

# Baza de date pentru un sistem de transport public în comun

***Coheci Cristiana***

***Grupa 152***

***Iunie 2023***

## Cuprins

Descrierea modelului real, a utilitatii acestuia si a regulilor de functionare. ....	3
<b>Prezentarea constrangerilor (restrictii, reguli) impuse asupra modelului.....</b>	<b>3</b>
<b>Descrierea entitatilor, incluzand precizarea cheii primare. ....</b>	<b>4</b>
<b>Descrierea relatiilor, incluzand precizarea cardinalitatii acestora. ....</b>	<b>5</b>
Descrierea atributelor, incluzand tipul de date si eventualele constrangeri, valori implicite, valori posibile ale atributelor. ....	6
Realizarea diagramei entitate-relatie corespunzatoare descrierii de la punctele 3-5. ....	11
Realizarea diagramei conceptuale corespunzatoare diagramei entitate-relatie proiectata la punctul 6. Diagrama conceptuala obtinuta trebuie sa contina minimum 6 tabele (fara considerarea subentitatilor), dintre care cel putin un tabel asociativ.....	12
Enumerarea schemelor relationale corespunzatoare diagramei conceptuale proiectata la punctul 7. ....	13
Realizarea normalizarii pana la forma normala 3 (FN1-FN3). ....	14
FN1 .....	14
FN2 .....	15
FN3 .....	16
Crearea unei secvente ce va fi utilizata in inserarea inregistrarilor in tabele (punctul 11).....	17
Crearea tabelelor in SQL si inserarea de date coerente in fiecare dintre acestea (minimum 5 inregistrari in fiecare tabel neasociativ; minimum 10 inregistrari in tabelele asociative) .....	18
Formulati in limbaj natural si implementati 5 cereri SQL complexe ce vor utiliza, in ansamblul lor, urmatoarele elemente: .....	28
Implementarea a 3 operatii de actualizare si de suprimare a datelor utilizand subcereri. ....	34
Formulati in limbaj natural si implementati in SQL: o cerere ce utilizeaza operatia outer-join pe minimum 4 tabele, o cerere ce utilizează operatia division si o cerere care implementează analiza top-n. Observație: Cele 3 cereri sunt diferite de cererile de la exercițiul 12. ....	37
Optimizarea unei cereri, aplicand regulile de optimizare ce deriva din proprietatile operatorilor algebrei relationale. Cererea va fi exprimata prin expresie algebrica, arbore algebric si limbaj (SQL), atat anterior cat si ulterior optimizarii. ..	39
Realizarea normalizarii BCNF, FN4, FN5. Aplicarea denormalizarii, justificand necesitatea acesteia. ....	45
BCNF .....	45
FN4 .....	45
FN5 .....	46
DENORMALIZARE .....	46

## Descrierea modelului real, a utilitatii acestuia si a regulilor de functionare.

Modelul de date va gestiona informatii legate de organizarea sistemului de transport in comun dintr-un oras. Asociatia are mai multe departamente, in functie de tipul de transport( terestru, subteran, acvatic), fiecare departament are mai multe sectii (ex: autobuz, troleibuz...) si mai multi angajati. Fiecare sectie este condusă de un manager, fiecare departament e condus de un manager. Fiecare angajat are un job si este repartizat intr-un sediu dintr-o anumită locație. O sectie se ocupa de mai multe trasee. Un angajat poate fi repartizat pe mai multe trasee. Pe un traseu circula mai multe autovehicule. Un autovehicul poate fi condus de mai multi soferi. Un autovehicul este repartizat într-un depou. Un traseu are mai multe stații. Un vehicul poate fi de mai multe tipuri, iar un tip de vehicul poate avea mai multe caracteristici de-a lungul timpului. Modelul este util pentru tinerea in evidenta a datelor companiei si raspunsul rapid la intrebari complexe.

## Prezentarea constrangerilor (restrictii, reguli) impuse asupra modelului.

- Un manager de departament este si manager de sectie.
- Fiecare departament are un singur manager.
- Angajatii sunt de doua tipuri, angajati obisnuiti, care au un job obisnuit si angajati soferi.
- Un sofer poate conduce mai multe vehicule ( trebuie sa conduca cel putin unul ), pe mai multe trasee, in intervale de timp diferite.
- Un traseu are mai multe statii, iar o statie apartine de cel putin un traseu (exista statii comune mai multor trasee).
- Statiile, depourile si filialele se afla in locatii. O locatie trebuie sa apartina de cel putin una dintre cele trei mai devreme enumerate.
- Un angajat lucreaza intr-o singura filiala. Unele filiale sunt fizice, au locatii altele nu au.
- Un vehicul are un set de caracteristici si poate avea mai multe caracteristici de-a lungul timpului. Un set de caracteristici poate aparea la mai multe vehicule.

## Descrierea entitatilor, incluzand precizarea cheii primare.

	Entitate	Primary Key
1	Departament transport	Tip_transport#
2	Sectie	Cod_sectie#
3	Angajat	Cod_angajat
4	SOFER (subentitate)	Cod_angajat
5	OBISNUIT (Subentitate)	Cod_angajat
6	Job	Cod_job
7	Locatie	Cod_locatie
8	Filiale	Cod_sectie, cod_locatie
9	Traseu	Cod_traseu
10	Statie	Cod_statie
11	Statie_Unica	Cod_statie, cod_traseu
12	Vehicul	Nr_inmatriculare
13	Tipuri_vehicule	Cod_tip_vehicul
14	Caracteristici	Cod_caracteristici
15	Caracteristica_tip	Cod_caracteristica, cod_tip_vehicul, data
16	Vehicul_Traseu_Sofer	Nr_inmatriculare, cod_traseu, cod_angajat
17	Depou	Cod_depou

1. DEPARTAMENT\_TRANSPORT = departamentul principal al companiei. Poate fi terestru, nautic, subteran si feroviar. Cheia primara este tip\_transport.
2. SECTIE (sectia nautic\_BARCI, sectia nautic hidro biciclete) = sectie dintr-un departament care se ocupa cu un anumit tip de vehicule. Cheia primara este cod\_sectie.
3. ANGAJATI= persoana fizica, angajata in cadrul Sistemului de Transport, care este responsabilă de anumite activități ale companiei. Cheia primara este cod\_angajat.
4. SOFER= subentitate a entitatii ANGAJAT, ce contine informatii specifice soferilor. Cheia primara este cod\_angajat.
5. OBISNUIT = subentitate a entitatii ANGAJAT, contine informatii specifice angajatilor obisnuiti. Cheia primara este cod\_angajat.
6. JOBURI = job pe care il poate avea un angajat in companie. Cheia primara este cod\_job.
7. LOCATII = locatie fizica in care se desfasoara anumite activitati ale firmei. Cheia primara este cod\_locatie.
8. FILIALE = subsectie a unei sectii, poate avea locatie fizica sau nu. Cheia primara este cod\_sectie+cod\_locatie.
9. TRASEE = ruta din oras pe care circula vehicule. Cheia primara este cod\_traseu.
10. STATII = statia de oprire pe parcursul traseelor. Cheia primara este cod\_statie.

11. STATIE\_UNICA = statie de oprire care apartine unui singur traseu. Cheia primara este cod\_statie+cod\_locatie.
12. VEHICULE = exemplar dintr-o masinarie ce are ca scop transportul de persoane. Cheia primară este nr\_înmatriculare.
13. TIP\_VEHICUL = un model de vehicul. Cheia primara este cod\_tip\_vehicul
14. CARACTERISTICI = un set de caracteristici ale vehiculelor. Cheia primara este cod\_caracteristica
15. CARACTERISTICA\_TIP = corespondenta dintre un tip de vehicul si ce caracteristici i se atribuie la ce moment de timp. Cheia primara este compusa din cod\_caracteristica, cod\_tip\_vehicul si data\_start.
16. VEHICUL\_TRASEU\_SOFER = corespondenta dintre un sofer, autovehiculul pe care il conduce si pe ce traseu il conduce (tabel asociativ). Cheia primara este compusa din nr\_inmatriculare, cod\_angajat si cod\_traseu.
17. DEPOURI = zona in care se depoziteaza vehiculele atunci cand nu sunt folosite. Cheia primara este cod\_depou.

## Descrierea relatiilor, incluzand precizarea cardinalitatii acestora.

**DEPARTAMENT\_TRANSPORT\_contine\_SECTII**= relatie dintre entitatile DEPARTAMENT\_TRANSPORT si SECTII reprezentand sectiile in subordinea fiecarui departament de transport. Relatia are cardinalitate minima 1:1 si cardinalitate maxima 1:n.

**ANGAJAT\_lucreaza\_in\_SECTIE**= relatie dintre entitatile ANGAJAT si SECTIE reprezentand sectiile in care lucreaza angajatii. Relatia are cardinalitate minima 1:0 (sectie trebuie sa aiba angajati, dar un angajat poate sa nu aiba sectie) si cardinalitate maxima n:1 (un angajat are o singura sectie, o sectie mai multi angajati).

**ANGAJAT\_conduce\_ANGAJATI**= relatie dintre entitatile ANGAJAT si ANGAJAT reprezentand relatia de subordonare intre angajati. Relatia are cardinalitate minima 0:0 si cardinalitate maxima 1:n (un manager poate sa aiba mai multi subordonati, un subordonat are un singur manager direct).

**ANGAJAT\_OBISNUIT\_are\_UN\_JOB**= relatie dintre subentitatea ANGAJAT\_OBISNUIT si entitatea JOBURI reprezentand jobul pe care il are fiecare angajat obisnuit (care nu e sofer). Relatia are cardinalitate minima 0:1 (un angajat trebuie sa aiba un job, un job poate sa nu aiba angajati) si cardinalitate maxima n:1 (in angajat poate sa aiba un singur job).

**ANGAJAT\_SOFER\_conduce\_pe\_TRASEU\_mai\_multe\_VEHICULE**= relatie de tip 3 ce leaga subentitatea ANGAJAT\_SOFER de entitatile TRASEE SI VEHICULE, reflectand soferul care conduce un anumit vehicul pe un anumit traseu. Denumirea acestei relatii este *conduce*.

