**Facultatea de Matematică și Informatică, Universitatea din București**

**PROIECT BAZE DE DATE**

**-Gestionarea datelor unei firme de productie-**

**Student: Tosa Denis Andrei**

**An universitar: 2022-2023**

**Grupa: 212**

-cuprins –

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.
2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.
3. Descrierea entităților, incluzând precizarea cheii primare.
4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.
5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.
6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.
7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.
8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.
9. Realizarea normalizării până la forma normală 3 (FN1-FN3).
10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).
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).
12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:

• subcereri sincronizate în care intervin cel puțin 3 tabele

• subcereri nesincronizate în clauza FROM

• grupări de date cu subcereri nesincronizate in care intervin cel putin 3 tabele, funcții grup, filtrare la nivel de grupuri (in cadrul aceleiasi cereri)

• ordonări si utilizarea funcțiilor NVL și DECODE (in cadrul aceleiasi cereri)

• utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE

• 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. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri. 2
2. Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.
3. 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.

Baza de date ERP-CRM

(enterprise resource planning-crm)

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

Baza de date “Enterprise resource planning” pe care am configurat-o ajuta la coordonarea unei firme din **Australia**. Firma isi va procura stocul de materiale pe care le poate comanda de la distribuitorii sai. Acestea pot avea “prelucrari de material” ( transportare, lustruire, depozitare etc.) si au ca scop confectionarea produselor pentru oameni cu dizabilitati, facute la comanda pentru nevoile acestora. De asemenea produsul confectionat la comanda poate insuma la randul lui “prelucrare de produs” si echipamentele utilizate la producerea acestora. Produsele finale se vor regasi pe factura, odata cu datele comenzii, datele clientului si datele firmei.

1. **Prezentarea constrangerilor (restrictii, reguli) impuse asupra modelului:**

* Atat emailul clientului dar si al firmei si al distribuitorului trebuie sa contina simbolul “@” si un domeniu.
* Toate numele clientilor trebuie sa contine doar litere
* Comisionul produselor trebuie sa fie mai mare sau egal decat 0
* GST-ul va fi egal cu 10% pentru persoanele care nu au dizabilitati si 0 pentru cei cu dizabilitati
* Data de livrare trebuie sa fie mai mare data intorcmirii facturii
* Materialele vor detine un cod de bare(SKU) care trebuie sa fie unic.
* Adresa clientilor trebuie sa fie valida, completa si trebuie sa detina litere si numere
* Profitul (comisionul) per produs trebuie sa fie minim 20%.
* Diferenta intre data crearii si data de livrare pe factura nu poate fi mai mica de 3 zile lucratoare
* Toate datele legate de pret trebuie sa fie mai mari sau egale decat 0.

1. Descrierea entitatilor, incluzand precizarea cheii primare.
2. Entitatea “Material” (cheie primara ID\_material) reprezinta inventarul de material al firmei si se rezuma prin date precum: nume, pret, cantitate, cod de bare
3. Entitatea “PrelucrareMaterial” (cheie primara ID\_PM)(cheie straina ID\_material) reprezinta costurile de productie/prelucrare pentru material si se rezuma prin date precum: nume, cost
4. Entitatea “ProdusPersonalizat” (cheie primara ID\_produspersonalizat) reprezinta produsul final si se rezuma prin date precum: nume, pret, commission, garantie
5. Entitatea “PrelucrareProdusPersonalizat” (cheie primara ID\_prelucrare) (cheie straina ID\_produspersonalizat) reprezinta costurile de productie/prelucrare pentru produsul final si se rezuma prin date precum nume, cost
6. Entitatea “Echipament” (cheie primara ID\_echipament) reprezinta totalitatea echipamentelor detinute de firma, ce ajuta la prelucrarea materialelor si a produselor personalizate. Acestea contin date precum: categorie, nume, model, pret, cod\_bare, data\_cumparare, descriere, data\_ultm\_revizie, depreciere
7. Entitatea “Revizie” (cheie primara ID\_revizie, cheie straina ID\_echipament) reprezinta mentenantele echipamentelor si contine: data, cost, comentarii
8. Entitatea “Furnizor” (cheie primara ID\_furnizor) reprezinta distribuitorul firmei si acesta poate fi alocat pe o comanda. Acesta contine atribute precum: nume, adresa, contact, email
9. Entitatea “Comanda” (cheie primara ID\_comanda, cheie straina “ID\_furnizor) aceasta reprezinta comanda facuta de firma catre furnizor pentru a achizitiona materiale noi. Comanda contine atribute precum : adresa , transport, deposit, tip\_plata, comentarii
10. Entitatea “Factura” (cheie primara ID\_factura, cheie straina ID\_client) reprezinta bonul dat de catre firma catre client si contine urmatoarele atribute: Transport, data, data\_livrare,GST, trimis, persoana\_fizica, comentarii, adresaFirma, contactFirma, emailFirma, contBancarFirma
11. Entiatea “Client” (cheie primara ID\_Client) reprezinta clientii firmei si contine atributele : nume, contact, adresa, email
12. Descrierea relatiilor, incluzand precizarea cardinalitatii acestora.

* Comanda are asociat un furnizor = relatie care leaga entitatile Comanda si Furnizor. Aceasta are cardinalitate obligatorie 1:0 (comanda trebuie sa aiba asociat un furnizor dar un funizor nu trebuie sa aiba asociata o comanda). Cardinalitatea optional este 1:M (un furnizor poate indeplini mai multe comenzi, dar comanda poate fi facuta doar de un furnizor)
* Comanda contine materiale = relatie care leaga entitatile Comanda si Material. Aceasta are cardinalitate obligatorie 1:0 (o comanda trebuie sa aiba macar un material, dar un material nu trebuie sa fie comandat de la furnizor). Cardinalitata optinala este M:M doarece pe o comanda pot fi mai multe materiale, si acelasi material se poate regasi in mai multe comenzi.
* Material contine PrelucrareMaterial = relatie care leaga entitatile Material si PrelucrareMaterial. Aceasta are cardinalitate obligatorie 1:0 (o prelucrare de material trebuie sa contina un material ,insa un material nu trebuie sa contina prelucrare de material). Cardinalitate optionala 1:M ( un material poate contine mai multe prelucrari de material ,dar o prelucrare de material poate sa aiba un material)
* Prelucrare material facuta cu echipament = relatie care leaga entitatile PrelucrareMaterial si Echipament. Aceasta are cardinalitate obligatorie 0:0 ( o prelucrare de material nu trebuie sa fie facuta de un echipament , iar un echipament nu trebuie sa faca o prelucrare de material). Cardinalitatea optionala este 1:M (un echipament poate face mai multe prelucrari de material, iar o prelucrare de material poate fi facuta de un echipament)
* ProdusPersonalizat contine material = relatie care leaga entitatile ProdusPersonalizat si Material. Aceasta are cardinalitate obligatorie 1:0 (un produs personalizat trebuie sa contina un material dar un material nu trebuie sa fie continut de produs personalizat). Cardinalitatea optional este M:M (un produs personalizat poate contine mai multe materiale , iar mai multe materiale pot fi continute de mai multe produse personalizate)
* ProdusPersonalizat contine prelucrare produs personalizat = relatie care leaga entitatile ProdusPersonalizat si PrelucrareProdusPersonalizat. Aceasta are cardinalitate 1:0 (un produs personalizat nu trebuie sa contina o prelucrare de produs personalizat, dar o prelucrare de produs personalizat trebuie sa contina un produs personlizat). Cardinalitatea optionala este 1:M deoarece un produs personalizat poate contine mai multe prelucrari de produs personalizat iar o prelucrare de produs personalizare poate fi continuta de macar un produs.
* Prelucrare de produs personalizat este facuta de echipament = relatie care leaga entitatile PrelucrareProdusPersonalizat si Echipament. Aceasta are cardinalitate obligatorie 0:0 (o prelucrare nu trebuie sa fie facuta de un echipament iar un echipament nu trebuie sa faca o prelucrare de produs personalizat). Cardinalitatea optionala este de M:1 deoarece o prelucrare de produs personalizat poate fi facuta de un echipament iar un echipament poate face mai multe prelucrari de produs personalizat.
* Factura contine produs personalizat (not. Produs) = relatie care leaga entitatile Factura si ProdusPersonalizat. Aceasta are o cardinalitate obligatorie 1:0 (o factura trebuie sa contina un produs dar un produs nu trebuie sa fie continut de o factura ca sa existe). Cardinalitatea optionala este M:M deoarece o factura poate contine mai multe produse iar aceste produs pot fi continute pe mai multe facturi.
* Factura are Client = relatie care leaga entitatile Factura si Client. Aceasta are o cardinalitate obligatorie de 1:0 deoarece o factura trebuie sa aiba un client dar un client nu trebuie sa aiba o factura pentru a exista. Cardinalitatea optionala este de M:1 deoarece un client poate avea mai multe facturi dar o factura poate avea doar un client.
* Echipamentul are revizie = relatie care leaga entitatile Echipament si Revizie. Aceasta are o legatura obligatorie de 0:1 pentru ca un echipament nu trebuie sa aiba o revizie dar o revizie trebuie sa aiba un echipament. Cardinalitatea optionala este 1:M deoarece un echipament poate avea mai multe revizii iar o revizie poate avea un echipament.

