**Databases Laboratory Work Nr 7**

**Title:** Diagrams, Schemes and Synonyms

**Prerequisites:** SQL Server 2019 and SSMS

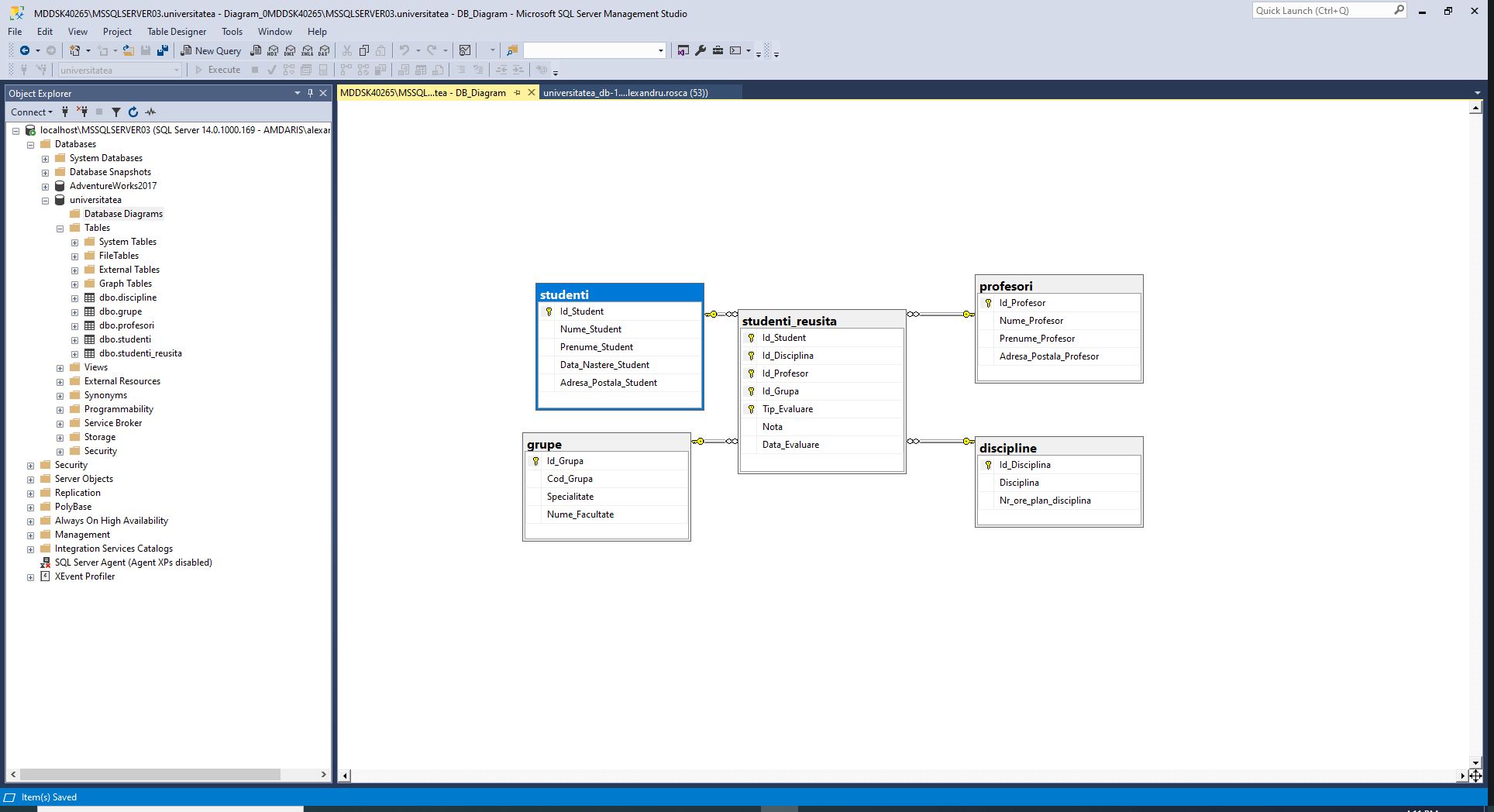
**Objectives:** get practical experience on creating diagrams and schemes. Learn how to create tables, primary and foreign keys and constraints in a diagram. Use synonyms to simplify SQL queries.

