# 6 Functies

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

* met de functies count, max, min, sum en avg kunt werken
* kunt groeperen
* met having kunt werken

Een functie herken je door de ronde haakjes die achter de naam van de functie staan. Soms staat er tussen de haakjes ook nog iets. Dat noemen we parameters of argumenten.

*De functie Count*

Met de functie count kun je tellen. De functie count geeft het aantal records dat eventueel aan de voorwaarde voldoet. In zijn eenvoudigste vorm ziet een opdracht om het aantal studenten te tellen er als volgt uit:

select count(\*) from student;

waarbij de wildcart (\*) staat voor alles. Als we alleen de studentes willen tellen, zullen we er een conditie aan moeten toevoegen:

select count(\*) from student where geslacht = 'v';

Dit geldt ook voor de andere functies die we nog gaan bespreken. Als we alle studenten willen tellen met telefoonnummer dan kunnen we dat als volgt oplossen:

select count(telefoon) from student;

De velden waarin een null staat, worden niet meegeteld.

Opdracht 6.1

Hoeveel studenten wonen er in Lelystad?

*De functie Max en Min*

Met de functie max bepaal je de grootste waarde van een veld en met min de kleinste waarde. Bijvoorbeeld de geboortedatum van de jongste student is:

select max(geboortedatum) from student;

Opdracht 6.2

Wat is de naam en de geboortedatum van de oudste student?

*De functie Sum*

Met de functie sum worden getallen bij elkaar opgeteld. Met de volgende query kan worden opgevraagd hoeveel schoolgeld de school had moeten innen.

select sum(schoolgeld) from student;

Opdracht 6.3

Hoeveel schoolgeld heeft de school geïnd?

*De functie Avg*

Met de functie avg bereken je het gemiddelde van velden. Zo kunnen we de gemiddelde berekenen van wat de studenten betaald hebben:

select avg(betaald) from student;

Opdracht 6.4

Hoeveel hebben studenten gemiddeld betaald die 500,- aan schoolgeld moeten betalen?

Daarnaast zijn er de functies ceil() en floor(). De functie ceil() geeft het eerst bovenliggende geheel getal en floor() kapt de cijfers af die achter de komma staan en geeft dus het geheel getal zonder de cijfers achter de komma.

select ceil(2.1) geeft 3.

select floor(2.1) geeft 2.

*Het groeperen*

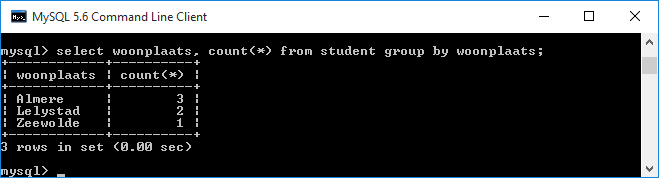
Vaak willen we niet het aantal studenten weten maar bijvoorbeeld hoeveel studenten uit een bepaalde plaats komen. Dat kunnen we doen door de volgende query:

select count(\*) from student where woonplaats = 'Lelystad';

Maar dan weten we alleen het aantal studenten dat in Lelystad woont. Als we dan Lelystad vervangen door Almere, weten we ook hoeveel studenten er in Almere wonen. Als het gaat om honderd verschillende woonplaatsen gaat, dan wordt dat ondoenlijk. We moeten dan gaan groeperen. Dat wil zeggen dat we per verschillende waarde in een veld het aantal kunnen bepalen. Zo’n query ziet er als volgt uit:

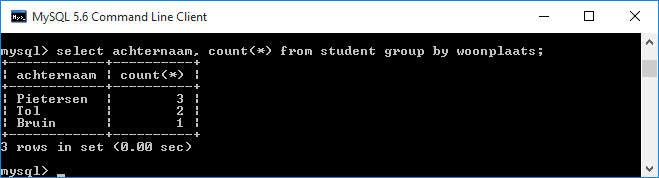
select woonplaats, count(\*) from student group by woonplaats;

We krijgen dan in een keer van alle woonplaatsen in de tabel het aantal studenten te zien:



Let wel op: Achter group by staat dezelfde veldnaam als achter select bij de naam van de functie. Als die niet hetzelfde zijn, krijg je onzin:

select achternaam, count(\*) from student group by woonplaats;



Opdracht 6.5

Laat door een query zien hoeveel mannelijke en vrouwelijke studenten er zijn ingeschreven.

