Universitatea din **București**

FMI

INFORMATICĂ

Baza de date a unei firme de transport de mobilă

Telu Mihai-Sebastian

Grupa 132

[1.Descrierea modelului real, a utilității acestuia și a regulilor de funcționare 2](#_Toc177950978)

[2.Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului 3](#_Toc177950979)

[3.Descrierea entităților, incluzând precizarea cheii primare 4](#_Toc177950980)

[4.Descrierea relațiilor, incluzând precizarea cardinalității acestora 5](#_Toc177950981)

[5.Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor 7](#_Toc177950982)

[6.Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5 13](#_Toc177950983)

[7.Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 7 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ 14](#_Toc177950984)

[8.Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7 15](#_Toc177950985)

[9.Realizarea normalizării până la forma normală 3 (FN1-FN3) 16](#_Toc177950986)

[10.Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11) 19](#_Toc177950987)

[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; maxim 30 de înregistrări în fiecare tabel) 20](#_Toc177950988)

[12.Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în 48](#_Toc177950989)

[ansamblul lor, următoarele elemente: 48](#_Toc177950990)

[a) subcereri sincronizate în care intervin cel puțin 3 tabele 48](#_Toc177950991)

[b) subcereri nesincronizate în clauza FROM 48](#_Toc177950992)

[c) grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate 48](#_Toc177950993)

[(în clauza de HAVING) în care intervin cel puțin 3 tabele (in cadrul aceleiași cereri) 48](#_Toc177950994)

[d) ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri) 48](#_Toc177950995)

[e) utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, 48](#_Toc177950996)

[a cel puțin unei expresii CASE 48](#_Toc177950997)

[f) utilizarea a cel puțin 1 bloc de cerere (clauza WITH) 48](#_Toc177950998)

[Observație: Într-o cerere se vor regăsi mai multe elemente dintre cele enumerate mai sus, astfel 48](#_Toc177950999)

[încât cele 5 cereri să le cuprindă pe toate. 48](#_Toc177951000)

[13.Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri 54](#_Toc177951001)

[14.Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă 57](#_Toc177951002)

[15.Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n 61](#_Toc177951003)

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

Proiectul constituie baza de date a unei firme intermediare între producătorii de mobilă și clienții lor, anume: o firma de transport-mobilă. Practic, mobila comandată de client de la mai mulți furnizori de mobilă asociați firmei va fi transportată clientului de această. Utilitate modelului vine din reducerea semnificativă a costului de transport al mobilei, din moment ce clientul plătește unui singur transportator, în loc să o facă cel puțin atâtor agenții de transport câte firme producătoare de mobilă a vizitat. De asemenea, un alt beneficiu al modelului se reflectă in organizarea produselor achiziționate sub spectrul unei singure entități, înainte de a fi livrate (poate, toate deoadată), evitându-se astfel intervalul de așteptare între livrarea diverselor piese de mobilier achiziționate, precum și eventualele neplăceri determinate de pierderea unor obiecte. Pentru o mai bună înțelegere, voi aduce un exemplu, mai exact: una dintre numeroasele firme de curierat ce deservește mai multe edituri.

Regulile de funcționare ar fi urmatoarele:

* Un client (persoana fizică) plasează o comandă cu obiecte de mobilier provenite de la mai multe furnizori asociați firmei. Comandă este preluată de unul dintre dispecerii firmei (firma având ca job-uri: CEO\_firma, contabil, secretar, dispecer, sofer, lucrator\_depozit, lucrator\_intretinere).
* În momentul preluării, ei i se asociază un cod unic (id\_comanda), suma necesară transportului, un traseu (determinat de locația ultimului depozit al firmei, vizitat, nu neapărat pentru a încarca ceva, de autoturismul transportator, înainte de a merge la client și locația clientului), precum și diverse pachete cu obiectele comandate.
* Pachetele menționate anterior sunt ținute la diverse depozite ale companiei, fiecare pachet fiind caracterizat printr-o greutate, dar și printr-un volum. Totodată, depozitele au și ele anumite capacități de stocare.
* Comanda făcută de client la o anumită dată va fi pusă într-un autovehicul, condus de unul dintre șoferi, devenind un transport propriu-zis, ce va ajunge la client.
* Piesele din pachete trebuie să fi fost puse la dispoziția firmei de transport de către vreun furnizor de mobilă înainte de crearea comenzii.

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

* O comandă este trasportată de un singur șofer, cu un singur autovehicul și într-un singur transport.
* Nu orice comandă este musai să fie livrată, mai exact-să apară că livrată în bază, să determine un transport.
* Fiecare pachet format conține doar piese de același tip, cel puțin una. Un pachet nu poate exista în doua comenzi și în doua depozite. Ca atare, pot exista pachete cu același conținut, dar care să determine obiecte diferite.
* Pot exista șoferi care nu au efectuat transporturi, dar și dispeceri care să nu fi preluat comenzi.
* Pachetele dintr-un depozit nu trebuie să depășească dimensiunea acestuia.
* O aceeași piesă de mobilier poate fi furnizată de mai mulți producători de mobilă.
* Pot exista piese furnizate de producători, care să nu fie puse în vreun pachet și, totodată, un furnizor nu este obligat să furnizeze firmei de transport toate piesele de mobilier pe care le deține.
* Un client poate avea o unică locație și un singur număr de telefon.
* Orice piesă de mobilier transportată este detinută de un anumit furnizor, iar orice furnizor trebuie să permită companiei sa transporte măcar una dintre piesele de mobilier produse de ea, altfel nu mai este furnizor.
* Șeful direct al unui angajat de pe un job este un angajat de pe același job sau CEO\_firma.
* Un ajngajat nu poate lucra simultan pe doua job-uri.
* Nu există doi angajați cu același nume și același prenume.
* Volumul pachetelor nu este suma volumelor pieselor, analog și greutatea.
* Denumirile utilizeaza alfabetul român, dar fără diactritice.
* Nu se pot transporta cu un autovehicul comenzi mai mari decât dimensiunea sa, se poate invers.
* Nu e neapărat ca de la punctul de plecare al unei comenzi să se încarce ceva.

# 3.Descrierea entităților, incluzând precizarea cheii primare

Entitatile modelului sunt urmatoarele:

1. ANGAJAT = entitate ce identifică persoanele (fizice) care lucrează pentru firma de transport. Cheia primară este *id\_angajat*.
2. JOB\_FIRMA= entitate care determină rolul în companie al fiecărui angajat. Cheia primară este *id\_job*.
3. ISTORICUL\_JOBURILOR = entitate ce reține toate job-urile pe care a lucrat un angajat de la venirea în firmă, chiar și cel prezent (data\_sfarsit=null). Cheia primară este compusă: *(id\_angajat, data\_inceput).*
4. CLIENT\_FIRMA = entitate ce identifică persoanele fizice care apelează la serviciile companiei. Cheia primară este *id\_client*.
5. COMANDA = entitate ce determină legătură dintre un client și un angajat. Ea conține datele despre achiziția clientului. Cheia ei primară este *id\_comanda.*
6. TRASEU = entitate ce redă un anumit traseu și limitele sale (apare pentru a reda traseul comenzii). Cheia primară este *id\_traseu*.
7. AUTOVEHICUL =   redă mijloacele de transport utilizate în livrarea comenzilor. Sunt de diferite tipuri si sunt alese in concordanță cu dimensiunea transportului. Cheia primară este *nr\_inmatriculare*.
8. PACHET = entitate care intră în alcătuirea unei comenzi. În interiorul acestor pachete sunt puse obiecte de mobilier identice, comandate de client. Aceste pachete au anumite dimensiuni. Cheia primară este *id\_pachet*.
9. DEPOZIT = memorează clădirile în care sunt depozitate pachetele corespunzătoare comenzilor, înainte de a fi livrate într-un transport. Cheia primară este *id\_depozit*.
10. PIESA =  indică pisele de mobilier funizate de producătorii de mobilă asociați. Cheia primară este *id\_piesa*.
11. FURNIZOR\_MOBILA = entitate care memorează companiile furnizoare de mobilă. Cheia primară este *id\_furnizor*.
12. LOCATIE = memorează diverse locații, printre care cele ale clienților și cele ale depozitelor. Cheia primară este *id\_locatie*.

# 4.Descrierea relațiilor, incluzând precizarea cardinalității acestora

Relatiile corespunzătoare modelului sunt următoarele (alături de entitățile pe care le leagă):

1. ANGAJAT\_**detine**\_JOB\_FIRMA= relație care leagă entitățile ANGAJAT și JOB\_FIRMA,  indicându-ne faptul că fiecare angajat al firmei are un anumit rol în interior acesteia. Cardinalitatea minimă este 1:1 și maximă este n:1.
2. ANGAJAT\_**conduce**\_ANGAJAT = relație care leagă entitățile ANGAJAT și ANGAJAT, indicându-ne faptul că fiecare angajat al firmei are un anumit superior direct (poate chiar si null, în cazul CEO-lui). Cardinalitatea minimă este 0 : 0 și maximă este 1 : n.
3. ANGAJAT\_**se\_gaseste**\_ISTORICUL\_JOBURILOR = relație care leagă entitățile ANGAJAT și ISTORICUL\_JOBURILOR, indicându-ne faptul că fiecare angajat al firmei are un anumit istoric al job-urilor pe care le-a ocupat în cadrul companiei. Cardinalitatea minimă este 1:1 și maximă este 1: n.
4. ANGAJAT\_**primeste**\_COMANDA = relație care leagă entitățile ANGAJAT și COMANDA, indicându-ne faptul că orice camandă plasată de un client va fi primită de un angajat (dispecer). Cardinalitatea minimă este 1: 0 și maximă este 1: n.
5. CLIENT\_FIRMA\_**plaseaza**\_COMANDA = relație care leagă entitățile CLIENT\_FIRMA și COMANDA, indicându-ne faptul că un client poate plasa o comandă de transport cu mobilă provenită de la asociații companiei. Cardinalitatea minimă este 1: 1 și maximă este 1 : n.
6. COMANDA\_**are**\_TRASEU = relație care leagă entitățile COMANDA și TRASEU, indicându-ne faptul că fiecare comandă dată are un anumit traseu, descris de ultimul depozit pe la care trece comanda (depozitele sunt foarte apropiate) și locația clientului. Cardinalitatea minimă este 0 : 1 și maximă este n : 1.
7. COMANDA\_**cuprinde\_**PACHET = relație care leagă entitățile COMANDA și PACHET, indicându-ne faptul că fiecare comandă este formată dintr-un anumit număr de pachete cu mobilă. Cardinalitatea minimă este 1 : 1 și maximă este 1 : n.
8. DEPOZIT\_**contine**\_PACHET = relație care leagă entitățile DEPOZIT și PACHET, indicându-ne faptul că fiecare pachet format pentru o comandă este pus într-un depozit până la transportul său. Cardinalitatea minimă este 1 : 0 și maximă este 1 : n.
9. PACHET\_**se\_formeaza**\_ PIESA = relație care leagă entitățile PACHET și PIESA, indicându-ne faptul că fiecare pachet dintr-o comandă este format din piese identice de mobilier. Cardinalitatea minimă este 0 : 1 și maximă este n : 1.
10. FURNIZOR\_MOBILA\_**furnizeaza**\_PIESA = relație (de tip many-to-many) care leagă entitățile FURNIZOR\_MOBILA și PIESA , indicându-ne faptul că fiecare piesă de mobilier livrată de companie este provenită de la anumiți furnizori. Cardinalitatea minimă este 1 : 1 și maximă este m : n.
11. DEPOZIT\_**se\_afla**\_LOCATIE = relație care leagă entitățile DEPOZIT și LOCATIE, indicându-ne faptul că fiecare depozit se află într-o locație. Cardinalitatea minimă este 0 : 1 și maximă este n : 1.
12. CLIENT\_FIRMA\_**se\_afla**\_LOCATIE = relație care leagă entitățile CLIENT\_FIRMA și LOCATIE, indicându-ne faptul că fiecare client se află într-o locație, unde trebuie adusă comanda. Cardinalitatea minimă este 0 : 1 și maximă este n : 1.
13. ANGAJAT\_**transporta**\_COMANDA\_**cu**\_AUTOMOBIL = relație de tip 3, care leagă entitățile ANGAJAT, COMANDA și AUTOMOBIL, indicându-ne faptul că fiecare comandă este livrată de un singur angajat cu un singur automobil („cardinalitatea” ar fi: 1:n:1). Denumirea acestei relaţii va fi *transporta.*

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

Entitatea ANGAJAT are următoarele atribute:

1. id\_angajat = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unui angajat (cheia primară);
2. nume = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele angajatului;
3. prenume = variabilă de tip caracter, de lungime maximă 25, care reprezintă prenumele angajatului (constrângere: (nume,prenume) să fie unic);
4. salariu = variabilă de tip întreg, de lungime maximă 6, care reprezintă salariul lunar al angajatului, in lei (constrângere: să fie mai mare decât 0);
5. data\_angajare = variabilă de tip dată calendaristică, care reprezintă data angajării angajatului in firmă (are ca valoare default: sysdate);
6. id\_sef = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul șefului direct al angajatului (cheie externă);
7. id\_job = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul job-lui angajatului (cheie externă; constrângere: să fie nenul).

Entitatea JOB\_FIRMA are următoarele atribute:

1. id\_job = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unui job (cheia primară);
2. denumire = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele job-lui (valori posibile: CEO\_firma, contabil, secretar, dispecer, sofer, lucrator\_depozit, lucrator\_intretinere);
3. salariu\_minim = variabilă de tip întreg, de lungime maximă 6, care reprezintă salariul minim al job-lui (constrângere: să fie mai mare decât 0);
4. salariu\_maxim = variabilă de tip întreg, de lungime maximă 6, care reprezintă salariul maxim al job-lui (constrângere: să fie mai mare decât salariu\_minim).

