***Proiect Baze de Date***

**Gestionarea unei Scoli**

Realizat de: Tiperciuc Radu

**1.Introducere**

Educația joacă un rol important în dezvoltarea oricărei țări,iar una dintre cele mai mari provocari in cadrul educatiei este calitatea acesteia si imbunatatirea, respectiv evolutia educatiei care trebuie sa tina pasul cu noile avansuri tehnologice, iar in paralel trebuia sa mentina o legatura reala cu cerintele necesare din campul muncii. Astfel unul dintre aspectele acestei îmbunătățiri este gestionarea resurselor școlare.

Școala este o organizație care dezvoltă o cultură bogată și diversificată. Orice încercare de perfecționare a activității școlare trebuie să pornească de la cunoașterea culturii școlii, așa cum se manifestă ea la nivelul managerilor, cadrelor didactice, părinților și elevilor.

Organizația școlară reprezintă unitatea de bază a sistemului de învățământ specializat în realizarea procesului de învățământ conform obiectivelor pedagogice generale și specifice stabilite la nivel de politică educațională. Analiza organizației școlare din perspectiva normativă evidențiază importanța structurii instituției care reflectă, pe de o parte, imperativele unor funcții manageriale stabile (planificare-organizare; orientare-îndrumare metodologică; reglare-autoreglare prin acţiuni de cercetare-perfecționare pedagogică), iar pe de altă parte,cerinţele comunității educative naţionale, teritoriale, zonale şi locale, aflate in continuă schimbare.

**2.Descrierea proiectului**

Proiectul consta in gestionare unei baze de date a unei scoli gimanziale de dimensiuni medii, in ceea ce priveste personalul scolii, bugetul acesteia , calitatea invatamantului, dar si din punctul de vedere al numarului de elevi ce studiaza in cadrul acestei institutii.  
Sistemul de management al unei scoli reprezinta ansamblul de actiuni si de masuri planificate care oferă toate serviciile unui institut educațional pentru a vă face viața mai ușoară și mai rapidă, al carei scop principal este performanta si eficenta, dar si de a pastra un structura unui sistem ușor de gestionare a utilizatorilor.

In vederea constructiei bazei de date s-au tinut cont de urmatoarele aspecte:

* Împarte informațiile în tabele bazate pe subiecte pentru a reduce datele redundante.
* Oferă accesul cu informațiile necesare pentru a uni informațiile din tabele împreună, după cum este necesar.
* proiectarea bazei de date astfel incat sa nu prezinte probleme la actualizarea ei , chiar si de alt dezvoltator.
* Ajută la asistență și asigură acuratețea și integritatea informațiilor.

Proiectul este un sistem care vă va oferi o imagine de ansamblu asupra funcționării întregului sistem educațional instituţie. Este un sistem informațional de management care ajută la gestionarea diferitelor procese dintr-o instituție educațională, cum ar fi Administrația generală, managementul personalului, , managementul elevilor și parintilor,costurilor etc. Informațiile sunt astfel gestionate incat sa ajute la luarea deciziilor mult mai rapid, eficient și mai ușor ca niciodată. De asemenea, ajută la îmbunătățirea calității generale a educației instituției

Baza de date conține informații generale (cum ar fi numele, adresa, informațiile de contact, data de inscriere a elevilor , cursurile etc.), notele sau fișierul de rezultate etc. O bază de date automată a studenților eficientizează administrația scolii într-o măsură considerabilă.

Pe scurt, Software-ul de management al educației gestioneaza instituția de învățământ, in cazul de fata o scoala gimnaziala de dimensiuni medii, simplificând și automatizând procesele și răspunzând nevoilor tuturor părților interesate, ajutându-i să fie mai eficienți în rolurile lor respective

Pe scurt caracteristicile proiectului sunt:

* Este foarte ușor de utilizat pentru fiecare utilizator;
* Crește eficiența și reduce costurile;
* Introduce o gestionare electronica a scolii;
* Ajuta utilizatorii sa ia deczii mai repede si mai usor
* Ajuta la imbunatariea securitatii datelor cu caracter personal;
* Asigura o intretinere usoara a datelor;
* Asigura stocarea datelor pe termen lung;
* Utilizatorul bazei de date poate vedea toate informatiile neceare, iar dupa caz le poate modifica;
* Ofera acces la istoricul datelor stocate

**3.Tehnologii folosite :**

Proiectul este realizat in cadrul unuia dintre cele mai eficiente, cunsocute si utilizate sisteme de baze de date: SQL Server

Microsoft SQL Server este un sistem de gestionare de [baze de date](https://ro.wikipedia.org/wiki/Baz%C4%83_de_date) relaționale (RDBMS) produs de compania [americană](https://ro.wikipedia.org/wiki/SUA) [Microsoft](https://ro.wikipedia.org/wiki/Microsoft) Corp. Limbajul de interogare este SQL iar extensia procedurală este T-SQL.

Suportă versiunea companiei Microsoft de [SQL](https://ro.wikipedia.org/wiki/SQL) (Structured Query Language - limbaj structurat de interogări), cel mai răspândit limbaj de interogare pentru bazele de date. Este considerat un sistem de gestionare a bazelor de date pentru întreprinderi - adică se poate aplica bazelor de date de dimensiuni foarte mari – însă la ultimele versiuni au venit cu mai multe tipuri de licențiere, de la cele free la unele avansate, astfel că în prezent putem spune că SQL Server se poate folosi pentru baze de date de orice dimensiune.

Microsoft SQL Server folosește o variantă de SQL numită T-SQL, sau Transact-SQL, o implementare de SQL-92 (standardul ISO pentru SQL) cu unele extensii. T-SQL în principal adaugă sintaxă adițională pentru procedurile stocate și pentru tranzacții. Standardele SQL necesită ACID; acesta este un acronim pentru cele 4 condiții ce trebuie îndeplinite de orice tranzacție: atomicitate, consistență, izolare, durabilitate. MS SQL Server suportă ODBC (Open Database Connectivity).

Stocarea datelor se face pe modelul relațional, sub forma unei colecții de tabele cu rânduri și coloane. Fiecare coloană este de un anumit tip: SQL Server suportă atât tipurile primare, precum întreg, zecimal, caracter, dată calendaristică, dar și tipuri mai complexe precum text, date binare, date geometrice, date spațiale, XML, etc.

.

**4.Date tehnice :**

Scopul principal al bazei de date trebuie sa fie gestionarea datelor scolii, a datelor personale ale angajatilor, profesorilor, elevilor, dar si ale parintilor acestora, iar nu in ultimul rand baza de date trebuie sa contina toate datele reprezentative ale scolii, cum ar fi obiectele ce compun inventarul acesteia, cat mai precis, date despre constructia scolii, dar si despre biblioteca scolii, cu componentele caracteristice acesteia: inventarul cartilor, al imprumuturilor, al obiectelor fizice din biblioteca, salii de studiu si multe altele.

Astfel in cadrului proiectului, baza de date contine :

- 22 de tabele;

- 480 de elevi cu datele personale ale acestora;

- 480 de inregistrari cu parintii copiilor (ambii parinti in aceeasi inregistrare);

- date despre Scoala ;

- date despre inventar si constructive (cladire);

- 30 de profesori ;

- un consiliu format din 10 profesori ;

- Notele pentru elevi (2 note pentru fiecare elev);

- 10 materii scoalre (cu posibilitatea de marire a acestui numar );

- date despre biblioteca scolii;

- date despre bibliotecari (2 bibliotecari);

- 27 de titluri de carti , aflate in biblioteca in numar de la 1 pana la 10 exemplare, cat si date despre carti(nume, autor, pret etc );

- o evidenta clara a imprumuturilor cartilor (baza de date contine 119 imprumuturi);

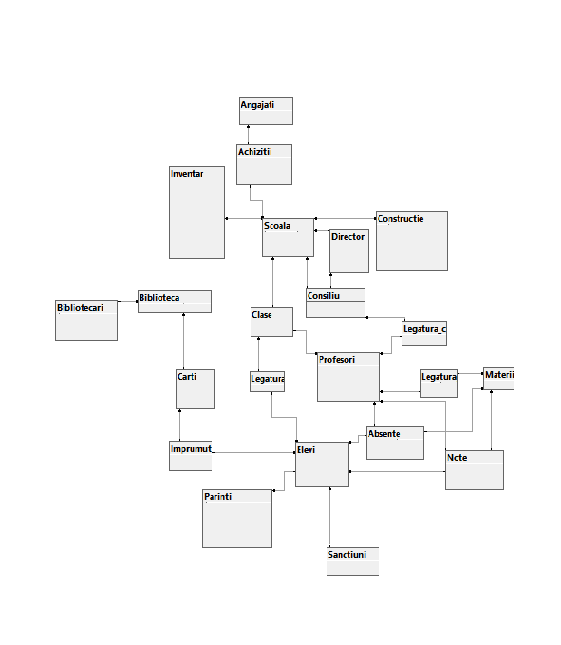
- o evidenta clara a achizitiilor facute ;

- o evidenta a sanctiunilor (8 la numar);

- informatii despre directorul scolii ;

- date despre angajatii scolii , altii decat profesorii ,

in numar de 15 , ce lucreaza in domenii precum : achizitii, curatenie , tamplarie , secretariat etc.

 **5.Tabelele din Baza de Date**

**Diagrama Bazei de Date**

Informatiile sunt stocate in tabele diferite astfel incat sa fie stocate toate datele dorite dar si sa fie reduse datele redundante,iar nu in ultimul rand trebuie pastratica logica funamentala de functionare a unei baze de date.

Inca de la inceput trebuie remarcat faptul ca cel mai important criteriu in construirea unei tabele este campul care contine cheia primara, Id – ul in cazul de fata.

Toate Id-urile trebuie sa fie unice, sa ofere un comfort administratorului in cazul prelucrarii acestora si tot odata sa fie usor de identificat. Astfel s-a creat urmatoarea regula de definre a acestora in cadrul proiectului:

* daca Id-ul este trecut cu albastru inseamna ca acela este doar prefixul id-ului.Metoda folosita in cazul tabelelor ce contin foarte mule inregistrari

\*Exemplu: tabela Elevi, Parinti etc

* daca Id-ul este portocaliu inseamna ca acel Id este unul fix, iar tabela are maxim 3 – 5 inregistrari.(Exemplu: tabela Scoala)

*Baza de Date* este compusa din urmatoarele tabele*:*

**SCOALA:**

ID\_Scoala(333)

Nume scoala

Adresa

**Constructie:**

Id\_Scoala

Nr clase

Nr laboratoare

Nr Sali de sport

Nr bazine de inot

Nr grupuri sanitare

Cancelarie profesori (bool)

Teren de fotbal / baschet/ handbal/ volei/tenis

Nr Sali Fitness

Biblioteca

**Inventar:**

Id\_scoala(333)

Clasa: nr.Scaune + nr.Banci + nr.Table+ nr.Bureti+nr.CutiiCreta + dulapuri

Laboratoare : calculatoare + scaune + mese + table + obiecte specifice

Sali\_de\_sport : echipamente(mingi, banci, saltele etc) + ventilatoare + nr\_dusuri + chiuvete + oglinzi

Grupuri sanitare: nr\_chiuvete + nr\_oglinzi + wc + nr\_sapunuri + uscatoare\_maini

Biblioteca : nr\_carti , nr\_banci, nr\_scaune

Utilizabile : Nr\_maturi + nr\_mopuri + nr\_galeti+ nr\_lopeti + foarfeci\_spatiu\_verde + nr\_becuri

**Elevi: -- - 480 elevi**

Id\_elev(10…) -> inregistrarile au toate prefixul ‘10’ (Exemplu: 101,10480)

Nume+Prenume

Data\_nastere

Nr\_telefon

E-mail

Data inscrierii

Id\_parinti

Cetatenie

**Parinti: -- - 480 inregistrari**

Id\_parinti(100..) -> inregistrarile au toate prefixul ‘100’ (Exemplu: 1001,100480)

Numele(+ Prenume) tatalui

Numele(+ Prenume) mamei

Nr\_telefon (x2)

E\_mail(x2)

Studii (gimnaziale/ liceale/ superioare)

Divortati(bool)

**Profesori: -- - 30 profesori**

ID(200..)

Nume + Prenume

Materie / Materii

E-mail

Nr telefon

Perioada in institutie (ani)

Vechime (experienta / ani de munca in domeniu)

Salariu

**Clasa: 🡪 24 clase : A \_ B\_ C**

ID\_clasa(13..)

Denumire clasa

Diriginte(ID\_profesor)

Desfiintata (bool)

Id\_scoala (333) - - > F.K.

**Legatura\_clase\_elevi**

Id\_clasa

Id\_elev

**Note: 2 note / elev**

Id\_nota(555..)

id\_elev

Materii -> nota

**Director:**

ID(30)

Nume+Prenume

Nr telefon

E-mail

Perioada(inceperea functiei)

Materia(Specialitatea (Ex: mate))

Salariu

Id\_scoala

**Sanctiuni: -- - 17 (total)**

Id\_sanctiune(14..)

Id\_elevi

Denumire sanctiune

Data

**Consiliu: -- - 10 profesori**

Id(444)

Denumire\_consiiliu

**Legatura\_consiliu\_profesori**

Id\_consiliu

Id\_profesori

**Achizitii: -- - 15 achizitii**

Id\_achizitie(15…)

Id\_responsabil achizitie 🡪 Id\_angajat (din tabela Angajati)

Id\_scoala la care se fac achizitiile

Data

Denumuire\_produs

Cantitate(nr. buc)

Pret

**Alti Angajati: -- - 10**

Id(16…)

Nume+Prenume

Domeniu de lucru (Achizitii, paza, mentenanta, instalator, tamplar)

Vechime in unitatea de invatamant

Salariu

**Biblioteca:**

(PK) ID(400)

ID Carti

ID Bibliotecari

Program

**Bibliotecari:**

Id (401+402)

Id\_biblioteca la care lucreaza

Nume + Prenume

Nr\_telefon

E-mail

Perioada in institutie

Salariu

**Carti: -- - 27 carti**

Id\_carte(888..)

