

# Proiect Baze de date

## Managementul unei ferme

Pîrvulescu Daria Maria

Grupa 151

### Cuprins:

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.
2. Prezentarea constrângerilor.
3. Descrierea entităților, incluzând precizarea cheii primare.
4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.
5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.
6. Diagrama Entitate-Relație.
7. Diagrama Conceptuală.
8. Schemele relaționale corespunzătoare diagramei conceptuale.
9. Realizarea normalizării până la forma normală 3 (FN1-FN3).
10. Secvențele ce vor fi utilizate în inserarea înregistrărilor în tabele.
11. Crearea tabelelor în SQL și inserarea de date coerente. (+ print screen)
12. 5 cereri SQL complexe. (+ print screen)
13. 3 operații de actualizare și de ștergere a datelor utilizând subcereri. (+ print screen)
14. Cerere ce utilizează operația outer-join, division și analiza top-n.
15. Optimizarea unei cereri (expresie algebrică, arbore algebraic și limbaj SQL).
16. a. Realizarea normalizării BCNF, FN4, FN5.  
b. Aplicarea denormalizării, justificând necesitatea acesteia.

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

Modelul de date va gestiona informațiile legate de organizarea și funcționarea unei ferme de animale și plante, din perspectiva comercializării produselor obținute. La această fermă există angajați care se ocupă de buna desfășurare a activităților. Aceștia se împart în mai multe categorii: fermieri (ce au grijă de animale și plante), contabili (ce se ocupă de finanțele fermei), șoferi (ce distribuie produsele) și comercianți (ce asigură buna funcționare a proceselor de vânzare-cumpărare).

Animalele sunt bine îngrijite de către fermieri, care au câte o specializare pentru a maximiza calitatea produselor obținute. Fiecare tip de animal are câte o cușcă special creată pentru a beneficia de cele mai bune condiții. Animalele de tipuri diferite nu împart aceeași cușcă din motive de siguranță. Animalele sunt hrănite cu plantele crescute chiar la fermă, pentru a se asigura sănătatea lor fizică.

Plantele sunt puse pe terenuri diferite în funcție de tip, însă mai multe plante pot fi plantate pe același teren în funcție de anotimp. Fiecare teren are un tip de sol.

Ferma este construită într-o locație care facilitează ajungerea rapidă a camioanelor de marfă, care sunt conduse de șoferii noștri. Camioanele sunt personalizate, fiecare șofer poate conduce doar propriul său camion.

Utilizatorii aplicației plasează comenzi. Comenzile sunt numeroase și sunt contorizate de comercianții noștri. Toate activitățile coordonate de fermă sunt organizate de contabili, după un calendar bine structurat.

Toți angajații noștri au un istoric unic de muncă, prin care se contorizează productivitatea în intervalul de timp în care lucrează.

## 2.) Prezentarea constrângerilor:

- Angajații se împart în patru subcategorii: fermieri, șoferi, contabili, comercianți.
- Unul sau mai mulți fermieri se ocupă de unul sau mai multe animale și/sau plante. Fermierii pot avea una sau mai multe specializări, iar într-o specializare se pot înscrie mai mulți fermieri.
- Animalele consumă plante. O plantă poate fi consumată de mai multe animale sau deloc, iar un animal poate consuma una sau mai multe plante.
- Un tip de plantă crește pe un singur tip de teren, dar pe un teren pot crește mai multe tipuri de plante.
- Un teren are unul sau mai multe tipuri de soluri, iar solurile pot fi prezente în mai multe tipuri de terenuri.
- Un șofer conduce un singur camion de marfă, iar un camion de marfă este condus de un singur șofer.
- Unul sau mai mulți contabili organizează una sau mai multe activități.
- Comercianții țin evidența comenzilor. O comandă poate fi gestionată de către un singur comerciant, dar un comerciant poate avea evidența uneia sau mai multor comenzi.

- Una sau mai multe comenzi sunt plasate de către un singur utilizator.
- Fiecare angajat deține un istoric unic.

### 3.) Descrierea entităților, incluzând precizarea cheii primare.

#### ENTITĂȚI

Pentru modelul de date referitor la organizarea și funcționarea unei ferme de animale și plante, din perspectiva comercializării produselor obținute, structurile ANIMAL, CUȘCĂ, PLANTA, TEREN, SOL, ANGAJAT, FERMIER, ȘOFER, CONTABIL, COMERCIANT, SPECIALIZARE, CAMION DE MARFA, UTILIZATOR, COMANDA, ACTIVITATE, ISTORIC reprezintă entități.

Vom prezenta entitățile modelului de date, dând o descriere completă a fiecăreia. De asemenea, pentru fiecare entitate se va preciza cheia primară.

Toate entitățile care vor fi prezentate sunt independente, cu excepția entităților dependente ANIMAL, PLANTA, TEREN, COMANDA și a subentităților FERMIER, ȘOFER, COMERCIANT, CONTABIL.

ANGAJAT = persoană fizică, angajată la fermă care are grijă de buna desfășurare a activităților, printre care se numără îngrijirea animalelor și a plantelor și contorizarea și distribuirea comenzilor. Cheia primară a acestei entități este id\_angajat.

FERMIER = subentitate a entității ANGAJAT, persoană fizică ce lucrează la ferma și are grija atât de animale cât și de plante, urmând câte o specializare. Cheia primară a acestei entități este id\_angajat.

ȘOFER = subentitate a entității ANGAJAT, persoană fizică ce lucrează la ferma și distribuie produsele cu ajutorul camioanelor. Cheia primară a acestei entități este id\_angajat.

CONTABIL = subentitate a entității ANGAJAT, persoană fizică ce lucrează la ferma prin planificarea activităților. Cheia primară a acestei entități este id\_angajat.

COMERCIANT = subentitate a entității ANGAJAT, persoană fizică ce lucrează la fermă și are grijă de comenzile plasate. Cheia primară a acestei entități este id\_angajat.

ANIMAL = entitate dependentă de CUȘCĂ, ce reține datele stocate pentru animalele din ferma, cum ar fi rasa și numărul acestora pentru fiecare specie. Cheia primară a entității este compusă din id\_cusca și id\_animal.

CUȘCĂ = obiect fizic și utilizat pentru a găzdui animalele din ferme, astfel încât attributele sale includ dimensiunile cuștii și capacitatea acesteia. Cheia primară a entității este

formată din id\_cusca, care este un identificator unic pentru fiecare cușcă înregistrată în sistemul de gestiune al fermei.

PLANTA = entitate dependentă de entitatea TEREN care este la rândul ei tot entitate dependentă de SOL, ce conține informații despre plantele crescute la ferma cum ar fi cantitatea și calitatea. Cheia primară a entității este compusă din cele 3 chei: id\_planta, id\_teren, id\_sol.

TEREN= entitate dependentă de SOL, ce reține lungimea terenurilor pe care sunt cultivate plantele din ferma. Cheia primară a acestei entități este compusă din id\_sol și id\_teren.

SOL= entitate independentă ce stochează informații despre densitatea solului pe care ferma îl folosește. Cheia primară a acestei entități este id\_sol.

COMANDA= entitate dependentă de UTILIZATOR, ce stochează datele comenzilor făcute de cumpărători, cum ar fi adresa și numărul de telefon. Cheia primară a acestei entități este compusă din id\_comanda și id\_user.

UTILIZATOR= persoană fizică sau juridică, ce cumpără bunurile puse la dispoziție de platformă online a fermei. Cheia primară a acestei entități este id\_user.

SPECIALIZARE= specializarea pe care fiecare fermier o are în funcție de mediul de lucru în care acesta își desfășoară activitatea. Cheia primară a acestei entități este id\_specializare.

CAMION DE MARFA= vehicul prin care se asigură transportul comenzilor și a materialelor necesare bunei desfășurări ale activităților de la ferma. Cheia primară a acestei entități este id\_camion.

ISTORIC= entitate ce reține date despre angajați, cum ar fi data angajării sau productivitatea. Cheia primară a acestei entități este id\_istoric.

ACTIVITATE= entitatea ce schematizează toate activitățile petrecute la ferma, fiind contorizată de către contabili. Cheia primară a acestei entități este id\_activitate.

#### 4) Descrierea relațiilor, incluzând precizarea cardinalității acestora.

### RELAȚII

Vom prezenta relațiile modelului de date, dând o descriere completă a fiecăreia. Pentru fiecare relație se va preciza cardinalitatea minimă și maximă.

ANIMAL\_sta\_în\_CUȘCĂ = relație care leagă entitățile ANIMAL și CUȘCĂ, reprezentând locul în care animalele trăiesc. Relația are o cardinalitate minimă de 0:1 (o rasă de animale trebuie să stea în cel puțin o cușcă, dar o cușcă nu trebuie să fie populată tot

timpul de o rasă de animale și o cardinalitate maximă de M:1 (o rasă de animale poate sta în cel mult o cușcă, dar o cușcă poate avea mai multe rase de animale).

PLANTĂ\_crește\_pe\_TEREN = relație care leagă entitățile PLANTĂ și TEREN, reflectând legătura dintre acestea (plantele din fermă au un teren pe care cresc). Relația are o cardinalitate minimă de 0:1 (o plantă crește pe minim un teren, dar pe un teren nu trebuie să crească neapărat ceva) și o cardinalitate maximă de M:1 (pe un teren pot crește mai multe plante, dar o plantă nu poate crește pe mai multe terenuri).

TEREN\_are\_SOL = relație de tip *many-to-many* care leagă entitățile TEREN și SOL, reflectând legătura dintre acestea (fiecare tip de teren are un tip de sol specific). Relația are o cardinalitate minimă de 1:1 (fiecare tip de teren are și un tip de sol, și fiecare sol are un tip de teren) și o cardinalitate maximă de M:M (un teren poate avea mai multe tipuri de sol, iar un tip de sol se poate găsi pe mai multe terenuri).

FERMIER\_urmează\_SPECIALIZARE = relație de tip *many-to-many* care leagă entitățile FERMIER și SPECIALIZARE, reflectând legătura dintre acestea (fiecare fermier are o specializare). Relația are o cardinalitate minimă de 1:1 (fiecare fermier are cel puțin câte o specializare, și toate specializările prezentate aparțin câte unui fermier) și o cardinalitate maximă de M:M (un fermier poate avea mai multe specializări, iar o specializare poate fi urmată de mai mulți fermieri).

ȘOFER\_conduce\_CAMION\_DE\_MARFA = relație care leagă entitățile ȘOFER și CAMION\_DE\_MARFA, reflectând legătura dintre acestea (șoferii conduc camioanele de marfă pentru a distribui produsele). Relația are o cardinalitate minimă de 0:1 (un șofer conduce cel puțin un camion, dar nu toate camioanele trebuie să fie conduse de un șofer) și o cardinalitate maximă de 1:1 (fiecare șofer conduce un singur camion personalizat, și fiecare camion este condus de către un șofer).

ANGAJAT\_deține\_ISTORIC = relație care leagă entitățile ANGAJAT și ISTORIC, reflectând legătura dintre acestea (angajații au câte un istoric de muncă). Relația are o cardinalitate minimă și maximă de 1:1 (fiecare angajat are câte un istoric unic).

CONTABIL\_organizează\_ACTIVITATE = relație de tip *many-to-many* care leagă entitățile CONTABIL și ACTIVITATE, reflectând legătura dintre acestea (contabilii organizează și planifică activitățile specifice fermei). Relația are o cardinalitate minimă de 1:1 (o activitate trebuie organizată de cel puțin un contabil, și un contabil organizează cel puțin o activitate) și o cardinalitate maximă de M:M (un contabil poate organiza mai multe activități, iar o activitate poate fi organizată de un grup de contabili care se consultă între ei).

COMERCIANT\_contorizează\_COMANDĂ = relație care leagă entitățile COMERCIANT și COMANDĂ, reflectând legătura dintre acestea (în cadrul fermei există mai multe comenzi care sunt preluate, contorizate și procesate de către comercianții care lucrează la fermă). Relația are o cardinalitate minimă de 1:1 (fiecare comandă are cel puțin o persoană care o contorizează, și fiecare comerciant care lucrează la fermă are cel puțin o comandă de luat) și o cardinalitate maximă de 1:M (un contabil poate organiza mai multe comenzi, dar o comandă este organizată de un singur comerciant).

UTILIZATOR\_plasează\_COMANDĂ = relație care leagă entitățile UTILIZATOR și COMANDĂ, reflectând legătura dintre acestea (toate produsele comandate sunt cerute de

către utilizatorii aplicației). Relația are o cardinalitate minimă de 1:1 (pentru a avea statutul de utilizator trebuie să plasezi cel puțin o comandă, și o comandă este plasată de un utilizator) și o cardinalitate maximă de 1:M (un utilizator poate plasa mai multe comenzi, dar comenzile sunt unice și pot fi plasate de un singur utilizator).

FERMIER\_îngrijește\_ANIMAL\_mănâncă\_PLANTĂ = relație de tip trei care leagă entitățile FERMIER, ANIMAL și PLANTĂ. Între toate aceste entități există o *relație many-to-many*, reflectând faptul că fermierul are grijă atât de animale, cât și de plante, iar animalele se hrănesc cu plantele de la fermă. Vom numi această relație "îngrijește".

FERMIER\_este\_un\_ANGAJAT = relație de tip ISA care leagă entitatea FERMIER și ANGAJAT (un fermier este un fel de angajat). Relația are o cardinalitate minimă de 0:1 și maximă de 1:1.

ȘOFER\_este\_un\_ANGAJAT = relație de tip ISA care leagă entitatea ȘOFER și ANGAJAT (un șofer este un fel de angajat). Relația are o cardinalitate minimă de 0:1 și maximă de 1:1.

CONTABIL\_este\_un\_ANGAJAT = relație de tip ISA care leagă entitatea CONTABIL și ANGAJAT (un contabil este un fel de angajat). Relația are o cardinalitate minimă de 0:1 și maximă de 1:1.

COMERCIANT\_este\_un\_ANGAJAT = relație de tip ISA care leagă entitatea COMERCIANT și ANGAJAT (un comerciant este un fel de angajat). Relația are o cardinalitate minimă de 0:1 și maximă de 1:1.

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

### **ATTRIBUTE**

Entitatea independentă CUȘCĂ are ca atribute:

*id\_cusca*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unei custi.

*lungime*= variabilă de tip întreg, de lungime maximă 5, care reprezintă lungimea cuștii în care animalele stau, măsurată în metrii.

*lațime*= variabilă de tip întreg, de lungime maximă 5, care reprezintă lățimea cuștii în care animalele stau, măsurată în metrii.

*capacitate*= variabilă de tip întreg, de lungime maximă 5, care reprezintă capacitatea cuștii în care animalele stau, măsurată în metrii cubi.

Entitatea independentă SOL are ca atribute:

*id\_sol*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unui tip de sol.

*densitate*= variabilă de tip real, de lungime maximă 5, care reprezintă densitatea solului în care cresc plantele, măsurată în kg/m<sup>3</sup>;

*grad\_de\_fertilitate*= variabilă de tip caracter, care ia valori dintre cuvintele *crescută* sau *scazută*, care reprezintă cât de fertil a fost solul într-un anumit an.

Entitatea independentă ANGAJAT are ca atribute:

*id\_angajat*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unui angajat.

*nume* = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele angajatului.

*prenume* = variabilă de tip caracter, de lungime maximă 25, care reprezintă prenumele angajatului.

*salariu*= variabilă de tip întreg, de lungime maximă 15, care reprezintă salariu câștigat de fiecare angajat anual.

*data\_nastere* = variabilă de tip dată calendaristică, care reprezintă data nașterii angajatului respectiv.

*sex* = variabilă de tip caracter, luând valorile m sau f, de lungime 1, care reprezintă sexul angajatului.

*naționalitate* = variabilă de tip caracter, de lungime maximă 12, care reprezintă naționalitatea unui angajat .

Entitatea independentă SPECIALIZARE are ca atribute:

*id\_specializare*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic a unei specializări.

*data\_inființării*= variabilă de tip dată calendaristică, care reprezintă data la care s-a înființat o specializare în fermă.

*nume\_s*= variabilă de tip caracter, de lungime maximă 50, care reprezintă numele unei specializări .

Entitatea independentă CAMION\_DE\_MARFA are ca atribute:

*id\_camion*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic a unui camion.

*capacitate*= variabilă de tip întreg, de lungime maximă 10, care reprezintă capacitatea unui camion, pentru a vedea câte tone poate încărca.

*id\_angajat*= corespunde valorii cheii primare din tabelul ANGAJAT.

Entitatea independentă ISTORIC are ca attribute:

*id\_istoric*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unui istoric al unui angajat.

*data\_angajării*= variabilă de tip dată calendaristică, care reprezintă data de la care un angajat a început să lucreze.

*data\_promovării*= variabilă de tip dată calendaristică, care reprezintă data la care un angajat a primit o promovare sau o mărire de salariu.

*productivitate*= variabilă de tip caracter, care ia valori dintre cuvintele *crescuta* sau *scazuta*, care reprezintă cat de productiv a fost un angajat într-un anumit an.

*id\_angajat*= corespunde valorii cheii primare din tabelul ANGAJAT.

Entitatea independentă ACTIVITATE are ca attribute:

*id\_activitate*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unei activitati.

*data\_desfășurare*= variabilă de tip dată calendaristică, care reprezintă data în care un eveniment are loc.

*ora*= variabilă de tip caracter, care reprezintă ora în format de 24h la care un eveniment are loc.

Entitatea independentă UTILIZATOR are ca attribute:

*id\_user*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unui user.

