Aplicatie pentru
Gestionarea unei firme de dezvoltare personala

Sd. Cap. Chiuta Mihai – C112C

ACADEMIA TEHNICA MILITARA “FERDINAND I”

1 Contents

2 Introducere

Acest proiect are ca scop gestionarea unei firme de dezvoltare profesionala (Kompetent).

In firma de fata ar trebui sa se aiba in vedere angajatii cu datele lor personale (Employees), care vor apartine cate unui department (Departments) si va avea o anumita functie (EmployeeRole).

De asemenea, activitatea principala a firmei este organizarea cursurilor (Courses).

Fiecare curs va putea fi planificat de mai multe ori (CourseSchedule), iar fiecare planificare va presupune un lector (Instructors) care sa predea cursul, o locatie in care sa se desfasoare

(Facilities), si materialele necesare (CourseMaterials) (e.g. : pixuri si mape).

Materialele (Materials) trebuie comandate (Orders) de la diferiti furnizori (Suppliers) si vor apartine unor categorii (MaterialCategories) (e.g: laptop-urile sunt echipamente din categoria

IT)

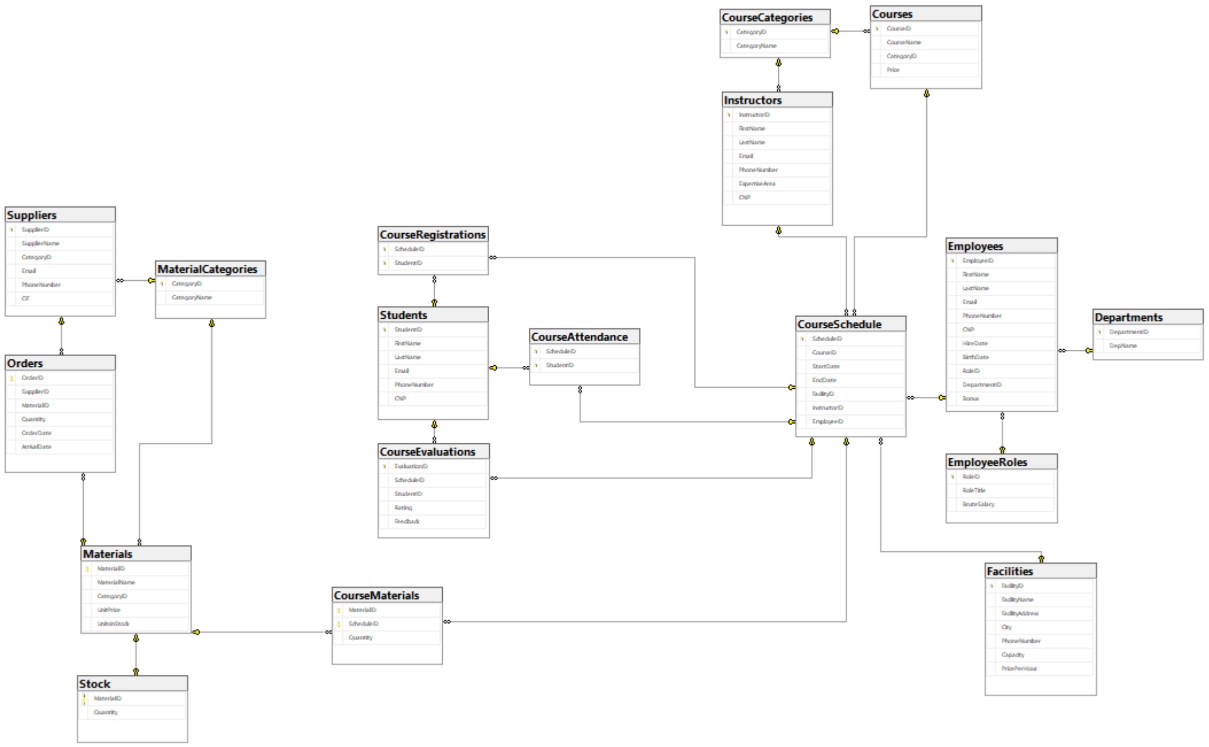
Cursantii (Students) vor fi si ei salvati in baza de date, precum si inscrierile

(CourseRegistrations), prezenta lor (CourseAttendances), dar si recenziile

(CourseEvaluations).

Tabela CourseSchedule si Employees vor fi legate in cerintele extra.

3 Diagrama bazei de date



Atentie!! Diagrama de mai sus este valabila de abia dupa rularea cerintelor extra (pana atunci tabela de Employees nu va fi legata de CourseSchedule)

4 Crearea bazei de date

```
GO
CREATE DATABASE Kompetent
ON PRIMARY
(
      Name = Data1,
      FileName = 'D:\School\Anul 2\Semestrul 2\Baze de
date\Tema\Kompetent\Data1.ndf',
      size = 10MB,
      maxsize = unlimited,
      filegrowth = 1GB
),
(
      Name = Data2,
      FileName = 'D:\School\Anul 2\Semestrul 2\Baze de
date\Tema\Kompetent\Data2.ndf',
      size = 10MB,
      maxsize = unlimited,
      filegrowth = 1GB
),
(
      Name = Data3,
      FileName = 'D:\School\Anul 2\Semestrul 2\Baze de
date\Tema\Kompetent\Data3.ndf',
      size = 10MB,
      maxsize = unlimited,
      filegrowth = 1GB
)
LOG ON
(
      Name = Log1,
      FileName = 'D:\School\Anul 2\Semestrul 2\Baze de
date\Tema\Kompetent\Log1.ldf',
      size = 10MB,
      maxsize = unlimited,
      filegrowth = 1GB
),
(
      Name = Log2,
      FileName = 'D:\School\Anul 2\Semestrul 2\Baze de
date\Tema\Kompetent\Log2.ldf',
      size = 10MB,
      maxsize = unlimited,
      filegrowth = 1GB
)

USE Kompetent;
```

5 Crearea tabelelor

Am ales sa encapsulez crearea fiecarui tabel intr-o singura tranzactie.

Astfel daca unul din tabele esueaza sa fie creat, nu vor fi create nici celalalte.

```
------------------------------------------------------
--------------------- TABLES -------------------------
------------------------------------------------------
GO
BEGIN TRAN CREATE_TABLES
```

5.1 Crearea tabelului Students

```
GO
IF OBJECT_ID('Students', 'U') IS NOT NULL
      DROP TABLE Students
GO
CREATE TABLE Students (
    StudentID INT PRIMARY KEY IDENTITY(1,1),
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Email VARCHAR(50) NOT NULL,
    PhoneNumber VARCHAR(20) NOT NULL,
      CNP VARCHAR(13) NOT NULL UNIQUE
);
```

5.2 Crearea tabelului Departments

```
GO
IF OBJECT_ID('Departments', 'U') IS NOT NULL
      DROP TABLE Departments
GO
CREATE TABLE Departments(
DepartmentID INT PRIMARY KEY IDENTITY(1,1),
DepName VARCHAR(50) NOT NULL
);
```

5.3 Crearea tabelului EmployeeRoles

```
GO
IF OBJECT_ID('EmployeeRoles', 'U') IS NOT NULL
      DROP TABLE EmployeeRoles
GO
CREATE TABLE EmployeeRoles (
RoleID INT PRIMARY KEY IDENTITY(1,1),
RoleTitle VARCHAR(50) NOT NULL,
BruteSalary DECIMAL(10,2) NOT NULL
);
```

5.4 Crearea tabelului Employees

```
GO
IF OBJECT_ID('Employees', 'U') IS NOT NULL
      DROP TABLE Employees
GO
CREATE TABLE Employees (
EmployeeID INT PRIMARY KEY IDENTITY(1,1),
FirstName VARCHAR(50) NOT NULL,
LastName VARCHAR(50) NOT NULL,
Email VARCHAR(50) NOT NULL,
PhoneNumber VARCHAR(20) NOT NULL,
CNP VARCHAR(13) NOT NULL,
HireDate DATE NOT NULL,
BirthDate DATE NOT NULL,
RoleID INT NOT NULL,
DepartmentID INT NOT NULL,
Bonus INT NOT NULL DEFAULT 0,
FOREIGN KEY (RoleID) REFERENCES EmployeeRoles(RoleID),
FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID),
UNIQUE (CNP)
);
```

5.5 Crearea tabelului CourseCategories

```
GO
IF OBJECT_ID('CourseCategories', 'U') IS NOT NULL
      DROP TABLE CourseCategories
GO
CREATE TABLE CourseCategories (
CategoryID INT PRIMARY KEY IDENTITY(1,1),
CategoryName VARCHAR(50) UNIQUE
)
```

5.6 Crearea tabelului Instructors

```
GO
IF OBJECT_ID('Instructors', 'U') IS NOT NULL
      DROP TABLE Instructors
GO
CREATE TABLE Instructors (
InstructorID INT PRIMARY KEY IDENTITY(1,1),
FirstName VARCHAR(50) NOT NULL,
LastName VARCHAR(50) NOT NULL,
Email VARCHAR(50) NOT NULL,
PhoneNumber VARCHAR(20) NOT NULL,
ExpertiseArea INT NOT NULL,
CNP VARCHAR(13) NOT NULL UNIQUE,
FOREIGN KEY (ExpertiseArea) REFERENCES CourseCategories(CategoryID)
);
```

5.7 Crearea tabelului Courses

```
GO
IF OBJECT_ID('Courses', 'U') IS NOT NULL
      DROP TABLE Courses
GO
CREATE TABLE Courses (
CourseID INT PRIMARY KEY IDENTITY(1,1),
CourseName VARCHAR(50) UNIQUE NOT NULL,
CategoryID INT NOT NULL,
Price DECIMAL(10,2) NOT NULL,
FOREIGN KEY (CategoryID) REFERENCES CourseCategories(CategoryID)
);
```

5.8 Crearea tabelului Facilities

```
GO
IF OBJECT_ID('Facilities', 'U') IS NOT NULL
      DROP TABLE Facilities
GO
-- n-am adaugat email fiindca majoritatea hotelurilor n-au
CREATE TABLE Facilities (
FacilityID INT PRIMARY KEY IDENTITY(1,1),
FacilityName VARCHAR(50) NOT NULL,
FacilityAddress VARCHAR(100) NOT NULL,
City VARCHAR(20) NOT NULL,
PhoneNumber VARCHAR(20),
Capacity INT NOT NULL,
PricePerHour DECIMAL(8,2)
);
```

5.9 Crearea tabelului CourseSchedule

```
GO
IF OBJECT_ID('CourseSchedule', 'U') IS NOT NULL
      DROP TABLE CourseSchedule
GO
CREATE TABLE CourseSchedule (
ScheduleID INT PRIMARY KEY IDENTITY(1,1),
CourseID INT NOT NULL,
StartDate DATE NOT NULL,
EndDate DATE NOT NULL,
FacilityID INT NOT NULL,
InstructorID INT NOT NULL,
FOREIGN KEY (CourseID) REFERENCES Courses(CourseID),
FOREIGN KEY (FacilityID) REFERENCES Facilities(FacilityID),
FOREIGN KEY (InstructorID) REFERENCES Instructors(InstructorID)
);
```

5.10 Crearea tabelului CourseRegistrations

```
GO
IF OBJECT_ID('CourseRegistrations', 'U') IS NOT NULL
      DROP TABLE CourseRegistrations
GO
CREATE TABLE CourseRegistrations (
ScheduleID INT NOT NULL,
StudentID INT NOT NULL,
PRIMARY KEY (ScheduleID, StudentID),
FOREIGN KEY (ScheduleID) REFERENCES CourseSchedule(ScheduleID),
FOREIGN KEY (StudentID) REFERENCES Students(StudentID)
);
```

5.11 Crearea tabelului CourseAttendance

```
GO
IF OBJECT_ID('CourseAttendance', 'U') IS NOT NULL
```

```
      DROP TABLE CourseAttendance
GO
CREATE TABLE CourseAttendance (
ScheduleID INT NOT NULL,
StudentID INT NOT NULL,
PRIMARY KEY (ScheduleID, StudentID),
FOREIGN KEY (ScheduleID) REFERENCES CourseSchedule(ScheduleID),
FOREIGN KEY (StudentID) REFERENCES Students(StudentID)
);
```

5.12 Crearea tabelului MaterialCategories

```
GO
IF OBJECT_ID('MaterialCategories', 'U') IS NOT NULL
      DROP TABLE MaterialCategories
GO
CREATE TABLE MaterialCategories (
CategoryID INT PRIMARY KEY IDENTITY(1,1),
CategoryName VARCHAR(50) UNIQUE
)
```

5.13 Crearea tabelului Materials

```
GO
IF OBJECT_ID('Materials', 'U') IS NOT NULL
      DROP TABLE Materials
GO
CREATE TABLE Materials (
MaterialID INT PRIMARY KEY IDENTITY(1,1),
MaterialName VARCHAR(50) NOT NULL,
CategoryID INT NOT NULL,
UnitPrice DECIMAL(8,2) NOT NULL,
UnitsInStock INT NOT NULL,
FOREIGN KEY (CategoryID) REFERENCES MaterialCategories(CategoryID)
);
```

5.14 Crearea tabelului CourseMaterials

```
GO
IF OBJECT_ID('CourseMaterials', 'U') IS NOT NULL
      DROP TABLE CourseMaterials
GO
CREATE TABLE CourseMaterials (
MaterialID INT NOT NULL,
ScheduleID INT NOT NULL,
Quantity INT NOT NULL,
PRIMARY KEY (ScheduleID, MaterialID),
FOREIGN KEY (ScheduleID) REFERENCES CourseSchedule(ScheduleID),
FOREIGN KEY (MaterialID) REFERENCES Materials(MaterialID)
);
```

5.15 Crearea tabelului CourseEvaluations

```
GO
IF OBJECT_ID('CourseEvaluations', 'U') IS NOT NULL
```

```
      DROP TABLE CourseEvaluations
GO
CREATE TABLE CourseEvaluations (
EvaluationID INT PRIMARY KEY IDENTITY(1,1),
ScheduleID INT NOT NULL,
StudentID INT NOT NULL,
Rating INT NOT NULL CHECK (Rating <= 100 AND Rating >= 0),
Feedback TEXT NOT NULL,
FOREIGN KEY (ScheduleID) REFERENCES CourseSchedule(ScheduleID),
FOREIGN KEY (StudentID) REFERENCES Students(StudentID)
);
```

5.16 Crearea tabelului Suppliers

```
GO
IF OBJECT_ID('Suppliers', 'U') IS NOT NULL
      DROP TABLE Suppliers
GO
CREATE TABLE Suppliers (
SupplierID INT PRIMARY KEY IDENTITY(1,1),
SupplierName VARCHAR(50) NOT NULL,
CategoryID INT NOT NULL,
Email VARCHAR(50) NOT NULL,
PhoneNumber VARCHAR(20) NOT NULL,
CIF VARCHAR(13) NOT NULL,
FOREIGN KEY (CategoryID) REFERENCES MaterialCategories(CategoryID)
);
```

5.17 Crearea tabelului Orders

```
GO
IF OBJECT_ID('Orders', 'U') IS NOT NULL
      DROP TABLE Orders
GO
CREATE TABLE Orders (
OrderID INT PRIMARY KEY IDENTITY(1,1),
SupplierID INT NOT NULL,
MaterialID INT NOT NULL,
Quantity INT NOT NULL,
OrderDate DATE NOT NULL,
ArrivalDate DATE NULL,
FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID),
FOREIGN KEY (MaterialID) REFERENCES Materials(MaterialID)
);
```

5.18 Crearea tabelului Stock

```
GO
IF OBJECT_ID('Stock', 'U') IS NOT NULL
      DROP TABLE Stock
GO
CREATE TABLE Stock(
MaterialID INT PRIMARY KEY FOREIGN KEY REFERENCES Materials(MaterialID),
Quantity INT NOT NULL,
);
```

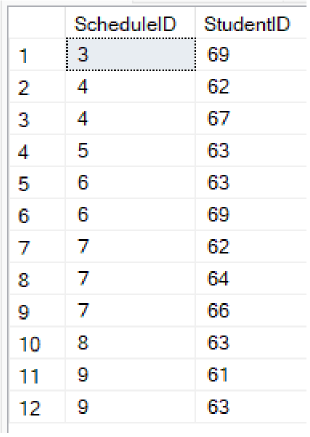
```
COMMIT CREATE_TABLES;
GO
```

6 Vederi

```
-------------------------------------------------------
---------------------- VIEWS -------------------------
-------------------------------------------------------
```

6.1 Sa se creeze un view care sa afiseze numele furnizorului si suma totala a comenzilor pentru fiecare furnizor

```
--------------------- VIEW 1 -------------------------

IF OBJECT_ID('OrdersBySupplier', 'V') IS NOT NULL
      DROP VIEW OrdersBySupplier

GO
CREATE VIEW OrdersBySupplier AS
SELECT S.SupplierName, SUM(O.Quantity * M.UnitPrice) as 'Total'
FROM Orders O
INNER JOIN Suppliers S
ON O.SupplierID = S.SupplierID
INNER JOIN Materials M
ON O.MaterialID = M.MaterialID
GROUP BY S.SupplierName
GO
```

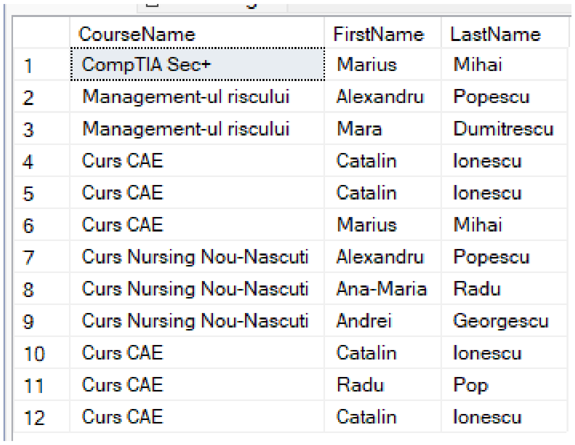


	ScheduleID	StudentID
1	3	69
2	4	62
3	4	67
4	5	63
5	6	63
6	6	69
7	7	62
8	7	64
9	7	66
10	8	63
11	9	61
12	9	63

6.2 Sa se creeze un view care sa arate toate perechile (nume lector, nume curs) potrivite din punct de vedere a expertizei lectorului

```
--------------------- VIEW 2 -------------------------


IF OBJECT_ID('CoursesInstructorsPairByExpertise', 'V') IS NOT NULL
```

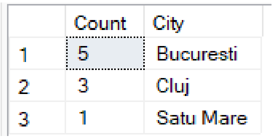
```
      DROP VIEW CoursesInstructorsPairByExpertise

GO
CREATE VIEW CoursesInstructorsPairByExpertise AS
SELECT C.CourseName, I.FirstName, I.LastName
FROM Courses C
INNER JOIN Instructors I
ON C.CategoryID = I.ExpertiseArea
GO
```



	CourseName	FirstName	LastName
1	CompTIA Sec+	Marius	Mihai
2	Management-ul riscului	Alexandru	Popescu
3	Management-ul riscului	Mara	Dumitrescu
4	Curs CAE	Catalin	Ionescu
5	Curs CAE	Catalin	Ionescu
6	Curs CAE	Marius	Mihai
7	Curs Nursing Nou-Nascuti	Alexandru	Popescu
8	Curs Nursing Nou-Nascuti	Ana-Maria	Radu
9	Curs Nursing Nou-Nascuti	Andrei	Georgescu
10	Curs CAE	Catalin	Ionescu
11	Curs CAE	Radu	Pop
12	Curs CAE	Catalin	Ionescu

6.3 Sa se creeze un view care sa afiseze toate absentele la cursuri

```
--------------------- VIEW 3 -------------------------

IF OBJECT_ID('Absences','V') IS NOT NULL
      DROP VIEW Absences

GO
CREATE VIEW Absences AS
SELECT CR.ScheduleID, CR.StudentID
FROM CourseRegistrations CR
EXCEPT
SELECT CA.ScheduleID, CA.StudentID
FROM CourseAttendance CA
GO
```

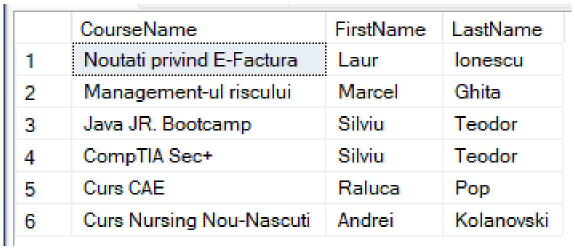


	Count	City
1	5	Bucuresti
2	3	Cluj
3	1	Satu Mare

6.4 Sa se creeze un view care sa afiseze toate absentele la cursuri, afisand numele cursurilor si al studentilor

```
--------------------- VIEW 4 -------------------------

IF OBJECT_ID('AbsencesWithNames', 'V') IS NOT NULL
      DROP VIEW AbsencesWithNames
```

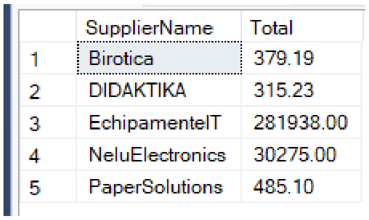
```
GO
CREATE VIEW AbsencesWithNames AS
SELECT C.CourseName, S.FirstName, S.LastName
FROM Absences A
INNER JOIN CourseSchedule CS ON CS.ScheduleID = A.ScheduleID
INNER JOIN Courses C          ON C.CourseID = CS.CourseID
INNER JOIN Students S         ON S.StudentID = A.StudentID
GO
```



	CourseName	FirstName	LastName
1	Noutati privind E-Factura	Laur	Ionescu
2	Management-ul riscului	Marcel	Ghita
3	Java JR. Bootcamp	Silviu	Teodor
4	CompTIA Sec+	Silviu	Teodor
5	Curs CAE	Raluca	Pop
6	Curs Nursing Nou-Nascuti	Andrei	Kolanovski

6.5 Sa se creeze un view care sa afiseze numarul de cursuri planificate pe fiecare oras

```
--------------------- VIEW 5 -------------------------

IF OBJECT_ID('CoursesCountByCity', 'V') IS NOT NULL
      DROP VIEW CoursesCountByCity

GO
CREATE VIEW CoursesCountByCity AS
SELECT COUNT(*) as 'Count', F.City as 'City'
FROM CourseSchedule CS INNER JOIN Facilities F
ON CS.FacilityID = F.FacilityID
GROUP BY F.City
GO
```



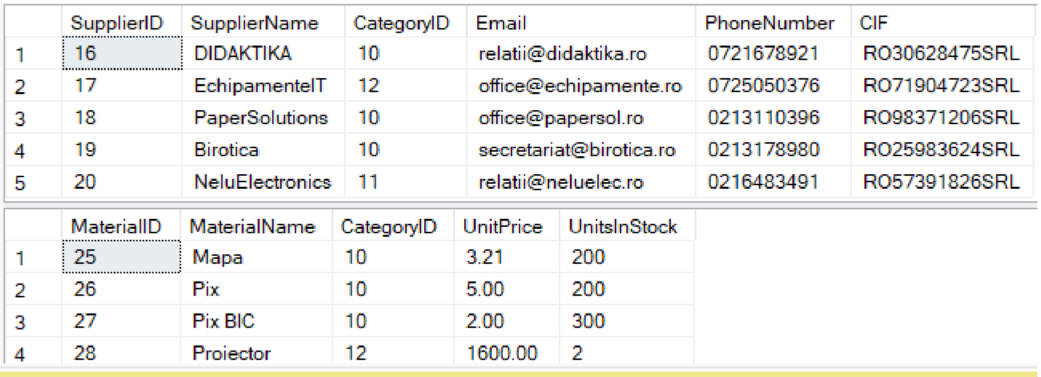
	SupplierName	Total
1	Birotica	379.19
2	DIDAKTIKA	315.23
3	EchipamenteIT	281938.00
4	NeluElectronics	30275.00
5	PaperSolutions	485.10

7 Declansatoare

```
------------------------------------------------------
--------------------- Triggers -----------------------
------------------------------------------------------
```

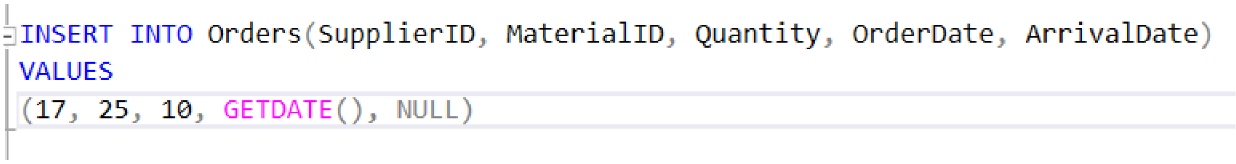
7.1 Sa se creeze un trigger pentru cazul in care se comanda un produs de la un furnizor care nu are produse din acea categorie

```
-------------------- Trigger 1 -----------------------

-- trigger for when ordering a product from a supplier which doesn't have
a matching categoryID
GO
IF OBJECT_ID('tr_OrderSupplierCtgMatch', 'TR') IS NOT NULL
      DROP TRIGGER tr_OrderSupplierCtgMatch
GO

CREATE TRIGGER tr_OrderSupplierCtgMatch
ON Orders
AFTER INSERT, UPDATE
AS
BEGIN

      IF EXISTS(
            SELECT *
            FROM inserted I
            INNER JOIN Suppliers S
            ON I.SupplierID = S.SupplierID
            INNER JOIN Materials M
            ON I.MaterialID = M.MaterialID
            WHERE M.CategoryID != S.CategoryID
      )
      BEGIN
            ;THROW 50000, 'One or more of the inserted suppliers don''t
have materials of that category.', 0
      END

      RETURN;
END
```

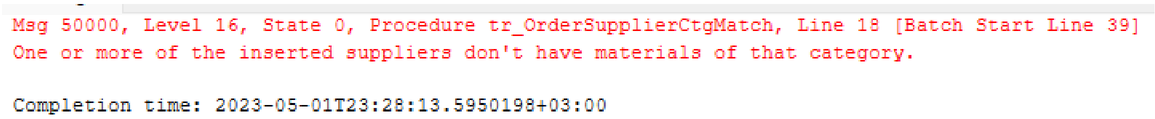


	SupplierID	SupplierName	CategoryID	Email	PhoneNumber	CIF
1	16	DIDAKTIKA	10	relatii@didaktika.ro	0721678921	RO30628475SRL
2	17	EchipamenteIT	12	office@echipamente.ro	0725050376	RO71904723SRL
3	18	PaperSolutions	10	office@papersol.ro	0213110396	RO98371206SRL
4	19	Birotica	10	secretariat@birotica.ro	0213178980	RO25983624SRL
5	20	NeluElectronics	11	relatii@neluelec.ro	0216483491	RO57391826SRL

	MaterialID	MaterialName	CategoryID	UnitPrice	UnitsInStock
1	25	Mapa	10	3.21	200
2	26	Pix	10	5.00	200
3	27	Pix BIC	10	2.00	300
4	28	Proiector	12	1600.00	2



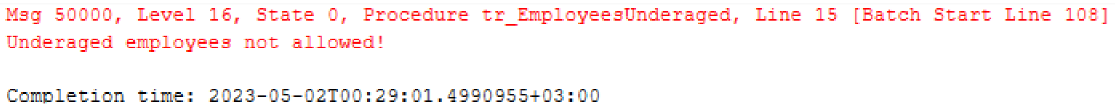
```
INSERT INTO Orders(SupplierID, MaterialID, Quantity, OrderDate, ArrivalDate)
VALUES
(17, 25, 10, GETDATE(), NULL)
```



```
Msg 50000, Level 16, State 0, Procedure tr_OrderSupplierCtgMatch, Line 18 [Batch Start Line 39]
One or more of the inserted suppliers don't have materials of that category.

Completion time: 2023-05-01T23:28:13.5950198+03:00
```

7.2 Sa se creeze un trigger care sa impiedice angajarea minorilor

```
-------------------- Trigger 2 -----------------------

GO
IF OBJECT_ID('tr_EmployeesUnderaged','TR') IS NOT NULL
      DROP TRIGGER tr_EmployeesUnderaged

GO
CREATE TRIGGER tr_EmployeesUnderaged
ON Employees
AFTER INSERT, UPDATE
AS
BEGIN
      IF @@ROWCOUNT = 0 RETURN;
      SET NOCOUNT ON;

      IF EXISTS (
            SELECT *
            FROM inserted I
            WHERE DATEDIFF(YEAR, I.BirthDate, GETDATE()) < 18
      )
      BEGIN
            ;THROW 50000, 'Underaged employees not allowed!', 0
      END

      RETURN;
END
```

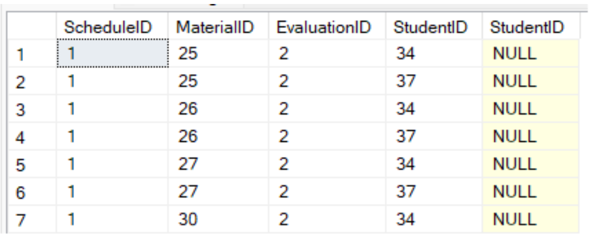


```
Msg 50000, Level 16, State 0, Procedure tr_EmployeesUnderaged, Line 15 [Batch Start Line 108]
Underaged employees not allowed!

Completion time: 2023-05-02T00:29:01.4990955+03:00
```

