



Syllabus

Databases

Mileto Di Marco
Kristien Roels
Bart Soete

Inhoudsopgave

H3 - SQL Server en SQL Server Management Studio	3
1 Inleiding	3
2 Datatypes in SQL Server	3
2.1 Gehele getallen	3
2.2 Exacte gebroken getallen	3
2.3 Gebroken getallen	4
2.4 Datum/Tijd	4
2.5 Karakter datatypes	4
2.6 Binaire gegevens	5
2.7 Overige	5
3 Kennismaking met SQL Server Management Studio (SSMS)	6
4 Acties met SQL Server Management Studio (SSMS)	7
4.1 Voorbeelddatabase MijnFilms	7
4.2 Een nieuwe database aanmaken	9
4.3 Aanmaken tabellen	9
4.4 Verwijderen tabellen	20
4.5 Relaties	21
4.6 Invoeren data	26

H3 - SQL Server en SQL Server Management Studio

1 Inleiding

We hernemen nog even wat we tot nu toe hebben gezien :

- MS SQL Server is een Relationeel Database Management Systeem (RDBMS) dat zich manifesteert als een service. Dit betekent dat SQL Server zich als een programma zal aanbieden dat op de achtergrond draait (wat je bijgevolg dus niet te zien krijgt).
- Om de dataserver te beheren (server en databases) is er een tool voorhanden die je op je PC dient te installeren: de SQL Server Management Studio (SSMS). Deze tool werkt **onafhankelijk** van het feit of er al dan niet een SQL Server op de machine geïnstalleerd is. Met deze tool kan je om het even welke SQL Server gaan beheren zolang je er maar verbinding mee kunt maken.

2 Datatypes in SQL Server

Binnen SQL Server zijn er verschillende datatypes mogelijk. Onderstaande tabellen geven een overzicht.

Voor een volledige bespreking van alle datatypes, zie <https://docs.microsoft.com/nl-be/sql/t-sql/data-types/data-types-transact-sql>

2.1 Gehele getallen

Gegevenstype	Toegestane waarden	Grootte
Bit	1, 0 of NULL (NULL: speciale waarde om aan te duiden dat de waarde ongekend is)	1 byte voor 1-8 bit kolommen
Tinyint	0 tot 255	1 byte
Smallint	Gehele waarden van -2^{15} tot en met $2^{15}-1$	2 bytes
Int	Gehele waarden van -2^{31} tot en met $2^{31}-1$	4 bytes
Bigint	Gehele waarden van -2^{63} tot en met $2^{63}-1$	8 bytes

2.2 Exacte gebroken getallen

Gegevenstype	Toegestane waarden	Grootte	
Decimal(p [,s]) of Numeric(p [,s]) Expressies tussen [] zijn optioneel	Numerieke gegevens met een vastgelegde precisie (p) en schaal (s) p: maximaal aantal cijfers zowel voor als na de komma : 1-38 (default 18) s: maximaal aantal cijfers rechts van de komma : 0-p	Gebaseerd op p :	
		p	Bytes
		1-9	5
		9-19	9
		20-28	13
		29-38	17
Smallmoney	Geldwaarden, kommagetallen nauwkeurig tot op 0,0001 -214748,3648 tot en met +214748,3647	4 bytes	
Money	Geldwaarden, kommagetallen nauwkeurig tot op 0,0001 -922337203685477,5808 tot en met 922337203685477,5807	8 bytes	

2.3 Gebroken getallen

Gegevenstype	Toegestane waarden	Grootte						
Float[(n)]	<p>Floating point waarden van $-1,79 \cdot 10^{308}$ tot en met $1,79 \cdot 10^{308}$ waarbij:</p> <p>n (1-53): aantal bits dat gebruikt wordt om de mantisse van het getal omgezet in wetenschappelijke notatie op te slaan</p> <p>n bepaalt bijgevolg de nauwkeurigheid en de opslagruimte</p> <p>Float waarden zijn benaderingen</p> <p>Opm: float(53) is equivalent met double precision</p>	<p>Afhankelijk van n</p> <table><tr><th>n</th><th>Bytes</th></tr><tr><td>1-24</td><td>4</td></tr><tr><td>25-53</td><td>8</td></tr></table>	n	Bytes	1-24	4	25-53	8
n	Bytes							
1-24	4							
25-53	8							
Real	Floating point waarden van $-3,40 \cdot 10^{38}$ tot en met $3,40 \cdot 10^{38}$	4 bytes						

2.4 Datum/Tijd

Gegevenstype	Toegestane waarden	Grootte
Smalldatetime	Datum- en tijdgegevens vanaf 1 januari 1900 tot en met 6 juni 2079 nauwkeurig tot op 1 minuut	4 bytes
Datetime	Datum- en tijdgegevens vanaf 1 januari 1753 tot en met 31 december 9999 nauwkeurig op 3 ms	8 bytes
Datetime2 (verdient voorkeur boven datetime)	Datum- en tijdgegevens vanaf 1 januari 0001 tot en met 31 december 9999 nauwkeurig op 100 ns	Maximaal 8 bytes
Date	Datumgegevens vanaf 1 januari 0001 tot en met 31 december 9999	3 bytes
Time	Tijdgegevens nauwkeurig op 100 ns	5 bytes

2.5 Karakter datatypes

Gegevenstype	Toegestane waarden	Grootte
Opmerking: Non-unicode karakters worden opgeslagen in 1 byte: slechts 256 tekens worden ondersteund. Unicode karakters worden opgeslagen in 2 bytes en zijn veel diverser: ook tekens uit vreemde talen worden ondersteund.		
Char[(n)]	Non-unicode karakters met een vaste lengte tot maximaal 8000 tekens n: stringlengte (default 1) Char(30) betekent 30 karakters (er wordt aangevuld met spaties)	N bytes
Varchar[(n MAX)]	Non-unicode karakters met een variabele lengte tot maximaal 8000 tekens (worden niet aangevuld met spaties). n: stringlengte (default 1) MAX: maximale opslag (2 GB)	1 byte per karakter
Nchar[(n)]	Unicode karakters met vaste lengte tot maximaal 4000 tekens n: stringlengte (default 1)	N * 2 bytes
Nvarchar[(n MAX)]	Unicode karakters met variabele lengte tot maximaal 4000 tekens. n: stringlengte (default 1) MAX: maximale opslag (2 GB)	2 bytes per karakter

2.6 Binaire gegevens

Gegevenstype	Toegestane waarden	Grootte
Binary[(n)]	Binaire gegevens met vaste lengte tot maximaal 8000 bytes n: lengte (default 1)	N bytes
Varbinary[(n MAX)]	Binaire gegevens met variabele lengte tot maximaal 8000 bytes n: lengte (default 1) MAX: maximale opslag (2 GB)	Actuele lengte

2.7 Overige

Gegevenstype	Toegestane waarden	Grootte
Cursor	Een verwijzing naar een cursor (entiteit die een verwijzing maakt naar een rij in een resultaatset)	
Rowversion	Eén kolom in een tabel kan van het type rowversion zijn. Dan wordt per record bijgehouden wanneer de laatste wijziging is gebeurd.	
SQL_Variant	Waarden van een willekeurig datatype (met enkele beperkingen: zie Help)	Max. 8016 bytes
Uniqueidentifier	Een Global Unique Identifier (GUID) is een 16-byte unieke numerieke waarde gevormd door SQL Server Een voorbeeld: 6F9619FF-8B86-D011-B42D-00C04FC964FF	16 bytes
Table	Gegevenstype op een resultaatset op te slaan voor latere verwerking	
XML	Type voor de opslag van XML-data	

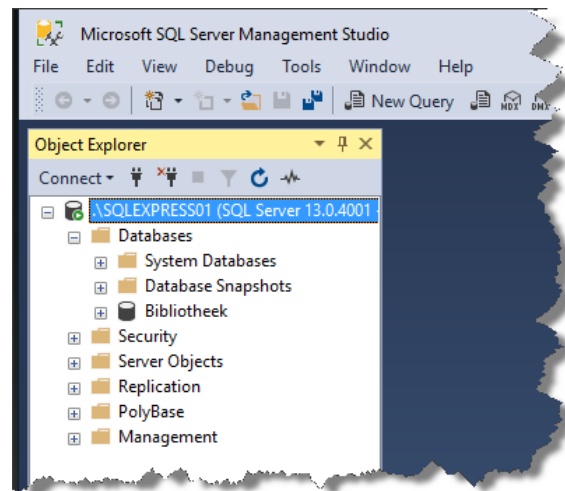
3 Kennismaking met SQL Server Management Studio (SSMS)

Na het starten van de tool wordt gevraagd verbinding te maken met een (SQL) Server. Gezien we de database lokaal hebben geïnstalleerd geef je je computernaam in gevolgd door de service die je wenst te gebruiken:

- *LocalePC*\sqlExpress

Gezien de server lokaal draait kan je hier ook `.\SQLEXPRESS` ingeven (. staat voor de lokale server). We kiezen voor Windows Authentication.

