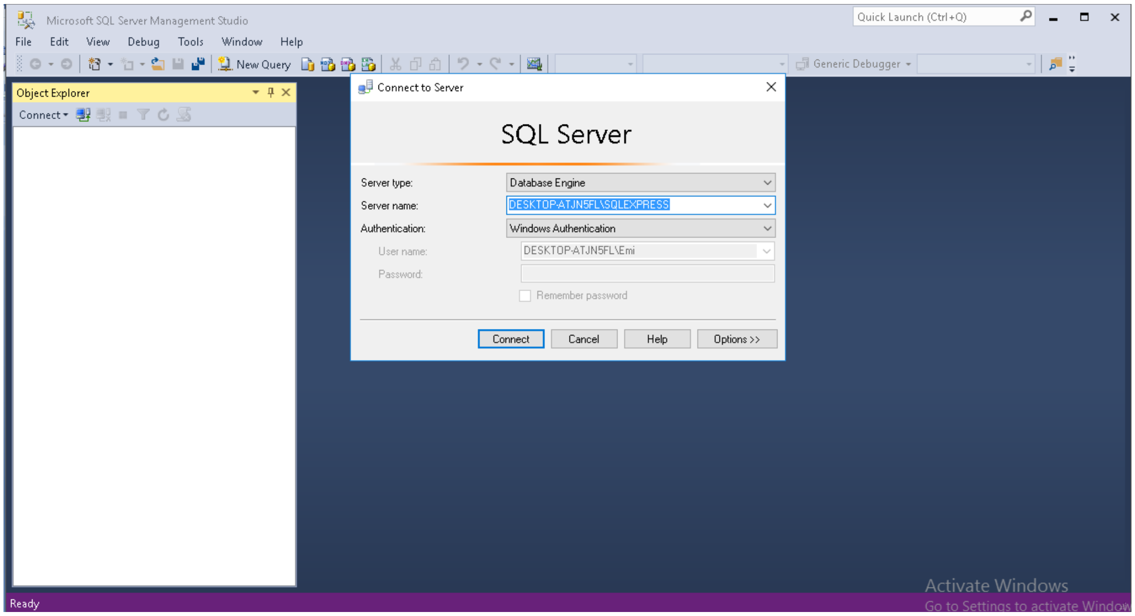
Problem 1. Database design - 2 weeks

Imagine a simple application that would require a database. Represent the application data in a relational structure and implement the structure in an SQL Server database. The database should contain *at least 10 tables* and should implement *at least one 1 to many relationship* and one *many to many relationship*.
Before you start working send an e-mail to laboratory assistant with the subject for which you want to create a database (a short description will be appreciated). In case of conflicts, first in the list wins, so please check existing entries here to make sure your subject is not taken.

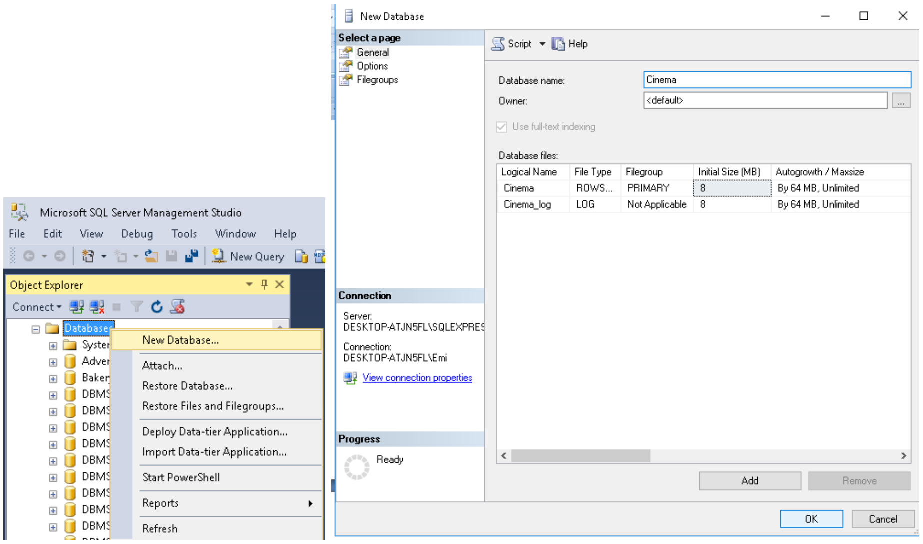
Example: Cinema database

- Connect to the application



Server type: Database Engine
Server name: DESKTOP-ATJN5FL\SQLEXPRESS
Authentication: Windows Authentication
User name: DESKTOP-ATJN5FL\Emi
Password:
□ Remember password