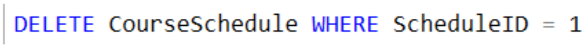
7.3 Sa se creeze un trigger care in cazul anularii unei planificari, sa se stearga si toate obiectele associate cu acea planificare inainte

```
-------------------- Trigger 3 -----------------------

-- delete CourseSchedule dependencies

-- here we can delete the schedules together with their dependencies
-- only if the courses haven't happened yet
-- ( e.g. : cancelling an upcoming course )
GO
IF OBJECT_ID('tr_DeleteCourseScheduleDependencies','TR') IS NOT NULL
      DROP TRIGGER tr_DeleteCourseScheduleDependencies

GO

CREATE TRIGGER tr_DeleteCourseScheduleDependencies
ON CourseSchedule
INSTEAD OF DELETE
AS
BEGIN
      IF @@ROWCOUNT = 0 RETURN;
      SET NOCOUNT ON;

      IF EXISTS(
            SELECT * FROM Deleted D
            WHERE DATEDIFF(DAY, CAST( GETDATE() AS DATE ), D.StartDate) <
0
      )
      BEGIN
            ;THROW 50000, 'You can''t delete past schedules!', 0
      END

      DELETE CA
      FROM CourseAttendance CA INNER JOIN Deleted D
      ON CA.ScheduleID = D.ScheduleID

      DELETE CR
      FROM CourseRegistrations CR INNER JOIN Deleted D
      ON CR.ScheduleID = D.ScheduleID

      DELETE CE
      FROM CourseEvaluations CE INNER JOIN Deleted D
      ON CE.ScheduleID = D.ScheduleID

      DELETE CM
      FROM CourseMaterials CM INNER JOIN Deleted D
      ON CM.ScheduleID = D.ScheduleID
```

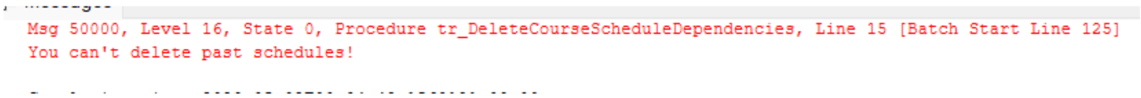
```
	DELETE CS
	FROM CourseSchedule CS INNER JOIN Deleted D
	ON CS.ScheduleID = D.ScheduleID

	RETURN;
END
```



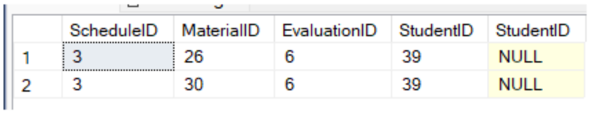
	ScheduleID	MaterialID	EvaluationID	StudentID	StudentID
1	1	25	2	34	NULL
2	1	25	2	37	NULL
3	1	26	2	34	NULL
4	1	26	2	37	NULL
5	1	27	2	34	NULL
6	1	27	2	37	NULL
7	1	30	2	34	NULL



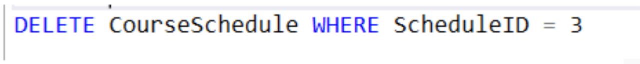
```
DELETE CourseSchedule WHERE ScheduleID = 1
```



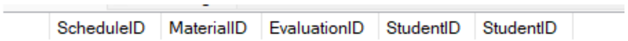
```
Msg 50000, Level 16, State 0, Procedure tr_DeleteCourseScheduleDependencies, Line 15 [Batch Start Line 125]
You can't delete past schedules!
```



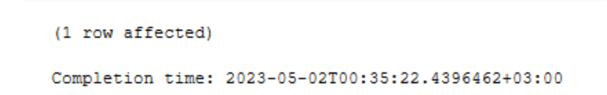
	ScheduleID	MaterialID	EvaluationID	StudentID	StudentID
1	3	26	6	39	NULL
2	3	30	6	39	NULL



```
DELETE CourseSchedule WHERE ScheduleID = 3
```



ScheduleID	MaterialID	EvaluationID	StudentID	StudentID

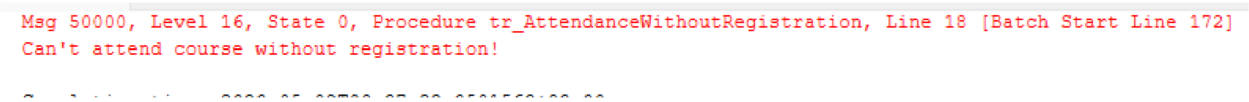


```
(1 row affected)

Completion time: 2023-05-02T00:35:22.4396462+03:00
```

7.4 Sa se creeze un trigger care sa impiedice prezenta unui student la un curs la care nu este inscris

```
-------------------- Trigger 4 -----------------------

-- prevent inserting course attendance where there was no registration
GO
IF OBJECT_ID('tr_AttendanceWithoutRegistration','TR') IS NOT NULL
	DROP TRIGGER tr_AttendanceWithoutRegistration

GO
CREATE TRIGGER tr_AttendanceWithoutRegistration
ON CourseAttendance
AFTER INSERT, UPDATE
AS
BEGIN
	IF @@ROWCOUNT = 0 RETURN;
	SET NOCOUNT ON;

	IF EXISTS
```

```
      (
            SELECT I.ScheduleID, I.StudentID
            FROM Inserted I
            EXCEPT
            SELECT CR.ScheduleID, CR.StudentID
            FROM CourseRegistrations CR
      )
      BEGIN
            ;THROW 50000, 'Can''t attend course without registration!', 0
      END

      RETURN;
END
```



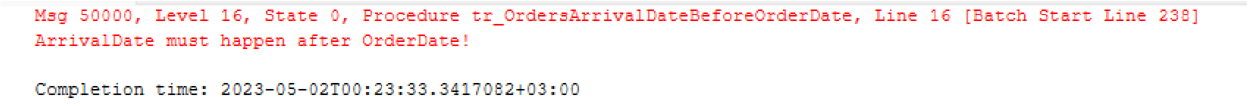
```
Msg 50000, Level 16, State 0, Procedure tr_AttendanceWithoutRegistration, Line 18 [Batch Start Line 172]
Can't attend course without registration!
```

7.5 Sa se scrie un trigger care sa impiedice prezenta a mai multor personae decat capacitatea salii

```
-------------------- Trigger 5 -----------------------

-- prevent inserting course attendance more than capacity
GO
IF OBJECT_ID('tr_CourseAttendanceCapacity','TR') IS NOT NULL
      DROP TRIGGER tr_CourseAttendanceCapacity

GO
CREATE TRIGGER tr_CourseAttendanceCapacity
ON CourseAttendance
AFTER INSERT,UPDATE
AS
BEGIN
      IF @@ROWCOUNT = 0 RETURN;
      SET NOCOUNT ON;


      DECLARE @temp AS TABLE ( ScheduleID INT, NrOfPeople INT );

      WITH CT(ScheduleID, StudentID) AS (
            SELECT CA.ScheduleID as ScheduleID, CA.StudentID as StudentID
            FROM CourseAttendance CA
            UNION
            SELECT I.ScheduleID, I.StudentID
            FROM Inserted I
      )
      INSERT INTO @temp
      SELECT CS.ScheduleID, COUNT(*)
      FROM CT
      INNER JOIN CourseSchedule CS
      ON CT.ScheduleID = CS.ScheduleID
      INNER JOIN Facilities F
      ON F.FacilityID = CS.FacilityID
      GROUP BY CS.ScheduleID, F.Capacity
      HAVING COUNT(*) > F.Capacity

      IF EXISTS( SELECT * FROM @temp )
      BEGIN
            SELECT * FROM @temp
            ;THROW 50000, 'Capacity limit reached for above facilities', 0
      END

END
```

7.6 In tabelul Orders, sa se creeze un trigger care sa impiedice o data de sosire a comenzii mai devreme decat data in care ea a fost plasata

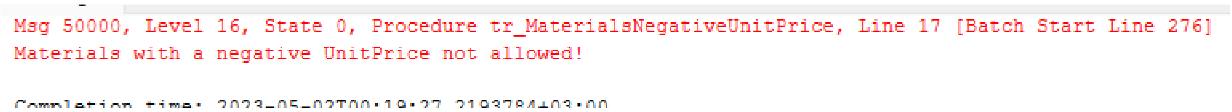
```
-------------------- Trigger 6 -----------------------

GO
IF OBJECT_ID('tr_OrdersArrivalDateBeforeOrderDate','TR') IS NOT NULL
       DROP TRIGGER tr_OrdersArrivalDateBeforeOrderDate

GO
CREATE TRIGGER tr_OrdersArrivalDateBeforeOrderDate
ON Orders
AFTER INSERT, UPDATE
AS
BEGIN
      IF @@ROWCOUNT = 0 RETURN;
      SET NOCOUNT ON;

      IF
      EXISTS (
            SELECT *
            FROM inserted I
            WHERE DATEDIFF(DAY, I.OrderDate, I.ArrivalDate) < 0
      )
      BEGIN
            ;THROW 50000, 'ArrivalDate must happen after OrderDate!', 0
      END

      RETURN;
END
```



```
Msg 50000, Level 16, State 0, Procedure tr_OrdersArrivalDateBeforeOrderDate, Line 16 [Batch Start Line 238]
ArrivalDate must happen after OrderDate!

Completion time: 2023-05-02T00:23:33.3417082+03:00
```

7.7 Sa se creeze un trigger care sa impiedice un pret unitar negativ

```
-------------------- Trigger 7 -----------------------
GO
IF OBJECT_ID('tr_MaterialsNegativeUnitPrice','TR') IS NOT NULL
	DROP TRIGGER tr_MaterialsNegativeUnitPrice

GO

CREATE TRIGGER tr_MaterialsNegativeUnitPrice
ON Materials
AFTER INSERT, UPDATE
AS
BEGIN
	IF @@ROWCOUNT = 0 RETURN;
	SET NOCOUNT ON;

	IF
	EXISTS (
		SELECT *
		FROM inserted I
		WHERE I.UnitPrice < 0
	)
	BEGIN
		;THROW 50000, 'Materials with a negative UnitPrice not
allowed!', 0
	END


	RETURN;
END
```

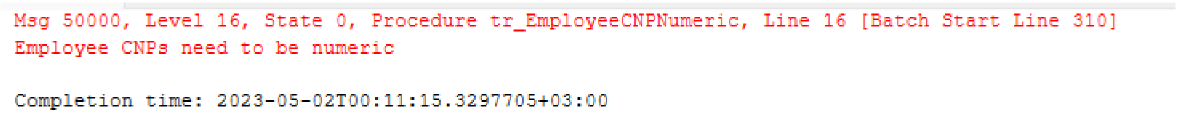


```
Msg 50000, Level 16, State 0, Procedure tr_MaterialsNegativeUnitPrice, Line 17 [Batch Start Line 276]
Materials with a negative UnitPrice not allowed!

Completion time: 2023-05-03T00:19:27.2193784+03:00
```

7.8 In tabelul Employees, sa se creeze un trigger care sa impiedice existenta CNP-urile non-numerice

```
-------------------- Trigger 8 -----------------------
GO
IF OBJECT_ID('tr_EmployeeCNPNumeric','TR') IS NOT NULL
	DROP TRIGGER tr_EmployeeCNPNumeric

GO
CREATE TRIGGER tr_EmployeeCNPNumeric
ON Employees
AFTER INSERT, UPDATE
AS
BEGIN
	IF @@ROWCOUNT = 0 RETURN;
	SET NOCOUNT ON;
```

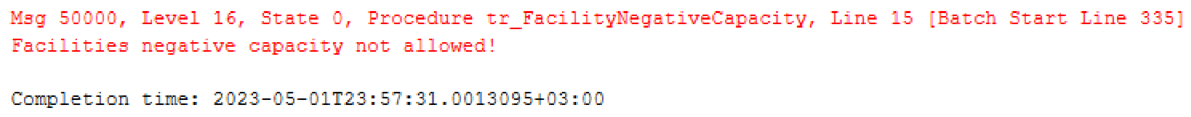
```
      IF EXISTS(
            SELECT *
            FROM Inserted I
            WHERE
            ISNUMERIC(I.CNP) = 0
      )
      BEGIN
            ;THROW 50000, 'Employee CNPs need to be numeric', 0
      END

      RETURN;
END
```



```
Msg 50000, Level 16, State 0, Procedure tr_EmployeeCNPNumeric, Line 16 [Batch Start Line 310]
Employee CNPs need to be numeric

Completion time: 2023-05-02T00:11:15.3297705+03:00
```

7.9 In tabelul Facilities, sa se creeze un trigger care sa impiedice capacitatea negative a unei locatii

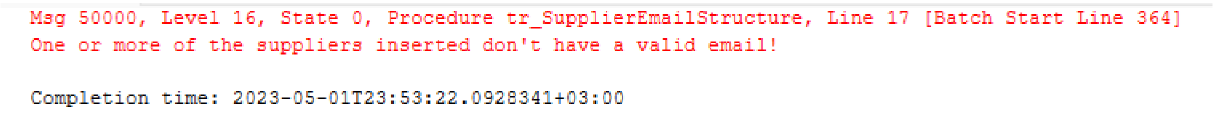
```
-------------------- Trigger 9 -----------------------

GO
IF OBJECT_ID('tr_FacilityNegativeCapacity','TR') IS NOT NULL
      DROP TRIGGER tr_FacilityNegativeCapacity

GO
CREATE TRIGGER tr_FacilityNegativeCapacity
ON Facilities
AFTER INSERT, UPDATE
AS
BEGIN
      IF @@ROWCOUNT = 0 RETURN;
      SET NOCOUNT ON;

      IF EXISTS(
            SELECT *
            FROM Facilities F
            WHERE F.Capacity < 0
      )
      BEGIN
            ;THROW 50000, 'Facilities negative capacity not allowed!', 0
      END

      RETURN;
END
```



```
Msg 50000, Level 16, State 0, Procedure tr_FacilityNegativeCapacity, Line 15 [Batch Start Line 335]
Facilities negative capacity not allowed!

Completion time: 2023-05-01T23:57:31.0013095+03:00
```

7.10 In tabelul Suppliers, sa se creeze un trigger care sa verifice structura adreselor de email

```
-------------------- Trigger 10 ----------------------

GO
IF OBJECT_ID('tr_SupplierEmailStructure','TR') IS NOT NULL
      DROP TRIGGER tr_SupplierEmailStructure

GO

CREATE TRIGGER tr_SupplierEmailStructure
ON Suppliers
AFTER INSERT, UPDATE
AS
BEGIN
      IF @@ROWCOUNT = 0 RETURN;
      SET NOCOUNT ON;

      IF
      EXISTS (
            SELECT *
            FROM inserted I
            WHERE I.Email NOT LIKE '%@%.%'
      )
      BEGIN
            ;THROW 50000, 'One or more of the suppliers inserted don''t
have a valid email!', 0
      END

      RETURN;
END
```



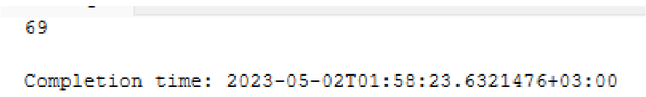
```
Msg 50000, Level 16, State 0, Procedure tr_SupplierEmailStructure, Line 17 [Batch Start Line 364]
One or more of the suppliers inserted don't have a valid email!

Completion time: 2023-05-01T23:53:22.0928341+03:00
```

8 Proceduri Stocate

```
-------------------------------------------------------
-------------------- Procedures -----------------------
-------------------------------------------------------
```

8.1 Sa se creeze o procedura stocata pentru calcularea rating-ului mediu al planificarii unui curs

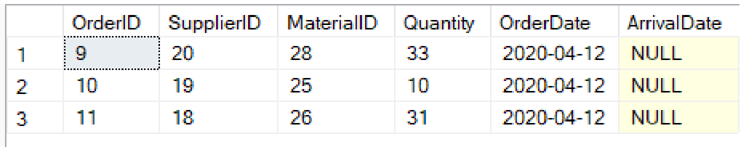
```
---------------------  PROC 1  -----------------------

GO

IF OBJECT_ID('GetCourseAvgRating', 'P') IS NOT NULL
      DROP PROC GetCourseAvgRating;
GO

CREATE PROC GetCourseAvgRating
      @targetScheduleId AS INT,
      @avgRating AS INT = 0 OUTPUT
AS
BEGIN
      SET @avgRating = (
            SELECT AVG(Rating)
            FROM CourseEvaluations
            WHERE ScheduleID = @targetScheduleId
      )
      PRINT @avgRating
      RETURN;
END
```



```
69

Completion time: 2023-05-02T01:58:23.6321476+03:00
```

8.2 Sa se creeze o procedura stocata care sa afiseze toate comenzile dintr-o data anume

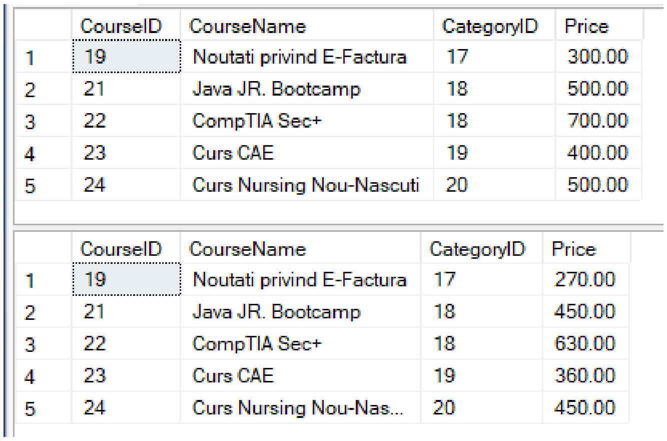
```
---------------------  PROC 2  -----------------------
GO

IF OBJECT_ID('GetOrdersOnDate','P') IS NOT NULL
      DROP PROC GetOrdersOnDate;
GO

CREATE PROC GetOrdersOnDate
      @orderDate AS DATE
AS
BEGIN
      SELECT *
      FROM Orders
      WHERE OrderDate = @orderDate
      RETURN;
END
```



	OrderID	SupplierID	MaterialID	Quantity	OrderDate	ArrivalDate
1	9	20	28	33	2020-04-12	NULL
2	10	19	25	10	2020-04-12	NULL
3	11	18	26	31	2020-04-12	NULL

8.3 Sa se creeze o procedura stocata care sa aplice un discount X tuturor cursurilor dintr-o categorie, sau a tuturor cursurilor daca aceasta nu se specifica

```
---------------------  PROC 3  -----------------------
GO

IF OBJECT_ID('ApplyDiscountToCourses', 'P') IS NOT NULL
      DROP PROC ApplyDiscountToCourses;
GO

CREATE PROC ApplyDiscountToCourses
      @percentage AS REAL,
      @categoryID AS INT = NULL
AS
BEGIN
      IF @categoryID IS NULL
      BEGIN
            UPDATE Courses
            SET Price = Price - Price * @percentage
      END
      ELSE
      BEGIN
            UPDATE Courses
            SET Price = Price - Price * @percentage
            WHERE CategoryID = @categoryID
      END

      RETURN;
END
```

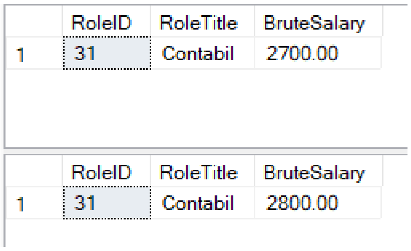


	CourseID	CourseName	CategoryID	Price
1	19	Noutati privind E-Factura	17	300.00
2	21	Java JR. Bootcamp	18	500.00
3	22	CompTIA Sec+	18	700.00
4	23	Curs CAE	19	400.00
5	24	Curs Nursing Nou-Nascuti	20	500.00

	CourseID	CourseName	CategoryID	Price
1	19	Noutati privind E-Factura	17	270.00
2	21	Java JR. Bootcamp	18	450.00
3	22	CompTIA Sec+	18	630.00
4	23	Curs CAE	19	360.00
5	24	Curs Nursing Nou-Nas...	20	450.00

8.4 Sa se creeze o procedura stocata care sa mareasca salariul brut pentru o functie X cu o valoare de Y

```
----------------------  PROC 4  ------------------------
GO

IF OBJECT_ID('RaiseBruteSalary','P') IS NOT NULL
      DROP PROC RaiseBruteSalary
GO

CREATE PROC RaiseBruteSalary
      @roleTitle AS VARCHAR(50),
      @bonus AS INT = 0
AS
BEGIN
      IF @roleTitle IS NULL
      BEGIN
            ;THROW 5000, '@roleTitle cannot be null!', 0;
      END

      SELECT *
      FROM EmployeeRoles
      WHERE RoleTitle = @roleTitle;

      IF @@ROWCOUNT = 0
      BEGIN
            ;THROW 5000, 'No role found with that title.', 0;
      END

      UPDATE EmployeeRoles
      SET BruteSalary = BruteSalary + @bonus
      WHERE RoleTitle = @roleTitle

      RETURN;
END
```



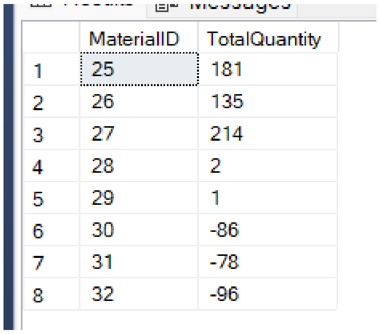
	RoleID	RoleTitle	BruteSalary
1	31	Contabil	2700.00

	RoleID	RoleTitle	BruteSalary
1	31	Contabil	2800.00

8.5 Sa se creeze o procedura stocata care sa intoarca diferenta dintre stock si cantitatea necesara pentru un curs pe fiecare material in parte

```
----------------------  PROC 5  ------------------------

-- returns the difference between materials for in stock and materials
needed for a course
GO
```

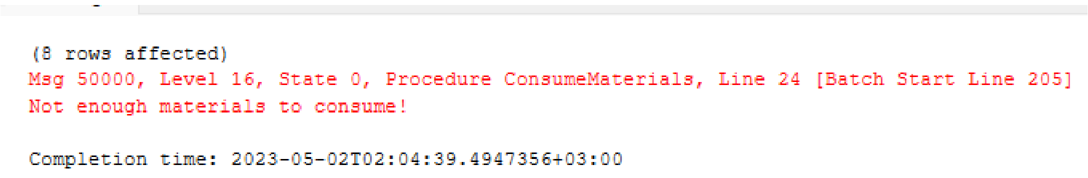
```
IF OBJECT_ID('DiffStockCourse','P') IS NOT NULL
      DROP PROC DiffStockCourse
GO

CREATE PROC DiffStockCourse
      @scheduleID AS INT
AS
BEGIN
      SELECT MaterialID, SUM(Quantity) as TotalQuantity
      FROM
      (
            SELECT CM.MaterialID AS MaterialID, -CM.Quantity AS Quantity
            FROM CourseMaterials CM
            WHERE CM.ScheduleID = @scheduleID
            UNION
            SELECT M.MaterialID, M.UnitsInStock
            FROM Materials M
      ) t
      GROUP BY MaterialID

      RETURN;
END
```



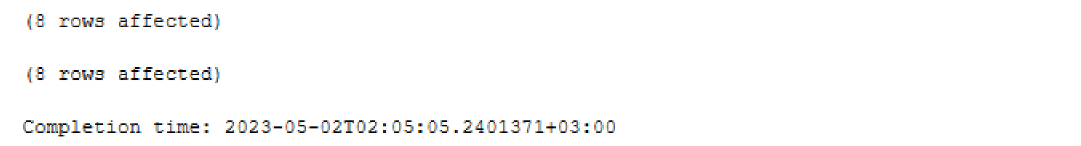
	MaterialID	TotalQuantity
1	25	181
2	26	135
3	27	214
4	28	2
5	29	1
6	30	-86
7	31	-78
8	32	-96

8.6 Sa se creeze o procedura stocata care sa consume toate materialele din stock in functie de cat are nevoie planificarea X

```
----------------------  PROC 6  ------------------------
GO

IF OBJECT_ID('ConsumeMaterials','P') IS NOT NULL
      DROP PROC ConsumeMaterials;

GO

CREATE PROC ConsumeMaterials
      @scheduleID AS INT
AS
BEGIN

      DECLARE @table AS Table(
            MaterialID INT,
		Quantity INT
	)

	INSERT INTO @table
	EXEC DiffStockCourse @scheduleID

	DECLARE @min AS INT

	SET @min = (
		SELECT MIN(Quantity)
		FROM @table
	)

	IF @min < 0
	BEGIN
		;THROW 50000,'Not enough materials to consume!',0;
	END

	UPDATE M
	SET M.UnitsInStock = T.Quantity
	FROM Materials M INNER JOIN @table AS T
	ON M.MaterialID = T.MaterialID

	RETURN;
END
```



```
(8 rows affected)
Msg 50000, Level 16, State 0, Procedure ConsumeMaterials, Line 24 [Batch Start Line 205]
Not enough materials to consume!

Completion time: 2023-05-02T02:04:39.4947356+03:00
```

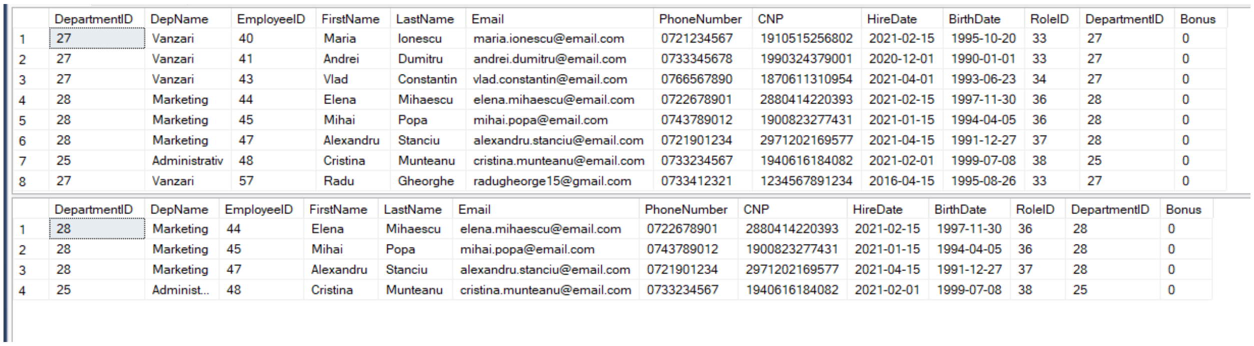


```
(8 rows affected)

(8 rows affected)

Completion time: 2023-05-02T02:05:05.2401371+03:00
```

8.7 Sa se creeze o procedura stocata care sa dea afara pe toti angajatii din departamentul cu numele X

```
---------------------  PROC 7  -----------------------
GO

IF OBJECT_ID('FireAllFromDepartment','P') IS NOT NULL
	DROP PROC FireAllFromDepartment;

GO

CREATE PROC FireAllFromDepartment
	@depName as VARCHAR(50)
AS
```

```
BEGIN

      DELETE E
      FROM Employees E
      WHERE E.DepartmentID = (
            SELECT TOP 1 D.DepartmentID
            FROM Departments D
            WHERE D.DepName = @depName
      )

      RETURN;
END
```

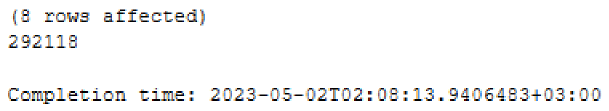


	DepartmentID	DepName	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus
1	27	Vanzari	40	Maria	Ionescu	maria.ionescu@email.com	0721234567	1910515256802	2021-02-15	1995-10-20	33	27	0
2	27	Vanzari	41	Andrei	Dumitru	andrei.dumitru@email.com	0733345678	1990324379001	2020-12-01	1990-01-01	33	27	0
3	27	Vanzari	43	Vlad	Constantin	vlad.constantin@email.com	0766567890	1870611310954	2021-04-01	1993-06-23	34	27	0
4	28	Marketing	44	Elena	Mihaescu	elena.mihaescu@email.com	0722678901	2880414220393	2021-02-15	1997-11-30	36	28	0
5	28	Marketing	45	Mihai	Popa	mihai.popa@email.com	0743789012	1900823277431	2021-01-15	1994-04-05	36	28	0
6	28	Marketing	47	Alexandru	Stanciu	alexandru.stanciu@email.com	0721901234	2971202169577	2021-04-15	1991-12-27	37	28	0
7	25	Administrativ	48	Cristina	Munteanu	cristina.munteanu@email.com	0733234567	1940616184082	2021-02-01	1999-07-08	38	25	0
8	27	Vanzari	57	Radu	Gheorghe	radugheorge15@gmail.com	0733412321	1234567891234	2016-04-15	1995-08-26	33	27	0

	DepartmentID	DepName	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus
1	28	Marketing	44	Elena	Mihaescu	elena.mihaescu@email.com	0722678901	2880414220393	2021-02-15	1997-11-30	36	28	0
2	28	Marketing	45	Mihai	Popa	mihai.popa@email.com	0743789012	1900823277431	2021-01-15	1994-04-05	36	28	0
3	28	Marketing	47	Alexandru	Stanciu	alexandru.stanciu@email.com	0721901234	2971202169577	2021-04-15	1991-12-27	37	28	0
4	25	Administ...	48	Cristina	Munteanu	cristina.munteanu@email.com	0733234567	1940616184082	2021-02-01	1999-07-08	38	25	0

