

PROIECT

-GESTIUNEA GRADINILOR ZOOLOGICE-

Putinelu Andrei Bogdan

Grupa 133

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.

Modelul gestionează activitatea mai multor grădini zoologice. Se gestionează mai multe aspecte ale grădinilor zoologice precum:

- Angajații grădinilor zoologice și ce poziții ocupă
- Produsele vândute (de exemplu bilete pentru spectacole, bilete de intrare, bilete cu reduceri speciale, consumabile precum apă, popcorn etc.)
- Hrana pentru animale deținute de grădini
- Animalele ce aparțin grădinilor zoologice, detalii despre acestea, precum și eventuale restricții în urma unor evenimente neprevăzute (de exemplu după o vizită medicală un animal trebuie să stea la repaus 2 zile)
- Partenerii cu care grădinile zoologice fac tranzacții, acestea pot cumpăra servicii (de exemplu hârtie pentru bilete), acestea pot vinde servicii (de exemplu un spectacol cu animale privat) și 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).

2. Prezentarea constrângerilor (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, iar un partener va lucra cu minim o gradina zoologica.
- 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.

3. Descrierea entităților, incluzând precizarea cheii primare.

Entitatea GRADINA_ZOOLOGICA cu cheia primara id_zoo# se refera la gradinile zoologice gestionate.

Entitatea PRODUSE cu cheia primara id_produs# se refera la produsele vandute de catre gradinile zoologice, cum ar fii bilete, suvenire sau consumabile.

Entitatea PARTENER cu cheia primara id_part# se refera la partenerii gradinilor zoologice.

Entitatea SERVICIU cu cheia primara id_serviciu# reprezinta serviciile ce au loc intre gradinile zoologice si partenerii acestora, cum ar fii un serviciu de cumparare hartie pentru bilete.

Entitatea ANGAJATI cu cheia primara id_ang# reprezinta angajatii gradinilor zoologice.

Entitatea JOBS cu cheia primara id_job# reprezinta rolul unui angajat intr-o gradina zoologica.

Entitatea ANIMALE cu cheia primara id_animal# se refera la animalele din gradina zoologica.

Entitatea SPECIE cu cheia primara id_specie# se refera la specia animalelor din gradinile zoologice.

Entitatea HRANA cu cheia primara id_hrana# reprezinta hrana pentru animale pe care o detine o gradina zoologica.

Entitatea TIP_HRANA cu cheia primara id_tip_hrana# reprezinta tipurile de hrana pentru animale ce se gasesc in gradinile zoologice.

Entitatea RESTRICTIE cu cheia primara id_restrictie# reprezinta restrictiile la care sunt supuse animalele, spre exemplu, in urma unei accidentari un tigru trebuie sa stea la repaus timp de 2 zile pentru a se recupera.

Entitatea TIP_RESTRICTIE cu cheia primara tip_restrictie# se refera la tipul de restrictii.

4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

Relatia beneficiaza este o relatie de tip 3, stabilita intre entitatile PARTENER, GRADINA_ZOOLOGICA si SERVICIU. Aceasta gestioneaza activitatile dintre gradinile zoologice si partenerii acestora. Relatia are cardinalitatea maxima MANY-MANY-MANY si cardinalitatea minima 1-1-1.

Relatia vand este o relatie de tip 2, stabilita intre entitatile PRODUSE si GRADINA_ZOOLOGICA. Aceasta leaga produsele de gradina zoologica care le vinde si are cardinalitatea maxima MANY-MANY, iar cardinalitatea minima 1-1.

Relatia lucreaza_la este o relatie de tip 2, stabilita intre GRADINA_ZOOLOGICA si ANGAJATI. Aceasta leaga angajatii de gradina zoologica la care lucreaza. Cardinalitatea maxima este 1-MANY, iar cardinalitatea minima este 1-1.

Relatia angajat_ca este o relatie de tip 2, stabilita intre ANGAJATI si JOBS. Aceasta leaga angajatii de rolul lor in gradina zoologica. Are cardinalitatea maxima MANY-1 si cardinalitatea minima 1-1.

Relatia apartine este o relatie de tip 2, stabilita intre GRADINA_ZOOLOGICA si ANIMALE. Aceasta leaga animalele de gradina zoologica de care apartin. Cardinalitatea maxima este 1-MANY, iar cardinalitatea minima este 1-1.

Relatia detine este o relatie de tip 2, stabilita intre GRADINA_ZOOLOGICA si HRANA. Aceasta leaga hrana de gradina zoologica care o detine. Cardinalitatea maxima este 1-MANY, iar cardinalitatea minima este 1-1.

Relatia se_incadreaza este o relatie de tip 2, stabilita intre SPECIE si ANIMALE, care leaga animalele de specia din care fac parte. Cardinalitatea maxima este 1-MANY, iar cardinalitatea minima este 1-1.

Relatia are este o relatie de tip 2, stabilita intre RESTRICTIE si ANIMALE, leaga animalul de restrictiile pe care trebuie sa le urmeze. Cardinalitatea maxima este MANY-1, iar cardinalitatea minima este 0-1.

Relatia contine este o relatie de tip 2, stabilita intre RESTRICTIE si TIP_RESTRICTIE. Aceasta leaga tipul de restrictie de restrictia impusa. Cardinalitatea maxima este MANY-1, iar cardinalitatea minima este 1-1.

Relatia este este o relatie de tip 2, stabilita intre TIP_HRANA si HRANA, care leaga tipul hranei de hrana detinuta. Cardinalitatea maxima este de 1-MANY, iar cardinalitatea minima este 1-1.

Relatia hraneste este o relatie de tip 3, stabilita intre HRANA, ANGAJATI si ANIMALE. Aceasta leaga angajatii de animalele pe care le-au hranit si de hrana cu care le-au dat-o. Cardinalitatea maxima este MANY-MANY-MANY, iar cardinalitatea minima este 1-1-1.

5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.

In entitatea PARTENER:

- id_part# - este cheia primara, va fi de tipul NUMBER(4) si va avea constrangerea de cheie primara. (de ex. 10)
- nume_partener – se refera la numele firmei partenerilor, va fi de tipul VARCHAR2(30) si va avea constrangerea de not null. (de ex. SUFIPAPER)
- nume_contact – se refera la numele de familie al persoanei de contact al firmei pe care o reprezinta, va avea tipul VARCHAR2(25) si va avea constrangere de not null. (de ex. Smith)
- email_contact – se refera la email-ul persoanei de contact, are tipul VARCHAR2(30) si va avea constrangerile de unique si not null. (de ex. lester.smith@gmail.com)
- nr_tel_contact – se refera la numarul de telefon al persoanei de contact, are tipul VARCHAR2(15) si are constrangerea de unique. (de ex. 0777755552)
- oras_sediu – se refera la orasul unde se afla sediul partenerului si unde se afla persoana de contact, are tipul VARCHAR2(25). (de ex. New York)

In entitatea SERVICIU:

- id_serviciu# - este cheia primara, va fi de tipul NUMBER(4), si va avea constrangerea de cheie primara. (de ex. 120)
- nume_serviciu – se refera la obiectele sau serviciile tranzactionate, va avea tipul VARCHAR2(60) si va avea constrangerea de not null. (de ex. Morcovi)
- tip_serviciu – se refera la unul din cele 3 tipuri de servicii admise, adica Donatie, Vanzare sau Achizitionare (Donatie – gradina zoologica a primit o donatie, Vanzare – gradina zoologica a facut vanzarea, Achizitionare – gradina zoologica a achizitionat), este de tipul VARCHAR2(15) si va avea o constrangere de not null si o

constrangere de check care va verifica ca, valoarea sa fie una din cele 3. (de ex. Achizitionare)

In relatia beneficiaza:

- cantitate – este o mica descriere pentru serviciul facut (cu precizarea ca, pentru hrana se vor introduce numai cantitati masurate in kilograme), va fi de tipul VARCHAR2(30), si va avea constrangerea not null. (de ex. pentru exemplul anterior cu Achizitionare Morcovi, putem avea : '20 kilograme', insa pentru Achizitionare Hartie putem avea : '5000 bucati')

In entitatea GRADINA_ZOOLOGICA:

- id_zoo# - este cheia primara, are tipul NUMBER(4) si are constrangerea de cheie primara. (de ex. 1)
- locatie – se refera la orasul unde este localizata gradina zoologica, are tipul VARCHAR2(30) si are constrangerea de not null. (de ex. Sydney)
- dimensiune -se refera la dimensiunea gradinii zoologice masurata in m^2 , este de tipul NUMBER(4). (de ex. 425)

In entitatea PRODUSE:

- id_produs# - este cheia primara, are tipul NUMBER(4) si are constrangerea de cheie primara. (de ex. 20)
- nume_produs – se refera la numele produsului pe care il poate vinde o gradina zoologica, este de tip VARCHAR2(30), si are constrangerea de not null. (de ex. Bilet Elev)
- cerinta_produs – se refera la cerinta care trebuie indeplinita pentru a achizitiona produsul (poate sa nu existe nicio cerinta), are tipul VARCHAR2(30). (de ex. Carnet Elev)
- valabilitate_zile – se refera la numarul de zile in care produsul va fi valabil, va fi de tipul NUMBER(4), si va avea o constrangere de not null si un check sa fie mai mare ca 0. (de ex. 1)

In relatia vand:

- cost – se refera la costul in euro al produselor vandute de gradina zoologica care poate sa difere de la o gradina la alta, este de tip

NUMBER(4) si are constrangere de not null si un check sa fie mai mare ca 0. (de ex. 5)

In entitatea ANGAJATI:

- id_ang# - este cheia primara, este de tipul NUMBER(4) si are constrangere de cheie primara. (de ex. 300)
- nume – se refera la numele de familie al angajatului, are tipul VARCHAR2(30) si constrangerea de not null. (de ex. Jones)
- prenume – se refera la prenumele angajatului, are tipul VARCHAR2(30). (de ex. Luke)
- salariu – se refera la salariul in euro al angajatului, are tipul NUMBER(8) si are constrangerile not null si check sa fie mai mare ca 0. (de ex. 5000)
- id_zoo - se refera la gradina zoologica din care angajatul face parte, este de tipul NUMBER(4) si are constrangere de not null si de foreign key la entitatea GRADINA_ZOOLOGICA. (de ex. 120)
- id_job – face referire la rolul angajatului, are tipul VARCHAR2(20) si are constrangere de not null si de foreign key la entitatea JOBS. (de ex. VANZ_PROD)
- nr_tel – se refera la numarul de telefon al angajatului, are tipul VARCHAR2(15) si are constrangerea de unique. (de ex. 0778888889)
- email - se refera la email-ul angajatului, are tipul VARCHAR2(30) si va avea constrangerile de unique si not null. (de ex. luke.jones@gmail.com)
- data_ang – se refera la data angajarii, are tipul DATE si are ca valoare implicita ziua in care va fi inregistrat in baza de date (sysdate). (de ex. 17-JUN-97)