**Tasks:**

1. Create Diagram of database which is described in practical part of chapter 4.
2. Add necessary constraints for column ‘Sef\_grupa’ and ‘Prof\_Indrumator’ (described in task3, chapter 6) from table ‘grupe’.
3. Add to the constructed diagram new table – ‘orarul’, described in chapter 6 with following columns: Id\_Disciplina, Id\_Profesor, Bloc. Table’s key is built from 3 columns: Id\_Grupa, Zi, Auditoriu.
4. ‘orarul’ table must have 2 foreign keys: (Zi, Ora, Id\_Grupa, Id\_Profesor) and (Zi, Ora, Id\_Grupa, Id\_Disciplina).
5. Table should also contain referential constraints for Id\_Disciplinam Id\_Profesor, Id\_Grupa.
6. Create 3 new schemes: cadre\_didactice, plan\_studii, studenti. Move table ‘profesori’ into ‘cadre\_didactice’ scheme. Move table ‘orarul’, ‘discipline’ into ‘plan\_studii’ scheme. Move tables ‘sudenti’, ‘studenti\_reusita’ into ‘studenti’ scheme.
7. Modify 2-3 queries from practical tasks from chapter 4, so table names will have explicitly defined names (including new schemes).
8. Create synonyms in order to simplify queries from previous tasks.

