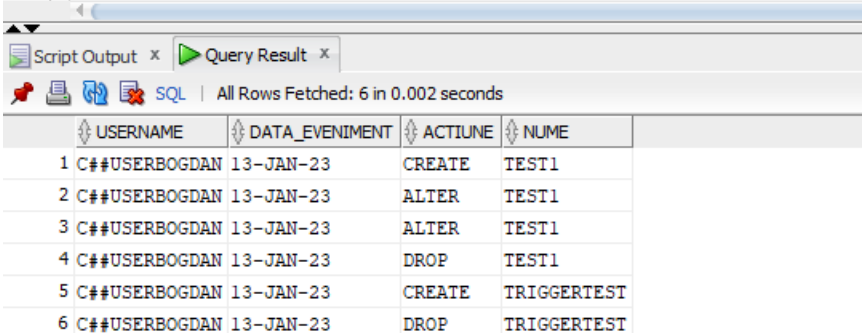
```
CREATE OR replace TRIGGER detalii
  AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
  INSERT INTO evenimente
    (username,
     data_eveniment,
     actiune,
     nume)
  VALUES
    (USER,
     SYSDATE,
     ora_sysevent,
     ora_dict_obj_name);
END;

/
SELECT *
FROM   evenimente;
```

Dupa cateva teste pentru declansarea trigger-ului

```
create or replace trigger detalii
  after create or drop or alter on schema
begin
  insert into evenimente(username, data_eveniment, actiune, nume)
  values (user,sysdate,ora_sysevent,ora_dict_obj_name);
end;
/

select *
from evenimente;
```



The screenshot shows a database interface with a 'Query Result' tab. It displays 6 rows of data from the 'evenimente' table. The columns are USERNAME, DATA\_EVENTIMENT, ACTIUNE, and NUME. The first four rows show events for 'C##USERBOGDAN' on '13-JAN-23' with actions 'CREATE', 'ALTER', 'ALTER', and 'DROP' respectively, all for 'TEST1'. The last two rows show events for 'C##USERBOGDAN' on '13-JAN-23' with actions 'CREATE' and 'DROP' respectively, for 'TRIGGERTEST'.

	USERNAME	DATA_EVENTIMENT	ACTIUNE	NUME
1	C##USERBOGDAN	13-JAN-23	CREATE	TEST1
2	C##USERBOGDAN	13-JAN-23	ALTER	TEST1
3	C##USERBOGDAN	13-JAN-23	ALTER	TEST1
4	C##USERBOGDAN	13-JAN-23	DROP	TEST1
5	C##USERBOGDAN	13-JAN-23	CREATE	TRIGGERTEST
6	C##USERBOGDAN	13-JAN-23	DROP	TRIGGERTEST