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



Database name: Cinema
Owner: <default>
☑ Use full-text indexing

Database files:

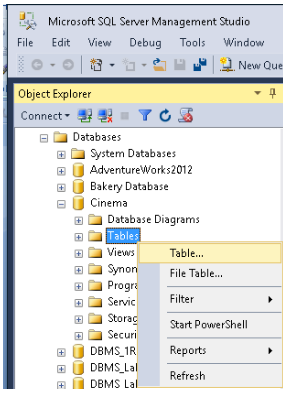
Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
Cinema	ROWS...	PRIMARY	8	By 64 MB, Unlimited
Cinema_log	LOG	Not Applicable	8	By 64 MB, Unlimited

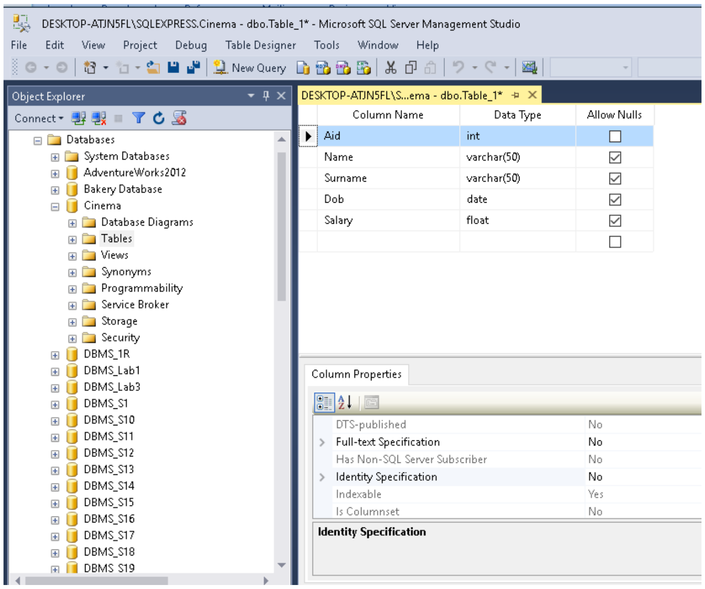
Server:
DESKTOP-ATJN5FL\SQLEXPRES

Connection:
DESKTOP-ATJN5FL\Emi

Then refresh on the Databases.

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





Column Name	Data Type	Allow Nulls
Aid	int	☐
Name	varchar(50)	☑
Surname	varchar(50)	☑
Dob	date	☑
Salary	float	☑
		☐

Column Properties

DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	No
Indexable	Yes
Is Columnset	No

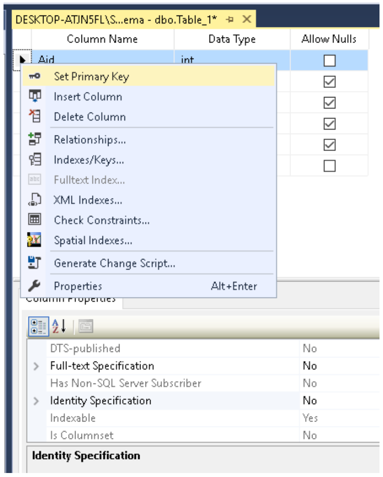
Identity Specification

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.

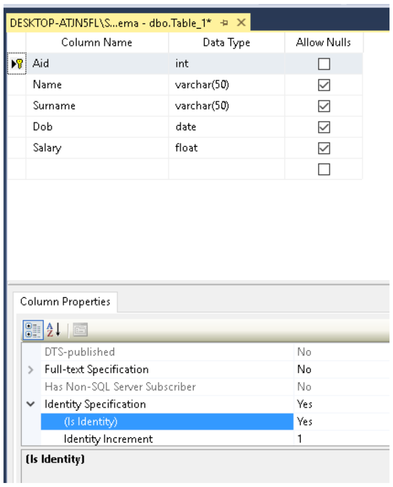
```
CREATE TABLE Actor(
        Aid int IDENTITY(1,1) NOT NULL,
        Name varchar(50) NULL,
        Surname varchar(50) NULL,
        Dob date NULL,
        Salary float NULL,
        CONSTRAINT PK_Actor PRIMARY KEY
)
```

OR by design view:



DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	No
Indexable	Yes
Is Columnset	No