In entitatea JOBS:

- id_job# - este cheia primara, are tipul VARCHAR2(20) si are constrangere de cheie primara. (de ex. VANZ_PROD)
- titlu_job – se refera la rolul angajatului, are tipul VARCHAR2(35) si are constrangere de not null. (de ex. VANZATOR PRODUSE)

- salariu_min – se refera la salariul minim in euro pe care il poate castiga un angajat cu rolul respectiv, are tipul NUMBER(8) si are constrangere not null si check sa fie mai mare ca 0. (de ex. 1000)
- salariu_max – se refera la salariul maxim in euro pe care il poate castiga un angajat cu rolul respectiv, are tipul NUMBER(8) si are constrangere not null si check sa fie mai mare ca 0 si sa fie mai mare ca salariu_min. (de ex. 5000)

In entitatea ANIMALE:

- id_animal# - este cheia primara, este de tipul NUMBER(4) si are constrangere de cheie primara. (de ex. 1)
- nume – se refera la numele animalului, are tipul VARCHAR2(25) si are valoarea implicita 'No Name', in caz ca animalul nu a primit inca un nume. (de ex. Rocky)
- varsta – se refera la varsta animalului in ani, are tipul NUMBER(4) si are constrangerea not null si check sa fie mai mare sau egal ca 0, in caz ca nu a implinit animalul 1 an. (de ex. 5)
- id_zoo - se refera la gradina zoologica la care animalul apartine, este de tipul NUMBER(4) si are constrangere de not null si de foreign key la entitatea GRADINA_ZOOLOGICA. (de ex. 120)
- id_specie – se refera la specia animalului, are tipul VARCHAR2(25) si are constrangerea not null si de foreign key la entitatea SPECIE. (de ex. TIG_SIB)
- temperament – se refera la temperamental animalului, are tipul VARCHAR2(60). (de ex. Prietenos)
- varsta_luni – se refera la lunile din varsta animalului care nu intregesc un an, are tipul NUMBER(4) si are constrangerea de not null si check sa fie mai mare sau egal cu 0 si mai mic strict ca 12. (de ex. 6)

In entitatea SPECIE:

- id_specie# - este cheia primara, are tipul VARCHAR2(25) si are constrangerea de cheie primara. (de ex. TIG_SIB)
- tip_dieta – se refera la dieta animalului, carnivor, erbivor sau omnivor, are tipul VARCHAR2(10) si are constrangerea de not null si de check sa fie o valoare dintre cele 3. (de ex. Carnivor)

- Colorit – se refera la coloritul animalului, se poate oferi o scurta descriere, are tipul VARCHAR2(60). (de ex. Portocaliu cu dungi negre)
- tip_animal – in functie de activitatea animalului, acesta poate fi nocturn, diurn sau amandoua (crepuscular), are tip VARCHAR2(20), are constrangere de not null si check sa fie una dintre cele 3 valori. (de ex. Crepuscular)
- varsta_medie – se refera la durata medie, in ani, de viata a specie, are tip NUMBER(4), are constrangere de not null. (de ex. 9)
- clasificare – se refera la clasificarea specie, are tipul VARCHAR2(20), are constrangere de not null. (de ex. Mamifer)
- nume_specie – se refera la numele specie, are tipul VARCHAR2(35) si are constrangerea de not null. (de ex. TIGRU SIBERIAN)

In entitatea RESTRICTIE:

- id_restrictie# - este cheia primara, este de tipul NUMBER(4) si are constrangere de cheie primara. (de ex. 10)
- data_inceput – se refera la data inceperii restrictiei, are tipul DATE, are valoarea default sysdate (ziua in care a fost introdusa in baza de date) si are constrangere de not null. (de ex. 21-SEP-89)
- data_sfarsit – se refera la data terminarii restrictiei, are tipul DATE, are valoarea default sysdate (ziua in care a fost introdusa in baza de date) si are constrangere de not null si check daca este mai mare sau egala ca data_inceput. (de ex. 22-SEP-89)
- id_animal – se refera la animalul care primeste restrictia , este de tipul NUMBER(4) si are constrangere de not null si de foreign key la entitatea ANIMALE. (de ex. 1)
- tip_restrictie – se refera la tipul restrictiei animalului, are tipul VARCHAR2(20) si are constrangere de not null si de foreign key la entitatea TIP_RESTRICTIE. (de ex. REP_US)

In entitatea TIP_RESTRICTIE:

- tip_restrictie# – este cheia primara, are tipul VARCHAR2(20) si are constrangere de cheie primara. (de ex. REP_US)

- nume_restrictie – se refera la nume restrictiei, are tipul VARCHAR2(30) si are constrangere not null. (de ex. REPAUS USOR)
- descriere_restrictie – descrie ce presupune restrictia, are tipul VARCHAR2(80) si are constrangere de not null. (de ex. Repaus usor pentru oboseala)

In entitatea HRANA:

- id_hrana# - este cheia primara, este de tipul NUMBER(4) si are constrangere de cheie primara. (de ex. 55).
- cantitate – se refera la cantitatea, in kilograme, de hrana detinuta, are tipul NUMBER(6) si are constraint de not null si check sa fie mai mare ca 0. (de ex. 200)
- id_zoo – se refera la gradina zoologica care detine hrana, este de tipul NUMBER(4) si are constrangere de not null si foreign key la entitatea GRADINA_ZOOLOGICA. (de ex. 120)
- id_tip_hrana – se refera la tipul de hrana, are tipul VARCHAR2(20) si are constrangerea de not null si foreign key la entitatea TIP_HRANA. (de ex. MRCV)

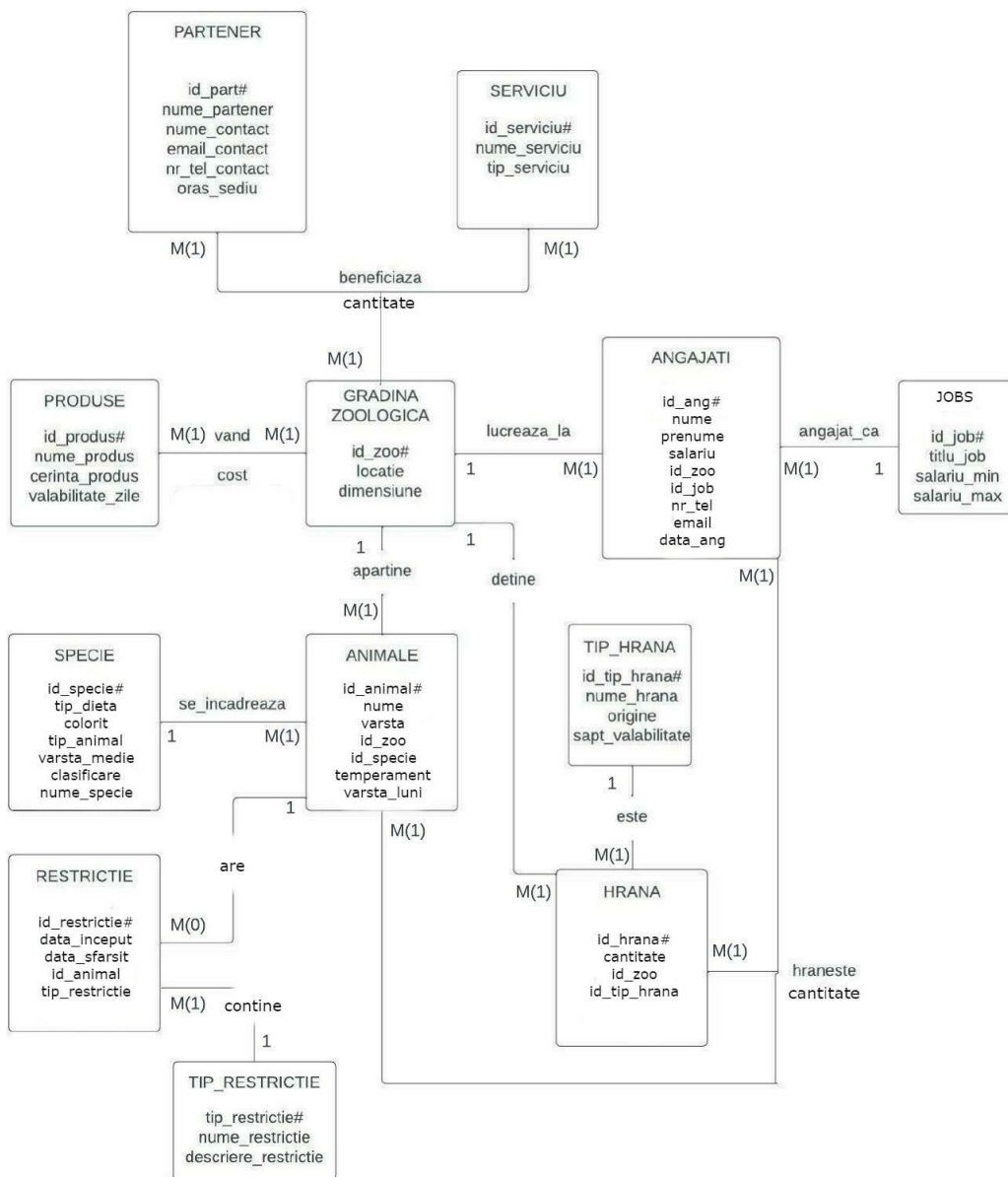
In entitatea TIP_HRANA:

- id_tip_hrana# – este cheia primara, are tipul VARCHAR2(20) si are constrangerea de cheie primara. (de ex. MRCV)
- nume_hrana – reprezinta numele hranii, are tipul VARCHAR2(30) si are constrangerea de not null. (de ex. MORCOVI)
- origine – reprezinta tara din care a fost exportata hrana, are tip VARCHAR2(30) si are constrangere de not null. (de ex. CHINA)
- sapt_valabilitate – reprezinta numarul de saptamani pentru care hrana nu expira si este buna de mancat, are tipul NUMBER(4) si are constrangerea de not null si check sa fie mai mare ca 0. (de ex. 2)

