

Database

Wat is een database?

Heel wat jaren terug in de tijd, toen er nog nauwelijks computers gebruikt werden, werden belangrijke gegevens bijgehouden in een kaartenbak.



Op een kaart werd informatie geschreven over bijvoorbeeld een persoon. De kaarten werden op alfabet gesorteerd (op naam van de personen) in de bak gezet, zodat ze weer snel en eenvoudig teruggevonden konden worden.

Een database is niets meer dan een digitale kaartenbak. Deze digitale kaartenbak bestaat uit tabellen met gegevens. Elke tabel is te vergelijken met de kaartenbak, waarbij elke rij gegevens in die tabel een afzonderlijke kaart is.

Eenvoudige SQL-commando's

Er zijn verschillende soorten SQL-commando's. Je kan met een commando gegevens uit de database opvragen, nieuwe gegevens invoeren in de database, gegevens aanpassen of verwijderen. In deze opdracht zullen we alleen kijken naar het commando om gegevens uit de database op te halen.

Het SQL-commando om gegevens uit de database op te halen heeft de volgende structuur (In het voorbeeld staat het in hoofdletters, maar dat hoeft niet. Het commando staat nu ook op meerdere regels, maar ook dat is niet noodzakelijk. Het mag ook op 1 regel):

```
SELECT <kolomnaam of namen>  
FROM <tabelnaam of namen>  
WHERE <bepaalde voorwaarde>
```

We zullen een voorbeeld bekijken. Hieronder staat een voorbeeldtabel waar we een commando op los gaan laten:

Tabelnaam: studenten

id	voornaam	achternaam	adres	plaats	postcode
2	Isabella	van der Smeede	den Teulingsingel 4-7	Giessen	6823JD
3	Annabel	Sanders	Köseweg 41-w	Julianadorp	7942JG
4	Collin	Vos	van de Bergsteeg 94-13	Delfzijl	4614GX
6	Nadia	Maaswinkel	de Graafsteeg 24-70	Enschede	2631VK
7	Stan	Hussain	Chevressonsingel 1i	Froombosch	3015XB
8	Xavi	van de Ven	Kleinlaan 5	Stieltjeskanaal	8573WP
9	Brian	Lansink	Veenstrahof 35i	Margraten	6165XE
10	Zoey	de Haan	Aschmansingel 25-o	Cromvoirt	4131NE
11	Aron	Beourgeois	van Wijlandsingel 60u	Heukelum	1339VJ
12	Ravi	Scherms	van de Bergpad 53-f	Almen	6942LX
13	Bilal	Ulrich	Dijkstrastraat 5092	Vinkenbuurt	9752BK
14	Yfke	van der Leek	Broekhovenpad 95-64	Gelderswoude	1334DP
15	Elisa	van Amstel	Zhangweg 9	Netterden	9752BK
16	Fenna	van Leuven	Dekkersingel 39j	Sluis	4814NJ
17	Senna	Beernink	Sardjoering 1u	Nij Beets	1333EJ
18	Lieve	Ünal	Özcanstraat 2-5	Volkel	1072NL
19	Nora	Vos Specht	van den Heuvelweg 8998	Hollandscheveld	1509GP
20	Fabian	Bosch	de Leeuwring 5	Mantgum	4131NE

Bovenstaande tabel heeft de tabelnaam studenten en bestaat uit 6 kolommen met de kolomnamen id, voornaam, achternaam, adres, plaats en postcode.

Voorbeeld 1:

Als we willen weten wat de woonplaats van Xavi van de Ven is kunnen we het volgende SQL-commando geven:

```
SELECT plaats FROM studenten WHERE voornaam = 'Xavi' AND achternaam = 'van de Ven'
```

Het antwoord van de database is ook altijd een tabel (In dit geval van 1 rij en 1 kolom):

plaats
Stieltjeskanaal

In het WHERE-gedeelte zie je dat we zoeken op basis van twee voorwaarden, namelijk de voornaam en de achternaam. Deze voorwaarden worden gescheiden door het woord AND. In dat geval moeten beide voorwaarden gelden.