Identity Specification

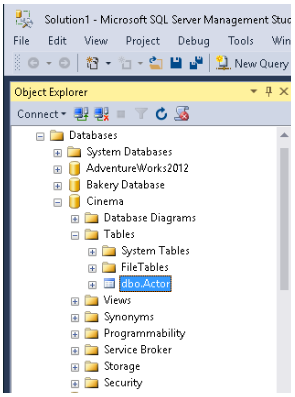


Column Name	Data Type	Allow Nulls
Aid	int	☐
Name	varchar(50)	☑
Surname	varchar(50)	☑
Dob	date	☑
Salary	float	☑
		☐

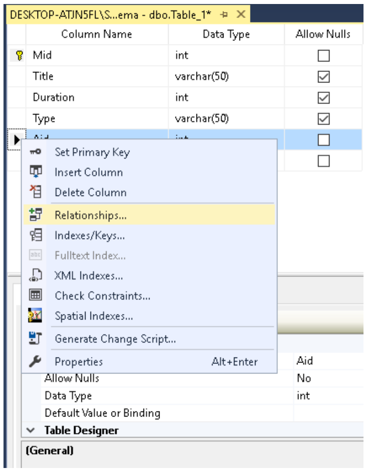
Column Properties

DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1

(Is Identity)



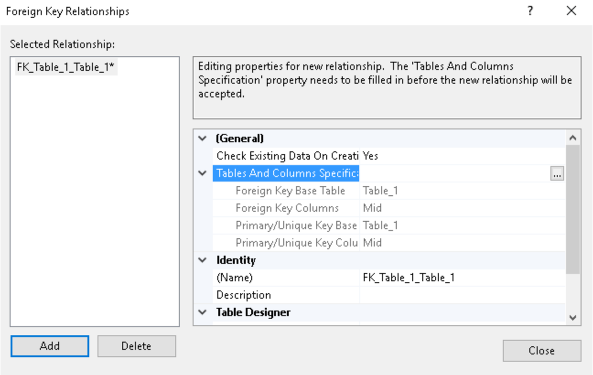
Foreign key = the primary key from the table in which is in a relationship, of the same type (int, varchar, ..) and the same values (but the values can be used for one or more records) and not null.


DESKTOP-ATJN5FL\S...ema - dbo.Table_1*

Column Name	Data Type	Allow Nulls
Mid	int	□
Title	varchar(50)	☑
Duration	int	☑
Type	varchar(50)	☑
Aid	int	□
		□

(Name)	Aid
Allow Nulls	No
Data Type	int
Default Value or Binding	

Table Designer
(General)

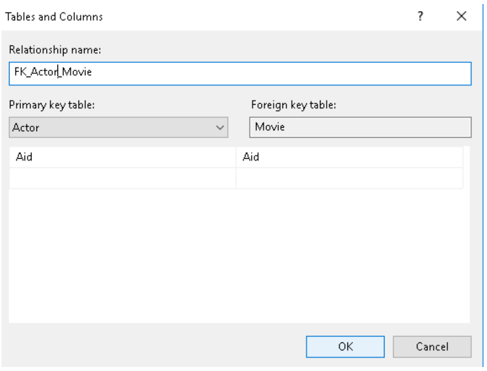

Foreign Key Relationships

Selected Relationship:

FK_Table_1_Table_1*

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

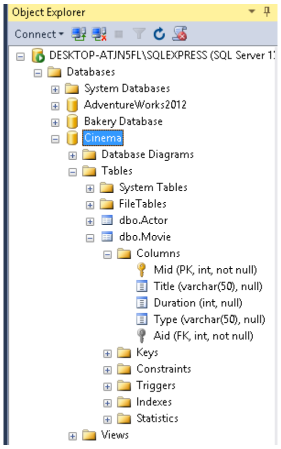
(General)	
Check Existing Data On Creati	Yes
Tables And Columns Specific	
Foreign Key Base Table	Table_1
Foreign Key Columns	Mid
Primary/Unique Key Base	Table_1
Primary/Unique Key Colu	Mid
Identity	
(Name)	FK_Table_1_Table_1
Description	
Table Designer	


Tables and Columns

Relationship name:

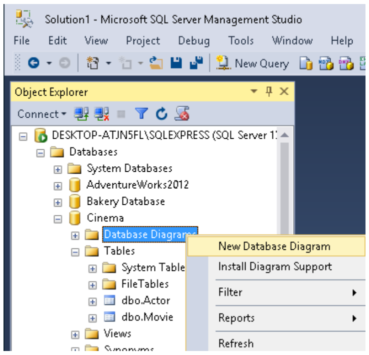
FK_Actor_Movie

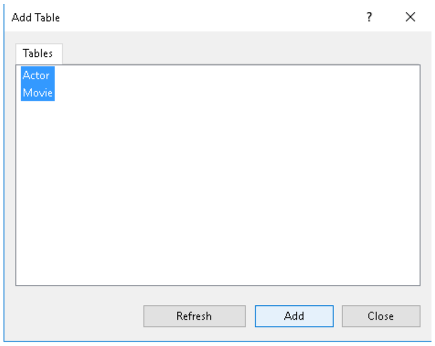
Primary key table:	Foreign key table:
Actor	Movie
Aid	Aid

