

6 SQL

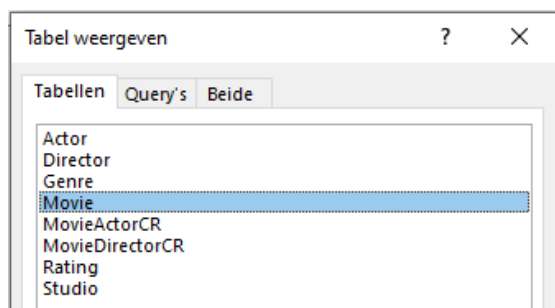
6.1 Inhoud tonen van een tabel

Het opvragen van gegevens doe je via SQL-opdrachten. We hebben al wat hiervan gezien in het vorige hoofdstuk, maar we gaan in dit hoofdstuk er dieper op in.

We hebben nu een Movie database gemaakt waar we in kunnen gaan kijken. Er zijn in Access mooie grafische hulp schermen die de onderstaande statements kunnen maken. Om de statements te maken is het handig om die te gebruiken, maar het gaat er in dit hoofdstuk vooral om dat je de SQL-statement begrijpt. Ook zou je zelf de SQL-opdrachten moeten kunnen maken. We maken een nieuwe SQL-Statement door in de ribben {Maken} de Queryontwerp te selecteren.

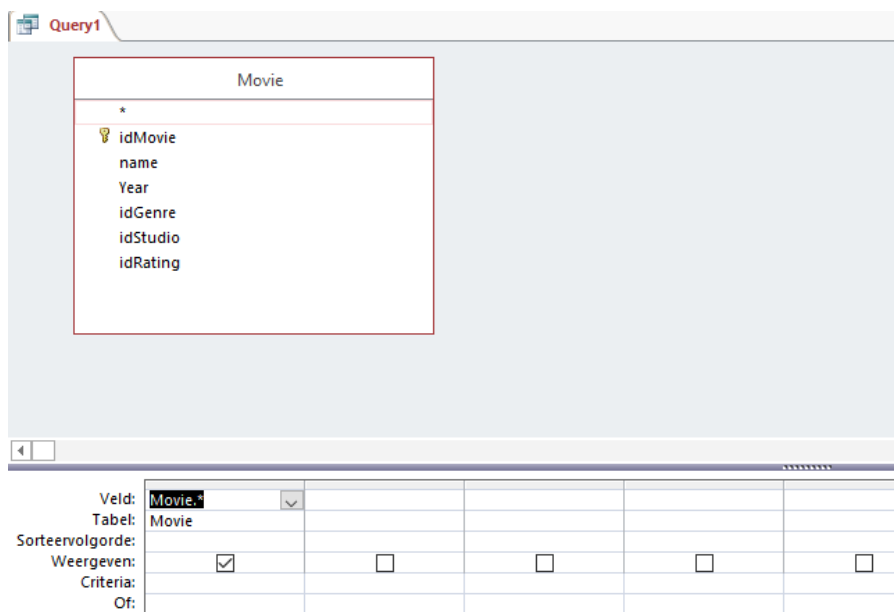


Het volgende scherm is een lijst van alle tabellen die we in de database hebben gemaakt.



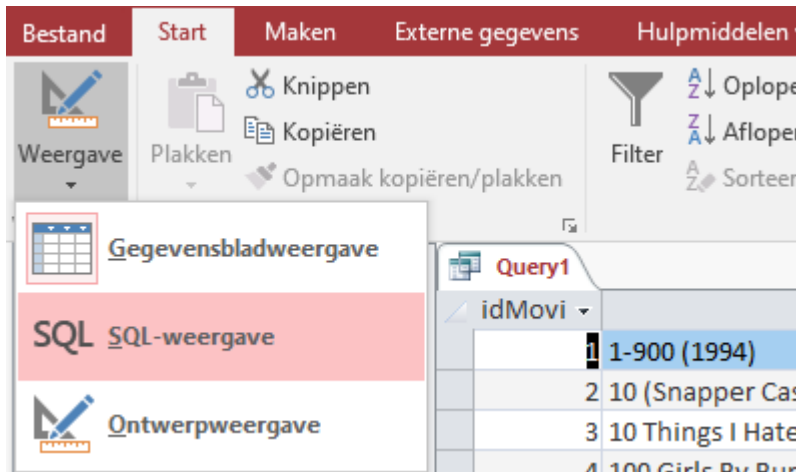
Als we nu op Movie drukken en op Toevoegen kunnen we dit scherm sluiten.

