Gestionarea unui lant de centre de adoptie pentru animale

Student: Tudoroiu Simona

Grupa: 151

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

* 1. ***Descrierea:***

Modelul de date ofera informatii despre modul de functionare al unui lant de centre de adoptie pentru animale, oferind detalii despre centrul de adoptie, animalele disponibile, personalul centrului de adoptie impreuna cu departamentele si joburile disponibile.

Pot exista animale ce sufera de anumite boli. Aceste vor beneficia de un tratament special fata de restul animalelor, primind astfel un medicament adecvat bolii.

Animalele detin un carnet de vaccinuri. Acesta va retine istoricul vaccinurilor pana in momentul de fata.

Un animal poate fi de rasa. Acest lucru se poate atesta printr-un pedigree.

In functie de animal, centrul de adoptie trebuie sa retina anumite tipuri de hrana, speciala pentru acel animal.

Pentru ca centrul de adoptie sa functioneze, este nevoie de un personal calificat. Personalul este impartit in mai multe departamente, de exemplu: departamentul doctorilor veterinari, departamentul de curatenie, departamentul de administratie etc., fiind disponibile mai multe joburi.

* 1. ***Regulile de funcționare ale modelului de date:***
* Un centru de adoptie pentru animale are o locatie stabilita, adica un oras din care face parte, si o tara, de unde apartine orasul.
* Un centru de adoptie poate adaposti mai multe animale, dar in cazul in care toate animalele sunt adoptate, atunci centrul nu mai adaposteste niciun animal pana in momentul aparitiei altui animal. Un animal poate fi adapostit de un singur centru de adoptie.
* Un animal poate detine un singur carnet de vaccinuri, acesta reprezentand istoricul de vaccinuri facute de acel animal. Insa animalul nu este obligat sa aiba un carnet de vaccinuri, fiind posibil ca persoanele ce adopta animalul sa il faca.
* Unui animal i se poate face mai multe vaccinuri, sau niciunul. In cazul in care unui animal ii este administrat cel putin un vaccin, atunci i se intocmeste un carnet de vaccinuri, unde vor aparea vaccinurile facute pana in acel moment.
* Un animal poate suferi de o boala pe care doctorii veterinari din centrul de adoptie o detecteaza. Aceasta boala se trateaza in centru folosind un anume medicament. Poate exista situatia in care un animal ce sufera de o boala nu poate fi tratat cu niciun medicament.
* Centrul de adoptie trebuie sa detina hrana pentru animale, pentru a putea ingriji animalele adapostite pana cand acestea sunt adoptate.
* Centrul de adoptie poate angaja maxim un personal pentru ca acesta sa functioneze.
* Personalul este impartit in mai multe departamente, de exemplu: departamentul doctorilor veterinar, cel de curatenie, de administratie etc.
* Pentru fiecare departament exista mai multe joburi, dar este necesar sa existe minim un job la fiecare departament.
* Pentru fiecare job sunt angajate mai multe persoane, dar este necesar sa fie angajate minim o persoane la fiecare job disponibil.

1. ***Prezentarea constrângerilor(restricții, reguli) impuse asupra modelului.***

* Un centru de adoptie trebuie sa aiba minim o hrana de animale.
* Un animal detine doar un carnet de vaccinuri.
* Un animal trebuie sa fie adapostit de cel putin un centru de adoptie.
* Intr-un carnet de vaccinuri trebuie sa existe cel putin un vaccin pentru ca aceste sa fie valid.
* Un medicament trebuie sa trateze cel putin o boala detectata.
* Un centru de adoptie poate sa angajeze maxim un personal.
* Un personal este format din minim un departament.
* Un departament trebuie sa fie compus din minim un job.
* La un job trebuie sa lucreze minim un angajat.

1. ***Entitati.***

Modelul de date cuprinde urmatoarele entitati: CENTRU\_DE\_ADOPTIE, ANIMAL, CARNET\_VACCINURI, VACCIN, HRANA\_ANIMALE, BOALA, MEDICAMENT, ORAS, TARA, PERSONAL, DEPARTAMENT, JOB, ANGAJAT. Modelul de date cuprinde si subentitatea ANIMAL\_DE\_RASA. Entitatea dependenta este CARNET\_VACCINURI.

CENTRU\_DE\_ADOPTIE = locatia ce adaposteste animale de pe strada si are ca scop gasirea unei noi case pentru acestea. Cheie primara: *id\_centru.*

ANIMAL = vietate care in cazul modelului de date este data spre adoptie. Cheie primara: *id\_animal.*

CARNET\_VACCINURI = act care contine informatii despre ce vaccinuri a facut un animal respectiv si in ce data, si totodata data vaccinului urmator. Aceasta entitate depinde de entitatile ANIMAL si VACCIN. Cheie primara compusa: *id\_carnet, id\_animal, id\_vaccin.*

VACCIN = substanta administrata in cazul modelului de date unui animal pentru a-l feri de boli si pentru a mentine sanatatea animalului cat se poate de buna. Cheie primara: *id\_vaccin.*

HRANA\_ANIMALE = mancare speciala administrarii doar animalelor. Cheie primara: *id\_hrana.*

BOALA = proces patologic care afecteaza organismul, in cazul modelului de date, organismul animalului. Cheie primara: *id\_hrana.*

MEDICAMENT = substanta administrata cu scopul de a trata o anumita boala, sau mai multe boli. Cheie primara: *id\_medicament.*

ORAS = entitate ce ajuta la identificarea orasului in care se afla centrul de adoptie. Cheie primara: *id\_oras.*

TARA = entitate ce ajuta la identificarea tarii corespunzatoare orasului in care se afla centrul de adoptie. Cheie primara: *id\_tara.*

PERSONAL = persoane desemnate sa ingrijeasca o firma si sa ajute la functionarea ei, in cazul de fata este vorba despre centrul de adoptie. Cheie primara: *id\_personal.*

DEPARTAMENT = diviziune a personalului, caruia ii este atribuita un anumit tip de datorie. Cheie primara: *id\_departament.*

JOB = slujba care este atribuita unui om la momentul angajarii in firma. Cheie primara: *id\_job.*

ANGAJAT = persoana ce ii este oferita sansa sa munceasca intr-o firma in schimbul unui salariu, avand un anumit job. Cheie primara: *id\_angajat.*

ANIMAL\_DE\_RASA = subentitate a entitatii ANIMAL care desemneaza un animal ce are o genetica buna, avand ambii parinti de rasa, recunoscandu-se acest lucru printr-un pedigree. Cheie primara: *id\_animal.*

1. ***Descrierea relațiilor, incluzând precizarea cardinalității acestora.***

CENTRU\_DE\_ADOPTIE\_*se\_afla\_*ORAS = relatie care leaga entitatile CENTRU\_DE\_ADOPTIE si ORAS si care reflecta legatura dintre acestea (in ce orase exista centre de adoptie pentru animale). Ea are cardinalitatea minima 1:1 (un centru de adoptie trebuie sa se afla in cel putin un oras, iar intr-un oras trebuie sa se afle cel putin un centru de adoptie) si cardinalitatea maxima M:1 (un centru de adoptie se poate afla intr-un singur oras, dar intr-un oras se pot afla mai multe centre de adoptie).

ORAS\_*apartine\_*TARA = relatie care leaga entitatile ORAS si TARA si care reflecta legatura dintre acestea (carei tari ii corespunde orasul respectiv in care se afla centre de adoptie). Ea are cardinalitatea minima 1:1 (un oras trebuie sa apartina de cel putin o tara, iar intr-o tara trebuie sa se afle cel putin un oras) si cardinalitatea maxima M:1(un oras poate apartine de o singura tara, iar intr-o tara se pot afla mai multe orase).

CENTRU\_DE\_ADOPTIE\_*adaposteste\_*ANIMAL = relatie care leaga entitatile CENTRU\_DE\_ADOPTIE si ANIMAL si care reflecta legatura dintre acestea (ce animale sunt adapostite de care centre de adoptie). Ea are cardinalitatea minima 1:0 (un centru de adoptie nu trebuie sa adaposteasca animale, la un moment dat, toate animalele fiind date spre adoptie, dar un animal trebuie sa fie adapostit de un centru de adoptie) si cardinalitatea maxima 1:M (un centru de adoptie poate adaposti mai multe animale, dar un animal poate fi adapostit de un singur centru de adoptie).

CENTRU\_DE\_ADOPTIE\_*are\_*HRANA\_ANIMALE = relatie care leaga entitatile CENTRU\_DE\_ADOPTIE si HRANA\_ANIMALE si care reflecta legatura dintre acestea (ce hrana de animale au centrele de adoptie). Ea are cardinalitatea minima 0:1 (un centru de adoptie trebuie sa aiba cel putin o hrana de animale, dar o hrana de animale nu trebuie sa fie detinuta de niciun centru de adoptie) si cardinalitatea maxima 1:M (un centru de adoptie poate avea mai multe hrane de animale, dar o hrana de animale poate fi detinuta de un singur centru de adoptie).

CENTRU\_DE\_ADOPTIE\_*angajeaza\_*PERSONAL = relatie care leaga entitatile CENTRU\_DE\_ADOPTIE si PERSONAL si care reflecta legatura dintre acestea (ce personal este angajat de ce centru de adoptie). Ea are cardinalitatea minima 1:1 (un centru de adoptie trebuie sa angajeze cel putin un personal, iar un personal trebuie sa fie angajat de cel putin un centru de adoptie) si cardinalitatea maxima 1:1 (un centru de adoptie poate angaja un singur personal, iar un personal poate fi angajat de un singur centru de adoptie).

ANIMAL\_*mananca\_*HRANA\_ANIMALE = relatie care leaga entitatile ANIMAL si HRANA\_ANIMALE si care reflecta legatura dintre acestea (ce hrana de animale mananca care animal). Ea are cardinalitatea minima 1:1 (un animal trebuie sa manance cel putin o hrana de animale, iar o hrana de animale trebuie sa fie mancata de cel putin un animal) si cardinalitatea maxima M:M (un animal poate manca mai multe hrane de animal, iar o hrana de animal poate fi mancata de mai multe animale).

ANIMAL\_*detine\_*CARNET\_VACCINURI = relatie care leaga entitatile ANIMAL si CARNET\_VACCINURI si care reflecta legatura dintre acestea (ce carnet de vaccinuri detine care animal). Ea are cardinalitatea minima 1:0 (un animal nu trebuie sa detina niciun carnet de vaccinuri, dar un carnet de vaccinuri trebuie sa fie detinut de un animal) si cardinalitatea maxima 1:1 (un animal poate detine un singur carnet de vaccinuri, iar un carnet de vaccinuri poate fi detinut de un singur animal).

ANIMAL\_*face\_*VACCIN = relatie care leaga entitatile ANIMAL si VACCIN si care reflecta legatura dintre acestea (ce vaccin a facut care animal). Ea are cardinalitatea minima 0:0 (un animal nu trebuie sa faca niciun vaccin, iar un vaccin nu trebuie sa fie facut de niciun animal) si cardinalitatea maxima 1:M (un animal poate face mai multe vaccinuri, dar un vaccin poate fi facut de un singur animal).

VACCIN\_*apare\_*CARNET\_VACCINURI = relatie care leaga entitatile VACCIN si CARNET\_VACCINURI si care reflecta legatura dintre acestea (ce vaccin apare in care carnet de vaccinuri). Ea are cardinalitatea minima 1:0 (un vaccin nu trebuie sa apare in niciun carnet de vaccinuri, dar intr-un carnet de vaccinuri trebuie sa apara cel putin un vaccin) si cardinalitatea maxima M:1 (un vaccin poate aparea intr-un singur carnet de vaccinuri, dar intr-un carnet de vaccinuri pot aparea mai multe vaccinuri).

ANIMAL\_*sufera\_*BOALA = relatie care leaga entitatile ANIMAL si BOALA si care reflecta legatura dintre acestea (de care boala sufera care animal). Ea are cardinalitatea minima 0:0 (un animal nu trebuie sa sufere de nicio boala, iar o boala nu trebuie sa apara la niciun animal) si cardinalitatea maxima M:M (un animal poate suferi de mai multe boli, iar o boala poate aparea la mai multe animale).

BOALA\_*se\_trateaza\_*MEDICAMENT = relatie care leaga entitatile BOALA si MEDICAMENT si care reflecta legatura dintre acestea (de care medicament este nevoie pentru a tratata care boala). Ea are cardinalitatea minima 1:0 (o boala nu trebuie sa se trateze cu niciun medicament, ori se trateaza de la sine, ori nu este tratabila, dar un medicament trebuie sa trateze cel putin o boala) si cardinalitatea maxima M:M (o boala se poate trata cu mai multe medicamente, iar un medicament poate trata mai multe boli).

PERSONAL\_*este\_format\_*DEPARTAMENT = relatie care leaga entitatile PERSONAL si DEPARTAMENT si care reflecta legatura dintre acestea (din ce departamente este formal personalul). Ea are cardinalitatea minima 1:1 (un personal trebuie sa fie format din cel putin un departament, iar un departament trebuie sa apartina de cel putin un personal) si cardinalitatea maxima 1:M (un personal poate fi format din mai multe departamente, dar un departament poate apartine de un singur personal).

DEPARTAMENT\_*este\_compus\_*JOB = relatie care leaga entitatile DEPARTAMENT si JOB si care reflecta legatura dintre acestea (din care joburi este compus care departament). Ea are cardinalitatea minima 1:1 (un departament trebuie sa fie compus din cel putin un job, iar un job trebuie sa apartina de cel putin un departament) si cardinalitatea maxima 1:M (un departament poate fi compus din mai multe joburi, dar un job poate apartine de un singur departament).

JOB\_*este\_atribuit\_*ANGAJAT = relatie care leaga entitatile JOB si ANGAJAT si care reflecta legatura dintre acestea (la ce job lucreaza care angajat). Ea are cardinalitatea minima 1:1 (un job trebuie sa fie atribuit cel putin unui angajat, iar un angajat trebuie sa lucreze la cel putin un job) si cardinalitatea maxima 1:M (un job poate fi atribuit mai multor angajati, iar un angajat poate lucra la un singur job).

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

Entitatea TARA are ca atribute:

* *id\_tara =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul unei tari.
* *nume =* variabila de tip caracter, de lungime maxima 25, care reprezinta numele tarii.

Entitatea ORAS are ca atribute:

* *id\_oras =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui oras.
* *nume =* variabila de tip caracter, de lungime maxima 25, care reprezinta numele orasului
* *id\_tara =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul tarii in care se afla orasul. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul TARA (FK).

Entitatea CENTRU\_DE\_ADOPTIE are ca atribute:

* *id\_centru =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui centru de adoptie.
* *nume =* variabila de tip caracter, de lungime maxima 50, care reprezinta numele unui centru de adoptie.
* *data\_aparitie =* variabila de tip data calendaristica, care reprezinta data aparitiei centrului de adoptie.
* *nr\_animale\_adoptate =* variabila de tip intreg, de lungime maxima 10, care reprezinta numarul de animale date spre adoptie in cadrul centrului de adoptie pentru animale.
* *id\_oras =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul orasului in care se afla centrul de adoptie. Atributul trebuie sa corespunda unei valori a cheii primare din tablul ORAS (FK).

