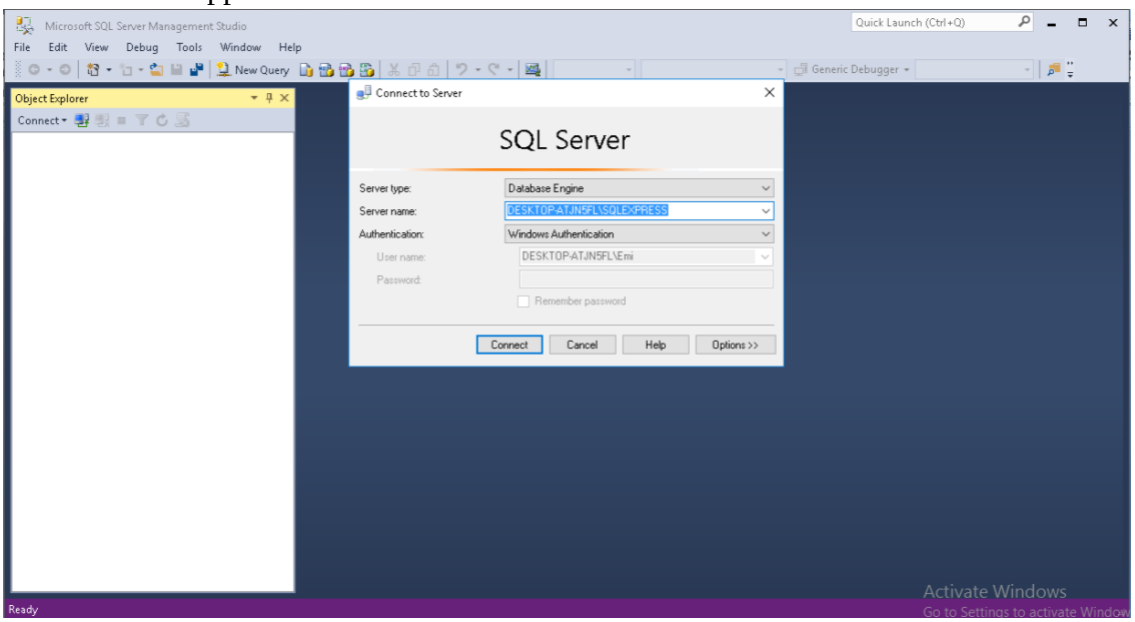
Problema 1. Proiectare bază de date - 2 săptămâni

Imaginaţi o aplicaţie simplă ce necesită o bază de date. Reprezentaţi datele aplicaţiei într-o structură relaţională şi implementaţi structura într-o bază de date Microsoft SQL Server. Baza de date trebuie să conţină *cel puţin 10 tabele* şi trebuie să implementeze *cel puţin o relaţie 1-m* şi *cel puţin o relaţie m-n*.

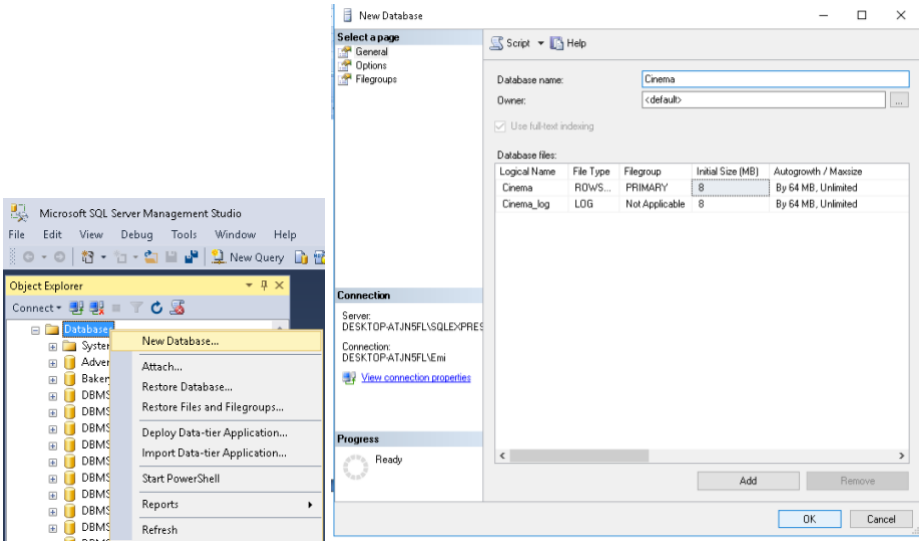
Înainte de a începe proiectarea trimiteţi un e-mail cu tema aleasă asistentului de laborator cu subiectul pentru care doriţi să creaţi baza de date (alături de o scurtă descriere). În cazul în care există un conflict cu tema propusă de un alt coleg, are câştig de cauză cel care a trimis primul e-mail cu propunerea.