Het volgende wat we zien is de tabel Movie met zijn kolommen. Als we op de ster [*] drukken zien we onderaan het scherm die ster worden toegevoegd. Als we op de ontwerp-ribben op uitvoeren drukken dan wordt de inhoud van de tabel Movie getoond.



6.1.1 SELECT

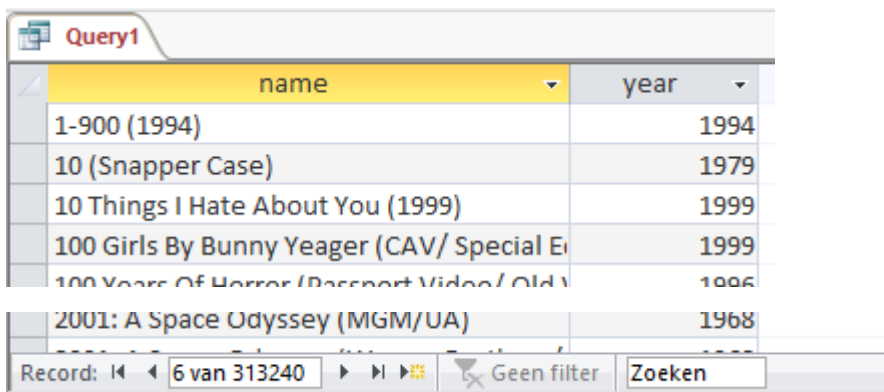
Als we het resultaat van de Movies bekijken kunnen we vervolgens op {Weergave}{SQL-Weergave} het SQL-commando zien.



```
SELECT *  
FROM Movie
```

Deze regel moet je als volgt lezen. Uit (FROM) de tabel Movie selecteren (SELECT) we alle velden. De ster [*] is een handige manier om even snel alle velden te tonen. **We moeten alleen erg uitkijken met deze manier van tonen.** De volgorde van de velden is afhankelijk van de volgorde hoe de velden in de database staan. Wordt de database uitgebreid met een extra veld, dan krijg je dit veld er automatisch later bij. Dit kan tot problemen leiden. De volgorde van de velden kunnen ook veranderd worden in de database. Het is daarom belangrijk dat je expliciet aangeeft welke velden je wilt zien. We vervangen de ster <*> met de velden {name} en {year}.

```
SELECT name, year  
FROM Movie
```



name	year
1-900 (1994)	1994
10 (Snapper Case)	1979
10 Things I Hate About You (1999)	1999
100 Girls By Bunny Yeager (CAV/ Special E	1999
100 Years Of Horror (Barnport Video/ Old	1996
2001: A Space Odyssey (MGM/UA)	1968

Onderaan het scherm zien we hoeveel records we terugkrijgen bij deze opdracht. Dit getal hadden we ook via een SQL-statement kunnen krijgen.

```
SELECT COUNT(name)  
FROM Movie;
```

Als we COUNT(name) na de select zetten dan zien we hoeveel resultaten er zijn.

6.1.2 DISTINCT

Met de instructie {DISTINCT} geven we aan dat we alleen unieke rijen willen zien. Als we alle {year} uit de {Movies} tabel bekijken krijgen we 313.240 record terug.

```
SELECT DISTINCT year  
FROM Movie |
```

Bij het uitvoeren van bovenstaande opdracht zien we goed het verschil in het aantal resultaten. Als we de opdracht uitvoeren zonder {DISTINCT} krijgen we 313.240 record terug, zetten we {DISTINCT} ervoor dan is het resultaat nog maar 116 records.

Hoeveel records krijg je terug als je nu het veld {name} voor {year} neerzet?

```
SELECT DISTINCT name, year  
FROM Movie;
```

We krijgen dan weer alle 313.240 record terug. Dit komt doordat {DISTINCT} kijkt of de gehele regel uniek is, niet alleen het veld dat erachter staat.

