



# SQL

## ***Webleren***

*School je gratis bij via het internet. Waar en wanneer je wilt.*

[www.vdab.be/webleren](http://www.vdab.be/webleren)

21/01/2017

## © COPYRIGHT 2016 VDAB

Niets uit deze syllabus mag worden verveelvoudigd, bewerkt, opgeslagen in een database en/of openbaar gemaakt door middel van druk, fotokopie, microfilm, geluidsband, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van VDAB.

Hoewel deze syllabus met zeer veel zorg is samengesteld, aanvaardt VDAB geen enkele aansprakelijkheid voor schade ontstaan door eventuele fouten en/of onvolkomenheden in deze syllabus en of bijhorende bestanden.

## Inhoud

Hoofdstuk 1.	Inleiding .....	10
1.1.	Wat is SQL? .....	10
1.1.1.	Introductie .....	10
1.1.2.	Categorieën .....	10
1.2.	Enkele begrippen .....	12
1.2.1.	Database, -server en -taal .....	12
1.2.2.	Het relationeel model .....	13
1.3.	Standaardisatie .....	17
1.3.1.	Verschillende dialecten .....	17
1.3.2.	Ontstaan .....	17
1.4.	Opgaven .....	18
Hoofdstuk 2.	Databases .....	19
2.1.	Intro .....	19
2.2.	Bieren .....	19
2.3.	Planten .....	20
2.4.	Video .....	22
Hoofdstuk 3.	Software .....	24
3.1.	Wat heb je nodig .....	24
3.1.1.	Workbench .....	24
3.1.2.	Installatie .....	24
3.2.	Workbench .....	24
3.2.1.	Database importeren .....	24
3.2.2.	Query uitvoeren .....	24
3.2.3.	Schema tonen .....	24
Hoofdstuk 4.	Gegevens selecteren .....	25
4.1.	Selecteren .....	25
4.1.1.	Syntax .....	25
4.1.2.	Voorbeelden .....	25
4.1.3.	Opdracht .....	28
4.2.	Where-voorwaarde .....	28
4.2.1.	Syntax .....	28
4.2.2.	Like .....	28

4.2.3.	Voorbeelden .....	29
4.2.4.	And, Or en Not.....	32
4.2.5.	Voorbeelden .....	32
4.2.6.	Opdracht.....	34
4.3.	Sorteren.....	34
4.3.1.	Syntax .....	34
4.3.2.	Voorbeelden .....	34
4.3.3.	Opdracht.....	36
4.4.	Opgaven 1.....	36
4.4.1.	Opgave 1.....	36
4.4.2.	Opgave 2.....	37
4.4.3.	Opgave 3.....	37
4.4.4.	Opgave 4.....	37
4.4.5.	Opgave 5.....	38
4.4.6.	Opgave 6.....	38
4.4.7.	Opgave 7.....	38
4.4.8.	Opgave 8.....	39
4.4.9.	Opgave 9.....	39
4.4.10.	Opgave 10.....	39
4.4.11.	Opgave 11.....	39
4.4.12.	Opgave 12.....	40
4.4.13.	Opgave 13.....	40
4.4.14.	Opgave 14.....	41
4.4.15.	Opgave 15.....	41
4.4.16.	Opgave 16.....	41
4.4.17.	Opgave 17.....	42
4.4.18.	Opgave 18.....	42
4.4.19.	Opgave 19.....	42
4.4.20.	Opgave 20.....	42
4.4.21.	Opgave 21.....	43
4.4.22.	Opgave 22.....	43
4.4.23.	Opgave 23.....	44
4.4.24.	Opgave 24.....	44
4.4.25.	Opgave 25.....	44

4.4.26.	Opgave 26.....	44
4.4.27.	Opgave 27.....	45
4.5.	Aggregate functions .....	45
4.5.1.	Theorie.....	45
4.5.2.	Voorbeelden .....	46
4.5.3.	Opdracht.....	47
4.6.	Berekeningen.....	47
4.6.1.	Theorie.....	47
4.6.2.	Voorbeelden .....	48
4.7.	Opgaven.....	48
4.7.1.	Opgave 1.....	49
4.7.2.	Opgave 2.....	49
4.7.3.	Opgave 3.....	49
4.7.4.	Opgave 4.....	49
4.7.5.	Opgave 5.....	49
4.7.6.	Opgave 6.....	49
4.7.7.	Opgave 7.....	50
4.7.8.	Opgave 8.....	50
4.7.9.	Opgave 9.....	50
4.8.	Groeperen .....	51
4.8.1.	Syntax .....	51
4.8.2.	Regel .....	51
4.8.3.	Voorbeelden .....	51
4.8.4.	Opdracht.....	54
4.9.	Opgaven 3.....	54
4.9.1.	Opgave 1.....	54
4.9.2.	Opgave 2.....	54
4.9.3.	Opgave 3.....	55
4.9.4.	Opgave 4.....	55
4.9.5.	Opgave 5.....	56
4.9.6.	Opgave 6.....	56
4.9.7.	Opgave 7.....	56
4.9.8.	Opgave 8.....	57
4.9.9.	Opgave 9.....	57

4.9.10.	Opgave 10.....	57
4.9.11.	Opgave 11.....	57
4.9.12.	Opgave 12.....	58
4.9.13.	Opgave 13.....	58
4.9.14.	Opgave 14.....	58
4.10.	Opdracht voor de coach 1 .....	59
Hoofdstuk 5.	Meerdere tables .....	60
5.1.	Inner join .....	60
5.1.1.	Syntax .....	60
5.1.2.	Voorbeelden .....	61
5.2.	Outer join.....	63
5.2.1.	Syntax .....	63
5.2.2.	Left versus Right .....	64
5.2.3.	Voorbeelden .....	64
5.2.4.	Combinaties.....	66
5.3.	Self join .....	67
5.3.1.	Syntax .....	67
5.3.2.	Voorbeeld .....	67
5.3.3.	Opgaven.....	68
5.3.4.	Opgave 1.....	68
5.3.5.	Opgave 2.....	69
5.3.6.	Opgave 3.....	69
5.3.7.	Opgave 4.....	70
5.3.8.	Opgave 5.....	70
5.3.9.	Opgave 6.....	71
5.3.10.	Opgave 7.....	71
5.3.11.	Opgave 8.....	71
5.3.12.	Opgave 9.....	72
5.3.13.	Opgave 10.....	72
Hoofdstuk 6.	Extra mogelijkheden.....	73
6.1.	Union .....	73
6.1.1.	Syntax .....	73
6.1.2.	Voorbeelden .....	73
6.2.	Subqueries Intro .....	76

6.2.1.	Voorbeelden .....	76
6.2.2.	Wanneer gebruik je een subquery en wanneer join? .....	80
6.3.	Opgaven.....	81
6.3.1.	Opgave 1.....	81
6.3.2.	Opgave 2.....	82
6.3.3.	Opgave 3.....	82
6.3.4.	Opgave 4.....	83
6.3.5.	Opgave 5.....	83
6.3.6.	Opgave 6.....	84
6.3.7.	Opgave 7.....	84
6.3.8.	Opgave 8.....	84
6.3.9.	Opgave 9.....	84
6.3.10.	Opgave 10.....	85
6.3.11.	Opgave 11.....	85
6.4.	Opdracht voor de coach 2 .....	86
Hoofdstuk 7.	Gegevens aanpassen .....	87
7.1.	Inleiding .....	87
7.2.	Insert.....	87
7.2.1.	Zelf waarden opgeven .....	87
7.2.2.	Uit andere tables .....	88
7.2.3.	Voorbeelden .....	88
7.3.	Update.....	91
7.3.1.	Syntax .....	91
7.3.2.	Voorbeelden .....	92
7.4.	Delete .....	94
7.4.1.	Syntax .....	94
7.4.2.	Voorbeelden .....	94
7.4.3.	Opgaven.....	96
7.4.4.	Opgave 1.....	96
7.4.5.	Opgave 2.....	96
7.4.6.	Opgave 3.....	96
7.4.7.	Opgave 4.....	96
7.4.8.	Opgave 5.....	97
7.4.9.	Opgave 6.....	97

Hoofdstuk 8.	Beheer van tables en relaties .....	98
8.1.	Inleiding .....	98
8.2.	Table maken .....	98
8.2.1.	Een lege table maken .....	98
8.2.2.	Datatypes.....	99
8.2.3.	Gegevenstype float versus decimal.....	100
8.2.4.	Een nieuwe table op basis van geselecteerde records .....	101
8.2.5.	Voorbeelden .....	101
8.3.	Table verwijderen.....	102
8.3.1.	Syntax .....	103
8.3.2.	Voorbeelden .....	103
8.4.	Table wijzigen .....	103
8.4.1.	Syntax .....	104
8.4.2.	Voorbeelden .....	104
8.5.	Sleutels .....	105
8.5.1.	Constraint .....	105
8.5.2.	Beperkingen.....	105
8.5.3.	Syntax .....	107
8.5.4.	Voorbeelden .....	107
8.5.5.	Indexen .....	110
8.5.6.	Algemeen.....	110
8.5.7.	Waarom? - Voorbeeld .....	111
8.5.8.	Syntax .....	111
8.5.9.	Effect van een index .....	111
8.5.10.	Voorbeelden .....	112
8.6.	Opgaven.....	112
8.6.1.	Opgave 1.....	112
8.6.2.	Opgave 2.....	113
8.6.3.	Opgave 3.....	113
8.6.4.	Opgave 4.....	113
8.6.5.	Opgave 5.....	114
8.6.6.	Opgave 6.....	114
Hoofdstuk 9.	Views .....	115
9.1.	Wat zijn views?.....	115



9.1.1.	Definitie .....	115
9.1.2.	Toepassing .....	115
9.2.	Maken en verwijderen .....	115
9.2.1.	Maken .....	116
9.2.2.	Verwijderen .....	116
9.2.3.	Voorbeelden .....	116
9.3.	Opgaven .....	117
9.3.1.	Opgave 1 .....	117
9.3.2.	Opgave 2 .....	117
9.3.3.	Opgave 3 .....	118
9.3.4.	Opgave 4 .....	118
9.3.5.	Opgave 5 .....	118
9.3.6.	Opgave 6 .....	118
9.3.7.	Opgave 7 .....	118
9.4.	Opdracht voor de coach 3 .....	118
Hoofdstuk 10.	Einde cursus .....	120
10.1.1.	Eindoefening .....	120
10.1.2.	Wat nu? .....	120

## Hoofdstuk 1. Inleiding

### 1.1. Wat is SQL?

#### 1.1.1. Introductie

Sinds het begin van de jaren tachtig is de van oorsprong universiteitstaal **Structured Query Language** (SQL) niet weer weg te denken als kern van de diverse aansturingstalen.

SQL is een **database opvraag taal**. Je kan gegevens uit een (relationele) database ophalen en bewerken. Het is geen programmeertaal, maar je kan wel functies aanroepen en verschillende scenario's maken.

De databasegegevens worden beheerd door een afzonderlijk systeem: Relational Database Management System (zoals Access). Het RDBMS zelf is verantwoordelijk voor de structuur, het bewaren en ophalen van gegevens. Met behulp van SQL kan je opdrachten geven aan het RDBMS.

SQL is een **gestandaardiseerde taal** en dus bruikbaar voor elke soort relationele database. Het kan de basis vormen van elke toekomstige database-koppeling die je als programmeur, analist of database-gebruiker gaat maken.



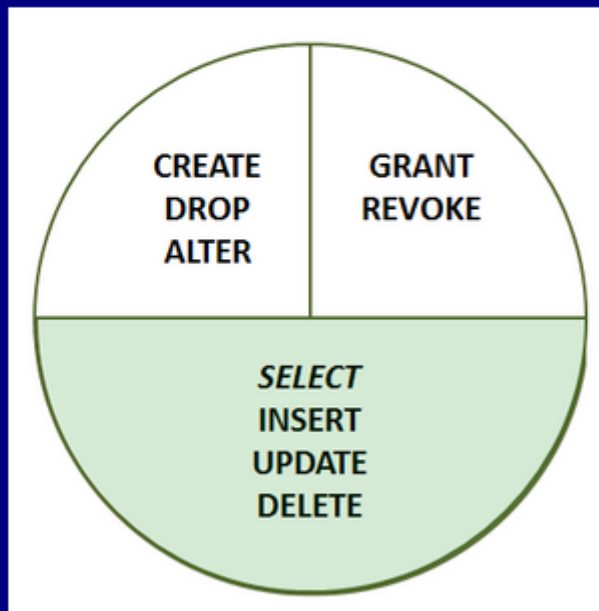
Bekijk zeker ook de [animatie](#) op YouTube.

#### 1.1.2. Categorieën

De SQL-instructies worden in 3 categorieën opgesplitst:

# DML

## Data Manipulation Language

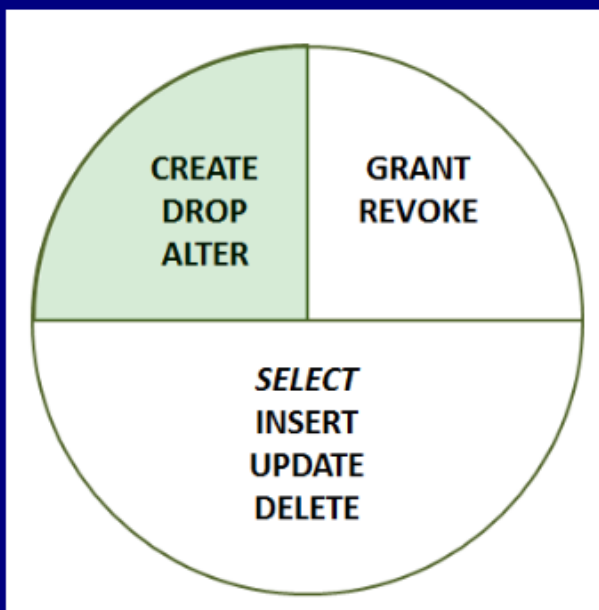


Gegevens ...

- selecteren
- toevoegen
- wijzigen
- wissen

# DDL

## Data Definition Language

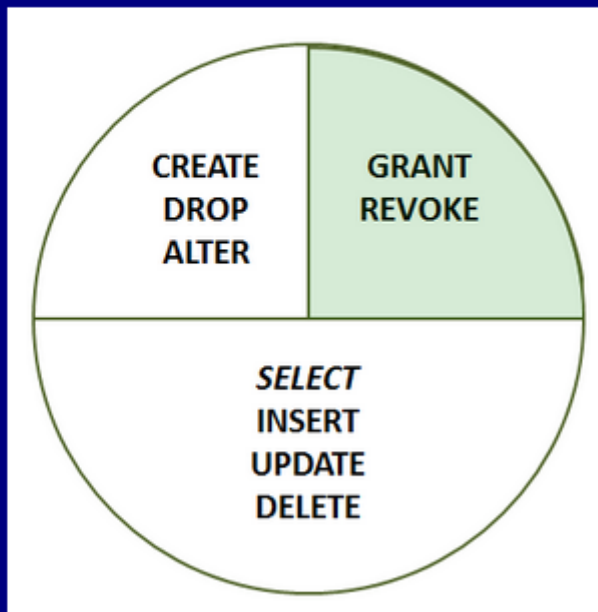


Creatie en  
onderhoud van

- databases
- tables
- views
- indexen

# DCL

## Data Control Language



Gaat over beveiliging van gegevens.  
Voorbeeld:

- personeelsdirecteur: toegang tot lonen van personeelsleden
- secretaresse: enkel toegang tot namen en personeelsnummer

Deze instructies komen niet aan bod in deze cursus.

## 1.2. Enkele begrippen

Voor we aan het leren van SQL kunnen beginnen, moeten we eerst een aantal begrippen uitleggen.

### 1.2.1. Database, -server en -taal

We keren terug naar de definitie van SQL:

**SQL** is een databasetaal waarin opdrachten geformuleerd kunnen worden die door een databaseserver verwerkt worden.

Deze zin bevat drie belangrijke begrippen: **database**, **databaseserver** en **databasetaal**. We lichten elk van de begrippen kort toe.

#### *Database*

Dit zou een definitie kunnen zijn:

Een **database** is een gecentraliseerde, gestructureerde set gegevens die op computer bewaard wordt.

Databases bestaan tegenwoordig in alle maten en soorten, gaande van kleine lokale databases voor persoonlijk gebruik tot supergrote server databases waarop wel duizenden mensen tegelijkertijd data kunnen raadplegen. Een kaartenbak wordt dus niet gezien als een database. De grote geautomatiseerde bestanden van banken, verzekeringsmaatschappijen, organisaties,... voldoen wel aan deze definitie.

### *Databaseserver*

Gegevens in een database worden pas waardevol wanneer er iets mee gedaan wordt. Volgens de definitie worden gegevens in de database beheerd door een apart managementsysteem. Dit systeem wordt een databaseserver genoemd.

Zo kan een databaseserver

- het aanmelden afhandelen,
- query's afhandelen,
- wijzigingen registreren,
- back-ups aanmaken,
- ...

Zonder databaseserver is het niet mogelijk om de gegevens te raadplegen, te wijzigen of te verwijderen. Alleen de databaseserver weet waar welke gegevens bewaard worden.

### *Databasetaal*

De databaseserver zal nooit uit zichzelf gegevens wijzigen of verwijderen. Iemand moet daartoe opdracht geven. Die opdrachten worden in speciale talen geformuleerd: de databasetalen.

Opdrachten, ook wel instructies genoemd, - volgens de regels van de databasetaal geformuleerd - worden door de gebruiker met behulp van speciale programma's ingevoerd en door de databaseserver uitgevoerd.

Elke databaseserver bezit een databasetaal. Tussen al die talen bestaan verschillen waardoor ze in grote groepen te verdelen zijn. Eén van de groepen is die van de relationele databasetalen. SQL behoort tot deze groep.

#### **1.2.2. Het relationeel model**

SQL is zoals gezegd een relationele databasetaal. Ze is gebaseerd op het relationele model. Meer uitleg hierover vind je in de webcursus **Gegevensanalyse**.

De belangrijkste begrippen willen we toch nog even op een rij zetten:

# Table

## Entiteit

Tweedimensionale gegevensstructuur bestaande uit rijen en kolommen.

PersoneelsID	Familienaam	Voornaam	Straat en nummer	Postcode	gsm	Geboortedatum
1054	Vandenabeele	Jos	Dapperstraat 16	8490	0497 66 35 13	27/01/1966
1055	Vanacker	Veerle	Land van Belofte 26	8000	0497 36 36 86	02/07/1969
1056	Verlinde	Marc	H. Serruyslaan 66	8400	0497 49 71 96	08/12/1975
1057	Liever☺	Ward	Noordstraat 23	8320	0497 66 73 37	29/05/1980
1058	Dedeurwaerder	Dirk	Zevende Hemel 7	8730	0497 09 72 20	20/02/1983
2012	Quaghebuur	Mieke	Langestraat 98	9000	0496 78 45 12	03/08/1969
2013	Dedecker	Eric	Zoeten Inval 144	9000	0478 74 85 96	29/01/1970
2014	Lieckens	Roger	St-Hubertuslaan 41	9000	0479 50 21 12	18/12/1975
2015	Delporte	Saartje	Vijfwegenstraat 164	9000		09/05/1960
2016	Lievemans	Chris	Trapiestendreef 69	9000	0476 45 67 89	27/01/1966
2017	Lindemans	Klaas	Rue Les Misérables 2	9000		19/02/1975
3088	Filliers	Dirk	De Keyserlei	2000		08/01/1976
3089	Moens	Johan	Statiestraat 7	2950		06/09/1979
3090	Vanbergen	Jacqueline	Turnhoutsebaan 16	2140	0496 49 93 57	29/12/1975
3091	Vanmarcke	Inge	Turnhoutsebaan 194	2110	0478 44 38 89	30/12/1975

# Rij

## Instantie

Bevat gegevens die bij elkaar horen.

