*VERDEȘ MARIA-IOANA*

*GRUPA 131*

# Proiect Baze de Date 2023

# **Mercerie**

Cuprins

[***Proiect Baze de Data 1***](#_Toc135999313)

[***Mercerie 1***](#_Toc135999314)

[***1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare 3***](#_Toc135999315)

[***2. Prezentarea constrângerilor(restricții, reguli) impuse asupra modelului. 3***](#_Toc135999316)

[***3. Descrierea entităților, incluzând precizarea cheii primare 3***](#_Toc135999317)

[***4. Descrierea relațiilor, incluzând precizarea cardinalității acestora 4***](#_Toc135999318)

[***5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor 5***](#_Toc135999319)

[***6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5 9***](#_Toc135999320)

[***7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6 9***](#_Toc135999321)

[***8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7 10***](#_Toc135999322)

[***9. Realizarea normalizării până la forma normală 3(FN1-FN3) 10***](#_Toc135999323)

[***10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele 11***](#_Toc135999324)

[***11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative) 15***](#_Toc135999325)

[***12. Formulați în limbaj natural și implementați 5 cereri SQL complexe 38***](#_Toc135999326)

[***13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri 44***](#_Toc135999327)

[***14. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n 46***](#_Toc135999328)

[***15. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. 49***](#_Toc135999329)

[***16. a. Realizarea normalizării BCNF, FN4, FN5. 52***](#_Toc135999330)

[***b. Aplicarea denormalizării, justificând necesitatea acesteia. 52***](#_Toc135999331)

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

Merceria Granny Mary oferă clienților săi, pe lângă servicii clasice de vânzare-cumpărare, oportunitatea de a se înscrie la cursuri de croșetat, împletit sau brodat. Astfel angajații săi se împart în instructori și vânzători, despre care se cunosc, în plus pe lângă detaliile asociate fiecărui angajat (nume, număr de telefon, email), data angajării, respectiv tura pe care lucrează.

Există atât cursuri de avansați, cât și de începători, despre care sunt cunoscute data de început și durata acestora, măsurată în săptămâni. În funcție de ziua din săptămână, acestea se organizează în unul sau mai multe sedii, ce se află în București sau în Piatra Neamț. În cadrul cursurilor, cursanții înregistrați realizează diverse proiecte pe care primesc note. Notele se iau în considerare pentru realizarea mediei finale a cursantului la cursul respectiv doar dacă data de predare a proiectului precedă termenul limită.

Vânzările sunt înregistrate sub forma unor tranzacții. Fiecare tranzacție realizată de un vânzător la o anumită dată se împarte de fapt în una sau mai multe comenzi către stoc, fiecare comandă solicitând o anumită cantitate dintr-un singur produs. Pentru fiecare produs din stoc se cunoaște numele, prețul și tipul acestuia. Furnizorii pot livra mai multe tipuri de produse, dar nu există mai mulți furnizori a căror ofertă să se refere la același tip de produs.

## Prezentarea constrângerilor(restricții, reguli) impuse asupra modelului.

* Pentru sedii nu este obligatorie menționarea străzii, dar trebuie menționat orașul, care va fi fie București, fie Piatra-Neamț.
* Tipul unui curs poate avea următoarele valori: croșetat, împletit, brodat.
* Nivelul unui curs poate fi fie ”avansați”, fie ”începători”.
* Durata unui curs se măsoară în săptămâni.
* Dificultatea proiectelor este un număr natural între 1 și 5.
* Emailul angajaților, cursanților sau a furnizorilor nu este obligatoriu, dar numărul de telefon este.
* Un angajat trebuie să fie instructor sau vânzător, dar nu ambele.
* Tura unui vânzător poate avea valori: dimineață, după-amiază, seară.

## Descrierea entităților, incluzând precizarea cheii primare

|  |  |  |
| --- | --- | --- |
| ENTITATE | DESCRIERE | CHEIE PRIMARĂ |
| ANGAJAȚI | Informații comune tuturor angajaților merceriei. | id\_angajat |
| INSTRUCTOR | Informații specifice angajaților care predau cursuri. | id\_angajat |
| VÂNZĂTOR | Informații specifice angajaților care se ocupă de vânzarea produselor. | id\_angajat |
| CURS | Informații cu privire la cursurile susținute de către angajații merceriei. | id\_curs |
| SEDII | Informații cu privire la sediile în care se organizează cursurile. | id\_sediu |
| PROIECT | Informații generale despre proiectele atribuite cursanților în cadrul cursurilor. | id\_proiect |
| CURSANT | Informații despre cei care s-au înscris la cel puțin un curs organizat de mercerie. | id\_cursant |
| TRANZACȚII | Informații despre tranzacțiile realizate de către vânzătorii merceriei. | id\_tranzacție |
| COMANDĂ | Informații despre comenzile către stoc care compun tranzacțiile realizare de vânzători. | id\_tranzactie,  id\_comanda |
| STOC | Informații despre produsele care se regăsesc în stocul merceriei. | id\_produs |
| TIPURI\_PRODUSE | Informații generale despre tipurile de produse vândute de mercerie. | id\_tip |
| FURNIZORI | Informații generale despre furnizorii produselor merceriei. | id\_furnizor |

## Descrierea relațiilor, incluzând precizarea cardinalității acestora

|  |  |  |  |
| --- | --- | --- | --- |
| RELAȚIE | ENTITĂȚI | CARDINALITATE | DESCRIERE |
| se\_organizează\_în | SEDII – CURS | many-to-many | Un curs se organizează în unul sau mai multe sedii, iar într-un sediu se pot organiza unul sau mai multe cursuri diferite. În funcție de ziua din săptămână, un curs se poate organiza de mai multe ori în același sediu. |
| participă\_la | CURS – PROIECT – CURSANT | many-to-many | Un cursant participă la unul sau mai multe cursuri, în cadrul cărora realizează unul sau mai multe proiecte. Un proiect poate fi realizat de către mai mulți cursanți în cadrul a mai multe cursuri. Fiecare proiect are un termen limită în funcție de cursul în cadrul căruia trebuie realizat, este predat la o anumită dată și primește o anumită notă. |
| predă | INSTRUCTOR – CURS | one-to-many | Un instructor predă mai multe cursuri, iar un curs este predat de un singur instructor. |
| ISA | ANGAJAȚI –  INSTRUCTOR,  ANGAJAȚI –  VÂNZĂTOR | one-to-one | Fiecare angajat este fie un instructor, fie un vânzător. Un instructor este un singur angajat, iar un vânzător este un singur angajat. |
| realizează | VÂNZĂTOR –  TRANZACȚII | one-to-many | Un vânzător realizează mai multe tranzacții, la o anumită data. O tranzacție este realizată de un singur vânzător. |
| implică | TRANZACȚII - COMANDĂ | one-to-many | O tranzacție implică mai multe comenzi către stoc. O comandă aparține unei singure tranzacții. |
| necesită | STOC - COMANDĂ | one-to-many | Un produs din stoc aparține mai multor comenzi. O comandă necesită un singur produs din stoc. |
| se\_încadrează\_în | TIPURI\_PRODUSE – STOC | one-to-many | Un tip de produs este asociat mai multor produse din stoc, iar un produs din stoc se încadrează într-un singur tip. |
| furnizează | FURNIZORI – TIPURI\_PRODUSE | one-to-many | Un furnizor furnizează mai multe tipuri de produse, iar un anumit tip de produs este furnizat de către un singur furnizor. |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ANGAJAȚI | | | | |
| ATRIBUT | TIP DE DATE | DIMENSIUNE | VALORI DEFAULT / POSIBILE | CONSTRÂNGERI |
| id\_angajat | number | 4 | - | PK |
| nume | varchar2 | 20 | - | NOT NULL,  UNIQUE (împreună cu prenume) |
| prenume | varchar2 | 20 | - | NOT NULL, UNIQUE (împreună cu nume) |
| număr\_de\_telefon | varchar2 | 20 | - | NOT NULL, UNIQUE |
| email | varchar2 | 20 | - | UNIQUE |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| INSTRUCTOR | | | | |
| ATRIBUT | TIP DE DATE | DIMENSIUNE | VALORI DEFAULT / POSIBILE | CONSTRÂNGERI |
| id­\_angajat | number | 4 | - | PK, FK (on delete cascade) |
| data\_angajare | date | - | default sysdate | - |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VÂNZĂTOR | | | | |
| ATRIBUT | TIP DE DATE | DIMENSIUNE | VALORI DEFAULT / POSIBILE | CONSTRÂNGERI |
| id\_angajat | number | 4 | - | PK, FK (on delete cascade) |
| tura | varchar2 | 15 | dimineața / după-amiaza / seara | NOT NULL |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CURS | | | | |
| ATRIBUT | TIP DE DATE | DIMENSIUNE | VALORI DEFAULT / POSIBILE | CONSTRÂNGERI |
| id\_curs | number | 3 | - | PK |
| id\_angajat | number | 4 | - | FK (on delete set null) |
| tip | varchar2 | 15 | croșetat / împletit / brodat | - |
| nivel | varchar2 | 15 | avansați / începători | NOT NULL |
| data\_început | date | - | default sysdate | - |
| durata | number | 2 | - | CHECK (> 0) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SEDII | | | | |
| ATRIBUT | TIP DE DATE | DIMENSIUNE | VALORI DEFAULT / POSIBILE | CONSTRÂNGERI |
| id\_sediu | number | 3 | - | PK |
| strada | varchar2 | 25 | - | - |
| oraș | varchar2 | 25 | - | NOT NULL |
| email | varchar2 | 25 | - | UNIQUE |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CURSANT | | | | |
| ATRIBUT | TIP DE DATE | DIMENSIUNE | VALORI DEFAULT / POSIBILE | CONSTRÂNGERI |
| id\_cursant | number | 4 | - | PK |
| nume\_cursant | varchar2 | 20 | - | NOT NULL, UNIQUE (împreună cu prenume) |
| prenume\_cursant | varchar2 | 20 | - | NOT NULL, UNIQUE (împreună cu nume) |
| data\_înregistrare | date | - | default sysdate | - |
| număr\_de\_telefon | varchar2 | 20 | - | NOT NULL, UNIQUE |
| email | varchar2 | 20 | - | UNIQUE |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PROIECT | | | | |
| ATRIBUT | TIP DE DATE | DIMENSIUNE | VALORI DEFAULT / POSIBILE | CONSTRÂNGERI |
| id\_proiect | number | 3 | - | PK |
| nume | varchar2 | 25 | - | NOT NULL |
| dificultate | number | 1 | 1 / 2 / 3 / 4 / 5 | CHECK(> 0), CHECK(< 6) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TRANZACȚII | | | | |
| ATRIBUT | TIP DE DATE | DIMENSIUNE | VALORI DEFAULT / POSIBILE | CONSTRÂNGERI |
| id\_tranzacție | number | 3 | - | PK |
| id\_angajat | number | 4 | - | FK (on delete set null) |
| dată | date | - | default sysdate | - |

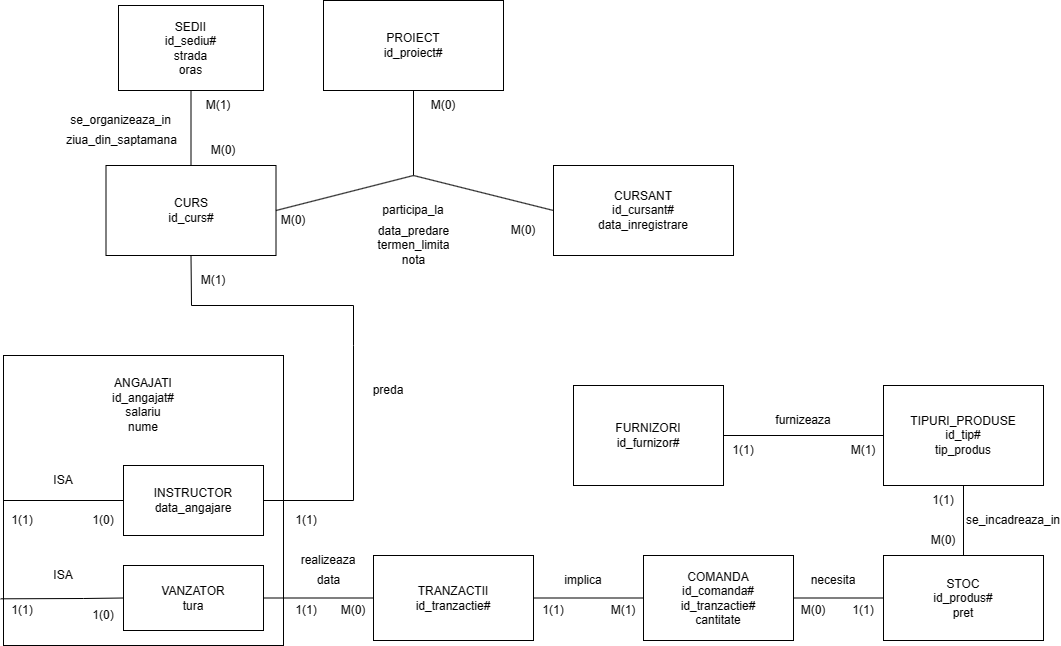
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COMANDĂ | | | | |
| ATRIBUT | TIP DE DATE | DIMENSIUNE | VALORI DEFAULT / POSIBILE | CONSTRÂNGERI |
| id\_comandă | number | 3 | - | PK |
| id\_tranzacție | number | 3 | - | PK, FK (on delete cascade) |
| id\_produs | number | 3 | - | FK (on delete set null) |
| cantitate | number | 2 | - | CHECK (> 0) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STOC | | | | |
| ATRIBUT | TIP DE DATE | DIMENSIUNE | VALORI DEFAULT / POSIBILE | CONSTRÂNGERI |
| id\_produs | number | 3 | - | PK |
| id\_tip | number | 3 | - | FK (on delete set null) |
| nume\_produs | varchar2 | 25 | - | NOT NULL |
| preț | number | 5, 2 | - | CHECK (> 0) |

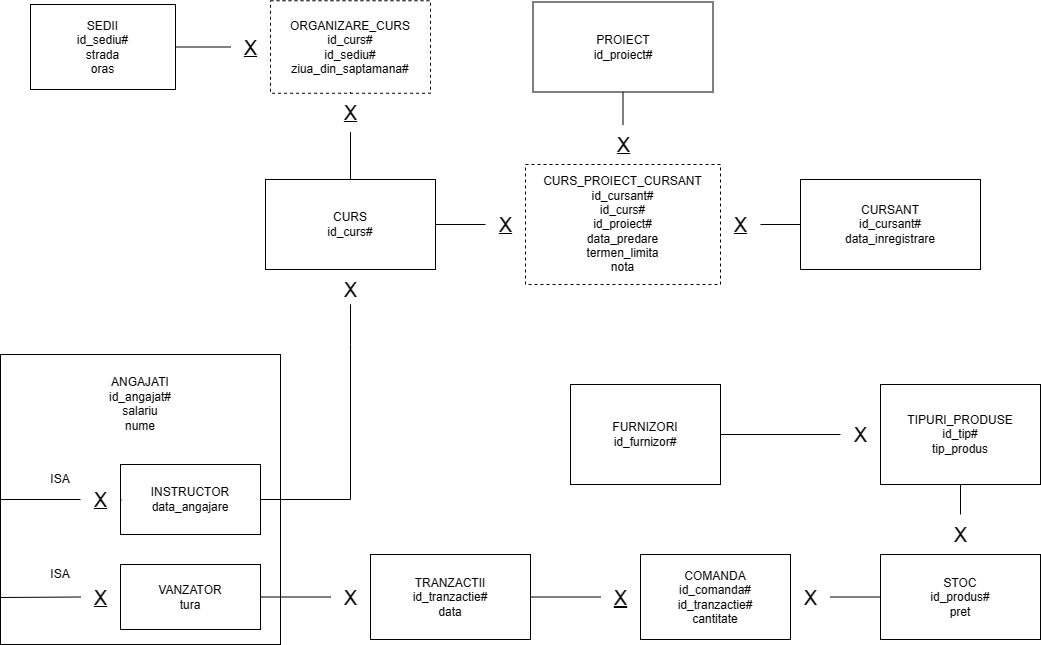
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TIPURI\_PRODUSE | | | | |
| ATRIBUT | TIP DE DATE | DIMENSIUNE | VALORI DEFAULT / POSIBILE | CONSTRÂNGERI |
| id\_tip | number | 3 | - | PK |
| id\_furnizor | number | 4 | - | FK (on delete set null) |
| tip\_produs | varchar2 | 25 | - | NOT NULL |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FURNIZORI | | | | |
| ATRIBUT | TIP DE DATE | DIMENSIUNE | VALORI DEFAULT / POSIBILE | CONSTRÂNGERI |
| id\_furnizor | number | 4 | - | PK |
| nume\_furnizor | varchar2 | 25 | - | NOT NULL |
| număr\_de\_telefon | varchar2 | 20 | - | NOT NULL, UNIQUE |
| email | varchar2 | 20 | - | UNIQUE |

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



## Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6



## Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7

ANGAJATI(id\_angajat#, nume, prenume, numar\_de\_telefon, email);

INSTRUCTOR(id\_angajat#, data\_angajare);

VANZATOR(id\_angajat#, tura);

SEDII(id\_sediu#, strada, oras, email);

CURS(id\_curs#, id\_angajat, tip, nivel, data\_inceput, durata);

ORGANIZARE\_CURS(id\_curs#, id\_sediu#, ziua\_din\_saptamana#);

CURSANT(id\_cursant#, nume\_cursant, prenume\_cursant, data\_inregistrare, numar\_de\_telefon, email);

PROIECT(id\_proiect#, nume, dificultate, data\_predare, termen\_limita, nota);

CURS\_PROIECT\_CURSANT(id\_cursant#, id\_curs#, id\_proiect#);

TRANZACTII(id\_tranzactie#, id\_angajat, data);

COMANDA(id\_comanda#, id\_tranzactie#, id\_produs, cantitate);

STOC(id\_produs#, id\_tip, nume\_produs, pret);

TIPURI\_PRODUSE(id\_tip#, id\_furnizor, tip\_produs);

FURNIZORI(id\_furnizor#, nume\_furnizor, numar\_de\_telefon, email);

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

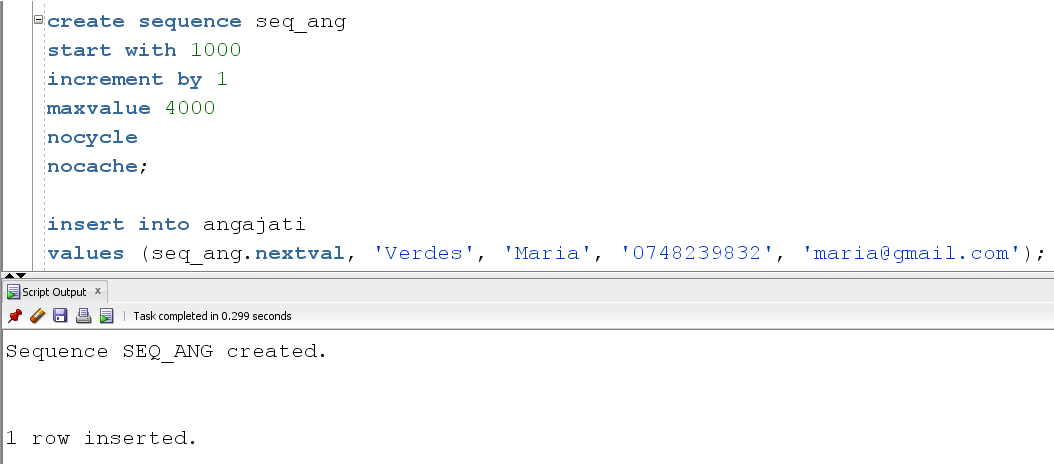
**FN1** – Considerăm tabela STOC cu următoarea schemă relațională: STOC(id\_produs#, id\_tip#, nume\_produs, pret, tip\_produs, nume\_furnizor, contact\_furnizor). Aducerea unei tabele în FN1 presupune înlocuirea tuturor atributelor sale ce au valori divizabile spre a obține o tabelă compusă exclusiv din atribute atomice. Așadar, observăm faptul că tabela noastră STOC nu se află în FN1, iar pentru a îndrepta acest lucru împărțim atributul *contact\_furnizor* în două atribute diferite și indivizabile: *numar\_de\_telefon\_furnizor* și *email\_furnizor*.