*nume* = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele angajatului.

*prenume* = variabilă de tip caracter, de lungime maximă 25, care reprezintă prenumele angajatului.

Entitatea ANIMAL are ca attribute: *id\_cusca* descris mai sus,

*id\_animal*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unui tip de animal din fermă.

*rasă*= variabilă de tip caracter, de lungime maximă 25, care reprezintă rasa animalului.

*nr\_animale*= variabilă de tip întreg, de lungime maximă 10, care reprezintă numărul de animale din fiecare rasă.

Entitatea TEREN are ca attribute: *id\_sol* descrie mai sus,



*id\_teren*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unui tip de teren.

*arie*= variabilă de tip întreg, de lungime maximă 10, care reprezintă suprafața terenurilor cultivate cu plante în hectare.

Entitatea PLANTĂ are ca atribute: *id\_sol* descris mai sus, *id\_teren* descris mai sus,

*id\_plantă*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unui tip de plante.

*calitate*= variabila de tip caracter, care ia valori dintre cuvintele *intai*, *doi* sau *trei*, care reprezintă cat de buna este recolta pe un anumit tip de plante în acel an.

*cantitate*= variabila de tip întreg, de lungime maximă 10, care reprezintă cate kilograme din fiecare tip de plantă au fost recoltate.

Entitatea COMANDĂ are ca atribute: *id\_user* descris mai sus,

*id\_comandă*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unei comenzi plasate de un utilizator.

*adresă*= variabilă de tip caracter, de maxim 50 de caractere, care reprezintă adresa unde trebuie livrată comanda.

*nr\_telefon*= variabilă de tip întreg, de lungime maximă 10, care reprezintă numărul de telefon care este folosit pentru confirmarea comenzii.

*id\_angajat*= corespunde valorii cheii primare din tabelul ANGAJAT.

Relația TEREN\_are\_SOL are ca atribute:

*id\_sol*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unui tip de sol. Atributul trebuie să corespundă cu o valoare a cheii primare din tabelul SOL.

*id\_teren*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unui tip de teren. Atributul trebuie să corespundă cu o valoare a cheii primare din tabelul TEREN.

Relația FERMIER\_urmează\_SPECIALIZARE are ca atribute:

*id\_specializare*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic a unei specializări. Atributul trebuie să corespundă cu o valoare a cheii primare din tabelul SPECIALIZARE.

*id\_angajat*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unui angajat. Atributul trebuie să corespundă cu o valoare a cheii primare din tabelul FERMIER.

*experiență în domeniu*= variabilă de tip întreg, de lungime maximă 5, care reprezintă experiența pe care fiecare fermier ce urmează o specializare o are, ea se măsoară în ani.

Relația CONTABIL\_organizeaza\_ACTIVITATE are ca attribute:

*id\_angajat*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unui angajat. Atributul trebuie să corespundă cu o valoare a cheii primare din tabelul CONTABIL.

*id\_activitate*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unei activități. Atributul trebuie să corespundă cu o valoare a cheii primare din tabelul ACTIVITATE.

*locuție*= variabilă de tip caracter, de lungime maximă 40, care reprezintă locul desfășurării unui eveniment. Variabila va folosi prescurtări.

Relația FERMIER îngrijește ANIMAL mănâncă PLANTĂ are ca attribute: *id\_angajat*, *id\_animal*, *id\_cușcă*, *id\_plantă*, *id\_teren*, *id\_sol*. Toate aceste attribute au fost descrise mai sus. Atributele trebuie să corespundă cu o valoare a cheii primare din tabelele ANGAJAT, ANIMAL, respective PLANTĂ.

Subentitatea FERMIER are ca attribute :

*id\_angajat*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unui angajat.

*nume* = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele angajatului.

*prenume* = variabilă de tip caracter, de lungime maximă 25, care reprezintă prenumele angajatului.

*salariu*= variabilă de tip întreg, de lungime maximă 5, care reprezintă salariu câștigat de fiecare angajat anual.

*data\_naștere* = variabilă de tip dată calendaristică, care reprezintă data nașterii angajatului respectiv.

*sex* = variabilă de tip caracter, luând valorile m sau f, de lungime 1, care reprezintă sexul angajatului.

*naționalitate* = variabilă de tip caracter, de lungime maximă 12, care reprezintă naționalitatea unui angajat .

*certificări*= variabilă de tip caracter, de lungime maximă 12, care reprezintă informații despre certificările, standardele sau calificările deținute de fermier. Acest atribut poate include informații despre practicile agricole sustenabile, metodele de creștere a animalelor sau alte aspecte legate de calitatea și siguranța produselor.

Subentitatea ȘOFER are ca attribute :

*id\_angajat*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unui angajat.

*nume* = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele angajatului.

*prenume* = variabilă de tip caracter, de lungime maximă 25, care reprezintă prenumele angajatului.

*salariu*= variabilă de tip întreg, de lungime maximă 5, care reprezintă salariu castigat de fiecare angajat anual.

*data\_nastere* = variabilă de tip dată calendaristică, care reprezintă data nașterii angajatului respectiv.

*sex* = variabilă de tip caracter, luând valorile m sau f, de lungime 1, care reprezintă sexul angajatului.

*naționalitate* = variabilă de tip caracter, de lungime maximă 12, care reprezintă naționalitatea unui angajat.

*categorie\_permis*= variabilă de tip caracter, de lungime maximă 20, care reprezintă toate categoriile de permis pe care fiecare sofer le are. Acestea vor fi enumerate cu virgula între ele.

Subentitatea CONTABIL are ca attribute :

*id\_angajat*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unui angajat.

*nume* = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele angajatului.

*prenume* = variabilă de tip caracter, de lungime maximă 25, care reprezintă prenumele angajatului.

*salariu*= variabilă de tip întreg, de lungime maximă 5, care reprezintă salariu castigat de fiecare angajat anual.

*data\_nastere* = variabilă de tip dată calendaristică, care reprezintă data nașterii angajatului respectiv.

*sex* = variabilă de tip caracter, luând valorile m sau f, de lungime 1, care reprezintă sexul angajatului.

*naționalitate* = variabilă de tip caracter, de lungime maximă 12, care reprezintă naționalitatea unui angajat.

*educație*= variabilă de tip caracter, de lungime maximă 60, care reprezintă facultatea sau cursurile în domeniu pe care un contabil le-a urmat. Aceasta variabila va conține abrevieri.

Subentitatea COMERCIANT are ca atribute :

*id\_angajat*= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unic al unui angajat.

*nume* = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele angajatului.

*prenume* = variabilă de tip caracter, de lungime maximă 25, care reprezintă prenumele angajatului.

*salariu*= variabilă de tip întreg, de lungime maximă 5, care reprezintă salariu castigat de fiecare angajat anual.

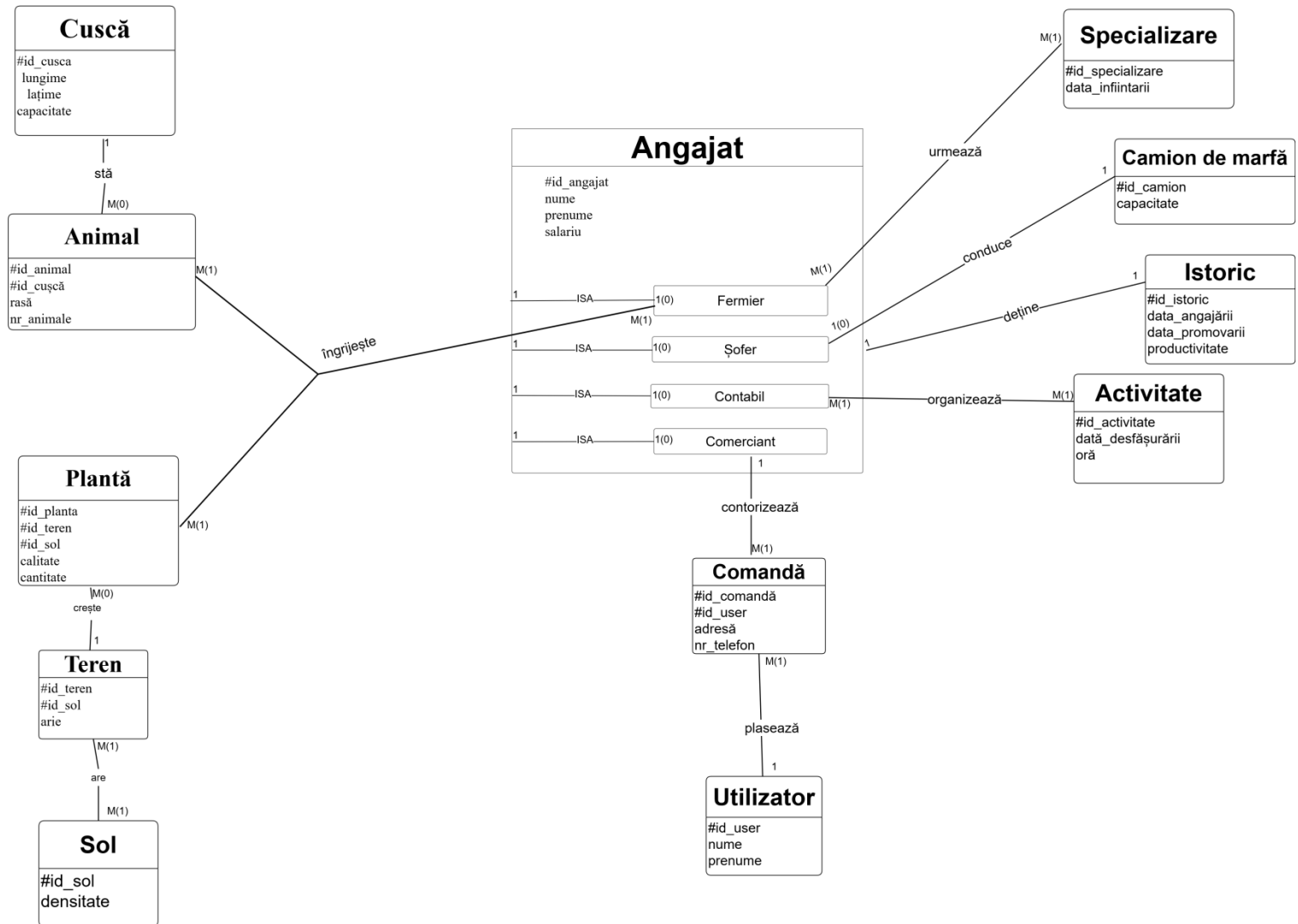
*data\_naștere* = variabilă de tip dată calendaristică, care reprezintă data nașterii angajatului respectiv.

*sex* = variabilă de tip caracter, luând valorile m sau f, de lungime 1, care reprezintă sexul angajatului.

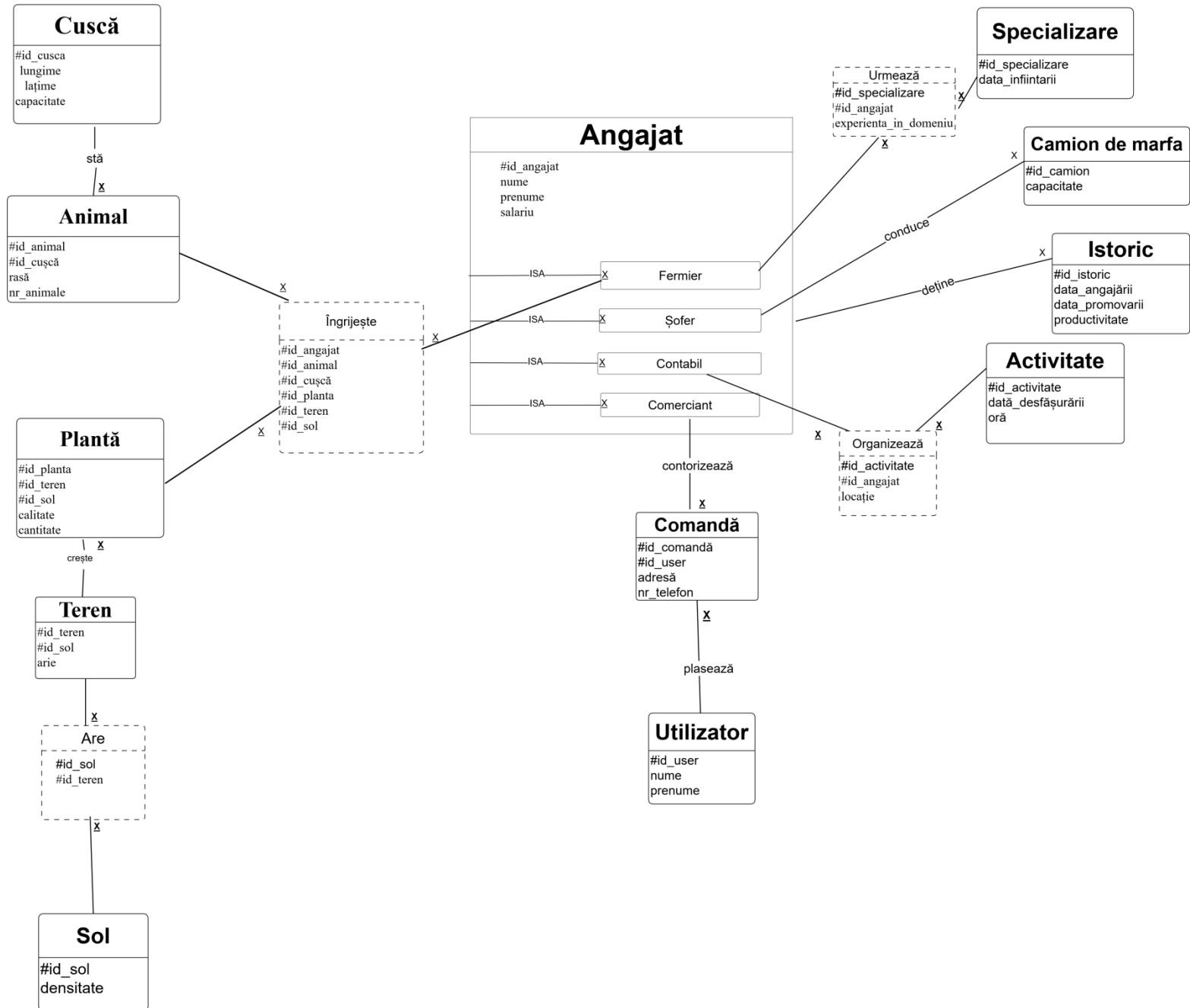
*naționalitate* = variabilă de tip caracter, de lungime maximă 12, care reprezintă naționalitatea unui angajat.

*volumul \_de\_ vânzări*= variabila de tip caracter, care ia valori dintre cuvintele *scazut*, *mediu* sau *crescut*, care reprezinta cat de multe produse a reusit un comerciant sa contorizeze în acel an. Acest atribut oferă perspectiva asupra dimensiunii și performanței comerciantului.

## 6.) Diagrama Entitate-Relație



## 7.) Diagrama conceptuală



8.) Schemele relaționale corespunzătoare diagramei conceptuale sunt următoarele:

CUSCA (#id\_cusca, lungime, latime, capacitate)

ANIMAL (#id\_animal, #id\_cușcă, rasă, nr\_animale)

INGRIJESTE ( #id\_angajat, #id\_animal, #id\_cușcă, #id\_planta, #id\_teren, #id\_sol)

PLANTĂ (#id\_planta, #id\_teren, #id\_sol, calitate, cantitate)

TEREN (#id\_teren, #id\_sol, arie)

ARE (#id\_sol, #id\_teren)

SOL(#id\_sol, densitate, grad\_de\_fertilitate)

ANGAJAT (#id\_angajat, nume, prenume, salariu, data\_nastere, sex, nationalitate)

FERMIER (#id\_angajat, nume, prenume, salariu, data\_nastere, sex, nationalitate,certificari)

SOFER (#id\_angajat, nume, prenume, salariu, data\_nastere, sex, nationalitate, categorie\_permis)

COMERCIANT (#id\_angajat, nume, prenume, salariu, data\_nastere, sex, nationalitate, educatie)

CONTABIL (#id\_angajat, nume, prenume, salariu, data\_nastere, sex, nationalitate, volum\_de\_vanzari)

COMANDA (#id\_comandă, #id\_user, #id\_angajat, adresă, nr\_telefon)

UTILIZATOR (#id\_user, nume, prenume)

ORGANIZEAZA (#id\_activitate, #id\_angajat, locație)

ACTIVITATE (#id\_activitate, dată\_desfășurării,oră)

ISTORIC (#id\_istoric, #id\_angajat ,data\_angajării, data\_promovarii, productivitate)

CAMION\_DE\_MARFA (#id\_camion, #id\_angajat ,capacitate)

SPECIALIZARE (#id\_specializare, nume, data\_infiintarii)

URMEAZA (#id\_specializare, #id\_angajat, experienta\_in\_domeniu)

## 9.) NORMALIZARE

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

## FN1

O relație este în prima formă normală dacă fiecărui atribut care o compune îi corespunde o valoare indivizibilă (atomică).

De exemplu, tabelul ORGANIZEAZA nu s-ar afla în FN1 dacă o instanța din acest tabel ar avea mai multe valori în atributul locație.

| Id_angajat# | Id_activitate# | Locatie                          |
|-------------|----------------|----------------------------------|
| 210         | 60             | Hamaru Farm - Barn 1, Barn 2     |
| 211         | 61             | Hamaru Farm – Greenhouse, Barn 3 |

## FN2

O relație R este în a doua formă normală dacă și numai dacă:

- relația R este în FN1;
- fiecare atribut care nu este cheie (nu participă la cheia primară) este dependent de întreaga cheie primară.