Na de installatie zien we de lokale SQLEXPRESS Databaseserver waarbij door de gebruiker één database werd toegevoegd: Bibliotheek (zie vorig hoofdstuk).



Vanuit dit scherm is het onder andere mogelijk om :

- Een nieuwe database aan te maken
- Nieuwe tabellen toe te voegen aan de (bestaande) database
- Bestaande tabellen te wijzigen
- Tabellen te wissen
- Inhoud van de tabellen te bekijken
- ...

4 Acties met SQL Server Management Studio (SSMS)

4.1 Voorbeelddatabase MijnFilms

In onderstaand (uitgewerkt) voorbeeld maken we een nieuwe database aan genaamd “MijnFilms”.

We wensen volgende data in te voeren:

Titel	Acteur	Jaar	Stemmen	Score
Cast Away	Tom Hanks	2000	1	75
Cast Away	Hellen Hunt	2000	1	75
The Terminal	Tom Hanks	2004	1	62
The Terminal	Catherine Zeta Jones	2004	1	62
The Da Vinci Code	Tom Hanks	2006	1	82
The Da Vinci Code	Audrey Tautou	2006	1	82
The Da Vinci Code	Ian McKellen	2006	1	82
The Shawshank Redemption	Tim Robins	1994	1	93
The Shawshank Redemption	Morgan Freeman	1994	1	93
The Circle	Tom Hanks	2017	1	65
The Circle	Emma Watson	2017	1	65
Shrek	Mike Meyers	2001	1	75
Shrek	Cameron Diaz	2001	1	75
Shrek	Eddy Murphy	2001	1	75
Bad Teacher	Cameron Diaz	2011	1	63
Bad Teacher	Justin Timberlake	2011	1	63
The Hunger Games	Jennifer Lawrence	2012	1	70
The Hunger Games	Woody Harrelson	2012	1	70
The Hunger Games	Lenny Kravitz	2012	1	70
X-Men : First Class	James McAvoy	2011	1	60
X-Men : First Class	Kevin Bacon	2011	1	60
X-Men : First Class	Jennifer Lawrence	2011	1	60
RED	Bruce Willis	2010	1	65
RED	John Malkovich	2010	1	65
RED	Morgan Freeman	2010	1	65
RED	Hellen Mirren	2010	1	65

Na normaliseren wordt de data op volgende manier ingegeven:

Tabel Film (**FilmID**, Titel, Jaar, Stemmen, Score)

FilmID	Titel	Jaar	Stemmen	Score
1	Cast Away	2000	1	75
2	The Terminal	2004	1	62
3	The Da Vinci Code	2006	1	82
4	Shawshank Redemption	1994	1	93
5	The Circle	2017	1	65
6	Shrek	2001	1	75
7	Bad Teacher	2011	1	63
8	The Hunger Games	2012	1	70
9	X-Men : First Class	2011	1	60

10	RED	2010	1	65
----	-----	------	---	----

Tabel Acteur (**ActeurID**, ActeurNaam)

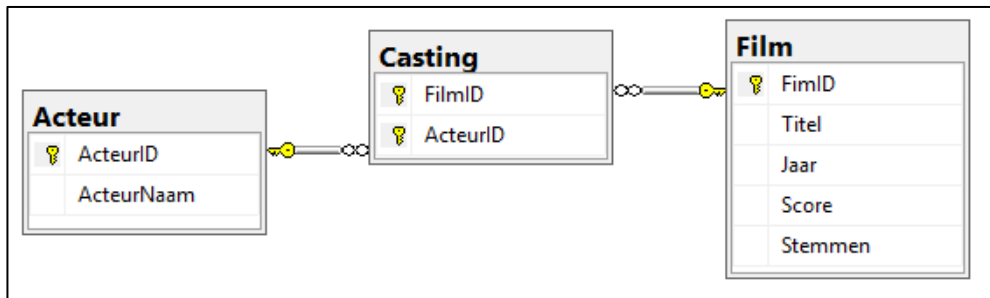
ActeurID	ActeurNaam
1	Tom Hanks
2	Tim Robins
3	Morgan Freeman
4	Catherine Zita Jones
5	Audrey Tautou
6	Ian McKellen
7	Emma Watson
8	Mike Meyers
9	Cameron Diaz
10	Eddy Murphy
11	Justin Timberlake
12	Jennifer Lawrence
13	Woody Harrelson
14	Lenny Kravitz
15	James McAvoy
16	Kevin Bacon
17	Bruce Willis
18	Hellen Hunt
19	John Malkovich
20	Hellen Mirren

Koppeltabel Casting (**FilmID**, **ActeurID**)

FilmID	ActeurID
1	1
1	18
2	1
2	4
3	1
3	5
3	6
4	2
4	3
5	1
5	7
6	8
6	9
6	10
7	9
7	11
8	12
8	13
8	14

9	12
9	15
9	16
10	3
10	19
10	20

Een globaal overzicht :



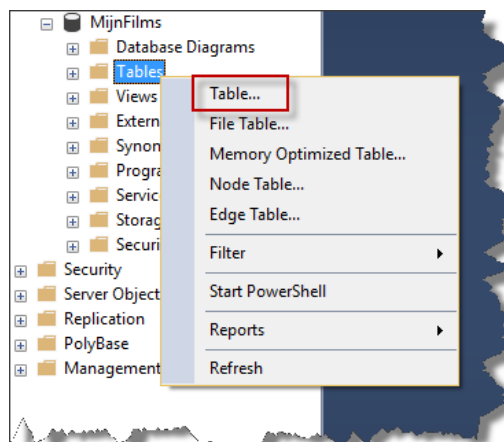
- In de tabel Film bewaren we de algemene informatie over een film.
Film (**FilmID**, Titel, Jaar, Score, Stemmen)
- In de tabel Acteur bewaren we informatie over de belangrijkste auteurs.
Acteur (**ActeurID**, ActeurNaam)
- In een koppelingstabel Casting houden we informatie bij welke auteur in welke film gespeeld heeft.
Casting (**FilmID**, **ActeurID**)

4.2 Een nieuwe database aanmaken

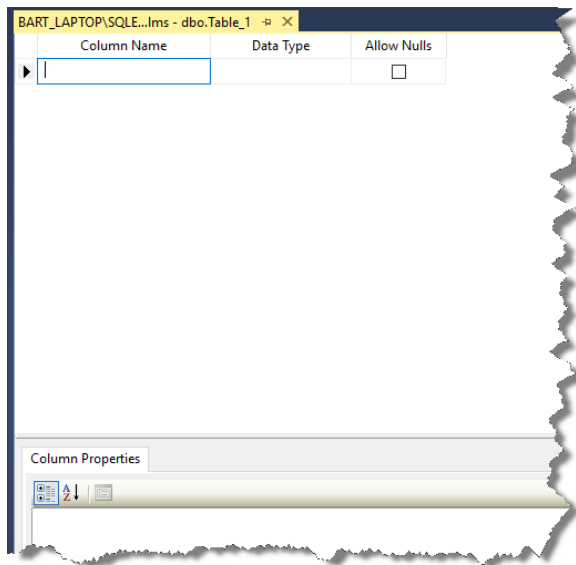
Een nieuwe database aanmaken doe je door in de node "Databases" te rechtsklikken. Je selecteert hierna "New Database" en geeft de database de naam "MijnFilms".

4.3 Aanmaken tabellen

In de database "MijnFilms" worden standaard een aantal mappen aangemaakt. Na het rechtsklikken op het mapje Tables selecteren we "Table...".