Id\_bibliotca in care se afla

Nume

Autor

Pret

Nr exemplare in biblioteca

Categorie Carte - - - > SF/ Istorie / Dezvoltare personala/ Psihologie/ Actiune / Filosofie

**Imprumuturi: -- - 119**

(PK) Id inregistrare(23..)

Id\_carte

Id\_elev

Data

Durata

**Absente – 5 inregistrari**

(PK) Id\_absenta(210..)

Id\_elev

Data\_absenta

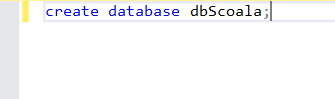
Motivata (DA / NU)

Id\_profesor

Id\_materie

**6.Crearea Bazei de Date**

Pentru a crea baza de date **dbScoala** am folosit instructiunea SQL :



**7.Crearea Tabelelor**

Baza de date **dbScoala** contine 19 tabele .

In crearea tabelelor am folosit 2 metode de creare si inserare a inregistrarilor :

1. Folosind comanda SQL : “CREATE” si dupa inseram inregistrarile cu comanda SQL : “INSERT” ;

2. Folosind comanda “Import Flat File “ in SQL Server;

Metoda 1 :

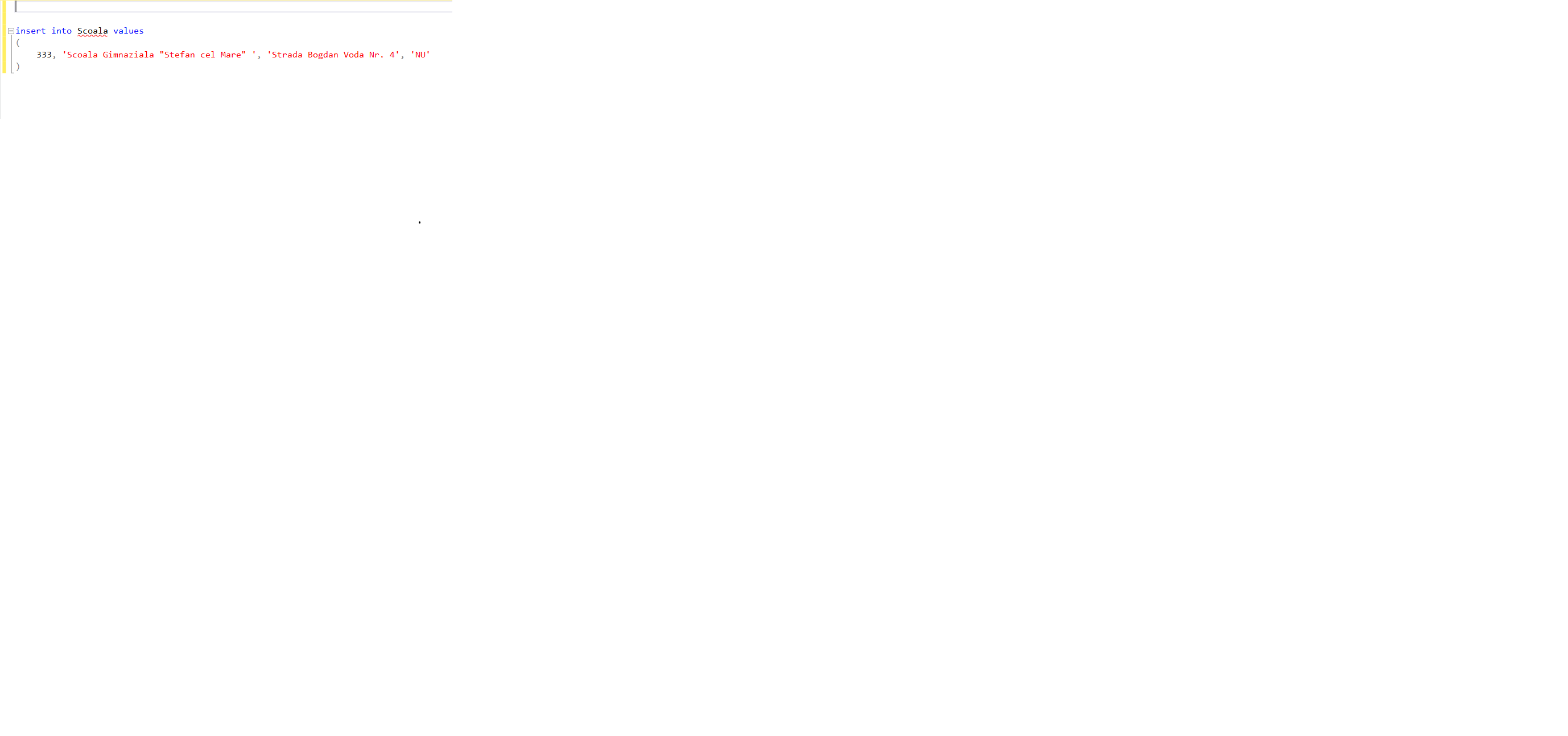
Prin prima metoda s-au creat urmatoarele tabele :

* Inventar
* Scoala
* Director
* Constructie
* Consiliu
* Biblioteca
* Bibliotecari
* Absente
* Legatura\_profesori\_materii

\* Exemplu de creare prin metoda 1:



\* Exemplu de inserare a inregistrarilor prin Metoda 1:



Metoda 2 :

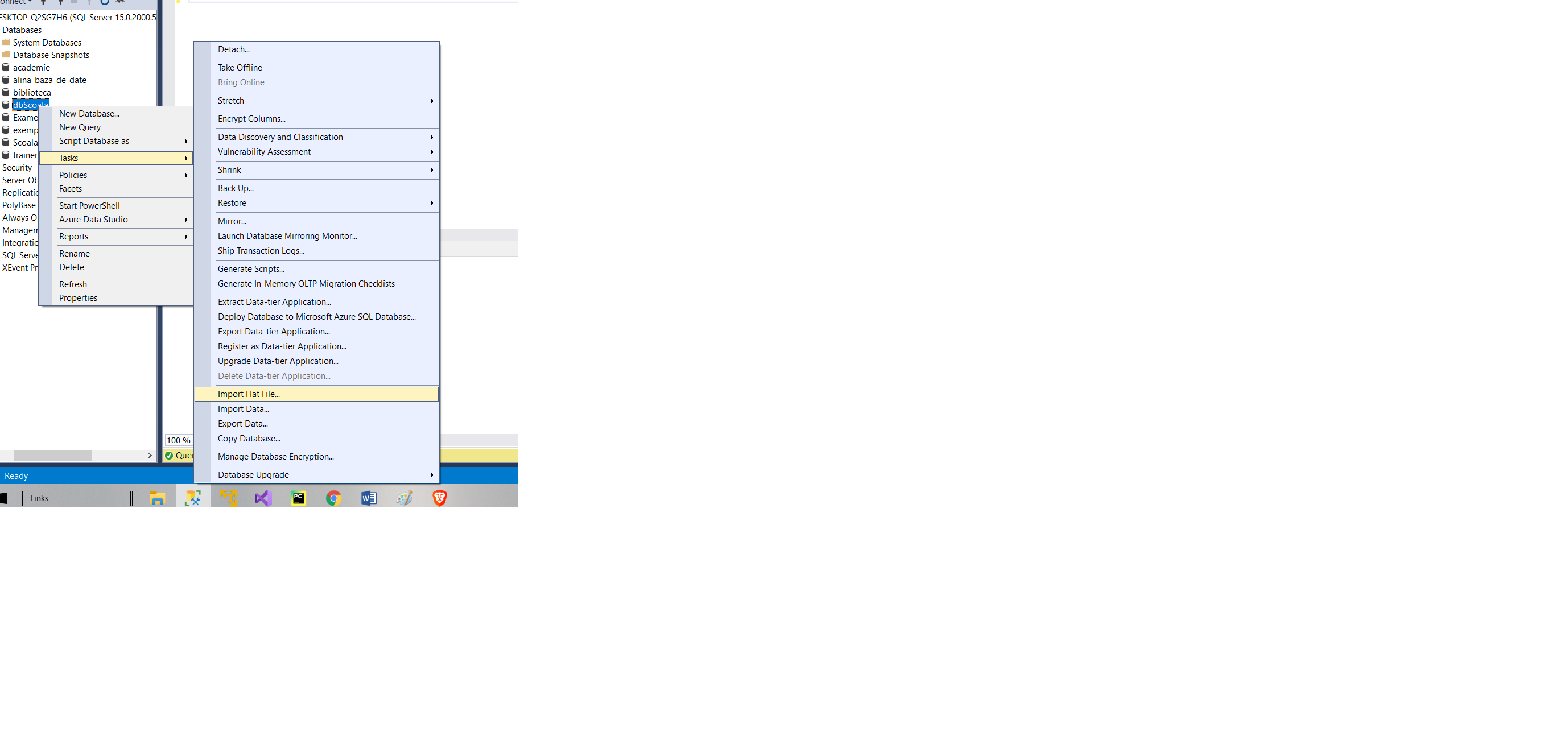
Prin aceasta metoda s-au creat urmatoarele tabele :

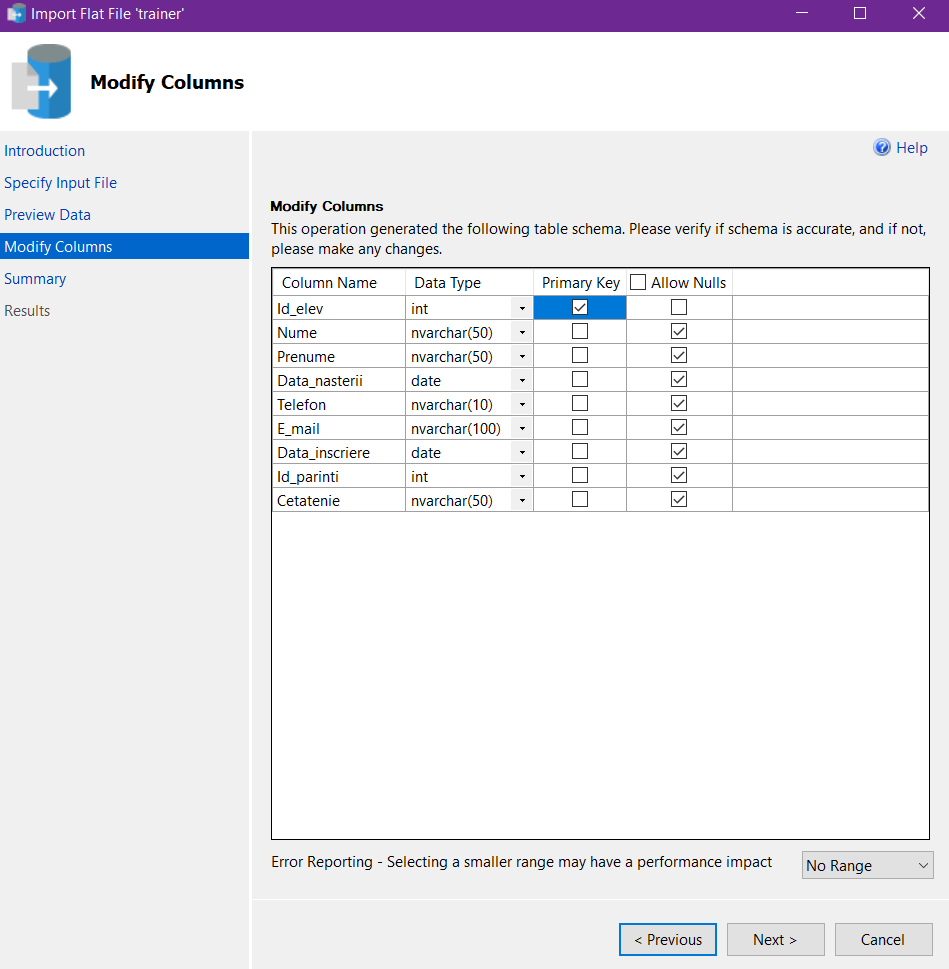
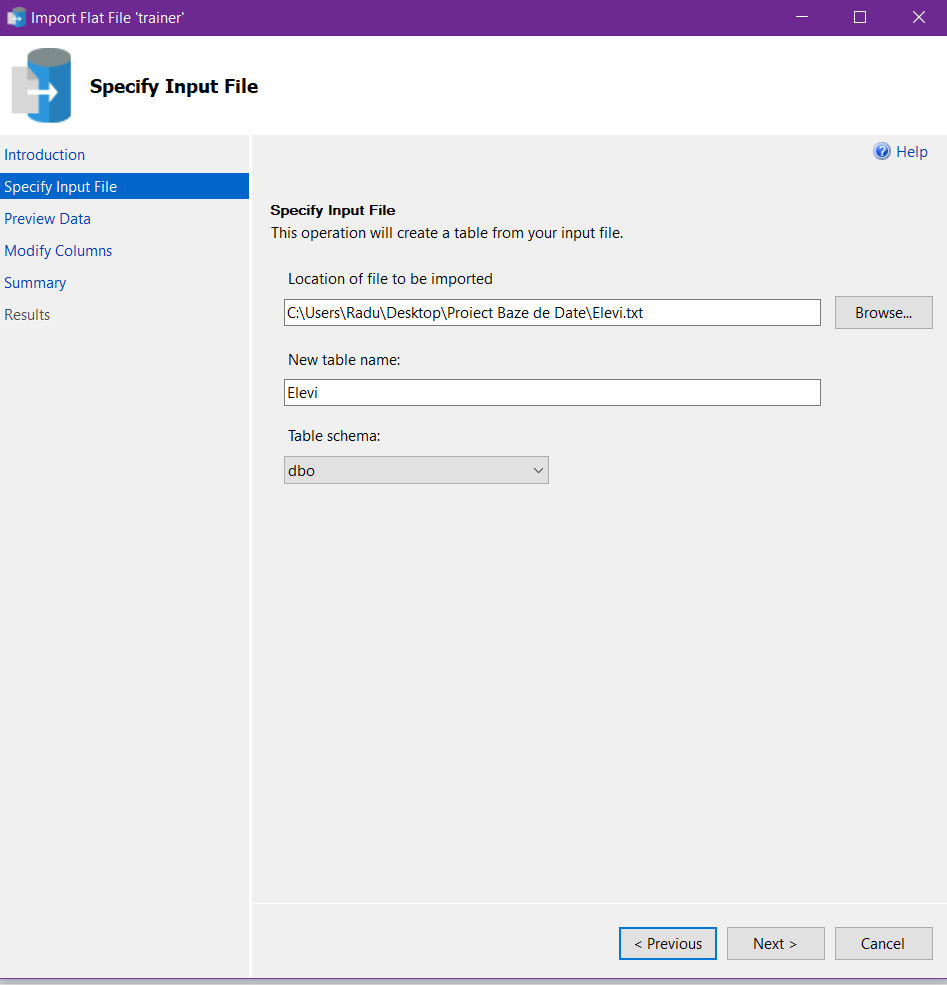
* Elevi
* Parinti
* Profesori
* Clase
* Legatura\_clase\_elevi
* Legatura\_consiliu\_profesori
* Angajati
* Achizitii
* Note
* Sanctiuni
* Imprumuturi
* Carti
* Materii