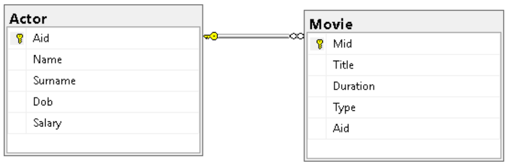


```
CREATE TABLE Movie(
        Mid int NOT NULL PRIMARY KEY,
        Title varchar(50) NULL,
        Duration int NULL,
        Type varchar(50) NULL,
        Aid int NOT NULL,
        CONSTRAINT FK_Actor_Movie FOREIGN KEY(Aid) REFERENCES Actor (Aid)
)
```

In the diagram can see directly the relation



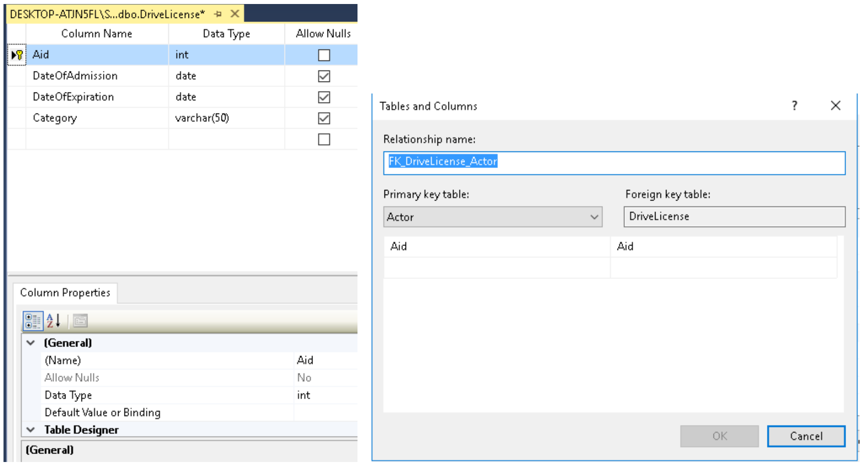




Relationships between the tables: 1-1, 1-n, m-n

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.)



DESKTOP-ATJN5FL\S...dbo.DriveLicense*

Column Name	Data Type	Allow Nulls
Aid	int	□
DateOfAdmission	date	☑
DateOfExpiration	date	☑
Category	varchar(50)	☑
		□

Column Properties

(General)
(Name) Aid
Allow Nulls No
Data Type int
Default Value or Binding
Table Designer
(General)

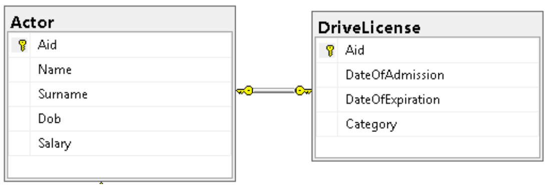
Tables and Columns

Relationship name: FK_DriveLicense_Actor

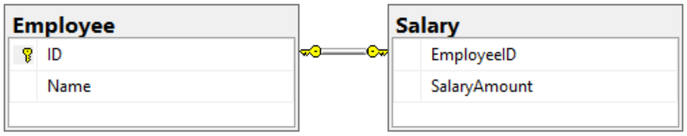
Primary key table: Actor

Foreign key table: DriveLicense

Actor	DriveLicense
Aid	Aid



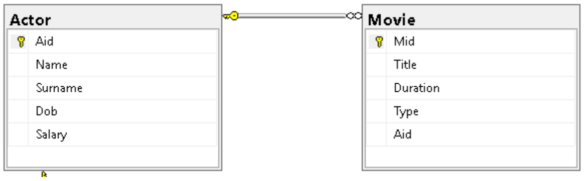
Create table Employee (ID int primary key, Name varchar(50))	Create table Salary (EmployeeID int primary key, SalaryAmount int, Constraint FK_Salary_Employee Foreign key(EmployeeID) References Employee(ID))



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

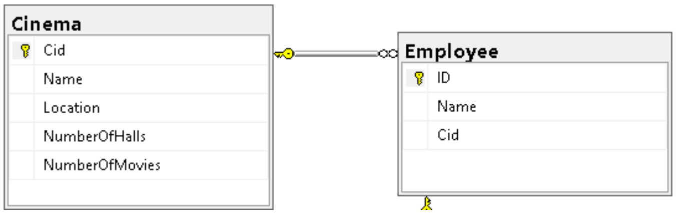
Relation 1-m (one to many)

