# 3 Raadplegen van gegevens

Leerdoelen

Na het bestuderen van dit hoofdstuk wordt van je verwacht dat je:

* met de select … from … where … kunt werken
* weet wat een conditie is en ermee kunt werken
* weet wat een logische operator is en ermee kunt werken
* weet wat een relatieteken is en ermee kunt werken
* de records kunnen sorteren

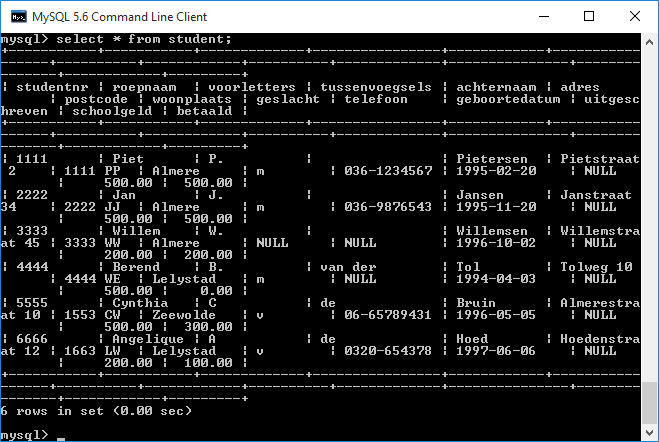
Er is een DBMS geïnstalleerd. Daarin hebben we de database school aangemaakt met daarin de tabel student. Aan die tabel hebben we een aantal records toegevoegd. De vraag is hoe kunnen we die gegevens raadplegen.

De eerste selectie

Dat kan met behulp van de SQL-instructie select from, onderdeel van DML. Een sterretje (\*) achter select betekent dat je alle velden van een record wilt zien. Daarachter staat de from-component met de naam van de tabel:

select \* from student;

We krijgen dan de gehele inhoud van de tabel student te zien:



In plaats van het sterretje kunnen we de veldnamen noemen van de velden die we willen zien:

select roepnaam, tussenvoegsels, achternaam from student;



We zien dan alleen de velden van alle records die genoemd zijn in de SQL-instuctie dus een selectie van de kolommen uit de tabel.

*Opdracht 3.1*

Geef een overzicht van alle studenten met hun adressen en woonplaatsen.

Bij deze opdracht is het niet duidelijk uit welke elementen de namen van de studenten moeten bestaan: voorletters of roepnaam. Ook is niet duidelijk uit of de postcode wel of niet erbij hoort. In dit geval gaat het om de voorletters en niet de roepnaam en moet de postcode er wel bij staan.

Een conditie

Tot nu toe hebben we het aantal kolommen beperkt, we kunnen ook het aantal records (= regels) beperken door de where-component. Deze komt direct achter de from-component. Er wordt een relatie gelegd tussen een veld en een waarde waaraan het veld wel of niet gelijk moet zijn. Deze relatie wordt weergegeven door een van de volgende relatietekens:

= is gelijk aan

< is kleiner dan

> is groter dan

<= is kleiner dan of gelijk aan

>= is groter dan of gelijk aan

<> is ongelijk aan

Als er gevraagd wordt naar een overzicht van alle studenten die in Almere wonen, dan kun je de volgende query gebruiken:

select \* from student where woonplaats = 'Almere';

We zien nu maar drie records in plaats van zes. Het aantal regels is dus beperkt.

Opdracht 3.2

Geef een overzicht van alle studenten (voorletters, tussenvoegsels en achternaam) die in Lelystad wonen.

Opdracht 3.3

Geef een overzicht van alle studenten (voorletters, tussenvoegsels en achternaam) die niet in Lelystad wonen.

Opdracht 3.4

Geef een overzicht van alle studenten (voorletters, tussenvoegsels en achternaam) die vrouw zijn.

Opdracht 3.5

Geef een overzicht van alle studenten (voorletters, tussenvoegsels en achternaam) die niet man zijn.

Opdracht 3.6

Geef een overzicht van alle studenten (voorletters, tussenvoegsels en achternaam) van wie het geslacht niet bekend is.

Meerdere condities

De laatste opdracht kan op een andere manier opgelost worden.Alleen moeten we dan twee condities aan elkaar koppelen. Dat gebeurt met logische operatoren:

not Geeft het tegengestelde van de conditie

and Aan beide condities moet voldaan zijn

or Aan een van de twee condities moet voldaan zijn

De eerste mogelijkheid is met gebruikmaking van de logische operator and:

select voorletters, tussenvoegsels, achternaam from student where geslacht <> 'm' and geslacht <> 'v';

Als we or gebruiken in plaats van and:

select voorletters, tussenvoegsels, achternaam from student where geslacht <> 'm' or geslacht <> 'v';

dan zien we alle records. Alle records voldoen aan een van de condities: In alle records is het geslacht of ongelijk aan m of ongelijk aan v.