**SECTIE\_se\_ocupa\_de\_TRASEE**= relatie dintre entitatile SECTIE si TRASEE reprezentand carei sectii apartine fiecare traseu. Relatia are cardinalitate minima 1:1 (nu exista sectie fara traseu sau traseu fara sectie) si cardinalitate maxima 1:n (un traseu are o singura sectie, o sectie poate sa aiba toate traseele).

**SECTIE\_se\_afla\_in\_LOCATIE**= relatie dintre entitatile SECTIE si LOCATIE reprezentand locatia la care se afla un sediu al companiei. La o locatie se pot afla mai multe sectii. O sectie poate fi impartita pe mai multe filiale, la mai multe locatii. Relatia are cardinalitate minima 0:1 si cardinalitate maxima n:n.

**DEPOU\_se\_afla\_in\_LOCATIE**= relatie dintre entitatile DEPOU si LOCATIE reprezentand locatia la care se afla un depou al companiei. Relatia are cardinalitate minima 0:1 si cardinalitate maxima n:1.

**VEHICUL\_este\_parcata\_in\_DEPOU**= relatie dintre entitatile VEHICUL si DEPOU reprezentand depoul in care este parcat regulat un vehicul. Relatia are cardinalitate minima 1:1 si cardinalitate maxima n:1.

**VEHICUL\_este\_de\_TIP\_VEHICUL**= relatie dintre entitatile VEHICUL si TIP\_VEHICUL reprezentand tipul de care apartine un vehicul. Relatia are cardinalitate minima 1:1 si cardinalitate maxima n:1.

**TIP\_VEHICUL\_are\_CHARACTERISTICI**= relatie dintre entitatile TIPURI\_VEHICULE si CHARACTERISTICI reprezentand caracteristicile pe care le are un tip de vehicul. Relatia are cardinalitate minima 1:1 si cardinalitate maxima 1:n.

**TRASEU\_are\_STATII** = relatie dintre entitatile TRASEE si STATII reprezentand statiile pe care le are un traseu. Relatia are cardinalitate minima 1:1 si cardinalitate maxima n:n (un traseu poate avea mai multe statii si o statie poate apartine de mai multe trasee).

**STATIE\_se\_afla\_in\_LOCATIE** = relatie dintre entitatile STATII si LOCATII reprezentand locatia la care se afla o statie. Relatia are cardinalitate minima 0:1 si cardinalitate maxima 1:1.

## Descrierea atributelor, incluzand tipul de date si eventualele constrangeri, valori implicite, valori posibile ale atributelor.

Subentitatea "**obisnuit\_stc**" are ca atribute:

- **Cod\_angajat**= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul ofertei. Acest atribut trebuie sa corespunda valorii cheii primare din superentitatea angajati\_stc.
- **Cod\_job**= variabila de tip intreg, de lungime maxima 5, care reprezinta codul tipului de vehicul. Atributul trebuie sa corespunda valorii cheii primare din tabelul JOBURI.

Subentitatea "**sofer\_stc**" are ca atribute:

- **cod\_angajat**: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al angajatului. Acest atribut trebuie sa corespunda valorii cheii primare din superentitatea angajati\_stc.
- **tip\_permis**: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând tipul de permis de conducere al șoferului. Acest atribut nu poate fi vid.

Entitatea independentă "**angajati\_stc**" are ca atribute:

- **cod\_angajat**: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al angajatului.

- o nume: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând numele angajatului.
- o data\_angajare: Variabilă de tip dată, reprezentând data angajării angajatului.
- o salariu: Variabilă numerică cu o lungime maximă de 10 cifre, reprezentând salariul angajatului.
- o Cod\_sef: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al sefului. Acest atribut trebuie sa corespunda cheii primare din entitatea angajati\_stc.
- o Cod\_sectie: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând parte din codul filialei in care lucreaza. Acest atribut trebuie sa corespunda cheii primare din entitatea filiale\_stc.
- o Cod\_locatie: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând parte din codul locatiei filialei in care lucreaza. Acest atribut trebuie sa corespunda cheii primare din entitatea filiale\_stc.

Entitatea independentă "**joburi**" are ca attribute:

- o cod\_job: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al jobului.
- o nume: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând numele jobului.
- o ore\_pe\_zi: Variabilă numerică cu o lungime maximă de 2 cifre, reprezentând numărul de ore lucrate pe zi pentru acest job.

Entitatea independentă "**departament\_transport\_stc**" are ca attribute:

- o tip\_transport: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând tipul de transport.
- o responsabil: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al angajatului responsabil pentru acest departament. Acest atribut este o cheie străină care face referire la coloana "cod\_angajat" din tabela "angajati\_stc".
- o nume\_departament: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând numele departamentului. Acest atribut nu poate fi vid.

Entitatea independentă "**sectii\_stc**" are ca attribute:

- o cod\_sectie: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al secției.
- o responsabil: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al angajatului responsabil pentru această secție. Acest atribut este o cheie străină care face referire la coloana "cod\_angajat" din tabela "angajati\_stc".
- o tip\_transport: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând tipul de transport al secției. Acest atribut este o cheie străină care face referire la coloana "tip\_transport" din tabela "departament\_transport\_stc".
- o nume\_sectie: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând numele secției.

Entitatea independentă "**locatii\_stc**" are ca attribute:

- o cod\_locatie: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al locației.

- o nume: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând numele locației. Acest atribut trebuie să fie unic.
- o adresa: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând adresa locației. Acest atribut trebuie să fie unic.
- o cod\_postal: Variabilă numerică cu o lungime maximă de 7 cifre, reprezentând codul poștal al locației.

Tabela asociativă "**filiale\_stc**" are ca attribute:

- o cod\_sectie: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al secției. Această coloană este o cheie străină care face referire la coloana "cod\_sectie" din tabela "sectii\_stc".
- o cod\_locatie: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al locației. Această coloană este o cheie străină care face referire la coloana "cod\_locatie" din tabela "locatii\_stc".
- o responsabil: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al angajatului responsabil pentru această filială. Acest atribut este o cheie străină care face referire la coloana "cod\_angajat" din tabela "angajati\_stc".

Entitatea independentă "**trasee\_stc**" are ca attribute:

- o cod\_traseu: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al traseului.
- o capat1: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând primul capăt al traseului. Acest atribut este o cheie străină care face referire la coloana "cod\_statie" din tabela "statii\_stc".
- o capat2: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând al doilea capăt al traseului. Acest atribut este o cheie străină care face referire la coloana "cod\_statie" din tabela "statii\_stc".
- o lungime: Variabilă numerică cu o lungime maximă de 7 cifre, reprezentând lungimea traseului.
- o cod\_sectie: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al secției. Această coloană este o cheie străină care face referire la coloana "cod\_sectie" din tabela "sectii\_stc".

Entitatea independentă "**depouri\_stc**" are ca attribute:

- o cod\_depou: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al depoului.
- o nume\_depou: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând numele depoului.
- o capacitate: Variabilă numerică cu o lungime maximă de 5 cifre, reprezentând capacitatea depoului.
- o cod\_locatie: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al locației. Această coloană este o cheie străină care face referire la coloana "cod\_locatie" din tabela "locatii\_stc".

Entitatea independentă "**statii\_stc**" are ca attribute:



- **cod\_statie:** Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al stației.
- **nume\_statie:** Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând numele stației.
- **cod\_locatie:** Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al locației. Această coloană este o cheie străină care face referire la coloana "cod\_locatie" din tabela "locatii\_stc".

Tabela asociativa "**statie\_unica\_stc**" are ca attribute:

- **cod\_statie:** Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al stației. Această coloană este o cheie străină care face referire la coloana "cod\_statie" din tabela "statii\_stc".
- **cod\_traseu:** Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al traseului. Această coloană este o cheie străină care face referire la coloana "cod\_traseu" din tabela "trasee\_stc".

Entitatea independentă "**tipuri\_vehicule\_stc**" are ca attribute:

- **cod\_tip\_vehicul:** Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al tipului de vehicul.
- **nume:** Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând numele tipului de vehicul.
- **marca:** Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând marca tipului de vehicul.

Entitatea independentă "**caracteristici\_stc**":

- **cod\_caract:** Variabilă numerică cu o lungime maximă de 3 cifre, reprezentând codul unic al caracteristicii.
- **nr\_locuri:** Variabilă numerică cu o lungime maximă de 3 cifre, reprezentând numărul de locuri al vehiculului.
- **nr\_roti:** Variabilă numerică cu o lungime maximă de 2 cifre, reprezentând numărul de roți al vehiculului.
- **seria:** Variabilă de tip text cu lungimea maximă de 10 caractere, reprezentând seria vehiculului.
- **model:** Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând modelul vehiculului.
- **an:** Variabilă de tip dată calendaristică, reprezentând anul producției vehiculului. Acest atribut are o valoare implicită data inserării.

Entitatea independentă "**caracteristica\_tip**":

- **cod\_caract:** Variabilă numerică cu o lungime maximă de 3 cifre, reprezentând codul unic al caracteristicii. Această coloană este o cheie străină care face referire la coloana "cod\_caract" din tabela "caracteristici\_stc".
- **cod\_tip\_vehicul:** Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al tipului de vehicul. Această coloană este o cheie străină care face referire la coloana "cod\_tip\_vehicul" din tabela "tipuri\_vehicule\_stc".
- **data\_start:** Variabilă de tip dată calendaristică, reprezentând data de început a valabilității caracteristicii pentru tipul de vehicul.

- data\_expirare: Variabilă de tip dată calendaristică, reprezentând data de expirare a valabilității caracteristicii pentru tipul de vehicul.