### 13. Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.

*Codul:*

```
CREATE OR replace PACKAGE gradini_zoo
IS
    PROCEDURE angajati_hranesc(
        id_partener IN partener.id_part%TYPE);
    PROCEDURE numar_vanzatori(
        id_prod IN produse.id_produs%TYPE);
    FUNCTION Donator_maxim(
        id_z gradina_zoologica.id_zoo%TYPE)
    RETURN partener.nume_partener%TYPE;
    PROCEDURE animale_specie(
        id_z IN gradina_zoologica.id_zoo%TYPE);
END;

/

CREATE OR replace PACKAGE BODY gradini_zoo
IS
    PROCEDURE Angajati_hranesc(id_partener IN partener.id_part%TYPE)
    IS
        TYPE ang IS RECORD (
            id_ang angajati.id_ang%TYPE,
            nume angajati.nume%TYPE,
            prenume angajati.prenume%TYPE,
            titlu_job jobs.titlu_job%TYPE );
        TYPE lista_ang
            IS TABLE OF ANG;
        TYPE gradzoo
            IS TABLE OF LISTA_ANG INDEX BY PLS_INTEGER;
        TYPE nume_job
            IS TABLE OF jobs.titlu_job%TYPE INDEX BY VARCHAR2(10);
        no_partner_found EXCEPTION;
        no_zoo_found EXCEPTION;
        verificare          NUMBER(3);
        angajat_job         NUME_JOB;
        CURSOR joburi IS
            SELECT id_job,
                   titlu_job
            FROM    jobs;
        CURSOR gradini_zoo(
            id_p partener.id_part%TYPE) IS
            SELECT DISTINCT id_zoo
            FROM    beneficiaza
            WHERE   id_part = id_p;
        CURSOR cursor_angajati(
            id_p partener.id_part%TYPE) IS
            SELECT DISTINCT a.id_ang,
                           a.nume,
                           a.prenume,
                           a.id_job,
                           a.id_zoo
            FROM      angajati a,
                   hraneste h
```



```

WHERE a.id_zoo IN (SELECT DISTINCT id_zoo
                    FROM   beneficiaza
                    WHERE  id_part = id_p)
      AND h.id_ang = a.id_ang
      AND a.id_job != 'INGRJ';
lista_angajati GRADZOO;
aux           ANG;
i             PLS_INTEGER;
nr_ordine     NATURAL;
BEGIN
  SELECT Count(*)
  INTO   verificare
  FROM   partener
  WHERE  id_part = id_partener;

  IF verificare = 0 THEN
    RAISE no_partner_found;
  END IF;

  SELECT Count(*)
  INTO   verificare
  FROM   beneficiaza
  WHERE  id_part = id_partener;

  IF verificare = 0 THEN
    RAISE no_zoo_found;
  END IF;

  FOR elem IN joburi LOOP
    Angajat_job(elem.id_job) := elem.titlu_job;
  END LOOP;

  FOR elem IN gradini_zoo(id_partener) LOOP
    Lista_angajati(elem.id_zoo) := Lista_ang();
  END LOOP;

  FOR elem IN cursor_angajati(id_partener) LOOP
    aux.id_ang := elem.id_ang;

    aux.nume := elem.nume;

    aux.prenume := elem.prenume;

    aux.titlu_job := Angajat_job(elem.id_job);

    Lista_angajati(elem.id_zoo).extend;

    lista_angajati(elem.id_zoo)(Lista_angajati(elem.id_zoo).last) := aux;
  END LOOP;

  dbms_output.Put_line('Gradinile zoologice cu care a interactionat partenerul cu id '
                        || id_partener
                        || ' sunt:');

  dbms_output.new_line;

  dbms_output.new_line;

```

```

i := lista_angajati.first;

WHILE i <= lista_angajati.last LOOP
    dbms_output.Put_line('-----');

    dbms_output.Put_line('Gradina zoologica '
                        || i);

    dbms_output.Put_line('-----');

    IF Lista_angajati(i).count = 0 THEN
dbms_output.Put_line('Nu exista angajati care au hranit animalele si sa nu fie ingriji
tori. ');
    ELSE
        nr_ordine := 1;

        FOR j IN Lista_angajati(i).first..Lista_angajati(i).last LOOP
            dbms_output.Put_line(nr_ordine
                                || ' '
                                || lista_angajati(i)(j).id_ang
                                || ' '
                                || lista_angajati(i)(j).prenume
                                || ' '
                                || lista_angajati(i)(j).nume
                                || ' '
                                || lista_angajati(i)(j).titlu_job);

            nr_ordine := nr_ordine + 1;
        END LOOP;
    END IF;

    dbms_output.new_line;

    dbms_output.new_line;

i := lista_angajati.NEXT(i);
END LOOP;
EXCEPTION
    WHEN no_partner_found THEN
        Raise_application_error(-20001,
            'Nu exista un partener cu acest id. ');
    ; WHEN no_zoo_found THEN
        Raise_application_error(-20002,
            'Acest partener nu interactioneaza cu nicio gradina zoologica. ');
END;

PROCEDURE Numar_vanzatori(id_prod IN produse.id_produș%TYPE)
IS
    CURSOR numar_produce(
        id_p produse.id_produș%TYPE) IS
        SELECT *
        FROM (SELECT Count(*) nr_prod,
                    id_zoo
                FROM vand
                WHERE id_zoo IN (SELECT v.id_zoo
                                FROM vand v
                                WHERE v.id_produș = id_p)

                GROUP BY id_zoo
                ORDER BY 1 DESC)