Entitatea ANIMAL are ca atribute:

* *id\_animal =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui animal.
* *nume\_rasa =* variabila de tip caracter, de lungime maxima 25, care reprezinta numele rasei unui animal.
* *varsta =* variabila de tip intreg, de lungime maxima 3, care reprezinta varsta unui animal.
* *nume =* variabila de tip caracter, de lungime maxima 25, care reprezinta numele unui animal.
* *id\_centru =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul centrului de adoptie din care face parte animalul. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul CENTRU\_DE\_ADOPTIE (FK)

Entitatea ANIMAL\_DE\_RASA are ca atribute:

* *id\_animal =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui animal.
* *pedigree =* variabila de tip caracter, de lungime maxima 10, care atesta daca animalul are sau nu pedigree

Entitatea CARNET\_VACCINURI are ca atribute:

* *id\_carnet =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui carnet de vaccinuri
* *id\_animal =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul animalului care detine carnetul de vaccinuri. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul ANIMAL
* *id\_vaccin =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul vaccinului ce apare in carnetul de vaccinuri. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul VACCIN
* *nr\_vaccinuri =* variabila de tip intreg, de lungime maxima 2, care reprezinta numarul de vaccinuri pe care animalul le are administrate pana in prezent
* *data\_urmatorului\_vaccin =* variabila de tip data calendaristica, care reprezinta data urmatorului vaccin

Entitatea VACCIN are ca atribute:

* *id\_vaccin =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui vaccin
* *nume =* variabila de tip caracter, de lungime maxima 30, care reprezinta numele vaccinului
* *data\_administrare =* variabila de tip data calendaristica, care reprezinta data administrarii vaccinului respectiv
* *id\_animal =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul animalului caruia i-a fost administrat vaccinul

Entitatea HRANA\_ANIMALE are ca atribute:

* *id\_hrana =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui tip de hrana de animale
* *nume =* variabila de tip caracter, de lungime maxima 30, care reprezinta numele unui tip de hrana de animale
* *cantitate =* variabila de tip intreg, de lungime maxima 3, care reprezinta cantitatea unui tip de hrana de animale
* *id\_centru =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul centrului de adoptie care detine tipul de hrana de animale Atributul trebuie sa corespunda unei valori a cheii primare din tabelul CENTRU\_ANIMALE (FK)

Entitatea BOALA are ca atribute:

* *id\_boala =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul unei boli
* *nume =* variabila de tip caracter, de lungime maxima 30, care reprezinta denumirea unei boli
* *gravitate =* variabila de tip intreg, de lungime maxima 2, care reprezinta gradul de gravitate al bolii

Entitatea MEDICAMENT are ca atribute:

* *id\_medicament =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui medicament
* *nume =* variabila de tip caracter, de lungime maxima 30, care reprezinta denumirea medicamentului
* *doza\_pe\_zi =* variabila de tip intreg, de lungime maxima 2, care reprezinta doza recomandata de administrare a medicamentului pe zi

Entitatea PERSONAL are ca atribute:

* *id\_personal =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui personal
* *nr\_angajati* = variabile de tip intreg, de lungime maxima 4, care reprezinta numarul de angajati
* *salariu\_minim* = variabila de tip intreg, de lungime maxima 5, care reprezinta salariul minim al unui personal
* *salariu\_maxim* = variabila de tip intreg, de lungime maxima 5, care reprezinta salariul maxim al unui personal
* *id\_centru* = variabila de tip intreg, de lungime maxima 5, care reprezinta codul centrului unde este angajat personalul. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul CENTRU\_DE\_ADOPTIE(FK)

Entintate DEPARTAMENT are ca atribute:

* *id\_departament* = variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui departament
* *nume* = variabila de tip caracter, de lungime maxima 25, care reprezinta numele departamentului
* *nr\_persoane* = variabila de tip intreg, de lungime maxima 3, care reprezinta numarul de persoane dintr-un departament
* *salariu\_minim* = variabila de tip intreg, de lungime maxima 5, care reprezinta salariul minim al unui departament
* *salariu\_maxim* = variabila de tip intreg, de lungime maxima 5, care reprezinta salariul maxim al unui departament
* *id\_personal* = variabila de tip intreg, de lungime maxima 5, care reprezinta condul personalului din care face parte departamentul. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul PERSONAL(FK)

Entitatea JOB are ca atribute:

* *id\_job* = variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui job
* *nume* = variabila de tip caracter, de lungime maxima 30, care reprezinta numele jobului
* *salariu* = variabila de tip intreg, de lungime maxima 5, care reprezinta salariul primit in cadrul jobului
* *id\_departament* = varabila de tip intreg, de lungime maxima 5, care reprezinta codul departamentului din care face parte jobul. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul DEPARTAMENT(FK)

Entitatea ANGAJAT are ca atribute:

* *id\_angajat* = variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui angajat
* *nume* = variabila de tip caracter, de lungime maxima 30, care reprezinta numele angajatului
* *prenume* = variabila de tip caracter, de lungime maxima 30, care reprezinta prenumele angajatului
* *varsta* = variabila de tip intreg, de lungime maxima 2, care reprezinta varsta angajatului
* *data\_nasterii* = variabila de tip data calanderistica, care reprezinta data nasterii angajatului
* *salariu* = variabila de tip intreg, de lungime maxima 5, care reorezinta salariul angajatului
* *id\_job* = variabila de tip intreg, de lungime maxima 5, care reprezinta codul jobului angajatului. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul JOB(FK)

Relatia ANIMAL\_*sufera\_*BOALA are ca atribute:

* *id\_animal =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui animal. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul ANIMAL
* *id\_boala =* varabila de tip intreg, de lungime maxima 5, care reprezinta codul unei boli. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul BOALA

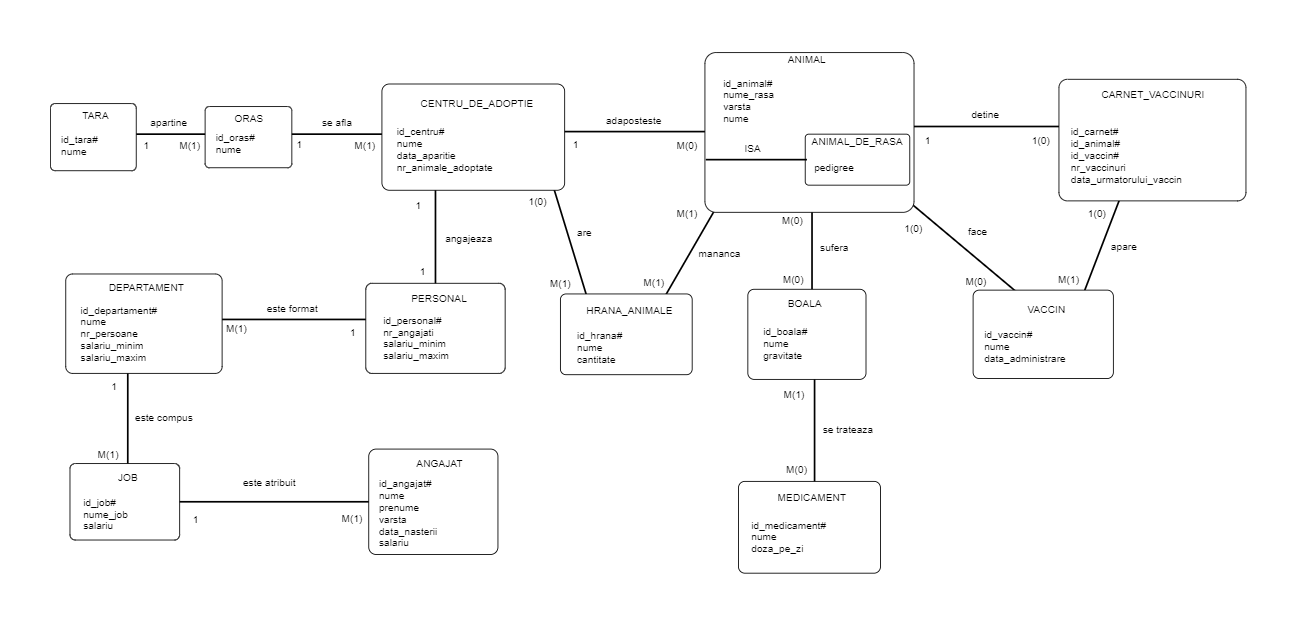
Relatia BOALA\_*se\_trateaza\_*MEDICAMENT are ca atribute:

* *id\_boala =* varabila de tip intreg, de lungime maxima 5, care reprezinta codul unei boli. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul BOALA
* *id\_medicament =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui medicament. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul MEDICAMENT

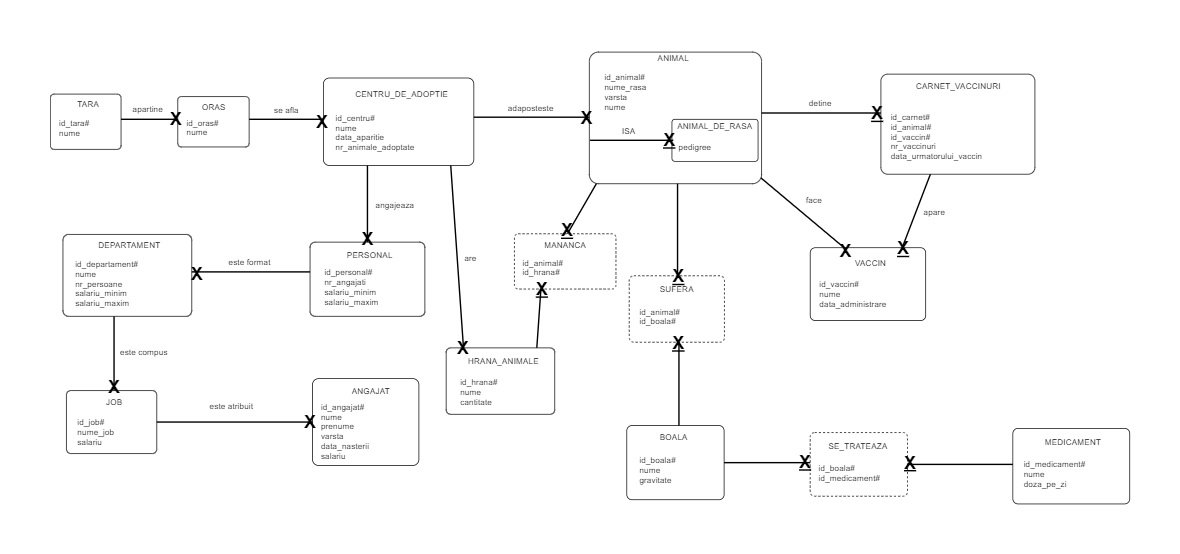
Relatia ANIMAL\_*mananca\_*HRANA\_ANIMALE are ca atribute:

* *id\_animal =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui animal. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul ANIMAL
* *id\_hrana =* variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui tip de hrana de animal. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul HRANA\_ANIMALE

***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.***

* TARA(*id\_tara#,* nume)
* ORAS(*id\_oras#,* nume, id\_tara)
* CENTRU\_DE\_ADOPTIE(*id\_centru#,* nume, data\_aparitie, nr\_animale\_adoptate, id\_oras)
* ANIMAL(*id\_animal#,* nume\_rasa, varsta, nume, id\_centru)
* ANIMAL\_DE\_RASA(*id\_animal#,* pedigree)
* CARNET\_VACCINURI(*id\_carnet#, id\_animal#, id\_vaccin#,* nr\_vaccinuri, data\_urmatorului\_vaccin)
* VACCIN(*id\_vaccin#,* nume, data\_administrare, id\_animal)
* SUFERA(*id\_animal#, id\_boala#)*
* BOALA(*id\_boala#,* nume, gravitate)
* SE\_TRATEAZA(*id\_boala#, id\_medicament#)*
* MEDICAMENT(*id\_medicament#,* nume, doza\_pe\_zi)
* MANANCA(*id\_animal#, id\_hrana#)*
* HRANA\_ANIMALE(*id\_hrana#,* nume, cantitate, id\_centru)
* PERSONAL(*id\_personal#,* nr\_angajati, salariu\_minim, salariu\_maxim, id\_centru)
* DEPARTAMENT(*id\_departament#,* nume, nr\_persoane, salariu\_minim, salariu\_maxim, id\_personal)
* JOB(*id\_job#,* nume\_job, salariu, id\_departament)
* ANGAJAT(*id\_angajat#,* nume, prenume, varsta, data\_nasterii, salariu, id\_job)

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

* *Forma normala 1 (FN1)*

Demonstratie pe cateva exemple:

Relatia CENTRU\_DE\_ADOPTIE are drept cheie primara *id\_centru,* fiind un identificator unic, deoarece pentru fiecare centru de adoptie exista un id unic. Astfel, atributul *id\_centru* este o valoare indivizibila, deci relatia se afla in forma normala 1, existand un identificator unic pentru toate intrarile din tabel.

Relatia CARNET\_VACCINURI are o cheie primara compusa formata din: *id\_centru, id\_animal, id\_vaccin.* Cheia primara este un identificator unic pentru entitatea CARNET\_VACCINURI, atributele ce formeaza cheia primara fiind impreuna o valora indivizibila. Astfel, relatia se afla in forma normala 1.

Sa presupunem ca acelasi vaccin a fost administrat de mai multe ori. Initial, relatia VACCIN\_nonFN1 va arata astfel:

|  |  |
| --- | --- |
| *id\_vaccin#* | data\_administrare |
| 1 | 11-02-2002, 24-11-2004 |
| 2 | 05-06-2018 |
| 3 | 16-12-2020, 14-06-2021 |

Relatia de mai sus nu este in FN1, deoarece atributului data\_administrare nu ii corespunde o valoara indivizibila.

Dupa aplicarea primei forme normale, relatia va arata astfel:

|  |  |
| --- | --- |
| *id\_vaccin#* | *data\_administrare#* |
| 1 | 11-02-2002 |
| 1 | 24-11-2004 |
| 2 | 05-06-2018 |
| 3 | 16-12-2020 |
| 3 | 14-06-2021 |

Dupa normalizare, fiecarui atribut ii corespunde o valoare indivizibila.

* *Forma normala 2 (FN2)*

Initial, relatia CARNET\_VACCINURI arata astfel:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *id\_carnet#* | *id\_animal#* | *id\_vaccin#* | nr\_vaccinuri | data\_urmatorului  \_vaccin | nume\_vaccin | data\_administrare |
| 1 | 1 | 1 | 4 | 12-06-2020 | eurican | 12-01-2020 |
| 3 | 3 | 2 | 3 | 05-07-2022 | rabisin | 04-02-2022 |
| 5 | 5 | 1 | 1 | 10-09-2022 | purevax | 10-04-2022 |
| 7 | 7 | 5 | 3 | 03-04-2021 | primodog | 01-11-2020 |

Relatia se afla in FN1, deoarece exista identificator unic pentru toate intrarile in tabel.

Pentru ca relatia sa se afle in FN2, trebuie ca fiecare atribut care nu este cheie sa fie dependent de intreaga cheie primara. Asadar, atributele *nr\_vaccinuri, data\_urmatorului\_vaccin, nume\_vaccin, data\_administrare* trebuie sa depinda de intreaga cheie primara, si anume ***id\_carnet#, id\_animal#, id\_vaccin#.*** Se observa totusi ca acest lucru nu este respectat, din cauza dependentei directe dintre ***id\_vaccin#,*** *nume\_vaccin* si *data\_administrare,* ceea ce inseamna ca cele doua atribute, *nume\_vaccin* si *data\_administrare,* depind doar de o parte a cheii primare, mai exact de ***id\_vaccin#***.

Astfel, relatia nu se afla in forma normala 2.

Dependentele vor fi urmatoarele:

* {*id\_vaccin#*} -> {nume\_vaccin, data\_administrare}
* {*id\_carnet#, id\_animal#, id\_vaccin#*} -> {nr\_vaccinuri, data\_urmatorului\_vaccin}

In final, relatia transformata in FN2 va arata astfel:

CARNET\_VACCINURI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *id\_carnet#* | *id\_animal#* | *id\_vaccin#* | nr\_vaccinuri | data\_urmatorului\_vaccin |
| 1 | 1 | 1 | 4 | 12-06-2020 |
| 3 | 3 | 2 | 3 | 05-07-2022 |
| 5 | 5 | 1 | 1 | 10-09-2022 |
| 7 | 7 | 5 | 3 | 03-04-2021 |

VACCIN

|  |  |  |
| --- | --- | --- |
| *id\_vaccin#* | nume\_vaccin | data\_administrare |
| 1 | eurican | 12-01-2020 |
| 2 | rabisin | 04-02-2022 |
| 1 | purevax | 10-04-2022 |
| 5 | primodog | 01-11-2020 |

* *Forma normala 3 (FN3)*

Pentru a putea exemplifica aceasta forma normala, vom realiza un

exemplu fictiv pentru ANGAJAT, presupunand ca atributul *salariu* depinde de atributul *varsta,* fapt ce nu se respecta si in cazul diagramei construie.