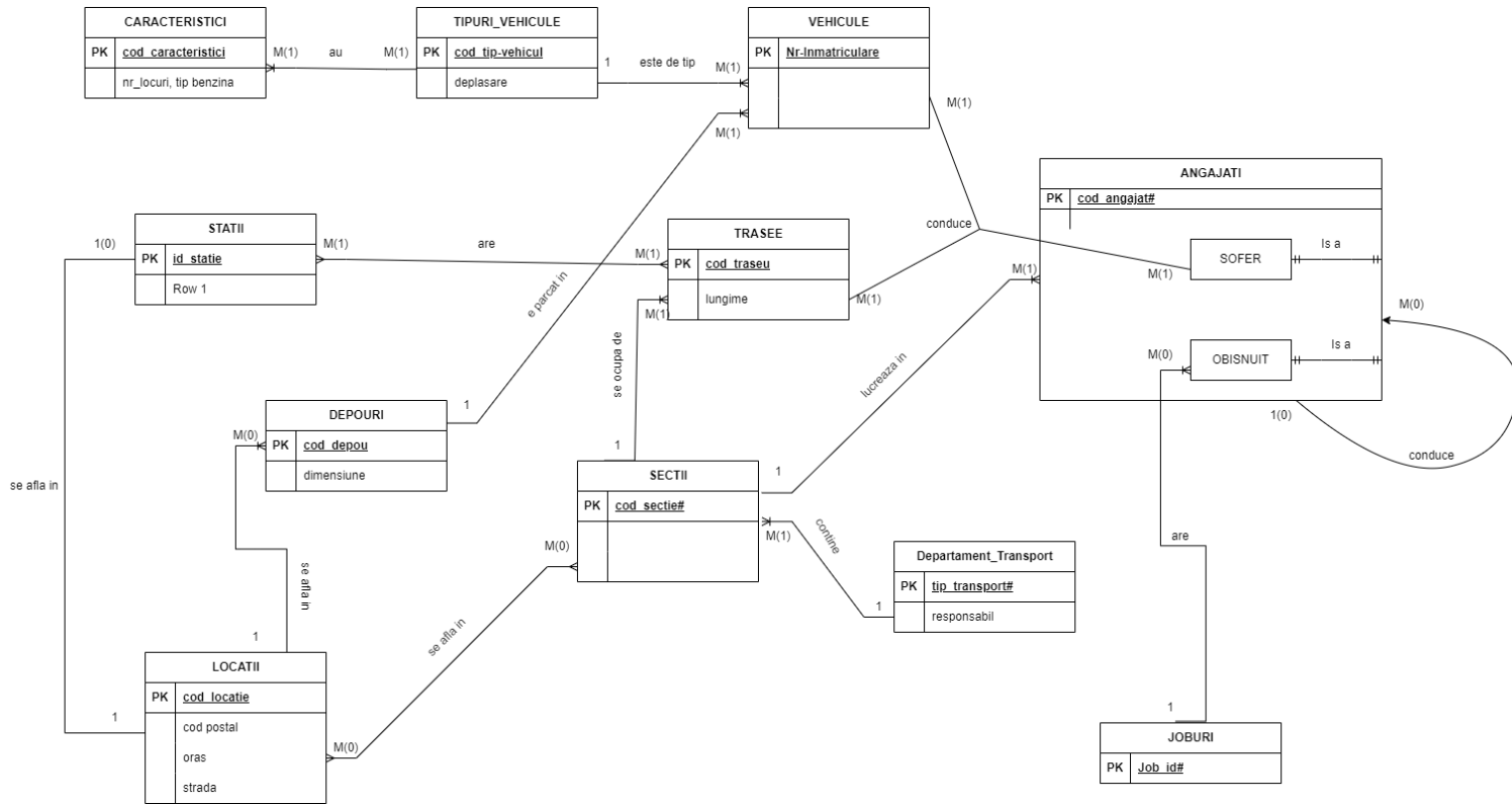
Entitatea independentă "**vehicule\_stc**":

- nr\_inmatriculare: Variabilă de tip text cu lungimea maximă de 10 caractere, reprezentând numărul de înmatriculare al vehiculului. Această coloană reprezintă cheia primară a tabelului.
- cod\_tip\_vehicul: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al tipului de vehicul. Această coloană este o cheie străină care face referire la coloana "cod\_tip\_vehicul" din tabela "tipuri\_vehicule\_stc".
- cod\_depou: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al depoului. Această coloană este o cheie străină care face referire la coloana "cod\_depou" din tabela "depouri\_stc".

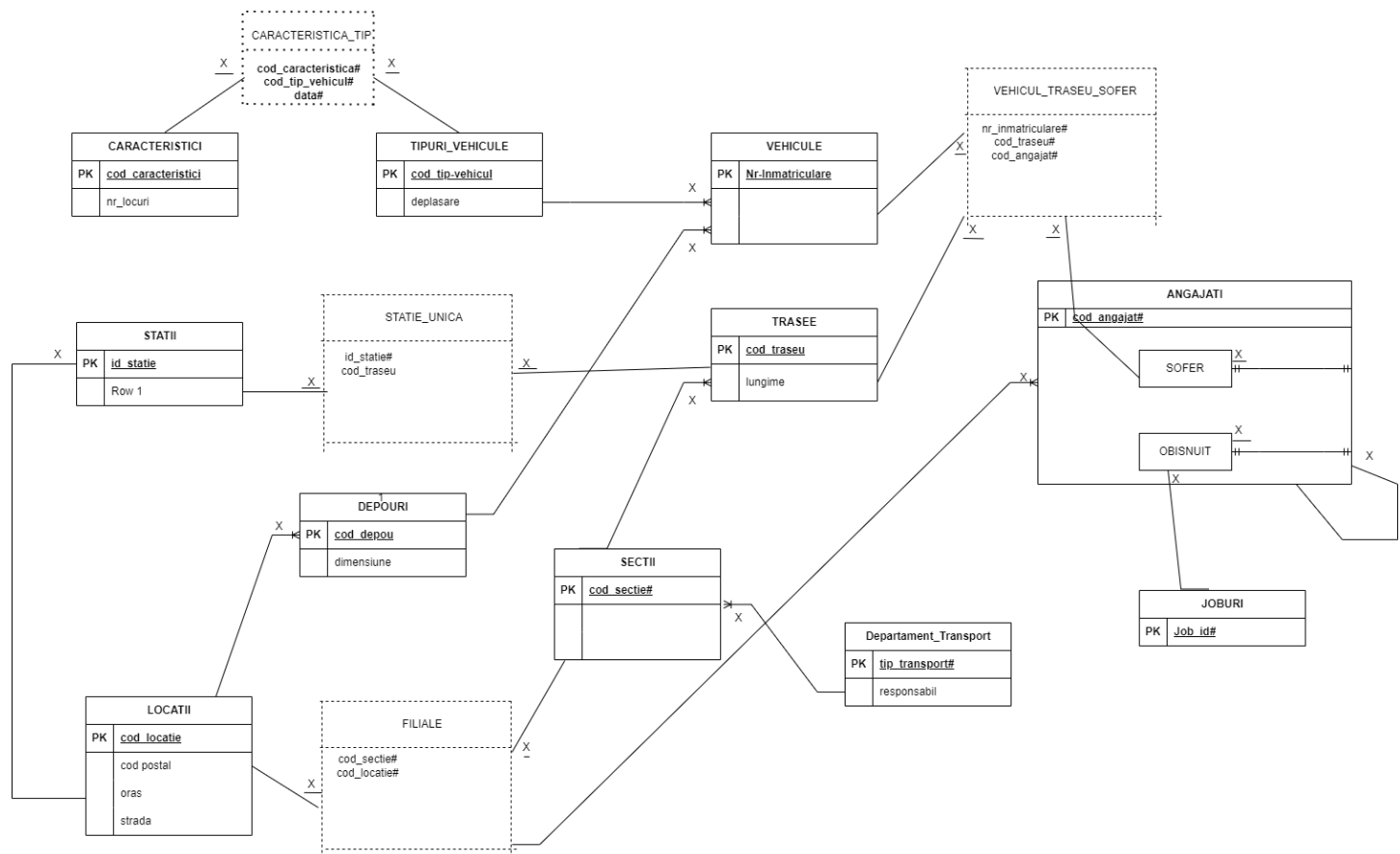
Tabela asociativă "**vehicul\_traseu\_sofer**":

- cod\_angajat: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al angajatului. Această coloană este o cheie străină care face referire la coloana "cod\_angajat" din tabela "sofer\_stc".
- nr\_inmatriculare: Variabilă de tip text cu lungimea maximă de 10 caractere, reprezentând numărul de înmatriculare al vehiculului. Această coloană este o cheie străină care face referire la coloana "nr\_inmatriculare" din tabela "vehicule\_stc".
- cod\_traseu: Variabilă de tip text cu lungimea maximă de 25 de caractere, reprezentând codul unic al traseului. Această coloană este o cheie străină care face referire la coloana "cod\_traseu" din tabela "trasee\_stc".

Realizarea diagramei entitate-relatie corespunzatoare descrierii de la punctele 3-5.



Realizarea diagramei conceptuale corespunzatoare diagramei entitate-relatie proiectata la punctul 6. Diagrama conceptuala obtinuta trebuie sa contina minimum 6 tabele (fara considerarea subentitatilor), dintre care cel putin un tabel asociativ.



Enumerarea schemelor relationale corespunzatoare diagramei conceptuale proiectata la punctul 7.