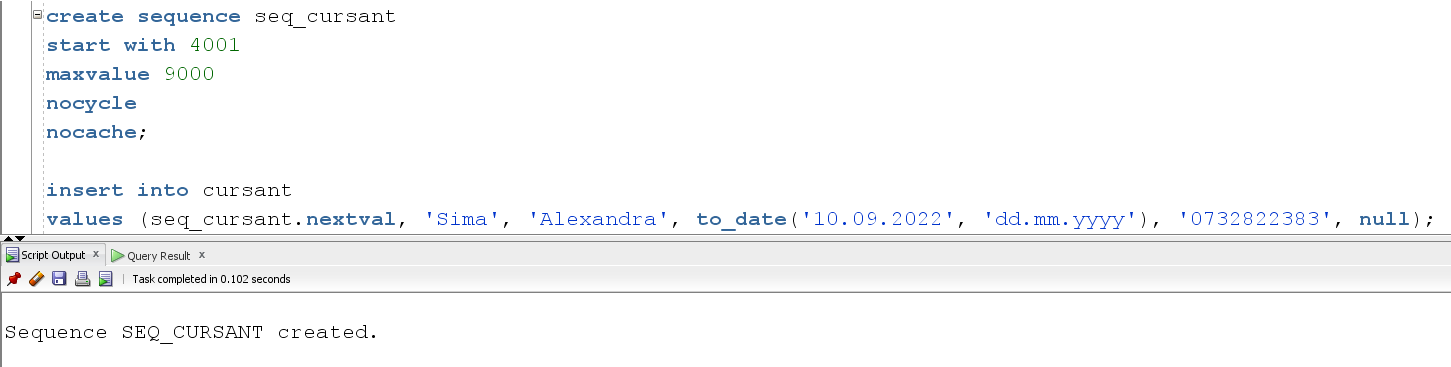
**FN2** – O tabelă în FN2 este în FN1 și fiecare atribut al său care nu face parte din cheie este dependent de cheia primară în întregimea ei. Tabela STOC nu este în FN2 deoarece observăm că atributele *tip\_pro*dus, *nume\_furnizor, numar\_de\_telefon\_furnizor* și *email\_furnizor* depind doar de *id\_tip*. Astfel, pentru aducerea tabelei la FN2, alegem să introducem o nouă tabelă, TIPURI\_PRODUSE(id\_tip#, tip\_produs, nume\_furnizor, numar\_de\_telefon, email\_furnizor), în timp ce tabelei STOC îi revine următoarea schemă relațională: STOC(id\_produs#, id\_tip, nume\_produs, pret).

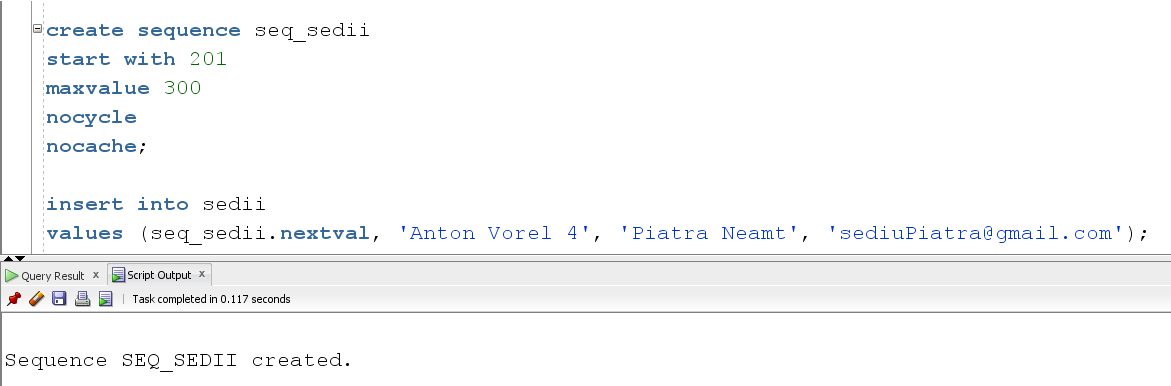
**FN3** – Analizând mai atent tabela TIPURI\_PRODUSE, observăm că aceasta nu se află în FN3. Pentru ca o tabelă să fie în FN3, aceasta trebuie să fie în FN2, iar fiecare atribut al ei care nu face parte din cheia primară trebuie să depindă direct de aceasta din urmă. Cu toate acestea, sesizăm faptul că *nume\_furnizor*, *numar\_de\_telefon* și *email\_furnizor* depind în aceeași măsură de *id\_tip* și de *tip\_produs*. Pentru a rezolva dependența tranzitivă depistată construim o nouă tabelă, FURNIZORI(id\_furnizor#, nume\_furnizor, numar\_de\_telefon, email).

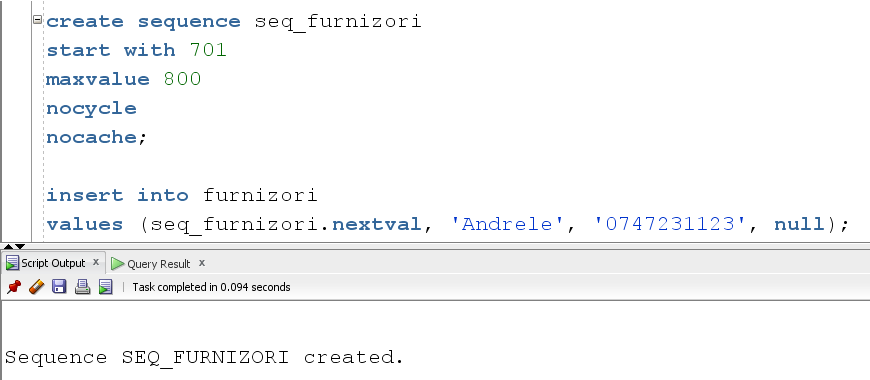
## Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele

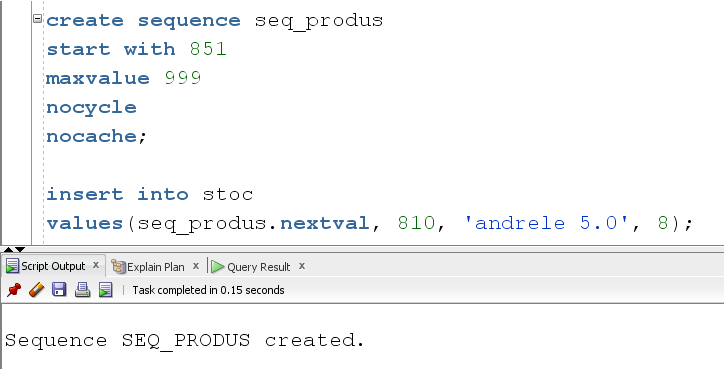
Pentru inserarea datelor corespunzătoare în tabelele bazei mele de date, am ales să creez mai multe secvențe:

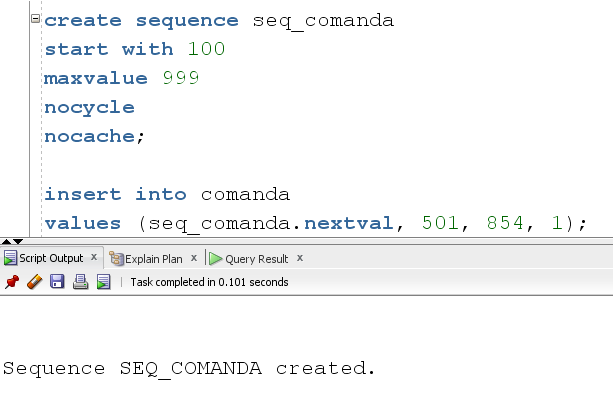


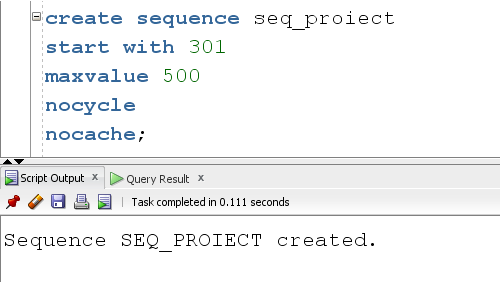
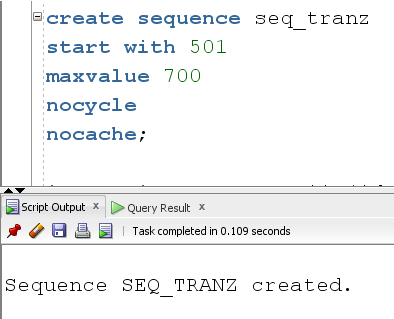


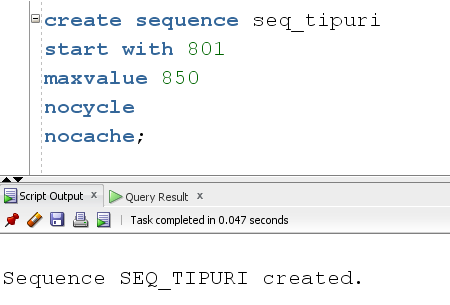
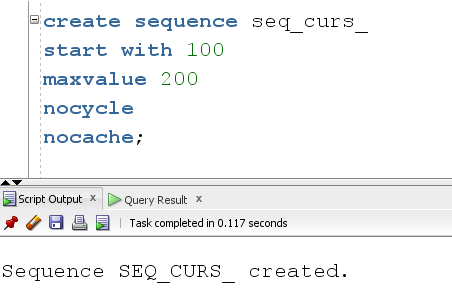








**COD:**

create sequence seq\_ang

start with 1000

increment by 1

maxvalue 4000

nocycle

nocache;

create sequence seq\_curs\_

start with 100

maxvalue 200

nocycle

nocache;

create sequence seq\_sedii

start with 201

maxvalue 300

nocycle

nocache;

create sequence seq\_proiect

start with 301

maxvalue 500

nocycle

nocache;

create sequence seq\_cursant

start with 4001

maxvalue 9000

nocycle

nocache;

create sequence seq\_tranz

start with 501

maxvalue 700

nocycle

nocache;

create sequence seq\_furnizori

start with 701

maxvalue 800

nocycle

nocache;

create sequence seq\_tipuri

start with 801

maxvalue 850

nocycle

nocache;

create sequence seq\_produs

start with 851

maxvalue 999

nocycle

nocache;

create sequence seq\_comanda

start with 100

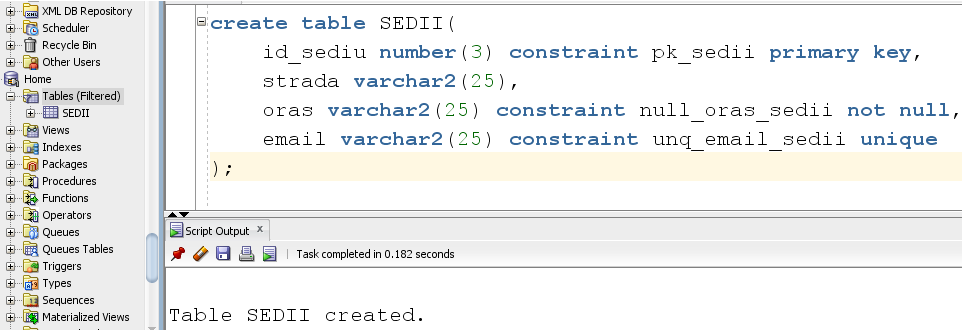
maxvalue 999

nocycle

nocache;

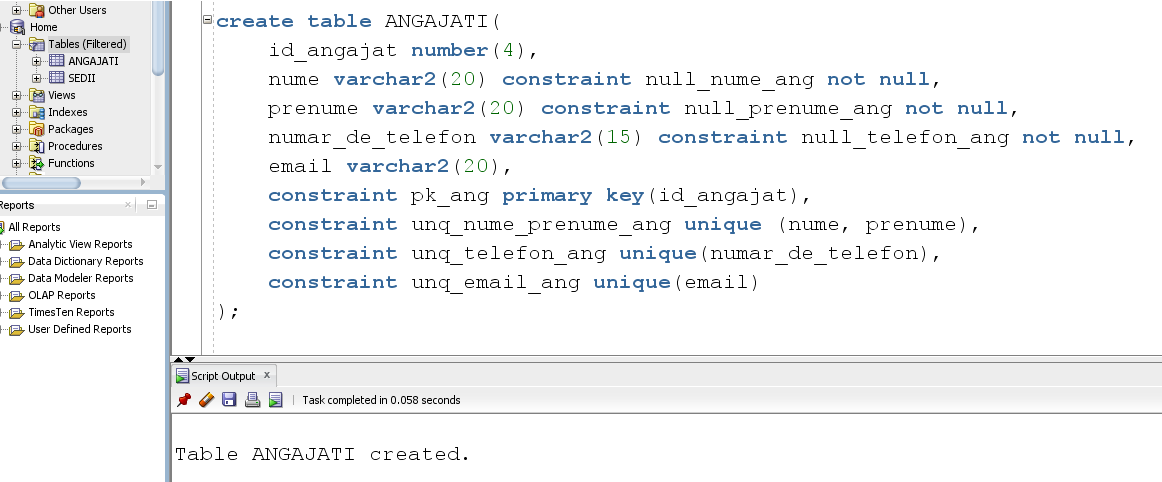
## Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative)

