Proiect Baze de Date

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

O fabrica de mobila, a carei locatie si nume il cunoastem, dispune de un numar de hale.

In interiorul acestora sunt organizate departamente diferite.

In evidenta angajatilor, regasim toate informatiile necesare despre acestia.

Fabrica dispune de un stoc al materialelor si de un catalog, in care clientii pot vedea ce modele de mobila sunt produse.

In momentul in care cantitatea unui material sau obiect ce intra in alcatuirea constructiei mobilei scade sub 5, se efectueaza o comanda catre furnizor.

Clientii pot comanda produse din catalog, sau pot da o comanda personalizata. Pentru comanda personalizata se va atasa un link catre un document ce cuprinde informatiile necesare. Cererea le poate fi acceptata sau nu. Dupa plasarea comenzii, vor primi termenul limita pana la care le va fi expediata.

Transportul mobilei este realizata de un curier.

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

Fabrica cuprinde mai multe hali.

O hala cuprinde unul sau mai multe departamente.

Un department apartine unei singure hali.

Un angajat lucreaza intr-un singur departament.

Un anumit tip de material se achizitioneaza de la un singur furnizor.

Furnizorii vand mai multe tipuri de materiale.

Clientii comanda unul sau mai multe obiecte.

Fiecare comanda personalizata primește un calificativ: admis sau respins.

Curierul livreaza una sau mai multe comenzi finalizate.

1. Descrierea entităților, incluzând precizarea cheii primare.

|  |  |  |
| --- | --- | --- |
| ENTITATE | CHEIE PRIMARA | OBSERVATII |
| Fabrica | Id\_fabrica | Fabrica pentru confectionarea si vinderea mobilei. |
| Hala | Id\_hala | Locatiile in care se afla departamentele. |
| Departament | Id\_departament | Fiecare este dedicata diferitelor etape de constructie si ansamblare a unui obiect de mobila. |
| Angajat | Id\_angajat | Informatii generale despre persoana angajata. |
| Material | Id\_material | Detalii despre materialul din stoc. |
| Furnizor | Id\_furnizor | Informatii despre furnizorii care aprovizioneaza stocul. |
| Produs | Id\_produs | Informatii despre obiectele standard ce se produc in fabrica |
| Factura | Id\_factura | Date despre fabrica, cumparator si produsele achizitionate |
| Client | Id\_client | Informatii despre client |
| Curier | Id\_curier | Persoana care este angajat al fabricii si livreaza clientilor mobila comandata. |
| Comanda\_produs | Id\_comanda\_produs | Informatii generale despre comandarea unui produs de catre un client. |
| Comanda standard | Id\_comanda\_produs | Informatii specifice despre comenzile realizate din catalogul cu produse standard al fabricii  Subentitate a entitatii comanda\_produs |
| Comanda personalizata | Id\_comanda\_produs | Informatii specifice despre comandarea unui produs modificat (pdf)  Subentitate a entitatii comanda\_produs |

1. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

* FURNIZOR\_**vinde**\_MATERIAL => relatia binara **vinde** este de tip one-to-many si uneste entitatile FURNIZOR si MATERIAL, prezentand legatura dintre acestea ( un furnizor vinde mai multe tipuri de materiale, iar un material este cumparat de la un singur furnizor ). Asadar avem cardinalitatea maxima si minima 1(1) to M(1), deoarece de la un furnizor trebuie sa se cumpere minim un tip de material, ca acesta sa apara in baza de date, iar un material trebuie sa fie cumparat de la un furnizor.
* FABRICA\_**are**\_MATERIAL => relatia binara **are** este de tip one-to-many si uneste entitatile FABRICA si MATERIAL, prezentand legatura dintre acestea (o fabrica are in stoc mai multe tipuri de materiale, iar un material are un cod unic apartinand unei singure fabrici). Asadar avem cardinalitatea maxima si minima 1(1) to M(1), deoarece fabrica are in stoc cel putin un tip de material, iar un material apartine obligatoriu unei fabrici.
* FABRICA\_**cuprinde**\_HALA => relatia binara **cuprinde** este de tip one-to-many si uneste entitatile FABRICA si HALA, prezentand legatura dintre acestea (o fabrica cuprinde mai multe hali, iar o hala apartine unei singure fabrici). Asadar avem cardinalitatea maxima si minima 1(1) to M(1), deoarece o fabrica trebuie sa aiba cel putin o hala, iar o hala apartine obligatoriu unei fabrici.
* HALA\_**cuprinde**\_DEPARTAMENT => relatia binara **cuprinde** este de tip one-to-many si uneste entitatile HALA si DEPARTAMENT, prezentand legatura dintre acestea (o hala cuprinde mai multe departamente, iar un departament se afla in cadrul unei singure hali). Asadar avem cardinalitatea maxima si minima 1(1) to M(1), deoarece o hala trebuie sa aiba cel putin un departament, iar un departament apartine obligatoriu unei hali.
* DEPARTAMENT\_**lucreaza**\_ANGAJAT => relatia binara **lucreaza** este de tip one-to-many si uneste entitatile DEPARTAMENT si ANGAJAT, prezentand legatura dintre acestea (intr-un departament lucreaza mai multi angajati, iar un angajat lucreaza in cadrul unui singur departament). Asadar avem cardinalitatea maxima si minima 1(1) to M(1), deoarece intr-un departament trebuie sa lucreze cel putin un angajat, iar un angajat trebuie sa apartina unui departament.
* FABRICA\_**proceseaza**\_COMANDA\_PRODUS => relatia binara **proceseaza** este de tip one-to-many si uneste entitatile FABRICA si COMANDA\_PRODUS, prezentand legatura dintre acestea (o fabrica proceseaza mai multe comenzi, iar o comanda este procesata de catre o singura fabrica). Asadar avem cardinalitatea maxima si minima 1(1) to M(1), deoarece o fabrica proceseaza cel putin o comanda, iar o comanda este procesata obligatoriu de catre o fabrica.
* COMANDA\_PRODUS \_**cuprinde**\_PRODUS => relatia binara **cuprinde** este de tip many-to-many si uneste entitatile COMANDA\_PRODUS si PRODUS, prezentand legatura dintre acestea (o comanda trebuie sa cuprinda mai multe produse sau cel putin unul, iar un tip de produs poate sa apartina mai multor comenzi sau niciuneia). Asadar avem cardinalitatea maxima si minima M(0) to M(1).
* CLIENT\_**efectueaza**\_COMANDA\_PRODUS => relatia binara **efectueaza** este de tip one-to-many si uneste entitatile CLIENT si COMANDA\_PRODUS, prezentand legatura dintre acestea (un client efectueaza mai multe comenzi, iar o comanda este efectuata de catre un singur client). Asadar avem cardinalitatea maxima si minima 1(1) to M(1), deoarece un client trebuie sa efectueze cel putin o comanda, iar o comanda este efectuata in mod obligatoriu de catre un client.
* CLIENT\_**plateste**\_FACTURA => relatia binara **plateste** este de tip one-to-many si uneste entitatile CLIENT si FACTURA, prezentand legatura dintre acestea (un client plateste mai multe facturi, iar o factura este platita de catre un singur client). Asadar avem cardinalitatea maxima si minima 1(1) to M(1), deoarece un client trebuie sa plateasca cel putin o factura, iar o factura este platita in mod obligatoriu de catre un client.
* FACTURA \_**insumeaza**\_PRODUS => relatia binara **insumeaza** este de tip many-to-many si uneste entitatile FACTURA si PRODUS, prezentand legatura dintre acestea (o factura trebuie sa cuprinda mai multe produse sau cel putin unul, iar un tip de produs poate sa apartina mai multor facturi sau niciuneia). Asadar avem cardinalitatea maxima si minima M(0) to M(1).

|  |  |  |
| --- | --- | --- |
| **RELATIE** | **CARDINALITATE** | **OBSERVATII** |
| livreaza | curier-(client,produs)  one-to-many  client-(curier,produs)  one-to-many  produs-(curier, client)  one-to-many | Un curier poate livra mai multor clienti mai multe produse.  Un client poate primi de la mai multi curieri mai multe produse.  Un produs poate fi livrat mai multor clienti, de catre mai multi curieri. |

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

ENTITATE: ***FABRICA***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune/ precizie | Valori posibile si valori default | Observații, obligatoriu/opțional |
| Id\_fabrica | integer |  |  | PK |
| Locatie | string | 200 |  | NOT NULL  Orasul unde este amplasata |
| Nume | string | 100 |  | NOT NULL |

ENTITATE: ***HALA***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune/ precizie | Valori posibile si valori default | Observații, obligatoriu/opțional |
| Id\_hala | Integer |  |  | PK |
| Id\_fabrica | integer |  |  | FK |
| Suprafata | float |  | [150,1000] | Unitate de masura: m^2  NOT NULL |
| Nume | string | 100 |  | Not null |

ENTITATE: ***DEPARTAMENT***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune/ precizie | Valori posibile si valori default | Observații, obligatoriu/opțional |
| Id\_departament | integer |  |  | PK |
| Id\_hala | integer |  |  | FK |
| Nume | string | 100 |  | NOT NULL |

ENTITATE: ***ANGAJAT***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune/ precizie | Valori posibile si valori default | Observații, obligatoriu/opțional |
| Id\_angajat | integer |  |  | PK |
| Id\_departament | integer |  |  | FK |
| Nume | string | 50 |  | NOT NULL |
| Prenume | string | 100 |  | NOT NULL |
| CNP | integer | 13 |  | NOT NULL |
| Salariu | float |  | Valoare default = 0 |  |
| Data\_angajare | date |  |  | NOT NULL |
| Telefon | integer |  |  | NOT NULL |
| Mail | string | 100 |  | Optional  UNIQUE |
| Sex | String | 10 | F/M | NOT NULL |

ENTITATE: ***MATERIAL***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune/ precizie | Valori posibile si valori default | Observații, obligatoriu/opțional |
| Id\_material | integer |  |  | PK |
| Id\_fabrica | Integer |  |  | FK |
| Id\_furnizor | Integer |  |  | FK |
| Nume | String | 100 |  | NOT NULL |
| Cantitate | Integer |  | Valoare default = 0 |  |
| Categorie | string | 100 |  | NOT NULL |