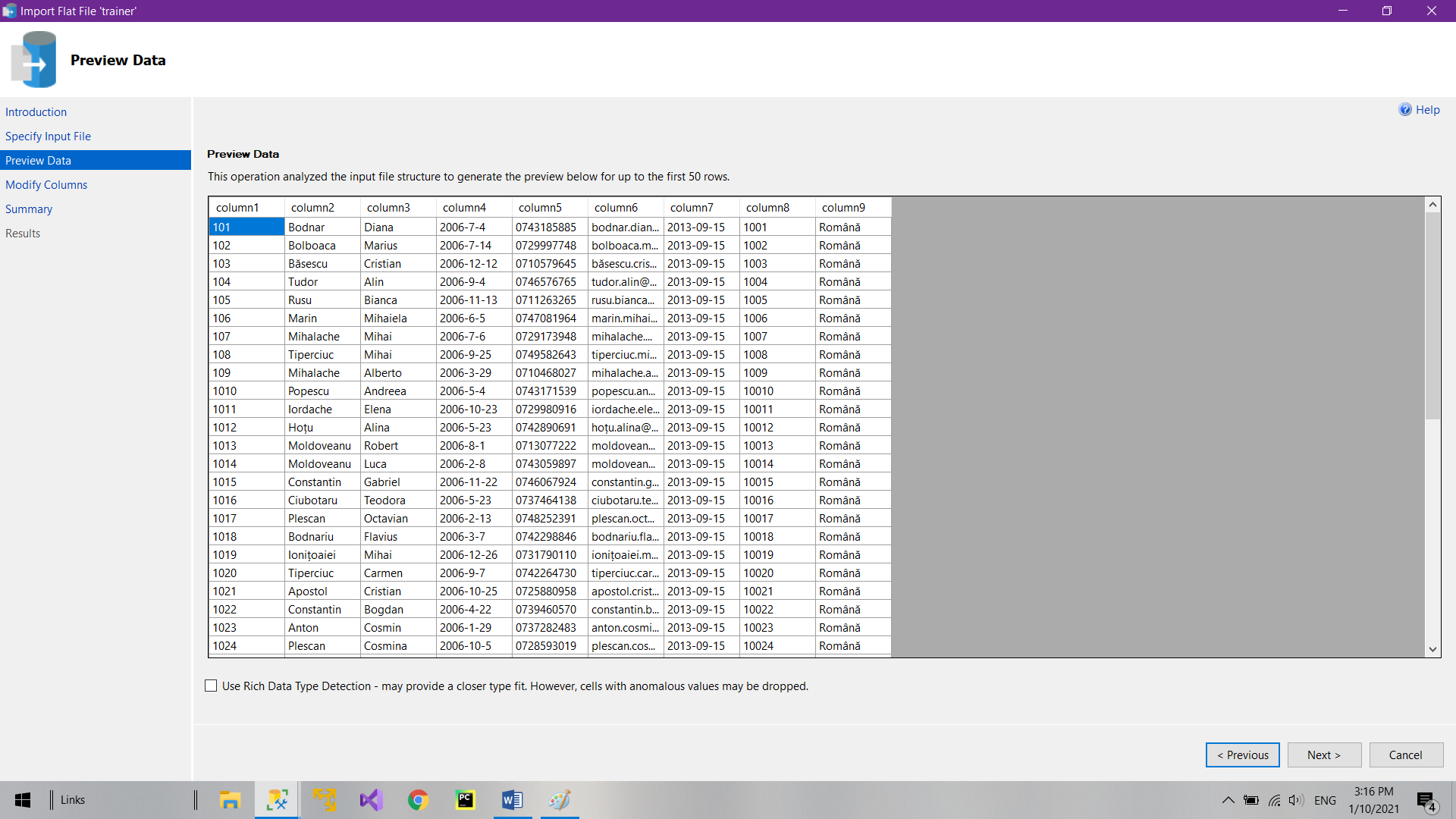
Pentru a folosi aceasta metoda trebuie sa avem un fisier exteren ‘.txt’ , care sa contina inregistrarile din tabelul pe care vrem sa il cream, dupa care din SQL Server vom folosi comanda : Import Flat File

Observatie : Folosind aceasta comanda tabelul se creeaza automat , noi doar trebuie sa definim campurile .

\* Exemplu de creare a unui tabel prin Metoda 2 :

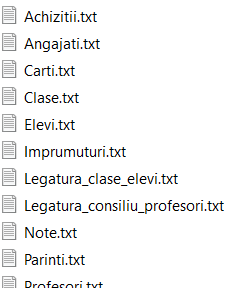


** **



**8.Crearea inregistrarilor in Python**

Pentru a folosi metoda a doua avem nevoie de fisiere exterioare , care sa contina datele(inregistrarile).

****

Pentru a crea date am folosind limbajul Python, iar datele au fost introduse cu comenzile COPY + PASTE, in fisierile ’.txt’ deja create.

**Prezentarea codului folosit in Python pentru a crea datele**

**1.Cod pentru crearea elevilor**

import random  
  
# need 30 Names  
Nume\_elevi\_l = [  
 'Anton', 'Tiperciuc', 'Hamza', 'Ionițoaiei', 'Radu', 'Apostol', 'Marin', 'Tudor', 'Bolboaca', 'Iordache'  
 , 'Constantin', 'Ponta', 'Băsescu', 'Amaruței', 'Ciubotaru', 'Bodnar', 'Bodnariu', 'Irimescu', 'Gheorge'  
 , 'Strugariu', 'Moldoveanu', 'Cristescu', 'Rusu', 'Sandu', 'Plescan', 'Popescu', 'Mortescu', 'Jacotă'  
 , 'Balmoș', 'Ciobâcă', 'Coțoveanu', 'Hoțu', 'Șpagoveanu', 'Mihalache', 'Ariton'  
 ]  
Prenume\_elevi\_l = [  
 'Andrei', 'Mihai', 'Gavril', 'Alin', 'Alina', 'Alexandra', 'Alex', 'Daniel', 'Dana', 'Carmen', 'Diana',  
 'Alin', 'Andreea', 'Teodora', 'Rebeca', 'Mihaiela', 'Flavius', 'Cristian', 'Ionut', 'Stefan', 'Marius',  
 'Bogdan', 'Cosmin', 'Petrica', 'Robert', 'Octavian', 'Luca', 'Vlad', 'Cosmina', 'Cristina', 'Bianca',  
 'Gabriel', 'Elena', 'George', 'Rafael', 'Alberto'  
 ]  
  
  
for i in range(1, 9):  
 for j in range(1, 61):  
 nume\_elev = random.choice(Nume\_elevi\_l)  
 prenume\_elev = random.choice(Prenume\_elevi\_l)  
 an\_nastere = 2005 + i  
 data\_nasterii = f'{an\_nastere}-{random.randint(1,12)}-{random.randint(1,31)}'  
 telefon = f'07{random.randint(100,500)}{random.randint(500,999)}{random.randint(10,99)}'  
  
 # data inscrierii : # anul de referinta este 2020  
 an\_incriere = str(7 + int(data\_nasterii[0:4]))  
 data\_inscrierii = f'{an\_incriere}-09-15'  
  
 id\_parinti = '100' + str((i-1)\*60+j)  
  
 Reccord = '10' + str((i-1)\*60+j) + ', ' + nume\_elev + ', ' + prenume\_elev + ', ' + data\_nasterii + ', ' + telefon + ', ' + \  
 nume\_elev.lower()+'.'+prenume\_elev.lower()+'@gmail.com' + ', ' + data\_inscrierii + ', ' + id\_parinti + \  
 ', ' + 'Română'  
 print(Reccord)

**2.Cod pentru crearea parintilor**

import random  
  
with open('prelucrare\_date.txt', 'r') as e:  
 lines = e.readlines()  
  
nume\_parinti\_l = [  
 'Anton', 'Tiperciuc', 'Hamza', 'Ionițoaiei', 'Radu', 'Apostol', 'Marin', 'Tudor', 'Bolboaca', 'Iordache'  
 , 'Constantin', 'Ponta', 'Băsescu', 'Amaruței', 'Ciubotaru', 'Bodnar', 'Bodnariu', 'Irimescu', 'Gheorge'  
 , 'Strugariu', 'Moldoveanu', 'Cristescu', 'Rusu', 'Sandu', 'Plescan', 'Popescu', 'Mortescu', 'Jacotă'  
 , 'Balmoș', 'Ciobâcă', 'Coțoveanu', 'Hoțu', 'Șpagoveanu', 'Mihalache', 'Ariton'  
 ]  
  
prenume\_tata\_l = [  
 'Cristi', 'Dumitru', 'Gheorghe', 'George', 'Petrică', 'Andrei', 'Marius', 'Lică', 'Ionut', 'Vladimir', 'Mihai',  
 'Octavian', 'Vasile', 'Costel', 'Ion', 'Nicu', 'Daniel', 'David', 'Iustin', 'Nelu', 'Ghiță', 'Antonio', 'Paul',  
 'Cosmin'  
 ]  
  
prenume\_mama\_l = ['Ana', 'Lăcrămioara', 'Elena', 'Gabriela', 'Aurica', 'Rodica', 'Gheorghița', 'Andreea',  
 'Diana', 'Alexandra', 'Daniela', 'Carmen', 'Mihaiela', 'Larisa', 'Antonia', 'Laura', 'Ionela',  
 'Cosmina', 'Bianca', 'Delia', 'Adriana', 'Ioana', 'Cristina', 'Doina', 'Dana', 'Monica', 'Andra'  
 ]  
  
for line in lines:  
 id\_e, nume\_e, prenume\_elev, d\_nastere, tel\_elv, mail\_elv, inscr\_elv, id\_parinti, cetatenie\_elv = line.split(',')  
  
 data\_nastere\_mama = f'19{random.randint(54, 89)}-{random.randint(1, 12)}-{random.randint(1,31)}'  
 data\_nastere\_tata = f'19{random.randint(54, 89)}-{random.randint(1, 12)}-{random.randint(1,31)}'  
  
 telefon\_mama = f'07{random.randint(100,500)}{random.randint(500,999)}{random.randint(50,99)}'  
 telefon\_tata = f'07{random.randint(500,900)}{random.randint(500,999)}{random.randint(10,50)}'  
  
 nume\_parinti = random.choice(nume\_parinti\_l)  
 prenume\_tata = random.choice(prenume\_tata\_l)  
 prenume\_mama = random.choice(prenume\_mama\_l)  
  
 e\_mail\_mama = f'{nume\_parinti.lower()}.{prenume\_mama.lower()}@gmail.com'  
 e\_mail\_tata = f'{nume\_parinti.lower()}.{prenume\_tata.lower()}@gmail.com'  
  
 cetatenie\_mama = 'Română'  
 cetatenie\_tata = 'Română'  
  
 Reccord = str(id\_parinti) + ', ' + nume\_parinti + ', ' + prenume\_mama + ', ' + data\_nastere\_mama + ', ' + \  
 telefon\_mama + ', ' + e\_mail\_mama + ', ' + cetatenie\_mama + ', ' + nume\_parinti + ', ' + prenume\_tata + ', ' + \  
 data\_nastere\_tata + ', ' + telefon\_tata + ', ' + e\_mail\_tata + ', ' + cetatenie\_tata  
 print(Reccord)