8.8 Sa se creeze o procedura stocata care sa calculeze costul pentru materialele necessitate de planificarea X

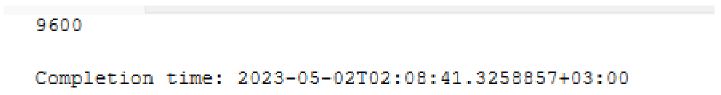
```
---------------------  PROC 8  -----------------------

GO

IF OBJECT_ID('CalculateMaterialCost', 'P') IS NOT NULL
      DROP PROC CalculateMaterialCost;
GO

CREATE PROC CalculateMaterialCost
      @scheduleID AS INT,
      @materialCost AS INT = 0 OUTPUT
AS
BEGIN
      DECLARE @table AS TABLE (
            MaterialID INT,
            Quantity INT
      )

      INSERT INTO @table
      EXEC DiffStockCourse @scheduleID

      SET @materialCost = (
            SELECT SUM((-T.Quantity) * M.UnitPrice)
            FROM @table as T INNER JOIN Materials as M
            ON T.MaterialID = M.MaterialID
            WHERE T.Quantity < 0
      )


      RETURN;
END
```



```
(8 rows affected)
292118

Completion time: 2023-05-02T02:08:13.9406483+03:00
```

8.9 Sa se creeze o procedura stocata pentru calcularea de costuri legate de locatia in care se va desfasura o planificare cu id-ul X

```
---------------------  PROC 9  -----------------------

GO

IF OBJECT_ID('CalculateFacilityCost', 'P') IS NOT NULL
      DROP PROC CalculateFacilityCost
GO

CREATE PROC CalculateFacilityCost
      @scheduleID AS INT,
      @facilityCost AS INT = 0 OUTPUT
AS
BEGIN
      SET @facilityCost = (
            SELECT F.PricePerHour * 24 * DATEDIFF(DAY, CS.StartDate, 
CS.EndDate)
            FROM Facilities F INNER JOIN CourseSchedule CS
            ON F.FacilityID = CS.FacilityID
            WHERE CS.ScheduleID = @scheduleID
      )
      RETURN;
END
```

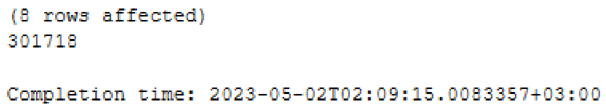


```
9600

Completion time: 2023-05-02T02:08:41.3258857+03:00
```

8.10 Sa se creeze o procedura stocata care sa calculeze costul total, avand in vedere costul locatiei si costul materialelor necesare pentru o planificare cu id-ul X

```
--------------------  PROC 10  -----------------------

GO
```

```
IF OBJECT_ID('CalculateTotalCost','P') IS NOT NULL
      DROP PROC CalculateTotalCost
GO

CREATE PROC CalculateTotalCost
      @scheduleID AS INT,
      @totalCost AS INT = 0 OUTPUT
AS
BEGIN

      DECLARE @out1 AS INT = 0;
      DECLARE @out2 AS INT = 0;

      EXEC CalculateMaterialCost
            @scheduleID ,
            @out1 OUTPUT

      EXEC CalculateFacilityCost
            @scheduleID,
            @out2 OUTPUT

      SET @totalCost = @out1 + @out2

      RETURN;
END
```



```
(8 rows affected)
301718

Completion time: 2023-05-02T02:09:15.0083357+03:00
```

9 Inserarea de date

```
-------------------------------------------------------
--------------------- SEEDING ------------------------
-------------------------------------------------------

BEGIN TRY
      BEGIN TRANSACTION
```

9.1 Popularea tabelului Departments

```
      INSERT INTO Departments(DepName)
      VALUES
      ('Resurse Umane'),
      ('Comercial'),
      ('Financiar'),
      ('Administrativ'),
      ('IT'),
      ('Vanzari'),
      ('Marketing');
```

9.2 Popularea tabelului MaterialCategories

```
      INSERT INTO MaterialCategories
      VALUES
      ('Birotica'),
      ('Electronice'),
      ('IT');
```

9.3 Popularea tabelului Facilities

```
      INSERT INTO
Facilities(FacilityName,FacilityAddress,City,Capacity,PricePerHour,PhoneNu
mber)
      VALUES
      ('Rin Grand Hotel',' Soseaua Vitan-Barzesti
7D','Bucuresti',50,200,'0310052102'),
      ('Moxy','Strada Doamnei 17-19','Bucuresti',30,400,'0311114400'),
      ('Novotel','Calea Victoriei 37B','Bucuresti',45,300,'0213088500'),
      ('Belvedere','Str. Calarasilor nr.1','Cluj',50,400,'0724211991'),
      ('Sunny Hill','Strada Fagetului 31A','Cluj',45,350,'0264480328'),
      ('Hotel Satu Mare','Bulevardul Vasile Lucaciu 42','Satu
Mare',35,200,'0261 877 870');
```

9.4 Popularea tabelului CourseCategories

```
      INSERT INTO CourseCategories
      VALUES
      ('Management'),
      ('Finante'),
      ('IT'),
      ('Limbi Straine'),
      ('Medical');
```

9.5 Popularea tabelului Students

```
      INSERT INTO Students(FirstName, LastName, Email, PhoneNumber, CNP)
      VALUES
      ('Radu','Pop','radu.pop1990@gmail.com','0721324565',
'1890422167893'),
      ('Alexandru', 'Popescu', 'alex.popescu_1995@gmail.com',
'0734567890', '2940214239076'),
      ('Catalin', 'Ionescu', 'catalin.ionescu@gmail.ro', '0741122334',
'2900227264302'),
      ('Ana-Maria', 'Radu', 'anamaria.radu90@gmail.com', '0723456789',
'1980813290516'),
      ('Elena', 'Pop', 'elena.pop.mail@gmail.com', '0721324565',
'2990418306923'),
      ('Andrei', 'Georgescu', 'a_georgescu96@gmail.ro', '0721765432',
'1930411276598'),
      ('Mara', 'Dumitrescu', 'mara.dumitrescu8_8@gmail.com', '0733123456',
'1960917251307'),
      ('Radu', 'Balan', 'radu.balan01@gmail.com', '0745654321',
'2950513290467'),
      ('Marius', 'Mihai', 'marius.mihai_75@gmail.com', '0722890132',
'2920816239078'),
      ('Corina', 'Stanescu', 'corina.stanescu95@gmail.ro', '0734567123',
'1971011253948');
```

9.6 Popularea tabelului EmployeeRoles

```
      INSERT INTO EmployeeRoles(RoleTitle, BruteSalary)
      VALUES
      ('Contabil',2700),
      ('Contabil Sef',4000),
      ('Agent Vanzari',2300),
      ('Director Vanzari',7000),
      ('Tehnician IT',3500),
      ('Agent Marketing',2300),
      ('Director Marketing',7000),
      ('Organizator',4000),
      ('Director General', 10000),
      ('Expert Vanzari', 5000);
```

9.7 Popularea tabelului Materials

```
      DECLARE @tempMaterials TABLE
      (
           MaterialName VARCHAR(50),
           CategoryName VARCHAR(50),
           UnitPrice DECIMAL(8,2),
           UnitsInStock INT
      )

      INSERT INTO @tempMaterials
      VALUES
      ('Mapa'                ,'Birotica'      ,3.21 ,200  ),
      ('Pix'                 ,'Birotica'      ,5.00 ,200  ),
      ('Pix BIC'       ,'Birotica'      ,2.00 ,300  ),
      ('Proiector'     ,'IT'            ,1600 ,2          ),
      ('Cafetiera'     ,'Electronice'   ,300  ,1          ),
      ('Imprimanta'    ,'IT'            ,409  ,2          ),
      ('Laptop'        ,'IT'            ,3000 ,10         ),
      ('Microunde'     ,'Electronice'   ,239  ,1          );

      INSERT INTO Materials(MaterialName, CategoryID, UnitPrice,
UnitsInStock)
      SELECT TM.MaterialName, MC.CategoryID, TM.UnitPrice, TM.UnitsInStock
      FROM @tempMaterials as TM INNER JOIN MaterialCategories MC
      ON TM.CategoryName = MC.CategoryName;
```

9.8 Popularea tabelului Suppliers

```
      DECLARE @tempSuppliers TABLE(
           SupplierName VARCHAR(50),
           CategoryName VARCHAR(50),
           Email VARCHAR(50),
           PhoneNumber VARCHAR(20),
           CIF VARCHAR(13)
      );

      INSERT INTO @tempSuppliers
      VALUES
      ('DIDAKTIKA'           ,'Birotica'      ,'relatii@didaktika.ro'
      ,'0721678921'    ,'RO30628475SRL'),
      ('EchipamenteIT' ,'IT'            ,'office@echipamente.ro'
      ,'0725050376'    ,'RO71904723SRL'),
      ('PaperSolutions','Birotica'      ,'office@papersol.ro'
      ,'0213110396'    ,'RO98371206SRL'),
      ('Birotica'            ,'Birotica'      ,'secretariat@birotica.ro'
      ,'0213178980'    ,'RO25983624SRL'),
      ('NeluElectronics'     ,'Electronice'   ,'relatii@neluelec.ro'
      ,'0216483491'    ,'RO57391826SRL');

      INSERT INTO Suppliers(SupplierName, CategoryID, Email, PhoneNumber,
CIF)
      SELECT TS.SupplierName, MC.CategoryID, TS.Email, TS.PhoneNumber,
TS.CIF
      FROM @tempSuppliers AS TS INNER JOIN MaterialCategories AS MC
```

```
ON TS.CategoryName = MC.CategoryName;
```

9.9 Popularea tabelului Courses

```
DECLARE @tempCourses TABLE
(
      CourseName VARCHAR(50),
      CategoryName VARCHAR(50),
      Price DECIMAL(10,2)
);

INSERT INTO @tempCourses
VALUES
(
'Noutati privind E-Factura',
'Finante',
300.00
),
(
'Management-ul riscului',
'Management',
200.00
),
(
'Java JR. Bootcamp',
'IT',
500.00
),
(
'CompTIA Sec+',
'IT',
700.00
),
(
'Curs CAE',
'Limbi Straine',
400.00
),
(
'Curs Nursing Nou-Nascuti',
'Medical',
500.00
);

INSERT INTO Courses(CourseName, CategoryID, Price)
SELECT TC.CourseName, CC.CategoryID, TC.Price
FROM @tempCourses as TC INNER JOIN CourseCategories CC
ON TC.CategoryName = CC.CategoryName;
```

9.10 Popularea tabelului Instructors

```
      DECLARE @tempInstructors TABLE
      (
            FirstName VARCHAR(50),
            LastName VARCHAR(50),
            Email VARCHAR(50),
            PhoneNumber VARCHAR(20),
            ExpertiseAreaName VARCHAR(50),
            CNP VARCHAR(13)
      );

      INSERT INTO @tempInstructors
      VALUES
      ('Marcel'   ,'Ghita'         ,'marcel.ghita@gmail.com'
      ,'0768060210'    ,'Management'    , '1930605239087'),
      ('Laur'          ,'Ionescu'       ,'laurionescu_60@gmail.com'
      ,'0722842101'    ,'Finante'       , '2950410276394'),
      ('Silviu'   ,'Teodor'        ,'silviuT1980@gmail.com'
      ,'0731567432'    ,'IT'            , '1980813214590'),
      ('Andrei'   ,'Kolanovski'    ,'andreikolo@gmail.com'
      ,'0764845193'    ,'Medical'       , '2900205298765'),
      ('Raluca'   ,'Pop'                 ,'raluka2009@gmail.com'
      ,'0726498668'    ,'Limbi Straine', '2910507309164');

      INSERT INTO
Instructors(FirstName,LastName,Email,PhoneNumber,ExpertiseArea, CNP)
      SELECT     TI.FirstName, TI.LastName, TI.Email, TI.PhoneNumber,
CC.CategoryID, TI.CNP
      FROM @tempInstructors as TI INNER JOIN CourseCategories AS CC
      ON TI.ExpertiseAreaName = CC.CategoryName;
```

9.11 Popularea tabelului Employees

```
      DECLARE @tempEmployees TABLE
      (
            FirstName VARCHAR(50),
            LastName VARCHAR(50),
            Email VARCHAR(50),
            PhoneNumber VARCHAR(20),
            HireDate DATE,
            BirthDate DATE,
            RoleTitle VARCHAR(50),
            DepName VARCHAR(50),
            CNP VARCHAR(13)
      );

      INSERT INTO @tempEmployees( FirstName, LastName, Email, PhoneNumber,
HireDate, BirthDate, RoleTitle, DepName, CNP )
      VALUES
      ('Maria'    , 'Apreutesei'   , 'maria_apr@gmail.com'
      , '0726347811',  '2009-05-15', '1986-03-12', 'Contabil Sef'
      ,'Financiar'           , '1860517212475'),
      ('Ion'            , 'Vasilescu'     , 'iceman2012@gmail.com'
      , '0724321123',  '2018-07-20', '1999-01-26', 'Tehnician IT'
      ,'IT'                   , '2930328439087'),
      ('Ion'            , 'Popescu'       , 'ion.popescu@email.com'
      , '0765123456',  '2021-01-01', '1990-05-15', 'Contabil'
      , 'Financiar'           , '2980126394176'),
      ('Maria'    , 'Ionescu'       , 'maria.ionescu@email.com'         ,
'0721234567',    '2021-02-15', '1995-10-20', 'Agent Vanzari'          ,
'Vanzari'               , '1910515256802'),
      ('Andrei'   , 'Dumitru'       , 'andrei.dumitru@email.com' ,
'0733345678',    '2020-12-01', '1992-08-10', 'Agent Vanzari'          ,
'Vanzari'               , '1990324379001'),
      ('Ana'            , 'Birsan'        , 'ana.birsan@email.com'
      , '0744456789',  '2021-03-15', '1998-02-18', 'Agent Vanzari'
      , 'Vanzari'             , '2940422175406'),
      ('Vlad'           , 'Constantin'    , 'vlad.constantin@email.com'
      , '0766567890',  '2021-04-01', '1993-06-23', 'Director Vanzari'
      , 'Vanzari'             , '1870611310954'),
      ('Elena'    , 'Mihaescu'      , 'elena.mihaescu@email.com' ,
'0722678901',    '2021-02-15', '1997-11-30', 'Agent Marketing' ,
'Marketing'      , '2880414220393'),
      ('Mihai'    , 'Popa'          , 'mihai.popa@email.com'            ,
'0743789012',    '2021-01-15', '1994-04-05', 'Agent Marketing' ,
'Marketing'      , '1900823277431'),
      ('Adina'    , 'Petrescu'      , 'adina.petrescu@email.com' ,
'0765890123',    '2021-03-01', '1996-09-12', 'Director General'       ,
'Administrativ'  , '2930817242188'),
      ('Alexandru', 'Stanciu'             , 'alexandru.stanciu@email.com'
      , '0721901234',  '2021-04-15', '1991-12-27', 'Director Marketing',
'Marketing'      , '2971202169577'),
      ('Cristina' , 'Munteanu'      , 'cristina.munteanu@email.com'     ,
'0733234567',    '2021-02-01', '1999-07-08', 'Organizator'            ,
'Administrativ'  , '1940616184082');

      INSERT INTO Employees( FirstName, LastName, Email, PhoneNumber,
HireDate, BirthDate, RoleID, DepartmentID, CNP )
      SELECT TE.FirstName, TE.LastName, TE.Email, TE.PhoneNumber,
TE.HireDate, TE.BirthDate, ER.RoleID, D.DepartmentID, TE.CNP
      FROM @tempEmployees AS TE
      INNER JOIN Departments D ON TE.DepName = D.DepName
      INNER JOIN EmployeeRoles ER ON TE.RoleTitle = ER.RoleTitle;
```

9.12 Popularea tabelului Orders

```
      DECLARE @ordersStartDate DATE = '2022-01-01'
      DECLARE @ordersMaxDayOffset INT = 800
      DECLARE @ordersMaxsDaysUntilArrival INT = 20

      INSERT INTO Orders(SupplierID, MaterialID, Quantity, OrderDate,
ArrivalDate)
      SELECT TOP 15
      SupplierID,
      MaterialID,
      ABS(CHECKSUM(NEWID()))%101,
      DATEADD(DAY, ABS(CHECKSUM(NEWID())) % @ordersMaxDayOffset,
@ordersStartDate),
      NULL
      FROM Suppliers S INNER JOIN Materials M
      ON S.CategoryID = M.CategoryID
      ORDER BY NEWID();

      UPDATE O
      SET O.ArrivalDate = DATEADD(DAY, ABS(CHECKSUM(NEWID())) %
@ordersMaxsDaysUntilArrival, O.OrderDate)
      FROM Orders O
```

9.13 Popularea tabelului CourseSchedule

```
      INSERT INTO CourseSchedule(CourseID, InstructorID, StartDate,
EndDate, FacilityID)
      VALUES
      (
            (SELECT TOP 1 CourseID FROM Courses ORDER BY NEWID()),
            (SELECT TOP 1 InstructorID FROM Instructors ORDER BY NEWID()),
            '2023-02-05',
            '2023-02-07',
            (SELECT TOP 1 FacilityID FROM Facilities ORDER BY NEWID())
      ),
      (
            (SELECT TOP 1 CourseID FROM Courses ORDER BY NEWID()),
            (SELECT TOP 1 InstructorID FROM Instructors ORDER BY NEWID()),
            '2023-03-18',
            '2023-03-21',
            (SELECT TOP 1 FacilityID FROM Facilities ORDER BY NEWID())
      ),
      (
            (SELECT TOP 1 CourseID FROM Courses ORDER BY NEWID()),
            (SELECT TOP 1 InstructorID FROM Instructors ORDER BY NEWID()),
            '2023-07-20',
            '2023-07-21',
            (SELECT TOP 1 FacilityID FROM Facilities ORDER BY NEWID())
      ),
      (
            (SELECT TOP 1 CourseID FROM Courses ORDER BY NEWID()),
            (SELECT TOP 1 InstructorID FROM Instructors ORDER BY NEWID()),
            '2023-08-15',
            '2023-08-17',
            (SELECT TOP 1 FacilityID FROM Facilities ORDER BY NEWID())
      ),
      (
            (SELECT TOP 1 CourseID FROM Courses ORDER BY NEWID()),
            (SELECT TOP 1 InstructorID FROM Instructors ORDER BY NEWID()),
            '2024-01-26',
            '2024-01-28',
            (SELECT TOP 1 FacilityID FROM Facilities ORDER BY NEWID())
      ),
      (
            (SELECT TOP 1 CourseID FROM Courses ORDER BY NEWID()),
            (SELECT TOP 1 InstructorID FROM Instructors ORDER BY NEWID()),
            '2024-02-14',
            '2024-02-15',
            (SELECT TOP 1 FacilityID FROM Facilities ORDER BY NEWID())
      ),
      (
            (SELECT TOP 1 CourseID FROM Courses ORDER BY NEWID()),
            (SELECT TOP 1 InstructorID FROM Instructors ORDER BY NEWID()),
            '2024-03-06',
            '2024-03-09',
            (SELECT TOP 1 FacilityID FROM Facilities ORDER BY NEWID())
      ),
      (
            (SELECT TOP 1 CourseID FROM Courses ORDER BY NEWID()),
            (SELECT TOP 1 InstructorID FROM Instructors ORDER BY NEWID()),
            '2023-06-23',
            '2023-06-25',
            (SELECT TOP 1 FacilityID FROM Facilities ORDER BY NEWID())
      ),
      (
            (SELECT TOP 1 CourseID FROM Courses ORDER BY NEWID()),
            (SELECT TOP 1 InstructorID FROM Instructors ORDER BY NEWID()),
            '2024-07-08',
            '2024-07-11',
            (SELECT TOP 1 FacilityID FROM Facilities ORDER BY NEWID())
      );
```

9.14 Popularea tabelului CourseEvaluations

```
      DECLARE @tempCourseEval TABLE ( Feedback TEXT );

      INSERT INTO @tempCourseEval
      VALUES
      ('Excelent curs, cu un continut foarte bine structurat si
informativ. Recomand tuturor!'),
      ('Din pacate, nu am fost multumit de curs. Profesorul nu a fost prea
implicat si mi s-a parut ca informatiile au fost prezentate haotic.'),
      ('Am apreciat mult faptul ca am putut invata la propriul meu ritm si
ca am avut acces la multe resurse online. Multumesc!'),
      ('Cursul a fost interesant si am invatat multe lucruri noi. Cu toate
acestea, as fi preferat mai multe exercitii practice.'),
      ('Un curs excelent pentru cei care doresc sa isi dezvolte
abilitatile de comunicare. Profesorul a fost foarte bine pregatit si a
creat o atmosfera prietenoasa si deschisa.'),
      ('Nu recomand acest curs. Am avut probleme cu accesul la materiale
si mi s-a parut ca informatiile prezentate nu au fost suficient de
clare.'),
      ('Cursul a fost exact ceea ce cautam. Informatii relevante si utile,
prezentate intr-un mod accesibil si captivant. Multumesc!'),
      ('Am fost dezamagit de acest curs. Am asteptat mai multe informatii
si exercitii practice. Mi s-a parut ca nu am invatat prea multe lucruri
noi.'),
      ('Profesorul a fost minunat. A fost foarte rabdator si a raspuns la
toate intrebarile noastre. Am apreciat mult efortul depus pentru a ne
ajuta sa intelegem subiectul.'),
      ('Cursul a fost foarte bine organizat si am apreciat mult faptul ca
am avut acces la feedback constant. Recomand acest curs cu incredere!');

      INSERT INTO CourseEvaluations(ScheduleID, StudentID, Rating,
Feedback)
      SELECT TOP 10 CS.ScheduleID, S.StudentID,
ABS(CHECKSUM(NEWID()))%100, TCE.Feedback
      FROM CourseSchedule CS CROSS JOIN Students S CROSS JOIN
@tempCourseEval TCE
      ORDER BY NEWID();
```

9.15 Popularea tabelului CourseMaterials

```
      INSERT INTO CourseMaterials(MaterialID, ScheduleID, Quantity)
      SELECT TOP 20 M.MaterialID, CS.ScheduleID,
ABS(CHECKSUM(NEWID()))%100
      FROM CourseSchedule CS CROSS JOIN Materials M
      ORDER BY NEWID();
```

9.16 Popularea tabelului CourseRegistrations

```
      INSERT INTO CourseRegistrations(ScheduleID, StudentID)
      SELECT TOP 20 CS.ScheduleID, S.StudentID
      FROM CourseSchedule CS CROSS JOIN Students S
      ORDER BY NEWID();
```

9.17 Popularea tabelului CourseAttendance

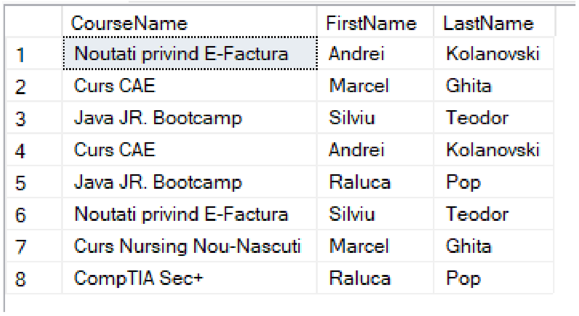
```
      INSERT INTO CourseAttendance(ScheduleID, StudentID)
      SELECT TOP 10 CR.ScheduleID, CR.StudentID
      FROM CourseRegistrations CR INNER JOIN CourseSchedule CS
      ON CR.ScheduleID = CS.ScheduleID
      WHERE DATEDIFF(DAY, CS.EndDate, CAST( GETDATE() AS DATE )) >= 0;

      COMMIT TRANSACTION
END TRY
BEGIN CATCH
      IF @@TRANCOUNT > 0
      BEGIN
            PRINT XACT_STATE();
            ROLLBACK TRANSACTION
      END
      PRINT ERROR_MESSAGE();
      THROW 50000, 'The transaction was cancelled', 0
END CATCH
```

10 SELECT

```
------------------------------------------------------
----------------- SELECT QUERIES ---------------------
------------------------------------------------------

-- 32 SELECT-uri

---- join pe cel putin 3 tabele
```

10.1 Sa se afiseze fiecare curs cu numele si prenumele lectorului

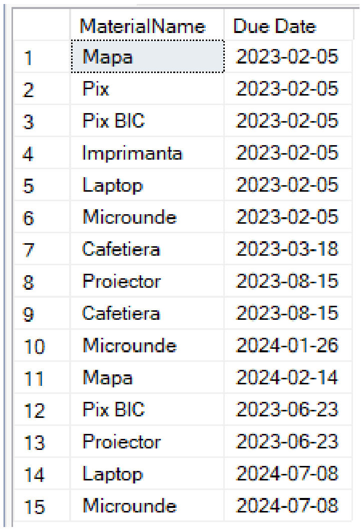
```
SELECT C.CourseName, I.FirstName, I.LastName
FROM CourseSchedule CS
INNER JOIN Courses C
ON CS.CourseID = C.CourseID
INNER JOIN Instructors I
ON CS.InstructorID = I.InstructorID
```



	CourseName	FirstName	LastName
1	Noutati privind E-Factura	Andrei	Kolanovski
2	Curs CAE	Marcel	Ghita
3	Java JR. Bootcamp	Silviu	Teodor
4	Curs CAE	Andrei	Kolanovski
5	Java JR. Bootcamp	Raluca	Pop
6	Noutati privind E-Factura	Silviu	Teodor
7	Curs Nursing Nou-Nascuti	Marcel	Ghita
8	CompTIA Sec+	Raluca	Pop

10.2 Sa se afiseze ce materiale (MaterialName) trebuie cumparate si pana cand (CourseSchedule.StartDate)

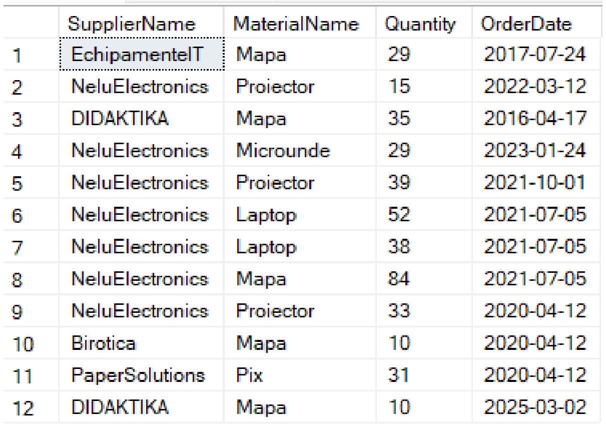
```
SELECT M.MaterialName, CS.StartDate as 'Due Date'
FROM CourseMaterials CM
INNER JOIN Materials M
ON CM.MaterialID = M.MaterialID
INNER JOIN CourseSchedule CS
ON CM.ScheduleID = CS.ScheduleID
```



	MaterialName	Due Date
1	Mapa	2023-02-05
2	Pix	2023-02-05
3	Pix BIC	2023-02-05
4	Imprimanta	2023-02-05
5	Laptop	2023-02-05
6	Microunde	2023-02-05
7	Cafetiera	2023-03-18
8	Proiector	2023-08-15
9	Cafetiera	2023-08-15
10	Microunde	2024-01-26
11	Mapa	2024-02-14
12	Pix BIC	2023-06-23
13	Proiector	2023-06-23
14	Laptop	2024-07-08
15	Microunde	2024-07-08

10.3 Sa se afiseze ce numele materialelor care au fost cumparate, cantitatea, data si numele furnizorului

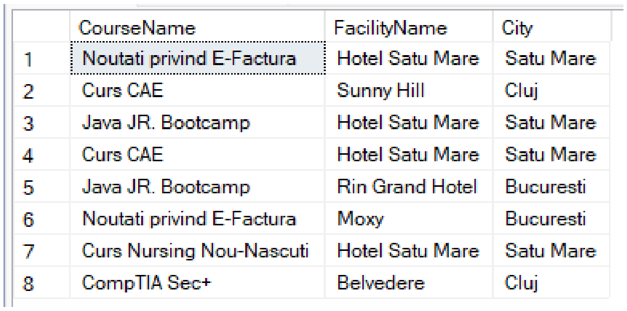
```
SELECT S.SupplierName, M.MaterialName, O.Quantity, O.OrderDate
FROM Orders O
INNER JOIN Suppliers S
ON O.SupplierID = S.SupplierID
INNER JOIN Materials M
ON O.MaterialID = M.MaterialID
```



	SupplierName	MaterialName	Quantity	OrderDate
1	EchipamenteIT	Mapa	29	2017-07-24
2	NeluElectronics	Proiector	15	2022-03-12
3	DIDAKTIKA	Mapa	35	2016-04-17
4	NeluElectronics	Microunde	29	2023-01-24
5	NeluElectronics	Proiector	39	2021-10-01
6	NeluElectronics	Laptop	52	2021-07-05
7	NeluElectronics	Laptop	38	2021-07-05
8	NeluElectronics	Mapa	84	2021-07-05
9	NeluElectronics	Proiector	33	2020-04-12
10	Birotica	Mapa	10	2020-04-12
11	PaperSolutions	Pix	31	2020-04-12
12	DIDAKTIKA	Mapa	10	2025-03-02

10.4 se afiseze orasul, in ce cladire va avea fiecare curs si numele cursului

```
SELECT C.CourseName, F.FacilityName, F.City
FROM CourseSchedule CS
INNER JOIN Courses C
ON CS.CourseID = C.CourseID
INNER JOIN Facilities F
ON CS.FacilityID = F.FacilityID
```

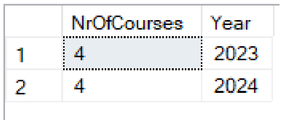


	CourseName	FacilityName	City
1	Noutati privind E-Factura	Hotel Satu Mare	Satu Mare
2	Curs CAE	Sunny Hill	Cluj
3	Java JR. Bootcamp	Hotel Satu Mare	Satu Mare
4	Curs CAE	Hotel Satu Mare	Satu Mare
5	Java JR. Bootcamp	Rin Grand Hotel	Bucuresti
6	Noutati privind E-Factura	Moxy	Bucuresti
7	Curs Nursing Nou-Nascuti	Hotel Satu Mare	Satu Mare
8	CompTIA Sec+	Belvedere	Cluj

```
---- gruparea datelor
```