ENTITATE: ***FURNIZOR***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune/ precizie | Valori posibile si valori default | Observații, obligatoriu/opțional |
| Id\_furnizor | Integer |  |  | PK |
| Nume | String | 100 |  | NOT NULL |
| Telefon | integer |  |  | NOT NULL |
| Mail | string | 100 |  | UNIQUE |

ENTITATE: ***COMANDA\_PRODUS***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune/ precizie | Valori posibile si valori default | Observații, obligatoriu/opțional |
| Id\_comanda | Integer |  |  | PK |
| Id\_fabrica | Integer |  |  | FK |
| Id\_client | Integer |  |  | FK |
| Data | Date |  |  | NOT NULL |
| Deadline | date |  |  | NOT NULL |

ENTITATE: ***COMANDA\_STANDARD***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune/ precizie | Valori posibile si valori default | Observații, obligatoriu/opțional |
| Id\_comanda | Integer |  |  | FK |
| Nume\_produs | String | 100 |  | NOT NULL |

ENTITATE: ***COMANDA\_PERSONALIZATA***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune/ precizie | Valori posibile si valori default | Observații, obligatoriu/opțional |
| Id\_comanda | Integer |  |  | FK |
| Admisa | String | 20 | T / F | NOT NULL |
| Link | string | 300 |  | NOT NULL |

ENTITATE: ***PRODUS\_COMANDAT***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune/ precizie | Valori posibile si valori default | Observații, obligatoriu/opțional |
| Id\_produs\_comandat | Integer |  |  | PK |
| Id\_produs | Integer |  |  | FK |
| Id\_comanda | Integer |  |  | FK |
| Id\_factura | Integer |  |  | FK |
| Pret | float |  |  | NOT NULL |

ENTITATE: ***PRODUS***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune/ precizie | Valori posibile si valori default | Observații, obligatoriu/opțional |
| Id\_produs | Integer |  |  | PK |
| Categorie | String | 100 |  | NOT NULL |
| Nume | String | 100 |  | NOT NULL |
| Pret | Float |  |  | NOT NULL |
| Dimensiune | string | 100 |  | NOT NULL |

ENTITATE: ***CLIENT***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune/ precizie | Valori posibile si valori default | Observații, obligatoriu/opțional |
| Id\_client | Integer |  |  | PK |
| Nume | String | 100 |  | NOT NULL |
| Prenume | String | 100 |  | NOT NULL |
| Telefon | Integer |  |  | NOT NULL |
| Mail | string | 100 |  | UNIQUE  NOT NULL |

ENTITATE: ***FACTURA***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune/ precizie | Valori posibile si valori default | Observații, obligatoriu/opțional |
| Id\_factura | Integer |  |  | PK |
| Id\_client | Integer |  |  | FK |
| Total\_plata | Float |  |  | NOT NULL |
| Data | Date |  |  | NOT NULL |

ENTITATE: ***CURIER***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune/ precizie | Valori posibile si valori default | Observații, obligatoriu/opțional |
| Id\_curier | Integer |  |  | PK |
| Nume | String | 100 |  | NOT NULL |
| Prenume | String | 100 |  | NOT NULL |
| CNP | Integer | 13 |  | NOT NULL |
| Telefon | Number |  |  | NOT NULL |
| Mail | string | 100 |  | Optional  UNIQUE |

RELATIE: ***LIVREAZA***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune/ precizie | Valori posibile si valori default | Observații, obligatoriu/opțional |
| Id\_client | Integer |  |  | FK |
| Id\_curier | Integer |  |  | FK |
| Id\_produs\_comandat | Integer |  |  | FK |
| Data | Date |  |  | NOT NULL |
| Ora\_livrare | Time |  |  | Optional  Livrarea se va realiza dupa ora mentionata. |

1. **ERD (DIAGRAMA ENTITATE-RELATIE)**

**lucreaza**

M(1)

ANGAJAT

FURNIZOR

1

1

**livreaza**

DEPARTAMENT

FACTURA

CURIER

CLIENT

PRODUS

MATERIAL

HALA

FABRICA

**vinde**

M(1)

M(1)

**cuprinde**

1

M(1)

M(1)

1

1

**are**

**cuprinde**

1

**proceseaza**

COMANDA\_PRODUS

M(1)

M(1)

1

1(0)

STANDARD

M(1)

**cuprinde**

M(0)

1(0)

ISA

1

M(1)

M

ISA

PERSONALIZATA

**efectueaza**

1

**insumeaza**

M

1

M(0)

M

M(1)

**plateste**

1. **DIAGRAMA CONCEPTUALA**

X

ANGAJAT

FURNIZOR

DEPARTAMENT

FACTURA

CLIENT

MATERIAL

HALA

FABRICA

X

X

X

X

X

PRODUS

COMANDA\_PRODUS

X

X

X

STANDARD

X

X

PRODUS\_COMANDAT

PERSONALIZATA

X

X

X

LIVREAZA

X

X

CURIER

1. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

FABRICA (#Id\_fabrica, Locatie, Nume)

HALA (#Id\_hala, #Id\_fabrica, Suprafata, Nume)

DEPARTAMENT (#Id\_departament, #Id\_hala, Nume)

ANGAJAT (#Id\_angajat, #Id\_departament, Nume , Prenume, CNP, Salariu , Data\_angajare, Telefon, E-mail)

MATERIAL (#Id\_material, #Id\_fabrica, #Id\_furnizor, Nume , Cantitate ,Categorie)

FURNIZOR (#Id\_furnizor, Nume , Telefon , E-mail)

COMANDA\_PRODUS (#Id\_comanda, #Id\_fabrica, #Id\_client, Data, Deadline)

COMANDA\_STANDARD (#Id\_comanda, Nume\_produs)

COMANDA\_PERSONALIZATA (#Id\_comanda, Admisa, Link)

PRODUS\_COMANDAT (#Id\_produs\_comandat, #Id\_produs, #Id\_comanda, #Id\_factura, Pret)

PRODUS (#Id\_produs, Categorie, Nume, Pret, Dimensiune)

CLIENT (#Id\_client, Nume, Prenume, Telefon, E-mail)

FACTURA (#Id\_factura, #Id\_client, Total\_plata, Data)

CURIER (#Id\_curier, Nume, Prenume, CNP, Telefon, E-mail)

LIVREAZA (#Id\_client, #Id\_curier, #Id\_produs\_comandat, Data, Interval\_orar)

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

• **Forma normală 1 (FN1)**

O relaţie este în FN1 dacă domeniile pe care sunt definite atributele relaţiei sunt constituite numai din valori atomice. Un tuplu nu trebuie să conţină atribute sau grupuri de atribute repetitive.

Aducerea relaţiilor în FN1 presupune eliminarea atributelor compuse şi a celor repetitive.

Forma normală 1 este cea care impune ca fiecare înregistrare să fie definită astfel încât să fie identificată unic prin intermediul unei chei primare.

Voi ilustra FN1 printr-un exemplu, folosindu-ma de entitatile FABRICA si MATERIAL.

Stim ca in cadrul unei fabrici, denumirile materialelor nu se repeta, dar nu ni se garanteaza acest lucru si pentru totalitatea fabricilor.

1. Mai multe valori seminifică același câmp

|  |  |
| --- | --- |
| FABRICA | MATERIAL |
| Nume1 | Lemn stejar, pal lucios, sticla securizata |
| Nume2 | Lemn stejar, pal lucios |
| Nume3 | Pal lucios, sticla securizata |

|  |  |
| --- | --- |
| FABRICA | MATERIAL |
| Nume1 | Lemn stejar |
| Nume1 | Pal lucios |
| Nume1 | Sticla securizata |
| Nume2 | Lemn stejar |
| Nume2 | Pal lucios |
| Nume3 | Pal lucios |
| Nume3 | Sticla securizata |

|  |  |  |
| --- | --- | --- |
| ID | FABRICA | MATERIAL |
| 1 | Nume1 | Lemn stejar |
| 2 | Nume1 | Pal lucios |
| 3 | Nume1 | Sticla securizata |
| 4 | Nume2 | Lemn stejar |
| 5 | Nume2 | Pal lucios |
| 6 | Nume3 | Pal lucios |
| 7 | Nume3 | Sticla securizata |

1. Mai multe coloane reprezintă același tip de date.

|  |  |  |  |
| --- | --- | --- | --- |
| FABRICA | MATERIAL | MATERIAL(1) | MATERIAL(2) |
| Nume1 | Lemn stejar | pal lucios | sticla securizata |
| Nume2 | Lemn stejar | pal lucios |  |
| Nume3 | Pal lucios | sticla securizata |  |

|  |  |
| --- | --- |
| FABRICA | MATERIAL |
| Nume1 | Lemn stejar |
| Nume2 | Lemn stejar |
| Nume3 | Pal lucios |
| Nume1 | pal lucios |
| Nume2 | pal lucios |
| Nume3 | sticla securizata |
| Nume1 | Sticla securizata |

|  |  |  |
| --- | --- | --- |
| ID | FABRICA | MATERIAL |
| 1 | Nume1 | Lemn stejar |
| 2 | Nume1 | Pal lucios |
| 3 | Nume1 | Sticla securizata |
| 4 | Nume2 | Lemn stejar |
| 5 | Nume2 | Pal lucios |
| 6 | Nume3 | Pal lucios |
| 7 | Nume3 | Sticla securizata |

Unicitatea unei inregistrari este asigurata prin utilizarea unei chei primare. In exemplul de mai sus aceasta fiind reprezentata prin coloana aditionala ID, in care, prin introducerea unui intreg, se asigura unicitatea fiecarui material introdus.

• **Forma normală 2 (FN2)**

O relaţie se află în a doua formă normală FN2 dacă:

- se află în forma normală FN1

- fiecare atribut care nu este cheie este dependent de întreaga cheie primară.

Asadar, daca un atribut este dependent functional doar de o parte a cheii, atunci acesta va fi mutat intr-un alt tabel. Daca tabela nu are o cheie primara compusa, atunci ea este automat in FN2.