Voorbeeld 2:

In het volgende voorbeeld bestaat het resultaat uit meerdere rijen en kolommen en worden in het WHERE-deel de voorwaarden gescheiden door het woord OR.

```
SELECT voornaam, achternaam FROM studenten WHERE plaats = 'Volkel' OR plaats = 'Delfzijl'
```

De gestelde vraag is: 'Wie woont er in Volkel of in Delfzijl'

Op deze vraag is het antwoord van de database:

voornaam	achternaam
Collin	Vos
Lieve	Ünal

In het Select-deel worden de kolomnamen opgenomen die we in het resultaat willen hebben, steeds met een komma ertussen.

Voorbeeld 3:

In dit voorbeeld maken we gebruik van de kolom id. Dit is een kolom waarbij elke student een unieke waarde toegekend krijgt. Deze is te vergelijken met jouw studentnummer. Zo'n unieke waarde wordt ook wel een **primaire sleutel** genoemd.

```
SELECT voornaam, achternaam FROM studenten WHERE id = 10
```

Met deze query wordt de volgende vraag beantwoord: 'Wat is de voor- en achternaam van de student met nummer 10.'

Het antwoord is dan:

voornaam	achternaam
Zoey	de Haan

In het WHERE-deel staat id = 10 in plaats van id = '10'.

Als een bepaalde waarde in de database als getal is opgeslagen (zoals de id's van de studenten), dan mogen de ' tekens niet gebruikt worden. In alle andere gevallen moeten ze er wel staan.

Voorbeeld 4:

Als je alle kolommen uit de database in het antwoord wilt hebben, kan je een * in het SELECT-deel zetten. Het commando behorende bij de vraag 'geef alle gegevens van de student met nummer 10' wordt dan:

```
SELECT * FROM studenten WHERE id = 10
```

Het resultaat ziet er dan als volgt uit:

id	voornaam	achternaam	adres	plaats	postcode
10	Zoey	de Haan	Aschmansingel 25-o	Cromvoirt	4131NE

Voorbeeld 5:

Je kan in het WHERE ook vergelijken of een waarde groter of kleiner is dan een bepaalde waarde. Het commando op de vraag 'geef alle gegevens van de studenten met een id kleiner dan 10' wordt dan

```
SELECT * FROM studenten WHERE id < 10
```

Het resultaat ziet er dan als volgt uit:

id	voornaam	achternaam	adres	plaats	postcode
2	Isabella	van der Smeede	den Teulingsingel 4-7	Giessen	6823JD
3	Annabel	Sanders	Köseweg 41-w	Julianadorp	7942JG
4	Collin	Vos	van de Bergsteeg 94-13	Delfzijl	4614GX
6	Nadia	Maaswinkel	de Graafsteeg 24-70	Enschede	2631VK
7	Stan	Hussain	Chevressonsingel 1i	Froombosch	3015XB
8	Xavi	van de Ven	Kleinlaan 5	Stieltjeskanaal	8573WP
9	Brian	Lansink	Veenstrahof 35i	Margraten	6165XE

Andere vergelijkingsopties kan je vinden op:

https://www.w3schools.com/sql/sql_where.asp

Uit bovenstaande voorbeelden kunnen we halen dat je altijd bepaalde kennis van de tabellen in de database nodig hebt. Hieronder een opsomming van de gegevens die je nodig hebt:

- Naam van de tabel of tabellen
- Namen van de kolommen
- Wat is er in die kolom opgeslagen, een getal of een tekst (er zijn nog meer mogelijkheden, maar die bekijken we nu niet)?
- Welke kolom bevat de unieke waarde (primaire sleutel)?

SQL-commando's met sortering

Het resultaat van een SQL-commando levert, zoals eerder aangegeven, weer een tabel op. Deze resultatentabel kan gesorteerd worden. Dit kunnen we in het SQL-commando aangeven.

Hieronder een aantal voorbeelden (uitgevoerd op de studententabel uit de eerdere voorbeelden).