**3.Cod pentru crearea notelor**

import random  
  
with open('Elevi.txt', 'r') as elv:  
 lines = elv.readlines()  
  
for i, line in enumerate(lines):  
 id\_elev, nume\_elev, pren\_elev, d\_nastere, e\_tel, e\_mail\_elev, d\_insc\_elev, id\_parinti, cetatenie\_elv = line.split(',')  
  
 data = f'2014-{random.randint(1, 12)}-{random.randint(1,31)}'  
  
 Reccord = data + ', ' + id\_elev + ', ' + str(random.randint(2, 10)) + ', ' + str(random.randint(2, 10)) + ', ' + str(random.randint(2, 10)) + ', ' + str(random.randint(2, 10)) + ', ' + str(random.randint(2, 10)) + ', ' + str(random.randint(2, 10)) + ', ' + str(random.randint(2, 10)) + ', ' + str(random.randint(2, 10)) + ', ' + str(random.randint(2, 10)) + ', ' + str(random.randint(2, 10)) + ', ' + '555'+str(i+1)  
  
 print(Reccord)

**4.Cod pentru crearea angajatilor**

import random  
  
nume = ['Gusteru', 'Hîrtoapă', 'Cotleț', 'Ciubotariu', 'Apostol', 'Irimin', 'Velea', 'Grigorescu', 'Tomescu', 'Puiu',  
 'Badea', 'Moraru', 'Crihan', 'Ursan', 'Bălan']  
  
prenume = ['Andrei', 'Andreea', 'Iulia', 'Alexandru', 'Ionel', 'Ionut', 'Marinache', 'Petru', 'Ovidiu', 'Cristina', 'Marin']  
  
domeniu = ['contabilitate', 'achizitii', 'curatenie', 'secretariat', 'salubrizare']  
  
for i in range(1, 17):  
 vechime = random.randint(1, 25)  
  
 Reccord = '16' + str(i) + ', ' + nume[i-1] + ', ' + str(random.choice(prenume)) + ', ' + str(random.choice(domeniu)) +\  
 ', ' + str(vechime) + ' ani' + ', ' + str(vechime - random.randint(1, 10)) + ', ' +str(random.randint(1500, 4500))  
 print(Reccord)

**5.Cod pentru crearea achizitiilor**

import random  
  
angajati\_l = [1613, 1614]  
denumire\_produs\_l = ['sapun', 'cutii creta', 'bureti tabla', 'pixuri', 'tocuri hartie A4', 'mopuri', 'maturi', 'solutii curatenie', 'ustensile gradinarit']  
for i in range(1, 10):  
 data = f'2020-{random.randint(1, 12)}-{random.randint(1, 31)}'  
 cantitate = random.randint(75, 500)  
 Reccord = '15' + str(i) + ', ' + str(random.choice(angajati\_l)) + ', ' + '333' + ', ' + data + ', ' + \  
 str(random.choice(denumire\_produs\_l)) + ', ' + str(cantitate) + ', ' + str(round((cantitate \* random.uniform(1.255, 23.65)), 3))  
 print(Reccord)

**6.Cod pentru crearea sanctiunilor**

import random  
  
sanctiuni\_l = ['preaviz de exmatriculare', 'mustrare scrisă', 'observație individuală',  
 ' retragere temporară a bursei de merit', ' retragere definitivă a bursei de merit'  
 ]  
  
for i in range(1, 9):  
 data = f'2020-{random.randint(9,11)}-{random.randint(1, 31)}'  
 Reccord = '14' + str(i) + ', ' + '10' + str(random.randint(1, 481)) + ', ' + random.choice(sanctiuni\_l) + ', ' + data  
 print(Reccord)

**7.Cod pentru crearea claselor**

den = [  
 'Clasa a-I-a A', 'Clasa a-I-a B', 'Clasa a-I-a C','Clasa a-II-a A', 'Clasa a-II-a A', 'Clasa a-II-a B',  
 'Clasa a-III-a A','Clasa a-III-a B','Clasa a-III-a C','Clasa a-IV-a A','Clasa a-IV-a B','Clasa a-IV-a C','Clasa a-V-a A',  
 'Clasa a-V-a B','Clasa a-V-a C','Clasa a-VI-a A','Clasa a-VI-a B','Clasa a-VI-a C','Clasa a-VII-a A','Clasa a-VII-a B',  
 'Clasa a-VII-a C','Clasa a-VIII-a A','Clasa a-VIII-a B', 'Clasa a-VIII-a C'  
 ]  
  
for i in range(1, 25):  
 Reccord = '13' + str(i) + ', ' + den[i-1] + ', ' + '200' + str(i) + ', ' + 'NU' + ', ' + '333'  
 print(Reccord)

**9.Interogari**

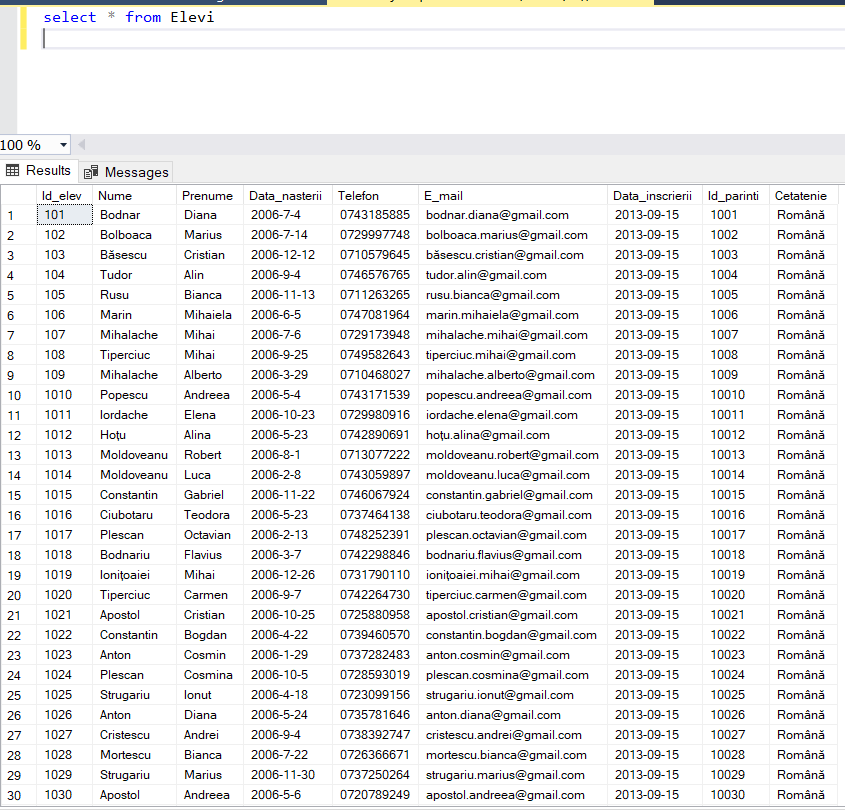
In cadrul proiectului s-au efectuat 120 de exemple de interogari folosind urmatoarele tipuri de interogari:

* Interogari cu Select
* Interogari cu Joinn-uri
* Interogari cu Proceduri Stocate
* Interogari cu View-uri
* Interogari Declansatoare

Pentru a economisi timp si spatiu, in documentatie sunt trecute doar primele 10 exemple din fiecare categorie cu cerinta cod SQL si date de iesire, iar restul exemplelor sunt compuse doar din cerinta si codul SQL valabil.

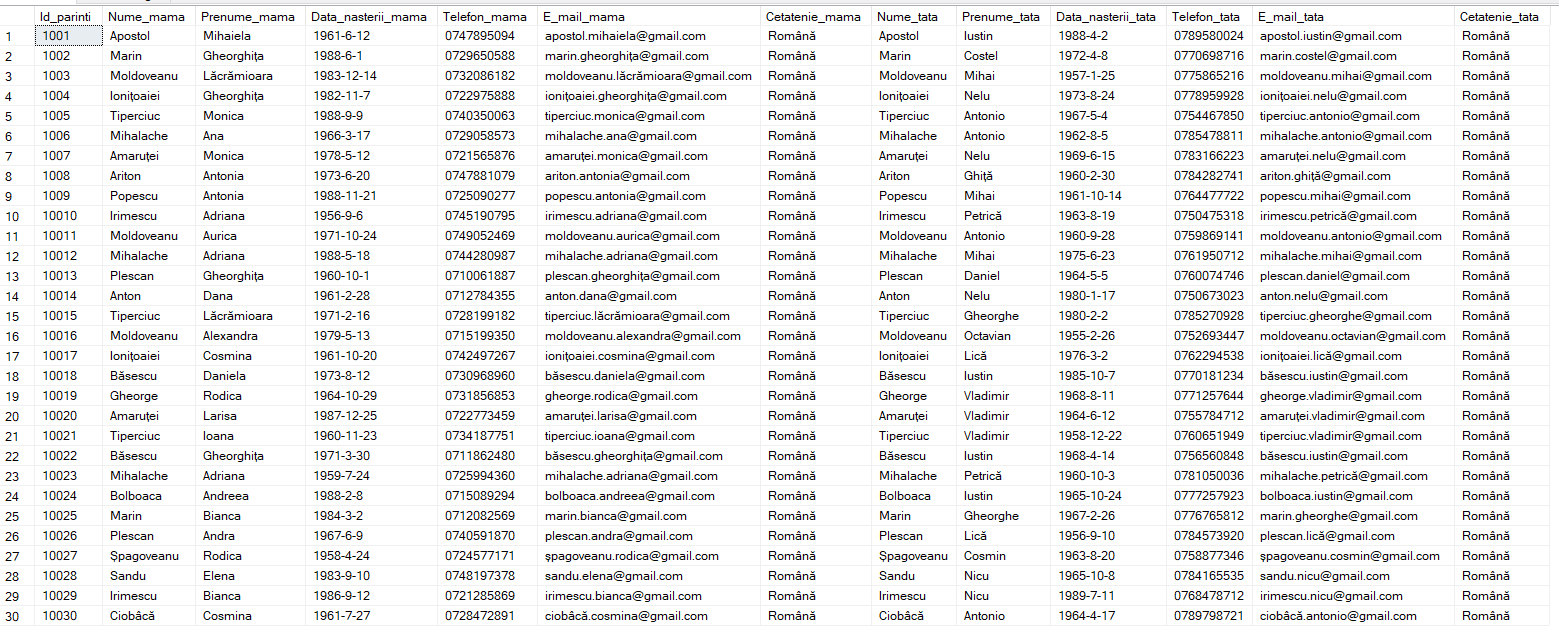
**I**.Interogarile cu Select

1) Selectati toate inregistrarile din tabela Elevi:

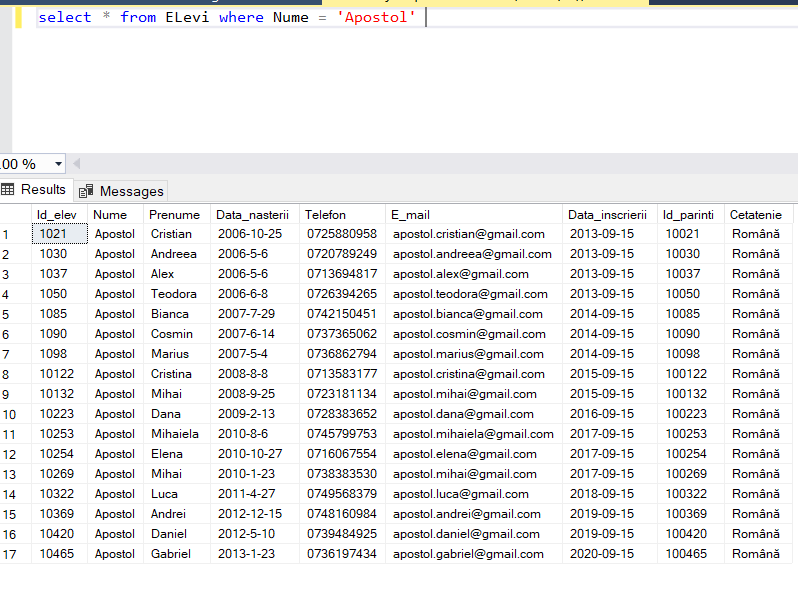


2) Afisati primele 30 de inregistrari cu parintii elevilor

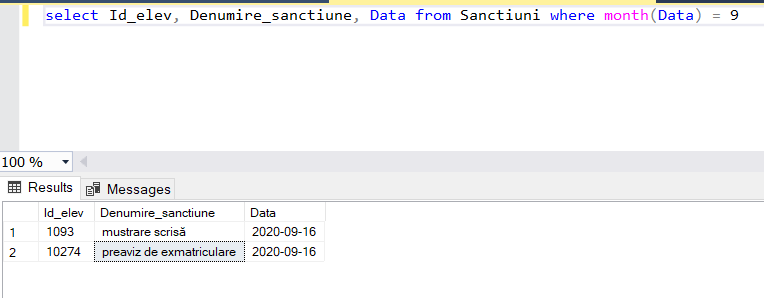




3) Afisati toti elevi al caror nume este “Apostol “

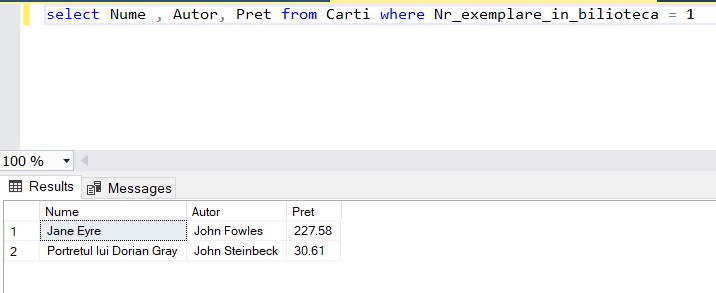


4) Afisati toti elevi care au luat o sanctiune in luna septembrie

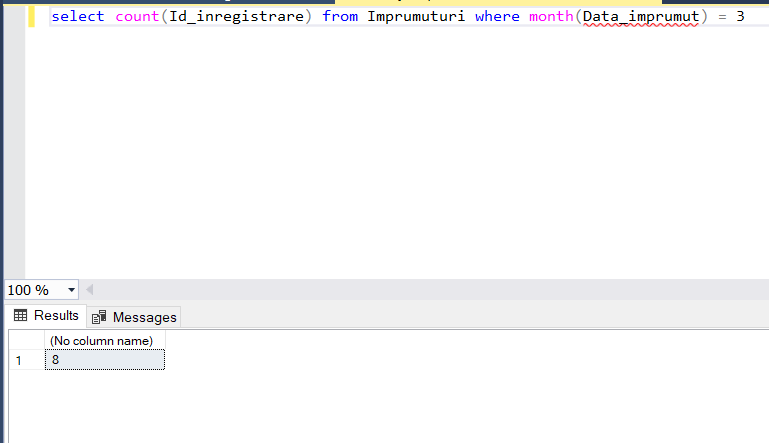
(afisati atat denumirea sanctiunii , cat si data completa)****