10.5 Sa se afiseze numarul cursurilor pe fiecare an

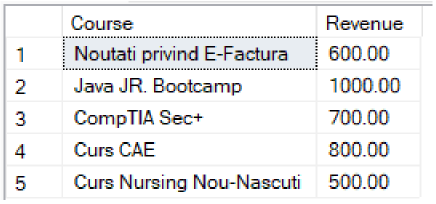
```
SELECT COUNT(C.CourseID) as 'NrOfCourses', DATEPART(YEAR, CS.StartDate) as
'Year'
FROM Courses C INNER JOIN CourseSchedule CS
ON C.CourseID = CS.CourseID
GROUP BY DATEPART(YEAR, CS.StartDate)
```



	NrOfCourses	Year
1	4	2023
2	4	2024

10.6 Sa se afiseze venitul total de la fiecare curs

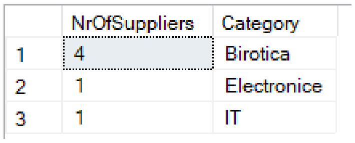
```
SELECT C.CourseName as 'Course',SUM(C.Price) as 'Revenue'
FROM CourseSchedule CS INNER JOIN Courses C
ON CS.CourseID = C.CourseID
GROUP BY C.CourseName
```



	Course	Revenue
1	Noutati privind E-Factura	600.00
2	Java JR. Bootcamp	1000.00
3	CompTIA Sec+	700.00
4	Curs CAE	800.00
5	Curs Nursing Nou-Nascuti	500.00

10.7 Sa se afiseze cati furnizori sunt pentru fiecare categorie de material

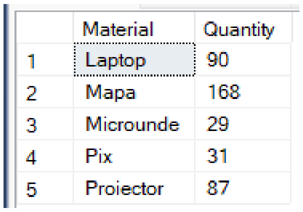
```
SELECT COUNT(S.SupplierID) as 'NrOfSuppliers', MC.CategoryName as
'Category'
FROM Suppliers S INNER JOIN MaterialCategories MC
ON S.CategoryID = MC.CategoryID
GROUP BY MC.CategoryName
```



	NrOfSuppliers	Category
1	4	Birotica
2	1	Electronice
3	1	IT

10.8 Sa se afiseze cantitatea totala comandata din fiecare material

```
SELECT M.MaterialName as 'Material', SUM(O.Quantity) as 'Quantity'
FROM Orders O INNER JOIN Materials M
ON O.MaterialID = M.MaterialID
GROUP BY M.MaterialName
```

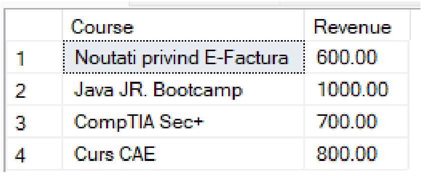


	Material	Quantity
1	Laptop	90
2	Mapa	168
3	Microunde	29
4	Pix	31
5	Proiector	87

```
---- filtrarea de grup
```

10.9 Sa se afiseze cursurile care au generat un venit mai mare de 500 lei

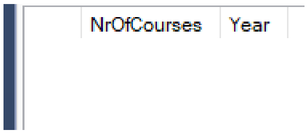
```
SELECT C.CourseName as 'Course',SUM(C.Price) as 'Revenue'
FROM CourseSchedule CS INNER JOIN Courses C
ON CS.CourseID = C.CourseID
GROUP BY C.CourseName
HAVING SUM(C.Price) > 500
```



	Course	Revenue
1	Noutati privind E-Factura	600.00
2	Java JR. Bootcamp	1000.00
3	CompTIA Sec+	700.00
4	Curs CAE	800.00

10.10 Sa se afiseze anii in care au avut loc mai mult de 4 cursuri

```
SELECT COUNT(C.CourseID) as 'NrOfCourses', DATEPART(YEAR, CS.StartDate) as
'Year'
FROM Courses C INNER JOIN CourseSchedule CS
ON C.CourseID = CS.CourseID
GROUP BY DATEPART(YEAR, CS.StartDate)
HAVING COUNT(C.CourseID) > 4
```

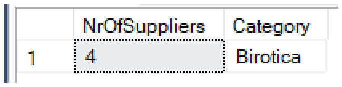


	NrOfCourses	Year

10.11 Sa se afiseze categoriile de materiale pentru care exista mai multi furnizori

```
SELECT COUNT(S.SupplierID) as 'NrOfSuppliers', MC.CategoryName as
'Category'
FROM Suppliers S INNER JOIN MaterialCategories MC
ON S.CategoryID = MC.CategoryID
GROUP BY MC.CategoryName
```

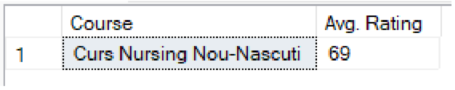
```
HAVING COUNT(S.SupplierID) > 1
```



	NrOfSuppliers	Category
1	4	Birotica

10.12 Sa se afiseze cursurile cu rating mediu mai mare de X

```
GO
DECLARE @AVGRating INT = 60

SELECT C.CourseName as 'Course', AVG(CE.Rating) as 'Avg. Rating'
FROM CourseSchedule CS
INNER JOIN Courses C
ON CS.CourseID = C.CourseID
INNER JOIN CourseEvaluations CE
ON CS.ScheduleID = CE.ScheduleID
GROUP BY C.CourseName
HAVING AVG(CE.Rating) > @AVGRating
```

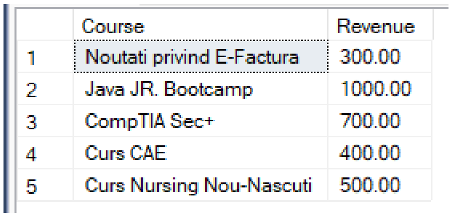


	Course	Avg. Rating
1	Curs Nursing Nou-Nascuti	69

```
---- gruparea datelor filtrate
```

10.13 Sa se afiseze venitul total de la fiecare curs viitor

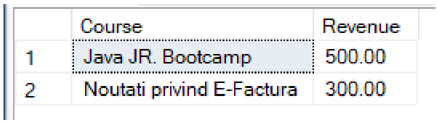
```
SELECT C.CourseName as 'Course',SUM(C.Price) as 'Revenue'
FROM CourseSchedule CS INNER JOIN Courses C
ON CS.CourseID = C.CourseID
WHERE DATEDIFF(DAY, CAST(GETDATE() AS DATE), CS.StartDate) >= 0
GROUP BY C.CourseName
```



	Course	Revenue
1	Noutati privind E-Factura	300.00
2	Java JR. Bootcamp	1000.00
3	CompTIA Sec+	700.00
4	Curs CAE	400.00
5	Curs Nursing Nou-Nascuti	500.00

10.14 Sa se afiseze venitul total de la fiecare curs din Bucuresti

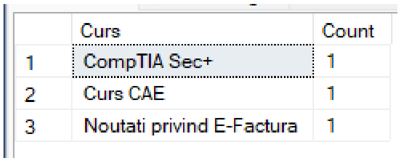
```
SELECT C.CourseName as 'Course',SUM(C.Price) as 'Revenue'
FROM CourseSchedule CS
INNER JOIN Courses C
ON CS.CourseID = C.CourseID
INNER JOIN Facilities F
ON CS.FacilityID = F.FacilityID
WHERE F.City = 'Bucuresti'
GROUP BY C.CourseName
```



	Course	Revenue
1	Java JR. Bootcamp	500.00
2	Noutati privind E-Factura	300.00

10.15 se afiseze cate planificari cu o durata mai mare de 2 zile sunt pentru fiecare curs

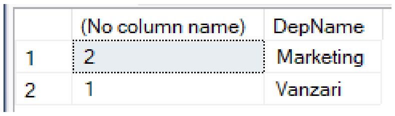
```
SELECT C.CourseName as 'Curs', COUNT(CS.ScheduleID) as 'Count'
FROM CourseSchedule CS
INNER JOIN Courses C
ON CS.CourseID = C.CourseID
WHERE DATEDIFF(DAY,CS.StartDate,CS.EndDate) > 2
GROUP BY C.CourseName
```



	Curs	Count
1	CompTIA Sec+	1
2	Curs CAE	1
3	Noutati privind E-Factura	1

10.16 Sa se afiseze cati angajati sunt in reteaua vodafone (nr. de telefon incepe cu 072) din fiecare departament

```
SELECT COUNT(E.EmployeeID), D.DepName
FROM Employees E
INNER JOIN Departments D
ON E.DepartmentID = D.DepartmentID
WHERE E.PhoneNumber LIKE '072%'
GROUP BY D.DepName
```

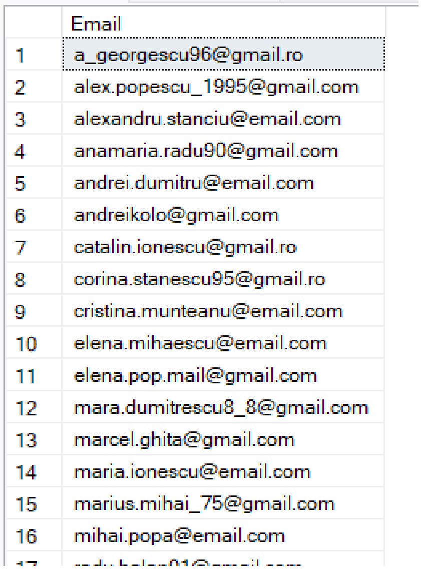


	(No column name)	DepName
1	2	Marketing
2	1	Vanzari

```
---- operatori pe seturi de date
```

10.17 Sa se afiseze toate email-urile din baza de date

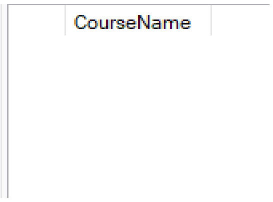
```
SELECT S.Email
FROM Students S
UNION
SELECT I.Email
FROM Instructors I
UNION
SELECT E.Email
FROM Employees E
```



	Email
1	a_georgescu96@gmail.ro
2	alex.popescu_1995@gmail.com
3	alexandru.stanciu@email.com
4	anamaria.radu90@gmail.com
5	andrei.dumitru@email.com
6	andreikolo@gmail.com
7	catalin.ionescu@gmail.ro
8	corina.stanescu95@gmail.ro
9	cristina.munteanu@email.com
10	elena.mihaescu@email.com
11	elena.pop.mail@gmail.com
12	mara.dumitrescu8_8@gmail.com
13	marcel.ghita@gmail.com
14	maria.ionescu@email.com
15	marius.mihai_75@gmail.com
16	mihai.popa@email.com
17	radu.balan91@gmail.com

10.18 Sa se afiseze cursurile care nu s-au vandut

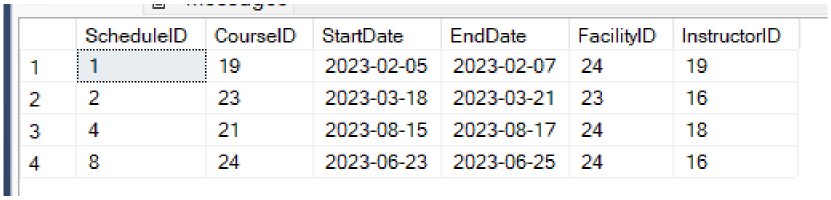
```
SELECT C.CourseName
FROM Courses C
WHERE C.CourseID IN
(
      SELECT C.CourseID
      FROM Courses C
      EXCEPT
      SELECT CS.CourseID
      FROM CourseSchedule CS
)
```



	CourseName

10.19 Sa se afiseze cursurile care s-au vandut anul asta si inca nu sunt in plan anul viitor

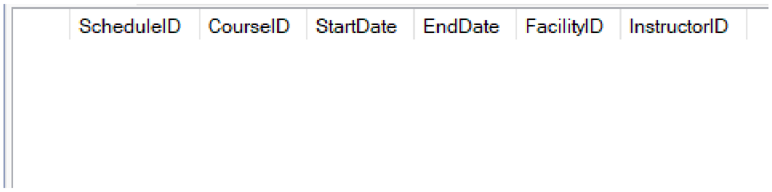
```
SELECT *
FROM CourseSchedule
WHERE DATEPART(YEAR, StartDate) = DATEPART(YEAR, GETDATE())
EXCEPT
SELECT *
FROM CourseSchedule
WHERE DATEPART(YEAR, StartDate) = DATEPART(YEAR, GETDATE()) + 1
```



	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID
1	1	19	2023-02-05	2023-02-07	24	19
2	2	23	2023-03-18	2023-03-21	23	16
3	4	21	2023-08-15	2023-08-17	24	18
4	8	24	2023-06-23	2023-06-25	24	16

10.20 Sa se afiseze cursurile care s-au vandut si anul asta si anul trecut

```
SELECT *
FROM CourseSchedule
WHERE DATEPART(YEAR, StartDate) = DATEPART(YEAR, GETDATE())
INTERSECT
SELECT *
FROM CourseSchedule
WHERE DATEPART(YEAR, StartDate) = DATEPART(YEAR, GETDATE()) - 1
```

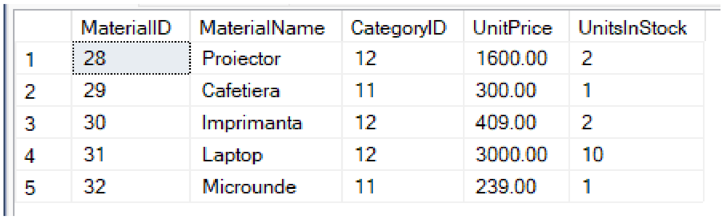


	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID

```
---- alte select-uri
```

10.21 Sa se afiseze toate materialele mai scumpe de 200 lei

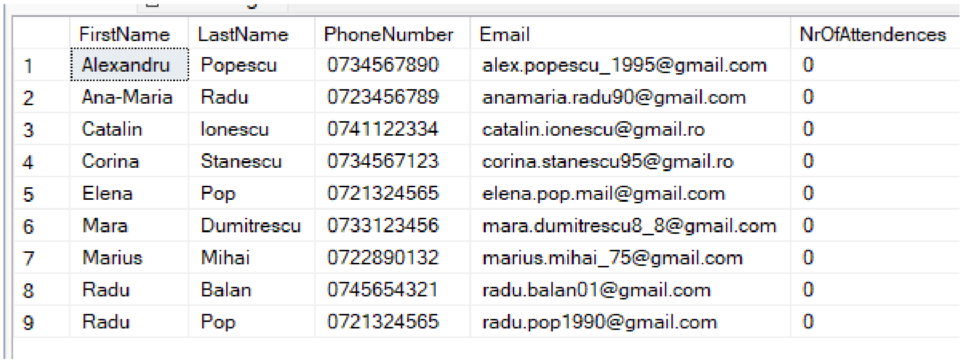
```
SELECT * FROM Materials WHERE UnitPrice >= 200
```



	MaterialID	MaterialName	CategoryID	UnitPrice	UnitsInStock
1	28	Proiector	12	1600.00	2
2	29	Cafetiera	11	300.00	1
3	30	Imprimanta	12	409.00	2
4	31	Laptop	12	3000.00	10
5	32	Microunde	11	239.00	1

10.22 Sa se afiseze studentii care nu au participat la niciun singur curs

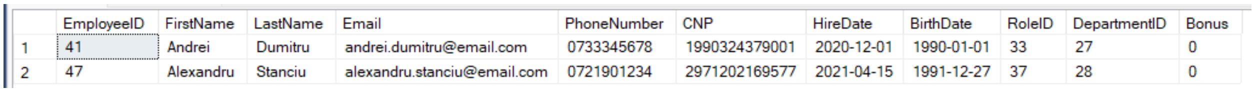
```
SELECT S.FirstName, S.LastName, S.PhoneNumber, S.Email ,
COUNT(CA.StudentID) as 'NrOfAttendences'
FROM Students S
LEFT JOIN CourseAttendance CA
ON S.StudentID = CA.StudentID
GROUP BY S.FirstName, S.LastName, S.PhoneNumber, S.Email
HAVING COUNT(CA.StudentID) = 0
```



	FirstName	LastName	PhoneNumber	Email	NrOfAttendences
1	Alexandru	Popescu	0734567890	alex.popescu_1995@gmail.com	0
2	Ana-Maria	Radu	0723456789	anamaria.radu90@gmail.com	0
3	Catalin	Ionescu	0741122334	catalin.ionescu@gmail.ro	0
4	Corina	Stanescu	0734567123	corina.stanescu95@gmail.ro	0
5	Elena	Pop	0721324565	elena.pop.mail@gmail.com	0
6	Mara	Dumitrescu	0733123456	mara.dumitrescu8_8@gmail.com	0
7	Marius	Mihai	0722890132	marius.mihai_75@gmail.com	0
8	Radu	Balan	0745654321	radu.balan01@gmail.com	0
9	Radu	Pop	0721324565	radu.pop1990@gmail.com	0

10.23 Sa se afiseze angajatii nascuti inainte de 1992

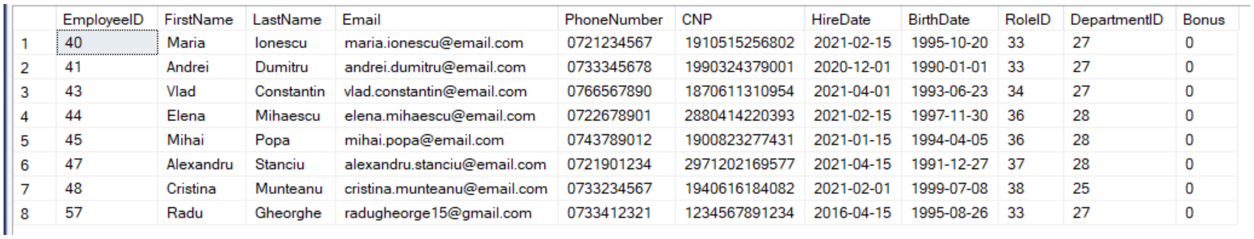
```
SELECT *
FROM Employees
WHERE DATEPART(YEAR, BirthDate) <= 1992
```



	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus
1	41	Andrei	Dumitru	andrei.dumitru@email.com	0733345678	1990324379001	2020-12-01	1990-01-01	33	27	0
2	47	Alexandru	Stanciu	alexandru.stanciu@email.com	0721901234	2971202169577	2021-04-15	1991-12-27	37	28	0

10.24 Sa se afiseze angajatii angajati dupa 2012

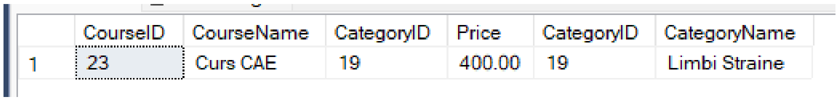
```
SELECT *
FROM Employees
WHERE DATEPART(YEAR, HireDate) >= 2012
```



	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus
1	40	Maria	Ionescu	maria.ionescu@email.com	0721234567	1910515256802	2021-02-15	1995-10-20	33	27	0
2	41	Andrei	Dumitru	andrei.dumitru@email.com	0733345678	1990324379001	2020-12-01	1990-01-01	33	27	0
3	43	Vlad	Constantin	vlad.constantin@email.com	0766567890	1870611310954	2021-04-01	1993-06-23	34	27	0
4	44	Elena	Mihaescu	elena.mihaescu@email.com	0722678901	2880414220393	2021-02-15	1997-11-30	36	28	0
5	45	Mihai	Popa	mihai.popa@email.com	0743789012	1900823277431	2021-01-15	1994-04-05	36	28	0
6	47	Alexandru	Stanciu	alexandru.stanciu@email.com	0721901234	2971202169577	2021-04-15	1991-12-27	37	28	0
7	48	Cristina	Munteanu	cristina.munteanu@email.com	0733234567	1940616184082	2021-02-01	1999-07-08	38	25	0
8	57	Radu	Gheorghe	radugheorge15@gmail.com	0733412321	1234567891234	2016-04-15	1995-08-26	33	27	0

10.25 Sa se afiseze cursurile de limbi straine

```
SELECT *
FROM Courses C
INNER JOIN CourseCategories CC
ON C.CategoryID = CC.CategoryID
WHERE CC.CategoryName = 'Limbi Straine'
```

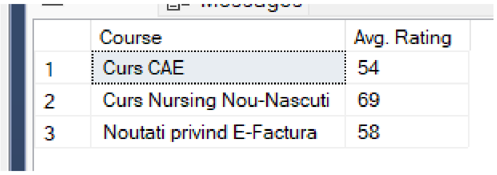


	CourseID	CourseName	CategoryID	Price	CategoryID	CategoryName
1	23	Curs CAE	19	400.00	19	Limbi Straine

```
-- extra
```

10.26 Sa se afiseze rating-ul fiecarui curs

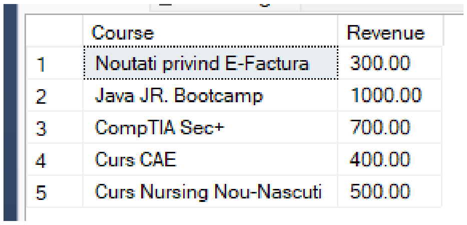
```
SELECT C.CourseName as 'Course', AVG(CE.Rating) as 'Avg. Rating'
FROM CourseSchedule CS
INNER JOIN Courses C
ON CS.CourseID = C.CourseID
INNER JOIN CourseEvaluations CE
ON CS.ScheduleID = CE.ScheduleID
GROUP BY C.CourseName
```



	Course	Avg. Rating
1	Curs CAE	54
2	Curs Nursing Nou-Nascuti	69
3	Noutati privind E-Factura	58

10.27 Sa se afiseze venitul total de la fiecare curs trecut

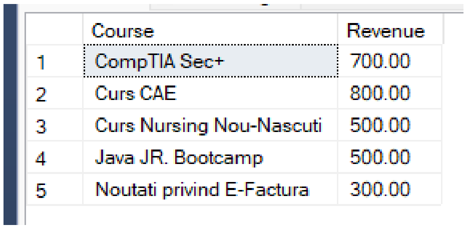
```
SELECT C.CourseName as 'Course',SUM(C.Price) as 'Revenue'
FROM CourseSchedule CS INNER JOIN Courses C
ON CS.CourseID = C.CourseID
WHERE DATEDIFF(DAY, CAST(GETDATE() AS DATE), CS.StartDate) >= 0
GROUP BY C.CourseName
```



	Course	Revenue
1	Noutati privind E-Factura	300.00
2	Java JR. Bootcamp	1000.00
3	CompTIA Sec+	700.00
4	Curs CAE	400.00
5	Curs Nursing Nou-Nascuti	500.00

10.28 Sa se afiseze venitul total de la fiecare curs dinafara Bucuresti-ului

```
SELECT C.CourseName as 'Course',SUM(C.Price) as 'Revenue'
FROM CourseSchedule CS
INNER JOIN Courses C
ON CS.CourseID = C.CourseID
INNER JOIN Facilities F
ON CS.FacilityID = F.FacilityID
WHERE F.City <> 'Bucuresti'
GROUP BY C.CourseName
```

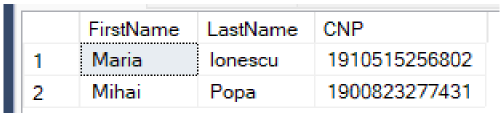


	Course	Revenue
1	CompTIA Sec+	700.00
2	Curs CAE	800.00
3	Curs Nursing Nou-Nascuti	500.00
4	Java JR. Bootcamp	500.00
5	Noutati privind E-Factura	300.00

10.29 Sa se afiseze angajatii al caror nume incepe cu M

```
SELECT FirstName, LastName, CNP
```

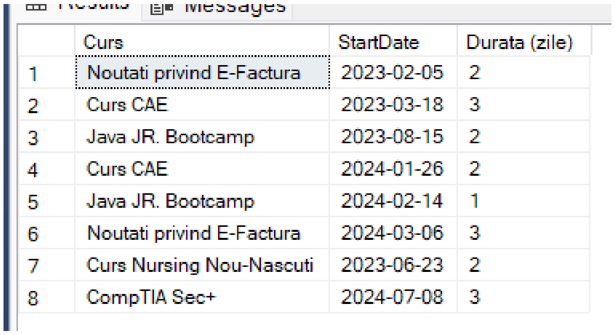
```
FROM Employees
WHERE FirstName LIKE 'M%'
```



	FirstName	LastName	CNP
1	Maria	Ionescu	1910515256802
2	Mihai	Popa	1900823277431

10.30 se afiseze cate zile tine fiecare curs planificat si cand incepe

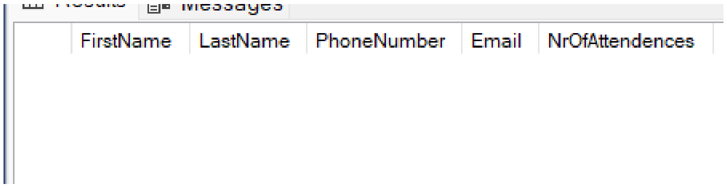
```
SELECT C.CourseName as 'Curs', CS.StartDate,
DATEDIFF(DAY,CS.StartDate,CS.EndDate) as 'Durata (zile)'
FROM CourseSchedule CS
INNER JOIN Courses C
ON CS.CourseID = C.CourseID
```



	Curs	StartDate	Durata (zile)
1	Noutati privind E-Factura	2023-02-05	2
2	Curs CAE	2023-03-18	3
3	Java JR. Bootcamp	2023-08-15	2
4	Curs CAE	2024-01-26	2
5	Java JR. Bootcamp	2024-02-14	1
6	Noutati privind E-Factura	2024-03-06	3
7	Curs Nursing Nou-Nascuti	2023-06-23	2
8	CompTIA Sec+	2024-07-08	3

10.31 Sa se afiseze studentii care au participat la mai multe cursuri

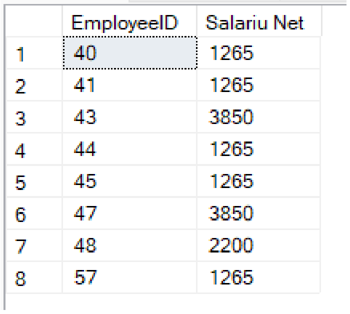
```
SELECT S.FirstName, S.LastName, S.PhoneNumber, S.Email ,
COUNT(CA.StudentID) as 'NrOfAttendences'
FROM Students S
LEFT JOIN CourseAttendance CA
ON S.StudentID = CA.StudentID
GROUP BY S.FirstName, S.LastName, S.PhoneNumber, S.Email
HAVING COUNT(CA.StudentID) > 1
```



	FirstName	LastName	PhoneNumber	Email	NrOfAttendences

10.32 Sa se afiseze salariul net al angajatilor

```
DECLARE @impozit REAL = 0.45

SELECT E.EmployeeID ,( ER.BruteSalary + E.Bonus ) - ( ER.BruteSalary +
E.Bonus ) * @impozit as 'Salariu Net'
FROM Employees E
INNER JOIN EmployeeRoles ER
ON E.RoleID = ER.RoleID
```



	EmployeeID	Salariu Net
1	40	1265
2	41	1265
3	43	3850
4	44	1265
5	45	1265
6	47	3850
7	48	2200
8	57	1265

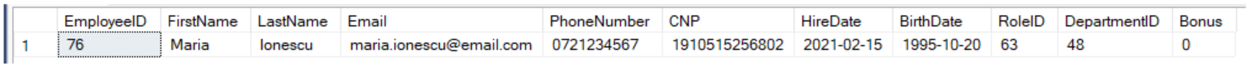
11 UPDATE

```
-------------------------------------------------------
----------------- UPDATE QUERIES ---------------------
-------------------------------------------------------
```

11.1 Schimba data de angajare al unui angajat dupa CNP

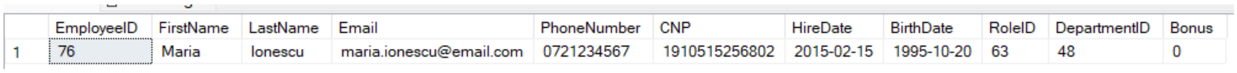
```
GO
DECLARE @CNP VARCHAR(13) = '1910515256802'
DECLARE @newHireDate DATE = '2015-02-15'

UPDATE Employees
SET HireDate = @newHireDate
WHERE CNP = @CNP
```

Before



	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus
1	76	Maria	Ionescu	maria.ionescu@email.com	0721234567	1910515256802	2021-02-15	1995-10-20	63	48	0

After

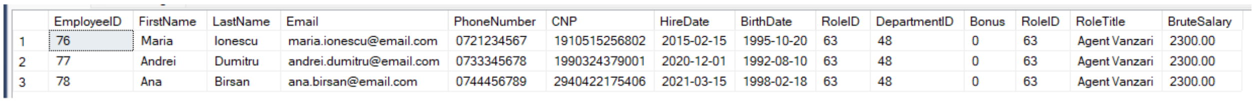


	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus
1	76	Maria	Ionescu	maria.ionescu@email.com	0721234567	1910515256802	2015-02-15	1995-10-20	63	48	0