ANGAJATI\_STC (cod\_angajat#, nume, data\_angajare, salariu, cod\_sef, cod\_sectie, cod\_locatie);  
JOBURI (cod\_job#, nume, ore\_pe\_zi);  
OBISNUIT\_STC (cod\_angajat#, cod\_job);  
SOFER\_STC (cod\_angajat#, tip\_permis);  
DEPARTAMENT\_TRANSPORT\_STC ( tip\_transport#, responsabil, nume\_departament);  
SECTII\_STC (cod\_sectie#, responsabil, tip\_transport, nume\_sectie);  
LOCATII\_STC (cod\_locatie#, nume, adresa, cod\_postal);  
FILIALE\_STC (cod\_sectie#, cod\_locatie#, responsabil);  
TRASEE\_STC (cod\_traseu#, capat1, capat2, lungime, cod\_sectie);  
DEPOURI\_STC (cod\_depou#, nume\_depou, capacitate, cod\_locatie);  
STATII\_STC (cod\_statie#, nume\_statie, cod\_locatie);  
STATIE\_UNICA\_STC (cod\_statie#, cod\_traseu#);  
TIPURI\_VEHICULE\_STC (cod\_tip\_vehicul#, nume, marca);  
CARACTERISTICI\_STC(cod\_caract#, nr\_locuri, nr\_roti, combustibil, seria, model, an);  
CARACTERISTICA\_TIP (cod\_caract#, cod\_tip\_vehicul#, data\_start#, data\_expirare);  
VEHICULE\_STC (nr\_inmatriculare#, cod\_tip\_vehicul, cod\_depou);  
VEHICUL\_TRASEU\_SOFER (cod\_angajat#, nr\_inmatriculare#, cod\_traseu#);

## Realizarea normalizarii pana la forma normala 3 (FN1-FN3).

### FN1

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

Exemplu de relație care nu ar fi fost în FN1:

DEPARTAMENTE\_TRANSPORT\_STC (non FN1):

Tip_transport#	Nume_departament	Sectii
DT0	Terestru_auto	Autobuz, troleibuz, microbuz
DT1	Tren	Tren, tramvai
DT2	Subteran	Metrou, metrou expres
DT3	Nautic	Vaporas, barca, gondola, hidrobicicleta
DT4	Aerian	Parapanta, elicopter

DEPARTAMENTE\_TRANSPORT\_STC (FN1):

Tip_transport#	Nume_departament	Sectii
DT0	Terestru_auto	Autobuz
DT0	Terestru auto	troleibuz
DT0	Terestru Auto	microbuz
DT1	Tren	Tren
DT1	Tren	tramvai
DT2	Subteran	Metrou
DT2	Subteran	Metrou expres
DT3	Nautic	Vaporas
DT3	Nautic	Barca
DT3	Nautic	Gondola
DT3	Nautic	Hidrobicicleta
DT4	Aerian	Parapanta
DT4	Aerian	elicopter

## 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;

### FILIALE (FN1)

Cod_sectie#	Cod_locatie#	Adresa	Cod_postal
S0	L0	Str. Ceahlăul, nr 3	060371
S1	L1	Calea Serban Voda, nr 288	040223
S2	L2	Str. Manolescu Grig, nr 2	011234
S3	L3	Blv. Regele Mihai I	015449

Cum adresa si codul postal sunt dependente numai de cod\_locatie si nu de intreaga cheie primara, tabelul de mai sus nu este in forma normala 2 si trebui transformat asa:

### FILIALE (FN2)

Cod_sectie#	Cod_locatie#
S0	L0
S1	L1
S2	L2
S3	L3

### LOCATII (FN2)

Cod_locatie#	Adresa	Cod_postal
L0	Str. Ceahlăul, nr 3	060371
L1	Calea Serban Voda, nr 288	040223
L2	Str. Manolescu Grig, nr 2	011234
L3	Blv. Regele Mihai I	015449

### FN3

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 situația de mai jos atributul secției nu depinde direct de cheia primară tip\_transport, depinde de departament.

DEPARTAMENTE\_TRANSPORT\_STC (FN2):

Tip_transport#	Nume_departament	Sectii
DT0	Terestru_auto	Autobuz
DT0	Terestru auto	troleibuz
DT0	Terestru Auto	microbuz
DT1	Tren	Tren
DT1	Tren	tramvai
DT2	Subteran	Metrou
DT2	Subteran	Metrou expres
DT3	Nautic	Vaporas
DT3	Nautic	Barca
DT3	Nautic	Gondola
DT3	Nautic	Hidrobicicleta
DT4	Aerian	Parapanta
DT4	Aerian	elicopter

DEPARTAMENTE\_TRANSPORT\_STC (FN3):

Tip_transport#	Nume_departament
DT0	Terestru_auto
DT1	Tren
DT2	Subteran
DT3	Nautic
DT4	Aerian

SECTII

Sectie#	Nume_departament
Autobuz	Terestru_auto
Troleibuz	Terestru auto
Microbuz	Terestru auto
Tren	Tren
Tramvai	Tren
Metrou	Subteran
Metrou expres	Subteran
Hidrobicicleta	Nautic
Elicopter	Aerian



## Crearea unei secvente ce va fi utilizata in inserarea inregistrarilor in tabele (punctul 11)

**Demo** (complet in fisierul special pentru crearea de tabele):

```
CREATE TABLE angajati_stc(
    cod_angajat VARCHAR2(25) primary key,
    nume VARCHAR(25) CONSTRAINT nume_vid NOT NULL,
    data_angajare DATE default sysdate,
    salariu NUMBER(10) default 2500
);

CREATE TABLE joburi(
    cod_job VARCHAR2(25) primary key,
    nume VARCHAR(25) CONSTRAINT numeJob_vid NOT NULL,
    ore_pe_zi NUMBER(2)
);

CREATE TABLE obisnuit_stc(
    cod_angajat VARCHAR2(25) primary key references angajati_stc(cod_angajat),
    cod_job VARCHAR2(25),
    constraint codJob_fk foreign key(cod_job) references joburi (cod_job)
);

CREATE TABLE sofer_stc(
    cod_angajat VARCHAR2(25) primary key references angajati_stc(cod_angajat),
    tip_permis VARCHAR2(25) constraint permis_conducere NOT NULL
);

ALTER TABLE angajati_stc
ADD (cod_sef VARCHAR2(25) REFERENCES angajati_stc(cod_angajat));

desc angajati_stc;

ALTER TABLE sofer_stc
MODIFY tip_permis default 'z-profesionist';

CREATE TABLE departament_transport_stc(
    tip_transport VARCHAR2(25) primary key,
```

```

    responsabil VARCHAR2(25) references angajati_stc(cod_angajat)
);

ALTER TABLE departament_transport_stc
ADD (nume_departament VARCHAR(25) NOT NULL);

```

Crearea tabelelor in SQL si inserarea de date coerente in fiecare dintre acestea (minimum 5 inregistrari in fiecare tabel neasociativ; minimum 10 inregistrari in tabelele asociative) .

## 1 Departament transport

```

-----
TIP_TRANSPORT      NOT NULL VARCHAR2 (25)
RESPONSABIL                VARCHAR2 (25)
NUME_DEPARTAMENT NOT NULL VARCHAR2 (25)

```

	⚡ TIP_TRANSPORT	⚡ RESPONSABIL	⚡ NUME_DEPARTAMENT	
1	DT0	A0	Terestru_auto	
2	DT1	A3	Tren	
3	DT2	A6	Subteran	
4	DT3	A9	Nautic	
5	DT4	A12	Aerian	

## 2 Sectie

```

-----
COD_SECTIE      NOT NULL VARCHAR2 (25)
RESPONSABIL                VARCHAR2 (25)
TIP_TRANSPORT                VARCHAR2 (25)
NUME_SECTIE                VARCHAR2 (25)

```

	COD_SECTIE	RESPONSABIL	TIP_TRANSPORT	NUME_SECTIE
1	S0	A0	DT0	autobuz
2	S1	A1	DT0	troleibuz
3	S2	A2	DT0	microbuz
4	S3	A3	DT1	tren
5	S4	A4	DT1	tramvai
6	S5	A5	DT2	metrou
7	S6	A6	DT2	metrou expres
8	S7	A7	DT3	vaporas
9	S8	A8	DT3	barca
10	S9	A9	DT3	gondola
11	S10	A10	DT3	hidrobicicleta
12	S11	A11	DT4	parapanta
13	S12	A12	DT4	elicopter

## 2 Angajat

```

-----
COD_ANGAJAT    NOT NULL VARCHAR2 (25)
NUME           NOT NULL VARCHAR2 (25)
DATA_ANGAJARE          DATE
SALARIU          NUMBER (10)
COD_SEF          VARCHAR2 (25)
COD_SECTIE       VARCHAR2 (25)
COD_LOCATIE      VARCHAR2 (25)

```

SQL | All Rows Fetched: 26 in 0.029 seconds

	COD_ANGAJAT	NUME	DATA_ANGAJARE	SALARIU	COD_SEF	COD_SECTIE	COD_LOC...
1	A0	Augustus	12-DEC-90	10000 (null)	S0	L0	
2	A1	Bartolomeu	12-DEC-90	10000 A0	S1	L3	
3	A2	Cicero	12-DEC-90	10000 A0	S2	L8	
4	A3	Dionys	12-DEC-90	10000 (null)	S3	(null)	
5	A4	Elena	12-DEC-90	10000 A3	S4	(null)	
6	A5	Faust	12-DEC-90	10000 A6	S5	(null)	
7	A6	Gregorio	12-DEC-90	10000 (null)	S6	(null)	
8	A7	Heracle	12-DEC-90	10000 A9	S7	(null)	
9	A8	Iulius	12-DEC-90	10000 A9	S8	L4	
10	A9	Juna	12-DEC-90	10000 (null)	S9	(null)	
11	A10	Klaus	12-DEC-90	10000 A9	S10	L10	
12	A11	Lucretia	12-DEC-90	10000 A12	S11	(null)	
13	A12	Manolo	12-DEC-90	10000 (null)	S12	(null)	
14	A13	Nero	12-DEC-95	8000 A0	S0	L1	
15	A14	Ophelia	12-DEC-95	8000 A1	S1	L6	
16	A15	Patrocle	12-DEC-95	8000 A2	S2	L9	
17	A16	Quince	12-DEC-95	8000 A3	S3	(null)	
18	A17	Ruth	12-DEC-95	8000 A4	S4	(null)	
19	A18	Santos	12-DEC-95	8000 A5	S5	(null)	
20	A19	Tremaine	12-DEC-95	8000 A6	S6	(null)	
21	A20	Ulise	12-DEC-95	8000 A7	S7	(null)	
22	A21	Venetia	12-DEC-95	8000 A8	S8	L8	
23	A22	Waldo	12-DEC-95	8000 A9	S9	(null)	
24	A23	Xenia	12-DEC-95	8000 A10	S10	L7	
25	A24	Yuri	12-DEC-95	8000 A11	S11	(null)	
26	A25	Zenobia	12-DEC-95	8000 A12	S12	(null)	

#### 4 SOFER (subentitate)

```
COD_ANGAJAT NOT NULL VARCHAR2(25)
TIP_PERMIS NOT NULL VARCHAR2(25)
```

	COD_ANGAJAT	TIP_PERMIS
1	A13	z-profesionist
2	A14	z-profesionist
3	A15	z-profesionist
4	A16	z-profesionist
5	A17	z-profesionist
6	A18	z-profesionist
7	A19	z-profesionist
8	A20	z-profesionist
9	A21	z-profesionist
10	A22	z-profesionist
11	A23	z-profesionist
12	A24	z-profesionist
13	A25	z-profesionist