Example: Cinema database

- Connect to the application

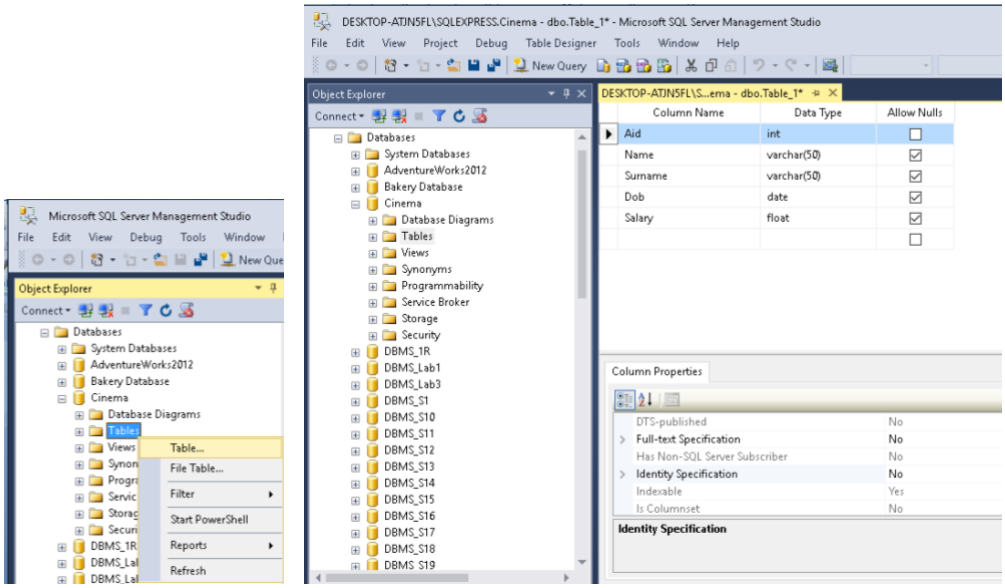


- Create the database: it will be located in C:\Program Files\Microsoft SQL Server\MSSQL13.SQLEXPRESS\MSSQL\DATA\



Then refresh on the Databases.

- Create the tables: primary key, relationships, foreign key, …

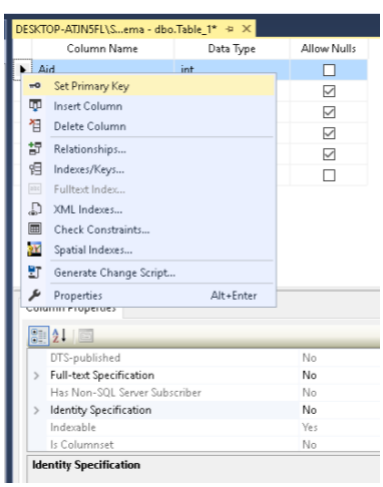


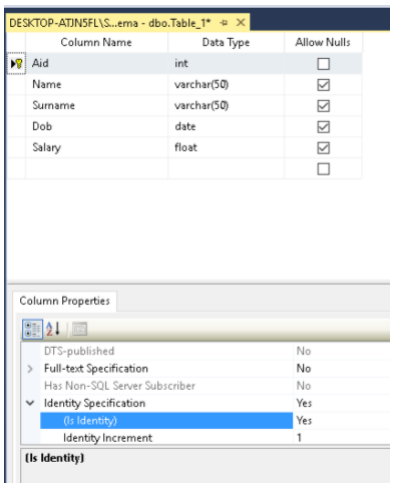
Set the primary key = the values that are unique for each record and not null.