**SEDII**



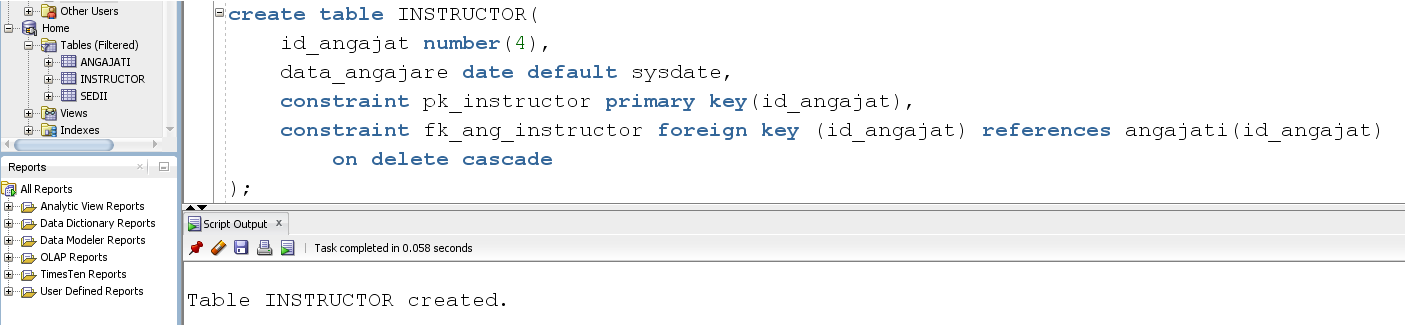


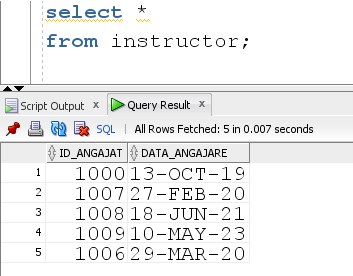
**ANGAJATI**



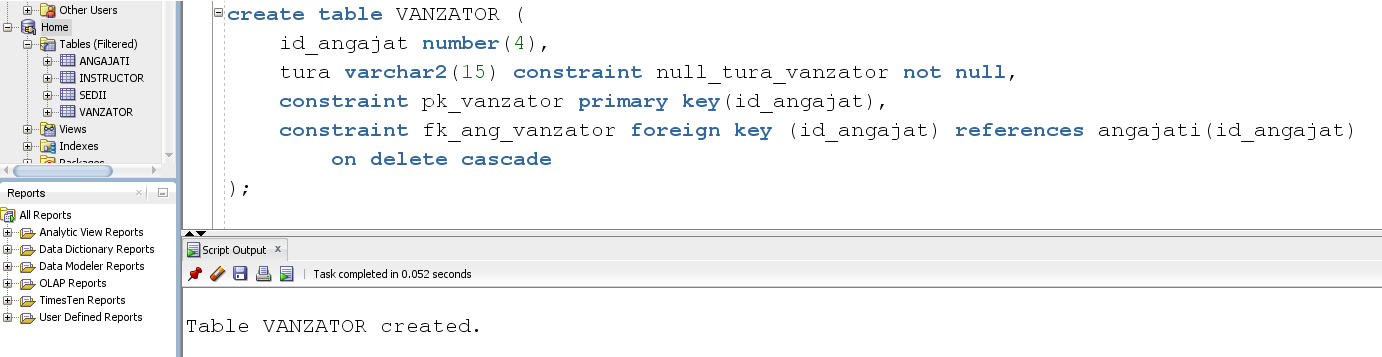


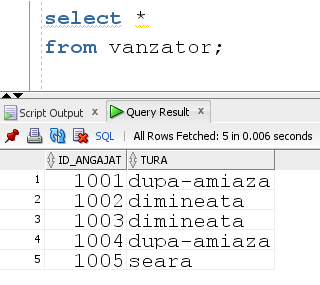
**INSTRUCTOR**



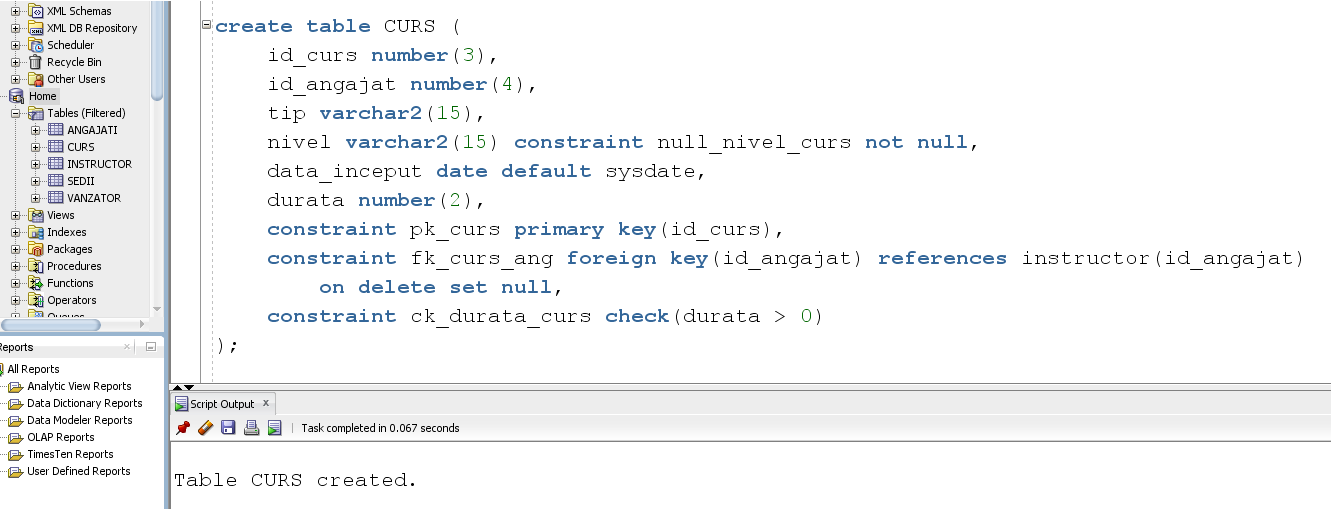


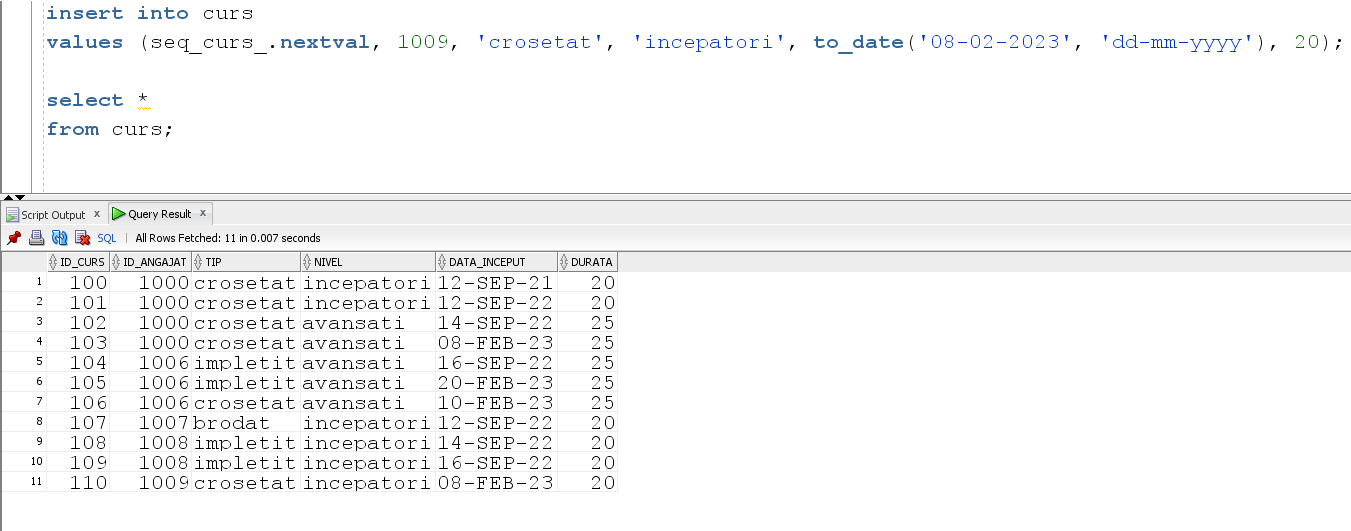
**VANZATOR**



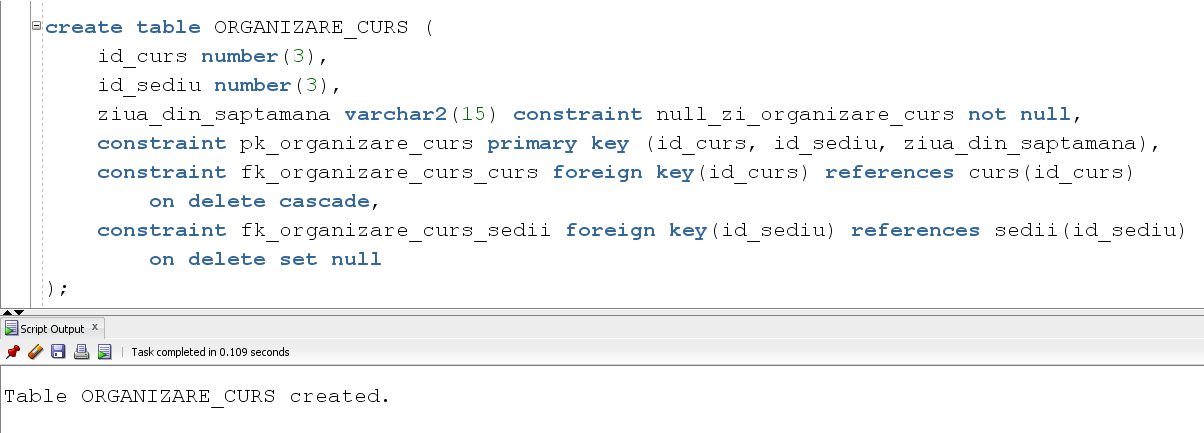


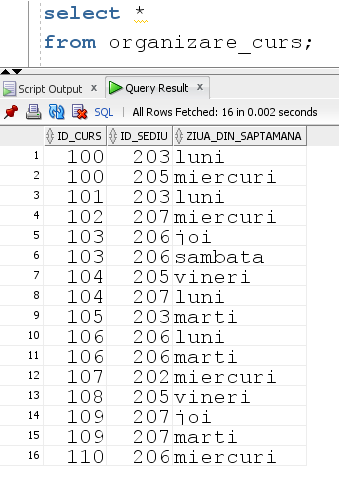
**CURS**



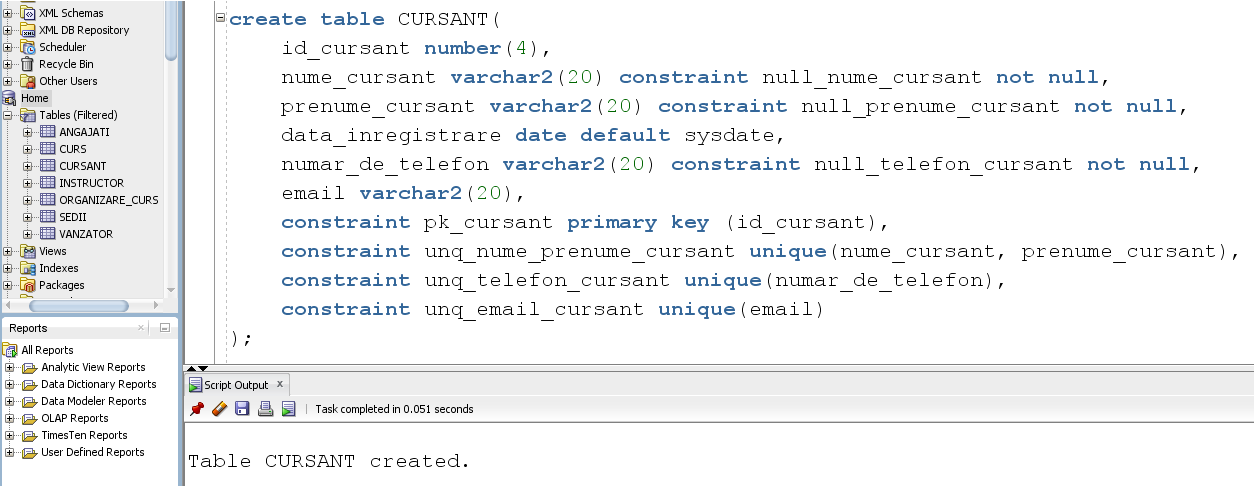


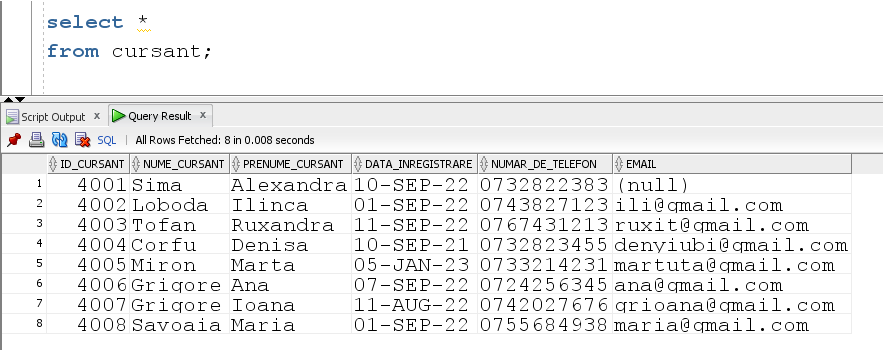
**ORGANIZARE\_CURS**



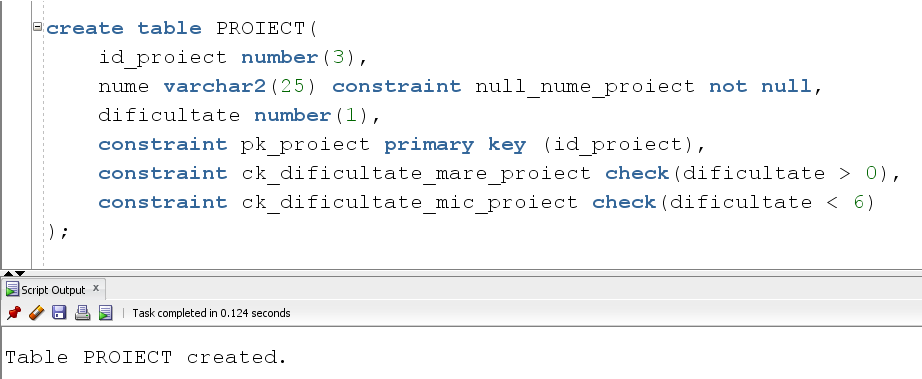


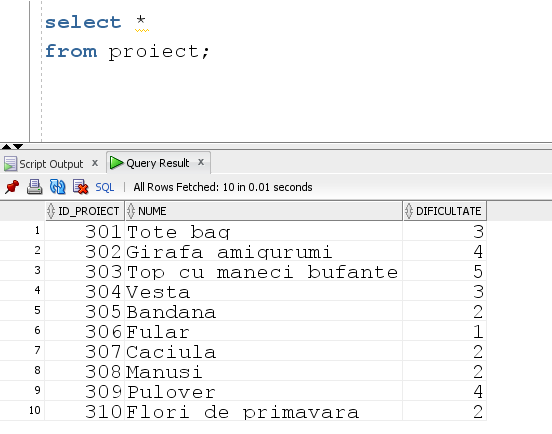
**CURSANT**



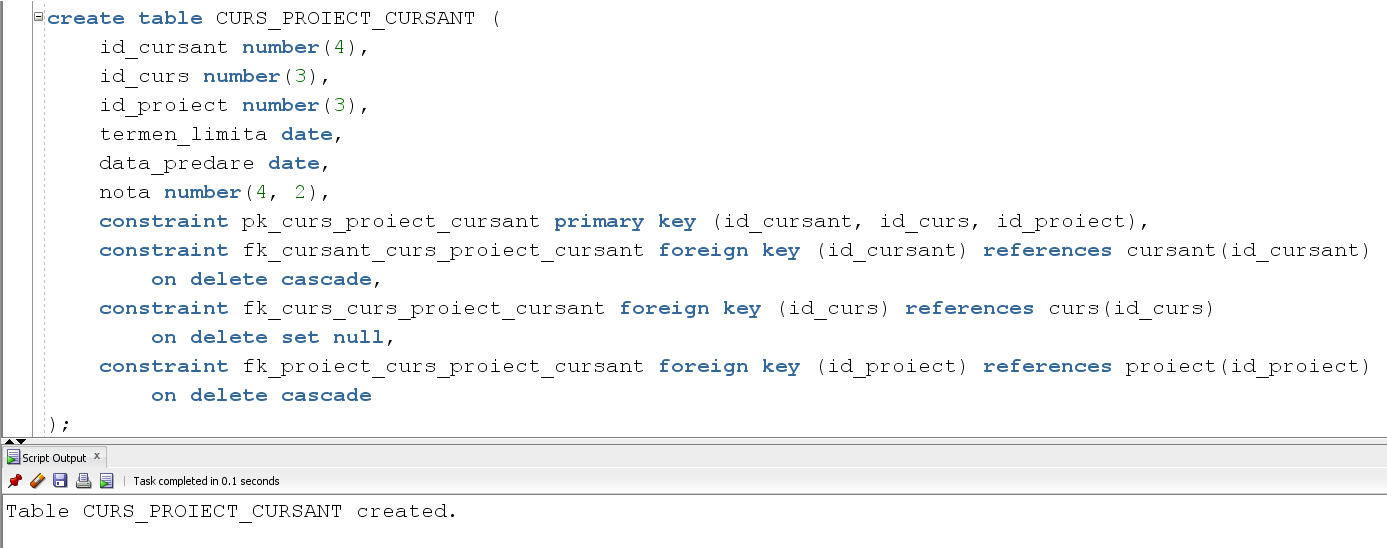


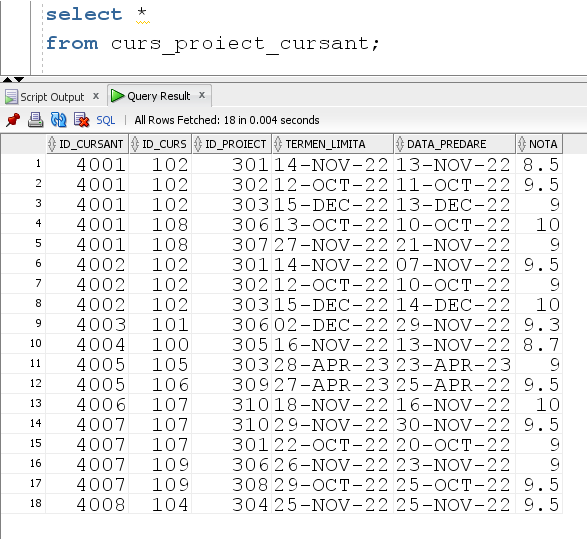
**PROIECT**



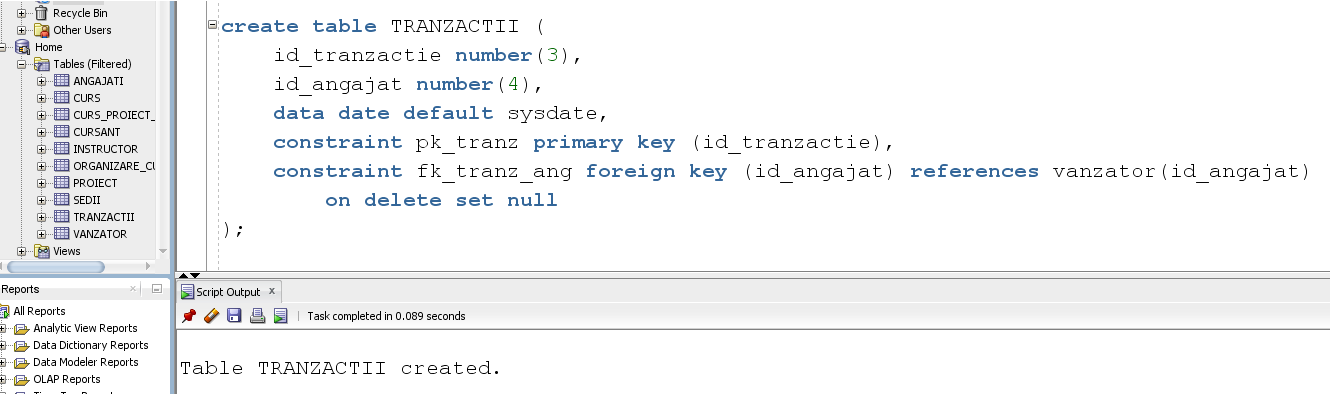


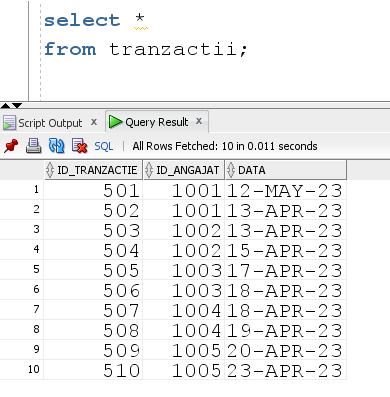
**CURS\_PROIECT\_CURSANT**



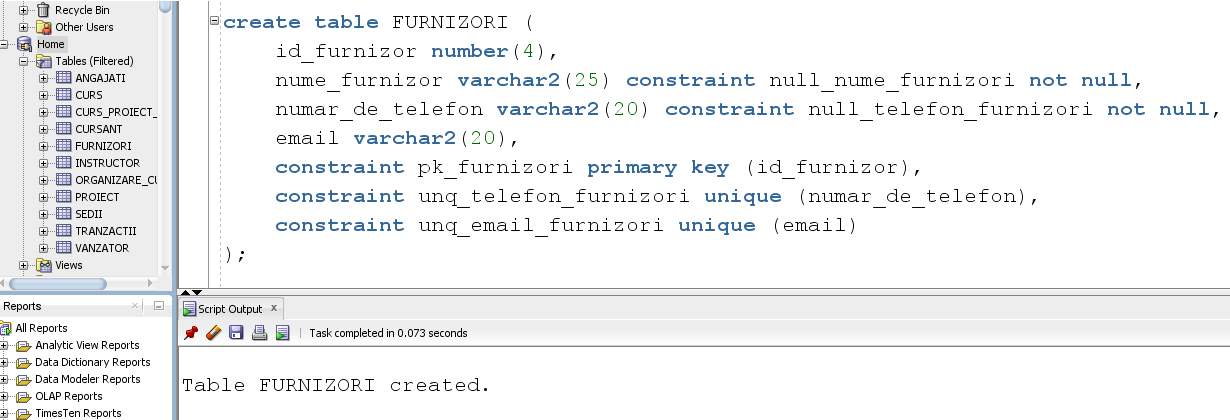


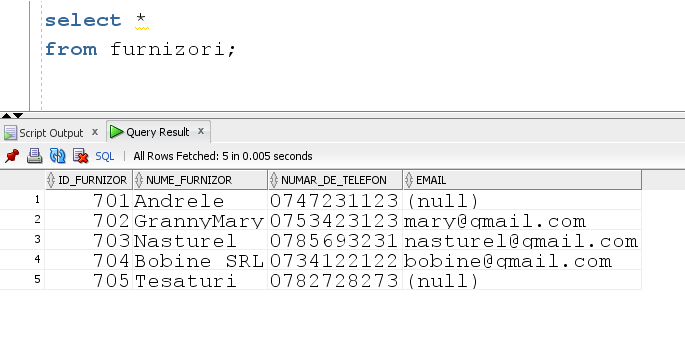
**TRANZACTII**

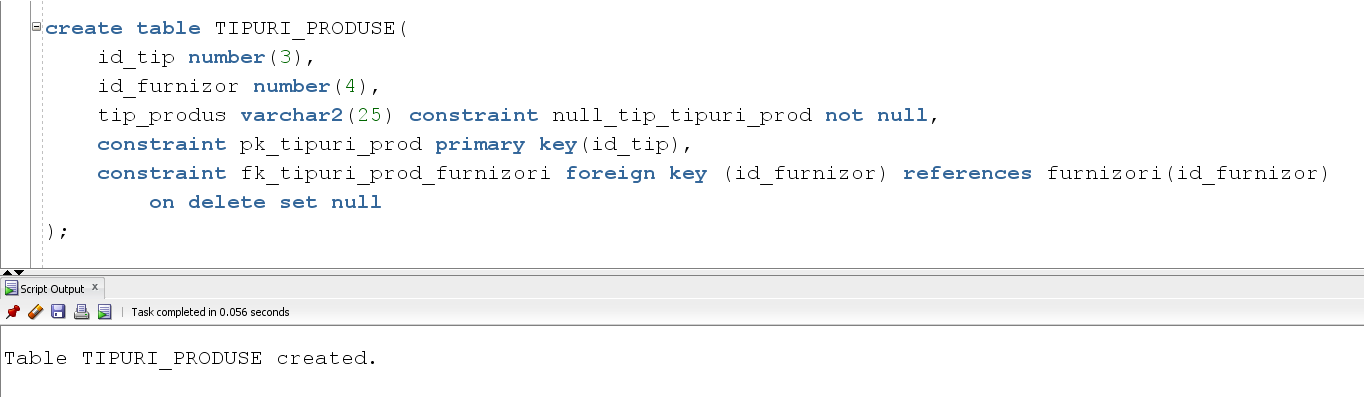


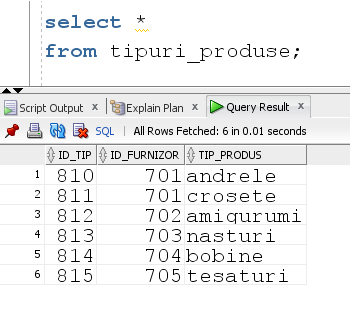


**FURNIZORI**

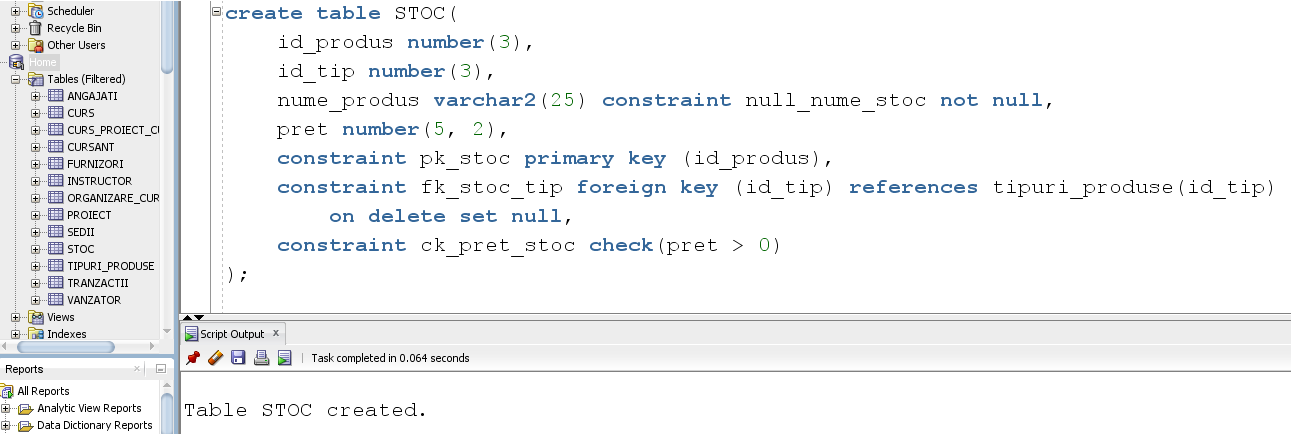


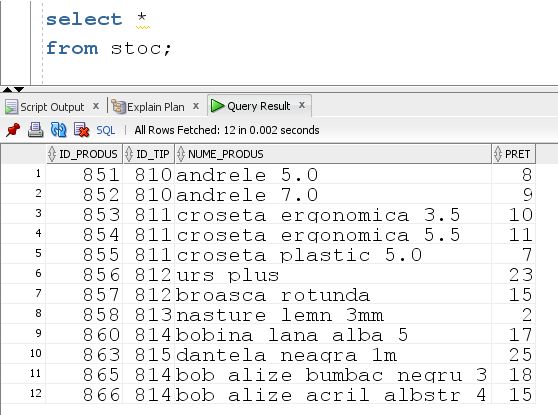


**TIPURI\_PRODUSE**

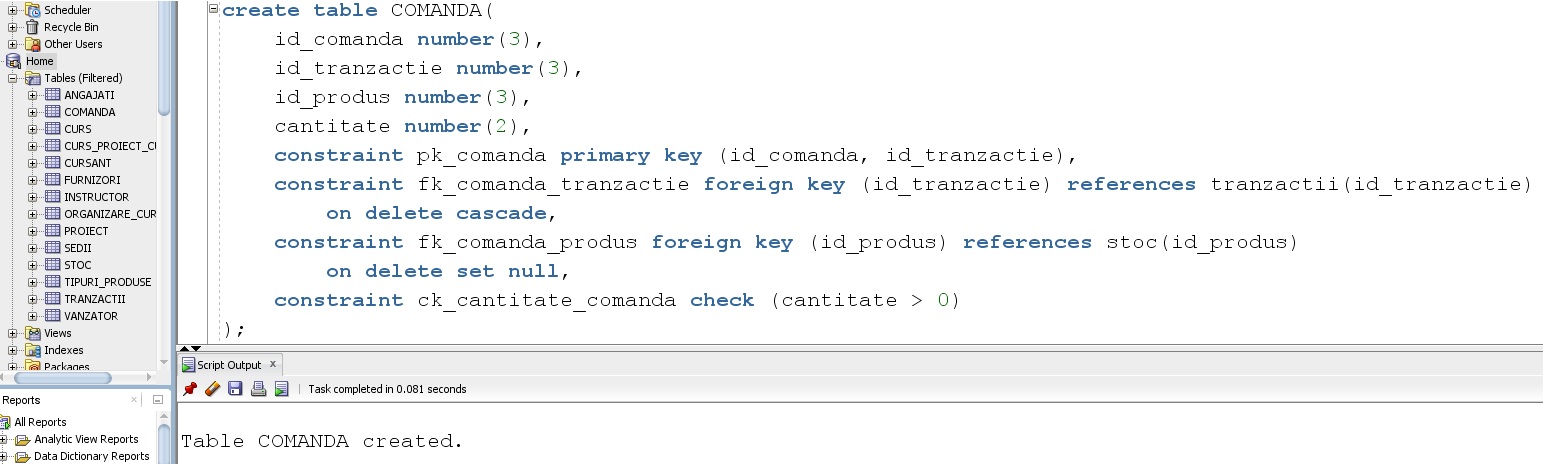


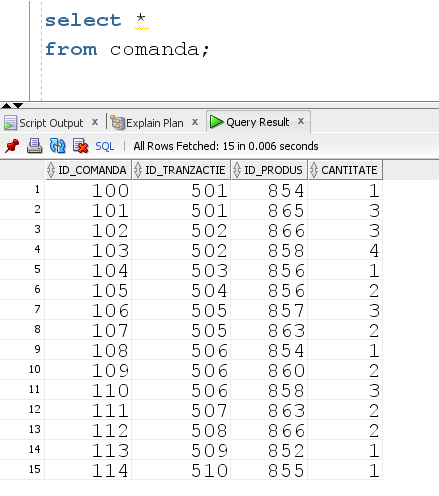
**STOC**





**COMANDA**





**COD:**

create table SEDII(

id\_sediu number(3) constraint pk\_sedii primary key,

strada varchar2(25),

oras varchar2(25) constraint null\_oras\_sedii not null,

email varchar2(25) constraint unq\_email\_sedii unique

);

create table ANGAJATI(

id\_angajat number(4),

nume varchar2(20) constraint null\_nume\_ang not null,

prenume varchar2(20) constraint null\_prenume\_ang not null,

numar\_de\_telefon varchar2(15) constraint null\_telefon\_ang not null,

email varchar2(20),

constraint pk\_ang primary key(id\_angajat),

constraint unq\_nume\_prenume\_ang unique (nume, prenume),

constraint unq\_telefon\_ang unique(numar\_de\_telefon),

constraint unq\_email\_ang unique(email)

);

create table INSTRUCTOR(

id\_angajat number(4),

data\_angajare date default sysdate,

constraint pk\_instructor primary key(id\_angajat),

constraint fk\_ang\_instructor foreign key (id\_angajat) references angajati(id\_angajat)

on delete cascade

);

create table VANZATOR (

id\_angajat number(4),

tura varchar2(15) constraint null\_tura\_vanzator not null,

constraint pk\_vanzator primary key(id\_angajat),

constraint fk\_ang\_vanzator foreign key (id\_angajat) references angajati(id\_angajat)

on delete cascade

);

create table CURS (

id\_curs number(3),

id\_angajat number(4),

tip varchar2(15),

nivel varchar2(15) constraint null\_nivel\_curs not null,

data\_inceput date default sysdate,

durata number(2),

constraint pk\_curs primary key(id\_curs),

constraint fk\_curs\_ang foreign key(id\_angajat) references instructor(id\_angajat)

on delete set null,

constraint ck\_durata\_curs check(durata > 0)

);

create table ORGANIZARE\_CURS (

id\_curs number(3),

id\_sediu number(3),

ziua\_din\_saptamana varchar2(15) constraint null\_zi\_organizare\_curs not null,

constraint pk\_organizare\_curs primary key (id\_curs, id\_sediu, ziua\_din\_saptamana),

constraint fk\_organizare\_curs\_curs foreign key(id\_curs) references curs(id\_curs)

on delete cascade,

constraint fk\_organizare\_curs\_sedii foreign key(id\_sediu) references sedii(id\_sediu)

on delete set null