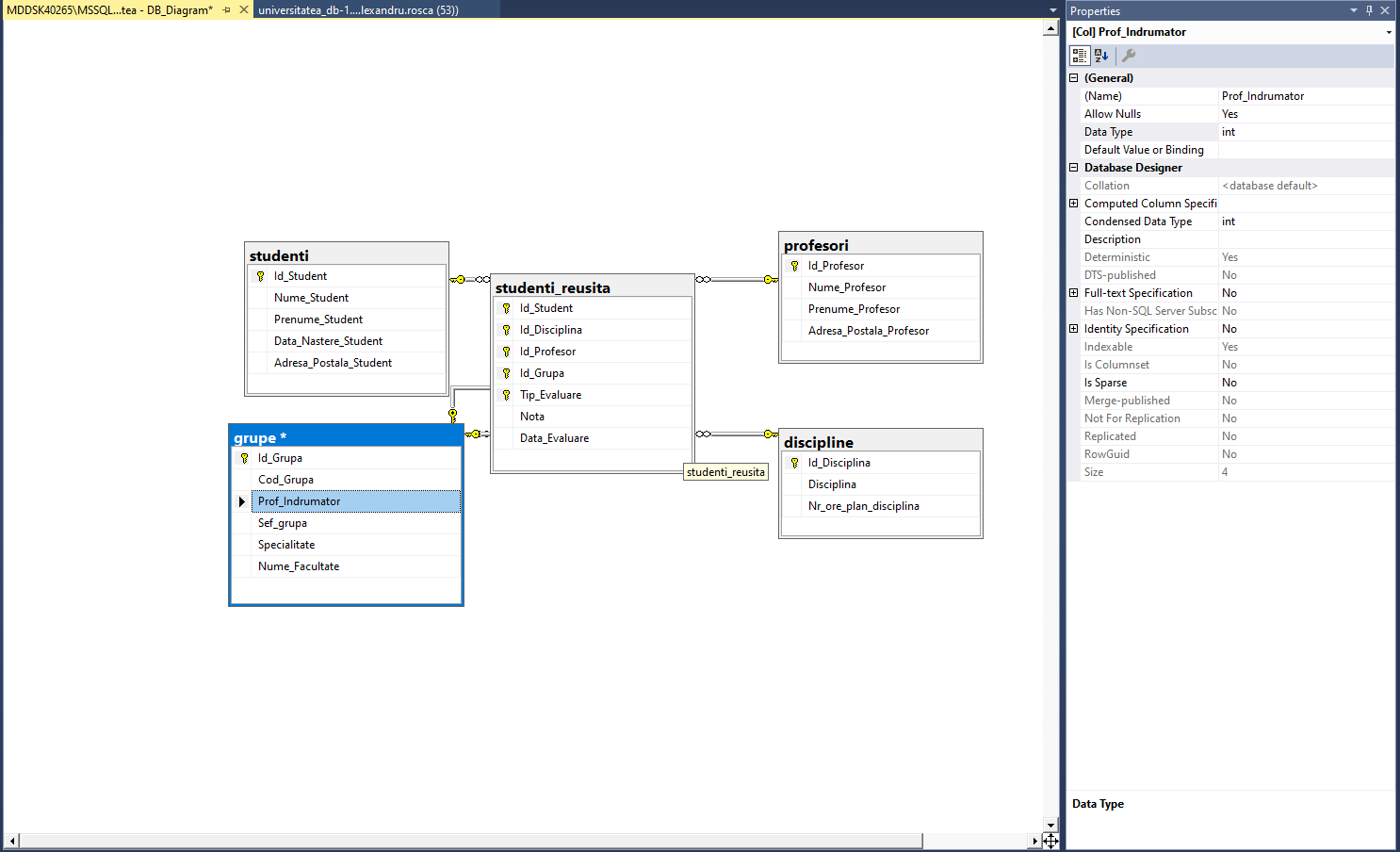
**Implementation:**

1. Created diagram for ‘universitatea’ database.

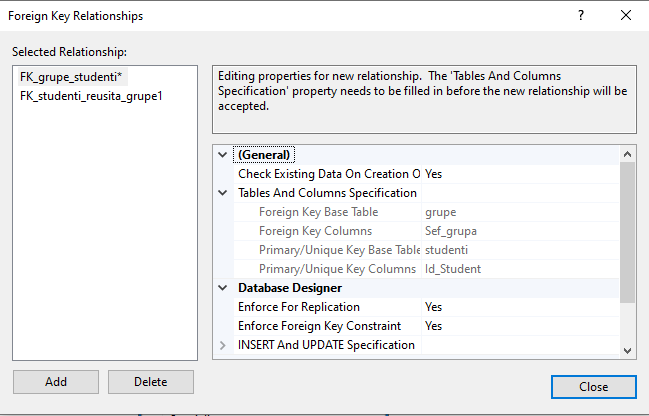
****

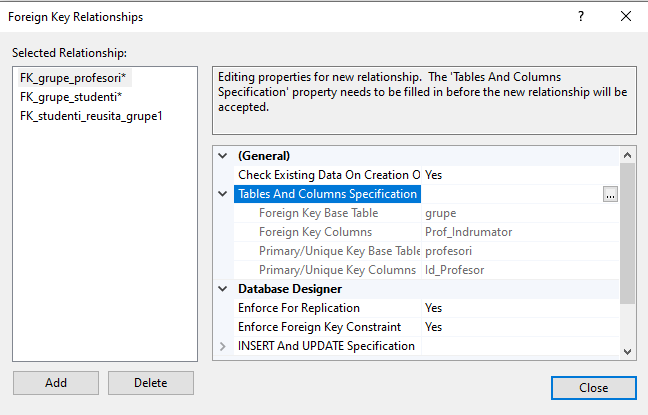
**2.**

I’ve created 2 column in table ‘grupe’ and changed their type to ‘int’