6.1.3 WHERE

Met een SQL-instructie willen we meestal niet alle resultaten terugzien maar een subset van de resultaten. We willen heel precies aangeven waar we geïnteresseerd in zijn. We zouden alles aan de database kunnen opvragen en in onze Python code alle resultaten weg kunnen gooien die we niet nodig hebben. Dit is alleen een oplossing die veel trager is dan gelijk met SQL het die vraag te stellen. Als we in Python hert filteren doen moet wel alle data eerst bij elkaar gezocht worden, en alle data moet dan misschien wel via een internetverbinding, en vervolgens via draadloze verbinding naar jouw programma gestuurd worden, en dan nog moet regel voor regel bekeken worden. Je DBMS is hier juist helemaal op geoptimaliseerd en weet via indexering waarschijnlijk al hoe dit filteren snel gedaan moet worden. Het filteren van de data wordt gedaan met de WHERE-instructie.

```
SELECT name  
FROM Movie  
WHERE year = 1999
```

Het veld {year} staat niet in het {SELECT}-gedeelte van de opdracht. Het is namelijk al bekend dat het resultaat 1999 zal zijn.

Voor het filteren van meerdere velden tegelijk kunnen we de het filter combineren met AND en OR instructies. In de tabel {GENRE} zien we dat Anime de sleutel = 4 heeft. Als we dan alle Anime uit 1999 willen zien doen we dat met de volgende instructie.

```
SELECT name  
FROM Movie  
WHERE year = 1999 AND idGenre = 4
```

In het {WHERE} gedeelte kunnen we de volgende operatoren gebruiken om de selectie mee te maken.

- = Gelijk
- > Groter dan
- < Kleiner dan
- >= Groter of gelijk aan
- <= Kleiner of gelijk aan

6.1.4 ORDER BY

We zien dat het resultaat uit het vorige hoofdstuk alle regels door elkaar heen staan. We zijn wel dat alle resultaten door elkaar heen staan.

name
Monster Rancher (A.D. Vision) #1: Let The Games Begin!
Monster Rancher (A.D. Vision) #2: Catch A Tiger By The Tail!
Pokemon #18: Water Blast!
Pokemon: The First Movie (Special Edition)
Sol Bianca: The Legacy #1: Lost Treasures
Sonic The Hedgehog: The Movie (Old Version)
Tenchi Muyo: The Movie 3: Tenchi Forever (Old Version)
Wild Cardz (Old Version)
Sol Bianca: The Legacy #2: Separation
Jubei-Chan: The Ninja Girl (Bandai Entertainment) #1: Secret Of The Lonely Eyepatch
Pokemon #21: The Po-Ke Corral!
Pokemon #22: Hang Ten, Pikachu
Jubei-Chan: The Ninja Girl (Bandai Entertainment) #2: Basic Ninja Training (Old Versi

We zouden denken dat het resultaat in de volgorde staat van hoe het in de tabel staat opgeslagen. Dit is vaak het geval, maar dat is niet per definitie zo. Zolang er geen ORDER BY in de instructie staat is de volgorde altijd onbepaald. Je hebt dus geen garantie van de volgorde. Het kan zijn dat je instructie via verschillende DBMS-systemen gevonden wordt. Degene die als eerste klaar is geeft gelijk zijn resultaat, en dat wordt getoond. Als een DBMS net even wat drukker is omdat deze ook een andere opdracht nog moest uitvoeren kan deze die normaal als eerste klaar is opeens later klaar zijn, met als gevolg dat zijn gegevens later in de lijst staan. Is de volgorde van belang dat moeten we dat meegeven aan de SQL-instructie door middel van {ORDER BY}.

```
SELECT name
FROM Movie
WHERE year = 1999 AND idGenre = 4
ORDER BY name
```

Het resultaat zijn dezelfde regels, alleen krijgen we nu het resultaat in alfabetische volgorde.

name
A.D. Police: To Protect And Serve
A.Li.Ce (a.k.a. Alice/ Special Edition)
Amazing Nurse Nanako #1: Memories Of You
Amazing Nurse Nanako #1: Memories Of You (Signature Series)
Amazing Nurse Nanako #2: Fire-Crackers
Amazing Nurse Nanako #3: The Last Spiral
Angel Links (Bandai Entertainment) #1 - 4 (Collector's Edition)
Angel Links (Bandai Entertainment) #1 - 4 (Collector's Edition / Old Version)

Standaard is de volgorde oplopend, oftewel *ascending*. We kunnen dit ook meegeven door ASC achter ieder veld in de ORDER BY instructie te zetten. Als we het resultaat in omgekeerde volgorde willen dan gebruiken we de term *descending* oftewel DESC.