);

create table CURSANT(

id\_cursant number(4),

nume\_cursant varchar2(20) constraint null\_nume\_cursant not null,

prenume\_cursant varchar2(20) constraint null\_prenume\_cursant not null,

data\_inregistrare date default sysdate,

numar\_de\_telefon varchar2(20) constraint null\_telefon\_cursant not null,

email varchar2(20),

constraint pk\_cursant primary key (id\_cursant),

constraint unq\_nume\_prenume\_cursant unique(nume\_cursant, prenume\_cursant),

constraint unq\_telefon\_cursant unique(numar\_de\_telefon),

constraint unq\_email\_cursant unique(email)

);

create table PROIECT(

id\_proiect number(3),

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

dificultate number(1),

constraint pk\_proiect primary key (id\_proiect),

constraint ck\_dificultate\_mare\_proiect check(dificultate > 0),

constraint ck\_dificultate\_mic\_proiect check(dificultate < 6)

);

create table CURS\_PROIECT\_CURSANT (

id\_cursant number(4),

id\_curs number(3),

id\_proiect number(3),

termen\_limita date,

data\_predare date,

nota number(4, 2),

constraint pk\_curs\_proiect\_cursant primary key (id\_cursant, id\_curs, id\_proiect),

constraint fk\_cursant\_curs\_proiect\_cursant foreign key (id\_cursant) references cursant(id\_cursant)

on delete cascade,

constraint fk\_curs\_curs\_proiect\_cursant foreign key (id\_curs) references curs(id\_curs)

on delete set null,

constraint fk\_proiect\_curs\_proiect\_cursant foreign key (id\_proiect) references proiect(id\_proiect)

on delete cascade

);

create table TRANZACTII (

id\_tranzactie number(3),

id\_angajat number(4),

data date default sysdate,

constraint pk\_tranz primary key (id\_tranzactie),

constraint fk\_tranz\_ang foreign key (id\_angajat) references vanzator(id\_angajat)

on delete set null

);

create table FURNIZORI (

id\_furnizor number(4),

nume\_furnizor varchar2(25) constraint null\_nume\_furnizori not null,

numar\_de\_telefon varchar2(20) constraint null\_telefon\_furnizori not null,

email varchar2(20),

constraint pk\_furnizori primary key (id\_furnizor),

constraint unq\_telefon\_furnizori unique (numar\_de\_telefon),

constraint unq\_email\_furnizori unique (email)

);

create table TIPURI\_PRODUSE(

id\_tip number(3),

id\_furnizor number(4),

tip\_produs varchar2(25) constraint null\_tip\_tipuri\_prod not null,

constraint pk\_tipuri\_prod primary key(id\_tip),

constraint fk\_tipuri\_prod\_furnizori foreign key (id\_furnizor) references furnizori(id\_furnizor)

on delete set null

);

create table STOC(

id\_produs number(3),

id\_tip number(3),

nume\_produs varchar2(25) constraint null\_nume\_stoc not null,

pret number(5, 2),

constraint pk\_stoc primary key (id\_produs),

constraint fk\_stoc\_tip foreign key (id\_tip) references tipuri\_produse(id\_tip)

on delete set null,

constraint ck\_pret\_stoc check(pret > 0)

);

create table COMANDA(

id\_comanda number(3),

id\_tranzactie number(3),

id\_produs number(3),

cantitate number(2),

constraint pk\_comanda primary key (id\_comanda, id\_tranzactie),

constraint fk\_comanda\_tranzactie foreign key (id\_tranzactie) references tranzactii(id\_tranzactie)

on delete cascade,

constraint fk\_comanda\_produs foreign key (id\_produs) references stoc(id\_produs)

on delete set null,

constraint ck\_cantitate\_comanda check (cantitate > 0)