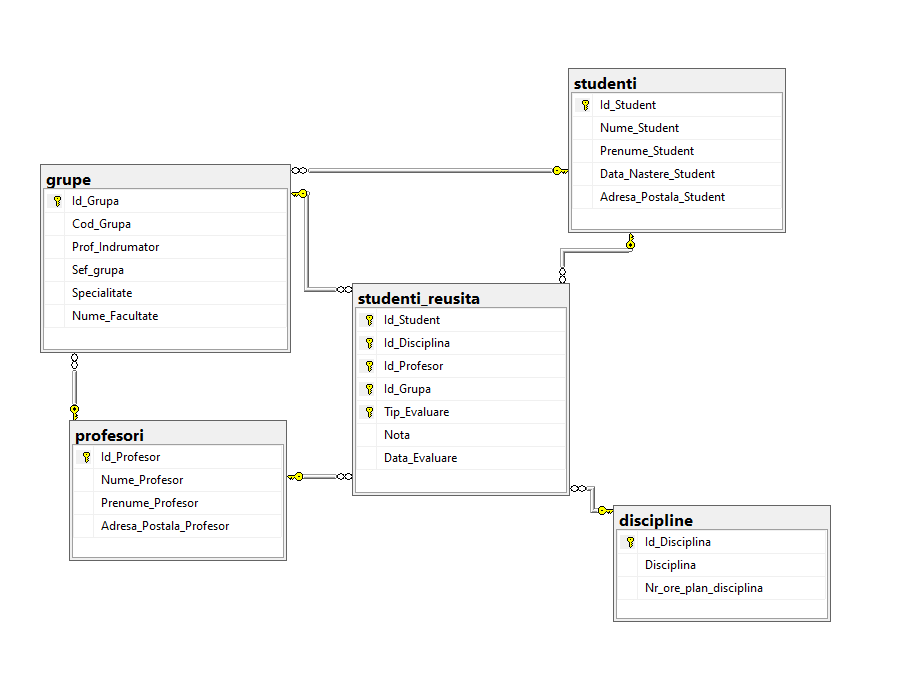


Then I’ve added a foreign key constraint as it is required in task

****

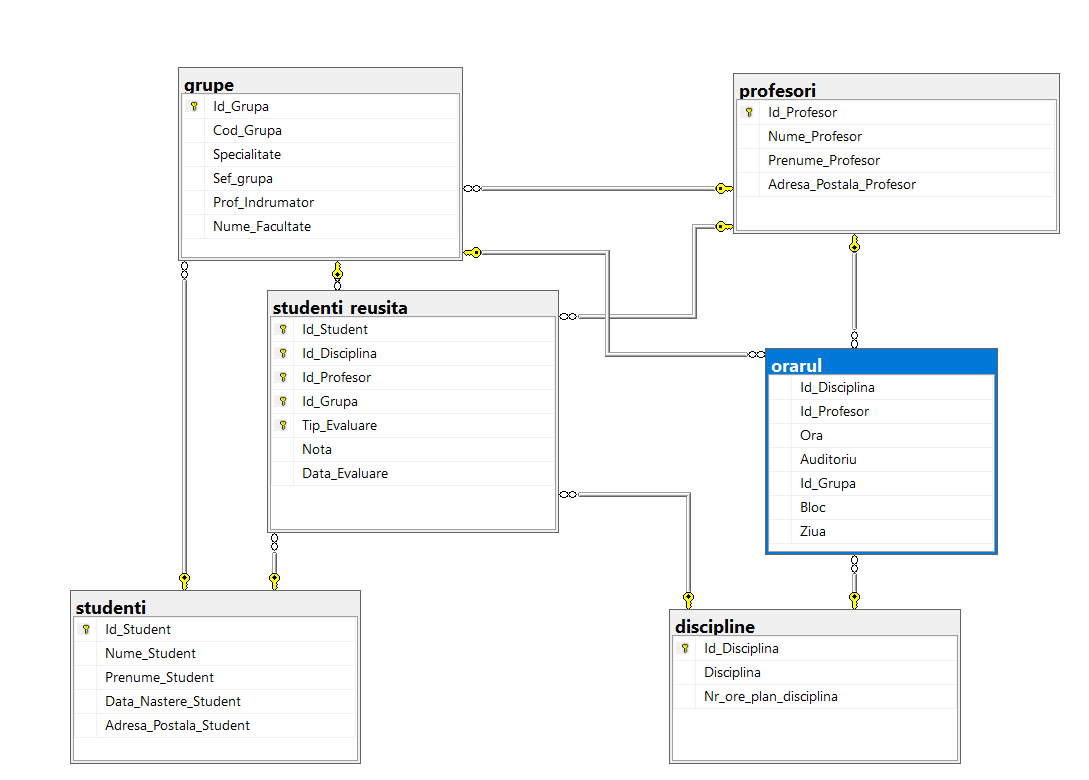