Voi lua ca exemplu entitatea FACTURA.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| #Id\_factura | Nume client | #Id\_client | Total\_plata | Data |
| F1 | N1 | C1 | 2500 | 12-03-2022 |
| F2 | N1 | C1 | 4000 | 27-06-2020 |
| F3 | N2 | C2 | 3200 | 03-09-2021 |
| F4 | N3 | C3 | 1500 | 14-10-2022 |
| F5 | N3 | C3 | 2300 | 07-08-2021 |

Un client poate plati mai multe facturi, iar o factura este platita de catre un singur client. Tabelul se afla in FN1, deoarece avem un identificator unic pentru toate datele din tabel, anume *Id\_factura.*

Observam ca atributul *Nume\_client* nu este dependent de intreaga cheie primara (*#Id\_factura, #Id\_client*), ci doar de *Id\_client -> relatia nu se afla in FN2.*

Astfel avem ca:

• {#Id\_client} -> {Nume\_client}

• {#Id\_factura, #Id\_client} -> {Total\_plata, Data}

Aplicand regula Casey Delobel rezulta că, pentru a avea relația în FN2, numele clientului trebuie să fie doar în entitatea CLIENT.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| #Id\_factura | Nume client | #Id\_client | Total\_plata | Data |
| F1 | N1 | C1 | 2500 | 12-03-2022 |
| F2 | N1 | C1 | 4000 | 27-06-2020 |
| F3 | N2 | C2 | 3200 | 03-09-2021 |
| F4 | N3 | C3 | 1500 | 14-10-2022 |
| F5 | N3 | C3 | 2300 | 07-08-2021 |

|  |  |  |  |
| --- | --- | --- | --- |
| #Id\_factura | #Id\_client | Total\_plata | Data |
| F1 | C1 | 2500 | 12-03-2022 |
| F2 | C1 | 4000 | 27-06-2020 |
| F3 | C2 | 3200 | 03-09-2021 |
| F4 | C3 | 1500 | 14-10-2022 |
| F5 | C3 | 2300 | 07-08-2021 |

• **Forma normală 3 (FN3)**

O relatie este în FN3 daca:

* se gaseste în FN2
* fiecare atribut care nu este cheie (nu participa la o cheie) depinde direct de cheia primara

Voi lua ca si exemplu tabela MATERIAL.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| #Id\_material | #Id\_fabrica | Nume | Cantitate | Categorie | Nume\_furnizor | Telefon |
| M1 | F1 | Lemn cires | 50 | Lemn | D1 | 0761456782 |
| M2 | F1 | Lemn stejar | 90 | Lemn | D2 | 0762674429 |
| M3 | F2 | Sticla neagra | 40 | Sticla | D3 | 0765567822 |

Aceasta tabela nu se afla in FN3, deoarece putem observa cum entitatea *Telefon* depinde de *Nume\_furnizor*, care depinde la randul sau de cheia primara *#Id\_material.*

Pentru a normaliza, vom separa cele doua atribute, mutandu-le in tabela FURNIZOR, iar in locul acestora, in tabela MATERIAL va aparea cheia straina *#Id\_furnizor*. Astfel apare relatia one-to-many intre FURNIZOR si MATERIAL.

MATERIAL:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| #Id\_material | #Id\_fabrica | Nume | Cantitate | Categorie | #Id\_furnizor |
| M1 | F1 | Lemn cires | 50 | Lemn | Cod1 |
| M2 | F1 | Lemn stejar | 90 | Lemn | Cod2 |
| M3 | F2 | Sticla neagra | 40 | Sticla | Cod3 |

FURNIZOR:

|  |  |  |  |
| --- | --- | --- | --- |
| #Id\_furnizor | Nume\_furnizor | Telefon | E-mail |
| Cod1 | D1 | 0761456782 | E1 |
| Cod2 | D2 | 0762674429 | E2 |
| Cod3 | D3 | 0765567822 | E3 |

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

b. Aplicarea denormalizării, justificând necesitatea acesteia.

**• Forma normală Boyce-Codd (BCNF)**

Forma normală Boyce-Codd se bazează pe dependenţele funcţionale care iau în consideraţie toate cheile candidat dintr-o relaţie.

Daca avem relatii cu o singura cheie candidat, atunci FN3 si BCNF sunt echivalente. (exemplu: FURNIZOR, FABRICA, PRODUS, CLIENT, CURIER)

Voi alege relația MATERIAL (#Id\_material, #Id\_fabrica, #Id\_furnizor, Nume , Cantitate ,Categorie)

Id\_furnizor

Id\_fabrica

Cantitate

Regula Casey Delobel pentru MATERIAL (#Id\_fabrica, #Id\_furnizor, Cantitate) având Cantitate -> Id\_fabrica

Asadar, rezulta

MATERIAL\_1(#Id\_furnizor, Cantitate)

MATERIAL\_2(Cantitate, #Id\_fabrica)

**• Forma nomală 4 (FN4)**

O relație este în a patra formă nomală dacă și numai dacă este în BCNF și nu conține relații m:n independente.

Pentru relația FURNIZOR (#Id\_furnizor, Nume , Telefon , E-mail) presupun ca un furnizor poate avea mai multe numere de telefon si mai multe nume (spre exemplu, numele firmei si numele personal al celor/celui ce o detin(e))

#Id\_furnizor -> nume

#Id\_furnizor -> telefon

Relația FURNIZOR este in BCNF. Pentru a aduce relația în FN4 o vom descompune prin proiecție în două relații:

FURNIZOR1(#Id\_furnizor, nume)

FURNIZOR2(#Id\_furnizor, telefon)

🡺 FURNIZOR = JOIN(FURNIZOR1, FURNIZOR2)

**• Forma normală 5 (FN5)**

O relaţie R este în FN5 (numită şi forma normală proiecţie-uniune) dacă şi numai dacă orice dependenţă de uniune a lui R este o consecinţă a unei chei candidat a lui R.

Orice relaţie care este în FN5 este şi în FN4, deoarece fiecare dependenţă multivaloare poate fi privită ca un caz particular de dependenţă de uniune. Orice relaţie poate fi descompusă fără pierderi la uniune într-o mulţime de relaţii care sunt în FN5.

Pentru a preciza dacă o relaţie este în FN5, este suficient să cunoaştem cheile candidate şi toate dependenţele de uniune din R.

Aducerea în FN5 presupune eliminarea join dependențelor.

Voi lua relația PRODUS si voi presupune că exista o join dependență în aceasta – iau mulțimea (Pret, Nume, Dimensiune) si presupun ca există multiple dependențe între fiecare dintre perechile din mulțimi (Pret, Nume), (Nume, Dimensiune) și (Pret, Dimensiune). Se pierde faptul că în acest caz Dimensiune se duce in Pret.

**• Denormalizare**

Denormalizarea este procesul invers al procesului de normalizare. Denormalizarea funcționează adăugând date redundante sau grupând date pentru a optimiza performanța.

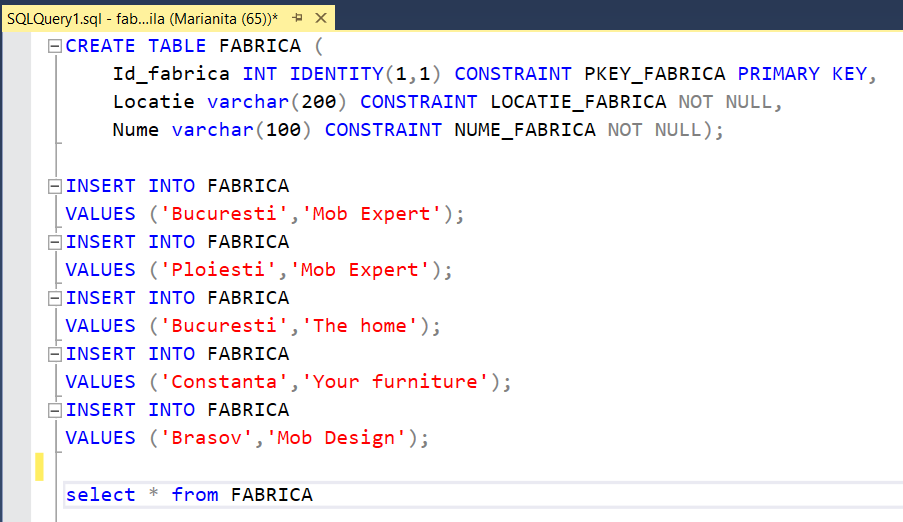
Chiar dacă adăugarea de date redundante pare contraproductivă, uneori denormalizarea este un proces foarte important pentru a depăși unele dintre deficiențele software-ului bazei de date relaționale, care pot suporta sancțiuni de performanță ridicate cu baze de date normalizate (chiar și pentru o performanță mai ridicată).

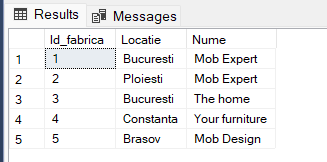
Acest lucru se datorează faptului că uneia dintre mai multe relații (care sunt rezultatul normalizării) pentru a produce un rezultat unei interogări poate fi uneori lentă în funcție de implementarea fizică reală a sistemelor de baze de date.