PersoneelsID	Familienaam	Voornaam	Straat en nummer	Postcode	gsm	Geboortedatum
1054	Vandenabeele	Jos	Dapperstraat 16	8490	0497 66 35 13	27/01/1966
1055	Vanacker	Veerle	Land van Belofte 26	8000	0497 36 36 86	02/07/1969
1056	Verlinde	Marc	H. Serruyslaan 66	8400	0497 49 71 96	08/12/1975
1057	Liever☺	Ward	Noordstraat 23	8320	0497 66 73 37	29/05/1980
1058	Dedeurwaerder	Dirk	Zevende Hemel 7	8730	0497 09 72 20	20/02/1983
2012	Quaghebuur	Mieke	Langestraat 98	9000	0496 78 45 12	03/08/1969
2013	Dedecker	Eric	Zoeten Inval 144	9000	0478 74 85 96	29/01/1970
2014	Lieckens	Roger	St-Hubertuslaan 41	9000	0479 50 21 12	18/12/1975
2015	Delporte	Saartje	Vijfwegenstraat 164	9000		09/05/1960
2016	Lievemans	Chris	Trapiestendreef 69	9000	0476 45 67 89	27/01/1966
2017	Lindemans	Klaas	Rue Les Misérables 2	9000		19/02/1975
3088	Filliers	Dirk	De Keyserlei	2000		08/01/1976
3089	Moens	Johan	Statiestraat 7	2950		06/09/1979
3090	Vanbergen	Jacqueline	Turnhoutsebaan 16	2140	0496 49 93 57	29/12/1975
3091	Vanmarcke	Inge	Turnhoutsebaan 194	2110	0478 44 38 89	30/12/1975

# Kolom

## Attribuut

Bevat gegevens van dezelfde soort.

PersoneelsID	Familienaam	Voornaam	Straat en nummer	Postcode	gsm	Geboortedatum
1054	Vandenabeele	Jos	Dapperstraat 16	8490	0497 66 35 13	27/01/1966
1055	Vanacker	Veerle	Land van Belofte 26	8000	0497 36 36 86	02/07/1969
1056	Verlinde	Marc	H. Serruyslaan 66	8400	0497 49 71 96	08/12/1975
1057	Liever	Ward	Noordstraat 23	8320	0497 66 73 37	29/05/1980
1058	Dedeurwaerder	Dirk	Zevende Hemel 7	8730	0497 09 72 20	20/02/1983
2012	Quaghebuur	Mieke	Langestraat 98	9000	0496 78 45 12	03/08/1969
2013	Dedecker	Eric	Zoeten Inval 144	9000	0478 74 85 96	29/01/1970
2014	Lieckens	Roger	St-Hubertuslaan 41	9000	0479 50 21 12	18/12/1975
2015	Delporte	Saartje	Vijfwegenstraat 164	9000		09/05/1960
2016	Lievemans	Chris	Trapiestendreef 69	9000	0476 45 67 89	27/01/1966
2017	Lindemans	Klaas	Rue Les Misérables 2	9000		19/02/1975
3088	Filliers	Dirk	De Keyserlei	2000		08/01/1976
3089	Moens	Johan	Statiestraat 7	2950		06/09/1979
3090	Vanbergen	Jacqueline	Turnhoutsebaan 16	2140	0496 49 93 57	29/12/1975
3091	Vanmarcke	Inge	Turnhoutsebaan 194	2110	0478 44 38 89	30/12/1975

# Veld

## Kruispunt van een rij en een kolom

Bevat specifieke informatie van een specifiek iets.

PersoneelsID	Familienaam	Voornaam	Straat en nummer	Postcode	gsm	Geboortedatum
1054	Vandenabeele	Jos	Dapperstraat 16	8490	0497 66 35 13	27/01/1966
1055	Vanacker	Veerle	Land van Belofte 26	8000	0497 36 36 86	02/07/1969
1056	Verlinde	Marc	H. Serruyslaan 66	8400	0497 49 71 96	08/12/1975
1057	Liever	Ward	Noordstraat 23	8320	0497 66 73 37	29/05/1980
1058	Dedeurwaerder	Dirk	Zevende Hemel 7	8730	0497 09 72 20	20/02/1983
2012	Quaghebuur	Mieke	Langestraat 98	9000	0496 78 45 12	03/08/1969
2013	Dedecker	Eric	Zoeten Inval 144	9000	0478 74 85 96	29/01/1970
2014	Lieckens	Roger	St-Hubertuslaan 41	9000	0479 50 21 12	18/12/1975
2015	Delporte	Saartje	Vijfwegenstraat 164	9000		09/05/1960
2016	Lievemans	Chris	Trapiestendreef 69	9000	0476 45 67 89	27/01/1966
2017	Lindemans	Klaas	Rue Les Misérables 2	9000		19/02/1975
3088	Filliers	Dirk	De Keyserlei	2000		08/01/1976
3089	Moens	Johan	Statiestraat 7	2950		06/09/1979
3090	Vanbergen	Jacqueline	Turnhoutsebaan 16	2140	0496 49 93 57	29/12/1975
3091	Vanmarcke	Inge	Turnhoutsebaan 194	2110	0478 44 38 89	30/12/1975



# Primary key

## Primaire sleutel

Veld dat de instantie uniek definieert.

PersoneelsID	Familienaam	Voornaam	Straat en nummer	Postcode	gsm	Geboortedatum
1054	Vandenabeele	Jos	Dapperstraat 16	8490	0497 66 35 13	27/01/1966
1055	Vanacker	Veerle	Land van Belofte 26	8000	0497 36 36 86	02/07/1969
1056	Verlinde	Marc	H. Serruyslaan 66	8400	0497 49 71 96	08/12/1975
1057	Liever	Ward	Noordstraat 23	8320	0497 66 73 37	29/05/1980
1058	Dedeurwaerder	Dirk	Zevende Hemel 7	8730	0497 09 72 20	20/02/1983
2012	Quaghebuur	Mieke	Langestraat 98	9000	0496 78 45 12	03/08/1969
2013	Dedecker	Eric	Zoeten Inval 144	9000	0478 74 85 96	29/01/1970
2014	Lieckens	Roger	St-Hubertuslaan 41	9000	0479 50 21 12	18/12/1975
2015	Delporte	Saartje	Vijfwegenstraat 164	9000		09/05/1960
2016	Lievemans	Chris	Trapiestendreef 69	9000	0476 45 67 89	27/01/1966
2017	Lindemans	Klaas	Rue Les Misérables 2	9000		19/02/1975
3088	Filliers	Dirk	De Keyserlei	2000		08/01/1976
3089	Moens	Johan	Statiestraat 7	2950		06/09/1979
3090	Vanbergen	Jacqueline	Turnhoutsebaan 16	2140	0496 49 93 57	29/12/1975
3091	Vanmarcke	Inge	Turnhoutsebaan 194	2110	0478 44 38 89	30/12/1975

# Foreign key

De foreign key is de verbindende schakel tussen twee tables.

						Postcode Gemeente	
						2000	Antwerpen
						2110	Wijnegem
						2140	Borgerhout
						2950	Kapellen
						8000	Brugge
PersoneelsID	Familienaam	Voornaam	Straat en nummer	Postcode	gsm	Geboortedatum	
1054	Vandenabeele	Jos	Dapperstraat 16	8490	0497 66 35 13	27/01/1966	
1055	Vanacker	Veerle	Land van Belofte 26	8000	0497 36 36 86	02/07/1969	
1056	Verlinde	Marc	H. Serruyslaan 66	8400	0497 49 71 96	08/12/1975	
1057	Liever	Ward	Noordstraat 23	8320	0497 66 73 37	29/05/1980	
1058	Dedeurwaerder	Dirk	Zevende Hemel 7	8730	0497 09 72 20	20/02/1983	
2012	Quaghebuur	Mieke	Langestraat 98	9000	0496 78 45 12	03/08/1969	
2013	Dedecker	Eric	Zoeten Inval 144	9000	0478 74 85 96	29/01/1970	
2014	Lieckens	Roger	St-Hubertuslaan 41	9000	0479 50 21 12	18/12/1975	
2015	Delporte	Saartje	Vijfwegenstraat 164	9000		09/05/1960	
2016	Lievemans	Chris	Trapiestendreef 69	9000	0476 45 67 89	27/01/1966	
2017	Lindemans	Klaas	Rue Les Misérables 2	9000		19/02/1975	



# Index

Lijkt op de index achteraan in een boek.  
Met een zoekwoord kan je de juiste bladzijde, of het juiste rijnummer in je table, vinden.

PersoneelsID	Familienaam	Voornaam	Straat en nummer	Postcode	gsm	Geboortedatum
1054	Vandenabeele	Jos	Dapperstraat 16	8490	0497 66 35 13	27/01/1966
1055	Vanacker	Veerle	Land van Belofte 26	8000	0497 36 36 86	02/07/1969
1056	Verlinde	Marc	H. Serruyslaan 66	8400	0497 49 71 96	08/12/1975
1057	Liever	Ward	Noordstraat 23	8320	0497 66 73 37	29/05/1980
1058	Dedeurvaerder	Dirk	Zevende Hemel 7	8730	0497 09 72 20	20/02/1983
2012	Quaghebeur	Mieke	Langestraat 98	9000	0496 78 45 12	03/08/1969
2013	Dedecker	Eric	Zoeten Inval 144	9000	0478 74 85 96	29/01/1970
2014	Lieckens	Roger	St-Hubertuslaan 41	9000	0479 50 21 12	18/12/1975
2015	Delporte	Saartje	Vijfwegenstraat 164	9000		09/05/1960
2016	Lievemans		Leestendreef 69	9000	0476 45 67 89	27/01/1966
2017	Lindemans		Les Misérables 2	9000		19/02/1975
3088	Filliers		eyserlei	2000		08/01/1976
3089	Moens		estraat 7	2950		06/09/1979
3090	Vanbergen		houtsebaan 16	2140	0496 49 93 57	29/12/1975
3091	Vanmarcke		houtsebaan 194	2110	0478 44 38 89	30/12/1975

## 1.3. Standaardisatie

### 1.3.1. Verschillende dialecten

SQL is reeds door veel leveranciers als databasetaal geïmplementeerd: IBM, Microsoft, Oracle,... Het is niet zo dat SQL de naam is van één bepaald product. Elke leverancier heeft een eigen "dialect" gemaakt. Al deze dialecten lijken wel op elkaar, maar ze zijn niet 100% hetzelfde.

Sommige producten kennen meer instructies dan andere. Ook de mogelijkheden van een bepaalde instructie kunnen uiteenlopen. Om de verschillen tussen deze dialecten te beperken, werd al vroeg beslist om een **standaard voor SQL** te definiëren.

### 1.3.2. Ontstaan

Rond 1983 zijn de ISO (International Standardization Organization) en ANSI (American National Standards Institute) begonnen met de ontwikkeling van deze standaard. In 1986 verscheen de eerste SQL-standaard, ook wel **SQL-86-standaard** of **SQL1** genoemd.

SQL1 biedt voor het 'beschermen' van de integriteit van de gegevens in een database maar weinig mogelijkheden. **SQL2** (waarvan de standaard is vastgelegd in 1992) biedt ons wat integriteitsbewaking betreft veel meer: concepten zoals 'primary key', 'foreign key' en 'check'-mogelijkheden vormen enkele van de uitbreidingen van SQL1 naar SQL2.

In deze cursus zullen we de instructies bespreken die door de belangrijkste SQL-producten (MySQL, Oracle, SQL Server, Access,...) ondersteund worden.

## 1.4. Opgaven

Maak de twee opdrachten in de webcursus

## Hoofdstuk 2. Databases

### 2.1. Intro

In deze cursus gebruiken we drie databases.

- Alle voorbeelden uit de cursus zijn gebaseerd op de database BIEREN.
- Alle oefeningen kan je maken in de database PLANTEN.
- De opdrachten voor de coach moet je maken in de database VIDEO.

In dit onderdeel bespreken we de inhoud van de drie databases.

### 2.2. Bieren

De database BIEREN bevat gegevens over verschillende bieren en de brouwerijen waar deze bieren gebrouwen worden.

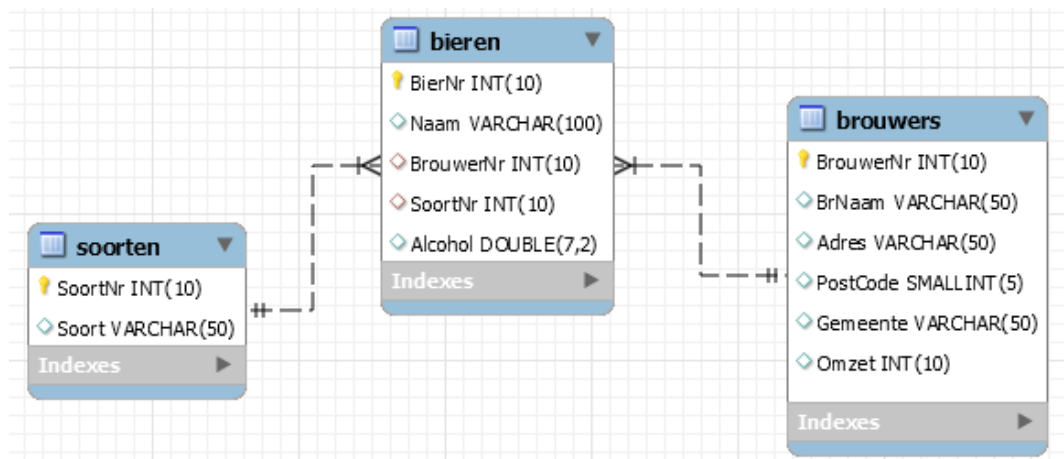
#### Tables

De database Bieren bestaat uit drie tables:

Table	Inhoud
bieren	Bevat een lijst van alle bieren <i>Velden:</i> BierNr, Naam, BrouwerNr, SoortNr, Alcohol. <i>Primary Key:</i> BierNr Het veld BrouwerNr verwijst naar de brouwerij uit de table brouwers. Het veld SoortNr verwijst naar de biersoort uit de table soorten.
brouwers	Bevat een lijst van alle brouwerijen <i>Velden:</i> BrouwerNr, BrNaam, Adres, Postcode, Gemeente, Omzet <i>Primary Key:</i> BrouwerNr
soorten	Bevat een lijst van alle biersoorten <i>Velden:</i> SoortNr, Soort <i>Primary Key:</i> SoortNr

Deze database bevat ook reeds een table *bieren\_oud* die we later nodig hebben. Deze table is een kopie van de structuur van de table *bieren*.

## Schema



Met het script worden er geen relaties gegenereerd. Je zal deze zelf toevoegen in het onderdeel Beheer van tables en relaties.

## Download

Het bijhorende SQL-bestand kan je downloaden in de webversie van de cursus.

## 2.3. Planten

De database PLANTEN bevat de gegevens van een tuincentrum: gegevens over de planten, de leveranciers, aangevraagde offertes en de geplaatste bestellingen.

### Tables

De database Planten bestaat uit zes tables.

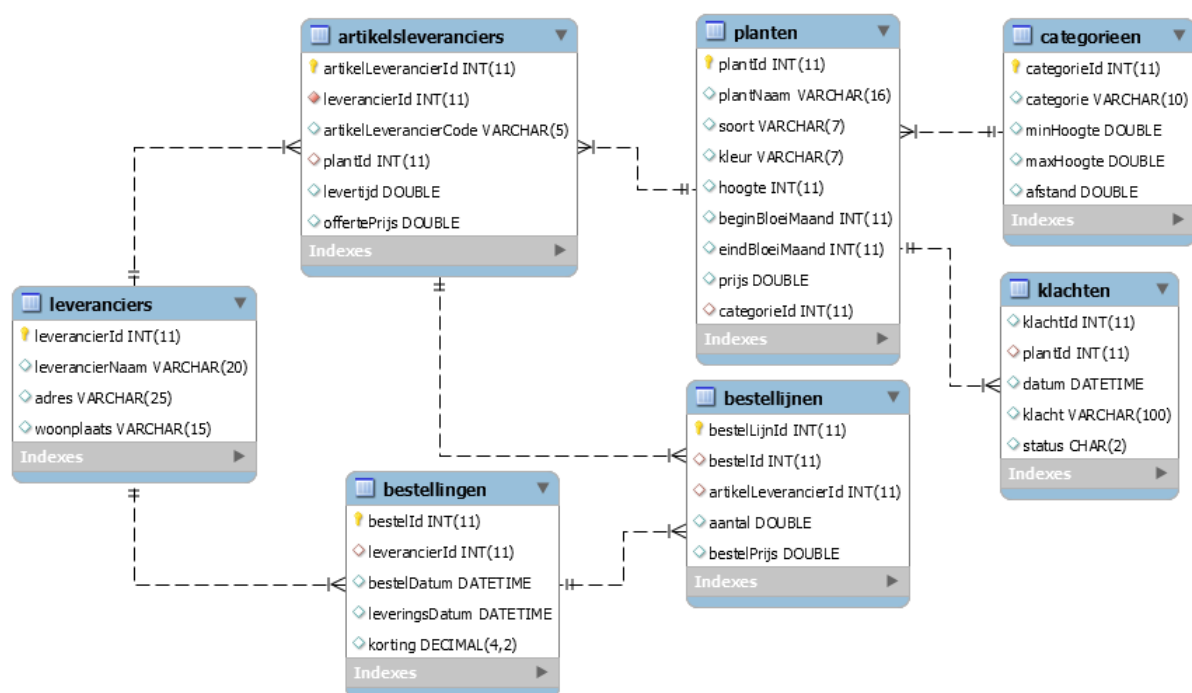
De prijzen zijn uitgedrukt in €, de hoogte in cm.

Alle primaire sleutels zijn autonummeringsvelden.

Deze database bevat ook een table *planten\_oud* die we later nodig hebben. Deze table is een kopie van de structuur van de table *planten*.

Table	Inhoud
planten	Bevat de gegevens van de planten <i>Velden:</i> plantId, plantNaam, soort, kleur, hoogte, beginBloeimaand, eindBloeimaand, prijs, categorieId <i>Primary key:</i> plantId plantNaam bevat de naam van de plant. beginBloeimaand en eindBloeimaand bevat de maanden (1 tot 12) wanneer de bloeiperiode van de plant begint en eindigt. Waarde 0 betekent dat het begin of einde van de bloei niet gekend is.
artikelsleveranciers	Bevat de catalogusgegevens van onze leveranciers <i>Velden:</i> artikelLeverancierId, leverancierId, artikelLeverancierCode, plantId, levertijd, offertePrijs <i>Primary key:</i> artikelLeverancierId artikelLeverancierCode is de code die de leverancier aan de plant geeft.
leveranciers	Bevat de gegevens van de leveranciers <i>Velden:</i> leverancierId, leverancierNaam, adres, woonplaats <i>Primary key:</i> leverancierId
bestellingen	Bevat de bestellingen die wij geplaatst hebben bij de leveranciers <i>Velden:</i> bestelId, leverancierId, bestelDatum, leveringsDatum, korting <i>Primary key:</i> bestelId
bestellijnen	Bevat de detaillijnen van de bestelling <i>Velden:</i> bestelLijnId, bestelId, artikelLeverancierId, aantal, bestelPrijs <i>Primary key:</i> bestelLijnId
categorieën	Bevat de gegevens van de verschillende soorten categorieën van planten <i>Velden:</i> categorieId, categorie, minHoogte, maxHoogte, afstand <i>Primary key:</i> categorieId

## Schema



Met het script worden er geen relaties gegenereerd. Je zal de relaties en de table *klachten* zelf toevoegen in het onderdeel *Beheer van tables en relaties*.