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

NN = Not Null

Entitate Furnizor:

1. ID\_furnizor NUMBER, lungime 10, PK, NN
2. Nume VARCHAR2, lungime 50 ,NN, unic
3. Adresa VARCHAR2, lungime 100, NN
4. Contact VARCHAR2,lungime 10, NN
5. Email VARCHAR2,lungime 50, NN, unic

Entitate Comanda:

1. ID\_comanda NUMBER, lungime 10, PK, NN
2. ID\_furnizor NUMBER, lungime 10 , FK ,NN
3. Adresa VARCHAR2,lungime 100, NN
4. Transport NUMBER,lungime 10,decimale 2 , NN
5. Deposit VARCHAR2,lungime 50,
6. tip\_plata VARCHAR2,lungime 50, NN
7. comentarii VARCHAR2,lungime, 200

Entitate ComandaMaterial:

1. ID\_comanda\_material NUMBER, lungime 10, PK,NN
2. ID\_comanda NUMBER,lungime 10, FK, NN
3. ID\_material NUMBER,lungime 10, FK, NN
4. Pret\_cumparare NUMBER,lungime 10,decimale 2,NN
5. Cantitate NUMBER,lungime 10,NN

Entitate Material:

1. ID\_material – identificator unic, cheie primara, NUMBER,lungime 10 , NN
2. Nume VARCHAR2,lungime 50, NN
3. Pret NUMBER, lungime (10,2),NN valoare implicita 0
4. Cantitate NUMBER, lungime(10),NN
5. Cod\_bare VARCHAR2,lungime 50, NN

Entitate PrelucrareMaterial:

1. ID\_PM –identificator unic, cheie primara, number,lungime 10, NN
2. Material\_id, FK, NUMBER, lungime 10, NN
3. Nume VARCHAR2,lungime 20 , NN
4. Cost NUMBER, lungime(10,2),NN, valoare implicita 0

Entitate EchipamentMaterial:

1. ID\_EM identificator unic, cheie primara, number,lungime 10, NN
2. ID\_PM, FK, NUMBER, lungime 10, NN
3. ID\_echipament, FK, NUMBER, lungime 10, NN

Entitate MaterialProdusPersonalizat:

1. ID\_MPP identificator unic, cheie primara, number,lungime 10, NN
2. ID\_material , FK, NUMBER, lungime 10, NN
3. ID\_produspersonalizat , FK, NUMBER, lungime 10, NN

Entitate ProdusPersonalizat:

1. Id\_produspersonalizat identificator unic, cheie primara, number,lungime 10, NN
2. Nume VARCHAR2,lungime 50, NN
3. Pret NUMBER, lungime (10,2), NN valoare implicita 0
4. Commission NUMBER, lungime (10,2), NN
5. Garantie NUMBER, lungime (10), NN

Entitate PrelucrareProdusPersonalizat:

1. ID\_prelucrare (cheie primara) NUMBER, lungime 10, NN
2. Id\_produspersonalizat (cheie straina) NUMBER, lungime 10, NN
3. Nume VARCHAR2,lungime 50, NN
4. Cost NUMBER, lungime 10, NN

Entitate EchipamentMaterial:

1. ID\_EM(cheie primara) NUMBER, lungime 10 ,NN
2. ID\_prelucrare(cheie straina) NUMBER, lungime 10 ,NN
3. ID\_echipament(cheie straina) NUMBER, lungime 10 ,NN

Entitate Echipament:

1. ID\_echipament PK, NUMBER, lungime 10,NN
2. Categorie VARCHAR2,lungime 50, NN
3. Nume VARCHAR2,lungime 40, NN
4. Model VARCHAR2,lungime 40, NN
5. Pret NUMBER, lungime (10,2), NN
6. Cod VARCHAR2,lungime 30, NN
7. Data\_cumparare DATE
8. Descriere VARCHAR2,lungime 200, NN
9. Data\_ultima\_revizie DATE
10. Depreciere NUMBER(10,2)

Entitate Revizie:

1. ID\_revizie PK, NUMBER, lungime 10, NN
2. ID\_echipament FK, NUMBER, lungime 10, NN
3. Data date
4. Cost NUMBER, lungime(10,2), val. implicita 0
5. Comentarii VARCHAR2, lungime 100

Entitate ProdusFactura:

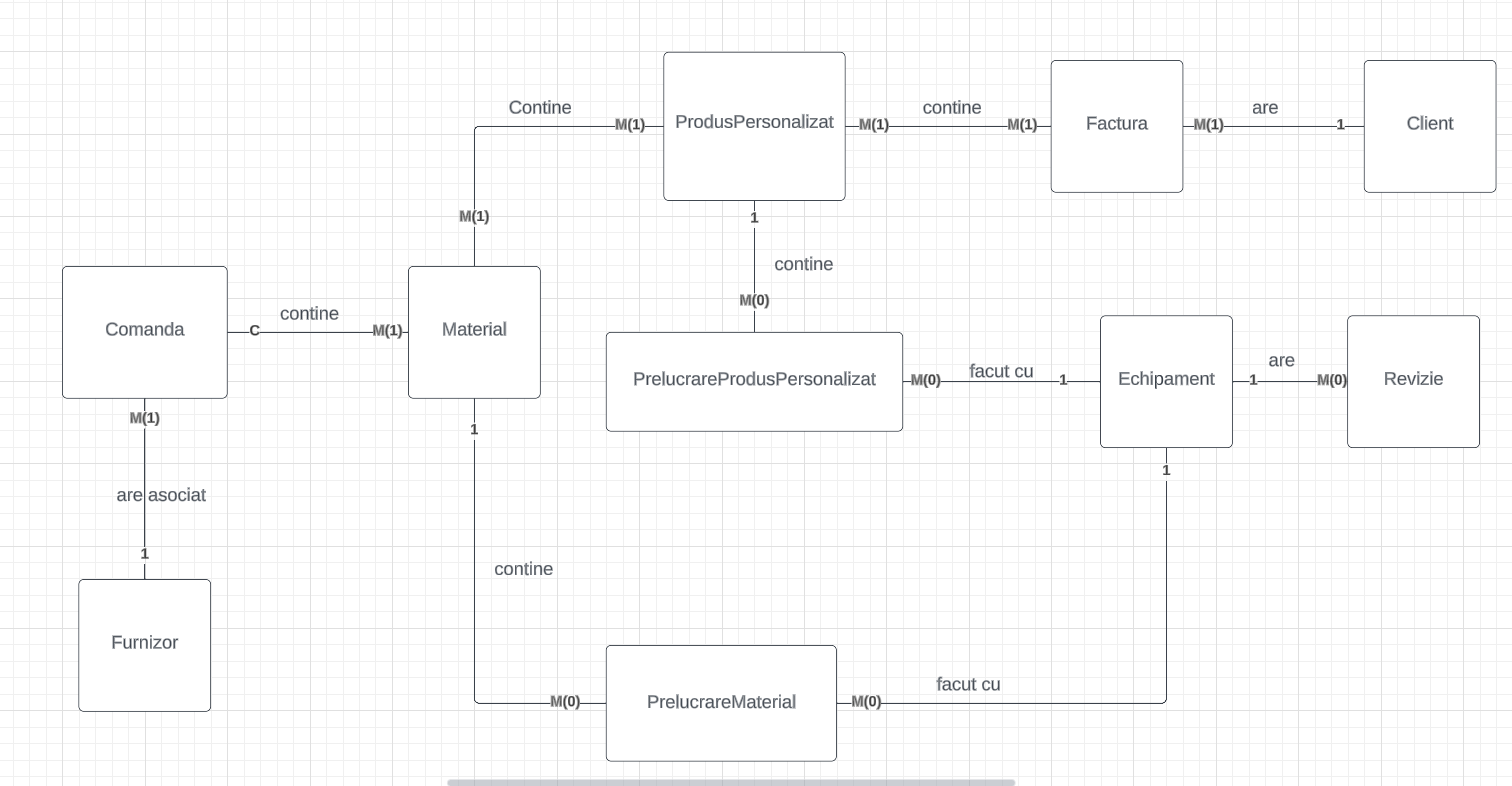
1. ID\_produsfactura PK, NUMBER, lungime 10, NN
2. ID\_produspersonalizat FK, NUMBER, lungime 10, NN
3. ID\_Factura FK, NUMBER, lungime 10, NN

Entitate Factura:

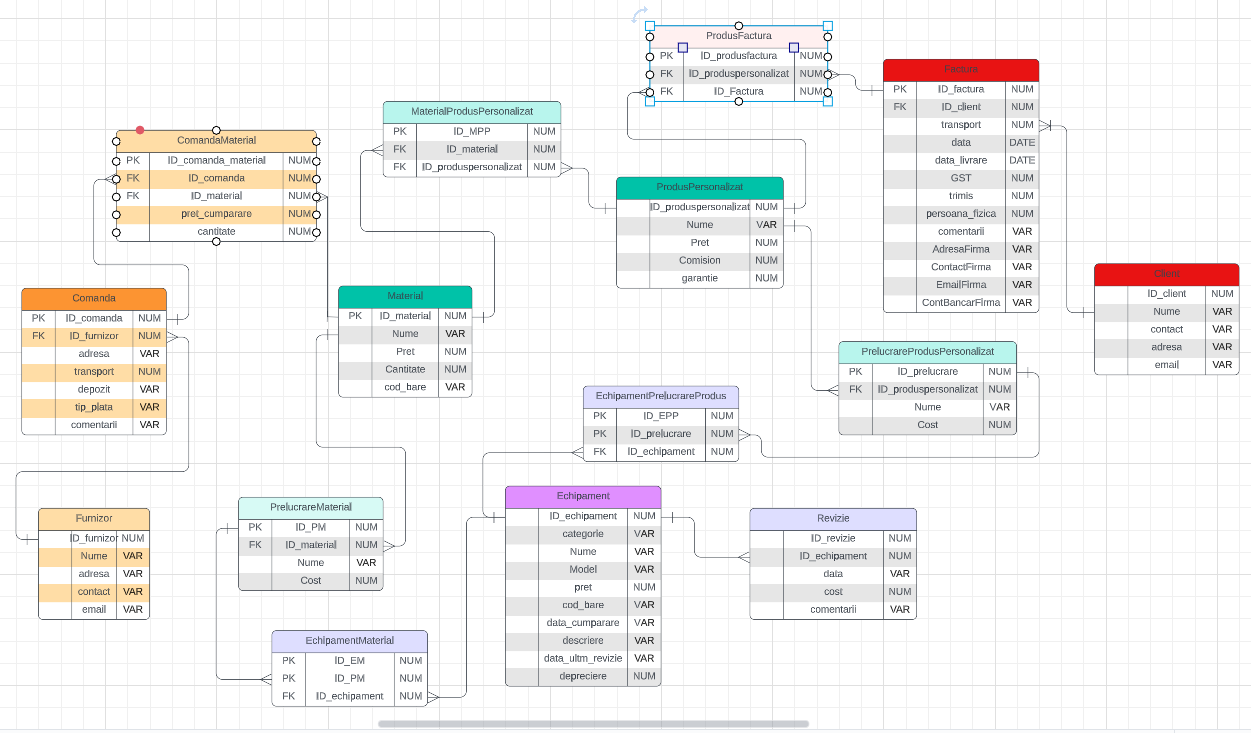
1. ID\_Factura PK, NUMBER, lungime 10, NN
2. ID\_client FK, NUMBER, lungime 10, NN
3. Transport NUMBER, lungime(10,2), default 0, NN
4. Data date
5. Data\_livrare date
6. GST NUMBER, lungime 2,default 10, NN
7. Trimis NUMBER, lungime 1, default 0
8. Persoana\_fizica NUMBER, lungime 1 , default 1
9. Comentarii VARCHAR2,lungime 200, NN
10. AdresaFirma VARCHAR2,lungime 100, NN
11. ContactFirma VARCHAR2,lungime 10, NN
12. EmailFirma VARCHAR2,lungime 50, NN
13. ContBancarFirma VARCHAR2 50,lungime, NN

Entitate Client:

1. ID\_client PK, NUMBER 10, NN
2. Nume VARCHAR2,lungime 50, NN
3. Telefon VARCHAR2,lungime 20, NN, unic
4. Adresa VARCHAR2,lungime 100, NN
5. Email VARCHAR2,lungime 50, NN, unic
6. Realizarea diagramei entitate-relatie corespunzatoare descrierii de la punctele 3-5



1. Realizarea diagramei conceptual corespunzatoare diagramei entitate-realtie proiectate la punctual 6. Diagrama conceptual obtinuta trebuie sa contina minimum 6 tabele, care dintre care cel putin un table asociativ.



1. Enumerarea schemelor relationale corespunzatoare diagramei conceptual proiectate la punctual 7.

* Furnizor (ID\_furnizor#,nume,adresa,contact,email)
* Comanda(ID\_comanda#,ID\_furnizor#,adresa,transport,deposit,tip\_plata,comentarii)
* ComandaMaterial(ID\_comanda\_material#,ID\_comanda#,ID\_material#, pret\_cumparare,cantitate)
* Material(ID\_material#,nume,pret,cantitate,cod\_bare)
* PrelucrareMaterial(ID\_PM#,ID\_material#,nume,cost)
* MaterialProdusPersonalizat(ID\_MPP#,ID\_material#,ID\_produspersonalizat#)
* ProdusPersonalizat(ID\_produspersonalizat#,nume,pret,commission,garantie)
* PrelucrareProdusPersonalizat(ID\_prelucrare#,ID\_personalizat,nume,cost)
* EchipamentPrelucrareProdus(ID\_EPP#,ID\_prelucrare#,ID\_echipament#)
* EchipamentMaterial(ID\_EM#,ID\_PM#,ID\_echipament#)
* Echipament(ID\_echipament#,categorie,nume,model,pret,cod\_bare,descriere,data\_cumparare,data\_ultm\_rezivie,depreciere)
* Revizie(ID\_revizie#,ID\_echipament#,data,cost,comentarii)
* ProdusFactura(ID\_produsfactura#,ID\_produspersonalizat#,ID\_Factura#)
* Factura(ID\_factura#,ID\_client#,transport,data,data\_livrare,GST,trimis,persoana\_fizica,comentarii, AdresaFirma,ContactFirma,EmailFirma,ContBancarFirma)
* Client(ID\_client#,nume,contact,adresa,email)

1. Realizarea normalizarii pana la forma normal 3(FN1-FN3)

* Forma Normala 1 (FN1)
* O relatie se afla in forma normala 1 daca fiecarui atribut care o compune ii corespunde o valoare indivizibila. Mai precis, o relatie se afla in forma normala 1 daca exista un indentificator unic.
* De exemplu, in entitatea ProdusPersonalizat cheia primara este id\_produspersonalizat, ceea ce inseamna ca este un indentificator unic (fiecare produs are un id unic). In acest fel, atributul id\_produspersonalizat fiind o valoare indivizibila, relatia se afla in forma normala 1(exista identificator unic pentru toate intrarile din tabel).
* Pentru a exemplifica cum se aplica aceasta normalizare se va crea un exemplu, mai précis se va presupune ca pentru un Produs Personalizat intra in compozitia sa mai multe material.
* Cum arata initial relatia produspersonalizat\_nonFN1:

|  |  |
| --- | --- |
| ID\_produspersonalizat | ID\_material |
| 1 | 2,3,4 |
| 2 | 4,2,1 |
| 3 | 6,4 |
| 4 | 2,5 |
| 5 | 6,7 |

Astfel se paote observa ca relatia de mai sus mentionata nu este in FN, deoarece atributul id\_matertial nu ii corespunde o valoare indivizibila.

* Aplicarea prima forma normal, relatia va arata astfel:

|  |  |
| --- | --- |
| ID\_produspersonalizat | ID\_material |
| 1 | 2 |
| 1 | 3 |
| 1 | 4 |
| 2 | 4 |
| 2 | 2 |
| 2 | 1 |
| 3 | 6 |
| 3 | 4 |
| 4 | 2 |
| 4 | 5 |
| 5 | 6 |
| 5 | 7 |

* Se poate observa ca dupa normalizarea fiecarui atribut care o compune ii corespunde o valoarea indivizibila
* **FORMA NORMALA2(FN2)**
  + Definiția acestei forme normale este următoarea: O relație se află în formă normală 2 dacă și numai dacă aceasta se află în prima formă normală și dacă fiecare atribut care nu face parte din cheie este dependent de întreaga cheie primară
  + Vom exemplifica o relatie non\_FN2, pentru tabelul facture (ce va fi reprezentat nu se va regasi in in baza de date, este un exemplu imaginar)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id\_factura | Id\_client# | Id\_plata# | Tip\_plata | Transport |
| 1 | 1 | 1 | Cash | 100 |
| 2 | 2 | 2 | Wire transfer | 50 |
| 3 | 3 | 3 | Card | 200 |

Relatia de mai sus, din tabel, se afla in FN1 deoarece exista identificator unic pentru toate intrarile din tabel.

Pentru a fi in FN2 trebuie fiecare atribut ce nu eeste cheie sa depinda de intreaga cheie primara. In acest caz atributele id\_factura, id\_client, id\_plata trebuie sa depinda de intreaga cheie primara (id\_factura#, id\_client#, id\_plata #). Se observa clar transport este dependent doar de cheia id\_factura, iar tip\_plata de id\_plata#, ceea ce inseamna ca exita dependeta partial, astfel, relatia prezentata nu se afla in FN2.