11.2 Avansarea dupa un anume numar de ani

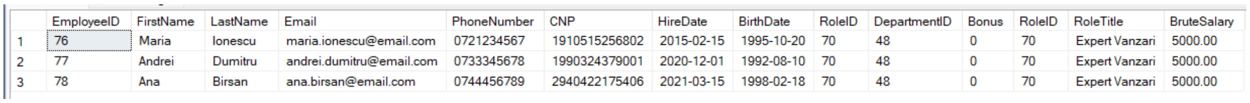
```
GO
DECLARE @oldTitle VARCHAR(50) = 'Agent Vanzari'
DECLARE @newTitle VARCHAR(50) = 'Expert Vanzari'
DECLARE @seniority INT = 2

UPDATE E
SET E.RoleID = ( SELECT TOP 1 RoleID FROM EmployeeRoles WHERE RoleTitle =
@newTitle )
FROM Employees E
INNER JOIN EmployeeRoles ER
ON E.RoleID = ER.RoleID
WHERE DATEDIFF(YEAR, E.HireDate, GETDATE()) >= @seniority AND ER.RoleTitle
= @oldTitle
```

Before



	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus	RoleID	RoleTitle	BruteSalary
1	76	Maria	Ionescu	maria.ionescu@email.com	0721234567	1910515256802	2015-02-15	1995-10-20	63	48	0	63	Agent Vanzari	2300.00
2	77	Andrei	Dumitru	andrei.dumitru@email.com	0733345678	1990324379001	2020-12-01	1992-08-10	63	48	0	63	Agent Vanzari	2300.00
3	78	Ana	Birsan	ana.birsan@email.com	0744456789	2940422175406	2021-03-15	1998-02-18	63	48	0	63	Agent Vanzari	2300.00

After

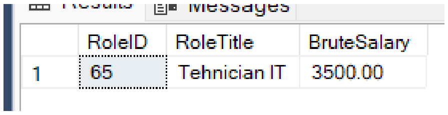


	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus	RoleID	RoleTitle	BruteSalary
1	76	Maria	Ionescu	maria.ionescu@email.com	0721234567	1910515256802	2015-02-15	1995-10-20	70	48	0	70	Expert Vanzari	5000.00
2	77	Andrei	Dumitru	andrei.dumitru@email.com	0733345678	1990324379001	2020-12-01	1992-08-10	70	48	0	70	Expert Vanzari	5000.00
3	78	Ana	Birsan	ana.birsan@email.com	0744456789	2940422175406	2021-03-15	1998-02-18	70	48	0	70	Expert Vanzari	5000.00

11.3 Actualizarea salariului brut al unei specializari

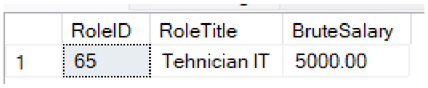
```
GO
DECLARE @newSalary INT = 5000
DECLARE @roleTitle VARCHAR(50) = 'Tehnician IT'

UPDATE EmployeeRoles
SET BruteSalary = @newSalary
WHERE RoleTitle = @roleTitle
```

Before



	RoleID	RoleTitle	BruteSalary
1	65	Tehnician IT	3500.00

After

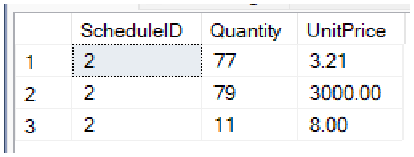


	RoleID	RoleTitle	BruteSalary
1	65	Tehnician IT	5000.00

11.4 Sa se reduca cu un procent cantitatea materialelor mai scumpe de un anumit pret folosite pentru un curs anume

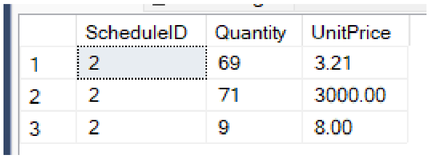
```
GO
DECLARE @courseName VARCHAR(50) = 'Noutati privind E-Factura'
DECLARE @percentage REAL = 0.1
DECLARE @maxPrice INT = 3

UPDATE CM
SET CM.Quantity = CM.Quantity - CM.Quantity * @percentage
FROM CourseMaterials CM
INNER JOIN Materials M
ON CM.MaterialID = M.MaterialID
INNER JOIN CourseSchedule CS
ON CM.ScheduleID = CS.ScheduleID
INNER JOIN Courses C
ON CS.CourseID = C.CourseID
WHERE C.CourseName = @courseName AND M.UnitPrice >= @maxPrice
```

Before



	ScheduleID	Quantity	UnitPrice
1	2	77	3.21
2	2	79	3000.00
3	2	11	8.00

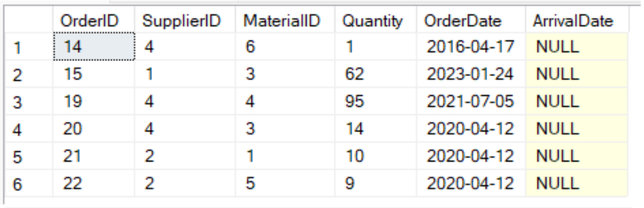
After



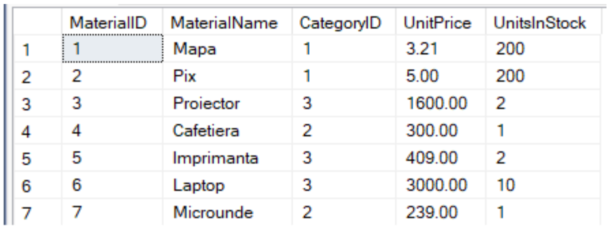
	ScheduleID	Quantity	UnitPrice
1	2	69	3.21
2	2	71	3000.00
3	2	9	8.00

11.5 Reducerea cantitatii de material comandate daca in stoc se afla deja un nr. destul de mare si daca nu a ajuns inca

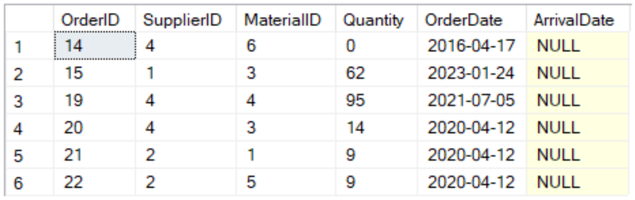
```
GO
DECLARE @Percentage REAL = 0.1
DECLARE @UnitsInStockThreshold INT = 5

UPDATE O
SET O.Quantity = O.Quantity - O.Quantity * @Percentage
FROM Orders O
INNER JOIN Materials M
ON O.MaterialID = M.MaterialID
WHERE M.UnitsInStock >= @UnitsInStockThreshold AND O.ArrivalDate IS NULL

SELECT * FROM Materials
SELECT * FROM Orders WHERE ArrivalDate IS NULL
```



	OrderID	SupplierID	MaterialID	Quantity	OrderDate	ArrivalDate
1	14	4	6	1	2016-04-17	NULL
2	15	1	3	62	2023-01-24	NULL
3	19	4	4	95	2021-07-05	NULL
4	20	4	3	14	2020-04-12	NULL
5	21	2	1	10	2020-04-12	NULL
6	22	2	5	9	2020-04-12	NULL



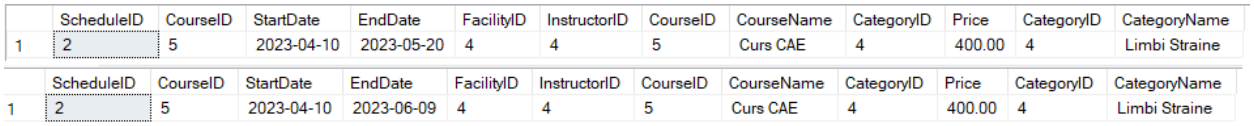
	MaterialID	MaterialName	CategoryID	UnitPrice	UnitsInStock
1	1	Mapa	1	3.21	200
2	2	Pix	1	5.00	200
3	3	Proiector	3	1600.00	2
4	4	Cafetiera	2	300.00	1
5	5	Imprimanta	3	409.00	2
6	6	Laptop	3	3000.00	10
7	7	Microunde	2	239.00	1



	OrderID	SupplierID	MaterialID	Quantity	OrderDate	ArrivalDate
1	14	4	6	0	2016-04-17	NULL
2	15	1	3	62	2023-01-24	NULL
3	19	4	4	95	2021-07-05	NULL
4	20	4	3	14	2020-04-12	NULL
5	21	2	1	9	2020-04-12	NULL
6	22	2	5	9	2020-04-12	NULL

11.6 Lungeste cursurile dintr-o anumita categorie

```
GO
DECLARE @categorie VARCHAR(50) = 'Limbi Straine'
DECLARE @prelungireInZile INT = 20

UPDATE CS
SET CS.EndDate = DATEADD(DAY, @prelungireInZile ,CS.EndDate)
FROM CourseSchedule CS
INNER JOIN Courses C
ON C.CourseID = CS.CourseID
INNER JOIN CourseCategories CC
ON CC.CategoryID = C.CategoryID
WHERE CC.CategoryName = @categorie
```

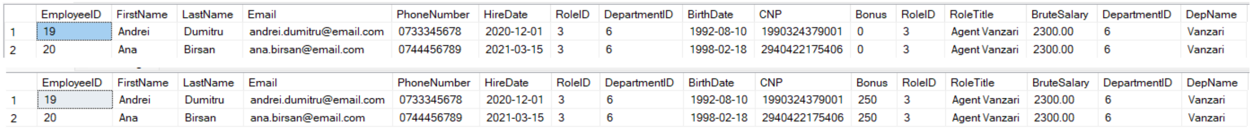


	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID	CourseID	CourseName	CategoryID	Price	CategoryID	CategoryName
1	2	5	2023-04-10	2023-05-20	4	4	5	Curs CAE	4	400.00	4	Limbi Straine

	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID	CourseID	CourseName	CategoryID	Price	CategoryID	CategoryName
1	2	5	2023-04-10	2023-06-09	4	4	5	Curs CAE	4	400.00	4	Limbi Straine

11.7 Creste bonusul angajatilor dintr-un anumit departament cu o anumita functie

```
GO
DECLARE @departmentName VARCHAR(50) = 'Vanzari'
DECLARE @roleTitle VARCHAR(50) = 'Agent Vanzari'
DECLARE @amount INT = 250

UPDATE E
SET E.Bonus = E.Bonus + @amount
FROM Employees E
INNER JOIN EmployeeRoles ER
ON E.RoleID = ER.RoleID
INNER JOIN Departments D
ON E.DepartmentID = D.DepartmentID
WHERE D.DepName = @departmentName AND ER.RoleTitle = @roleTitle
```

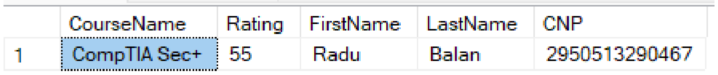


	EmployeeID	FirstName	LastName	Email	PhoneNumber	HireDate	RoleID	DepartmentID	BirthDate	CNP	Bonus	RoleID	RoleTitle	BruteSalary	DepartmentID	DepName
1	19	Andrei	Dumitru	andrei.dumitru@email.com	0733345678	2020-12-01	3	6	1992-08-10	1990324379001	0	3	Agent Vanzari	2300.00	6	Vanzari
2	20	Ana	Birsan	ana.birsan@email.com	0744456789	2021-03-15	3	6	1998-02-18	2940422175406	0	3	Agent Vanzari	2300.00	6	Vanzari

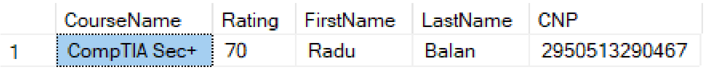
	EmployeeID	FirstName	LastName	Email	PhoneNumber	HireDate	RoleID	DepartmentID	BirthDate	CNP	Bonus	RoleID	RoleTitle	BruteSalary	DepartmentID	DepName
1	19	Andrei	Dumitru	andrei.dumitru@email.com	0733345678	2020-12-01	3	6	1992-08-10	1990324379001	250	3	Agent Vanzari	2300.00	6	Vanzari
2	20	Ana	Birsan	ana.birsan@email.com	0744456789	2021-03-15	3	6	1998-02-18	2940422175406	250	3	Agent Vanzari	2300.00	6	Vanzari

11.8 Schimba rating-ul unui curs dupa nume oferit de un anumit student

```
GO
DECLARE @newRating INT = 70
DECLARE @courseName VARCHAR(50) = 'CompTIA Sec+'
DECLARE @CNPStudent VARCHAR(13) = '2950513290467'

UPDATE CE
SET CE.Rating = @newRating
FROM CourseEvaluations CE
INNER JOIN Students S
ON CE.StudentID = S.StudentID
INNER JOIN CourseSchedule CS
ON CE.ScheduleID = CS.ScheduleID
INNER JOIN Courses C
ON C.CourseID = CS.ScheduleID
WHERE CNP = @CNPStudent AND C.CourseName = @courseName
```



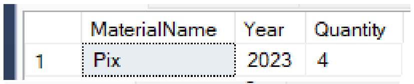
	CourseName	Rating	FirstName	LastName	CNP
1	CompTIA Sec+	55	Radu	Balan	2950513290467



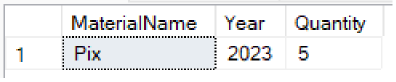
	CourseName	Rating	FirstName	LastName	CNP
1	CompTIA Sec+	70	Radu	Balan	2950513290467

11.9 Creste cantitatea dintr-un material anume folosit pentru un curs anume planificat intr-un an anume

```
GO
DECLARE @matName VARCHAR(50) = 'Pix'
DECLARE @year INT = 2023
DECLARE @percentage REAL = 0.1

UPDATE CM
SET CM.Quantity = CM.Quantity + CEILING(CM.Quantity * @percentage)
FROM CourseMaterials CM
INNER JOIN Materials M
ON CM.MaterialID = M.MaterialID
INNER JOIN CourseSchedule CS
ON CS.ScheduleID = CM.ScheduleID
WHERE DATEPART(YEAR, CS.StartDate) = @year AND M.MaterialName = @matName
```



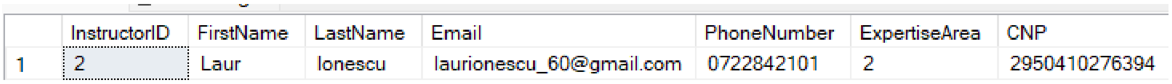
	MaterialName	Year	Quantity
1	Pix	2023	4



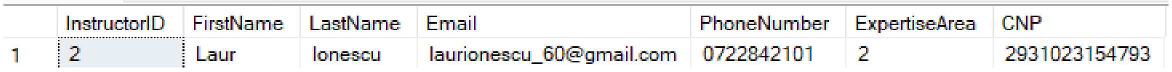
	MaterialName	Year	Quantity
1	Pix	2023	5

11.10 Sa se schimbe CNP-ul unui lector

```
GO
DECLARE @oldCNPInstructor VARCHAR(13) = '2950410276394'
DECLARE @newCNPInstructor VARCHAR(13) = '2931023154793'

UPDATE I
SET I.CNP = @newCNPInstructor
FROM Instructors I
WHERE I.CNP = @oldCNPInstructor
```



	InstructorID	FirstName	LastName	Email	PhoneNumber	ExpertiseArea	CNP
1	2	Laur	Ionescu	laurionescu_60@gmail.com	0722842101	2	2950410276394

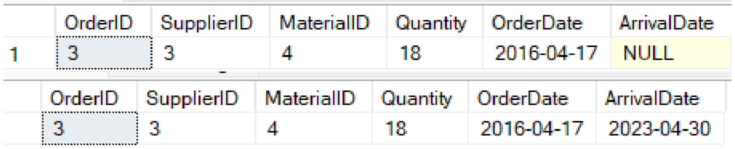


	InstructorID	FirstName	LastName	Email	PhoneNumber	ExpertiseArea	CNP
1	2	Laur	Ionescu	laurionescu_60@gmail.com	0722842101	2	2931023154793

11.11 Sa se seteze data in care un colet a fost livrat

```
GO
DECLARE @OrderID INT = 3

UPDATE O
SET O.ArrivalDate = GETDATE()
FROM Orders O
```

```
WHERE OrderID = @OrderID
```

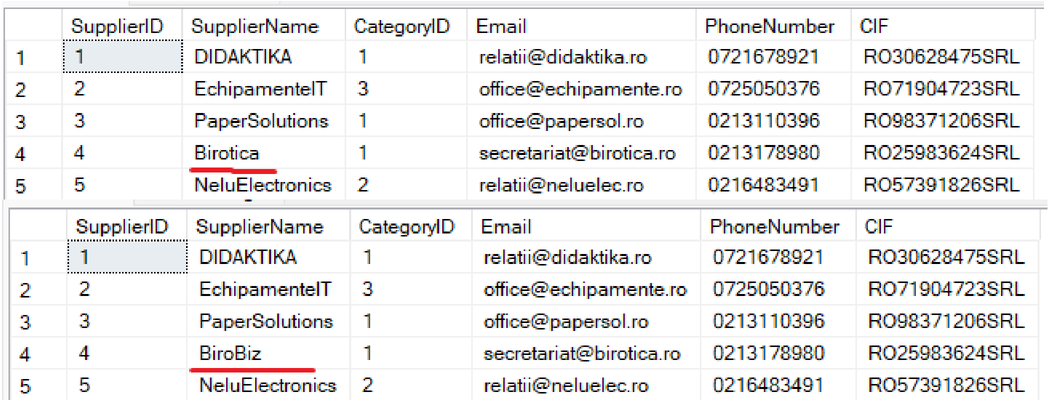


	OrderID	SupplierID	MaterialID	Quantity	OrderDate	ArrivalDate
1	3	3	4	18	2016-04-17	NULL

	OrderID	SupplierID	MaterialID	Quantity	OrderDate	ArrivalDate
	3	3	4	18	2016-04-17	2023-04-30

11.12 Sa se schimbe numele unui furnizor

```
GO
DECLARE @oldSupName VARCHAR(50) = 'Birotica'
DECLARE @newSupName VARCHAR(50) = 'BiroBiz'

UPDATE S
SET S.SupplierName = @newSupName
FROM Suppliers S
WHERE S.SupplierName = @oldSupName
```

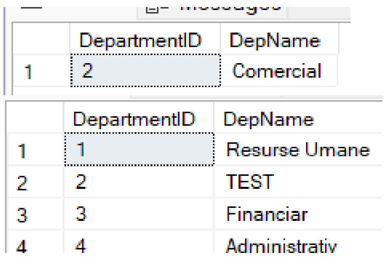


	SupplierID	SupplierName	CategoryID	Email	PhoneNumber	CIF
1	1	DIDAKTIKA	1	relatii@didaktika.ro	0721678921	RO30628475SRL
2	2	EchipamenteIT	3	office@echipamente.ro	0725050376	RO71904723SRL
3	3	PaperSolutions	1	office@papersol.ro	0213110396	RO98371206SRL
4	4	Birotica	1	secretariat@birotica.ro	0213178980	RO25983624SRL
5	5	NeluElectronics	2	relatii@neluelec.ro	0216483491	RO57391826SRL

	SupplierID	SupplierName	CategoryID	Email	PhoneNumber	CIF
1	1	DIDAKTIKA	1	relatii@didaktika.ro	0721678921	RO30628475SRL
2	2	EchipamenteIT	3	office@echipamente.ro	0725050376	RO71904723SRL
3	3	PaperSolutions	1	office@papersol.ro	0213110396	RO98371206SRL
4	4	BiroBiz	1	secretariat@birotica.ro	0213178980	RO25983624SRL
5	5	NeluElectronics	2	relatii@neluelec.ro	0216483491	RO57391826SRL

11.13 Sa se schimbe numele departamentelor care incep cu o litera in altul

```
GO
DECLARE @letter AS VARCHAR(1) = 'C'
DECLARE @newName AS VARCHAR(50) = 'TEST'

UPDATE D
SET D.DepName = @newName
FROM Departments D
WHERE D.DepName LIKE @letter + '%'
```

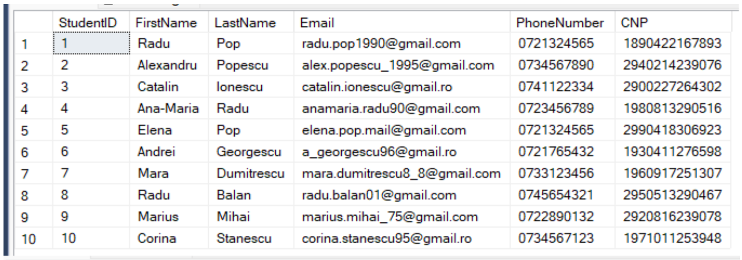


	DepartmentID	DepName
1	2	Comercial

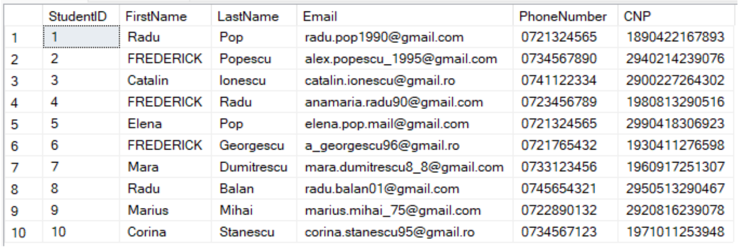
	DepartmentID	DepName
1	1	Resurse Umane
2	2	TEST
3	3	Financiar
4	4	Administrativ

11.14 Sa se inlocuiasca toate numele de familie studentilor care se termina intr-o litera anume cu acelasi nume

```
GO
DECLARE @letter VARCHAR(1) = 'A'
DECLARE @newName VARCHAR(50) = 'FREDERICK'

UPDATE S
SET S.FirstName = @newName
FROM Students S
WHERE S.FirstName LIKE @letter + '%'
```



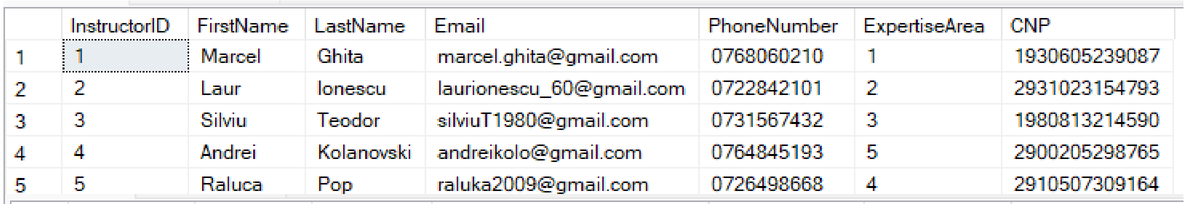
	StudentID	FirstName	LastName	Email	PhoneNumber	CNP
1	1	Radu	Pop	radu.pop1990@gmail.com	0721324565	1890422167893
2	2	Alexandru	Popescu	alex.popescu_1995@gmail.com	0734567890	2940214239076
3	3	Catalin	Ionescu	catalin.ionescu@gmail.ro	0741122334	2900227264302
4	4	Ana-Maria	Radu	anamaria.radu90@gmail.com	0723456789	1980813290516
5	5	Elena	Pop	elena.pop.mail@gmail.com	0721324565	2990418306923
6	6	Andrei	Georgescu	a_georgescu96@gmail.ro	0721765432	1930411276598
7	7	Mara	Dumitrescu	mara.dumitrescu8_8@gmail.com	0733123456	1960917251307
8	8	Radu	Balan	radu.balan01@gmail.com	0745654321	2950513290467
9	9	Marius	Mihai	marius.mihai_75@gmail.com	0722890132	2920816239078
10	10	Corina	Stanescu	corina.stanescu95@gmail.ro	0734567123	1971011253948



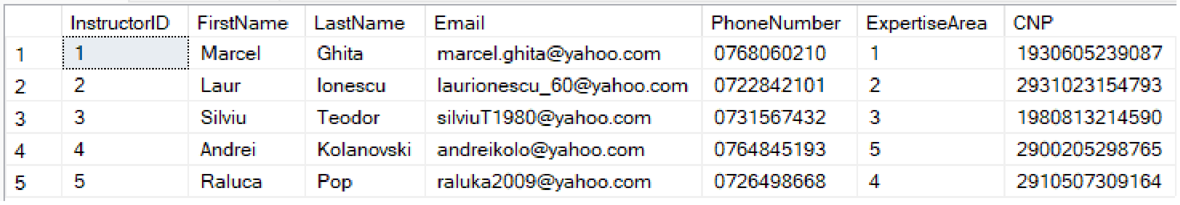
	StudentID	FirstName	LastName	Email	PhoneNumber	CNP
1	1	Radu	Pop	radu.pop1990@gmail.com	0721324565	1890422167893
2	2	FREDERICK	Popescu	alex.popescu_1995@gmail.com	0734567890	2940214239076
3	3	Catalin	Ionescu	catalin.ionescu@gmail.ro	0741122334	2900227264302
4	4	FREDERICK	Radu	anamaria.radu90@gmail.com	0723456789	1980813290516
5	5	Elena	Pop	elena.pop.mail@gmail.com	0721324565	2990418306923
6	6	FREDERICK	Georgescu	a_georgescu96@gmail.ro	0721765432	1930411276598
7	7	Mara	Dumitrescu	mara.dumitrescu8_8@gmail.com	0733123456	1960917251307
8	8	Radu	Balan	radu.balan01@gmail.com	0745654321	2950513290467
9	9	Marius	Mihai	marius.mihai_75@gmail.com	0722890132	2920816239078
10	10	Corina	Stanescu	corina.stanescu95@gmail.ro	0734567123	1971011253948

11.15 Sa se schimbe toate gmail-urile in mail-uri de yahoo ale lectorilor

```
UPDATE I
SET I.Email = REPLACE(I.Email,'gmail','yahoo')
FROM Instructors I
```



	InstructorID	FirstName	LastName	Email	PhoneNumber	ExpertiseArea	CNP
1	1	Marcel	Ghita	marcel.ghita@gmail.com	0768060210	1	1930605239087
2	2	Laur	Ionescu	laurionescu_60@gmail.com	0722842101	2	2931023154793
3	3	Silviu	Teodor	silviuT1980@gmail.com	0731567432	3	1980813214590
4	4	Andrei	Kolanovski	andreikolo@gmail.com	0764845193	5	2900205298765
5	5	Raluca	Pop	raluka2009@gmail.com	0726498668	4	2910507309164



	InstructorID	FirstName	LastName	Email	PhoneNumber	ExpertiseArea	CNP
1	1	Marcel	Ghita	marcel.ghita@yahoo.com	0768060210	1	1930605239087
2	2	Laur	Ionescu	laurionescu_60@yahoo.com	0722842101	2	2931023154793
3	3	Silviu	Teodor	silviuT1980@yahoo.com	0731567432	3	1980813214590
4	4	Andrei	Kolanovski	andreikolo@yahoo.com	0764845193	5	2900205298765
5	5	Raluca	Pop	raluka2009@yahoo.com	0726498668	4	2910507309164

11.16 Sa se schimbe data in care fiecare comanda a ajuns astfel incat durata de livrare a tuturor comenzilor sa nu fie mai mare de 3 zile

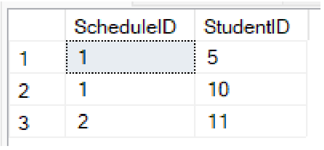
```
UPDATE O
SET O.ArrivalDate = DATEADD(DAY, 3, O.OrderDate)
FROM Orders O
WHERE DATEDIFF(DAY, OrderDate, ArrivalDate) > 3

-- test
INSERT INTO Orders(SupplierID, MaterialID, Quantity, OrderDate,
ArrivalDate)
VALUES
(31, 49, 10, '2022-05-03', '2022-06-03'),
(31, 49, 10, '2023-04-15', '2023-07-12'),
(31, 49, 10, '2021-12-12', '2023-04-04')


SELECT *
FROM Orders O
WHERE DATEDIFF(DAY, OrderDate, ArrivalDate) > 3
```

12 DELETE

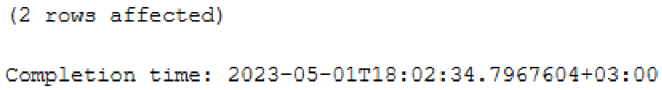
```
------------------------------------------------------
----------------- DELETE QUERIES ---------------------
------------------------------------------------------
```

12.1 Sa se stearga prezenta de pe o planificare

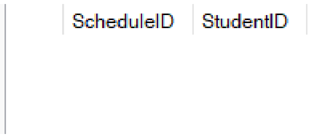
```
GO
DECLARE @ScheduleID INT = 1

DELETE CA
FROM CourseAttendance CA
WHERE CA.ScheduleID = @ScheduleID
```



	ScheduleID	StudentID
1	1	5
2	1	10
3	2	11



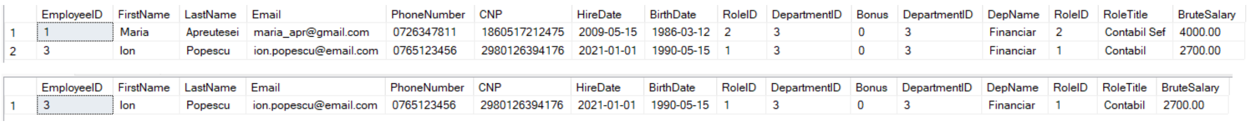
```
(2 rows affected)

Completion time: 2023-05-01T18:02:34.7967604+03:00
```



	ScheduleID	StudentID

12.2 Sa se stearga angajatii din departamentul X cu salariu mai mare decat Y

```
GO
DECLARE @depName VARCHAR(50) = 'Financiar'
DECLARE @bruteSalary DECIMAL(10,2) = 2800.0

DELETE E
FROM Employees E
INNER JOIN Departments D
ON E.DepartmentID = D.DepartmentID
INNER JOIN EmployeeRoles ER
ON E.RoleID = ER.RoleID
WHERE D.DepName = @depName AND ER.BruteSalary > @bruteSalary
```