De exemplu, tabelul URMEAȘA nu ar fi în FN2 dacă atributul lui experienta\_in\_domeniu ar depinde doar de o parte din cheia primară compusă.

| Id_specializare# | id_angajat# | Experienta_in_domeniu |
|------------------|-------------|-----------------------|
| 13               | <b>201</b>  | 18                    |
| 12               | <b>201</b>  | 18                    |

În această situație, observăm că experienta\_in\_domeniu depinde doar de id\_angajat.

## FN3

Intuitiv, o relație R este în a treia formă normală dacă și numai dacă:

- relația R este în FN2;
- fiecare atribut care nu este cheie (nu participă la o cheie) depinde direct de cheia primară;

În proiectul meu, toate relațiile sunt deja în FN3. Dacă nu ar fi în FN3, tabelul ANIMAL ar putea să arate în felul următor:

| Id_animal# | rasa  | Id_cusca# | capacitate |
|------------|-------|-----------|------------|
| 51         | porc  | 6         | 20         |
| 63         | gasca | 8         | 18         |

Se observă că atributul capacitate depinde tranzitiv de id\_animal, deoarece acesta depinde de id\_cusca care depinde de id\_animal.

### 10.) Crearea de secvențe:



Am creat cate o secventa pentru fiecare tabel neasociativ sau care nu este subentitate.

```
CREATE SEQUENCE SEQ_CUSCA  
INCREMENT by 1  
START WITH 1  
MAXVALUE 9  
NOCYCLE;
```

```
CREATE SEQUENCE SEQ_SOL  
INCREMENT by 1  
START WITH 10  
MAXVALUE 19  
NOCYCLE;
```

```
CREATE SEQUENCE SEQ_ANGAJAT  
INCREMENT by 1  
START WITH 200  
MAXVALUE 221  
NOCYCLE;
```

```
CREATE SEQUENCE SEQ_SPECIALIZARE  
INCREMENT by 1  
START WITH 30  
MAXVALUE 39  
NOCYCLE;
```

```
CREATE SEQUENCE SEQ_CAMION_DE_MARFA  
INCREMENT by 1  
START WITH 40  
MAXVALUE 49  
NOCYCLE;
```

```
CREATE SEQUENCE SEQ_ISTORIC  
INCREMENT by 1  
START WITH 500  
MAXVALUE 520  
NOCYCLE;
```

```
CREATE SEQUENCE SEQ_ACTIVITATE  
INCREMENT by 1  
START WITH 60  
MAXVALUE 69  
NOCYCLE;
```

```
CREATE SEQUENCE SEQ_UTILIZATOR  
INCREMENT by 1  
START WITH 70  
MAXVALUE 79  
NOCYCLE;
```

```
CREATE SEQUENCE SEQ_ANIMAL  
INCREMENT by 1  
START WITH 80  
MAXVALUE 89  
NOCYCLE;
```

```
CREATE SEQUENCE SEQ_TEREN  
INCREMENT by 1  
START WITH 90  
MAXVALUE 99  
NOCYCLE;
```

```
CREATE SEQUENCE SEQ_PLANTA  
INCREMENT by 1  
START WITH 100  
MAXVALUE 109  
NOCYCLE;
```

```
CREATE SEQUENCE SEQ_COMANDA  
INCREMENT by 1  
START WITH 110  
MAXVALUE 119  
NOCYCLE;
```

## **CREATE**

-cream tablele

```
CREATE TABLE CUSCA(  
    id_cusca number(5) constraint pkid_cusca primary key,  
    lungime number(5),  
    latime number(5),  
    capacitate number(5)  
);
```

```
CREATE TABLE SOL (  
    id_sol number(5)constraint pkid_sol primary key,  
    densitate number(5,2),  
    grad_de_fertilitate CHAR(8) CHECK (grad_de_fertilitate IN ('crescuta', 'scazuta'))  
);
```

```
CREATE TABLE ANGAJAT (  
    id_angajat number(5) constraint pkid_angajat primary key,  
    nume VARCHAR(25),  
    prenume VARCHAR(25),  
    salariu number(15)constraint salary not NULL,  
    data_nastere DATE,  
    sex CHAR(4) CHECK (sex IN ('f', 'm')),  
    nationalitate VARCHAR(12)  
);
```

```
CREATE TABLE SPECIALIZARE (  
    id_specializare number(5) constraint pkid_specializare primary key,  
    data_infiintarii date,  
    nume_s varchar(50)  
);
```

```
CREATE TABLE CAMION_DE_MARFA (  
    id_camion number(5) constraint pkid_camion primary key,  
    capacitate number(10),  
    id_angajat number(5) constraint fk_camion_angajat references ANGAJAT(id_angajat)  
);
```

```
CREATE TABLE ISTORIC (  
    id_istoric number(5) constraint pkid_istoric primary key,  
    data_angajarii DATE,  
    data_promovarii DATE,  
    productivitate varchar(8) CHECK (productivitate IN ('crescuta', 'scazuta')),  
    id_angajat number(5) constraint fk_istoric_angajat references ANGAJAT(id_angajat)  
);
```

```
CREATE TABLE ACTIVITATE (  
    id_activitate number(5) constraint pkid_activitate primary key,  
    data_desfasurare DATE,  
    ora varchar(25)  
);
```

```
CREATE TABLE UTILIZATOR (  
    id_user number(5) constraint pkid_user primary key,  
    nume VARCHAR(25),  
    prenume VARCHAR(25)  
);
```

```
CREATE TABLE ANIMAL (  
    id_animal number(5) constraint pkid_animal primary key,  
    id_cusca number(5),  
    rasa VARCHAR(25),  
    nr_animale number(10),  
    constraint fk_cod_cusca foreign key (id_cusca) references CUSCA(id_cusca)  
);
```

```
CREATE TABLE TEREN (  
    id_teren number(5) constraint pkid_teren primary key,  
    id_sol number(5),  
    arie number(10),  
    constraint fk_cod_sol foreign key (id_sol) references SOL(id_sol)  
);
```

```
CREATE TABLE PLANTA (  
    id_planta number(5) constraint pkid_planta primary key,  
    id_sol number(5),
```

```
id_teren number(5),
calitate VARCHAR(5) CHECK (calitate IN ('intai', 'doi', 'trei')),
cantitate number(10),
constraint fk_cod_soll foreign key (id_sol) references SOL(id_sol),
constraint fk_cod_teren foreign key (id_teren) references TEREN(id_teren)
```

);

```
CREATE TABLE COMANDA (
    id_comanda number(5) constraint pkid_comanda primary key,
    id_user number(5),
    adresa VARCHAR(50),
    nr_telefon number(10) constraint telefon not NULL unique,
    constraint fk_cod_user foreign key (id_user) references UTILIZATOR(id_user),
    id_angajat number(5) constraint fk_comanda_angajat references ANGAJAT(id_angajat)
);
```

```
CREATE TABLE ARE_ (
    id_sol number(5) constraint codul_solului references SOL(id_sol),
    id_teren number(5) constraint codul_terenului references TEREN(id_teren),
    constraint are_pk primary key(id_sol, id_teren)
);
```

```
CREATE TABLE ANGAJAT (
    id_angajat number(5) constraint pkid_angajat primary key,
    nume VARCHAR(25),
    prenume VARCHAR(25),
    salariu number(15) constraint salary not NULL,
    data_nastere DATE,
    sex CHAR(4) CHECK (sex IN ('f', 'm')),
    nationalitate VARCHAR(12)
);
```

```
CREATE TABLE FERMIER(
    id_angajat number(5) primary key references ANGAJAT(id_angajat),
    certificari VARCHAR(12)
);
```

```
CREATE TABLE SOFER(
    id_angajat number(5) primary key references ANGAJAT(id_angajat),
    categorii_permis VARCHAR(20)
);
```

```
CREATE TABLE CONTABIL(
    id_angajat number(5) primary key references ANGAJAT(id_angajat),
    educatie VARCHAR (60)
);
```

```
CREATE TABLE COMERCIANT(
    id_angajat number(5) primary key references ANGAJAT(id_angajat),
    volumul_de_vanzari VARCHAR(12) CHECK ( volumul_de_vanzari IN ('scazut',
    'mediu','crescut'))
);
```

```
CREATE TABLE INGRIJESTE(
    id_sol number(5) constraint codul_sol references SOL(id_sol),
    id_teren number(5) constraint codul_teren references TEREN(id_teren),
    id_planta number(5) constraint codul_planta references PLANTA(id_planta),
    id_cusca number(5) constraint codul_cusca references CUSCA(id_cusca),
    id_animal number(5) constraint codul_animalului references ANIMAL(id_animal),
    id_angajat number(5) constraint codul_angajatului references ANGAJAT(id_angajat),
    constraint ingrijeste_pk primary key(id_sol, id_teren, id_planta, id_cusca, id_animal,
    id_angajat)
);
```

```
CREATE TABLE URMEAZA(
    id_specializare number(5) constraint codul_specializarii references
    SPECIALIZARE(id_specializare),
    id_angajat number(5) constraint codul_angajat references ANGAJAT(id_angajat),
    experienta_in_domeniu number(3) not null,
    constraint urmeaza_pk primary key(id_specializare, id_angajat)
);
```

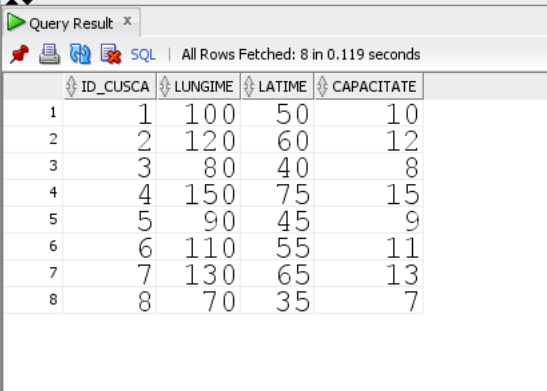
```
CREATE TABLE ORGANIZEAZA(
    id_activitate number(5) constraint codul_activitatii references
    ACTIVITATE(id_activitate),
    id_angajat number(5) constraint codul_angajat2 references ANGAJAT(id_angajat),
    locatie VARCHAR(40),
    constraint organizeaza_pk primary key(id_activitate, id_angajat)
);
```

## INSERT

-inseram datele in tabele

```
INSERT INTO CUSCA (id_cusca, lungime, latime, capacitate)
VALUES(SEQ_CUSCA.NEXTVAL, 100, 50, 10);
INSERT INTO CUSCA (id_cusca, lungime, latime, capacitate)
VALUES (SEQ_CUSCA.NEXTVAL, 120, 60, 12);
INSERT INTO CUSCA (id_cusca, lungime, latime, capacitate)
VALUES(SEQ_CUSCA.NEXTVAL, 80, 40, 8);
INSERT INTO CUSCA (id_cusca, lungime, latime, capacitate)
VALUES(SEQ_CUSCA.NEXTVAL, 150, 75, 15);
INSERT INTO CUSCA (id_cusca, lungime, latime, capacitate)
VALUES(SEQ_CUSCA.NEXTVAL, 90, 45, 9);
INSERT INTO CUSCA (id_cusca, lungime, latime, capacitate)
```

```
VALUES(SEQ_CUSCA.NEXTVAL, 110, 55, 11);
INSERT INTO CUSCA (id_cusca, lungime, latime, capacitate)
VALUES(SEQ_CUSCA.NEXTVAL, 130, 65, 13);
INSERT INTO CUSCA (id_cusca, lungime, latime, capacitate)
VALUES (SEQ_CUSCA.NEXTVAL, 70, 35, 7);
```

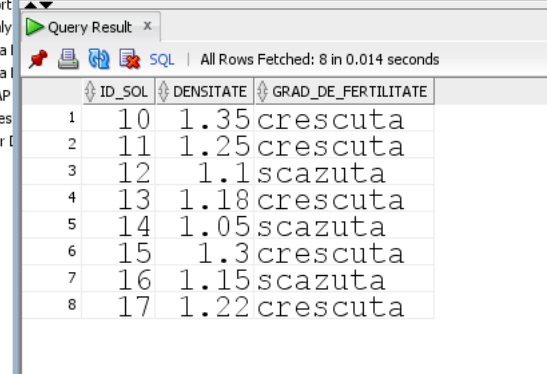


Query Result x

All Rows Fetched: 8 in 0.119 seconds

|   | ID_CUSCA | LUNGIME | LATIME | CAPACITATE |
|---|----------|---------|--------|------------|
| 1 | 1        | 100     | 50     | 10         |
| 2 | 2        | 120     | 60     | 12         |
| 3 | 3        | 80      | 40     | 8          |
| 4 | 4        | 150     | 75     | 15         |
| 5 | 5        | 90      | 45     | 9          |
| 6 | 6        | 110     | 55     | 11         |
| 7 | 7        | 130     | 65     | 13         |
| 8 | 8        | 70      | 35     | 7          |

```
INSERT INTO SOL (id_sol, densitate, grad_de_fertilitate)
VALUES (SEQ_SOL.NEXTVAL, 1.25, 'crescuta');
INSERT INTO SOL (id_sol, densitate, grad_de_fertilitate)
VALUES (SEQ_SOL.NEXTVAL, 1.10, 'scazuta');
INSERT INTO SOL (id_sol, densitate, grad_de_fertilitate)
VALUES (SEQ_SOL.NEXTVAL, 1.35, 'crescuta');
INSERT INTO SOL (id_sol, densitate, grad_de_fertilitate)
VALUES (SEQ_SOL.NEXTVAL, 1.18, 'crescuta');
INSERT INTO SOL (id_sol, densitate, grad_de_fertilitate)
VALUES (SEQ_SOL.NEXTVAL, 1.05, 'scazuta');
INSERT INTO SOL (id_sol, densitate, grad_de_fertilitate)
VALUES (SEQ_SOL.NEXTVAL, 1.30, 'crescuta');
INSERT INTO SOL (id_sol, densitate, grad_de_fertilitate)
VALUES(SEQ_SOL.NEXTVAL, 1.15, 'scazuta');
INSERT INTO SOL (id_sol, densitate, grad_de_fertilitate)
VALUES(SEQ_SOL.NEXTVAL, 1.22, 'crescuta');
```



Query Result x

All Rows Fetched: 8 in 0.014 seconds

|   | ID_SOL | DENSITATE | GRAD_DE_FERTILITATE |
|---|--------|-----------|---------------------|
| 1 | 10     | 1.35      | crescuta            |
| 2 | 11     | 1.25      | crescuta            |
| 3 | 12     | 1.1       | scazuta             |
| 4 | 13     | 1.18      | crescuta            |
| 5 | 14     | 1.05      | scazuta             |
| 6 | 15     | 1.3       | crescuta            |
| 7 | 16     | 1.15      | scazuta             |
| 8 | 17     | 1.22      | crescuta            |

```
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
```

```

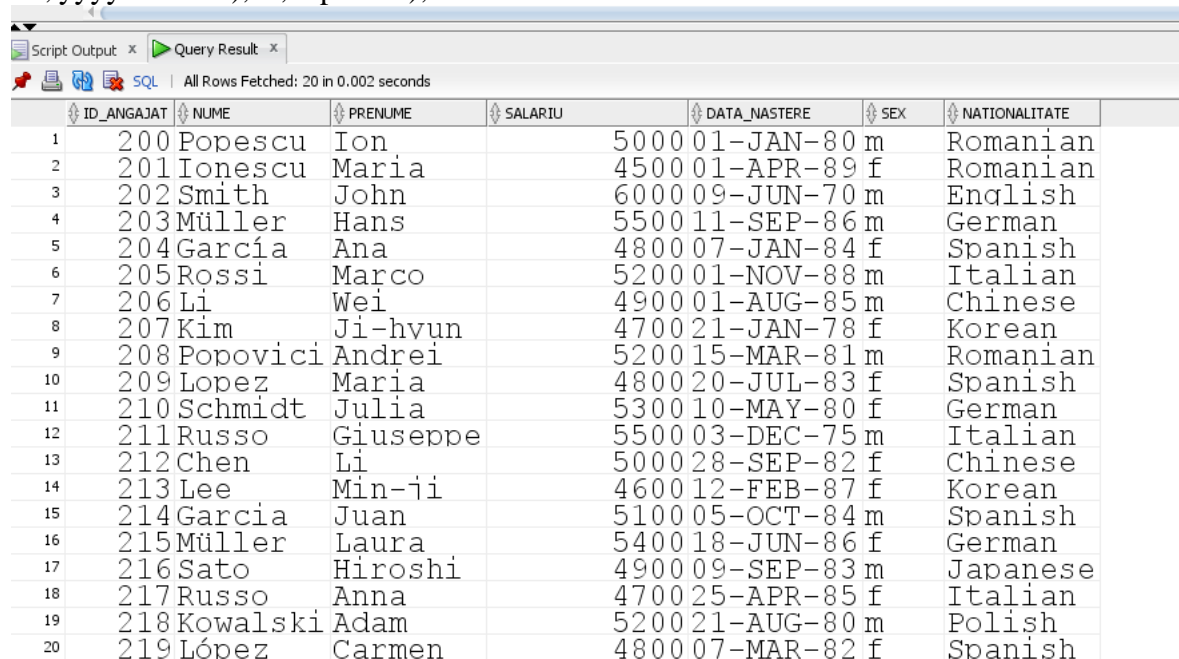
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Popescu', 'Ion', 5000, to_date('1980-01-01','yyyy-
mm-dd'), 'm', 'Romanian');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Ionescu', 'Maria', 4500, to_date('1989-04-
01','yyyy-mm-dd'), 'f', 'Romanian');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Smith', 'John', 6000, to_date('1970-06-09','yyyy-
mm-dd'), 'm', 'English');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Müller', 'Hans', 5500, to_date('1986-09-
11','yyyy-mm-dd'), 'm', 'German');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'García', 'Ana', 4800, to_date('1984-01-07','yyyy-
mm-dd'), 'f', 'Spanish');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Rossi', 'Marco', 5200, to_date('1988-11-01','yyyy-
mm-dd'), 'm', 'Italian');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Li', 'Wei', 4900, to_date('1985-08-01','yyyy-mm-
dd'), 'm', 'Chinese');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Kim', 'Ji-hyun', 4700, to_date('1978-01-21','yyyy-
mm-dd'), 'f', 'Korean');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Popovici', 'Andrei', 5200, to_date('1981-03-
15','yyyy-mm-dd'), 'm', 'Romanian');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Lopez', 'Maria', 4800, to_date('1983-07-20','yyyy-
mm-dd'), 'f', 'Spanish');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Schmidt', 'Julia', 5300, to_date('1980-05-10','yyyy-
mm-dd'), 'f', 'German');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Russo', 'Giuseppe', 5500, to_date('1975-12-
03','yyyy-mm-dd'), 'm', 'Italian');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Chen', 'Li', 5000, to_date('1982-09-28','yyyy-mm-
dd'), 'f', 'Chinese');