****

Know diagram looks like this

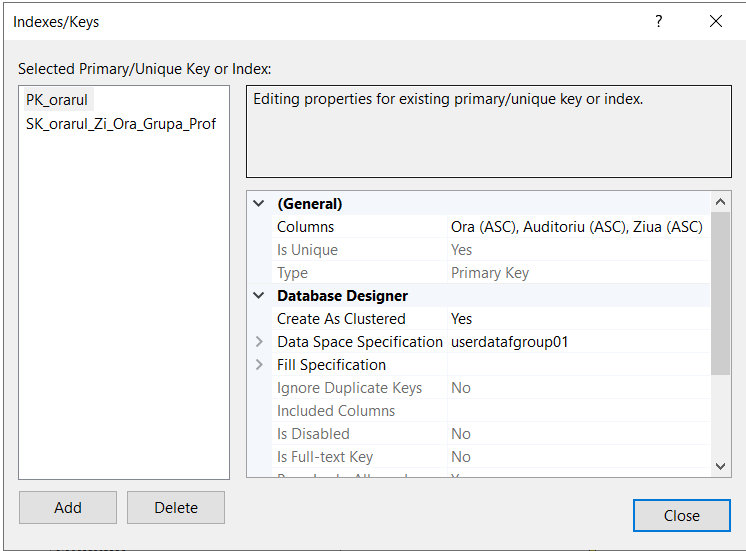


**3.**

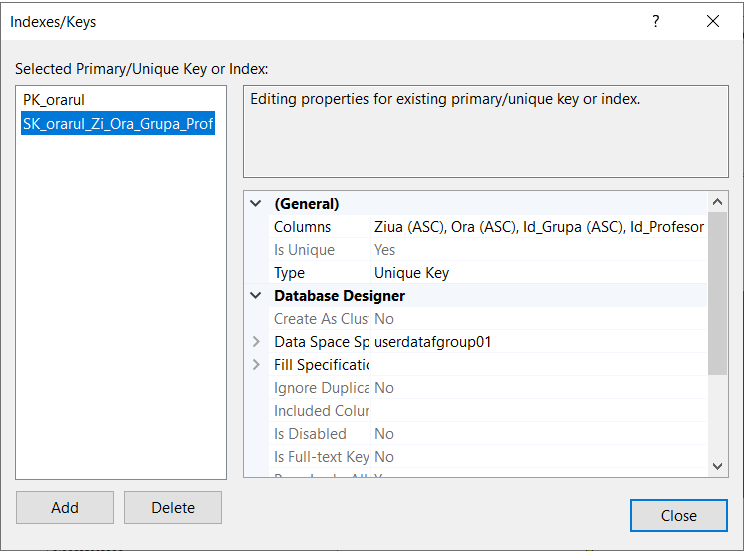
Create new table with all required columns