);

insert into angajati

values (seq\_ang.nextval, 'Verdes', 'Maria', '0748239832', 'maria@gmail.com');

insert into angajati

values (seq\_ang.nextval, 'Bing', 'Chandler', '0734213422', 'friends14@gmail.com');

insert into angajati

values (seq\_ang.nextval, 'Geller', 'Monica', '0748453212', 'friends23@gmail.com');

insert into angajati

values (seq\_ang.nextval, 'Geller', 'Ross', '0748232432', 'friends24@gmail.com');

insert into angajati

values (seq\_ang.nextval, 'Green', 'Rachel', '0756234231', 'friends34@gmail.com');

insert into angajati

values (seq\_ang.nextval, 'Tribbiani', 'Joey', '0745534345', 'friends@gmail.com');

insert into angajati

values (seq\_ang.nextval, 'Buffay', 'Phoebe', '0734456456', 'friends32@gmail.com');

insert into angajati

values (seq\_ang.nextval, 'Cozma', 'Irinel', '0734452356', 'ico@gmail.com');

insert into angajati

values (seq\_ang.nextval, 'Asavinei', 'Maria', '0724321234', null);

insert into angajati

values (seq\_ang.nextval, 'Leahu', 'Silvia-Ioana', '0749229329', 'yoana@gmail.com');

select \*

from angajati;

insert into instructor

values (1000, to\_date('13-10-2019', 'dd-mm-yyyy'));

insert into instructor

values (1007, to\_date('27-02-2020', 'dd-mm-yyyy'));

insert into instructor

values (1008, to\_date('18-06-2021', 'dd-mm-yyyy'));

insert into instructor (id\_angajat)

values (1009);

insert into instructor

values (1006, to\_date('29-03-2020', 'dd-mm-yyyy'));

select \*

from instructor;

insert into vanzator

values (1001, 'dupa-amiaza');

insert into vanzator

values (1002, 'dimineata');

insert into vanzator

values (1003, 'dimineata');

insert into vanzator

values (1004, 'dupa-amiaza');

insert into vanzator

values (1005, 'seara');

select \*

from vanzator;

insert into curs

values (seq\_curs\_.nextval, 1000, 'crosetat', 'incepatori', to\_date('12-09-2021', 'dd-mm-yyyy'), 20);

insert into curs

values (seq\_curs\_.nextval, 1000, 'crosetat', 'incepatori', to\_date('12-09-2022', 'dd-mm-yyyy'), 20);

insert into curs

values (seq\_curs\_.nextval, 1000, 'crosetat', 'avansati', to\_date('14-09-2022', 'dd-mm-yyyy'), 25);

insert into curs

values (seq\_curs\_.nextval, 1000, 'crosetat', 'avansati', to\_date('08-02-2023', 'dd-mm-yyyy'), 25);

insert into curs

values (seq\_curs\_.nextval, 1006, 'impletit', 'avansati', to\_date('16-09-2022', 'dd-mm-yyyy'), 25);

insert into curs

values (seq\_curs\_.nextval, 1006, 'impletit', 'avansati', to\_date('20-02-2023', 'dd-mm-yyyy'), 25);

insert into curs

values (seq\_curs\_.nextval, 1006, 'crosetat', 'avansati', to\_date('10-02-2023', 'dd-mm-yyyy'), 25);

insert into curs

values (seq\_curs\_.nextval, 1007, 'brodat', 'incepatori', to\_date('12-09-2022', 'dd-mm-yyyy'), 20);

insert into curs

values (seq\_curs\_.nextval, 1008, 'impletit', 'incepatori', to\_date('14-09-2022', 'dd-mm-yyyy'), 20);

insert into curs

values (seq\_curs\_.nextval, 1008, 'impletit', 'incepatori', to\_date('16-09-2022', 'dd-mm-yyyy'), 20);

insert into curs

values (seq\_curs\_.nextval, 1009, 'crosetat', 'incepatori', to\_date('08-02-2023', 'dd-mm-yyyy'), 20);

select \*

from curs;

insert into sedii

values (seq\_sedii.nextval, 'Anton Vorel 4', 'Piatra Neamt', 'sediuPiatra@gmail.com');

insert into sedii

values (seq\_sedii.nextval, 'Traian Popovici 130', 'Bucuresti', 'sediuDristor@gmail.com');

insert into sedii

values (seq\_sedii.nextval, 'Constantin Nacu 9', 'Bucuresti', 'sediuUni@gmail.com');

insert into sedii

values (seq\_sedii.nextval, 'Splaiul Independentei', 'Bucuresti', 'sediuRegie@gmail.com');

insert into sedii

values (seq\_sedii.nextval, 'Pictor Arthur Verona', 'Bucuresti', 'sediuRomana@gmail.com');

select \*

from sedii;

insert into organizare\_curs

values (100, 203, 'luni');

insert into organizare\_curs

values (100, 205, 'miercuri');

insert into organizare\_curs

values (101, 203, 'luni');

insert into organizare\_curs

values (102, 207, 'miercuri');

insert into organizare\_curs

values (103, 206, 'joi');

insert into organizare\_curs

values (103, 206, 'sambata');

insert into organizare\_curs

values (104, 205, 'vineri');

insert into organizare\_curs

values (104, 207, 'luni');

insert into organizare\_curs

values (105, 203, 'marti');

insert into organizare\_curs

values (106, 206, 'luni');

insert into organizare\_curs

values (106, 206, 'marti');

insert into organizare\_curs

values (107, 202, 'miercuri');

insert into organizare\_curs

values (108, 205, 'vineri');

insert into organizare\_curs

values (109, 207, 'joi');

insert into organizare\_curs

values (109, 207, 'marti');

insert into organizare\_curs

values (110, 206, 'miercuri');

select \*

from organizare\_curs;

insert into proiect

values (seq\_proiect.nextval, 'Tote bag', 3);

insert into proiect

values(seq\_proiect.nextval, 'Girafa amigurumi', 4);

insert into proiect

values(seq\_proiect.nextval, 'Top cu maneci bufante', 5);

insert into proiect

values(seq\_proiect.nextval, 'Vesta', 3);

insert into proiect

values(seq\_proiect.nextval, 'Bandana', 2);

insert into proiect

values(seq\_proiect.nextval, 'Fular', 1);

insert into proiect

values(seq\_proiect.nextval, 'Caciula', 2);

insert into proiect

values(seq\_proiect.nextval, 'Manusi', 2);

insert into proiect

values(seq\_proiect.nextval, 'Pulover', 4);

insert into proiect

values(seq\_proiect.nextval, 'Flori de primavara', 2);

select \*

from proiect;

insert into cursant

values (seq\_cursant.nextval, 'Sima', 'Alexandra', to\_date('10.09.2022', 'dd.mm.yyyy'), '0732822383', null);

insert into cursant

values (seq\_cursant.nextval, 'Loboda', 'Ilinca', to\_date('01.09.2022', 'dd.mm.yyyy'), '0743827123', 'ili@gmail.com');

insert into cursant

values (seq\_cursant.nextval, 'Tofan', 'Ruxandra', to\_date('11.09.2022', 'dd.mm.yyyy'), '0767431213', 'ruxit@gmail.com');

insert into cursant

values (seq\_cursant.nextval, 'Corfu', 'Denisa', to\_date('10.09.2021', 'dd.mm.yyyy'), '0732823455', 'denyiubi@gmail.com');

insert into cursant

values (seq\_cursant.nextval, 'Miron', 'Marta', to\_date('05.01.2023', 'dd.mm.yyyy'), '0733214231', 'martuta@gmail.com');

insert into cursant

values (seq\_cursant.nextval, 'Grigore', 'Ana', to\_date('07.09.2022', 'dd.mm.yyyy'), '0724256345', 'ana@gmail.com');

insert into cursant

values (seq\_cursant.nextval, 'Grigore', 'Ioana', to\_date('11.08.2022', 'dd.mm.yyyy'), '0742027676', 'grioana@gmail.com');

insert into cursant

values (seq\_cursant.nextval, 'Savoaia', 'Maria', to\_date('01.09.2022', 'dd.mm.yyyy'), '0755684938', 'maria@gmail.com');

select \*

from cursant;

insert into curs\_proiect\_cursant

values(4001, 102, 301, to\_date('14.11.2022', 'dd.mm.yyyy'), to\_date('13.11.2022', 'dd.mm.yyyy'), 8.5);

insert into curs\_proiect\_cursant

values(4001, 102, 302, to\_date('12.10.2022', 'dd.mm.yyyy'), to\_date('11.10.2022', 'dd.mm.yyyy'), 9.5);

insert into curs\_proiect\_cursant

values(4001, 102, 303, to\_date('15.12.2022', 'dd.mm.yyyy'), to\_date('13.12.2022', 'dd.mm.yyyy'), 9);

insert into curs\_proiect\_cursant

values(4001, 108, 306, to\_date('13.10.2022', 'dd.mm.yyyy'), to\_date('10.10.2022', 'dd.mm.yyyy'), 10);

insert into curs\_proiect\_cursant

values(4001, 108, 307, to\_date('27.11.2022', 'dd.mm.yyyy'), to\_date('21.11.2022', 'dd.mm.yyyy'), 9);

insert into curs\_proiect\_cursant

values(4002, 102, 301, to\_date('14.11.2022', 'dd.mm.yyyy'), to\_date('07.11.2022', 'dd.mm.yyyy'), 9.5);

insert into curs\_proiect\_cursant

values(4002, 102, 302, to\_date('12.10.2022', 'dd.mm.yyyy'), to\_date('10.10.2022', 'dd.mm.yyyy'), 9);

insert into curs\_proiect\_cursant

values(4002, 102, 303, to\_date('15.12.2022', 'dd.mm.yyyy'), to\_date('14.12.2022', 'dd.mm.yyyy'), 10);

insert into curs\_proiect\_cursant

values(4003, 101, 306, to\_date('02.12.2022', 'dd.mm.yyyy'), to\_date('29.11.2022', 'dd.mm.yyyy'), 9.3);

insert into curs\_proiect\_cursant

values(4004, 100, 305, to\_date('16.11.2022', 'dd.mm.yyyy'), to\_date('13.11.2022', 'dd.mm.yyyy'), 8.7);

insert into curs\_proiect\_cursant

values(4005, 105, 303, to\_date('28.04.2023', 'dd.mm.yyyy'), to\_date('23.04.2023', 'dd.mm.yyyy'), 9);

insert into curs\_proiect\_cursant

values(4005, 106, 309, to\_date('27.04.2023', 'dd.mm.yyyy'), to\_date('25.04.2022', 'dd.mm.yyyy'), 9.5);

insert into curs\_proiect\_cursant

values(4006, 107, 310, to\_date('18.11.2022', 'dd.mm.yyyy'), to\_date('16.11.2022', 'dd.mm.yyyy'), 10);

insert into curs\_proiect\_cursant

values(4007, 107, 310, to\_date('29.11.2022', 'dd.mm.yyyy'), to\_date('30.11.2022', 'dd.mm.yyyy'), 9.5);

insert into curs\_proiect\_cursant

values(4007, 107, 301, to\_date('22.10.2022', 'dd.mm.yyyy'), to\_date('20.10.2022', 'dd.mm.yyyy'), 9);

insert into curs\_proiect\_cursant

values(4007, 109, 306, to\_date('26.11.2022', 'dd.mm.yyyy'), to\_date('23.11.2022', 'dd.mm.yyyy'), 9);

insert into curs\_proiect\_cursant

values(4007, 109, 308, to\_date('29.10.2022', 'dd.mm.yyyy'), to\_date('25.10.2022', 'dd.mm.yyyy'), 9.5);

insert into curs\_proiect\_cursant

values(4008, 104, 304, to\_date('25.11.2022', 'dd.mm.yyyy'), to\_date('25.11.2022', 'dd.mm.yyyy'), 9.5);

select \*

from curs\_proiect\_cursant;

insert into tranzactii (id\_tranzactie, id\_angajat)

values (seq\_tranz.nextval, 1001);

insert into tranzactii

values (seq\_tranz.nextval, 1001, to\_date('13.04.2023', 'dd.mm.yyyy'));

insert into tranzactii

values (seq\_tranz.nextval, 1002, to\_date('13.04.2023', 'dd.mm.yyyy'));

insert into tranzactii

values (seq\_tranz.nextval, 1002, to\_date('15.04.2023', 'dd.mm.yyyy'));

insert into tranzactii

values (seq\_tranz.nextval, 1003, to\_date('17.04.2023', 'dd.mm.yyyy'));

insert into tranzactii

values (seq\_tranz.nextval, 1003, to\_date('18.04.2023', 'dd.mm.yyyy'));

insert into tranzactii

values (seq\_tranz.nextval, 1004, to\_date('18.04.2023', 'dd.mm.yyyy'));

insert into tranzactii

values (seq\_tranz.nextval, 1004, to\_date('19.04.2023', 'dd.mm.yyyy'));

insert into tranzactii

values (seq\_tranz.nextval, 1005, to\_date('20.04.2023', 'dd.mm.yyyy'));

insert into tranzactii

values (seq\_tranz.nextval, 1005, to\_date('23.04.2023', 'dd.mm.yyyy'));

select \*

from tranzactii;

insert into furnizori

values (seq\_furnizori.nextval, 'Andrele', '0747231123', null);

insert into furnizori

values (seq\_furnizori.nextval, 'GrannyMary', '0753423123', 'mary@gmail.com');

insert into furnizori

values (seq\_furnizori.nextval, 'Nasturel', '0785693231', 'nasturel@gmail.com');

insert into furnizori

values (seq\_furnizori.nextval, 'Bobine SRL', '0734122122', 'bobine@gmail.com');

insert into furnizori

values (seq\_furnizori.nextval, 'Tesaturi', '0782728273', null);

select \*

from furnizori;

insert into tipuri\_produse

values (seq\_tipuri.nextval, 701, 'andrele');

insert into tipuri\_produse

values (seq\_tipuri.nextval, 701, 'crosete');

insert into tipuri\_produse

values (seq\_tipuri.nextval, 702, 'amigurumi');

insert into tipuri\_produse

values (seq\_tipuri.nextval, 703, 'nasturi');

insert into tipuri\_produse

values (seq\_tipuri.nextval, 704, 'bobine');

insert into tipuri\_produse

values (seq\_tipuri.nextval, 705, 'tesaturi');

select \*

from tipuri\_produse;

insert into stoc

values(seq\_produs.nextval, 810, 'andrele 5.0', 8);

insert into stoc

values(seq\_produs.nextval, 810, 'andrele 7.0', 9);

insert into stoc

values(seq\_produs.nextval, 811, 'croseta ergonomica 3.5', 10);

insert into stoc

values(seq\_produs.nextval, 811, 'croseta ergonomica 5.5', 11);

insert into stoc

values(seq\_produs.nextval, 811, 'croseta plastic 5.0', 7);

insert into stoc

values(seq\_produs.nextval, 812, 'urs plus', 23);

insert into stoc

values(seq\_produs.nextval, 812, 'broasca rotunda', 15);

insert into stoc

values(seq\_produs.nextval, 813, 'nasture lemn 3mm', 2);

insert into stoc

values(seq\_produs.nextval, 813, 'nasture plastic negru 4.5mm', 1.5);

insert into stoc

values(seq\_produs.nextval, 814, 'bobina lana alba 5', 17);

insert into stoc

values(seq\_produs.nextval, 814, 'bob alize bumbac negru 3', 18);

insert into stoc

values(seq\_produs.nextval, 814, 'bob alize acril albstr 4', 15);

insert into stoc

values(seq\_produs.nextval, 815, 'dantela neagra 1m', 25);

select \*

from stoc;

insert into comanda

values (seq\_comanda.nextval, 501, 854, 1);

insert into comanda

values (seq\_comanda.nextval, 501, 865, 3);

insert into comanda

values (seq\_comanda.nextval, 502, 866, 3);

insert into comanda

values (seq\_comanda.nextval, 502, 858, 4);

insert into comanda

values (seq\_comanda.nextval, 503, 856, 1);

insert into comanda

values (seq\_comanda.nextval, 504, 856, 2);

insert into comanda

values (seq\_comanda.nextval, 505, 857, 3);

insert into comanda

values (seq\_comanda.nextval, 505, 863, 2);

insert into comanda

values (seq\_comanda.nextval, 506, 854, 1);

insert into comanda

values (seq\_comanda.nextval, 506, 860, 2);

insert into comanda

values (seq\_comanda.nextval, 506, 858, 3);

insert into comanda

values (seq\_comanda.nextval, 507, 863, 2);

insert into comanda

values (seq\_comanda.nextval, 508, 866, 2);

insert into comanda

values (seq\_comanda.nextval, 509, 852, 1);

insert into comanda

values (seq\_comanda.nextval, 510, 855, 1);

select \*

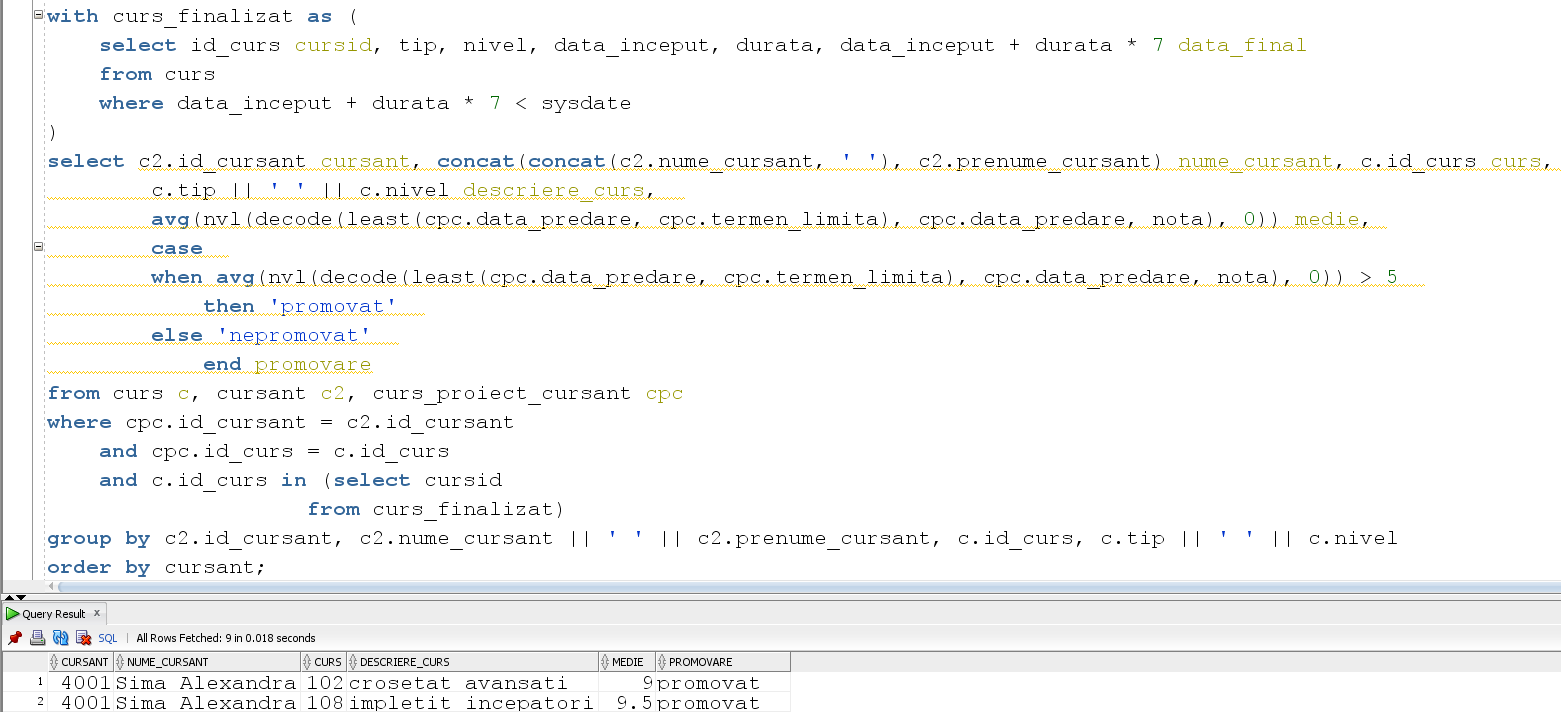
from comanda;