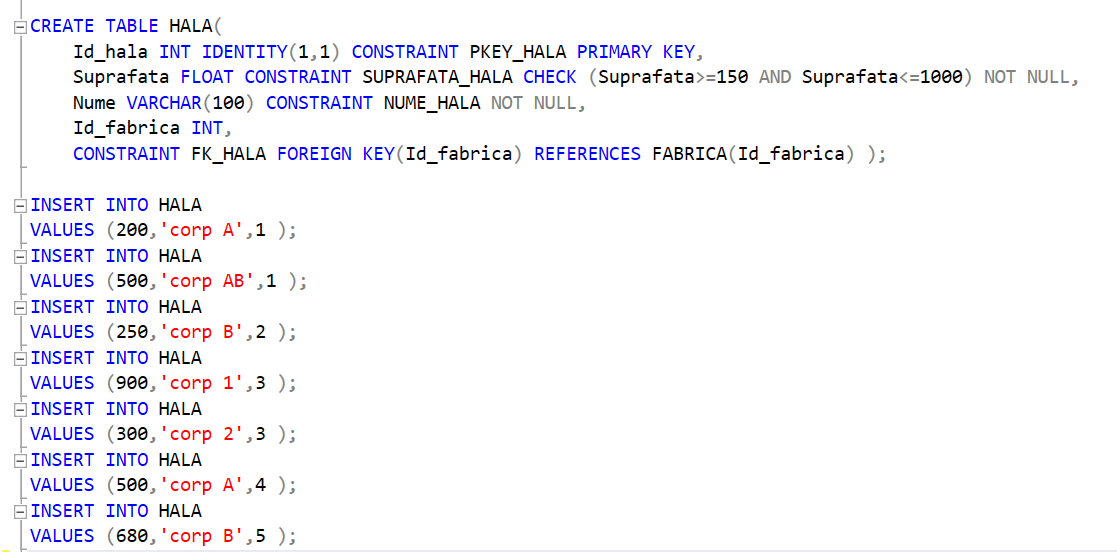
Astfel, plasând toate datele într-un singur loc, ar putea elimina necesitatea căutării acelor fișiere multiple pentru a colecta aceste date.

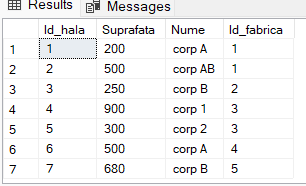
În cadrul bazei mele de date, luând spre exemplu relația FURNIZOR și descompunerea ei de la FN4, este inutil și mult mai costisitor din punct de vedere al timpului de executare să parcurgem datele si din FURNIZOR1 și FURNIZOR2.

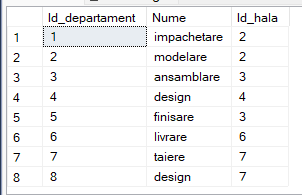
1. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).