ANGAJAT\_nonFN3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *id\_angajat#* | nume | prenume | varsta | data\_nasterii | salariu | id\_job(FK) |
| 13 | Gheorghe | Marius | 30 | 05-03-1992 | 2000 | 3 |
| 16 | Ionescu | Ana | 22 | 01-02-2000 | 1670 | 2 |
| 20 | Marian | Andrei | 30 | 08-04-1992 | 2000 | 3 |
| 25 | Petre | Ioana | 20 | 06-06-2002 | 1540 | 5 |
| 26 | Agache | Cosmina | 21 | 09-05-2001 | 1600 | 1 |

Aceasta relatia este in prima forma normala deoarece exista identificator unic pentru toate intrarile din tabel, si anume *id\_angajat#.*

Relatia se afla si in forma normala 2, deoarece atributele *nume, prenume, varsta, data\_nasterii, salariu* sunt dependente de intreaga cheie primara ***id\_angajat#,*** nefacand parte din aceasta.

Se observa ca in exemplul de fata, atributul salariu depinde de cheia primara

***id\_angajat#*** prin intermediul atributului *varsta.*

Astfel, exemplul dat nu se afla in FN3, acest lucru fiind observabil din relatiile de dependenta urmatoare:

* {*id\_angajat#*} -> {nume, prenume, varsta, data\_nasterii}
* {*id\_angajat#*} -> {varsta} -> {salariu}

Vom folosi regula Casey-Delobel pentru a aduce relatia ANGAJAT\_nonFN3 in forma normala 3. Aceasta regula presupune descompunerea relatiei prin eliminarea dependentei functionale tranzitive. Se va inlocui relatia prin doua proiectii:

* ANGAJAT\_FN3(*id\_angajat#,* nume, prenume, varsta, data\_nasterii)
* VARSTA\_SALARIU(varsta, salariu)

ANGAJAT\_FN3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *id\_angajat#* | nume | prenume | varsta | data\_nasterii | id\_job(FK) |
| 13 | Gheorghe | Marius | 30 | 05-03-1992 | 3 |
| 16 | Ionescu | Ana | 22 | 01-02-2000 | 2 |
| 20 | Marian | Andrei | 30 | 08-04-1992 | 3 |
| 25 | Petre | Ioana | 20 | 06-06-2002 | 5 |
| 26 | Agache | Cosmina | 21 | 09-05-2001 | 1 |

VARSTA\_SALARIU

|  |  |
| --- | --- |
| varsta | salariu |
| 30 | 2000 |
| 22 | 1670 |
| 30 | 2000 |
| 20 | 1540 |
| 21 | 1600 |

***10. 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).***

**Crearea tabelelor:**

create table tara

(

id\_tara number(5),

nume varchar(25) constraint nume\_tara\_nn not null,

unique (nume)

);

alter table tara

add constraint id\_tara\_pk primary key(id\_tara);

create table oras

(

id\_oras number(5),

nume varchar(25) constraint nume\_oras\_nn not null,

id\_tara number(5),

unique (nume)

);

alter table oras

add constraint id\_oras\_pk primary key(id\_oras);

alter table oras

add constraint id\_tara\_fk foreign key(id\_tara) references tara(id\_tara);

create table centru\_de\_adoptie

(

id\_centru number(5),

nume varchar(50) constraint nume\_centru\_nn not null,

data\_aparitie date constraint aparitie\_nn not null,

nr\_animale\_adoptate number(10),

id\_oras number(5),

unique (nume)

);

alter table centru\_de\_adoptie

add constraint id\_centru\_pk primary key(id\_centru);

alter table centru\_de\_adoptie

add constraint id\_oras\_fk foreign key(id\_oras) references oras(id\_oras);

create table hrana\_animale

(

id\_hrana number(5),

nume varchar(30) constraint nume\_hrana\_nn not null,

cantitate number(3) constraint cantitate\_hrana\_nn not null,

id\_centru number(5),

constraint id\_hrana\_pk primary key(id\_hrana),

constraint id\_centru\_fk foreign key(id\_centru) references centru\_de\_adoptie(id\_centru)

);

create table animal

(

id\_animal number(5),

nume\_rasa varchar(25),

varsta number(3) constraint varsta\_animal\_nn not null,

nume varchar(25),

id\_centru number(5),

pedigree varchar(10) constraint pedigree\_nn not null,

constraint id\_animal\_pk primary key(id\_animal),

constraint id\_centru\_2\_fk foreign key(id\_centru) references centru\_de\_adoptie(id\_centru),

constraint pedigree\_a check (pedigree='are' or pedigree='nu are')

);

create table vaccin

(

id\_vaccin number(5),

nume varchar(30) constraint nume\_vaccin\_nn not null,

data\_administrare date constraint data\_vaccin\_nn not null,

id\_animal number(5),

constraint id\_vaccin\_pk primary key(id\_vaccin),

constraint id\_animal\_fk foreign key(id\_animal) references animal(id\_animal)

);

create table carnet\_vaccinuri

(

id\_carnet number(5),

id\_animal number(5),

id\_vaccin number(5),

nr\_vaccinuri number(2) constraint nr\_vaccinuri\_nn not null,

data\_urmatorului\_vaccin date,

constraint pk\_carnet\_animal\_vaccin primary key(id\_carnet, id\_animal, id\_vaccin),

constraint id\_animal\_2\_fk foreign key (id\_animal) references animal (id\_animal),

constraint id\_vaccin\_fk foreign key (id\_vaccin) references vaccin (id\_vaccin)

);

create table boala

(

id\_boala number(5),

nume varchar(30) constraint nume\_boala\_nn not null,

gravitate number(2) constraint gravitate\_boala\_nn not null,

constraint id\_boala\_pk primary key (id\_boala)

);

create table medicament

(

id\_medicament number(5),

nume varchar(30) constraint nume\_medicament\_nn not null,

doza\_pe\_zi number(2) constraint doza\_medicament\_nn not null,

constraint id\_medicament\_pk primary key (id\_medicament)

);

create table personal

(

id\_personal number(5),

nr\_angajati number(4) constraint nr\_angajati\_nn not null,

salariu\_minim number(5) constraint sal\_min\_p\_nn not null,

salariu\_maxim number(5) constraint sal\_max\_p\_nn not null,

id\_centru number(5),

constraint id\_personal\_pk primary key (id\_personal),

constraint id\_centru\_3\_fk foreign key (id\_centru) references centru\_de\_adoptie (id\_centru)

);

create table departament

(

id\_departament number(5),

nume varchar(25) constraint nume\_departament\_nn not null,

nr\_persoane number(3) constraint nr\_persoane\_d\_nn not null,

salariu\_minim number(5) constraint sal\_min\_d\_nn not null,

salariu\_maxim number(5) constraint sal\_max\_d\_nn not null,

id\_personal number(5),

constraint id\_departament\_pk primary key (id\_departament),

constraint id\_personal\_fk foreign key (id\_personal) references personal (id\_personal)

);

create table job

(

id\_job number(5),

nume varchar(30) constraint nume\_job\_nn not null,

salariu number(5) constraint salariu\_job\_nn not null,

id\_departament number(5),

constraint id\_job\_pk primary key (id\_job),

constraint id\_departament\_fk foreign key (id\_departament) references departament (id\_departament)

);

create table angajat

(

id\_angajat number(5),

nume varchar(30) constraint nume\_angajat\_nn not null,

prenume varchar(30) constraint prenume\_angajat\_nn not null,

varsta number(2) constraint varsta\_angajat\_nn not null,

data\_nasterii date constraint data\_angajat\_nn not null,

salariu number(5) constraint salariu\_angajat\_nn not null,

id\_job number(5),

constraint id\_angajat\_pk primary key(id\_angajat),

constraint id\_job\_fk foreign key(id\_job) references job (id\_job)

);

create table sufera

(

id\_animal number(5),

id\_boala number(5),

constraint pk\_animal\_boala primary key (id\_animal, id\_boala),

constraint id\_animal\_4\_fk foreign key (id\_animal) references animal(id\_animal),

constraint id\_boala\_2\_fk foreign key (id\_boala) references boala (id\_boala)

);

create table se\_treateaza

(

id\_boala number(5),

id\_medicament number(5),

constraint pk\_boala\_medicament primary key (id\_boala, id\_medicament),

constraint id\_boala\_3\_fk foreign key (id\_boala) references boala (id\_boala),

constraint id\_medicament\_fk foreign key (id\_medicament) references medicament(id\_medicament)

);

create table mananca

(

id\_animal number(5),

id\_hrana number(5),

constraint pk\_animal\_hrana primary key (id\_animal, id\_hrana),

constraint id\_animal\_5\_fk foreign key (id\_animal) references animal(id\_animal),

constraint id\_hrana\_2\_fk foreign key (id\_hrana) references hrana\_animale (id\_hrana)

);

**Inserarea de date:**

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

create sequence seq\_tara

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into tara

values(seq\_tara.nextval, 'Romania');

insert into tara

values(seq\_tara.nextval, 'Franta');

insert into tara

values(seq\_tara.nextval, 'Germania');

insert into tara

values(seq\_tara.nextval, 'Italia');

insert into tara

values(seq\_tara.nextval, 'Bulgaria');

insert into tara

values(seq\_tara.nextval, 'Ucraina');

insert into tara

values(seq\_tara.nextval, 'Spania');

insert into tara

values(seq\_tara.nextval, 'Ungaria');

insert into tara

values(seq\_tara.nextval, 'Polonia');

insert into tara

values(seq\_tara.nextval, 'Danemarca');

commit;

create sequence seq\_oras

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into oras

values(seq\_oras.nextval, 'Constanta', 1);

insert into oras

values(seq\_oras.nextval, 'Craiova', 1);

insert into oras

values(seq\_oras.nextval, 'Roma', 4);

insert into oras

values(seq\_oras.nextval, 'Barcelona', 7);

insert into oras

values(seq\_oras.nextval, 'Navodari', 1);

insert into oras

values(seq\_oras.nextval, 'Bucuresti', 1);

insert into oras

values(seq\_oras.nextval, 'Verona', 4);

insert into oras

values(seq\_oras.nextval, 'Timisoara', 1);

insert into oras

values(seq\_oras.nextval, 'Alba-Iulia', 1);

insert into oras

values(seq\_oras.nextval, 'Iasi', 1);

commit;

create sequence seq\_centru

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into centru\_de\_adoptie

values(seq\_centru.nextval, 'Animalutele fericite', TO\_DATE('12-03-2004', 'dd-mm-yyyy'), 100, 2);

insert into centru\_de\_adoptie

values(seq\_centru.nextval, 'Adopta un prieten', TO\_DATE('02-11-2010', 'dd-mm-yyyy'), 150, 6);

insert into centru\_de\_adoptie

values(seq\_centru.nextval, 'Animalul tau', TO\_DATE('25-06-2012', 'dd-mm-yyyy'), 98, 10);

insert into centru\_de\_adoptie

values(seq\_centru.nextval, 'Adopta-ma!', TO\_DATE('16-09-2006', 'dd-mm-yyyy'), 200, 4);

insert into centru\_de\_adoptie

values(seq\_centru.nextval, 'Animale prietenoase', TO\_DATE('05-10-2013', 'dd-mm-yyyy'), 59, 5);

insert into centru\_de\_adoptie

values(seq\_centru.nextval, 'Animalutele fericite', TO\_DATE('12-03-2004', 'dd-mm-yyyy'), 100, 2);

commit;

create sequence seq\_hrana

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into hrana\_animale

values(seq\_hrana.nextval, 'Chonk', 300, 1);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Cocomel', 579, 2);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Papbun', 400, 3);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Falrel', 280, 4);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Varel', 600, 5);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Chonk', 250, 2);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Chonk', 190, 5);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Cocomel', 398, 1);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Cocomel', 400, 4);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Varel', 387, 3);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Papbun', 376, 1);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Falrel', 578, 5);

commit;

create sequence seq\_animal

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into animal

values(seq\_animal.nextval, 'British', 3, 'Mimi', 2, 'are');

insert into animal

values(seq\_animal.nextval, 'Scotish', 1, 'Misha', 5, 'nu are');

insert into animal

values(seq\_animal.nextval, 'nu stim', 2, 'Cara', 4,'nu are');

insert into animal

values(seq\_animal.nextval, 'Labrador', 4, 'Rex', 1,'are');

insert into animal

values(seq\_animal.nextval, 'Pug', 1, 'Boni', 3,'nu are');

commit;

create sequence seq\_vaccin

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into vaccin

values(seq\_vaccin.nextval, 'Eurican', TO\_DATE('12-04-2005', 'dd-mm-yyyy'),1);

insert into vaccin

values(seq\_vaccin.nextval, 'Rabisim', TO\_DATE('04-03-2010', 'dd-mm-yyyy'),2);

insert into vaccin

values(seq\_vaccin.nextval, 'Purevax', TO\_DATE('24-11-2006', 'dd-mm-yyyy'),3);

insert into vaccin

values(seq\_vaccin.nextval, 'Primodog', TO\_DATE('15-12-2013', 'dd-mm-yyyy'),4);

insert into vaccin

values(seq\_vaccin.nextval, 'Eurican', TO\_DATE('02-09-2012', 'dd-mm-yyyy'),5);

commit;

create sequence seq\_carnet

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into carnet\_vaccinuri

values(seq\_carnet.nextval, 1, 3, 1, TO\_DATE('23-07-2022', 'dd-mm-yyyy'));

insert into carnet\_vaccinuri

values(seq\_carnet.nextval, 2, 1, 3, TO\_DATE('12-09-2022', 'dd-mm-yyyy'));

insert into carnet\_vaccinuri

values(seq\_carnet.nextval, 3, 5, 6, TO\_DATE('15-10-2022', 'dd-mm-yyyy'));

insert into carnet\_vaccinuri

values(seq\_carnet.nextval, 4, 2, 2, TO\_DATE('13-11-2022', 'dd-mm-yyyy'));

insert into carnet\_vaccinuri

values(seq\_carnet.nextval, 5, 4, 1, TO\_DATE('25-08-2022', 'dd-mm-yyyy'));

commit;

create sequence seq\_boala

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into boala

values(seq\_boala.nextval, 'Parazitoza', 3);

insert into boala

values(seq\_boala.nextval, 'Salmoneloza', 6);

insert into boala

values(seq\_boala.nextval, 'Stenoza Aortica', 10);

insert into boala

values(seq\_boala.nextval, 'Toxopalsmoza', 4);

insert into boala

values(seq\_boala.nextval, 'Otita', 5);

commit;

create sequence seq\_medicament

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into medicament

values(seq\_medicament.nextval, 'Brufil', 3);

insert into medicament

values(seq\_medicament.nextval, 'Iosman', 2);

insert into medicament

values(seq\_medicament.nextval, 'Calmarin', 1);

insert into medicament

values(seq\_medicament.nextval, 'Maldon', 1);

insert into medicament

values(seq\_medicament.nextval, 'Aurilo', 2);

commit;

create sequence seq\_personal

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into personal

values(seq\_personal.nextval, 100, 1300, 3000, 1);

insert into personal

values(seq\_personal.nextval, 210, 1250, 2998, 2);

insert into personal

values(seq\_personal.nextval, 87, 1100, 2500, 3);

insert into personal

values(seq\_personal.nextval, 180, 1400, 3100, 4);

insert into personal

values(seq\_personal.nextval, 150, 1247, 2876, 5);

commit;

create sequence seq\_departament

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into departament

values(seq\_departament.nextval, 'Curatenie',30, 1300, 1400, 1);

insert into departament

values(seq\_departament.nextval, 'Curatenie',25, 1250, 1300, 2);

insert into departament

values(seq\_departament.nextval, 'Curatenie',15, 1100, 1278, 3);

insert into departament

values(seq\_departament.nextval, 'Curatenie',40, 1400, 1500, 4);

insert into departament

values(seq\_departament.nextval, 'Curatenie',35, 1247, 1350, 5);

insert into departament

values(seq\_departament.nextval, 'Curatenie',30, 1300, 1400, 1);

insert into departament

values(seq\_departament.nextval, 'Doctori',50, 2500, 3000, 1);

insert into departament

values(seq\_departament.nextval, 'Doctori',100, 2480, 2998, 2);

insert into departament

values(seq\_departament.nextval, 'Doctori',45, 2000, 2500, 3);

insert into departament

values(seq\_departament.nextval, 'Doctori',68, 2657, 3100, 4);