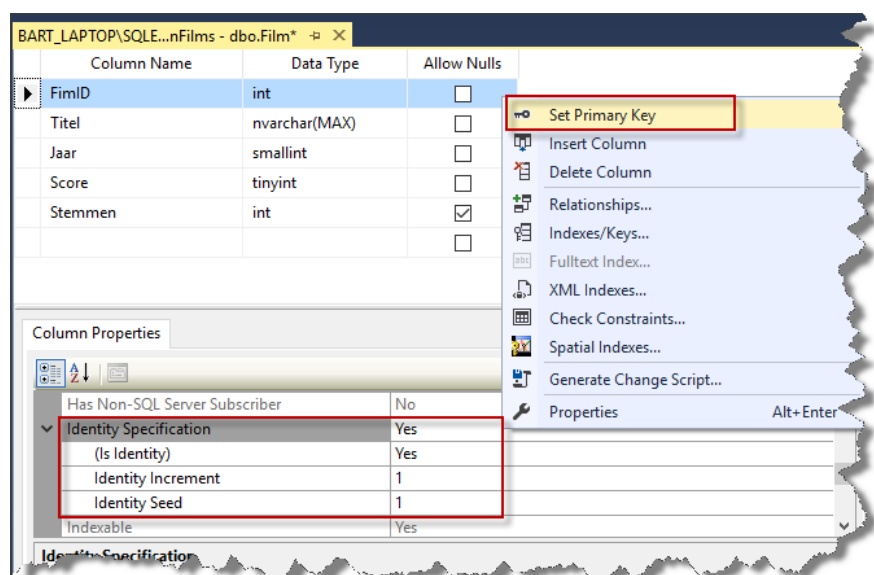
Aan de rechterkant van het scherm verschijnt een nieuw venster waar je het ontwerp van een tabel kunt gaan aanmaken.



In het bovenste gedeelte zal je de namen en de datatypes van de nieuwe kolommen opgeven. Onderaan kan je dan verdere eigenschappen van een kolom ingeven.

We voorzien volgende kolommen in de tabel Film:

- FilmID: unieke waarde waarmee de film kan geïdentificeerd worden
 - We voorzien hier dat dit veld als primaire sleutel wordt gedefinieerd en automatisch wordt opgevuld.

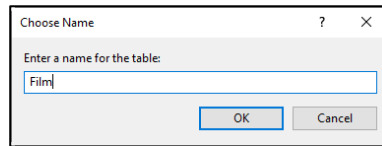



- Titel: titel van de film
 - Dit is een tekstveld. Om zeker te zijn dat langere titels bewaard kunnen worden maken we hier een nvarchar(MAX) van. We staan niet toe dat hier lege velden (null) worden ingegeven.
- Jaar: jaar waarin de film werd uitgebracht
 - Jaar is een numerieke waarde. Gezien de waarden (1922-huidig jaar) eerder beperkt zijn volstaat hier een smallint. We staan niet toe dat null wordt ingegeven.
- Score: een waarderingscijfer tussen 0 en 100
 - Score is een numerieke waarde. Hier volstaat een tinyint. We laten eveneens geen null-waarden toe.
- Stemmen: het aantal mensen dat op de film gestemd heeft
 - Stemmen bevat een getal. We kiezen hier een int waarde. Hier laten we echter wel lege velden toe.

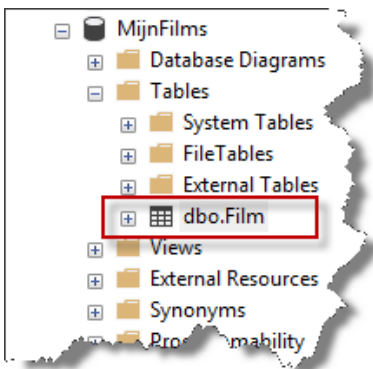
Je bewaart de tabel door bovenaan in de knoppenbalk op de knop opslaan te klikken.



Je voert de naam in en klikt op OK.



Wanneer je het ontwerpvenster sluit merk je (eventueel na een refresh ) dat de tabel werd toegevoegd aan de database.



dbo staat voor (zoals eerder vermeld) databaseowner.

Zolang je binnen je database blijft werken doet het er weinig toe wat als schemanaam gebruikt wordt. Wanneer je echter vanuit een andere database deze tabel wenst aan te spreken, dan moet je wel weten dat dit steeds als volgt dient te gebeuren :

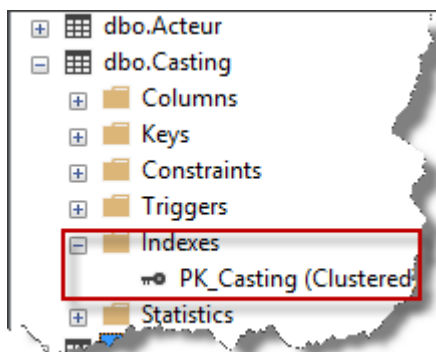
`<databasenaam>.<schemanaam>.<tabelnaam>`

Je kunt nu volgende tabellen toevoegen.

- Acteur:
 - ActeurID: int, primaire sleutel, Identity Seed
 - ActeurNaam: varchar(MAX), geen lege velden toegestaan
- Casting:
 - FilmID: int, primaire sleutel
 - ActeurID: int, primaire sleutel

Merk op dat er in de tabel Casting een samengestelde primaire sleutel is (een primaire sleutel die bestaat uit 2 velden). Om dit voor elkaar te krijgen kun je de beide velden selecteren (veld selecteren op grijze rij-indicator en ctrl-klik op het andere veld) vooraleer op het sleutel-icoontje de drukken.

Ter info: de gemaakte indexen worden in het mapje "Indexes" bijgehouden.





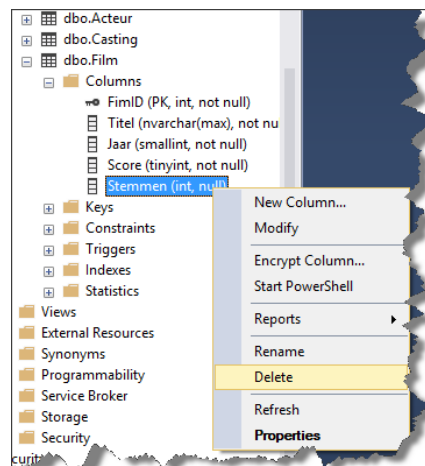
4.3.1 Kolommen uit een tabel verwijderen

Het is vrij eenvoudig om uit een bestaande tabel een kolom te verwijderen. Let op dat je door het verwijderen van een kolom uit een tabel de integriteit van de database niet beschadigt.

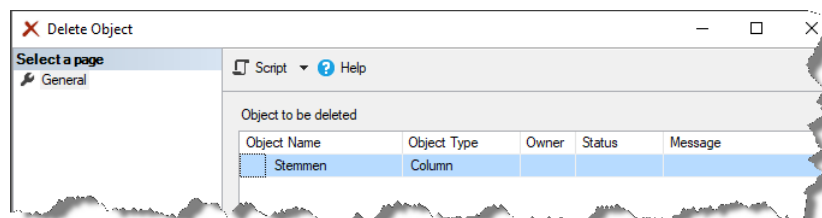
In onderstaand voorbeeld wens je de kolom Stemmen uit de tabel Film te verwijderen.

We gebruiken hiervoor de volgende werkwijze :

- Klik in de object browser op het  teken naast de tabel (je ziet alle verschillende submapjes zoals Columns, Keys,...)
- Klik op het  teken naast Columns
- Selecteer de kolom die je wenst te verwijderen
- Druk op de del-toets of klik er met de rechtermuisknop op en selecteer de optie "Delete"



- Het "Delete Object" venster verschijnt waarin afgebeeld staat welk(e) object(en) zal (zullen) verwijderd



worden.

- Klik onderaan op OK.

Opmerking: had het verwijderen van de kolom de integriteit van de database in het gedrang gebracht (bv. een veld verwijderen dat deel uitmaakt van een relatie) dan zou SQL Server geweigerd hebben deze kolom te verwijderen.

4.3.2 Kolommen invoegen en wijzigen

Het is vrij eenvoudig om kolommen toe te voegen aan de tabel of kolommen te wijzigen.

Er dient wel een onderscheid gemaakt te worden tussen de toestand waarbij zich nog geen gegevens (data) bevinden en de toestand waarin er wel al gegevens zijn ingebracht in de tabel.

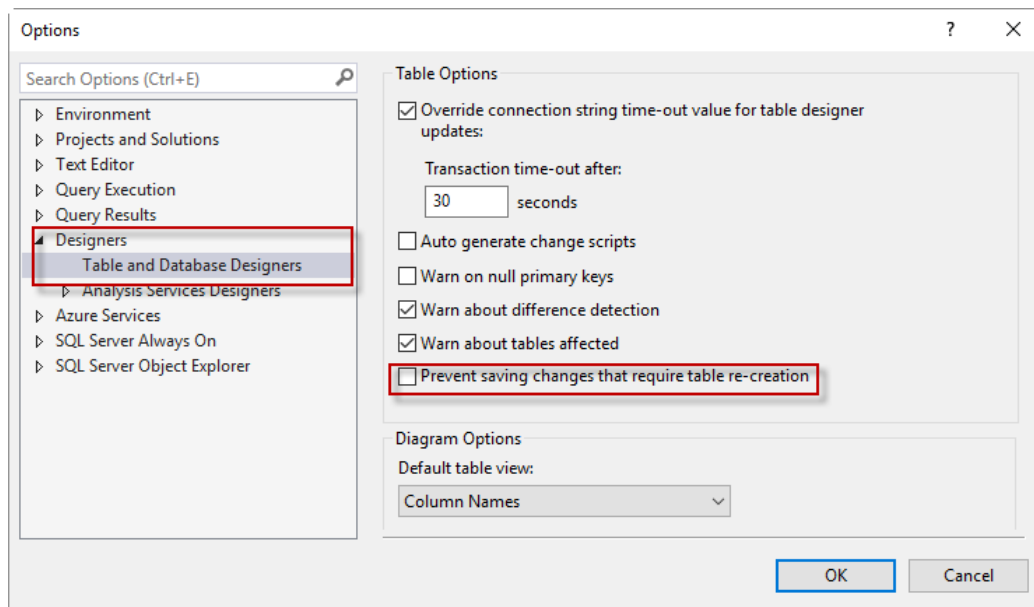
Indien er zich geen gegevens in de tabel bevinden, kan nagenoeg alles.