```

```

        WHERE ROWNUM <= 5;
CURSOR ang(
    id_grad gradina_zoologica.id_zoo%TYPE) IS
    SELECT prenume,
           nume
    FROM   angajati
    WHERE  id_zoo = id_grad
           AND id_job = 'VANZ_PROD';
id_z      gradina_zoologica.id_zoo%TYPE;
nr_p      NUMBER(3);
no_product_found EXCEPTION;
no_product_sold EXCEPTION;
verificare NUMBER(3);
nr_ordine NATURAL;
nr_zoo    NATURAL := 0;
BEGIN
    SELECT Count(*)
    INTO   verificare
    FROM   produse
    WHERE  id_produc = id_prod;

    IF verificare = 0 THEN
        RAISE no_product_found;
    END IF;

    SELECT Count(*)
    INTO   verificare
    FROM   vand
    WHERE  id_produc = id_prod;

    IF verificare = 0 THEN
        RAISE no_product_sold;
    END IF;

    OPEN numar_produce(id_prod);

    LOOP
        FETCH numar_produce INTO nr_p, id_z;

        exit WHEN numar_produce%NOTFOUND;

dbms_output.Put_line('-----
-----');

dbms_output.Put_line('Gradina zoologica '
                    || id_z
                    || ' vinde '
                    || nr_p
                    || ' tipuri de produse. Vanzatorii de produse sunt:');

dbms_output.Put_line('-----
-----');

nr_ordine := 1;

nr_zoo := nr_zoo + 1;

FOR elem IN ang(id_z) LOOP

```

```

        dbms_output.Put_line(nr_ordine
                               || ' '
                               || elem.prenume
                               || ' '
                               || elem.num);

        nr_ordine := nr_ordine + 1;
END LOOP;

dbms_output.new_line;

dbms_output.new_line;
END LOOP;

CLOSE numar_produce;

IF nr_zoo < 5 THEN
    dbms_output.Put_line('Produsul se afla doar in '
                          || nr_zoo
                          || ' gradini zoologice.');
```

END IF;

EXCEPTION

    WHEN no\_product\_found THEN

        Raise\_application\_error(-20001, 'Nu exista un produs cu acest id.')

    ; WHEN no\_product\_sold THEN

        Raise\_application\_error(-20002,

            'Acest produs nu este vandut de nicio gradina zoologica.');

END;

FUNCTION Donator\_maxim(id\_z gradina\_zoologica.id\_zoo%TYPE)

RETURN partener.nume\_partener%TYPE

IS

    nume          partener.nume\_partener%TYPE;

    no\_zoo\_found EXCEPTION;

    verificare NUMBER(1);

BEGIN

    SELECT Count(\*)

    INTO  verificare

    FROM  gradina\_zoologica

    WHERE  id\_zoo = id\_z;

    IF verificare = 0 THEN

        RAISE no\_zoo\_found;

    END IF;

    SELECT p.nume\_partener

    INTO  nume

    FROM  beneficiaza b,

          partener p,

          serviciu s

    WHERE  b.id\_zoo = id\_z

          AND b.id\_part = p.id\_part

          AND b.id\_serviciu = s.id\_serviciu

          AND s.tip\_serviciu = 'Donatie'

    GROUP BY p.nume\_partener

    HAVING Count(\*) = (SELECT Max(Count(\*))

                      FROM  beneficiaza b,

                          serviciu s

                      WHERE  b.id\_zoo = id\_z

```

        AND b.id_serviciu = s.id_serviciu
        AND s.tip_serviciu = 'Donatie'
    GROUP BY b.id_part);

    RETURN nume;
EXCEPTION
    WHEN no_zoo_found THEN
        Raise_application_error(-20001,
            'Nu exista o gradina zoologica cu acest id.');
```

WHEN no\_data\_found THEN

```

        Raise_application_error(-20002,
            'Aceasta gradina zoologica nu a primit nicio donatie.');
```

too\_many\_rows THEN

```

        Raise_application_error(-20003,
            'Prea multi parteneri cu donatii maxime.');
```

END;

PROCEDURE Animale\_specie(id\_z IN gradina\_zoologica.id\_zoo%TYPE)

IS

```

    TYPE animal IS RECORD (
        id_animal animale.id_animal%TYPE,
        nume animale.nume%TYPE,
        varsta animale.varsta%TYPE,
        id_zoo gradina_zoologica.id_zoo%TYPE,
        temperament animale.temperament%TYPE );
    TYPE lista_animale
        IS TABLE OF ANIMAL INDEX BY PLS_INTEGER;
    lista_animalista LISTA_ANIMALE;
    sp specie.id_specie%TYPE;
    no_zoo_found EXCEPTION;
    verificare NUMBER(1);
    nr_ordine NATURAL := 1;
```