## Download

Het bijhorende SQL-bestand kan je downloaden in de webversie van de cursus.

## 2.4. Video

De database VIDEO bevat de gegevens van een videotheek: klantgegevens, info over de films, de maatschappijen. Ook wie welke film gehuurd heeft staat in de database.

### Tables

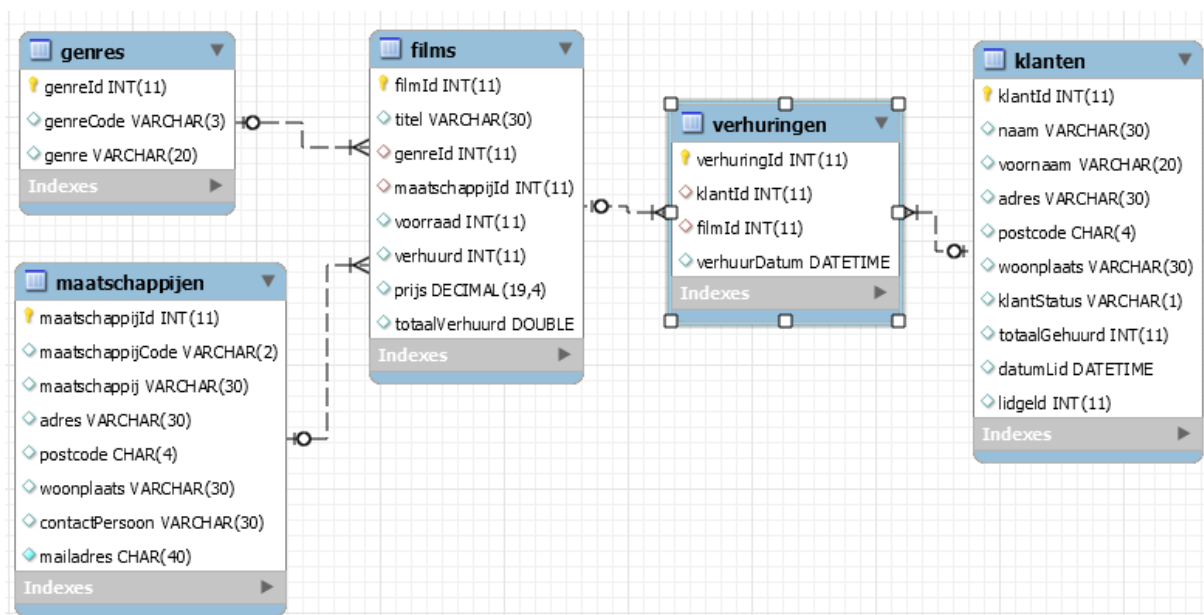
De database Video bestaat uit vijf tables:

Table	Inhoud
films	Bevat de gegevens van de films Velden: filmId, titel, genreId, maatschappijId, voorraad, verhuurd, prijs, totaalVerhuurd Primary key: filmId
genres	Bevat de gegevens van de verschillende filmgenres Velden: genreId, genreCode, genre Primary key: genreId
klanten	Bevat de gegevens van de klanten Velden: klantId, naam, voornaam, adres, postcode, gemeente, klantStatus, totaalGehuurd, datumLid, lidgeld Primary key: klantId De kolom lidgeld bevat de waarde 0 of -1. Als het lidgeld betaald is, staat hier -1. 0 betekent dat het lidgeld nog niet betaald is.
maatschappijen	Bevat de gegevens van de maatschappijen die de films uitgeven Velden: maatschappijId, maatschappijCode, maatschappij, adres, postcode, gemeente, contactPersoon Primary key: maatschappijId
verhuringen	Bevat de verhuringen Velden: verhuringId, klantId, filmId, verhuurDatum Primary key: verhuringId

De prijzen zijn uitgedrukt in €.

Alle primaire sleutels zijn autonummeringsvelden.

### Schema



Met het script worden er geen relaties gegenereerd. Je zal deze zelf toevoegen in het onderdeel *Beheer van tables en relaties*.

### ***Download***

Het bijhorende SQL-bestand kan je downloaden in de webversie van de cursus.

## Hoofdstuk 3. Software

### 3.1. Wat heb je nodig

#### 3.1.1. Workbench

In deze cursus werken we met het programma **MySQL Workbench**.

Het is een uniforme, visuele tool voor databasearchitecten en ontwikkelaars. MySQL Workbench biedt datamodeltering, SQL ontwikkeling en uitgebreide beheertools voor serverconfiguratie, gebruikersbeheer, back-up en nog veel meer .

MySQL Workbench is beschikbaar voor Windows, Linux en Mac OS X.

#### 3.1.2. Installatie

Installeer het programma Workbench. Onderstaande animatie loodst je stap voor stap door de installatie.

In de webcursus vind je stap-voor-stap-animaties voor de installatie van Workbench op Windows en Mac.

### 3.2. Workbench

#### 3.2.1. Database importeren

Voor je aan de slag kan, moet je de databases importeren.

Start het programma MySQL Workbench en log in. Volg nadien de stappen uit de animatie.

#### 3.2.2. Query uitvoeren

Bekijk de animatie in de webcursus om te zien hoe je een SQL-instructie kan uitvoeren.

#### 3.2.3. Schema tonen

Soms is het handig om in één oogopslag de verschillende tables en eventuele relaties te kunnen bekijken.

In de animatie in de webcursus tonen we stap voor stap hoe je dit moet doen.



## Hoofdstuk 4. Gegevens selecteren

### 4.1. Selecteren

In dit hoofdstuk bekijken we de `SELECT`-instructie, de kern van SQL.

We bekijken uitgebreid de verschillende componenten waaruit deze instructie opgebouwd is.

#### 4.1.1. Syntax

Met de `SELECT`-instructie kunnen we vragen stellen aan onze **database**:

- Wat is het adres van werknemer Willy Maes?
- Hoeveel schoenen worden er per regio per maand verkocht?
- Geef een lijst van alle klanten die deze maand iets besteld hebben.
- ....

De **syntax** ziet er als volgt uit:

```
SELECT [predikaat] { * | table.* | [table.]veld1 [, [table.]veld2 [,...]] }
FROM tableexpressie
```

De **instructie** bevat volgende onderdelen:

Onderdeel	Betekenis
<b>predikaat</b>	Een van de volgende predikaten: <i>ALL</i> of <i>DISTINCT</i> . Met een predikaat kan je het aantal records in het resultaat beperken. Als je geen predikaat opgeeft, gaat SQL uit van de standaardinstelling <i>ALL</i> .
<b>*</b>	Bepaalt dat alle velden uit de opgegeven table of tables geselecteerd worden.
<b>table</b>	De naam van de table die de velden bevat waaruit records worden geselecteerd. Dit wordt vooral gebruikt als je gegevens uit meerdere tables selecteerd.
<b>veld1, veld2</b>	De namen van de velden waaruit gegevens worden opgehaald. Als je meer dan één veld opgeeft, worden deze opgehaald in de opgegeven volgorde.
<b>tableexpressie</b>	De naam van de table of de namen van de tables waaruit je gegevens wilt ophalen.

In de volgende hoofdstukken zullen we deze syntax verder uitbreiden.

#### 4.1.2. Voorbeelden

De minimale instructie ziet er als volgt uit:

```
SELECT velden FROM table
```

Hieronder vind je vier voorbeelden. Test elk van de voorbeelden uit in de database **BIEREN**.

### Voorbeeld 1

Je wil een lijst met de namen van alle brouwerijen.

*Oplossing:*

```
SELECT brnaam FROM brouwers
```

*Resultaat:* 118 records

1 • SELECT brnaam FROM brouwers

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

brnaam
Achouffe
Alken
Ambly
Anker
Artois
Bavik
Belle Vue
Belle Vue - Molenbeek
Belle Vue - Zuun
Bie (De)
Binchoise
Bios
Blauw

brouwers 1 x

### Voorbeeld 2

Je wil een lijst met de naam en het alcoholpercentage van alle bieren.

*Oplossing:*

```
SELECT naam, alcohol FROM bieren
```

*Resultaat:* 1215 records

1 • SELECT naam, alcohol FROM bieren

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows: |

naam	alcohol
A.C.O.	7.00
Aalbeeks St. Corneliusbier (=Kapittel pater (Het))	6.50
Aardbeien witbier	2.50
Aarschots kruikenbier (=St. Sebastiaan grand cru)	7.60
Abt Bijbier (Nen)	7.00
Adler	6.75
Aerts 1900	7.00
Affligem blond (Abdij)	7.00
Affligem christmas ale (Abdij)	9.00
Affligem dubbel (Abdij)	7.00
Affligem patersvat	7.00
Affligem tripel (Abdij)	8.50
Alka nileener	5.00

bieren 2 x

### Voorbeeld 3

Je wil een lijst met alle gegevens van alle bieren.

*Oplossing:*

```
SELECT * FROM bieren
```

*Resultaat:* 1215 records

BierNr	Naam	BrouwerNr	SoortNr	Alcohol
4	A.C.O.	104	18	7.00
5	Aalbeeks St. Corneliusbier (=Kapittel pater (Het))	113	18	6.50
7	Aardbeien witbier	56	53	2.50
8	Aarschots kruikenbier (=St. Sebastiaan grand cru)	105	15	7.60
10	Abt Bijbier (Nen)	33	18	7.00
11	Adler	51	42	6.75
12	Aerts 1900	81	14	7.00
13	Affligem blond (Abdij)	100	33	7.00
14	Affligem christmas ale (Abdij)	100	36	9.00
15	Affligem dubbel (Abdij)	100	14	7.00
16	Affligem patersvat	100	33	7.00
17	Affligem tripel (Abdij)	100	59	8.50
18	Akila pilsener	68	42	5.00

### Voorbeeld 4

Je wil een lijst van gemeentes waar de brouwerijen gevestigd zijn.

*Oplossing:*

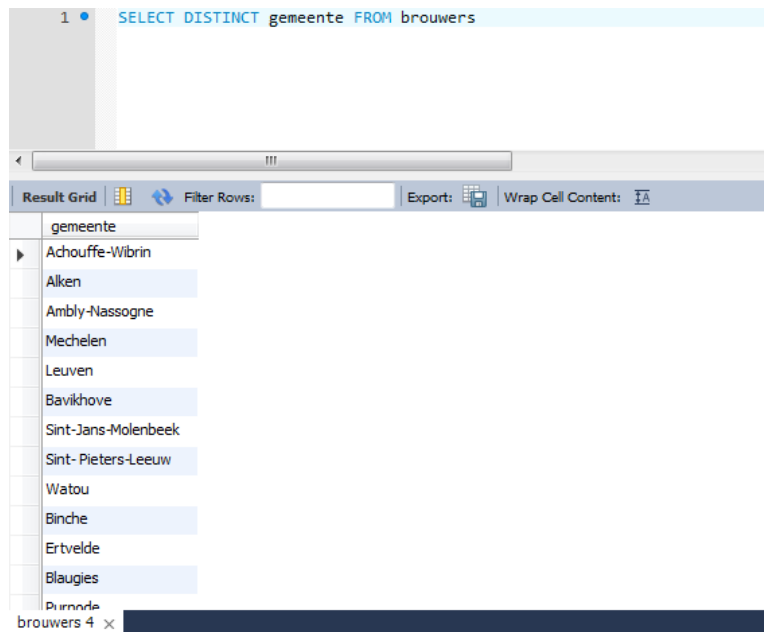
```
SELECT gemeente FROM brouwers
```

Sommige gemeenten komen meerdere keren voor op deze lijst omdat er meerdere brouwerijen in dezelfde gemeente gevestigd zijn. Om elke gemeente slechts één keer op de lijst te zien, gebruiken we het predikaat **DISTINCT**. DISTINCT zorgt er voor dat elke rij uniek is.

*Oplossing:*

```
SELECT DISTINCT gemeente FROM brouwers
```

*Resultaat:* 109 records



### 4.1.3. Opdracht

Maak de twee opdrachten in de webversie.

## 4.2. Where-voorwaarde

In de voorgaande voorbeelden werden steeds alle records getoond.

In dit onderdeel bekijken we hoe je een deel van de records kan selecteren.

### 4.2.1. Syntax

De **where**-component kan je zien als een filter. Alleen de rijen die aan de **voorwaarde** voldoen, worden in het eindresultaat getoond.

De **syntax** van onze SELECT ziet er dan als volgt uit:

```
SELECT [predikaat] { * | table.* | [table.]veld1 [AS alias1] [, [table.]veld2 [AS alias2] [, ...]] }
FROM tableexpressie [, ...]
[WHERE attribuut operator constante | attribuut ]
```

De **operator** kan zijn: <, >, <=, >=, <>, =, like, between ... and ... , in (...)

De **constante** kan een getal of een reeks karakters zijn. Een karakterreeks moet altijd beginnen en eindigen met een quote (').

De vergelijking en sortering van **karakterreeksen** is afhankelijk van de ingestelde collation van je databaseserver. De standaard collation van MySQL maakt geen onderscheid tussen hoofd- en kleine letters.

### 4.2.2. Like

De operator **like** biedt een aantal speciale mogelijkheden. In onderstaande tabel staan de belangrijkste opgesomd.

Aard van de selectie	Patroon	Waarden die in het patroon passen	Waarden die niet in het patroon passen
Meerdere karakters	a%a %ab% ab%	aa, aBa, aBBBa abc, AABb, Xab abcdefg, abc, ab	aBC aZb, bac cab, aab
Eén karakter	a_a	aaa, a3a, aBa	aBBBa

Deze mogelijkheden kunnen natuurlijk ook gecombineerd worden.

#### 4.2.3. Voorbeelden

Hieronder vind je 7 voorbeelden op het toepassen van WHERE. Probeer elk voorbeeld uit in de database BIENEN.

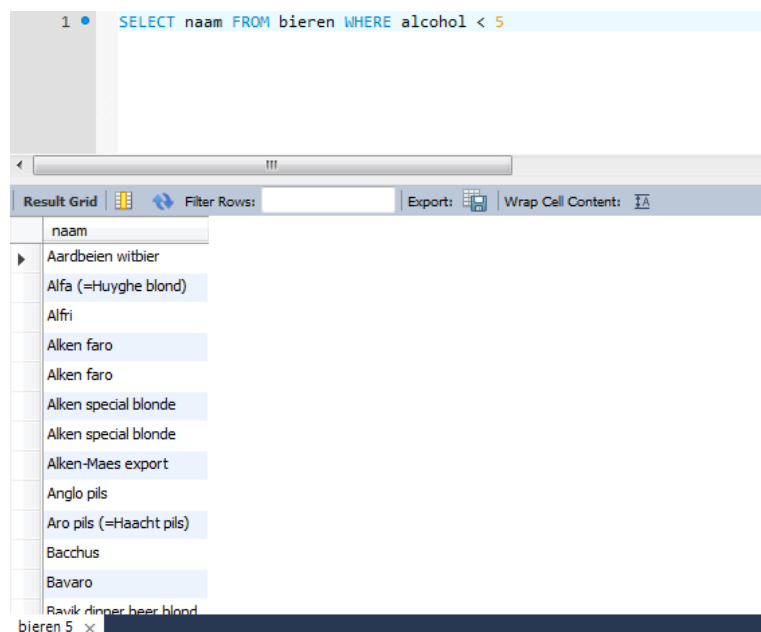
##### Voorbeeld 1

Je wil een lijst met alle bieren met een alcoholpercentage lager dan 5%.

*Oplossing:*

```
SELECT naam FROM bieren WHERE alcohol < 5
```

*Resultaat:* 249 records



##### Voorbeeld 2

Je wil een lijst van alle brouwerijen in Beersel.

*Oplossing:*

```
SELECT brnaam FROM brouwers WHERE gemeente='Beersel'
```

*Resultaat:* 2 records

SQL

1 • `SELECT brnaam FROM brouwers WHERE gemeente='Beersel'`

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
brnaam			
Drie Fonteynen			
Vandervelden Henri			

### Voorbeeld 3

Je wil een lijst met bieren waarbij de naam de tekst 'ale' bevat.

*Oplossing:*

`SELECT naam FROM bieren WHERE naam like '%ale%'`

*Resultaat:* 72 records

1 • `SELECT naam FROM bieren WHERE naam like '%ale%'`

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
naam			
Affligem christmas ale (Abdij)			
Aldegonde speciale			
Ale brau (=Vega pils)			
Alexander Rodenbach			
Balens kruierke			
Bass pale ale			
Bastions cuvee speciale (Les)			
Belgian ale traditional gueuze			
Bergeotte normale (=Regal (Saison))			
Bergeotte speciale (=St. Benoit brune (brassin ...			
Bousval blonde speciale (=Floreffe blonde)			
Bousval brune speciale (=Floreffe double)			
River christmas beer (=Affligem christmas ale)			

bieren 7 x

Ook in dit voorbeeld zie je het effect van de collation. Het bier Ale brau (=Vega pils) wordt ook getoond.

### Voorbeeld 4

Je wil een lijst met alle bieren met een alcoholpercentage tussen 5% en 7%.

*Oplossing:*

`SELECT naam FROM bieren WHERE alcohol between 5 and 7`

*Resultaat:* 536 records

De bieren met alcoholpercentage 5 of 7 worden ook meegenomen in het resultaat.

1 • `SELECT naam FROM bieren WHERE alcohol between 5 and 7`

naam
A.C.O.
Aalbeeks St. Corneliusbier (=Kapittel pater (Het))
Abt Bijbier (Nen)
Adler
Aerts 1900
Affligem blond (Abdij)
Affligem dubbel (Abdij)
Affligem patersvat
Akila pilsener
Ale brau (=Vega pils)
Alexander Rodenbach
All Black
Amelia carise

bieren 8 x

### Voorbeeld 5

Je wil een lijst met alle bieren met 0%, 5% of 8% als alcoholpercentage.

*Oplossing:*

```
SELECT naam FROM bieren WHERE alcohol in (0, 5, 8)
```

*Resultaat:* 289 records

1 • `SELECT naam FROM bieren WHERE alcohol in (0, 5, 8)`

naam
Akila pilsener
Ale brau (=Vega pils)
Ambiorix dubbel
Amelie
Amuzantje (=Lamoral Degmont)
Ankerpils (=Wieze pils)
Arabier
Ardenne blonde (Biere d')
Argus pils (=Premium pilsener)
Argus witte (=Limburgse witte)
Arjaun (=Lamoral Degmont)
Astor
ΔHac (=Vega pils)

bieren 9 x

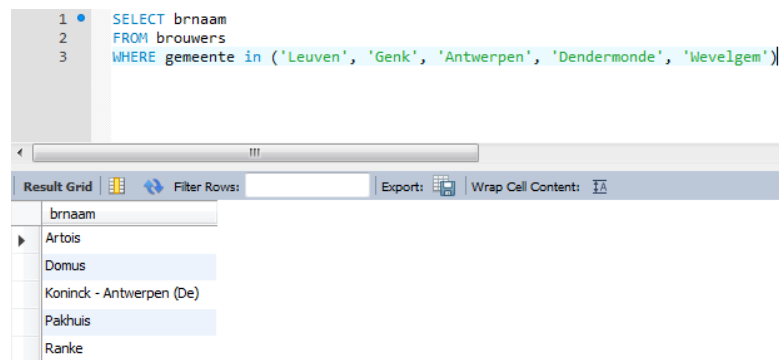
### Voorbeeld 6

Je wil een lijst van de brouwerijen uit Leuven, Genk, Antwerpen, Dendermonde en Wevelgem.