insert into departament

values(seq\_departament.nextval, 'Doctori',48, 2150, 2876, 5);

insert into departament

values(seq\_departament.nextval, 'Administratie',20, 1500, 2500, 1);

insert into departament

values(seq\_departament.nextval, 'Administratie',85, 1458, 2480, 2);

insert into departament

values(seq\_departament.nextval, 'Administratie',27, 1376, 2000, 3);

insert into departament

values(seq\_departament.nextval, 'Administratie',72, 2000, 2657, 4);

insert into departament

values(seq\_departament.nextval, 'Administratie',67, 1500, 2150, 5);

commit;

create sequence seq\_job

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into job

values(seq\_job.nextval, 'Ingrijitor animale',1300, 1);

insert into job

values(seq\_job.nextval, 'Om de serviciu',1400, 1);

insert into job

values(seq\_job.nextval, 'Ingrijitor animale',1250, 2);

insert into job

values(seq\_job.nextval, 'Om de serviciu',1300, 2);

insert into job

values(seq\_job.nextval, 'Ingrijitor animale',1100, 3);

insert into job

values(seq\_job.nextval, 'Om de serviciu',1278, 3);

insert into job

values(seq\_job.nextval, 'Ingrijitor animale',1400, 4);

insert into job

values(seq\_job.nextval, 'Om de serviciu',1500, 4);

insert into job

values(seq\_job.nextval, 'Ingrijitor animale',1247, 5);

insert into job

values(seq\_job.nextval, 'Om de serviciu',1350, 5);

insert into job

values(seq\_job.nextval, 'Medic veterinar',2500, 6);

insert into job

values(seq\_job.nextval, 'Chirurg veterinar',3000, 6);

insert into job

values(seq\_job.nextval, 'Medic veterinar',2480, 7);

insert into job

values(seq\_job.nextval, 'Chirurg veterinar',2998, 7);

insert into job

values(seq\_job.nextval, 'Medic veterinar',2000, 8);

insert into job

values(seq\_job.nextval, 'Chirurg veterinar',2500, 8);

insert into job

values(seq\_job.nextval, 'Medic veterinar',2657, 9);

insert into job

values(seq\_job.nextval, 'Chirurg veterinar',3100, 9);

insert into job

values(seq\_job.nextval, 'Medic veterinar',2150, 10);

insert into job

values(seq\_job.nextval, 'Chirurg veterinar',2876, 10);

insert into job

values(seq\_job.nextval, 'Casier',1500, 11);

insert into job

values(seq\_job.nextval, 'Manager',2500, 11);

insert into job

values(seq\_job.nextval, 'Casier',1458, 12);

insert into job

values(seq\_job.nextval, 'Manager',2480, 12);

insert into job

values(seq\_job.nextval, 'Casier',1376, 13);

insert into job

values(seq\_job.nextval, 'Manager',2000, 13);

insert into job

values(seq\_job.nextval, 'Casier',2000, 14);

insert into job

values(seq\_job.nextval, 'Manager',2657, 14);

insert into job

values(seq\_job.nextval, 'Casier',1500, 15);

insert into job

values(seq\_job.nextval, 'Manager',2150, 15);

commit;

create sequence seq\_angajat

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into angajat

values(seq\_angajat.nextval, 'Agache','Cosmin', 20,TO\_DATE('12-03-2002', 'dd-mm-yyyy'), 1300, 1);

insert into angajat

values(seq\_angajat.nextval, 'Popa','Andrei', 35, TO\_DATE('09-10-1987', 'dd-mm-yyyy'),1400, 2);

insert into angajat

values(seq\_angajat.nextval, 'Marius','Andrei', 48, TO\_DATE('10-02-1974', 'dd-mm-yyyy'), 1250, 3);

insert into angajat

values(seq\_angajat.nextval, 'Marian','Maria', 40, TO\_DATE('20-12-1982', 'dd-mm-yyyy'),1300, 4);

insert into angajat

values(seq\_angajat.nextval, 'Coman','Alexandru', 38, TO\_DATE('24-06-1984', 'dd-mm-yyyy'),1100, 5);

insert into angajat

values(seq\_angajat.nextval, 'Popescu','Alina', 28,TO\_DATE('30-09-1994', 'dd-mm-yyyy'), 1278, 6);

insert into angajat

values(seq\_angajat.nextval, 'Alexandru','Ion', 50,TO\_DATE('14-01-1972', 'dd-mm-yyyy'), 1400, 7);

insert into angajat

values(seq\_angajat.nextval, 'Oleanu','Marian', 43,TO\_DATE('08-08-1987', 'dd-mm-yyyy'), 1500, 8);

insert into angajat

values(seq\_angajat.nextval, 'Cornel','Maria', 35,TO\_DATE('24-06-1987', 'dd-mm-yyyy'), 1247, 9);

insert into angajat

values(seq\_angajat.nextval, 'Gheorghe','Andrei', 46,TO\_DATE('09-06-1976', 'dd-mm-yyyy'), 1350, 10);

insert into angajat

values(seq\_angajat.nextval, 'Stoica','Carina', 40, TO\_DATE('05-11-1982', 'dd-mm-yyyy'),2500, 11);

insert into angajat

values(seq\_angajat.nextval, 'Petru','Matei', 45,TO\_DATE('13-05-1977', 'dd-mm-yyyy'), 3000, 12);

insert into angajat

values(seq\_angajat.nextval, 'Marius','David', 37,TO\_DATE('04-12-1985', 'dd-mm-yyyy'), 2480, 13);

insert into angajat

values(seq\_angajat.nextval, 'Ion','Marian', 50,TO\_DATE('09-06-1972', 'dd-mm-yyyy'), 2998, 14);

insert into angajat

values(seq\_angajat.nextval, 'Cornel','Ionut', 34, TO\_DATE('25-10-1988', 'dd-mm-yyyy'),2000, 15);

insert into angajat

values(seq\_angajat.nextval, 'Gheorghe','Ana', 39,TO\_DATE('29-11-1983', 'dd-mm-yyyy'), 2500, 16);

insert into angajat

values(seq\_angajat.nextval, 'Pavel','George', 51, TO\_DATE('03-05-1971', 'dd-mm-yyyy'),2657, 17);

insert into angajat

values(seq\_angajat.nextval, 'Marius','Ion', 30,TO\_DATE('08-12-1992', 'dd-mm-yyyy'), 3100, 18);

insert into angajat

values(seq\_angajat.nextval, 'Agache','Maria', 40, TO\_DATE('04-10-1982', 'dd-mm-yyyy'),2150, 19);

insert into angajat

values(seq\_angajat.nextval, 'Petru','Costel', 35,TO\_DATE('04-05-1987', 'dd-mm-yyyy'), 2876, 20);

insert into angajat

values(seq\_angajat.nextval, 'Alexandru','Ioana', 25, TO\_DATE('10-10-1997', 'dd-mm-yyyy'),1500, 21);

insert into angajat

values(seq\_angajat.nextval, 'Ionescu','Carmen', 30,TO\_DATE('05-11-1992', 'dd-mm-yyyy'), 2500, 22);

insert into angajat

values(seq\_angajat.nextval, 'Pavel','Mirel', 22,TO\_DATE('10-12-2000', 'dd-mm-yyyy'), 1458, 23);

insert into angajat

values(seq\_angajat.nextval, 'Mitrel','Ionut', 28,TO\_DATE('12-09-1994', 'dd-mm-yyyy'), 2480, 24);

insert into angajat

values(seq\_angajat.nextval, 'Cordean','Marin', 32,TO\_DATE('12-02-1990', 'dd-mm-yyyy'), 1376, 25);

insert into angajat

values(seq\_angajat.nextval, 'Ionescu','Alexandra', 34, TO\_DATE('04-03-1988', 'dd-mm-yyyy'),2000, 26);

insert into angajat

values(seq\_angajat.nextval, 'Petre','Alin', 27,TO\_DATE('05-11-1995', 'dd-mm-yyyy'), 2000, 27);

insert into angajat

values(seq\_angajat.nextval, 'Cornel','Ioan', 34,TO\_DATE('10-10-1988', 'dd-mm-yyyy'), 2657, 28);

insert into angajat

values(seq\_angajat.nextval, 'Ion','Ana', 24,TO\_DATE('03-04-1998', 'dd-mm-yyyy'), 1500, 29);

insert into angajat

values(seq\_angajat.nextval, 'Popescu','Elena', 34, TO\_DATE('12-10-1988', 'dd-mm-yyyy'),2150, 30);

commit;

insert into sufera

values(1, 1);

insert into sufera

values(1, 2);

insert into sufera

values(1, 4);

insert into sufera

values(2, 1);

insert into sufera

values(2, 3);

insert into sufera

values(3, 1);

insert into sufera

values(3, 2);

insert into sufera

values(3, 3);

insert into sufera

values(3, 5);

insert into sufera

values(5, 1);

commit;

insert into se\_treateaza

values(1, 1);

insert into se\_treateaza

values(1, 2);

insert into se\_treateaza

values(1, 4);

insert into se\_treateaza

values(2, 1);

insert into se\_treateaza

values(2, 3);

insert into se\_treateaza

values(3, 1);

insert into se\_treateaza

values(3, 2);

insert into se\_treateaza

values(3, 3);

insert into se\_treateaza

values(3, 5);

insert into se\_treateaza

values(5, 1);

commit;

insert into mananca

values(1, 1);

insert into mananca

values(1, 12);

insert into mananca

values(1, 4);

insert into mananca

values(2, 10);

insert into mananca

values(2, 3);

insert into mananca

values(3, 9);

insert into mananca

values(3, 11);

insert into mananca

values(3, 2);

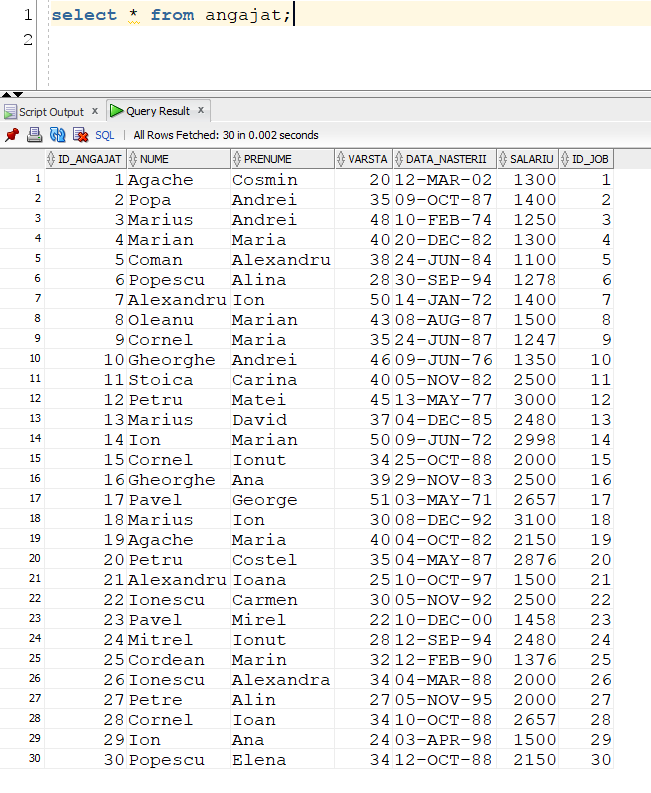
insert into mananca

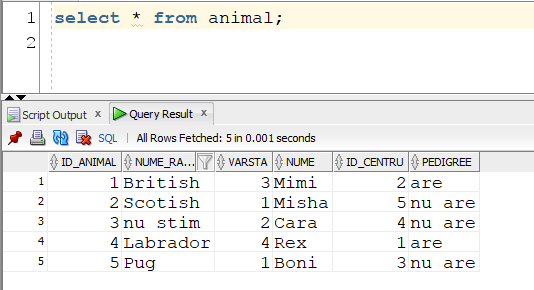
values(3, 12);

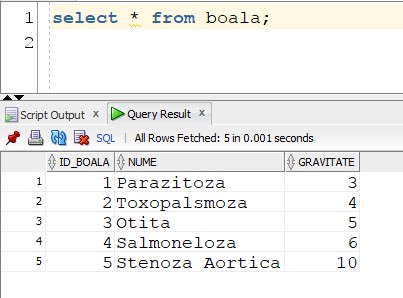
insert into mananca

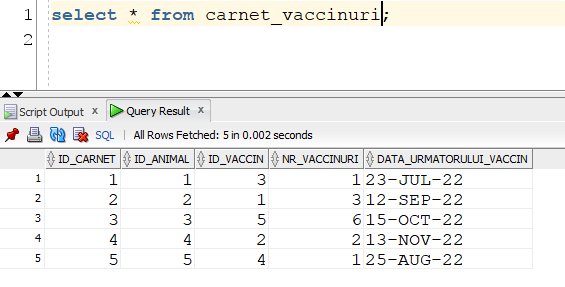
values(5, 10);

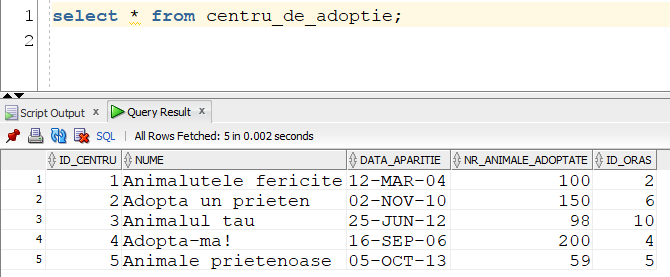
commit;