BEGIN

```

    BEGIN
        SELECT Count(*)
        INTO verificare
        FROM gradina_zoologica
        WHERE id_zoo = id_z;

        IF verificare = 0 THEN
            RAISE no_zoo_found;
        END IF;

        SELECT id_specie
        INTO sp
        FROM animale
        WHERE id_zoo = id_z
        GROUP BY id_specie
        HAVING Count(*) = (SELECT Max(Count(*))
                           FROM animale
                           WHERE id_zoo = id_z
                           GROUP BY id_specie);
```

EXCEPTION

```

    WHEN no_zoo_found THEN
        Raise_application_error(-20001,
            'Nu exista o gradina zoologica cu acest id.');
```

WHEN no\_data\_found THEN

```

        Raise_application_error(-20002,
            'Aceasta gradina zoologica nu adaposteste inca niciun animal.');
```

```

        WHEN too_many_rows THEN
            Raise_application_error(-20003, 'Prea multe specii comune.');
```

END;

```

SELECT ani.id_animal,
       ani.nume,
       ani.varsta,
       ani.id_zoo,
       ani.temperament
bulk   collect INTO lista_ani
FROM   animale ani,
       hrana hr,
       angajati ang,
       hraneste hran,
       tip_hrana t
WHERE  ani.id_animal = hran.id_animal
      AND ang.id_ang = hran.id_ang
      AND hr.id_hrana = hran.id_hrana
      AND hr.id_tip_hrana = t.id_tip_hrana
      AND hran.data_hranire = '16-MAY-22'
      AND t.sapt_valabilitate = 2
      AND ani.id_specie = sp
      AND ( To_char(hran.ora_hranire, 'HH24:MI') > '14:00'
            OR To_char(hran.ora_hranire, 'HH24:MI') < '08:00' )
      AND ang.id_ang IN (SELECT a.id_ang
                        FROM   angajati a
                        WHERE  a.salariu <= (SELECT ( salariu_max +
                                                    salariu_min ) / 2
                        FROM   jobs
                        WHERE  a.id_job = id_job));

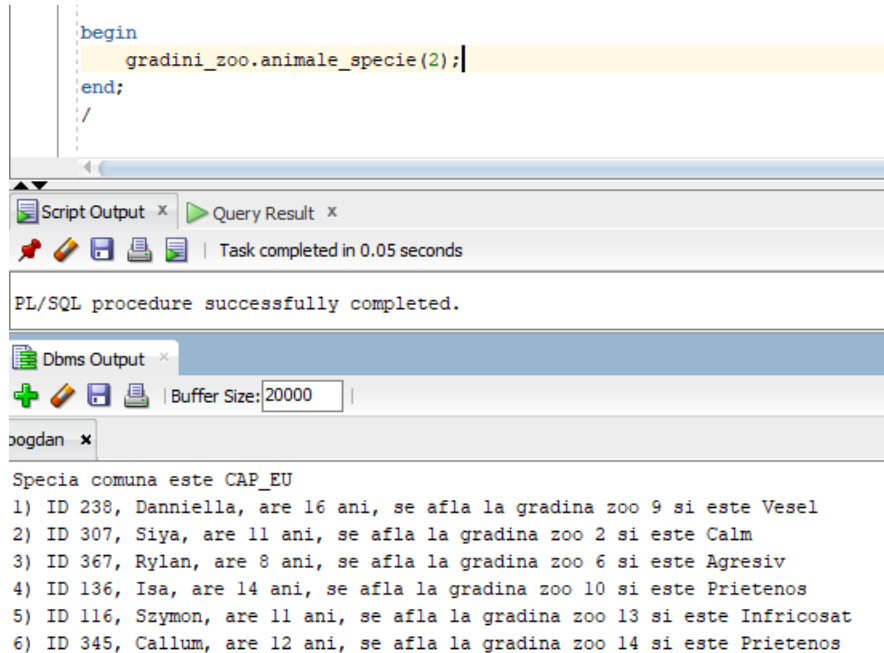
dbms_output.Put_line('Specia comuna este '
                    || sp);

FOR i IN lista_ani.first..lista_ani.last LOOP
    dbms_output.Put_line(nr_ordine
                        || ') ID '
                        || Lista_ani(i).id_animal
                        || ', '
                        || Lista_ani(i).nume
                        || ', are '
                        || Lista_ani(i).varsta
                        || ' ani, se afla la gradina zoo '
                        || Lista_ani(i).id_zoo
                        || ' si este '
                        || Lista_ani(i).temperament);

    nr_ordine := nr_ordine + 1;
END LOOP;
END;
END;
/
```

Package GRADINI\_ZOO compiled

Package Body GRADINI\_ZOO compiled



```
begin
    gradini_zoo.animale_specie(2);
end;
/
```

Script Output x Query Result x

Task completed in 0.05 seconds

PL/SQL procedure successfully completed.