Dependetele vor fi:

* (id\_factura#) -> (transport)
* (id\_plata#,id\_factura#,id\_client#) -> (tip\_plata)

Transformarea in FN2:

**Plata Factura**

|  |  |
| --- | --- |
| ID\_plata # | Tip\_plata |
| 1 | Cash |
| 2 | Wire Transfer |
| 3 | Card |

**Factura**

|  |  |  |  |
| --- | --- | --- | --- |
| Id factura | Id\_client# | Id\_plata# | Transport |
| 1 | 1 | 1 | 100 |
| 2 | 2 | 2 | 50 |
| 3 | 3 | 3 | 200 |

* **FORMA NORMALA3(FN3)**
  + Definiția acestei forme normale este următoarea: O relație se află în formă normală 3 dacă și numai dacă se află în formă normală 2 și dacă fiecare atribut care nu este cheie depinde direct de cheia primară și numai de ea.’

Pentru a exemplifica aceasta forma normal s-a ales sa se realizeze un exemplu fictive pentru Factura(se presupune ca atributul transport depinde de tip\_factura si pret,fapt ce nu este adevarat in cazul diagramei construite)

Asfel, daca ar fi aceste date in Factura ( acest tabel este creat doar pentru a exemplifica FN3):

|  |  |  |
| --- | --- | --- |
| ID factura | transport | Tip\_factura |
| 1 | 200 | Invoice |
| 2 | 400 | Overdue Invoice |
| 3 | 0 | Timesheet Invoice |
| 4 | 200 | Invoice |

Aceasta relatie este in FN1, pentru ca exista indentificator unic pentru toate intrarile din tabel, anume **id\_factura#**

Se afla si in FN2 deoarece se poate remawrca faptul ca atributele: pret, tip\_factura care nu sunt in component cheii primare sunt dependente de intreaga cheie primara **id\_factura#**.

In acest exemplu se observa ca atributul transport depinde tranzitiv de **tip\_factura**.

Astfel, acest exemplu nu se afla in FN3, avand in vedere relatiile de dependenta intalnite:

* + Id\_factura -> tip factura
  + Id\_factura - > tip facture -> transport

Pentru a adduce relatia facture\_nonFN3 in forma normala 3 se aplica regula Casey-Delobel care presupune descompunerea relatiei prin eliminarea dependentei functionale transitive. In acest mod, din cause dependentei functionale transport-tip\_factura care surprinde faptul ca relatia nu se afla in FN3, se inlocuieste relatia prin doua proiectii:

* + Factura\_FN3 (id\_factura#, tip\_factura)
  + Tip\_factura\_transport (tip\_factura, transport)

Factura\_FN3:

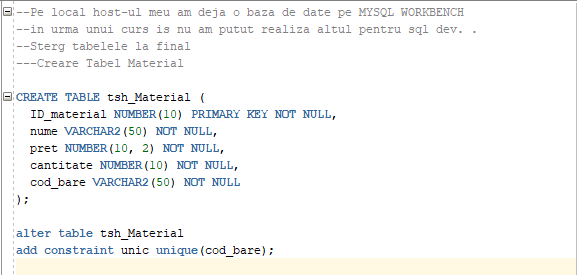
|  |  |
| --- | --- |
| Id factura | Tip factura |
| 1 | invoice |
| 2 | Overdue invoice |
| 3 | Timesheet invoice |
| 4 | invoice |

Tip\_factura\_tansport

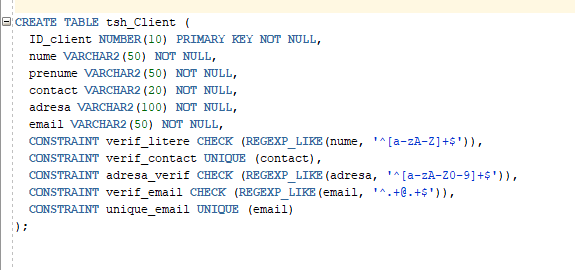
|  |  |
| --- | --- |
| Tip factura | Transport |
| Invoice | 200 |
| Overdue invoice | 400 |
| Timesheet invoice | 0 |
| invoice | 200 |

1. Crearea unei secvente ce va fi utilizata in inserarea inregistrarilor in tabele (pct 11).

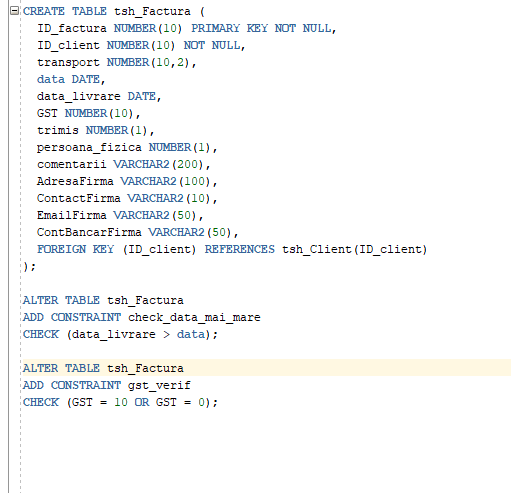
* Creare Tabel Material



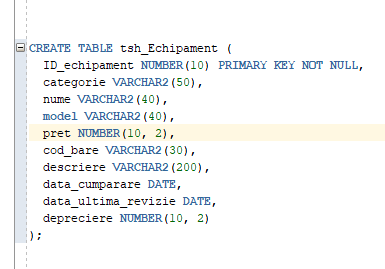
* Creare Tabel Client



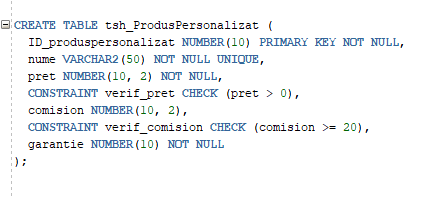
* Creare Tabel Factura



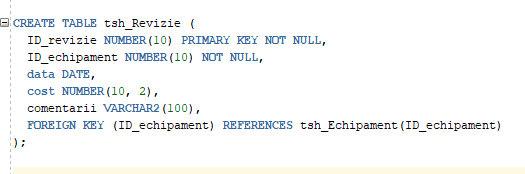
* Creare Tabel Echipament

****

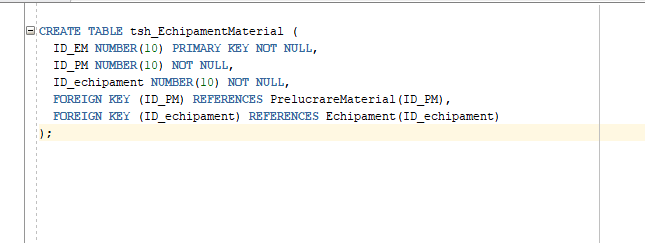
* Creare Tabel ProdusPersonalizat

****

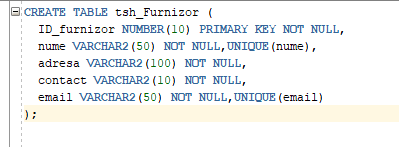
* Creare Tabel Revizie



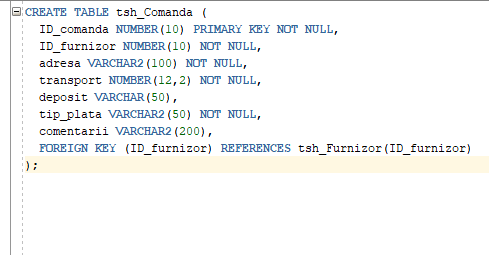
* Creare Tabel Echipament Material



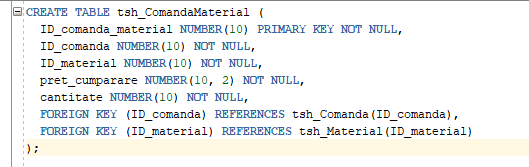
* Creare Tabel Furnizor



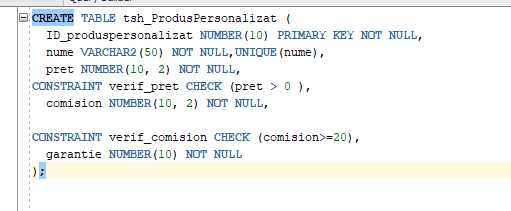
* Creare Tabel Comanda



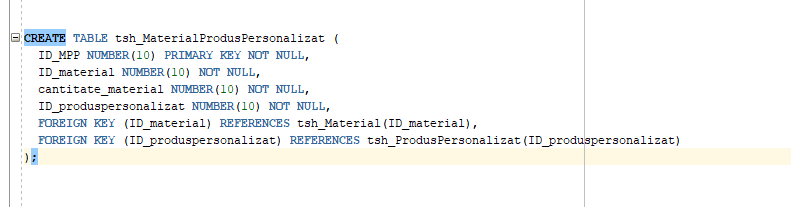
* Creare Tabel ComandaMaterial



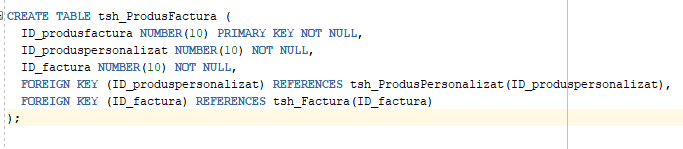
* Creare Tabel ProdusPersonalizat



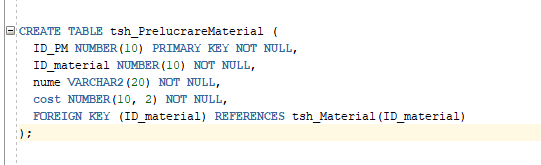
* Creare Tabel MaterialProdusPersonalizat