Entitatea ISTORICUL\_JOBURILOR are următoarele atribute:

1. id\_angajat = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unui angajat (cheie primară+cheie externă);
2. data\_inceput = variabilă de tip dată calendaristică, care reprezintă data angajării angajatului pe job-ul respectiv (cheie primară);
3. data\_sfarsit = variabilă de tip dată calendaristică, care reprezintă data din care angajatul respectiv nu mai lucrează pe job-ul respectiv;
4. id\_job = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul job-lui pe care lucrează angajatul între cele 2 date.

Entitatea CLIENT\_FIRMA are urmatoarele atribute:

1. id\_client = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unui client (cheia primară);
2. nume = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele clientului;
3. prenume = variabilă de tip caracter, de lungime maximă 25, care reprezintă prenumele clientului;
4. nr\_telefon = variabilă de tip caracter, de lungime maximă 10, care reprezintă numărul de telefon al clientului (constrângeri: să fie nenul, să fie unic);
5. id\_locatie = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul locației clientului (cheie externă; constrângere: să fie nenul).

Entitatea COMANDA are urmatoarele atribute:

1. id\_comanda = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unei comenzi (cheia primară);
2. id\_client = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul clientului ce plasează comanda (cheie externă; constrângere: să fie diferit de null);
3. id\_angajat = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul angajatului ce preia comanda (cheie externă; constrângere: să fie diferit de null);
4. id\_traseu = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul traseului comenzii (cheie externă; constrângere: să fie diferit de null);
5. suma = variabilă de tip întreg, de lungime maximă 6, care reprezintă suma plătită de client pe comandă, in lei (constrângere: să fie mai mare decât 0);
6. data\_comanda = variabilă de tip dată calendaristică, care reprezintă data când este facută comanda (are ca valoare default: sysdate).

Entitatea TRASEU are următoarele atribute:

1. id\_traseu = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unui tarseu (cheia primară);
2. pct\_plecare = variabilă de tip întreg, de lungime maximă 4, care reprezintă punctul/id-ul punctului de plecare al traseului (locația depozitului vizitat ultima oară de transportator);
3. pct\_sosire = variabilă de tip întreg, de lungime maximă 4, care reprezintă punctul de sosire al traseului (locația clientului).

Entitatea AUTOVEHICUL are următoarele atribute:

1. nr\_inmatriculare = variabilă de tip caracter, de lungime maximă 8, care reprezintă numărul de înmatriculare al unui autovehicul folosit pentru transport (cheia primară);
2. tip = variabilă de tip caracter, de lungime maximă 8, care reprezintă tipul autovehicului (valori posibile: VAN, AVIA, TIR, TIR\_rem; acestea au niște dimensiuni (de transport) standard, respectiv: ( VAN, 13m3, 1.5t), ( AVIA, 40m3, 3.5t), ( TIR, 90m3, 23t), ( TIR\_rem, 120m3, 30t); întrucât sunt puține tipuri de autovehicul pentru transport, s-a evitat construirea unui nou tabel cu cheia primară tip și atributele volum, greutate).

Entitatea LOCATIE are următoarele atribute:

1. id\_locatie = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unei locații (cheia primară);
2. tara = variabilă de tip caracter, de lungime maximă 25, care reprezintă țara locației;
3. oras = variabilă de tip caracter, de lungime maximă 25, care reprezintă orașul corespunzător locației.

Entitatea DEPOZIT are următoarele atribute:

1. id\_depozit = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unui depozit (cheia primară);
2. capacitate = variabilă de tip întreg, de lungime maximă 4, care reprezintă capacitatea de stocare a depozitului (în metri cub; constrângere: să fie mai mare decât 0);
3. id\_locatie = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul locației depozitului (cheie externă; constrângere: să fie nenul).

Entitatea PIESA are urmatoarele atribute:

1. id\_piesa = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unei piese (cheia primară);
2. denumire = variabilă de tip caracter, de lungime maximă 25, care reprezintă denumirea piesei (valori posibile: scaun\_x, canapea\_x, pat\_x, fotoliu\_x, masa\_x, noptiera\_x, birou\_x, dulap\_x, etajera\_x etc, unde x este un cod din cifre și litere ce determină caracteristici ale obiectului, care nu prezintă interes pentru modelul descris, astfel că nu determină o nouă coloană sau un nou tabel).

Entitatea PACHET are următoarele atribute:

1. id\_pachet = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unui pachet (cheia primară);
2. id\_comanda = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul comenzii în care este inclus pachetul (cheie externă; constrângere: să fie diferit de null);
3. id\_depozit = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unui depozit în se află pachetul (cheie externă; constrângere: să fie diferit de null);
4. id\_piesa = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul piesei din care este format pachetul (cheie externă; constrângere: să fie diferit de null);
5. nr\_piese = variabilă de tip întreg, de lungime maximă 4, care reprezintă numărul de piese din pachetul respectiv (constrângere: să fie mai mare decât 0) ;
6. volum = variabilă de tip întreg, de lungime maximă 4, care reprezintă volumul pachetului (în m3 ; constrângere: să fie mai mare decât 0);
7. greutate = variabilă de tip întreg, de lungime maximă 4, care reprezintă greutatea pachetului (în kg; constrângere: să fie mai mare decât 0).

Entitatea FURNIZOR\_MOBILA are următoarele atribute:

1. id\_furnizor = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unui furnizor de mobilă (cheia primară);
2. denumire = variabilă de tip caracter, de lungime maximă 25, care reprezintă denumirea furnizorului.

Relaţia FURNIZOR\_MOBILA\_**furnizeaza**\_PIESA are următorul atribut:

1. data\_furnizare = variabilă de tip dată calendaristică, care reprezintă data de când furnizorul de mobilă permite firmei de transport să trasporte obiectul de mobilier respectiv, produs de către ea.

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

AUTOVEHICUL

nr\_inmatriculare#

transporta

DEPOZIT

id\_depozit#

contine

M(0)

1

FURNIZOR\_MOBILA

id\_furnizor#

PACHET

id\_pachet#

TRASEU

id\_traseu#

are

M(0)

1

CLIENT\_FIRMA

id\_client#

plaseaza

M(1)

1

COMANDA

id\_comanda#

primeste

M(0)

1

conduce

1(0)

M(0)

ISTORICUL\_JOBURILOR

id\_angajat#

data\_inceput#

ANGAJAT

id\_angajat#

JOB\_FIRMA

id\_job#

detine

M(1)

1

se\_gaseste

M(1)

1

cuprinde

M(1)

1

PIESA

id\_piesa#

se\_formeaza

1

M(0)

furnizeaza

M(1)

M(1)

LOCATIE

id\_locatie#

M(0)

1

se\_afla

M(0)

1

se\_afla

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

AUTOVEHICUL

nr\_inmatriculare#

TRANSPORTA

id\_comanda#

id\_angajat#

nr\_inmatriculare#

DEPOZIT

id\_depozit#

contine

FURNIZOR\_MOBILA

id\_furnizor#

PACHET

id\_pachet#

TRASEU

id\_traseu#

are

CLIENT\_FIRMA

id\_client#

plaseaza

COMANDA

id\_comanda#

primeste

conduce

ISTORICUL\_JOBURILOR

id\_angajat#

data\_inceput#

ANGAJAT

id\_angajat#

JOB\_FIRMA

id\_job#

detine

se\_gaseste

cuprinde

PIESA

id\_piesa#

se\_formeaza

FURNIZEAZA

id\_furnizor#

id\_piesa#

LOCATIE

id\_locatie#

se\_afla

se\_afla

.

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

ANGAJAT(id\_angajat#, nume, prenume, salariu, data\_angajare, id\_sef, id\_job);

JOB\_FIRMA(id\_job#, denumire, salariu\_minim, salariu\_maxim);

ISTORICUL\_JOBURILOR(id\_angajat#, data\_inceput#, data\_sfarsit, id\_job);

COMANDA(id\_comanda#, id\_client, id\_angajat, id\_traseu, suma, data\_comanda);

TRANSPORTA(id\_comanda#, id\_angajat#, nr\_inmatriculare#);

AUTOVEHICUL(nr\_inmatriculare#, tip);

TRASEU(id\_traseu#, pct\_plecare, pct\_sosire);

CLIENT\_FIRMA(#id\_client, nume, prenume, nr\_telefon, id\_locatie);

PACHET(id\_pachet#, id\_comanda, id\_depozit, id\_piesa, nr\_piese, volum, greutate);

DEPOZIT(id\_depozit#, capacitate, id\_locatie);

PIESA(id\_piesa#, denumire);

FURNIZOR\_MOBILA(id\_furnizor#, denumire);

FURNIZEAZA(id\_piesa#, id\_furnizor#, data\_furnizare);

LOCATIE(id\_locatie#, tara, oras).

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

**FN1:**

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

Un contraexemplu relativ la diagrama conceptuală ar fi dacă entitatea ISTORICUL\_JOBURILOR nu ar fi avut în  componența cheii primare și atributul: data\_inceput (sau data\_sfarsit). Problemele ar fi apărut când un angajat ar fi lucrat de mai multe ori pe același job ori când ar fi avut mai multe job-uri (căci apar valori multiple/compuse pe unele coloane, iar relaţia nu mai este în FN1). Rezolvarea s-ar face adăungand linii și reconsiderând cheia primară.

ISTORICUL\_JOBURILOR

|  |  |  |  |
| --- | --- | --- | --- |
| id\_angajat# | data\_inceput | data\_sfarsit | id\_job |
| 101 | 01-Jan-04, 02-Feb-04 | 01-Feb-04,  null | 200,210 |
| 110 | 03-Mar-06, 05-Mar-07 | 04-Jan-07,  null | 201 |

REZOLVARE (în diagrama):

ISTORICUL\_JOBURILOR

|  |  |  |  |
| --- | --- | --- | --- |
| id\_angajat# | data\_inceput# | data\_sfarsit | id\_job |
| 101 | 01-Jan-04 | 01-Feb-04 | 200 |
| 101 | 02-Feb-04 | null | 210 |
| 110 | 03-Mar-06 | 04-Jan-07 | 201 |
| 110 | 05-Mar-07 | null | 201 |

**FN2:**

O relaţie R este în a doua formă normală dacă şi numai dacă:

 relaţia R este în FN1;

 fiecare atribut care nu este cheie (nu participă la cheia primară) este

dependent de întreaga cheie primară.

Un contraexemplu relativ la diagrama conceptuală ar fi dacă entitatea AUTOVEHICUL nu ar fi existat, iar relația TRANSPORTA s-ar fi numit TRANSPORT și ar fi determinat un tabel din diagramă entitate-relație, care, față de tabelul asociativ, ar fi avut în plus și atributul „tip” (pentru tipul vehiculului). În mod evident, tip ar fi dependent funcțional de nr\_inmatriculare, deci de o parte a cheii primare, că atare, relația nu ar fi în FN2. Rezolvarea s-ar face aplicând regulă „Casey-Delobel”, obținându-se tabelele TRANSPORT (TRANSPORTA din diagramă conceptuală) și AUTOVEHICUL (adică, prin crearea unei noi relații AUTOVEHICUL( nr\_inmatriculare#, tip) și legarea de tabelul TRANSPORT, din care se elimina atributul tip).

TRANSPORT:

|  |  |  |  |
| --- | --- | --- | --- |
| id\_comanda# | id\_angajat# | nr\_inmatriculare# | tip |
| 300 | 200 | AG01CTM | TIR |
| 301 | 201 | AG01CTM | TIR |
| 302 | 200 | AG02CTM | VAN |
| 303 | 200 | AG02CTM | VAN |

REZOLVARE:

TRANSPORT:

|  |  |  |
| --- | --- | --- |
| id\_comanda# | id\_angajat# | nr\_inmatriculare# |
| 300 | 200 | AG01CTM |
| 301 | 201 | AG01CTM |
| 302 | 200 | AG02CTM |
| 303 | 200 | AG02CTM |

AUTOVEHICUL:

|  |  |
| --- | --- |
| nr\_inmatriculare# | tip |
| AG01CTM | TIR |
| AG02CTM | VAN |

**FN3:**

O relaţie este în FN3 dacă şi numai dacă fiecare atribut (coloană) care nu este cheie, depinde de cheie, de întreaga cheie şi numai de cheie.

Un contraexemplu relativ la diagrama conceptuală ar fi dacă entitatea AUTOVEHICUL ar conţine, de exemplu, şi atributul „volum” (fără a-l considera implicit). Se va observa cum acest atribut depinde de atributul „tip” al relaţie, care nu constituie cheia primară, deci relaţia nu este în FN3. Rezolvarea s-ar face aplicând regulă „Casey-Delobel”, obținându-se tabelele AUTOVEHICUL(nr\_inmatriculare#, tip) şi TIP\_AUTOVEHICUL(#tip,volum).

AUTOVEHICUL:

|  |  |  |
| --- | --- | --- |
| nr\_inmatriculare# | tip | volum |
| AG01CTM | TIR | 90m3 |
| AG02CTM | TIR | 90m3 |
| AG03CTM | TIR | 90m3 |
| AG04CTM | VAN | 15m3 |

REZOLVARE:

AUTOVEHICUL:

|  |  |
| --- | --- |
| nr\_inmatriculare# | tip |
| AG01CTM | TIR |
| AG02CTM | TIR |
| AG03CTM | TIR |
| AG04CTM | VAN |

TIP\_AUTOVEHICUL:

|  |  |
| --- | --- |
| tip# | volum |
| TIR | 90m3 |
| VAN | 15m3 |

OBS: În cazul entității PACHET, nu există o dependență funcțională între volum/greutate și id\_piesa (nr\_piese), întrucât volumul pachetului nu este suma volumelor pieselor, analog și greutatea!