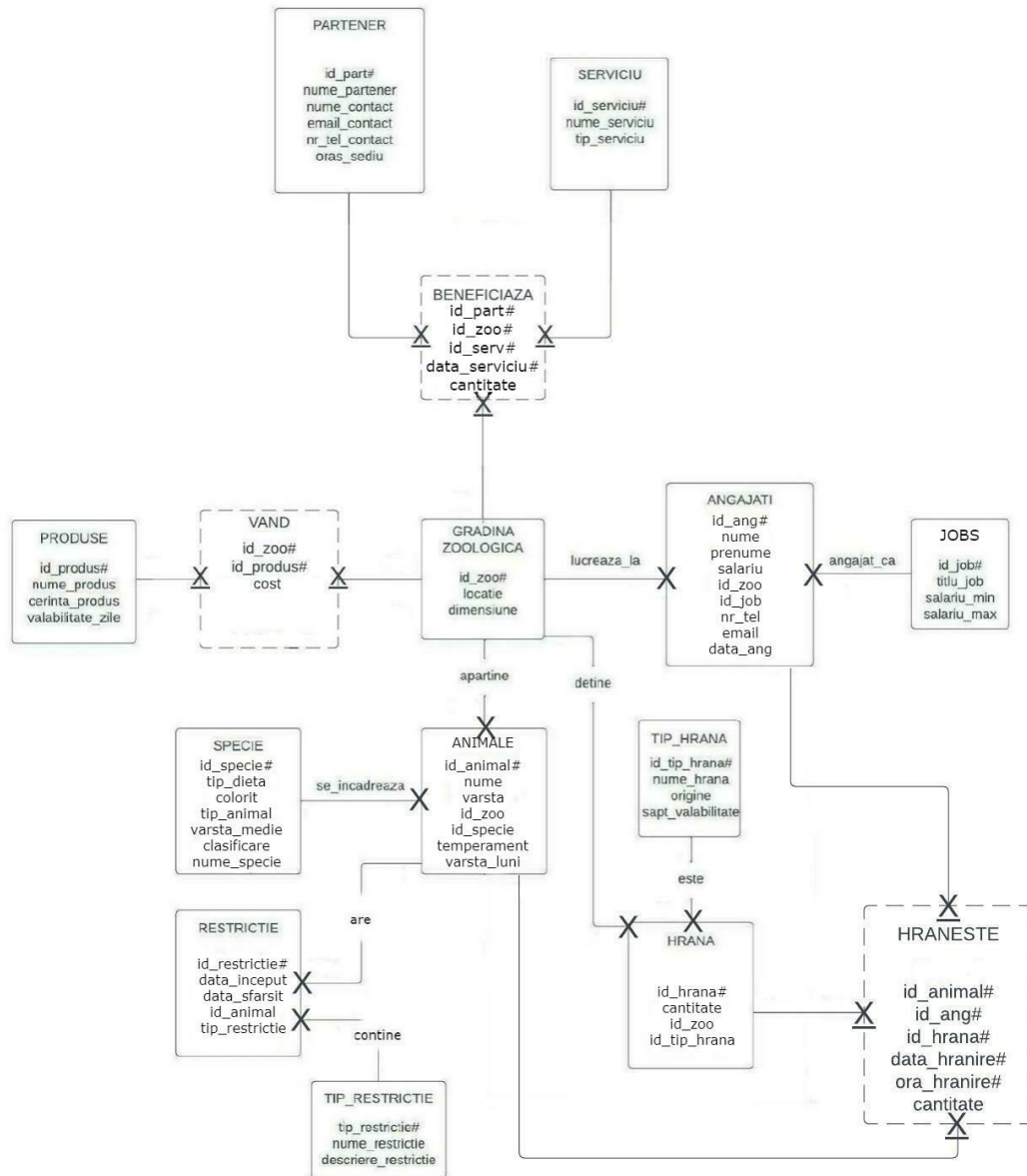
In relatia hraneste:

- cantitate – se refera la numarul de kilograme de hrana oferita animalului, are tipul NUMBER(4) si are constrangerea de not null si check sa fie mai mare ca 0. (de ex. 10)

6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.



8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

PARTENER(id_part#, nume_partener, nume_contact, email_contact, nr_tel_contact, oras_sediu);

SERVICIU(id_serviciu#, nume_serviciu, tip_serviciu);

BENEFICIAZA(id_part#, id_zoo#, id_serv#, data_serviciu#, cantitate);

GRADINA_ZOOLOGICA(id_zoo#, locatie, dimensiune);

VAND(id_zoo#, id_produs#, cost);

PRODUSE(id_produs#, nume_produs, cerinta_produs, valabilitate_zile);

ANGAJATI(id_ang#, nume, prenume, salariu, id_zoo, id_job, nr_tel, email, data_ang);

JOBS(id_job#, titlu_job, salariu_min, salariu_max);

ANIMALE(id_animal#, nume, varsta, id_zoo, id_specie, temperament, varsta_luni);

SPECIE(id_specie#, tip_dieta, colorit, tip_animal, varsta_medie, clasificare, nume_specie);

RESTRICTIE(id_restrictie#, data_inceput, data_sfarsit, id_animal, tip_restrictie);

TIP_RESTRICTIE(tip_restrictie#, nume_restrictie, descriere_restrictie);

HRANA(id_hrana#, cantitate, id_zoo, id_tip_hrana);

TIP_HRANA(id_tip_hrana#, nume_hrana, origine, sapt_valabilitate);

HRANESTE(id_animal#, id_ang#, id_hrana#, data_hranire#, ora_hranire#, cantitate);

9. Realizarea normalizării până la forma normală 3 (FN1-FN3).

Exemplele urmatoare se vor raporta la diagrama conceptuala de mai sus.

Un exemplu de non-FN1 a diagramei conceptuale poate fi urmatorul:

In entitatea PARTENER avem attributele:

- id_part# - cheia primara
- nume_partener – numele firmei
- informatii_contact – informatiile persoanei de contact

PARTENER

id_part#	nume_partener	informatii_contact
1	CORTOS	Smith, j.smith@gmail.com, 0761111112, California
2	MIRES	Frost, andrew.frost@gmail.com, 0789568887, Sydney
3	PAPOO	Crone, ashleycrone@gmail.com, 0772236564, Madrid

Nu se afla in FN1, deoarece atributul informatii_contact nu este atomic.

Astfel o sa impartim atributul informatii_contact in mai multe attribute atomice.

Se realizeaza urmatoarea transformare pentru a trece in FN1:

PARTENER

id_part#	nume_partener	nume_contact	email_contact	nr_tel_contact	oras_sediu
1	CORTOS	Smith	j.smith@gmail.com	0761111112	California
2	MIRES	Frost	andrew.frost@gmail.com	0789568887	Sydney
3	PAPOO	Crone	ashleycrone@gmail.com	0772236564	Madrid

Un exemplu de non-FN2 a diagramei conceptuale poate fi urmatorul:

Entitatea PRODUSE nu exista. In entitatea VAND avem attributele:

- id_zoo# - face parte din cheia primara
- id_produș# - face parte din cheia primara
- cost – costul produsului
- nume_produș – numele produsului
- cerinta_produș – conditia pentru a vinde produsul
- valabilitate_zile – numarul de zile in care produsul va fii valabil, de la cumparare

VAND

id_zoo#	id_produș#	nume_produș	cerinta_produș	valabilitate_zile	cost
1	20	Bilet Elev	Carnet Elev	1	3
2	20	Bilet Elev	Carnet Elev	1	4
3	20	Bilet Elev	Carnet Elev	1	5
1	10	Bilet Student	Legitimatie Student	2	10

Nu se afla in FN2, deoarece attributele nume_produș, cerinta_produș si valabilitate_zile depind functional de id_produș#.

Vom realiza urmatoarea transformare pentru a trece in FN2:

Vom crea entitatea PRODUSE cu attributele:

- id_produș# - cheia primara
- nume_produș – numele produsului
- cerinta_produș – conditia pentru a vinde produsul
- valabilitate_zile – numarul de zile in care produsul va fii valabil, de la cumparare

Entitatea VAND va ramane cu attributele:

- id_zoo# - face parte din cheia primara
- id_produș# - face parte din cheia primara
- cost – costul produsului

VAND

id_zoo#	id_produs#	cost
1	20	3
2	20	4
3	20	5
1	10	10

PRODUSE

id_produs#	nume_produs	cerinta_produs	valabilitate_zile
20	Bilet Elev	Carnet Elev	1
10	Bilet Student	Legitimatie Student	2

Un exemplu de non-FN3 a diagramei conceptuale poate fi urmatorul:

Entitatea TIP_HRANA nu exista, iar entitatea HRANA are attributele:

- id_hrana# - este cheia primara
- cantitate – cantitatea hranii
- id_zoo – face referire la gradina zoologica care detine hrana
- id_tip_hrana – face referire la tipul de hrana
- nume_hrana – numele hranii
- origine – originea hranii
- sapt_valabilitate – numarul de saptamani in care hrana este comestibila

HRANA

id_hrana#	cantitate	id_zoo	id_tip_hrana	nume_hrana	origine	sapt_valabilitate
22	200	1	MRCV	Morcovi	China	2
33	150	2	MRCV	Morcovi	China	2
44	175	3	MRCV	Morcovi	China	2
55	100	1	CRTF	Cartofi	Italia	4

Nu se afla in FN3, deoarece attributele nume_hrana, origine, sapt_valabilitate depind functional de atributul id_tip_hrana.

Vom realiza urmatoarea transformare pentru a trece in FN3:

Vom crea entitatea TIP_HRANA cu attributele:

- id_tip_hrana# – cheia primara
- nume_hrana – numele hranii
- origine – originea hranii
- sapt_valabilitate – numarul de saptamani in care hrana este comestibila

Entitatea HRANA va ramane cu attributele:

- id_hrana# - este cheia primara
- cantitate – cantitatea hranii
- id_zoo – face referire la gradina zoologica
- id_tip_hrana – face referire la tipul de hrana

HRANA

id_hrana#	cantitate	id_zoo	id_tip_hrana
22	200	1	MRCV
33	150	2	MRCV
44	175	3	MRCV
55	100	1	CRTF

TIP_HRANA

id_tip_hrana#	nume_hrana	origine	sapt_valabilitate
MRCV	Morcovi	China	2
CRTF	Cartofi	Italia	4

**10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în
tabele (punctul 11).**

```
--
10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).
--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;
```

```
--10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).

--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
```

Script Output x Query Result x

Task completed in 0.036 seconds

Sequence SECV_ZOO created.

Sequence SECV_ANIMALE created.

Sequence SECV_PART created.