*Oplossing:*

```
SELECT brnaam FROM brouwers
WHERE gemeente in ('Leuven', 'Genk', 'Antwerpen', 'Dendermonde', 'Wevelgem')
```

Resultaat: 5 records



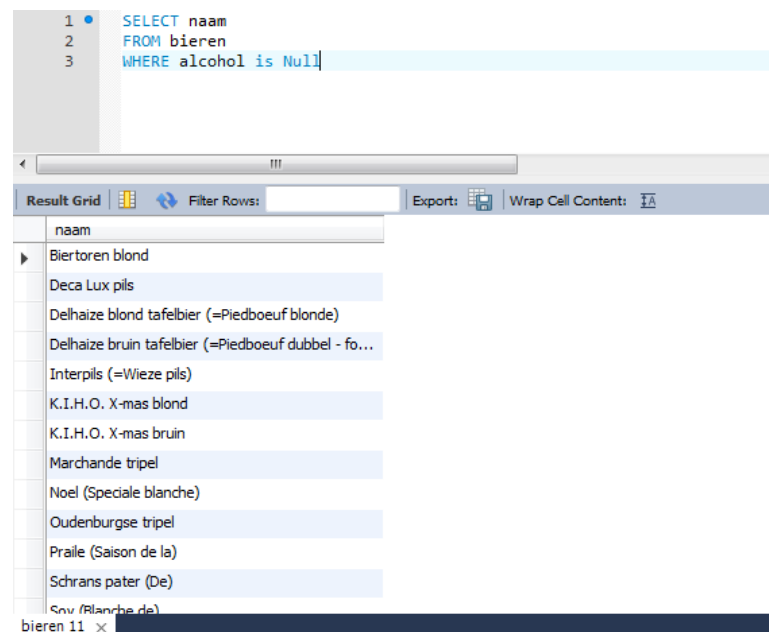
### Voorbeeld 7

Je wil een lijst met alle bieren waarbij het alcoholpercentage niet ingevuld is.

Oplossing:

```
SELECT naam FROM bieren WHERE alcohol is Null
```

Resultaat: 20 records



### 4.2.4.And, Or en Not

Binnen je where kan je verschillende voorwaarden combineren met AND, OR en NOT.

- AND: alle voorwaarden moeten waar zijn
- OR: één van de voorwaarden moet waar zijn
- NOT: negatie van de voorwaarde, waar wordt onwaar en omgekeerd

### 4.2.5.Voorbeelden

Hier vind je twee voorbeelden op het combineren van voorwaarden.

Probeer deze voorbeelden ook uit in de database BIEREN.



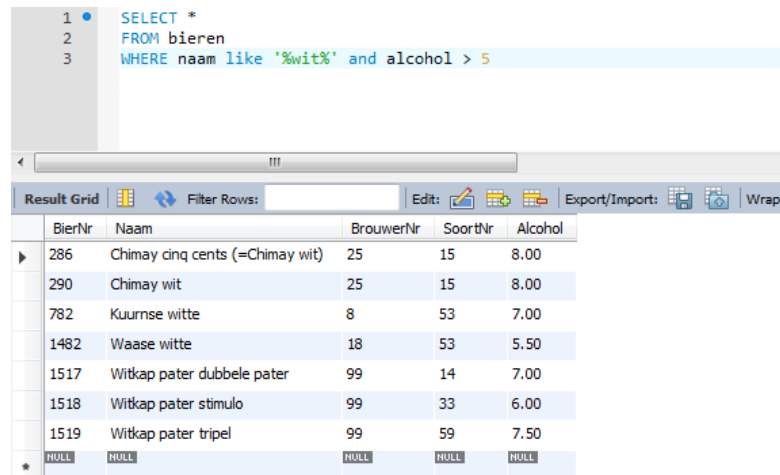
### Voorbeeld 1

Je wil een lijst met alle bieren waarbij in de naam het woord 'wit' voorkomt en met een alcoholpercentage hoger dan 5.

*Oplossing:*

```
SELECT * FROM bieren WHERE naam like '%wit%' and alcohol > 5
```

*Resultaat: 7 records*



The screenshot shows a SQL query editor with the following query:

```
1 SELECT *
2 FROM bieren
3 WHERE naam like '%wit%' and alcohol > 5
```

Below the query is a result grid with the following data:

BierNr	Naam	BrouwerNr	SoortNr	Alcohol
286	Chimay cinq cents (=Chimay wit)	25	15	8.00
290	Chimay wit	25	15	8.00
782	Kuurnse witte	8	53	7.00
1482	Waase witte	18	53	5.50
1517	Witkap pater dubbele pater	99	14	7.00
1518	Witkap pater stimulo	99	33	6.00
1519	Witkap pater tripel	99	59	7.50
NULL	NULL	NULL	NULL	NULL

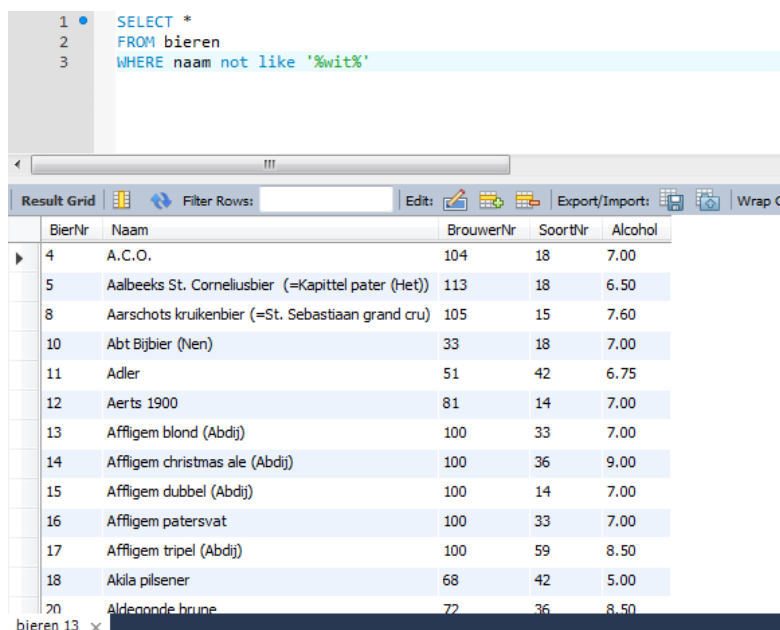
### Voorbeeld 2

Je wil een lijst met alle bieren waarbij het woord 'wit' niet in de naam voorkomt.

*Oplossing:*

```
SELECT * FROM bieren WHERE naam not like '%wit%'
```

*Resultaat: 1185 records*



The screenshot shows a SQL query editor with the following query:

```
1 SELECT *
2 FROM bieren
3 WHERE naam not like '%wit%'
```

Below the query is a result grid with the following data:

BierNr	Naam	BrouwerNr	SoortNr	Alcohol
4	A.C.O.	104	18	7.00
5	Aalbeeks St. Corneliusbier (=Kapittel pater (Het))	113	18	6.50
8	Aarschots kruikenbier (=St. Sebastiaan grand cru)	105	15	7.60
10	Abt Bijbier (Nen)	33	18	7.00
11	Adler	51	42	6.75
12	Aerts 1900	81	14	7.00
13	Affligem blond (Abdij)	100	33	7.00
14	Affligem christmas ale (Abdij)	100	36	9.00
15	Affligem dubbel (Abdij)	100	14	7.00
16	Affligem patersvat	100	33	7.00
17	Affligem tripel (Abdij)	100	59	8.50
18	Akila pilsener	68	42	5.00
20	Aldegonde brune	72	36	8.50

#### 4.2.6. Opdracht

Maak de opdrachten in de webversie van deze cursus.

### 4.3. Sorteren

Tot nu worden de gegevens getoond zoals ze in de achterliggende table bewaard worden.

In dit onderdeel bekijken we hoe je de volgorde zelf kan bepalen door te sorteren.

Denk eraan:

De sortering van karakterreeksen is afhankelijk van de ingestelde collation van MySQL. De standaard collation van MySQL maakt geen onderscheid tussen hoofd- en kleine letters.

#### 4.3.1. Syntax

Op SQL-gebaseerde databasesystemen sorteren niet automatisch. Je zal hiertoe zelf de opdracht moeten geven.

Syntax:

```
SELECT [predikaat] { * | table.* | [table.]veld1 [AS alias1] [, [table.]veld2 [AS alias2] [, ...]] }  
FROM tableexpressie [, ...]  
[WHERE ...]  
[ORDER BY attribuut asc|desc [, attribuut asc|desc [,...]] ]
```

Als attribuut kan je een veldnaam gebruiken. Je kan ook het kolomnummer (eerste kolom = 1) gebruiken.

Met asc sorteer je van klein naar groot, met desc van groot naar klein.

#### 4.3.2. Voorbeelden

Hieronder vind je 3 voorbeelden. Test ze alle drie uit in de database BIEREN.

##### *Voorbeeld 1*

Je wil een alfabetische lijst van de bieren met hun alcoholpercentage.

*Oplossing:*

```
SELECT naam, alcohol FROM bieren ORDER BY naam asc
```

*Resultaat:* 1215 records

```

1 • SELECT naam, alcohol
2   FROM bieren
3   ORDER BY naam asc

```

naam	alcohol
A.C.O.	7.00
Aalbeeks St. Corneliusbier (=Kapittel pater (Het))	6.50
Aardbeien witbier	2.50
Aarschots kruikenbier (=St. Sebastiaan grand cru)	7.60
Abt Bijbier (Nen)	7.00
Adler	6.75
Aerts 1900	7.00
Affligem blond (Abdij)	7.00
Affligem christmas ale (Abdij)	9.00
Affligem dubbel (Abdij)	7.00
Affligem patersvat	7.00
Affligem tripel (Abdij)	8.50

bieren 14 ×

### Voorbeeld 2

Je wil een alfabetische lijst van de bieren met hun alcoholpercentage.

*Oplossing:*

```
SELECT naam, alcohol FROM bieren ORDER BY 1 asc
```

We willen sorteren op naam. Naam staat als eerste in de select, dus gebruiken we het cijfer 1.

*Resultaat:* 1215 records

```

1 • SELECT naam, alcohol
2   FROM bieren
3   ORDER BY 1 asc

```

naam	alcohol
A.C.O.	7.00
Aalbeeks St. Corneliusbier (=Kapittel pater (Het))	6.50
Aardbeien witbier	2.50
Aarschots kruikenbier (=St. Sebastiaan grand cru)	7.60
Abt Bijbier (Nen)	7.00
Adler	6.75
Aerts 1900	7.00
Affligem blond (Abdij)	7.00
Affligem christmas ale (Abdij)	9.00
Affligem dubbel (Abdij)	7.00
Affligem patersvat	7.00
Affligem tripel (Abdij)	8.50

bieren 15 ×

### Voorbeeld 3

Je wil een lijst met alle bieren (naam en alcoholpercentage). De bieren met het hoogste alcoholpercentage staan bovenaan. Zorg ook voor een sortering op naam.

Oplossing:

```
SELECT naam, alcohol FROM bieren ORDER BY alcohol desc, naam asc
```

Resultaat: 1215 records

naam	alcohol
Fitt	15.00
Bush beer	12.00
Bush de Noel	12.00
Fantome brewery (The best of)	12.00
Gaverhopke (t) bruin 12	12.00
Kasteel van Ramegnies-Chin (Bier v/h)	12.00
Rochefort 10	11.30
Kasteelbier Ingelmunster	11.00
Kasteelbier Ooidonk (=Kasteelbier Ingelmunster)	11.00
Sioen Maeseycker kloosterbier	11.00
Westvleteren 12	11.00
Boucanier (Biere du) (=Piraat)	10.50

### 4.3.3. Opdracht

Maak de opdrachten in de webversie.

## 4.4. Opgaven 1

Hieronder krijg je een hele reeks oefeningen op:

- gegevens selecteren
- where-voorwaarde toepassen
- sorteren

Om deze oefeningen uit te proberen, heb je de database PLANTEN nodig. Bij elk van de oefeningen krijg je (een deel van) het resultaat van je instructie te zien.

De oplossingen van deze oefeningen vind je in de webcursus.

### 4.4.1. Opgave 1

Geef een overzicht, met alle gegevens, van de leveranciers uit Aalsmeer.

Resultaat: 4 records

	leverancierId	leverancierNaam	adres	woonplaats
▶	4	SPITMAN EN ZN.	ACHTERTUIN 9	AALSMEER
	6	MOOIWEER FA.	VERLENGDE ZOMERSTR. 24	AALSMEER
	10	DE GROENE KAS BV.	GLASWEG 1	AALSMEER
	11	FLORA BV.	OEVERSTRAAT 76	AALSMEER
*	NULL	NULL	NULL	NULL

#### 4.4.2. Opgave 2

Geef een alfabetisch overzicht op plantennaam, met plantId, plantennaam en prijs, van alle planten.

*Resultaat:* 119 records

*Begin van de lijst:*

	plantId	plantNaam	prijs
▶	48	ACACIA	17.5
	52	AFRIKAANTJE	0.6
	55	AKELEI	2.5
	88	ALYSSUM	0.6
	53	ANEMOON	3.5
	117	ANJER	3
	110	ASTER	0.75
	107	AZALEA	17.5
	84	AZIJBLOOM	9.5
	59	BASILICUM	2
	1	BEGONIA	0.65

#### 4.4.3. Opgave 3

Welke planten beginnen in de maand maart te bloeien?

Toon plantId, plantennaam, en de beginmaand van de bloeiperiode.

*Resultaat:* 3 records

	plantId	plantNaam	beginBloeimaand
▶	86	FORSYTHIA	3
	95	POPULIER	3
	102	VIOOLTJE	3
*	NULL	NULL	NULL

#### 4.4.4. Opgave 4

Maak een overzicht uit de table *artikelsleveranciers*, gesorteerd op plantId en binnen plantId een sortering op artikelcode van de leverancier.

Toon alleen de gegevens plantId, artikelcode van de leverancier en leverancierId.

*Resultaat:* 218 records

*Begin van de lijst:*

	plantId	artikelLeverancierCode	leverancierId
▶	1	209	7
	1	498-1	5
	1	B101	4
	1	BEGO	10
	2	HEDE	10
	2	ST3P2	11
	3	257-V	5
	3	CHMA	10
	3	CHRY	6
	4	ST4P1	11
	5	LUPI	6

#### 4.4.5. Opgave 5

Maak een gesorteerd overzicht van alle waterplanten. Sorteer op hoogte, grootste voorop.

Resultaat: 9 records

	plantId	plantNaam	soort	kleur	hoogte	beginBloeimaand	eindBloeimaand	prijs	categorieId
▶	7	LISDODDE	WATER	GEEL	200	8	9	4.5	5
	40	CYPERGRAS	WATER	NULL	100	0	0	5	3
	29	KALMOES	WATER	NULL	90	0	0	4.5	3
	77	DOTTERBLOEM	WATER	GEEL	30	4	6	4.5	2
	119	WOLGRAS	WATER	WIT	30	5	6	3.5	2
	4	WATERLELIE	WATER	WIT	0	0	0	12	1
	12	WATERHYACINT	WATER	BLAUW	0	6	9	5	1
	41	BLAASJESKRUID	WATER	GEEL	0	7	8	2.5	1
	80	KIKKERBEET	WATER	WIT	0	7	8	1.25	1
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#### 4.4.6. Opgave 6

Maak een lijst van de verschillende kleuren die bij de planten uit de table *planten* horen.

Resultaat: 13 records

	kleur
▶	NULL
	BLAUW
	BRUIN
	GEEL
	GEMENGD
	GROEN
	LILA
	ORANJE
	PAARS
	ROOD
	ROZE
	VIOLET
	WIT

#### 4.4.7. Opgave 7

Maak een lijst van alle planten waarvan de kolom kleur niet ingevuld is.

Resultaat: 11 records

	plantId	plantNaam	soort	kleur	hoogte	beginBloeimaand	eindBloeimaand	prijs	categorieId
▶	2	KLIMOP	KLIM	NULL	0	0	0	7.5	5
	6	JENEVERBES	BOOM	NULL	250	0	0	6.5	6
	11	DWERG CYPRES	BOOM	NULL	500	0	0	13.5	6
	22	PETERSELIE	KRUID	NULL	25	0	0	2	2
	25	ZILVERSPAR	BOOM	NULL	3000	0	0	19.5	6
	29	KALMOES	WATER	NULL	90	0	0	4.5	3
	40	CYPERGRAS	WATER	NULL	100	0	0	5	3
	70	HULST	HEESTER	NULL	700	0	0	22.5	6
	78	BERK	BOOM	NULL	2000	0	0	7.5	6
	93	SPAR	BOOM	NULL	3000	0	0	12.5	6
	101	WIJNSTOK	KLIM	NULL	600	0	0	10	6
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#### 4.4.8. Opgave 8

Toon de verschillende soorten planten in de table *planten*.

*Resultaat:* 10 records

soort
1-JARIG
KLIM
VAST
WATER
BOOM
HEESTER
KRUID
BOL
2-JARIG
HEIDE

#### 4.4.9. Opgave 9

Geef een overzicht van alle vaste planten met gele bloemen. Van iedere plant toon je de volgende gegevens: plantId, plantennaam, hoogte en de beginmaand van de bloeiperiode.

*Resultaat:* 3 records

plantId	plantNaam	hoogte	beginBloeimaand
23	WOLFSMELK	60	4
105	DOVENETEL	25	4
113	ZONNEBLOEM	150	8
NULL	NULL	NULL	NULL

#### 4.4.10. Opgave 10

Geef een overzicht van alle planten met een prijs boven de 10 € die niet tot de soort bomen behoren.

*Resultaat:* 8 records

plantId	plantNaam	soort	kleur	hoogte	beginBloeimaand	eindBloeimaand	prijs	categorieId
4	WATERLELIE	WATER	WIT	0	0	0	12	1
32	PEPERBOOMPJE	HEESTER	ROZE	125	2	3	15	4
38	SERING	HEESTER	PAARS	500	5	6	19.5	6
39	RODODENDRON	HEESTER	ROOD	125	5	7	19.5	4
54	MAGNOLIA	HEESTER	WIT	1000	4	5	14.5	6
70	HULST	HEESTER	NULL	700	0	0	22.5	6
107	AZALEA	HEESTER	ORANJE	200	4	5	17.5	5
112	TOVERHAZELAAR	HEESTER	GEEL	500	1	2	32	6
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#### 4.4.11. Opgave 11

Maak een lijst van alle planten die in juni beginnen te bloeien en witte bloemen geven en tevens van alle planten die in augustus voor het eerst bloeien en gele bloemen hebben.

Plaats alle beschikbare gegevens in het overzicht.

Resultaat: 9 records

	plantId	plantNaam	soort	kleur	hoogte	beginBloeimaand	eindBloeimaand	prijs	categorieId
▶	3	MARGRIET	VAST	WIT	70	6	8	2.5	3
	7	LISDODDE	WATER	GEEL	200	8	9	4.5	5
	37	VUURDOORN	HEESTER	WIT	0	6	6	5	5
	48	ACACIA	BOOM	WIT	2500	6	6	17.5	6
	51	KAMILLE	VAST	WIT	70	6	7	3	3
	81	GIPSKRUID	1-JARIG	WIT	50	6	7	1	2
	113	ZONNEBLOEM	VAST	GEEL	150	8	9	2	4
	115	ZONNEBLOEM	1-JARIG	GEEL	150	8	10	1.8	4
	117	ANJER	VAST	WIT	40	6	8	3	2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#### 4.4.12. Opgave 12

Welke planten met gemengde bloeikleuren worden maximum 60 cm hoog?  
Toon plantId, plantennaam en hoogte.