5) Afisati cartile care se afla intr-un singur exemplar in biblioteca

(se afiseaza numele , autorul si pretul)

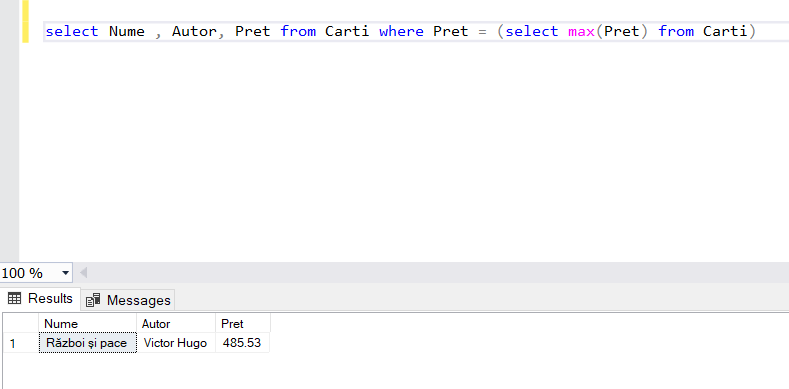


6) Afisati cate carti s-au imprumutat in luna martie

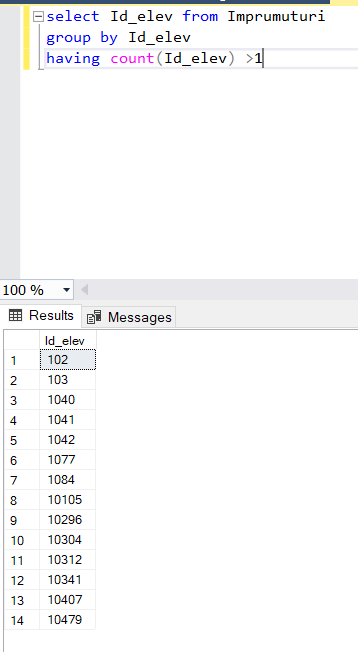


7) Sa se afiseze cea mai scumpa carte

(se va afisa numele , autorul si pretul cartii)

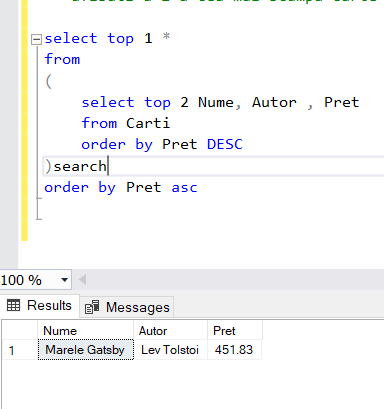


8) Afisati elevi care au imprumutat mai mult de o carte



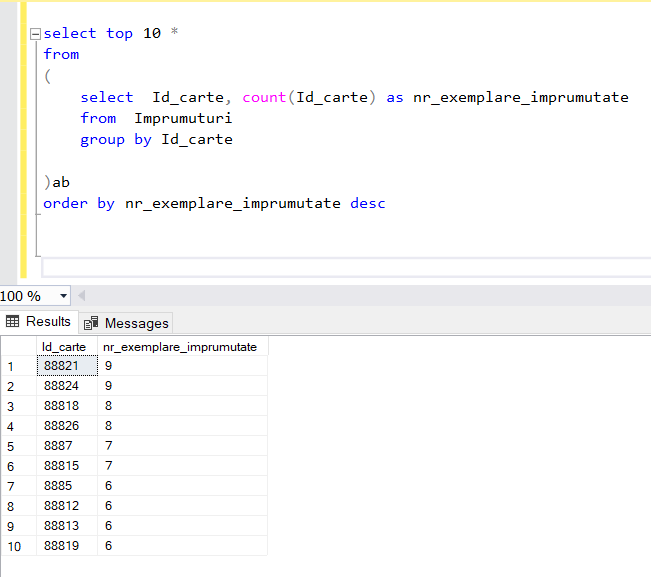
9) Sa se afiseze a doua cea mai scumpa carte

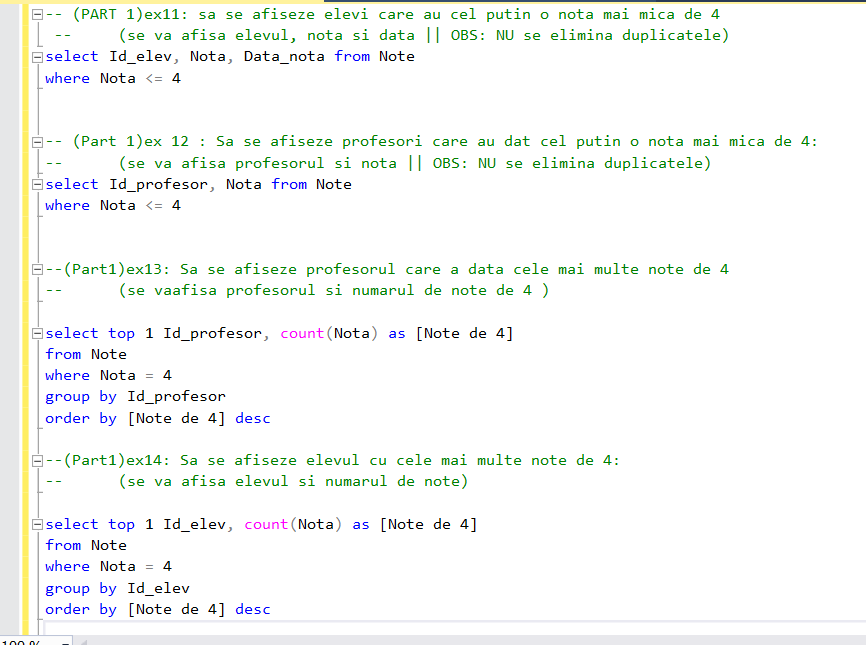
(se va afisa numele , autorul si pretul cartii)

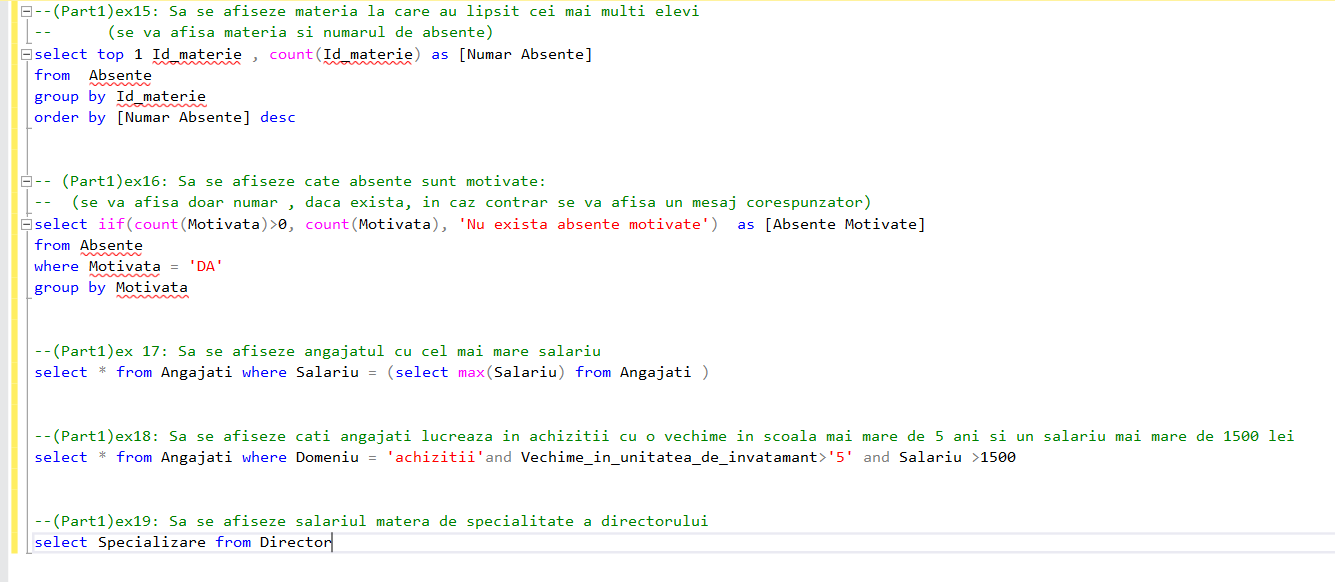
****

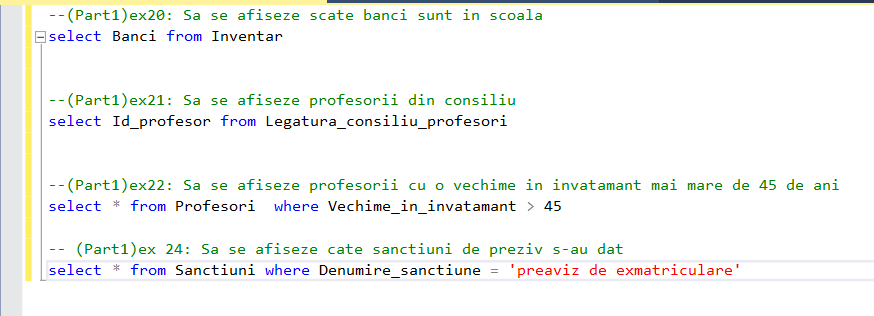
10) Sa se afiseze 10 cele mai imprumutate carti

( se va afisa cartea si numarul de exemplare imprumutate)