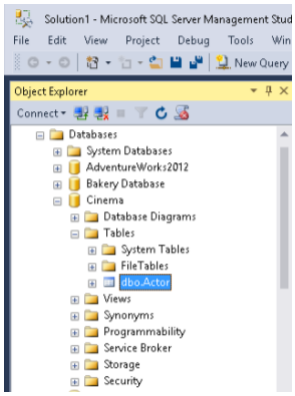
One can make the primary key to be identity (autoincrement). Save the table.

```
CREATE TABLE Actor(
        Aid int IDENTITY(1,1) NOT NULL,
        Name varchar(50) NULL,
        Surname varchar(50) NULL,
        Dob date NULL,
        Salary float NULL,
        CONSTRAINT PK_Actor PRIMARY KEY
)
```

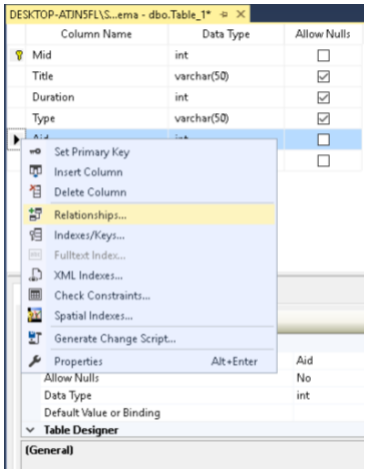
OR by design view:

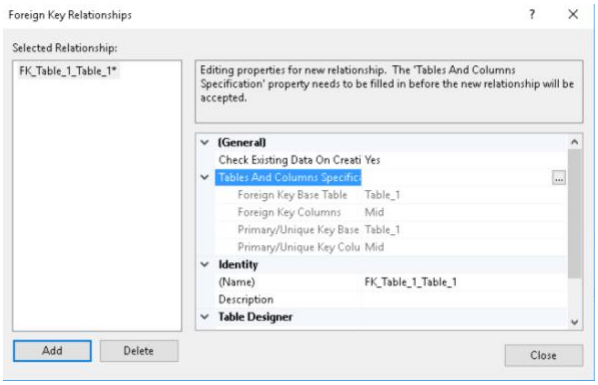


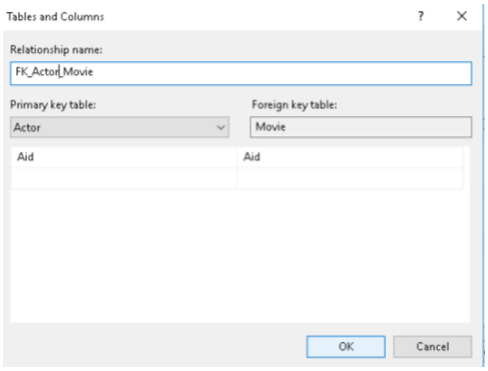


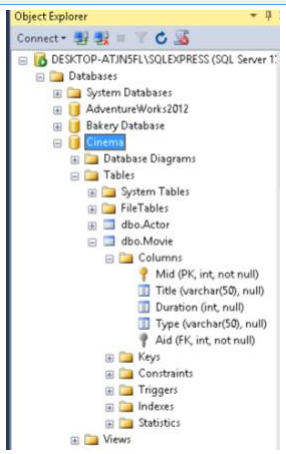


Foreign key = the primary key from the table in which is in a relationship, of the same type (int, varchar, ..) and the same values (but the values can be used for one or more records) and not null.