Wanneer er wel gegevens in de tabel zijn opgeslagen, dan zal je hier rekening moeten mee houden.

Enkele voorbeelden:

- Stel dat je een kolom hebt waarvan je hebt vastgelegd dat er zich null-waarden in mogen bevinden en er zijn reeds null-waarden aanwezig, dan kan je dit niet aanpassen vooraleer alle null-waarden zijn ingevuld.
- Stel dat je een kolom hebt met numerieke waarden (datatype). Je kan dit probleemloos omzetten naar alfanumerieke waarden. Omgekeerd kan alleen als de reeds ingegeven alfanumerieke waarden geïnterpreteerd kunnen worden als een numerieke waarde en het gekozen numerieke datatype groot genoeg is om de waarden in op te slaan.
- Stel dat je een nieuwe kolom wenst toe te voegen en de optie “Allow null” wenst uit te schakelen. Dit zal niet lukken, je dient eerst de kolom aan te maken met de optie “Allow null” ingeschakeld, vervolgens alle nodige waarden toevoegen waarna je de optie alsnog kunt uitschakelen.

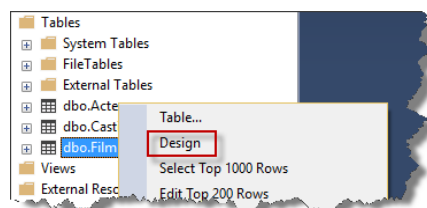
Voor bepaalde aanpassingen moet de tabel opnieuw worden aangemaakt. Hiervoor dien je via Tool > Options > ... de optie “Prevent saving changes that require table re-creation” uit te schakelen.



Opgepast, in een bedrijfsomgeving is het best deze optie terug aan te vinken, nadat de wijzigingen zijn uitgevoerd, zodat de database beschermd wordt tegen “abuis” dataverlies.

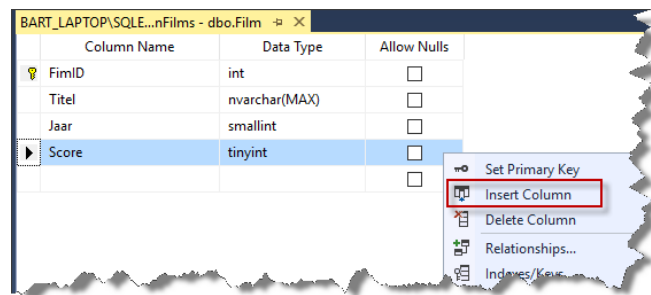
We gaan de kolom “Stemmen” opnieuw toevoegen en kiezen hiervoor volgende werkwijze:

- Klik in de object browser met de rechtermuisknop op de tabel “Film”.
- Selecteer in het snelmenu de optie “Design”.



- We voegen de kolom “Stemmen” toe, vóór de kolom “Score”.


- Klik met de rechtermuistoets op de kolom “Score” en selecteer in het snelmenu voor de optie “Insert Column”



- In de vrijgekomen ruimte voer je de gegevens in van de nieuwe kolom (naam=Stemmen, datatype=int)
- Bewaar de tabel en sluit deze af.

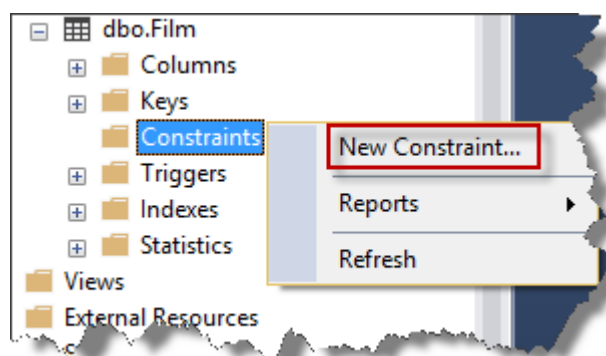
4.3.3 Constraints aanmaken, wijzigen en verwijderen

Constraints (integriteitsregels of integriteitsvoorwaarden) zorgen ervoor dat enkel gegevens kunnen ingegeven worden die aan bepaalde voorwaarden voldoen.

Om een constraint te maken dien je in de object browser op het  teken naast de tabel te klikken waarvoor je een integriteitsregel wenst in te geven. Je kan dit gaan doen in het mapje Constraints.

We passen dit toe op de tabel “Film”:

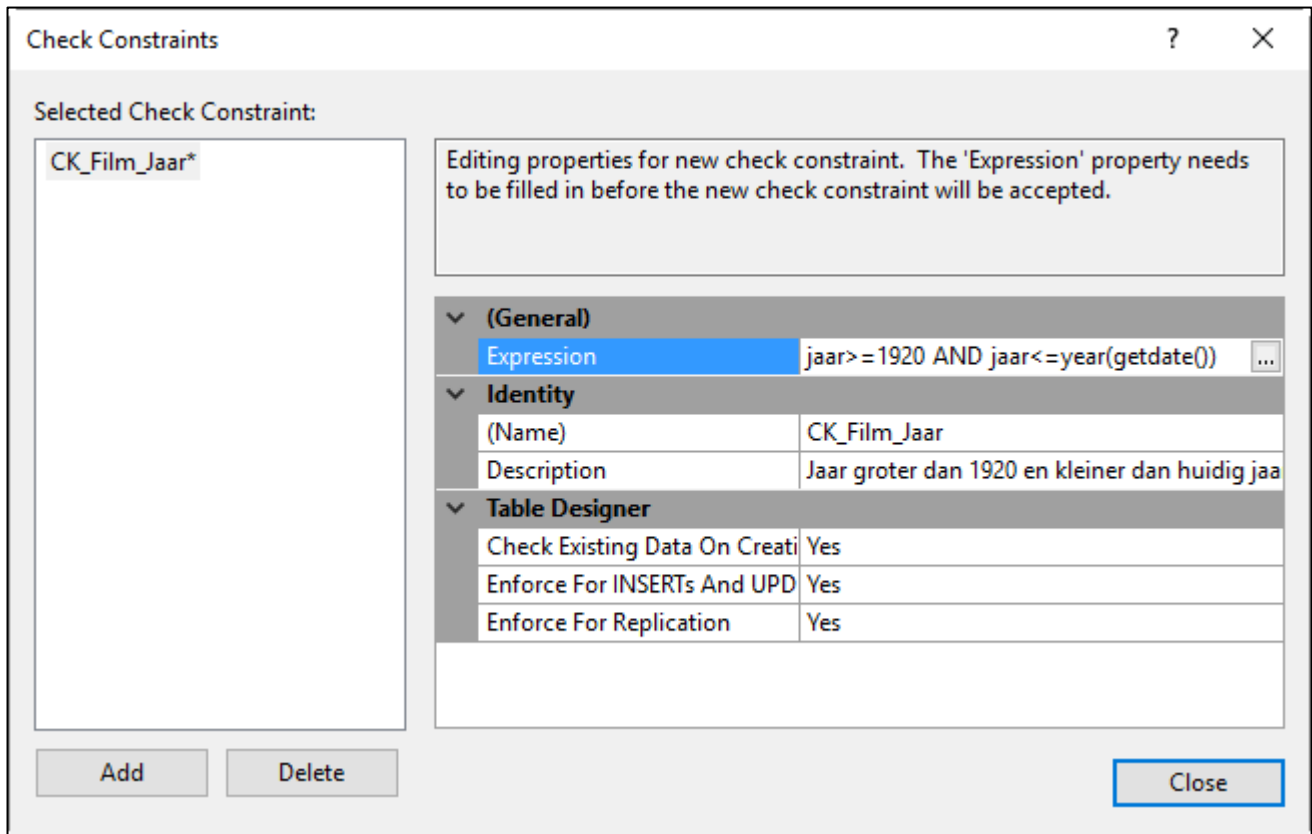
- Rechtsklik op het mapje “Constraints” en selecteer in het snelmenu de optie “New Constraint...”