Sequence SECV_SERV created.

Sequence SECV_PRODUS created.

Sequence SECV_ANGAJAT created.

Sequence SECV_RESTRICTIE created.

11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).

De mentionat este faptul ca o buna parte a comenzilor de insert au fost generate de niste script-uri scrise in python de mine (acestea vor fi incluse la trimiterea proiectului).

```
--11. Crearea tabelelor în SQL si inserarea de date coerente în fiecare dintre acestea
--
(minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).
--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_produs  NUMBER(4) CONSTRAINT fk_produs REFERENCES produse(id_produs) ON
        DELETE
        CASCADE,
    cost       NUMBER(4) CONSTRAINT null_cost NOT NULL CONSTRAINT check_cost
        CHECK(cost>0),
    CONSTRAINT pk_zoo_produs PRIMARY KEY(id_zoo, id_produs)
);

--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)
);

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

```

INSERT INTO produse
(id_produc, nume_produc, cerinta_produc, valabilitate_zile)
VALUES
(secv_produc.nextval, 'Popcorn', NULL, 1);

INSERT INTO produse
(id_produc, nume_produc, cerinta_produc, valabilitate_zile)
VALUES
(secv_produc.nextval, 'Popcorn cu caramel', NULL, 1);

INSERT INTO produse
(id_produc, nume_produc, cerinta_produc, valabilitate_zile)
VALUES
(secv_produc.nextval, 'Apa', NULL, 1);

INSERT INTO produse
(id_produc, nume_produc, cerinta_produc, valabilitate_zile)
VALUES
(secv_produc.nextval, 'Vata de zahar', NULL, 1);

INSERT INTO produse
(id_produc, nume_produc, cerinta_produc, valabilitate_zile)
VALUES
(secv_produc.nextval, 'Inghetata', NULL, 1);

INSERT INTO produse
(id_produc, nume_produc, cerinta_produc, valabilitate_zile)
VALUES
(secv_produc.nextval, 'Popcorn cu ciocolata', NULL, 1);

INSERT INTO produse
(id_produc, nume_produc, cerinta_produc, valabilitate_zile)
VALUES
(secv_produc.nextval, 'Chipsuri', NULL, 1);

INSERT INTO produse
(id_produc, nume_produc, cerinta_produc, valabilitate_zile)
VALUES
(secv_produc.nextval, 'Slushie Coacaze', NULL, 1);

INSERT INTO produse
(id_produc, nume_produc, cerinta_produc, valabilitate_zile)
VALUES
(secv_produc.nextval, 'Slushie Capsuni', NULL, 1);

INSERT INTO produse
(id_produc, nume_produc, cerinta_produc, valabilitate_zile)
VALUES
(secv_produc.nextval, 'Slushie Portocale', NULL, 1);

INSERT INTO produse
(id_produc, nume_produc, cerinta_produc, valabilitate_zile)
VALUES
(secv_produc.nextval, 'Slushie Cirese', NULL, 1);

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

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

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

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

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

```

```

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

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

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

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

INSERT INTO vand
(id_zoo, id_produs, cost)
VALUES
(1, 1090, 28);

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

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

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

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

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

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

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

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

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

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

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

```



```

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

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

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

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

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

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

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

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

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

--inseram inregistrari in ANGAJATI
INSERT INTO angajati
(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
(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
(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
(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
(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
      (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
      (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

```

```

VALUES      (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
VALUES      (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
VALUES      (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
VALUES      (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
VALUES      (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
VALUES      (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
VALUES      (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
VALUES      (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
VALUES      (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
VALUES      (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
VALUES      (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
VALUES      (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
VALUES      (
secv_restrictie.nextval,'20-OCT-2007','21-OCT-2007',165,'INT_CUP_FEM');

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

INSERT INTO restrictie
VALUES      (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
VALUES      (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

```

```

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

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

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

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

INSERT INTO hrana
VALUES      (id_hrana,cantitate,id_zoo,id_tip_hrana)
VALUES      (secv_hrana.nextval,260,1,'POR_CAN');

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

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

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

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

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

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

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

--inseram inregistrari in HRANESTE
INSERT INTO hraneste
VALUES      (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
VALUES      (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
VALUES      (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
VALUES      (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);
```

nume_restrictie varchar2(30) constraint null_nume_restrictie not null,

Script Output x Query Result x
Task completed in 0.029 seconds

Table PARTENER created.

Table SERVICIU created.

Table GRADINA_ZOOLOGICA created.

Table BENEFICIAZA created.

Table PRODUSE created.

Table VAND created.

Table JOBS created.

Table ANGAJATI created.

Table SPECIE created.

Table ANIMALE created.

Table TIP_RESTRICTIE created.

Table RESTRICTIE created.

Table TIP_HRANA created.

Table HRANA created.

Table HRANESTE created.

```
--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)
);
```

Script Output x Query Result x

 SQL | All Rows Fetched: 14 in 0.007 seconds

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)
);
```

Script Output x Query Result x					
SQL Fetched 50 rows in 0.009 seconds					
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_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)
);
```

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_produs number(4) constraint fk_produs references PRODUSE(id_produs) on delete cascade,
    cost number(4) constraint null_cost not null constraint check_cost check(cost>0),
    constraint pk_zoo_produs primary key(id_zoo,id_produs)
);
```

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	Amfibian	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

12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:

- operație *join* pe cel puțin 4 tabele
- filtrare la nivel de linii
- subcereri sincronizate în care intervin cel puțin 3 tabele
- subcereri nesincronizate în care intervin cel puțin 3 tabele
- grupări de date, funcții grup, filtrare la nivel de grupuri
- ordonări
- utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a funcțiilor NVL și DECODE, a cel puțin unei expresii CASE
- utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

```

--Cererea 1
--
Sa se afiseze informatii (numele si tipul serviciului, numele partenerului, id_zoo, data in care
s-a produs serviciul si cantitatea referitoare la serviciu)
--
despre tranzactiile produse intre parteneri si gradinile zoologice in care exista angajati care c
astiga salariul mediu
--
pe jobul lor in gradina zoologica in care ei lucreaza, care au numele de familie format din maxim
5 litere si care au fost angajati in 2020.
--rezultatele vor fi ordonate crescator dupa ziua din luna a datelor cand s-
au produs tranzactiile
--41 rezultate
WITH salariu_mediu
AS (SELECT Round(Avg(salariu)),
        id_zoo,
        id_job
    --se calculeaza salariul mediu pe fiecare job la fiecare gradina zoo
    FROM angajati
    GROUP BY id_zoo,
        id_job)
SELECT p.num_e_partener "Nume Partener",
        g.id_zoo "Gradina Zoo",
        s.num_e_serviciu "Nume Serviciu",
        s.tip_serviciu "Tip Serviciu",
        b.cantitate "Cantitate Serviciu",
        To_char(data_serviciu, 'DD-MM-YYYY') "Data Serviciului"
FROM partener p,
        serviciu s,
        gradina_zoologica g,
        beneficiaza b
WHERE p.id_part = b.id_part
    --se face join pe tabelele partener,serviciu,gradina_zoologica si beneficiaza
    AND s.id_serviciu = b.id_serviciu
    AND g.id_zoo = b.id_zoo
    AND g.id_zoo IN (SELECT id_zoo
        --
        se selecteaza gradinile zoo care au cel putin un angajat care castiga salariul mediu pe jobul sau
        in aceeasi gradina,
        FROM angajati
        --
        care are numele format din maxim 5 litere si care a fost angajat in anul 2020
        WHERE ( salariu, id_zoo, id_job ) IN (SELECT *
            FROM
                salariu_mediu)
        AND Length(num_e) <= 5
        AND To_char(data_ang, 'YYYY') = '2020')
ORDER BY To_char(data_serviciu, 'DD');