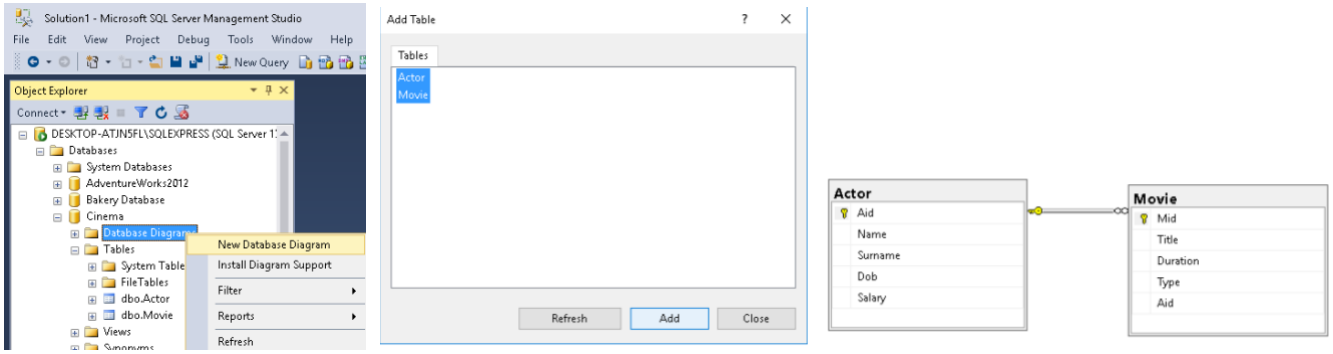





```
CREATE TABLE Movie(
        Mid int NOT NULL PRIMARY KEY,
        Title varchar(50) NULL,
        Duration int NULL,
        Type varchar(50) NULL,
        Aid int NOT NULL,
        CONSTRAINT FK_Actor_Movie FOREIGN KEY(Aid) REFERENCES Actor (Aid)
```

)

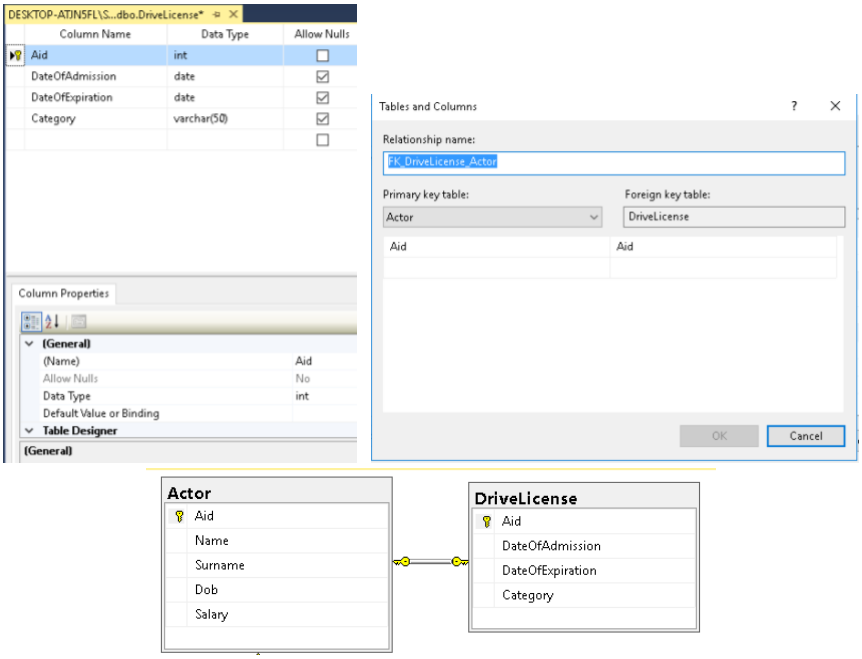
In the diagram can see directly the relation



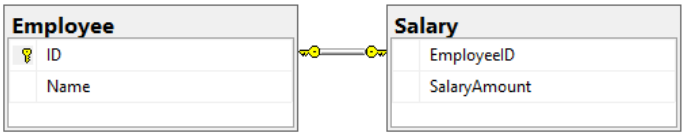
Relationships between the tables: 1-1, 1-n, m-n

Relation 1-1 (1 to 1): Actor-DriveLicense, Actor-IdentityCard, Cinema-Director(Manager), Employee-Salary

It is realized with the primary keys from the 2 tables. (Set the foreign key as a primary key, and then set the relationship on both primary key fields. You should see a key sign on both ends of the relationship line. This represents a one to one relationship.)