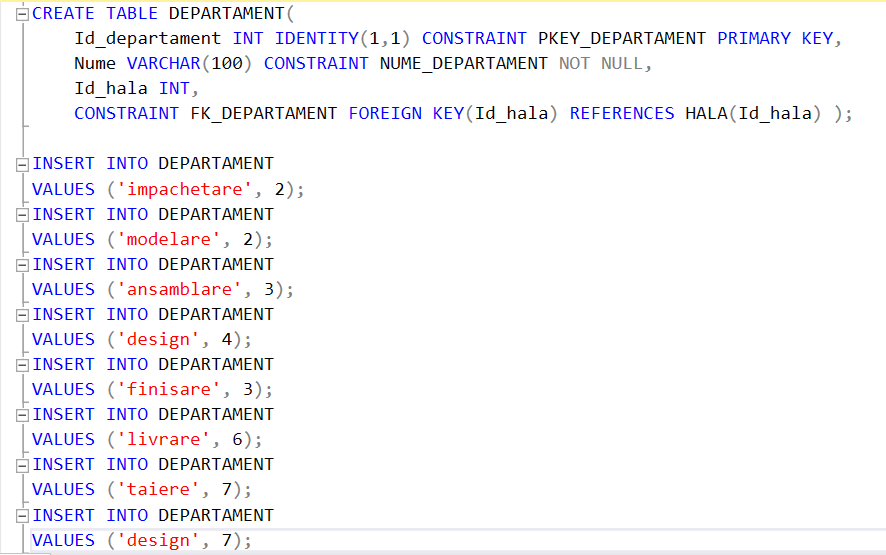


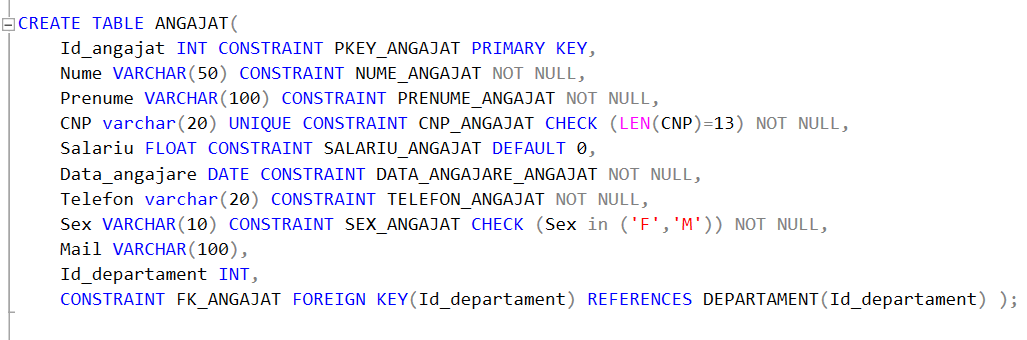


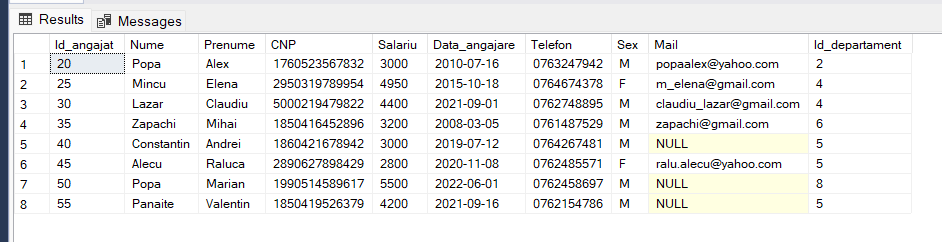




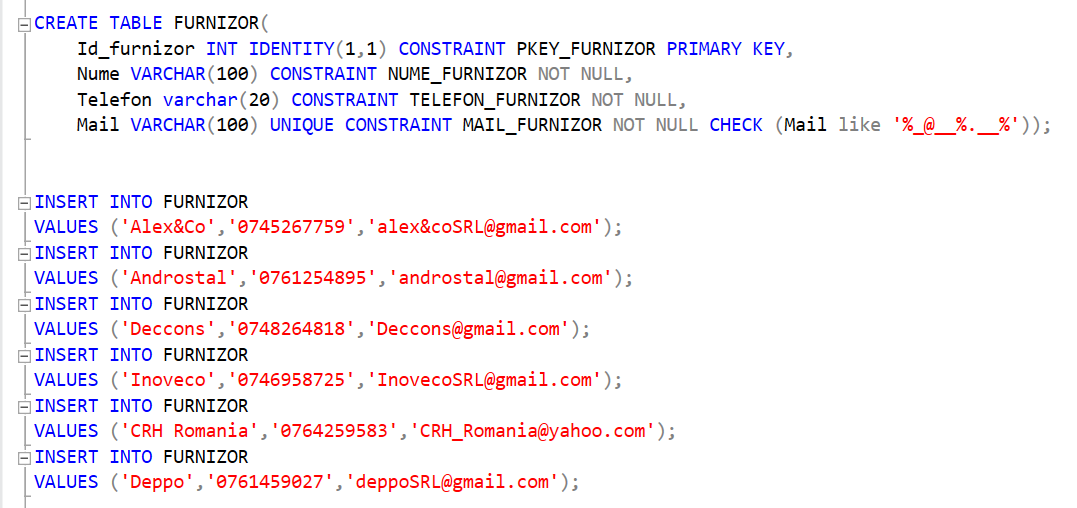




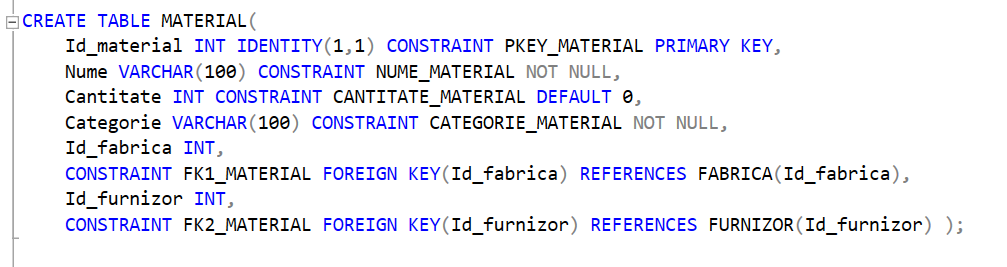


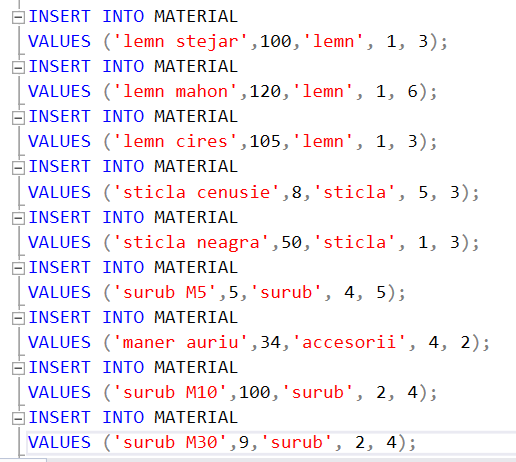


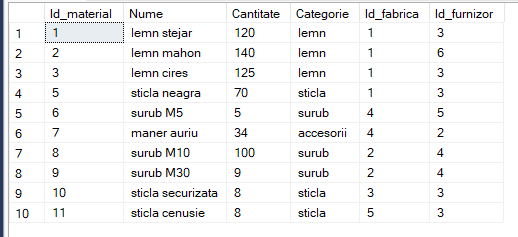


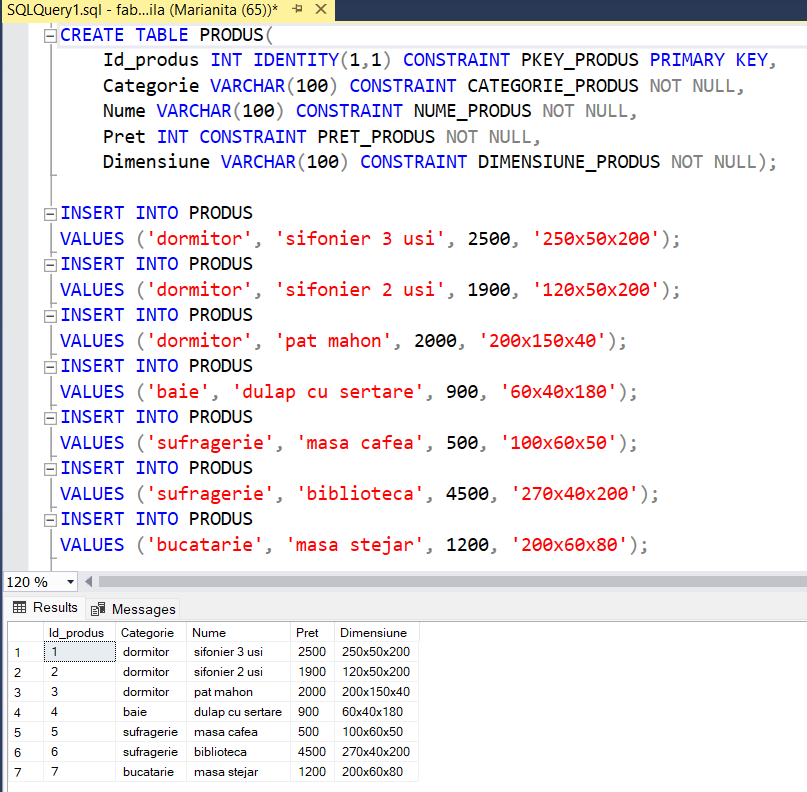


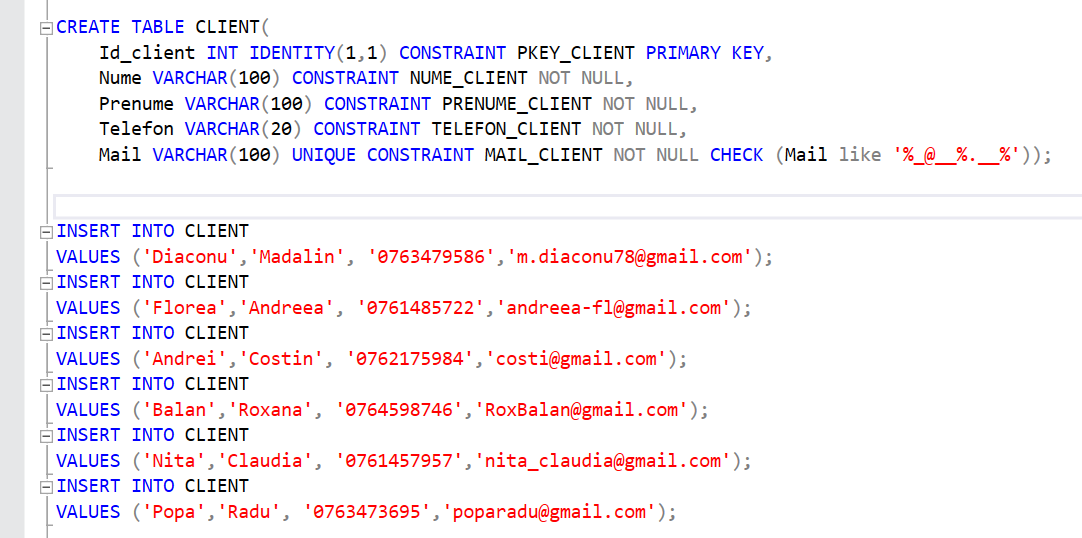


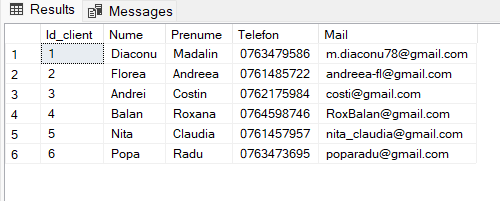


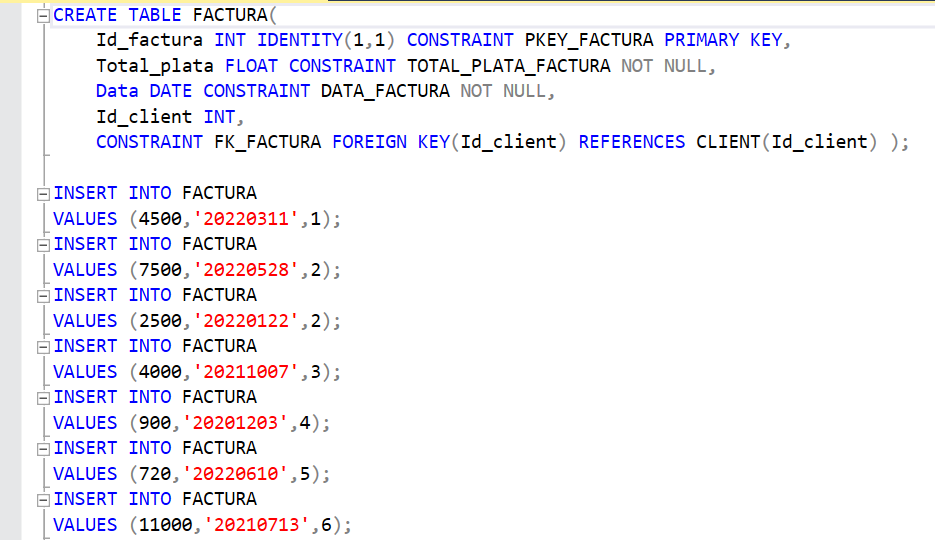


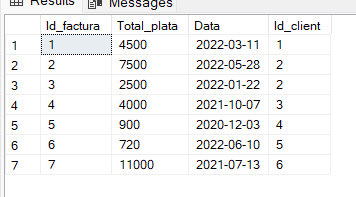


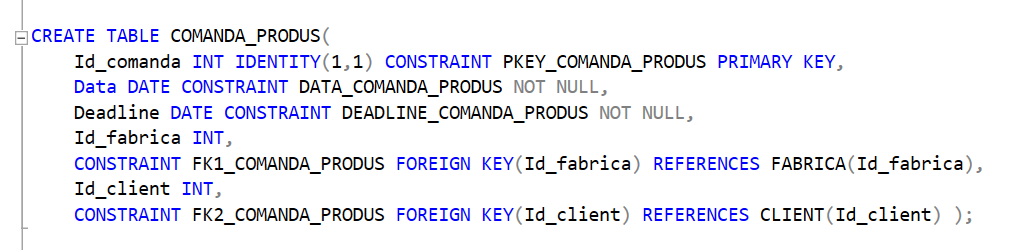


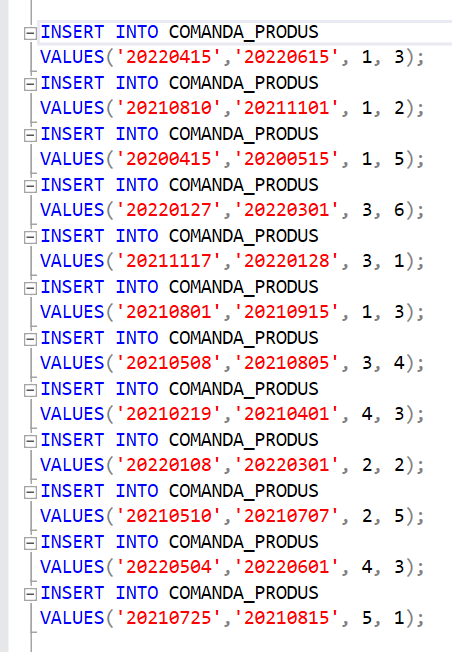


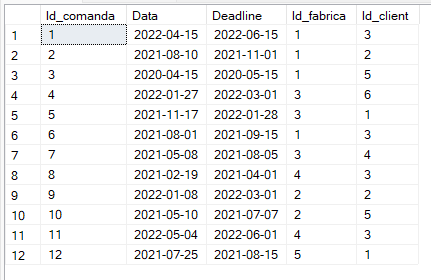


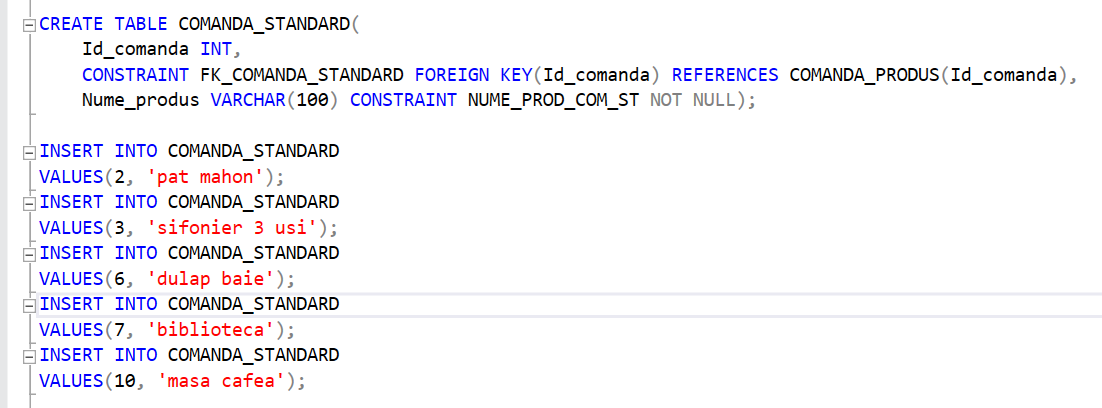


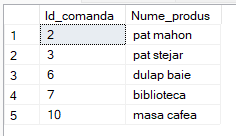




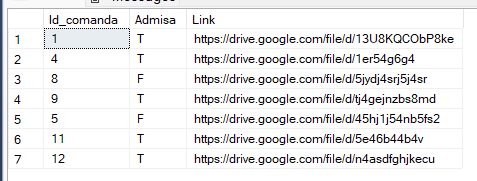


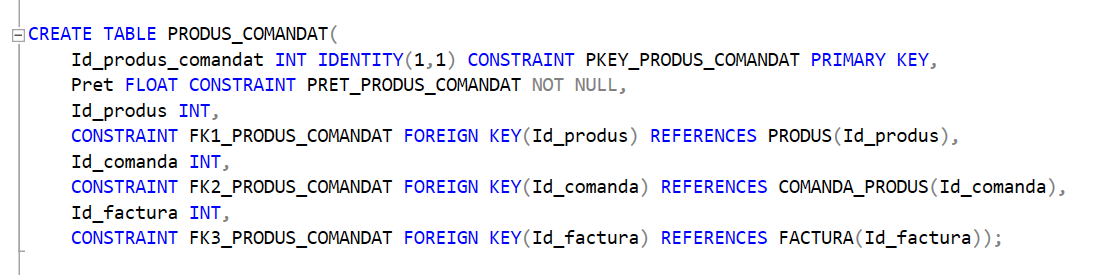


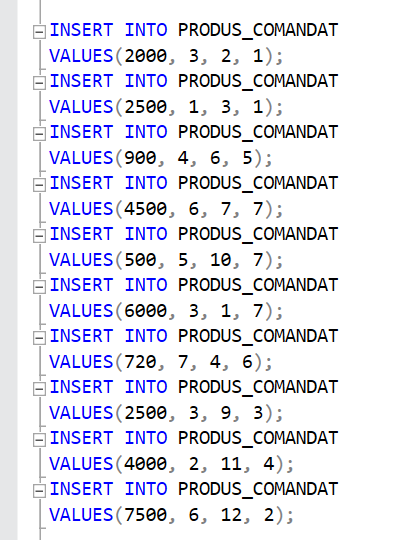


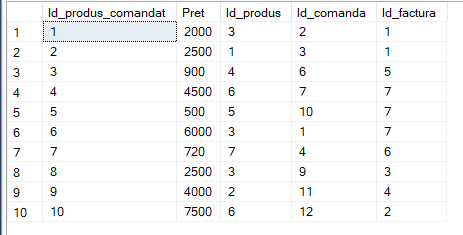


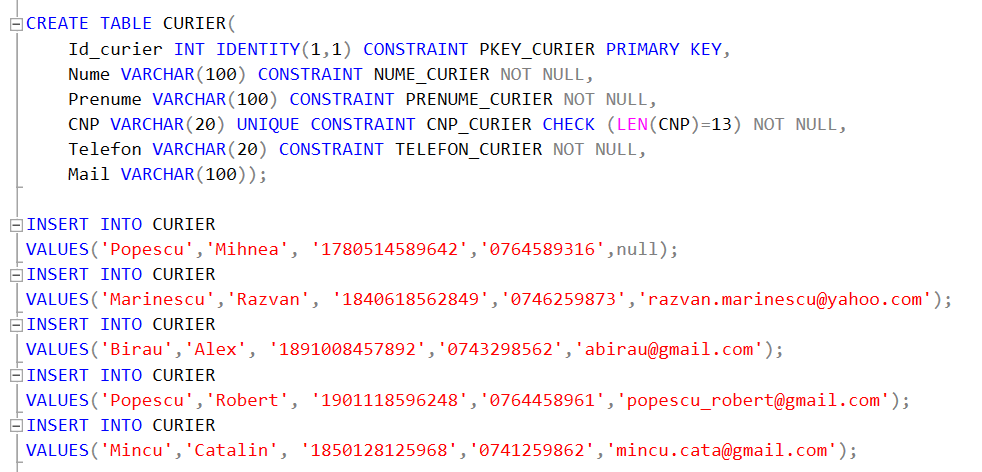


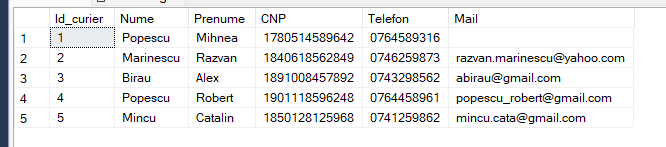