Voorbeeld 6:

```
SELECT * FROM studenten WHERE id < 10 ORDER BY id DESC
```

Met ORDER BY geef je aan op basis van welke kolom de sortering moet plaatsvinden. Door DESC (descending) erachter te zetten, geef je aan dat de sortering van hoog naar laag moet plaatsvinden. Het resultaat van deze sortering wordt dan:

id ▼ 1	voornaam	achternaam	adres	plaats	postcode
9	Brian	Lansink	Veenstrahof 35i	Margraten	6165XE
8	Xavi	van de Ven	Kleinlaan 5	Stieltjeskanaal	8573WP
7	Stan	Hussain	Chevressonsingel 1i	Froombosch	3015XB
6	Nadia	Maaswinkel	de Graafsteeg 24-70	Enschede	2631VK
4	Collin	Vos	van de Bergsteeg 94-13	Delfzijl	4614GX
3	Annabel	Sanders	Köseweg 41-w	Julianadorp	7942JG
2	Isabella	van der Smeede	den Teulingsingel 4-7	Giessen	6823JD

Voorbeeld 7:

```
SELECT * FROM studenten WHERE id < 10 ORDER BY achternaam, voornaam
```

Je kan ook sorteren op alfabetische volgorde en meerdere kolommen. In bovenstaande commando wordt er eerst gesorteerd op achternaam. Indien een bepaalde achternaam meerdere keren voorkomt (dit is nu niet het geval), worden deze rijen gesorteerd op de voornaam.

id	voornaam ▲ 2	achternaam ▲ 1	adres	plaats	postcode
7	Stan	Hussain	Chevressonsingel 1i	Froombosch	3015XB
9	Brian	Lansink	Veenstrahof 35i	Margraten	6165XE
6	Nadia	Maaswinkel	de Graafsteeg 24-70	Enschede	2631VK
3	Annabel	Sanders	Köseweg 41-w	Julianadorp	7942JG
8	Xavi	van de Ven	Kleinlaan 5	Stieltjeskanaal	8573WP
2	Isabella	van der Smeede	den Teulingsingel 4-7	Giessen	6823JD
4	Collin	Vos	van de Bergsteeg 94-13	Delfzijl	4614GX

SQL-commando's op meerdere tabellen

Het komt regelmatig voor dat de gegevens die je uit een database wilt halen in meerdere tabellen gevonden moeten worden. In het voorbeeld van de studententabel kan er een tweede tabel bestaan met daarin cijfers voor een bepaald vak. Een deel van deze tabel ziet er als volgt uit.

Tabelnaam: cijfers

cijferID	studentID	vak	cijfer
1	2	wiskunde	7
2	2	Engels	5
3	2	Nederlands	6
4	3	wiskunde	4
5	3	Engels	7
6	3	Nederlands	9
7	4	wiskunde	6
8	4	Engels	6
9	4	Nederlands	7
10	4	Nederlands	7

Tabelnaam: studenten

id	voornaam	achternaam	adres
2	Isabella	van der Smeede	den Teu
3	Annabel	Sanders	Kösewe
4	Collin	Vos	van de l
6	Nadia	Maaswinkel	de Graa
7	Stan	Hussain	Chevres
8	Xavi	van de Ven	Kleinlaa
9	Brian	Lansink	Veenstr

In deze tabel staan niet de namen van de studenten, maar een verwijzing naar het id in de studententabel. Student met studentID 2 kan in de andere tabel (de studententabel) teruggevonden worden. Het gaat dan om Isabella van der Smeede. De verwijzing naar de andere tabel is altijd een verwijzing naar de unieke kolom in de andere tabel (het id).

Hoe je gegevens uit meerdere tabellen kan halen via SQL-commando's zie je in de volgende voorbeelden:

Voorbeeld 8:

```
SELECT voornaam, vak , cijfer
FROM studenten, cijfers
WHERE studenten.id = cijfers.studentID
```

In het SELECT-deel geef je aan welke kolommen je in het resultaat wilt opnemen. Deze kunnen uit verschillende tabellen gehaald worden. Als er in de tabellen die je gebruikt gelijke kolomnamen voorkomen, moet je de kolom die je wilt gebruiken aanduiden door eerst de tabelnaam te noemen gevolgd door een punt en dan de kolomnaam (zoals in het WHERE-deel gedaan is).

In het FROM-deel noem je alle tabellen die je gebruikt met steeds een komma ertussen.

In het WHERE-deel stel je de waarde van de koppelkolom uit de ene tabel gelijk aan de unieke (primaire sleutel) waarde uit de andere tabel. In dit voorbeeld zijn de twee tabellen

gekoppeld op basis van het studenten ID. Daarom pak je het id uit de studententabel en stel je die gelijk aan het studentID uit de cijfertabel.