Resultaat: 13 records

	plantId	plantNaam	hoogte
▶	8	RIDDERSPoor	50
	24	VIOLIER	60
	49	LEEUWEBEKJE	50
	74	VINGERHOEDSKRUID	0
	82	STRUJKHEIDE	30
	90	VLAMBLOEM	30
	94	KLAPROOS	40
	97	SLEUTELBLOEM	25
	98	DAHLIA	40
	102	VIOOLTJE	15
	104	VIOLIER	60
	108	KLAPROOS	35
	110	ASTER	50
*	NULL	NULL	NULL

#### 4.4.13. Opgave 13

Geef een overzicht van alle leveranciers die niet in Hillegom wonen.

Resultaat: 9 records

	leverancierId	leverancierNaam	adres	woonplaats
▶	1	HOVENIER G.H.	ZANDWEG 50	LISSE
	3	STRUJK BV.	BESSENLAAN 1	LISSE
	4	SPITMAN EN ZN.	ACHTERTUIN 9	AALSMEER
	5	DEZAAIER L.J.A.	DE GRONDEN 101	LISSE
	6	MOOIWEER FA.	VERLENGDE ZOMERSTR. 24	AALSMEER
	8	TRA A.	KOELEPLEKSTRAAT 10	LISSE
	9	ERICA BV.	BERKENWEG 87	HEEMSTED
	10	DE GROENE KAS BV.	GLASWEG 1	AALSMEER
	11	FLORA BV.	OEVERSTRAAT 76	AALSMEER
*	NULL	NULL	NULL	NULL



#### 4.4.14. Opgave 14

Van welke planten is de kleur onbekend? Geef plantId en plantennaam.

Resultaat: 11 records

	plantId	plantNaam
▶	2	KLIMOP
	6	JENEVERBES
	11	DWERG CYPRES
	22	PETERSELIE
	25	ZILVERSPAR
	29	KALMOES
	40	CYPERGRAS
	70	HULST
	78	BERK
	93	SPAR
	101	WIJNSTOK
★	NULL	NULL

#### 4.4.15. Opgave 15

Welke planten bloeien in ieder geval in de periode augustus tot en met oktober? Geef plantId en plantennaam.

Resultaat: 6 records

	plantId	plantNaam
▶	52	AFRIKAANTJE
	53	ANEMOON
	87	PETUNIA
	98	DAHLIA
	110	ASTER
	115	ZONNEBLOEM
★	NULL	NULL

#### 4.4.16. Opgave 16

Welke planten bloeien in ieder geval in de maand september? Geef plantId en plantennaam.

Resultaat: 26 records

Begin van de lijst:

	plantId	plantNaam
▶	1	BEGONIA
	7	LISDODDE
	9	OLJFWILG
	12	WATERHYACINT
	13	VUURPIJL
	18	GANZERIJK
	21	BEREKLAUW
	27	PAMPUSGRAS
	28	MONNIKS KAP
	42	DOPHEIDE
	43	BOOMHEIDE
	47	STOKROOS
	52	AFRIKAANTJE
	53	ANEMOON

#### 4.4.17. Opgave 17

Geef een overzicht van alle vaste planten met een prijs tussen 3 € en 5 €.

*Resultaat: 17 records*

*Begin van de lijst:*

	plantId	plantNaam	soort	kleur	hoogte	beginBloeimaand	eindBloeimaand	prijs	categorieId
▶	13	VUURPIJL	VAST	ROOD	120	6	9	3.5	4
	14	KRUISDISTEL	VAST	BLAUW	75	6	7	3	3
	16	RIDDERSPOOR	VAST	LILA	150	6	7	3	4
	21	BEREKLAUW	VAST	WIT	100	7	9	3.5	3
	28	MONNIKSKAP	VAST	VIOLET	120	8	9	3	4
	30	NAGELKRUID	VAST	ORANJE	50	7	8	3	2
	51	KAMILLE	VAST	WIT	70	6	7	3	3
	53	ANEMOON	VAST	ROZE	50	8	10	3.5	2
	58	KLAPROOS	VAST	ROOD	70	5	6	3	3
	60	BOTERBLOEM	VAST	WIT	50	5	6	3	2
	62	KOGELDISTEL	VAST	BLAUW	175	6	7	3	5
	73	GIPSKRUID	VAST	WIT	90	7	8	4.5	3
	79	DAGLELIE	VAST	ROOD	80	6	8	3	3
	100	DOVENETEL	VAST	ROOD	50	6	7	4.5	2

#### 4.4.18. Opgave 18

Geef een overzicht van alle planten die in maart, april, september of oktober beginnen te bloeien.

*Resultaat: 15 records*

*Begin van de lijst:*

	plantId	plantNaam	soort	kleur	hoogte	beginBloeimaand	eindBloeimaand	prijs	categorieId
▶	9	OLIJFWILG	HEESTER	GEEL	400	9	10	10	6
	23	WOLFSMELK	VAST	GEEL	60	4	4	2.5	2
	26	TULP	BOL	GEEL	30	4	6	0.4	2
	27	PAMPUSGRAS	VAST	WIT	300	9	10	9.5	5
	34	MUURBLOEM	2-JARIG	BRUIN	50	4	5	1.8	2
	54	MAGNOLIA	HEESTER	WIT	1000	4	5	14.5	6
	64	BREM	HEESTER	GEEL	150	4	7	5	4
	77	DOTTERBLOEM	WATER	GEEL	30	4	6	4.5	2
	86	FORSYTHIA	HEESTER	GEEL	250	3	4	5.5	5
	95	POPULIER	BOOM	WIT	1000	3	4	4.5	6
	97	SLEUTELBLOEM	2-JARIG	GEME...	25	4	5	2	2
	102	VIOOLTJE	2-JARIG	GEME...	15	3	8	0.5	1
	105	DOVENETEL	VAST	GEEL	25	4	5	2	2
	107	AZALEA	HEESTER	ORANJE	200	4	5	17.5	5

#### 4.4.19. Opgave 19

Van welke planten is zowel de kleur als de hoogte onbekend? Geef plantId, plantennaam, kleur en hoogte.

*Resultaat: geen records*

#### 4.4.20. Opgave 20

Bij welke planten komt het woord BOOM voor? Geef plantId en plantennaam.

SQL

Resultaat: 5 records

	plantId	plantNaam
▶	32	PEPERBOOMPJE
	43	BOOMHEIDE
	44	TULPEBOOM
	50	JUDASBOOM
	84	AZIJBOM
*	NULL	NULL

#### 4.4.21. Opgave 21

Geef plantId en plantennaam van alle planten die als derde letter een N hebben.

Resultaat: 11 records

	plantId	plantNaam
▶	6	JENEVERBES
	17	LINDE
	18	GANZERIJK
	19	MUNT
	28	MONNIKSCHAP
	61	WINTERHEIDE
	74	VINGERHOEDSKRUID
	89	HONDSTONG
	96	HONDSTONG
	113	ZONNEBLOEM
	115	ZONNEBLOEM
*	NULL	NULL

#### 4.4.22. Opgave 22

Welke 1- en 2-jarige planten staan er in de table *planten*?

Probeer deze vraag met en zonder de operator LIKE op te lossen.

Resultaat: 23 records

Begin van de lijst:

	plantId	plantNaam	soort	kleur	hoogte	beginBloeimaand	eindBloeimaand	prijs	categorieId
▶	1	BEGONIA	1-JARIG	ROOD	15	6	9	0.65	1
	8	RIDDERSPOOR	1-JARIG	GEMENGD	50	7	8	1	2
	20	KORENBLOEM	1-JARIG	GEMENGD	80	7	8	1	3
	24	VIOLIER	1-JARIG	GEMENGD	60	6	8	1	2
	34	MUURBLOEM	2-JARIG	BRUIN	50	4	5	1.8	2
	49	LEEUWEBEKJE	1-JARIG	GEMENGD	50	7	8	0.8	2
	52	AFRIKAANTJE	1-JARIG	GEEL	25	7	10	0.6	2
	57	CHRYSAANT	1-JARIG	GEEL	80	6	8	0.8	3
	63	KLOKJESBLOEM	2-JARIG	BLAUW	70	6	8	1.8	3
	81	GIPSKRUID	1-JARIG	WIT	50	6	7	1	2
	83	JUDASPENNING	2-JARIG	LILA	70	5	7	1	3
	87	PETUNIA	1-JARIG	ROZE	25	7	10	0.8	2
	88	ALYSSUM	1-JARIG	PAARS	10	6	9	0.6	1
	89	HONDSTONG	2-JARIG	BLAUW	30	5	6	1	2

#### 4.4.23. Opgave 23

Geef een overzicht van alle planten, behalve de bomen en de heesters, die tussen de 100 en 200 cm hoog zijn, rode of blauwe bloemen geven, en vóór augustus beginnen te bloeien.

Alle gegevens behalve de prijs zijn belangrijk.

Sorteer de lijst op soort en binnen soort op plantennaam.

*Resultaat:* 4 records

	plantId	plantNaam	soort	kleur	hoogte	beginBloeimaand	eindBloeimaand
▶	36	IRIS	BOL	BLAUW	100	5	7
	33	ROZEMARIJN	KRUID	BLAUW	150	5	5
	62	KOGELDISTEL	VAST	BLAUW	175	6	7
	13	VUURPIJL	VAST	ROOD	120	6	9
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#### 4.4.24. Opgave 24

Bij welke planten, die niet behoren tot de soort kruid, komt het woord KRUID voor in hun plantennaam?

Geef plantId en plantennaam.

*Resultaat:* 6 records

	plantId	plantNaam
▶	30	NAGELKRUID
	41	BLAASJESKRUID
	73	GIPSKRUID
	74	VINGERHOEDSKRUID
	81	GIPSKRUID
	85	LEVERKRUID
*	NULL	NULL

#### 4.4.25. Opgave 25

Geef plantId en plantennaam van alle planten die beginnen met de letter L en eindigen met de letter E.

*Resultaat:* 4 records

	plantId	plantNaam
▶	5	LUPINE
	7	LISDODDE
	17	LINDE
	49	LEEUWEBEKJE
*	NULL	NULL

#### 4.4.26. Opgave 26

Welke planten hebben een plantennaam van precies 5 letters lang? Geef plantId en plantennaam.

*Resultaat:* 7 records

	plantId	plantNaam
▶	17	LINDE
	70	HULST
	92	SALIE
	100	PIOEN
	110	ASTER
	111	DILLE
	117	ANJER
★	HULL	HULL

#### 4.4.27. Opgave 27

Welke planten hebben een plantennaam van minimum 5 letters lang? Geef plantId en plantennaam.

*Resultaat:* 111 records

*Begin van de lijst:*

	plantId	plantNaam
▶	1	BEGONIA
	2	KLIMOP
	3	MARGRIET
	4	WATERLELIE
	5	LUPINE
	6	JENEVERBES
	7	LISDODDE
	8	RIDDERSPoor
	9	OLIJFWILG
	10	GOUDEN REGEN
	11	DWERCYPRES
	12	WATERHYACINT

### 4.5. Aggregate functions

In dit onderdeel bekijken we hoe je minima, gemiddelde,... kan berekenen.

#### 4.5.1. Theorie

SQL kent verschillende functies die gebruikt kunnen worden om de data in de database te manipuleren.

Je kan aan een veldnaam een **aggregate function** toevoegen. Deze functies voeren een berekening uit met alle waarden van de veldnaam van alle geselecteerde records. Lege waarden (Null) worden hierbij niet meegenomen.

In deze cursus bespreken we de volgende aggregate functions:

- COUNT - aantal,
- SUM - totaal,
- AVG - gemiddelde,
- MAX - grootste waarde,
- MIN - kleinste waarde.

Het **aanroepen** van functies in SQL lijkt op het aanroepen van functies in programmeertalen zoals C++, PHP en JavaScript. Eerst komt de functienaam, gevolgd door een haakje openen, parameters, haakje sluiten, ofwel:

```
functie(parameters)
```

Vaak gaan we het gebruik van de functie combineren met een **alias**. Door deze alias zal de kolom toch een naam krijgen als we het resultaat van de instructie bekijken.

```
SELECT AVG(leeftijd) AS gem
FROM leden ;
```

berekent de gemiddelde leeftijd van alle leden. De kolom krijgt als naam *gem*.

#### 4.5.2. Voorbeelden

Hieronder vind je twee voorbeelden die je kan uittesten in de database BIEREN.

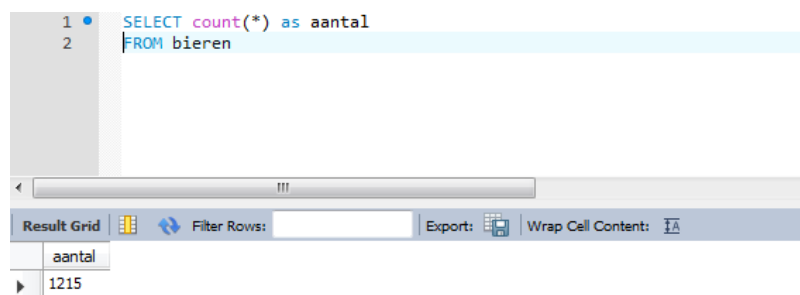
##### Voorbeeld 1

Je wil het totaal aantal bieren kennen.

*Oplossing:*

```
SELECT count(*) as aantal FROM bieren
```

*Resultaat:*



aantal
1215

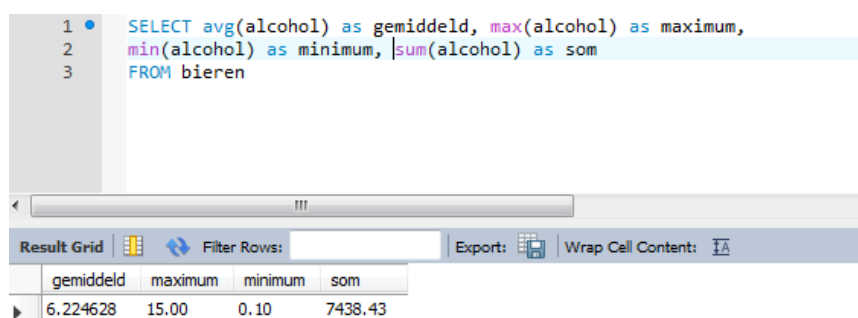
##### Voorbeeld 2

Je wil het gemiddelde, het grootste, het kleinste en het totale alcoholpercentage uit de table *bieren*.

*Oplossing:*

```
SELECT avg(alcohol) as gemiddeld, max(alcohol) as maximum, min(alcohol) as minimum,
sum(alcohol) as som
FROM bieren
```

*Resultaat:*



gemiddeld	maximum	minimum	som
6.224628	15.00	0.10	7438.43

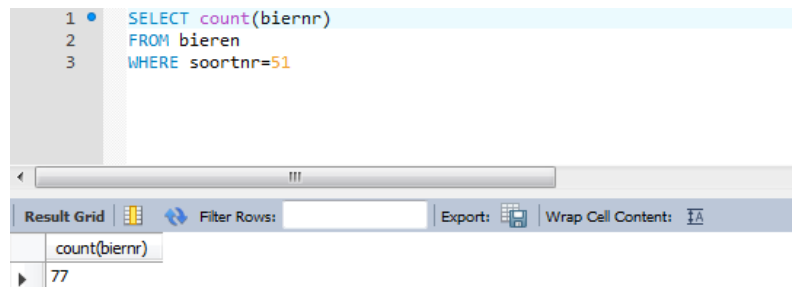
### Voorbeeld 3

Je wil het aantal tafelbieren (soortnr 51) kennen.

#### Oplossing

```
SELECT count(biernr) FROM bieren WHERE soortnr=51
```

#### Resultaat



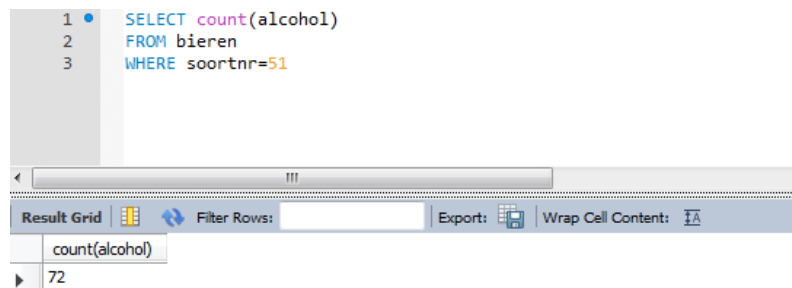
The screenshot shows a SQL query editor with the following query:

```
1 SELECT count(biernr)
2 FROM bieren
3 WHERE soortnr=51
```

Below the query editor, the 'Result Grid' is displayed with the following data:

count(biernr)
77

Als je op het veld alcohol telt, krijg je dit resultaat:



The screenshot shows a SQL query editor with the following query:

```
1 SELECT count(alcohol)
2 FROM bieren
3 WHERE soortnr=51
```

Below the query editor, the 'Result Grid' is displayed with the following data:

count(alcohol)
72

#### Van waar komt dit verschil?

De aggregate functies houden geen rekening met de waarde Null.

Er zijn 5 bieren met als alcoholpercentage Null. Deze worden niet meegenomen als je telt op het veld alcohol. Als je telt op het veld biernr worden deze wel meegenomen.

Daarom dat we in praktijk meestal count(\*) gebruiken. Zo ben je zeker dat alle rijen geteld worden.

### 4.5.3. Opdracht

Maak de opdrachten in de cursus.

## 4.6. Berekeningen

Naast de 5 aggregate functies kan je ook berekeningen laten uitvoeren.

In dit onderdeel leggen we uit hoe je dat moet doen.

### 4.6.1. Theorie

In je query kan je onmiddellijk **berekeningen** laten uitvoeren. Je kan hiervoor slechts **4 operators** gebruiken:

- optellen +
- aftrekken -
- delen /
- vermenigvuldigen \*

Rekenen kan uiteraard alleen met getallen en niet met tekst.

Ook hier gaan we vaak combineren met een **alias** om de kolom een naam te geven.

```
SELECT naam, brutoloon*0.6 AS nettoloon
FROM werknemers
```

berekent het nettoloon (60% van je brutoloon) van alle werknemers.

#### 4.6.2. Voorbeeld

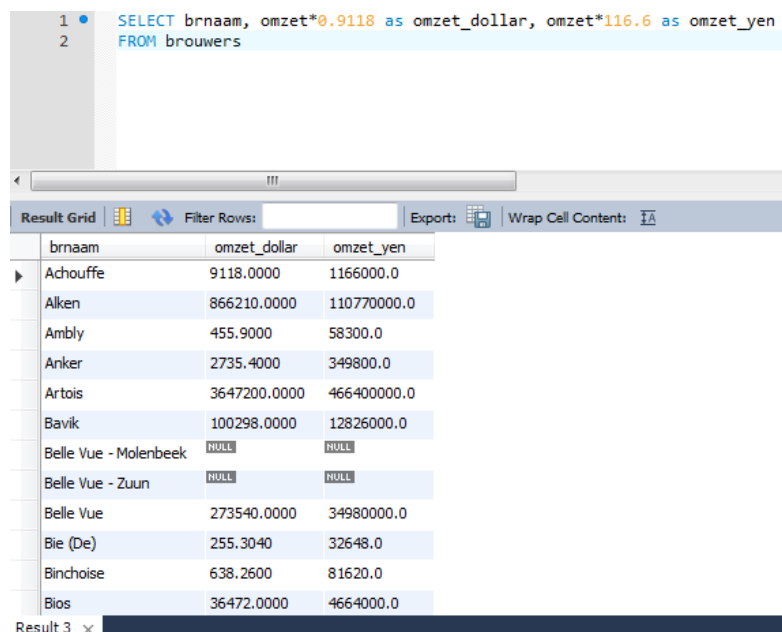
Volgend voorbeeld kan je weer uittesten in de database Bieren.

Maak een lijst van alle brouwerijen met hun omzet in dollar en in yen.