```

<pre>--Cererea 1 --Sa se afiseze informatii (numele si tipul serviciului, numele partenerului, id_zoo, data in care s-a produs serviciul si cantitatea referitoare la serviciu) --despre tranzactiile produse intre parteneri si gradinile zoologice in care exista angajati care castiga salariul mediu --pe jobul lor in gradina zoologica in care ei lucreaza, care au numele de familie format din maxim 5 litere si care au fost angajati in 2020. --rezultatele vor fi ordonate crescator dupa ziua din luna a datelor cand s-au produs tranzactiile --41 rezultate with salariu_mediu as (select round(avg(salariu)),id_zoo,id_job --se calculeaza salariul mediu pe fiecare job la fiecare gradina zoo from angajati group by id_zoo,id_job) select p.numa_partener "Nume Partener",g.id_zoo "Gradina Zoo",s.numa_serviciu "Nume Serviciu", s.tip_serviciu "Tip Serviciu", b.cantitate "Cantitate Serviciu",to_char(data_serviciu,'DD-MM-YYYY') "Data Serviciului" from partener p,serviciu s, gradina_zoologica g, beneficiaza b where p.id_part=b.id_part --se face join pe tabelele partener,serviciu,gradina_zoologica si beneficiaza and s.id_serviciu=b.id_serviciu and g.id_zoo=b.id_zoo and g.id_zoo in (select id_zoo --se selecteaza gradinile zoo care au cel putin un angajat care castiga salariul mediu pe jobul sau in aceeasi gradina, from angajati --care are numele format din maxim 5 litere si care a fost angajat in anul 2020 where (salariu,id_zoo,id_job) in (select * from salariu_mediu) and length(nume)<=5 and to_char(data_ang,'YYYY')='2020') order by to_char(data_serviciu,'DD');</pre>					
<pre>Script Output x Query Result x All Rows Fetched: 41 in 0.004 seconds</pre>					
Nume Partener	Gradina Zoo	Nume Serviciu	Tip Serviciu	Cantitate Serviciu	Data Serviciului
27 BLICH	13 Hartie	Achizitionare	3000 bucati	18-07-2005	
28 KINGUS	8 Carne de vita	Achizitionare	150 kilograme	18-01-2006	
29 BLICH	8 Medicamente	Achizitionare	40 flacoane	18-03-2008	
30 BLICH	8 Suplimente	Achizitionare	90 flacoane	20-06-2002	
31 ROWNE	8 Spectacol	Vanzare	2 spectacole	20-05-2013	
32 KINGUS	13 Vitamine	Donatie	80 flacoane	22-05-2013	
33 MEROS	13 Morcovi	Achizitionare	600 kilograme	22-07-2020	
34 SUFIPAPER	13 Morcovi	Achizitionare	500 kilograme	23-11-2016	
35 FARKIN	13 Medicamente	Achizitionare	10 flacoane	23-06-2015	
36 ROWNE	13 Morcovi	Achizitionare	400 kilograme	24-04-2014	
37 SCHLET	13 Spectacol	Vanzare	1 spectacol	25-10-2008	
38 TANG	8 Eveniment Privat	Vanzare	1 eveniment	26-04-2003	
39 TANG	8 Spectacol	Vanzare	4 spectacole	27-10-2012	
40 SCHLET	13 Bani	Donatie	10000 euro	27-06-2000	
41 ROWNE	8 Vitamine	Achizitionare	70 flacoane	28-11-2001	

```
--Cererea 2
--
Se vrea marirea salariilor anumitor angajati. Sa se afiseze informatii despre acesti angajati (id
_zoo,id_ang,id_job, prenumele si numele
--
in aceeasi coloana, nr_tel,email,salariul_vechi, salariul_marit, data_ang), unde salariul_marit e
ste salariul vechi crescut
--
cu 5% daca angajatul lucreaza ca dresor , cu 10% daca angajatul lucreaza ca veterinar, cu 15% dac
a lucreaza ca ingrijitor animale.
--
Daca angajatul nu are numar de telefon se va afisa "Nu exista numar de telefon". Angajatii carora
li se va marii salariul sunt cei
--
care nu au fost angajati in ultima zi din luna, dar au fost angajati in ultimii 5 ani si care luc
reaza la o gradina zoologica
--ce contine cel putin un animal carnivora caruia i-au fost impuse minim 2 restrictii.
--Rezultatele vor fii ordonate crescator dupa id_zoo.
--14 rezultate
SELECT id_zoo "Gradina Zoologica",
       id_ang "Angajat",
       id_job "Job",
       prenume || ' ' || nume "Nume Complet",
       Nvl(nr_tel, 'Nu exista numar de telefon') "Numar Telefon",
       email "Email",
       salariu "Salariu Vechi",
       CASE Lower(id_job) --calculam salariul pentru cele 3 job-uri
```

```

        WHEN 'drsr' THEN salariu * 1.05
        WHEN 'vtrnr' THEN salariu * 1.1
        WHEN 'ingrj' THEN salariu * 1.15
    END
    data_ang                                "Salariu Marit",
    angajati                                "Data Angajare"
FROM
    angajati
WHERE
    Initcap(id_job) IN ( 'Drsr', 'Vtrnr', 'Ingrj' )
    --verificam ca job-ul angajatului sa fie unul din cele 3
    AND id_zoo IN (SELECT DISTINCT id_zoo
        --
        furnizez gradinile zoologice care au cel putin un animal carnivor cu minim 2 restrictii
        FROM
            animale
        WHERE
            id_animal IN (SELECT id_animal
                --verific daca animalul are minim 2 restrictii
                FROM
                    restrictie
                GROUP BY id_animal
                HAVING Count(*) >= 2)
            AND id_specie IN (SELECT id_specie
                --verific ca specia sa fie o specie carnivora
                FROM
                    specie
                WHERE
                    Lower(tip_dieta) = 'carnivor'))
    AND Last_day(data_ang) != data_ang
    AND Months_between(sysdate, data_ang) < 60
    --verific ca angajatul sa fie angajat in ultimii 5 ani
ORDER BY id_zoo;

```

```
--Cererea 2

--Se vrea marirea salariilor anumitor angajati. Sa se afiseze informatii despre acesti angajati (id_zoo,id_ang,id_job, prenumele si numele
--in aceeasi coloana, nr_tel,email,salariul_vechi, salariul_marit, data_ang), unde salariul_marit este salariul vechi crescut
--cu 5% daca angajatul lucreaza ca dresor , cu 10% daca angajatul lucreaza ca veterinar, cu 15% daca lucreaza ca ingrijitor animale.
--Daca angajatul nu are numar de telefon se va afisa "Nu exista numar de telefon". Angajatii carora li se va marii salariul sunt cei
--care nu au fost angajati in ultima zi din luna, dar au fost angajati in ultimii 5 ani si care lucreaza la o gradina zoologica
--ce contine cel putin un animal carnivor caruia i-au fost impuse minim 2 restrictii.
--Rezultatele vor fii ordonate crescator dupa id_zoo.
--14 rezultate

select id_zoo "Gradina Zoologica",id_ang "Angajat",id_job "Job", prenume || ' ' || nume "Nume Complet",
       nvl(nr_tel,'Nu exista numar de telefon') "Numar Telefon", email "Email",
       salariu "Salariu Vechi",case lower(id_job) --calculam salariul pentru cele 3 job-uri
       when 'drsr' then
           salariu*1.05
       when 'vtrnr' then
           salariu*1.1
       when 'ingrj' then
           salariu*1.15
       end "Salariu Marit" ,data_ang "Data Angajare"
from angajati
where initcap(id_job) in ('Drsr','Vtrnr','Ingrj') --verificam ca job-ul angajatului sa fie unul din cele 3
and id_zoo in (select distinct id_zoo --furnizez gradinile zoologice care au cel putin un animal carnivor cu minim 2 restrictii
               from animale
               where id_animal in (select id_animal --verific daca animalul are minim 2 restrictii
```

	Gradina Zoologica	Angajat	Job	Nume Complet	Numar Telefon	Email	Salariu Vechi	Salariu Marit	Data Angajare
1	1	517	INGRJ	Fenne Ng	0726718448	fenne.ng@gmail.com	5000	5750	23-MAY-17
2	1	513	VTRNR	Kapel Hildegard	0744811814	kapel.hildegard@gmail.com	8900	9790	06-OCT-17
3	1	512	VTRNR	Emilia Dwight	0752272512	emilia.dwight@gmail.com	8000	8800	20-OCT-19
4	2	532	VTRNR	Olive Orr	0723142135	olive.orr@gmail.com	8700	9570	07-JAN-18
5	3	556	INGRJ	Eilish Pritchard	0745913643	eilish.pritchard@gmail.com	5100	5865	07-MAR-18
6	5	592	INGRJ	Daryl Bonner	0744752041	daryl.bonner@gmail.com	6000	6900	06-JUL-17
7	5	801	INGRJ	Jun Donnelly	0785666366	jun.donnelly@gmail.com	3100	3565	07-OCT-21
8	5	588	VTRNR	Shaurya Beach	0789304954	shaurya.beach@gmail.com	5400	5940	15-JUL-18
9	5	589	VTRNR	Kyra Marshall	0765018851	kyra.marshall@gmail.com	9900	10890	01-SEP-18
10	8	647	INGRJ	Helena Turnbull	0769482656	helena.turnbull@gmail.com	4700	5405	06-JUL-19
11	8	638	DRSR	Conall Peck	0780671983	conall.peck@gmail.com	5000	5250	05-MAY-20
12	9	656	DRSR	Conah Salter	0734098755	conah.salter@gmail.com	2700	2835	24-MAY-20
13	12	720	VTRNR	Ibraheem Andrew	0772215941	ibraheem.andrew@gmail.com	9200	10120	25-JAN-21
14	12	796	VTRNR	Sia Portillo	0794330452	sia.portillo@gmail.com	8500	9350	13-OCT-17

```
--Cererea 3
--
Sa se afiseze specia animalelor care, in ziua de 16 mai 2022, au fost hranite in afara intervalul
ui orar 08:00 - 14:00
--
cu un tip de hrana ce este valabil 2 saptamani de la cumparare, de catre un angajat care castiga
salariul mediu al job-ului sau,
--adica media dintre salariul minim si maxim al job-
ului. Se vor ordona rezultatele dupa numele speciei.
--2 rezultate
SELECT s.nume_specie "Numele Speciei"
FROM   specie s
WHERE  EXISTS (SELECT ani.id_specie
                FROM   animale ani,
                      hrana hr,
                      angajati ang,
                      hraneste hran,
                      tip_hrana t
                WHERE  s.id_specie = ani.id_specie
                      AND 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 data_hranire = '16-MAY-22'
                      AND t.sapt_valabilitate = 2
```

```

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 id_ang
                    --angajatii care castiga salariul mediu de job
                    FROM   angajati a
                    WHERE  salariu IN(SELECT ( salariu_max
                                              +
                                              salariu_min
                                              ) / 2
                                      FROM   jobs
                                      WHERE  a.id_job = id_job)))

ORDER BY nume_specie;

```

--Cererea 3

--Sa se afiseze specia animalelor 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 salariul mediu al job-ului sau,
 --adica media dintre salariul minim si maxim al job-ului. Se vor ordona rezultatele dupa numele speciei.
 --2 rezultate

```

select s.nume_specie "Numele Speciei"
from specie s
where exists (select ani.id_specie
              from animale ani, hrana hr, angajati ang, hraneste hran, tip_hrana t
              where s.id_specie=ani.id_specie
                  and 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 data_hranire='16-MAY-22'
                  and t.sapt_valabilitate=2
                  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 id_ang --angajatii care castiga salariul mediu de job
                                    from angajati a
                                    where salariu in(select (salariu_max+salariu_min)/2
                                                            from jobs
                                                            where a.id_job=id_job)))
order by nume_specie;

```

Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0.004 seconds

Numele Speciei
1 Anaconda Verde
2 Raton

--Cererea 4

--

Sa se afiseze gradina zoologica, numele animalelor si daca acestea sunt batrane sau tinere (un animal este batran daca are varsta mai mare --decat jumatatea varstei medii) pentru animalele carora nu le-au fost impuse restrictii si care fac parte din gradinile zoologice ale caror suma a preturilor produselor

--

se afla peste media sumelor preturilor produselor din gradinile zoologice. Rezultatele vor fii ordonate descrescator dupa coloana in care apare batran/tanar.

--109 rezultate

```

WITH animale_fara_restr
AS (SELECT id_animal --animalele fara restrictii
FROM   animale
minus
SELECT UNIQUE id_animal
FROM   restrictie)
SELECT a.id_zoo "Gradina Zoologica",

```



```

a.nume      "Nume",
aux.stadiu  "Stadiu"
FROM animale a,
(SELECT Decode(a.varsta, s.varsta_medie / 2, 'Tanar',
               'Batran') stadiu,
      a.id_animal --stadiul tuturor animalelor
FROM animale a,
specie s
WHERE a.id_specie = s.id_specie) aux
WHERE a.id_animal = aux.id_animal
AND a.id_animal IN (SELECT *
                   FROM animale_fara_restr)
AND a.id_zoo IN (SELECT id_zoo
                --gradinile zoo care au suma preturilor mai mare decat media sumelor
FROM vand
HAVING Sum(cost) > (SELECT Avg(suma)
                   FROM (SELECT Sum(cost) suma,
                                id_zoo
                           FROM vand
                           GROUP BY id_zoo))
                   GROUP BY id_zoo)
ORDER BY aux.stadiu DESC;

```

--Cererea 4

--Sa se afiseze gradina zoologica, numele animalelor si daca acestea sunt batrane sau tinere (un animal este batran daca are varsta mai mare decat jumatatea varstei medii) pentru animalele carora nu le-au fost impuse restrictii si care fac parte din gradinile zoologice ale caror suma a preturilor produselor --se afla peste media sumelor preturilor produselor din gradinile zoologice. Rezultatele vor fi ordonate descrescator dupa coloana in care apare batran/tanar.