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

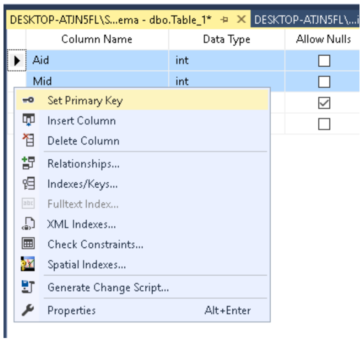


Relation m-m (many to many)

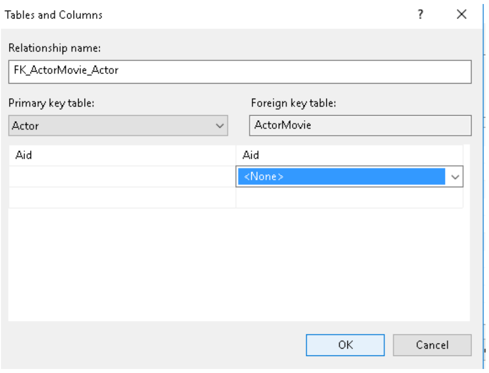
- 2 relations 1-m – this relation can be implemented through an intermediate table.

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"



Column Name	Data Type	Allow Nulls
Aid	int	☐
Mid	int	☐
		☑
		☐

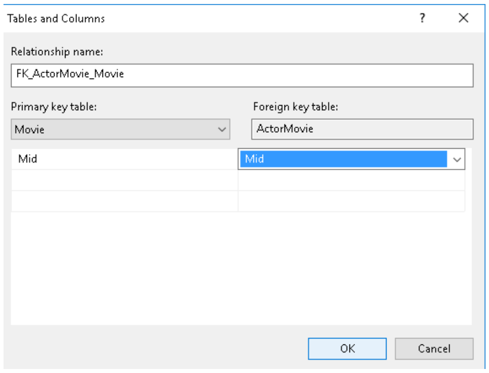


Relationship name: FK_ActorMovie_Actor

Primary key table: Actor

Foreign key table: ActorMovie

Actor	ActorMovie
Aid	Aid
	<None>

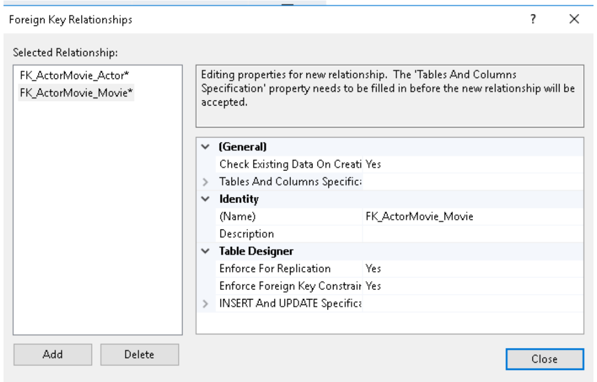


Relationship name: FK_ActorMovie_Movie

Primary key table: Movie

Foreign key table: ActorMovie

Movie	ActorMovie
Mid	Mid

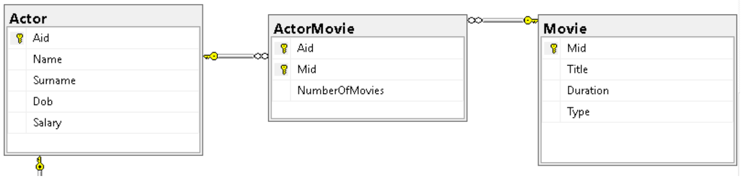


Selected Relationship:

FK_ActorMovie_Actor*
FK_ActorMovie_Movie*

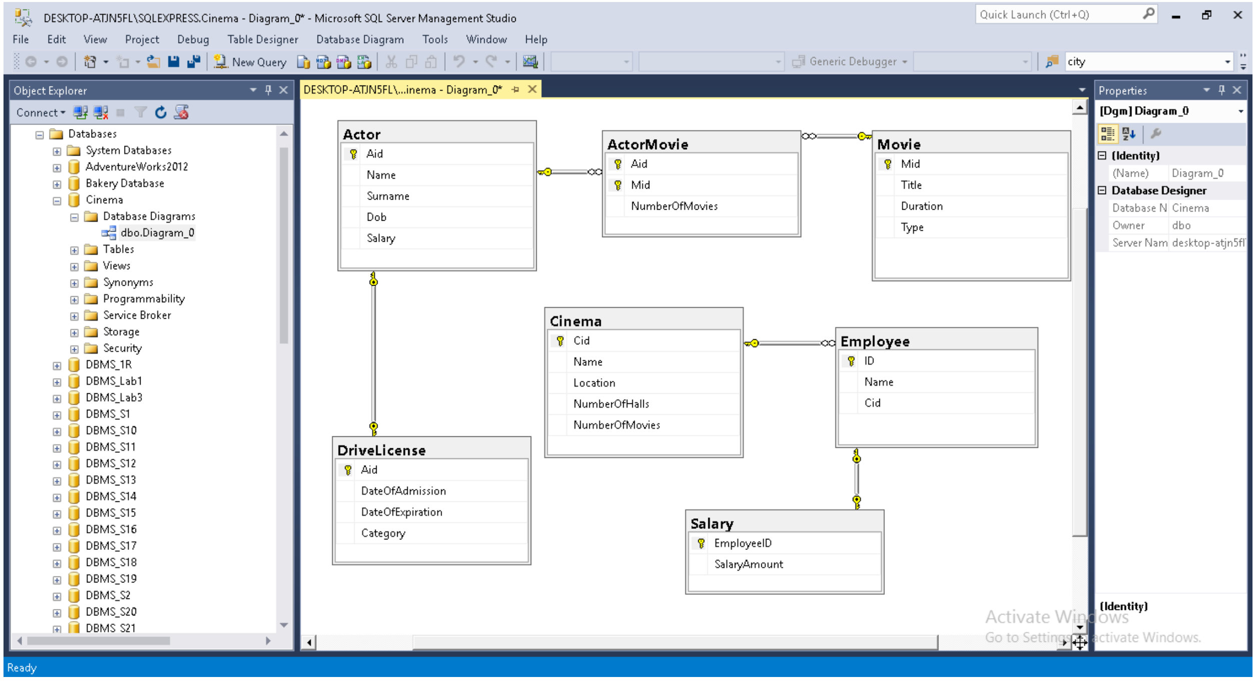
Editing properties for new relationship. The 'Tables And Columns Specification' property needs to be filled in before the new relationship will be accepted.

(General)	
Check Existing Data On Creati	Yes
Tables And Columns Specific:	
Identity	
(Name)	FK_ActorMovie_Movie
Description	
Table Designer	
Enforce For Replication	Yes
Enforce Foreign Key Constrair	Yes
INSERT And UPDATE Specific:	



- 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:



DESKTOP-ATJN5FL\SQLEXPRESS.Cinema - Diagram_0* - Microsoft SQL Server Management Studio

Actor
- Aid
- Name
- Surname
- Dob
- Salary

ActorMovie
- Aid
- Mid
- NumberOfMovies

Movie
- Mid
- Title
- Duration
- Type

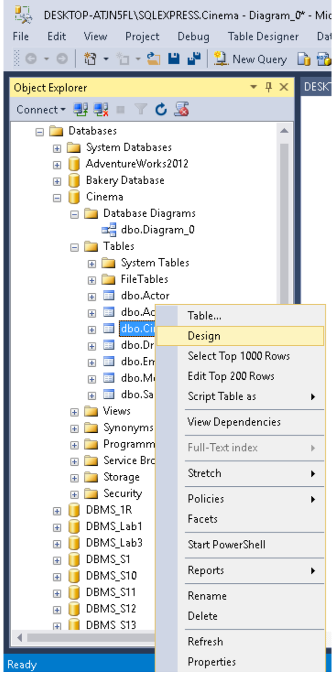
Cinema
- Cid
- Name
- Location
- NumberOfHalls
- NumberOfMovies