****

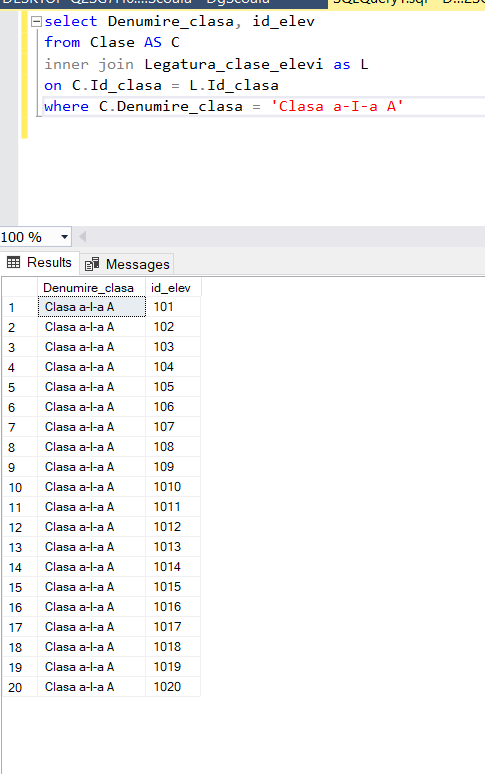
****

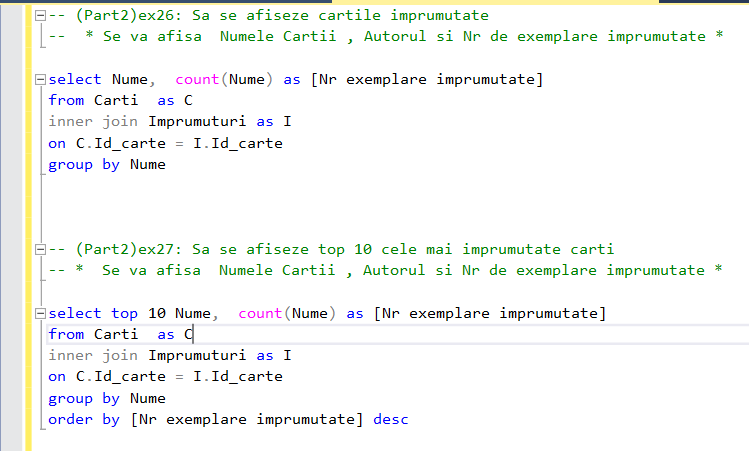
****

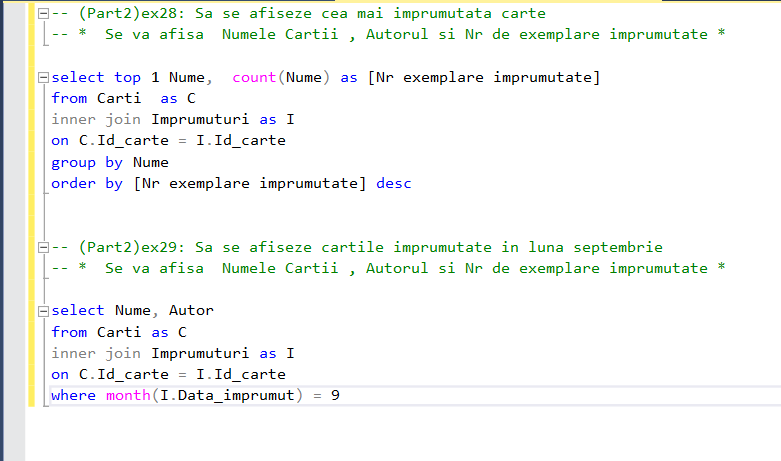
****

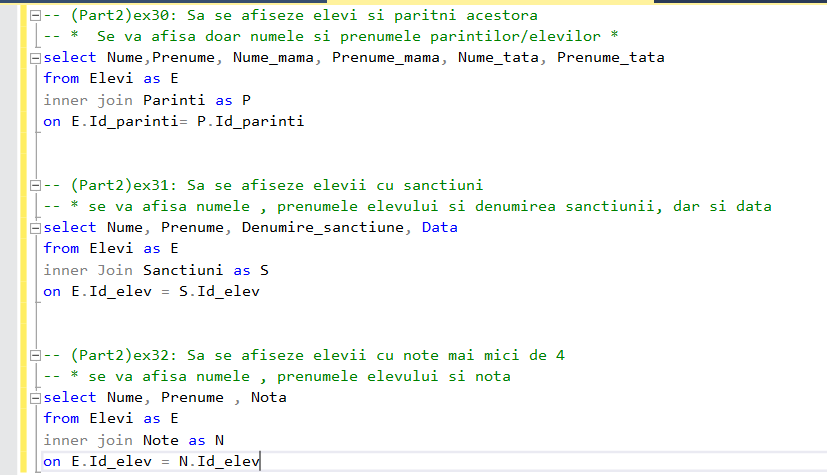
II Interogari cu Join

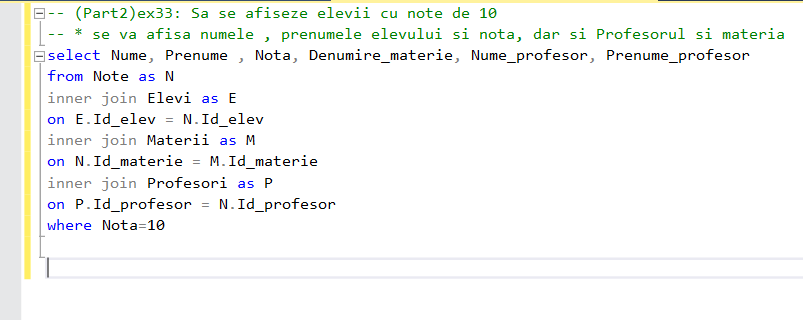
25) Afisati pe ecran toti elevi din clasa a-I-a A:

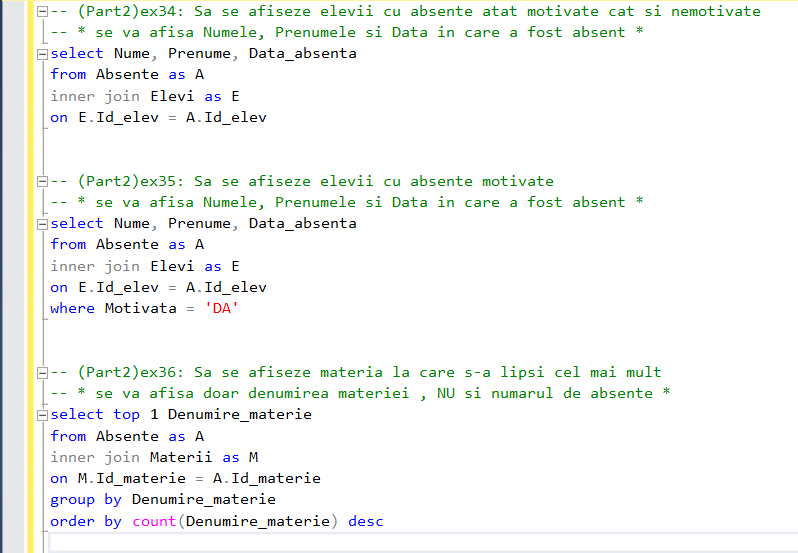


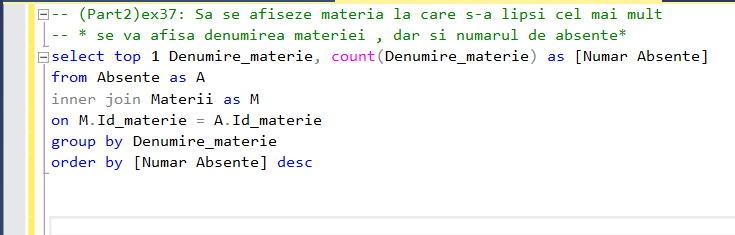
****

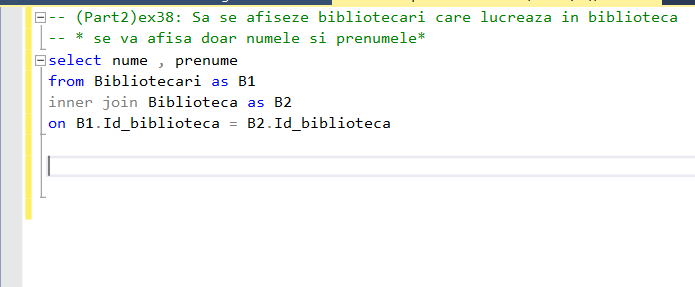
****

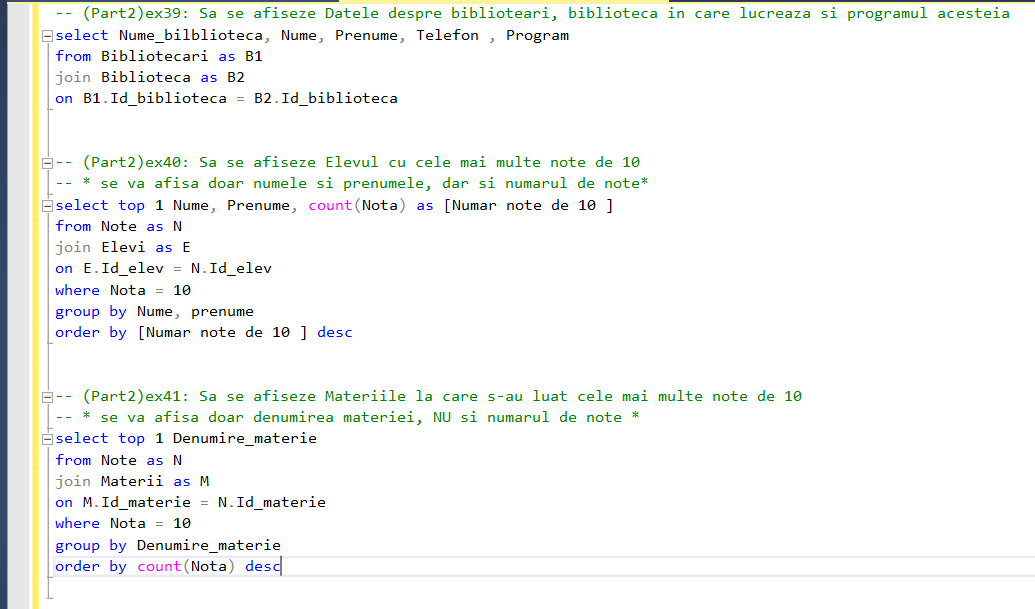
****

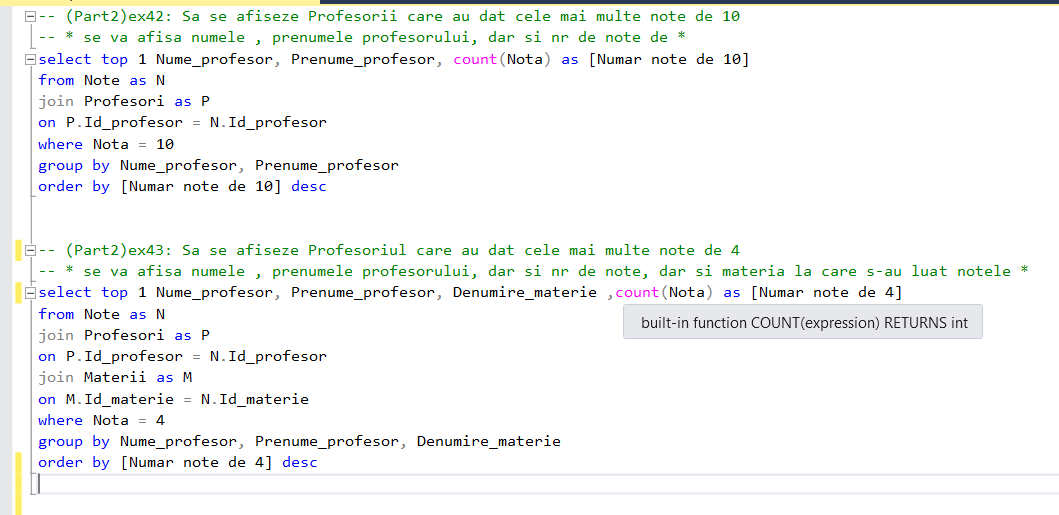
****

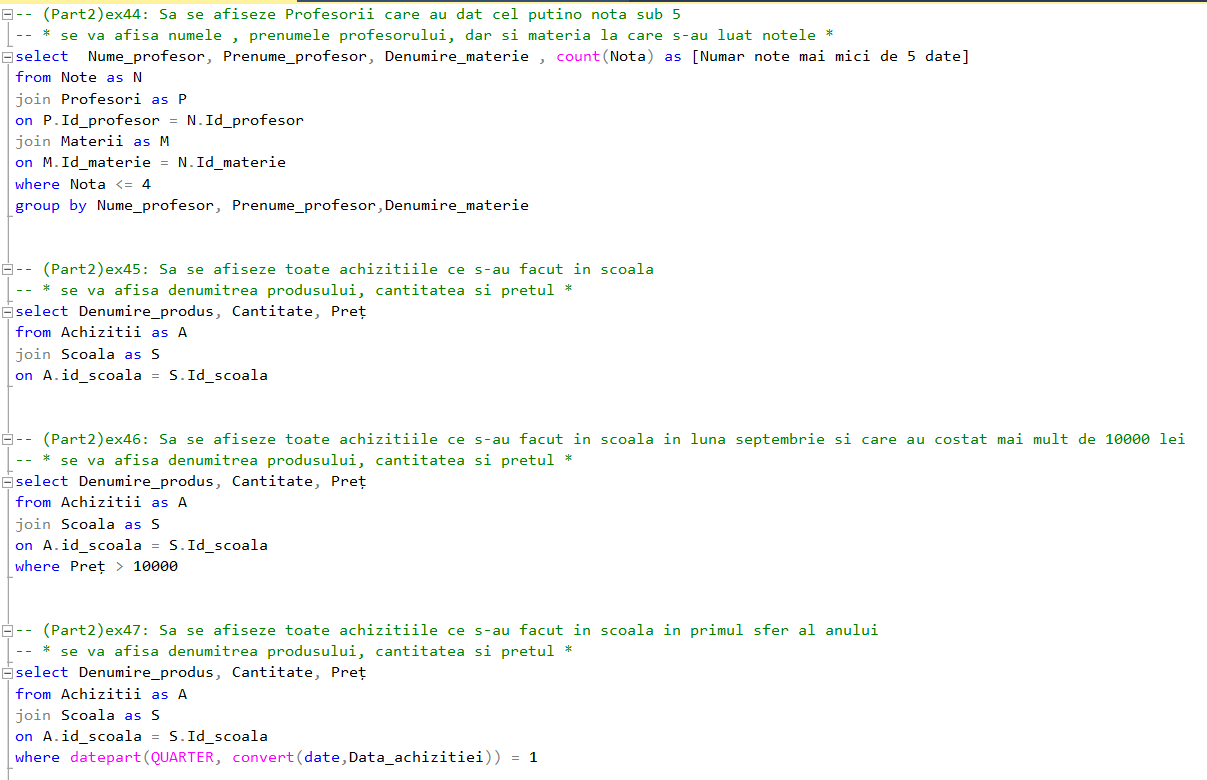
****

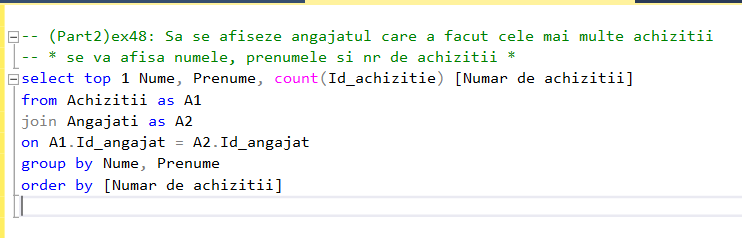
****

****

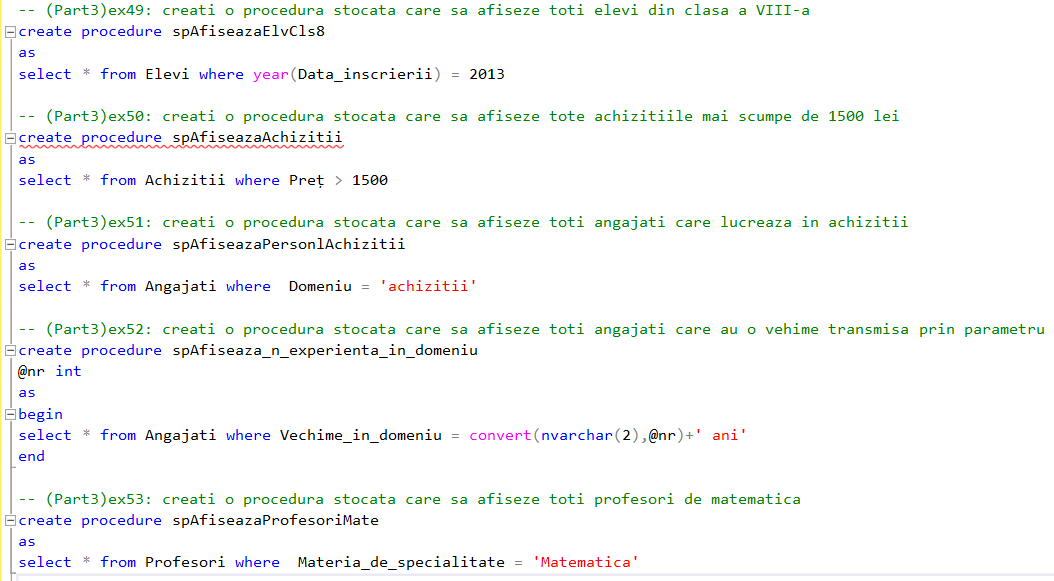
****

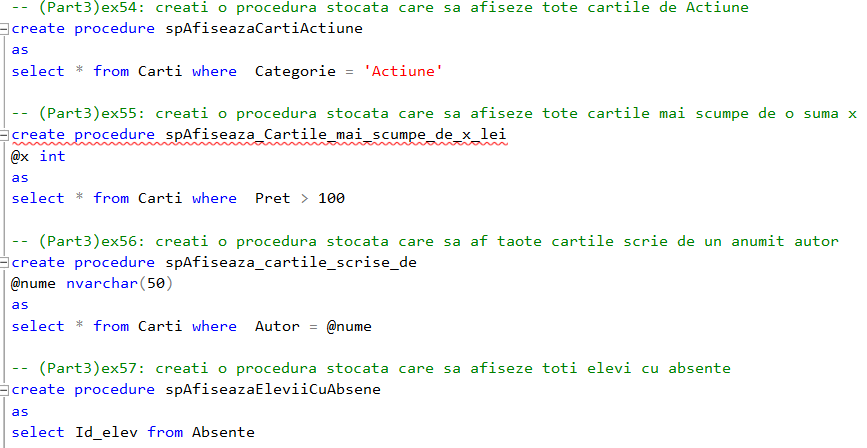
****

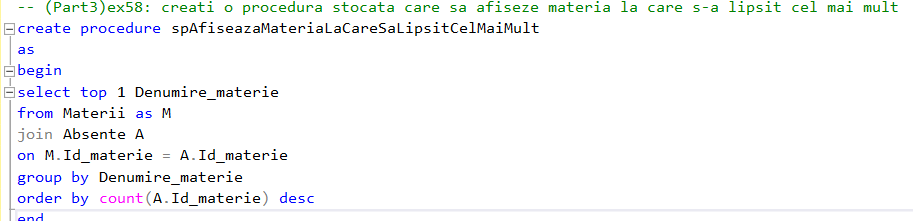
****

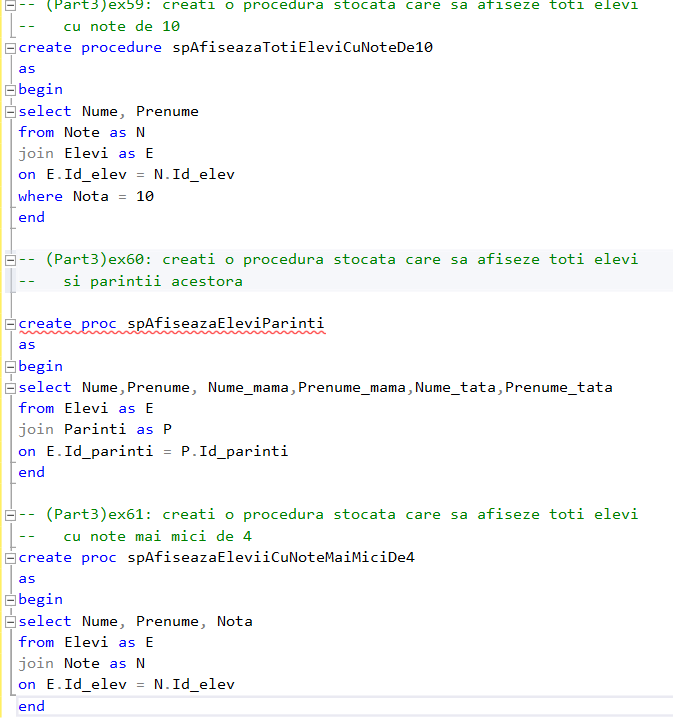
****

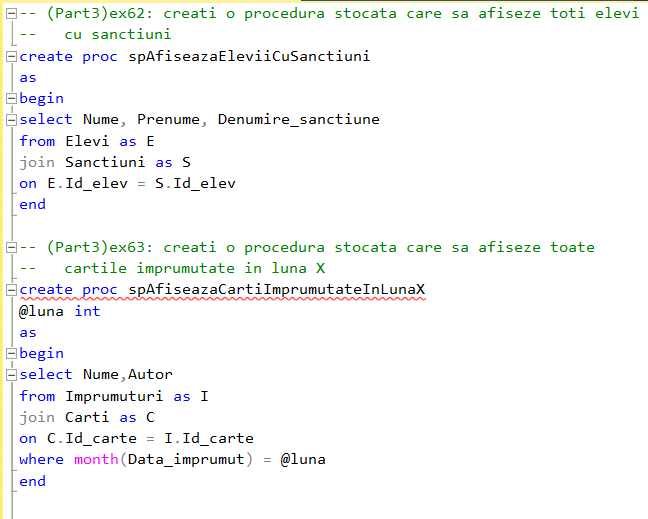
**III** Interogari cu Proceduri Stocate

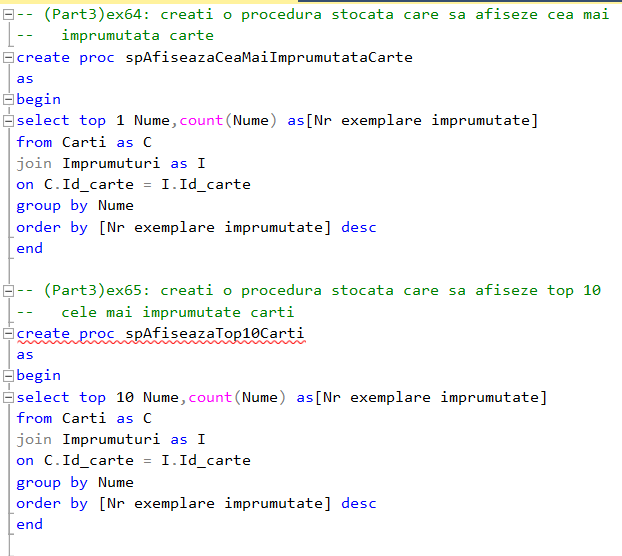


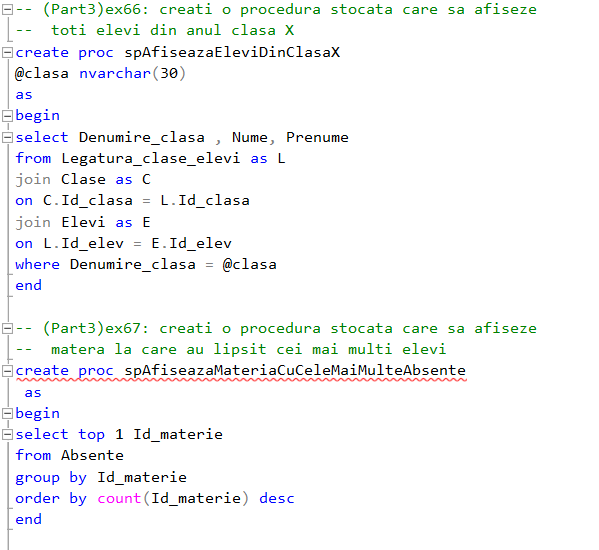
****

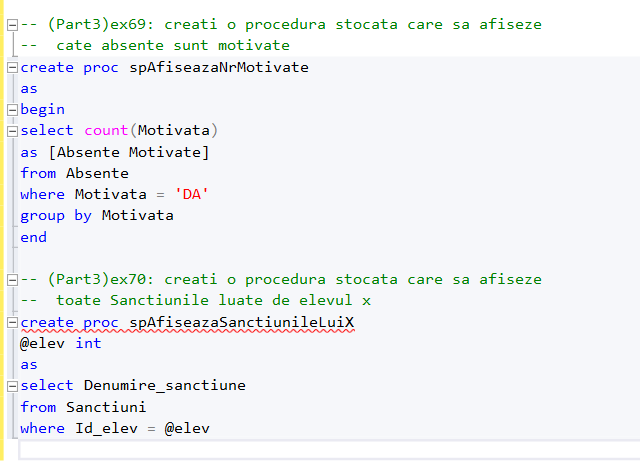
****

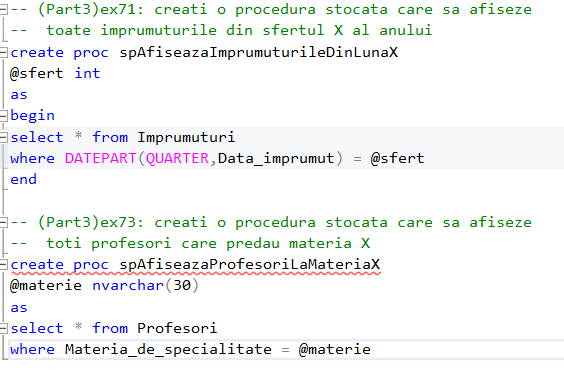
****

****

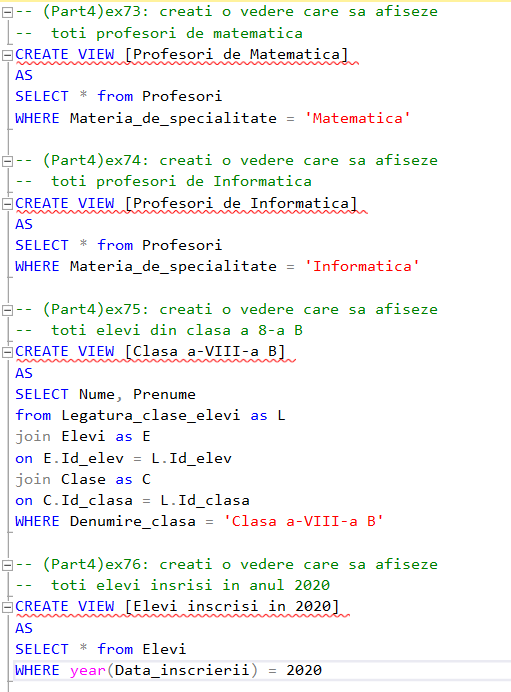
****

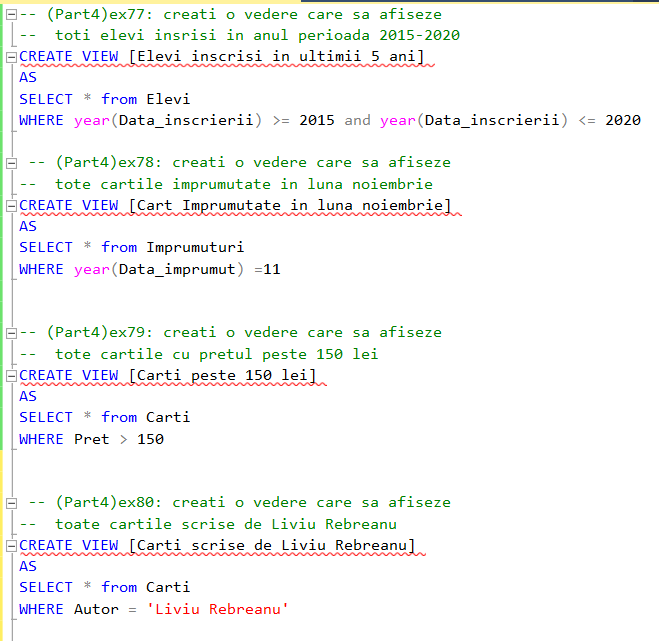
****

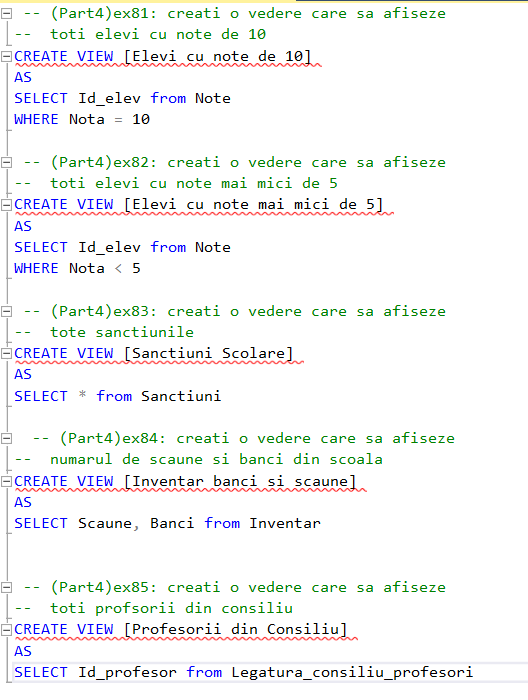
****

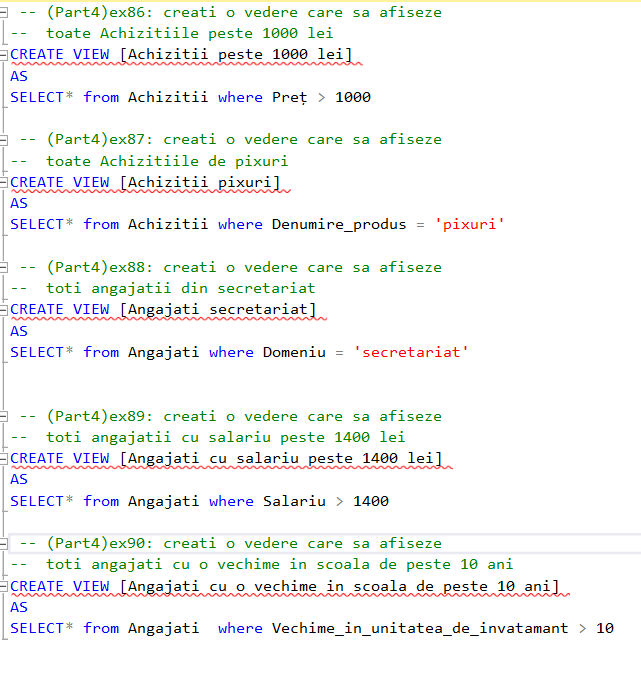
****

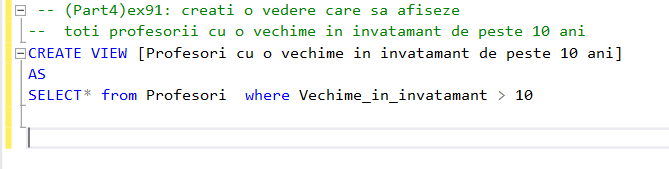