--109 rezultate

```

with animale_fara_restr as (select id_animal --animalele fara restrictii
                             from animale
                             minus
                             select unique id_animal
                             from restrictie)
select a.id_zoo "Gradina Zoologica",a.nume "Nume",aux.stadiu "Stadiu"
from animale a,(select decode(a.varsta,s.varsta_medie/2,'Tanar','Batran') stadiu,a.id_animal --stadiul tuturor animalelor
                from animale a,specie s
                where a.id_specie=s.id_specie) aux
where a.id_animal=aux.id_animal
and a.id_animal in (select * from animale_fara_restr)
and a.id_zoo in (select id_zoo --gradinile zoo care au suma preturilor mai mare decat media sumelor
                from vand
                having sum(cost) > (select avg(suma)
                                   from (select sum(cost) suma,id_zoo
                                           from vand
                                           group by id_zoo))
                group by id_zoo)
order by aux.stadiu desc;

```

Gradina Zoologica	Nume	Stadiu
96	1 Rojin	Batran
97	3 Kirstie	Batran
98	3 Sandra	Batran
99	9 Haydon	Batran
100	1 Naomi	Batran
101	3 Aditya	Batran
102	10 Salim	Batran
103	9 Clarke	Batran
104	9 Ami	Batran
105	1 Nichole	Batran
106	10 Connagh	Batran
107	1 Nancie	Batran
108	8 Eamon	Batran
109	5 Elliott	Batran

--Cererea 5

--

Sa se afiseze informatiile angajatilor care au hranit in ziua de 16 mai 2022 animale care contin sirul 'an' in nume,

--

ce nu lucreaza ca ingrijitori si care apartin gradinilor zoologice care au tranzactionat, in ultimi 5 ani, cu un partener

```

--al carei persoana de contact nu are numar de telefon.
--6 rezultate
SELECT id_ang "ID Angajat",
       prenume
       || ' '
       || nume "Nume Angajat",
       salariu "Salariu",
       id_zoo "Gradina Zoologica",
       id_job "Job",
       nr_tel "Numar Telefon",
       email "Email",
       data_ang "Data Angajarii"
FROM   angajati
WHERE  id_ang IN (SELECT DISTINCT id_ang
                  --angajatii care au hranit animalele ce au sirul 'an' in nume, pe 16 mai 2022
                  FROM   hraneste
                  WHERE  id_animal IN (SELECT DISTINCT id_animal
                                      FROM   animale
                                      WHERE  nume LIKE '%an%')
                  AND data_hranire = '16-MAY-22')
AND Lower(id_job) != 'ingrj'
AND id_zoo IN (SELECT DISTINCT gz.id_zoo
               --gradinile zoologice ce au tranzactionat in ultimii 5 ani cu parteneri
               FROM   gradina_zoologica gz,
                     beneficiaza b,
                     partener p
               --al caror persoana de contact nu are numar de telefon
               WHERE  gz.id_zoo = b.id_zoo
                     AND p.id_part = b.id_part
                     AND nr_tel_contact IS NULL
                     AND Months_Between(sysdate, data_serviciu) < 60);

```

```

--Cererea 5

--Sa se afiseze informatiile angajatilor care au hranit in ziua de 16 mai 2022 animale care contin sirul 'an' in nume,
--ce nu lucreaza ca ingrijitori si care apartin gradinilor zoologice care au tranzactionat, in ultimii 5 ani, cu un partener
--al carei persoana de contact nu are numar de telefon.
--6 rezultate

select id_ang "ID Angajat", prenume || ' ' || nume "Nume Angajat", salariu "Salariu", id_zoo "Gradina Zoologica",
       id_job "Job", nr_tel "Numar Telefon", email "Email", data_ang "Data Angajarii"
from   angajati
where  id_ang in (select distinct id_ang --angajatii care au hranit animalele ce au sirul 'an' in nume, pe 16 mai 2022
                  from   hraneste
                  where  id_animal in (select distinct id_animal
                                      from   animale
                                      where  nume like '%an%')
                  and data_hranire='16-MAY-22')
and lower(id_job)!='ingrj'
and id_zoo in (select distinct gz.id_zoo --gradinile zoologice ce au tranzactionat in ultimii 5 ani cu parteneri
               from   gradina_zoologica gz, beneficiaza b, partener p --al caror persoana de contact nu are numar de telefon
               where  gz.id_zoo=b.id_zoo
                     and p.id_part=b.id_part
                     and nr_tel_contact is null
                     and months_between(sysdate,data_serviciu)<60);

```

ID Angajat	Nume Angajat	Salariu	Gradina Zoologica	Job	Numar Telefon	Email	Data Angajarii
1	500 Brit Lam	2700		1 SRVC	0787513945	brit.lam@gmail.com	19-MAY-17
2	506 Eliza Beck	5400		1 DRSR	0756554712	eliza.beck@gmail.com	22-SEP-16
3	796 Sia Portillo	8500		12 VTRNR	0794330452	sia.portillo@gmail.com	13-OCT-17
4	797 Loren Zhang	4100		12 VTRNR	0795680693	loren.zhang@gmail.com	13-JUL-07
5	626 Ines Nolan	6500		7 VTRNR	0760464366	ines.nolan@gmail.com	01-JAN-02
6	668 Azra Foster	2500		10 SRVC	0706601216	azra.foster@gmail.com	15-DEC-02

13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

```
--Actualizare
--Sa se mareasca salariile ingrijitorilor de animale cu 10% in gradinile zoologice unde
--numarul animalelor este 7 ori mai mare ca numarul de ingrijitori
--19 rows updated
UPDATE angajati
SET    salariu = salariu * 1.10
WHERE  Lower(id_job) = 'ingrj'
      AND id_zoo IN (SELECT anim.id_zoo
                     FROM    (SELECT Count(*) nr_ang,
                                      id_zoo
                               FROM    angajati
                               WHERE  Lower(id_job) = 'ingrj'
                               GROUP BY id_zoo) ang,
                     --numarul ingrijitorilor din gradina zoologica
                     (SELECT Count(*) nr_anim,
                                      id_zoo
                               FROM    animale
                               GROUP BY id_zoo) anim
                     WHERE  anim.id_zoo = ang.id_zoo
                     AND nr_anim > 7 * nr_ang);

ROLLBACK; --pentru a reveni la valorile de dinainte de actualizare
COMMIT;  --daca vrem sa permanentizam modificarile
```

```

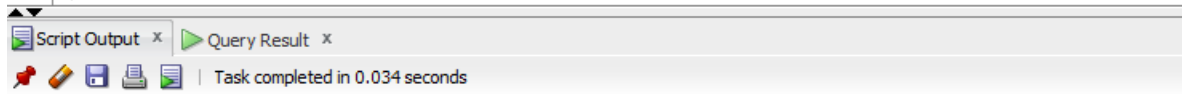
--Actualizare
--Sa se mareasca salariile ingrijitorilor de animale cu 10% in gradinile zoologice unde
--numarul animalelor este 7 ori mai mare ca numarul de ingrijitori
--19 rows updated

update angajati
set salariu=salariu*1.10
where lower(id_job)='ingrj'
and id_zoo in (select anim.id_zoo
               from (select count(*) nr_ang,id_zoo --numarul ingrijitorilor din gradina zoologica
                     from angajati
                     where lower(id_job)='ingrj'
                     group by id_zoo) ang,
               (select count(*) nr_anim,id_zoo --numarul animalelor din gradina zoologica
                 from animale
                 group by id_zoo) anim
               where anim.id_zoo=ang.id_zoo
               and nr_anim>7*nr_ang);

rollback; --pentru a reveni la valorile de dinainte de actualizare

commit; --daca vrem sa permanentizam modificarile

```



19 rows updated.

Rollback complete.

```

--Actualizare
--Sa se mareasca salariile minime si maxime cu 20% pentru job-
urile, mai putin ingrijitor, care s-au implicat in hranirea animalelor
--8 rows updated
UPDATE jobs j
SET      ( salariu_min, salariu_max ) = (SELECT salariu_min * 1.2,
                                             salariu_max * 1.2
                                             FROM   jobs
                                             WHERE  id_job = j.id_job)

WHERE  Lower(id_job) != 'ingrj'
      AND id_job IN (SELECT DISTINCT id_job
                     --toate job-urile angajatilor care s-au implicat in hranirea animalelor
                     FROM   hraneste h,
                           angajati a
                     WHERE  a.id_ang = h.id_ang);

ROLLBACK; --pentru a reveni la valorile de dinainte de actualizare
COMMIT;  --daca vrem sa permanentizam modificarile

```

```
--Actualizare
--Sa se mareasca salariile minime si maxime cu 20% pentru job-urile, mai putin ingrijitor, care s-au implicat in hranirea animalelor
--8 rows updated
```

```
update jobs j
set (salariu_min,salariu_max) = (select salariu_min*1.2,salariu_max*1.2
                                from jobs
                                where id_job=j.id_job)
where lower(id_job)!='ingrij'
and id_job in (select distinct id_job --toate job-urile angajatilor care s-au implicat in hranirea animalelor
               from hraneste h,angajati a
               where a.id_ang=h.id_ang );

rollback; --pentru a reveni la valorile de dinainte de actualizare

commit; --daca vrem sa permanentizam modificarile
```

Script Output x Query Result x
Task completed in 0.038 seconds

8 rows updated.

