

Overzicht van de taal SQL

```
SELECT <kolommen en/of functies>
FROM <tabellen>
[ WHERE <voorwaarden> ]
[ GROUP BY <groepeer-kolommen> ]
[ HAVING <groep-voorwaarden> ]
[ ORDER BY <sorteer-kolommen> ]
```

De eerste twee regels zijn verplicht. De andere hoeven niet voor te komen, dit wordt aangegeven door de blokhaken.

SELECT

Kolommen	Achter SELECT geef je op welke kolommen de resultaat tabel van de query moet hebben.
Kolomnamen	Je duidt kolommen aan met hun naam. Kolomnamen worden gescheiden door komma's. Wanneer er verwarring mogelijk is doordat er kolommen met dezelfde naam voorkomen (koppelingen), geef je kolommen aan met <i><tabelnaam>.<kolomnaam></i> (punt ertussen).
Alle kolommen	Alle kolommen uit een tabel krijg je met * of <i><tabelnaam>.*</i>
Functies	Je kunt functies over een hele kolom laten uitrekenen: Aantal rijen: <i>COUNT(*)</i> Totaal: <i>SUM(<kolomnaam>)</i> Gemiddelde: <i>AVG(<kolomnaam>)</i> Maximum: <i>MAX(<kolomnaam>)</i> Minimum: <i>MIN(<kolomnaam>)</i> Meestal selecteer je of functies, of kolommen, niet door elkaar. Uitzondering is het groeperen, zie daar.
Andere namen voor kolommen	Je kunt een kolom een nieuwe naam geven door er <i>AS <nieuwenaam></i> achter te zetten. Dit is vooral handig bij kolommen met de resultaten van berekende functies
DISTINCT	Door dit sleutelwoord direct achter SELECT te zetten, zullen er in het resultaat geen dubbele rijen verschijnen.

FROM

Brontabellen	Hier geef je de tabel of tabellen op waaruit de gebruikte gegevens gehaald moeten worden. Tussen de tabelnamen moeten komma's.
Pseudoniemen	Wanneer je direct achter een tabelnaam een afkorting van de tabelnaam zet, kun je die afkorting in de rest van de query gebruiken in plaats van de tabelnaam zelf.
Koppelingen	Wanneer je meer dan één tabel gebruikt, maakt het systeem een <u>koppeling</u> tussen de tabellen. Dit betekent dat alle mogelijke combinaties van rijen gemaakt zullen worden. Meestal is dit niet gewenst, bij WHERE kun je dan door middel van een <u>koppelvoorwaarde</u> aangeven welke combinaties gemaakt moeten worden.

WHERE

Selectievoorwaarden	Achter WHERE geef je met behulp van een voorwaarde aan welke rijen uit de brontabellen geselecteerd moeten worden.
Koppelvoorwaarden	Sommige voorwaarden dienen om aan te geven welke combinaties bij een koppeling gemaakt moeten worden. Meestal bestaat zo'n voorwaarde uit het gelijk stellen van kolommen die in beide tabellen voorkomen (bijvoorbeeld: <i>leerlingen.llnr = uitleningen.llnr</i>).

AND, OR, NOT	Afzonderlijke voorwaarden verbind je met AND of OR. In een reeks van AND en OR wordt de AND het eerst gedaan. Gebruik waar nodig haakjes. Met NOT kun je een voorwaarde omkeren.
Waarden	Zet enkele aanhalingstekens '...' om tekstconstanten of datums. Getallen kun je zonder aanhalingstekens invoeren. NB: Notatie van datums verschilt per systeem.
Operatoren	Gebruik =, <, >, >=, <= of <> om te vergelijken. De < en > duiden bij tekst op alfabetische volgorde: < betekent dan 'eerder in het alfabet'.
LIKE	Gebruik LIKE in plaats van = als je niet exacte gelijkheid bedoelt. In de tekstconstante kun je dan _ (Access: ?) gebruiken voor één onbekende letter, % (Access: *) voor willekeurig veel letters. Voorbeeld: <i>NAAM LIKE '_ob%'</i> selecteert 'Bob' 'Rob' en 'Robert', maar niet 'Jacob'.
IS NULL	Gebruik <i>IS NULL</i> om te testen of er in een veld iets is ingevuld.
<u>Subquery</u> , IN	Binnen een query kun je een subquery maken voor een tussenresultaat. De voorwaarde is dan: <i><kolom> IN (<subquery>)</i> . Hiermee kun je nagaan of de gegevens in <kolom> voorkomen in de tabel die door de subquery wordt opgeleverd. De subquery moet een tabel met één kolom opleveren. Als je zeker weet dat de subquery maar één rij oplevert, bijvoorbeeld doordat er een functie als MAX gebruikt wordt, kun je in plaats van IN ook = gebruiken.
Subquery, NOT IN	Om uit te zoeken of er in een andere tabel juist <i>geen</i> samenhangende gegevens zijn, kun je een subquery met NOT IN gebruiken. De voorwaarde is dan: <i><kolom> NOT IN (<subquery>)</i> .
EXISTS (alleen verdieping)	De voorwaarde: <i>EXISTS (<subquery>)</i> test of de subquery een tabel met één of meer rijen oplevert. <i>NOT EXISTS (<subquery>)</i> test juist of de subquery niets oplevert.

ORDER BY

Sorteervolgorde	Achter ORDER BY kunnen een of meer kolommen worden opgegeven, door middel van de kolomnaam of het rangnummer van de bij SELECT opgegeven kolommen. Er wordt gesorteerd op de eerstgenoemde kolom, bij gelijkheid op de tweede enzovoort.
ASC, DESC	Na elke kolom kun je ASC (oplopend) of DESC (aflopend) opgeven. Standaard is oplopend.

GROUP BY HAVING

Groeperen	Met GROUP BY kun je de rijen van een tabel indelen in groepen die dezelfde gegevens in een kolom (of combinatie van kolommen) hebben. Die kolom(men) geef je achter GROUP BY op.
Functies	Bij SELECT zet je dezelfde kolommen als achter GROUP BY, plus functies voor het aantal, totalen en dergelijke. Deze functies worden dan voor elke groep apart uitgerekend.
HAVING (alleen verdieping)	Bij HAVING kun je voorwaarden opgeven die voor de groepen moeten gelden. Er is een belangrijk verschil met de voorwaarden die je bij WHERE opgeeft. Met de voorwaarden bij WHERE worden rijen geselecteerd, voordat er gegroepeerd wordt. De voorwaarden bij HAVING gelden voor de groepen en de functies die hierin worden gebruikt. Groepen met meer dan tien rijen selecteer je bijvoorbeeld met <i>HAVING COUNT(*) > 10</i> .

Gegevens invoeren/wijzigen/verwijderen

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...)
```

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition
```

```
DELETE FROM table_name  
WHERE condition;
```