* 1. De namen van werknemers waarvan het salaris (nog) niet bekend is maar die wel

een chef hebben.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) naam, Salaris, Chef FROM werknemers

WHERE Salaris [IS](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#operator_is) NULL [OR](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_or) Chef [IS](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#operator_is) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) NULL

* 1. De namen van alle werknemers oplopend geordend naar hun code.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) naam, Code FROM werknemers ORDER BY Code ASC

* 1. Alle woonplaatsen slechts eenmaal genoemd, alfabetisch aflopend geordend.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) naam, Woonplaats FROM werknemers ORDER BY Woonplaats ASC

* 1. Alle namen van alle werknemers, aflopend geordend naar salaris.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) naam, Salaris FROM werknemers ORDER BY Salaris ASC

* 1. Alle namen van alle werknemers, geordend naar datum van indiensttreding.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) naam, Datum\_in\_d FROM werknemers ORDER BY Datum\_in\_d ASC

* 1. Alle namen van alle werknemers met een salaris tussen 2000 en 4000 Euro,

geordend naar woonplaats en per woonplaats naar naam.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) naam, Salaris, Woonplaats FROM werknemers WHERE Salaris BETWEEN 2000 [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) 4000 ORDER BY Woonplaats,Naam ASC

Geavanceerde functies voor het opvragen van gegevens

## GROUP BY, HAVING, COUNT, AVG, SUM, MAX en MIN

In veel gevallen zullen we niet alleen geïnteresseerd zijn in informatie in de vorm van rijen uit een tabel, maar willen we ook inzicht krijgen in gegroepeerde gegevens. In SQL hebben we hiervoor de GROUP BY instructie tot onze beschikking. Vaak maken we daarbij bovendien gebruik van de zogenaamde "groep functies". Hiermee kunnen we bijvoorbeeld tellen hoeveel rijen er in een groep zitten of wat het gemiddelde van een numeriek gegeven in een groep is.

De GROUP BY instructie zorgt er voor dat bepaalde rijen gegroepeerd worden zodat we berekeningen en selecties kunnen uitvoeren met groepen rijen in plaats van individuele rijen. We zullen ons eerst eens af gaan vragen hoe de GROUP BY instructie nu eigenlijk werkt.

Omdat de GROUP BY instructie onmiddellijk na de WHERE component wordt uitgevoerd kunnen we ons het volgende voorstellen.

We zouden van de rijen in de tabel WERKNEMERS een aantal gegroepeerde rijen kunnen maken, bijvoorbeeld per WOONPLAATS. Om dit te bereiken zouden we dus na de WHERE component toe moeten voegen

GROUP BY WOONPLAATS

Het gevolg hiervan is dat er als het ware per woonplaats een aantal rijen tot één rij gecombineerd worden. Dit betekent automatisch dat we met een opdracht als

SELECT NAAM, WOONPLAATS

FROM WERKNEMERS

GROUP BY WOONPLAATS

niets kunnen beginnen. Immers er wordt gegroepeerd, dus eigenlijk ontstaat er iets als het volgende

NAAM WOONPLAATS

BUCK, BRASSER, VLIET, NEVE, PAREE, JANSMA AMSTERDAM UMBGROVE VLISSINGEN

LIEVENSE LEEUWARDEN

MAAS, MERK, WILLEGEN, JOBSE, BOS, MAAS, KONING ROTTERDAM GEEL, LOOF UTRECHT

ZEEUW, VLIET EINDHOVEN

JONGE MAASTRICHT

Het op het scherm zetten van de woonplaatsen is niet zo'n probleem. Maar wat moet er nu afgedrukt worden bij NAAM?

We zullen er dus rekening mee moeten houden dat we bij de GROUP BY optie alleen vragen om die kolommen waar op gegroepeerd wordt of gebruik maken van de zogenaamde groep functies. De groep functies die we in iedere SQL implementatie aan kunnen treffen zijn:

COUNT

AVG

SUM

MAX

MIN

We zouden ons bijvoorbeeld af kunnen vragen hoeveel werknemers er in iedere plaats wonen:

SELECT WOONPLAATS, COUNT(\*)

FROM WERKNEMERS

GROUP BY WOONPLAATS

Dit heeft tot gevolg dat er bij het tussenresultaat gekeken zal worden hoeveel rijen er in iedere groep zitten (we geven door het \* teken aan dat het aantal rijen geteld moet worden) en dat de uitkomst op het scherm afgedrukt zal worden. Dat geeft dan het volgende resultaat

WOONPLAATS COUNT(\*)

AMSTERDAM 6

VLISSINGEN 1

LEEUWARDEN 1

ROTTERDAM 7

UTRECHT 2

EINDHOVEN 2

MAASTRI CHT 1

We kunnen COUNT ook gebruiken om alleen het aantal ingevulde waarden per kolom te tellen. We gebruiken dan de kolom naam in plaats van het \* teken.