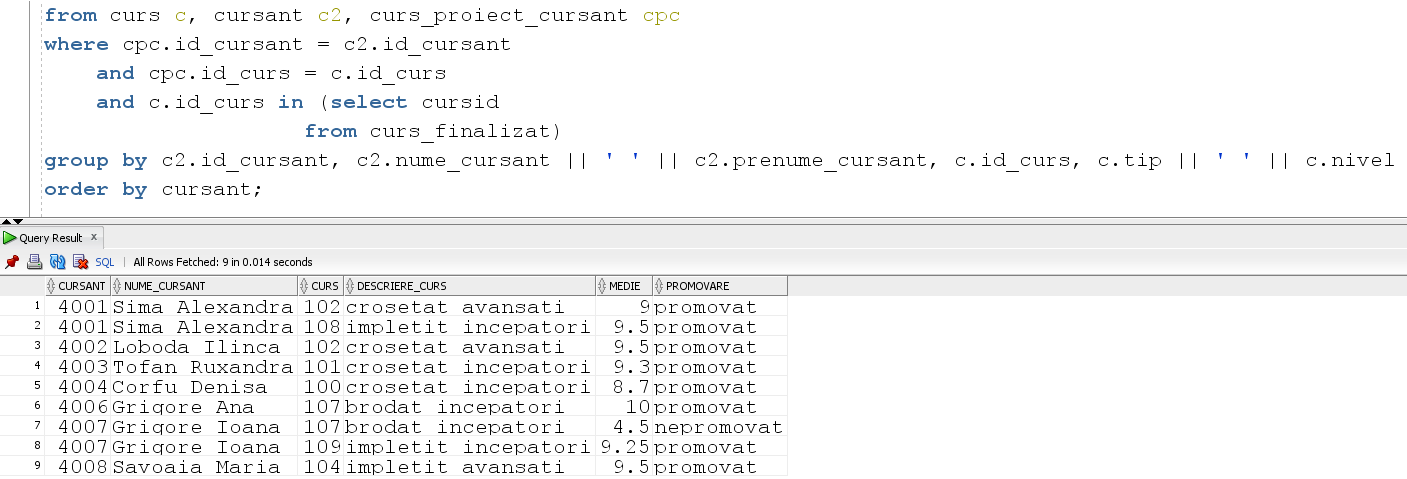
## Formulați în limbaj natural și implementați 5 cereri SQL complexe

**CEREREA 1**: Pentru fiecare cursant, să i se afișeze id-ul, numele complet și media finală obținută la fiecare curs pe care l-a absolvit, împreună cu id-ul și descrierea cursului (tipul nivelul cursului), dar și starea de promovabilitate a cursantului la cursul respectiv. Rezultatele vor fi ordonate după id-ul cursantului.

***COMPLETĂRI***: Un curs se consideră absolvit dacă acesta s-a încheiat. Un cursant promovează un curs dacă media sa este mai mare decât 5. Media nu se rotunjește și se calculează ca fiind media algebrică a notelor obținute de cursant la proiectele din cadrul cursului. Notele obținute la proiecte predate după termenul limită se vor ignora, iar proiectele vor fi considerate ca fiind punctate cu nota 0.

***FOLOSIM***: clauza WITH, CASE, ordonări si funcțiile NVL și DECODE (în cadrul aceleiași cereri), 2 funcții pe date calendaristice (*sysdate* și *least*), 1 funcție pentru șiruri de caractere (*concat*)





**COD:**

with curs\_finalizat as (

select id\_curs cursid, tip, nivel, data\_inceput, durata, data\_inceput + durata \* 7 data\_final

from curs

where data\_inceput + durata \* 7 < sysdate

)

select c2.id\_cursant cursant, concat(concat(c2.nume\_cursant, ' '), c2.prenume\_cursant) nume\_cursant, c.id\_curs curs,

c.tip || ' ' || c.nivel descriere\_curs,

avg(nvl(decode(least(cpc.data\_predare, cpc.termen\_limita), cpc.data\_predare, nota), 0)) medie,

case

when avg(nvl(decode(least(cpc.data\_predare, cpc.termen\_limita), cpc.data\_predare, nota), 0)) > 5

then 'promovat'

else 'nepromovat'

end promovare

from curs c, cursant c2, curs\_proiect\_cursant cpc

where cpc.id\_cursant = c2.id\_cursant

and cpc.id\_curs = c.id\_curs

and c.id\_curs in (select cursid

from curs\_finalizat)

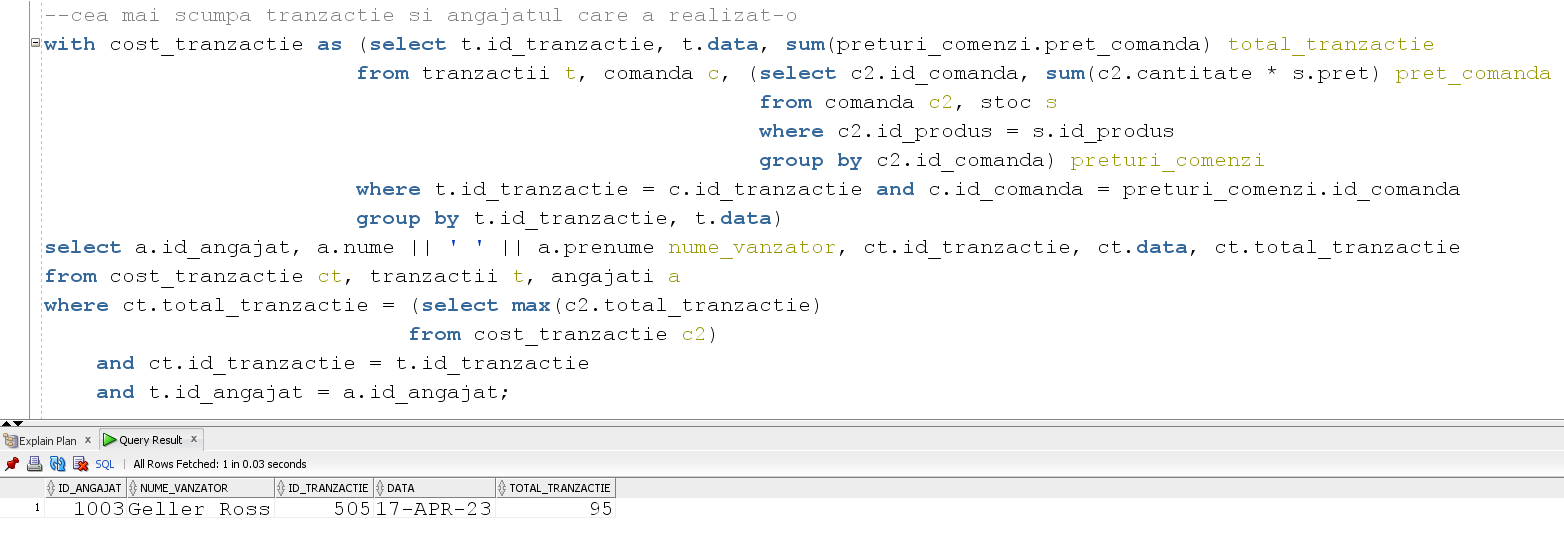
group by c2.id\_cursant, c2.nume\_cursant || ' ' || c2.prenume\_cursant, c.id\_curs, c.tip || ' ' || c.nivel

order by cursant;

**CEREREA 2**: Să se afișeze suma totală a celei mai costisitoare tranzacții, împreună cu id-ul și data la care a fost realizată, dar și id-ul și numele complet al vânzătorului care a realizat-o.

***COMPLETĂRI***: Amintim faptul că o tranzacție este compusă din mai multe comenzi către stoc. Acestea din urmă se referă la un singur produs, din care solicită o anumită cantitate.

***FOLOSIM***: clauza WITH



**COD:**

with cost\_tranzactie as (select t.id\_tranzactie, t.data, sum(preturi\_comenzi.pret\_comanda) total\_tranzactie

from tranzactii t, comanda c, (select c2.id\_comanda, sum(c2.cantitate \* s.pret) pret\_comanda

from comanda c2, stoc s

where c2.id\_produs = s.id\_produs

group by c2.id\_comanda) preturi\_comenzi

where t.id\_tranzactie = c.id\_tranzactie and c.id\_comanda = preturi\_comenzi.id\_comanda

group by t.id\_tranzactie, t.data)

select a.id\_angajat, a.nume || ' ' || a.prenume nume\_vanzator, ct.id\_tranzactie, ct.data, ct.total\_tranzactie

from cost\_tranzactie ct, tranzactii t, angajati a

where ct.total\_tranzactie = (select max(c2.total\_tranzactie)

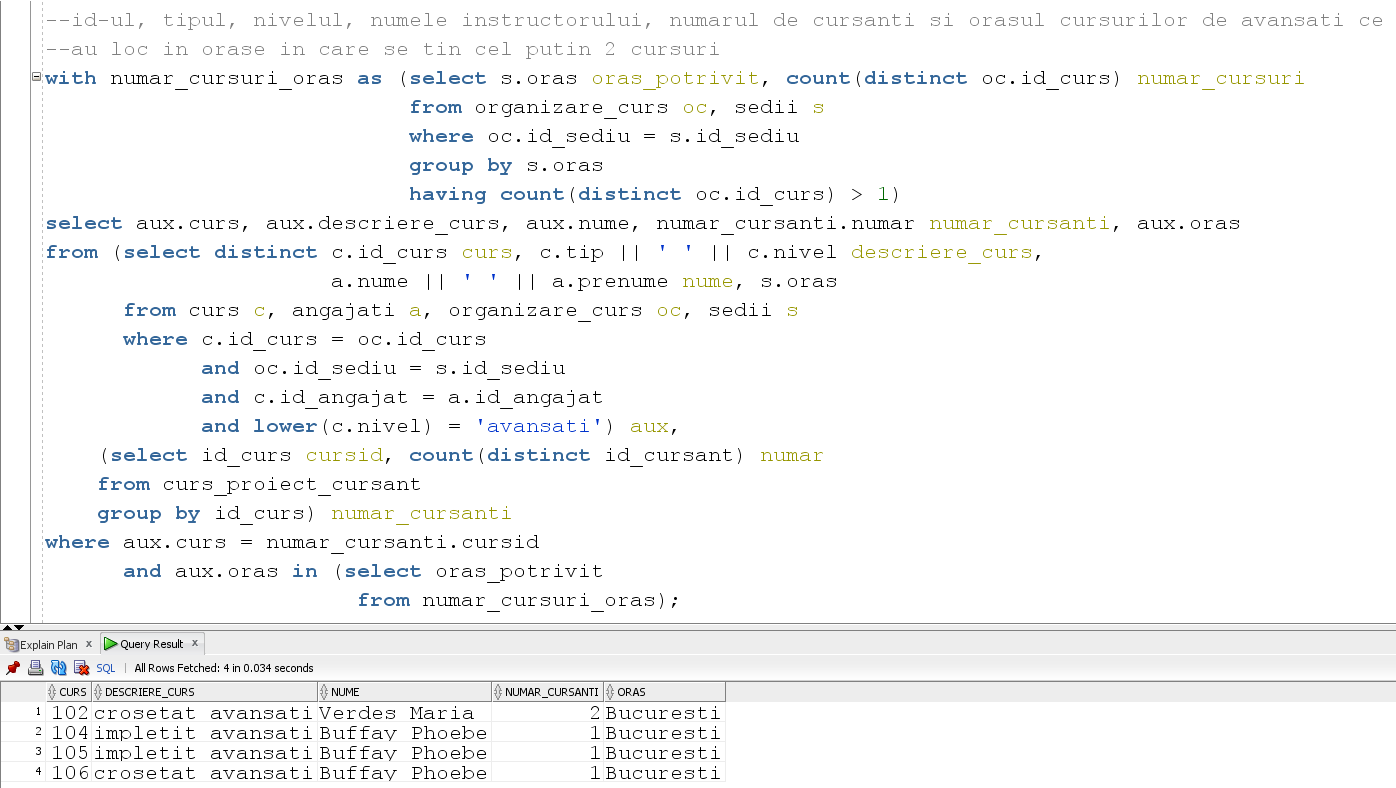
from cost\_tranzactie c2)

and ct.id\_tranzactie = t.id\_tranzactie

and t.id\_angajat = a.id\_angajat;

**CEREREA 3**: Să se afișeze id-ul, descrierea (tipul concatenat cu nivelul), numele instructorului, numărul de cursanți și orașul în care se organizează cursurile de avansați ce au loc în orașe în care se organizează cel puțin 2 cursuri.

***FOLOSIM***: subcerere nesincronizată în clauza FROM, clauza WITH, o funcție pentru șiruri de caractere (*lower*)



**COD:**

with numar\_cursuri\_oras as (select s.oras oras\_potrivit, count(distinct oc.id\_curs) numar\_cursuri

from organizare\_curs oc, sedii s

where oc.id\_sediu = s.id\_sediu

group by s.oras

having count(distinct oc.id\_curs) > 1)

select aux.curs, aux.descriere\_curs, aux.nume, numar\_cursanti.numar numar\_cursanti, aux.oras

from (select distinct c.id\_curs curs, c.tip || ' ' || c.nivel descriere\_curs,

a.nume || ' ' || a.prenume nume, s.oras

from curs c, angajati a, organizare\_curs oc, sedii s

where c.id\_curs = oc.id\_curs

and oc.id\_sediu = s.id\_sediu

and c.id\_angajat = a.id\_angajat

and lower(c.nivel) = 'avansati') aux,

(select id\_curs cursid, count(distinct id\_cursant) numar

from curs\_proiect\_cursant

group by id\_curs) numar\_cursanti

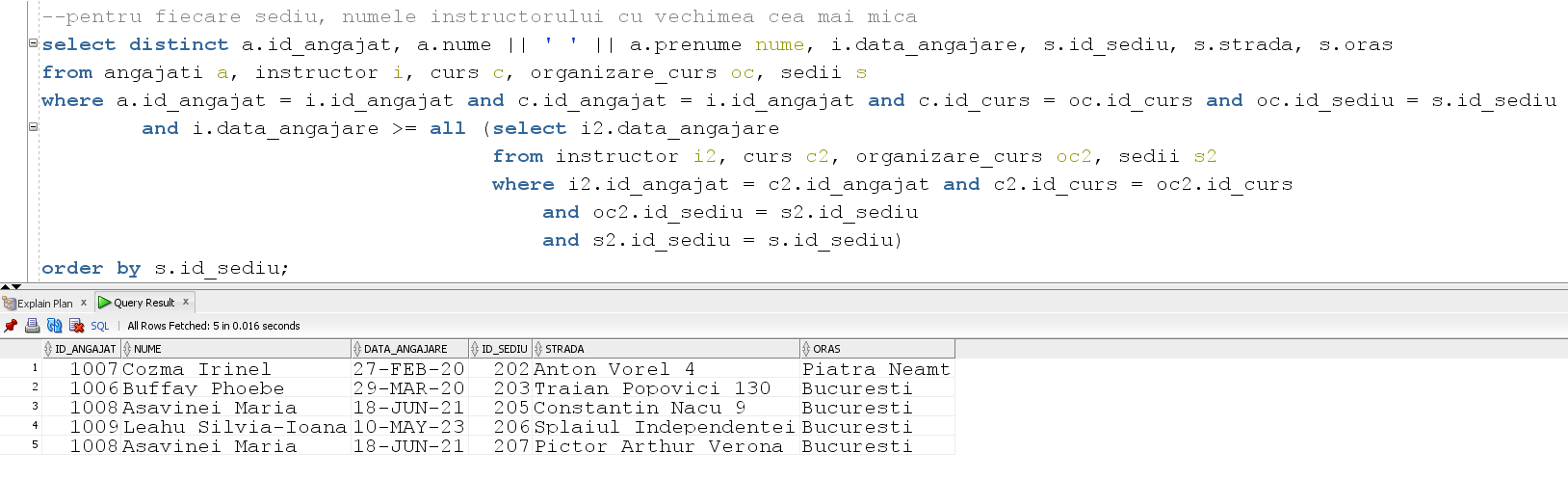
where aux.curs = numar\_cursanti.cursid

and aux.oras in (select oras\_potrivit

from numar\_cursuri\_oras);

**CEREREA 4**: Să se afișeze pentru fiecare sediu în parte id-ul, strada și orașul, dar și id-ul, numeleși data angajării instructorului cu vechimea cea mai mică.