Dbms Output x

Buffer Size: 20000

bogdan x

Specia comuna este CAP\_EU

- 1) ID 238, Danniella, are 16 ani, se afla la gradina zoo 9 si este Vesel
- 2) ID 307, Siya, are 11 ani, se afla la gradina zoo 2 si este Calm
- 3) ID 367, Rylan, are 8 ani, se afla la gradina zoo 6 si este Agresiv
- 4) ID 136, Isa, are 14 ani, se afla la gradina zoo 10 si este Prietenos
- 5) ID 116, Szymon, are 11 ani, se afla la gradina zoo 13 si este Infricosat
- 6) ID 345, Callum, are 12 ani, se afla la gradina zoo 14 si este Prietenos

**14. Definiți un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite (minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri).**

*Codul:*

```
CREATE OR replace PACKAGE conducere_zoo
IS
TYPE specie_animal
IS
    RECORD
    (
        id_specie specie.id_specie%TYPE,
        nume_specie specie.nume_specie%TYPE,
        tip_dieta specie.tip_dieta%TYPE );
TYPE lista_specii
IS
    TABLE OF SPECIE_ANIMAL INDEX BY PLS_INTEGER;
TYPE restrictie_animal
```

```

IS
  RECORD
  (
    nume animale.nume%TYPE,
    data_inceput restrictie.data_inceput%TYPE,
    data_sfarsit restrictie.data_sfarsit%TYPE,
    nume_restrictie tip_restrictie.nume_restrictie%TYPE );
TYPE lista_restrictii
IS
  TABLE OF RESTRICTIE_ANIMAL INDEX BY PLS_INTEGER;
  -- Sa se returneze speciile de care are grija un ingrijitor dat
FUNCTION Angajat_specii(id_angajat angajati.id_ang%TYPE)
  RETURN LISTA_SPECII;
  -- Sa se returneze cantitatea mancata de un animal dat, intr-o zi data
FUNCTION Cantitate(zi DATE,
                   id_animal animale.id_animal%TYPE)
  RETURN NUMBER;
  -- Sa se afiseze restrictiile impuse asupra animalelor dintr-o gradina zoo data
PROCEDURE restrictii_animale(id_z IN gradina_zoologica.id_zoo%TYPE);
  -- Sa se afiseze cate tipuri de hrana provin din tara data
PROCEDURE provine(tara tip_hrana.origine%TYPE);
END;/CREATE OR replace PACKAGE BODY
  conducere_zoo
IS
FUNCTION Angajat_specii(id_angajat angajati.id_ang%TYPE)
  RETURN LISTA_SPECII
IS
  no_zookeeper_found EXCEPTION;
  verificare          NUMBER(1);
  lista LISTA_SPECII;
  aux SPECIE_ANIMAL;
  indice NATURAL := 1;
BEGIN
  SELECT Count(*)
  INTO   verificare
  FROM   angajati
  WHERE  id_ang=id_angajat
  AND    id_job = 'INGRJ';

  IF verificare = 0 THEN
    RAISE no_zookeeper_found;
  END IF;
  FOR elem IN
    (
      SELECT DISTINCT s.id_specie,
                     s.nume_specie,
                     s.tip_dieta
      FROM           hraneste h,
                     specie s,
                     animale a
      WHERE          h.id_ang=783
      AND            h.id_animal=a.id_animal
      AND            a.id_specie=s.id_specie)
  LOOP
    aux.id_specie := elem.id_specie;
    aux.nume_specie := elem.nume_specie;
    aux.tip_dieta := elem.tip_dieta;
    Lista(indice) := aux;
  
```



```

        indice := indice + 1;
    END LOOP;
    RETURN lista;
EXCEPTION
WHEN no_zookeeper_found THEN
    Raise_application_error(-20001, 'Angajatul nu este ingrijitor.');
```