SELECT WOONPLAATS, COUNT(CHEF)

FROM WERKNEMERS

GROUP BY WOONPLAATS

geeft ons per woonplaats het aantal werknemers met een CHEF. Met andere woorden: ROTTERDAM zal dan door het getal 6 in plaats van 7 worden vergezeld omdat de in ROTTERDAM woonachtige DE KONING geen CHEF heeft en in dit geval dus niet meetelt.

De functie AVG wordt gebruikt om het gemiddelde (AVERAGE) van een bepaalde groep te berekenen. Zo zouden we het gemiddelde salaris per woonplaats op kunnen vragen door

SELECT WOONPLAATS, AVG(SALARIS)

FROM WERKNEMERS

GROUP BY WOONPLAATS

AVG berekent het gemiddelde en houdt daarbij rekening met het feit dat niet ingevulde kolommen (NULL waarden) genegeerd moeten worden. Ook is het in sommige RDBMS-en mogelijk om DISTINCT te gebruiken zodat het gemiddelde van alle verschillende waarden berekend wordt. AVG(DISTINCT(GETAL)) bij de waarden 1, 1, 1 en 3 levert dan 2 op in plaats van 1,5.

De SUM functie totaliseert een numerieke kolom:

SELECT WOONPLAATS, SUM(SALARIS)

FROM WERKNEMERS

GROUP BY WOONPLAATS

levert per WOONPLAATS het totaal aan salarissen op.

De MAX en de MIN functie berekenen respectievelijk de hoogste en de laagste waarde binnen een groep.

SELECT WOONPLAATS, MAX(SALARIS)

FROM WERKNEMERS

GROUP BY WOONPLAATS

geeft ons per woonplaats het hoogste salaris, terwijl

SELECT WOONPLAATS, MIN(SALARIS)

FROM WERKNEMERS

GROUP BY WOONPLAATS

ons het laagste salaris per woonplaats oplevert.

Het is ook toegestaan om te groeperen op meerdere kolommen. Als we

bijvoorbeeld een overzicht willen hebben van het aantal werknemers per woonplaats en binnen woonplaats per postcode-gebied kunnen we opvragen

SELECT WOONPLAATS, POSTCODE, COUNT(\*)

FROM WERKNEMERS

GROUP BY WOONPLAATS, POSTCODE

De GROUP FUNCTIES kunnen ook gebruikt worden zonder het GROUP BY

gedeelte. Dat komt met name van pas als we zo'n functie willen gebruiken voor een hele tabel. We beschouwen dan in feite de gehele tabel als een groep.

De opdracht

SELECT COUNT(\* ), AVG(SALARIS), SUM(SALARIS), MAX(SALARIS), MIN (SALARIS)

FROM WERKNEMERS

geeft ons een overzicht van achtereenvolgens het aantal rijen, het gemiddelde salaris, het totaalbedrag van alle salarissen, het hoogste salaris en het laagste salaris van alle werknemers in de WERKNEMERS tabel.

Aan gegroepeerde rijen kunnen we ook typische 'groepsvoorwaarden' stellen. Zo zouden we bijvoorbeeld het aantal werknemers per woonplaats op kunnen vragen maar daar tegelijkertijd aan toe kunnen voegen dat we in het overzicht alleen die woonplaatsen willen zien die meer dan 5 werknemers hebben.

Dan moeten we gebruik maken van HAVING.

SELECT WOONPLAATS, COUNT(\*)

FROM WERKNEMERS

GROUP BY WOONPLAATS HAVING COUNT(\*) > 5

Dat geeft dan het volgende resultaat:

WOONPLAATS COUNT(\*)

AMSTERDAM 6

ROTTERDAM 7

Het RDBMS zal nog steeds de eerste stappen uit de eerder genoemde opdracht zonder HAVING uitvoeren, maar zal voordat het resultaat op het scherm wordt afgedrukt als het ware nog die groepen uitfilteren die niet aan de, in het HAVING gedeelte gestelde, voorwaarde voldoen.