Rollback complete.

```
--Suprimare
--Sa se stearga toate restrictiile impuse asupra animalelor din speciile crepusculare
--21 rows updated
DELETE FROM restrictie
WHERE id_animal IN (SELECT id_animal --animalele din speciile crepusculare
                       FROM   animale a,
                              specie s
                       WHERE   a.id_specie = s.id_specie
                              AND Lower(s.tip_animal) = 'crepuscular');

ROLLBACK; --pentru a reveni la valorile de dinainte de actualizare
COMMIT; --daca vrem sa permanentizam modificarile
```

```
--Suprimare
--Sa se stearga toate restrictiile impuse asupra animalelor din speciile crepusculare
--21 rows updated
```

```
delete from restrictie
where id_animal in (select id_animal --animalele din speciile crepusculare
                      from animale a, specie s
                      where a.id_specie=s.id_specie
                            and lower(s.tip_animal)='crepuscular');

rollback; --pentru a reveni la valorile de dinainte de actualizare

commit; --daca vrem sa permanentizam modificarile
```

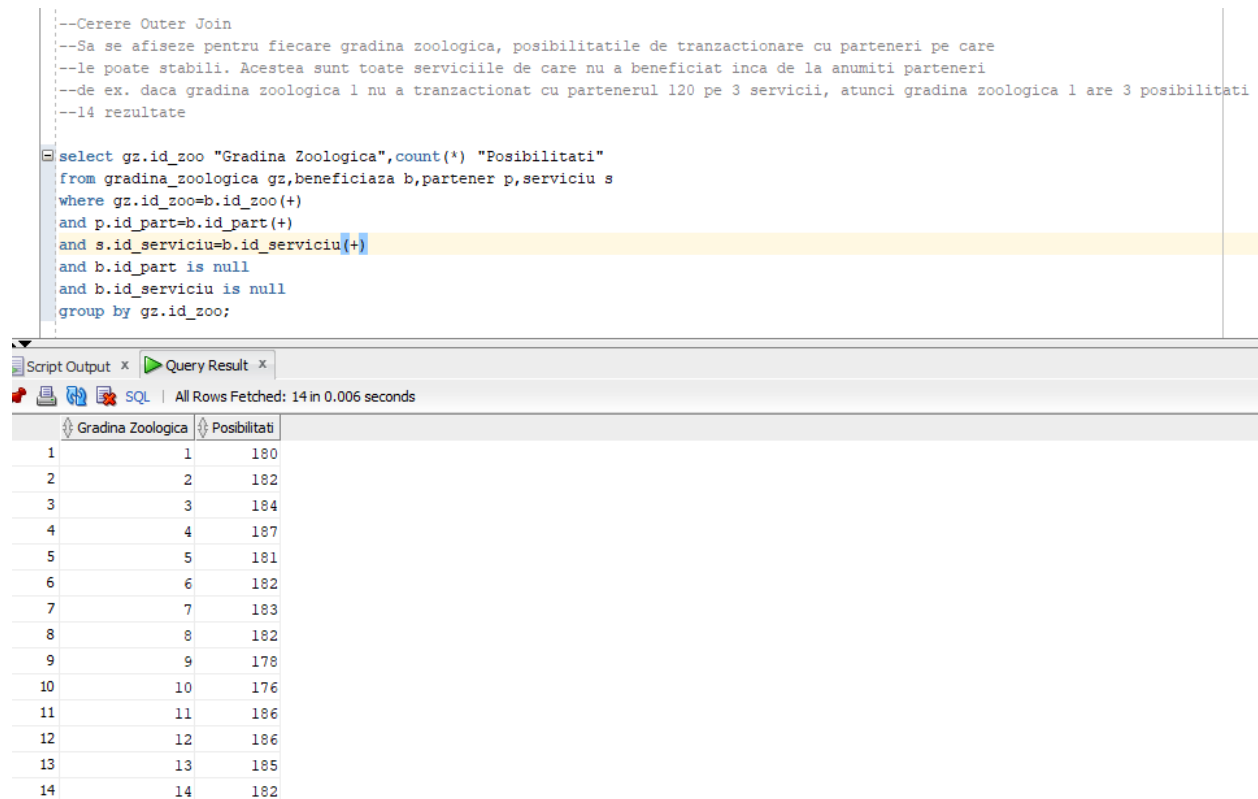
Script Output x Query Result x
Task completed in 0.038 seconds

21 rows deleted.

Rollback complete.

16. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația *outer-join* pe minimum 4 tabele și două cereri ce utilizează operația *division*.

```
--Cerere Outer Join
--
--Sa se afiseze pentru fiecare gradina zoologica, posibilitatile de tranzactionare cu parteneri pe
care
--
--le poate stabili. Acestea sunt toate serviciile de care nu a beneficiat inca de la anumiti parten
eri
--
--de ex. daca gradina zoologica 1 nu a tranzactionat cu partenerul 120 pe 3 servicii, atunci gradin
a zoologica 1 are 3 posibilitati
--14 rezultate
SELECT gz.id_zoo "Gradina Zoologica",
       Count(*) "Posibilitati"
FROM   gradina_zoologica gz,
       beneficiaza b,
       partener p,
       serviciu s
WHERE  gz.id_zoo = b.id_zoo(+)
       AND p.id_part = b.id_part(+)
       AND s.id_serviciu = b.id_serviciu(+)
       AND b.id_part IS NULL
       AND b.id_serviciu IS NULL
GROUP BY gz.id_zoo;
```



```
--Cerere Outer Join
--Sa se afiseze pentru fiecare gradina zoologica, posibilitatile de tranzactionare cu parteneri pe
care
--le poate stabili. Acestea sunt toate serviciile de care nu a beneficiat inca de la anumiti parteneri
--de ex. daca gradina zoologica 1 nu a tranzactionat cu partenerul 120 pe 3 servicii, atunci gradina zoologica 1 are 3 posibilitati
--14 rezultate

select gz.id_zoo "Gradina Zoologica",count(*) "Posibilitati"
from gradina_zoologica gz,beneficiaza b,partener p,serviciu s
where gz.id_zoo=b.id_zoo(+)
and p.id_part=b.id_part(+)
and s.id_serviciu=b.id_serviciu(+)
and b.id_part is null
and b.id_serviciu is null
group by gz.id_zoo;
```

Gradina Zoologica	Posibilitati
1	180
2	182
3	184
4	187
5	181
6	182
7	183
8	182
9	178
10	176
11	186
12	186
13	185
14	182

```

--Cerere 1 Division
--Sa se afiseze produsele vandute de toate gradinile zoologice care au minim 25 de animale
--1 rezultat
SELECT id_produs "ID Produs"
FROM vand
WHERE id_zoo IN(SELECT id_zoo
                  FROM animale
                  GROUP BY id_zoo
                  HAVING Count(*) >= 25)
GROUP BY id_produs
HAVING Count(id_zoo) = (SELECT Count(Count(*))
                        FROM animale
                        GROUP BY id_zoo
                        HAVING Count(*) >= 25);

```

The screenshot shows a SQL IDE interface. The top pane displays a SQL query with syntax highlighting. The query is identical to the one in the first block. The bottom pane shows the 'Query Result' tab with a single row of data.

ID Produs
1 1150

```

--Cerere 2 Division
--
Sa se afiseze partenerii atasati tuturor gradinilor zoologice care au doar 2 angajati pe pozitia
de administrator
--care castiga mai mult sau egal ca salariul mediu pe jobul de administrator.
--9 rezultate
WITH zoo_distincte
AS (SELECT Count(DISTINCT id_zoo) nr,
        id_part --calculez pentru fiecare partener numarul
        FROM beneficiaza
        --de gradini zoologice distincte care satisfac conditia
        WHERE id_zoo IN (SELECT id_zoo
                          --deoarece un partener poate tranzactiona cu aceeaasi
                          FROM angajati --gradina zoologica de mai multe ori
                          WHERE salariu >= (SELECT ( salariu_min + salariu_max
                                                    ) /
                                                    2
                                              FROM jobs
                                              WHERE Lower(id_job) = 'admn')
                          AND Lower(id_job) = 'admn')
        GROUP BY id_zoo

```

```

HAVING Count(*) = 2)
GROUP BY id_part)
SELECT id_part "ID Partener"
FROM beneficiaza b
WHERE id_zoo IN(SELECT id_zoo
FROM angajati
WHERE salariu >= (SELECT ( salariu_min + salariu_max ) / 2
FROM jobs
WHERE Lower(id_job) = 'admn')
AND Lower(id_job) = 'admn'
GROUP BY id_zoo
HAVING Count(*) = 2)
GROUP BY id_part
HAVING (SELECT nr
FROM zoo_distincte
WHERE id_part = b.id_part) = (SELECT Count(Count(*))
FROM angajati
WHERE salariu >= (SELECT
( salariu_min +
salariu_max ) / 2
FROM jobs
WHERE
Lower(id_job) = 'admn'
)
AND Lower(id_job) = 'admn'
GROUP BY id_zoo
HAVING Count(*) = 2);

```


<pre> --Cerere 2 Division --Sa se afiseze partenerii atasati tuturor gradinilor zoologice care au doar 2 angajati pe pozitia de administrator --care castiga mai mult sau egal ca salariul mediu pe jobul de administrator. --9 rezultate with zoo_distincte as(select count(distinct id_zoo) nr,id_part --calculez pentru fiecare partener numarul from beneficiaza --de gradini zoologice distincte care satisfac conditia where id_zoo in (select id_zoo --deoarece un partener poate tranzactiona cu aceeasi from angajati --gradina zoologica de mai multe ori where salariu >=(select (salariu_min+salariu_max)/2 from jobs where lower(id_job)='admn') and lower(id_job)='admn' group by id_zoo having count(*)=2) group by id_part) select id_part "ID Partener" from beneficiaza b where id_zoo in(select id_zoo from angajati where salariu >=(select (salariu_min+salariu_max)/2 from jobs where lower(id_job)='admn') and lower(id_job)='admn' group by id_zoo having count(*)=2) group by id_part having (select nr from zoo_distincte where id_part=b.id_part)=(select count(count(*)) from angajati where lower(id_job)='admn' group by id_job having count(*)=1) </pre>																					
Script Output x	Query Result x																				
<p>SQL All Rows Fetched: 9 in 0.017 seconds</p> <table> <thead> <tr> <th>ID Partener</th> <th></th> </tr> </thead> <tbody> <tr><td>1</td><td>10</td></tr> <tr><td>2</td><td>20</td></tr> <tr><td>3</td><td>50</td></tr> <tr><td>4</td><td>60</td></tr> <tr><td>5</td><td>70</td></tr> <tr><td>6</td><td>80</td></tr> <tr><td>7</td><td>90</td></tr> <tr><td>8</td><td>100</td></tr> <tr><td>9</td><td>120</td></tr> </tbody> </table>		ID Partener		1	10	2	20	3	50	4	60	5	70	6	80	7	90	8	100	9	120
ID Partener																					
1	10																				
2	20																				
3	50																				
4	60																				
5	70																				
6	80																				
7	90																				
8	100																				
9	120																				

17. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebrică, arbore algebric și limbaj (SQL), atât anterior cât și ulterior optimizării.

