Databases Laboratory Work Nr 7

Title: Diagrams, Schemes and Synonyms

Prerequisites: SQL Server 2019 and SSMS

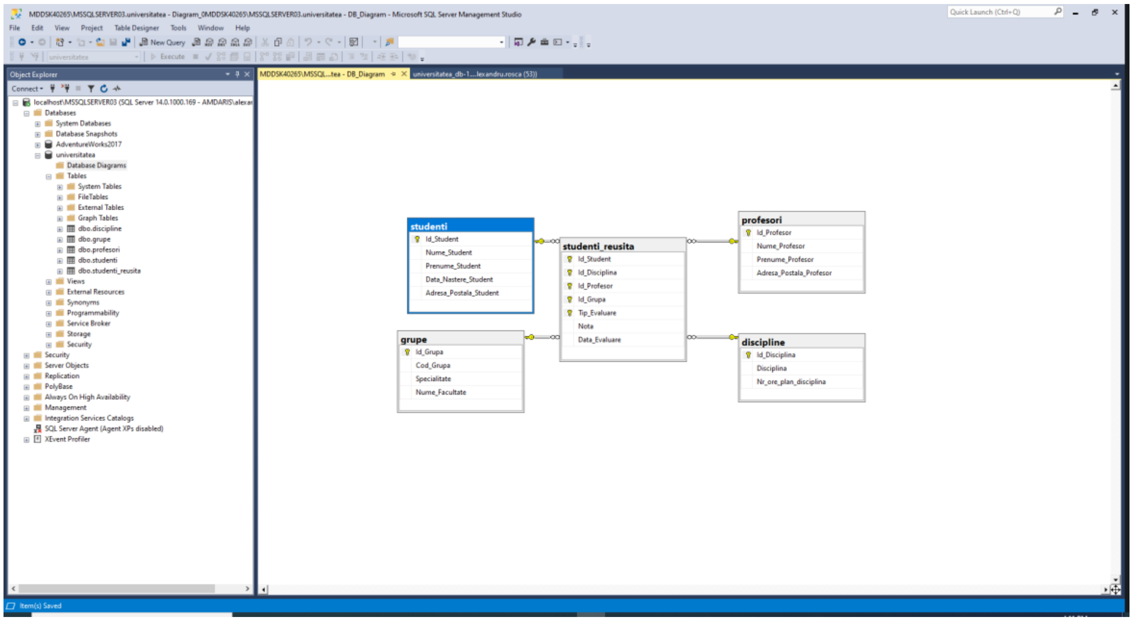
Objectives: get practical experience on creating diagrams and schemes. Learn how to create tables, primary and foreign keys and constraints in a diagram. Use synonyms to simplify SQL queries.

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.