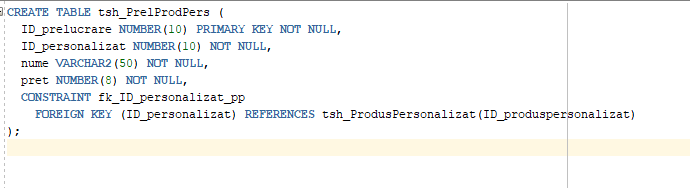
* Creare Tabel ProdusFactura



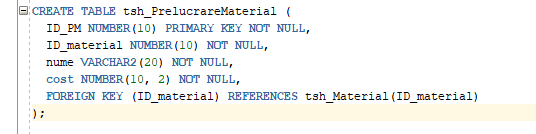
* Creare Tabel PrelucrareMaterial



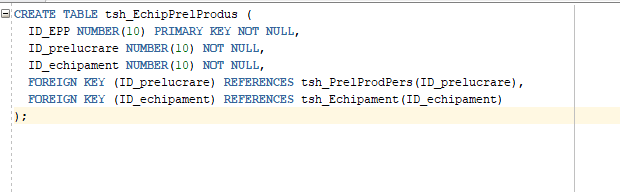
* Creare Tabel PrelucrareProdusPersonalizat



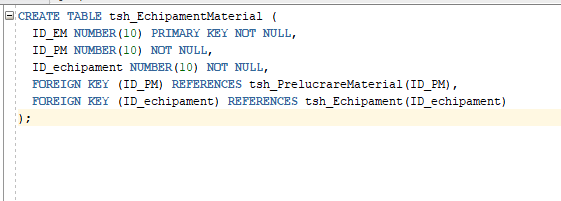
* Creare Tabel PrelucrareMaterial



* Creare Tabel EchipamentPrelugrareProdus

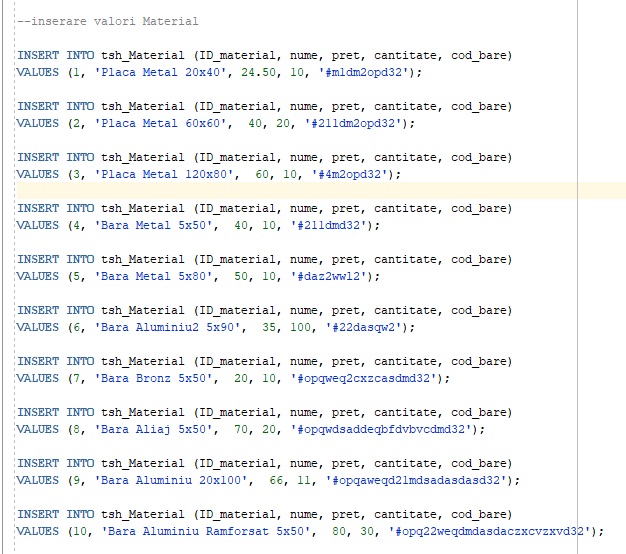


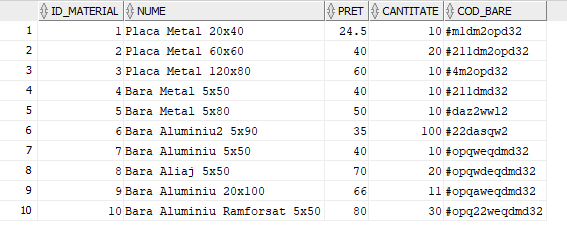
* Creare Tabel EchipamentMaterial



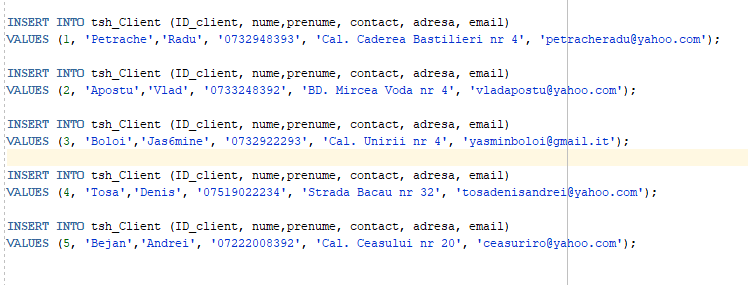
1. Crearea tabelelor in SQL si inserarea de date coerenta in fiecare dintre aecstea (minimum 5 inregistrari in fiecare table seasociativ, minimum 10 inregistrari in tabelele asociative)

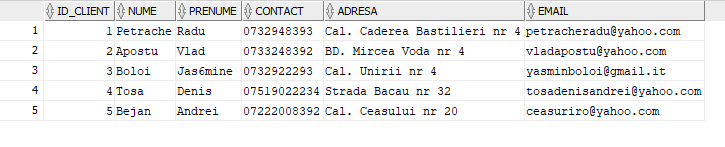
* Inserare date tabel material



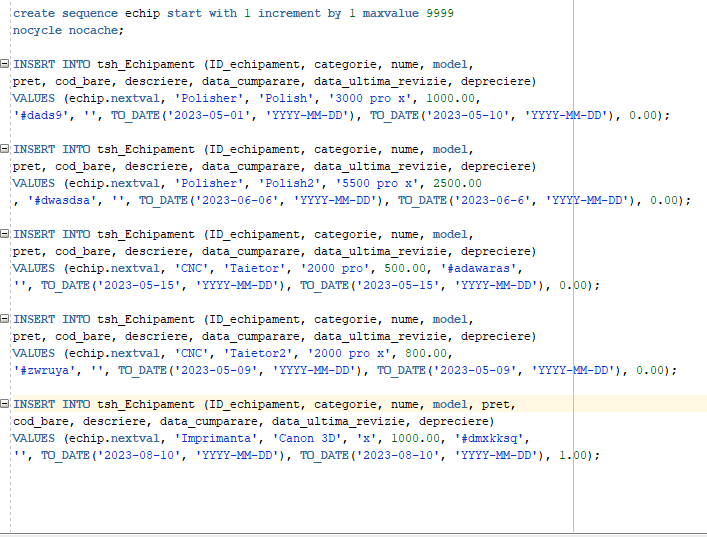


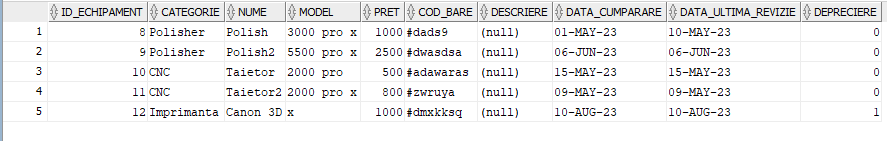
* Inserare date tabel client





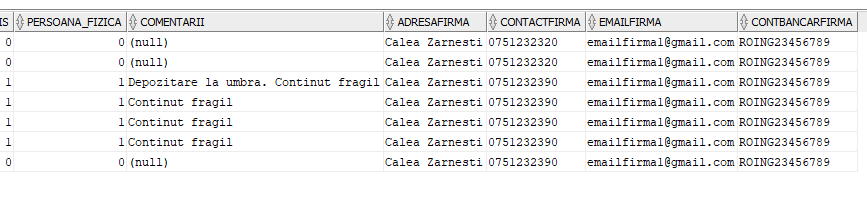
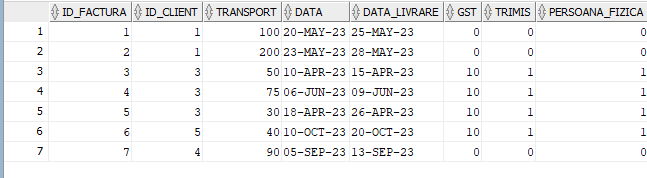
* Inserare date tabel echipament