Sa se afiseze speciile carnivore care nu sunt crepusculare si care nu caracterizeaza un animal pe nume Ivy.

```

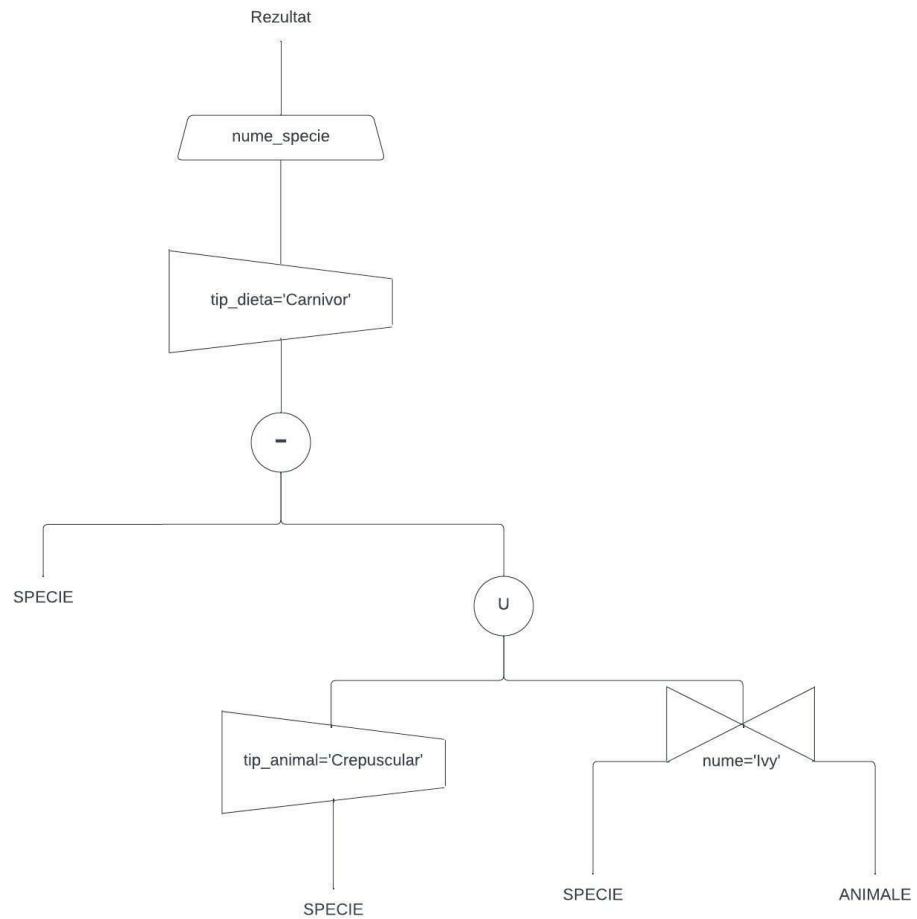
SELECT nume_specie "Nume Specie"
FROM (SELECT *
      FROM specie

```

```

minus
(SELECT *
FROM specie
WHERE tip_animal = 'Crepuscular'
UNION
SELECT s.*
FROM specie s,
animale a
WHERE s.id_specie = a.id_specie
AND a.num = 'Ivy'))
WHERE tip_dieta = 'Carnivor';

```



Rezultat= $\Pi_{nume_specie}(\sigma_{tip_dieta='Carnivor'}(SPECIE - (\sigma_{tip_animal='Crepuscular'}(SPECIE) \cup (SPECIE \bowtie_{nume='Ivy'} ANIMALE))))$

```

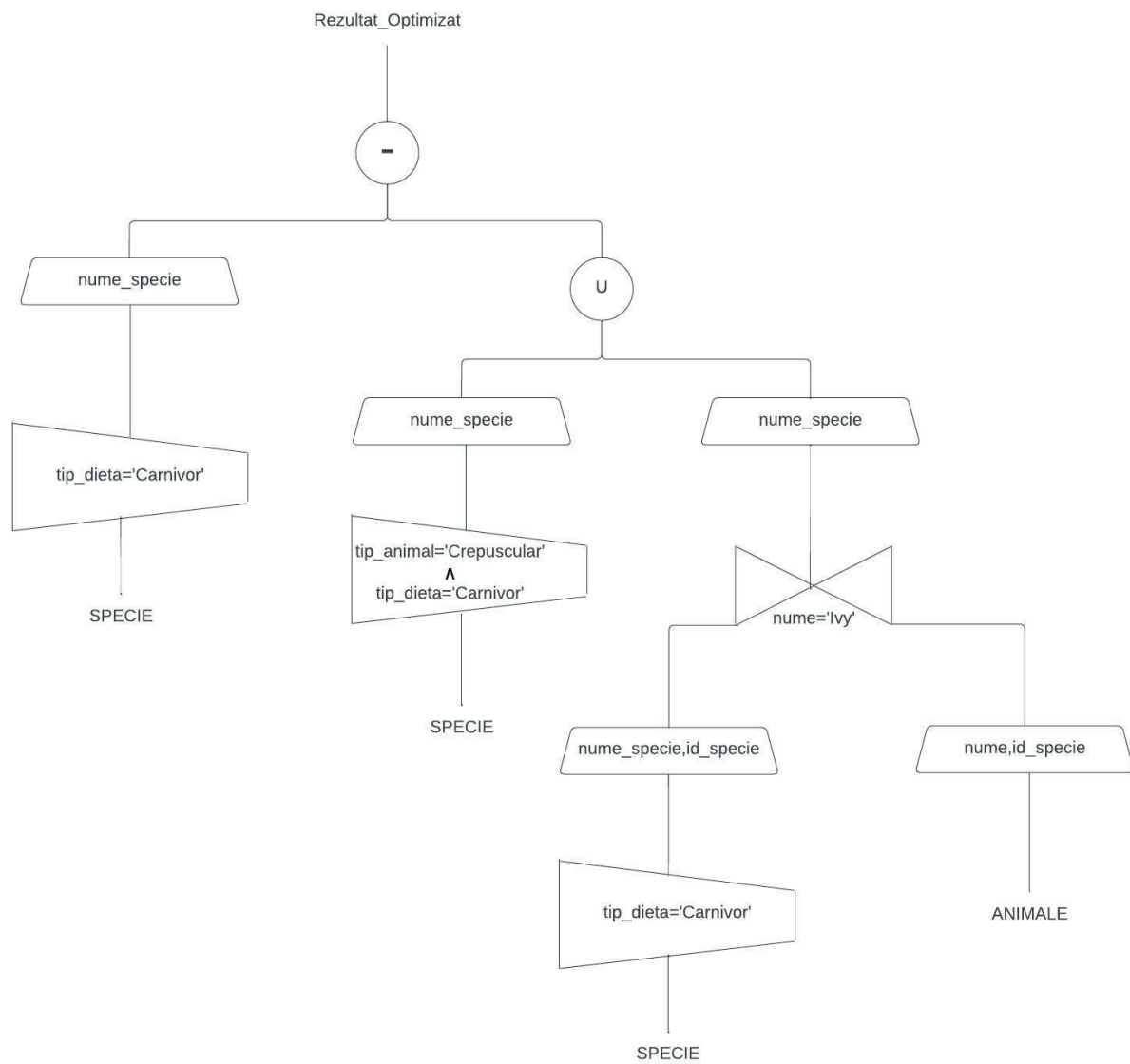
SELECT nume_specie
FROM specie
WHERE tip_dieta = 'Carnivor'
minus

```

```

(SELECT nume_specie
FROM specie
WHERE tip_animal = 'Crepuscular'
AND tip_dieta = 'Carnivor'
UNION
SELECT sp.nume_specie
FROM (SELECT nume_specie,
id_specie
FROM specie
WHERE tip_dieta = 'Carnivor') sp,
(SELECT nume,
id_specie
FROM animale) an
WHERE sp.id_specie = an.id_specie
AND an.nume = 'Ivy');

```



$$\text{Rezultat_Optimizat} = \Pi_{\text{nume_specie}}(\sigma_{\text{tip_dieta}='Carnivor'}(\text{SPECIE})) -$$

$$(\Pi_{\text{nume_specie}}(\sigma_{\text{tip_dieta}='Carnivor'}(\text{SPECIE})) \cup$$

$$\quad \wedge$$

$$\quad \text{tip_animal}='Crepuscular')$$

$$(\Pi_{\text{nume_specie}}(\Pi_{\text{nume_specie}, \text{id_specie}}(\sigma_{\text{tip_dieta}='Carnivor'}(\text{SPECIE}))) \bowtie_{\text{nume}='Ivy'}$$

$$\Pi_{\text{nume}, \text{id_specie}}(\text{ANIMALE}))$$

18. a. Realizarea normalizării BCNF, FN4, FN5.
 b. Aplicarea denormalizării, justificând necesitatea acesteia.

Un exemplu de non-BCNF este urmatorul:

Sa presupunem ca exista tabela INGRIJIRE cu attributele (id_zoo#, id_animal#, ingrijitor) care are ca scop sa plaseze animalul in grija ingrijitorului.

{id_zoo, id_animal} → ingrijitor, iar cum un ingrijitor poate lucra la o singura gradina zoologica avem ca

ingrijitor → id_zoo

rezulta ca nu este in forma BCNF, si ca sa o transformam realizam urmatoarele modificari:

avem prima tabela INGRIJIRE1 (id_animal#, ingrijitor)

si a doua tabela INGRIJIRE2 (ingrijitor #, id_zoo)

Un exemplu de non-FN4 este urmatorul:

Luam tabela HRANESTE din diagrama conceptuala. Pentru acest exemplu vom considera doar cele 3 chei primare id_ang#, id_animal#, id_hrana#. Un animal este hranit, indiferent de angajat, cu aceeași hrana.

$\text{id_animal} \rightarrow \rightarrow \text{id_hrana} \Rightarrow \text{id_animal} \rightarrow \rightarrow \text{id_ang}$

Astfel nu este în FN4 și se vor produce următoarele transformări pentru a aduce în FN4:

Se creează tabelele HRANESTE1(id_animal#,id_hrana#) și HRANESTE2(id_animal#,id_ang#).

Aplicarea denormalizării:

În tabela HRANESTE vom adăuga atributul id_zoo, ceea ce determină ieșirea din FN2. Adăugarea atributului id_zoo este justificată de faptul că astfel se evită un join pentru aflarea lui id_zoo.