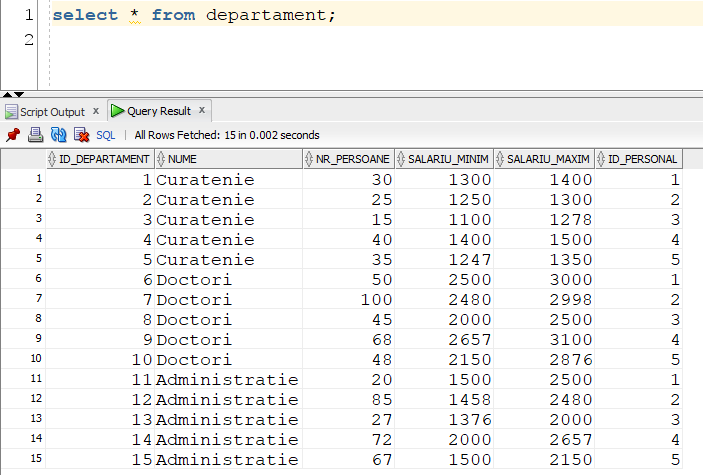


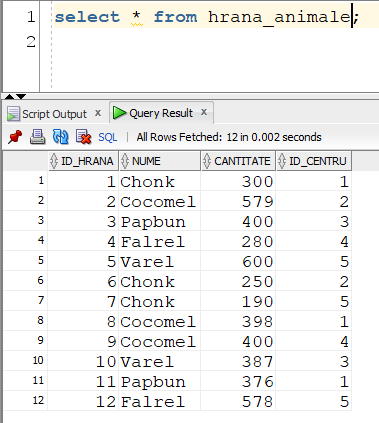


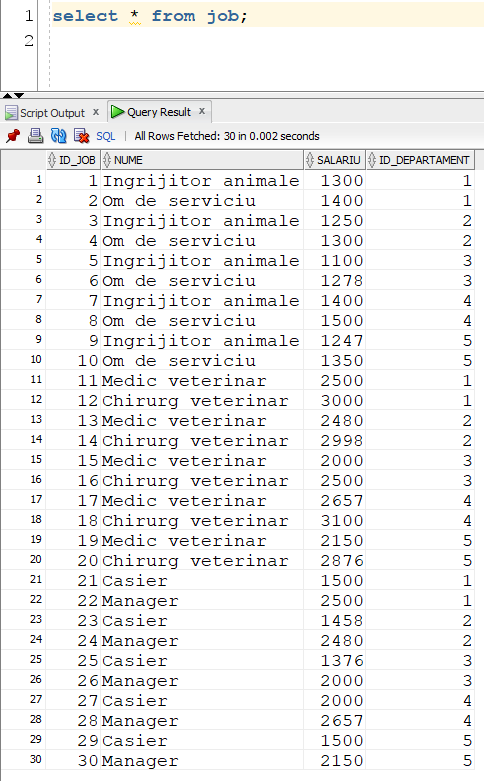


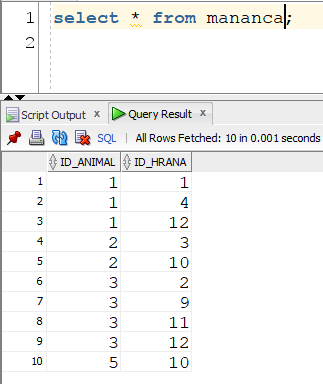


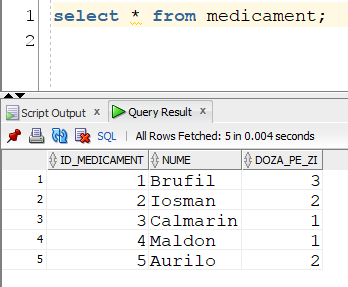


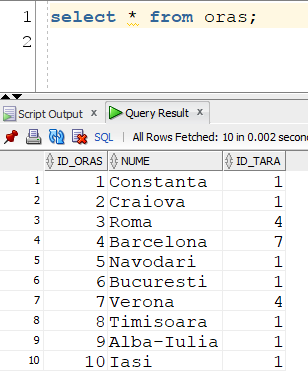


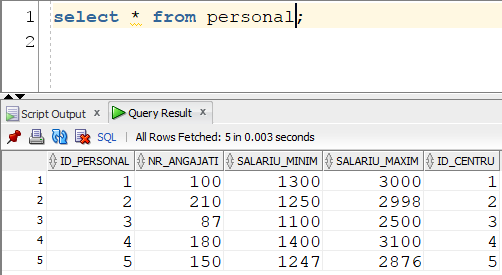


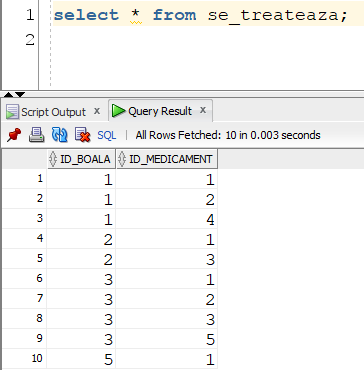


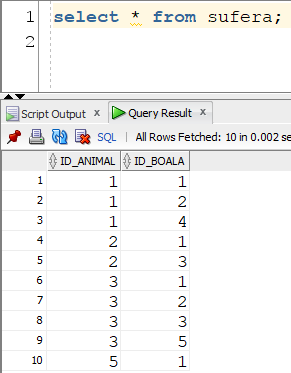


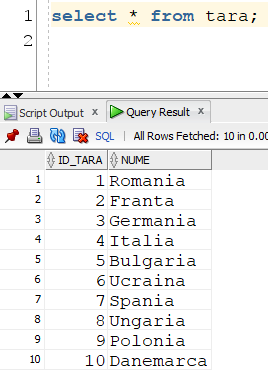


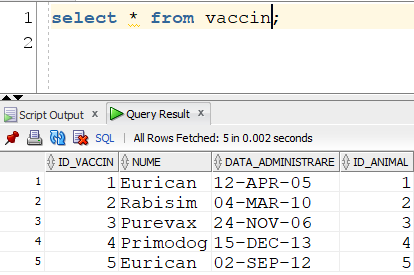












***11. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:***

***• 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)***

--1

--Afisati numele si numarul de vaccinuri facute ale animalelor ce fac parte din centrele de adoptie

--ce se afla in orasele care au incepand cu a 3-a litera, subsirul "vodari", fara nimic altceva dupa acesta.

select a.nume, cv.nr\_vaccinuri

from carnet\_vaccinuri cv join animal a on (cv.id\_animal = a.id\_animal)

join centru\_de\_adoptie ca on (a.id\_centru = ca.id\_centru)

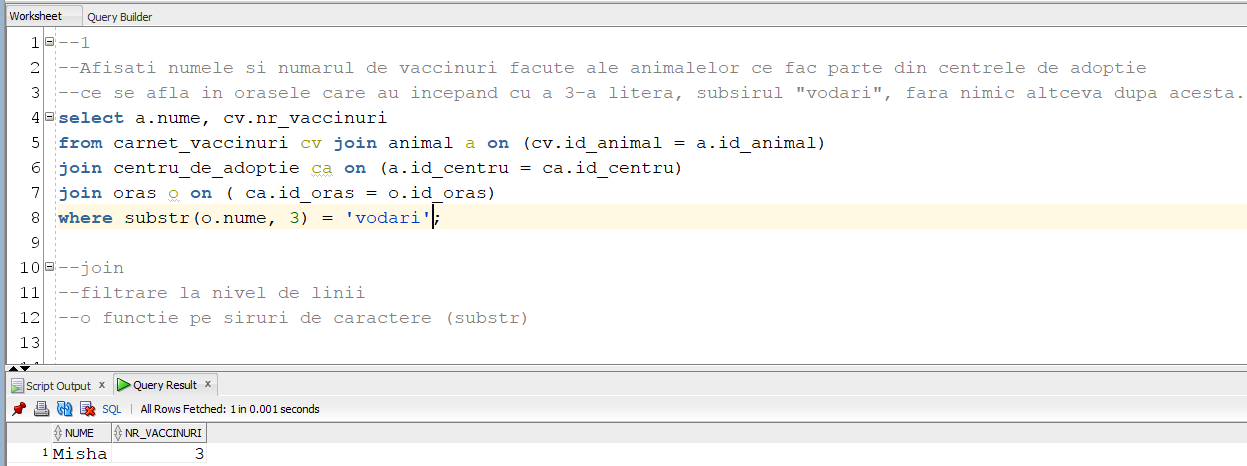
join oras o on ( ca.id\_oras = o.id\_oras)

where substr(o.nume, 3) = 'vodari';

--join

--filtrare la nivel de linii

--o functie pe siruri de caractere (substr)



--2

--Afisati numele centrelor de adoptie, orasele din care fac parte si tara, unde numele tarii scris cu litere mari este "ROMANIA"

select c.nume centru, o.nume oras, t.nume tara

from centru\_de\_adoptie c join oras o on (c.id\_oras = o.id\_oras)

join tara t on (o.id\_tara = t.id\_tara)

where t.nume in (select t2.nume

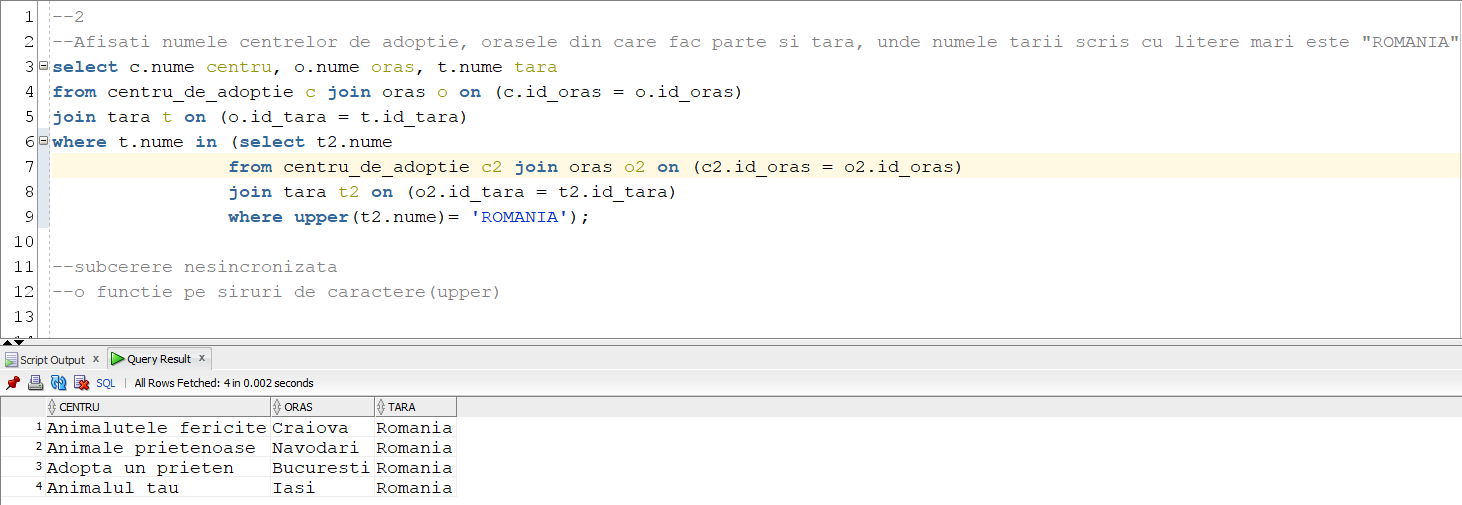
from centru\_de\_adoptie c2 join oras o2 on (c2.id\_oras = o2.id\_oras)

join tara t2 on (o2.id\_tara = t2.id\_tara)

where upper(t2.nume)= 'ROMANIA');

--subcerere nesincronizata

--o functie pe siruri de caractere(upper)



--3

--Afisati numele animalelor, numele bolii de care sufera, gravitatea bolii si cate tipuri de hrana mananca pentru

--animalele ale caror varsta este mai mica sau egala cu gravitatea bolii

select a.nume, b.nume, b.gravitate, count(h.nume)

from animal a, sufera s, boala b, mananca m, hrana\_animale h

where a.id\_animal = s.id\_animal and s.id\_boala = b.id\_boala and a.id\_animal = m.id\_animal and m.id\_hrana = h.id\_hrana and

a.varsta in (select b2.gravitate

from animal a2 join sufera s2 on (a2.id\_animal = s2.id\_animal)

join boala b2 on (s2.id\_boala = b2.id\_boala)

where b2.gravitate <= a.varsta)

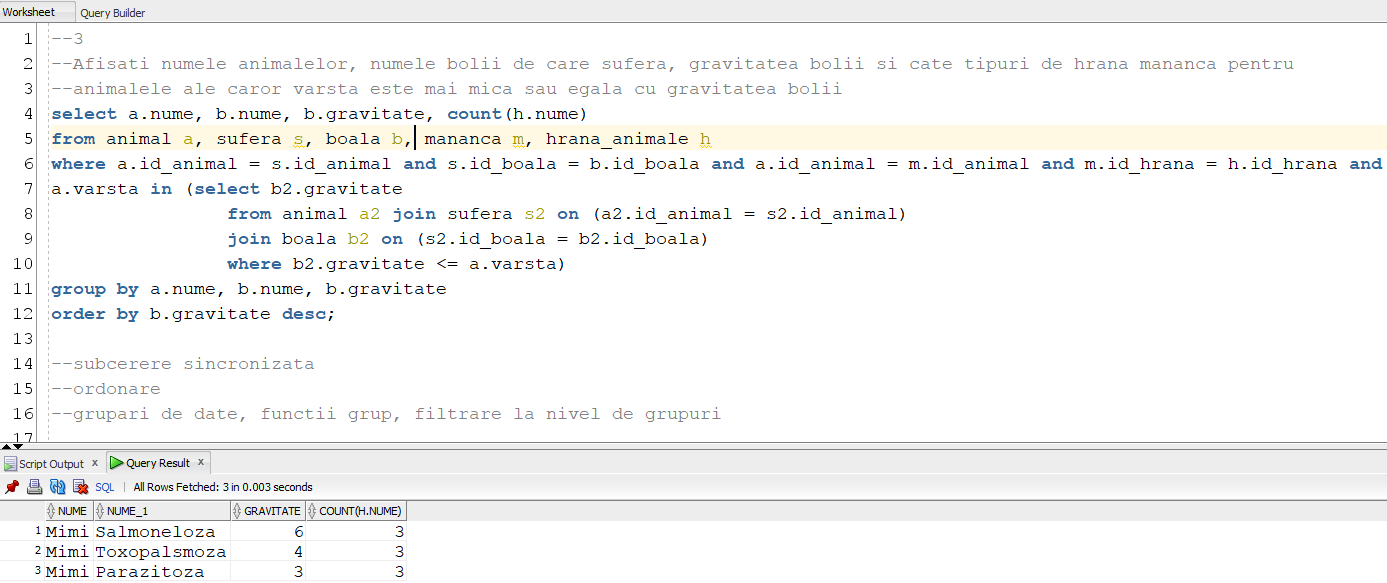
group by a.nume, b.nume, b.gravitate

order by b.gravitate desc;

--subcerere sincronizata

--ordonare

--grupari de date, functii grup, filtrare la nivel de grupuri



--4

--Afisati numele animalului(in cazul in care nu are, afiseaza "nu are nume"), data urmatorului vaccin, si un mesaj de tipul:

--"Vaccinul trebuie facut" daca data urmatorului vaccin se afla dupa doua luni si 5 zile de la data curenta,

--"Vaccinul trebuie trecut in carnetul de vaccinuri" daca data urmatorului vaccin se afla inainte de doua luni si 5 zile de la data curenta,

--"Este ziua vaccinului" in oricare alt caz. Vor fi afisate aceste lucruri pentru animalele

--care au data urmatorului vaccin dupa ultima zi a lunii curente.

select nvl(a.nume, 'nu are nume'), cv.data\_urmatorului\_vaccin,

case

when cv.data\_urmatorului\_vaccin > add\_months(sysdate +5, 2) then 'Vaccinul trebuie facut'

when cv.data\_urmatorului\_vaccin < add\_months(sysdate +5, 2) then 'Vaccinul trebuie trecut in carnetul de vaccinuri'

else 'Este ziua vaccinului'

end as mesaj

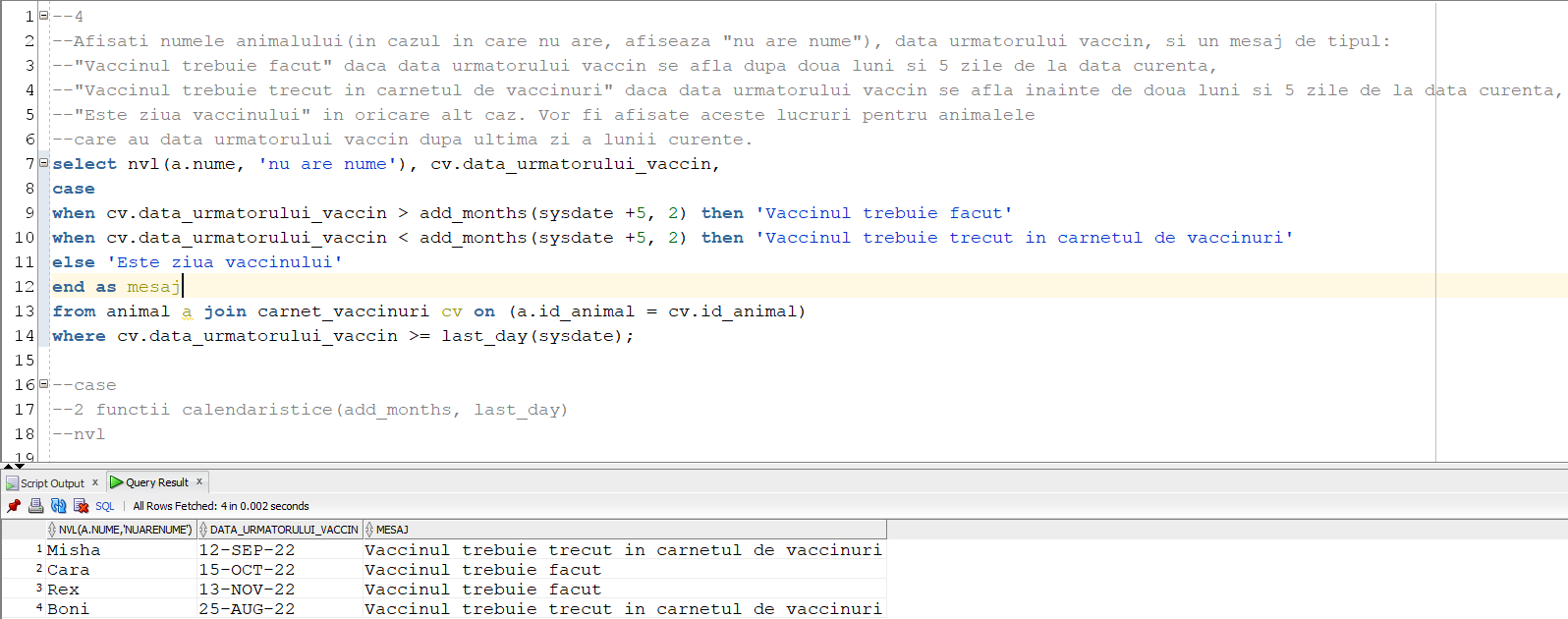
from animal a join carnet\_vaccinuri cv on (a.id\_animal = cv.id\_animal)

where cv.data\_urmatorului\_vaccin >= last\_day(sysdate);

--case

--2 functii calendaristice(add\_months, last\_day)

--nvl



--5

--Afisati numele, prenumele si un mesaj de tipul:

--"Are salariul cel mai mic" pentru angajatii care au salariul cel mai mic dintre toate joburile,

--"Are salariul cel mai mare" pentru angajatii care au salariul cel mai mare dintre toate joburile,

--"Are salariul mediu" pentru angajatii care au salariul egal cu media salariilor dintre toate joburile,

--"Nu conteaza" in celelalte cazuri. Aceste mesaje se afiseaza pentru angajatii din departamentul cu cele mai putine persoane.