```

```

INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Lee', 'Min-ji', 4600, to_date('1987-02-12','yyyy-
mm-dd'), 'f', 'Korean');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Garcia', 'Juan', 5100, to_date('1984-10-05','yyyy-
mm-dd'), 'm', 'Spanish');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Müller', 'Laura', 5400, to_date('1986-06-
18','yyyy-mm-dd'), 'f', 'German');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Sato', 'Hiroshi', 4900, to_date('1983-09-09','yyyy-
mm-dd'), 'm', 'Japanese');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Russo', 'Anna', 4700, to_date('1985-04-25','yyyy-
mm-dd'), 'f', 'Italian');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'Kowalski', 'Adam', 5200, to_date('1980-08-
21','yyyy-mm-dd'), 'm', 'Polish');
INSERT INTO ANGAJAT (id_angajat, nume, prenume, salariu, data_nastere, sex,
nationalitate)
VALUES (SEQ_ANGAJAT.NEXTVAL, 'López', 'Carmen', 4800, to_date('1982-03-
07','yyyy-mm-dd'), 'f', 'Spanish');

```



Script Output x Query Result x

SQL | All Rows Fetched: 20 in 0.002 seconds

|    | ID_ANGAJAT | NUME     | PRENUME  | SALARIU | DATA_NASTERE | SEX | NATIONALITATE |
|----|------------|----------|----------|---------|--------------|-----|---------------|
| 1  | 200        | Popescu  | Ion      | 5000    | 01-JAN-80    | m   | Romanian      |
| 2  | 201        | Ionescu  | Maria    | 4500    | 01-APR-89    | f   | Romanian      |
| 3  | 202        | Smith    | John     | 6000    | 09-JUN-70    | m   | English       |
| 4  | 203        | Müller   | Hans     | 5500    | 11-SEP-86    | m   | German        |
| 5  | 204        | García   | Ana      | 4800    | 07-JAN-84    | f   | Spanish       |
| 6  | 205        | Rossi    | Marco    | 5200    | 01-NOV-88    | m   | Italian       |
| 7  | 206        | Li       | Wei      | 4900    | 01-AUG-85    | m   | Chinese       |
| 8  | 207        | Kim      | Ji-hyun  | 4700    | 21-JAN-78    | f   | Korean        |
| 9  | 208        | Popovici | Andrei   | 5200    | 15-MAR-81    | m   | Romanian      |
| 10 | 209        | Lopez    | Maria    | 4800    | 20-JUL-83    | f   | Spanish       |
| 11 | 210        | Schmidt  | Julia    | 5300    | 10-MAY-80    | f   | German        |
| 12 | 211        | Russo    | Giuseppe | 5500    | 03-DEC-75    | m   | Italian       |
| 13 | 212        | Chen     | Li       | 5000    | 28-SEP-82    | f   | Chinese       |
| 14 | 213        | Lee      | Min-ji   | 4600    | 12-FEB-87    | f   | Korean        |
| 15 | 214        | Garcia   | Juan     | 5100    | 05-OCT-84    | m   | Spanish       |
| 16 | 215        | Müller   | Laura    | 5400    | 18-JUN-86    | f   | German        |
| 17 | 216        | Sato     | Hiroshi  | 4900    | 09-SEP-83    | m   | Japanese      |
| 18 | 217        | Russo    | Anna     | 4700    | 25-APR-85    | f   | Italian       |
| 19 | 218        | Kowalski | Adam     | 5200    | 21-AUG-80    | m   | Polish        |
| 20 | 219        | López    | Carmen   | 4800    | 07-MAR-82    | f   | Spanish       |

```

INSERT INTO SPECIALIZARE (id_specializare, data_infiintarii, nume_s)

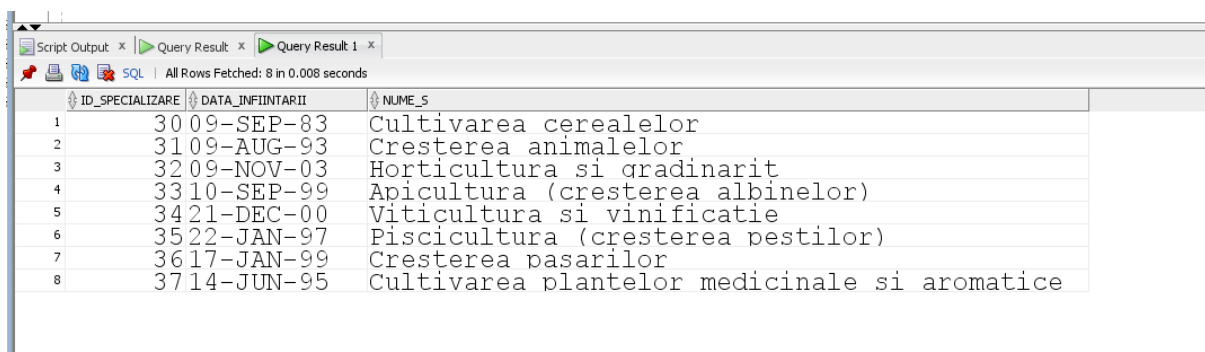
```



```

VALUES (SEQ_SPECIALIZARE.NEXTVAL, to_date('1983-09-09','yyyy-mm-dd'),
'Cultivarea cerealelor');
INSERT INTO SPECIALIZARE (id_specializare, data_infiintarii, nume_s)
VALUES (SEQ_SPECIALIZARE.NEXTVAL, to_date('1993-08-09','yyyy-mm-dd'),
'Cresterea animalelor');
INSERT INTO SPECIALIZARE (id_specializare,data_infiintarii, nume_s)
VALUES (SEQ_SPECIALIZARE.NEXTVAL, to_date('2003-11-09','yyyy-mm-dd'),
'Horticultura si gradinarit');
INSERT INTO SPECIALIZARE (id_specializare, data_infiintarii, nume_s)
VALUES (SEQ_SPECIALIZARE.NEXTVAL, to_date('1999-09-10','yyyy-mm-dd'),
'Apicultura (cresterea albinelor)');
INSERT INTO SPECIALIZARE (id_specializare, data_infiintarii, nume_s)
VALUES (SEQ_SPECIALIZARE.NEXTVAL, to_date('2000-12-21','yyyy-mm-dd'),
'Viticultura si vinificatie');
INSERT INTO SPECIALIZARE (id_specializare, data_infiintarii, nume_s)
VALUES (SEQ_SPECIALIZARE.NEXTVAL, to_date('1997-01-22','yyyy-mm-dd'),
'Piscicultura (cresterea pestilor)');
INSERT INTO SPECIALIZARE (id_specializare, data_infiintarii, nume_s)
VALUES (SEQ_SPECIALIZARE.NEXTVAL, to_date('1999-01-17','yyyy-mm-dd'),
'Cresterea pasarilor');
INSERT INTO SPECIALIZARE (id_specializare, data_infiintarii, nume_s)
VALUES (SEQ_SPECIALIZARE.NEXTVAL, to_date('1995-06-14','yyyy-mm-dd'),
'Cultivarea plantelor medicinale si aromatice');

```



The screenshot shows a SQL query result window with the following data:

| ID_SPECIALIZARE | DATA_INFIINTARII | NUME_S                                       |
|-----------------|------------------|--|
| 1               | 30 09-SEP-83     | Cultivarea cerealelor                        |
| 2               | 31 09-AUG-93     | Cresterea animalelor                         |
| 3               | 32 09-NOV-03     | Horticultura si gradinarit                   |
| 4               | 33 10-SEP-99     | Apicultura (cresterea albinelor)             |
| 5               | 34 21-DEC-00     | Viticultura si vinificatie                   |
| 6               | 35 22-JAN-97     | Piscicultura (cresterea pestilor)            |
| 7               | 36 17-JAN-99     | Cresterea pasarilor                          |
| 8               | 37 14-JUN-95     | Cultivarea plantelor medicinale si aromatice |

```

INSERT INTO CAMION_DE_MARFA (id_camion, capacitate,id_angajat)
VALUES(SEQ_CAMION_DE_MARFA.NEXTVAL, 50,205);
INSERT INTO CAMION_DE_MARFA (id_camion, capacitate,id_angajat)
VALUES(SEQ_CAMION_DE_MARFA.NEXTVAL, 60,206);
INSERT INTO CAMION_DE_MARFA (id_camion, capacitate,id_angajat)
VALUES(SEQ_CAMION_DE_MARFA.NEXTVAL, 45,207);
INSERT INTO CAMION_DE_MARFA (id_camion, capacitate, id_angajat)
VALUES(SEQ_CAMION_DE_MARFA.NEXTVAL, 55,208);
INSERT INTO CAMION_DE_MARFA (id_camion, capacitate, id_angajat)
VALUES(SEQ_CAMION_DE_MARFA.NEXTVAL, 4,209);

```

|   | ID_CAMION | CAPACIT... | ID_ANGAJAT |
|---|-----------|------------|------------|
| 1 | 40        | 60         | 206        |
| 2 | 41        | 50         | 205        |
| 3 | 42        | 45         | 207        |
| 4 | 43        | 55         | 208        |
| 5 | 44        | 4          | 209        |

```

INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii, productivitate,
id_angajat )
VALUES (SEQ_ISTORIC.NEXTVAL, TO_DATE('15-01-2020', 'dd-mm-yyyy'),
TO_DATE('10-03-2022', 'dd-mm-yyyy'), 'crescuta',200);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('20-06-2019', 'dd-mm-yyyy'),
TO_DATE('05-09-2021', 'dd-mm-yyyy'), 'scazuta',201);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('01-02-2021', 'dd-mm-yyyy'), NULL,
'crescuta',202);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('10-09-2020', 'dd-mm-yyyy'),
TO_DATE('01-05-2022', 'dd-mm-yyyy'), 'crescuta',203);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('25-11-2019', 'dd-mm-yyyy'), NULL,
'scazuta',204);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('03-04-2020', 'dd-mm-yyyy'),
TO_DATE('15-01-2022', 'dd-mm-yyyy'), 'crescuta',205);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('01-03-2021', 'dd-mm-yyyy'),
TO_DATE('20-04-2022', 'dd-mm-yyyy'), 'crescuta',206);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('10-07-2005', 'dd-mm-yyyy'), NULL,
'scazuta',207);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('10-06-2020', 'dd-mm-yyyy'),
TO_DATE('28-07-2022', 'dd-mm-yyyy'), 'scazuta',208);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('28-07-2011', 'dd-mm-yyyy'),
TO_DATE('18-09-2022', 'dd-mm-yyyy'), 'scazuta',209);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)

```

```

VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('11-07-2011', 'dd-mm-yyyy'),
TO_DATE('18-08-2022', 'dd-mm-yyyy'), 'scazuta',210);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('09-06-2004', 'dd-mm-yyyy'),
TO_DATE('28-02-2022', 'dd-mm-yyyy'), 'crescuta',211);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('10-03-2003', 'dd-mm-yyyy'), NULL,
'scazuta',212);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('19-07-2019', 'dd-mm-yyyy'),
TO_DATE('21-04-2022', 'dd-mm-yyyy'), 'scazuta',213);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('10-09-2020', 'dd-mm-yyyy'),
TO_DATE('16-11-2022', 'dd-mm-yyyy'), 'crescuta',214);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('18-02-2015', 'dd-mm-yyyy'), NULL,
'scazuta',215);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('10-01-2015', 'dd-mm-yyyy'),
TO_DATE('18-04-2022', 'dd-mm-yyyy'), 'crescuta',216);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('02-05-2018', 'dd-mm-yyyy'), NULL,
'crescuta',217);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('01-08-2019', 'dd-mm-yyyy'),
TO_DATE('19-09-2022', 'dd-mm-yyyy'), 'scazuta',218);
INSERT INTO ISTORIC (id_istoric, data_angajarii, data_promovarii,
productivitate,id_angajat)
VALUES(SEQ_ISTORIC.NEXTVAL, TO_DATE('11-09-2005', 'dd-mm-yyyy'),
TO_DATE('01-05-2022', 'dd-mm-yyyy'), 'crescuta',219);

```

| ID_ISTORIC | DATA_ANGAJARII | DATA_PROMOVARII | PRODUCTIVITATE | ID_ANGAJAT |
|------------|----------------|-----------------|----------------|------------|
| 1          | 50015-JAN-20   | 10-MAR-22       | crescuta       | 200        |
| 2          | 50120-JUN-19   | 05-SEP-21       | scazuta        | 201        |
| 3          | 50225-NOV-19   | (null)          | scazuta        | 204        |
| 4          | 50303-APR-20   | 15-JAN-22       | crescuta       | 205        |
| 5          | 50401-MAR-21   | 20-APR-22       | crescuta       | 206        |
| 6          | 50510-JUL-05   | (null)          | scazuta        | 207        |
| 7          | 50610-JUN-20   | 28-JUL-22       | scazuta        | 208        |
| 8          | 50728-JUL-11   | 18-SEP-22       | scazuta        | 209        |
| 9          | 50811-JUL-11   | 18-AUG-22       | scazuta        | 210        |
| 10         | 50909-JUN-04   | 28-FEB-22       | crescuta       | 211        |
| 11         | 51010-MAR-03   | (null)          | scazuta        | 212        |
| 12         | 51119-JUL-19   | 21-APR-22       | scazuta        | 213        |
| 13         | 51210-SEP-20   | 16-NOV-22       | crescuta       | 214        |
| 14         | 51318-FEB-15   | (null)          | scazuta        | 215        |
| 15         | 51410-JAN-15   | 18-APR-22       | crescuta       | 216        |
| 16         | 51502-MAY-18   | (null)          | crescuta       | 217        |
| 17         | 51601-AUG-19   | 19-SEP-22       | scazuta        | 218        |
| 18         | 51711-SEP-05   | 01-MAY-22       | crescuta       | 219        |
| 19         | 51801-FEB-21   | (null)          | crescuta       | 202        |
| 20         | 51910-SEP-20   | 01-MAY-22       | crescuta       | 203        |

```

INSERT INTO ACTIVITATE (id_activitate, data_desfasurare, ora)
VALUES(SEQ_ACTIVITATE.NEXTVAL, TO_DATE('2023-01-15', 'yyyy-mm-dd'),
'09:00');
INSERT INTO ACTIVITATE (id_activitate, data_desfasurare, ora)
VALUES(SEQ_ACTIVITATE.NEXTVAL, TO_DATE('2023-02-20', 'yyyy-mm-dd'),
'14:30');
INSERT INTO ACTIVITATE (id_activitate, data_desfasurare, ora)
VALUES(SEQ_ACTIVITATE.NEXTVAL, TO_DATE('2023-03-10', 'yyyy-mm-dd'),
'16:00');
INSERT INTO ACTIVITATE (id_activitate, data_desfasurare, ora)
VALUES(SEQ_ACTIVITATE.NEXTVAL, TO_DATE('2023-04-05', 'yyyy-mm-dd'),
'11:15');
INSERT INTO ACTIVITATE (id_activitate, data_desfasurare, ora)
VALUES(SEQ_ACTIVITATE.NEXTVAL, TO_DATE('2023-05-12', 'yyyy-mm-dd'),
'13:45');
INSERT INTO ACTIVITATE (id_activitate, data_desfasurare, ora)
VALUES(SEQ_ACTIVITATE.NEXTVAL, TO_DATE('2023-06-18', 'yyyy-mm-dd'),
'10:30');
INSERT INTO ACTIVITATE (id_activitate, data_desfasurare, ora)
VALUES(SEQ_ACTIVITATE.NEXTVAL, TO_DATE('2023-07-25', 'yyyy-mm-dd'),
'15:00');
INSERT INTO ACTIVITATE (id_activitate, data_desfasurare, ora)
VALUES(SEQ_ACTIVITATE.NEXTVAL, TO_DATE('2023-08-30', 'yyyy-mm-dd'),
'17:30');