**IV** Interogari cu View-uri

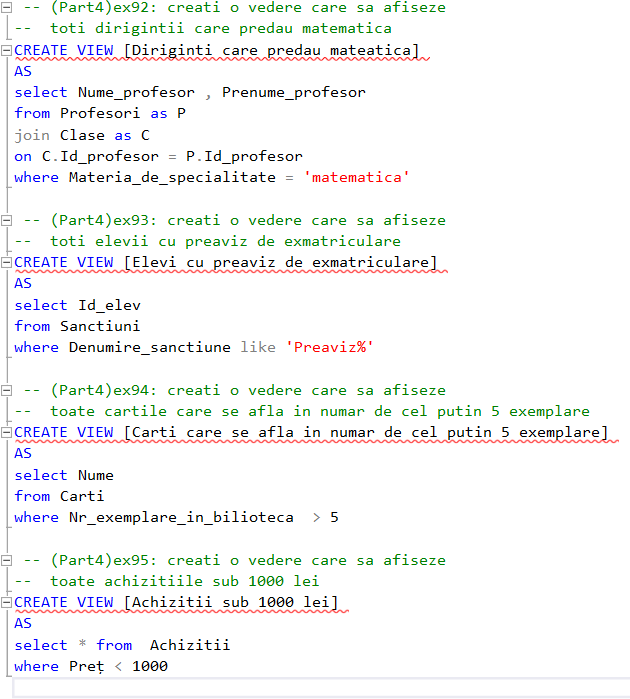


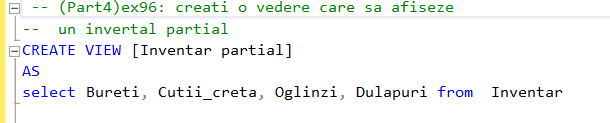
****

****

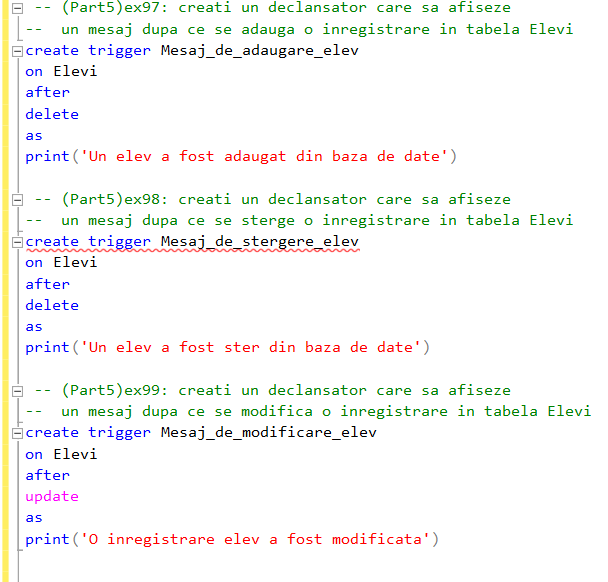
****

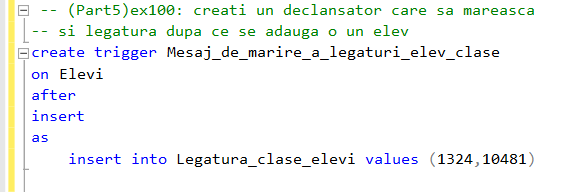
****

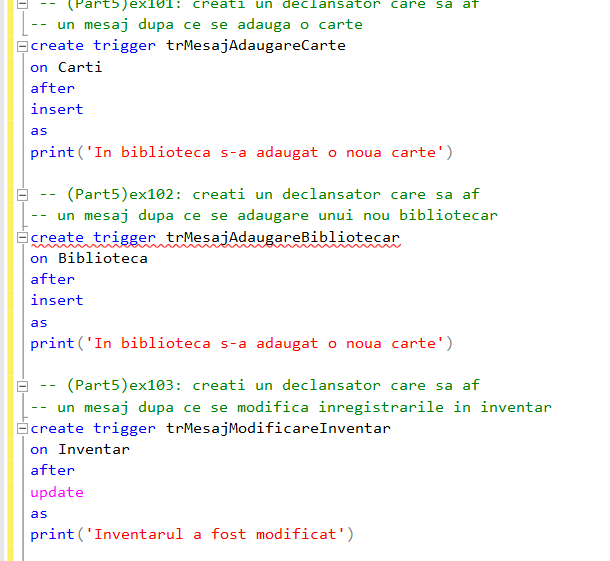
****

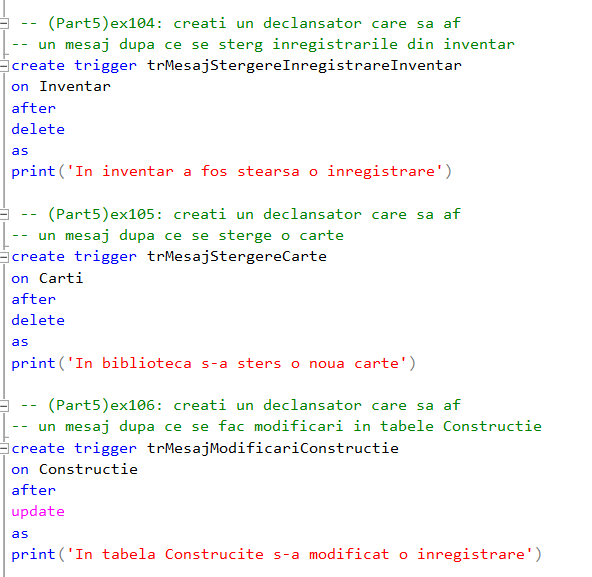
****

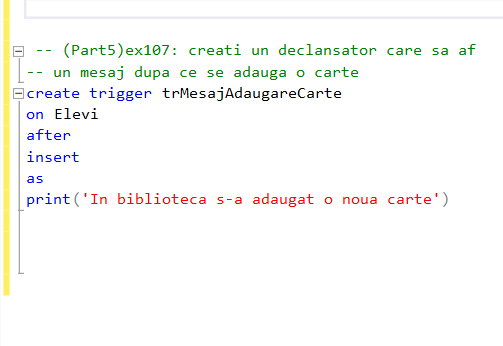
**V** Interogari cu Declansatoare

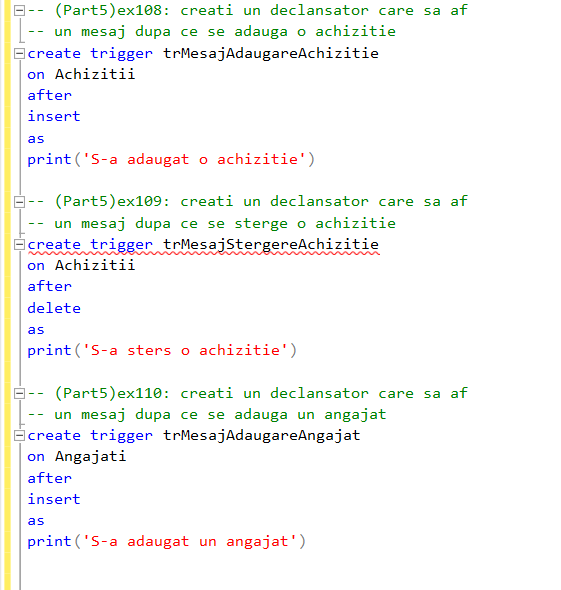


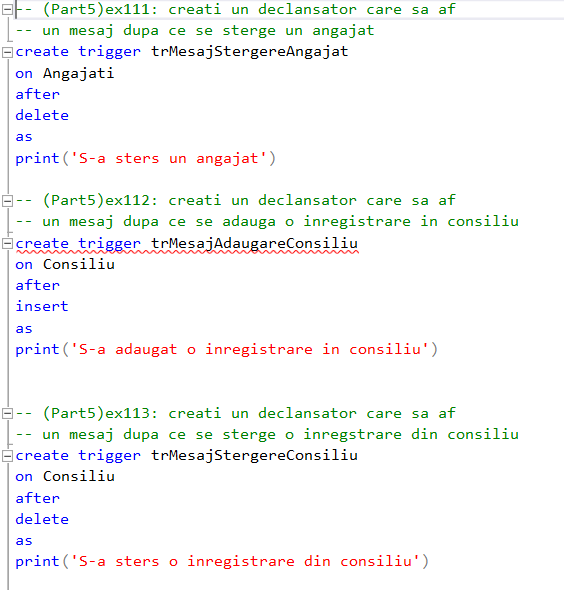


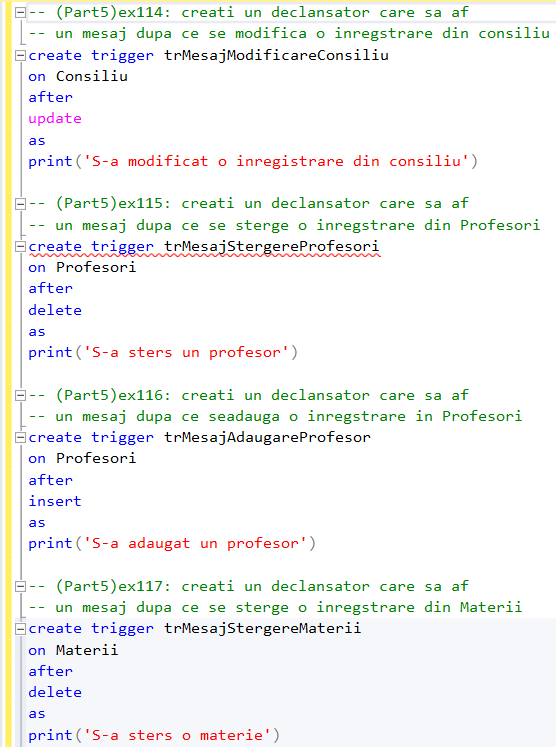


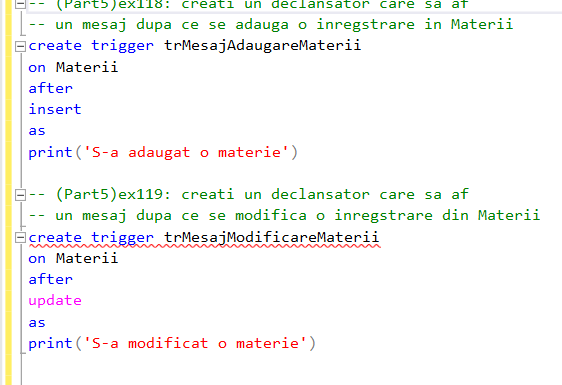
****

****

****

****

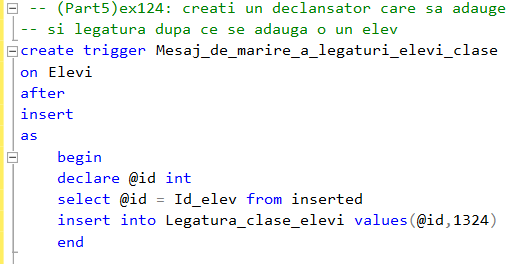
****

****

(Part5) ex 120:

In crearea trigger-ului se va avea in vedere faptul ca doar clasa ‘a-VIII-a C’ mai poate primi elevi .

* La introducerea datelor (prin comanda “insert”) tb sa se tina cont de criterial de varsta
* Clasa a-VIII-a C este initializata cu 20 de elevi

****

Ex124 :

Creati o SP care sa se afiseze vehimea angajatilor (ani + luni +zile):



**Rezultat:**