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

create sequence incrementare

start with 100

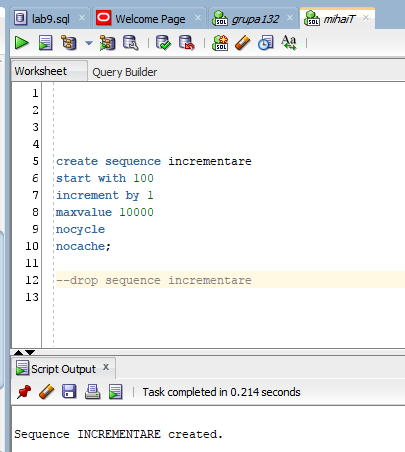
increment by 1

maxvalue 10000

nocycle

nocache;

--drop sequence incrementare



# 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; maxim 30 de înregistrări în fiecare tabel)

**Crearea tabelelor:**

create table JOB\_FIRMA(

id\_job number(4),

denumire varchar2(25),

salariu\_minim number(6),

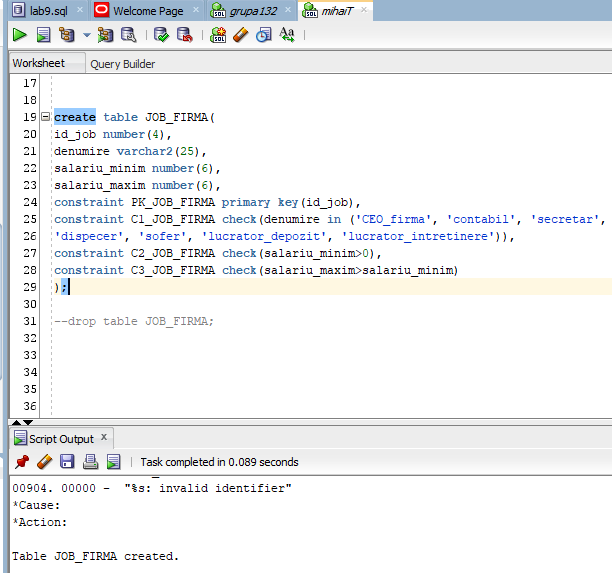
salariu\_maxim number(6),

constraint PK\_JOB\_FIRMA primary key(id\_job),

constraint C1\_JOB\_FIRMA check(denumire in ('CEO\_firma', 'contabil', 'secretar', 'dispecer', 'sofer', 'lucrator\_depozit', 'lucrator\_intretinere')),

constraint C2\_JOB\_FIRMA check(salariu\_minim>0),

constraint C3\_JOB\_FIRMA check(salariu\_maxim>salariu\_minim)

);--drop table JOB\_FIRMA

create table ANGAJAT(

id\_angajat number(4),

nume varchar2(25),

prenume varchar2(25),

salariu number(6),

data\_angajare date default sysdate,

id\_sef number(4),

id\_job number(4),

constraint PK\_ANGAJAT primary key(id\_angajat),

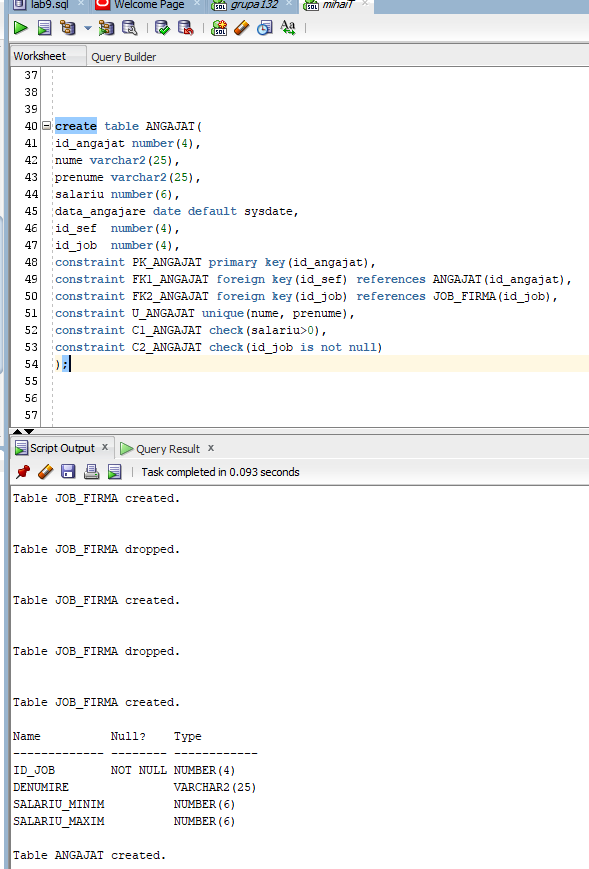
constraint FK1\_ANGAJAT foreign key(id\_sef) references ANGAJAT(id\_angajat),

constraint FK2\_ANGAJAT foreign key(id\_job) references JOB\_FIRMA(id\_job),

constraint U\_ANGAJAT unique(nume, prenume),

constraint C1\_ANGAJAT check(salariu>0),

constraint C2\_ANGAJAT check(id\_job is not null));



create table ISTORICUL\_JOBURILOR(

id\_angajat number(4),

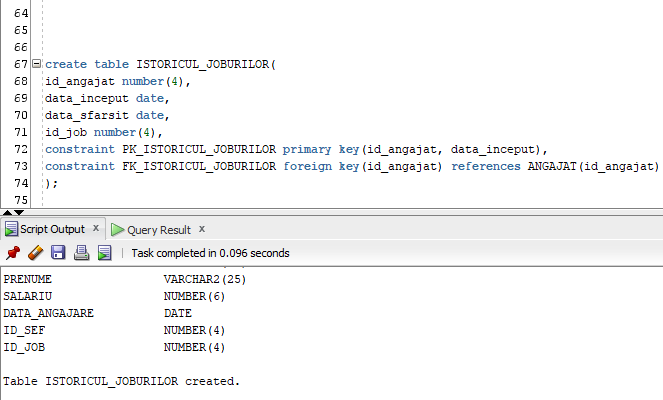
data\_inceput date,

data\_sfarsit date,

id\_job number(4),

constraint PK\_ISTORICUL\_JOBURILOR primary key(id\_angajat, data\_inceput),

constraint FK\_ISTORICUL\_JOBURILOR foreign key(id\_angajat) references ANGAJAT(id\_angajat));

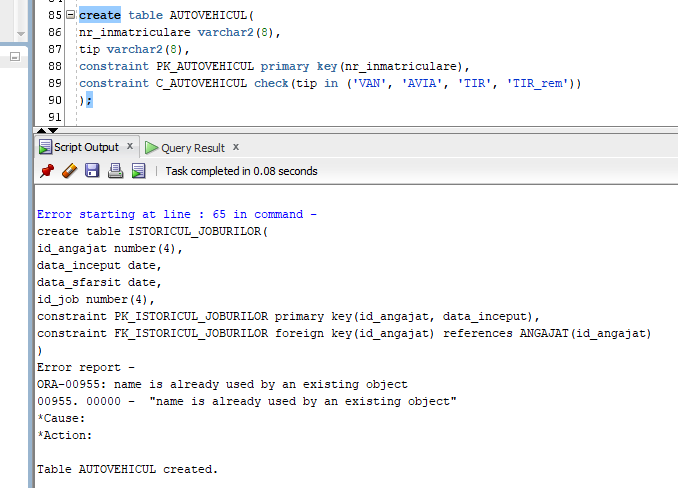


create table AUTOVEHICUL(

nr\_inmatriculare varchar2(8),

tip varchar2(8),

constraint PK\_AUTOVEHICUL primary key(nr\_inmatriculare),

constraint C\_AUTOVEHICUL check(tip in ('VAN', 'AVIA', 'TIR', 'TIR\_rem'))); 

create table TRASEU(

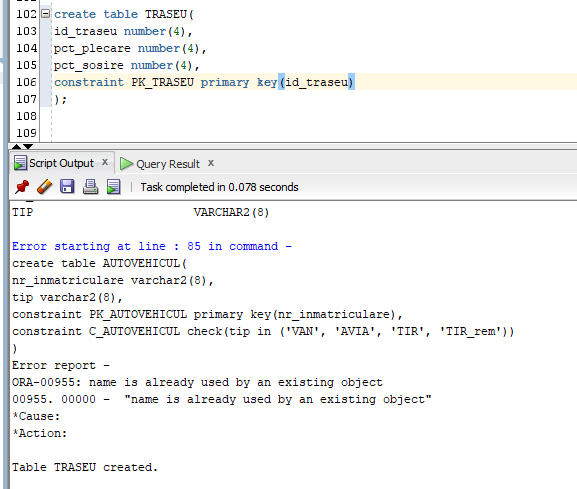
id\_traseu number(4),

pct\_plecare number(4),

pct\_sosire number(4),

constraint PK\_TRASEU primary key(id\_traseu)

);



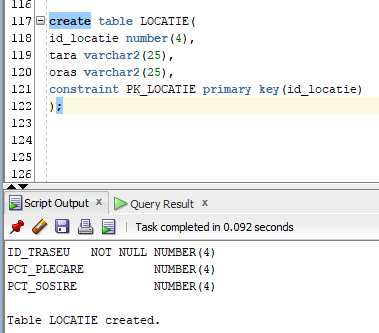
create table LOCATIE(

id\_locatie number(4),

tara varchar2(25),

oras varchar2(25),

constraint PK\_LOCATIE primary key(id\_locatie));



create table CLIENT\_FIRMA(

id\_client number(4),

nume varchar2(25),

prenume varchar2(25),

nr\_telefon varchar2(10),

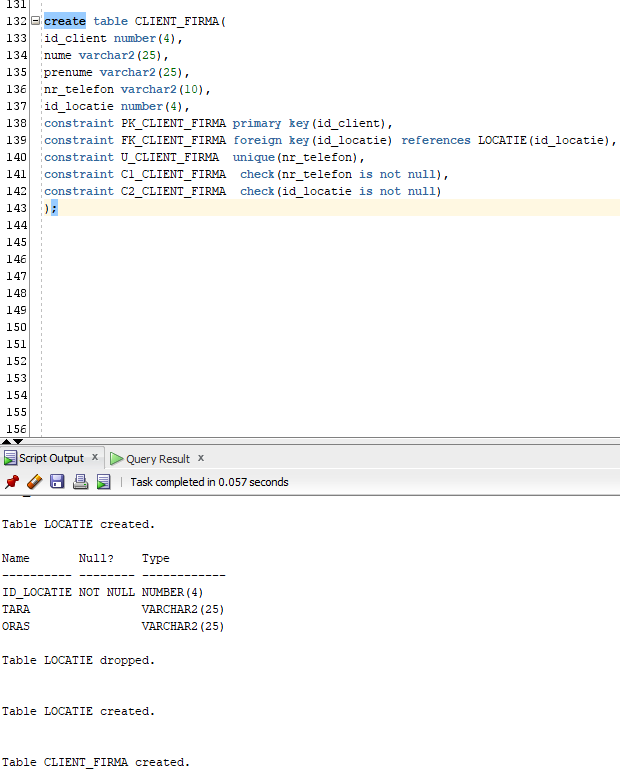
id\_locatie number(4),

constraint PK\_CLIENT\_FIRMA primary key(id\_client),

constraint FK\_CLIENT\_FIRMA foreign key(id\_locatie) references LOCATIE(id\_locatie),

constraint U\_CLIENT\_FIRMA unique(nr\_telefon),

constraint C1\_CLIENT\_FIRMA check(nr\_telefon is not null),

constraint C2\_CLIENT\_FIRMA check(id\_locatie is not null)); 

create table DEPOZIT(

id\_depozit number(4),

capacitate number(4),

id\_locatie number(4),

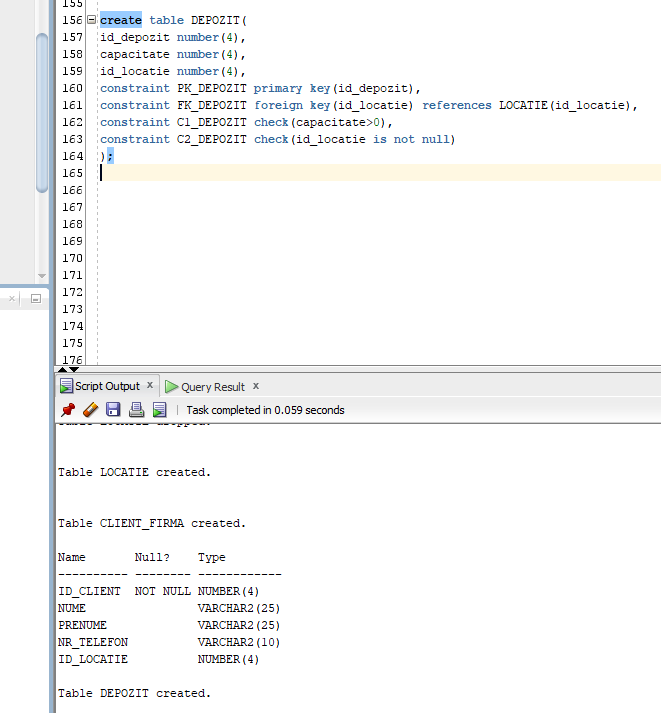
constraint PK\_DEPOZIT primary key(id\_depozit),

constraint FK\_DEPOZIT foreign key(id\_locatie) references LOCATIE(id\_locatie),

constraint C1\_DEPOZIT check(capacitate>0),

constraint C2\_DEPOZIT check(id\_locatie is not null)