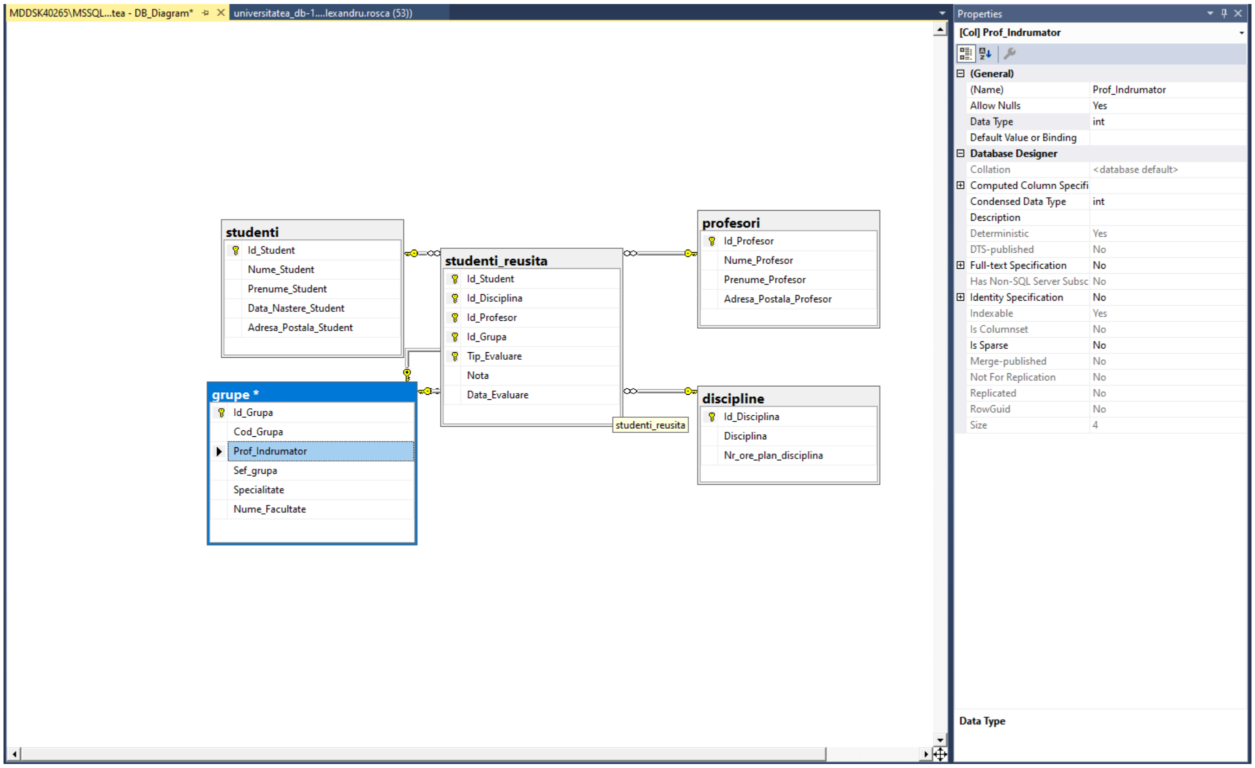
Implementation:

1. Created diagram for ‘universitatea’ database.



2.

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

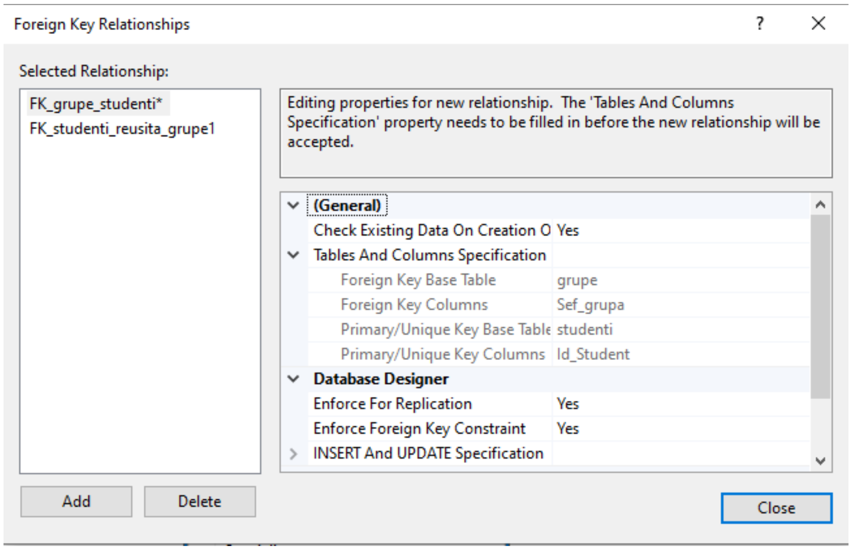


[Col] Prof_Indrumator

(General)	
(Name)	Prof_Indrumator
Allow Nulls	Yes
Data Type	int
Default Value or Binding	
Database Designer	
Collation	<database default>
Computed Column Specifi	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subsc	No
Identity Specification	No
Indexable	Yes
Is Columnset	No
Is Sparse	No
Merge-published	No
Not For Replication	No
Replicated	No
RowGuid	No
Size	4