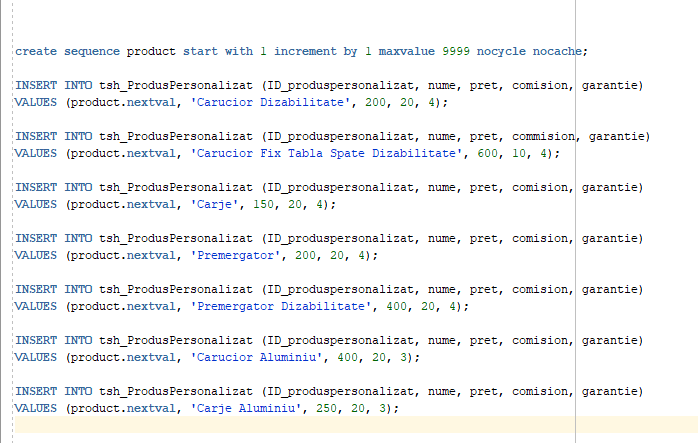


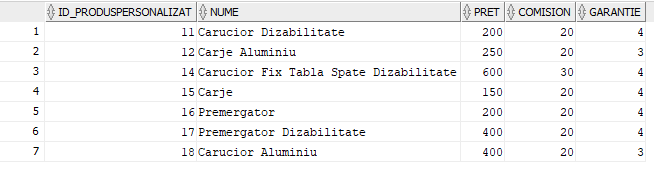
* Inserare date tabel factura



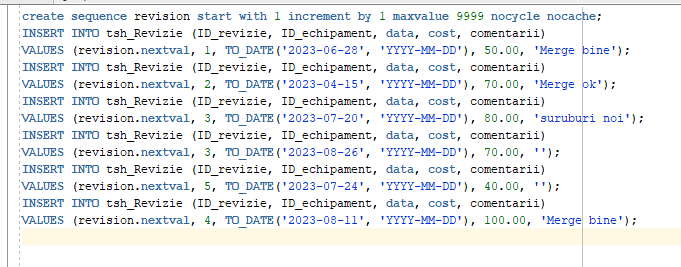


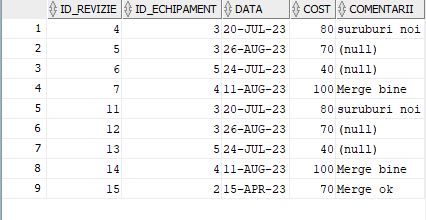
* Inserare date tabel ProdusPersonalizat



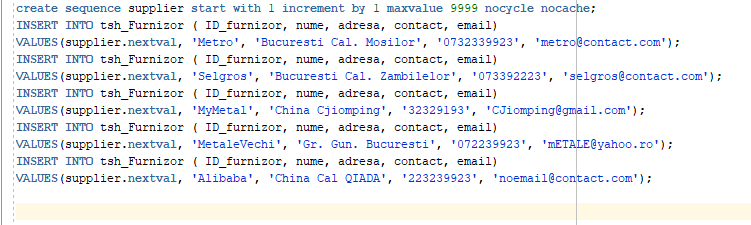


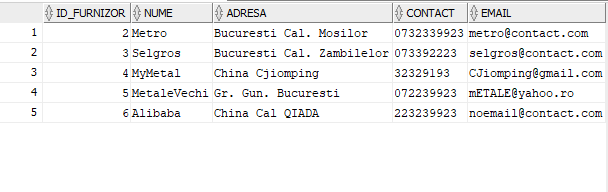
* Inserare date tabel Revizie



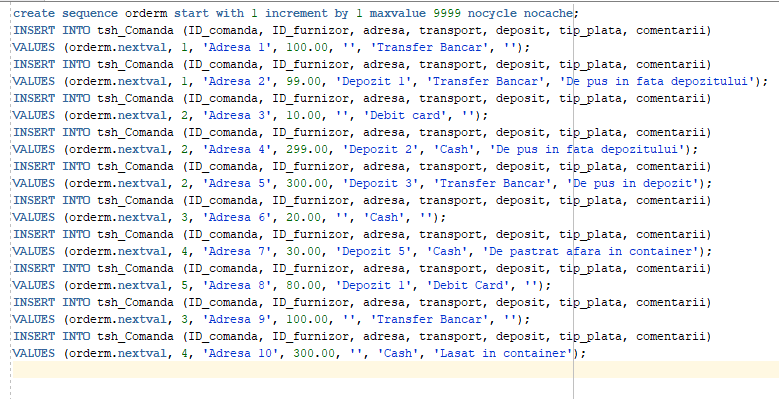


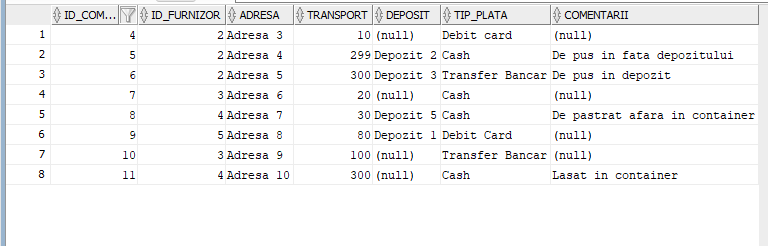
* Inserare date tabel Furnizor



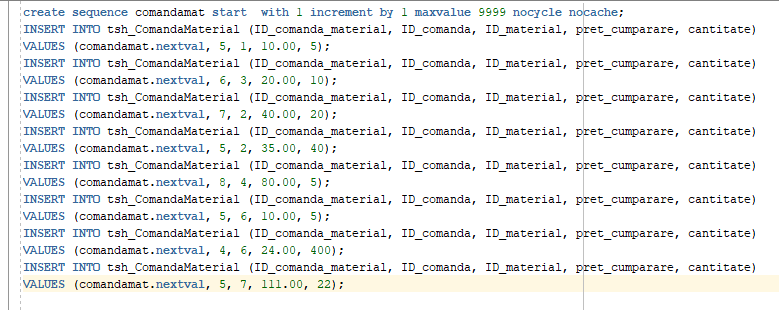


* Inserare date tabel Comanda





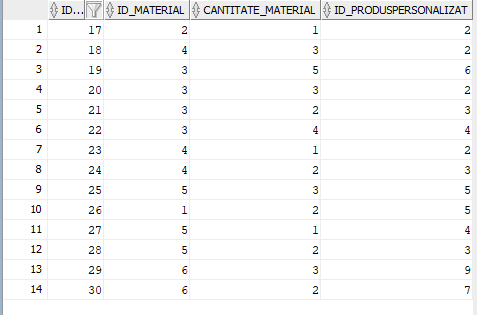
* Inserare date tabel ComandaMaterial



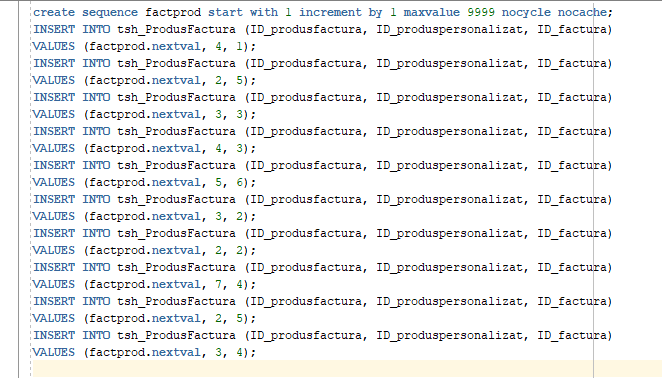


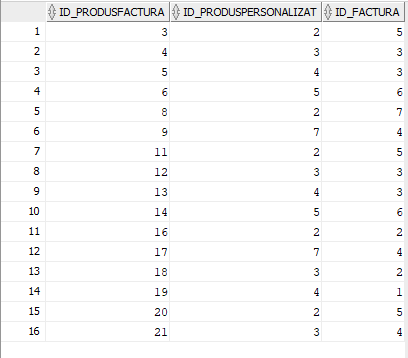
* Inserare date tabel MaterialProdusPersonalizat



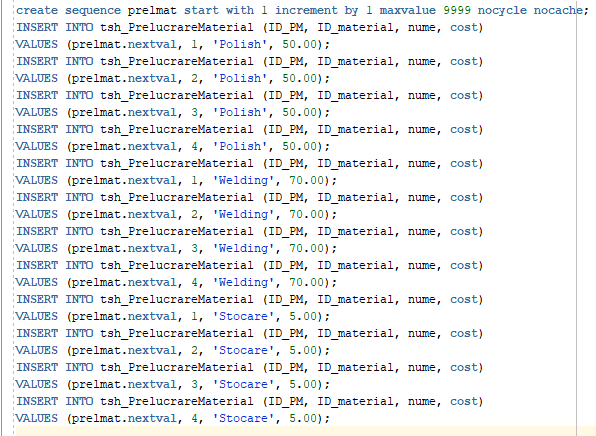


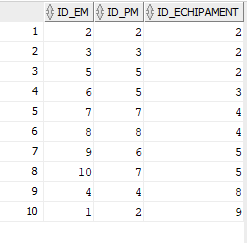
* Inserare date tabel ProdusFactura



****

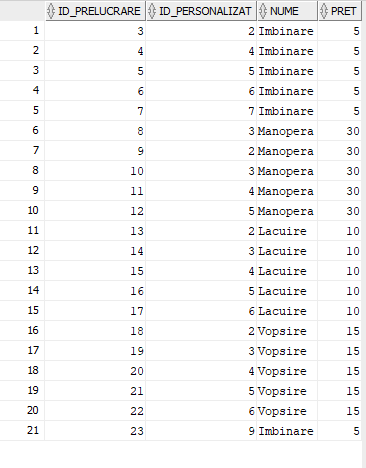
* Inserare date tabel PrelucrareMaterial