Ook met de not-operator is opdracht 3.5 op te lossen:

select voorletters, tussenvoegsels, achternaam from student where not geslacht = 'm' and not geslacht = 'v';

In deze query worden de logische operatoren and en not gecombineerd.

Een logische operator kan ook meerdere keren in een where-component gebruikt worden. Bijvoorbeeld als je een overzicht wilt geven van studenten die in Almere wonen en die geboren zijn in het jaar 1995:

select voorletters, tussenvoegsels, achternaam from student where woonplaats = 'Almere' and geboortedatum >= '1995-1-1' and geboortedatum < '1996-1-1';

Ook kunnen verschillende operatoren, zoals and en or, gecombineerd worden. Bijvoorbeeld in het overzicht van alle studenten die in 1995 en in 1997 geboren zijn:

select voorletters, tussenvoegsels, achternaam from student where geboortedatum >= '1995-1-1' and geboortedatum < '1996-1-1' or geboortedatum >= '1997-1-1' and geboortedatum < '1998-1-1' ;

Opdracht 3.7

Geef een overzicht van alle studenten (voorletters, tussenvoegsels en achternaam) die in Lelystad wonen en vrouw zijn.

Opdracht 3.8

Geef een overzicht van alle studenten (voorletters, tussenvoegsels en achternaam) die in Almere en Lelystad wonen..

Unieke records

Als er wordt gevraagd naar een overzicht van alle woonplaatsen van alle studenten dan kun je de volgende query gebruiken:

select woonplaats from student;

Almere en Lelystad komen meerdere malen in de lijst voor. Door distinct aan de query toe te voegen, komen alle waarden in het overzicht slechts een keer voor:

select distinct woonplaats from student;

Alle woonplaatsen komen nu een keer in de lijst voor.

Sorteren

Bij sorteren zorg je ervoor dat de records een bepaalde volgorde hebben. Om de gegevens in een bepaalde volgorde te zien zet je achter de query order by en daarachter de veldnaam of veldnamen waarop gesorteerd moet worden. Order by wordt altijd als laatste in de query geplaatst. Je wilt de namen van de studenten in alfabetische volgorde in het overzicht hebben, dan stuur je de volgende query naar de database:

select voorletters, tussenvoegsels, achternaam from student order by achternaam;

Als er studenten zijn met dezelfde achternaam, kun je de query uitbreiden:

select voorletters, tussenvoegsels, achternaam from student order by achternaam, tussenvoegsels, voorletters;

Je kunt er ook nog asc (afkorting van ascending: oplopend) achter de veldnaam plaatsen:

select voorletters, tussenvoegsels, achternaam from student order by achternaam asc, tussenvoegsels asc, voorletters asc;

In dit geval is het antwoord van de database in alle drie de gevallen gelijk. Op de laatste twee queries geeft de database altijd hetzelfde antwoord. Bij desc (afkorting vandescending: aflopend) is het antwoord anders, dan geeft de database de omgekeerde volgorde:

select voorletters, tussenvoegsels, achternaam from student order by achternaam desc, tussenvoegsels desc, voorletters desc;

Opdracht 3.9

Geef een overzicht van alle studenten (voorletters, tussenvoegsels en achternaam en woonplaats) met de woonplaats in alfabetische volgorde.

Opdracht 3.10

Geef een overzicht van alle studenten (voorletters, tussenvoegsels en achternaam en woonplaats) met de woonplaats in omgekeerde volgorde.

Vaktaal

* **Select [distinct] … from ...: where …**: query waarmee gegevens opgevraagd kunnen worden. Achter select staan de veldnamen die getoond moeten worden, achter from staan de tabellen waaruit de gegevens opgehaald moeten worden en achter where staat de conditie of condities..
* **Conditie**: Voorwaarde waaraan records moeten voldoen, voordat ze getoond worden.
* **Relatieteken**: Geeft de relatie aan tussen een veld en een waarde
* **Logische operator**: Geeft aan welke logische relatie er is tussen de verschillende condities in dezelfde where-component
* **Order by**: Sorteren van records in een bepaalde volgorde oplopend of aflopend
* **Empty set**: Reactie van het DBMS als de query wel volgens de taalregels is maar er geen records opgehaald kunnen worden die aan de conditie(s) in de query voldoen