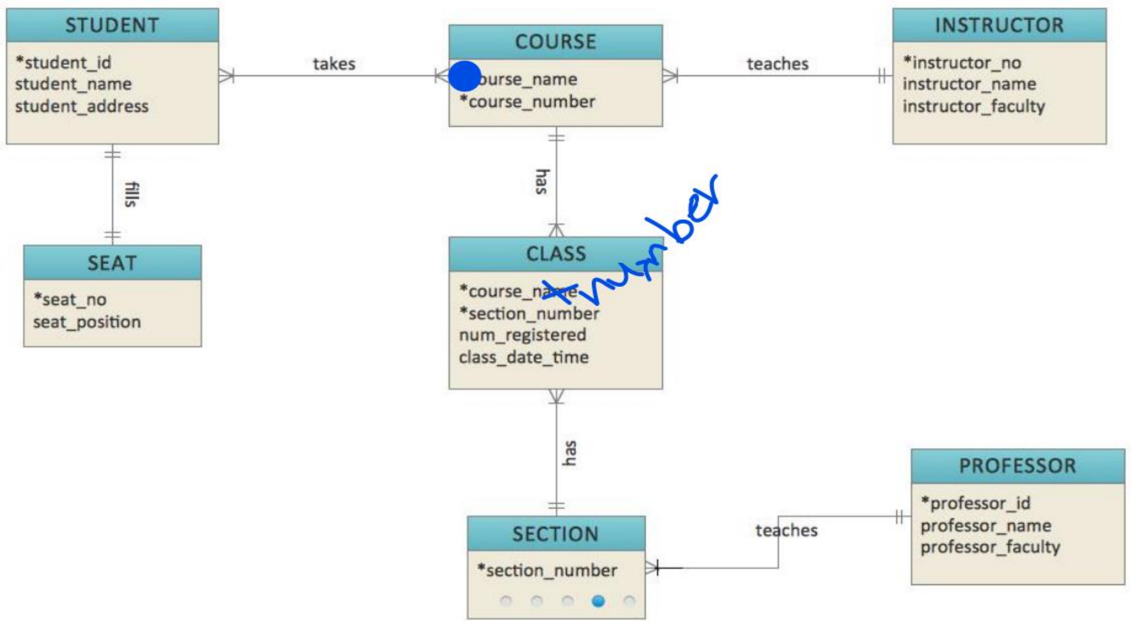
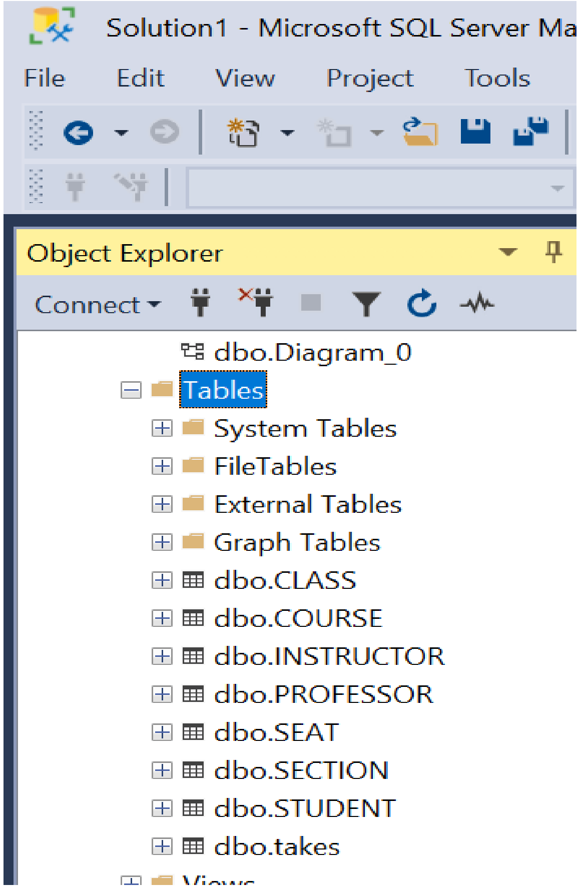
Amirkabir University of Technology (AUT)
Tehran, Iran
Lab Database, Computer Engineering Department
Report 1

Part 1) MSSMS ER model report:

After opening up a new project on MSSMS (name: Lab_ass_1) and connecting to server to the database, the entities (tables) and their relations should be designed according to the lab description file. Towards that end, first, the entities are made according to the ER diagram on page 16 of the lab description file.

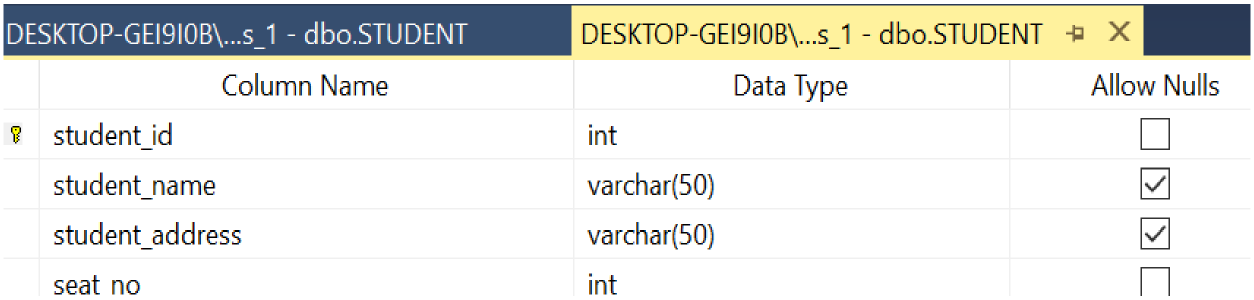


Next, their relations (many-to-one, one-to-one, and one-to-many) with their according foreign keys are specified, according to page 6 of the lab description file.