Data Type

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

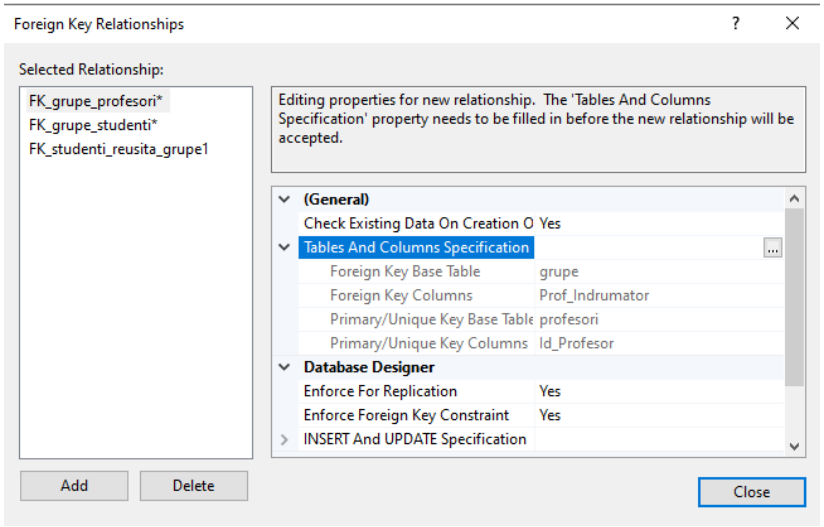


Selected Relationship:

FK_grupe_studenti*
FK_studenti_reusita_grupe1

Editing properties for new relationship. The 'Tables And Columns Specification' property needs to be filled in before the new relationship will be accepted.

(General)	
Check Existing Data On Creation O	Yes
Tables And Columns Specification	
Foreign Key Base Table	grupe
Foreign Key Columns	Sef_grupa
Primary/Unique Key Base Table	studenti
Primary/Unique Key Columns	Id_Student
Database Designer	
Enforce For Replication	Yes
Enforce Foreign Key Constraint	Yes
INSERT And UPDATE Specification	



Foreign Key Relationships

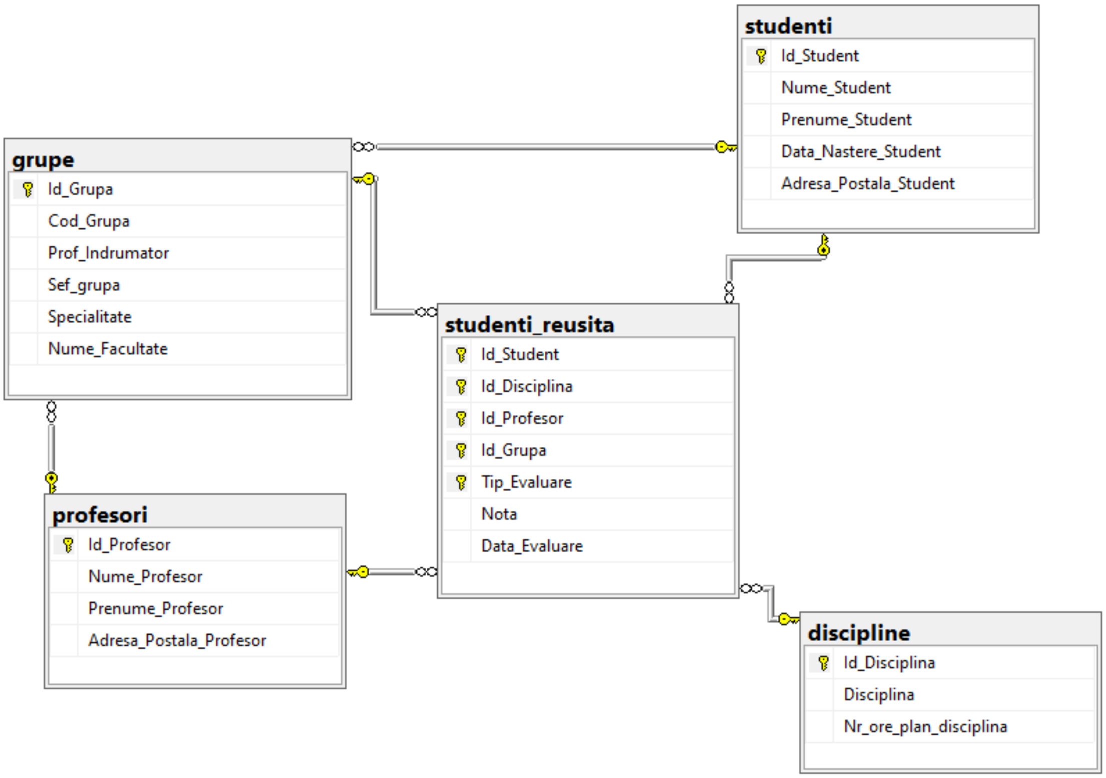
Selected Relationship:

- FK_grupe_profesori*
- FK_grupe_studenti*
- FK_studenti_reusita_grupe1

Editing properties for new relationship. The 'Tables And Columns Specification' property needs to be filled in before the new relationship will be accepted.