END;

```

FUNCTION Cantitate(zi DATE,
                  id_ani animale.id_animal%TYPE)
    RETURN NUMBER
IS
    cant          NUMBER;
    verificare     NUMBER;
    no_animal_found EXCEPTION;
    no_feeding_found EXCEPTION;
BEGIN
    SELECT Count(*)
    INTO   cant
    FROM   animale
    WHERE  id_animal=id_ani;

    IF cant = 0 THEN
        RAISE no_animal_found;
    END IF;
    SELECT SUM(cantitate),
           Count(cantitate)
    INTO   cant,
           verificare
    FROM   hraneste
    WHERE  data_hranire = zi
    AND    id_animal=id_ani;

    IF verificare = 0 THEN
        RAISE no_feeding_found;
    END IF;
    RETURN cant;
EXCEPTION
WHEN no_feeding_found THEN
    Raise_application_error(-20001, 'Animalul nu a fost hranit in ziua respectiva');
```

WHEN no\_animal\_found THEN

```

    Raise_application_error(-20002, 'Nu exista un animal cu acest id');
```

END;

```

PROCEDURE Restrictii_animale(id_z IN gradina_zoologica.id_zoo%TYPE)
IS
    lista LISTA_RESTRICTII;
    no_zoo_found EXCEPTION;
    verificare NUMBER(1);
BEGIN
    SELECT Count(*)
    INTO   verificare
    FROM   gradina_zoologica
    WHERE  id_zoo=id_z;

    IF verificare = 0 THEN
        RAISE no_zoo_found;
    END IF;
    SELECT  a.nume,
            r.data_inceput,
```

```

        r.data_sfarsit,
        t.num Restrictie bulk collect
INTO lista
FROM Restrictie r,
     animale a,
     tip_restrictie t
WHERE r.id_animal=a.id_animal
AND a.id_zoo=id_z
AND r.tip_restrictie=t.tip_restrictie
ORDER BY a.num;

IF lista.Count() = 0 THEN
    RAISE no_data_found;
END IF;
FOR i IN lista.first..lista.last
LOOP
    dbms_output.Put_line(Lista(i).num
    || ' '
    || Lista(i).num Restrictie
    || ' '
    || Lista(i).data_inceput
    || ' - '
    || Lista(i).data_sfarsit);
END LOOP;
EXCEPTION
WHEN no_zoo_found THEN
    Raise_application_error(-20001, 'Nu exista o gradina zoologica cu acest id.');
```

```

WHEN no_data_found THEN
    Raise_application_error(-20001, 'Nu exista restrictii.');
```

```

END;
PROCEDURE Provine(tara tip_hrana.origine%TYPE)
IS
    cant NUMBER(3);
    no_country_found EXCEPTION;
BEGIN
    SELECT Count(*)
    INTO cant
    FROM tip_hrana
    WHERE Lower(origine)=Lower(tara);

    IF cant = 0 THEN
        RAISE no_country_found;
    ELSE
        dbms_output.Put_line('Din '
        || tara
        || ' provin '
        || cant
        || ' tipuri de hrana.');
```

```

    END IF;
EXCEPTION
WHEN no_country_found THEN
    Raise_application_error(-20001, 'Nu exista hrana din aceasta tara');
```

```

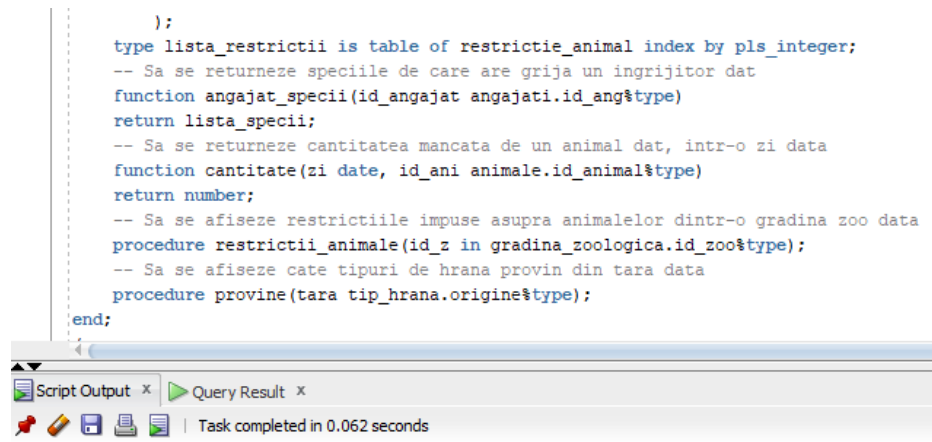
END;
END;
DECLARE
    lista conducere_zoo.lista_specii;
    cantitate NUMBER;
BEGIN
    -- lista := conducere_zoo.angajat_specii(788);