```

| Query Result x                             |               |                  |       |
|--|---------------|------------------|-------|
| SQL   All Rows Fetched: 8 in 0.005 seconds |               |                  |       |
|  | ID_ACTIVITATE | DATA_DESFASURARE | ORA   |
| 1  | 60            | 15-JAN-23        | 09:00 |
| 2  | 61            | 20-FEB-23        | 14:30 |
| 3  | 62            | 10-MAR-23        | 16:00 |
| 4  | 63            | 05-APR-23        | 11:15 |
| 5  | 64            | 12-MAY-23        | 13:45 |
| 6  | 65            | 18-JUN-23        | 10:30 |
| 7  | 66            | 25-JUL-23        | 15:00 |
| 8  | 67            | 30-AUG-23        | 17:30 |

```

INSERT INTO UTILIZATOR (id_user, nume, prenume)
VALUES(SEQ_UTILIZATOR.NEXTVAL, 'Smith', 'John');
INSERT INTO UTILIZATOR (id_user, nume, prenume)
VALUES(SEQ_UTILIZATOR.NEXTVAL, 'Johnson', 'Sarah');
INSERT INTO UTILIZATOR (id_user, nume, prenume)
VALUES(SEQ_UTILIZATOR.NEXTVAL, 'Williams', 'David');
INSERT INTO UTILIZATOR (id_user, nume, prenume)
VALUES(SEQ_UTILIZATOR.NEXTVAL, 'Brown', 'Emily');
INSERT INTO UTILIZATOR (id_user, nume, prenume)
VALUES(SEQ_UTILIZATOR.NEXTVAL, 'Jones', 'Michael');
INSERT INTO UTILIZATOR (id_user, nume, prenume)
VALUES(SEQ_UTILIZATOR.NEXTVAL, 'Taylor', 'Jessica');
INSERT INTO UTILIZATOR (id_user, nume, prenume)
VALUES(SEQ_UTILIZATOR.NEXTVAL, 'Davis', 'Daniel');
INSERT INTO UTILIZATOR (id_user, nume, prenume)
VALUES(SEQ_UTILIZATOR.NEXTVAL, 'Miller', 'Olivia');

```

| Query Result x                             |         |          |         |
|--|---------|----------|---------|
| SQL   All Rows Fetched: 8 in 0.006 seconds |         |          |         |
|  | ID_USER | NUME     | PRENUME |
| 1  | 70      | Smith    | John    |
| 2  | 71      | Johnson  | Sarah   |
| 3  | 72      | Williams | David   |
| 4  | 73      | Brown    | Emily   |
| 5  | 74      | Jones    | Michael |
| 6  | 75      | Taylor   | Jessica |
| 7  | 76      | Davis    | Daniel  |
| 8  | 77      | Miller   | Olivia  |

```

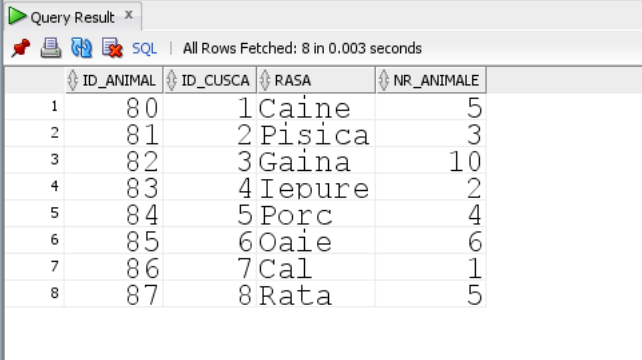
INSERT INTO ANIMAL (id_animal, id_cusca, rasa, nr_animale)
VALUES(SEQ_ANIMAL.NEXTVAL, 1, 'Caine', 5);
INSERT INTO ANIMAL (id_animal, id_cusca, rasa, nr_animale)
VALUES(SEQ_ANIMAL.NEXTVAL, 2, 'Pisica', 3);
INSERT INTO ANIMAL (id_animal, id_cusca, rasa, nr_animale)
VALUES(SEQ_ANIMAL.NEXTVAL, 3, 'Gaina', 10);
INSERT INTO ANIMAL (id_animal, id_cusca, rasa, nr_animale)
VALUES(SEQ_ANIMAL.NEXTVAL, 4, 'Iepure', 2);

```

```

INSERT INTO ANIMAL (id_animal, id_cusca, rasa, nr_animale)
VALUES(SEQ_ANIMAL.NEXTVAL, 5, 'Porc', 4);
INSERT INTO ANIMAL (id_animal, id_cusca, rasa, nr_animale)
VALUES(SEQ_ANIMAL.NEXTVAL, 6, 'Oaie', 6);
INSERT INTO ANIMAL (id_animal, id_cusca, rasa, nr_animale)
VALUES(SEQ_ANIMAL.NEXTVAL, 7, 'Cal', 1);
INSERT INTO ANIMAL (id_animal, id_cusca, rasa, nr_animale)
VALUES(SEQ_ANIMAL.NEXTVAL, 8, 'Rata', 5);

```



Query Result x

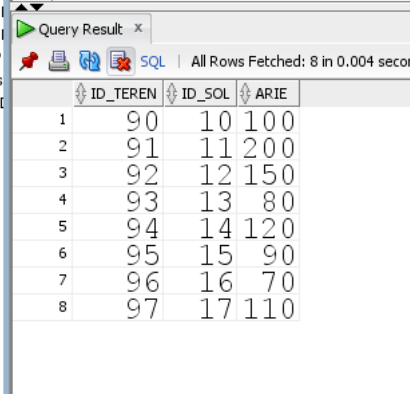
All Rows Fetched: 8 in 0.003 seconds

|   | ID_ANIMAL | ID_CUSCA | RASA   | NR_ANIMALE |
|---|-----------|----------|--------|------------|
| 1 | 80        | 1        | Caine  | 5          |
| 2 | 81        | 2        | Pisica | 3          |
| 3 | 82        | 3        | Gaina  | 10         |
| 4 | 83        | 4        | Iepure | 2          |
| 5 | 84        | 5        | Porc   | 4          |
| 6 | 85        | 6        | Oaie   | 6          |
| 7 | 86        | 7        | Cal    | 1          |
| 8 | 87        | 8        | Rata   | 5          |

```

INSERT INTO TERENCE (id_teren, id_sol, arie)
VALUES(SEQ_TERENCE.NEXTVAL, 10, 100);
INSERT INTO TERENCE (id_teren, id_sol, arie)
VALUES(SEQ_TERENCE.NEXTVAL, 11, 200);
INSERT INTO TERENCE (id_teren, id_sol, arie)
VALUES(SEQ_TERENCE.NEXTVAL, 12, 150);
INSERT INTO TERENCE (id_teren, id_sol, arie)
VALUES(SEQ_TERENCE.NEXTVAL, 13, 80);
INSERT INTO TERENCE (id_teren, id_sol, arie)
VALUES(SEQ_TERENCE.NEXTVAL, 14, 120);
INSERT INTO TERENCE (id_teren, id_sol, arie)
VALUES(SEQ_TERENCE.NEXTVAL, 15, 90);
INSERT INTO TERENCE (id_teren, id_sol, arie)
VALUES(SEQ_TERENCE.NEXTVAL, 16, 70);
INSERT INTO TERENCE (id_teren, id_sol, arie)
VALUES(SEQ_TERENCE.NEXTVAL, 17, 110);

```



Query Result x

All Rows Fetched: 8 in 0.004 seconds

|   | ID_TERENCE | ID_SOL | ARIE |
|---|------------|--------|------|
| 1 | 90         | 10     | 100  |
| 2 | 91         | 11     | 200  |
| 3 | 92         | 12     | 150  |
| 4 | 93         | 13     | 80   |
| 5 | 94         | 14     | 120  |
| 6 | 95         | 15     | 90   |
| 7 | 96         | 16     | 70   |
| 8 | 97         | 17     | 110  |

```

INSERT INTO PLANTA (id_planta, id_sol, id_teren, calitate, cantitate)

```

```

VALUES(SEQ_PLANTA.NEXTVAL, 10,90, 'intai', 100);
INSERT INTO PLANTA (id_planta, id_sol, id_teren, calitate, cantitate)
VALUES(SEQ_PLANTA.NEXTVAL, 11, 91, 'doi', 150);
INSERT INTO PLANTA (id_planta, id_sol, id_teren, calitate, cantitate)
VALUES(SEQ_PLANTA.NEXTVAL, 13, 92, 'trei', 200);
INSERT INTO PLANTA (id_planta, id_sol, id_teren, calitate, cantitate)
VALUES(SEQ_PLANTA.NEXTVAL, 17, 93, 'intai', 80);
INSERT INTO PLANTA (id_planta, id_sol, id_teren, calitate, cantitate)
VALUES(SEQ_PLANTA.NEXTVAL, 12, 94, 'doi', 120);
INSERT INTO PLANTA (id_planta, id_sol, id_teren, calitate, cantitate)
VALUES(SEQ_PLANTA.NEXTVAL, 10, 95, 'trei', 90);
INSERT INTO PLANTA (id_planta, id_sol, id_teren, calitate, cantitate)
VALUES(SEQ_PLANTA.NEXTVAL, 15,96, 'intai', 70);
INSERT INTO PLANTA (id_planta, id_sol, id_teren, calitate, cantitate)
VALUES(SEQ_PLANTA.NEXTVAL, 15,97, 'doi', 110);

```

select \* from utilizator;

Query Result x

All Rows Fetched: 8 in 0.005 seconds

|   | ID_PLANTA | ID_SOL | ID_TEREN | CALITATE | CANTITATE |
|---|-----------|--------|----------|----------|-----------|
| 1 | 100       | 10     | 90       | intai    | 100       |
| 2 | 101       | 11     | 91       | doi      | 150       |
| 3 | 102       | 13     | 92       | trei     | 200       |
| 4 | 103       | 17     | 93       | intai    | 80        |
| 5 | 104       | 12     | 94       | doi      | 120       |
| 6 | 105       | 10     | 95       | trei     | 90        |
| 7 | 106       | 15     | 96       | intai    | 70        |
| 8 | 107       | 15     | 97       | doi      | 110       |

```

INSERT INTO COMANDA (id_comanda, id_user, adresa, nr_telefon, id_angajat)
VALUES(SEQ_COMANDA.NEXTVAL, 70, 'Adresa 1', 1234567890, 215);
INSERT INTO COMANDA (id_comanda, id_user, adresa, nr_telefon, id_angajat)
VALUES(SEQ_COMANDA.NEXTVAL, 71, 'Adresa 2', 9876543210, 216);
INSERT INTO COMANDA (id_comanda, id_user, adresa, nr_telefon, id_angajat)
VALUES(SEQ_COMANDA.NEXTVAL, 72, 'Adresa 3', 4567890123, 217);
INSERT INTO COMANDA (id_comanda, id_user, adresa, nr_telefon, id_angajat)
VALUES(SEQ_COMANDA.NEXTVAL, 73, 'Adresa 4', 7890123456, 218);
INSERT INTO COMANDA (id_comanda, id_user, adresa, nr_telefon, id_angajat)
VALUES(SEQ_COMANDA.NEXTVAL, 74, 'Adresa 5', 2345678901, 219);
INSERT INTO COMANDA (id_comanda, id_user, adresa, nr_telefon, id_angajat)
VALUES(SEQ_COMANDA.NEXTVAL, 74, 'Adresa 6', 9345678901, 219);
INSERT INTO COMANDA (id_comanda, id_user, adresa, nr_telefon, id_angajat)
VALUES(SEQ_COMANDA.NEXTVAL, 72, 'Adresa 7', 2945678001, 217);
INSERT INTO COMANDA (id_comanda, id_user, adresa, nr_telefon, id_angajat)
VALUES(SEQ_COMANDA.NEXTVAL, 74, 'Adresa 8', 2346678901, 219);

```



| ID_COMANDA | ID_USER | ADRESA     | NR_TELEFON | ID_ANGAJAT |
|------------|---------|------------|------------|------------|
| 1          | 110     | 70Adresa 1 | 1234567890 | 215        |
| 2          | 111     | 71Adresa 2 | 9876543210 | 216        |
| 3          | 112     | 72Adresa 3 | 4567890123 | 217        |
| 4          | 113     | 73Adresa 4 | 7890123456 | 218        |
| 5          | 114     | 74Adresa 5 | 2345678901 | 219        |
| 6          | 115     | 74Adresa 6 | 9345678901 | 219        |
| 7          | 116     | 72Adresa 7 | 2945678001 | 217        |
| 8          | 117     | 74Adresa 8 | 2346678901 | 219        |

```

INSERT INTO ARE_ (id_sol, id_teren) VALUES (10, 90);
INSERT INTO ARE_ (id_sol, id_teren) VALUES (12, 91);
INSERT INTO ARE_ (id_sol, id_teren) VALUES (10, 92);
INSERT INTO ARE_ (id_sol, id_teren) VALUES (12, 93);
INSERT INTO ARE_ (id_sol, id_teren) VALUES (12, 94);
INSERT INTO ARE_ (id_sol, id_teren) VALUES (12, 95);
INSERT INTO ARE_ (id_sol, id_teren) VALUES (13, 96);
INSERT INTO ARE_ (id_sol, id_teren) VALUES (13, 97);
INSERT INTO ARE_ (id_sol, id_teren) VALUES (13, 91);
INSERT INTO ARE_ (id_sol, id_teren) VALUES (14, 95);
INSERT INTO ARE_ (id_sol, id_teren) VALUES (14, 96);
INSERT INTO ARE_ (id_sol, id_teren) VALUES (16, 95);

```

| ID_SOL | ID_TEREN |
|--------|----------|
| 1      | 10       |
| 2      | 10       |
| 3      | 12       |
| 4      | 12       |
| 5      | 12       |
| 6      | 12       |
| 7      | 13       |
| 8      | 13       |
| 9      | 13       |
| 10     | 14       |
| 11     | 14       |
| 12     | 16       |

```

INSERT INTO FERMIER (id_angajat, certificari)
VALUES (200, 'Certificat1');
INSERT INTO FERMIER (id_angajat, certificari)
VALUES (201, 'Certificat2');
INSERT INTO FERMIER (id_angajat, certificari)
VALUES (202, 'Certificat3');
INSERT INTO FERMIER (id_angajat, certificari)
VALUES (203, 'Certificat4');

```



```
INSERT INTO FERMIER (id_angajat, certificari)
VALUES (204, 'Certificat5');
```

Query Result x

All Rows Fetched: 5 in 0.011 seconds

|   | ID_ANGAJAT | CERTIFICARI |
|---|------------|-------------|
| 1 | 200        | Certificat1 |
| 2 | 201        | Certificat2 |
| 3 | 202        | Certificat3 |
| 4 | 203        | Certificat4 |
| 5 | 204        | Certificat5 |

```
INSERT INTO SOFER (id_angajat, categori_permiss)
VALUES (205, 'B');
INSERT INTO SOFER (id_angajat, categori_permiss)
VALUES (206, 'B, C');
INSERT INTO SOFER (id_angajat, categori_permiss)
VALUES (207, 'B, C, D');
INSERT INTO SOFER (id_angajat, categori_permiss)
VALUES (208, 'C');
INSERT INTO SOFER (id_angajat, categori_permiss)
VALUES (209, 'D');
```

Query Result x

All Rows Fetched: 5 in 0.006 seconds

|   | ID_ANGAJAT | CATEGORIE_PERMISS |
|---|------------|-------------------|
| 1 | 205        | B                 |
| 2 | 206        | B, C              |
| 3 | 207        | B, C, D           |
| 4 | 208        | C                 |
| 5 | 209        | D                 |

```
INSERT INTO CONTABIL (id_angajat, educatie)
VALUES (210, 'Facultatea de Economie');
INSERT INTO CONTABIL (id_angajat, educatie)
VALUES (211, 'Facultatea de Contabilitate si Informatica de Gestiune');
INSERT INTO CONTABIL (id_angajat, educatie)
VALUES (212, 'Facultatea de Finante si Banci');
INSERT INTO CONTABIL (id_angajat, educatie)
VALUES (213, 'Facultatea de Administratie si Afaceri');
INSERT INTO CONTABIL (id_angajat, educatie)
VALUES (214, 'Facultatea de Audit si Controlul Afacerilor');
```

Query Result x

All Rows Fetched: 5 in 0.007 seconds

| ID_ANGAJAT | EDUCATIE   |
|------------|--|
| 210        | Facultatea de Economie                                 |
| 211        | Facultatea de Contabilitate si Informatica de Gestiune |
| 212        | Facultatea de Finante si Banci                         |
| 213        | Facultatea de Administratie si Afaceri                 |
| 214        | Facultatea de Audit si Controlul Afacerilor            |

```

INSERT INTO COMERCIANT (id_angajat, volumul_de_vanzari)
VALUES (215, 'mediu');
INSERT INTO COMERCIANT (id_angajat, volumul_de_vanzari)
VALUES (216, 'crescut');
INSERT INTO COMERCIANT (id_angajat, volumul_de_vanzari)
VALUES (217, 'scazut');
INSERT INTO COMERCIANT (id_angajat, volumul_de_vanzari)
VALUES (218, 'mediu');
INSERT INTO COMERCIANT (id_angajat, volumul_de_vanzari)
VALUES (219, 'crescut');