#### 5 OBISNUIT (Subentitate)

```
COD_ANGAJAT NOT NULL VARCHAR2(25)
COD_JOB VARCHAR2(25)
```

	COD_ANGAJAT	COD_JOB
1	A0	J3
2	A1	J1
3	A2	J2
4	A3	J3
5	A4	J1
6	A5	J5
7	A6	J3
8	A7	J6
9	A8	J1
10	A9	J3
11	A10	J4
12	A11	J0
13	A12	J3

6 Job

```

-----
COD_JOB    NOT NULL VARCHAR2(25)
NUME       NOT NULL VARCHAR2(25)
ORE_PE_ZI          NUMBER(2)

```

	COD_JOB	NUME	ORE_PE_ZI
1	J0	Contabil	8
2	J1	Asistent_HR	6
3	J2	Programator	6
4	J3	Project_Manager	10
5	J4	Consultant	2
6	J5	Tehnician Securitate	5
7	J6	Mecanic auto	8

## 7 Locatie

```

-----
COD_LOCATIE NOT NULL VARCHAR2(25)
NUME          VARCHAR2(25)
ADRESA        VARCHAR2(25)
COD_POSTAL    NUMBER(7)

```

❖	COD_LOCATIE	❖	NUME	❖	ADRESA	❖	COD_POSTAL
1	L0		Crangasi		Str. Ceahlaul, nr 3		60371
2	L1		Serban		Calea Serban Voda, nr 288		40223
3	L2		Manolescu		Str. Manolescu Grig, nr 2		11234
4	L3		Herastrau		Blv. Regele Mihai I		15449
5	L4		Vianu		Str. Arh. Ion Mincu		18772
6	L5		Gara de Nord		Blv. Dinicu Golescu		60388
7	L6		Dambovita		Splaiul Independentei		25466
8	L7		Aeroport Baneasa		Sos. Bucuresti-Ploiesti		12211
9	L8		Pantelimon		Str. Campului, nr 99		39928
10	L9		Aparatorii Patriei		Str. Dumitru Dumitru		552134
11	L10		Parcul Sebastian		Calea 13 Septembrie, nr 4		59912

## 8 Filiale

```

-----
COD_SECTIE NOT NULL VARCHAR2(25)
COD_LOCATIE NOT NULL VARCHAR2(25)
RESPONSABIL VARCHAR2(25)

```

❖	COD_SECTIE	❖	COD_LOCATIE	❖	RESPONSABIL
1	S0		L0		A0
2	S0		L1		A0
3	S1		L3		A1
4	S1		L5		A1
5	S1		L6		A1
6	S2		L8		A2
7	S2		L9		A2
8	S10		L10		A10
9	S10		L7		A10
10	S8		L4		A8
11	S8		L8		A8

```

-----
COD_TRASEU NOT NULL VARCHAR2 (25)
CAPAT1          VARCHAR2 (25)
CAPAT2          VARCHAR2 (25)
LUNGIME         NUMBER (7)
COD_SECTIE      VARCHAR2 (25)

```

	⚡ COD_TRASEU	⚡ CAPAT1	⚡ CAPAT2	⚡ LUNGIME	⚡ COD_SECTIE
1	T0	St0	St7	13	S0
2	T1	St5	St7	10	S2
3	T2	St4	St3	4	S4
4	T3	St3	St6	15	S7
5	T4	St1	St2	25	S6
6	T5	St1	St7	20	S12

```

COD_STATIE NOT NULL VARCHAR2 (25)
NUME_STATIE VARCHAR2 (25)
COD_LOCATIE VARCHAR2 (25)

```

	⚡ COD_STATIE	⚡ NUME_STATIE	⚡ COD_LOCATIE
1	St0	Crangasi	L0
2	St1	Serban	L1
3	St2	Manolescu	L2
4	St3	Herastrau	L3
5	St4	Vianu	L4
6	St5	Gara de Nord	L5
7	St6	Dambovita	L6
8	St7	Aeroport Baneasa	L7

```

COD_STATIE NOT NULL VARCHAR2 (25)
COD_TRASEU NOT NULL VARCHAR2 (25)

```

	⌘ COD_STATIE	⌘ COD_TRASEU	
1	St0	T0	
2	St1	T0	
3	St1	T1	
4	St2	T4	
5	St2	T5	
6	St3	T3	
7	St4	T3	
8	St5	T1	
9	St6	T0	

```

-----
COD_TIP_VEHICUL NOT NULL VARCHAR2 (25)
NUME                VARCHAR2 (25)
MARCA                VARCHAR2 (25)

```



	NR_INMATRICULARE	COD_TIP_VEHICUL	COD_DEPOU
1	B-45-BUS	TV0	D0
2	B-46-BUS	TV0	D0
3	B-03-TRO	TV1	D0
4	B-04-TRO	TV1	D0
5	B-41-RAM	TV2	D1
6	B-11-RAM	TV2	D1
7	B-00-CAY	TV3	D3
8	B-01-CAY	TV3	D3
9	B-02-CAY	TV3	D3
10	B-04-CAY	TV3	D3
11	B-00-BAR	TV4	D3
12	B-01-BAR	TV4	D3
13	B-0-HELL	TV5	D4
14	B-1-HELL	TV5	D4
15	B-3-HELL	TV5	D4
16	B-99-REN	TV6	D2
17	B-97-REN	TV6	D2
18	B-92-REN	TV6	D2
19	B-30-MTR	TV7	D2
20	B-31-MTR	TV7	D2
21	B-66-MIC	TV8	D0

13 Tipuri\_vehicule

```

-----
COD_TIP_VEHICUL NOT NULL VARCHAR2 (25)
NUME                VARCHAR2 (25)
MARCA                VARCHAR2 (25)

```

SQL | All Rows Fetched: 9 in 0.009 seconds

	COD_TIP_VEICUL	NUME	MARCA
1	TV0	autobuz	bus2000
2	TV1	troleibuz	bus2000
3	TV2	tramvai	rail++
4	TV3	caiac	hidrofun
5	TV4	barca	hidrofun
6	TV5	elicopter	The Bold Eagle
7	TV6	tren	rail++
8	TV7	metrou	rail++
9	TV8	microbuz	bus2000

## 14 Caracteristici

```

-----
COD_CARACT    NOT NULL  NUMBER(3)
NR_LOCURI      NUMBER(3)
NR_ROTI        NUMBER(2)
SERIA          VARCHAR2(10)
MODEL          VARCHAR2(25)
AN             DATE
COMBUSTIBIL    VARCHAR2(25)

```

	⚡ COD_CARACT	⚡ NR_LOCURI	⚡ NR_ROTI	⚡ SERIA	⚡ MODEL	⚡ AN	⚡ COMBUSTIBIL	
1	100	40	4	rx33	infinitybus	17-DEC-15	benzina	
2	101	45	4	rx34	infinitybus2	17-DEC-23	benzina	
3	102	20	4	qqq0	legendtrolley	12-MAR-05	benzina	
4	103	3	0	5005	hydra	01-JAN-10	fara	
5	104	6	0	po21	hydra	17-AUG-08	fara	
6	105	4	3	aero	eagleland	28-MAY-20	diesel	
7	106	50	16	rr41	railmonster	02-MAY-01	electric	
8	107	50	16	rr4i1	railmonster	02-MAY-01	electric	

## 15 Caracteristica\_tip

```

-----
COD_CARACT          NOT NULL NUMBER(3)
COD_TIP_VEHICUL     NOT NULL VARCHAR2(25)
DATA_START          NOT NULL DATE
DATA_EXPIRARE       DATE

```

	⚡ COD_CARACT	⚡ COD_TIP_VEHICUL	⚡ DATA_START	⚡ DATA_EXPIRARE
1	100	TV0	17-DEC-15	16-DEC-23
2	101	TV0	17-DEC-23	(null)
3	107	TV7	02-MAY-01	02-MAY-03
4	106	TV7	03-MAY-03	01-MAY-05
5	107	TV7	02-MAY-05	(null)
6	107	TV6	03-MAY-01	(null)
7	102	TV1	12-MAR-05	(null)
8	107	TV2	03-MAY-01	(null)
9	103	TV3	01-JAN-10	(null)
10	104	TV4	17-AUG-08	(null)
11	105	TV5	28-MAY-20	(null)
12	100	TV8	17-DEC-15	(null)

## 16 Vehicul\_Traseu\_Sofer

```

Name                Null?      Type
-----
COD_ANGAJAT         NOT NULL VARCHAR2(25)
NR_INMATRICULARE    NOT NULL VARCHAR2(10)
COD_TRASEU          NOT NULL VARCHAR2(25)

```

	⚡ COD_ANGAJAT	⚡ NR_INMATRICULARE	⚡ COD_TRASEU
1	A24	B-0-HELL	T5
2	A20	B-00-BAR	T4
3	A22	B-00-CAY	T2
4	A23	B-00-CAY	T2
5	A21	B-01-BAR	T4
6	A23	B-04-CAY	T2
7	A14	B-04-TRO	T0
8	A14	B-04-TRO	T5
9	A25	B-1-HELL	T5
10	A25	B-3-HELL	T3
11	A18	B-30-MTR	T2
12	A19	B-30-MTR	T2
13	A19	B-31-MTR	T2
14	A17	B-41-RAM	T0
15	A13	B-45-BUS	T0
16	A13	B-46-BUS	T0
17	A15	B-66-MIC	T3
18	A16	B-92-REN	T1
19	A16	B-97-REN	T4
20	A16	B-99-REN	T3
21	A16	B-99-REN	T4

```

-----
COD_DEPOU    NOT NULL VARCHAR2 (25)
NUME_DEPOU      VARCHAR2 (25)
CAPACITATE      NUMBER (5)
COD_LOCATIE      VARCHAR2 (25)

```

	COD_DEPOU	NUME_DEPOU	CAPACITATE	COD_LOCATIE
1	D0	Crangasi	302	L0
2	D1	Parcul Sebastian	700	L10
3	D2	Gara de nord	50	L5
4	D3	Herastrau	147	L3
5	D4	Aeroport Baneasa	45	L7

Formulati in limbaj natural si implementati 5 cereri SQL complexe ce vor utiliza, in ansamblul lor, urmatoarele elemente:

- subcereri sincronizate in care intervin cel putin 3 tabele
- subcereri nesincronizate in clauza FROM
- grupari de date cu subcereri nesincronizate in care intervin cel putin 3 tabele, functii grup, filtrare la nivel de grupuri (in cadrul aceleiasi cereri)
- ordonari si utilizarea functiilor NVL si DECODE (in cadrul aceleiasi cereri)
- utilizarea a cel putin 2 functii pe siruri de caractere, 2 functii pe date calendaristice, a cel putin unei expresii CASE
- utilizarea a cel putin 1 bloc de cerere (clauza WITH) Observatie: Intr-o cerere se vor regasi mai multe elemente dintre cele enumerate mai sus, astfel incat cele 5 cereri sa le cuprinda pe toate.