Het resultaat van deze query ziet er als volgt uit:

voornaam	vak	cijfer
Isabella	wiskunde	7
Isabella	Engels	5
Isabella	Nederlands	6
Annabel	wiskunde	4
Annabel	Engels	7
Annabel	Nederlands	9
Collin	wiskunde	6
Collin	Engels	6
Collin	Nederlands	7
Collin	Nederlands	7

Voorbeeld 9:

Voor het antwoord op de vraag 'voor welk vak en door wie is er een onvoldoende gehaald?' kan je de volgende query gebruiken:

```
SELECT voornaam, vak  
FROM studenten, cijfers  
WHERE studenten.id = cijfers.studentID  
AND cijfer < 6
```

Het resultaat hiervan is:

voornaam	vak
Isabella	Engels
Annabel	wiskunde

Er komen ook nog andere koppelingen tussen drie tabellen voor, waarbij 1 tabel de koppeling vormt tussen de andere tabellen. Om hier een voorbeeld van te geven voegen we eerst twee extra tabellen aan onze voorbeelddatabase toe:

Tabelnaam: cursus

ID	naam	beschrijving
1	HTML5	Een leuke cursus over HTML
2	CSS3	Een leuke cursus over CSS
3	JavaScript	Een leuke cursus over JavaScript
4	PHP	Een leuke cursus over PHP
5	Laravel	Een leuke cursus over Laravel

Tabelnaam: cursus_deelname

cursusID	studentID
1	6
1	8
1	12
1	16
3	3
3	7
3	16

De studenten in ons voorbeeld kunnen kiezen voor een extra aangeboden cursus. De beschikbare cursussen staan in de tabelcursus. In de tabelcursus_deelname staan de inschrijvingen voor een bepaalde cursus. Op de eerste regel zie je dat voor cursus 1 (de HTML5-cursus) student nummer 6 (Nadia Maaswinkel) ingeschreven staat. Voor deze cursus staan ook student 8, 12 en 16 ingeschreven (regel 2 t/m 4).

De laatste tabel is dus de koppeling tussen de studententabel en de cursustabel.

Tabelnaam: cursus

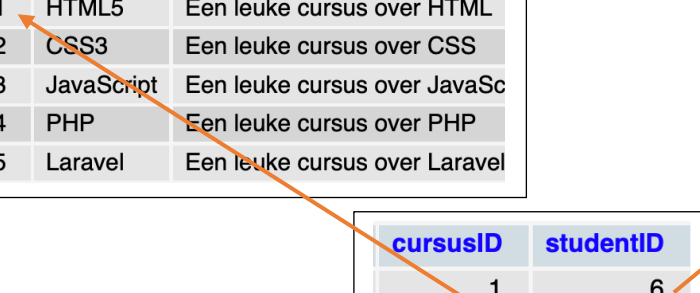
ID	naam	beschrijving
1	HTML5	Een leuke cursus over HTML
2	CSS3	Een leuke cursus over CSS
3	JavaScript	Een leuke cursus over JavaScript
4	PHP	Een leuke cursus over PHP
5	Laravel	Een leuke cursus over Laravel

Tabelnaam: studenten

id	voornaam	achternaam	ad
2	Isabella	van der Smeede	de
3	Annabel	Sanders	Kö
4	Collin	Vos	va
6	Nadia	Maaswinkel	de
7	Stan	Hussain	Ch
8	Xavi	van de Ven	Kle
9	Brian	Lansink	Ve

Tabelnaam: cursus_deelname

cursusID	studentID
1	6
1	8
1	12
1	16
3	3
3	7
3	16



Voorbeeld 10:

Om uit de database te halen wie er voor de cursus HTML5 ingeschreven staat, kunnen we de volgende query maken:

```
SELECT voornaam, achternaam  
FROM studenten, cursus, cursus_deelname  
WHERE studenten.id = cursus_deelname.studentID  
AND cursus_deelname.cursusID = cursus.ID  
AND cursus.naam = 'HTML5'
```

Hier geldt hetzelfde als in de vorige voorbeelden:

In het SELECT-deel noem je alle kolommen die je wilt opnemen in het resultaat.

In het FROM-deel noem je alle tabellen die je gebruikt.

In het WHERE- deel maak je nu twee keer een koppeling tussen twee tabellen. Hierbij is altijd de koppeltabel 1 van de 2 die je in de gelijkstelling gebruikt.