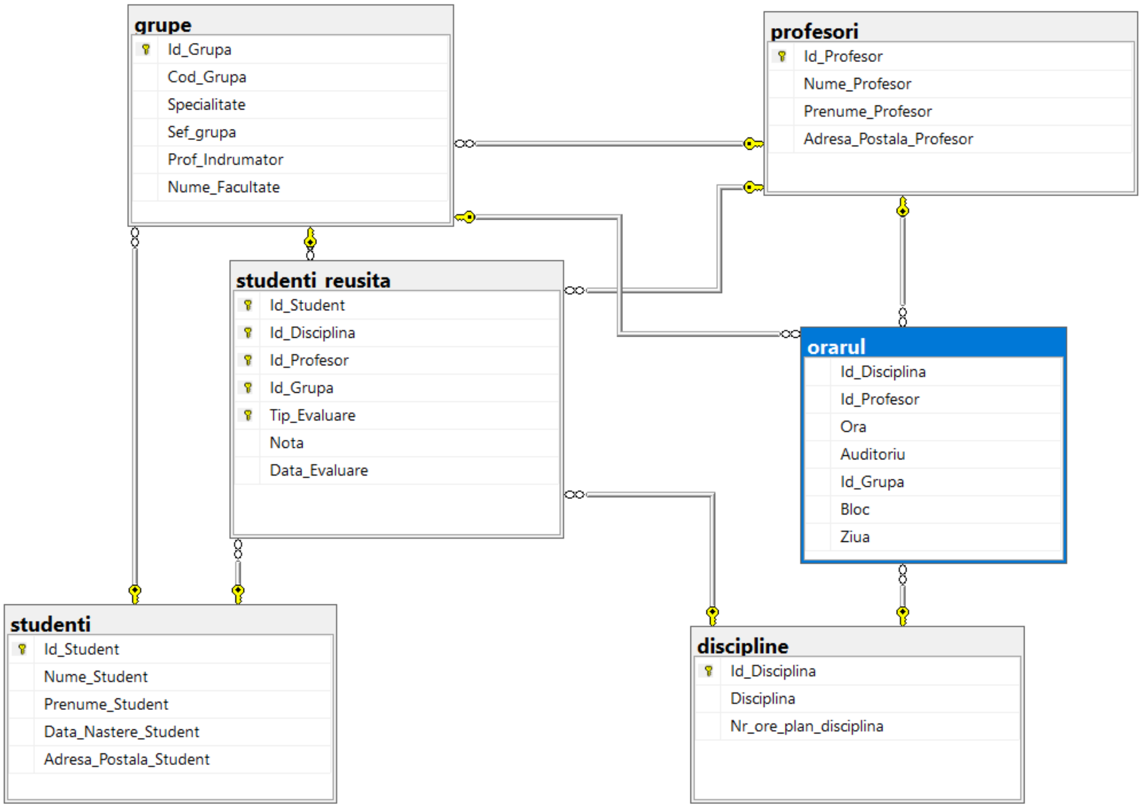
(General)	
Check Existing Data On Creation O	Yes
Tables And Columns Specification	
Foreign Key Base Table	grupe
Foreign Key Columns	Prof_Indrumator
Primary/Unique Key Base Table	profesori
Primary/Unique Key Columns	Id_Profesor
Database Designer	
Enforce For Replication	Yes
Enforce Foreign Key Constraint	Yes
INSERT And UPDATE Specification	