Achter de naam kan nog ASC of DESC gezet worden. Dit staat voor ASCENDING of te wel opvolgend of DESCENDING oftewel omgekeerd.

```
SELECT name
FROM Movie
WHERE year = 1999 AND idGenre = 4
ORDER BY name DESC
```

In onderstaand voorbeeld willen we {year} en {name} tonen van alle Anime gemaakt na 1999. {year} willen we in omgekeerde (DESC) volgorde tonen, en dan {name} is alfabetische volgorde.

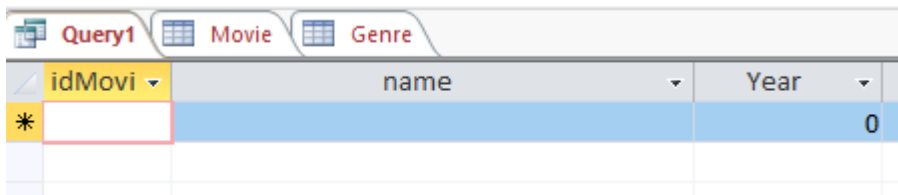
```
SELECT year, name
FROM Movie
WHERE year > 1999 AND idGenre = 4
ORDER BY year DESC, name
```

6.1.5 LIKE

We hebben in de voorgaande paragrafen een selectie gemaakt met de operatoren groter, kleiner en gelijk. Dit is bij getallen handig, maar we hebben heel veel te maken met tekst in onze database.

We willen filteren op Star Wars in de {Movie} tabel. Als we nu de volgende instructie doen dan zouden we kunnen denken dat er geen Star Wars dingen in onze database zitten.

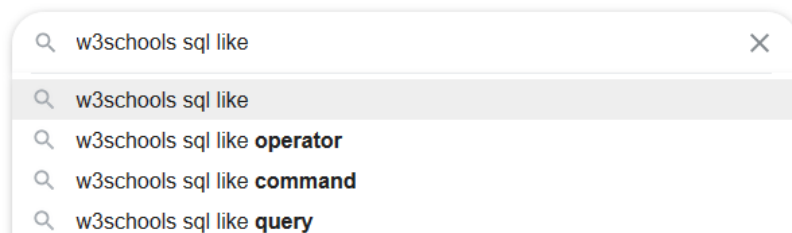
```
SELECT *  
FROM Movie  
WHERE name = "Star Wars"
```



The screenshot shows a database interface with tabs for 'Query1', 'Movie', and 'Genre'. Below the tabs is a table with columns 'idMovi', 'name', and 'Year'. The 'idMovi' column has a dropdown menu with an asterisk (*) selected. The 'name' column has a dropdown menu. The 'Year' column has a dropdown menu. The table shows 0 rows.

Er staat ook geen data in met precies de naam "Star wars". We willen dus zoeken waar het woord "Star Wars" in voorkomt.

In SQL gebruiken we hiervoor de operator 'like'. Ook SQL-opdrachten kan je opzoeken in W3Schools zoals we dat ook met Python deden.



Om de "Star wars" te zoeken moeten we dat dan doen met de onderstaande instructie waarna we 126 resultaten terugkrijgen uit de database.

```
SELECT *  
FROM Movie  
WHERE name LIKE "**Star Wars**"
```

6.1.6 NULL

Als we naar de *Movie*-tabel kijken dan is het veld *year* gedefinieerd als een lange integer, en op vereist staat Nee.