Uiteraard is het toegestaan om in een SELECT opdracht met een GROUP BY voorwaarden te stellen met behulp van een WHERE gedeelte. We moeten ons dan goed realiseren dat het WHERE gedeelte eerder wordt uitgevoerd dan het GROUP BY gedeelte, zodat er groepen gemaakt worden van alleen die rijen die aan de, in het WHERE gedeelte gestelde, voorwaarde voldoen.

SELECT WOONPLAATS, AVG(SALARIS)

FROM WERKNEMERS

WHERE SALARIS > 2500

GROUP BY WOONPLAATS

geeft ons per woonplaats het gemiddelde salaris van iedereen die meer dan 2500 Euro verdient.

Apart aandacht verdient nog de waarde NULL oftewel de niet ingevulde waarde. Bedenk dat er dus ook een groep kan ontstaan van records waarbij in de kolom die we in het GROUP BY gedeelte gebruiken, niets is ingevuld.

**Opdrachten**

Geef een opdracht in SQL voor het opvragen van de volgende overzichten

WERKNEMERS tabel.

* 1. Per woonplaats het gemiddelde salaris.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Woonplaats, [AVG](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_avg)(Salaris) FROM werknemers WHERE Salaris GROUP BY Woonplaats

* 1. Het hoogste salaris dat per chef door een ondergeschikte verdiend wordt.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Naam, Chef, [MAX](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_max)(Salaris) FROM werknemers WHERE Salaris GROUP BY chef

* 1. Idem als vraag 2 maar nu tellen alleen de werknemers die minder dan 4000 Euro verdienen mee.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Naam, Salaris, chef FROM werknemers WHERE Salaris <4000 GROUP BY chef

* 1. Het aantal werknemers dat onder iedere chef valt, chefs met minder dan twee ondergeschikten moeten niet vermeld worden.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Chef, [count](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_count)(\*) FROM werknemers GROUP by Chef HAVING [COUNT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_count)(\*) <2

Voor het raadplegen van gegevens uit meerdere tabellen hebben we de beschikking over een aantal mogelijkheden. We kennen de SUBQUERY: het gebruiken van meerdere SELECT opdrachten in één opdracht.

Naast de SUBQUERY kennen we de al eerder besproken JOIN waarbij we meerdere tabellen in het FROM gedeelte van de SELECT opdracht gebruiken.

In dit hoofdstuk zullen we zien dat we in veel gevallen zowel een SUBQUERY als een JOIN kunnen gebruiken om een antwoord te vinden op een bepaalde vraag.

De SUBQUERY

subquery De SUBQUERY is eigenlijk niets anders dan een vraag in een vraag oftewel een subvraag om bepaalde problemen op te lossen. Er is vrijwel geen enkel probleem te bedenken dat we niet ook met de JOIN op zouden kunnen lossen maar in veel gevallen is de SUBQUERY een stuk helderder en bovendien eenvoudiger te formuleren. Wel dienen we er rekening mee te houden dat veel RDBMS systemen beter presteren als we gebruik maken van een JOIN in plaats van een SUBQUERY.

Stel dat we op zoek zijn naar de namen van alle werknemers die in dezelfde plaats wonen als werknemer BOS. Natuurlijk kunnen we het antwoord hierop in twee stappen verkrijgen:

SELECT WOONPLAATS

FROM WERKNEMERS

WHERE NAAM = 'BOS'

Met als resultaat

WOONPLAATS

ROTTERDAM

en vervolgens

SELECT NAAM

FROM WERKNEMERS

WHERE WOONPLAATS = 'ROTTERDAM'

Met een SUBQUERY is het eigenlijk veel eenvoudiger op te lossen:

SELECT NAAM

FROM WERKNEMERS

WHERE WOONPLAATS =

(SELECT WOONPLAATS

FROM WERKNEMERS

WHERE NAAM = 'BOS')

Als we deze oplossing zien, zien we al vrij snel wat de bedoeling is, namelijk 'Geef de namen van alle werknemers met als woonplaats (de woonplaats van BOS)'. Dit is niet alleen wat sneller te doorzien maar sluit ook meer aan bij de wijze waarop we de vraag geformuleerd hadden.