```

```

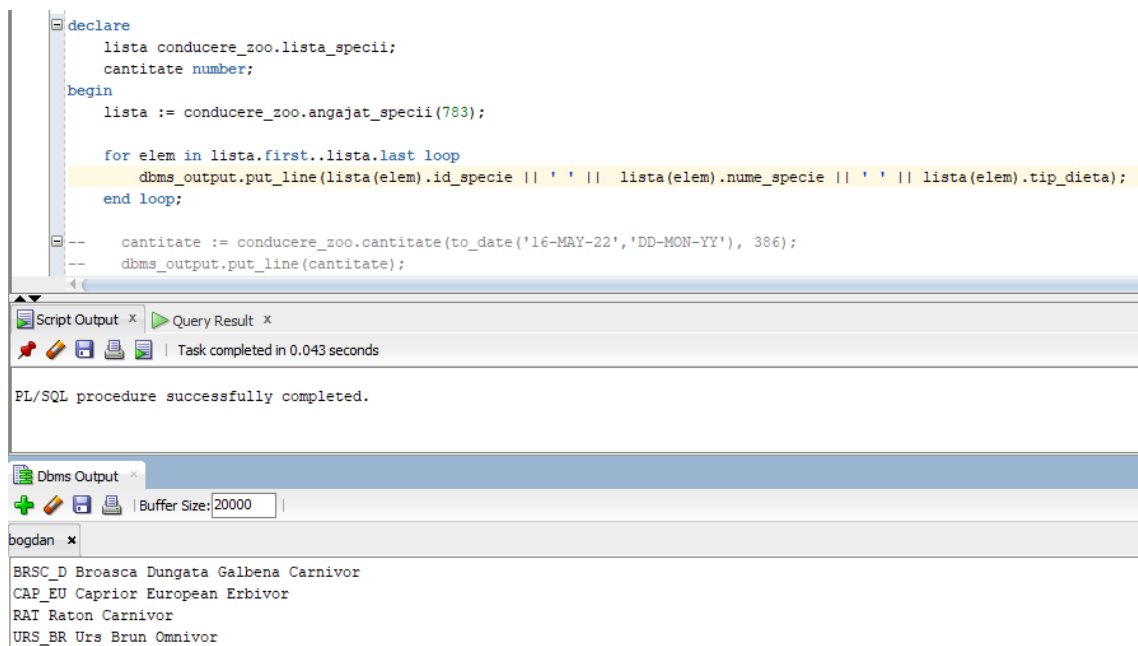
--
--   for elem in lista.first..lista.last loop
--
--       dbms_output.put_line(lista(elem).id_specie || ' ' || lista(elem).nume_spec
ie || ' ' || lista(elem).tip_dieta);
--   end loop;
--   cantitate := conducere_zoo.cantitate(to_date('16-MAY-22','DD-MON-
YY'), 386);
--   dbms_output.put_line('Cantitate: ' || cantitate);
--   conducere_zoo.restrictii_animale(16);
--   conducere_zoo.provine('Chile');
END;
/

```



Package CONDUCTERE\_ZOO compiled

Package Body CONDUCTERE\_ZOO compiled



```
declare
    lista conducere_zoo.lista_specii;
    cantitate number;
begin
    lista := conducere_zoo.angajat_specii(788);

    for elem in lista.first..lista.last loop
        dbms_output.put_line(lista(elem).id_specie || ' ' || lista(elem).nume_specie || ' ' || lista(elem).tip_dieta);
    end loop;

    -- cantitate := conducere_zoo.cantitate(to_date('16-MAY-22','DD-MON-YY'), 386);
    -- dbms_output.put_line(cantitate);
end;
```

Script Output x Query Result x

Task completed in 0.051 seconds

end;  
Error report -  
ORA-20001: Angajatul nu este ingrijitor.  
ORA-06512: at "C##USERBOGDAN.CONDUCERE\_ZOO", line 36  
ORA-06512: at line 5

```
-- dbms_output.put_line(lista(elem).id_specie || ' ' || lista(elem).nume_
-- end loop;

cantitate := conducere_zoo.cantitate(to_date('16-MAY-22','DD-MON-YY'), 386);
dbms_output.put_line('Cantitate: ' || cantitate);

-- conducere_zoo.restrictii_animale(1);
```

Script Output x Query Result x

Task completed in 0.047 seconds

PL/SQL procedure successfully completed.