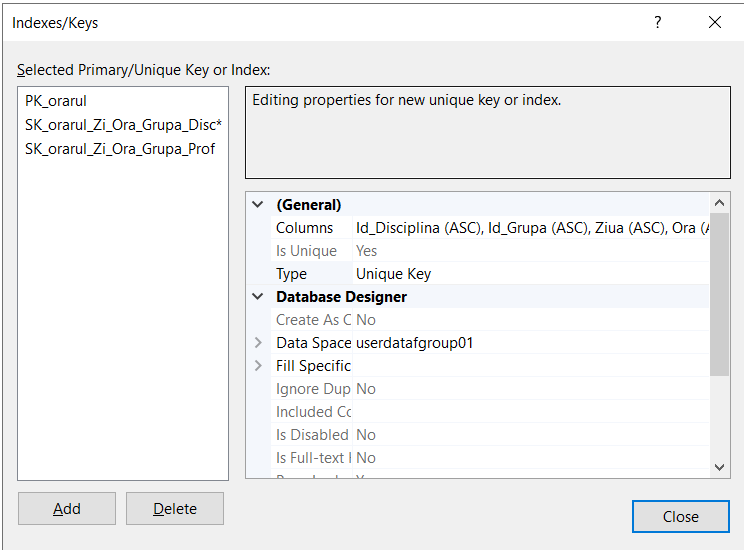
****

Then, create primary key by ‘Auditoriu’, ‘Ziua’, ‘Ora’ columns



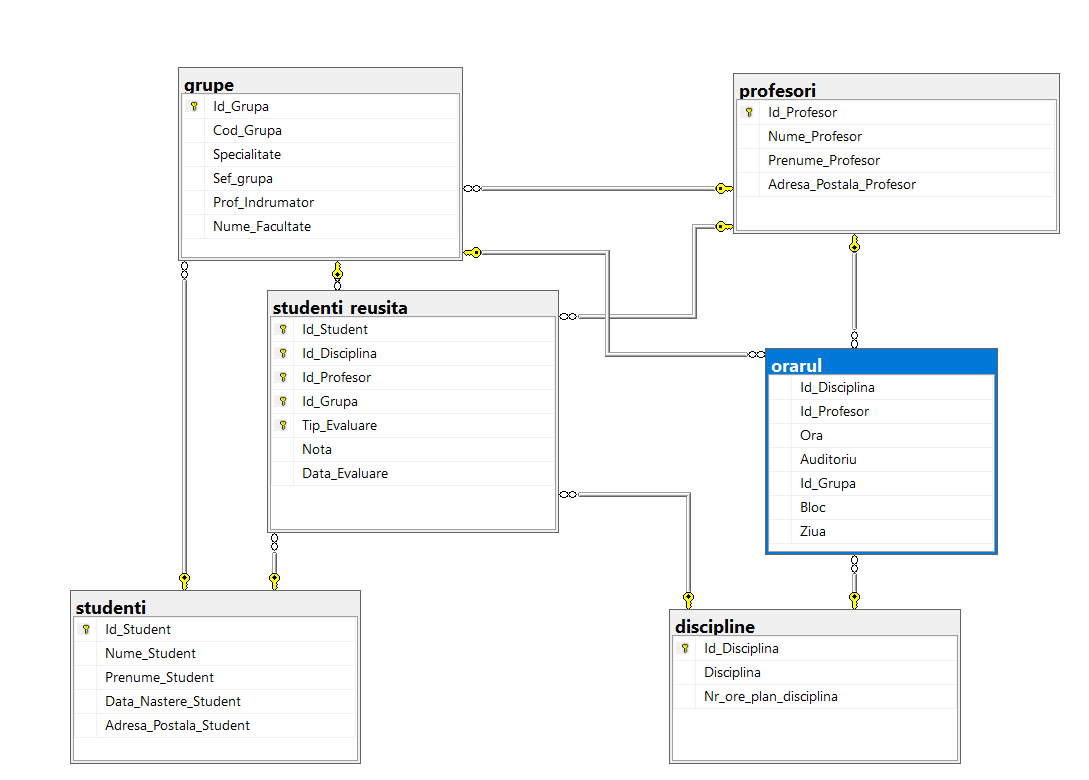
And add two secondary keys:





**5.**

Added FK constraints for ‘orarul’ table

****

**6.**

Created required schemes:

CREATE SCHEMA cadre\_didactice;

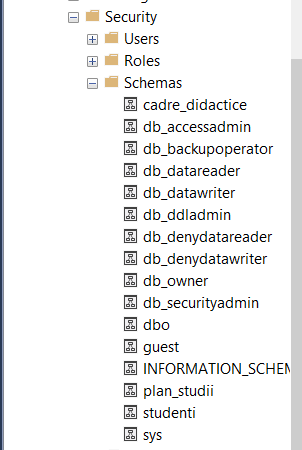
GO

CREATE SCHEMA plan\_studii;

GO

CREATE SCHEMA studenti;

GO



Changed the scheme using ALTER SCHEMA command:

USE universitatea;

GO

ALTER SCHEMA cadre\_didactice TRANSFER dbo.profesori;