- Het venster “Check Constraints” wordt geopend. Volgende zaken worden bij voorkeur ingegeven:
 - Naam (de naam van de constraint voorgedaan door CK_) => CK_Film_Jaar
 - De beschrijving (is optioneel maar ten eerste aan te raden) => Jaartal niet kleiner dan 1920
 - De beperkende maatregel zelf (Expression)

We willen ervoor zorgen dat in het veld “jaar” van de tabel “Film” enkel jaartallen kunnen worden ingevoerd tussen 1920 en het huidige jaar.

Hiervoor vullen we in het veld Expression volgende “code”: `jaar >= 1920 AND jaar <= year(getdate())`.

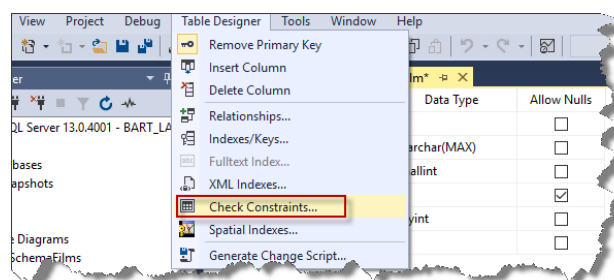


- Klik op Close.

SQL Server controleert of de ingegeven expressie correct kan geïnterpreteerd worden en zal eventueel foutboodschappen laten zien.

Wens je achteraf nog wijzigingen door te voeren aan de constraints dan open je de tabel terug in “Design”.

Via het nu beschikbare menuitem “Table Designer” selecteer je de optie “Check Constraints...”.



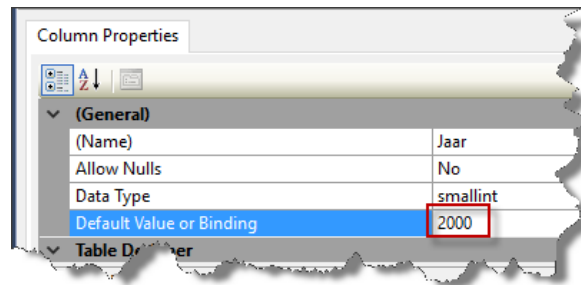
Via dit venster kan je constraints gaan verwijderen of wijzigen.

4.3.4 Defaults aanmaken en gebruiken

Tijdens de creatie van een tabel kan je bij de eigenschappen van een kolom een standaardwaarde (of default) opgeven. Deze waarde wordt dan automatisch toegekend wanneer een gebruiker een nieuwe rij toevoegt en voor deze kolom geen waarde opgeeft.

We nemen als voorbeeld terug de tabel “Film” waarin we als standaardwaarde voor de kolom jaar de waarde 2000 willen.

Dit kunnen we instellen door in de Design van de tabel de desbetreffende kolom aan te duiden en bij de eigenschap “Default Value or Binding” het getal 2000 in te geven.



We kunnen hier ook een functie ingeven. Willen we bijvoorbeeld standaard als jaartal het huidige jaartal instellen, dan vervangen we de waarde door de SQL-functie `YEAR(GETDATE())`.

Wanneer de functie nog complexer wordt en we willen deze waarde in meerdere tabellen gebruiken dan kan de waarde ook buiten de tabel gedefinieerd worden. Telkens als deze waarde nodig is, kan er een referentie naar gemaakt worden.

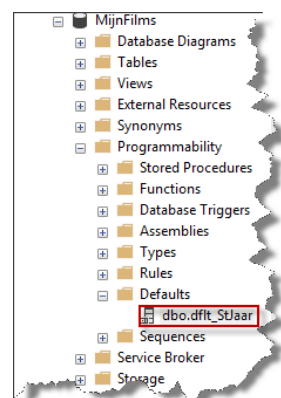
We kunnen hiervoor volgende stappen ondernemen:

- Klik bovenaan in de knoppenbalk “New Query” aan.
- Voorzie hierin volgende instructie:

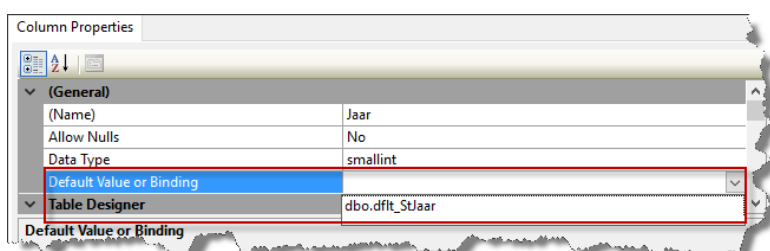
```
CREATE DEFAULT  
dflt_StJaar AS YEAR(GETDATE())-5
```

Met bovenstaande instructie wordt aangegeven dat je een algemene “Default” wenst aan te maken met de naam `dflt_StJaar` en als waarde het actuele jaar – 5.

- Klik op Execute zodat de instructie wordt uitgevoerd, onderaan verschijnt de boodschap “Command(s) completed successfully” wat aangeeft dat de default werd aangemaakt.
- Sluit het SQLQuery- venster, je hoeft de instructie niet op te slaan.
- Je vindt de aangemaakte default in het onderdeel “Defaults”.



- Editeer opnieuw de tabel: In de eigenschap “Default Value or Binding” kun je nu door middel van een keuzelijst de eerder aangemaakte Default gaan selecteren.



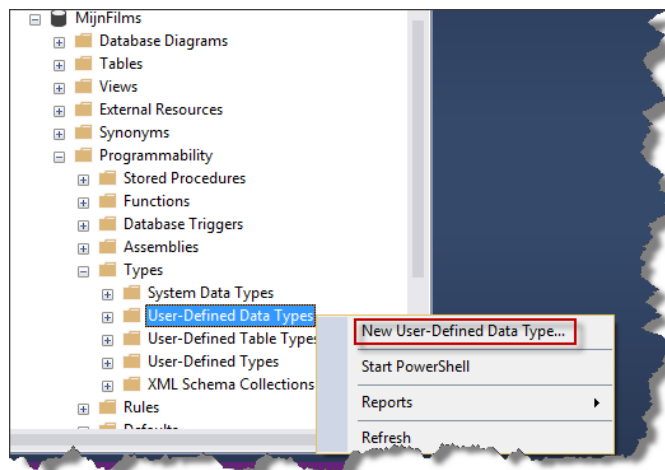
- Bewaar de tabel en sluit deze af.

4.3.5 User-defined data types (UDT)

Gebruikersgedefinieerde gegevenstypes zijn types die je zelf kan gaan aanmaken. Standaard zijn er voor het gegevenstype van een veld een aantal keuzes (int, varchar, ...). Deze gegevenstypes kan je verder gaan uitbreiden.

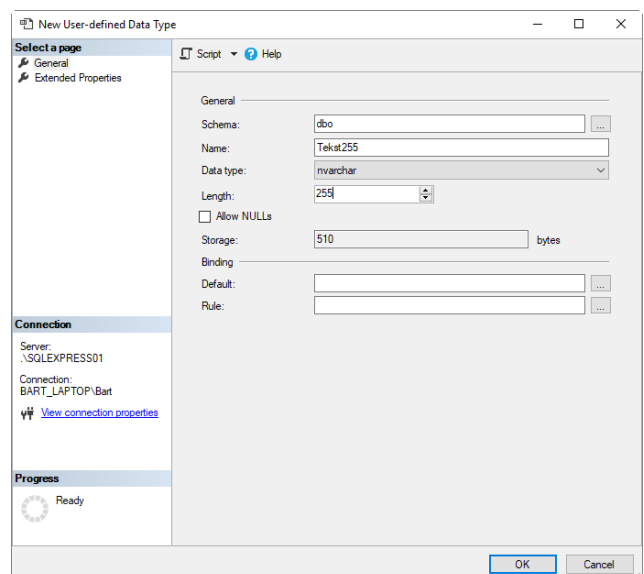
Hiervoor voer je volgende stappen uit:

- Klik met de rechtermuisknop op “User-defined Data Types” en kies in het snelmenu voor “New User Defined Data Type...”. Je vindt dit terug via <datasenaam> -> Programmability -> Types -> User Defined Data Types