);



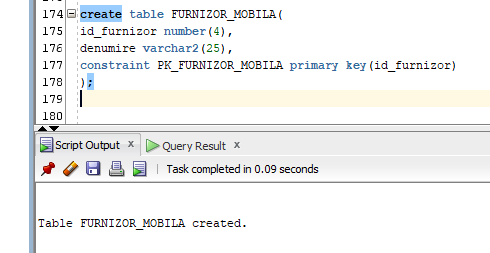
create table FURNIZOR\_MOBILA(

id\_furnizor number(4),

denumire varchar2(25),

constraint PK\_FURNIZOR\_MOBILA primary key(id\_furnizor)

);



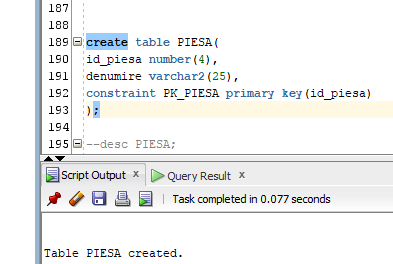
create table PIESA(

id\_piesa number(4),

denumire varchar2(25),

constraint PK\_PIESA primary key(id\_piesa)

);



create table FURNIZEAZA(

id\_piesa number(4),

id\_furnizor number(4),

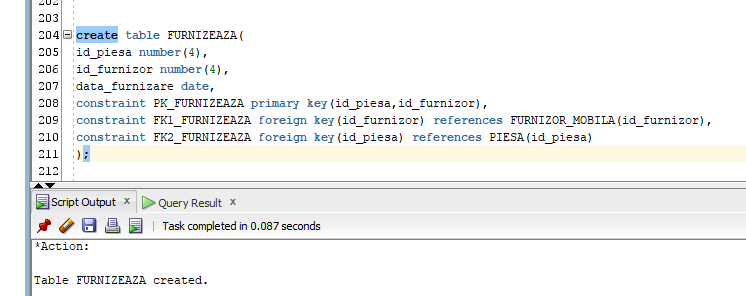
data\_furnizare date,

constraint PK\_FURNIZEAZA primary key(id\_piesa,id\_furnizor),

constraint FK1\_FURNIZEAZA foreign key(id\_furnizor) references FURNIZOR\_MOBILA(id\_furnizor),

constraint FK2\_FURNIZEAZA foreign key(id\_piesa) references PIESA(id\_piesa)

);



create table COMANDA(

id\_comanda number(4),

id\_client number(4),

id\_angajat number(4),

id\_traseu number(4),

suma number(6),

data\_comanda date default sysdate,

constraint PK\_COMANDA primary key(id\_comanda),

constraint FK1\_COMANDA foreign key(id\_client) references CLIENT\_FIRMA(id\_client),

constraint FK2\_COMANDA foreign key(id\_angajat) references ANGAJAT(id\_angajat),

constraint FK3\_COMANDA foreign key(id\_traseu) references TRASEU(id\_traseu),

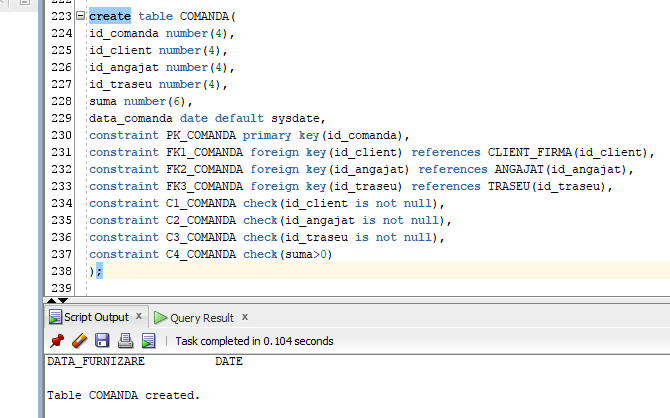
constraint C1\_COMANDA check(id\_client is not null),

constraint C2\_COMANDA check(id\_angajat is not null),

constraint C3\_COMANDA check(id\_traseu is not null),

constraint C4\_COMANDA check(suma>0)

);



create table TRANSPORTA(

id\_comanda number(4),

id\_angajat number(4),

nr\_inmatriculare varchar2(8),

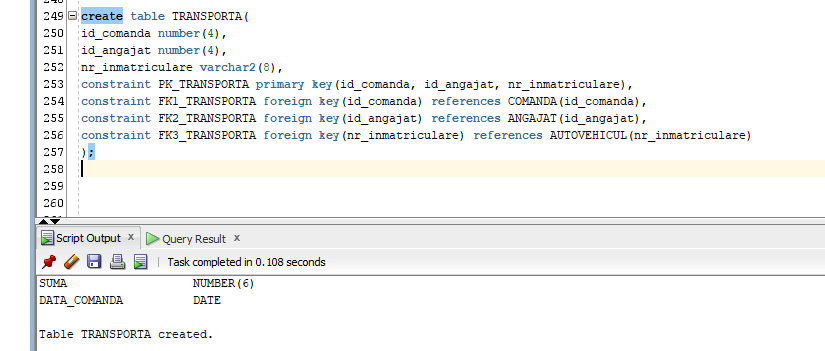
constraint PK\_TRANSPORTA primary key(id\_comanda, id\_angajat, nr\_inmatriculare),

constraint FK1\_TRANSPORTA foreign key(id\_comanda) references COMANDA(id\_comanda),

constraint FK2\_TRANSPORTA foreign key(id\_angajat) references ANGAJAT(id\_angajat),

constraint FK3\_TRANSPORTA foreign key(nr\_inmatriculare) references AUTOVEHICUL(nr\_inmatriculare)

);



create table PACHET(

id\_pachet number(4),

id\_comanda number(4) constraint C4 not null,

id\_depozit number(4) constraint C5 not null,

id\_piesa number(4) constraint C6 not null,

nr\_piese number(4),

volum number(4),

greutate number(4),

constraint PK\_PACHET primary key(id\_pachet),

constraint FK1\_PACHET foreign key(id\_comanda) references COMANDA(id\_comanda),

constraint FK2\_PACHET foreign key(id\_depozit) references DEPOZIT(id\_depozit),

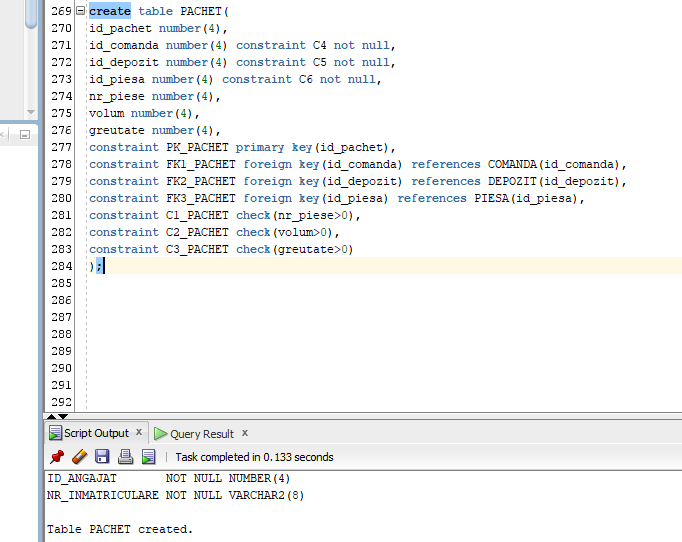
constraint FK3\_PACHET foreign key(id\_piesa) references PIESA(id\_piesa),

constraint C1\_PACHET check(nr\_piese>0),

constraint C2\_PACHET check(volum>0),

constraint C3\_PACHET check(greutate>0)

);



**Inserarea de date in tabele:**

* JOB\_FIRMA:

insert into JOB\_FIRMA values(incrementare.nextval, 'lucrator\_intretinere', 1000+incrementare.nextval\*10+(incrementare.nextval-100)\*210, 1000+incrementare.nextval\*15+(incrementare.nextval-100)\*210);

insert into JOB\_FIRMA values(incrementare.nextval, 'lucrator\_depozit', 1000+incrementare.nextval\*10+(incrementare.nextval-100)\*210, 1000+incrementare.nextval\*15+(incrementare.nextval-100)\*210);

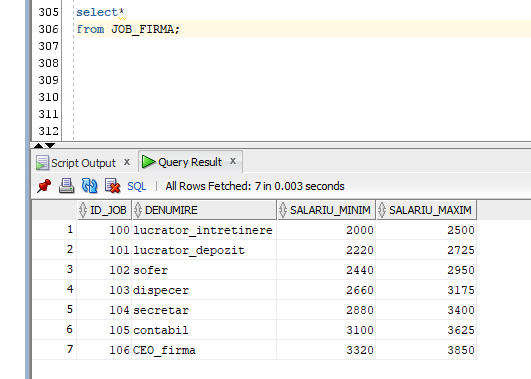
insert into JOB\_FIRMA values(incrementare.nextval, 'sofer', 1000+incrementare.nextval\*10+(incrementare.nextval-100)\*210, 1000+incrementare.nextval\*15+(incrementare.nextval-100)\*210);

insert into JOB\_FIRMA values(incrementare.nextval, 'dispecer', 1000+incrementare.nextval\*10+(incrementare.nextval-100)\*210, 1000+incrementare.nextval\*15+(incrementare.nextval-100)\*210);

insert into JOB\_FIRMA values(incrementare.nextval, 'secretar', 1000+incrementare.nextval\*10+(incrementare.nextval-100)\*210, 1000+incrementare.nextval\*15+(incrementare.nextval-100)\*210);

insert into JOB\_FIRMA values(incrementare.nextval, 'contabil', 1000+incrementare.nextval\*10+(incrementare.nextval-100)\*210, 1000+incrementare.nextval\*15+(incrementare.nextval-100)\*210);

insert into JOB\_FIRMA values(incrementare.nextval, 'CEO\_firma', 1000+incrementare.nextval\*10+(incrementare.nextval-100)\*210, 1000+incrementare.nextval\*15+(incrementare.nextval-100)\*210);



* ANGAJAT:

insert into ANGAJAT values(incrementare.nextval+4, 'Butculescu', 'Catalin', 3500, '02-Feb-10', null, 106);

insert into ANGAJAT values(incrementare.nextval+4, 'Deaconu', 'Catalin', 3200, '02-Feb-12', 111, 105);

insert into ANGAJAT values(incrementare.nextval+4, 'Stanciu', 'Silviana', 3000, '02-Feb-12', 111, 104);

insert into ANGAJAT values(incrementare.nextval+4, 'Paraschiv', 'Razvan', 2900, '01-Mar-12', 111, 103);

insert into ANGAJAT values(incrementare.nextval+4, 'Velcea', 'Ionut', 2800, '10-Mar-12', 114, 103);

insert into ANGAJAT values(incrementare.nextval+4, 'Bilous', 'Vlad', 2800, '10-Mar-12', 114, 103);

insert into ANGAJAT values(incrementare.nextval+4, 'Telu', 'Andrei', 2700, '01-Mar-12', 111, 102);

insert into ANGAJAT values(incrementare.nextval+4, 'Vanocea', 'Sebi', 2700, '12-Apr-12', 117, 102);

insert into ANGAJAT values(incrementare.nextval+4, 'Gheorghe', 'Andrei', 2700, '12-Apr-12', 117, 102);

insert into ANGAJAT values(incrementare.nextval+4, 'Stan', 'Antonio', 2500, '20-May-13', 117, 102);

insert into ANGAJAT values(incrementare.nextval+4, 'Epingeac', 'Robert', 2400, '15-May-12', 111, 101);

insert into ANGAJAT values(incrementare.nextval+4, 'Lazar', 'Dan', 2300, '16-Sep-13', 121, 101);

insert into ANGAJAT values(incrementare.nextval+4, 'Filip', 'Mirel', 2300, '16-May-12', 121, 101);

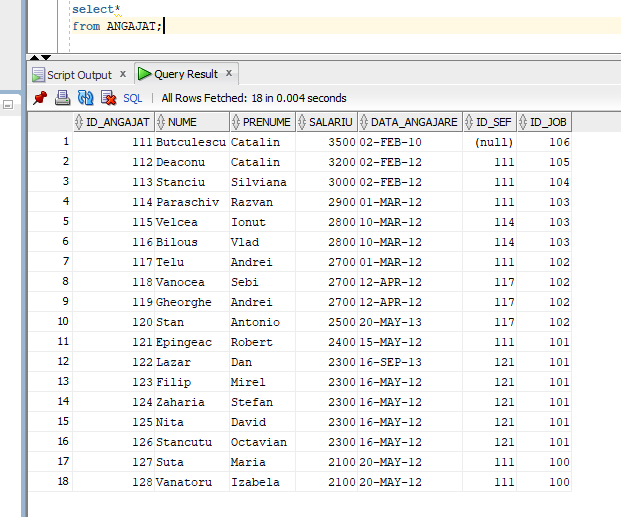
insert into ANGAJAT values(incrementare.nextval+4, 'Zaharia', 'Stefan', 2300, '16-May-12', 121, 101);

insert into ANGAJAT values(incrementare.nextval+4, 'Nita', 'David', 2300, '16-May-12', 121, 101);

insert into ANGAJAT values(incrementare.nextval+4, 'Stancutu', 'Octavian', 2300, '16-May-12', 121, 101);

insert into ANGAJAT values(incrementare.nextval+4, 'Suta', 'Maria', 2100, '20-May-12', 111, 100);

insert into ANGAJAT values(incrementare.nextval+4, 'Vanatoru', 'Izabela', 2100, '20-May-12', 111, 100);