--Pentru a stabili departamentul cu cele mai putine persoane, se va folosi un bloc de cerere.

with angajati\_putini(valoare) as

(select min(nr\_persoane)

from departament)

select a.nume, a.prenume, decode(a.salariu, (select min(salariu) from job), 'Are salariul cel mai mic',

(select max(salariu) from job), 'Are salariul cel mai mare',

(select avg(salariu) from job), 'Are salariul mediu', 'Nu conteaza salariul')

from angajat a join job j on a.id\_job = j.id\_job

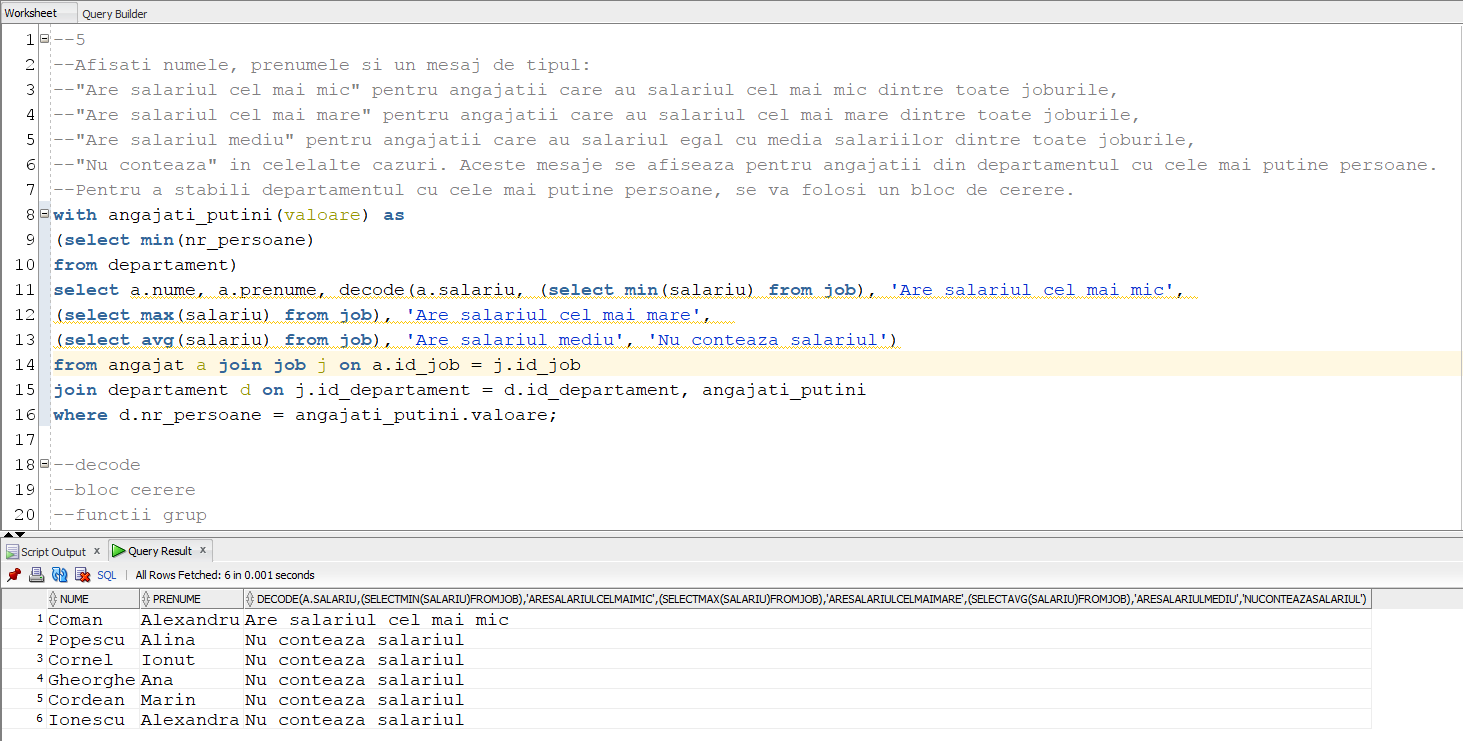
join departament d on j.id\_departament = d.id\_departament, angajati\_putini

where d.nr\_persoane = angajati\_putini.valoare;

--decode

--bloc cerere

--functii grup



***12. Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri***

--1

--Sa se majoreze salariul angajatilor care lucreaza ca doctori cu 15%.

--Sa se anuleze modificarile.

update angajat

set salariu = salariu\*1.15

where salariu in (select salariu

from angajat

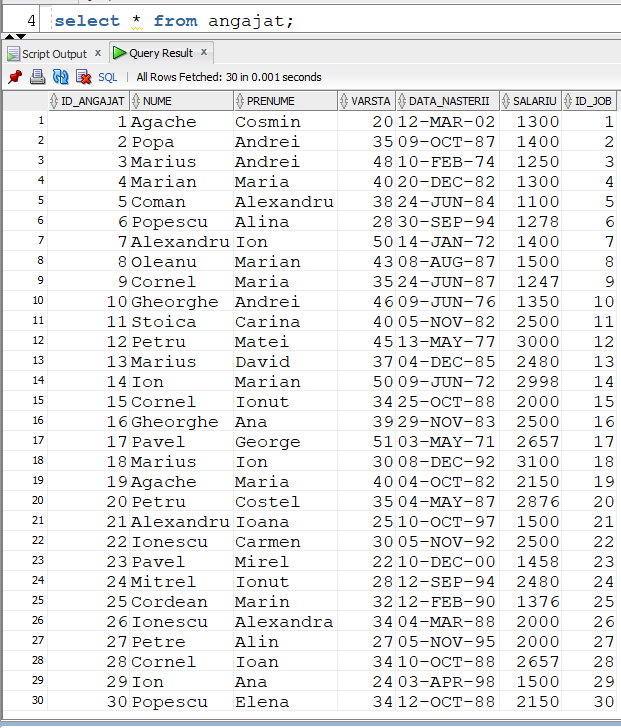
where id\_job in (select id\_job

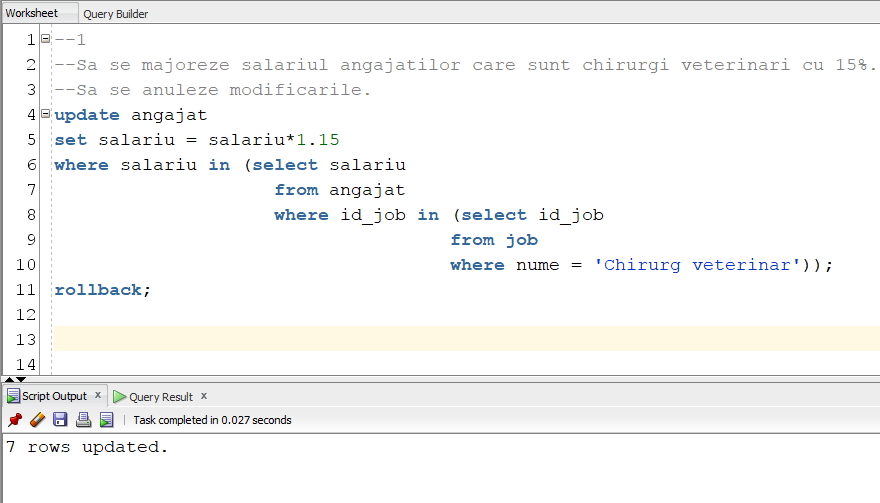
from job

where nume = 'Chirurg veterinar'));

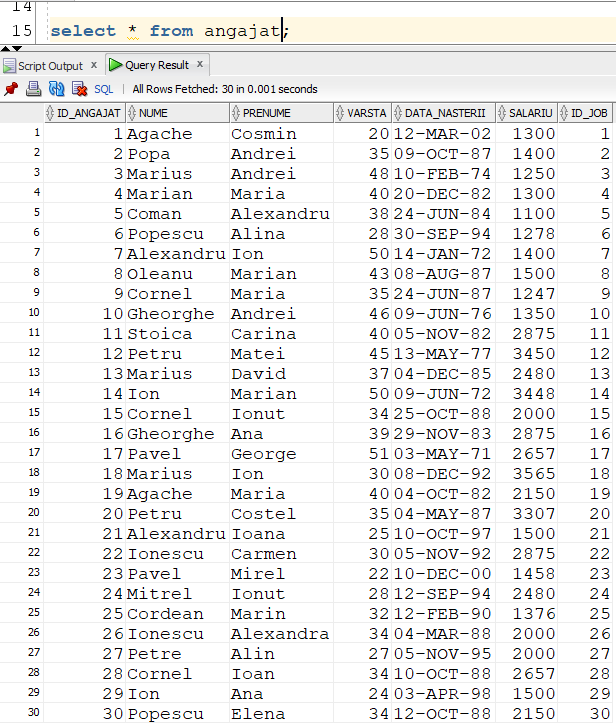
rollback;

**Tabelul inaite de update:**

****



**Tabelul dupa update:**

****

--2

--Sa se stearga inregistrarile din tabelul oras, acolo unde nu exista centre de adoptie pentru animale.

--Sa se anuleze modificarile.

delete

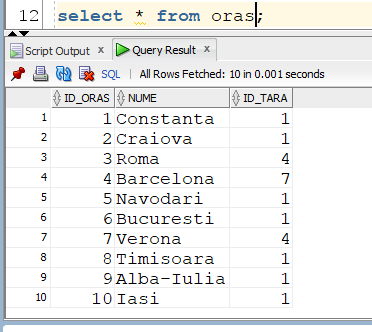
from oras

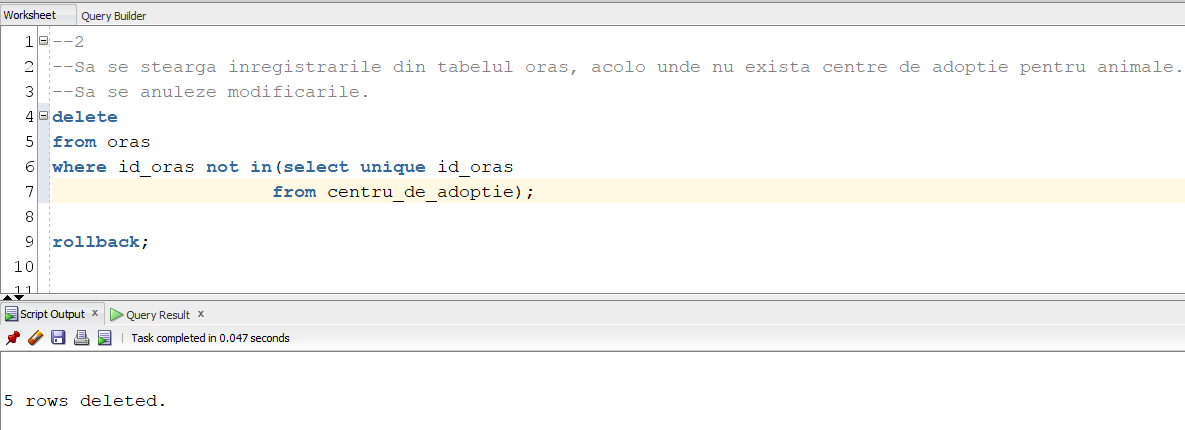
where id\_oras not in(select unique id\_oras

from centru\_de\_adoptie);

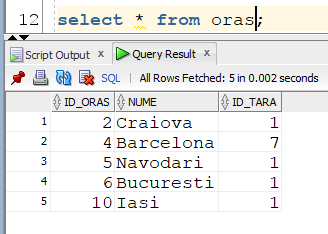
rollback;

**Tabelul inainte de update:**

****



**Tabelul dupa update:**

****

--3

--Sa se creasca doza pe zi cu una la medicamentele pentru bolile cu gravitate 10.

--Sa se anuleze modificarile.

update medicament

set doza\_pe\_zi = doza\_pe\_zi +1

where id\_medicament in (select id\_medicament

from se\_treateaza

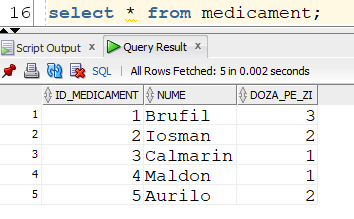
where id\_boala in (select id\_boala

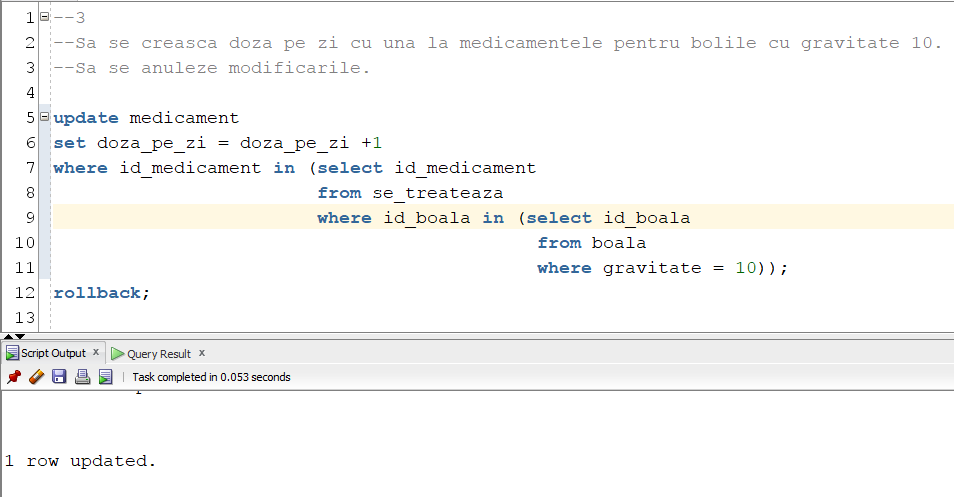
from boala

where gravitate = 10));

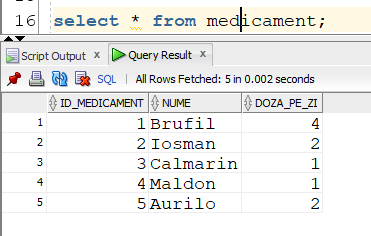
rollback;