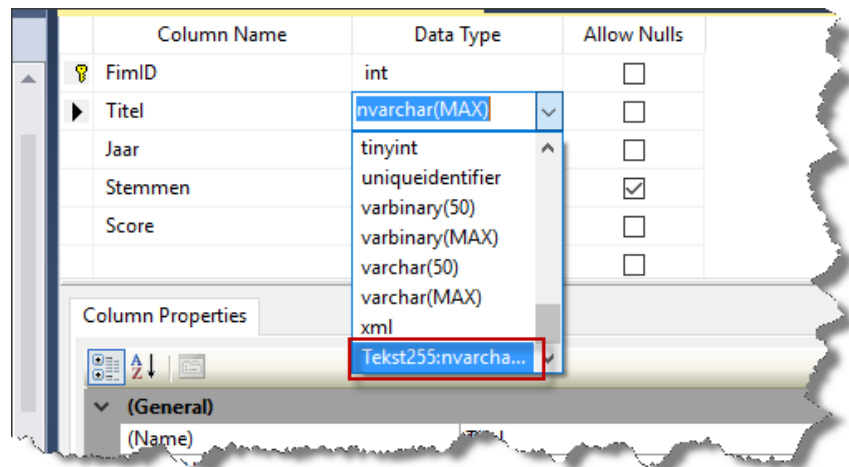
- Er wordt een nieuw venster geopend, waarin je het datatype kan gaan vastleggen.
 - Stel dat het type nvarchar(255) veel voorkomt. Wanneer je een standaard nvarchar gebruikt, worden bij default 50 karakters voorgesteld en je moet dit steeds gaan aanpassen. Je maakt een nieuw type aan met de naam “Tekst255” die eigenlijk niets anders is dan een voorstelling van een nvarchar(255).
 - Neem hiervoor volgende waarden over

Je kan hier ook een default instellen indien gewenst alsook rules gaan vastleggen (oude manier van constraints).



- Klik onderaan op OK.
- Het UDT verschijnt in de Object Explorer.
- Om dit toe te passen selecteer je bv. in de tabel “Film”, na selectie van “Design”, het veld Titel.

- Het datatype kan nu aangepast worden naar het UDT via de keuzelijst.



Het voordeel van gebruikersgedefiniëerde datatypes is dat je wanneer je een UDT zelf gaat wijzigen, deze wijziging meteen van toepassing is op alle kolommen uit alle tabellen die gebruik maken van dit datatype.

4.3.6 Indexen

De uitleg gebeurt aan de hand van een voorbeeld.

Stel: je hebt een tabel Klanten met 3 kolommen (**KlantID**, Naam, Gemeente) met de volgende inhoud:

Rec.Nr.	KlantID	Naam	Gemeente
1	1	Karel	Brugge
2	2	Wim	Brugge
3	3	Anne	Gent
4	5	Bart	Kortrijk
5	12	Hendrik	Gistel
6	13	Mia	Oostende

Je wenst regelmatig te gaan zoeken in deze tabel op basis van de naam van de klant.

Zoals de tabel nu geordend is, is de enige manier om een record terug te vinden op basis van de klantnaam, de tabel record per record te doorlopen en bij elk record te controleren of de waarde in het veld Naam overeenkomt met de zoekwaarde. Men spreekt in dit geval van sequentieel doorlopen of zoeken.

Karel zal in principe snel gevonden worden, maar dit is niet het geval voor Mia.

Wanneer je regelmatig een tabel moet doorzoeken op een specifieke veldnaam (in ons geval Naam) dan wordt de mogelijkheid geboden dit veld te indexeren.

Dit betekent dat een extra (onzichtbare) tabel wordt aangemaakt, met alleen maar de waarden uit te betrokken kolom, en dit op een gesorteerde manier. Naast de waarde wordt eveneens het recordnummer bewaard waar elke waarde kan teruggevonden worden in de originele tabel.

Als we dus het veld "Naam" indexeren, dan wordt volgende tabel aangemaakt:

Rec.Nr.	Naam	Rec.Nr in Klanten
1	Anne	3
2	Bart	4
3	Hendrik	5
4	Karel	1
5	Mia	6
6	Wim	2

Wanneer nu een zoekopdracht wordt gelanceerd op basis van klantnaam, dan zal het RDBMS, dat op de hoogte is van de indextabellen, niet in de tabel Klanten gaan zoeken maar in de indextabel. Omdat deze gesorteerd is op de zoekwaarde kan de tabel nu binair doorzocht worden, wat veel sneller is dan sequentieel.

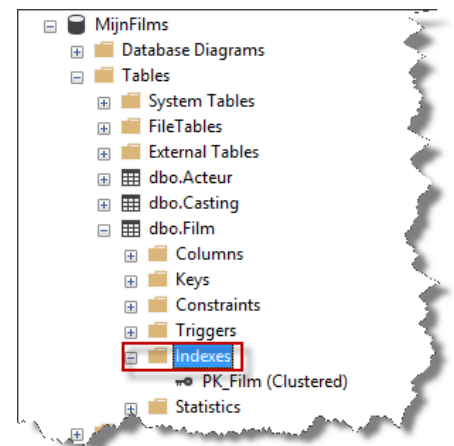
Eenmaal de waarde gevonden is, kent men ook meteen het recordnummer van het originele record in de Klanten tabel die dan ook op die manier razendsnel doorzocht kan worden.

Tot zover het goede nieuws in verband met indexen, maar dit systeem heeft ook een aantal nadelen :

- De omvang van de database zal duidelijk toenemen wanneer je veelvuldig indexen aanmaakt
- Het toevoegen, wijzigen en verwijderen van records zal meer tijd en resources in beslag nemen (de indextabellen moeten immers telkens bijgewerkt worden).

Indexen zijn uiterste interessante tools om te gebruiken, maar overdrijf hier niet mee.

We gaan in de tabel Film een nieuwe index aanmaken. Je vindt indexen terug onder <databasenaam> -> <tabelnaam> -> Indexes



Je merkt dat er reeds 1 index aanwezig is. Deze werd aangemaakt omdat het veld FilmID als primaire sleutel werd ingesteld.

We gaan nog een extra index voorzien op het veld "Titel" en volgen daarbij volgende werkwijze:

- Open via "Design" de tabel waar je de index wenst te voorzien
- Selecteer in de Menubalk onder "Table Designer" de optie "Indexes/Keys"
- Klik op "Add"
- Pas de kolomnaam en de naam van de Identity aan volgens onderstaand voorbeeld.
- Klik op Close en sla de tabel op (de index werd aangemaakt).

Je vindt de aangemaakte index in het Item Indexen.

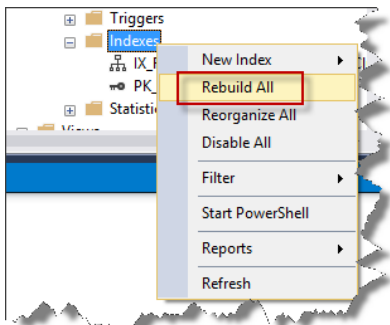


Eenmaal de index werd aangemaakt kan je deze niet meer wijzigen.

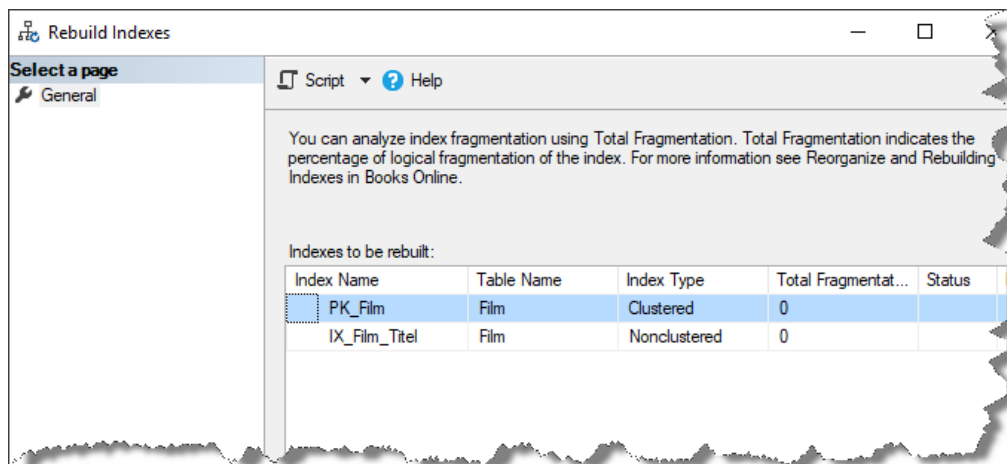
Wanneer je dit toch wenst te doen, verwijder je de index en je maakt die dan opnieuw aan.

Wat we tenslotte nog willen meegeven is dat het uitzonderlijk kan gebeuren dat de indextabellen niet meer overeenstemmen met de inhoud van de basistabel. Dit kan bijvoorbeeld gebeuren wanneer het toestel waarop de server draait onverwachts uitvalt.