Door gebruik te maken van een subvraag kunnen we eigenlijk op vrij eenvoudige wijze vraagstellingen zoals in het voorgaande oplossen.

Waar we in het voorafgaande aan voorbij zijn gegaan, is het feit dat een SUBQUERY in veel gevallen niet één, maar meerdere waarden oplevert. In het voorgaande voorbeeld kwam dit toevallig goed uit omdat in onze database slechts één werknemer BOS aanwezig is. Dat is bijvoorbeeld zeker niet het geval als we ons afvragen welke werknemers in dezelfde plaats wonen als de werknemers die als chef werknemer 40 hebben. Als we dit proberen op te lossen met:

SELECT naam

FROM werknemers

WHERE woonplaats =

(**SELECT woonplaats**

**FROM werknemers**

**WHERE chef= '40'**)

zal dit tot een foutmelding leiden omdat de SUBQUERY hier meerdere woonplaatsen oplevert en geen enkele waarde uit een rij kan gelijk zijn aan de waarden uit meerdere andere rijen. We gebruiken dan de (al eerder behandelde) IN operator.

Dus

SELECT naam

FROM werknemers

WHERE woonplaats IN

(**SELECT woonplaats**

**FROM werknemers**

**WHERE chef = '40')**

Het is ook mogelijk om meerdere SUBQUERIES aan elkaar te koppelen. De

voorgaande vraag zouden we ook kunnen formuleren als: Welke werknemers wonen in dezelfde plaats als de werknemers die als chef MERK hebben? Dat lossen we op door:

SELECT naam

FROM werknemers

WHERE woonplaats IN

(SELECT woonplaats

FROM werknemers

WHERE chef IN

(SELECT code

FROM werknmers

WHERE naam = 'MERK'))

**Opdrachten**

Geef een opdracht in SQL voor het opvragen van de volgende overzichten uit de WERKNEMERS, WERK, PROJECTEN en SCHALEN tabellen.

* 1. De namen van werknemers die dezelfde chef hebben als werknemer 40.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) naam, Code, Chef FROM werknemers WHERE Code [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Code FROM werknemers WHERE Code [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) code FROM werknemers WHERE Chef = 99))

* 1. De namen van werknemers die aan hetzelfde project werken als werknemer 35.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) code FROM werknemers WHERE werknemers.Code [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) code FROM werk WHERE proj\_code= ( [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) proj\_code FROM werk WHERE code = 35))

* 1. De namen van projecten die uitgevoerd worden in dezelfde plaats als de woonplaats van werknemer BOS.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Naam, Woonplaats FROM werknemers WHERE Woonplaats [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Woonplaats FROM werknemers WHERE Naam ="BOS")

* 1. De namen van werknemers die in een andere plaats wonen dan werknemer 99.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Code, Naam, Woonplaats FROM werknemers WHERE [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) Woonplaats [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Woonplaats FROM werknemers WHERE Code =99)

* 1. De namen van werknemers die een hoger salaris hebben dan werknemer 12.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Code, Naam, Salaris FROM werknemers WHERE Salaris [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in)([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Salaris FROM werknemers WHERE Salaris > 2149)

* 1. De namen van projecten waar werknemer MERK aan werkt.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) proj\_naam, proj\_leider FROM projecten WHERE proj\_leider [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) werknemers.Code FROM werknemers WHERE Naam ="MERK")

* 1. De woonplaatsen van werknemers die aan hetzelfde project werken als werknemer MERK.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Woonplaats, naam FROM werknemers WHERE woonplaats [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) projecten.proj\_plaats FROM projecten WHERE projecten.proj\_plaats = ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Woonplaats FROM werknemers WHERE naam = "MERK"))

* 1. De namen van alle werknemers die in dezelfde salarisschaal vallen als werknemer BOS.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Naam,schaal FROM werknemers, schalen WHERE werknemers.Salaris BETWEEN schalen.laagste [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) schalen.hoogste [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) schaal [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ( [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) schaal FROM werknemers,schalen WHERE werknemers.Salaris BETWEENschalen.laagste [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) schalen.hoogste [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) werknemers.Naam="BOS" )