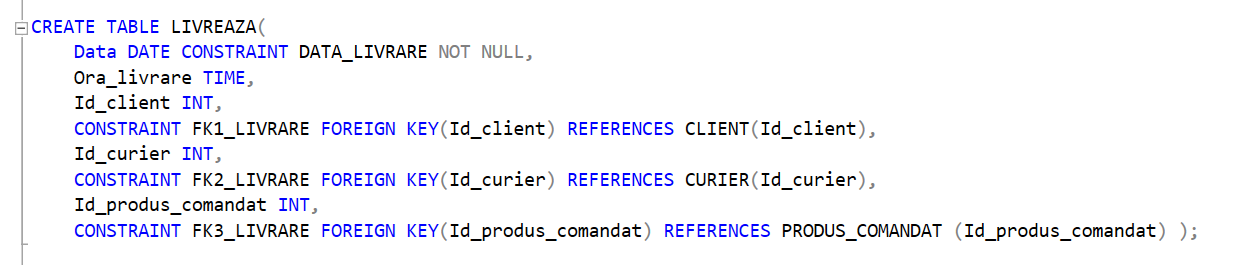


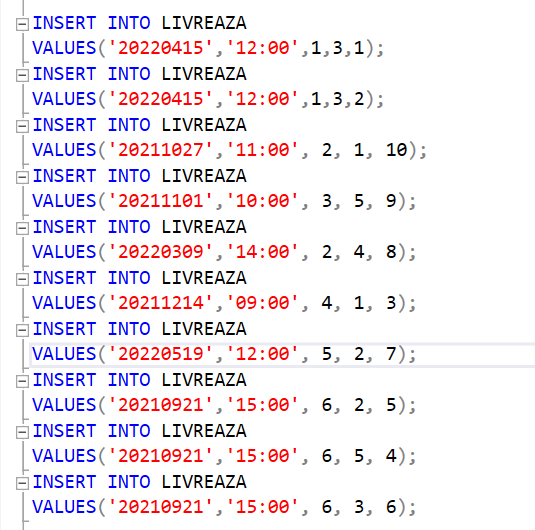


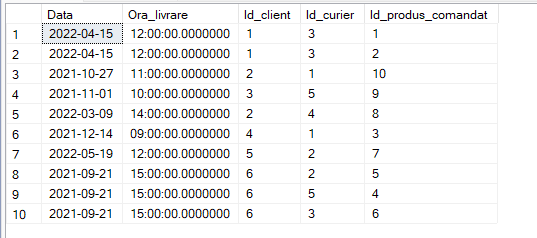












11. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:

• operație join pe cel puțin 4 tabele

• filtrare la nivel de linii

• subcereri sincronizate în care intervin cel puțin 3 tabele

• subcereri nesincronizate în care intervin cel puțin 3 tabele

• grupări de date, funcții grup, filtrare la nivel de grupuri

• ordonări

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

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

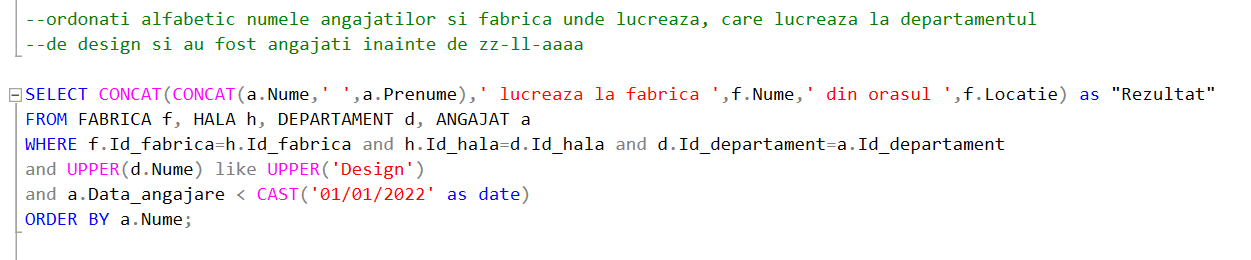
Id\_curier, nr prod livrate

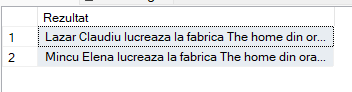
Select c.id\_curier, count(\*)

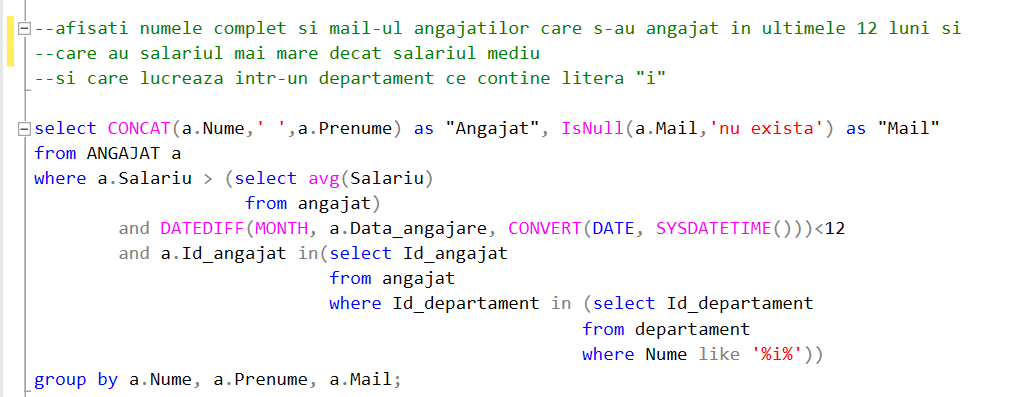
From curier c join livrare l left outer join produs\_comandat pc

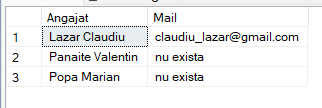
On c.Id\_curier = l.Id\_curier and l.Id\_produs\_comandat= pc. Id\_produs\_comandat