Movie		
	Veldnaam	Gegevenstype
idMovie		Numeriek
name		Korte tekst
year		Numeriek
idGenre		Numeriek
idStudio		Numeriek
idRating		Numeriek
Algemeen Opzoeken		
Veldlengte	Lange integer	
Notatie		
Aantal decimalen	Automatisch	
Invoermasker		
Bijschrift		
Standaardwaarde	0	
Validatieregel		
Validatietekst		
Vereist	Nee	
Geïndexeerd	Nee	
Tekstuitlijning	Algemeen	

Bekijken we de gegevens die in de database staan dan zien we ook dat niet bij ieder veld een waarde staat ingevuld. Deze lege velden zijn NULL-velden. We hebben hier dus geen 0 ingevoerd maar een NULL oftewel een leeg veld. Zo'n leeg veld moet je in een database lezen als : Onbekend.

Movie							
	idMovie	name	year	idGenre	idStudio	idRating	kl
+	634	Black Sabbath: The Best Of Musikladen	1999	20	6203	1	
+	635	Blondie: The Best Of Musikladen (Old Ver	1999	20	6203	1	
+	636	Dr. Hook & The Medicine Show: The Best (20	6203	1	
+	637	Hall & Oates: The Best Of Musikladen (Old	1977	20	6203	1	
+	638	Ike & Tina Turner: The Best Of Musikladen		20	6203	1	
+	639	Kool & The Gang: The Best Of Musikladen		20	6203	1	
+	640	Best Of Musikladen: Ladies Of Rock		20	6203	1	
+	641	Procol Harum: The Best Of Musikladen (Ol	1971	20	6203	1	
+	642	Roxy Music / T. Rex: The Best Of Musiklad	1998	20	6203	1	
+	643	Stephen Stills And Manassas: The Best Of		20	6203	1	
+	644	Yes: The Best Of Musikladen (Old Version		20	6203	1	
+	645	Best Of Ocean Adventures		28	6203	1	
+	646	Best Of Sci-Fi DVD: Cyborg 2 / Death Mach		25	8165	12	
+	647	Sessions At West: Best Of Sessions At We:	1997	20	7451	1	
+	648	Andy Griffith Show (United American): Be		32	8308	1	
+	649	Best Of The Best #1 & #2		1	1285	12	
+	650	Blues Brothers: Best Of The Blues Brother	1993	9	8308	1	
+	651	Chris Rock Show: Best Of The Chris Rock S	1999	27	2500	1	

Figuur 11: Null velden in year

Nu willen we alleen alle NULL-velden uit de database selecteren. Hiervoor hebben we een aparte methode.

```
SELECT *  
FROM Movie  
WHERE year is NULL
```

Als we alleen de *Movies* willen die een jaartal hebben, dan moeten we precies het tegenovergestelde vragen.

```
SELECT *  
FROM Movie  
WHERE year is not NULL
```

Omdat we het NULL-veld moeten lezen als “Onbekend” kan het zijn dat je alle films wilt hebben uit bijvoorbeeld 1984 of waarvan het niet bekend is uit welk jaar die film is.

```
SELECT *  
FROM Movie  
WHERE year = 1984 OR year is NULL
```

Bij getallen is het duidelijk te zien dat een veld niet is ingevuld en daardoor onbekend is.

6.1.7 Opdrachten

Van alle opdrachten moet je de SQL-Statement bepalen. En de lijst van resultaten.

1. Zoek in de {Movie} tabel alle films met “Matrix” in de titel (114 resultaten).
2. Van de Matrix films, toon alleen de resultaten waar een jaar van bekend is. (72 resultaten)
3. Sorteer de resultaten van opdracht 2 op jaartal, de nieuwste eerst.
4. Van opdracht 3 ben ik alleen geïnteresseerd in het “SciFi” genre. Filter het resultaat. (46 resultaten)
5. Toon in het resultaat van opgaven 4 alleen de verschillende idStudio (3 resultaten)
6. Welke Games (idGenre) hebben we uit het jaar 2007? (16 resultaten)
7. Hoeveel films zijn er gemaakt in je geboortjaar? Gebruik hiervoor de COUNT-instructie.
8. Zijn er films met jouw voornaam in de titel, of een acteur of director met jouw voornaam?
9. Hoeveel acteurs hebben als voornaam alleen “Kate” (333 resultaten)
10. Sorteer het resultaat uit 9 op naam.
11. In hoeveel films heeft “Kate Winslet” gespeeld? (146 resultaten)
12. Is zij ook de regisseur (Director) van een film?
13. Hoeveel met alleen de voornaam Kate hebben een film geregisseerd? (22 resultaten)