* ISTORICUL\_JOBURILOR:

insert into ISTORICUL\_JOBURILOR values(111,'02-Feb-10', null, 106);

insert into ISTORICUL\_JOBURILOR values(112,'02-Feb-12', null, 105);

insert into ISTORICUL\_JOBURILOR values(113,'02-Feb-12', null, 104);

insert into ISTORICUL\_JOBURILOR values(114,'01-Mar-12', null, 103);

insert into ISTORICUL\_JOBURILOR values(115,'10-Mar-12', null, 103);

insert into ISTORICUL\_JOBURILOR values(116,'10-Mar-12', null, 103);

insert into ISTORICUL\_JOBURILOR values(117,'01-Mar-12', null, 102);

insert into ISTORICUL\_JOBURILOR values(118,'12-Apr-12', null, 102);

insert into ISTORICUL\_JOBURILOR values(119,'12-Apr-12', null, 102);

insert into ISTORICUL\_JOBURILOR values(120,'20-May-13','01-Mar-15', 103);

insert into ISTORICUL\_JOBURILOR values(120,'02-Mar-15',null, 102);

insert into ISTORICUL\_JOBURILOR values(121,'15-May-12', null, 101);

insert into ISTORICUL\_JOBURILOR values(122,'16-Sep-13', '02-Dec-13', 100);

insert into ISTORICUL\_JOBURILOR values(122,'04-Dec-13', null, 101);

insert into ISTORICUL\_JOBURILOR values(123,'16-May-12', null, 101);

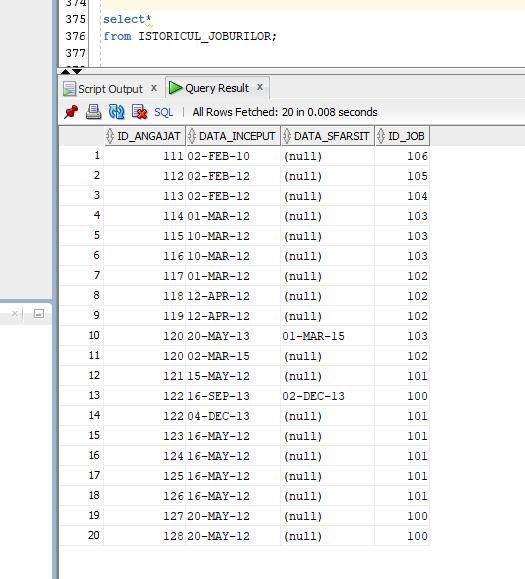
insert into ISTORICUL\_JOBURILOR values(124,'16-May-12', null, 101);

insert into ISTORICUL\_JOBURILOR values(125,'16-May-12', null, 101);

insert into ISTORICUL\_JOBURILOR values(126,'16-May-12', null, 101);

insert into ISTORICUL\_JOBURILOR values(127,'20-May-12', null, 100);

insert into ISTORICUL\_JOBURILOR values(128,'20-May-12', null, 100);



* AUTOVEHICUL:

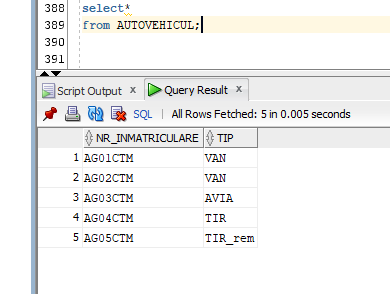
insert into AUTOVEHICUL values ('AG01CTM','VAN');

insert into AUTOVEHICUL values ('AG02CTM','VAN');

insert into AUTOVEHICUL values ('AG03CTM','AVIA');

insert into AUTOVEHICUL values ('AG04CTM','TIR');

insert into AUTOVEHICUL values ('AG05CTM','TIR\_rem');



* TRASEU:

insert into TRASEU values (200, 300, 400);

insert into TRASEU values (201, 300, 401);

insert into TRASEU values (202, 300, 402);

insert into TRASEU values (203, 301, 400);

insert into TRASEU values (204, 301, 403);

insert into TRASEU values (205, 301, 404);

insert into TRASEU values (206, 301, 402);

insert into TRASEU values (207, 302, 405);

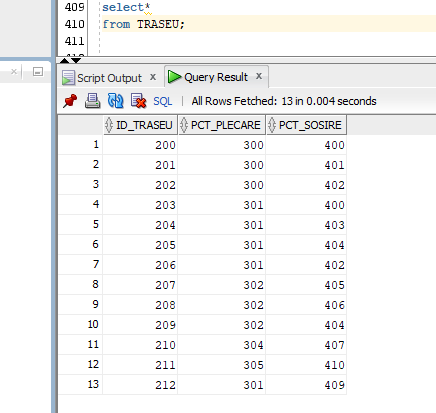
insert into TRASEU values (208, 302, 406);

insert into TRASEU values (209, 302, 404);

insert into TRASEU values (210, 304, 407);

insert into TRASEU values (211, 305, 410);

insert into TRASEU values (212, 301, 409);



* LOCATIE:

insert into LOCATIE values(300, 'Romania', 'Topoloveni');

insert into LOCATIE values(301, 'Romania', 'Pitesti');

insert into LOCATIE values(302, 'Romania', 'Gaesti');

insert into LOCATIE values(303, 'Romania', 'Mioveni');

insert into LOCATIE values(304, 'Romania', 'Titu');

insert into LOCATIE values(305, 'Romania', 'Braila');--depozit iesit din uz

insert into LOCATIE values(400, 'Romania', 'Stefanesti');

insert into LOCATIE values(401, 'Romania', 'Calinesti');

insert into LOCATIE values(402, 'Romania', 'Costesti');

insert into LOCATIE values(403, 'Romania', 'Campulung');

insert into LOCATIE values(404, 'Romania', 'Bradu');

insert into LOCATIE values(405, 'Romania', 'Targoviste');

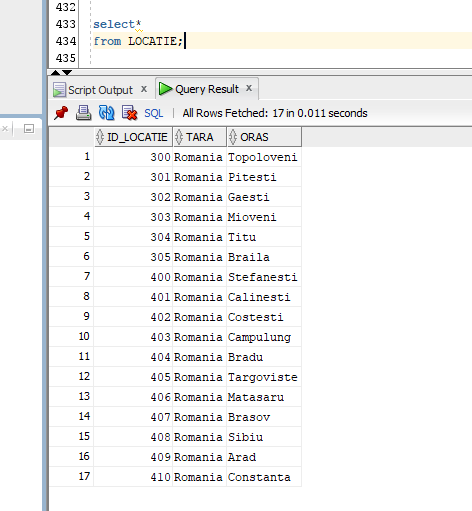
insert into LOCATIE values(406, 'Romania', 'Matasaru');

insert into LOCATIE values(407, 'Romania', 'Brasov');

insert into LOCATIE values(408, 'Romania', 'Sibiu');

insert into LOCATIE values(409, 'Romania', 'Arad');

insert into LOCATIE values(410, 'Romania', 'Constanta');



* CLIENT\_FIRMA:

insert into CLIENT\_FIRMA values(500, 'Popescu', 'Andrei', '0712444567', 400);

insert into CLIENT\_FIRMA values(501, 'Ionescu', 'Carmen', '0713464588', 401);

insert into CLIENT\_FIRMA values(502, 'Radu', 'Marin', '0725767321', 402);

insert into CLIENT\_FIRMA values(503, 'Dumitrescu', 'Cristina', '0734225789', 403);

insert into CLIENT\_FIRMA values(504, 'Vasile', 'Alexandru', '0747889022', 404);

insert into CLIENT\_FIRMA values(505, 'Marin', 'Alexandru', '0756442331', 405);

insert into CLIENT\_FIRMA values(506, 'Georgescu', 'Alexandru', '0763188256', 406);

insert into CLIENT\_FIRMA values(507, 'Popescu', 'Ana', '0778991333', 407);

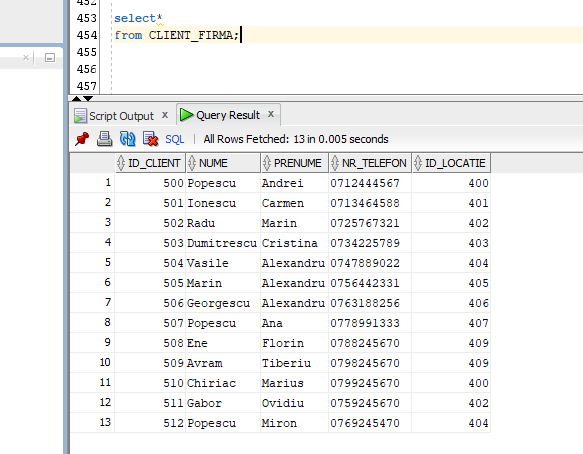
insert into CLIENT\_FIRMA values(508, 'Ene', 'Florin', '0788245670', 409);

insert into CLIENT\_FIRMA values(509, 'Avram', 'Tiberiu', '0798245670', 409);

insert into CLIENT\_FIRMA values(510, 'Chiriac', 'Marius', '0799245670', 400);

insert into CLIENT\_FIRMA values(511, 'Gabor', 'Ovidiu', '0759245670', 402);

insert into CLIENT\_FIRMA values(512, 'Popescu', 'Miron', '0769245470', 404);



* DEPOZIT:

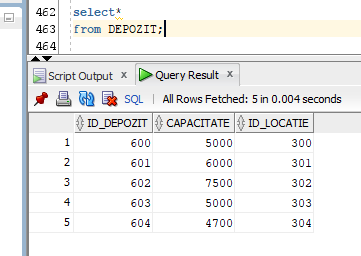
insert into DEPOZIT values(600,5000,300);

insert into DEPOZIT values(601,6000,301);

insert into DEPOZIT values(602,7500,302);

insert into DEPOZIT values(603,5000,303);

insert into DEPOZIT values(604,4700,304);



* FURNIZOR\_MOBILA:

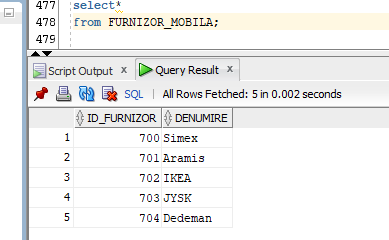
insert into FURNIZOR\_MOBILA values(700, 'Simex');

insert into FURNIZOR\_MOBILA values(701, 'Aramis');

insert into FURNIZOR\_MOBILA values(702, 'IKEA');

insert into FURNIZOR\_MOBILA values(703, 'JYSK');

insert into FURNIZOR\_MOBILA values(704, 'Dedeman');



* PIESA:

insert into PIESA values (800, 'scaun\_a2');

insert into PIESA values (801, 'scaun\_b2');

insert into PIESA values (802, 'canapea\_a2');

insert into PIESA values (803, 'canapea\_t6');

insert into PIESA values (804, 'pat\_s42');

insert into PIESA values (805, 'pat\_b15');

insert into PIESA values (806, 'pat\_a2');

insert into PIESA values (807, 'pat\_b2');

insert into PIESA values (808, 'canapea\_x4');

insert into PIESA values (809, 'fotoliu\_b53');

insert into PIESA values (810, 'fotoliu\_c4');

insert into PIESA values (811, 'masa\_d6');

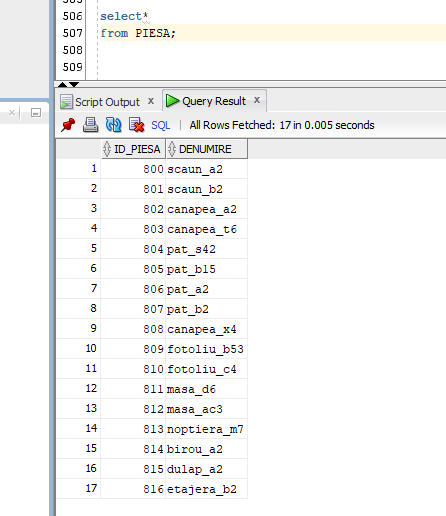
insert into PIESA values (812, 'masa\_ac3');

insert into PIESA values (813, 'noptiera\_m7');

insert into PIESA values (814, 'birou\_a2');

insert into PIESA values (815, 'dulap\_a2');

insert into PIESA values (816, 'etajera\_b2');



* FURNIZEAZA:

insert into FURNIZEAZA values(800, 700, '01-May-2010');

insert into FURNIZEAZA values(800, 704, '11-Jan-2011');

insert into FURNIZEAZA values(801, 700, '10-May-2010');

insert into FURNIZEAZA values(802, 701, '15-Jun-16');

insert into FURNIZEAZA values(803, 701, '07-Jan-2011');

insert into FURNIZEAZA values(804, 702, '10-Aug-2017');

insert into FURNIZEAZA values(805, 702, '17-Nov-2011');

insert into FURNIZEAZA values(806, 703, '24-Sep-2011');

insert into FURNIZEAZA values(807, 700, '22-Aug-2011');

insert into FURNIZEAZA values(807, 704, '13-Jun-2012');

insert into FURNIZEAZA values(808, 702, '29-Aug-2011');

insert into FURNIZEAZA values(809, 703, '05-Oct-2011');

insert into FURNIZEAZA values(810, 702, '06-Oct-2011');

insert into FURNIZEAZA values(811, 701, '21-Apr-2012');

insert into FURNIZEAZA values(812, 701, '09-Mar-2012');

insert into FURNIZEAZA values(813, 704, '16-Aug-2011');

insert into FURNIZEAZA values(814, 704, '16-Aug-2011');

insert into FURNIZEAZA values(814, 700, '18-Aug-2011');