Know diagram looks like this

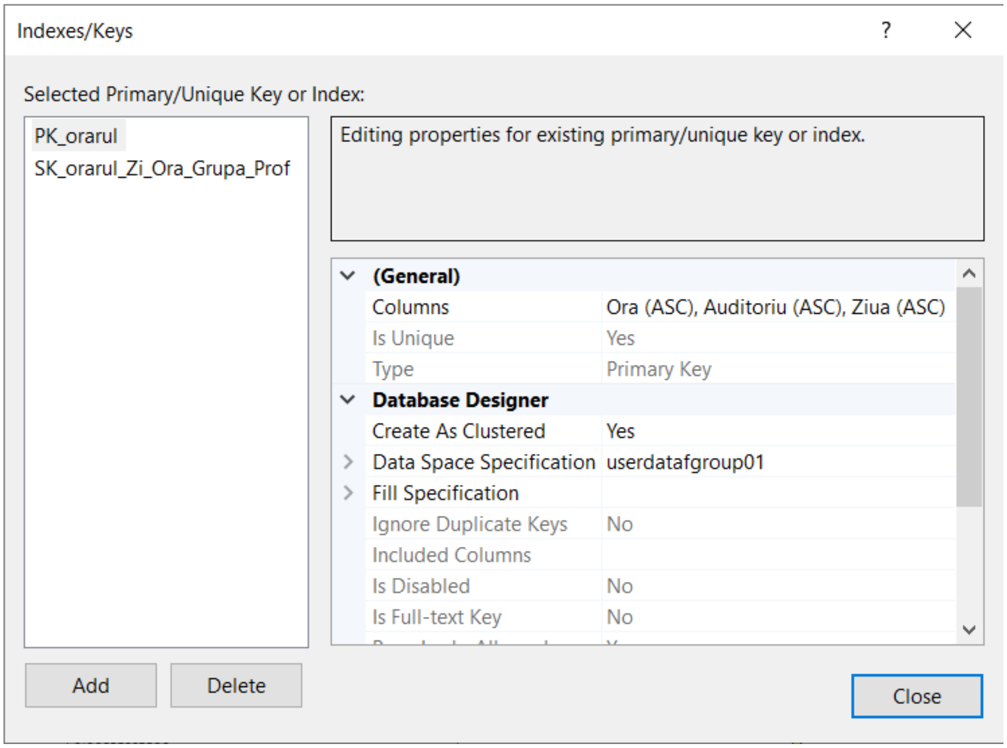


3.

Create new table with all required columns



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



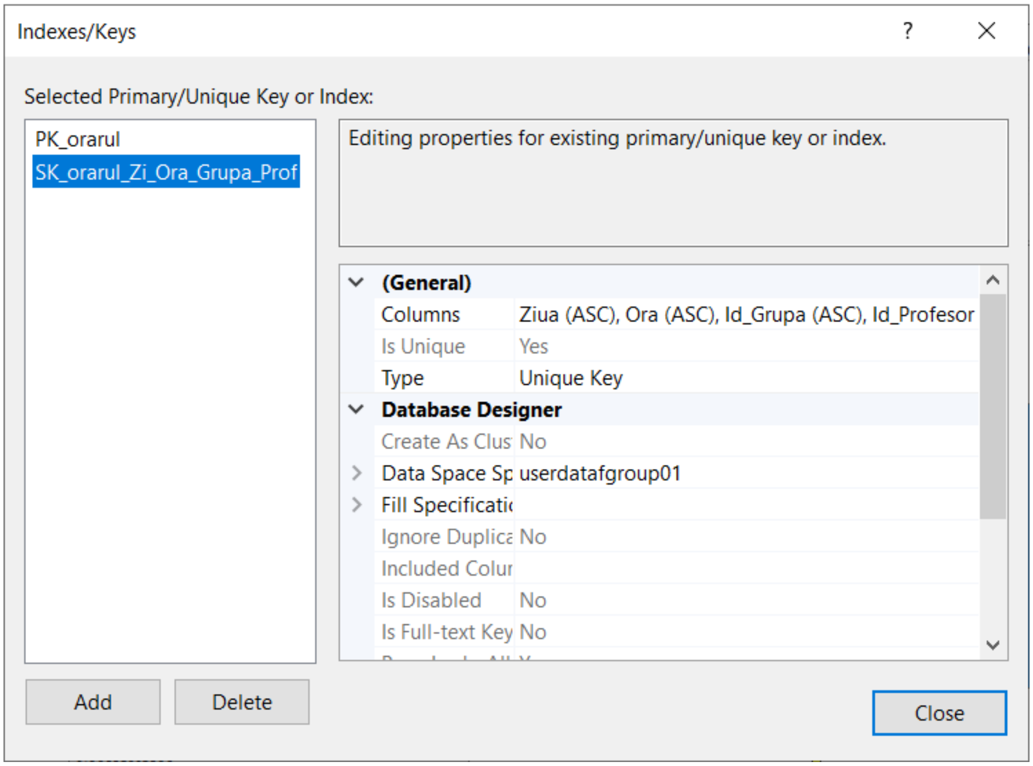
Selected Primary/Unique Key or Index:

PK_orarul
SK_orarul_Zi_Ora_Grupa_Prof

Editing properties for existing primary/unique key or index.