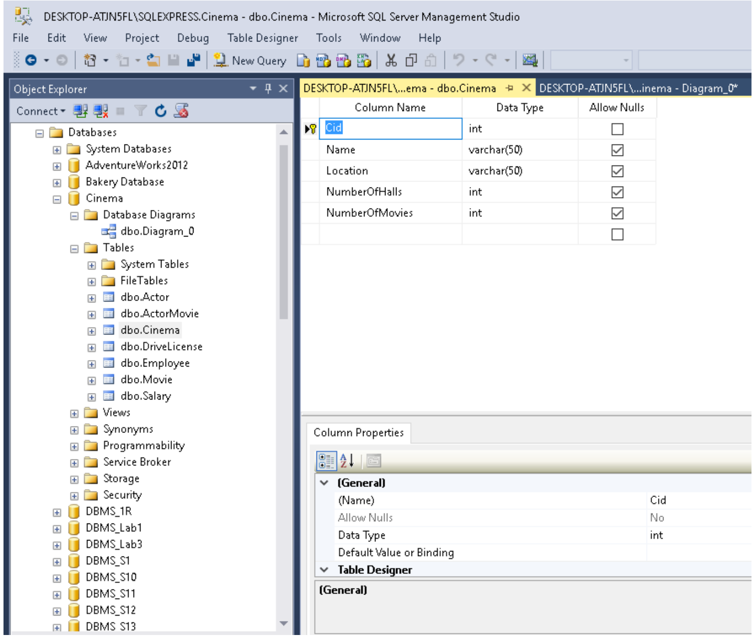
Employee
- ID
- Name
- Cid

DriveLicense
- Aid
- DateOfAdmission
- DateOfExpiration
- Category

Salary
- EmployeeID
- SalaryAmount

Modifications on tables (add new columns…)





DESKTOP-ATJN5FL\SQLEXPRESS.Cinema - dbo.Cinema - Microsoft SQL Server Management Studio

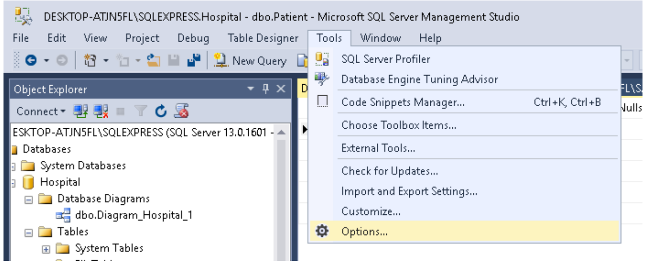
Column Name	Data Type	Allow Nulls
Cid	int	☐
Name	varchar(50)	☑
Location	varchar(50)	☑
NumberOfHalls	int	☑
NumberOfMovies	int	☑
		☐

Column Properties

(General)

(Name)	Cid
Allow Nulls	No
Data Type	int
Default Value or Binding	

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


DESKTOP-ATJN5FL\SQLEXPRESS.Hospital - dbo.Patient - Microsoft SQL Server Management Studio
File Edit View Project Debug Table Designer Tools Window Help
New Query
Object Explorer
Connect
Databases
System Databases
Hospital
Database Diagrams
dbo.Diagram_Hospital_1
Tables
System Tables
SQL Server Profiler
Database Engine Tuning Advisor
Code Snippets Manager... Ctrl+K, Ctrl+B
Choose Toolbox Items...
External Tools...
Check for Updates...
Import and Export Settings...
Customize...
Options...

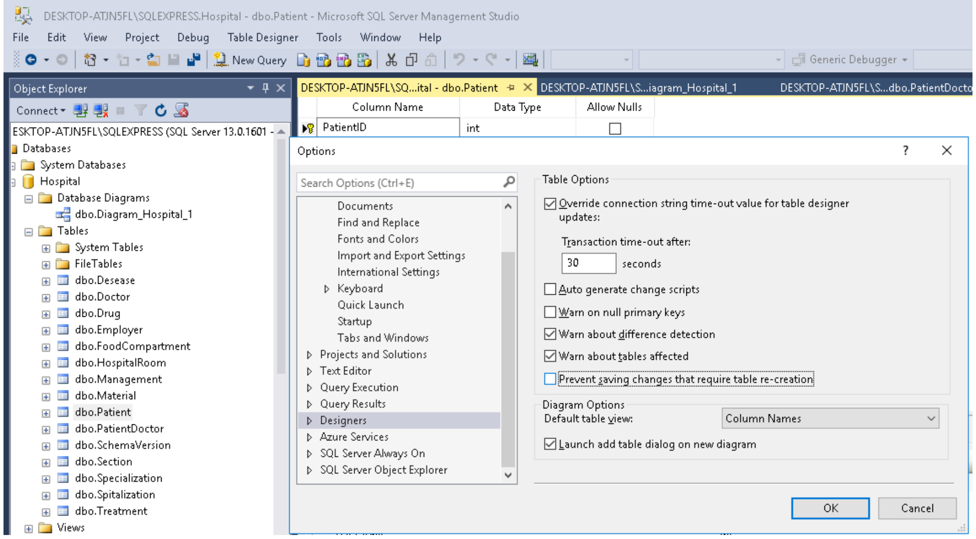


Table Options

☑ Override connection string time-out value for table designer updates:

Transaction time-out after: 30 seconds

☐ Auto generate change scripts

☐ Warn on null primary keys

☑ Warn about difference detection

☑ Warn about tables affected

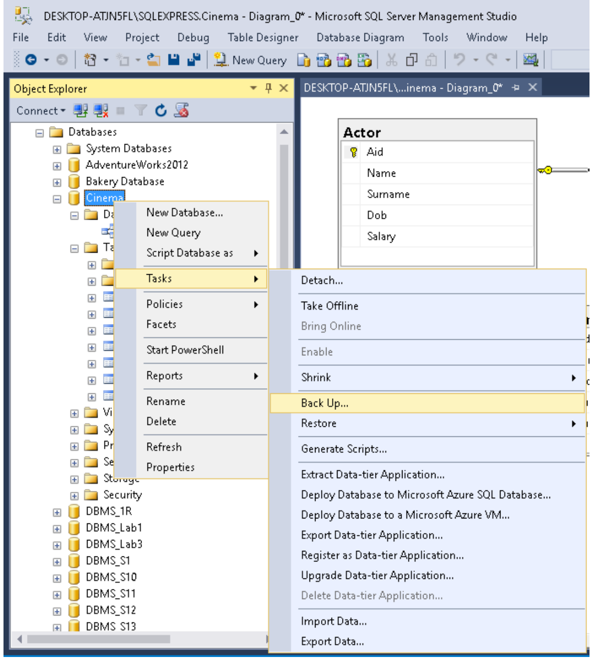
☐ Prevent saving changes that require table re-creation

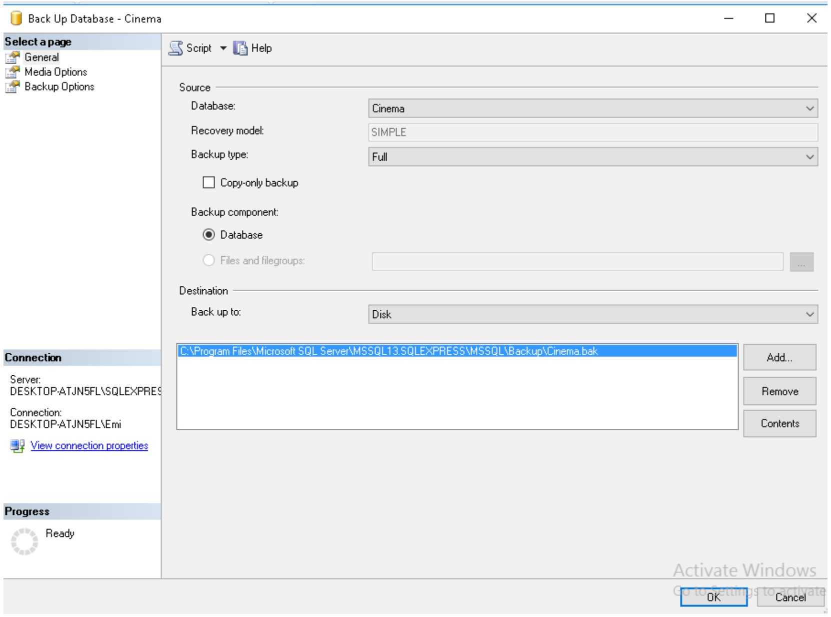
Diagram Options

Default table view: Column Names

☑ Launch add table dialog on new diagram

Back-up the database


DESKTOP-ATJN5FL\SQLEXPRESS.Cinema - Diagram_0* - Microsoft SQL Server Management Studio
File Edit View Project Debug Table Designer Database Diagram Tools Window Help
New Query
Object Explorer
Connect
DESKTOP-ATJN5FL\...inema - Diagram_0*
Databases
System Databases
AdventureWorks2012
Bakery Database
Cinema
Security
DBMS_1R
DBMS_Lab1
DBMS_Lab3
DBMS_S1
DBMS_S10
DBMS_S11
DBMS_S12
DBMS_S13
Actor
Aid
Name
Surname
Dob
Salary
New Database...
New Query
Script Database as
Tasks
Policies
Facets
Start PowerShell
Reports
Rename
Delete
Refresh
Properties
Detach...
Take Offline
Bring Online
Enable
Shrink
Back Up...
Restore
Generate Scripts...
Extract Data-tier Application...
Deploy Database to Microsoft Azure SQL Database...
Deploy Database to a Microsoft Azure VM...
Export Data-tier Application...
Register as Data-tier Application...
Upgrade Data-tier Application...
Delete Data-tier Application...
Import Data...
Export Data...



Back Up Database - Cinema

Select a page

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

Connection

Server:
DESKTOP-ATJN5FL\SQLEXPRES

Connection:
DESKTOP-ATJN5FL\Emi

Progress

Ready