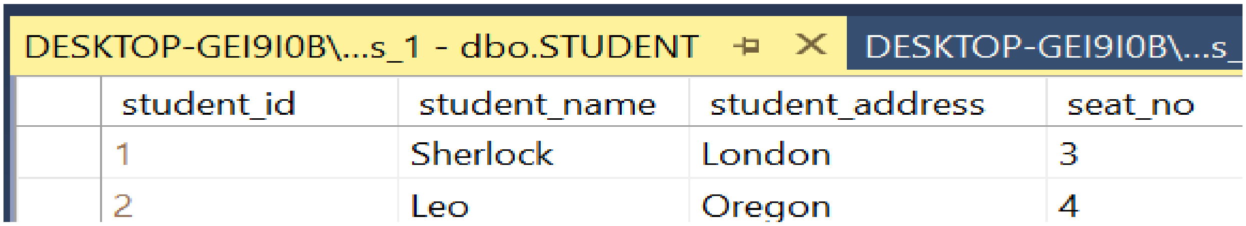


Here are the tables in detail:

1) **Table STUDENT:**

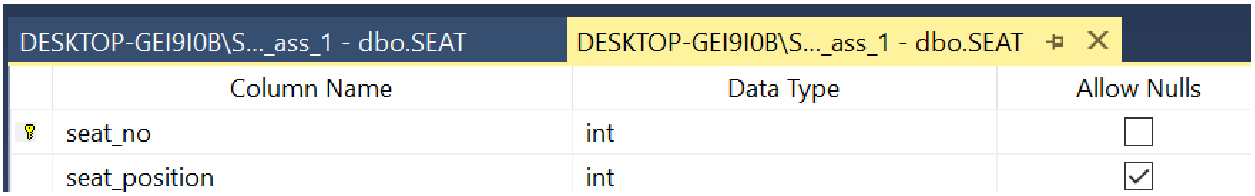

DESKTOP-GEI9I0B\...s_1 - dbo.STUDENT
DESKTOP-GEI9I0B\...s_1 - dbo.STUDENT

Column Name	Data Type	Allow Nulls
student_id	int	□
student_name	varchar(50)	☑
student_address	varchar(50)	☑
seat_no	int	□

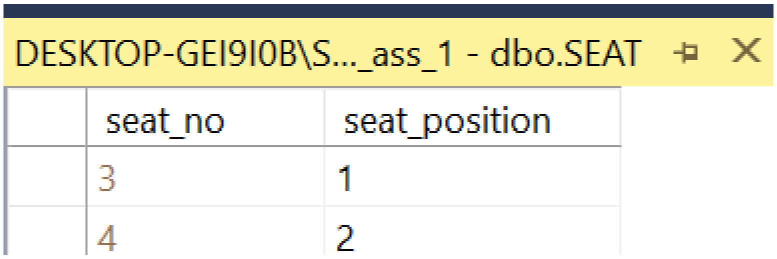


student_id	student_name	student_address	seat_no
1	Sherlock	London	3
2	Leo	Oregon	4

2) Table SEAT:

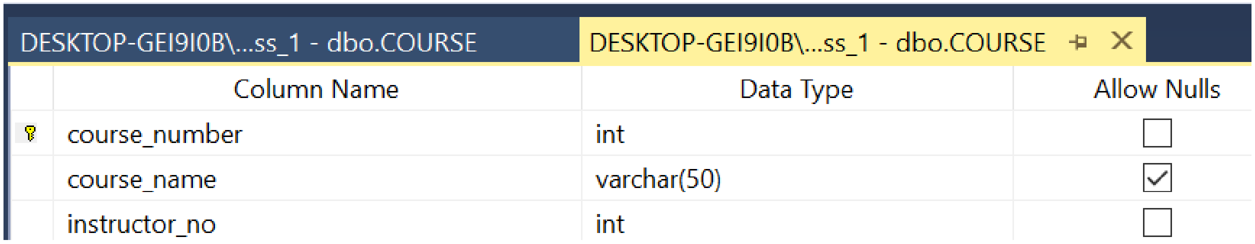

DESKTOP-GEI9I0B\S..._ass_1 - dbo.SEAT
DESKTOP-GEI9I0B\S..._ass_1 - dbo.SEAT

	Column Name	Data Type	Allow Nulls
🔑	seat_no	int	□
	seat_position	int	☑

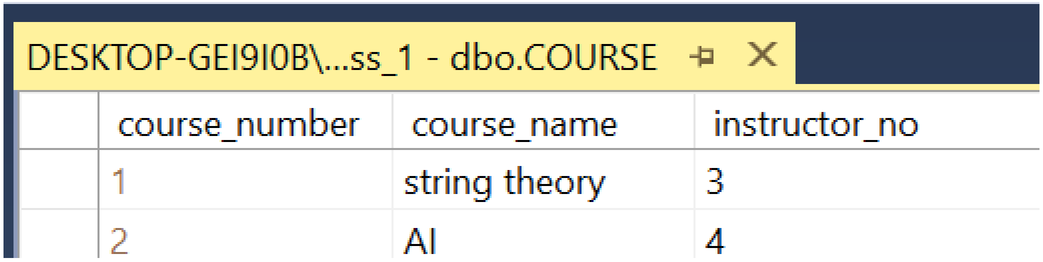


	seat_no	seat_position
	3	1
	4	2

3) Table COURSE:

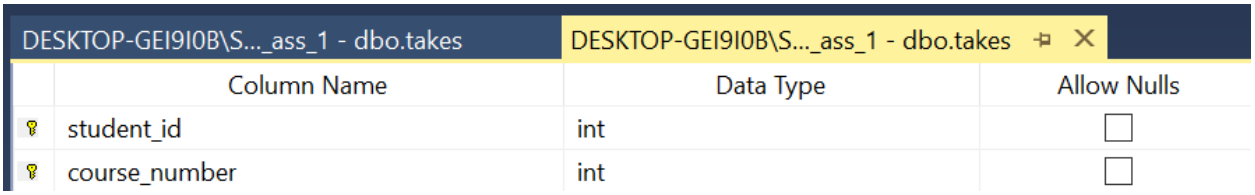

DESKTOP-GEI9I0B\...ss_1 - dbo.COURSE
DESKTOP-GEI9I0B\...ss_1 - dbo.COURSE

	Column Name	Data Type	Allow Nulls
🔑	course_number	int	□
	course_name	varchar(50)	☑
	instructor_no	int	□