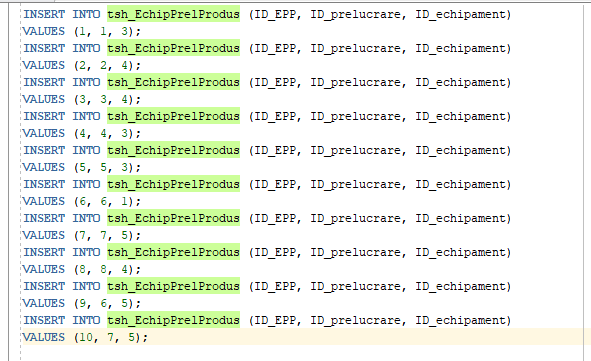


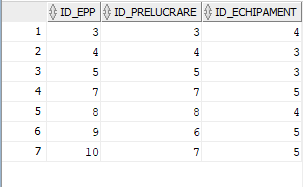
* Inserare date tabel PrelucrareProdusPersonalizat



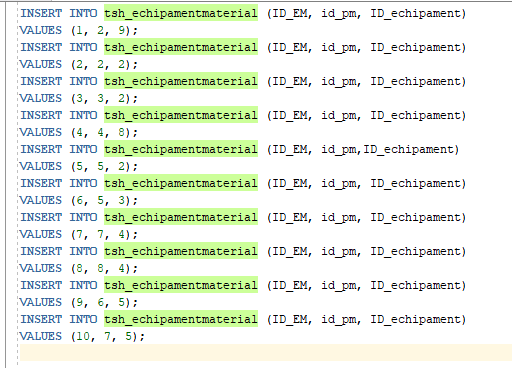
****

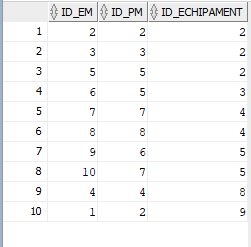
* Inserare date tabel EchipamentPrelucrareProdus





* Inserare date tabel EchipamentMATERIAL





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

•subcereri sincronizate în care intervin cel puțin 3 tabele

•subcereri nesincronizateîn clauza FROM

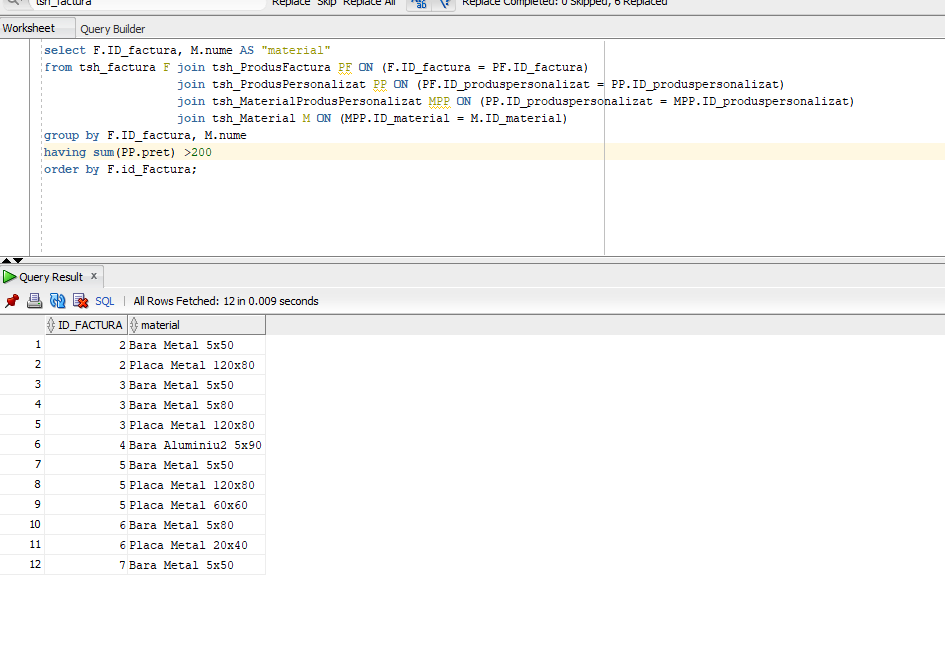
•grupări de date cu subcereri nesincronizate in care intervin cel putin 3 tabele, funcții grup, filtrare la nivel degrupuri(in cadrul aceleiasi cereri)

•ordonărisi utilizarea funcțiilor NVL și DECODE (in cadrul aceleiasi cereri)

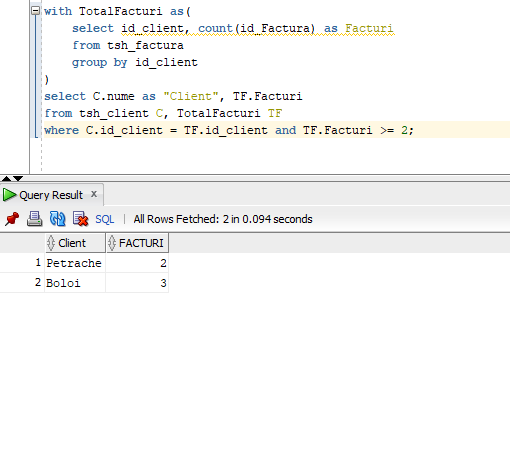
•utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE

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

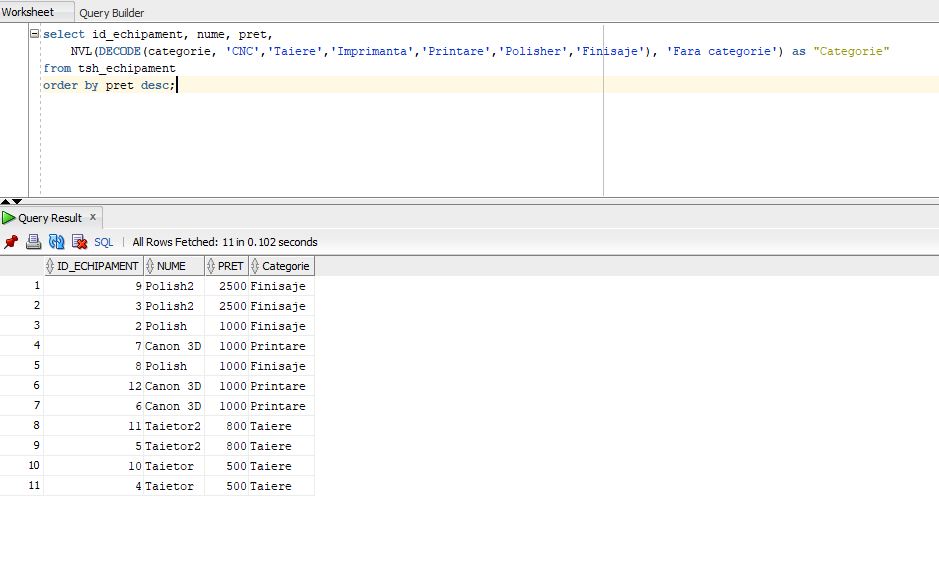
1. Prima cerere obtine toate materialele folosite in produsele personalizate de pe fiecare factura, doar daca, insumate, produsele costa mai mult de 200. (subcereri sincronizate si functii agregate)



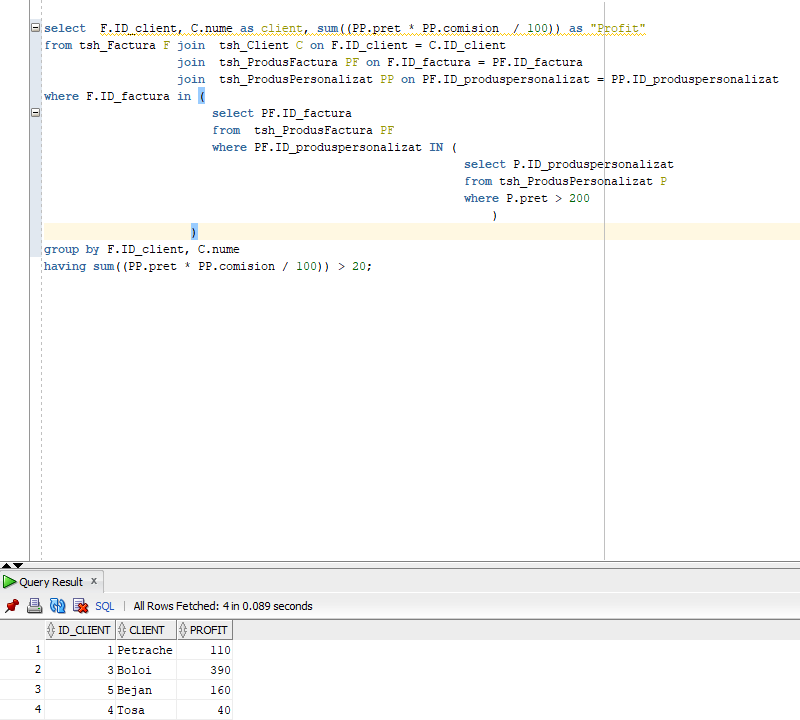
1. A doua cerere cauta clientii care au cel putin 2 facturi, si afiseaza nr. Acestora. (clauza with)



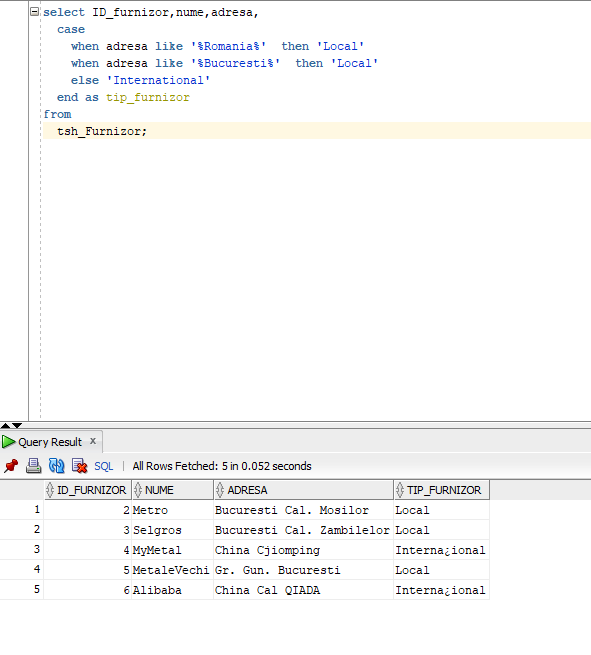
1. A treia cerere ordoneaza echipamentele in ordine descrescatoare dupa pret si foloseste functiile nvl si decode.