De group by-clause komt in een query altijd na de where-clause. Als we alleen het aantal studenten uit Almere en Lelystad willen weten, kunnen we dat doen met de volgende query:

select woonplaats, count(\*) from student where woonplaats = 'Almere' or woonplaats = 'Lelystad' group by woonplaats;

Opdracht 6.6

Maak een query waardoor we te weten komen hoeveel mannelijke en vrouwelijke studenten uit Lelystad komen.

Met de group by kunnen we ook de verschillende tussenvoegsels bepalen:

select tussenvoegsels from student group by tussenvoegsels;

We maken nu geen gebruik van functies en de database laat alle verschillende voorkomende tussenvoegsels zien.

Opdracht 6.7

Bepaal op een andere manier de verschillende tussenvoegsels.

*De having-opdracht*

De where-clause mag geen functies bevatten. Als we willen weten of er meerdere studenten uit een bepaalde plaats komen dan kunnen we dat niet oplossen met:

select woonplaats, count(\*) from student where count(\*) > 1 group by woonplaats;

We krijgen dan de foutmelding: ERROR 1111 (HY000): Invalid use of group function. Dat komt omdat in de where-clause de controle over de gehele tabel plaats vindt en niet per groep. De functie count(\*) in de where-clause moet per groep controleren of er aan de voorwaarde is voldaan. Daarvoor hebben we de having-opdracht.

select woonplaats, count(\*) from student group by woonplaats having count(\*) > 1;

Dan wordt per groep (per woonplaats dus) gecontroleerd of aan de voorwaarde is voldaan.

Opdracht 6.8

In welke plaats(en) wonen meer dan twee studenten?

Er kan ook op meerdere velden gegroepeerd worden. Als we willen weten hoeveel mannelijke en vrouwelijke studenten er zijn per woonplaats, dan kunnen we de volgende query gebruiken:

select woonplaats, geslacht, count(geslacht) from student group by woonpl

aats, geslacht;

Met functies kan ook gerekend worden. Als we willen weten hoeveel geld de school nog moet innen aan schoolgeld, dan gebruiken we de volgende query:

select woonplaats, geslacht, count(geslacht) from student group by woonplaats, geslacht;

Opdracht 6.9

Hoeveel moet er nog per woonplaats betaald worden?

Opdracht 6.10

Hoeveel moet er nog per woonplaats betaald worden? Zorg ervoor dat alleen die woonplaatsen getoond worden waar nog betaald moet worden.

Opdracht 6.11

Hoeveel moet er nog per woonplaats en per geslacht betaald worden? Zorg ervoor dat alleen die records getoond worden waar nog betaald moet worden.

Vaktaal

* **functie**: functie is een bewerking op alle rijen van een tabel of een groep records. Voorbeelden van functies zijn: count, sum, avg, max en min. Een functie is herkenbaar aan de ronde haken achter de naam van de functie.
* **Parameter**: Een gegeven dat dient als invoer van een functie die daarmee een bewerking kan uitvoeren. Voorbeeld daarvan is: sum(schoolgeld) waarbij sum de naam van de functie is en schoolgeld de parameter. Een parameter staat altijd tussen de ronde haken achter de naam van de functie.
* **Argument**: Zie **parameter**.
* **Groeperen**: rijen of records op basis van een of meerdere overeenkomstige gegevens in een groep plaatsen. In SQL wordt group by gebruikt om records te groeperen.
* **having:** Achter having staat de voorwaarde met daarin een functie waarop gecontroleerd wordt. Dit component is vergelijkbaar met where maar kan alleen in combinatie met group by gebruikt worden. Na where kan geen functie gebruikt worden, na having wel.