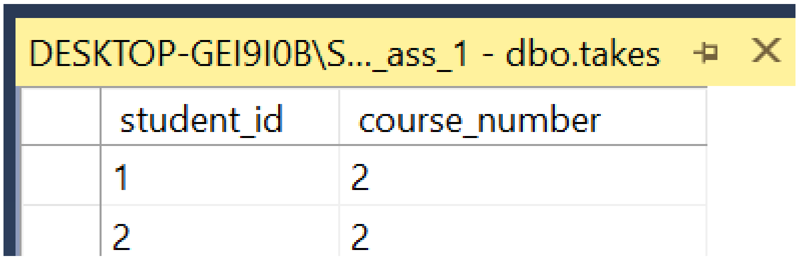


	course_number	course_name	instructor_no
	1	string theory	3
	2	AI	4

4) Relation Takes:

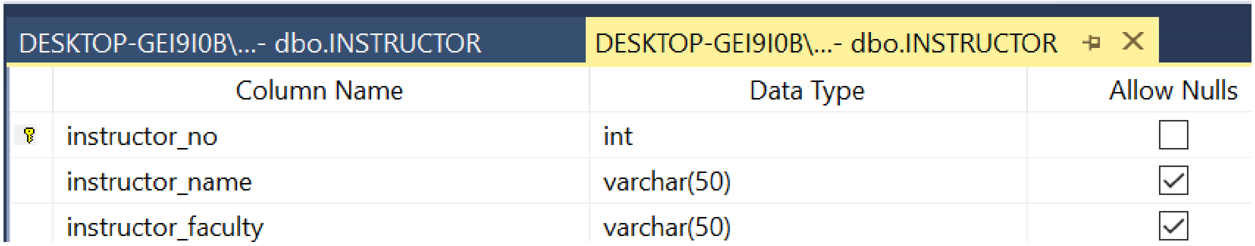

DESKTOP-GEI9I0B\S..._ass_1 - dbo.takes
DESKTOP-GEI9I0B\S..._ass_1 - dbo.takes

	Column Name	Data Type	Allow Nulls
🔑	student_id	int	☐
🔑	course_number	int	☐

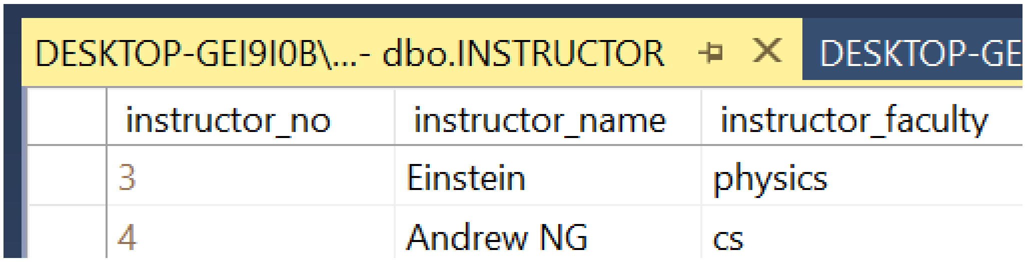


	student_id	course_number
	1	2
	2	2

5) Table INSTRUCTOR:

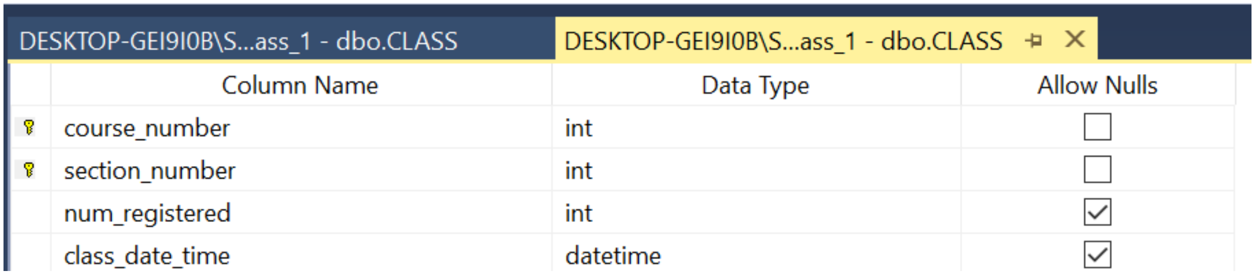

DESKTOP-GEI9I0B\...- dbo.INSTRUCTOR
DESKTOP-GEI9I0B\...- dbo.INSTRUCTOR

	Column Name	Data Type	Allow Nulls
🔑	instructor_no	int	☐
	instructor_name	varchar(50)	☑
	instructor_faculty	varchar(50)	☑

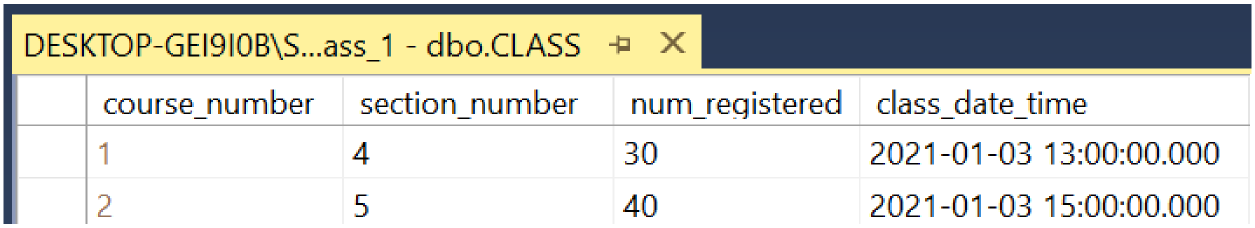


	instructor_no	instructor_name	instructor_faculty
	3	Einstein	physics
	4	Andrew NG	cs

6) Table CLASS:

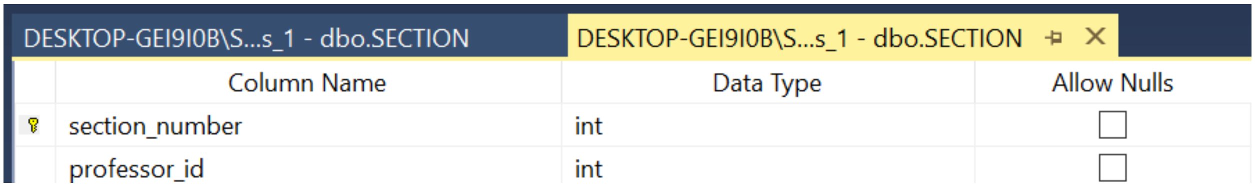

DESKTOP-GEI9I0B\S...ass_1 - dbo.CLASS
DESKTOP-GEI9I0B\S...ass_1 - dbo.CLASS

	Column Name	Data Type	Allow Nulls
🔑	course_number	int	☐
🔑	section_number	int	☐
	num_registered	int	☑
	class_date_time	datetime	☑

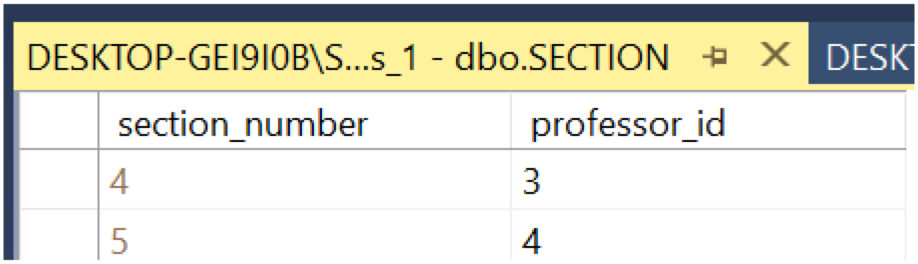


course_number	section_number	num_registered	class_date_time
1	4	30	2021-01-03 13:00:00.000
2	5	40	2021-01-03 15:00:00.000

7) Table SECTION:

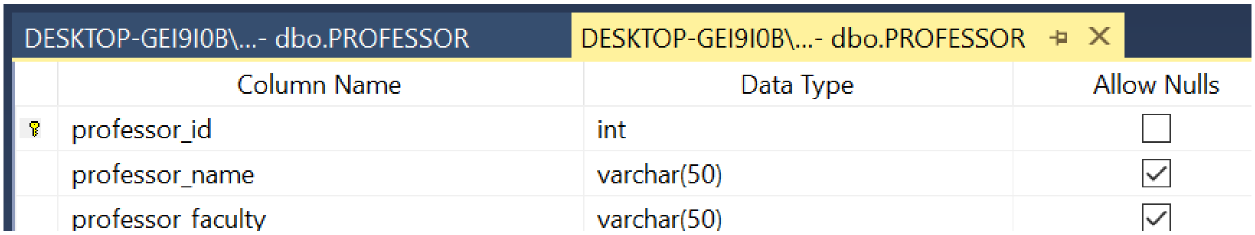

DESKTOP-GEI9I0B\S...s_1 - dbo.SECTION
DESKTOP-GEI9I0B\S...s_1 - dbo.SECTION

	Column Name	Data Type	Allow Nulls
🔑	section_number	int	☐
	professor_id	int	☐

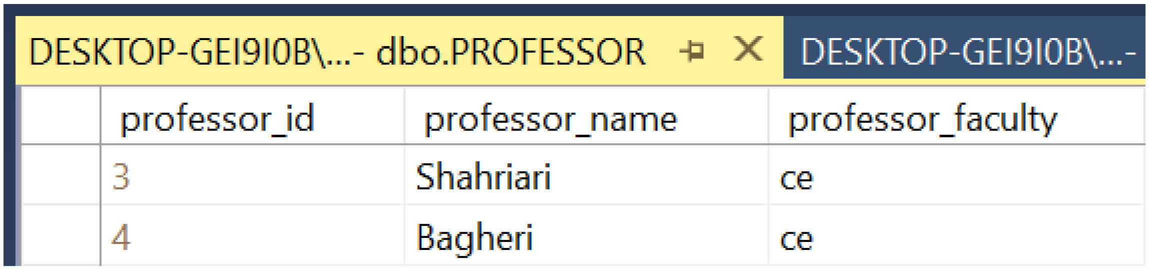


section_number	professor_id
4	3
5	4

8) Table PROFESSOR:


DESKTOP-GEI9I0B\...- dbo.PROFESSOR
DESKTOP-GEI9I0B\...- dbo.PROFESSOR

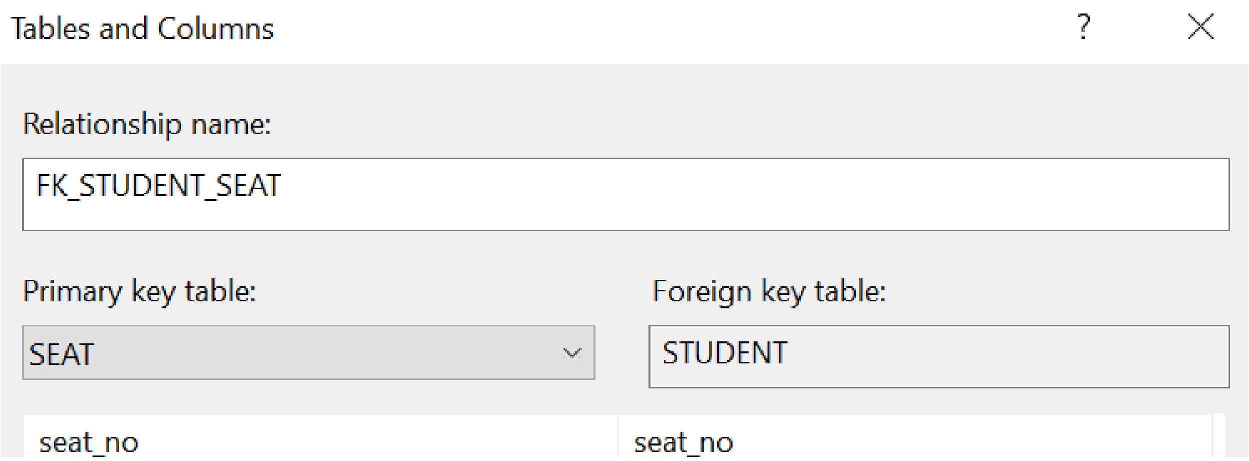
Column Name	Data Type	Allow Nulls
professor_id	int	☐
professor_name	varchar(50)	☑
professor_faculty	varchar(50)	☑



professor_id	professor_name	professor_faculty
3	Shahriari	ce
4	Bagheri	ce

Now, to see the foreign keys and relations in more detail:

1) **fills: (one-to-one relation)**

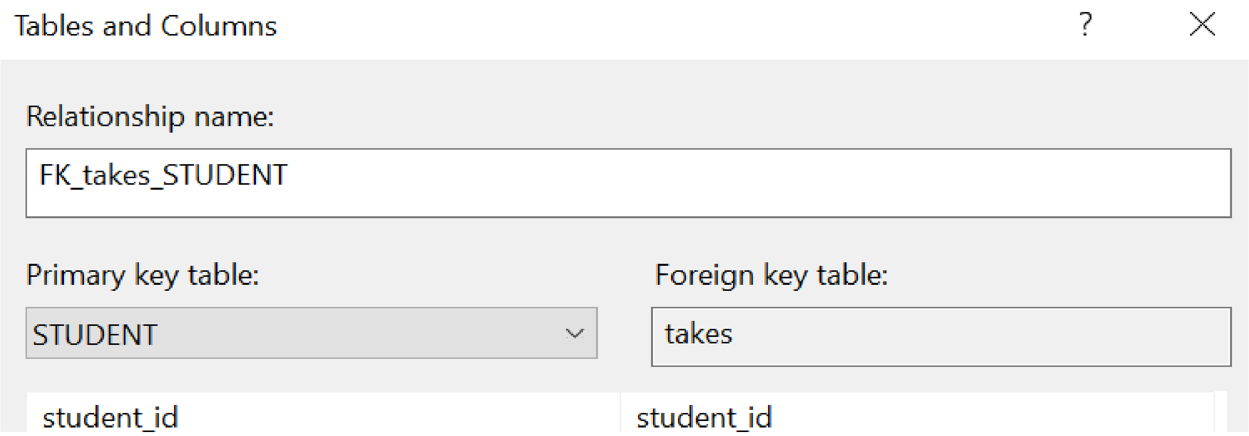


Relationship name:

FK_STUDENT_SEAT

Primary key table:	Foreign key table:
SEAT	STUDENT
seat_no	seat_no

2) takes: (many-to-many relation)



Tables and Columns

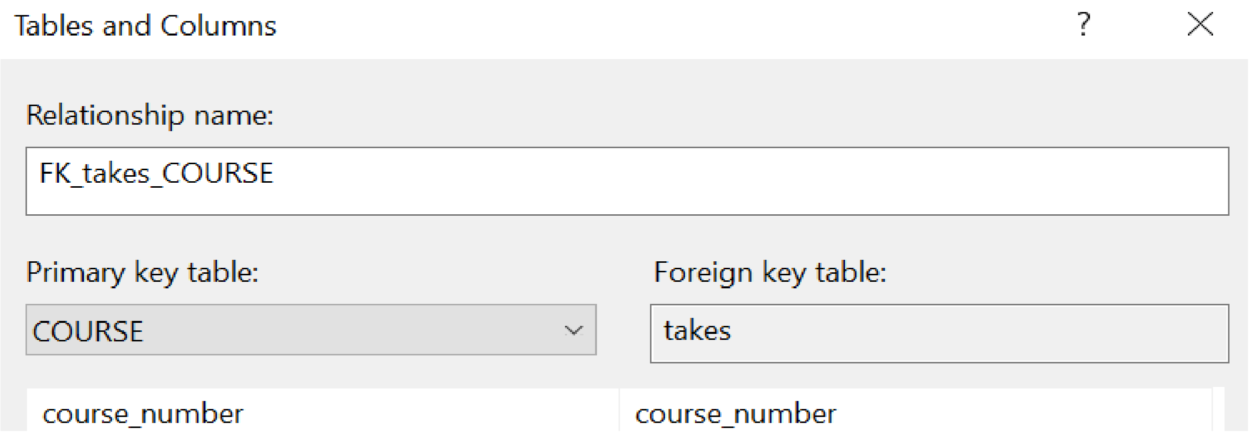
Relationship name:

FK_takes_STUDENT

Primary key table: STUDENT

Foreign key table: takes

student_id	student_id



Tables and Columns

Relationship name:

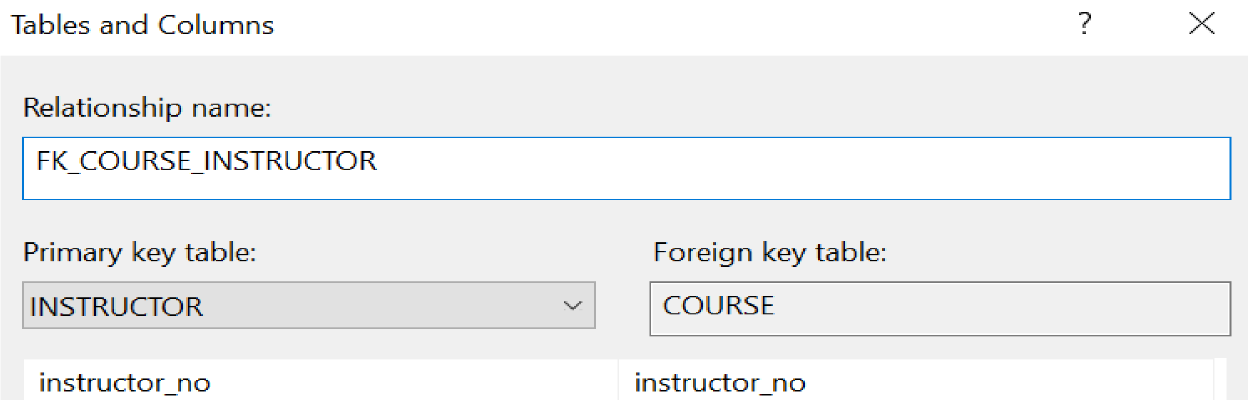
FK_takes_COURSE

Primary key table: COURSE

Foreign key table: takes

course_number	course_number

3) teaches: (many-to-one relation)