```

Query Result x

All Rows Fetched: 5 in 0.005 seconds

| ID_ANGAJAT | VOLUMUL_DE_VANZARI |
|------------|--------------------|
| 215        | mediu              |
| 216        | crescut            |
| 217        | scazut             |
| 218        | mediu              |
| 219        | crescut            |

```

INSERT INTO INGRIJESTE (id_sol, id_teren, id_planta, id_cusca, id_animal, id_angajat)
VALUES (10, 91, 101, 1, 80, 200);
INSERT INTO INGRIJESTE (id_sol, id_teren, id_planta, id_cusca, id_animal, id_angajat)
VALUES (17, 92, 102, 2, 82, 201);
INSERT INTO INGRIJESTE (id_sol, id_teren, id_planta, id_cusca, id_animal, id_angajat)
VALUES (16, 93, 103, 3, 81, 202);
INSERT INTO INGRIJESTE (id_sol, id_teren, id_planta, id_cusca, id_animal, id_angajat)
VALUES (13, 94, 104, 4, 83, 203);
INSERT INTO INGRIJESTE (id_sol, id_teren, id_planta, id_cusca, id_animal, id_angajat)
VALUES (14, 95, 105, 5, 87, 204);
INSERT INTO INGRIJESTE (id_sol, id_teren, id_planta, id_cusca, id_animal, id_angajat)
VALUES (11, 96, 106, 6, 83, 200);
INSERT INTO INGRIJESTE (id_sol, id_teren, id_planta, id_cusca, id_animal, id_angajat)
VALUES (16, 97, 107, 7, 84, 201);
INSERT INTO INGRIJESTE (id_sol, id_teren, id_planta, id_cusca, id_animal, id_angajat)
VALUES (15, 97, 100, 8, 86, 202);
INSERT INTO INGRIJESTE (id_sol, id_teren, id_planta, id_cusca, id_animal, id_angajat)
VALUES (15, 90, 100, 7, 85, 203);
INSERT INTO INGRIJESTE (id_sol, id_teren, id_planta, id_cusca, id_animal, id_angajat)
VALUES (12, 96, 101, 6, 84, 204);

```

| ID_SOL | ID_TEREN | ID_PLANTA | ID_CUSCA | ID_ANIMAL | ID_ANGAJAT |
|--------|----------|-----------|----------|-----------|------------|
| 1      | 10       | 91        | 101      | 1         | 80         |
| 2      | 11       | 96        | 106      | 6         | 83         |
| 3      | 12       | 96        | 101      | 6         | 84         |
| 4      | 13       | 94        | 104      | 4         | 83         |
| 5      | 14       | 95        | 105      | 5         | 87         |
| 6      | 15       | 90        | 100      | 7         | 85         |
| 7      | 15       | 97        | 100      | 8         | 86         |
| 8      | 16       | 93        | 103      | 3         | 81         |
| 9      | 16       | 97        | 107      | 7         | 84         |
| 10     | 17       | 92        | 102      | 2         | 82         |

```

INSERT INTO URMEAZA (id_specializare, id_angajat, experienta_in_domeniu)
VALUES (30, 200, 2);
INSERT INTO URMEAZA (id_specializare, id_angajat, experienta_in_domeniu)
VALUES (31, 201, 3);
INSERT INTO URMEAZA (id_specializare, id_angajat, experienta_in_domeniu)
VALUES (32, 202, 1);
INSERT INTO URMEAZA (id_specializare, id_angajat, experienta_in_domeniu)
VALUES (33, 203, 4);
INSERT INTO URMEAZA (id_specializare, id_angajat, experienta_in_domeniu)
VALUES (34, 204, 2);
INSERT INTO URMEAZA (id_specializare, id_angajat, experienta_in_domeniu)
VALUES (35, 200, 5);
INSERT INTO URMEAZA (id_specializare, id_angajat, experienta_in_domeniu)
VALUES (36, 201, 3);
INSERT INTO URMEAZA (id_specializare, id_angajat, experienta_in_domeniu)
VALUES (37, 202, 2);
INSERT INTO URMEAZA (id_specializare, id_angajat, experienta_in_domeniu)
VALUES (37, 203, 1);
INSERT INTO URMEAZA (id_specializare, id_angajat, experienta_in_domeniu)
VALUES (30, 204, 4);

```

| ID_SPECIALIZARE | ID_ANGAJAT | EXPERIENTA_IN_DOMENIU |
|-----------------|------------|-----------------------|
| 1               | 30         | 200                   |
| 2               | 31         | 201                   |
| 3               | 32         | 202                   |
| 4               | 33         | 203                   |
| 5               | 34         | 204                   |
| 6               | 35         | 200                   |
| 7               | 36         | 201                   |
| 8               | 37         | 202                   |
| 9               | 37         | 203                   |
| 10              | 30         | 204                   |

```

INSERT INTO ORGANIZEAZA (id_activitate, id_angajat, locatie)
VALUES (60, 210, 'Hamaru Farm - Barn 1');
INSERT INTO ORGANIZEAZA (id_activitate, id_angajat, locatie)
VALUES (61, 211, 'Hamaru Farm - Greenhouse');
INSERT INTO ORGANIZEAZA (id_activitate, id_angajat, locatie)
VALUES (62, 212, 'Hamaru Farm - Livestock Area');
INSERT INTO ORGANIZEAZA (id_activitate, id_angajat, locatie)
VALUES (63, 213, 'Hamaru Farm - Field 3');
INSERT INTO ORGANIZEAZA (id_activitate, id_angajat, locatie)
VALUES (64, 214, 'Hamaru Farm - Orchard');
INSERT INTO ORGANIZEAZA (id_activitate, id_angajat, locatie)
VALUES (65, 210, 'Green Fields Farm - Milking Parlor');
INSERT INTO ORGANIZEAZA (id_activitate, id_angajat, locatie)
VALUES (66, 211, 'Green Fields Farm - Crop Storage');
INSERT INTO ORGANIZEAZA (id_activitate, id_angajat, locatie)
VALUES (67, 212, 'Green Fields Farm - Poultry House');
INSERT INTO ORGANIZEAZA (id_activitate, id_angajat, locatie)
VALUES (67, 213, 'Green Fields Farm - Barn 2');
INSERT INTO ORGANIZEAZA (id_activitate, id_angajat, locatie)
VALUES (60, 214, 'Green Fields Farm - Vegetable Garden');

```

Query Results | All Rows Fetched: 10 in 0.006 seconds

|    | ID_ACTIVITATE | ID_ANGAJAT | LOCATIE                              |
|----|---------------|------------|--------------------------------------|
| 1  | 60            | 210        | Hamaru Farm - Barn 1                 |
| 2  | 61            | 211        | Hamaru Farm - Greenhouse             |
| 3  | 62            | 212        | Hamaru Farm - Livestock Area         |
| 4  | 63            | 213        | Hamaru Farm - Field 3                |
| 5  | 64            | 214        | Hamaru Farm - Orchard                |
| 6  | 65            | 210        | Green Fields Farm - Milking Parlor   |
| 7  | 66            | 211        | Green Fields Farm - Crop Storage     |
| 8  | 67            | 212        | Green Fields Farm - Poultry House    |
| 9  | 67            | 213        | Green Fields Farm - Barn 2           |
| 10 | 60            | 214        | Green Fields Farm - Vegetable Garden |

## 12)Implementarea a 5 cereri SQL complexe:

--a.)enunț: afisează numele, salariul,certificările, id\_angajat, id\_specializare si experiența fermierilor cu cel mai mic salariu în funcție de specializare. (fermierul cu salariul cel mai mic din fiecare specializare)

--Vom folosi: subcereri sincronizate in care intervin cel puțin 3 tabele  
--tabelele sunt angajat, fermier, urmeaza si specializare

```

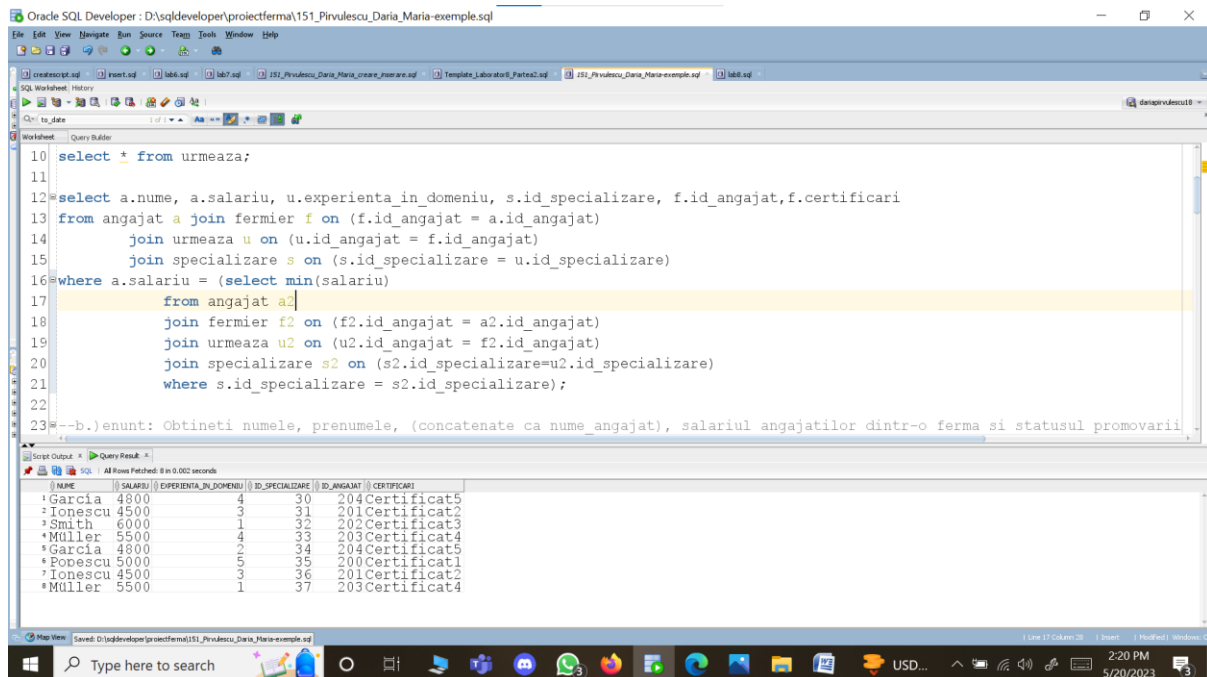
select a.nume, a.salariu, u.experienta_in_domeniu, s.id_specializare, f.id_angajat,f.certificari
from angajat a join fermier f on (f.id_angajat = a.id_angajat)
      join urmeaza u on (u.id_angajat = f.id_angajat)
      join specializare s on (s.id_specializare = u.id_specializare)

```

```

where a.salariu = (select min(salariu)
                  from angajat a2
                  join fermier f2 on (f2.id_angajat = a2.id_angajat)
                  join urmeaza u2 on (u2.id_angajat = f2.id_angajat)
                  join specializare s2 on (s2.id_specializare=u2.id_specializare)
                  where s.id_specializare = s2.id_specializare);

```



--b.)enunț: Obțineți numele, prenumele, (concatenate ca nume\_angajat), salariul angajaților dintr-o fermă și statusul promovării (daca angajatul urmează sau a fost promovat) se va afișa data, în caz contrar se va afișa "Lipsa promovare", atribuind o eticheta salariilor în functie de valoarea lor:

--'Mic' pentru salarii mai mici de 4700, 'Mediu' pentru salarii între 4700 și 5000, si 'Mare' pentru salarii mai mari de 5000.

--Ordonati dupa salarii.

--Vom folosi: ordonari si utilizarea functiilor NVL și DECODE (în cadrul aceleiasi cereri),  
 --utilizarea a unei funcții pe date calendaristice, a unei funcții pe șiruri de caractere și a unei expresii CASE.

```

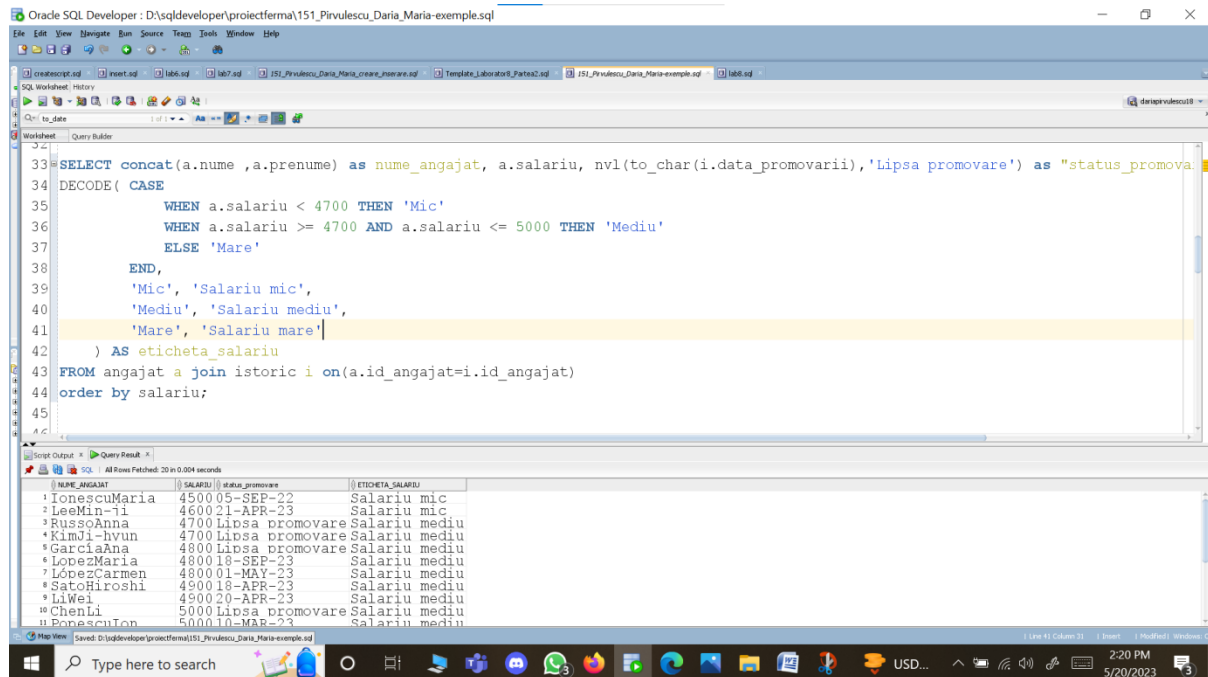
SELECT concat(a.nume ,a.prenume) as nume_angajat, a.salariu,
nvl(to_char(i.data_promovarii),'Lipsa promovare') as "status_promovare",
DECODE( CASE
  WHEN a.salariu < 4700 THEN 'Mic'
  WHEN a.salariu >= 4700 AND a.salariu <= 5000 THEN 'Mediu'
  ELSE 'Mare'
END,

```

```

'Mic', 'Salariu mic',
'Mediu', 'Salariu mediu',
'Mare', 'Salariu mare'
) AS eticheta_salariu
FROM angajat a join istoric i on(a.id_angajat=i.id_angajat)
order by salariu;

```



--c)enunt:Selectați id-ul angajatului, certificările, data de înființare și id-ul specializării pentru angajații fermieri care au experiență în domeniu maximă pentru fiecare specializare, având mai mult de o înregistrare în tabelul URMEAȘĂ și în care data de înființare a specializării este ulterioară datei de 1 ianuarie 1986.

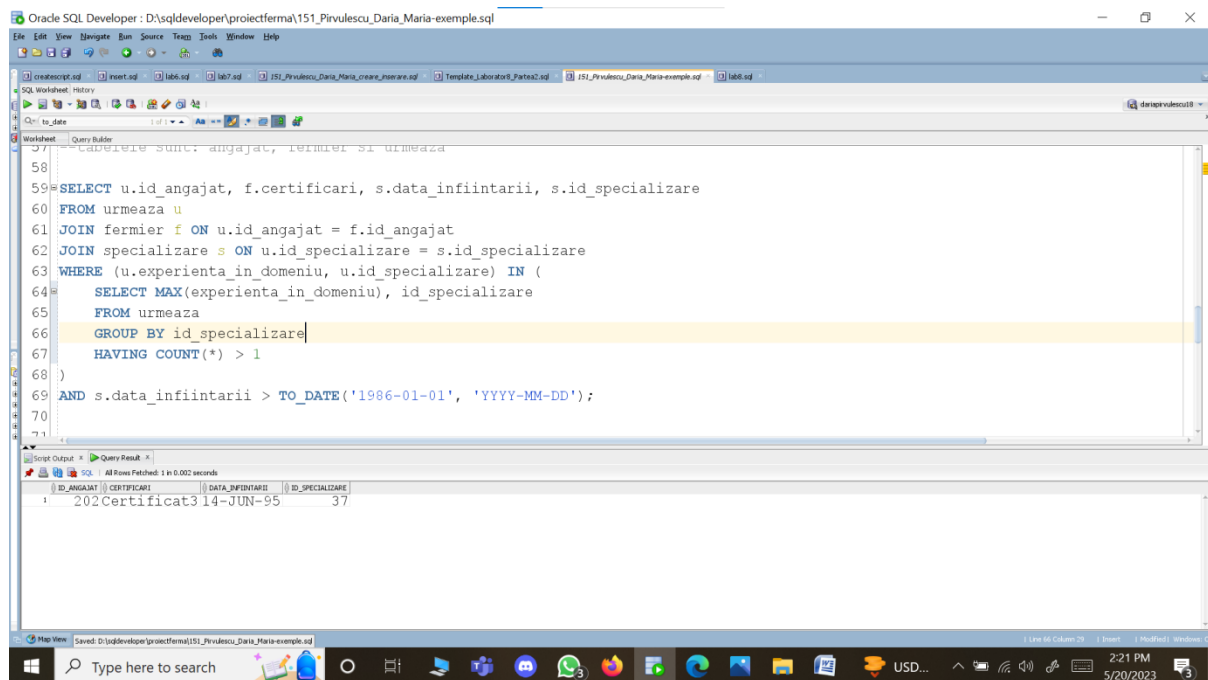
--vom folosi: grupari de date cu subcereri nesincronizate n care intervin cel puțin 3  
--tabele, funcții grup, filtrare la nivel de grupuri (în cadrul aceleiași cereri)  
--o funcție pe siruri de caractere

--tabelele sunt: angajat, fermier si urmeaza

```

SELECT u.id_angajat, f.certificari, s.data_infiintarii, s.id_specializare
FROM urmeaza u
JOIN fermier f ON u.id_angajat = f.id_angajat
JOIN specializare s ON u.id_specializare = s.id_specializare
WHERE (u.experienta_in_domeniu, u.id_specializare) IN (
    SELECT MAX(experienta_in_domeniu), id_specializare
    FROM urmeaza
    GROUP BY id_specializare
    HAVING COUNT(*) > 1
)
AND s.data_infiintarii > TO_DATE('1986-01-01', 'YYYY-MM-DD');

```



--d)enunt: Selectează numele, prenumele și numărul de comenzi ale utilizatorului cu cele mai multe comenzi în tabelul COMANDA.