**Tabelul inainte de update:**

****

****

**Tabelul dupa update:**

****

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

create sequence seq\_tara

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into tara

values(seq\_tara.nextval, 'Romania');

insert into tara

values(seq\_tara.nextval, 'Franta');

insert into tara

values(seq\_tara.nextval, 'Germania');

insert into tara

values(seq\_tara.nextval, 'Italia');

insert into tara

values(seq\_tara.nextval, 'Bulgaria');

insert into tara

values(seq\_tara.nextval, 'Ucraina');

insert into tara

values(seq\_tara.nextval, 'Spania');

insert into tara

values(seq\_tara.nextval, 'Ungaria');

insert into tara

values(seq\_tara.nextval, 'Polonia');

insert into tara

values(seq\_tara.nextval, 'Danemarca');

commit;

create sequence seq\_oras

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into oras

values(seq\_oras.nextval, 'Constanta', 1);

insert into oras

values(seq\_oras.nextval, 'Craiova', 1);

insert into oras

values(seq\_oras.nextval, 'Roma', 4);

insert into oras

values(seq\_oras.nextval, 'Barcelona', 7);

insert into oras

values(seq\_oras.nextval, 'Navodari', 1);

insert into oras

values(seq\_oras.nextval, 'Bucuresti', 1);

insert into oras

values(seq\_oras.nextval, 'Verona', 4);

insert into oras

values(seq\_oras.nextval, 'Timisoara', 1);

insert into oras

values(seq\_oras.nextval, 'Alba-Iulia', 1);

insert into oras

values(seq\_oras.nextval, 'Iasi', 1);

commit;

create sequence seq\_centru

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into centru\_de\_adoptie

values(seq\_centru.nextval, 'Animalutele fericite', TO\_DATE('12-03-2004', 'dd-mm-yyyy'), 100, 2);

insert into centru\_de\_adoptie

values(seq\_centru.nextval, 'Adopta un prieten', TO\_DATE('02-11-2010', 'dd-mm-yyyy'), 150, 6);

insert into centru\_de\_adoptie

values(seq\_centru.nextval, 'Animalul tau', TO\_DATE('25-06-2012', 'dd-mm-yyyy'), 98, 10);

insert into centru\_de\_adoptie

values(seq\_centru.nextval, 'Adopta-ma!', TO\_DATE('16-09-2006', 'dd-mm-yyyy'), 200, 4);

insert into centru\_de\_adoptie

values(seq\_centru.nextval, 'Animale prietenoase', TO\_DATE('05-10-2013', 'dd-mm-yyyy'), 59, 5);

insert into centru\_de\_adoptie

values(seq\_centru.nextval, 'Animalutele fericite', TO\_DATE('12-03-2004', 'dd-mm-yyyy'), 100, 2);

commit;

create sequence seq\_hrana

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into hrana\_animale

values(seq\_hrana.nextval, 'Chonk', 300, 1);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Cocomel', 579, 2);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Papbun', 400, 3);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Falrel', 280, 4);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Varel', 600, 5);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Chonk', 250, 2);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Chonk', 190, 5);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Cocomel', 398, 1);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Cocomel', 400, 4);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Varel', 387, 3);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Papbun', 376, 1);

insert into hrana\_animale

values(seq\_hrana.nextval, 'Falrel', 578, 5);

commit;

create sequence seq\_animal

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into animal

values(seq\_animal.nextval, 'British', 3, 'Mimi', 2, 'are');

insert into animal

values(seq\_animal.nextval, 'Scotish', 1, 'Misha', 5, 'nu are');

insert into animal

values(seq\_animal.nextval, 'nu stim', 2, 'Cara', 4,'nu are');

insert into animal

values(seq\_animal.nextval, 'Labrador', 4, 'Rex', 1,'are');

insert into animal

values(seq\_animal.nextval, 'Pug', 1, 'Boni', 3,'nu are');

commit;

create sequence seq\_vaccin

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into vaccin

values(seq\_vaccin.nextval, 'Eurican', TO\_DATE('12-04-2005', 'dd-mm-yyyy'),1);

insert into vaccin

values(seq\_vaccin.nextval, 'Rabisim', TO\_DATE('04-03-2010', 'dd-mm-yyyy'),2);

insert into vaccin

values(seq\_vaccin.nextval, 'Purevax', TO\_DATE('24-11-2006', 'dd-mm-yyyy'),3);

insert into vaccin

values(seq\_vaccin.nextval, 'Primodog', TO\_DATE('15-12-2013', 'dd-mm-yyyy'),4);

insert into vaccin

values(seq\_vaccin.nextval, 'Eurican', TO\_DATE('02-09-2012', 'dd-mm-yyyy'),5);

commit;

create sequence seq\_carnet

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into carnet\_vaccinuri

values(seq\_carnet.nextval, 1, 3, 1, TO\_DATE('23-07-2022', 'dd-mm-yyyy'));

insert into carnet\_vaccinuri

values(seq\_carnet.nextval, 2, 1, 3, TO\_DATE('12-09-2022', 'dd-mm-yyyy'));

insert into carnet\_vaccinuri

values(seq\_carnet.nextval, 3, 5, 6, TO\_DATE('15-10-2022', 'dd-mm-yyyy'));

insert into carnet\_vaccinuri

values(seq\_carnet.nextval, 4, 2, 2, TO\_DATE('13-11-2022', 'dd-mm-yyyy'));

insert into carnet\_vaccinuri

values(seq\_carnet.nextval, 5, 4, 1, TO\_DATE('25-08-2022', 'dd-mm-yyyy'));

commit;

create sequence seq\_boala

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into boala

values(seq\_boala.nextval, 'Parazitoza', 3);

insert into boala

values(seq\_boala.nextval, 'Salmoneloza', 6);

insert into boala

values(seq\_boala.nextval, 'Stenoza Aortica', 10);

insert into boala

values(seq\_boala.nextval, 'Toxopalsmoza', 4);

insert into boala

values(seq\_boala.nextval, 'Otita', 5);

commit;

create sequence seq\_medicament

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into medicament

values(seq\_medicament.nextval, 'Brufil', 3);

insert into medicament

values(seq\_medicament.nextval, 'Iosman', 2);

insert into medicament

values(seq\_medicament.nextval, 'Calmarin', 1);

insert into medicament

values(seq\_medicament.nextval, 'Maldon', 1);

insert into medicament

values(seq\_medicament.nextval, 'Aurilo', 2);

commit;

create sequence seq\_personal

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into personal

values(seq\_personal.nextval, 100, 1300, 3000, 1);

insert into personal

values(seq\_personal.nextval, 210, 1250, 2998, 2);

insert into personal

values(seq\_personal.nextval, 87, 1100, 2500, 3);

insert into personal

values(seq\_personal.nextval, 180, 1400, 3100, 4);

insert into personal

values(seq\_personal.nextval, 150, 1247, 2876, 5);

commit;

create sequence seq\_departament

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into departament

values(seq\_departament.nextval, 'Curatenie',30, 1300, 1400, 1);

insert into departament

values(seq\_departament.nextval, 'Curatenie',25, 1250, 1300, 2);

insert into departament

values(seq\_departament.nextval, 'Curatenie',15, 1100, 1278, 3);

insert into departament

values(seq\_departament.nextval, 'Curatenie',40, 1400, 1500, 4);

insert into departament

values(seq\_departament.nextval, 'Curatenie',35, 1247, 1350, 5);

insert into departament

values(seq\_departament.nextval, 'Curatenie',30, 1300, 1400, 1);

insert into departament

values(seq\_departament.nextval, 'Doctori',50, 2500, 3000, 1);

insert into departament

values(seq\_departament.nextval, 'Doctori',100, 2480, 2998, 2);

insert into departament

values(seq\_departament.nextval, 'Doctori',45, 2000, 2500, 3);

insert into departament

values(seq\_departament.nextval, 'Doctori',68, 2657, 3100, 4);

insert into departament

values(seq\_departament.nextval, 'Doctori',48, 2150, 2876, 5);

insert into departament

values(seq\_departament.nextval, 'Administratie',20, 1500, 2500, 1);

insert into departament

values(seq\_departament.nextval, 'Administratie',85, 1458, 2480, 2);

insert into departament

values(seq\_departament.nextval, 'Administratie',27, 1376, 2000, 3);

insert into departament

values(seq\_departament.nextval, 'Administratie',72, 2000, 2657, 4);

insert into departament

values(seq\_departament.nextval, 'Administratie',67, 1500, 2150, 5);

commit;

create sequence seq\_job

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into job

values(seq\_job.nextval, 'Ingrijitor animale',1300, 1);

insert into job

values(seq\_job.nextval, 'Om de serviciu',1400, 1);

insert into job

values(seq\_job.nextval, 'Ingrijitor animale',1250, 2);

insert into job

values(seq\_job.nextval, 'Om de serviciu',1300, 2);

insert into job

values(seq\_job.nextval, 'Ingrijitor animale',1100, 3);

insert into job

values(seq\_job.nextval, 'Om de serviciu',1278, 3);

insert into job

values(seq\_job.nextval, 'Ingrijitor animale',1400, 4);

insert into job

values(seq\_job.nextval, 'Om de serviciu',1500, 4);

insert into job

values(seq\_job.nextval, 'Ingrijitor animale',1247, 5);

insert into job

values(seq\_job.nextval, 'Om de serviciu',1350, 5);

insert into job

values(seq\_job.nextval, 'Medic veterinar',2500, 6);

insert into job

values(seq\_job.nextval, 'Chirurg veterinar',3000, 6);

insert into job

values(seq\_job.nextval, 'Medic veterinar',2480, 7);

insert into job

values(seq\_job.nextval, 'Chirurg veterinar',2998, 7);

insert into job

values(seq\_job.nextval, 'Medic veterinar',2000, 8);

insert into job

values(seq\_job.nextval, 'Chirurg veterinar',2500, 8);

insert into job

values(seq\_job.nextval, 'Medic veterinar',2657, 9);

insert into job

values(seq\_job.nextval, 'Chirurg veterinar',3100, 9);

insert into job

values(seq\_job.nextval, 'Medic veterinar',2150, 10);

insert into job

values(seq\_job.nextval, 'Chirurg veterinar',2876, 10);

insert into job

values(seq\_job.nextval, 'Casier',1500, 11);

insert into job

values(seq\_job.nextval, 'Manager',2500, 11);

insert into job

values(seq\_job.nextval, 'Casier',1458, 12);

insert into job

values(seq\_job.nextval, 'Manager',2480, 12);

insert into job

values(seq\_job.nextval, 'Casier',1376, 13);

insert into job

values(seq\_job.nextval, 'Manager',2000, 13);

insert into job

values(seq\_job.nextval, 'Casier',2000, 14);

insert into job

values(seq\_job.nextval, 'Manager',2657, 14);

insert into job

values(seq\_job.nextval, 'Casier',1500, 15);

insert into job

values(seq\_job.nextval, 'Manager',2150, 15);

commit;

create sequence seq\_angajat

increment by 1

start with 1

maxvalue 10000

nocycle;

insert into angajat

values(seq\_angajat.nextval, 'Agache','Cosmin', 20,TO\_DATE('12-03-2002', 'dd-mm-yyyy'), 1300, 1);

insert into angajat

values(seq\_angajat.nextval, 'Popa','Andrei', 35, TO\_DATE('09-10-1987', 'dd-mm-yyyy'),1400, 2);

insert into angajat

values(seq\_angajat.nextval, 'Marius','Andrei', 48, TO\_DATE('10-02-1974', 'dd-mm-yyyy'), 1250, 3);

insert into angajat

values(seq\_angajat.nextval, 'Marian','Maria', 40, TO\_DATE('20-12-1982', 'dd-mm-yyyy'),1300, 4);

insert into angajat

values(seq\_angajat.nextval, 'Coman','Alexandru', 38, TO\_DATE('24-06-1984', 'dd-mm-yyyy'),1100, 5);

insert into angajat

values(seq\_angajat.nextval, 'Popescu','Alina', 28,TO\_DATE('30-09-1994', 'dd-mm-yyyy'), 1278, 6);

insert into angajat

values(seq\_angajat.nextval, 'Alexandru','Ion', 50,TO\_DATE('14-01-1972', 'dd-mm-yyyy'), 1400, 7);

insert into angajat

values(seq\_angajat.nextval, 'Oleanu','Marian', 43,TO\_DATE('08-08-1987', 'dd-mm-yyyy'), 1500, 8);

insert into angajat

values(seq\_angajat.nextval, 'Cornel','Maria', 35,TO\_DATE('24-06-1987', 'dd-mm-yyyy'), 1247, 9);

insert into angajat

values(seq\_angajat.nextval, 'Gheorghe','Andrei', 46,TO\_DATE('09-06-1976', 'dd-mm-yyyy'), 1350, 10);

insert into angajat

values(seq\_angajat.nextval, 'Stoica','Carina', 40, TO\_DATE('05-11-1982', 'dd-mm-yyyy'),2500, 11);

insert into angajat

values(seq\_angajat.nextval, 'Petru','Matei', 45,TO\_DATE('13-05-1977', 'dd-mm-yyyy'), 3000, 12);

insert into angajat

values(seq\_angajat.nextval, 'Marius','David', 37,TO\_DATE('04-12-1985', 'dd-mm-yyyy'), 2480, 13);

insert into angajat

values(seq\_angajat.nextval, 'Ion','Marian', 50,TO\_DATE('09-06-1972', 'dd-mm-yyyy'), 2998, 14);

insert into angajat

values(seq\_angajat.nextval, 'Cornel','Ionut', 34, TO\_DATE('25-10-1988', 'dd-mm-yyyy'),2000, 15);

insert into angajat

values(seq\_angajat.nextval, 'Gheorghe','Ana', 39,TO\_DATE('29-11-1983', 'dd-mm-yyyy'), 2500, 16);

insert into angajat

values(seq\_angajat.nextval, 'Pavel','George', 51, TO\_DATE('03-05-1971', 'dd-mm-yyyy'),2657, 17);

insert into angajat

values(seq\_angajat.nextval, 'Marius','Ion', 30,TO\_DATE('08-12-1992', 'dd-mm-yyyy'), 3100, 18);

insert into angajat

values(seq\_angajat.nextval, 'Agache','Maria', 40, TO\_DATE('04-10-1982', 'dd-mm-yyyy'),2150, 19);

insert into angajat

values(seq\_angajat.nextval, 'Petru','Costel', 35,TO\_DATE('04-05-1987', 'dd-mm-yyyy'), 2876, 20);

insert into angajat

values(seq\_angajat.nextval, 'Alexandru','Ioana', 25, TO\_DATE('10-10-1997', 'dd-mm-yyyy'),1500, 21);

insert into angajat

values(seq\_angajat.nextval, 'Ionescu','Carmen', 30,TO\_DATE('05-11-1992', 'dd-mm-yyyy'), 2500, 22);

insert into angajat

values(seq\_angajat.nextval, 'Pavel','Mirel', 22,TO\_DATE('10-12-2000', 'dd-mm-yyyy'), 1458, 23);

insert into angajat

values(seq\_angajat.nextval, 'Mitrel','Ionut', 28,TO\_DATE('12-09-1994', 'dd-mm-yyyy'), 2480, 24);

insert into angajat

values(seq\_angajat.nextval, 'Cordean','Marin', 32,TO\_DATE('12-02-1990', 'dd-mm-yyyy'), 1376, 25);

insert into angajat

values(seq\_angajat.nextval, 'Ionescu','Alexandra', 34, TO\_DATE('04-03-1988', 'dd-mm-yyyy'),2000, 26);

insert into angajat

values(seq\_angajat.nextval, 'Petre','Alin', 27,TO\_DATE('05-11-1995', 'dd-mm-yyyy'), 2000, 27);

insert into angajat

values(seq\_angajat.nextval, 'Cornel','Ioan', 34,TO\_DATE('10-10-1988', 'dd-mm-yyyy'), 2657, 28);

insert into angajat

values(seq\_angajat.nextval, 'Ion','Ana', 24,TO\_DATE('03-04-1998', 'dd-mm-yyyy'), 1500, 29);

insert into angajat

values(seq\_angajat.nextval, 'Popescu','Elena', 34, TO\_DATE('12-10-1988', 'dd-mm-yyyy'),2150, 30);

commit;

***14. Crearea unei vizualizări compuse. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.***

Vizualizare care contine numele, prenumele si salariul unui angajat, si salariul mediu al jobului sau.