Indextabellen (beschadigde) kun je opnieuw laten opbouwen door met de rechtermuisknop op het Item "Indexes" te klikken en in het snelmenu te klikken op "Rebuild All".



Er wordt een nieuw venster geopend, waarin je de indexen ziet die op de tabel van toepassing zijn.



Door onderaan op OK te klikken worden de indexen opnieuw aangemaakt. Hoe lang dit duurt is afhankelijk van de grootte van de tabellen.

4.4 Verwijderen tabellen

In de grafische tool is het eenvoudig om een tabel te verwijderen.

Wens je een tabel te verwijderen, dan volstaat het om met de rechtermuistoets op de tabel te klikken en in het snelmenu te kiezen voor de optie "Delete". Na een bevestigingsscherm waarin je dan "OK" kan aanklikken wordt de geselecteerde tabel verwijderd.

Wordt de tabel gebruikt in een relatie (zie verder) dan zal geweigerd worden de tabel te wissen, omdat anders de integriteit van de database in het gedrang komt.

4.5 Relaties

4.5.1 Wat zijn databasediagrammen?

Databasediagrammen zijn objecten in een database die een goede manier zijn om de structuur en de onderlinge relaties tussen tabellen in de database visueel weer te geven. We spreken hier over het databaseschema.

Je kan databasediagrammen aanmaken voor de volledige database maar ook voor een stuk uit de database.

Naast de visuele weergave die je krijgt van de database, is het ook mogelijk tegelijkertijd een aantal integriteitsregels en automatismen in te bouwen.

Integriteitsregels

Stel dat we een relatie willen definiëren tussen de tabel Film en Casting (Film.FilmID <-> Casting.FilmID). Na creatie van deze relatie zal je er niet meer in slagen om in het veld FilmID van de tabel Casting nog een waarde te geven die niet als waarde in het veld FilmID van de tabel Film voorkomt.

Automatismen

Tijdens de creatie van een relatie kan je eveneens opgeven dat, wanneer het record met de primaire sleutel in de ene tabel wijzigt of verwijderd wordt, de wijziging of verwijdering meteen moet doorgevoerd worden naar de tabel met de foreign key.

Nemen we het bovenstaande voorbeeld van de relatie tussen tabel Film en Casting op het veld FilmID. Wanneer je tijdens de creatie van de relatie aangeeft dat de records in de tabel Casting automatisch dienen bijgewerkt te worden (enforce relationships for Inserts en Updates), dan worden, wanneer er een record uit de tabel Film wordt gewist, alle records met dezelfde FilmID in de tabel Casting automatisch verwijderd.

Databasediagrammen zijn niet alleen een handig visueel hulpmiddel, maar ook een krachtige tool voor het onderhoud van je database.

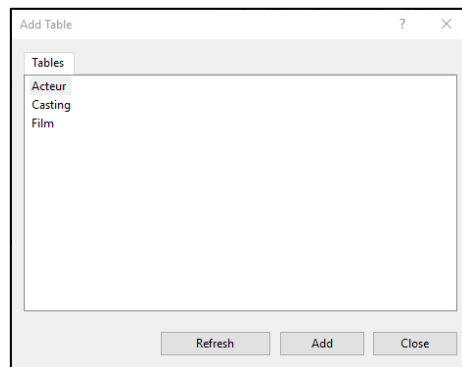
4.5.2 Databasediagrammen aanmaken

We gaan een databasediagram aanmaken. Hierbij dient opgemerkt dat vooraleer relaties kunnen worden aangemaakt tussen 2 of meerdere tabellen, in minstens 1 van de tabellen waar de relatie betrekking op heeft, een primaire sleutel moet gedefinieerd zijn.

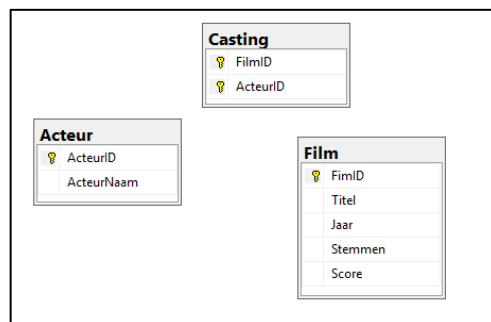
We volgen volgende stappen :

- Selecteer het onderdeel "Database Diagrams" in de database waar je een diagram wil maken (in ons geval MijnFilms).
- Als dit een eerste diagram is, zal je eerst moeten melden dat je weldegelijk een diagram wenst aan te maken.

- In het scherm selecteer je de tabellen die je in het diagram wenst te verwerken.

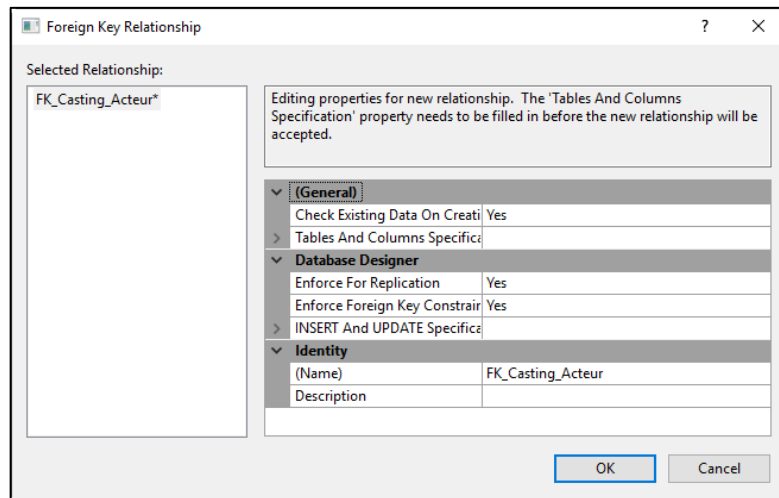


- Selecteer ze allemaal samen (of één voor één) en klik op Add. Ze verschijnen mogelijks wat random op het nieuw Diagram venster. Je kunt ze verslepen om ze te groeperen.



- Om een relatie aan te maken selecteer je de primaire sleutel (grijze vakje) van een tabel en sleep je deze naar de tabel waarnaar we een relatie willen.
- Er verschijnen 2 schermen nl. "Tables and Columns" en "Foreign Key Relationship".
 - In het eerste scherm leg je de relatie vast (van Primary Key naar Foreign Key).
 - Na controle klik je op OK.

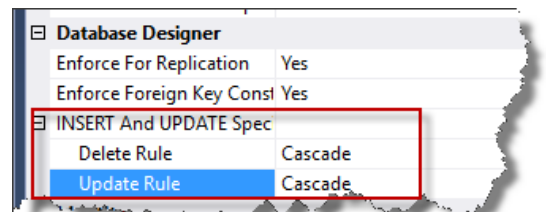
- In het 2^{de} scherm stel je de eigenschappen van de relatie in.



- “Check Existing Data on Creation”: met deze instelling wordt geforceerd dat je in de tabel Casting geen records kan aanmaken met een waarde voor ActeurID die niet in de tabel Acteur voorkomt. Dit laten we staan op “Yes”
- Wanneer je de eigenschap “INSERT and Update Specification” uitklapt dan zie je de eigenschappen “delete rule” en “update rule” verschijnen. Beiden hebben de waarde “No Action”

We kiezen voor beide eigenschappen de waarde “Cascade”: dit betekent dat, wanneer een acteur verwijderd wordt in de tabel Acteur, automatisch de records in de tabel Casting verwijderd worden met dezelfde ActeurID.

De update Rule heeft in ons geval weinig zin, gezien het nooit de bedoeling kan zijn om de waarde van een primaire sleutel te gaan wijzigen.



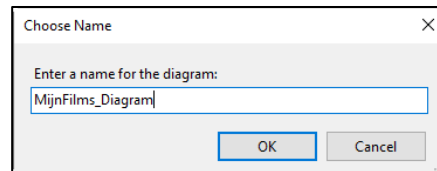
- De optie “Enforce For Replication” wordt later besproken. We geven mee dat het mogelijk is om data over verschillende SQL Servers te gaan verdelen. Met Replication is het dan mogelijk om toevoegingen, wijzigingen of verwijderingen van gegevens op 1 van de databaseservers automatisch te laten doorvoeren naar de andere databaseservers. Dit heeft enkel zin bij (zeer) grote databasetoepassingen.
- Wanneer je dit ingesteld hebt, klik je op OK. Je merkt dat er visueel een lijn getekend is tussen beide tabellen:



Aan de kant van Acteur komt een sleutel, om aan te geven dat de relatie in deze tabel gemaakt wordt op de primaire sleutel. Aan de kant van de tabel Casting komt een oneindig teken om aan te geven dat de relatie in deze tabel gemaakt wordt op de foreign key.