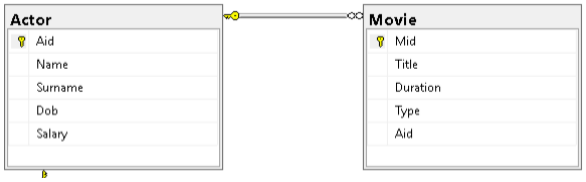
Create table Employee (ID int primary key, Name varchar(50))	Create table Salary (EmployeeID int primary key, SalaryAmount int, Constraint FK_Salary_Employee Foreign key(EmployeeID) References Employee(ID))



Address and Telephone (PhoneNumber) are tables!!! NOT fields (attributes)

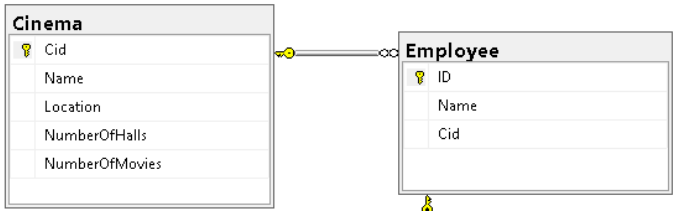
Relation 1-m (one to many)

Actor - Movie (An Actor play in one or more Movie) – Aid (from table Actor) is the primary key in Actor and is part *1 of the relationship* and Aid (from table Movie) is the foreign key in Movie and is part *m of the relationship*.



The relationship is made from the table where is the foreign key!

Cinema – Employee (In a Cinema can work one or more Employee, one or more Employee can work in one Cinema) – primary key in Cinema, foreign key in Employee

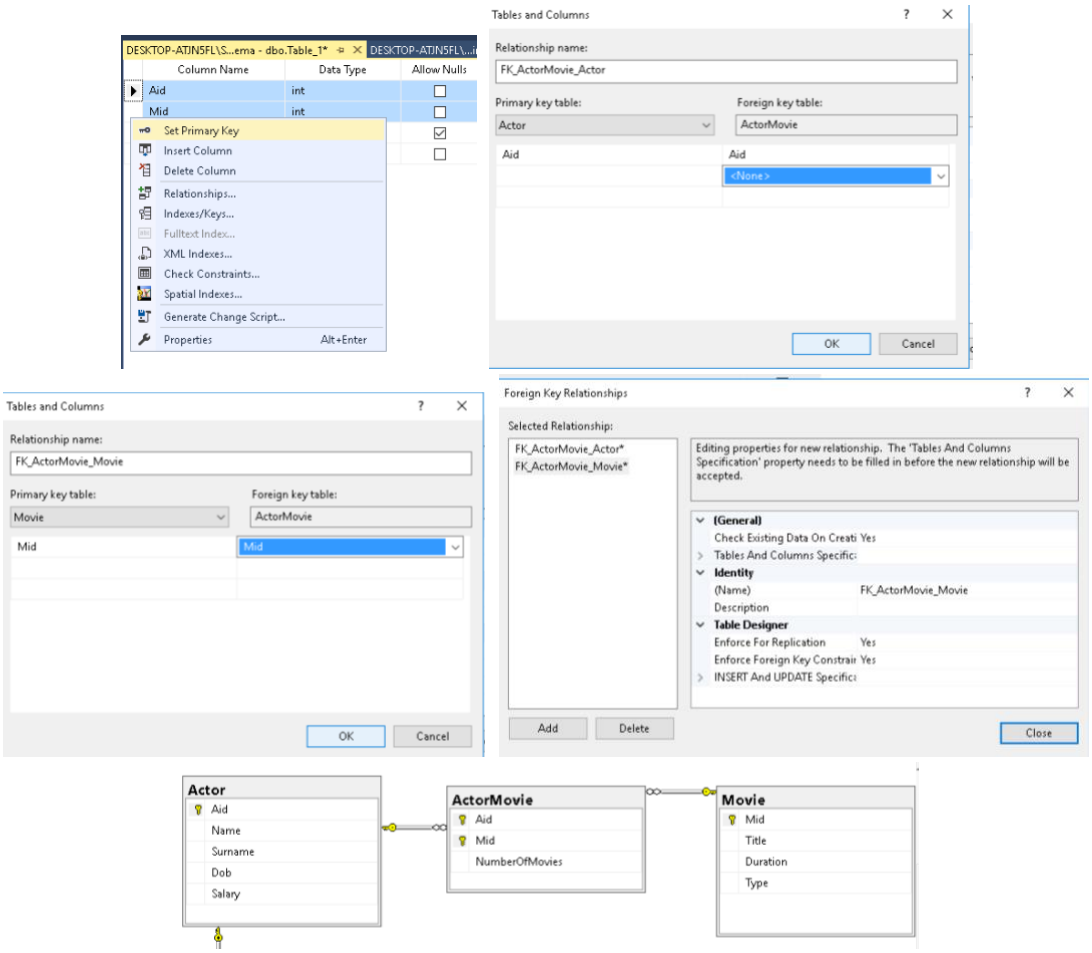


Relation m-m (many to many)

- 2 relations 1-m – this relation can be implemented through an intermediate table.