Dbms Output x

Buffer Size: 20000

bogdan x

Cantitate: 6

```

cantitate := conducere_zoo.cantitate(to_date('17-MAY-22','DD-MON-YY'), 386);
dbms_output.put_line('Cantitate: ' || cantitate);

```

Script Output x Query Result x  
Task completed in 0.053 seconds

```
-- end loop;
```

```

cantitate := conducere_zoo.cantitate(to_date('17-MAY-22','DD-MON-YY'), 386);
dbms_output.put_line('Cantitate: ' || cantitate);

```

```
-- conducere_zoo.restrictii_animale(1);
```

end;

Error report -

ORA-20001: Animalul nu a fost hranit in ziua respectiva

ORA-06512: at "C##USERBOGDAN.CONDUCERE\_ZOO", line 68

ORA-06512: at line 11

```

cantitate := conducere_zoo.cantitate(to_date('16-MAY-22','DD-MON-YY'), 1152);
dbms_output.put_line('Cantitate: ' || cantitate);

```

```
-- conducere_zoo.restrictii_animale(1);
```

end;

```
/
select * from angajati where id_ang=788;
```

Script Output x Query Result x  
Task completed in 0.039 seconds

```
-- end loop;
```

```

cantitate := conducere_zoo.cantitate(to_date('16-MAY-22','DD-MON-YY'), 1152);
dbms_output.put_line('Cantitate: ' || cantitate);

```

```
-- conducere_zoo.restrictii_animale(1);
```

end;

Error report -

ORA-20002: Nu exista un animal cu acest id

ORA-06512: at "C##USERBOGDAN.CONDUCERE\_ZOO", line 70

ORA-06512: at line 11

```
conducere_zoo.restrictii_animale(1);
```

```

-- conducere_zoo.provine('Germania');
end;

```

Script Output x Query Result x  
Task completed in 0.046 seconds

PL/SQL procedure successfully completed.

Dbms Output x  
Buffer Size: 20000  
bogdan x

Jamaal	Hrana	Moale	03-JUN-06	22-AUG-07
Matilda	Izolare		26-AUG-12	19-SEP-15
Orion	Interzicere	Cuplare	15-NOV-05	17-DEC-06
Orion	Interzicere	Cuplare	03-MAR-21	06-JUL-23
Sonia	Evitare	Rosie	07-SEP-08	18-SEP-10

```
conducere_zoo.restrictii_animale(15);

-- conducere_zoo.provine('Germania');

Script Output x Query Result x
Task completed in 0.044 seconds

-- cantitate := conducere_zoo.cantitate(to_date('16-MAY-22','DD-MON-YY'), 386);
-- dbms_output.put_line('Cantitate: ' || cantitate);

conducere_zoo.restrictii_animale(15);

-- conducere_zoo.provine('Germania');
end;
Error report -
ORA-20001: Nu exista restrictii.
ORA-06512: at "C##USERBOGDAN.CONDUCERE_ZOO", line 107
ORA-06512: at line 14
```

```
conducere_zoo.restrictii_animale(16);

-- conducere_zoo.provine('Germania');
end;
/

Script Output x Query Result x
Task completed in 0.063 seconds

-- cantitate := conducere_zoo.cantitate(to_date('16-MAY-22','DD-MON-YY'), 386);
-- dbms_output.put_line('Cantitate: ' || cantitate);

conducere_zoo.restrictii_animale(16);

-- conducere_zoo.provine('Germania');
end;
Error report -
ORA-20001: Nu exista o gradina zoologica cu acest id.
ORA-06512: at "C##USERBOGDAN.CONDUCERE_ZOO", line 105
ORA-06512: at line 14
```

```
conducere_zoo.provine('Germania');

Script Output x Query Result x
Task completed in 0.043 seconds

PL/SQL procedure successfully completed.

Dbms Output x
Buffer Size: 20000

bogdan x
Din Germania provin 1 tipuri de hrana.
```

```
    conducere_zoo.provine('Chile');  
end;  
/
```

Script Output x

Query Result x

Task completed in 0.051 seconds

```
--    conducere_zoo.restrictii_animale(16);
```

```
    conducere_zoo.provine('Chile');  
end;
```

Error report -

ORA-20001: Nu exista hrana din aceasta tara

ORA-06512: at "C##USERBOGDAN.CONDUCERE\_ZOO", line 127

ORA-06512: at line 16