Tables and Columns

Relationship name:

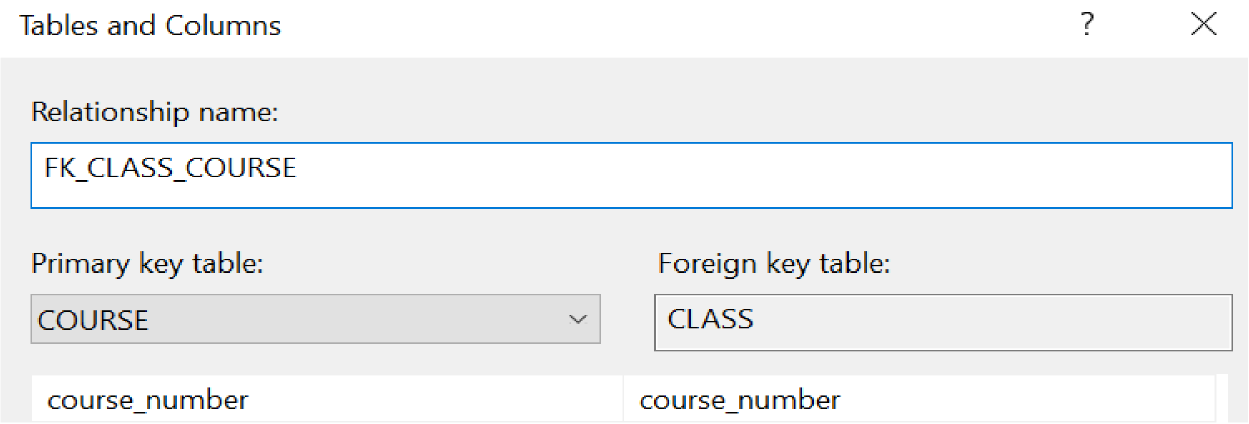
FK_COURSE_INSTRUCTOR

Primary key table: INSTRUCTOR

Foreign key table: COURSE

instructor_no	instructor_no

4) **has: (one-to-many relation)**

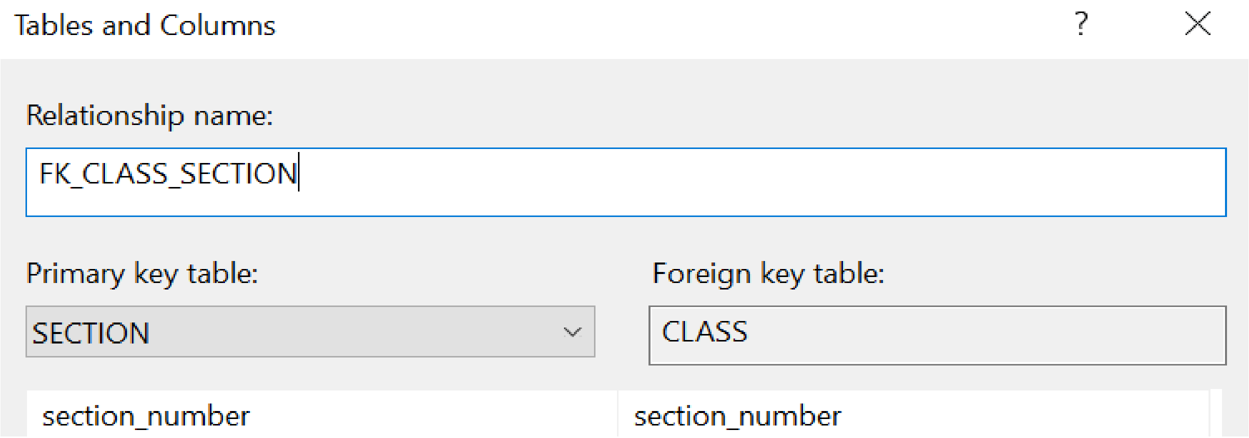

Tables and Columns

Relationship name:

FK_CLASS_COURSE

Primary key table:	Foreign key table:
COURSE	CLASS
course_number	course_number

5) **has: (many-to-one relation)**

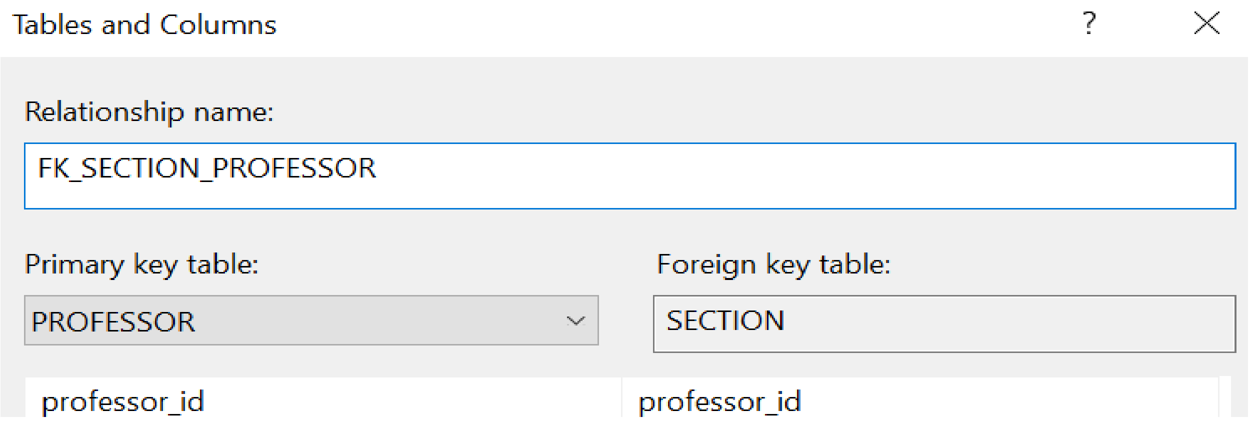

Tables and Columns

Relationship name:

FK_CLASS_SECTION

Primary key table:	Foreign key table:
SECTION	CLASS
section_number	section_number

6) **teaches: (many-to-one relation)**


Tables and Columns

Relationship name:

FK_SECTION_PROFESSOR

Primary key table:	Foreign key table:
PROFESSOR	SECTION
professor_id	professor_id

Part 2) Describe two different files with the extensions .mdf and .ldf in the data base:

.mdf: The file where data in the database is stored.

.ldf: The file where transactions and logs in the database is kept.

Every database in SQL Server stores at lease tow files with the extensions .mdf and .ldf where the former is for storing the data (data file), and the latter is for saving the logs (log file).

The data file contains information such as the tables, relations, indexes, procedures, and views. Common extension for these files are .mdf and also .ndf

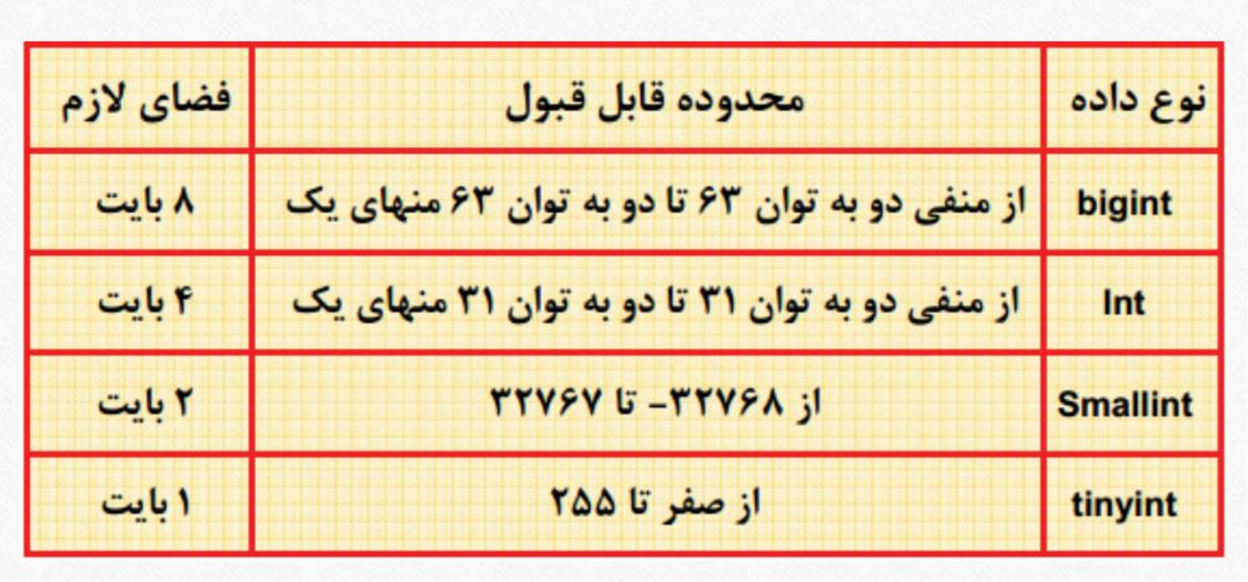
On the other hand, log files contain information that help recover transactions. Common extension for these files are .ldf

The path where these two files are being stored on my PC (and usually other PCs too) is:

C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA

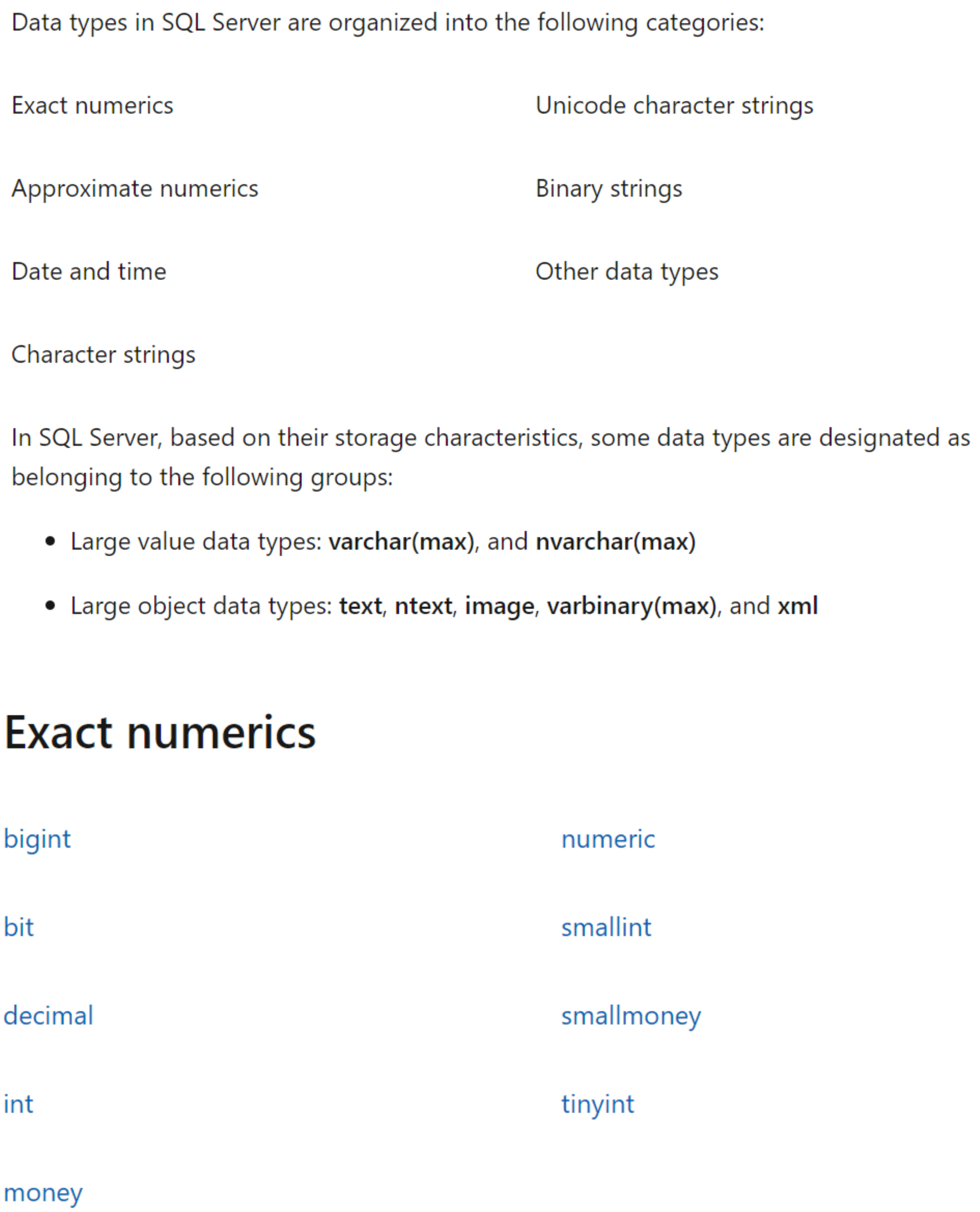
Part 3) Describe different data.