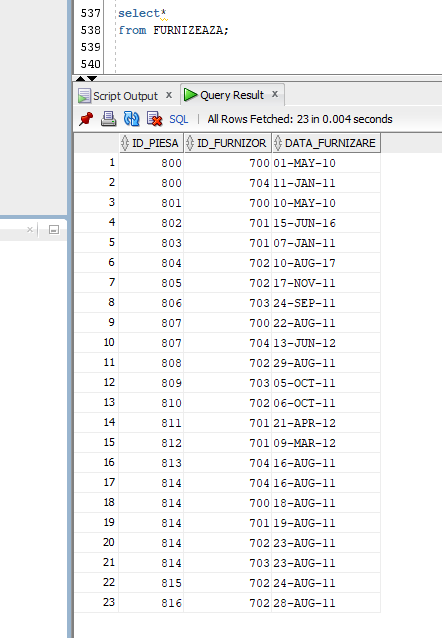
insert into FURNIZEAZA values(814, 701, '19-Aug-2011');

insert into FURNIZEAZA values(814, 702, '23-Aug-2011');

insert into FURNIZEAZA values(814, 703, '23-Aug-2011');

insert into FURNIZEAZA values(815, 702, '24-Aug-2011');

insert into FURNIZEAZA values(816, 702, '28-Aug-2011');



* COMANDA:

insert into COMANDA values(900, 500, 114, 200, 1000, '02-Mar-14');

insert into COMANDA values(901, 500, 116, 203, 1300, '07-Apr-15');

insert into COMANDA values(902, 501, 114, 201, 1800, '17-Jun-19');

insert into COMANDA values(903, 501, 115, 201, 2000, '15-Aug-19');

insert into COMANDA values(904, 502, 114, 202, 1250, '16-May-18');

insert into COMANDA values(905, 502, 114, 206, 1100, '17-Jun-20');

insert into COMANDA values(906, 503, 115, 204, 2200, '19-Jun-19');

insert into COMANDA values(907, 504, 114, 205, 800, '21-Jun-19');

insert into COMANDA values(908, 504, 115, 209, 1830, '14-Apr-2020');

insert into COMANDA values(909, 505, 115, 207, 1640, '12-Apr-2020');

insert into COMANDA values(910, 506, 114, 208, 1538, '10-Jul-2017');

insert into COMANDA values(911, 507, 115, 210, 2682, '10-Jan-2014');

insert into COMANDA values(912, 507, 116, 210, 1530, '14-Mar-2015');

insert into COMANDA values(913, 507, 115, 210, 1820, '14-Apr-2015');

insert into COMANDA values(914, 508, 115, 212, 3030, '22-Aug-2020');

insert into COMANDA values(915, 509, 114, 212, 1430, '13-Feb-2021');

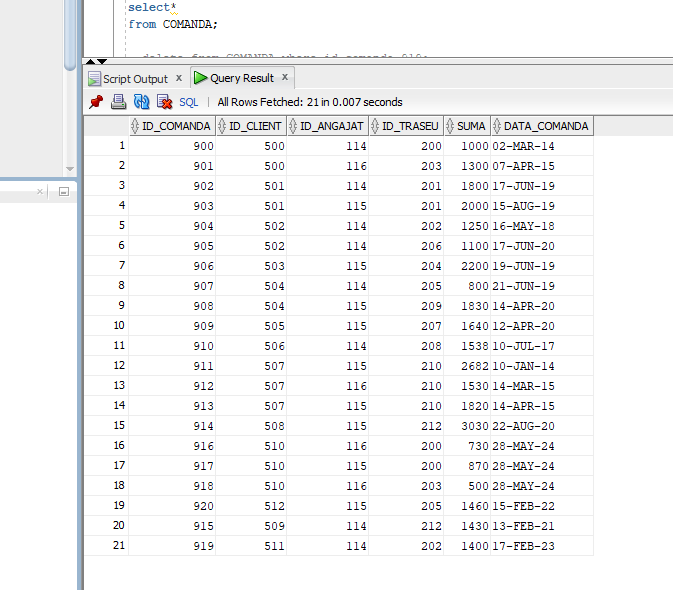
insert into COMANDA(id\_comanda, id\_client, id\_angajat, id\_traseu, suma) values(916, 510, 116, 200, 730);

insert into COMANDA(id\_comanda, id\_client, id\_angajat, id\_traseu, suma) values(917, 510, 115, 200, 870);

insert into COMANDA(id\_comanda, id\_client, id\_angajat, id\_traseu, suma) values(918, 510, 116, 203, 500);

insert into COMANDA values(919, 511, 114, 202, 1400, '17-Feb-2023');

insert into COMANDA values(920, 512, 115, 205, 1460, '15-Feb-2022');



* TRANSPORTA:

insert into TRANSPORTA values(900, 117, 'AG01CTM');

insert into TRANSPORTA values(901, 117, 'AG03CTM');

insert into TRANSPORTA values(902, 118, 'AG03CTM');

insert into TRANSPORTA values(903, 119, 'AG04CTM');

insert into TRANSPORTA values(904, 118, 'AG02CTM');

insert into TRANSPORTA values(905, 120, 'AG01CTM');

insert into TRANSPORTA values(906, 118, 'AG04CTM');

insert into TRANSPORTA values(907, 119, 'AG02CTM');

insert into TRANSPORTA values(908, 119, 'AG03CTM');

insert into TRANSPORTA values(909, 120, 'AG03CTM');

insert into TRANSPORTA values(910, 120, 'AG03CTM');

insert into TRANSPORTA values(911, 117, 'AG05CTM');

insert into TRANSPORTA values(913, 118, 'AG03CTM');

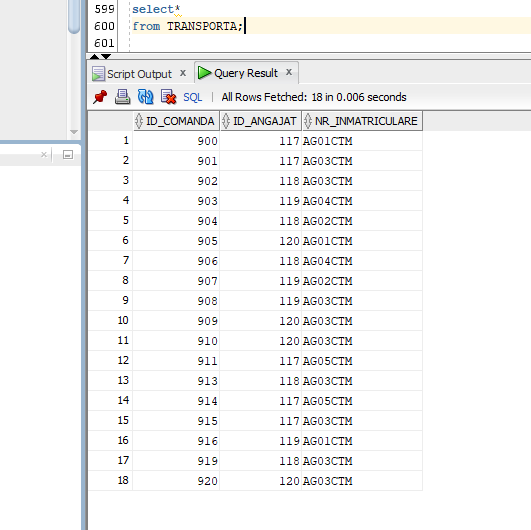
insert into TRANSPORTA values(914, 117, 'AG05CTM');

insert into TRANSPORTA values(915, 117, 'AG03CTM');

insert into TRANSPORTA values(916, 119, 'AG01CTM');

insert into TRANSPORTA values(919, 118, 'AG03CTM');

insert into TRANSPORTA values(920, 120, 'AG03CTM');



* PACHET:

insert into PACHET values(1000, 900, 600, 800, 10, 6, 300);

insert into PACHET values(1001, 900, 600, 803, 1, 7, 100);

insert into PACHET values(1002, 901, 601, 811, 5, 20, 1600);

insert into PACHET values(1003, 902, 601, 805, 2, 10, 800);

insert into PACHET values(1004, 902, 600, 809, 5, 15, 800);

insert into PACHET values(1005, 903, 600, 814, 20, 45, 2400);

insert into PACHET values(1006, 904, 602, 816, 4, 10, 200);

insert into PACHET values(1007, 905, 601, 815, 1, 6, 130);

insert into PACHET values(1008, 906, 600, 802, 10, 70, 1000);

insert into PACHET values(1009, 906, 601, 804, 1, 5, 200);

insert into PACHET values(1010, 907, 601, 807, 1, 6, 230);

insert into PACHET values(1011, 908, 602, 806, 3, 20, 600);

insert into PACHET values(1012, 909, 602, 807, 4, 25, 900);

insert into PACHET values(1013, 910, 604, 801, 26, 20, 100);

insert into PACHET values(1014, 911, 604, 808, 22, 100, 2500);

insert into PACHET values(1015, 912, 600, 810, 2, 5, 80);

insert into PACHET values(1016, 912, 604, 811, 1, 5, 60);

insert into PACHET values(1017, 913, 604, 813, 20, 34, 208);

insert into PACHET values(1018, 914, 600, 806, 6, 50, 650);

insert into PACHET values(1019, 914, 602, 801, 10, 13, 80);

insert into PACHET values(1020, 914, 601, 810, 40, 50, 3000);

insert into PACHET values(1021, 915, 601, 804, 4, 14, 430);

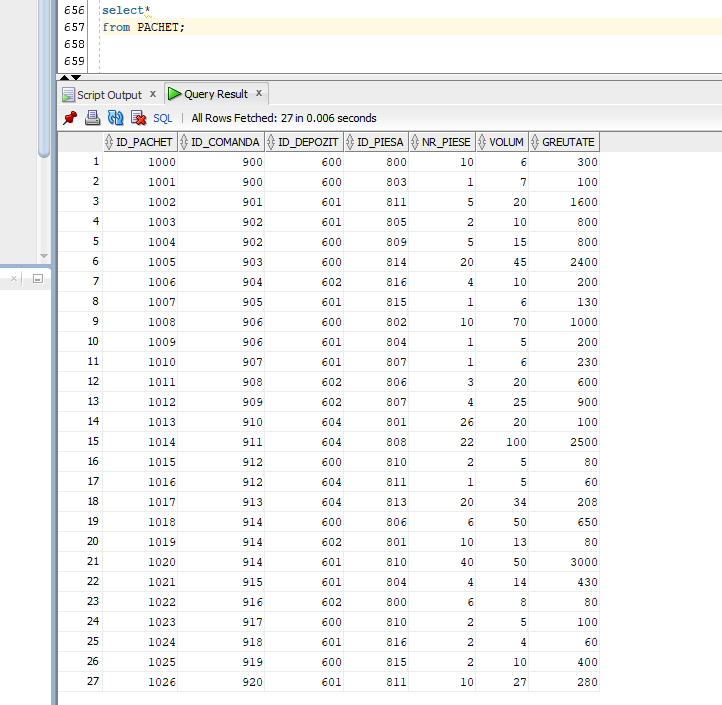
insert into PACHET values(1022, 916, 602, 800, 6, 8, 80);

insert into PACHET values(1023, 917, 600, 810, 2, 5, 100);

insert into PACHET values(1024, 918, 601, 816, 2, 4, 60);

insert into PACHET values(1025, 919, 600, 815, 2, 10, 400);

insert into PACHET values(1026, 920, 601, 811, 10, 27, 280);



# 12.Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în

# ansamblul lor, următoarele elemente:

# a) subcereri sincronizate în care intervin cel puțin 3 tabele

# b) subcereri nesincronizate în clauza FROM

# c) grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate

# (în clauza de HAVING) în care intervin cel puțin 3 tabele (in cadrul aceleiași cereri)

# d) ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)

# e) utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice,

# a cel puțin unei expresii CASE

# f) utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

# Observație: Într-o cerere se vor regăsi mai multe elemente dintre cele enumerate mai sus, astfel

# încât cele 5 cereri să le cuprindă pe toate.

1. Să se afișeze id-ul tuturor comenzilor preluate cu cel puțin 9 luni înainte de 01.01.2018, de un dispecer care nu este angajat in ultima zi a vreunei luni și livrate, apoi, de un angajat/șofer care a transportat cel puțin o comandă mai scumpă de  800 de lei cu un autoturism de tip VAN.

select t.id\_comanda

from transporta t, comanda c, angajat a

where (t.id\_comanda=c.id\_comanda and c.id\_angajat=a.id\_angajat)

and (months\_between('01-Jan-2018',c.data\_comanda)>=9 and a.data\_angajare<LAST\_DAY(a.data\_angajare))

and exists

(

select 5

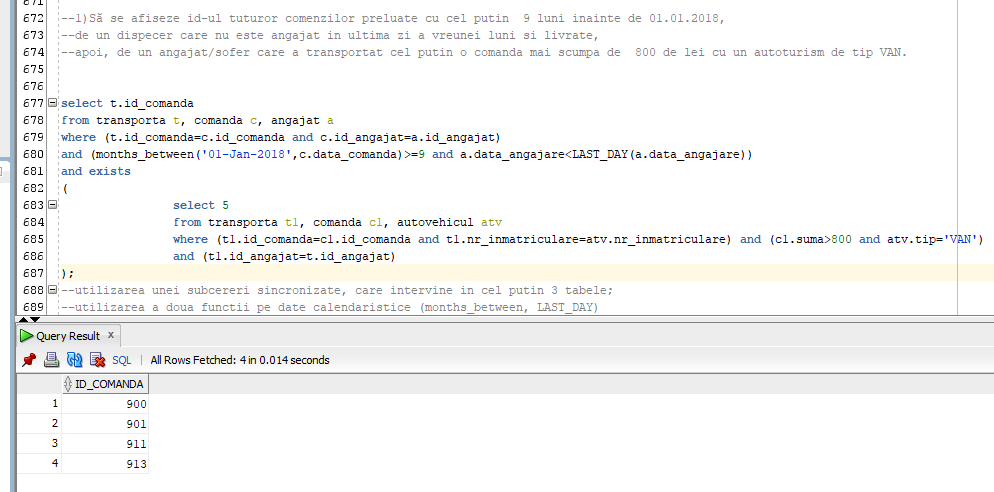
from transporta t1, comanda c1, autovehicul atv

where (t1.id\_comanda=c1.id\_comanda and t1.nr\_inmatriculare=atv.nr\_inmatriculare) and (c1.suma>800 and atv.tip='VAN') and (t1.id\_angajat=t.id\_angajat)

);

--utilizarea unei subcereri sincronizate, care intervine in cel putin 3 tabele;

--utilizarea a doua functii pe date calendaristice (months\_between, LAST\_DAY)