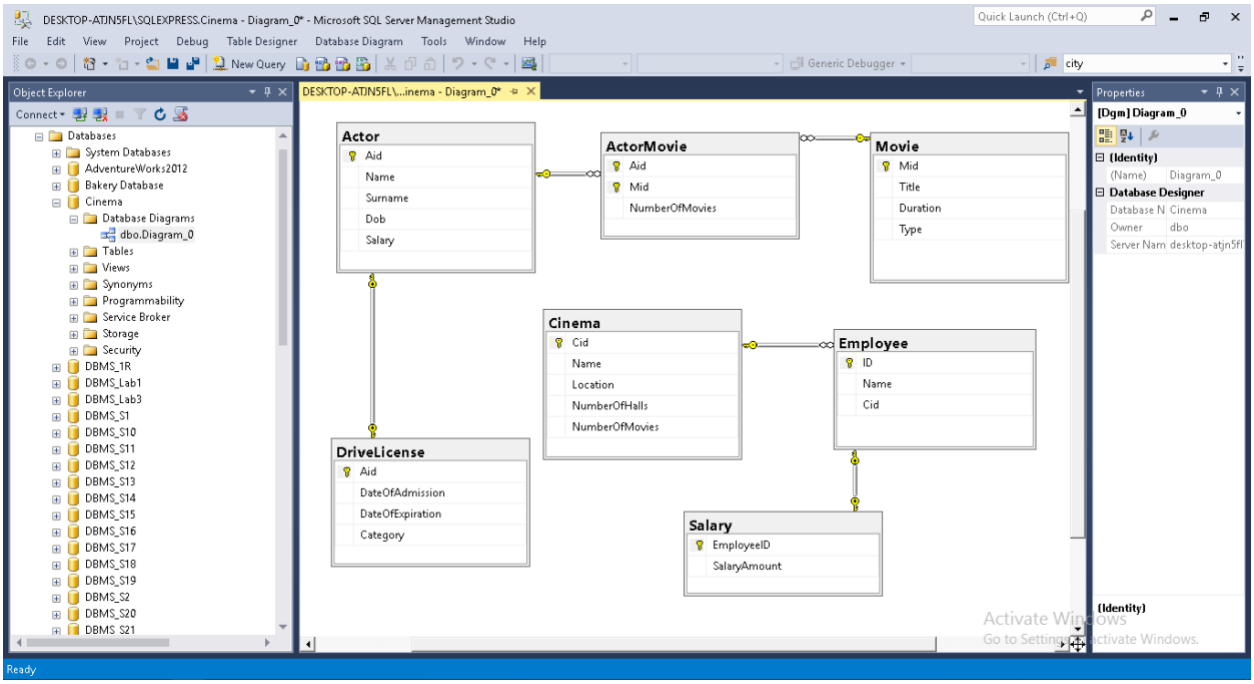
Actor – Movie (one or more Actor can play in one or more Movie) – we will have 2 relationships Actor – ActorMovie and ActorMovie – Movie, that are 1-m. ActorMovie is the intermediate table and contain the primary keys of the 2 tables (Actor, Movie) as primary key of this table, that are also foreign keys in the relationships.

- Create primary key “in pair”