**Crearea unei vizualizari compuse:**

create view angajat\_salariu\_mediu\_sal

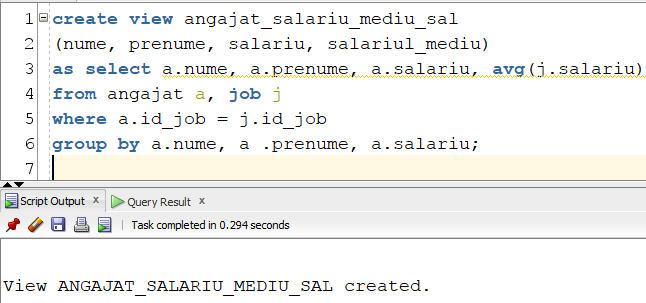
(nume, prenume, salariu, salariul\_mediu)

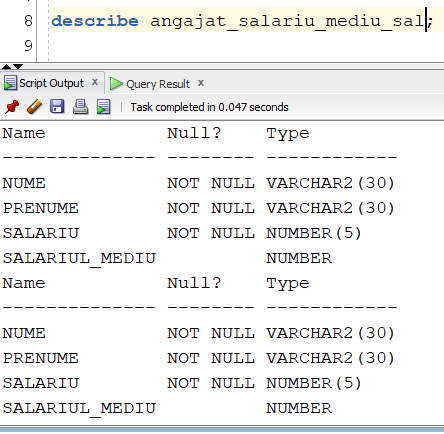
as select a.nume, a.prenume, a.salariu, avg(j.salariu)

from angajat a, job j

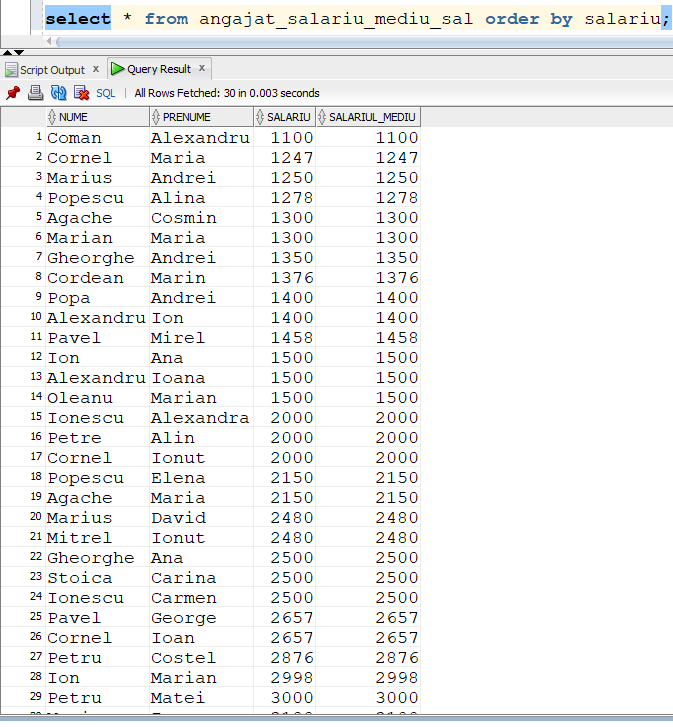
where a.id\_job = j.id\_job

group by a.nume, a .prenume, a.salariu;

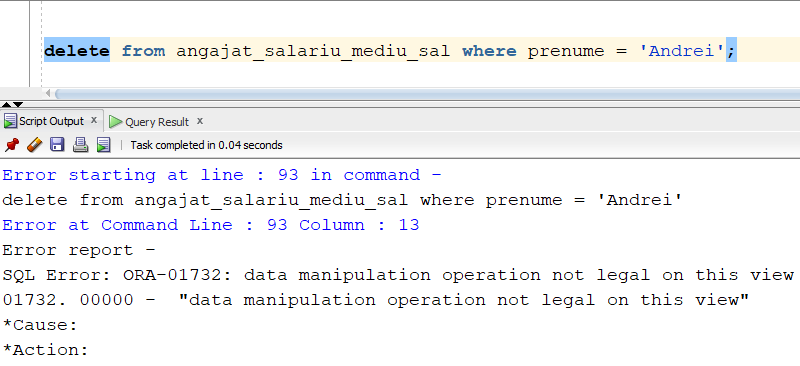




**Operatie LMD permisa:**

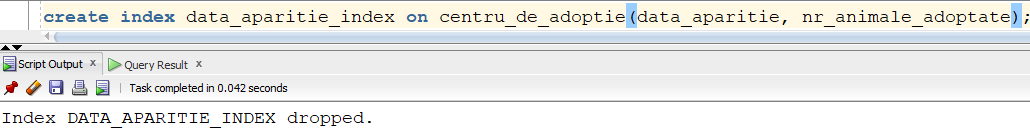


**Operatie LMD nepermisa:**



***15. Crearea unui index care să optimizeze o cerere de tip căutare cu 2 criterii. Specificați cererea.***

**Creare index:**

****

**Cererea:**

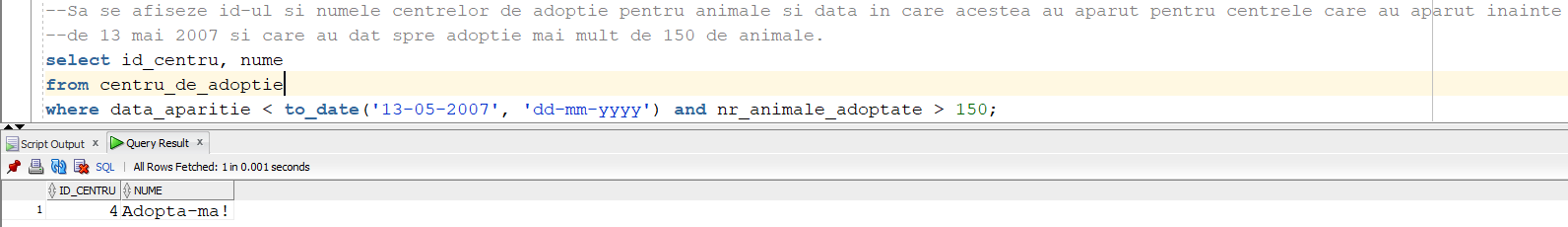
--Sa se afiseze id-ul si numele centrelor de adoptie pentru animale si data in care acestea au aparut pentru centrele care au aparut inainte de 13 mai 2007

--si care au dat spre adoptie mai mult de 150 de animale.

select id\_centru, nume

from centru\_de\_adoptie

where data\_aparitie < to\_date('13-05-2007', 'dd-mm-yyyy') and nr\_animale\_adoptate > 150;



***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.***

**Outer-join:**

--Pentru fiecare angajat, sa se afiseze numele, prenumele, departamentul si centrul de adoptie pentru animale la care lucreaza. In cazul in care exista

--job-uri fara angajati, se va afisa mesajul "Post liber".

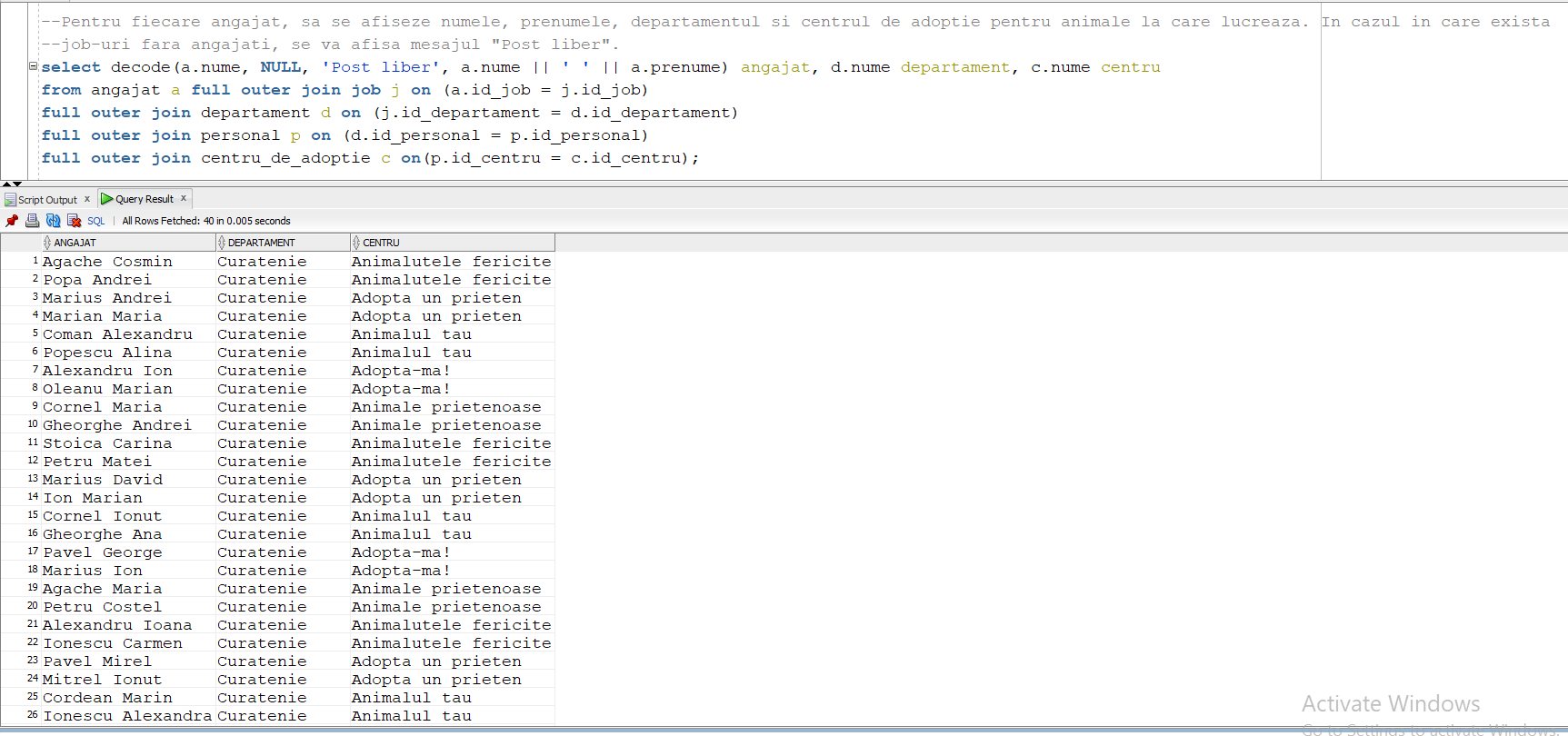
select decode(a.nume, NULL, 'Post liber', a.nume || ' ' || a.prenume) angajat, d.nume departament, c.nume centru

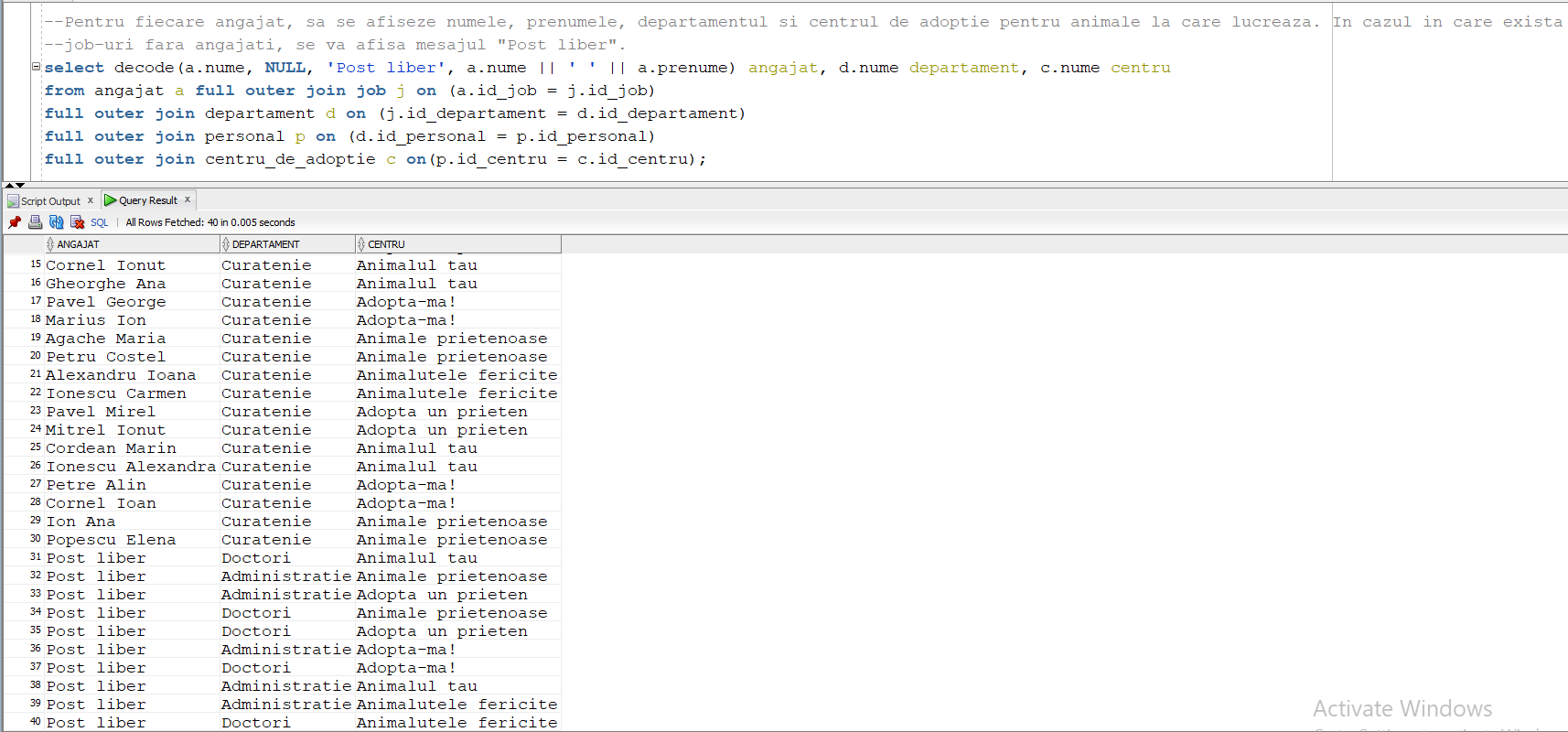
from angajat a full outer join job j on (a.id\_job = j.id\_job)

full outer join departament d on (j.id\_departament = d.id\_departament)

full outer join personal p on (d.id\_personal = p.id\_personal)

full outer join centru\_de\_adoptie c on(p.id\_centru = c.id\_centru);





**Division:**

**1.**

--Sa se afiseze id-ul si numele animalului care are cel putin

--aceleasi boli precum animalul cu id-ul 2;

select \* from sufera;

select a.id\_animal, a.nume

from animal a join sufera s on (a.id\_animal = s.id\_animal)

where s.id\_boala in (select id\_boala

from sufera

where id\_animal = 2)

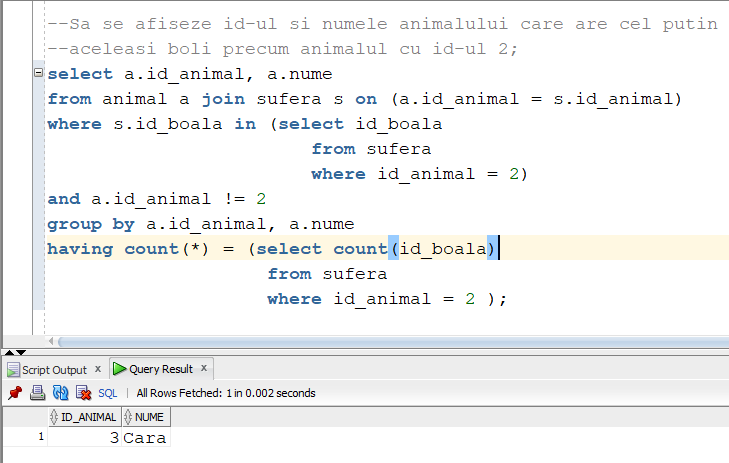
and a.id\_animal != 2

group by a.id\_animal, a.nume

having count(\*) = (select count(id\_boala)

from sufera

where id\_animal = 2 );



**2.**

--Sa se afiseze id-ul si numele animalului care are exact

--aceleasi boli precum animalul cu id-ul 5.

select a.id\_animal, a.nume

from animal a join sufera s on (a.id\_animal = s.id\_animal)

where s.id\_boala in (select id\_boala

from sufera

where id\_animal = 5)

and a.id\_animal != 5

group by a.id\_animal, a.nume

having count(\*) = (select count(id\_boala)

from sufera

where id\_animal = 5 )

minus

select a.id\_animal, a.nume

from animal a join sufera s on (a.id\_animal = s.id\_animal)

where s.id\_boala not in (select id\_boala

from sufera

where id\_animal = 5);