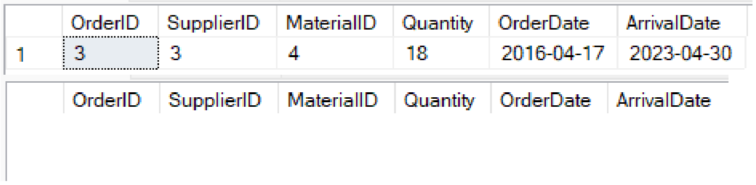
	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus	DepartmentID	DepName	RoleID	RoleTitle	BruteSalary
1	1	Maria	Apreutesei	maria_apr@gmail.com	0726347811	1860517212475	2009-05-15	1986-03-12	2	3	0	3	Financiar	2	Contabil Sef	4000.00
2	3	Ion	Popescu	ion.popescu@email.com	0765123456	2980126394176	2021-01-01	1990-05-15	1	3	0	3	Financiar	1	Contabil	2700.00

	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus	DepartmentID	DepName	RoleID	RoleTitle	BruteSalary
1	3	Ion	Popescu	ion.popescu@email.com	0765123456	2980126394176	2021-01-01	1990-05-15	1	3	0	3	Financiar	1	Contabil	2700.00

12.3 Sa se stearga comenzile care nu au ajuns in maxim X zile

```
GO
DECLARE @nrOfDays INT = 20

DELETE O
FROM Orders O
```

```
WHERE DATEDIFF(DAY, O.OrderDate, O.ArrivalDate) > @nrOfDays
```

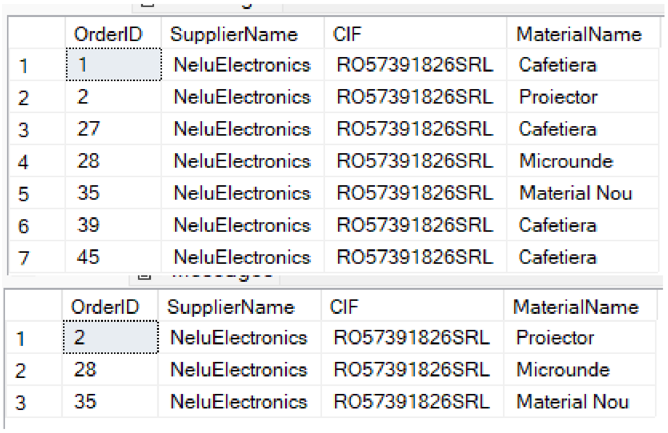


	OrderID	SupplierID	MaterialID	Quantity	OrderDate	ArrivalDate
1	3	3	4	18	2016-04-17	2023-04-30

	OrderID	SupplierID	MaterialID	Quantity	OrderDate	ArrivalDate

12.4 Sa se stearga comenzile de la furnizorul cu CIF-ul X pe materialul cu numele Y

```
GO
DECLARE @CIF VARCHAR(13) = 'RO57391826SRL'
DECLARE @MaterialName VARCHAR(50) = 'Cafetiera'

DELETE O
FROM Orders O
INNER JOIN Suppliers S
ON O.SupplierID = S.SupplierID
INNER JOIN Materials M
ON O.MaterialID = M.MaterialID
WHERE S.CIF = @CIF AND M.MaterialName = 'Cafetiera'
```

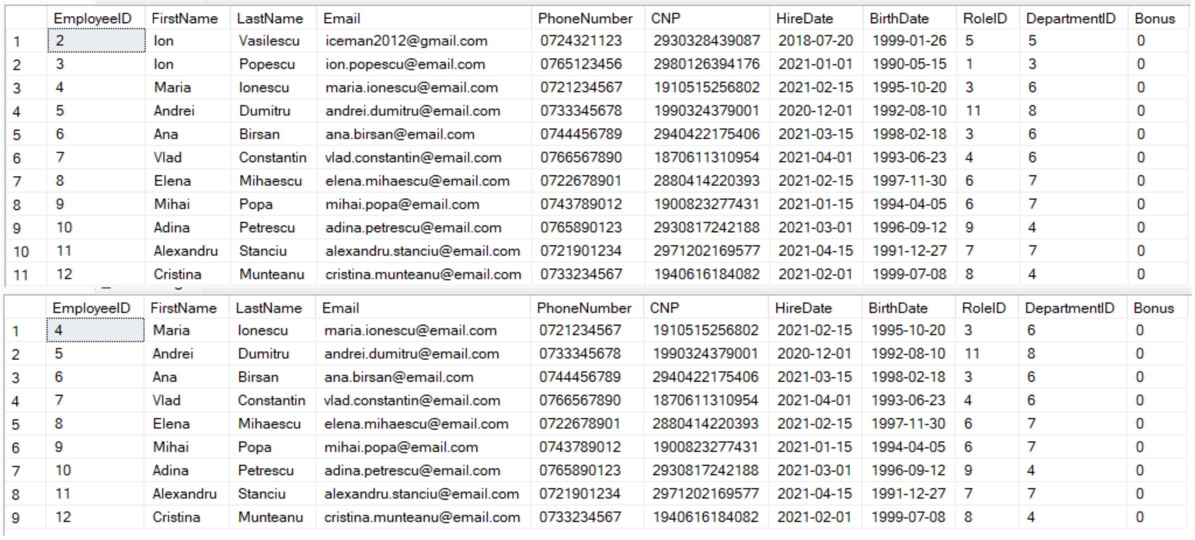


	OrderID	SupplierName	CIF	MaterialName
1	1	NeluElectronics	RO57391826SRL	Cafetiera
2	2	NeluElectronics	RO57391826SRL	Proiector
3	27	NeluElectronics	RO57391826SRL	Cafetiera
4	28	NeluElectronics	RO57391826SRL	Microunde
5	35	NeluElectronics	RO57391826SRL	Material Nou
6	39	NeluElectronics	RO57391826SRL	Cafetiera
7	45	NeluElectronics	RO57391826SRL	Cafetiera

	OrderID	SupplierName	CIF	MaterialName
1	2	NeluElectronics	RO57391826SRL	Proiector
2	28	NeluElectronics	RO57391826SRL	Microunde
3	35	NeluElectronics	RO57391826SRL	Material Nou

12.5 Sa se stearga angajatii al caror nume incepe cu o litera anume

```
GO
DECLARE @letter VARCHAR(1) = 'I'

DELETE E
FROM Employees E
WHERE E.FirstName LIKE @letter + '%'
```

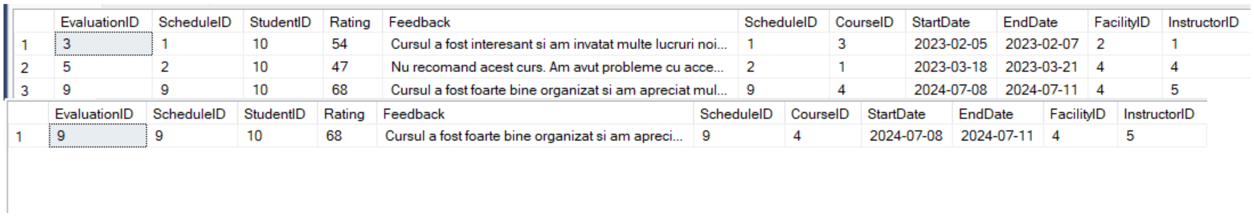


	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus
1	2	Ion	Vasilescu	iceman2012@gmail.com	0724321123	2930328439087	2018-07-20	1999-01-26	5	5	0
2	3	Ion	Popescu	ion.popescu@email.com	0765123456	2980126394176	2021-01-01	1990-05-15	1	3	0
3	4	Maria	Ionescu	maria.ionescu@email.com	0721234567	1910515256802	2021-02-15	1995-10-20	3	6	0
4	5	Andrei	Dumitru	andrei.dumitru@email.com	0733345678	1990324379001	2020-12-01	1992-08-10	11	8	0
5	6	Ana	Birsan	ana.birsan@email.com	0744456789	2940422175406	2021-03-15	1998-02-18	3	6	0
6	7	Vlad	Constantin	vlad.constantin@email.com	0766567890	1870611310954	2021-04-01	1993-06-23	4	6	0
7	8	Elena	Mihaescu	elena.mihaescu@email.com	0722678901	2880414220393	2021-02-15	1997-11-30	6	7	0
8	9	Mihai	Popa	mihai.popa@email.com	0743789012	1900823277431	2021-01-15	1994-04-05	6	7	0
9	10	Adina	Petrescu	adina.petrescu@email.com	0765890123	2930817242188	2021-03-01	1996-09-12	9	4	0
10	11	Alexandru	Stanciu	alexandru.stanciu@email.com	0721901234	2971202169577	2021-04-15	1991-12-27	7	7	0
11	12	Cristina	Munteanu	cristina.munteanu@email.com	0733234567	1940616184082	2021-02-01	1999-07-08	8	4	0

	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus
1	4	Maria	Ionescu	maria.ionescu@email.com	0721234567	1910515256802	2021-02-15	1995-10-20	3	6	0
2	5	Andrei	Dumitru	andrei.dumitru@email.com	0733345678	1990324379001	2020-12-01	1992-08-10	11	8	0
3	6	Ana	Birsan	ana.birsan@email.com	0744456789	2940422175406	2021-03-15	1998-02-18	3	6	0
4	7	Vlad	Constantin	vlad.constantin@email.com	0766567890	1870611310954	2021-04-01	1993-06-23	4	6	0
5	8	Elena	Mihaescu	elena.mihaescu@email.com	0722678901	2880414220393	2021-02-15	1997-11-30	6	7	0
6	9	Mihai	Popa	mihai.popa@email.com	0743789012	1900823277431	2021-01-15	1994-04-05	6	7	0
7	10	Adina	Petrescu	adina.petrescu@email.com	0765890123	2930817242188	2021-03-01	1996-09-12	9	4	0
8	11	Alexandru	Stanciu	alexandru.stanciu@email.com	0721901234	2971202169577	2021-04-15	1991-12-27	7	7	0
9	12	Cristina	Munteanu	cristina.munteanu@email.com	0733234567	1940616184082	2021-02-01	1999-07-08	8	4	0

12.6 Sa se stearga recenziile studentului X din anul Y

```
GO
DECLARE @StudentID INT = 10
DECLARE @Year INT = 2023

DELETE CE
FROM CourseEvaluations CE
INNER JOIN CourseSchedule CS
ON CE.ScheduleID = CS.ScheduleID
INNER JOIN Students S
ON CE.StudentID = S.StudentID
WHERE CE.StudentID = @StudentID AND DATEPART(YEAR, CS.StartDate) = @Year
```

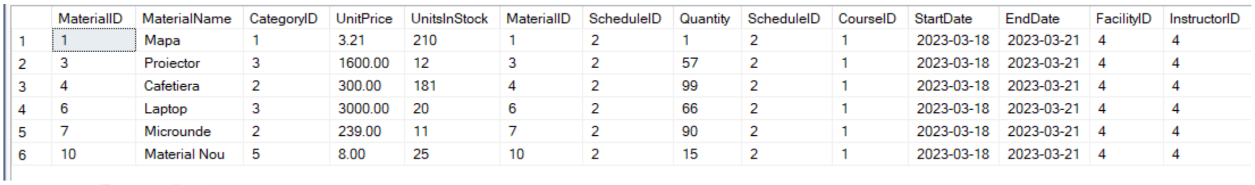


	EvaluationID	ScheduleID	StudentID	Rating	Feedback	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID
1	3	1	10	54	Cursul a fost interesant si am invatat multe lucruri noi...	1	3	2023-02-05	2023-02-07	2	1
2	5	2	10	47	Nu recomand acest curs. Am avut probleme cu acce...	2	1	2023-03-18	2023-03-21	4	4
3	9	9	10	68	Cursul a fost foarte bine organizat si am apreciat mul...	9	4	2024-07-08	2024-07-11	4	5

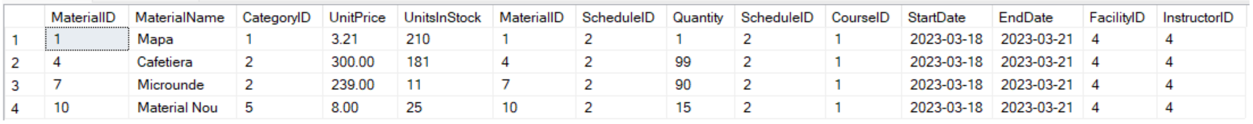
	EvaluationID	ScheduleID	StudentID	Rating	Feedback	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID
1	9	9	10	68	Cursul a fost foarte bine organizat si am apreci...	9	4	2024-07-08	2024-07-11	4	5

12.7 Sa se renunte la materialele mai scumpe de X pentru planificarile din luna Y anul asta

```
GO
DECLARE @Month INT = 3
DECLARE @Price INT = 500

DELETE CM
FROM Materials M
INNER JOIN CourseMaterials CM
ON M.MaterialID = CM.MaterialID
INNER JOIN CourseSchedule CS
ON CM.ScheduleID = CS.ScheduleID
WHERE
DATEPART(YEAR, GETDATE()) = DATEPART(YEAR, CS.StartDate) AND
DATEPART(MONTH, CS.StartDate) = @Month AND
M.UnitPrice > @Price
```



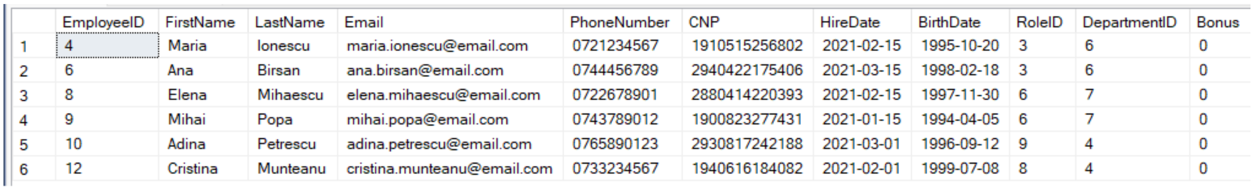
	MaterialID	MaterialName	CategoryID	UnitPrice	UnitsInStock	MaterialID	ScheduleID	Quantity	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID
1	1	Mapa	1	3.21	210	1	2	1	2	1	2023-03-18	2023-03-21	4	4
2	3	Proiector	3	1600.00	12	3	2	57	2	1	2023-03-18	2023-03-21	4	4
3	4	Cafetiera	2	300.00	181	4	2	99	2	1	2023-03-18	2023-03-21	4	4
4	6	Laptop	3	3000.00	20	6	2	66	2	1	2023-03-18	2023-03-21	4	4
5	7	Microunde	2	239.00	11	7	2	90	2	1	2023-03-18	2023-03-21	4	4
6	10	Material Nou	5	8.00	25	10	2	15	2	1	2023-03-18	2023-03-21	4	4



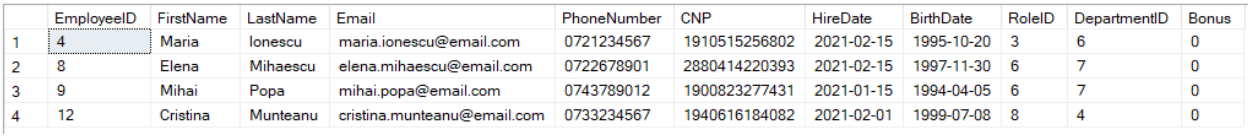
	MaterialID	MaterialName	CategoryID	UnitPrice	UnitsInStock	MaterialID	ScheduleID	Quantity	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID
1	1	Mapa	1	3.21	210	1	2	1	2	1	2023-03-18	2023-03-21	4	4
2	4	Cafetiera	2	300.00	181	4	2	99	2	1	2023-03-18	2023-03-21	4	4
3	7	Microunde	2	239.00	11	7	2	90	2	1	2023-03-18	2023-03-21	4	4
4	10	Material Nou	5	8.00	25	10	2	15	2	1	2023-03-18	2023-03-21	4	4

12.8 Sa se stearga oamenii angajati in luna X a anului Y

```
GO
DECLARE @Year INT = 2021
DECLARE @Month INT = 3

DELETE E
FROM Employees E
WHERE
DATEPART(YEAR, E.HireDate) = @Year AND
DATEPART(MONTH, E.HireDate) = @Month
```



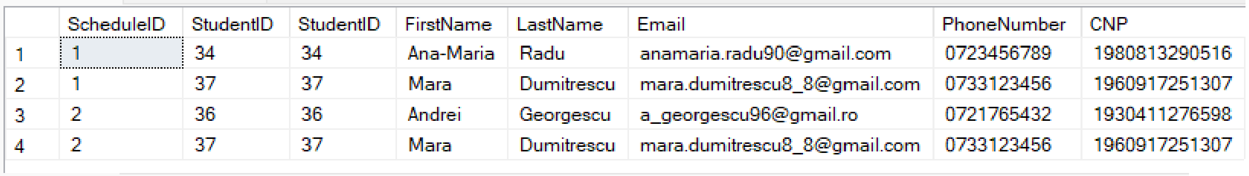
	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus
1	4	Maria	Ionescu	maria.ionescu@email.com	0721234567	1910515256802	2021-02-15	1995-10-20	3	6	0
2	6	Ana	Birsan	ana.birsan@email.com	0744456789	2940422175406	2021-03-15	1998-02-18	3	6	0
3	8	Elena	Mihaescu	elena.mihaescu@email.com	0722678901	2880414220393	2021-02-15	1997-11-30	6	7	0
4	9	Mihai	Popa	mihai.popa@email.com	0743789012	1900823277431	2021-01-15	1994-04-05	6	7	0
5	10	Adina	Petrescu	adina.petrescu@email.com	0765890123	2930817242188	2021-03-01	1996-09-12	9	4	0
6	12	Cristina	Munteanu	cristina.munteanu@email.com	0733234567	1940616184082	2021-02-01	1999-07-08	8	4	0



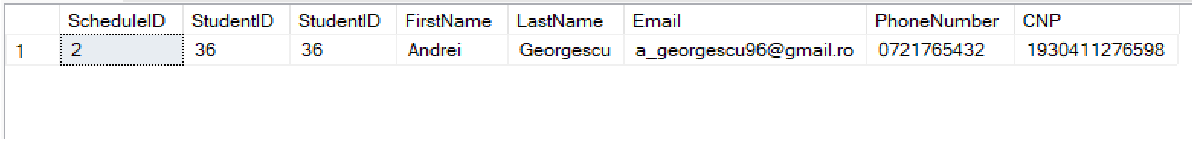
	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus
1	4	Maria	Ionescu	maria.ionescu@email.com	0721234567	1910515256802	2021-02-15	1995-10-20	3	6	0
2	8	Elena	Mihaescu	elena.mihaescu@email.com	0722678901	2880414220393	2021-02-15	1997-11-30	6	7	0
3	9	Mihai	Popa	mihai.popa@email.com	0743789012	1900823277431	2021-01-15	1994-04-05	6	7	0
4	12	Cristina	Munteanu	cristina.munteanu@email.com	0733234567	1940616184082	2021-02-01	1999-07-08	8	4	0

12.9 Sa se stearga prezenta studentilor al caror nume se termina cu A

```
GO
DELETE CA
FROM CourseAttendance CA
INNER JOIN Students S
ON CA.StudentID = S.StudentID
WHERE S.FirstName LIKE '%A'
```



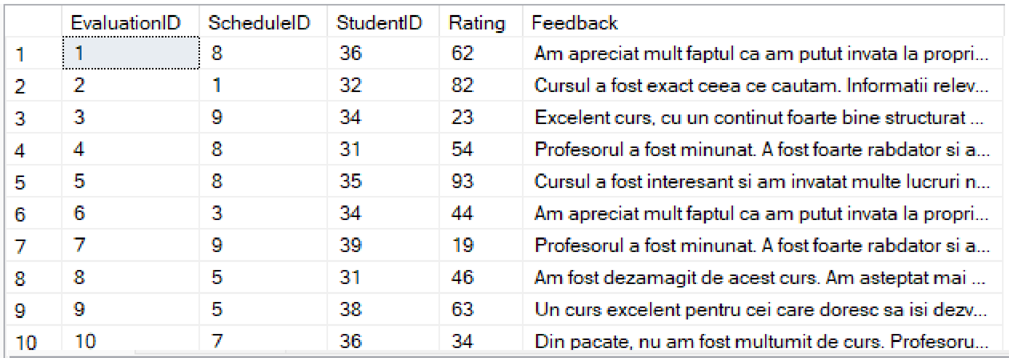
	ScheduleID	StudentID	StudentID	FirstName	LastName	Email	PhoneNumber	CNP
1	1	34	34	Ana-Maria	Radu	anamaria.radu90@gmail.com	0723456789	1980813290516
2	1	37	37	Mara	Dumitrescu	mara.dumitrescu8_8@gmail.com	0733123456	1960917251307
3	2	36	36	Andrei	Georgescu	a_georgescu96@gmail.ro	0721765432	1930411276598
4	2	37	37	Mara	Dumitrescu	mara.dumitrescu8_8@gmail.com	0733123456	1960917251307



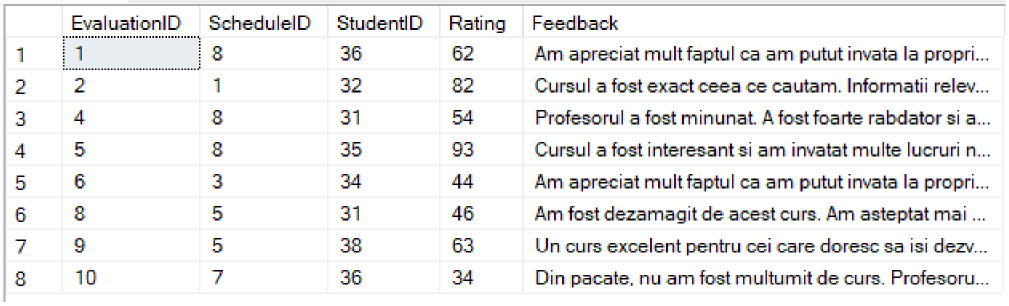
	ScheduleID	StudentID	StudentID	FirstName	LastName	Email	PhoneNumber	CNP
1	2	36	36	Andrei	Georgescu	a_georgescu96@gmail.ro	0721765432	1930411276598

12.10 Sa se stearga recenziile cu un rating mai mic de X

```
GO
DECLARE @Rating INT = 25

DELETE CE
FROM CourseEvaluations CE
WHERE CE.Rating < @Rating
```



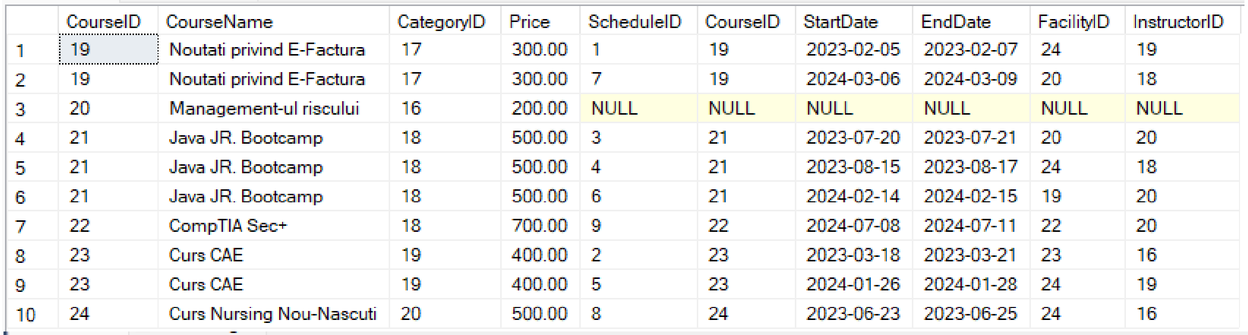
	EvaluationID	ScheduleID	StudentID	Rating	Feedback
1	1	8	36	62	Am apreciat mult faptul ca am putut invata la propri...
2	2	1	32	82	Cursul a fost exact ceea ce cautam. Informatii relev...
3	3	9	34	23	Excelent curs, cu un continut foarte bine structurat ...
4	4	8	31	54	Profesorul a fost minunat. A fost foarte rabdator si a...
5	5	8	35	93	Cursul a fost interesant si am invatat multe lucruri n...
6	6	3	34	44	Am apreciat mult faptul ca am putut invata la propri...
7	7	9	39	19	Profesorul a fost minunat. A fost foarte rabdator si a...
8	8	5	31	46	Am fost dezamagit de acest curs. Am asteptat mai ...
9	9	5	38	63	Un curs excelent pentru cei care doresc sa isi dezv...
10	10	7	36	34	Din pacate, nu am fost multumit de curs. Profesoru...



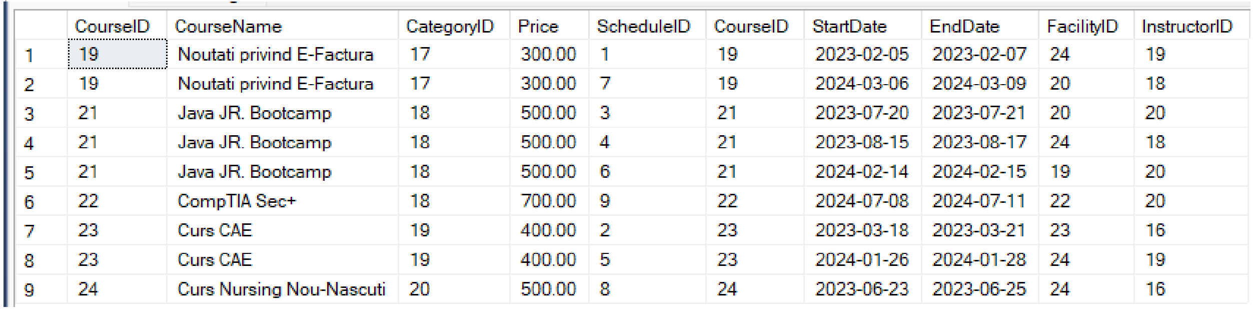
	EvaluationID	ScheduleID	StudentID	Rating	Feedback
1	1	8	36	62	Am apreciat mult faptul ca am putut invata la propri...
2	2	1	32	82	Cursul a fost exact ceea ce cautam. Informatii relev...
3	4	8	31	54	Profesorul a fost minunat. A fost foarte rabdator si a...
4	5	8	35	93	Cursul a fost interesant si am invatat multe lucruri n...
5	6	3	34	44	Am apreciat mult faptul ca am putut invata la propri...
6	8	5	31	46	Am fost dezamagit de acest curs. Am asteptat mai ...
7	9	5	38	63	Un curs excelent pentru cei care doresc sa isi dezv...
8	10	7	36	34	Din pacate, nu am fost multumit de curs. Profesoru...