1. Care e suma numarului de locuri ale vehiculelor conduse de soferii din subordinea lui Gregorio sau a subordonatilor sai directi? Atentie! Caracteristicile trebuie sa fie actuale, nu unele expirate!

---subcereri nesincronizate in clauza from, nvl, functie pe date (sysdate)

```
select sum(nr_locuri)
```

```
from(
```

```
select distinct v.nr_inmatriculare,nr_locuri, c.cod_caract, nvl(data_expirare,sysdate+1) exp
```

```
from vehicul_traseu_sofer v join (select cod_angajat
```

```
from angajati_stc
```

```
where cod_sef = (
```

```
select cod_angajat
```

```
from angajati_stc
```

```
where upper(ume) like 'GREGORIO'
```

```
)
```

```
union
```

```
select cod_angajat
```

```
from angajati_stc
```

```
where cod_sef IN (
```

```
select cod_angajat
```

```
from angajati_stc
```

```
where cod_sef = (
```

```
select cod_angajat
```

```
from angajati_stc
```

```
where upper(ume) like 'GREGORIO'
```

```
)
```

```
)
```

```
) aux on (v.cod_angajat=aux.cod_angajat)
```

```
join vehicule_stc v2 on (v.nr_inmatriculare= v2.nr_inmatriculare)
```

```
join tipuri_vehicule_stc tv on (v2.cod_tip_vehicul=tv.cod_tip_vehicul)
```

Query Result x

SQL | All Rows Fetched: 1 in 0.057 seconds

	SUM(NR_LOCUR)
1	100

--subcereri sincronizate, ordonari, 1 functie pe siruri (upper)

```

        where upper(ume)='DIONYS'))
    ) statii

```

```

order by 2 desc;

```

	COD_STATIE	LUNGIME
1	st2	45
2	st1	23
3	st4	15
4	st3	15
5	st5	10

3. Pentru fiecare angajat al firmei afisati numarul de ore lucrate pe zi.Despre soferi se stie ca lucreaza 7 ore pe zi stadard;

```

-- bloc de cerere, nvl, decode, ordonare

```

```

WITH temporar as(

```

```

    select a.cod_angajat, a.ume, nvl(j.cod_job,'SOF') cod_job, nvl(j.ume,'sofer') ume_job
    from angajati_stc a left join obisnuit_stc o on (a.cod_angajat=o.cod_angajat)
        left join joburi j on (o.cod_job=j.cod_job)
    )

```

```

select ume, DECODE (cod_job, 'SOF', 7,
    (select ore_pe_zi from joburi
        where cod_job = temporar.cod_job)) "ore pe zi"
from temporar
order by 2 desc;

```

	NUME	ore pe zi
1	Augustus	10
2	Manolo	10
3	Dionys	10
4	Juna	10
5	Gregorio	10
6	Lucretia	8
7	Heracle	8
8	Patrocle	7
9	Waldo	7
10	Nero	7
11	Quince	7
12	Tremaine	7
13	Xenia	7
14	Venetia	7
15	Zenobia	7
16	Ruth	7
17	Ulise	7
18	Yuri	7
19	Santos	7
20	Ophelia	7
21	Bartolomeu	6
22	Cicero	6
23	Elena	6
24	Iulius	6
25	Faust	5
26	Klaus	2

4. Media salariilor angajatilor care lucreaza in sectii care au filiale in locatii care au in codul postal secventa 99 si in care nu se afla niciun depou.

--grupari de date cu subcereri, functie grup, filtrare la nivel de grup

```
select cod_sectie, avg(salariu) avg_salarii
```

```
from angajati_stc
```

```
group by cod_sectie
```

```
having cod_sectie in(
```

```
    select cod_sectie
```

```
    from filiale_stc
```

```
    where cod_locatie in (
```



```

select cod_locatie
from locatii_stc l
where cod_postal like '%99%'
and cod_locatie not in
(select cod_locatie
from depouri_stc)
)
);

```

Query Result x		
SQL   All Rows Fetched: 2 in 0.02 seconds		
	COD_SECTIE	AVG_SALARII
1	S2	9000
2	S8	9000

5. Afisati suma numarului de roti a tipurilor vehiculelor din sistem. Se vor afisa separat sumele pentru vehiculele

care nu se mai folosesc, fiecare data de expirare separat, iar cele inca actuale impreuna. Vor fi luate in considerare

numai vehiculele care au fost cumparate intr-o luna cu 31 de zile (data de start= data de cumparare).

--caracteristicile care au data strat intr o luna cu 31 de zile

-- doua fct pentru date, case, initcap (fct pe siruri), subcerere sincronizata

```

WITH info_date as (
select cod_caract,extract(day from(LAST_DAY(data_start))) ult_zi,
CASE WHEN data_expirare > (select max(data_start)
from caracteristica_tip
where ct.cod_tip_vehicul=cod_tip_vehicul)
THEN 'something went wrong'

```

```

        WHEN data_expirare <= (select max(data_start)
                                from caracteristica_tip
                                where ct.cod_tip_vehicul=cod_tip_vehicul)
        THEN to_char(data_expirare)
        ELSE initcap('inca valabil')
        END as data_expirare
from caracteristica_tip ct
where extract(day from(LAST_DAY(data_start)))=31
)
select sum(nr_roti), data_expirare
from info_date i join caracteristici_stc c on (c.cod_caract= i.cod_caract)
group by data_expirare;

```

	⚡ SUM(NR_ROTI)	⚡ DATA_EXPIRARE	
1	16	01-MAY-05	
2	16	02-MAY-03	
3	63	Inca Valabil	
4	4	16-DEC-23	

Implementarea a 3 operatii de actualizare si de suprimare a datelor utilizand subcereri.

--1. updatam responsabilii pe angajati cu responsabil fiind responsabil de departament

```
select * from angajati_stc;
```

```
select * from sectii_stc;
```

```
UPDATE angajati_stc a
```

```
set cod_sef = (select d.responsabil
```

```
from departament_transport_stc d join sectii_stc s on(d.tip_transport=s.tip_transport)
```

```
where s.cod_sectie=a.cod_sectie and cod_angajat!=d.responsabil)
```

where salariu>8000;

13 rows updated.

----2. update filiale

select\*from filiale\_stc;

UPDATE filiale\_stc f

set responsabil = (select responsabil

from sectii\_stc

where cod\_sectie= f.cod\_sectie);

commit;

11 rows updated.

---3. suprimare date din locatii - stergem locatiile in care nu se gasesc nici filiale nici depouri nici statii

INSERT INTO locatii\_stc

VALUES ('L11','Nicaieri','Nicaieri, nr2', '000000');

select \* from locatii\_stc;

DELETE from locatii\_stc

WHERE cod\_locatie not in ((select cod\_locatie

from filiale\_stc)

union

(select cod\_locatie

from statii\_stc)

union

(select cod\_locatie

from depouri\_stc));

	⚡ COD_LOCATIE	⚡ NUME	⚡ ADRESA	⚡ COD_POSTAL
1	L11	Nicaieri	Nicaieri, nr2	0
2	L0	Crangasi	Str. Ceahlaul, nr 3	60371
3	L1	Serban	Calea Serban Voda, nr 288	40223
4	L2	Manolescu	Str. Manolescu Grig, nr 2	11234
5	L3	Herastrau	Blv. Regele Mihai I	15449
6	L4	Vianu	Str. Arh. Ion Mincu	18772
7	L5	Gara de Nord	Blv. Dinicu Golescu	60388
8	L6	Dambovita	Splaiul Independentei	25466
9	L7	Aeroport Baneasa	Sos. Bucuresti-Ploiesti	12211
10	L8	Pantelimon	Str. Campului, nr 99	39928
11	L9	Aparatorii Patriei	Str. Dumitru Dumitru	552134
12	L10	Parcul Sebastian	Calea 13 Septembrie, nr 4	59912

1 row deleted.

	⚡ COD_LOCATIE	⚡ NUME	⚡ ADRESA	⚡ COD_POSTAL
1	L0	Crangasi	Str. Ceahlaul, nr 3	60371
2	L1	Serban	Calea Serban Voda, nr 288	40223
3	L2	Manolescu	Str. Manolescu Grig, nr 2	11234
4	L3	Herastrau	Blv. Regele Mihai I	15449
5	L4	Vianu	Str. Arh. Ion Mincu	18772
6	L5	Gara de Nord	Blv. Dinicu Golescu	60388
7	L6	Dambovita	Splaiul Independentei	25466
8	L7	Aeroport Baneasa	Sos. Bucuresti-Ploiesti	12211
9	L8	Pantelimon	Str. Campului, nr 99	39928
10	L9	Aparatorii Patriei	Str. Dumitru Dumitru	552134
11	L10	Parcul Sebastian	Calea 13 Septembrie, nr 4	59912

Formulati in limbaj natural si implementati in SQL: o cerere ce utilizeaza operatia outer-join pe minimum 4 tabele, o cerere ce utilizează operatia division si o cerere care implementează analiza top-n. Observație: Cele 3 cereri sunt diferite de cererile de la exercițiul 12.