De lijn toont alleen aan dat er een relatie is tussen de tabellen. De lijn toont de betrokken velden niet noodzakelijk aan.

- Je kunt nu zelf de overige relatie(s) gaan instellen :
 - Film.FilmID -> Casting.FilmID
- Nadat de database is afgewerkt klik je op het bewaren-icoontje. Je geeft het diagram een duidelijke naam.



Je kunt binnen dezelfde database eventueel meerdere diagrammen aanmaken. Niet alle tabellen binnen een database hoeven immers tot dezelfde logische eenheid te behoren.

4.5.3 Databasediagrammen onderhouden

Relaties wijzigen

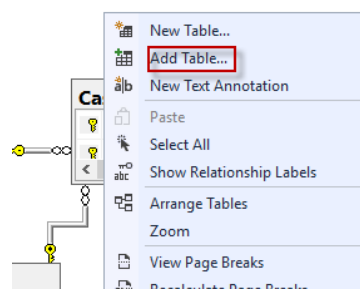
Eenmaal een databasediagram aangemaakt, kan dit achteraf nog gewijzigd worden.

Dit kan door met de rechtermuisknop op het naam van het diagram te klikken en de optie “Modify” te kiezen.

Een relatie zelf kan je niet wijzigen, wanneer je die toch wenst te wijzigen dan verwijder je eerst de bestaande relatie om deze dan opnieuw aan te maken. Een relatie kan gewist worden door met de rechtermuisknop op het relatielijntje te klikken en in het snelmenu de optie “Delete Relationship” te selecteren (vergeet de wijzigingen niet op te slaan).

Tabellen toevoegen

Wanneer je een andere bestaande tabel wenst te betrekken bij het databasediagram, kan klik je ergens op het witte vlak met de rechtermuisknop. In het snelmenu selecteer je de optie “Add Table...” waarna je de mogelijkheid krijgt een tabel toe te voegen. Na selectie van de tabel klik je op “Add” zodat de tabel wordt toegevoegd in het diagram.



Tabellen aanmaken in het diagram

Het is ook mogelijk om vanuit het diagram volledig nieuwe tabellen te gaan aanmaken. Je selecteert hiervoor in het snelmenu (na rechtsklik in het witte veld) de optie “New Table”.

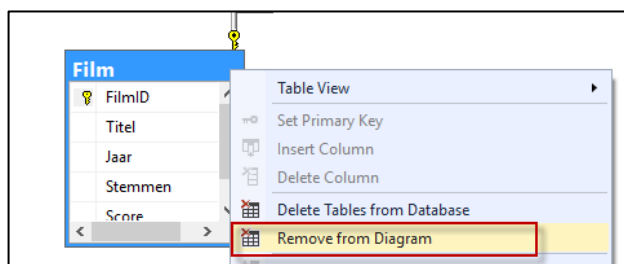
Na ingeven van de tabelnaam kan je de velden gaan definiëren.

Test *		
Column Name	Data Type	Allow Nulls
		<input type="checkbox"/>

Zolang bij de tabelnaam een sterretje staat, betekent dit dat de tabel nog niet bewaard werd. Dat gebeurt pas wanneer je het diagram bewaart.

Tabellen verwijderen uit het diagram

Klik met de rechtermuisknop op de tabel die je uit het diagram wenst te verwijderen en selecteer in het snelmenu de optie "Remove from Diagram". Hiermee worden ook meteen alle eventuele relaties die op deze tabel staan vernietigd.



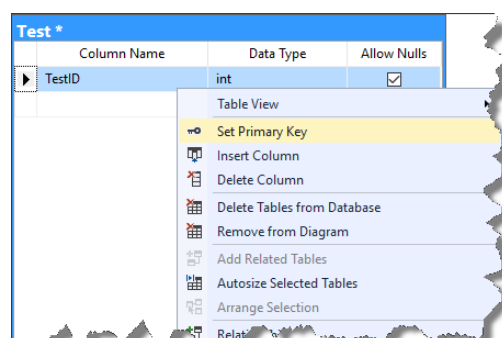
Tabellen verwijderen uit de database

Net zoals je nieuwe tabellen kunt aanmaken binnen het diagram kun je ook tabellen wissen uit de database zelf. Hiervoor selecteer je, nadat je op de betrokken tabel met de rechtermuisknop hebt geklikt, de optie "Delete Tables from Database".

Tabellen bewerken binnen het diagram

Binnen het diagram is het zelfs mogelijk om velden toe te voegen, te verwijderen, te wijzigen, primaire sleutels toe te voegen, indexen aan te maken,...

Klik hiervoor met de rechtermuisknop in een tabel. Je ziet hier de verschillende mogelijkheden.

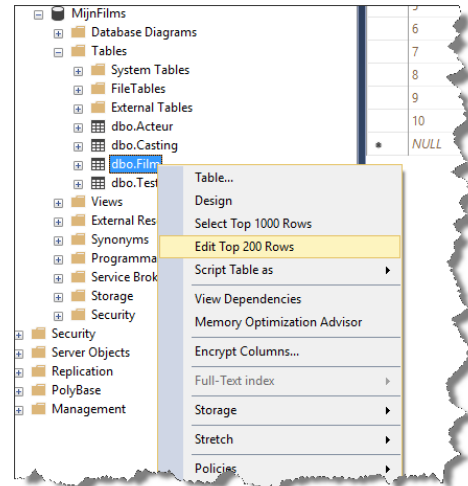


4.6 Invoeren data

Nadat we alle restricties hebben voorzien in onze database kunnen we data gaan invoeren.

Ingave gebeurt als volgt:

- Rechtsklik op de tabel die je wenst op te vullen en selecteer in het snelmenu “Edit top 200 Rows”

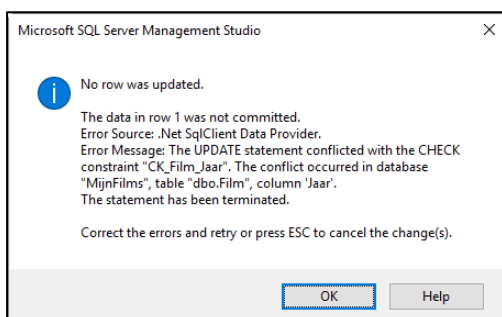


- In het rechtergedeelte wordt een nieuw venster geopend waarin alle records worden afgebeeld. Initieel is deze lijst leeg.
- Voer een record in (we testen eveneens de constraints):

	FilmID	Titel	Jaar	Stemmen	Score
	1	Test	1222	5	10

Het rode uitroepteken geeft aan dat de ingegeven informatie nog niet is opgenomen in de database. Dit zal pas gebeuren wanneer je een volgend record wenst in te geven.

Van zodra we naar een volgend record gaan, zal gepoogd worden de ingevoerde gegevens weg te schrijven in de tabel. Gezien we (bewust) het jaartal verkeerd hebben ingegeven verschijnt volgende foutboodschap.

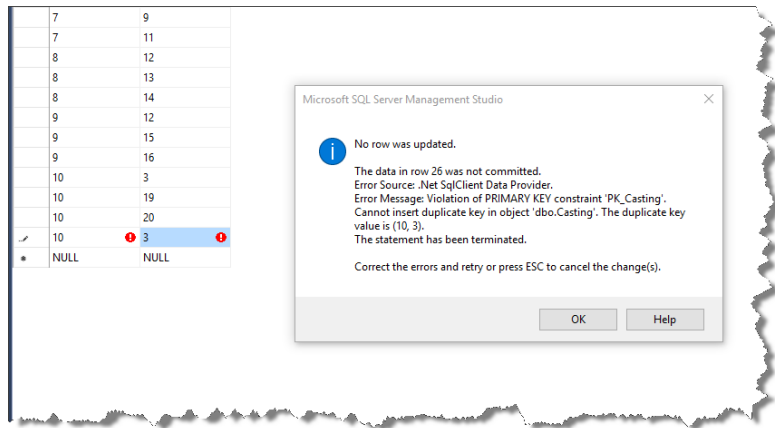


Er wordt aangegeven dat er data ingegeven werd in tegenspraak met de beperking op het jaartal. De data wordt in dit geval niet bewaard.

- Voer de records correct in voor de tabel Film
- Voer de records in voor de tabel Acteur
- Pas na het ingegeven van de tabellen Acteur en Film kunnen we de koppeltabel gaan invullen.

Let op:

- Combinatie van FilmID en ActeurID moet uniek zijn (samengestelde primaire sleutel)



- Er kunnen geen records worden toegevoegd van nog niet aangemaakte films of acteurs.