12.11 Sa se stearga cursurile neplanificate

```
GO
DELETE C
FROM Courses C
LEFT JOIN CourseSchedule CS
ON C.CourseID = CS.CourseID
WHERE CS.ScheduleID IS NULL
```



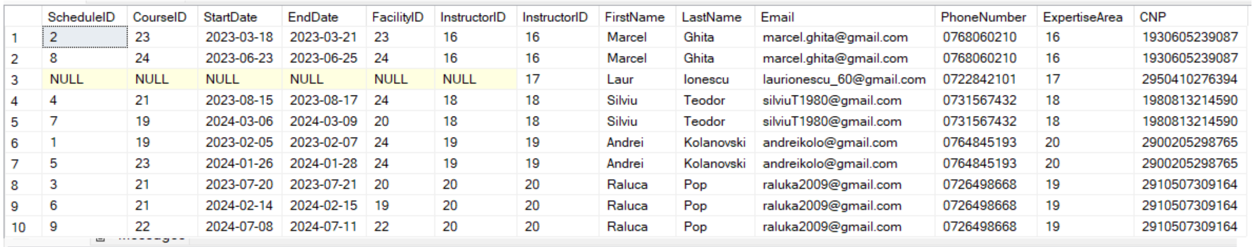
	CourseID	CourseName	CategoryID	Price	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID
1	19	Noutati privind E-Factura	17	300.00	1	19	2023-02-05	2023-02-07	24	19
2	19	Noutati privind E-Factura	17	300.00	7	19	2024-03-06	2024-03-09	20	18
3	20	Management-ul riscului	16	200.00	NULL	NULL	NULL	NULL	NULL	NULL
4	21	Java JR. Bootcamp	18	500.00	3	21	2023-07-20	2023-07-21	20	20
5	21	Java JR. Bootcamp	18	500.00	4	21	2023-08-15	2023-08-17	24	18
6	21	Java JR. Bootcamp	18	500.00	6	21	2024-02-14	2024-02-15	19	20
7	22	CompTIA Sec+	18	700.00	9	22	2024-07-08	2024-07-11	22	20
8	23	Curs CAE	19	400.00	2	23	2023-03-18	2023-03-21	23	16
9	23	Curs CAE	19	400.00	5	23	2024-01-26	2024-01-28	24	19
10	24	Curs Nursing Nou-Nascuti	20	500.00	8	24	2023-06-23	2023-06-25	24	16



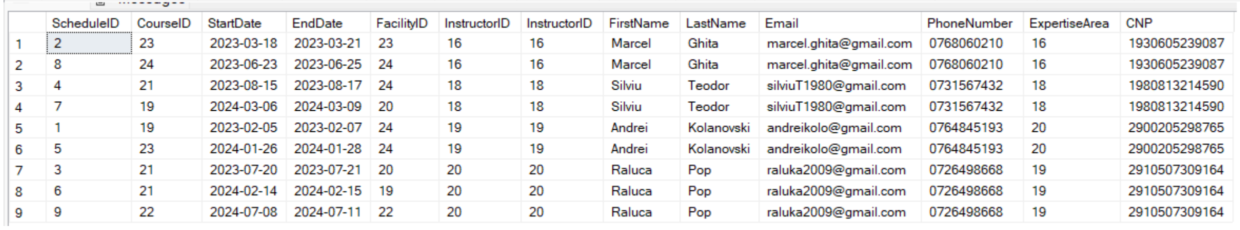
	CourseID	CourseName	CategoryID	Price	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID
1	19	Noutati privind E-Factura	17	300.00	1	19	2023-02-05	2023-02-07	24	19
2	19	Noutati privind E-Factura	17	300.00	7	19	2024-03-06	2024-03-09	20	18
3	21	Java JR. Bootcamp	18	500.00	3	21	2023-07-20	2023-07-21	20	20
4	21	Java JR. Bootcamp	18	500.00	4	21	2023-08-15	2023-08-17	24	18
5	21	Java JR. Bootcamp	18	500.00	6	21	2024-02-14	2024-02-15	19	20
6	22	CompTIA Sec+	18	700.00	9	22	2024-07-08	2024-07-11	22	20
7	23	Curs CAE	19	400.00	2	23	2023-03-18	2023-03-21	23	16
8	23	Curs CAE	19	400.00	5	23	2024-01-26	2024-01-28	24	19
9	24	Curs Nursing Nou-Nascuti	20	500.00	8	24	2023-06-23	2023-06-25	24	16

12.12 Sa se stearga lectorii care nu au planificat niciun curs

```
GO
DELETE I
FROM CourseSchedule CS
RIGHT JOIN Instructors I
ON CS.InstructorID = I.InstructorID
WHERE CS.ScheduleID IS NULL
```



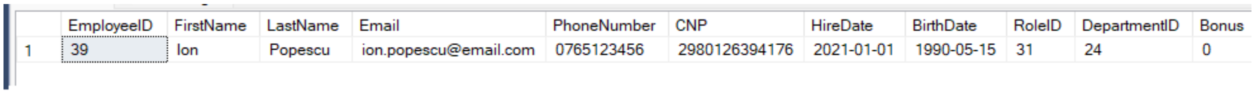
	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID	InstructorID	FirstName	LastName	Email	PhoneNumber	ExpertiseArea	CNP
1	2	23	2023-03-18	2023-03-21	23	16	16	Marcel	Ghita	marcel.ghita@gmail.com	0768060210	16	1930605239087
2	8	24	2023-06-23	2023-06-25	24	16	16	Marcel	Ghita	marcel.ghita@gmail.com	0768060210	16	1930605239087
3	NULL	NULL	NULL	NULL	NULL	NULL	17	Laur	Ionescu	laurionescu_60@gmail.com	0722842101	17	2950410276394
4	4	21	2023-08-15	2023-08-17	24	18	18	Silviu	Teodor	silviuT1980@gmail.com	0731567432	18	1980813214590
5	7	19	2024-03-06	2024-03-09	20	18	18	Silviu	Teodor	silviuT1980@gmail.com	0731567432	18	1980813214590
6	1	19	2023-02-05	2023-02-07	24	19	19	Andrei	Kolanovski	andreikolo@gmail.com	0764845193	20	2900205298765
7	5	23	2024-01-26	2024-01-28	24	19	19	Andrei	Kolanovski	andreikolo@gmail.com	0764845193	20	2900205298765
8	3	21	2023-07-20	2023-07-21	20	20	20	Raluca	Pop	raluka2009@gmail.com	0726498668	19	2910507309164
9	6	21	2024-02-14	2024-02-15	19	20	20	Raluca	Pop	raluka2009@gmail.com	0726498668	19	2910507309164
10	9	22	2024-07-08	2024-07-11	22	20	20	Raluca	Pop	raluka2009@gmail.com	0726498668	19	2910507309164



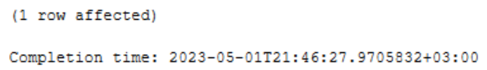
	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID	InstructorID	FirstName	LastName	Email	PhoneNumber	ExpertiseArea	CNP
1	2	23	2023-03-18	2023-03-21	23	16	16	Marcel	Ghita	marcel.ghita@gmail.com	0768060210	16	1930605239087
2	8	24	2023-06-23	2023-06-25	24	16	16	Marcel	Ghita	marcel.ghita@gmail.com	0768060210	16	1930605239087
3	4	21	2023-08-15	2023-08-17	24	18	18	Silviu	Teodor	silviuT1980@gmail.com	0731567432	18	1980813214590
4	7	19	2024-03-06	2024-03-09	20	18	18	Silviu	Teodor	silviuT1980@gmail.com	0731567432	18	1980813214590
5	1	19	2023-02-05	2023-02-07	24	19	19	Andrei	Kolanovski	andreikolo@gmail.com	0764845193	20	2900205298765
6	5	23	2024-01-26	2024-01-28	24	19	19	Andrei	Kolanovski	andreikolo@gmail.com	0764845193	20	2900205298765
7	3	21	2023-07-20	2023-07-21	20	20	20	Raluca	Pop	raluka2009@gmail.com	0726498668	19	2910507309164
8	6	21	2024-02-14	2024-02-15	19	20	20	Raluca	Pop	raluka2009@gmail.com	0726498668	19	2910507309164
9	9	22	2024-07-08	2024-07-11	22	20	20	Raluca	Pop	raluka2009@gmail.com	0726498668	19	2910507309164

12.13 Sa se stearga angajatii care sunt nascuti in anul X

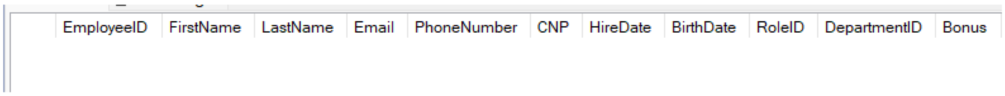
```
GO
DECLARE @Year INT = 1990

DELETE E
FROM Employees E
WHERE DATEPART(YEAR, E.BirthDate) = @Year
```



	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus
1	39	Ion	Popescu	ion.popescu@email.com	0765123456	2980126394176	2021-01-01	1990-05-15	31	24	0



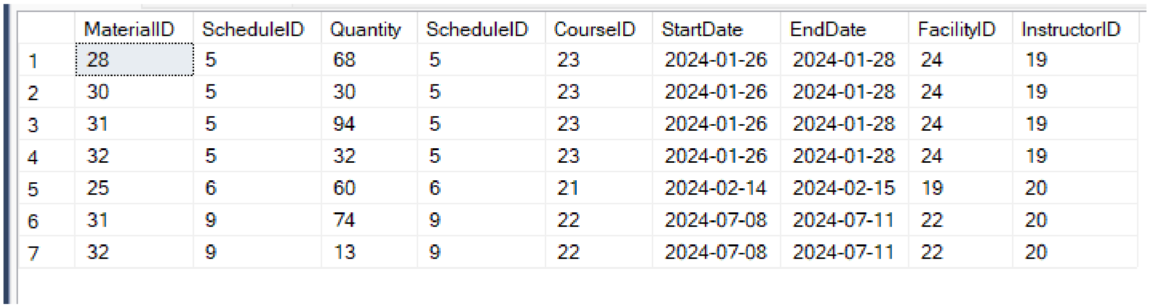
```
(1 row affected)

Completion time: 2023-05-01T21:46:27.9705832+03:00
```



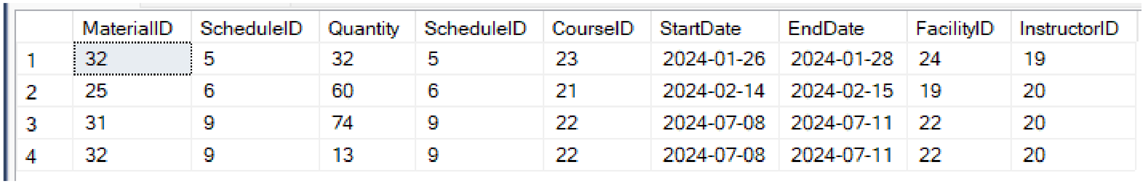
	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentID	Bonus

12.14 Sa se renunte la primele 3 rezultate dintre materialele pentru cursurile de anul viitor

```
GO
DELETE TOP(3) CM
FROM CourseMaterials CM
INNER JOIN CourseSchedule CS
ON CM.ScheduleID = CS.ScheduleID
WHERE DATEDIFF(YEAR, GETDATE(), CS.StartDate) >= 1
```



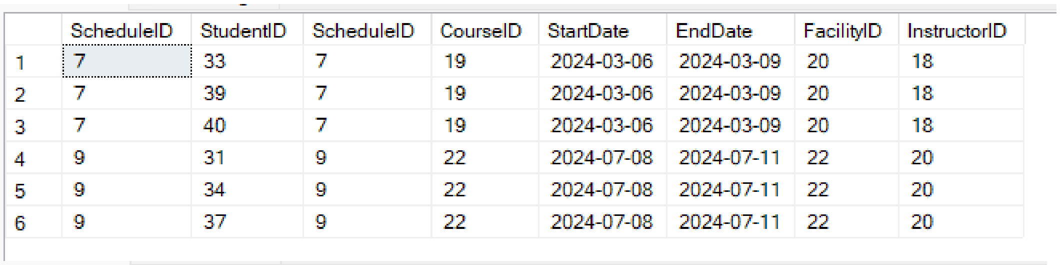
	MaterialID	ScheduleID	Quantity	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID
1	28	5	68	5	23	2024-01-26	2024-01-28	24	19
2	30	5	30	5	23	2024-01-26	2024-01-28	24	19
3	31	5	94	5	23	2024-01-26	2024-01-28	24	19
4	32	5	32	5	23	2024-01-26	2024-01-28	24	19
5	25	6	60	6	21	2024-02-14	2024-02-15	19	20
6	31	9	74	9	22	2024-07-08	2024-07-11	22	20
7	32	9	13	9	22	2024-07-08	2024-07-11	22	20



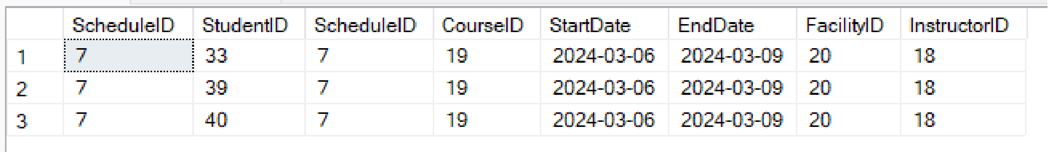
	MaterialID	ScheduleID	Quantity	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID
1	32	5	32	5	23	2024-01-26	2024-01-28	24	19
2	25	6	60	6	21	2024-02-14	2024-02-15	19	20
3	31	9	74	9	22	2024-07-08	2024-07-11	22	20
4	32	9	13	9	22	2024-07-08	2024-07-11	22	20

12.15 Sa se stearga inscrierile pentru planificarile cu mai mult de 350 zile in viitor

```
GO
DELETE CR
FROM CourseRegistrations CR
INNER JOIN CourseSchedule CS
ON CR.ScheduleID = CS.ScheduleID
WHERE DATEDIFF(DAY, GETDATE(), CS.StartDate) > 350
```



	ScheduleID	StudentID	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID
1	7	33	7	19	2024-03-06	2024-03-09	20	18
2	7	39	7	19	2024-03-06	2024-03-09	20	18
3	7	40	7	19	2024-03-06	2024-03-09	20	18
4	9	31	9	22	2024-07-08	2024-07-11	22	20
5	9	34	9	22	2024-07-08	2024-07-11	22	20
6	9	37	9	22	2024-07-08	2024-07-11	22	20



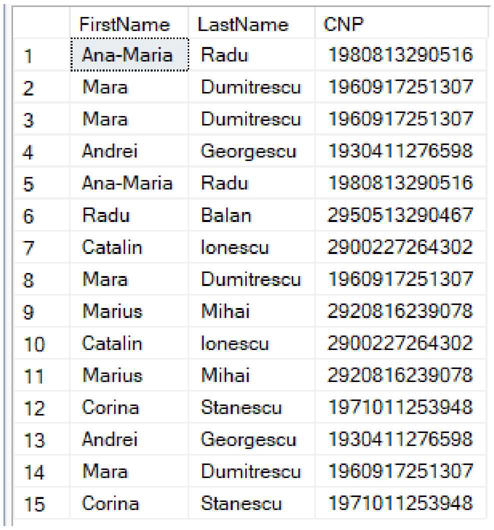
	ScheduleID	StudentID	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID
1	7	33	7	19	2024-03-06	2024-03-09	20	18
2	7	39	7	19	2024-03-06	2024-03-09	20	18
3	7	40	7	19	2024-03-06	2024-03-09	20	18

13 CTE-uri

```
------------------------------------------------------
----------------------- CTEs -------------------------
------------------------------------------------------
```

13.1 Sa se afiseze numele si CNP-ul studentilor care au absenta la cursuri, folosind CTE

```
WITH Absente AS (
      SELECT CR.ScheduleID, CR.StudentID
      FROM CourseRegistrations CR
      EXCEPT
      SELECT CA.ScheduleID, CA.StudentID
      FROM CourseAttendance CA
)
SELECT S.FirstName, S.LastName, S.CNP
FROM Students S
INNER JOIN Absente as A
ON S.StudentID = A.StudentID
```

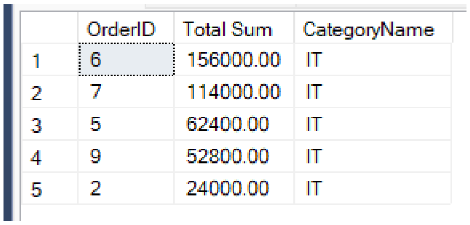


	FirstName	LastName	CNP
1	Ana-Maria	Radu	1980813290516
2	Mara	Dumitrescu	1960917251307
3	Mara	Dumitrescu	1960917251307
4	Andrei	Georgescu	1930411276598
5	Ana-Maria	Radu	1980813290516
6	Radu	Balan	2950513290467
7	Catalin	Ionescu	2900227264302
8	Mara	Dumitrescu	1960917251307
9	Marius	Mihai	2920816239078
10	Catalin	Ionescu	2900227264302
11	Marius	Mihai	2920816239078
12	Corina	Stanescu	1971011253948
13	Andrei	Georgescu	1930411276598
14	Mara	Dumitrescu	1960917251307
15	Corina	Stanescu	1971011253948

13.2 Folosind CTE-uri, sa se afiseze toate comenzile pentru materiale din categoria IT in ordine descrescatoare dupa suma de plata

```
WITH ITMaterials AS (
      SELECT O.OrderID, O.Quantity * M.UnitPrice as 'Total Sum', 
MC.CategoryName
      FROM Orders O
      INNER JOIN Materials M
      ON O.MaterialID = M.MaterialID
      INNER JOIN MaterialCategories MC
      ON M.CategoryID = MC.CategoryID
      WHERE MC.CategoryName = 'IT'
)
SELECT *
FROM ITMaterials
```

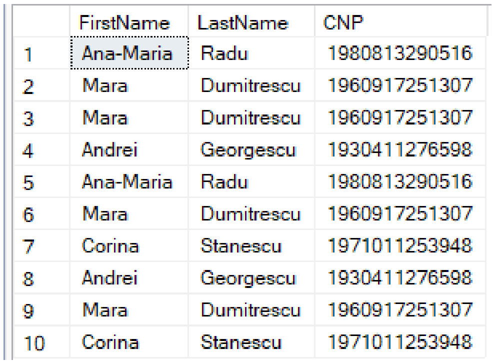
```
ORDER BY [Total Sum] DESC
```



	OrderID	Total Sum	CategoryName
1	6	156000.00	IT
2	7	114000.00	IT
3	5	62400.00	IT
4	9	52800.00	IT
5	2	24000.00	IT

13.3 Sa se afiseze numele si CNP-ul studentilor care au absenta la cursuri, al caror CNP incepe cu 1, folosind CTE

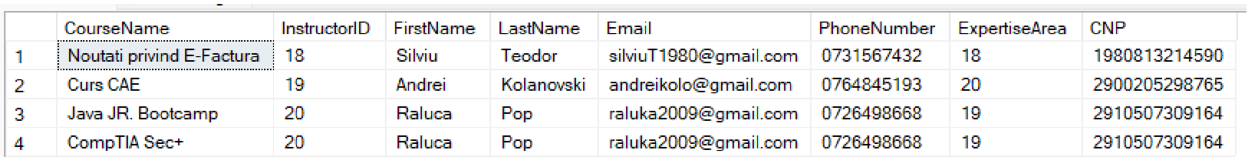
```
WITH Absente AS (
      SELECT CR.ScheduleID, CR.StudentID
      FROM CourseRegistrations CR
      EXCEPT
      SELECT CA.ScheduleID, CA.StudentID
      FROM CourseAttendance CA
),
Absenti AS (
SELECT S.FirstName, S.LastName, S.CNP
FROM Students S
INNER JOIN Absente as A
ON S.StudentID = A.StudentID
)
SELECT * FROM Absenti WHERE CNP LIKE '1%'
```



	FirstName	LastName	CNP
1	Ana-Maria	Radu	1980813290516
2	Mara	Dumitrescu	1960917251307
3	Mara	Dumitrescu	1960917251307
4	Andrei	Georgescu	1930411276598
5	Ana-Maria	Radu	1980813290516
6	Mara	Dumitrescu	1960917251307
7	Corina	Stanescu	1971011253948
8	Andrei	Georgescu	1930411276598
9	Mara	Dumitrescu	1960917251307
10	Corina	Stanescu	1971011253948

13.4 Sa se afiseze, folosind CTE, lectorii care vor veni la cursuri planificate anul viitor, si numele cursului respectiv

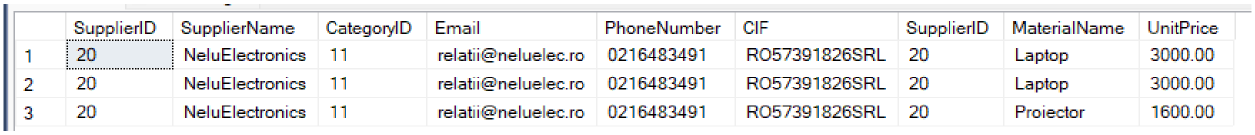
```
WITH NextYearSchedules AS (
      SELECT CourseID, InstructorID
      FROM CourseSchedule CS
      WHERE DATEDIFF(YEAR, GETDATE(), CS.StartDate) = 1
),
CourseNamesNextYear AS (
      SELECT C.CourseName, NYS.InstructorID
      FROM Courses C
      INNER JOIN NextYearSchedules NYS
      ON C.CourseID = NYS.CourseID
)
SELECT CNNY.CourseName, I.*
FROM Instructors I
INNER JOIN CourseNamesNextYear CNNY
ON I.InstructorID = CNNY.InstructorID
```



	CourseName	InstructorID	FirstName	LastName	Email	PhoneNumber	ExpertiseArea	CNP
1	Noutati privind E-Factura	18	Silviu	Teodor	silviuT1980@gmail.com	0731567432	18	1980813214590
2	Curs CAE	19	Andrei	Kolanovski	andreikolo@gmail.com	0764845193	20	2900205298765
3	Java JR. Bootcamp	20	Raluca	Pop	raluka2009@gmail.com	0726498668	19	2910507309164
4	CompTIA Sec+	20	Raluca	Pop	raluka2009@gmail.com	0726498668	19	2910507309164

13.5 Folosind CTE-uri, sa se afiseze furnizorii de la care s-au cumparat cele mai scumpe 3 materiale, acele materiale, si pretul lor

```
WITH Top3MostExpensive AS (
      SELECT TOP(3) O.SupplierID, M.MaterialName, M.UnitPrice
      FROM Orders O
      INNER JOIN Materials M
      ON O.MaterialID = M.MaterialID
      ORDER BY M.UnitPrice DESC
)
SELECT *
FROM Suppliers S
INNER JOIN Top3MostExpensive TME
ON S.SupplierID = TME.SupplierID
```



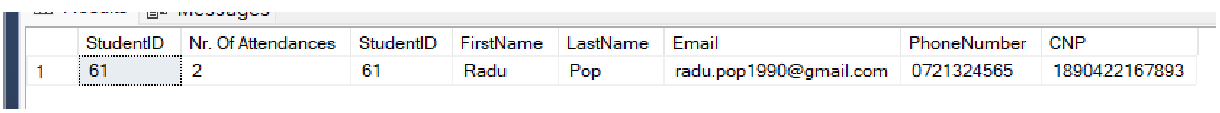
	SupplierID	SupplierName	CategoryID	Email	PhoneNumber	CIF	SupplierID	MaterialName	UnitPrice
1	20	NeluElectronics	11	relatii@neluelec.ro	0216483491	RO57391826SRL	20	Laptop	3000.00
2	20	NeluElectronics	11	relatii@neluelec.ro	0216483491	RO57391826SRL	20	Laptop	3000.00
3	20	NeluElectronics	11	relatii@neluelec.ro	0216483491	RO57391826SRL	20	Proiector	1600.00

13.6 Sa se schimbe prenumele studentului care a participat la cele mai multe planificari de cursuri in "Student Model"

```
WITH AttendanceCount AS (
      SELECT S.StudentID, COUNT(CA.StudentID) as 'Nr. Of Attendances'
      FROM Students S
      LEFT JOIN CourseAttendance CA
      ON S.StudentID = CA.StudentID
      GROUP BY S.StudentID
),
MostPresentStudent AS (
      SELECT TOP(1) *
      FROM AttendanceCount AC
      ORDER BY AC.[Nr. Of Attendances] DESC
)
UPDATE S
SET S.FirstName = 'Student Model'
FROM MostPresentStudent MPS
INNER JOIN Students S
ON MPS.StudentID = S.StudentID
```

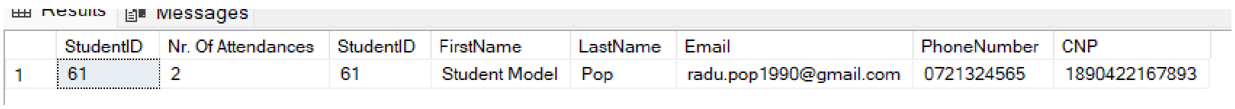
```
-- test

WITH AttendanceCount AS (
	SELECT S.StudentID, COUNT(CA.StudentID) as 'Nr. Of Attendances'
	FROM Students S
	LEFT JOIN CourseAttendance CA
	ON S.StudentID = CA.StudentID
	GROUP BY S.StudentID
),
MostPresentStudent AS (
	SELECT TOP(1) *
	FROM AttendanceCount AC
	ORDER BY AC.[Nr. Of Attendances] DESC
)
SELECT *
FROM MostPresentStudent MPS
INNER JOIN Students S
ON MPS.StudentID = S.StudentID
```

Before



	StudentID	Nr. Of Attendances	StudentID	FirstName	LastName	Email	PhoneNumber	CNP
1	61	2	61	Radu	Pop	radu.pop1990@gmail.com	0721324565	1890422167893

After



	StudentID	Nr. Of Attendances	StudentID	FirstName	LastName	Email	PhoneNumber	CNP
1	61	2	61	Student Model	Pop	radu.pop1990@gmail.com	0721324565	1890422167893

14 Tranzactii

```
------------------------------------------------------
------------------- Tranzactions ---------------------
------------------------------------------------------
```

14.1 Sa se creeze un nou material dintr-o noua categorie de materiale si sa se adauge la curs

```
GO
BEGIN TRY
      BEGIN TRANSACTION
      DECLARE @scheduleID INT = 2
      DECLARE @catName VARCHAR(50) = 'Categorie Noua'
      DECLARE @matName VARCHAR(50) = 'Material Nou'
      DECLARE @unitPrice DECIMAL(8,2) = 8.00
      DECLARE @unitsInStock INT = 15

      INSERT INTO MaterialCategories(CategoryName)
      VALUES
      (@catName)

      DECLARE @newCatID INT = SCOPE_IDENTITY()

      INSERT INTO Materials(MaterialName, CategoryID, UnitPrice,
UnitsInStock)
      VALUES
      (@matName, @newCatID, @unitPrice, @unitsInStock)

      DECLARE @newMatID INT = SCOPE_IDENTITY()

      INSERT INTO CourseMaterials(MaterialID, ScheduleID, Quantity)
      VALUES(@newMatID, @scheduleID, @unitsInStock)

      COMMIT TRANSACTION
END TRY
BEGIN CATCH
      IF @@TRANCOUNT > 0
      BEGIN
            PRINT XACT_STATE();
            ROLLBACK TRANSACTION
      END
      PRINT ERROR_MESSAGE();
      THROW 50000, 'An insert failed. The transaction was cancelled', 0;
END CATCH
```

14.2 Sa se cumpere o cantitate de material de la un furnizor nou si sa se afiseze pretul

```
GO
BEGIN TRY
      BEGIN TRANSACTION

      DECLARE @SupplierName VARCHAR(50) = 'Furnizor nou'
      DECLARE @Email VARCHAR(50) = 'furnizornou@furnizor.com'
      DECLARE @PhoneNumber VARCHAR(20) = '0213558080'
      DECLARE @CIF VARCHAR(13) = '000000'
      DECLARE @MaterialID INT = 49
      DECLARE @CategoryID INT
      DECLARE @Quantity INT = 20

      IF NOT EXISTS( SELECT * FROM Materials WHERE MaterialID = 
@MaterialID )
      BEGIN
            ;THROW 50000, 'Material doesn''t exist!', 0
      END

      SET @CategoryID = (
            SELECT M.CategoryID
            FROM Materials M
            WHERE M.MaterialID = @MaterialID
      )

      INSERT INTO Suppliers(SupplierName, CategoryID, Email, PhoneNumber, 
CIF)
      VALUES
      (@SupplierName, @CategoryID,@Email,@PhoneNumber,@CIF)

      DECLARE @SupplierID INT = SCOPE_IDENTITY();

      INSERT INTO Orders(SupplierID, MaterialID, Quantity, OrderDate, 
ArrivalDate)
      VALUES
      (@SupplierID, @MaterialID, @Quantity, GETDATE(), NULL)

      IF @@ROWCOUNT = 0
      BEGIN
            ;THROW 5000, 'Insert in Orders failed!', 0
      END

      SELECT O.Quantity * M.UnitPrice  as 'Sum'
      FROM Orders O
      INNER JOIN Materials M
      ON O.MaterialID = M.MaterialID
      WHERE O.SupplierID = @SupplierID

      COMMIT TRANSACTION
END TRY
BEGIN CATCH
      IF @@TRANCOUNT > 0
      BEGIN
            PRINT XACT_STATE();
            ROLLBACK TRANSACTION
      END
      PRINT ERROR_MESSAGE();
      THROW 50000, 'The transaction was cancelled', 0;
END CATCH
```

14.3 Sa se inregistreze in prezenta un cursant nou care nu se inscrisese

```
GO
BEGIN TRY
	BEGIN TRANSACTION

	DECLARE @FirstName VARCHAR(50) = 'Cursant'
	DECLARE @LastName VARCHAR(50) = 'Nou'
	DECLARE @Email VARCHAR(50) = 'cursantnou@gmail.com'
	DECLARE @PhoneNumber VARCHAR(20) = '0724123123'
	DECLARE @CNP VARCHAR(13) = '5000000000000'

	DECLARE @scheduleID INT = 2

	INSERT INTO Students(FirstName, LastName, Email, PhoneNumber, CNP)
	VALUES
	(@FirstName, @LastName, @Email, @PhoneNumber, @CNP)

	IF @@ROWCOUNT = 0
	BEGIN
		;THROW 50000, 'Insert into Students failed', 0
	END

	DECLARE @StdID INT = SCOPE_IDENTITY()

	INSERT INTO CourseRegistrations(ScheduleID, StudentID)
	VALUES
	(@scheduleID, @StdID)

	IF @@ROWCOUNT = 0
	BEGIN
		;THROW 50000, 'Insert into CourseRegistrations failed', 0
	END

	INSERT INTO CourseAttendance(ScheduleID, StudentID)
	VALUES
	(@scheduleID, @StdID)

	IF @@ROWCOUNT = 0
	BEGIN
		;THROW 50000, 'Insert into CourseAttendance failed', 0
	END

	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	IF @@TRANCOUNT > 0
	BEGIN
		PRINT XACT_STATE();
		ROLLBACK TRANSACTION
	END
	PRINT ERROR_MESSAGE();
	THROW 50000, 'The transaction was cancelled', 0
END CATCH
```