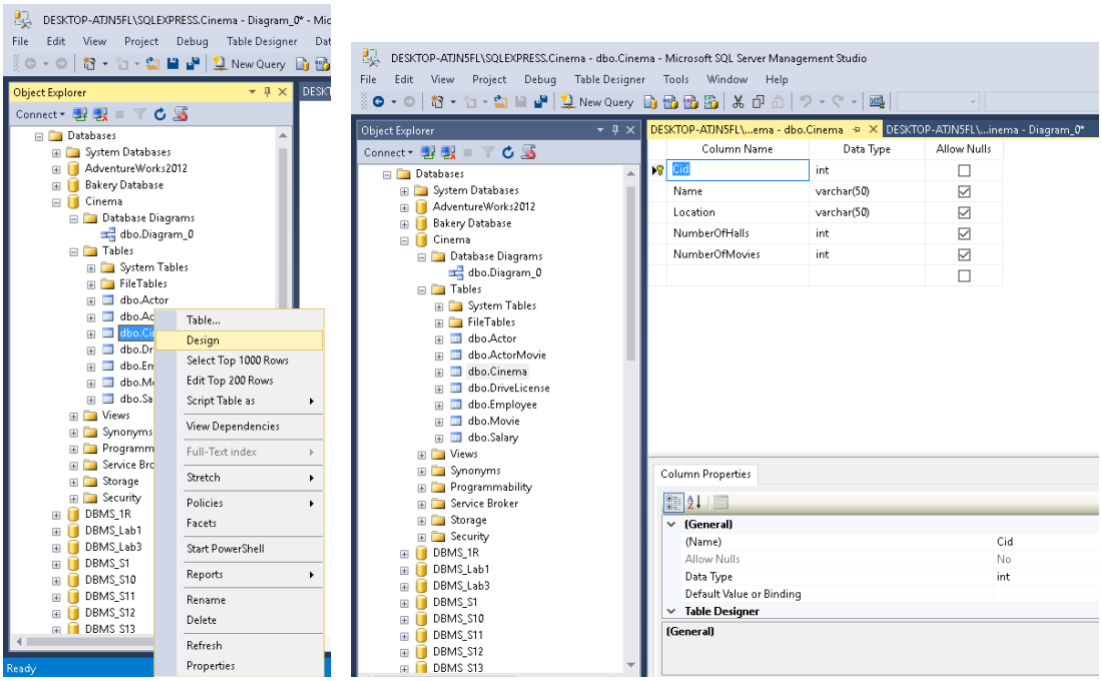


- It is not really correct to create a primary key that does not contain the primary keys of the 2 tables involved, because someone can have duplicates. For example, now we have pairs like (1,1), (1,2),…, but otherwise we could have 1 (1,1), 2 (1,1), ….

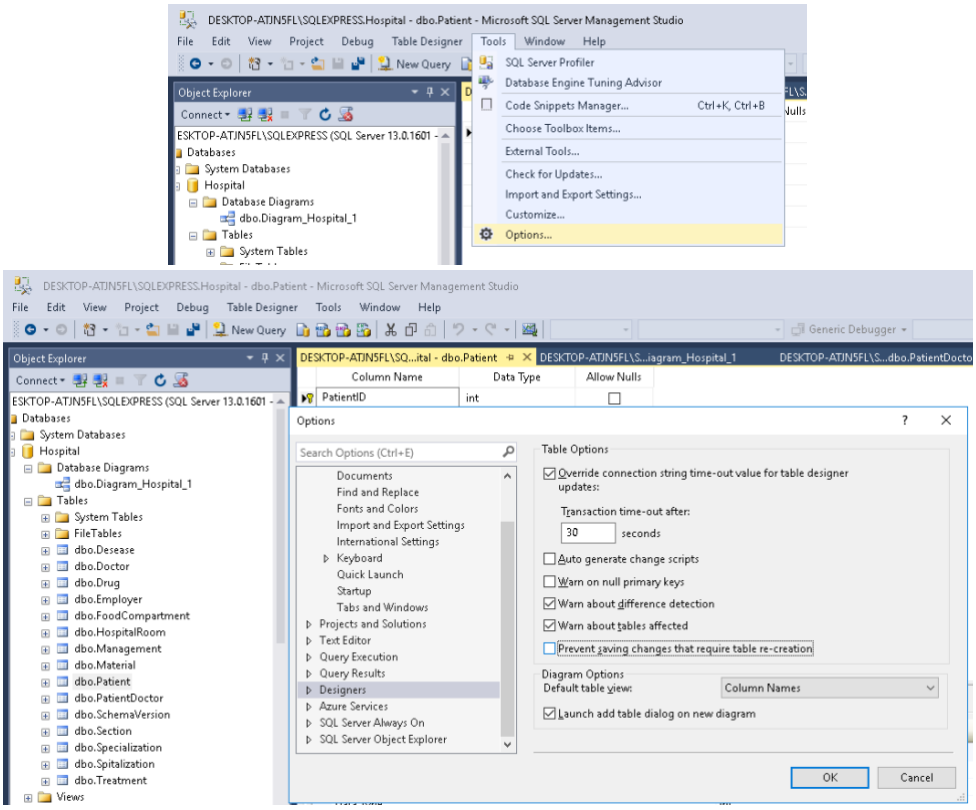
Diagram:



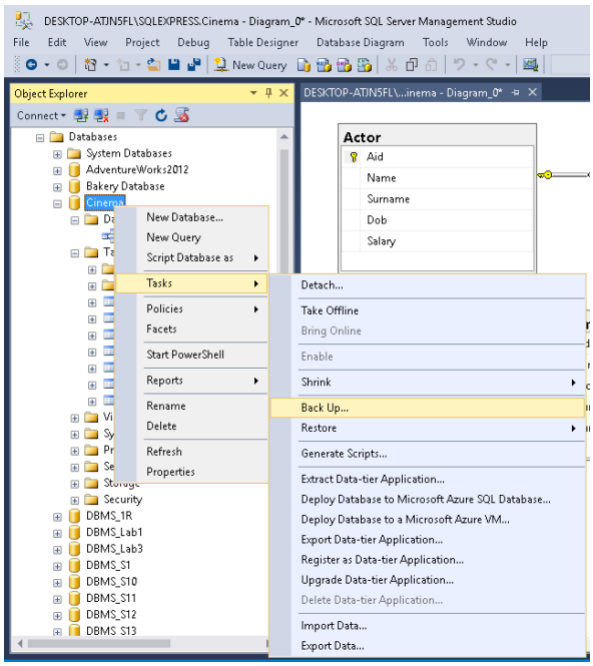
Modifications on tables (add new columns…)

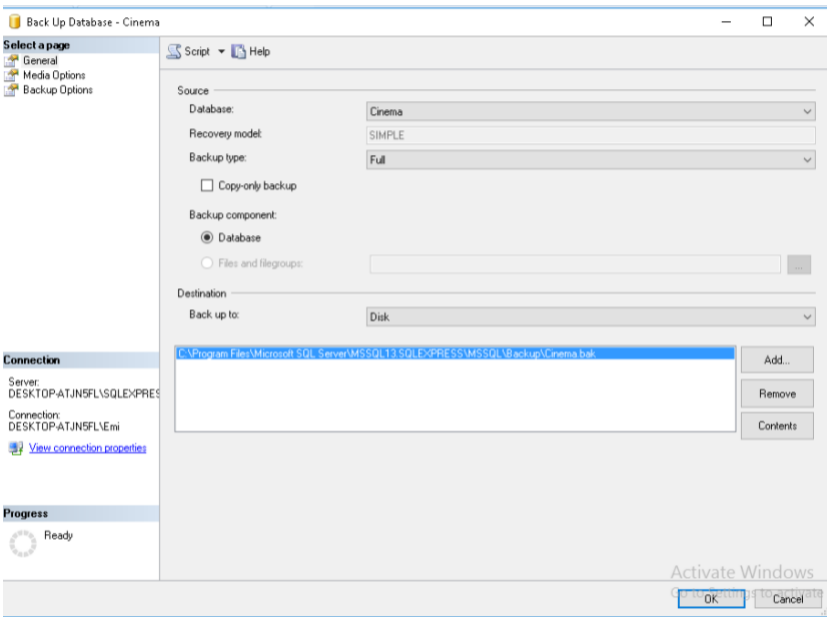


If someone cannot save: Tools menu -> Options -> Designers -> Uncheck - Prevent saving changes that require table re-creation -> ok



Back-up the database




Back Up Database - Cinema
Select a page
General
Media Options
Backup Options
Script
Help
Source
Database:
Cinema
Recovery model:
SIMPLE
Backup type:
Full
Copy-only backup
Backup component:
Database
Files and filegroups:
Destination
Back up to:
Disk
C:\Program Files\Microsoft SQL Server\MSSQL13.SQLEXPRESS\MSSQL\Backup\Cinema.bak
Add...
Remove
Contents
Connection
Server:
DESKTOP-ATJN5FL\SQLEXPRES
Connection:
DESKTOP-ATJN5FL\Emi
View connection properties
Progress
Ready
Activate Windows
OK
Cancel