*Oplossing:*

```
SELECT brnaam, omzet*0.9118 as omzet_dollar, omzet*116.6 as omzet_yen
FROM brouwers
```

*Resultaat:* 118 records



The screenshot shows a SQL query editor with the following query:

```
1 SELECT brnaam, omzet*0.9118 as omzet_dollar, omzet*116.6 as omzet_yen
2 FROM brouwers
```

Below the query, a 'Result Grid' is displayed with the following data:

brnaam	omzet_dollar	omzet_yen
Achouffe	9118.0000	1166000.0
Alken	866210.0000	110770000.0
Ambly	455.9000	58300.0
Anker	2735.4000	349800.0
Artois	3647200.0000	466400000.0
Bavik	100298.0000	12826000.0
Belle Vue - Molenbeek	NULL	NULL
Belle Vue - Zuun	NULL	NULL
Belle Vue	273540.0000	34980000.0
Bie (De)	255.3040	32648.0
Binchoise	638.2600	81620.0
Bios	36472.0000	4664000.0

#### 4.7. Opgaven

In dit onderdeel maken we een reeks oefeningen op:

- aggregate functies
- berekeningen

Alle oefeningen kan je maken in de database PLANTEN. De oplossingen vind je in de webversie.



#### 4.7.1.Opgave 1

Hoeveel leveranciers telt ons tuincentrum?

Resultaat:

	aantal
▶	11

#### 4.7.2.Opgave 2

Geef de gemiddelde prijs van alle waterplanten.

Resultaat:

	gemiddelde
▶	4.75

#### 4.7.3.Opgave 3

Geef de gemiddelde, de laagste en de hoogste offerteprijs van leverancier 4.

Resultaat:

	gemiddelde	laagste	hoogste
▶	3.7564516129032266	0.1	12.7

#### 4.7.4.Opgave 4

Wat is de maximale hoogte van de bomen in de table *planten*?

Resultaat:

	maxhoogte
▶	4000

#### 4.7.5.Opgave 5

Wat is de laagste offerteprijs voor artikel 1?

Resultaat:

	minimum
▶	0.35

#### 4.7.6.Opgave 6

Geef bestelnummer, artikelLeverancierId en het totale bestelbedrag per bestelrij uit de table *bestellijnen*.

Resultaat: 93 records

Begin van de lijst:

	bestelId	artikelLeverancierId	totaal
▶	1	50	67.5
	1	56	142.5
	1	59	56.25
	1	60	82.5
	1	70	95
	1	75	158.75
	2	6	67.5
	2	7	16.25
	2	8	17.5
	2	10	16.25
	3	1	37.5
	3	2	62.5
	3	3	80

#### 4.7.7. Opgave 7

Maak een overzicht van de heesters uit de table *planten*, waarbij je de prijzen met 5% verhoogt.

*Resultaat:* 14 records

*Begin van de lijst:*

	plantId	plantNaam	verhoogd
▶	9	OLIJFWILG	10.5
	15	ZUURBES	3.6750000000000003
	31	KORNOELJE	5.775
	32	PEPERBOOMPJE	15.75
	37	VUURDOORN	5.25
	38	SERING	20.475
	39	RODODENDRON	20.475
	54	MAGNOLIA	15.225000000000001
	56	LIGUSTER	0.42000000000000004
	64	BREM	5.25
	70	HULST	23.625
	86	FORSYTHIA	5.775
	107	AZALEA	18.375
	112	TOEPENHARTIG	33.6

#### 4.7.8. Opgave 8

Hoeveel stuks van leveranciersartikelnummer 59 zijn er besteld?

*Resultaat:*

	aantal
▶	75

#### 4.7.9. Opgave 9

Wat is het totale bestelbedrag (exclusief korting) voor leveranciersartikelnummer 8?

*Resultaat:*

	totaal
▶	47.75

## 4.8. Groeperen

In dit onderdeel bekijken we hoe je gegevens kan groeperen.

### 4.8.1. Syntax

Met behulp van **GROUP BY** en aggregate functies kan je berekeningen maken voor groepen records i.p.v. voor alle records samen. Een groep bestaat uit alle records met gelijke waarden voor alle velden (of uitdrukkingen) in de GROUP BY.

GROUP BY is niet verplicht in een select statement.

**HAVING** biedt je de mogelijkheid om een filter te plaatsen op basis van de berekeningen. Alleen de rijen die aan de voorwaarde voldoen, worden in het eindresultaat getoond.

Als je geen ORDER BY gebruikt, wordt het resultaat gesorteerd volgens de kolommen bij GROUP BY.

De (volledige) **syntaxis** is:

```
SELECT [predikaat] { * | table.* | [table.]veld1 [AS alias1] [, [table.]veld2 [AS alias2] [, ...]] }
FROM tableexpressie [, ...]
[WHERE ...]
[GROUP BY veld [, ...]]
[HAVING ...]
[ORDER BY attribuut asc|desc [, attribuut asc|desc [,...]] ]
```

### 4.8.2. Regel

Er is een belangrijke **regel** voor het gebruik van GROUP BY:

Bij het gebruik van group by moeten alle kolommen die in de select clause voorkomen en geen onderdeel zijn van een aggregate functie, opgenomen zijn in de group by clause

Voorbeeld:

```
SELECT artikelnummer, artikelLeverancierCode, avg(offertePrijs)
FROM offertes
GROUP BY artikelnummer
```

Dit is geen goede SQL-instructie: Het veld artikelLeverancierCode wordt vernoemd zonder aggregate functie en het staat niet bij de GROUP BY.

### 4.8.3. Voorbeelden

Hieronder krijg je 4 voorbeelden. Test ze allemaal uit in de database BIEREN.

#### *Voorbeeld 1*

Bereken het gemiddelde alcoholpercentage per brouwer.

*Oplossing:*

```
SELECT brouwernr, avg(alcohol) as gem
FROM bieren
GROUP BY brouwernr
```

*Resultaat:* 113 records

```

1 SELECT brouwnr, avg(alcohol) as gem
2 FROM bieren
3 GROUP BY brouwnr

```

brouwnr	gem
1	8.295556
2	3.011111
3	7.000000
4	7.050000
6	4.788235
8	4.742857
9	5.033333
10	5.000000
11	4.925000
12	7.375000
13	7.804167
14	7.467347

Result 1 x

### Voorbeeld 2

Bereken het aantal bieren per biersoort.

*Oplossing:*

```

SELECT soortnr, count(*) as aantal
FROM bieren
GROUP BY soortnr

```

*Resultaat:* 38 records

```

1 SELECT soortnr, count(*) as aantal
2 FROM bieren
3 GROUP BY soortnr
4

```

soortnr	aantal
2	5
3	6
4	5
5	2
6	28
8	8
11	9
12	3
13	4
14	139
15	193

Result 2 x

### Voorbeeld 3

Bepaal het minimum alcoholpercentage per brouwnr. De lijst toont enkel de brouwnr's en percentages die kleiner zijn dan 5%.

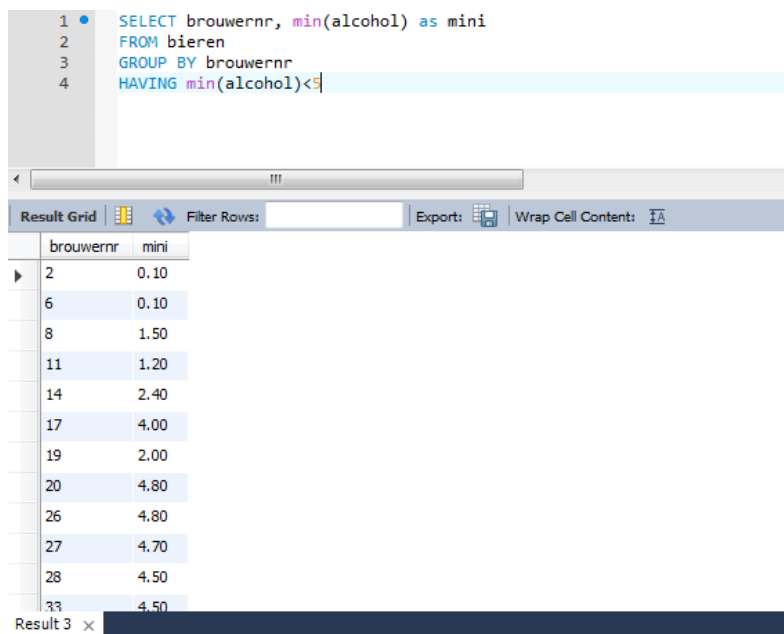
Om dit op te lossen gebruiken we **HAVING** i.p.v. **WHERE**.

Je gebruikt having als de selectie gebaseerd is op het resultaat van een bewerking met een aggregate functie. In alle andere gevallen gebruik je **WHERE**.

*Oplossing:*

```
SELECT brouwnr, min(alcohol) as mini
FROM bieren
GROUP BY brouwnr
HAVING min(alcohol)<5
```

*Resultaat:* 52 records



brouwnr	mini
2	0.10
6	0.10
8	1.50
11	1.20
14	2.40
17	4.00
19	2.00
20	4.80
26	4.80
27	4.70
28	4.50
33	4.50

#### Voorbeeld 4

Bereken het gemiddelde alcoholpercentage per brouwnr voor alle brouwers die meer dan 10 bieren produceren.

*Oplossing:*

```
SELECT brouwnr, avg(alcohol) as gem
FROM bieren
GROUP BY brouwnr
HAVING count(*)>10
```

*Resultaat:* 41 records

```

1 SELECT brouwer nr, avg(alcohol) as gem
2 FROM bieren
3 GROUP BY brouwer nr
4 HAVING count(*)>10

```

brouwer nr	gem
6	4.788235
8	4.742857
11	4.925000
13	7.804167
14	7.467347
17	6.981132
19	5.454545
21	6.538462
37	7.856667
43	5.264286
44	7.615789
51	4.418333

Result 4 x

#### 4.8.4. Opdracht

Maak de opdracht in de webversie van deze cursus.

### 4.9. Opgaven 3

In dit onderdeel krijg je een hele reeks oefeningen op het groeperen van gegevens.

Al deze oefeningen maak je in de database PLANTEN. Ook nu vind je de oplossingen in de webversie van deze cursus.

#### 4.9.1. Opgave 1

Tel het aantal planten per plantensoort uit de table *planten*.

*Resultaat:* 10 records

	soort	aantal
▶	1-JARIG	15
	2-JARIG	8
	BOL	5
	BOOM	17
	HEESTER	14
	HEIDE	4
	KLIM	4
	KRUID	12
	VAST	31
	WATER	9

#### 4.9.2. Opgave 2

Tel het aantal bestelregels per bestelling.

*Resultaat:* 15 records

*Begin van de lijst:*

	bestelId	aantal
▶	1	6
	2	4
	3	11
	4	5
	5	1
	6	3
	7	8
	8	16
	9	4
	10	6
	11	4
	12	8

### 4.9.3. Opgave 3

Wat is de gemiddelde prijs per plantensoort?

*Resultaat:* 10 records

	soort	gemiddelde
▶	1-JARIG	0.9533333333333334
	2-JARIG	1.2625
	BOL	0.9019999999999999
	BOOM	15.411764705882353
	HEESTER	12.528571428571428
	HEIDE	2.75
	KLIM	8.375
	KRUID	2.0416666666666665
	VAST	3.056451612903226
	WATER	4.75

### 4.9.4. Opgave 4

Hoeveel planten zijn er per plantensoort-kleurgroep?

*Resultaat:* 54 records

*Begin van de lijst:*

	soort	kleur	aantal
▶	1-JARIG	GEEL	3
	1-JARIG	GEMENGD	8
	1-JARIG	PAARS	1
	1-JARIG	ROOD	1
	1-JARIG	ROZE	1
	1-JARIG	WIT	1
	2-JARIG	BLAUW	2
	2-JARIG	BRUIN	1
	2-JARIG	GEMENGD	4
	2-JARIG	LILA	1
	BOL	BLAUW	3
	BOL	GEEL	1
	BOL	WIT	1

#### 4.9.5. Opgave 5

Maak een overzicht zodat duidelijk is welke kleur van de vaste planten de hoogste gemiddelde prijs heeft.

*Resultaat:* 10 records

	kleur	gemiddelde
▶	WIT	4.142857142857143
	ROZE	3.5
	ORANJE	3
	LILA	3
	VIOLET	3
	ROOD	2.892857142857143
	BLAUW	2.666666666666665
	PAARS	2.5
	GEMENGD	2.25
	GEEL	2.166666666666665

#### 4.9.6. Opgave 6

Laat per leverancierId het aantal artikelen zien dat de leverancier aanbiedt onder voorwaarde dat de levertijd van het artikel minder dan 18 dagen is.

*Resultaat:* 10 records

	leverancierId	aantal
▶	1	11
	3	11
	4	31
	5	30
	6	22
	7	15
	8	14
	9	11
	10	31
	11	18

#### 4.9.7. Opgave 7

Wat is de gemiddelde prijs per plantensoort, exclusief de geelbloemige planten?

*Resultaat:* 10 records

	soort	gemiddelde
▶	1-JARIG	0.9249999999999999
	2-JARIG	1.2625
	BOL	1.0275
	BOOM	12.833333333333334
	HEESTER	11.8625
	HEIDE	2.75
	KLIM	8
	KRUID	2.05
	VAST	3.1517857142857144
	WATER	5.4375



#### 4.9.8. Opgave 8

Maak een overzicht met de laagste en de hoogste offerteprijs per plant.

*Resultaat:* 119 records

*Begin van de lijst:*

	plantId	mini	maxi
▶	1	0.35	0.4
	2	4.65	4.95
	3	1.45	1.55
	4	7.9	7.9
	5	1.5	1.65
	6	2.95	2.95
	7	2.95	2.95
	8	0.65	0.65
	9	4.5	6.5
	10	9.9	9.9
	11	6.1	6.1

#### 4.9.9. Opgave 9

Wat is de gemiddelde prijs per plantensoort voor soorten met minstens 10 exemplaren in de table *planten*?

*Resultaat:* 5 records

	soort	gemiddelde
▶	1-JARIG	0.9533333333333334
	BOOM	15.411764705882353
	HEESTER	12.528571428571428
	KRUID	2.0416666666666665
	VAST	3.056451612903226

#### 4.9.10. Opgave 10

Hebben de planten met korte levertijden in het algemeen een hogere gemiddelde offerteprijs?

*Resultaat:* 4 records

	levertijd	gemiddelde
▶	7	0.6516129032258065
	10	2.567682926829269
	14	3.8944444444444457
	21	6.137500000000002

#### 4.9.11. Opgave 11

Maak een overzicht met de laagste en de hoogste bestelprijs per artikelLeverancierId.

*Resultaat:* 70 records

Begin van de lijst:

	artikelLeverancierId	laagste	hoogste
▶	1	0.75	1.25
	2	0.25	0.25
	3	0.2	0.2
	4	0.45	0.7
	5	0.4	0.7
	6	0.8	1.45
	7	0.45	0.7
	8	0.45	0.75
	9	0.7	0.7
	10	0.45	0.75

#### 4.9.12. Opgave 12

Geef een overzicht van het aantal beschikbare planten per beginBloeimaand/hoogte/kleur groep?

Resultaat: 109 records

Begin van de lijst:

	beginBloeimaand	hoogte	kleur	aantal
▶	0	0	NULL	1
	0	0	WIT	1
	0	20	ROOD	1
	0	25	NULL	1
	0	30	WIT	1
	0	90	NULL	1
	0	100	NULL	1
	0	250	NULL	1
	0	500	NULL	1
	0	600	NULL	1
	0	700	NULL	1

#### 4.9.13. Opgave 13

Wat is de laagste prijs per plantensoort van de planten die in ieder geval bloeien in de periode mei tot en met juni?

Resultaat: 5 records

	soort	minimum
▶	2-JARIG	0.5
	BOL	0.12
	HEESTER	3.5
	VAST	2.5
	WATER	3.5

#### 4.9.14. Opgave 14

Tel het aantal planten per prijs. De prijs moet je afronden op een geheel getal.

Resultaat: 17 records

	afgeronde_prijs	aantal
0		6
1		18
2		34
3		11
4		13
5		4
6		5
8		3
10		6
12		3
14		2
15		1
18		4
20		4
22		3
32		1
42		1

#### 4.10. Opdracht voor de coach 1

Hieronder krijg je een reeks opdrachten. Gebruik hiervoor de database VIDEO. Je kan deze in de webcursus downloaden.

Verzamel al je oplossingen in een tekstbestand en stuur dit door ter verbetering aan je **coach**. Vermeld als onderwerp "**Gegevens selecteren**".

Stel de SQL-instructies op om volgende opdrachten uit te voeren. Gebruik hiervoor de database Video.

1. Geef een lijst van de klanten uit Gent of Wetteren met de velden naam, voornaam, postcode, woonplaats en klantstatus. Sorteer op klantstatus en naam.
2. Maak een lijst van de klanten waarvan de postcode groter of gelijk is aan 9000 en de klanten waarvan het huuraantal groter is dan 200. Deze lijst moet gesorteerd worden op postcode
3. Geef een lijst van de klanten wiens naam niet begint met een 'd'.
4. Geef een lijst van klanten waar in de naam van de gemeente op de derde plaats een 'n' staat.
5. Bereken voor alle films de prijs incl BTW (21%).
6. Uit welke woonplaatsen komen onze klanten?
7. Maak een lijst van het aantal klanten per gemeente.
8. Maak een lijst die het mogelijk maakt een inzicht te krijgen in de verhuringen per gemeente. Op deze lijst moet de gemeente/stad met de meeste verhuringen bovenaan staan. Indien in een bepaalde gemeente geen 200 verhuringen gebeurd zijn, mag deze gemeente niet op de lijst staan.

Verzamel al je oplossingen in een tekstbestand en stuur dit door ter verbetering aan je coach. Vermeld als onderwerp "Gegevens selecteren".

## Hoofdstuk 5. Meerdere tables

In praktijk komt het vaak voor dat je gegevens nodig hebt uit meer dan 1 table. Hiervoor zul je query's moeten maken die tables onderling met elkaar verbinden en hieruit de juiste gegevens selecteren. Dit doen met een zogenaamde **join**.

Er zijn verschillende type joins. In deze cursus behandelen we de belangrijkste types:

- inner join
- outer join
- self join

### 5.1. Inner join

#### 5.1.1. Syntax

Met een **inner join** kan je records uit twee of meer tables combineren op basis van een voorwaarde met velden uit verschillende tables. Records uit één van beide tables die geen corresponderend record hebben in de andere table worden niet getoond.

Je kan deze toepassen op elke FROM-component. Het is de meest gebruikte type koppeling.

#### *Twee tables - 1 veld*

Om gegevens uit twee tables te selecteren, gebruik je volgende syntax:

```
SELECT velden
FROM table1 INNER JOIN table2
ON table1.veld1 operator table2.veld2
```

Deze instructie bevat volgende onderdelen:

Onderdeel	Betekenis
<b>table1, table2</b>	De namen van de tables waaruit records gecombineerd worden.
<b>veld1, veld2</b>	De namen van de velden die gekoppeld worden. Beide velden moeten hetzelfde gegevenstype hebben en dezelfde soort gegevens bevatten. Ze moeten niet dezelfde naam hebben.  Als de velden in de twee tables dezelfde naam hebben, maak je een onderscheid door de tablenaam ervoor te plaatsen:  table.veld
<b>operator</b>	Een willekeurige relationele vergelijkingsoperator:  = < > <= >= <>

#### *Twee tables - meerdere velden*

Je kan een join baseren op meerdere velden door de verschillende vergelijkingen te combineren met een logische operator.

```
SELECT velden
FROM table1 INNER JOIN table2
ON table1.veld1 operator table2.veld1 AND
   (table1.veld2 operator table2.veld2 OR
    table1.veld3 operator table2.veld3)
```

Deze syntax heb je nodig wanneer je twee tables wilt koppelen met een **samengestelde sleutel** (dit is een sleutel gebaseerd op de meerdere velden).