1. Aceasta cerere va returna Id ul si numele clientului si suma totala a profitului pentru facturile care contin produse mai scumpe de 200. Rezultate finale sunt filtrate pentru a afisa doar inregistrarile cu un profit mai mare de 20.

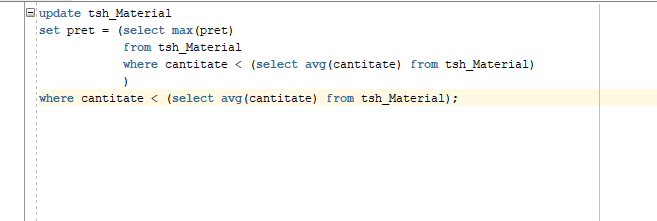


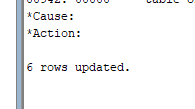
1. Aceasta cerere va verifica adresa furnizorilor pentru a determina daca acestia sunt locali sau internationali (folosing case si operatii pe string)



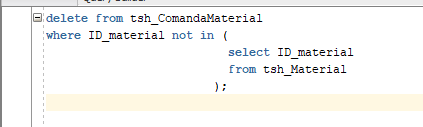
13.Implementarea 3 operatii de actualizare si de suprimare a datelor utilizand subcereri.

1. Se va actualiza pretul materialelor care au o cantitate mai mica decat media cantitatilor din tabelul Material (tsh\_Material) la valorea maxima de pret din tabel.

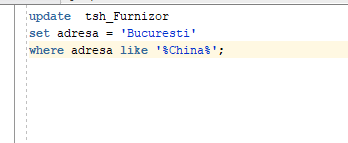




1. Se vor sterge toate intrarile din tabela ComandaMaterial pentru materialele care nu sunt disponibile in tabelul Material



1. Se vor modifica adresele furnizorilor din China cu Bucuresti.



Problema 14

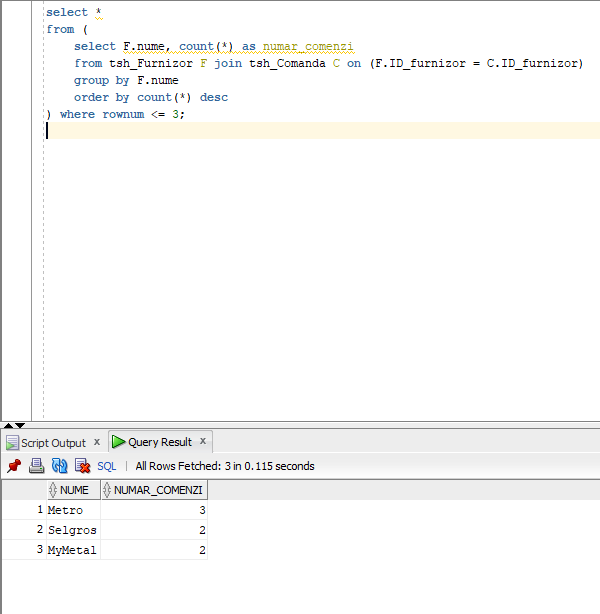
Pentru a adauga o inregistreare in tabelul Materal , o operatie LMD permisa ar fi :

INSERT INTO tsh\_Material (ID\_material, nume, pret, cantitate, cod\_bare)

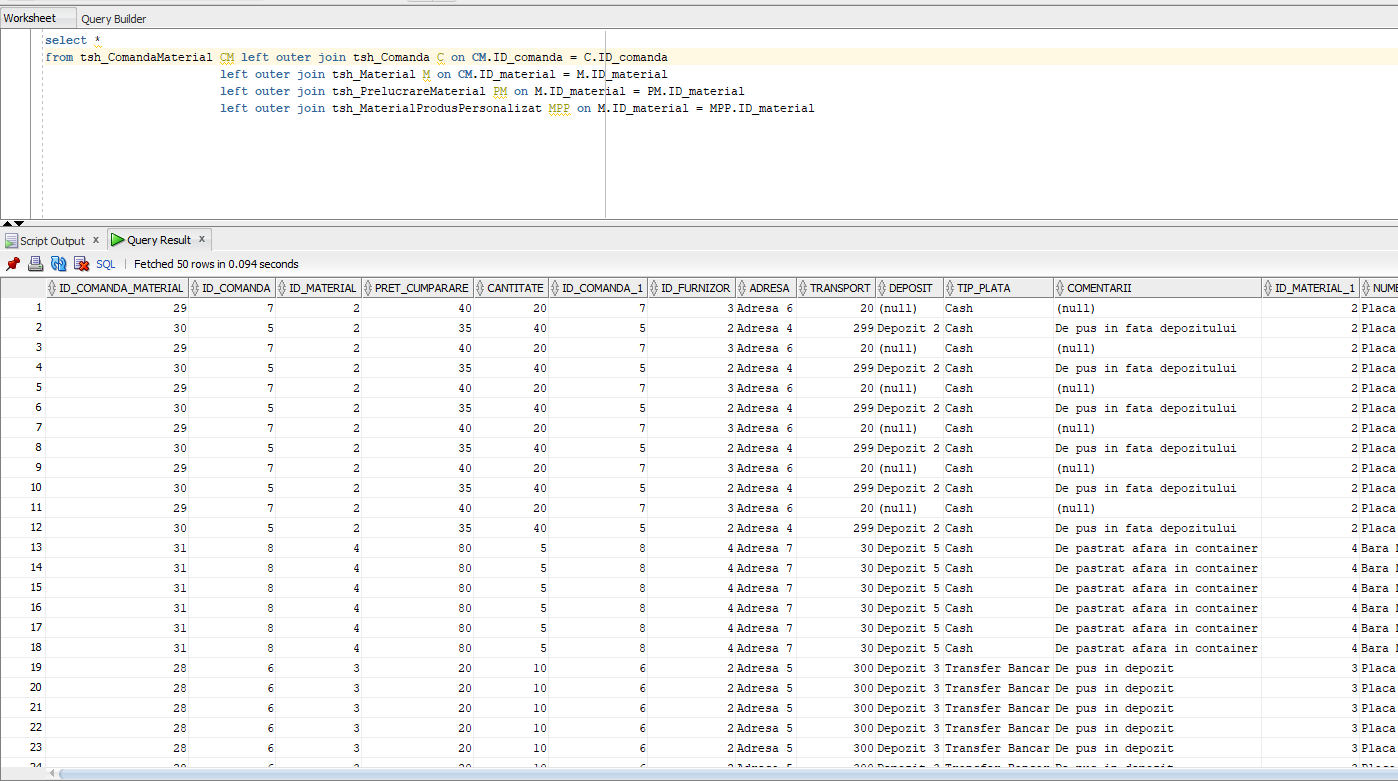
VALUES (10, 'Placa Metal 20x40', 24.50, 10, '#m1dm2opd32');

* Operatia este permisa deoarece respecta regulile si constrangerile
* ID\_materail este unic, NN
* Cantitatea este NN si mai mare decat 0
* Pretul este NN
* Codul de bare este unic
* O operatie nepermisa LMD ar fi daca incercam sa modificam codul de bare al altui material cu cel de mai sus '#m1dm2opd32' , pentru ca nu ar respecta constrangerea de unicitate / sa modificam valoarea primary key ul cu un id déjà existent folosit ca si cheie straina in alt tabel.

Problema 15.



1. Aceasta cerere returneaza primii 3 furnizori, ordanandu-i dupa numarul de comenzi in ordine descrescatoare.



1. Aceasta cerere returneaza toate datele despre tabelele selectate, cu valori corespunzatoare pentru inregistrarile care au o potrivire in functie de cheile primare si straine. In cazul in care nu exista o potrivirem, valorile pentru coloanele din tabelele asociate vor fi nule sau vor avea valorea implicita specificata.