--vom folosi: 1 bloc de cerere (clauza WITH)

WITH subq AS (

    SELECT id\_user, COUNT(\*) AS numar\_comenzi

    FROM comanda

    GROUP BY id\_user

    ORDER BY COUNT(\*) DESC

)

SELECT u.num, u.prenume, s.numar\_comenzi

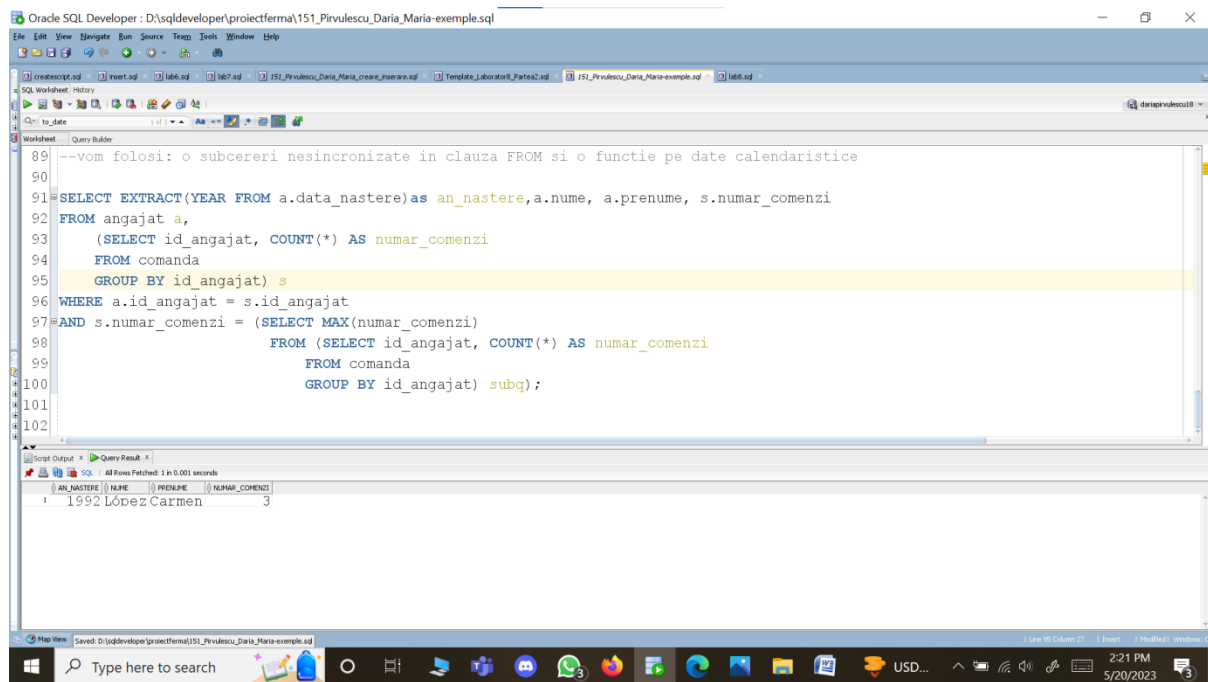
FROM utilizator u

JOIN subq s ON u.id\_user = s.id\_user

WHERE ROWNUM = 1;







13) Implementarea a 3 operatii de actualizare și de suprimare a datelor utilizand subcereri.

#### --1) actualizare

--a.)enunț: să se actualizeze salariul angajaților (să se crească cu 10% din salariul actual) care s-au nascut dupa Popescu Ion.

-- se va folosi comanda update

UPDATE angajat

SET salariu = salariu \*1.1


WHERE data\_nastere >(SELECT data\_nastere

FROM angajat

WHERE upper(nume)='POPESCU' and upper(prenume)='ION');

112 | FROM angajat

Script Output x Query Result x

 All Rows Fetched: 20 in 0.002 seconds

| ID_ANGAJAT | NUME | PRENUME         | SALARIU | DATA_NASTERE | SEX | NATIONALITATE |
|------------|------|-----------------|---------|--------------|-----|---------------|
| 1          | 200  | Popescu Ion     | 5000    | 01-JAN-90    | m   | Romanian      |
| 2          | 201  | Ionescu Maria   | 4950    | 01-APR-99    | f   | Romanian      |
| 3          | 202  | Smith John      | 6000    | 09-JUN-80    | m   | English       |
| 4          | 203  | Müller Hans     | 6050    | 11-SEP-96    | m   | German        |
| 5          | 204  | García Ana      | 5280    | 07-JAN-94    | f   | Spanish       |
| 6          | 205  | Rossi Marco     | 5720    | 01-NOV-98    | m   | Italian       |
| 7          | 206  | Li Wei          | 5390    | 01-AUG-95    | m   | Chinese       |
| 8          | 207  | Kim Ji-hyun     | 4700    | 21-JAN-88    | f   | Korean        |
| 9          | 208  | Popovici Andrei | 5720    | 15-MAR-91    | m   | Romanian      |
| 10         | 209  | Lopez Maria     | 5280    | 20-JUL-93    | f   | Spanish       |
| 11         | 210  | Schmidt Julia   | 5830    | 10-MAY-90    | f   | German        |
| 12         | 211  | Russo Giuseppe  | 5500    | 03-DEC-85    | m   | Italian       |
| 13         | 212  | Chen Li         | 5500    | 28-SEP-92    | f   | Chinese       |
| 14         | 213  | Lee Min-ji      | 5060    | 12-FEB-97    | f   | Korean        |
| 15         | 214  | Garcia Juan     | 5610    | 05-OCT-94    | m   | Spanish       |
| 16         | 215  | Müller Laura    | 5940    | 18-JUN-96    | f   | German        |
| 17         | 216  | Sato Hiroshi    | 5390    | 09-SEP-93    | m   | Japanese      |
| 18         | 217  | Russo Anna      | 5170    | 25-APR-95    | f   | Italian       |
| 19         | 218  | Kowalski Adam   | 5720    | 21-AUG-90    | m   | Polish        |
| 20         | 219  | López Carmen    | 5280    | 07-MAR-92    | f   | Spanish       |

--b.)enunț: Să se micșoreze salariul angajaților care au o productivitate scăzută cu 5%.

--vom folosi update

UPDATE angajat

SET salariu= salariu\*0.95

WHERE id\_angajat in (SELECT id\_angajat

FROM istoric

WHERE productivitate='scazuta');

134 | SET salariu= salariu\*0.95

| Script Output x Query Result x        |            |          |          |         |              |     |               |
|---------------------------------------|------------|----------|----------|---------|--------------|-----|---------------|
| All Rows Fetched: 20 in 0.001 seconds |            |          |          |         |              |     |               |
|                                       | ID_ANGAJAT | NUME     | PRENUME  | SALARIU | DATA_NASTERE | SEX | NATIONALITATE |
| 1                                     | 200        | Popescu  | Ion      | 5000    | 01-JAN-90    | m   | Romanian      |
| 2                                     | 201        | Ionescu  | Maria    | 4275    | 01-APR-99    | f   | Romanian      |
| 3                                     | 202        | Smith    | John     | 6000    | 09-JUN-80    | m   | English       |
| 4                                     | 203        | Müller   | Hans     | 5500    | 11-SEP-96    | m   | German        |
| 5                                     | 204        | García   | Ana      | 4560    | 07-JAN-94    | f   | Spanish       |
| 6                                     | 205        | Rossi    | Marco    | 5200    | 01-NOV-98    | m   | Italian       |
| 7                                     | 206        | Li       | Wei      | 4900    | 01-AUG-95    | m   | Chinese       |
| 8                                     | 207        | Kim      | Ji-hyun  | 4465    | 21-JAN-88    | f   | Korean        |
| 9                                     | 208        | Popovici | Andrei   | 4940    | 15-MAR-91    | m   | Romanian      |
| 10                                    | 209        | Lopez    | Maria    | 4560    | 20-JUL-93    | f   | Spanish       |
| 11                                    | 210        | Schmidt  | Julia    | 5035    | 10-MAY-90    | f   | German        |
| 12                                    | 211        | Russo    | Giuseppe | 5500    | 03-DEC-85    | m   | Italian       |
| 13                                    | 212        | Chen     | Li       | 4750    | 28-SEP-92    | f   | Chinese       |
| 14                                    | 213        | Lee      | Min-ji   | 4370    | 12-FEB-97    | f   | Korean        |
| 15                                    | 214        | Garcia   | Juan     | 5100    | 05-OCT-94    | m   | Spanish       |
| 16                                    | 215        | Müller   | Laura    | 5130    | 18-JUN-96    | f   | German        |
| 17                                    | 216        | Sato     | Hiroshi  | 4900    | 09-SEP-93    | m   | Japanese      |
| 18                                    | 217        | Russo    | Anna     | 4700    | 25-APR-95    | f   | Italian       |
| 19                                    | 218        | Kowalski | Adam     | 4940    | 21-AUG-90    | m   | Polish        |
| 20                                    | 219        | López    | Carmen   | 4800    | 07-MAR-92    | f   | Spanish       |

--c.) enunț: Să se actualizeze ora activităților planificate de contabilul care a mers la Facultatea de Economie.

--Ora va fi marita cu o unitatae

UPDATE activitate

SET ora = TO\_CHAR(TO\_DATE(ora, 'HH24:MI') + INTERVAL '1' HOUR, 'HH24:MI')

WHERE id\_activitate IN (

SELECT o.id\_activitate

FROM organizeaza o

JOIN contabil c ON o.id\_angajat = c.id\_angajat

WHERE UPPER(c.educatie) = 'FACULTATEA DE ECONOMIE'

);

| ID_ACTIVITATE | DATA_DESFASURARE | ORA   |
|---------------|------------------|-------|
| 1             | 60 15-JAN-23     | 10:00 |
| 2             | 61 20-FEB-23     | 14:30 |
| 3             | 62 10-MAR-23     | 16:00 |
| 4             | 63 05-APR-23     | 11:15 |
| 5             | 64 12-MAY-23     | 13:45 |
| 6             | 65 18-JUN-23     | 11:30 |
| 7             | 66 25-JUL-23     | 15:00 |
| 8             | 67 30-AUG-23     | 17:30 |

## --2) Suprimare:

--a.) enunț: Să se șteargă șoferii din tabelul ȘOFER alor căror camioane au capacitatea mai mica de 10 tone

--vom folosi instrucțiunea delete

DELETE FROM sofer

WHERE id\_angajat in (SELECT id\_angajat  
FROM camion\_de\_marfa  
WHERE capacitate < 10);

| ID_ANGAJAT | CATEGORIE_PERMIS |
|------------|------------------|
| 1          | 205 B            |
| 2          | 206 B, C         |
| 3          | 207 B, C, D      |
| 4          | 208 C            |

--b.) enunț: Să se șteargă șoferii români din tabelul șoferi.

DELETE FROM SOFER

WHERE id\_angajat in (SELECT id\_angajat  
FROM ANGAJAT  
WHERE UPPER(nationalitate)='ROMANIAN');

| ID_ANGAJAT | CATEGORIE_PERMIS |
|------------|------------------|
| 1          | 205 B            |
| 2          | 206 B, C         |
| 3          | 207 B, C, D      |
| 4          | 209 D            |

--c) enunț: Să se ștergă din istoric toți angajații de sex masculin

DELETE FROM istoric

WHERE id\_angajat IN (

SELECT id\_angajat

FROM angajat

WHERE sex='m'

);

| ID_ISTORIC | DATA_ANGAJARII | DATA_PROMOVARII | PRODUCTIVITATE | ID_ANGAJAT |
|------------|----------------|-----------------|----------------|------------|
| 1          | 50120-JUN-21   | 05-SEP-22       | scazuta        | 201        |
| 2          | 50225-NOV-21   | (null)          | scazuta        | 204        |
| 3          | 50510-JUL-02   | (null)          | scazuta        | 207        |
| 4          | 50728-JUL-08   | 18-SEP-23       | scazuta        | 209        |
| 5          | 50811-JUL-08   | 18-AUG-23       | scazuta        | 210        |
| 6          | 51010-MAR-00   | (null)          | scazuta        | 212        |
| 7          | 51119-JUL-21   | 21-APR-23       | scazuta        | 213        |
| 8          | 51318-FEB-12   | (null)          | scazuta        | 215        |
| 9          | 51502-MAY-20   | (null)          | crescuta       | 217        |
| 10         | 51711-SEP-02   | 01-MAY-23       | crescuta       | 219        |

14)Formulați în limbaj natural și implementați în SQL:

--a.) o cerere ce utilizează operația outer-join pe minimum 4 tabele

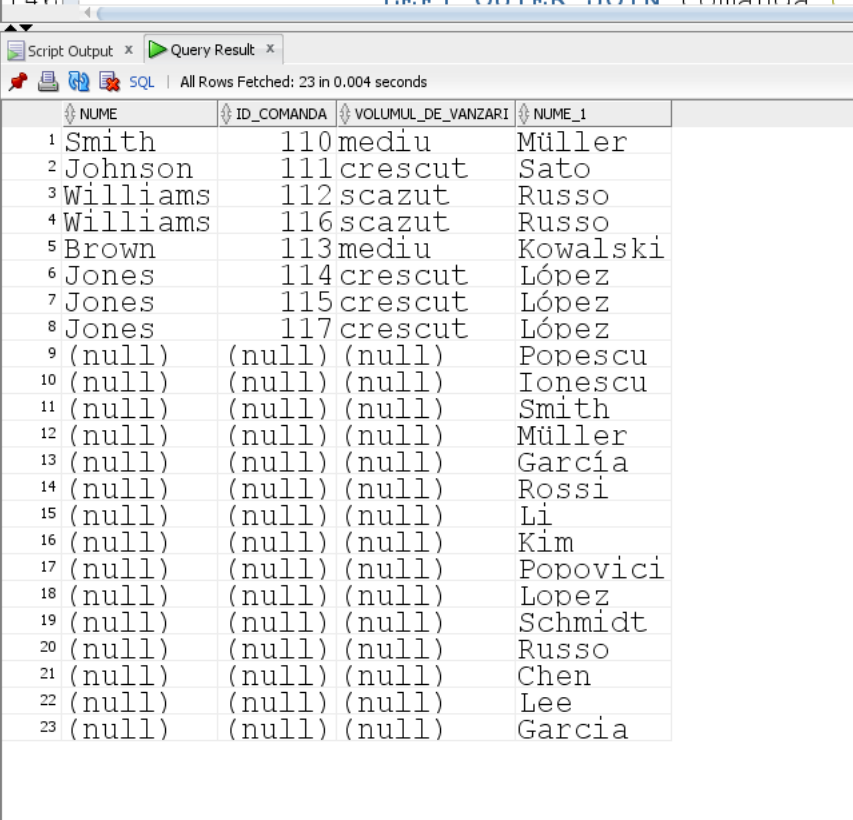
--enunț: Selectați numele utilizatorilor, id-urile comenzilor, volumul de vânzări și numele comercinaților.

SELECT u.num, c.id\_comanda, co.volumul\_de\_vanzari, a.num

FROM angajat a LEFT OUTER JOIN comerciant co on (a.id\_angajat=co.id\_angajat)

LEFT OUTER JOIN comanda c on (co.id\_angajat=c.id\_angajat)

LEFT OUTER JOIN utilizator u on (c.id\_user=u.id\_user);



|    | NUME     | ID_COMANDA | VOLUMUL_DE_VANZARI | NUME_1   |
|----|----------|------------|--------------------|----------|
| 1  | Smith    | 110        | mediu              | Müller   |
| 2  | Johnson  | 111        | crescut            | Sato     |
| 3  | Williams | 112        | scazut             | Russo    |
| 4  | Williams | 116        | scazut             | Russo    |
| 5  | Brown    | 113        | mediu              | Kowalski |
| 6  | Jones    | 114        | crescut            | López    |
| 7  | Jones    | 115        | crescut            | López    |
| 8  | Jones    | 117        | crescut            | López    |
| 9  | (null)   | (null)     | (null)             | Popescu  |
| 10 | (null)   | (null)     | (null)             | Ionescu  |
| 11 | (null)   | (null)     | (null)             | Smith    |
| 12 | (null)   | (null)     | (null)             | Müller   |
| 13 | (null)   | (null)     | (null)             | García   |
| 14 | (null)   | (null)     | (null)             | Rossi    |
| 15 | (null)   | (null)     | (null)             | Li       |
| 16 | (null)   | (null)     | (null)             | Kim      |
| 17 | (null)   | (null)     | (null)             | Popovici |
| 18 | (null)   | (null)     | (null)             | Lopez    |
| 19 | (null)   | (null)     | (null)             | Schmidt  |
| 20 | (null)   | (null)     | (null)             | Russo    |
| 21 | (null)   | (null)     | (null)             | Chen     |
| 22 | (null)   | (null)     | (null)             | Lee      |
| 23 | (null)   | (null)     | (null)             | Garcia   |

--b.) o cerere care implementează analiza top-n

--enunt: Sa se selecteze primii 10 angajati cu cele mai mici salarii.