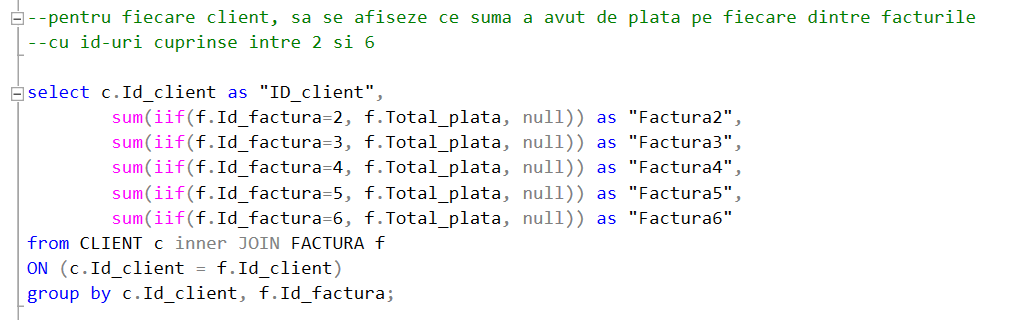
Group by Id\_curier

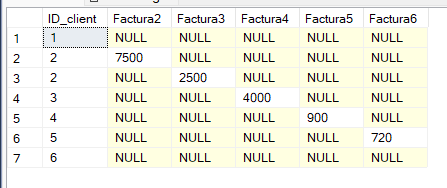


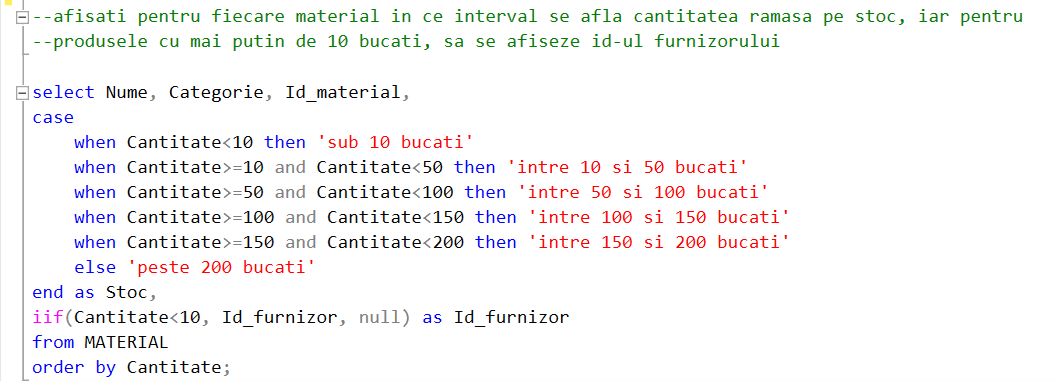


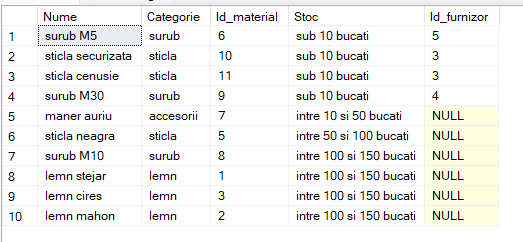


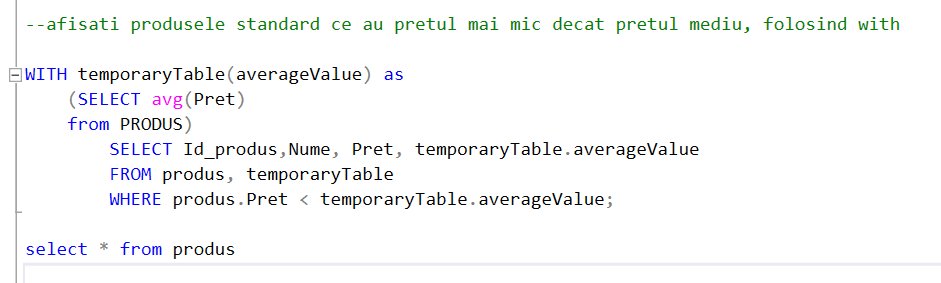


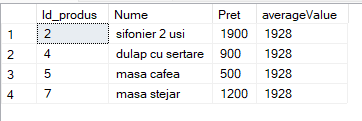




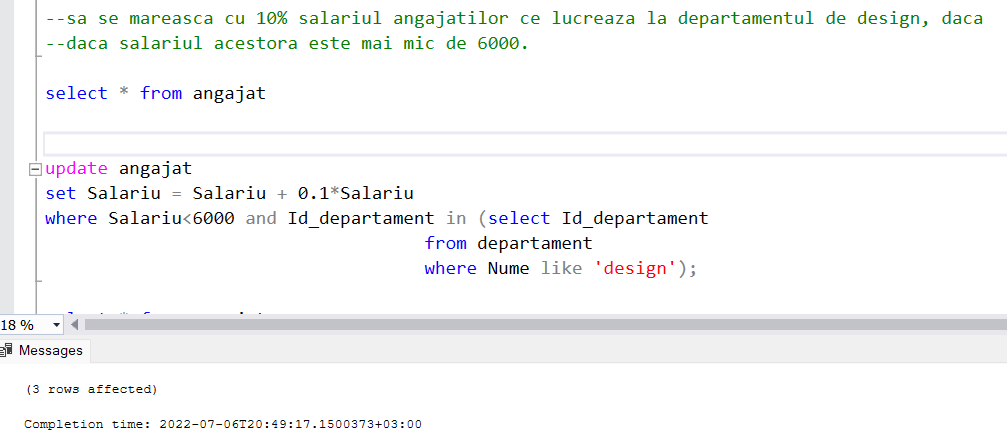


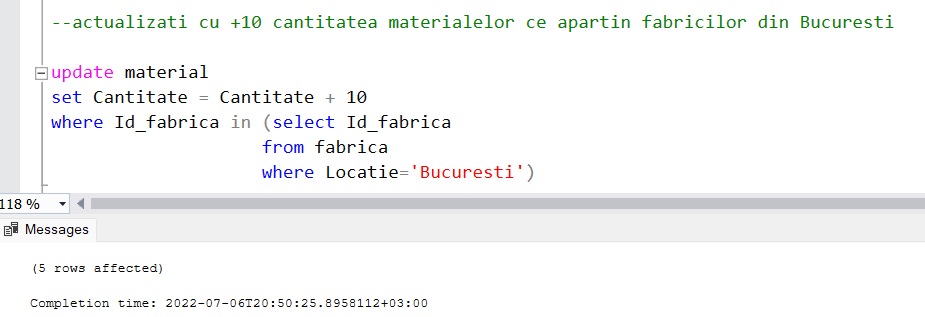


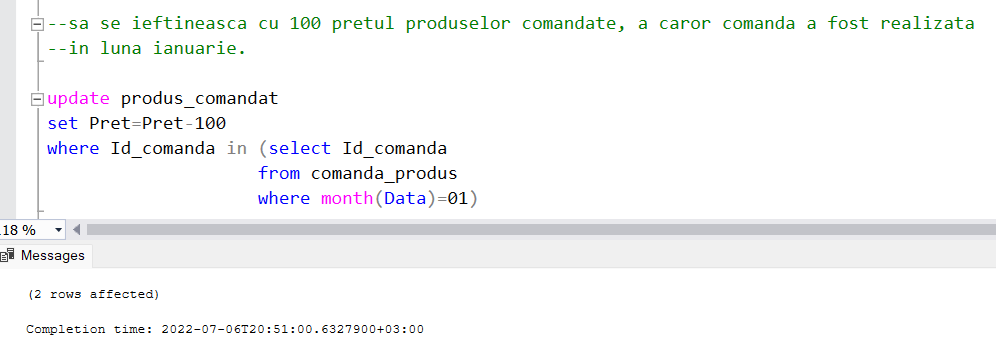




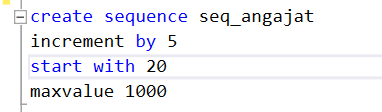
12. Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri.