***FOLOSIM***: subcerere sincronizată în care intervin cel puțin 3 tabele



**COD:**

select distinct a.id\_angajat, a.nume || ' ' || a.prenume nume, i.data\_angajare, s.id\_sediu, s.strada, s.oras

from angajati a, instructor i, curs c, organizare\_curs oc, sedii s

where a.id\_angajat = i.id\_angajat and c.id\_angajat = i.id\_angajat and c.id\_curs = oc.id\_curs and oc.id\_sediu = s.id\_sediu

and i.data\_angajare >= all (select i2.data\_angajare

from instructor i2, curs c2, organizare\_curs oc2, sedii s2

where i2.id\_angajat = c2.id\_angajat and c2.id\_curs = oc2.id\_curs

and oc2.id\_sediu = s2.id\_sediu

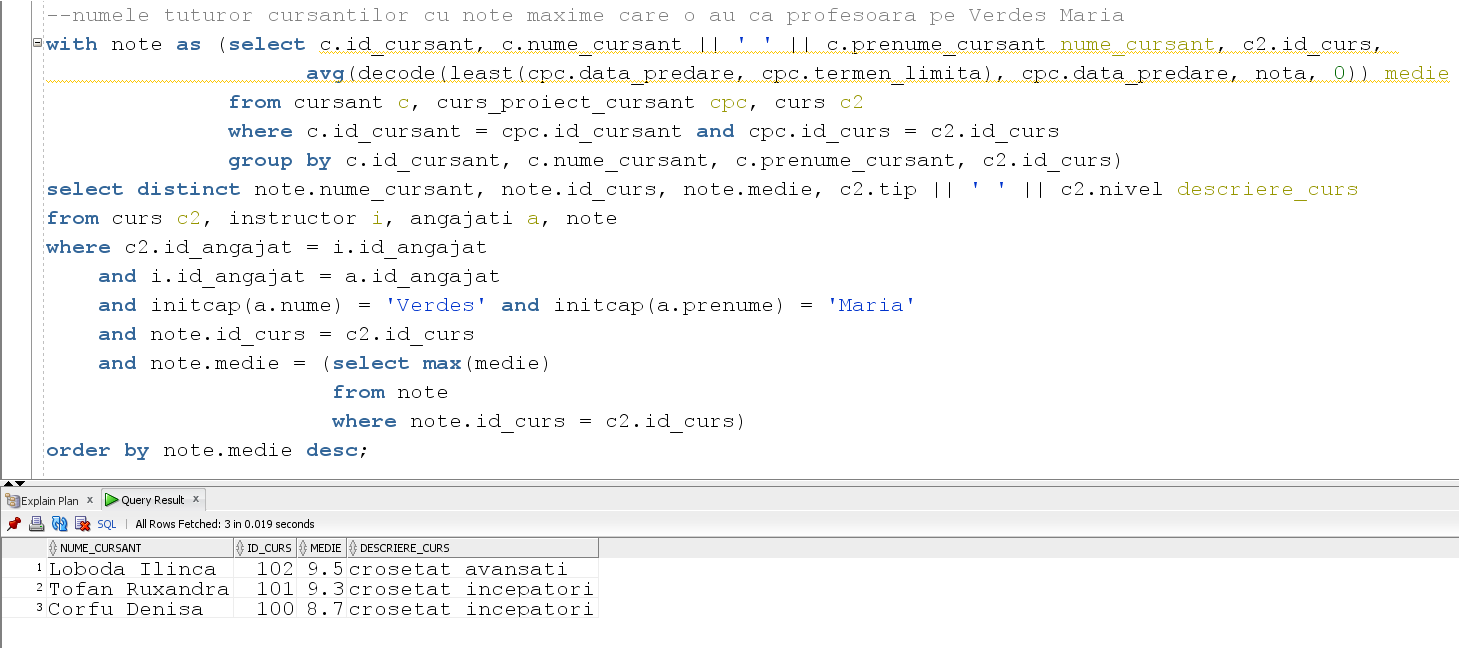
and s2.id\_sediu = s.id\_sediu)

order by s.id\_sediu;

**CEREREA 5**: Să se afișeze numele și media tuturor cursanților cu medii finale maxime la un curs predat de Verdes Maria, împreună cu id-ul și descrierea (tipul și nivelul) cursului respectiv. Rezultatele vor fi ordonate descrescător după medie.

***VARIANTA 1***

***FOLOSIM***: o funcție de șiruri de caractere (*initcap*), clauza WITH



**COD:**

with note as (select c.id\_cursant, c.nume\_cursant || ' ' || c.prenume\_cursant nume\_cursant, c2.id\_curs,

avg(decode(least(cpc.data\_predare, cpc.termen\_limita), cpc.data\_predare, nota, 0)) medie

from cursant c, curs\_proiect\_cursant cpc, curs c2

where c.id\_cursant = cpc.id\_cursant and cpc.id\_curs = c2.id\_curs

group by c.id\_cursant, c.nume\_cursant, c.prenume\_cursant, c2.id\_curs)

select distinct note.nume\_cursant, note.id\_curs, note.medie, c2.tip || ' ' || c2.nivel descriere\_curs

from curs c2, instructor i, angajati a, note

where c2.id\_angajat = i.id\_angajat

and i.id\_angajat = a.id\_angajat

and initcap(a.nume) = 'Verdes' and initcap(a.prenume) = 'Maria'

and note.id\_curs = c2.id\_curs

and note.medie = (select max(medie)

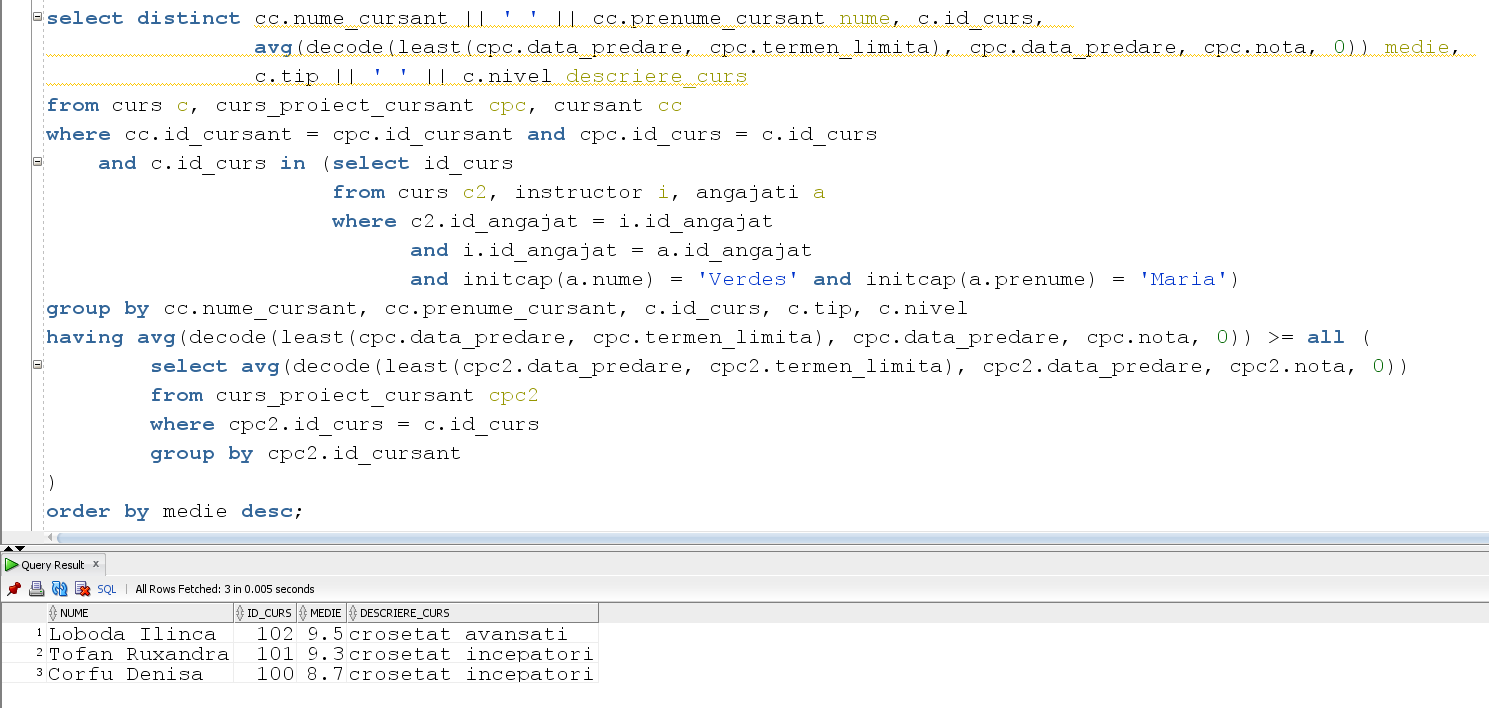
from note

where note.id\_curs = c2.id\_curs)

order by note.medie desc;

***VARIANTA 2***

***FOLOSIM:*** o funcție de șiruri de caractere (*initcap*), grupări de date cu subcerere nesincronizată în care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri (în cadrul aceleiași cereri)



**COD:**

select distinct cc.nume\_cursant || ' ' || cc.prenume\_cursant nume, c.id\_curs,

avg(decode(least(cpc.data\_predare, cpc.termen\_limita), cpc.data\_predare, cpc.nota, 0)) medie,

c.tip || ' ' || c.nivel descriere\_curs

from curs c, curs\_proiect\_cursant cpc, cursant cc

where cc.id\_cursant = cpc.id\_cursant and cpc.id\_curs = c.id\_curs

and c.id\_curs in (select id\_curs

from curs c2, instructor i, angajati a

where c2.id\_angajat = i.id\_angajat

and i.id\_angajat = a.id\_angajat

and initcap(a.nume) = 'Verdes' and initcap(a.prenume) = 'Maria')

group by cc.nume\_cursant, cc.prenume\_cursant, c.id\_curs, c.tip, c.nivel

having avg(decode(least(cpc.data\_predare, cpc.termen\_limita), cpc.data\_predare, cpc.nota, 0)) >= all (

select avg(decode(least(cpc2.data\_predare, cpc2.termen\_limita), cpc2.data\_predare, cpc2.nota, 0))

from curs\_proiect\_cursant cpc2

where cpc2.id\_curs = c.id\_curs

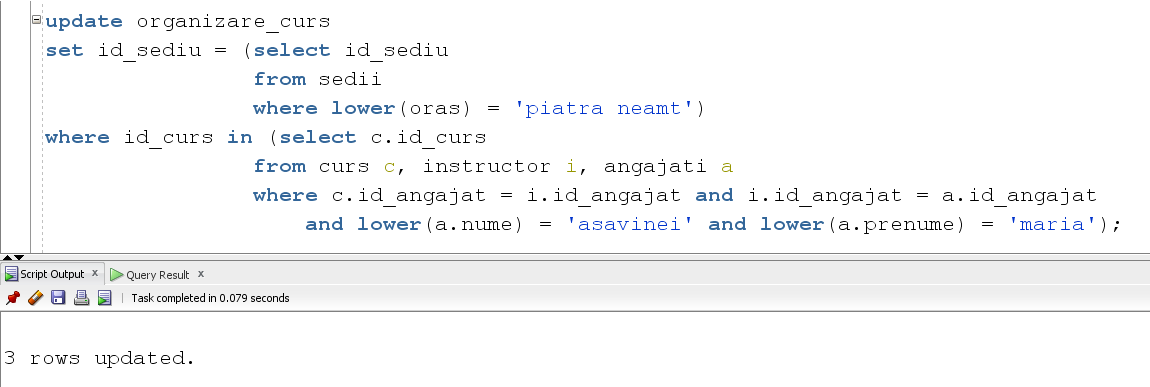
group by cpc2.id\_cursant

)

order by medie desc;

## Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri

**CEREREA 1**: Să se seteze sediul cursurilor predate de Asavinei Maria în sediul din Piatra Neamț.



**COD:**

update organizare\_curs

set id\_sediu = (select id\_sediu

from sedii

where lower(oras) = 'piatra neamt')

where id\_curs in (select c.id\_curs

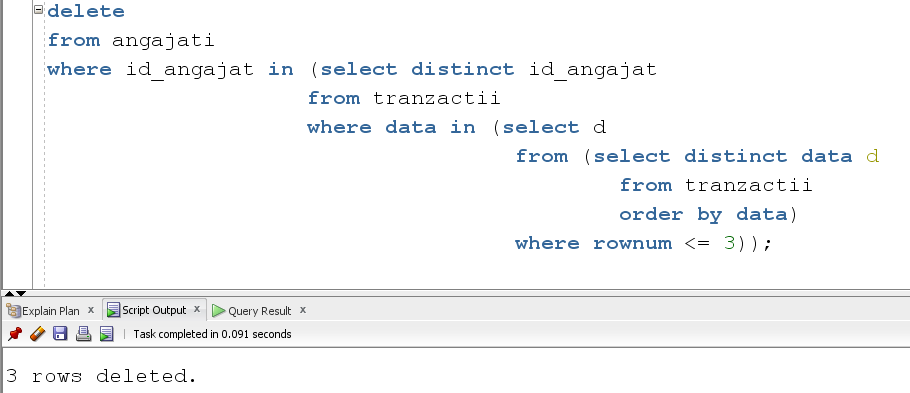
from curs c, instructor i, angajati a

where c.id\_angajat = i.id\_angajat and i.id\_angajat = a.id\_angajat

and lower(a.nume) = 'asavinei' and lower(a.prenume) = 'maria');

**CEREREA 2**: Să se șteargă acei vânzători care au realizat tranzacții în cele mai vechi 3 date calendaristice.

***COMPLETĂRI***: Având în vedere faptul că definirea constrângerii de cheie străină dintre tabelele ANGAJATI și VANZATOR asigură ștergerea vânzătorilor odată cu ștergerea acestora din tabela ANGAJAȚI, optăm să realizăm ștergerea în tabela ANGAJATI.



**COD:**

delete

from angajati

where id\_angajat in (select distinct id\_angajat

from tranzactii

where data in (select d

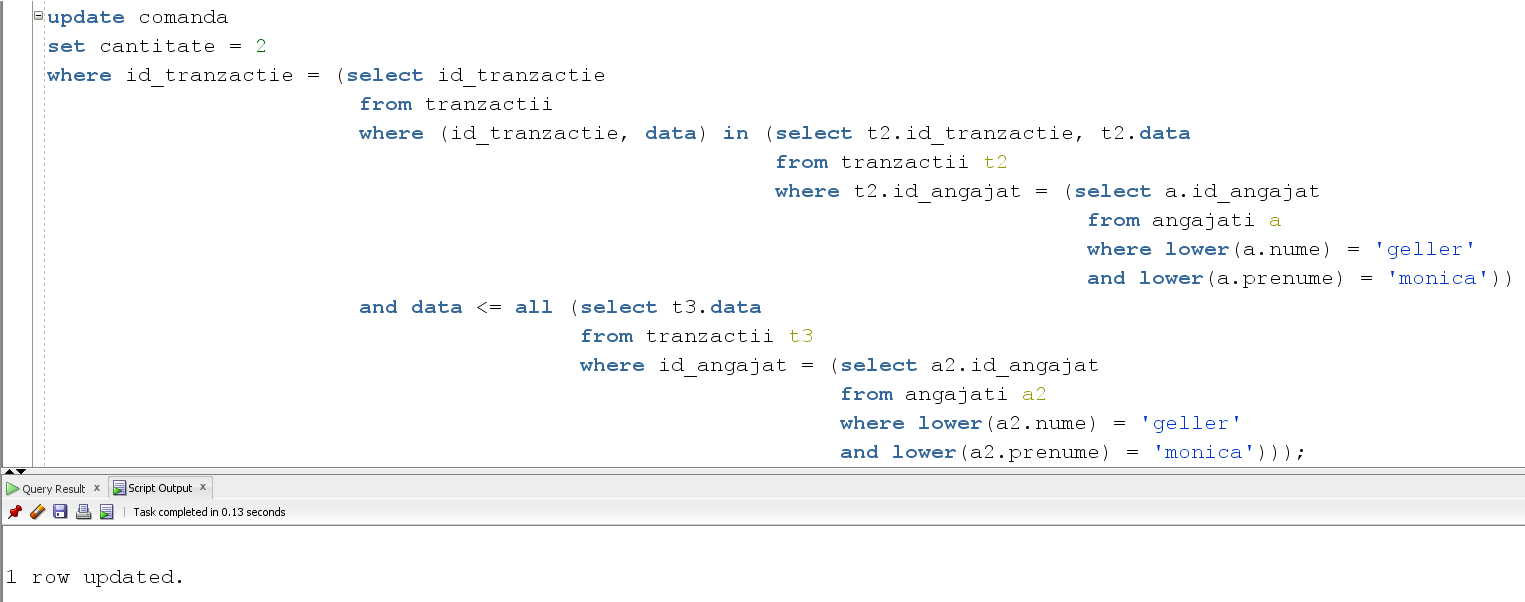
from (select distinct data d

from tranzactii

order by data)

where rownum <= 3));

**CEREREA 3**: Să se egaleze cantitățile corespunzătoare comenzilor din prima tranzacție realizată de Monica Geller cu 2.



**COD:**

update comanda

set cantitate = 2

where id\_tranzactie = (select id\_tranzactie

from tranzactii

where (id\_tranzactie, data) in (select t2.id\_tranzactie, t2.data

from tranzactii t2

where t2.id\_angajat = (select a.id\_angajat

from angajati a

where lower(a.nume) = 'geller'

and lower(a.prenume) = 'monica'))

and data <= all (select t3.data

from tranzactii t3

where id\_angajat = (select a2.id\_angajat

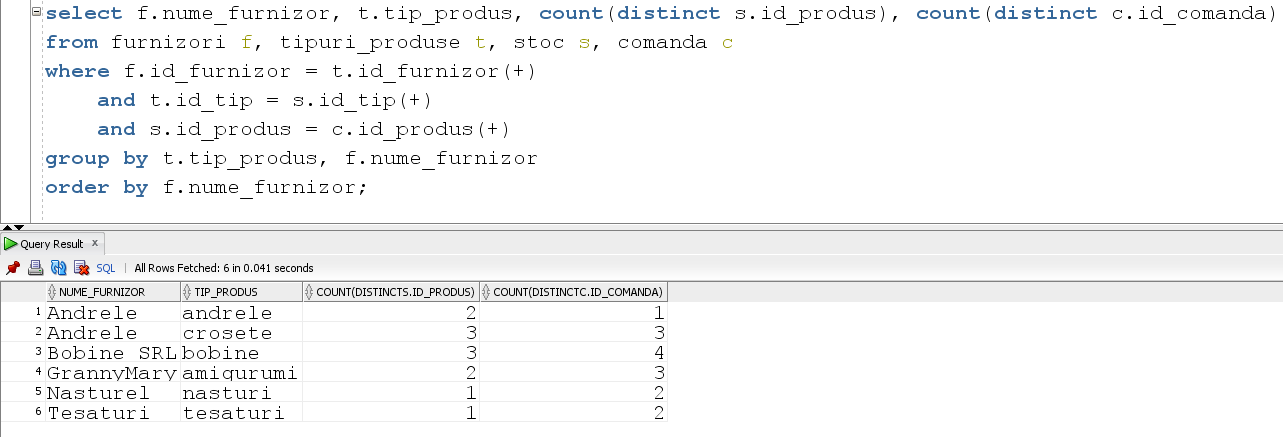
from angajati a2

where lower(a2.nume) = 'geller'

and lower(a2.prenume) = 'monica')));

## Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n

**CERERE OUTER-JOIN**: Afișați pentru fiecare furnizor tipurile de produse pe care le furnizează, numărul de produse de tipul respectiv din stoc și numărul de comenzi în care se regăsesc produse din fiecare tip inclusiv pentru produsele care nu se regăsesc în nicio comandă. Ordonați rezultatele după numele furnizorului.