--top n

1. Afisati primii 10 angajati din firma dupa lungimea numelui.

```
select length(ume), ume
from (
    select length(ume), ume
    from angajati_stc
    --where length(ume)>5
    order by 1 desc
)
where rownum<11;
```

	LENGTH(UME)	UME
1	10	Bartolomeu
2	8	Augustus
3	8	Gregorio
4	8	Tremaine
5	8	Patrocle
6	8	Lucretia
7	7	Heracle
8	7	Ophelia
9	7	Venetia
10	7	Zenobia

--division

2. Afisati numele tuturor soferilor care conduc vehicule pe exact aceleasi trasee ca si Waldo.

```

select a.numa
from vehicul_traseu_sofer v join angajati_stc a on (v.cod_angajat=a.cod_angajat)
where cod_traseu in ( select cod_traseu
                      from vehicul_traseu_sofer vts join angajati_stc a on (vts.cod_angajat=a.cod_angajat)
                      where lower(numa) ='waldo')
and lower(a.numa) !='waldo'
and (select count(*)
     from vehicul_traseu_sofer
     where v.cod_angajat=cod_angajat)= (select count(*)
                                         from
                                         (select cod_traseu
                                          from vehicul_traseu_sofer vts join angajati_stc a on (vts.cod_angajat=a.cod_angajat)
                                          where lower(numa) ='waldo')
                                         );

```



	NUMA
1	Santos

--outer join pe 4 tabele

3. Afisati numarul de locatii, apoi in cate locatii diferite se afla sectiile, depourile si statiile.

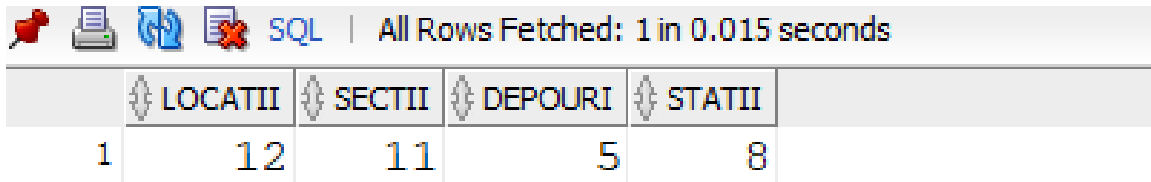
```

select count(cod_locatie) locatii, count(cod_sectie) sectii, count(cod_depou) depouri, count (cod_statie) statii
from(
    select l.cod_locatie , f.cod_sectie, d.cod_depou, s.cod_statie
    from filiale_stc f right join locatii_stc l on (l.cod_locatie=f.cod_locatie)
        left join depouri_stc d on (l.cod_locatie=d.cod_locatie)

```

```
left join statii_stc s on (s.cod_locatie=l.cod_locatie)
```

```
);
```



	LOCATII	SECTII	DEPOURI	STATII
1	12	11	5	8

Optimizarea unei cereri, aplicand regulile de optimizare ce deriva din proprietatile operatorilor algebrei relationale. Cererea va fi exprimata prin expresie algebrica, arbore algebric si limbaj (SQL), atat anterior cat si ulterior optimizarii.

```
select nume from
(
select *
from vehicul_traseu_sofer v join angajati_stc a on (v.cod_angajat=a.cod_angajat)
where cod_traseu in ( select cod_traseu
                      from vehicul_traseu_sofer vts join angajati_stc a on (vts.cod_angajat=a.cod_angajat)
                      where lower(nume) ='waldo')
and lower(a.nume) !='waldo'
and (select count(*)
     from vehicul_traseu_sofer
     where v.cod_angajat=cod_angajat)= (select count(*)
                                         from
                                         (select cod_traseu
                                          from vehicul_traseu_sofer vts join angajati_stc a on (vts.cod_angajat=a.cod_angajat)
                                          where lower(nume) ='waldo')
                                         ));
```

**Expresie algebrica:**

```
R1= SELECT (ANGAJATI_STC, lower(nume)= 'waldo');
```

R2= PROJECT (R1, cod\_angajat);  
R3= PROJECT (VEHICUL\_TRASEU\_SOFTER,cod\_angajat);  
R4= SEMIJOIN (R2,R3, cod\_angajat);  
R5= PROJECT (R4, cod\_traseu); --rez1  
R6 = SELECT (VEHICUL\_TRASEU\_SOFTER, cod\_traseu in R5);

R13 = SELECT (VEHICUL\_TRASEU\_SOFTER, cod\_angajat = v.cod\_angajat);  
R14= PROJECT (R13, count(\*));  
R15= PROJECT (ANGAJATI\_STC, cod\_angajat);  
R16= PROJECT (VEHICUL\_TRASEU\_SOFTER,cod\_angajat);  
R17= SEMIJOIN (R15,R16);  
R18= SELECT (R17, nume='waldo');  
R19=PROJECT (R18, count(\*));  
R20= SELECT(VEHICUL\_TRASEU\_SOFTER, R14=R19);

R21= INTERSECT (R6,R20);

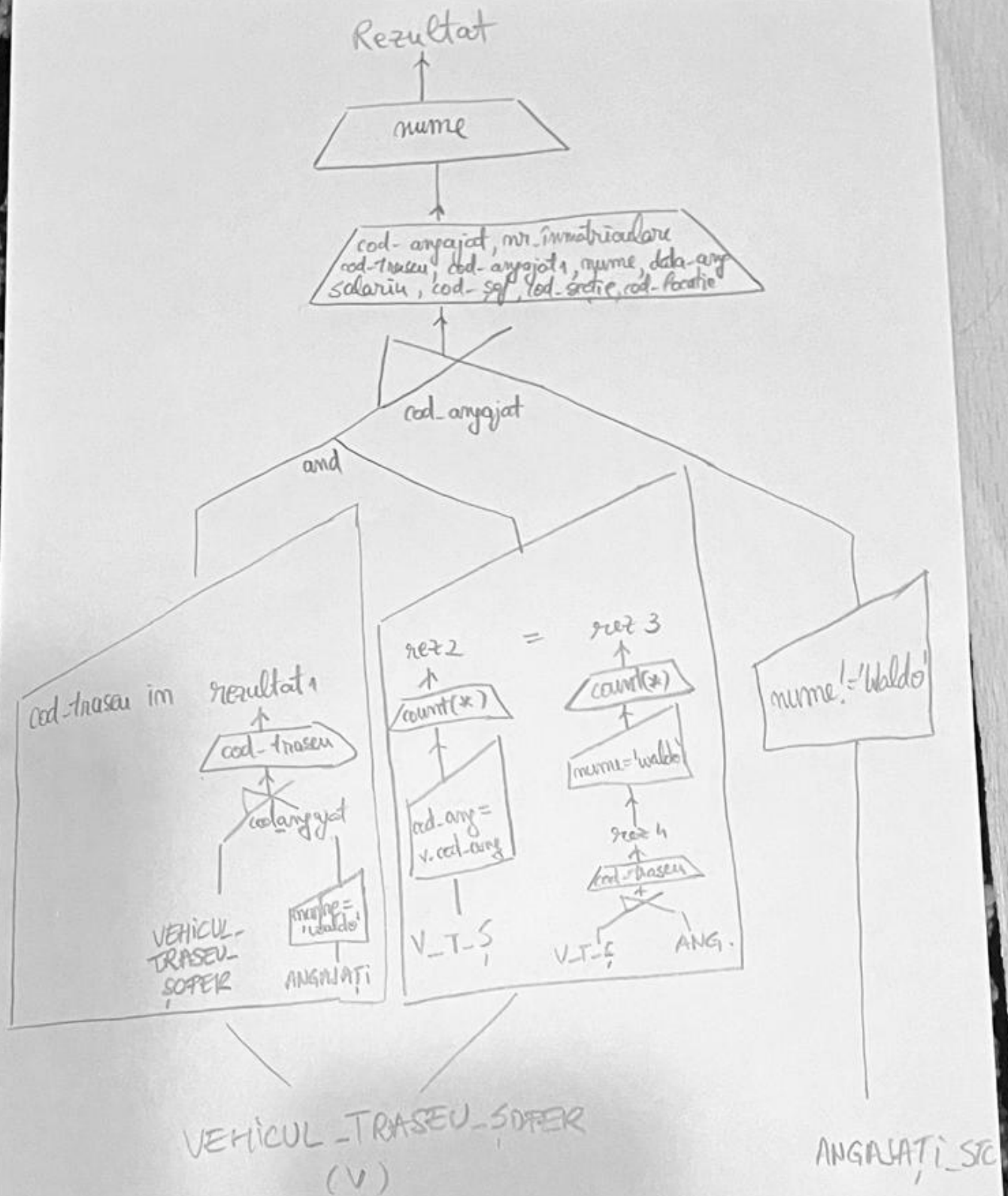
R7= PROJECT (R21, cod\_angajat);

R8= SELECT (ANGAJATI\_STC, lower(nume)='waldo');  
R9= PROJECT (R8, cod\_angajat);

R10= SEMIJOIN (R7, R9, cod\_angajat);  
R11= PROJECT(R10, \*);  
R12= PROJECT (R11, nume);

Arbore algebric:





Optimizarea se face aplicand regula 1, se aplica selecturile cat mai devreme posibil, deci scapam de selectul redundant de la final.

```
select a.nume
from vehicul_traseu_sofer v join angajati_stc a on (v.cod_angajat=a.cod_angajat)
where cod_traseu in ( select cod_traseu
                      from vehicul_traseu_sofer vts join angajati_stc a on (vts.cod_angajat=a.cod_angajat)
                      where lower(nume) ='waldo')
and lower(a.nume) !='waldo'
and (select count(*)
     from vehicul_traseu_sofer
     where v.cod_angajat=cod_angajat)= (select count(*)
                                         from
                                         (select cod_traseu
                                          from vehicul_traseu_sofer vts join angajati_stc a on (vts.cod_angajat=a.cod_angajat)
                                          where lower(nume) ='waldo')
                                         );
```

**Expresia algebrica** optimizata va fi;

R1= SELECT (ANGAJATI\_STC, lower(nume)= 'waldo');

R2= PROJECT (R1, cod\_angajat);

R3= PROJECT (VEHICUL\_TRASEU\_SOFTER,cod\_angajat);

R4= SEMIJOIN (R2,R3, cod\_angajat);

R5= PROJECT (R4, cod\_traseu); --rez1

R6 = SELECT (VEHICUL\_TRASEU\_SOFTER, cod\_traseu in R5);

R13 = SELECT (VEHICUL\_TRASEU\_SOFTER, cod\_angajat = v.cod\_angajat);

R14= PROJECT (R13, count(\*));

R15= PROJECT (ANGAJATI\_STC, cod\_angajat);

R16= PROJECT (VEHICUL\_TRASEU\_SOFTER,cod\_angajat);