14.4 Sa se planifice un nou curs sugerat de un nou instructor

```
GO
BEGIN TRY
      BEGIN TRANSACTION

            DECLARE @FirstName VARCHAR(50) = 'Victor'
            DECLARE @LastName VARCHAR(50) = 'Daniel'
            DECLARE @Email VARCHAR(50) = 'victordaniel@gmail.com'
            DECLARE @PhoneNumber VARCHAR(20) = '0768353134'
            DECLARE @CourseCategory INT = 2
            DECLARE @CNP VARCHAR(13) = '5000000000001'

            DECLARE @CourseName VARCHAR(50) = 'Curs Nou'
            DECLARE @Price INT = 250

            DECLARE @CourseID INT
            DECLARE @StartDate DATE = '2024-02-01'
            DECLARE @EndDate DATE = '2024-02-04'
            DECLARE @FacilityID INT = 2

            INSERT INTO Instructors(FirstName, LastName, Email,
PhoneNumber, ExpertiseArea, CNP)
            VALUES
            (@FirstName, @LastName, @Email, @PhoneNumber, @CourseCategory,
@CNP)

            IF @@ROWCOUNT = 0
            BEGIN
                  ;THROW 50000, 'Failed to insert into Instructors', 0
            END

            DECLARE @InstructorID INT = SCOPE_IDENTITY()

            INSERT INTO Courses(CourseName, CategoryID, Price)
            VALUES
            (@CourseName, @CourseCategory, @Price)

            IF @@ROWCOUNT = 0
            BEGIN
                  ;THROW 50000, 'Failed to insert into Courses', 0
            END

            SET @CourseID = SCOPE_IDENTITY()

            INSERT INTO CourseSchedule(CourseID, StartDate, EndDate,
FacilityID, InstructorID)
            VALUES
            (@CourseID, @StartDate, @EndDate, @FacilityID, @InstructorID)


      COMMIT TRANSACTION
END TRY
BEGIN CATCH
      IF @@TRANCOUNT > 0
      BEGIN
            PRINT XACT_STATE();
            ROLLBACK TRANSACTION
      END
      PRINT ERROR_MESSAGE();
      THROW 50000, 'The transaction was cancelled', 0
END CATCH
```

14.5 comanda materiale din primele 2 categorii de la toti furnizorii posibili

```
GO
BEGIN TRY
      BEGIN TRANSACTION

            DECLARE @Quantity INT = 50

            DECLARE @tempCategories TABLE (CategoryID INT, CategoryName
VARCHAR(50))

            INSERT INTO @tempCategories
            SELECT TOP 2 MC.CategoryID, MC.CategoryName
            FROM MaterialCategories MC

            IF @@ROWCOUNT = 0
            BEGIN
                  ;THROW 50000, 'Failed to get the first 2 categories', 0
            END

            DECLARE @supplierIDs TABLE(SupplierID INT)

            INSERT INTO @supplierIDs
            SELECT S.SupplierID
            FROM Suppliers S
            WHERE S.CategoryID IN (SELECT CategoryID FROM @tempCategories)

            IF @@ROWCOUNT = 0
            BEGIN
                  ;THROW 50000, 'Failed to get the supplierIDs', 0
            END

            INSERT INTO Orders(SupplierID, MaterialID, Quantity,
OrderDate, ArrivalDate)
            SELECT S.SupplierID, M.MaterialID, @Quantity, GETDATE(), NULL
            FROM Suppliers S CROSS JOIN Materials M
            WHERE S.CategoryID = M.CategoryID

            IF @@ROWCOUNT = 0
            BEGIN
                  ;THROW 50000, 'Couldn''t place orders!', 0
            END

      COMMIT TRANSACTION
END TRY
BEGIN CATCH
```

```
      IF @@TRANCOUNT > 0
      BEGIN
            PRINT XACT_STATE();
            ROLLBACK TRANSACTION
      END
      PRINT ERROR_MESSAGE();
      THROW 50000, 'The transaction was cancelled', 0;
END CATCH
```

14.6 Comanda o cantitate de material, actualizand stocul materialului si afisand suma de plata

```
GO
BEGIN TRY
      BEGIN TRANSACTION

            DECLARE @SupplierID INT = 5
            DECLARE @MaterialID INT = 10
            DECLARE @Quantity INT = 10

            INSERT INTO Orders
            VALUES
            (@SupplierID, @MaterialID, @Quantity, GETDATE(), NULL)

            IF @@ROWCOUNT = 0
            BEGIN
                  ;THROW 50000, 'Failed to insert in Orders', 0
            END

            UPDATE M
            SET M.UnitsInStock = M.UnitsInStock + @Quantity
            FROM Materials M

            IF @@ROWCOUNT = 0
            BEGIN
                  ;THROW 50000, 'Failed to update Materials', 0
            END

            SELECT M.UnitPrice * @Quantity as 'Sum'
            FROM Materials M
            WHERE M.MaterialID = @MaterialID

            IF @@ROWCOUNT = 0
            BEGIN
                  ;THROW 50000, 'Failed to select the material', 0
            END

      COMMIT TRANSACTION
END TRY
BEGIN CATCH
      IF @@TRANCOUNT > 0
      BEGIN
            PRINT XACT_STATE();
            ROLLBACK TRANSACTION
      END
      PRINT ERROR_MESSAGE();
      THROW 50000, 'The transaction was cancelled', 0;
END CATCH
```

14.7 Comanda toate materialele pentru un curs, actualizand stocurile si afisand suma de plata

```
GO
BEGIN TRY
      BEGIN TRANSACTION
            DECLARE @scheduleID INT = 3

            DECLARE @tempOrders TABLE (
                  SupplierID INT,
                  MaterialID INT,
                  Quantity INT,
                  OrderDate DATE,
                  ArrivalDate DATE
            )

            INSERT INTO @tempOrders(SupplierID, MaterialID, Quantity,
OrderDate, ArrivalDate)
            SELECT S.SupplierID, M.MaterialID, CM.Quantity, GETDATE(),
NULL
            FROM CourseMaterials CM
            INNER JOIN Materials M
            ON CM.MaterialID = M.MaterialID
            INNER JOIN Suppliers S
            ON M.CategoryID = S.CategoryID
            WHERE CM.ScheduleID = @scheduleID

            IF @@ROWCOUNT = 0
            BEGIN
                  ;THROW 50000, 'Failed to insert into temporary orders
table', 0
            END

            INSERT INTO Orders(SupplierID, MaterialID, Quantity,
OrderDate, ArrivalDate)
            SELECT tempO.SupplierID, tempO.MaterialID, tempO.Quantity,
tempO.OrderDate, tempO.ArrivalDate
            FROM @tempOrders AS tempO

            IF @@ROWCOUNT = 0
            BEGIN
                  ;THROW 50000, 'Failed to insert orders', 0
            END

            UPDATE M
            SET M.UnitsInStock = M.UnitsInStock + tempO.Quantity
            FROM Materials M INNER JOIN @tempOrders AS tempO
            ON M.MaterialID = tempO.MaterialID
            IF @@ROWCOUNT = 0
            BEGIN
                  ;THROW 50000, 'Failed to update stocks', 0
            END

      COMMIT TRANSACTION
END TRY
BEGIN CATCH
      IF @@TRANCOUNT > 0
      BEGIN
            PRINT XACT_STATE();
            ROLLBACK TRANSACTION
      END
      PRINT ERROR_MESSAGE();
      THROW 50000, 'The transaction was cancelled', 0;
END CATCH
```

14.8 Planifica un curs nou intr-o locatie noua cu un instructor al carei expertiza sa fie potrivita

```
GO
BEGIN TRY
      BEGIN TRANSACTION

            DECLARE @CourseName VARCHAR(50) = 'Curs Germana'
            DECLARE @CategoryID INT = 4
            DECLARE @Price DECIMAL(10,2) = 150

            DECLARE @FacilityName VARCHAR(50) = 'Hotel Ursul'
            DECLARE @FacilityAddress VARCHAR(100) = 'Aleea Vanatorilor nr.
9A'
            DECLARE @City VARCHAR(20) = 'Botosani'
            DECLARE @PhoneNumber VARCHAR(20) = '0724134763'
            DECLARE @Capacity INT = 40
            DECLARE @PricePerHour DECIMAL(8,2) = 200

            DECLARE @StartDate DATE = '2025-05-17'
            DECLARE @EndDate DATE = '2025-05-20'

            INSERT INTO Courses(CourseName, CategoryID, Price)
            VALUES
            (@CourseName, @CategoryID, @Price)

            IF @@ROWCOUNT = 0
            BEGIN
                  ;THROW 50000, 'Failed to insert into Courses', 0
            END

            DECLARE @CourseID INT = SCOPE_IDENTITY();

            INSERT INTO Facilities(FacilityName, FacilityAddress, City,
PhoneNumber, Capacity, PricePerHour)
            VALUES
            (@FacilityName, @FacilityAddress, @City, @PhoneNumber,
@Capacity, @PricePerHour)

            IF @@ROWCOUNT = 0
            BEGIN
                  ;THROW 50000, 'Failed to insert into Facilities', 0
            END

            DECLARE @FacilityID INT = SCOPE_IDENTITY();

            DECLARE @InstructorID INT = (
                  SELECT TOP 1 I.InstructorID FROM Instructors I WHERE
I.ExpertiseArea = @CategoryID
            )

            INSERT INTO CourseSchedule(CourseID, StartDate, EndDate,
FacilityID, InstructorID)
            VALUES
            (@CourseID, @StartDate, @EndDate, @FacilityID, @InstructorID)

      COMMIT TRANSACTION
END TRY
BEGIN CATCH
      IF @@TRANCOUNT > 0
      BEGIN
            PRINT XACT_STATE();
            ROLLBACK TRANSACTION
      END
      PRINT ERROR_MESSAGE();
      THROW 50000, 'The transaction was cancelled', 0;
END CATCH
```

14.9 Deschide un departament nou cu o functie noua si muta acolo un angajat

```
GO
BEGIN TRY
      BEGIN TRANSACTION

            DECLARE @depName VARCHAR(50) = 'Relatii cu publicul'
            DECLARE @roleTitle VARCHAR(50) = 'Purtator de cuvant'
            DECLARE @bruteSalary DECIMAL(10,2) = 5000
            DECLARE @employeeID INT = 5

            INSERT INTO Departments(DepName)
            VALUES
            (@depName)

            IF @@ROWCOUNT = 0
            BEGIN
                  ;THROW 50000, 'Failed to insert into Departments', 0
            END

            DECLARE @depID INT = SCOPE_IDENTITY()
            INSERT INTO EmployeeRoles(RoleTitle, BruteSalary)
            VALUES
            (@roleTitle, @bruteSalary)

            IF @@ROWCOUNT = 0
            BEGIN
                  ;THROW 50000, 'Failed to insert into EmployeeRoles', 0
            END

            DECLARE @roleID INT = SCOPE_IDENTITY()

            UPDATE E
            SET E.DepartmentID = @depID
            FROM Employees E
            WHERE E.EmployeeID = @employeeID

            UPDATE E
            SET E.RoleID = @roleID
            FROM Employees E
            WHERE E.EmployeeID = @employeeID

      COMMIT TRANSACTION
END TRY
BEGIN CATCH
      IF @@TRANCOUNT > 0
      BEGIN
            PRINT XACT_STATE();
            ROLLBACK TRANSACTION
      END
      PRINT ERROR_MESSAGE();
      THROW 50000, 'The transaction was cancelled', 0;
END CATCH
```

15 Extra

```
------------------------------------------------------
------------------ EXTRA QUERIES ---------------------
------------------------------------------------------
```

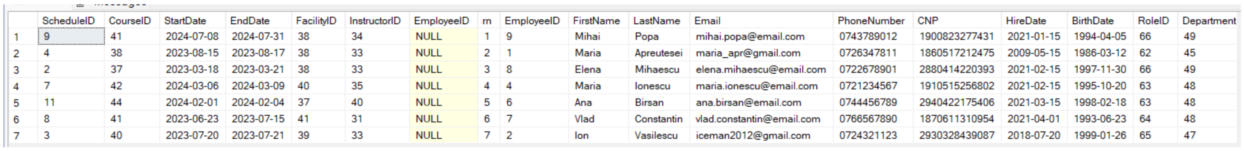
15.1 Sa se lege tabela Employees de CourseSchedule (de fiecare planificare este responsabil un angajat)

```
ALTER TABLE CourseSchedule
      ADD EmployeeID INT FOREIGN KEY REFERENCES Employees(EmployeeID)
```

15.2 Sa se asigneze la intamplare fiecarei planificari cate un angajat

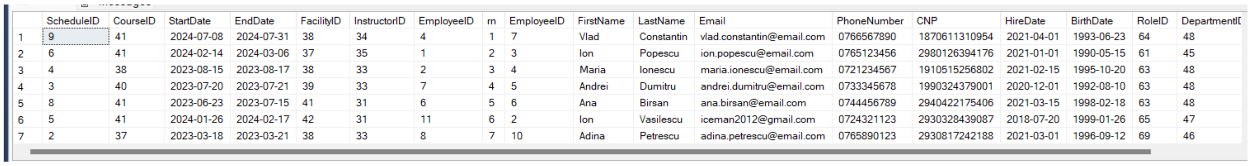
```
GO
WITH RandomlyOrderedEmployees AS
(
      SELECT *, ROW_NUMBER() OVER (ORDER BY NEWID()) as rn
      FROM Employees
),
RandomlyOrderedSchedules AS
(
      SELECT *, ROW_NUMBER() OVER (ORDER BY NEWID()) as rn
      FROM CourseSchedule
)
UPDATE ROS
SET ROS.EmployeeID = ROE.EmployeeID
FROM RandomlyOrderedSchedules ROS
INNER JOIN RandomlyOrderedEmployees ROE
ON ROS.rn = ROE.rn

-- test
GO
WITH RandomlyOrderedEmployees AS
(
      SELECT *, ROW_NUMBER() OVER (ORDER BY NEWID()) as rn
      FROM Employees
),
RandomlyOrderedSchedules AS
(
      SELECT *, ROW_NUMBER() OVER (ORDER BY NEWID()) as rn
      FROM CourseSchedule
)
SELECT *
FROM RandomlyOrderedSchedules ROS
INNER JOIN RandomlyOrderedEmployees ROE
ON ROS.rn = ROE.rn

SELECT * FROM CourseSchedule
```

Before



	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID	EmployeeID	rn	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	Department
1	9	41	2024-07-08	2024-07-31	38	34	NULL	1	9	Mihai	Popa	mihai.popa@email.com	0743789012	1900823277431	2021-01-15	1994-04-05	66	49
2	4	38	2023-08-15	2023-08-17	38	33	NULL	2	1	Maria	Apreutesei	maria_apr@gmail.com	0726347811	1860517212475	2009-05-15	1986-03-12	62	45
3	2	37	2023-03-18	2023-03-21	38	33	NULL	3	8	Elena	Mihaescu	elena.mihaescu@email.com	0722678901	2880414220393	2021-02-15	1997-11-30	66	49
4	7	42	2024-03-06	2024-03-09	40	35	NULL	4	4	Maria	Ionescu	maria.ionescu@email.com	0721234567	1910515256802	2021-02-15	1995-10-20	63	48
5	11	44	2024-02-01	2024-02-04	37	40	NULL	5	6	Ana	Birsan	ana.birsan@email.com	0744456789	2940422175406	2021-03-15	1998-02-18	63	48
6	8	41	2023-06-23	2023-07-15	41	31	NULL	6	7	Vlad	Constantin	vlad.constantin@email.com	0766567890	1870611310954	2021-04-01	1993-06-23	64	48
7	3	40	2023-07-20	2023-07-21	39	33	NULL	7	2	Ion	Vasilescu	iceman2012@gmail.com	0724321123	2930328439087	2018-07-20	1999-01-26	65	47

After

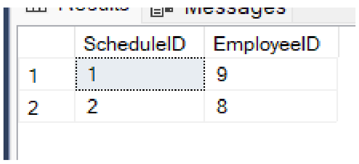


	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID	EmployeeID	rn	EmployeeID	FirstName	LastName	Email	PhoneNumber	CNP	HireDate	BirthDate	RoleID	DepartmentI
1	9	41	2024-07-08	2024-07-31	38	34	4	1	7	Vlad	Constantin	vlad.constantin@email.com	0766567890	1870611310954	2021-04-01	1993-06-23	64	48
2	6	41	2024-02-14	2024-03-06	37	35	1	2	3	Ion	Popescu	ion.popescu@email.com	0765123456	2980126394176	2021-01-01	1990-05-15	61	45
3	4	38	2023-08-15	2023-08-17	38	33	2	3	4	Maria	Ionescu	maria.ionescu@email.com	0721234567	1910515256802	2021-02-15	1995-10-20	63	48
4	3	40	2023-07-20	2023-07-21	39	33	7	4	5	Andrei	Dumitru	andrei.dumitru@email.com	0733345678	1990324379001	2020-12-01	1992-08-10	63	48
5	8	41	2023-06-23	2023-07-15	41	31	6	5	6	Ana	Birsan	ana.birsan@email.com	0744456789	2940422175406	2021-03-15	1998-02-18	63	48
6	5	41	2024-01-26	2024-02-17	42	31	11	6	2	Ion	Vasilescu	iceman2012@gmail.com	0724321123	2930328439087	2018-07-20	1999-01-26	65	47
7	2	37	2023-03-18	2023-03-21	38	33	8	7	10	Adina	Petrescu	adina.petrescu@email.com	0765890123	2930817242188	2021-03-01	1996-09-12	69	46

15.3 Sa se actualizeze planificarea astfel incat angajatul cu un CNP anume sa organizeze inca 2 planificari anume

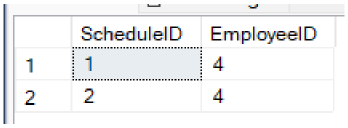
```
GO
DECLARE @CNP VARCHAR(13) = '1910515256802'
DECLARE @scheduleID1 INT = 1
DECLARE @scheduleID2 INT = 2

UPDATE CS
SET CS.EmployeeID = E.EmployeeID
FROM CourseSchedule CS
CROSS JOIN Employees E
WHERE E.CNP = @CNP AND ( CS.ScheduleID = @scheduleID1 OR CS.ScheduleID =
@scheduleID2 )

-- test
SELECT CS.ScheduleID, CS.EmployeeID
FROM CourseSchedule CS
CROSS JOIN Employees E
WHERE E.CNP = @CNP AND ( CS.ScheduleID = @scheduleID1 OR CS.ScheduleID =
@scheduleID2 )
```

Before



	ScheduleID	EmployeeID
1	1	9
2	2	8

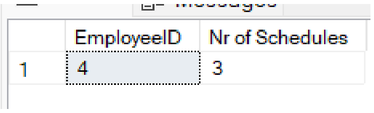
After



	ScheduleID	EmployeeID
1	1	4
2	2	4

15.4 Sa se creeze un view care sa afiseze angajatul cu cele mai multe cursuri organizate

```
GO
IF OBJECT_ID('Top1EmployeeBySchedule', 'V') IS NOT NULL
      DROP VIEW Top1EmployeeBySchedule

GO
CREATE VIEW Top1EmployeeBySchedule
AS
SELECT TOP(1) EmployeeID,COUNT(*) as 'Nr of Schedules'
FROM CourseSchedule
GROUP BY EmployeeID
ORDER BY [Nr of Schedules] DESC

-- test

SELECT * FROM Top1EmployeeBySchedule
```

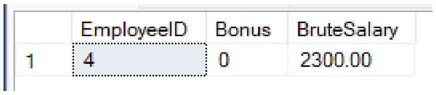


	EmployeeID	Nr of Schedules
1	4	3

15.5 Sa se ofere bonus de 25% din salariu angajatului cu cele mai multe cursuri organizate

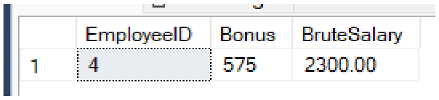
```
UPDATE E
SET E.Bonus = E.Bonus + ER.BruteSalary * 0.25
FROM Top1EmployeeBySchedule EBS
INNER JOIN Employees E
ON EBS.EmployeeID = E.EmployeeID
INNER JOIN EmployeeRoles ER
ON E.RoleID = ER.RoleID

-- test

SELECT E.EmployeeID, E.Bonus, ER.BruteSalary
FROM Top1EmployeeBySchedule EBS
INNER JOIN Employees E
ON EBS.EmployeeID = E.EmployeeID
INNER JOIN EmployeeRoles ER
ON E.RoleID = ER.RoleID
```

Before



	EmployeeID	Bonus	BruteSalary
1	4	0	2300.00

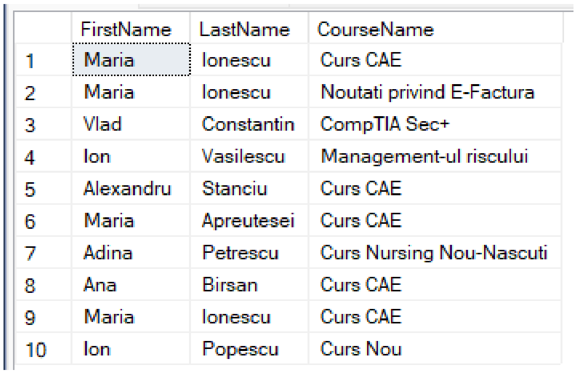
After



	EmployeeID	Bonus	BruteSalary
1	4	575	2300.00

15.6 Sa se afiseze ce cursuri (numele lor) organizeaza fiecare angajat (numele lor)

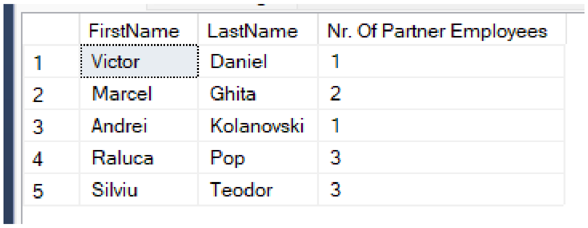
```
SELECT E.FirstName, E.LastName, C.CourseName
FROM CourseSchedule CS
INNER JOIN Employees E
ON CS.EmployeeID = E.EmployeeID
INNER JOIN Courses C
ON C.CourseID = CS.CourseID
```



	FirstName	LastName	CourseName
1	Maria	Ionescu	Curs CAE
2	Maria	Ionescu	Noutati privind E-Factura
3	Vlad	Constantin	CompTIA Sec+
4	Ion	Vasilescu	Management-ul riscului
5	Alexandru	Stanciu	Curs CAE
6	Maria	Apreutesei	Curs CAE
7	Adina	Petrescu	Curs Nursing Nou-Nascuti
8	Ana	Birsan	Curs CAE
9	Maria	Ionescu	Curs CAE
10	Ion	Popescu	Curs Nou

15.7 Sa se afiseze cati angajati au colaborat cu fiecare lector (si numele lectorului)

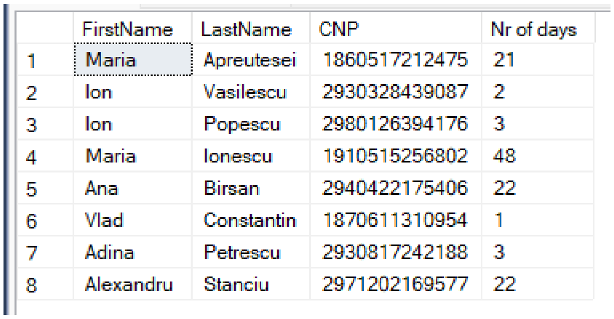
```
SELECT I.FirstName, I.LastName, COUNT ( DISTINCT CS.EmployeeID ) as 'Nr.
Of Partner Employees'
FROM CourseSchedule CS
INNER JOIN Instructors I
ON CS.InstructorID = I.InstructorID
GROUP BY I.FirstName, I.LastName
```



	FirstName	LastName	Nr. Of Partner Employees
1	Victor	Daniel	1
2	Marcel	Ghita	2
3	Andrei	Kolanovski	1
4	Raluca	Pop	3
5	Silviu	Teodor	3

15.8 Sa se afiseze cate zile de curs are fiecare angajat

```
SELECT E.FirstName, E.LastName, E.CNP, SUM(DATEDIFF(DAY, CS.StartDate,
CS.EndDate)) as 'Nr of days'
FROM CourseSchedule CS
INNER JOIN Employees E
ON CS.EmployeeID = E.EmployeeID
GROUP BY E.FirstName, E.LastName, E.CNP
```

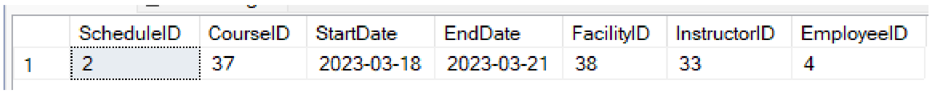


	FirstName	LastName	CNP	Nr of days
1	Maria	Apreutesei	1860517212475	21
2	Ion	Vasilescu	2930328439087	2
3	Ion	Popescu	2980126394176	3
4	Maria	Ionescu	1910515256802	48
5	Ana	Birsan	2940422175406	22
6	Vlad	Constantin	1870611310954	1
7	Adina	Petrescu	2930817242188	3
8	Alexandru	Stanciu	2971202169577	22

15.9 Sa se schimbe angajatul care se ocupa de o planificare anume

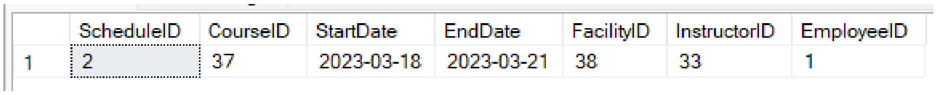
```
DECLARE @scheduleID INT = 2
DECLARE @newEmployeeID INT = 1

UPDATE CS
SET EmployeeID = @newEmployeeID
FROM CourseSchedule CS
WHERE ScheduleID = @scheduleID

-- test
SELECT *
FROM CourseSchedule CS
WHERE CS.ScheduleID = @scheduleID
```

Before



	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID	EmployeeID
1	2	37	2023-03-18	2023-03-21	38	33	4

After

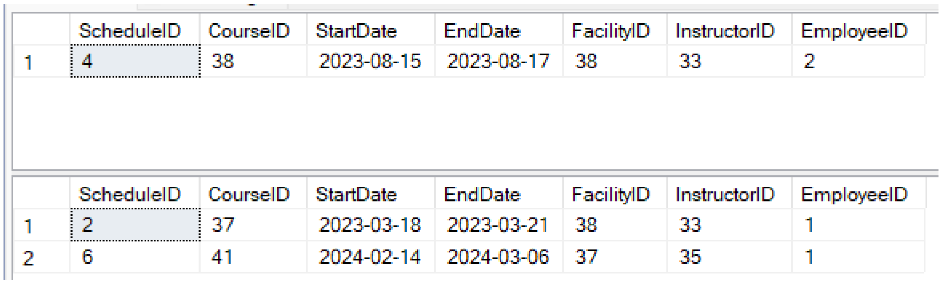


	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID	EmployeeID
1	2	37	2023-03-18	2023-03-21	38	33	1

15.10 Sa se mute toate planificarile unui angajat catre alt angajat

```
DECLARE @newEmployeeID INT = 1
DECLARE @oldEmployeeID INT = 2

UPDATE CS
SET CS.EmployeeID = @newEmployeeID
FROM CourseSchedule CS
WHERE CS.EmployeeID = @oldEmployeeID

-- test
SELECT *
FROM CourseSchedule CS
WHERE CS.EmployeeID = @oldEmployeeID

SELECT *
FROM CourseSchedule CS
WHERE CS.EmployeeID = @newEmployeeID
```

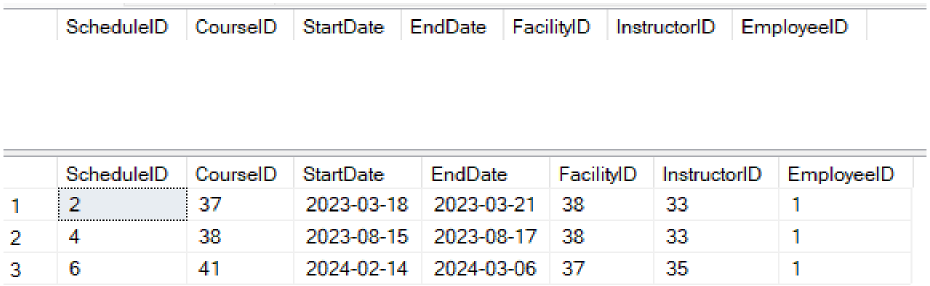
Before



	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID	EmployeeID
1	4	38	2023-08-15	2023-08-17	38	33	2

	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID	EmployeeID
1	2	37	2023-03-18	2023-03-21	38	33	1
2	6	41	2024-02-14	2024-03-06	37	35	1

After

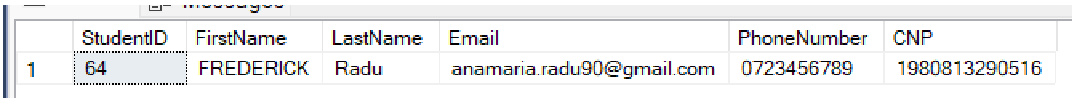


	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID	EmployeeID

	ScheduleID	CourseID	StartDate	EndDate	FacilityID	InstructorID	EmployeeID
1	2	37	2023-03-18	2023-03-21	38	33	1
2	4	38	2023-08-15	2023-08-17	38	33	1
3	6	41	2024-02-14	2024-03-06	37	35	1

15.11 Selectati toti studentii care nu sunt inscrisi la niciun curs

```
SELECT S.*
FROM Students S
LEFT JOIN CourseRegistrations CR
ON S.StudentID = CR.StudentID
WHERE CR.StudentID IS NULL
```



	StudentID	FirstName	LastName	Email	PhoneNumber	CNP
1	64	FREDERICK	Radu	anamaria.radu90@gmail.com	0723456789	1980813290516

16 Evidenta Query-urilor

Capitol	**Nr. Cerinte**
Creare baza de date	1
Creare tabele	18
View	6
Triggers	10
Proceduri stocate	10
INSERT	17
SELECT	32
UPDATE	16
DELETE	15
CTE	5
Tranzactii	9
Extra	11
Total	150

Atentie!! Desi sunt doar 9 cerinte in capitolul Tranzactii, minimul de 10 tranzactii a fost atins, fiindca exista o tranzactie care curprinde toate cerintele din crearea de tabele si nu a fost inclusa in capitolul Tranzactii.