**COD:**

select f.nume\_furnizor, t.tip\_produs, count(distinct s.id\_produs), count(distinct c.id\_comanda)

from furnizori f, tipuri\_produse t, stoc s, comanda c

where f.id\_furnizor = t.id\_furnizor(+)

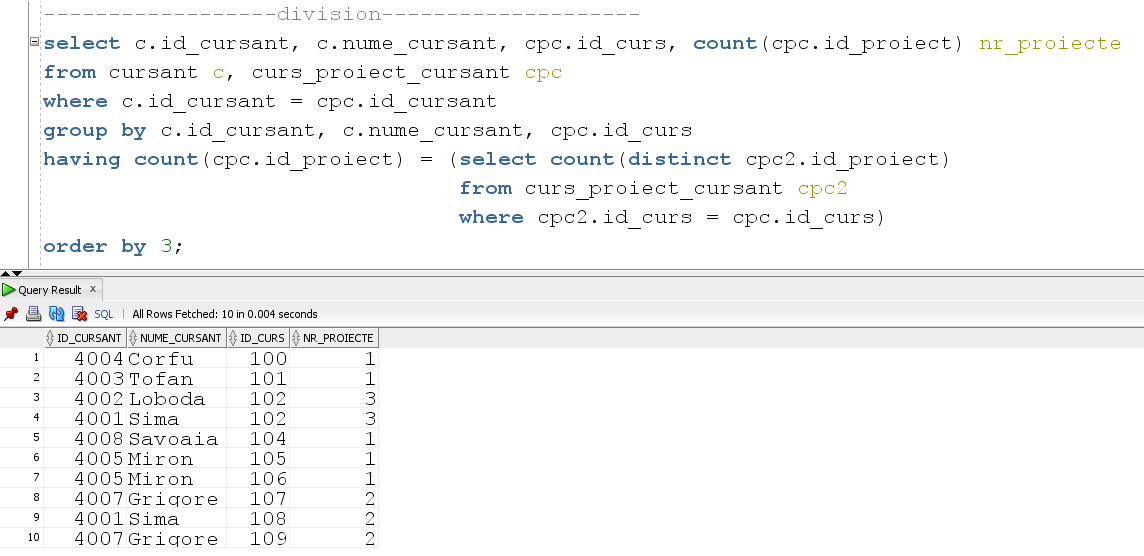
and t.id\_tip = s.id\_tip(+)

and s.id\_produs = c.id\_produs(+)

group by t.tip\_produs, f.nume\_furnizor

order by f.nume\_furnizor;

**CERERE DIVISION**: Să se afișeze id-ul și numele cursanților care au realizat toate proiectele din cadrul unui curs la care au participat, împreună cu id-ul cursului respectiv și numărul de proiecte. Ordonați rezultatele după id-urile cursurilor.



**COD:**

select c.id\_cursant, c.nume\_cursant, cpc.id\_curs, count(cpc.id\_proiect) nr\_proiecte

from cursant c, curs\_proiect\_cursant cpc

where c.id\_cursant = cpc.id\_cursant

group by c.id\_cursant, c.nume\_cursant, cpc.id\_curs

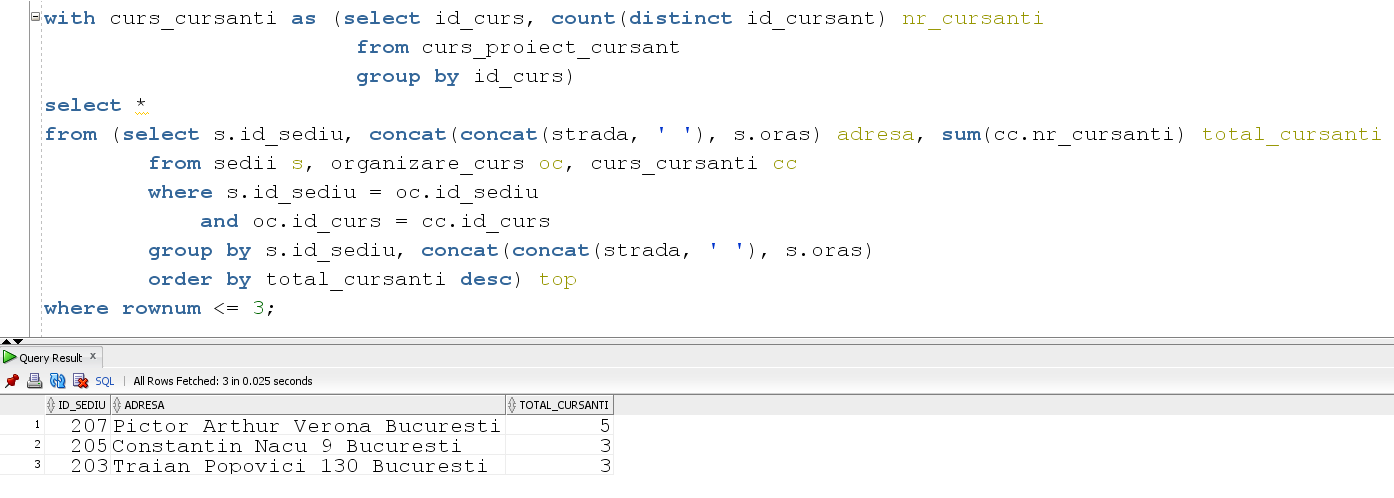
having count(cpc.id\_proiect) = (select count(distinct cpc2.id\_proiect)

from curs\_proiect\_cursant cpc2

where cpc2.id\_curs = cpc.id\_curs)

order by 3;

**CERERE TOP-N**: Să se afișeze cele top 3 sedii care găzduiesc cei mai mulți cursanți.



**COD:**

with curs\_cursanti as (select id\_curs, count(distinct id\_cursant) nr\_cursanti

from curs\_proiect\_cursant

group by id\_curs)

select \*

from (select s.id\_sediu, concat(concat(strada, ' '), s.oras) adresa, sum(cc.nr\_cursanti) total\_cursanti

from sedii s, organizare\_curs oc, curs\_cursanti cc

where s.id\_sediu = oc.id\_sediu

and oc.id\_curs = cc.id\_curs

group by s.id\_sediu, concat(concat(strada, ' '), s.oras)

order by total\_cursanti desc) top

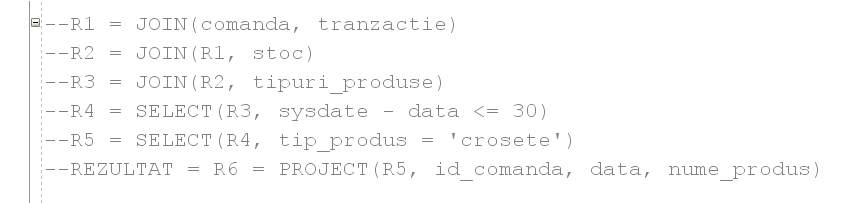
where rownum <= 3;

## Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale.

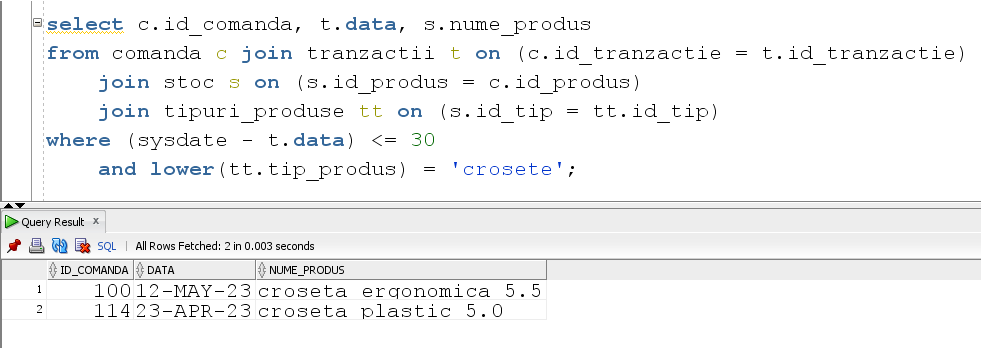
**CERERE**: Selectați comenzile care fac parte din tranzacții realizate în ultimele 30 de zile și care se referă la un produs de tip "croșete".

**ÎNAINTE DE OPTIMIZARE**

**EXPRESIE ALGEBRICĂ:**



**SQL:**



**COD:**

select c.id\_comanda, t.data, s.nume\_produs

from comanda c join tranzactii t on (c.id\_tranzactie = t.id\_tranzactie)

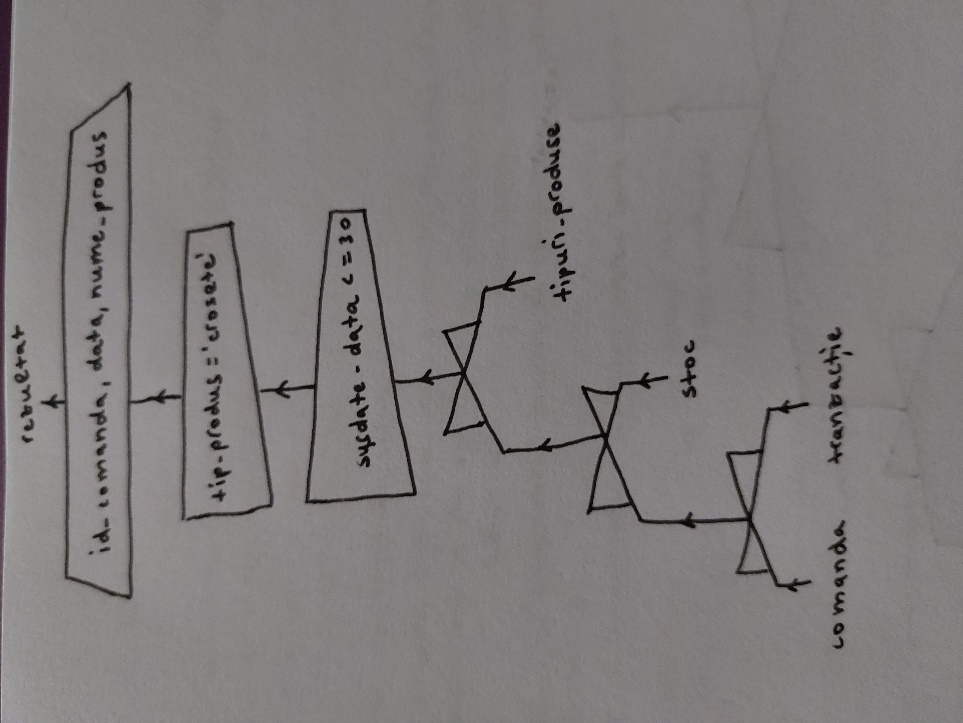
join stoc s on (s.id\_produs = c.id\_produs)

join tipuri\_produse tt on (s.id\_tip = tt.id\_tip)

where (sysdate - t.data) <= 30

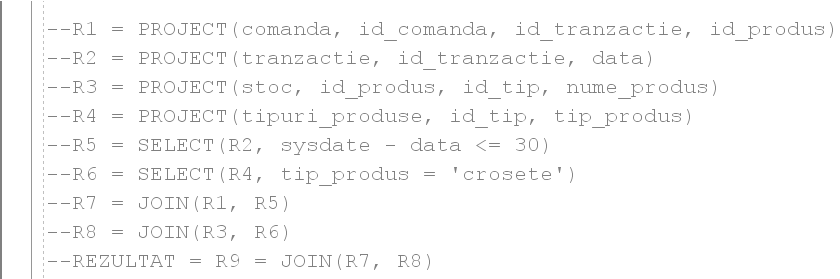
and lower(tt.tip\_produs) = 'crosete';

**ARBORE ALGEBRIC:**

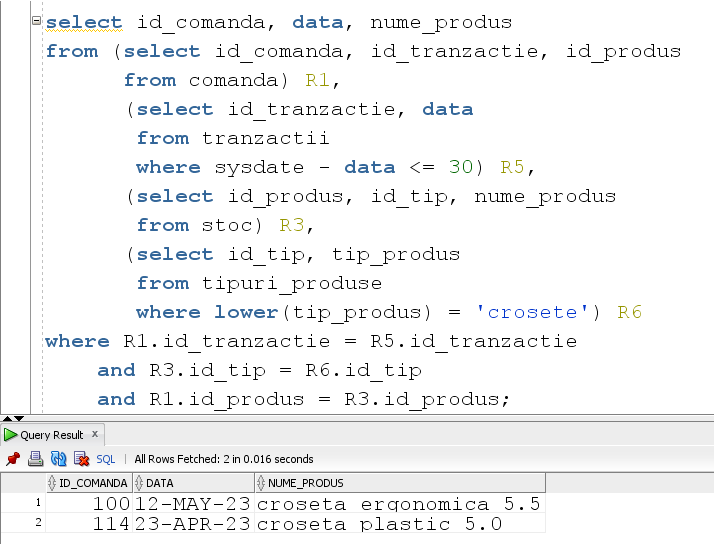
****

**DUPĂ OPTIMIZARE**

**EXPRESIE ALGEBRICĂ:**



**SQL:**



**COD:**

select id\_comanda, data, nume\_produs

from (select id\_comanda, id\_tranzactie, id\_produs

from comanda) R1,

(select id\_tranzactie, data

from tranzactii

where sysdate - data <= 30) R5,

(select id\_produs, id\_tip, nume\_produs

from stoc) R3,

(select id\_tip, tip\_produs

from tipuri\_produse

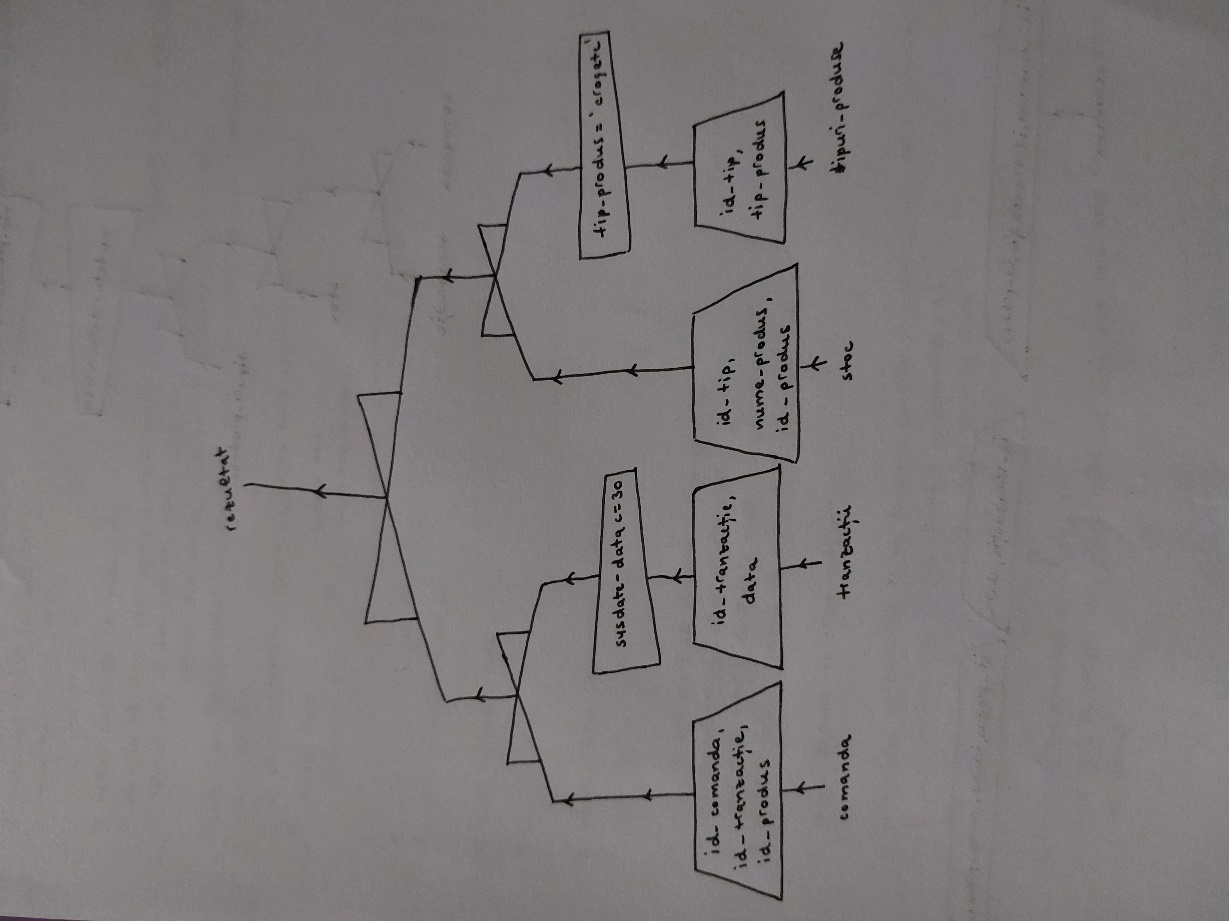
where lower(tip\_produs) = 'crosete') R6

where R1.id\_tranzactie = R5.id\_tranzactie

and R3.id\_tip = R6.id\_tip

and R1.id\_produs = R3.id\_produs;

**ARBORE ALGEBRIC:**

****

## a. Realizarea normalizării BCNF, FN4, FN5.

**BCNF**: Presupunem că tabelei PROIECT îi este adăugat atributul timp\_necesar. În acest caz, am avea dependența funcțională id\_proiect -> nume, dificultate, timp\_necesar, dar și o nouă dependență dificultate -> timp\_necesar. Această nouă dependență încalcă principiile BCNF prin faptul că dificultate este un determinant, dar nu este o cheie candidat. Pentru a rezolva acest lucru, modificăm tabela PROIECT astfel încât schema sa relațională devine PROIECT(id\_proiect#, nume, dificultate) și creăm o nouă tabelă, DIFICULTATE(dificultate#, timp\_necesar).

## b. Aplicarea denormalizării, justificând necesitatea acesteia.