### *Meerdere tables*

Het is ook mogelijk om meerdere tables met join te nesten:

```
SELECT velden
FROM table1
INNER JOIN table2 ON table1.veld1 operator table2.veld2
INNER JOIN table3 ON table2.veld2 operator table3.veld3
[INNER JOIN tablex ON table3.veld3 operator tablex.veldx [...]]
```

Bekijk ook de [animatie](#) op YouTube.

### **5.1.2. Voorbeelden**

Hieronder vind je weer twee voorbeelden die je kan uitproberen in de database BIEREN.

#### *Voorbeeld 1*

We willen een lijst van alle bieren met daarbij de naam van de brouwerij.

We moeten de tables *bieren* en *brouwers* koppelen via het veld *brouwnr*. Hierdoor kunnen we de naam van het bier en de naam van de bijhorende brouwer tonen.

*Oplossing:*

```
SELECT naam, brnaam
FROM bieren INNER JOIN brouwers
ON bieren.brouwnr = brouwers.brouwnr
```

*Resultaat:* 1215 records

```

1 SELECT naam, brnaam
2 FROM bieren INNER JOIN brouwers
3 ON bieren.brouwernr = brouwers.brouwernr

```

naam	brnaam
A.C.O.	Steedje
Aalbeeks St. Corneliusbier (=Kapittel pater (Het))	Van Eecke
Aardbeien witbier	Huyghe
Aarschots kruikenbier (=St. Sebastiaan grand cru)	Sterkens
Abt Bijbier (Nen)	Domus
Adler	Haacht
Aerts 1900	Palm
Affligem blond (Abdij)	Smedt (De)
Affligem christmas ale (Abdij)	Smedt (De)
Affligem dubbel (Abdij)	Smedt (De)
Affligem patersvat	Smedt (De)

Result 1 x

### Voorbeeld 2

Geef een lijst van alle bieren met de velden naam, brouwerij en soort.

*Oplossing:*

```

SELECT naam, brnaam, soort
FROM brouwers
INNER JOIN bieren ON bieren.brouwernr=brouwers.brouwernr
INNER JOIN soorten ON bieren.soortnr=soorten.soortnr

```

*OF*

```

SELECT naam, brnaam, soort
FROM soorten
INNER JOIN bieren ON soorten.soortnr = bieren.soortnr
INNER JOIN brouwers ON bieren.brouwernr = brouwers.brouwernr

```

*Resultaat:* 1215 records

```

1 • SELECT naam, brnaam, soort
2 FROM brouwers
3 INNER JOIN bieren ON bieren.brouwerNr=brouwers.brouwerNr
4 INNER JOIN soorten ON bieren.soortNr=soorten.soortNr

```

naam	brnaam	soort
A.C.O.	Steedje	Extra
Aalbeeks St. Corneliusbier (=Kapittel pater (Het))	Van Eecke	Extra
Aardbeien witbier	Huyghe	Tarwebier of witbier
Aarschots kruikenbier (=St. Sebastiaan grand cru)	Sterkens	Edelbier
Abt Bijbier (Nen)	Domus	Extra
Adler	Haacht	Pils
Aerts 1900	Palm	Dubbel Donker
Affligem blond (Abdij)	Smedt (De)	Lichtblond
Affligem christmas ale (Abdij)	Smedt (De)	Massieve Ale
Affligem dubbel (Abdij)	Smedt (De)	Dubbel Donker
Affligem patersvat	Smedt (De)	Lichtblond

Result 2 x

## 5.2. Outer join

### 5.2.1.Syntax

Ook met een **outer join** kan je records uit twee of meer tables combineren op basis van gelijke waarden in een gemeenschappelijk veld.

Om gegevens uit twee tables te selecteren, gebruik je volgende syntax:

```

SELECT velden
FROM table1 {LEFT | RIGHT} [OUTER] JOIN table2
ON table1.veld1 operator table2.veld2

```

Deze instructie bevat volgende onderdelen:

Onderdeel	Betekenis
<b>table1, table2</b>	De namen van de tables waaruit records gecombineerd worden.
<b>veld1, veld2</b>	De namen van de velden die gekoppeld worden. Beide velden moeten hetzelfde gegevenstype hebben en dezelfde soort gegevens bevatten. Ze moeten niet dezelfde naam hebben.  Als de velden in de twee tables dezelfde naam hebben, maak je een onderscheid door de tablenaam ervoor te plaatsen:  table.veld
<b>operator</b>	Een willekeurige relationele vergelijkingsoperator:  = < > <= >= <>

### 5.2.2. Left versus Right

Met een bewerking LEFT JOIN maak je een **left outer join**. In een left outer join worden alle records uit de eerste (linker-) table opgenomen, zelfs als er geen overeenkomende waarden in de tweede (rechter-) table zijn.

Met een bewerking RIGHT JOIN maak je een **right outer join**. In een right outer join worden alle records uit de tweede (rechter-) table opgenomen, zelfs als er geen overeenkomende waarden in de eerste (linker-) table zijn.

Het woord OUTER is niet verplicht in de instructie.

Voorbeeld:

Vertrek van de tables afdelingen (linkertable) en werknemers (rechttable)

- Met left join kan je alle afdelingen selecteren, inclusief afdelingen zonder personeelsleden. Bij de afdelingen zonder personeelsleden krijgen de velden van de personeelsleden de waarde Null in het resultaat.
- Als je alle personeelsleden wilt selecteren, dus met inbegrip van personeelsleden die niet werkzaam zijn op een bepaalde afdeling, gebruikt je right join. Bij de personeelsleden zonder afdeling krijgen de velden van de afdeling de waarde Null in het resultaat.

Bekijk zeker ook de animaties op YouTube:

- [Left outer join](#)
- [Right outer join](#)

### 5.2.3. Voorbeelden

Hieronder vind je weer twee voorbeelden die je kan uitproberen in de database BIEREN.

#### *Voorbeeld 1*

In volgend voorbeeld worden de tables *soorten* en *bieren* gekoppeld op het veld soortnr. Deze instructie genereert een lijst van alle soorten, inclusief de soorten waarvan er geen bieren aanwezig zijn in de table *bieren*.

*Oplossing:*

```
SELECT naam, soort
FROM soorten LEFT JOIN bieren
ON soorten.soortnr = bieren.soortnr
```

*Resultaat:* 1215 records



```

1 SELECT naam, soort
2 FROM soorten LEFT JOIN bieren
3 ON soorten.soortnr = bieren.soortnr

```

naam	soort
A.C.O.	Extra
Aalbeeks St. Corneliusbier (=Kapittel pater (Het))	Extra
Aardbeien witbier	Tarwebier of witbier
Aarschots kruikenbier (=St. Sebastiaan grand cru)	Edelbier
Abt Bijbier (Nen)	Extra
Adler	Pils
Aerts 1900	Dubbel Donker
Affligem blond (Abdij)	Lichtblond
Affligem christmas ale (Abdij)	Massieve Ale
Affligem dubbel (Abdij)	Dubbel Donker
Affligem patersvat	Lichtblond

Result 3

### Voorbeeld 2

In dit voorbeeld worden de tables *brouwers* en *bieren* gekoppeld. De select-instructie genereert een lijst van alle brouwers, inclusief de brouwerijen waarvan er geen bieren aanwezig zijn in de table *bieren*.

*Oplossing:*

```

SELECT brnaam, naam
FROM bieren RIGHT JOIN brouwers
ON bieren.brouwer nr = brouwers.brouwer nr
ORDER BY naam

```

*Resultaat:* 1220 records

```

1 SELECT brnaam, naam
2 FROM bieren RIGHT JOIN brouwers
3 ON bieren.brouwer nr = brouwers.brouwer nr
4 ORDER BY naam

```

brnaam	naam
Old Bailey	NULL
Pakhuis	NULL
Hopperd	NULL
Paeleman	NULL
Duwac	NULL
Steedje	A.C.O.
Van Eecke	Aalbeeks St. Corneliusbier (=Kapittel pater (Het))
Huyghe	Aardbeien witbier
Sterkens	Aarschots kruikenbier (=St. Sebastiaan grand cru)
Domus	Abt Bijbier (Nen)
Haacht	Adler

Result 4

## 5.2.4. Combinaties

### Left or right outer join?

```
SELECT brnaam, naam
FROM bieren
RIGHT JOIN brouwers ON bieren.brouwernr=brouwers.brouwernr
```

of

```
SELECT brnaam, naam
FROM brouwers
LEFT JOIN bieren ON bieren.brouwernr=brouwers.brouwernr
```

geven allebei hetzelfde resultaat (1220 records).

### Combineren met inner join

Je kan ook een outer join combineren met inner joins. Je plaatst de outer join als laatste in de reeks van joins.

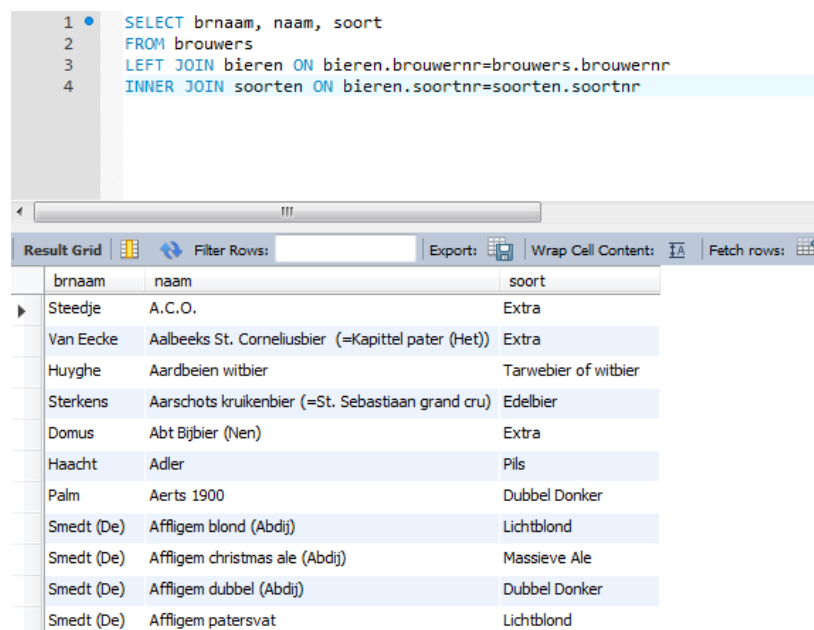
In onderstaande voorbeeld combineren we inner join en outer join. We baseren ons in beide voorbeelden op de tables *soorten*, *bieren* en *brouwers*.

In de eerste oplossing plaatsen we de outer join niet als laatste. In de tweede oplossing doen we dit wel. Je zal zien dat je een ander resultaat krijgt.

### Foute oplossing

```
SELECT brnaam, naam, soort
FROM brouwers
LEFT JOIN bieren ON bieren.brouwernr=brouwers.brouwernr
INNER JOIN soorten ON bieren.soortnr=soorten.soortnr
```

Resultaat: 1215 records



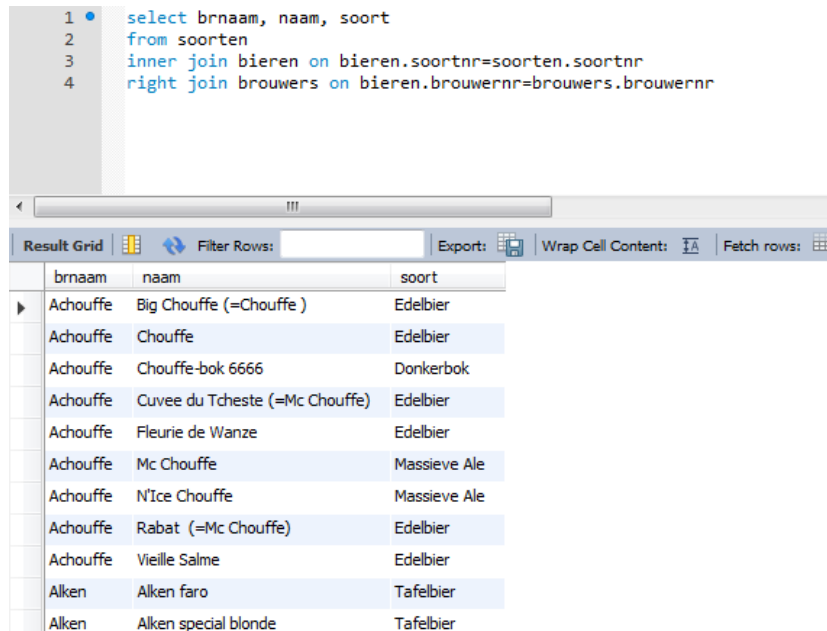
```
1 SELECT brnaam, naam, soort
2 FROM brouwers
3 LEFT JOIN bieren ON bieren.brouwernr=brouwers.brouwernr
4 INNER JOIN soorten ON bieren.soortnr=soorten.soortnr
```

	brnaam	naam	soort
▶	Steedje	A.C.O.	Extra
	Van Eecke	Aalbeeks St. Corneliusbier (=Kapittel pater (Het))	Extra
	Huyghe	Aardbeien witbier	Tarwebier of witbier
	Sterkens	Aarschots kruikenbier (=St. Sebastiaan grand cru)	Edelbier
	Domus	Abt Bijbier (Nen)	Extra
	Haacht	Adler	Pils
	Palm	Aerts 1900	Dubbel Donker
	Smedt (De)	Affligem blond (Abdij)	Lichtblond
	Smedt (De)	Affligem christmas ale (Abdij)	Massieve Ale
	Smedt (De)	Affligem dubbel (Abdij)	Dubbel Donker
	Smedt (De)	Affligem patersvat	Lichtblond

### Juiste oplossing

```
SELECT brnaam, naam, soort
FROM soorten
INNER JOIN bieren ON bieren.soortnr=soorten.soortnr
RIGHT JOIN brouwers ON bieren.brouwernr=brouwers.brouwernr
```

Resultaat: 1220 records



	brnaam	naam	soort
▶	Achouffe	Big Chouffe (=Chouffe )	Edelbier
	Achouffe	Chouffe	Edelbier
	Achouffe	Chouffe-bok 6666	Donkerbok
	Achouffe	Cuvee du Tcheste (=Mc Chouffe)	Edelbier
	Achouffe	Fleurie de Wanze	Edelbier
	Achouffe	Mc Chouffe	Massieve Ale
	Achouffe	N'Ice Chouffe	Massieve Ale
	Achouffe	Rabat (=Mc Chouffe)	Edelbier
	Achouffe	Vieille Salme	Edelbier
	Alken	Alken faro	Tafelbier
	Alken	Alken special blonde	Tafelbier

## 5.3. Self join

### 5.3.1.Syntax

Een laatste type dat we bekijken is een **self join**. Dit is een inner join of een outer join waarbij je twee keer dezelfde table gebruikt.

Om dit goed te laten verlopen moet je de tables een aliasnaam geven. Als je de tables een aliasnaam geeft, moet je bovendien in de select, on, group by en where clauses verwijzen naar de table met de aliasnaam i.p.v. van de tablenaam.

Syntax:

```
SELECT velden
FROM table as t1 {INNER | {RIGHT | LEFT} [OUTER]} JOIN table as t2
ON t1.veld operator t2.veld
```

De betekenis van de onderdelen is hetzelfde als bij de voorgaande syntaxen.

### 5.3.2. Voorbeeld

Ook hier weer een voorbeeld dat je kan uitproberen in de database BIEREN.

#### Voorbeeld 1

Maak een lijst van brouwers die in dezelfde gemeente wonen.

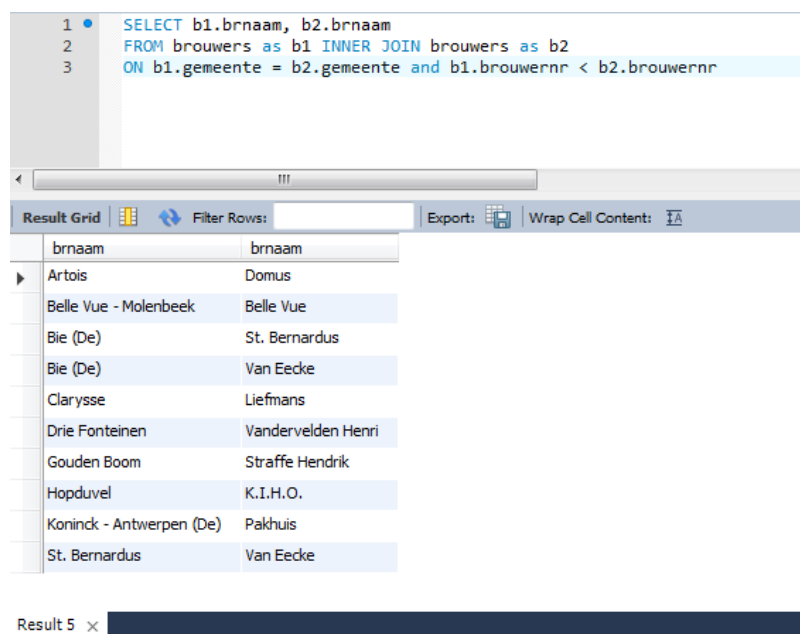
Aan de table *brouwers* wordt telkens een aliasnaam gegeven. Als je een kolomnaam gebruikt, ben je verplicht om die aliasnaam te gebruiken.

*Oplossing:*

```
SELECT b1.brnaam, b2.brnaam
FROM brouwers as b1 INNER JOIN brouwers as b2
ON b1.gemeente = b2.gemeente and b1.brouwernr < b2.brouwernr
```

$b1.brouwernr < b2.brouwernr$  zorgt ervoor dat de combinaties zoals Artois – Artois en dubbele records zoals Artois – Domus en Domus – Artois vermeden worden.

*Resultaat:* 10 records



The screenshot shows a SQL query editor with the following query:

```
1 SELECT b1.brnaam, b2.brnaam
2 FROM brouwers as b1 INNER JOIN brouwers as b2
3 ON b1.gemeente = b2.gemeente and b1.brouwernr < b2.brouwernr
```

Below the query editor is a 'Result Grid' showing 10 records. The grid has two columns: 'brnaam' and 'brnaam'. The records are as follows:

brnaam	brnaam
Artois	Domus
Belle Vue - Molenbeek	Belle Vue
Bie (De)	St. Bernardus
Bie (De)	Van Eecke
Clarysse	Liefmans
Drie Fonteinen	Vandervelden Henri
Gouden Boom	Straffe Hendrik
Hopduvel	K.I.H.O.
Koninck - Antwerpen (De)	Pakhuis
St. Bernardus	Van Eecke

At the bottom of the grid, it says 'Result 5' with a close button.

### 5.3.3. Opgaven

In dit onderdeel maken we een reeks oefeningen op het selecteren van gegevens uit meerdere tables.

Alle oefeningen kan je maken in de database PLANTEN. De oplossingen vind je in de webversie.

### 5.3.4. Opgave 1

Maak een overzicht van de bestellingen met het bestelnummer, de naam van de leverancier en de leverdatum.

*Resultaat:* 15 records

*Begin van de lijst:*

	bestelId	leverancierNaam	leveringsDatum
▶	1	SPITMAN EN ZN.	2016-01-31 00:00:00
	2	HOVENIER G.H.	2016-03-04 00:00:00
	3	HOVENIER G.H.	2016-03-06 00:00:00
	4	BAUMGARTEN R.	2016-03-27 00:00:00
	5	ERICA BV.	2016-03-16 00:00:00
	6	BLOEM L.Z.H.W.	2016-03-18 00:00:00
	7	DEZAAIER L.J.A.	2016-03-23 00:00:00
	8	SPITMAN EN ZN.	2016-03-27 00:00:00
	9	FLORA BV.	2016-03-23 00:00:00
	10	FLORA BV.	2016-03-23 00:00:00
	11	HOVENIER G.H.	2016-03-21 00:00:00
	12	HOVENIER G.H.	2016-04-02 00:00:00

### 5.3.5. Opgave 2

Welke planten hebben een levertijd van maximum 10 dagen?