Before inserting data in the database, its type should have been specified in the tables (relations). That is because when the data is being inserted in the memory, both the database and the operating system need to know how much memory should be dedicated to each data instance. And if the inserted data type violates the pre-specified data type, then the database server will probably give errors.



فضای لازم	محدوده قابل قبول	نوع داده
۸ بایت	از منفی دو به توان ۶۳ تا دو به توان ۶۳ منهای یک	bigint
۴ بایت	از منفی دو به توان ۳۱ تا دو به توان ۳۱ منهای یک	Int
۲ بایت	از -۳۲۷۶۸ تا ۳۲۷۶۷	Smallint
۱ بایت	از صفر تا ۲۵۵	tinyint

Data Type Categories:



Data types in SQL Server are organized into the following categories:

Exact numerics

Approximate numerics

Date and time

Character strings

Unicode character strings

Binary strings

Other data types

In SQL Server, based on their storage characteristics, some data types are designated as belonging to the following groups:

- Large value data types: **varchar(max)**, and **nvarchar(max)**
- Large object data types: **text**, **ntext**, **image**, **varbinary(max)**, and **xml**

Exact numerics

bigint

bit

decimal

int

money

numeric

smallint

smallmoney

tinyint

Approximate numerics

- float
- real

Date and time

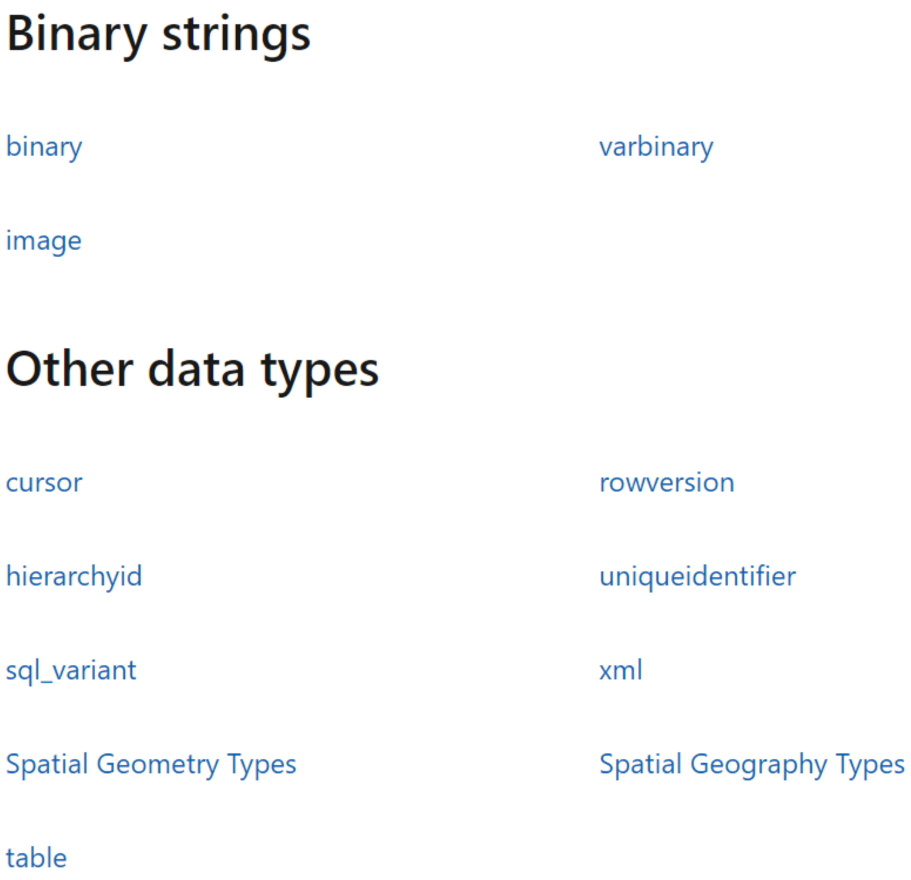
- date
- datetimeoffset
- datetime2
- smalldatetime
- datetime
- time

Character strings

- char
- varchar
- text

Unicode character strings

- nchar
- nvarchar
- ntext



Binary strings

binary

image

varbinary

Other data types

cursor

hierarchyid

sql_variant

Spatial Geometry Types

table

rowversion

uniqueidentifier

xml

Spatial Geography Types

Here were the different data and their types accepted by SQL.