(General)	
Columns	Ora (ASC), Auditoriu (ASC), Ziua (ASC)
Is Unique	Yes
Type	Primary Key
Database Designer	
Create As Clustered	Yes
Data Space Specification	userdatafgroup01
Fill Specification	
Ignore Duplicate Keys	No
Included Columns	
Is Disabled	No
Is Full-text Key	No

And add two secondary keys:

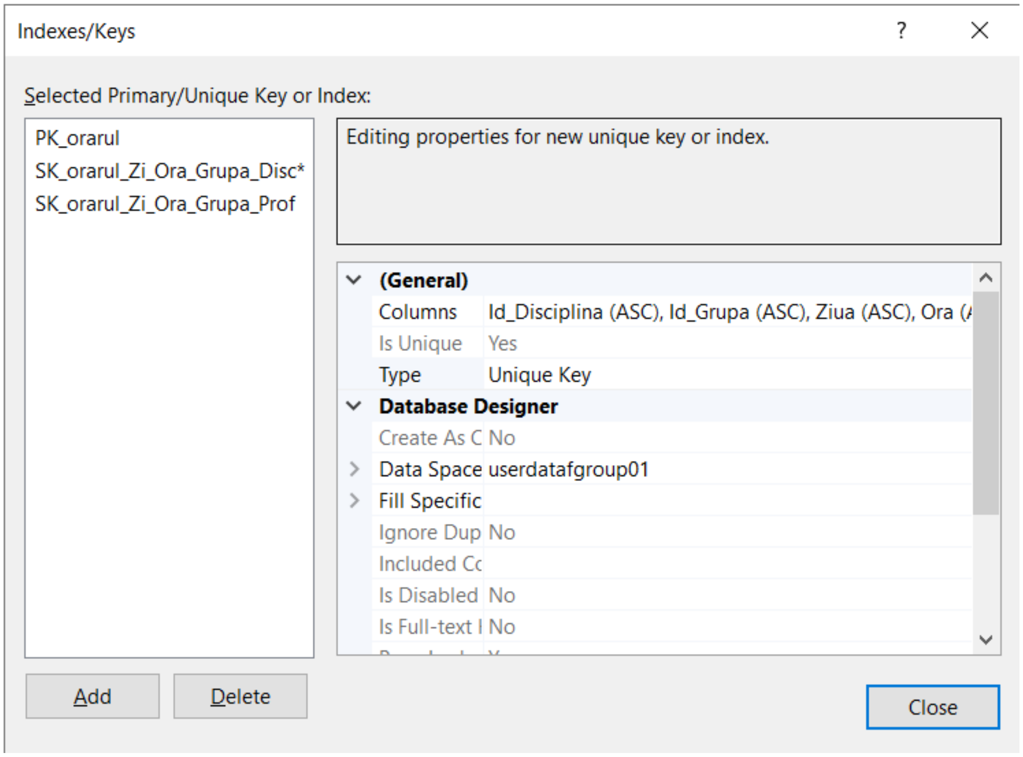


Selected Primary/Unique Key or Index:

PK_orarul
SK_orarul_Zi_Ora_Grupa_Prof

Editing properties for existing primary/unique key or index.