SELECT nume, prenume, salariu

FROM (

SELECT nume, prenume, salariu

FROM ANGAJAT

ORDER BY salariu ASC

)

WHERE ROWNUM <= 10;

| Script Output x Query Result x              |         |         |
|---|---------|---------|
| SQL   All Rows Fetched: 10 in 0.001 seconds |         |         |
| NUME  | PRENUME | SALARIU |
| 1 Ionescu                                   | Maria   | 4500    |
| 2 Lee                                       | Min-ji  | 4600    |
| 3 Kim                                       | Ji-hyun | 4700    |
| 4 Russo                                     | Anna    | 4700    |
| 5 García                                    | Ana     | 4800    |
| 6 López                                     | Carmen  | 4800    |
| 7 Lopez                                     | Maria   | 4800    |
| 8 Li  | Wei     | 4900    |
| 9 Sato                                      | Hiroshi | 4900    |
| 10 Popescu                                  | Ion     | 5000    |

--c.) o cerere care utilizeaza operatia division

--enunț: Să se obțină codurile angajaților atașați la toate activitățile care încep la ora 9 dimineța.

```
select * from activitate;
```

```
select * from organizeaza;
```

```
--UPDATE ACTIVITATE
```

```
--SET ora='09:00'
```

```
--WHERE id_activitate= 65;
```

```
--rollback;
```

--am actualizat datele pentru a vedea și alt exemplu

```
SELECT id_angajat
```

```
FROM ORGANIZEAZA
```

```
WHERE id_activitate in (SELECT id_activitate
```

```
FROM ACTIVITATE
```

```
WHERE ora='09:00')
```

```
GROUP BY id_angajat
```

```
HAVING count(id_activitate)= (SELECT count(*)
```

```
FROM ACTIVITATE
```

```
WHERE ora='09:00');
```

```

Worksheet  Query Builder
208 --UPDATE ACTIVITATE
209 --SET ora='09:00'
210 --WHERE id_activitate= 65;
211 --rollback;
212 --am actualizat datele pentru a vedea si alt exempl
213 SELECT id_angajat
214 FROM ORGANIZEAZA
215 WHERE id_activitate in (SELECT id_activitate
216                          FROM ACTIVITATE
217                          WHERE ora='09:00')
218 GROUP BY id_angajat
219 HAVING count(id_activitate)= (SELECT count(*)
220                               FROM ACTIVITATE
221                               WHERE ora='09:00') ;
222
223 --16)

```

Script Output x Query Result x

All Rows Fetched: 2 in 0.001 seconds

| ID_ANGAJAT |
|------------|
| 210        |
| 214        |

--15)

Cerere in limbaj natural:

--enunt: Sa se afle numele, id-ul angajatilor, salariul și data în care au fost angajați oamenii cu salariul între 5000 si 6000 și cu productivitatea crescută.

--cerere sql:

**SELECT** a.id\_angajat, salariu, nume, data\_angajarii

**FROM** istoric i **JOIN** angajat a **ON** (a.id\_angajat=i.id\_angajat)

**WHERE** salariu >5000 **AND** salariu <6000

**AND** lower(productivitate)='crescuta';



|   | NUME   | SALARIU | ID_ANGAJAT | DATA_ANGAJARII |
|---|--------|---------|------------|----------------|
| 1 | Müller | 5500    | 203        | 10-SEP-20      |
| 2 | Rossi  | 5200    | 205        | 03-APR-20      |
| 3 | Russo  | 5500    | 211        | 09-JUN-04      |
| 4 | Garcia | 5100    | 214        | 10-SEP-20      |

### Expresia algebrica - neoptima:

R1= PROJECT(ANGAJAT, id\_angajat, nume, salariu)

R2= SELECT (R1, salariu >5000)

R3= SELECT (R2, salariu <6000)

R4= PROJECT (ISTORIC, id\_angajat, data\_angajarii, productivitate)

R5= SELECT(R4, productivitate='crescuta')

R6= PROJECT( R5, id\_angajat, data\_angajarii)

Rezultat= JOIN(R6, R3, id\_angajat)

### Expresia algebrica - optima:

R1=SELECT(ANGAJAT, salariu>5000 AND salariu<6000)-Se aplica Proprietatea 4 – Compunere selectiilor

R2=PROJECT(R1, id\_angajat, salariu, nume)- Proprietatea 5 – Comutarea selectiei cu proiectia

R3=SELECT (ISTORIC, productivitate='crescuta')

R4=PROJECT(R3, id\_angajat, data\_angajarii)

Rezultat= JOIN(R4, R2, id\_angajat) -> rezultata optimizat

Asupra relatiilor R4 si R5 din expresia algebrica anterioara se aplica Proprietatea 5 – Comutarea selectiei cu proiectia, iar cele doua proiectii R4 si R6, prin Proprietatea 3 – Compunerea proiectiilor devin o singura proiectie.

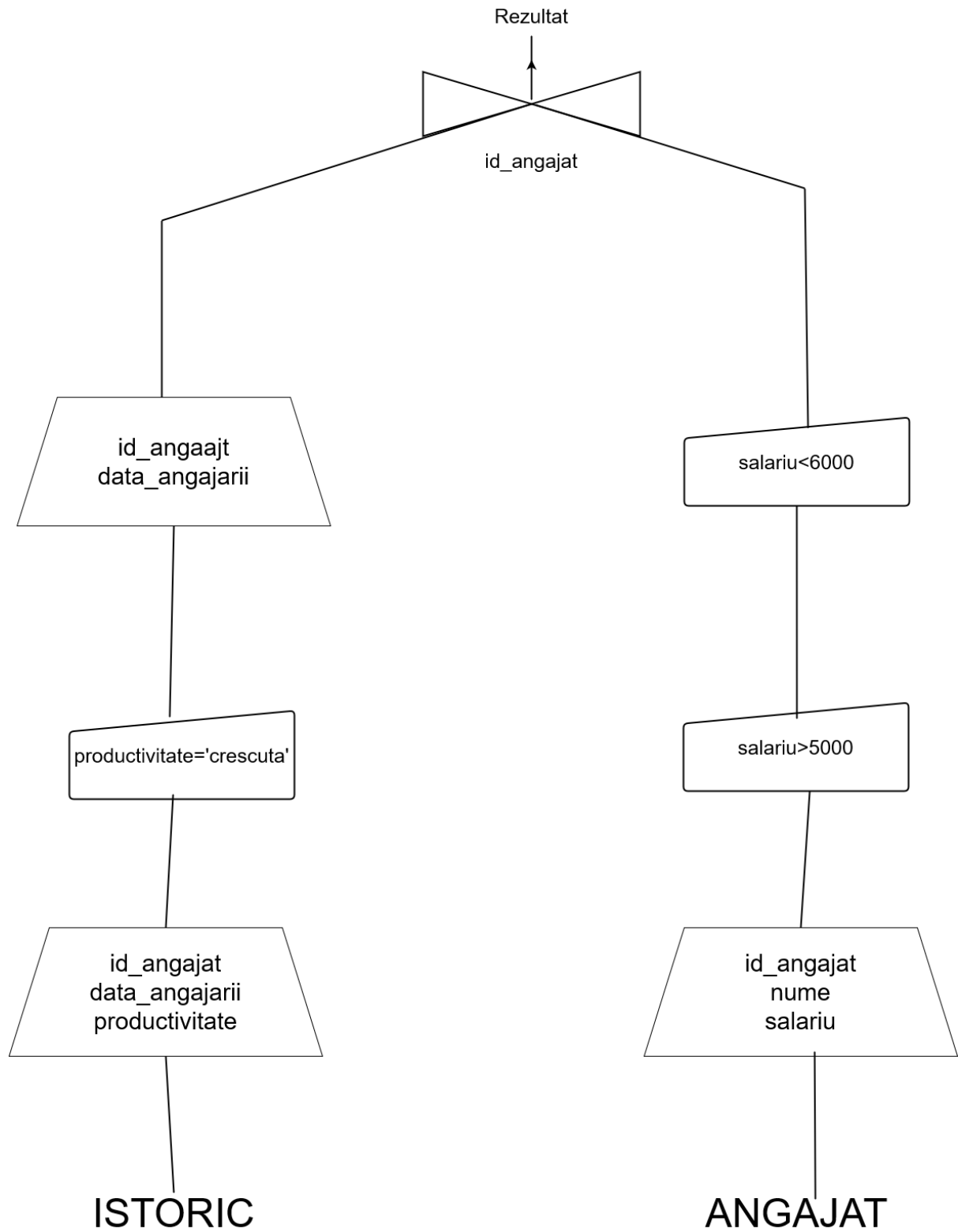
### Explicatii optimizare:

#### Reguli folosite:

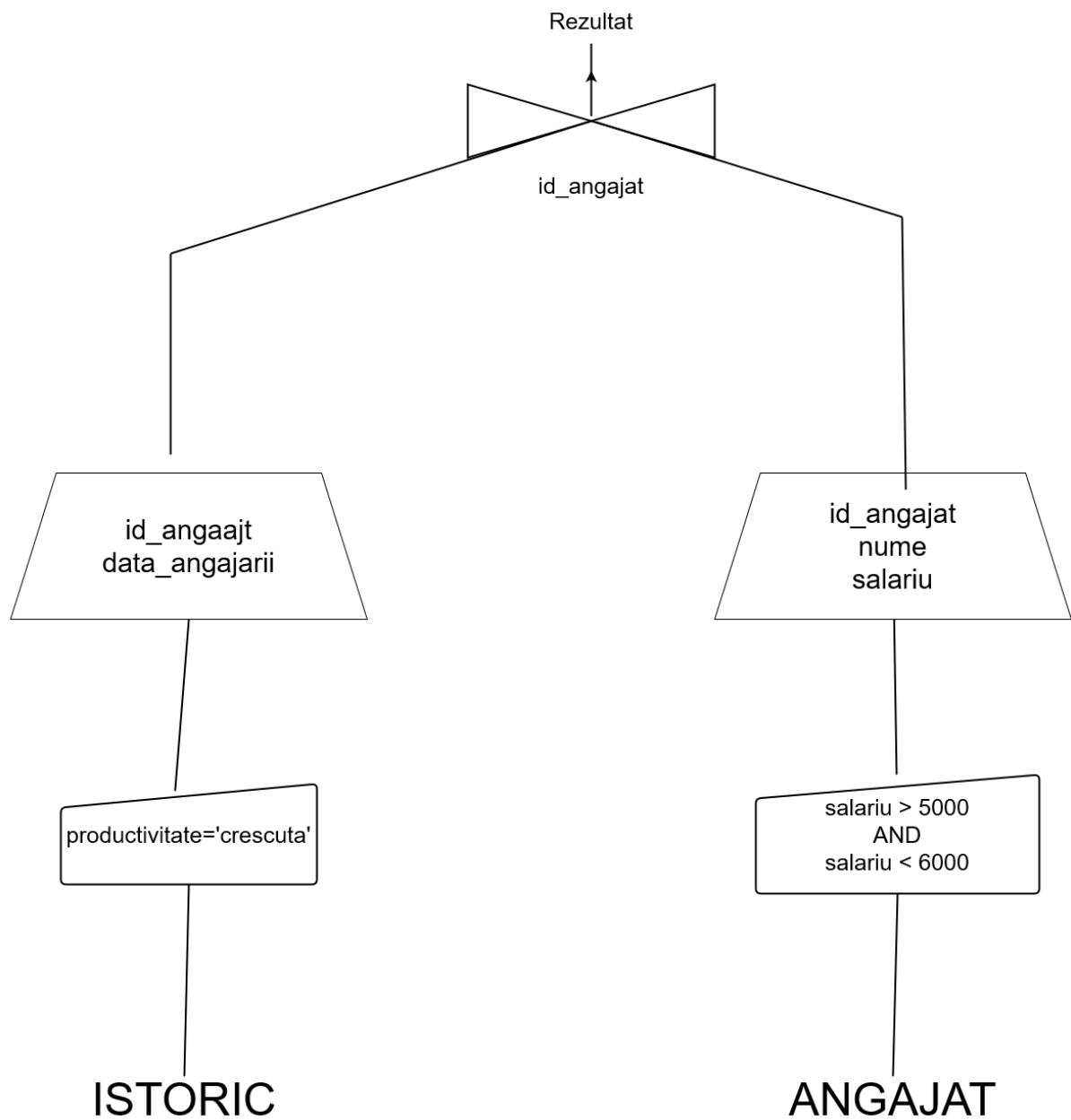
- Selectiile se executa cat mai devreme posibil – reducand substantial dimensiunea relatiilor;
- Produsele carteziene se inlocuiesc cu join-uri – s-a utilizat operatia join dupa coloana comuna cod\_contractant;

-Proiectiile s-au folosit la inceput pentru a se indeparta attributele nefolositoare.

**Arbore algebric neoptim:**



**Arbore algebric optim:**



16) a). Realizarea normalizarii BCNF, FN4, FN5.

### **BCNF**

O relație R este în forma normală Boyce-Codd dacă și numai dacă fiecare determinant este o cheie candidat.

Se poate identifica tabelul COMANDA care nu este în BCNF, deoarece nr\_telefon determină funcțional id\_user . Așadar, nr\_telefon este determinant, dar nu este cheie candidat.

Pentru a aduce în BCNF se poate aplica Regula Casey Delobel:  
COMANDA(id\_comanda#, id\_user#, nr\_telefon)—există dependență id\_user->nr\_telefon

COMANDA\_1(id\_comanda#,nr\_telefon)

COMANDA\_2(nr\_telefon#, id\_user)

### **FN4**

Formal, relația R este în a patra formă normală dacă și numai dacă:

- R este în BCNF;
- orice dependență multivaloare este o dependență funcțională.

Se poate vedea că tabelul INGRIJESTE nu se află în FN4 (chiar dacă este în BCNF), deoarece nu orice dependență multivaloare este o dependență funcțională.

Relația INGRIJESTE în FN4 ar arăta astfel:

INGRIJESTE\_1(id\_angajat#, id\_animal#, id\_cusca#)

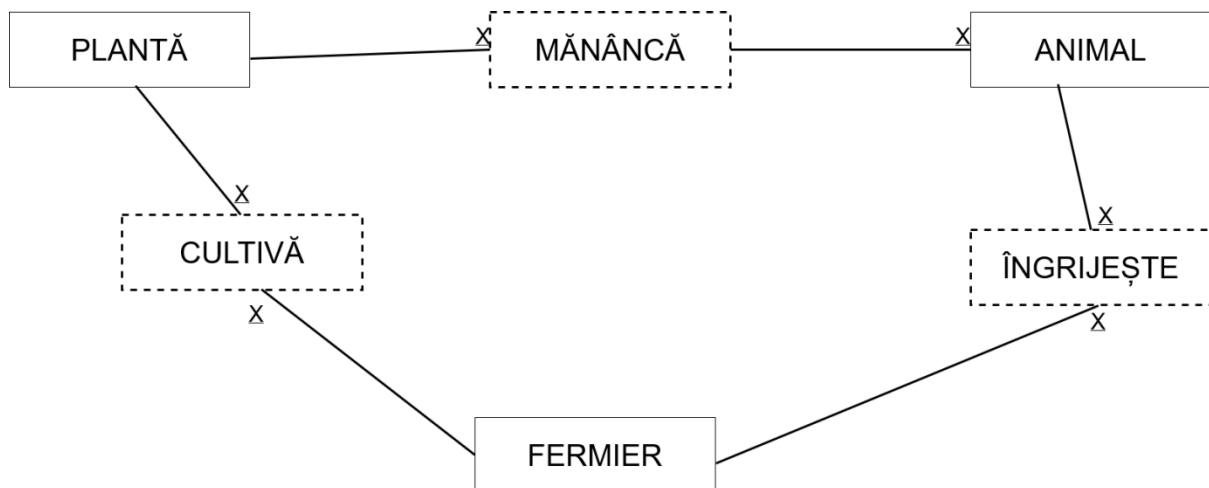
INGRIJESTE\_2(id\_animal#, id\_cusca#, id\_planta#, id\_teren#, id\_sol#)

### **FN5**

Intuitiv, o relație R este în forma normală 5 dacă și numai dacă:

- relația este în FN4;
- nu conține dependențe ciclice.

Modelul meu are relațiile în FN5. De exemplu, dacă relația ÎNGRIJEȘTE nu s-ar afla în FN5, ea ar conține dependențe circulare și ar arăta astfel:



b. Aplicarea denormalizării, justificând necesitatea acesteia.

Obiectivul denormalizării constă în reducerea numărului de join-uri efectuate pentru rezolvarea unei interogări, prin realizarea unora dintre acestea în avans, ca făcând parte din proiectarea bazei de date.

Normalizarea de la FN4 nu era necesară, deoarece se pierde informații. De aceea vrem să o denormalizăm. De exemplu, dacă un angajat hrănește un animal și același angajat îngrijește o plantă, nu este necesar ca fermierul să hrănească animalul cu acea plantă.