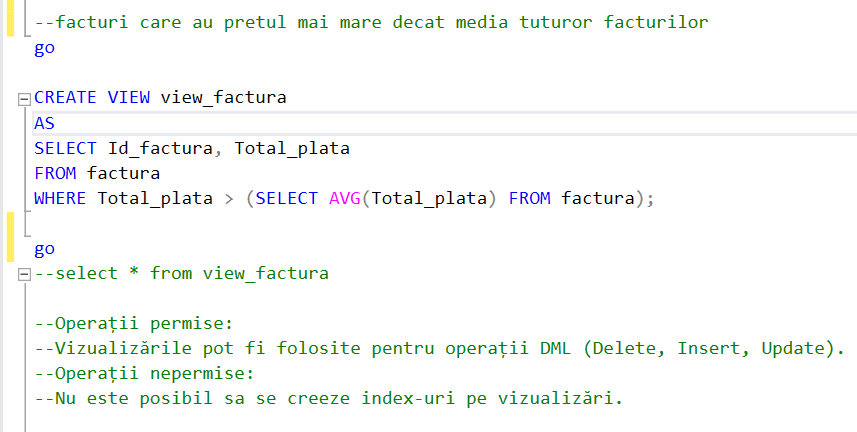


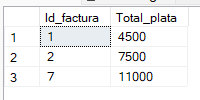


13. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele

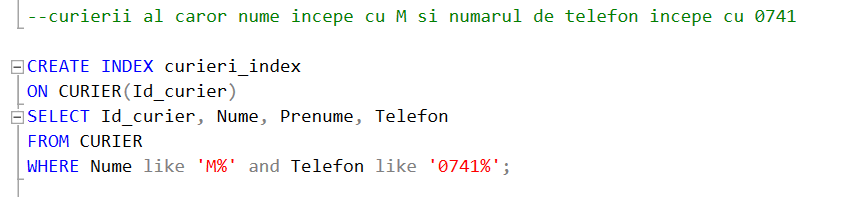


14. Crearea unei vizualizări compuse. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.



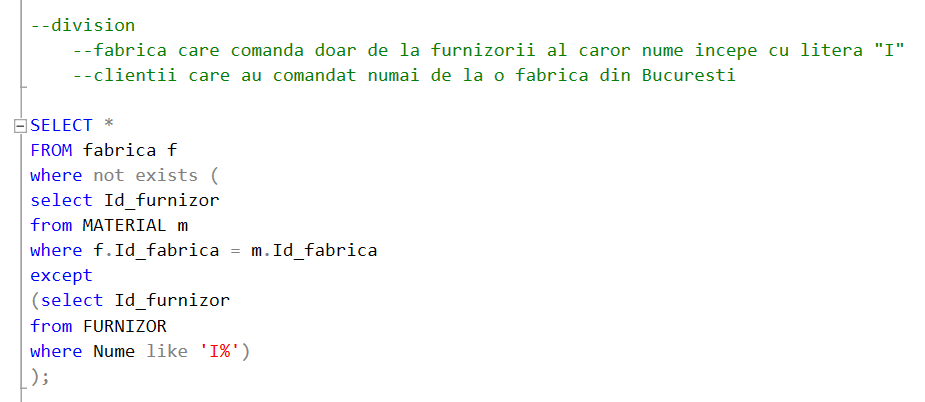


15. Crearea unui index care să optimizeze o cerere de tip căutare cu 2 criterii. Specificați cererea.

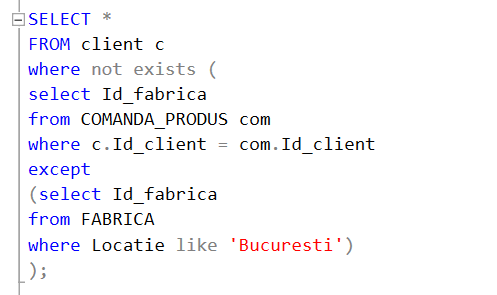


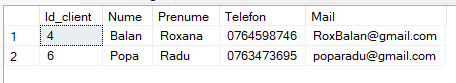


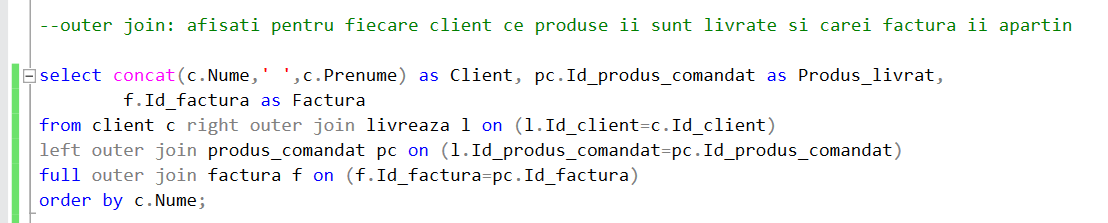
16. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minimum 4 tabele și două cereri ce utilizează operația division.

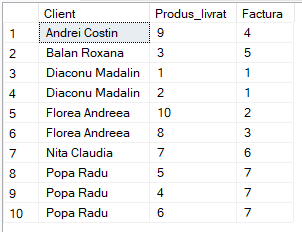








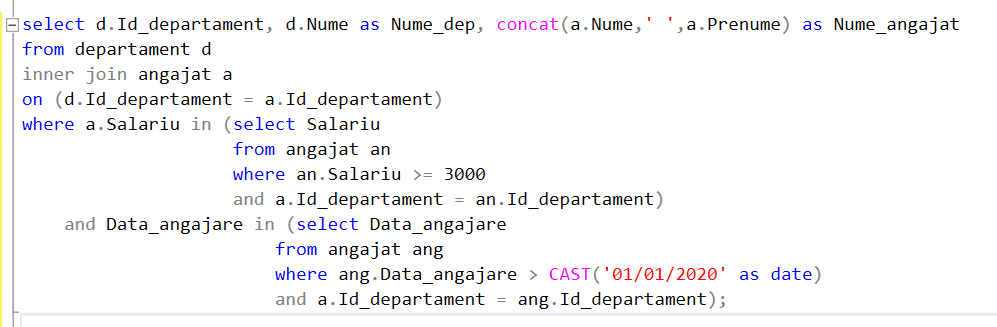




17. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebrică, arbore algebric și limbaj (SQL), atât anterior cât și ulterior optimizării.

--Sa se afiseze numele si id-ul departamentelor in care lucreaza angajati care au salariul mai mare de 3000 de lei si au fost angajati dupa 1 ianuarie 2020.

SQL:



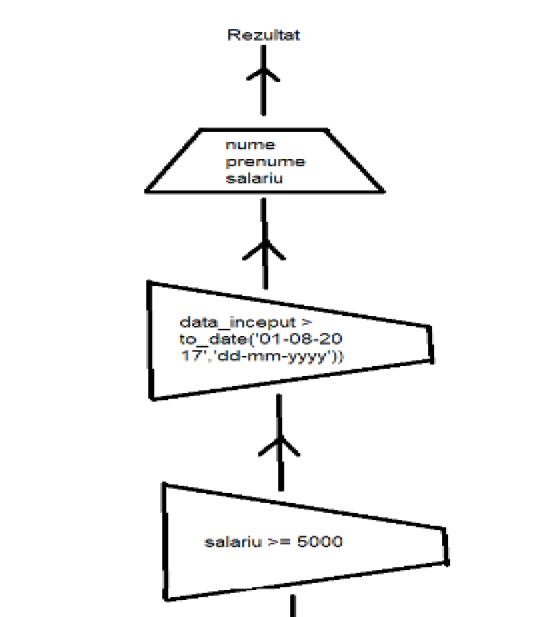
Expresie Algebrică:

R1 = JOIN(departament, angajat)

R2 = SELECT(R1, Salariu >= 3000)

R3 = SELECT(R2, Data\_angajare > CAST('01/01/2020' as date))

REZULTAT=R4=PROJECT(R3,Id\_departament, Nume, concat(angajat.Nume,' ',angajat.Prenume))



Salariu >= 3000

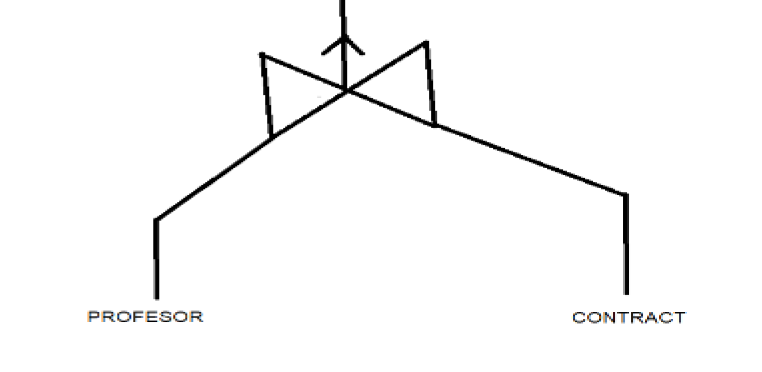
Data\_angajare > CAST('01/01/2020' as date)



REZULTAT

Nume, Prenume

concat(angajat.Nume, ' ',angajat.Prenume)

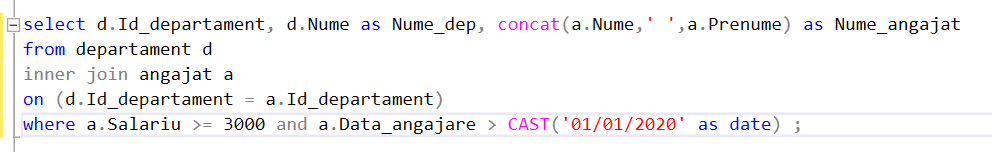


ANGAJAT

DEPARTAMENT

Etapa intermediara:

SQL:



Expresie Algebrică:

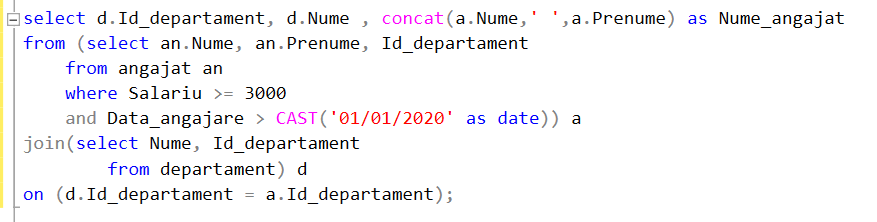
R1 = SELECT(angajat, Salariu >= 3000 and Data\_angajare > CAST('01/01/2020' as date))

R2 = JOIN(R1, departament)

R3 = PROJECT(R2, Id\_departament, Nume, concat(angajat.Nume,' ',angajat.Prenume))

Rezultat final:

SQL:



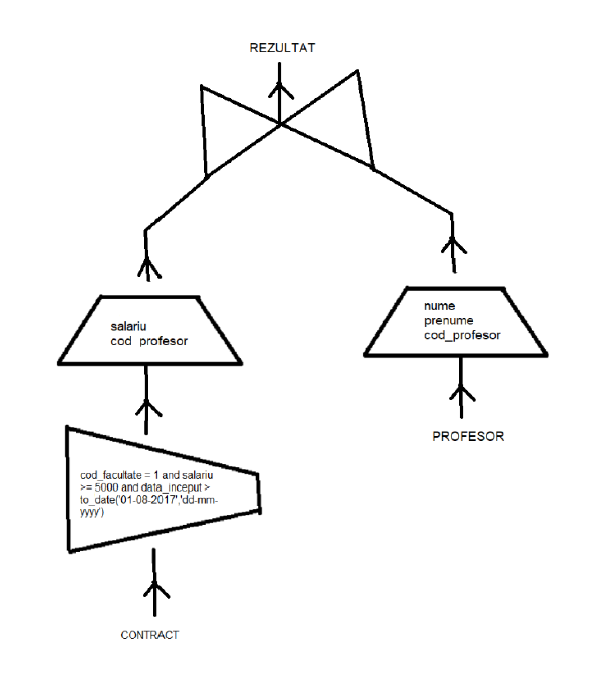
Expresie Algebrică:

R1 = SELECT(angajat, Salariu >= 3000 and Data\_angajare > CAST('01/01/2020' as date))

R2 = PROJECT(R1, Nume, Prenume, Id\_departament)

R3 = PROJECT(departament, Nume, Id\_departament)

REZULTAT = R4 = JOIN(R2, R3)



Salariu >= 3000 and

Data\_angajare >

CAST('01/01/2020' as date)

Nume, Prenume, Id\_departament

REZULTAT

DEPARTAMENT

ANGAJAT

Nume, Id\_departament