1. Pentru fiecare comandă care conține cel puțin un scaun, dar numărul total de scaune este mai mic decât cel maxim (posibil într-o comandă), să se afișeze id-ul comenzii, media volumelor pachetelor din ea, precum și media greutăților. Să se ordoneze rezultatele dupa greutate.

select aux.cod, NVL((avg(p2.volum)),0) Volum, NVL((avg(p2.greutate)),0) Greutate

from (--aici obtinem comenzile care contin scaune

select c.id\_comanda cod

from comanda c, pachet p, piesa pi

where c.id\_comanda=p.id\_comanda and p.id\_piesa=pi.id\_piesa and lower(substr(pi.denumire, 1, 5))='scaun'

group by c.id\_comanda

having sum(p.nr\_piese)<( --aici obtinem numarul maxim de scaune dintr-o comanda

select max(sum(p1.nr\_piese))

from comanda c1, pachet p1, piesa pi1

where c1.id\_comanda=p1.id\_comanda and p1.id\_piesa=pi1.id\_piesa and lower(substr(pi1.denumire, 1, 5))='scaun'

group by c1.id\_comanda

)

)aux, pachet p2

where p2.id\_comanda=aux.cod

group by aux.cod

order by 3;

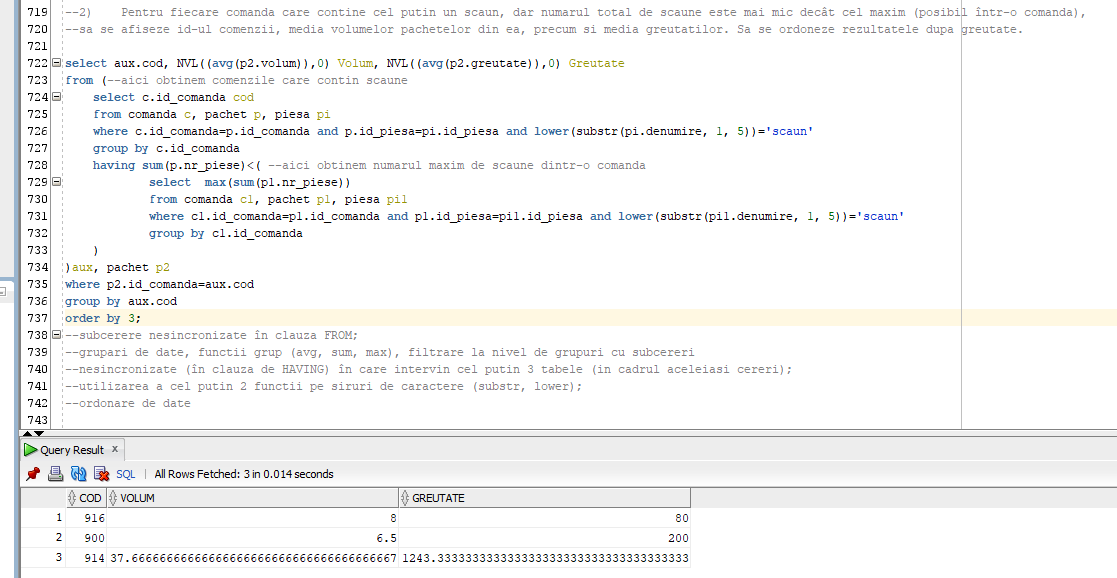
--subcerere nesincronizate în clauza FROM;

--grupari de date, functii grup (avg, sum, max), filtrare la nivel de grupuri cu subcereri

--nesincronizate (în clauza de HAVING) în care intervin cel putin 3 tabele (in cadrul aceleiasi cereri);

--utilizarea a cel putin 2 functii pe siruri de caractere (substr, lower);

--ordonare de date



3) Să se afișeze pentru fiecare comandă volumul total al pachetelor din fiecare depozit (se știe că id-urile depozitelor sunt 600, 601, 602, 603, 604), precum și volumul total al comenzii.

Să se ordoneze rezultatele după volumul total al comenzii.

select id\_comanda, NVL(SUM(DECODE(id\_depozit, 600, volum)),0) Dep600,

NVL(SUM(DECODE(id\_depozit, 601, volum)),0) Dep601, NVL(SUM(DECODE(id\_depozit, 602, volum)),0) Dep602,

NVL(SUM(DECODE(id\_depozit, 603, volum)),0) Dep603, NVL(SUM(DECODE(id\_depozit, 604, volum)),0) Dep604,

NVL(SUM(volum),0) Total

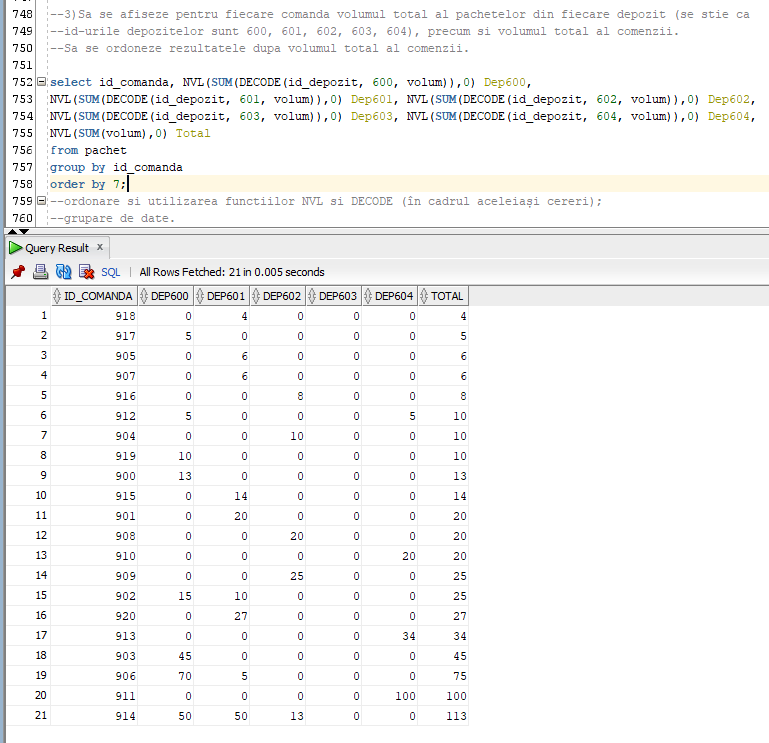
from pachet

group by id\_comanda

order by 7;

--ordonare si utilizarea functiilor NVL si DECODE (în cadrul aceleiași cereri);

--grupare de date.



4) Să afișeze o distribuire a comenzilor în tipurile de autovehicule corespunzătoare pentru transportul lor.

Evident, distribuirea se va face după greutatea și volumul comenzilor (adică, suma greutăților și volumelor pachetelor din comandă).

Afișarea va consta în id-ul comenzii, volumul și greutatea acesteia și un șir (de caractere) format din tipurile de autovehicule care pot livra comandă.

Aceste tipuri (de care dispune firma), alături de dimensiunile lor sunt:

( VAN, 13m3, 1.5t),  ( AVIA, 40m3, 3.5t),  ( TIR, 90m3, 23t), ( TIR\_rem, 120m3, 30t).

select aux.Comanda, aux.Volum, aux.Greutate,

case

when (Volum<=13 and Greutate<=1500) then 'VAN, AVIA, TIR, TIR\_rem'

when (Volum>13 or Greutate>1500) and (Volum<=40 and Greutate<=3500) then 'AVIA, TIR, TIR\_rem'

when (Volum>40 or Greutate>3500) and (Volum<=90 and Greutate<=23000) then 'TIR, TIR\_rem'

else 'TIR\_rem'

end as DISTRIBUIRE\_COMANDA

from (--OBS: nu puteam face direct gruparea in cererea principala, caci in select

--apare si coloana derivata din case, dar care nu se gaseste in clauza group by, astfel avem nevoie de subcerere!

select id\_comanda Comanda , sum(NVL(volum,0)) Volum, sum(NVL(greutate,0)) Greutate

from pachet

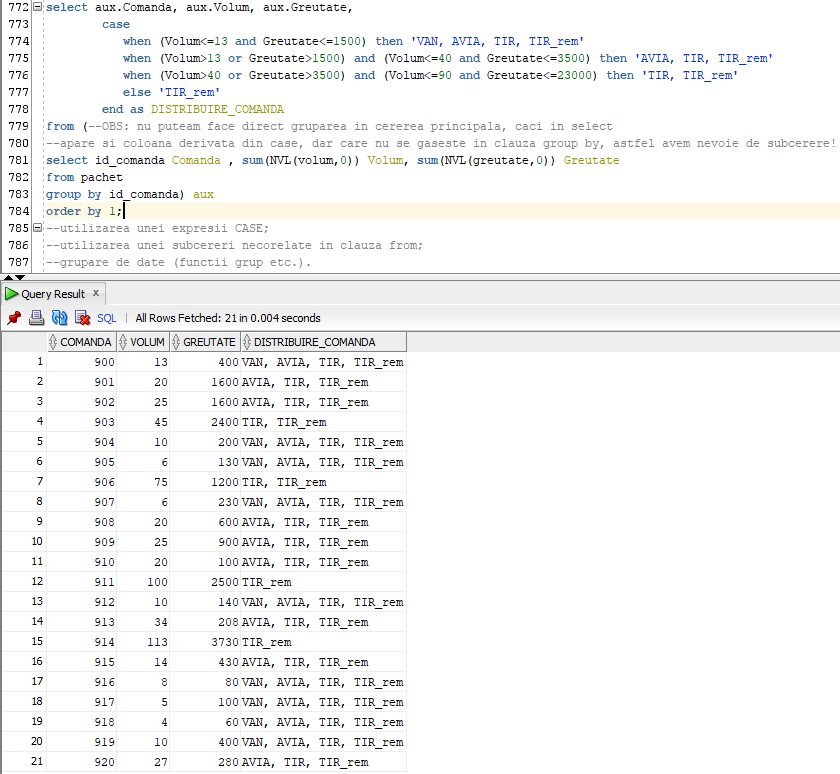
group by id\_comanda) aux

order by 1;

--utilizarea unei expresii CASE;

--utilizarea unei subcereri necorelate in clauza from;

--grupare de date (functii grup etc.).



1. Să se afişeze codul, numele şi prenumele (pe aceeaşi coloană) şi data angajării, ale subalternilor subordonaţi direct lui Catalin Butculescu care au cea mai mare vechime.

with subalt as(--aici gasesc subalternii

select a2.id\_angajat cod , a2.nume||' '||a2.prenume nume, a2.data\_angajare data\_ang

from angajat a, angajat a2

where a.id\_angajat=a2.id\_sef

and upper(a.nume)='BUTCULESCU' and upper(a.prenume)='CATALIN'

)

select cod, nume, data\_ang

from subalt

where data\_ang=

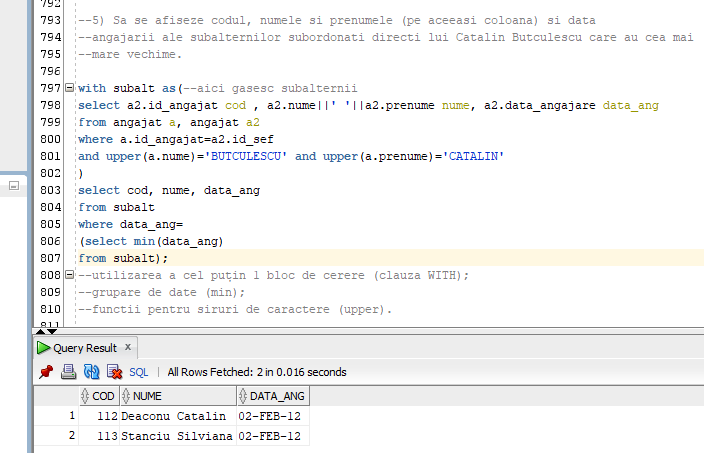
(select min(data\_ang)

from subalt);

--utilizarea a cel puțin 1 bloc de cerere (clauza WITH);

--grupare de date (min);

--functii pentru siruri de caractere (upper).