GO

ALTER SCHEMA studenti TRANSFER dbo.studenti;

ALTER SCHEMA studenti TRANSFER dbo.studenti\_reusita;

GO

ALTER SCHEMA plan\_studii TRANSFER dbo.orarul;

ALTER SCHEMA plan\_studii TRANSFER dbo.discipline;

GO

**7.**

Examples of queries with new schemas:

/\* 7 \*/

SELECT DISTINCT Cod\_Grupa, Adresa\_Postala\_Student FROM dbo.grupe

INNER JOIN studenti.studenti\_reusita ON

dbo.grupe.Id\_Grupa = studenti.studenti\_reusita.Id\_Grupa

INNER JOIN studenti.studenti ON

studenti.studenti\_reusita.Id\_Student = studenti.studenti.Id\_Student

WHERE Adresa\_Postala\_Student LIKE '%31 August%'

GO

/\* 10 \*/

SELECT DISTINCT Nume\_Student, Prenume\_Student, Nota FROM studenti.studenti

INNER JOIN studenti.studenti\_reusita ON

studenti.studenti.Id\_Student = studenti.studenti\_reusita.Id\_Student

INNER JOIN plan\_studii.discipline ON

studenti.studenti\_reusita.Id\_Disciplina = plan\_studii.discipline.Id\_Disciplina AND

plan\_studii.discipline.Disciplina = 'Baze de date'

WHERE YEAR(Data\_Evaluare) = 2018 AND Nota BETWEEN 4 AND 8

GO

/\* 17 \*/

SELECT DISTINCT Nume\_Profesor, Prenume\_Profesor FROM cadre\_didactice.profesori

INNER JOIN studenti.studenti\_reusita ON

cadre\_didactice.profesori.Id\_Profesor = studenti.studenti\_reusita.Id\_Profesor

INNER JOIN studenti.studenti ON

studenti.studenti.Id\_Student = studenti.studenti\_reusita.Id\_Student AND studenti.studenti\_reusita.Id\_Student = 100

GO

**8.**

Queries rewritten with synonyms:

CREATE SYNONYM stud FOR studenti.studenti;

GO

CREATE SYNONYM stud\_reusita FOR studenti.studenti\_reusita;

GO

CREATE SYNONYM gr FOR dbo.grupe;

GO

CREATE SYNONYM disc FOR plan\_studii.discipline;

GO

CREATE SYNONYM prof FOR cadre\_didactice.profesori

GO

/\* 7 \*/

SELECT DISTINCT Cod\_Grupa, Adresa\_Postala\_Student FROM gr

INNER JOIN stud\_reusita ON gr.Id\_Grupa = stud\_reusita.Id\_Grupa

INNER JOIN stud ON stud\_reusita.Id\_Student = stud.Id\_Student

WHERE Adresa\_Postala\_Student LIKE '%31 August%'

GO

/\* 10 \*/

SELECT DISTINCT Nume\_Student, Prenume\_Student, Nota FROM stud

INNER JOIN stud\_reusita ON stud.Id\_Student = stud\_reusita.Id\_Student

INNER JOIN disc ON stud\_reusita.Id\_Disciplina = disc.Id\_Disciplina

AND disc.Disciplina = 'Baze de date'

WHERE YEAR(Data\_Evaluare) = 2018 AND Nota BETWEEN 4 AND 8

GO

/\* 17 \*/

SELECT DISTINCT Nume\_Profesor, Prenume\_Profesor FROM prof

INNER JOIN stud\_reusita ON prof.Id\_Profesor = stud\_reusita.Id\_Profesor

INNER JOIN stud ON stud.Id\_Student = stud\_reusita.Id\_Student

AND stud\_reusita.Id\_Student = 100

GO

**Conclusion:**

In this work I had an opportunity to work with diagrams and tried to implement tasks from previous chapters using graphical interface. By the end of this work I’ve got a better understanding of how diagrams work and why they are useful in process of building a database. Also, now I have basic knowledge of creating and managing schemas in a database. Moreover, I had more practice with creating synonyms and applying them to in SQL queries.