(General)
Columns Ziua (ASC), Ora (ASC), Id_Grupa (ASC), Id_Profesor
Is Unique Yes
Type Unique Key
Database Designer
Create As Clus No
Data Space Sp userdatafgroup01
Fill Specificati
Ignore Duplica No
Included Colur
Is Disabled No
Is Full-text Key No



Selected Primary/Unique Key or Index:

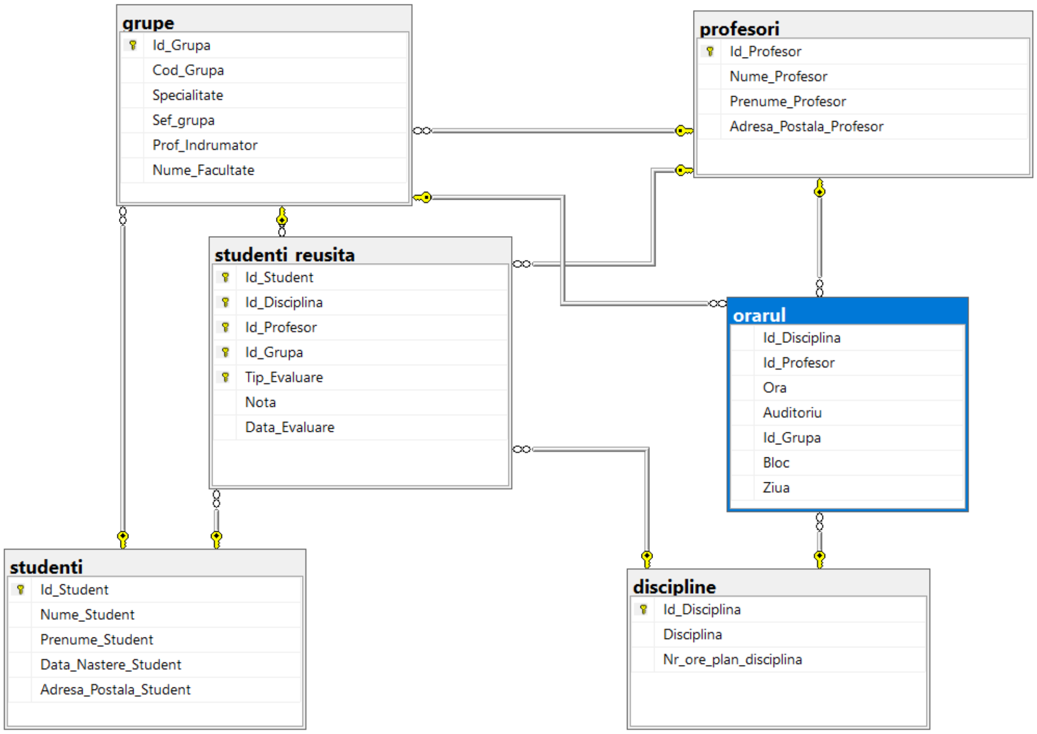
PK_orarul
SK_orarul_Zi_Ora_Grupa_Disc*
SK_orarul_Zi_Ora_Grupa_Prof

Editing properties for new unique key or index.

(General)
Columns Id_Disciplina (ASC), Id_Grupa (ASC), Ziua (ASC), Ora (
Is Unique Yes
Type Unique Key
Database Designer
Create As C No
Data Space userdatafgroup01
Fill Specific
Ignore Dup No
Included Cc
Is Disabled No
Is Full-text I No

5.

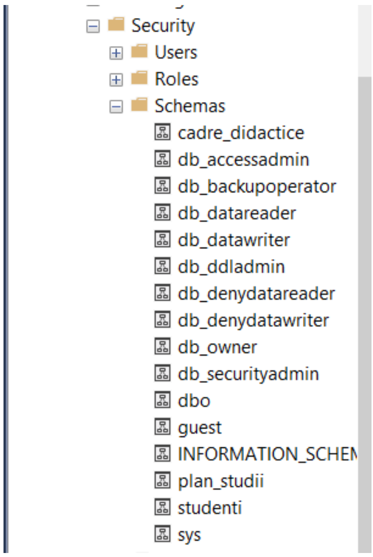
Added FK constraints for ‘orarul’ table



6.

Created required schemes:

```
CREATE SCHEMA cadre_didactice;
GO
CREATE SCHEMA plan_studii;
GO
CREATE SCHEMA studenti;
GO
```



```
⊟ Security
   ⊞ Users
   ⊞ Roles
   ⊟ Schemas
        cadre_didactice
        db_accessadmin
        db_backupoperator
        db_datareader
        db_datawriter
        db_ddladmin
        db_denydatareader
        db_denydatawriter
        db_owner
        db_securityadmin
        dbo
        guest
        INFORMATION_SCHEN
        plan_studii
        studenti
        sys
```

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.