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

1. Să se actualizeze la 2500 lei salariile tuturor șoferilor care au efectuat mai puțin de 5 transporturi și care au fost angajați după toți dispecerii.

update angajat

set salariu=2500

where id\_angajat in

(

select a.id\_angajat

from transporta t, angajat a

where t.id\_angajat=a.id\_angajat and a.data\_angajare>any(

select a2.data\_angajare

from comanda c, angajat a2

where c.id\_angajat=a2.id\_angajat

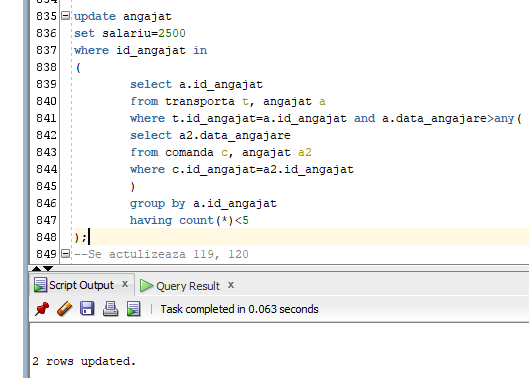
)

group by a.id\_angajat

having count(\*)<5

);

--Se actulizeaza 119, 120



1. Să se șteargă (din entitatea LOCATIE) fiecare locație care nu este a vreunui client sau a vreunui depozit.

delete from locatie

where id\_locatie in

(

select id\_locatie

from locatie

minus

(select distinct id\_locatie

from client\_firma

union

select distinct id\_locatie

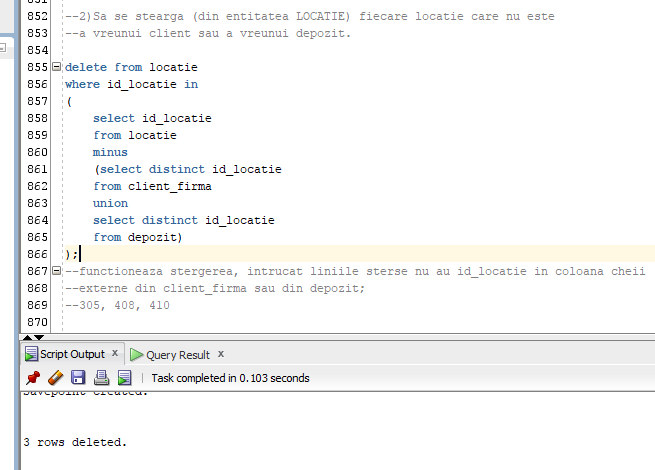
from depozit)

);

--functioneaza stergerea, intrucat liniile sterse nu au id\_locatie in coloana cheii

--externe din client\_firma sau din depozit;

--305, 408, 410



1. Să se actualizeze capacitatea fiecărui depozit în care a fost depozitat un volum mai mic de 80 m3 (de pachete). Capacitățile acestora de stocare se vor micșora cu 25%.

update depozit

set capacitate=capacitate\*0.75

where id\_depozit in(

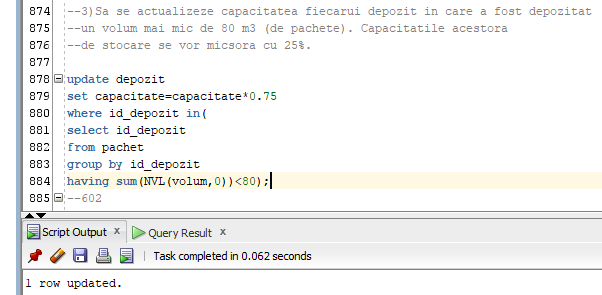
select id\_depozit

from pachet

group by id\_depozit

having sum(NVL(volum,0))<80);

--602



.

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

--Vizualizare complexa cu join intre 2 tabele, care contine

--id-ul, numele, prenumele, id-ul sefului, id-ul si denumirea

--job-lui pentru angajatii de pe job-urile 101, 102.

create or replace view angajat\_job\_firma as

(

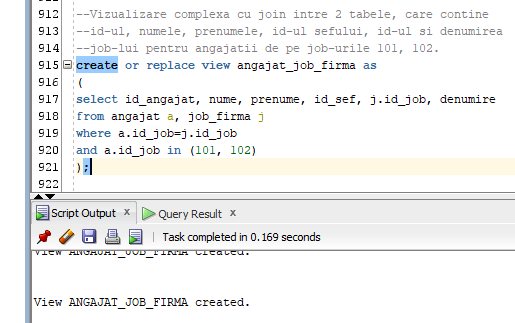
select id\_angajat, nume, prenume, id\_sef, j.id\_job, denumire

from angajat a, job\_firma j

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

and a.id\_job in (101, 102)

);



--Operatie LMD permisa--select (nu modifica niciun tabel,

--vizualizarea nu contine grupare ori alt element

--care sa impiedice LMD):

select \*

from angajat\_job\_firma

where id\_sef=

(

select id\_angajat

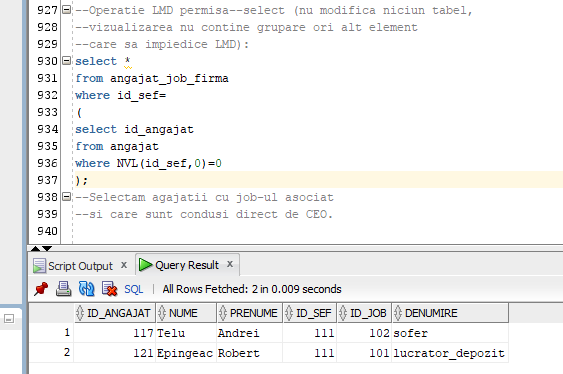
from angajat

where NVL(id\_sef,0)=0

);

--Selectam agajatii cu job-ul asociat

--si care sunt condusi direct de CEO.



--Operatie LMD nepermisa (in majoritatea cazurilor)--insert

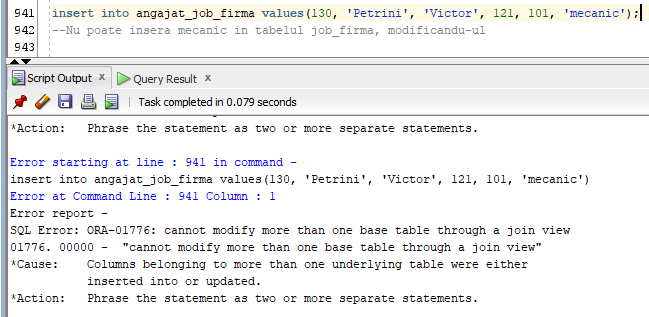
--(tabelul angajat este key preserved, celalalt nu, iar id\_job

--din tabelul angajat are constrangerea sa nu fie null si

--este cheie externa):

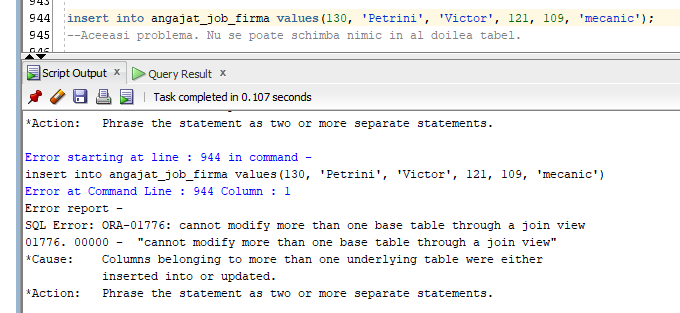
insert into angajat\_job\_firma values(130, 'Petrini', 'Victor', 121, 101, 'mecanic');

--Nu poate insera mecanic in tabelul job\_firma, modificandu-l



insert into angajat\_job\_firma values(130, 'Petrini', 'Victor', 121, 109, 'mecanic');

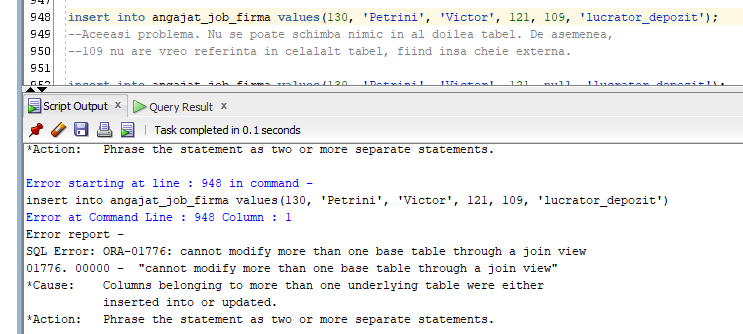
--Aceeasi problema. Nu se poate schimba nimic in al doilea tabel.



insert into angajat\_job\_firma values(130, 'Petrini', 'Victor', 121, 109, 'lucrator\_depozit');

--Aceeasi problema. Nu se poate schimba nimic in al doilea tabel. De asemenea,

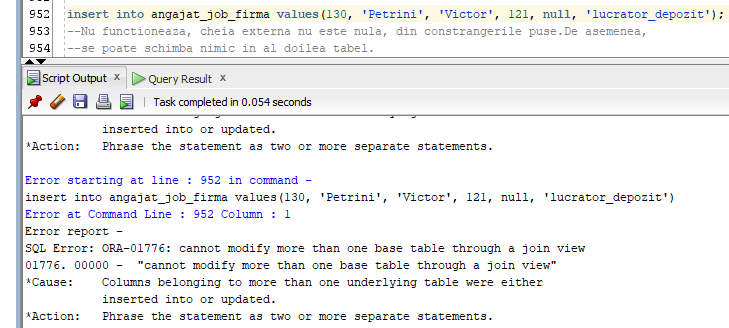
--109 nu are vreo referinta in celalalt tabel, fiind insa cheie externa.



insert into angajat\_job\_firma values(130, 'Petrini', 'Victor', 121, null, 'lucrator\_depozit');

--Nu functioneaza, cheia externa nu este nula, din constrangerile puse.De asemenea,

--se poate schimba nimic in al doilea tabel.



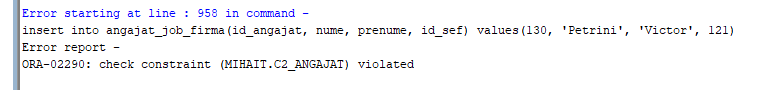
insert into angajat\_job\_firma values(130, 'Petrini', 'Victor', 121, 101, 'lucrator\_depozit');

insert into angajat\_job\_firma(id\_angajat, nume, prenume, id\_sef, id\_job) values(130, 'Petrini', 'Victor', 121, 101);

--Nu va functiona nici in cazurile de mai sus, caci tot se 'utilizeaza' tabela job\_firma.

insert into angajat\_job\_firma(id\_angajat, nume, prenume, id\_sef) values(130, 'Petrini', 'Victor', 121);

--Ar fi mers asa daca nu exista constrangerea din angajat de not-null pe id\_job.



.

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

1. Pentru fiecare comandă dată (nu se spune că și transportată), să se afișeze id-ul ei, numele clientului, orașul unde se află clientul, autoturismul cu care a fost transportată și numele angajatului care a transportat-o.

select c.id\_comanda, cln.nume, l.oras, t.nr\_inmatriculare, a.nume

from comanda c join client\_firma cln on c.id\_client=cln.id\_client

join locatie l on cln.id\_locatie=l.id\_locatie

left outer join transporta t on c.id\_comanda=t.id\_comanda

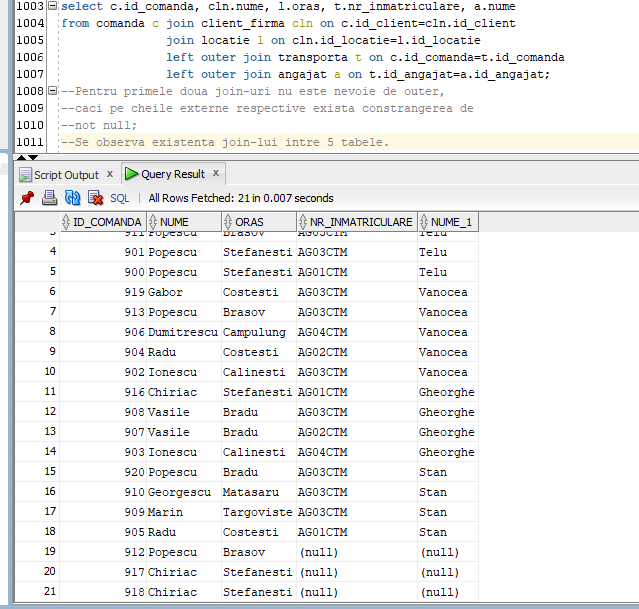
left outer join angajat a on t.id\_angajat=a.id\_angajat;

--Pentru primele doua join-uri nu este nevoie de outer,

--caci pe cheile externe respective exista constrangerea de

--not null;

--Se observa existenta join-lui intre 5 tabele.



1. Găsiți toți șoferii (id-ul lor) care au transportat comenzi cu toate autovehiculele de tip VAN.

select distinct t1.id\_angajat

from transporta t1

where not exists

(

select a.nr\_inmatriculare

from autovehicul a

where a.tip='VAN'

and not exists

(

select t2.nr\_inmatriculare

from transporta t2

where t2.nr\_inmatriculare=a.nr\_inmatriculare

and t2.id\_angajat=t1.id\_angajat

)

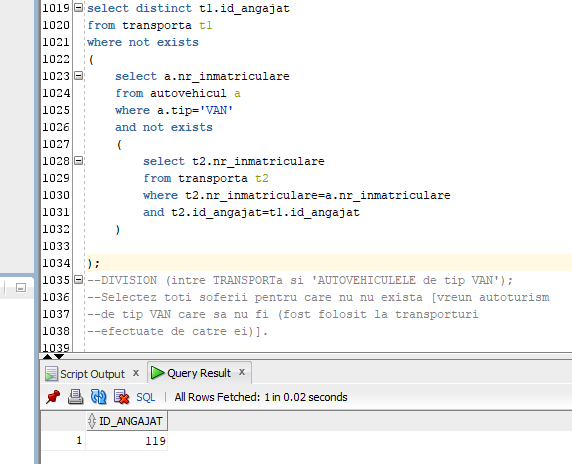
);

--DIVISION (intre TRANSPORTa si 'AUTOVEHICULELE de tip VAN');

--Selectez toti soferii pentru care nu nu exista [vreun autoturism

--de tip VAN care sa nu fi (fost folosit la transporturi

--efectuate de catre ei)].



1. Să se afișeze numărul de înmatriculare și greutatea totală a comenzilor transportate pentru autovehiculele care au transportat cele mai mari două cantități totale de mobilă.

select a.nr\_inmatriculare, sum(NVL(p.greutate,0))

from autovehicul a, transporta t, comanda c, pachet p

where a.nr\_inmatriculare=t.nr\_inmatriculare and t.id\_comanda=c.id\_comanda

and c.id\_comanda=p.id\_comanda

group by a.nr\_inmatriculare

having sum(NVL(p.greutate,0)) in

( select S

from

( select sum(NVL(p1.greutate,0)) S

from autovehicul a1, transporta t1, comanda c1, pachet p1

where a1.nr\_inmatriculare=t1.nr\_inmatriculare and t1.id\_comanda=c1.id\_comanda

and c1.id\_comanda=p1.id\_comanda

group by a1.nr\_inmatriculare

order by 1 desc

)

where rownum < 3

)

order by 2;

--Cerere care implementeaza analiza TOP-N.