R17= SEMIJOIN (R15,R16);

R18= SELECT (R17, nume='waldo');

R19=PROJECT (R18, count(\*));

R20= SELECT(VEHICUL\_TRASEU\_SOFTER, R14=R19);

R21= INTERSECT (R6,R20);

R7= PROJECT (R21, cod\_angajat);

R8= SELECT (ANGAJATI\_STC, lower(ume)='waldo');

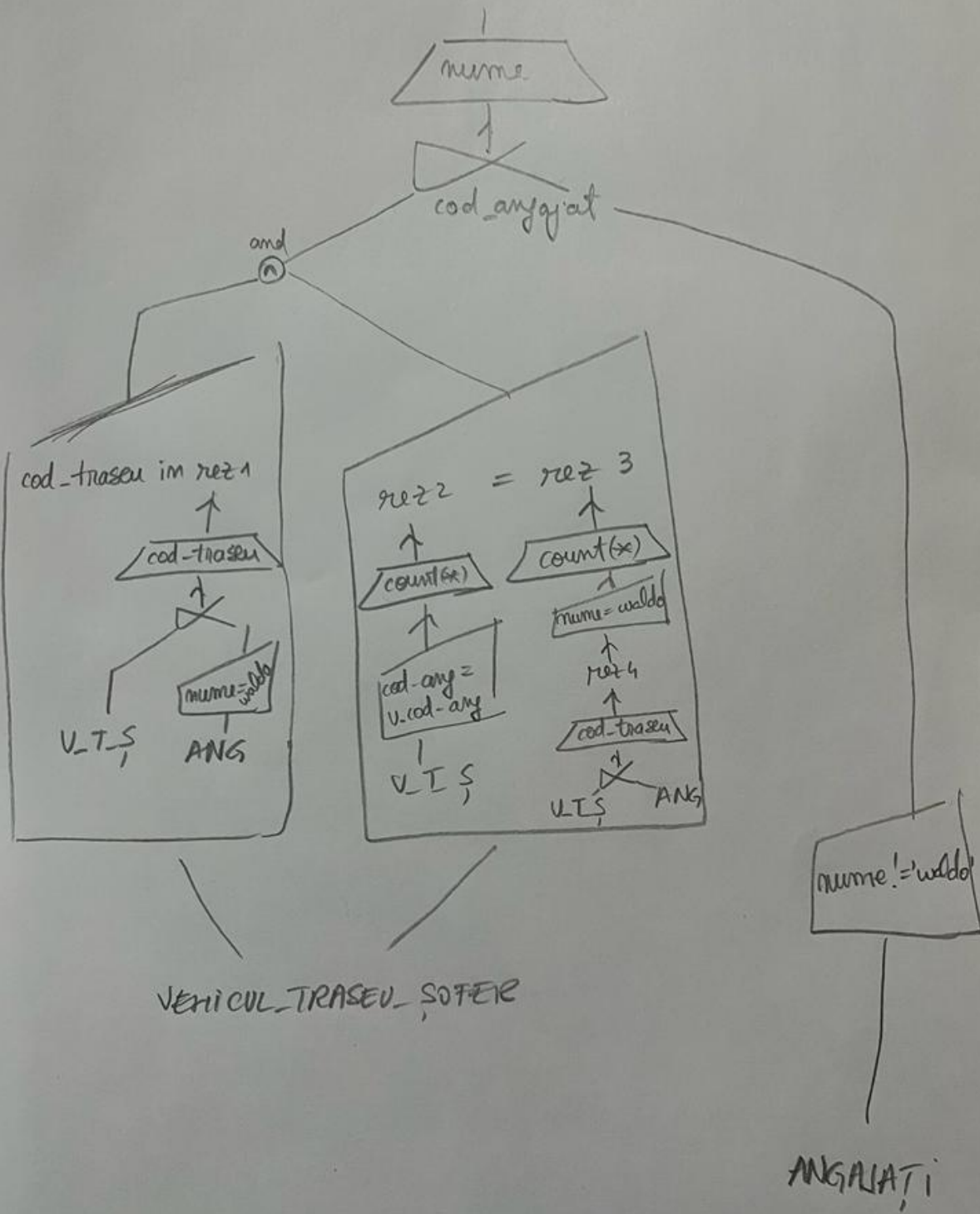
R9= PROJECT (R8, cod\_angajat);

R10= SEMIJOIN (R7, R9, cod\_angajat);

R11= PROJECT(R10, ume);

**Arbore algebric:**

OPTIMIZAT



Realizarea normalizarii BCNF, FN4, FN5. Aplicarea denormalizarii, justificand necesitatea acesteia.

BCNF

Consideram exemplul urmator de tabel non **BCNF**.

Avem VEHICUL\_TRASEU\_SOFER

Nr_inmatriculare#	Cod_angajat#	Cod traseu	Lungime
B-41-RAM	A17	T0	13
B-04-TRO	A14	T0	13
B-30-MTR	A19	T2	4

Se observa cheia candidat cod traseu. Pentru aducerea in BNCF se creeaza noul tabel: trasee\_stc, asa ca in baza de date prezentata in cadrul proiectului.

FN4

Pentru **forma normala 4** vom lua tabelul VEHICUL\_TRASEU\_SOFER.

VEHICUL\_TRASEU\_SOFER (cod\_angajat#, nr\_inmatriculare#, cod\_traseu#);

	⚡ COD_ANGAJAT	⚡ NR_INMATRICULARE	⚡ COD_TRASEU
1	A24	B-0-HELL	T5
2	A20	B-00-BAR	T4
3	A22	B-00-CAY	T2
4	A23	B-00-CAY	T2
5	A21	B-01-BAR	T4
6	A23	B-04-CAY	T2
7	A14	B-04-TRO	T0
8	A14	B-04-TRO	T5
9	A25	B-1-HELL	T5
10	A25	B-3-HELL	T3
11	A18	B-30-MTR	T2
12	A19	B-30-MTR	T2
13	A19	B-31-MTR	T2
14	A17	B-41-RAM	T0
15	A13	B-45-BUS	T0
16	A13	B-46-BUS	T0
17	A15	B-66-MIC	T3
18	A16	B-92-REN	T1
19	A16	B-97-REN	T4
20	A16	B-99-REN	T3
21	A16	B-99-REN	T4

Avem dependenta multipla :

Sofer ->-> Vehicul, respectiv Sofer ->->Traseu;

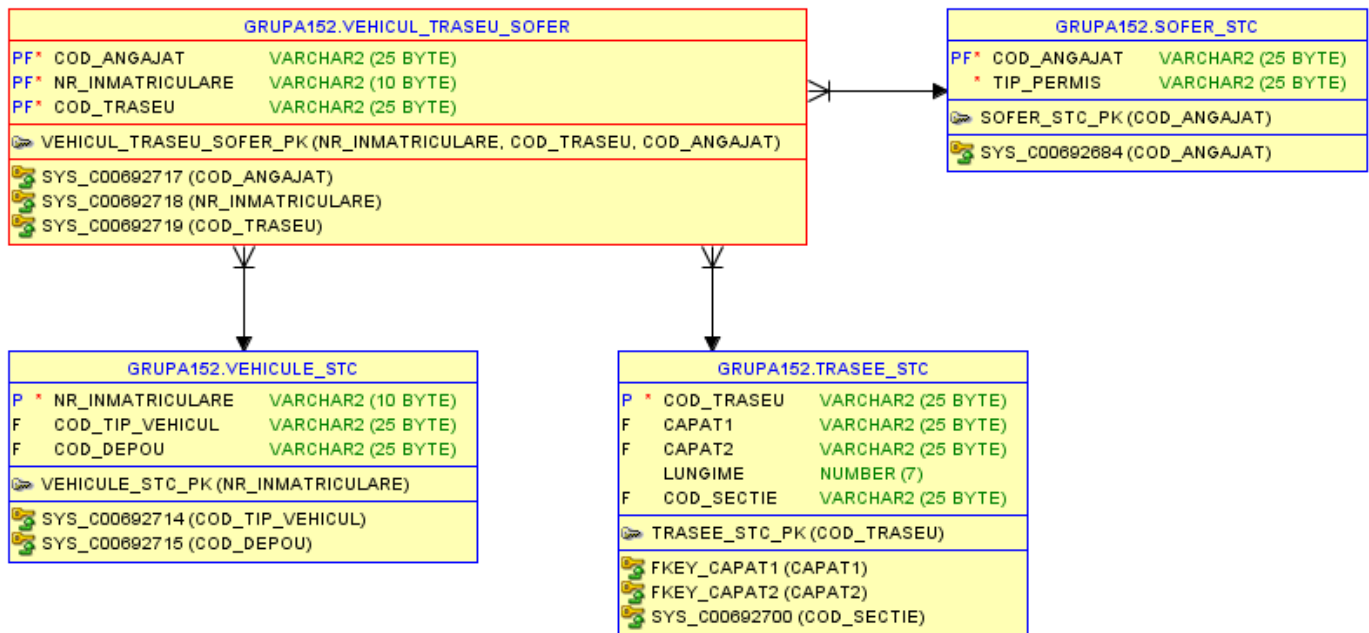
Relatiile dupa aplicarea lui FN4 sunt :

R1 (cod\_angajat#, nr\_inmatriculare#);

R2 (cod\_angajatr#, cod\_traseu#)

## FN5

Pentru forma normala 5 vom lua exemplu relatia de tip trei care contnine dependente ciclice dintre entitatile sofer\_stc, trasee\_stc si vehicule\_stc;



Pentru aducerea in forma normala 5 in loc de a avea un singur tabel asociativ cu trei chei primare, vom avea trei tabele asociative cu cate doua chei primare fiecare, deci vom avea relatiile:

R1= (cod\_angajat#, cod\_traseu#);

R2= (cod\_traseu#, cod\_vehicul#);

R3= (cod\_vehicul#, cod\_angajat#);

## DENORMALIZARE

Avand in vedere aducerea in forma normala 5, vom dori sa denormalizam, sa ne intoarcem la un singur tabel asociativ cu trei chei primare, pentru a micsora numarul de joinuri executate in procesarea cererilor sql din baza da date. Cu riscul de a mari redundanta, avem un timp de procesare mai mic, deci suntem mai eficienti.