Plaats in het overzicht de gegevens artikelcode, leveranciers-artikelcode, plantennaam en levertijd.

*Resultaat:* 113 records

*Begin van de lijst:*

	plantId	artikelLeverancierCode	plantNaam	levertijd
▶	115	A004	ZONNEBLOEM	7
	88	A075	ALYSSUM	7
	102	A103	VIOOLTJE	7
	108	A184	KLAPROOS	7
	24	A385	VIOLIER	7
	97	A421	SLEUTELBLOEM	7
	94	B148	KLAPROOS	7
	83	B331	JUDASPENNING	7
	34	B337	MUURBLOEM	7
	81	C274	GIPSKRUID	7
	87	D225	PFTI INTA	7

### 5.3.6. Opgave 3

Geef een overzicht waaruit blijkt bij welke leverancier(s) de planten besteld kunnen worden.

In het overzicht moeten de volgende gegevens staan:

plantid, plantennaam, leverancierscode, naam van de leverancier, levertijd en offerteprijs.

Sorteer het overzicht op plantennaam.

*Resultaat:* 218 records

*Begin van de lijst:*

	plantId	plantNaam	leverancierId	leverancierNaam	levertijd	offerteprijs
▶	48	ACACIA	4	SPITMAN EN ZN.	14	11.4
	52	AFRIKAANTJE	5	DEZAAIER L.J.A.	10	0.35
	52	AFRIKAANTJE	10	DE GROENE KAS BV.	14	0.35
	55	AKELEI	10	DE GROENE KAS BV.	14	1.55
	88	ALYSSUM	1	HOVENIER G.H.	7	0.35
	53	ANEMOON	10	DE GROENE KAS BV.	14	2.15
	53	ANEMOON	5	DEZAAIER L.J.A.	10	2
	117	ANJER	6	MOOIWEER FA.	14	1.8
	117	ANJER	5	DEZAAIER L.J.A.	10	1.7
	110	ASTER	5	DEZAAIER L.J.A.	10	0.45
	110	ASTER	7	BLOEM L.Z.H.W.	7	0.5
	107	ASTRILEA	8	TRABA	10	8.75

### 5.3.7. Opgave 4

Maak een overzicht van die planten die een prijs hebben welke minstens 50% boven de offerteprijs ligt.

Geef plantId, plantennaam, prijs van de plant, leveranciersId en offerteprijs.

Sorteer je lijst op plantennaam.

*Resultaat: 212 records*

*Begin van lijst:*

	plantId	plantNaam	prijs	leverancierId	offertePrijs
▶	48	ACACIA	17.5	4	11.4
	52	AFRIKAANTJE	0.6	5	0.35
	52	AFRIKAANTJE	0.6	10	0.35
	55	AKELEI	2.5	10	1.55
	88	ALYSSUM	0.6	1	0.35
	53	ANEMOON	3.5	10	2.15
	53	ANEMOON	3.5	5	2
	117	ANJER	3	5	1.7
	117	ANJER	3	6	1.8
	110	ASTER	0.75	7	0.5
	110	ASTER	0.75	5	0.45
	107	ASTRILEA	17.5	8	8.75

### 5.3.8. Opgave 5

Hoeveel verschillen de gehanteerde bestelprijzen met de huidige offerteprijzen?

Presenteer in de lijst de gegevens bestelnummer, artikelcode van de leverancier en het positieve of negatieve verschil.

*Resultaat: 93 records*

*Begin van de lijst:*

	bestelId	artikelLeverancierCode	verschil
▶	1	A154	-0.04999999999999999
	1	B006	-0.050000000000000044
	1	B111	-0.04999999999999982
	1	B396	0
	1	E098	-0.050000000000000044
	1	G202	-0.150000000000000036
	2	A421	0.150000000000000013
	2	B148	0.050000000000000044
	2	B331	0.09999999999999998
	2	C274	0.050000000000000044
	3	A004	-0.35000000000000001
	3	A075	0.00000000000000000

### 5.3.9. Opgave 6

In welke plaatsen kan het tuincentrum vaste planten bestellen?

Resultaat: 2 records

	woonplaats
▶	AALSMEER
	LISSE

### 5.3.10. Opgave 7

Maak een overzicht van alle rode planten geleverd door leveranciers die niet in Aalsmeer wonen. Vermeld de gegevens plantId, plantennaam, soort, naam van de leverancier, woonplaats. Sorteer op soort en plantennaam.

Resultaat: 10 records

	plantId	plantNaam	soort	leverancierNaam	woonplaats
▶	1	BEGONIA	1-JARIG	DEZAAIER L.J.A.	LISSE
	1	BEGONIA	1-JARIG	BLOEM L.Z.H.W.	HILLEGOM
	84	AZIJBLOOM	BOOM	BAUMGARTEN R.	HILLEGOM
	39	RODODENDRON	HEESTER	STRUJK BV.	LISSE
	39	RODODENDRON	HEESTER	TRA A.	LISSE
	39	RODODENDRON	HEESTER	BAUMGARTEN R.	HILLEGOM
	42	DOPHEIDE	HEIDE	STRUJK BV.	LISSE
	42	DOPHEIDE	HEIDE	ERICA BV.	HEEMSTEDE
	79	DAGLELIE	VAST	DEZAAIER L.J.A.	LISSE
	91	ENGELS GRAS	VAST	DEZAAIER L.J.A.	LISSE

### 5.3.11. Opgave 8

Bepaal voor iedere artikelcode de laagste offerteprijs.

Vermeld in het overzicht plantId, plantennaam en de bijbehorende offerteprijs.

Sorteer het overzicht op plantennaam.

Resultaat: 119 records

Begin van de lijst:

	plantId	plantNaam	minimum
▶	48	ACACIA	11.4
	52	AFRIKAANTJE	0.35
	55	AKELEI	1.55
	88	ALYSSUM	0.35
	53	ANEMOON	2
	117	ANJER	1.7
	110	ASTER	0.45
	107	AZALEA	7.9
	84	AZIJNBOOM	4.3
	59	BASILICUM	1.25
	1	BEGONIA	0.35

### 5.3.12. Opgave 9

Zoek de bestellingen welke een besteldatum hebben die gelijk is aan de leverdatum van één of meer andere bestellingen.

Geef bestelnummer 1<sup>e</sup> bestelling, besteldatum 1<sup>e</sup> bestelling, bestelnummer 2<sup>e</sup> bestelling, leveringsdatum 2<sup>e</sup> bestelling.

Resultaat: 2 records

	eerstenr	eerstedatum	tweedenr	tweeddatum
▶	4	2016-03-06 00:00:00	3	2016-03-06 00:00:00
	5	2016-03-06 00:00:00	3	2016-03-06 00:00:00

### 5.3.13. Opgave 10

Geef een overzicht van alle leveranciers en de bijbehorende bestellingen. Toon ook de leveranciers bij wie we nog niets besteld hebben.

We zijn enkel geïnteresseerd in de velden bestelId en leverancierNaam.

Resultaat: 17 records

	leverancierNaam	bestelId
▶	TRA A.	NULL
	STRUIK BV.	NULL
	SPITMAN EN ZN.	1
	HOVENIER G.H.	2
	HOVENIER G.H.	3
	BAUMGARTEN R.	4
	ERICA BV.	5
	BLOEM L.Z.H.W.	6



## Hoofdstuk 6. Extra mogelijkheden

### 6.1. Union

Soms is het nodig om resultaten van verschillende query's te combineren.

Dit kan je doen met **UNION**.

#### 6.1.1. Syntax

Met de optie **UNION** kan je in elke willekeurige combinatie de resultaten samenvoegen van twee of meer select-instructies.

Syntax:

```
query1 UNION [ALL] query2 [UNION [ALL] queryn [..]]
```

De union-bewerking bevat volgende onderdelen:

Onderdeel	Betekenis
<b>query1-n</b>	een select-instructie

Alle query's in een union-bewerking moeten hetzelfde **aantal velden** opvragen. Het is echter niet zo dat de velden ook dezelfde lengte of hetzelfde gegevenstype moeten hebben.

**Aliases** kan je alleen gebruiken in de eerste SELECT-component, omdat ze worden genegeerd in alle overige.

Je kan ook **sorteren**. In de ORDER BY-component verwijst je naar velden in de eerste SELECT-component. Je plaatst je ORDER BY ook helemaal op het einde.

#### 6.1.2. Voorbeelden

Probeer onderstaande voorbeelden uit in de database BIEREN.

Met de eerste drie voorbeelden willen we de werking van UNION tonen. In praktijk gebruik je dit beter niet op deze manier.

Voorbeeld 4 is wel een realistisch voorbeeld van het gebruik van UNION.

#### Voorbeeld 1

Selecteer alle bieren met soortnr 3 of 5

*Oplossing*

```
SELECT * FROM bieren WHERE soortnr=3
UNION
SELECT * FROM bieren WHERE soortnr=5
```

*Resultaat:* 8 records

```

1 • SELECT * FROM bieren WHERE soortnr=3
2   UNION
3   SELECT * FROM bieren WHERE soortnr=5

```

BierNr	Naam	BrouwerNr	SoortNr	Alcohol
428	Dommelsch malt	6	3	0.10
458	Edel-Brau	107	3	0.10
730	Jupiler NA	6	3	0.10
995	Palm green	81	3	0.10
1313	Stella Artois NA	6	3	0.10
1377	Tourtel malt	2	3	0.10
1188	Sezoens quattro	70	5	7.00
1475	Vleteren alt	30	5	8.00
*	NULL	NULL	NULL	NULL

bieren 1 x

### In praktijk

gebruik je de instructie

```
SELECT * FROM bieren WHERE soortnr in (3,5)
```

### Voorbeeld 2

Instructie

```

SELECT * FROM bieren
UNION
SELECT * FROM bieren

```

**Resultaat:** 1215 records

Alle gegevens van de table *bieren* worden slechts éénmaal getoond (1215 records). Standaard worden geen dubbele records getoond.

```

1 • SELECT * FROM bieren
2   UNION
3   SELECT * FROM bieren

```

BierNr	Naam	BrouwerNr	SoortNr	Alcohol
4	A.C.O.	104	18	7.00
5	Aalbeeks St. Corneliusbier (=Kapittel pater (Het))	113	18	6.50
7	Aardbeien witbier	56	53	2.50
8	Aarschots kruikenbier (=St. Sebastiaan grand cru)	105	15	7.60
10	Abt Bijbier (Nen)	33	18	7.00
11	Adler	51	42	6.75
12	Aerts 1900	81	14	7.00
13	Affligem blond (Abdij)	100	33	7.00
14	Affligem christmas ale (Abdij)	100	36	9.00
15	Affligem dubbel (Abdij)	100	14	7.00
16	Affligem patersvat	100	33	7.00

bieren 2 x

*In praktijk*

gebruik je de instructie

```
SELECT * FROM bieren
```

*Voorbeeld 3**Instructie*

```
SELECT * FROM bieren
UNION ALL
SELECT * FROM bieren
```

*Resultaat:* 2430 records

Met union all worden alle records getoond, ook de dubbele (2430 records). De query wordt ook sneller uitgevoerd.

*In praktijk*

ga je dit nooit doen.

*Voorbeeld 4*

Je wil een lijst met alle bieren. Elk bier krijgt de vermelding 'Alcoholvrij', 'Alcoholarm', 'Bevat alcohol' of 'Niet gekend'.

De juiste boodschap kan je bepalen op basis van het alcoholpercentage (<0,2; <0,5; >0,5 en IS NULL). Sorteer op alcoholpercentage.

*Oplossing*

```
SELECT naam, alcohol, 'Alcoholvrij' AS type FROM bieren WHERE alcohol<0.2
UNION
SELECT naam, alcohol, 'Alcoholarm' AS type FROM bieren WHERE alcohol>=0.2 and alcohol
<0.5
UNION
SELECT naam, alcohol, 'Bevat alcohol' AS type FROM bieren WHERE alcohol>=0.5
UNION
SELECT naam, alcohol, 'Niet gekend' AS type FROM bieren WHERE alcohol IS NULL
ORDER BY alcohol
```

*Resultaat:* 1215 records

```

1 • SELECT naam, alcohol, 'Alcoholvrij' AS type FROM bieren WHERE alcohol<0.2
2 UNION
3 SELECT naam, alcohol, 'Alcoholarm' AS type FROM bieren WHERE alcohol>=0.2 and alcohol <0.5
4 UNION
5 SELECT naam, alcohol, 'Bevat alcohol' AS type FROM bieren WHERE alcohol>=0.5
6 UNION
7 SELECT naam, alcohol, 'Niet gekend' AS type FROM bieren WHERE alcohol IS NULL
8 ORDER BY alcohol

```

naam	alcohol	type
XX Bitter	NULL	Niet gekend
Tourtel malt	0.10	Alcoholvrij
Dommelsch malt	0.10	Alcoholvrij
Jupiler NA	0.10	Alcoholvrij
Stella Artois NA	0.10	Alcoholvrij
Edel-Brau	0.10	Alcoholvrij
Palm green	0.10	Alcoholvrij
Brouwmeester alcohol vrij bier	0.40	Alcoholarm
Star bruin	0.50	Bevat alcohol

Result 4 x

## 6.2. Subqueries Intro

Een andere manier om queries te combineren is het gebruik van een subquery.

Bij een **subquery** wordt het resultaat van een SELECT-instructie gebruikt binnen een andere SQL-instructie.

Waar kun je een subquery in je query gebruiken?

- in de WHERE van een SELECT,
- in de HAVING van een SELECT,
- in de SELECT van een SELECT,
- in de FROM van een SELECT,
- in de WHERE van een INSERT (komt later aan bod),
- in de WHERE van een UPDATE (komt later aan bod),
- in de WHERE van een DELETE (komt later aan bod).

De voorbeelden op de volgende pagina laten zien hoe je subqueries kan gebruiken.

Een speciale subquery is de **gecorrleerde subquery**. Dit is een query waarin een kolom gebruikt wordt die tot een table behoort die in een ander queryblok gespecificeerd is. Ook hiervan krijg je een voorbeeld (voorbeeld 5) op de volgende pagina.

### 6.2.1. Voorbeelden

Probeer onderstaande voorbeelden uit in de database BIEREN.

#### Voorbeeld 1

Geef een lijst van alle bieren met het hoogste alcoholpercentage.

*Oplossing*

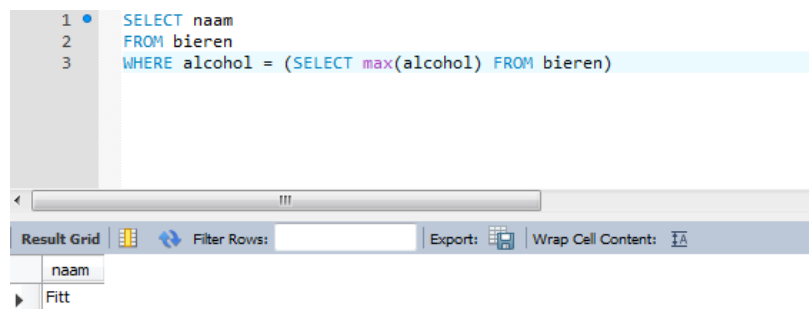
```

SELECT naam
FROM bieren
WHERE alcohol = (SELECT max(alcohol) FROM bieren)

```

In dit voorbeeld mag de subquery slechts **één waarde** als resultaat geven. Dit is zo voor de operatoren =, <, >, >=, <= en <>.

*Resultaat: 1 record*



bieren 5 × [redacted]

### Voorbeeld 2

Geef een lijst van alle bieren die in Oudenaarde gebrouwen worden.

*Oplossing*

```

SELECT naam
FROM bieren
WHERE brouwer nr in (SELECT brouwer nr FROM brouwers WHERE gemeente =
'Oudenaarde')

```

Alleen met de operator **in** kan je met een set van waarden werken.

*Resultaat: 14 records*

```

1 SELECT naam
2 FROM bieren
3 WHERE brouwer nr in (SELECT brouwer nr FROM brouwers WHERE gemeente = 'Oudenaarde')

```

naam
Felix kriebier
Felix Oudenaards kriebier
Felix Oudenaards oud bruin
Felix pils
Felix speciaal Oudenaards
Jan van Gent
Liefmans frambozenbier
Liefmans gluhkriek
Liefmans goudenband
Liefmans kriek
Liefmans odgar

bieren 6 ×

### Voorbeeld 3

Geef de soortnr van de soorten die maar door 1 brouwerij gebrouwen worden.

#### Oplossing

```

SELECT soortnr
FROM (SELECT DISTINCT soortnr, brouwer nr FROM bieren) as lijst
GROUP BY soortnr
HAVING count(*)=1

```

De subquery staat in de from-clausule. Eerst wordt de subquery uitgevoerd, nadien wordt de rest van de select-instructie uitgevoerd.

Als de substring in de from-component staat, moet deze een **alias** krijgen.

Resultaat: 5 records

```

1 SELECT soortnr
2 FROM (SELECT DISTINCT soortnr, brouwer nr FROM bieren) as lijst
3 GROUP BY soortnr
4 HAVING count(*)=1

```

soortnr
25
26
44
64
65

Result 7 ×

### Voorbeeld 4

Maak een lijst met het gemiddelde alcoholpercentage per soort.

#### Oplossing

```
SELECT soort, gemiddelde
FROM (SELECT soortnr, avg(alcohol) as gemiddelde FROM bieren GROUP BY soortnr) as r1
INNER JOIN soorten
ON r1.soortnr = soorten.soortnr
```

In de subquery wordt eerst het gemiddelde alcoholpercentage berekend per soortnr.

Gemiddelde is een aliasnaam uit de subquery die als kolomnaam gebruikt wordt in de buitenste query.

Om het resultaat van de subquery te kunnen joinen met een andere table moet je een aliasnaam geven aan de subquery.

**Resultaat:** 38 records

```
1 SELECT soort, gemiddelde
2 FROM (SELECT soortnr, avg(alcohol) as gemiddelde FROM bieren GROUP BY soortnr) as r1
3 INNER JOIN soorten
4 ON r1.soortnr = soorten.soortnr
```

soort	gemiddelde
Alcoholarm	3.440000
Alcoholvrij	0.100000
Ale	5.432000
Alt	7.500000
Amber	5.578571
Bierette	3.125000
Bitter	7.337500
Donkerbok	6.720000
Dort	5.300000
Dubbel Donker	7.078321
Flalkier	7.890155

Result 8

### Voorbeeld 5

Maak een lijst van alle bieren met een lagere alcoholpercentage dan het gemiddelde alcoholpercentage van zijn eigen soort.

#### Oplossing

```
SELECT b1.*
FROM bieren as b1
WHERE b1.alcohol < (SELECT avg(b2.alcohol) FROM bieren as b2 WHERE
b2.soortnr=b1.soortnr)
```

Hier wordt tweemaal gebruik gemaakt van de table *bieren*. Daarom wordt er aan de table een alias toegekend. Om aan te duiden tot welke table de velden behoren gebruiken we hier de alias i.p.v. de tablenaam bieren.

**Resultaat:** 603 records

```


1 SELECT b1.*
2 FROM bieren as b1
3 WHERE b1.alcohol < (SELECT avg(b2.alcohol) FROM bieren as b2 WHERE b2.soortnr=b1.soortnr)

```

BierNr	Naam	BrouwerNr	SoortNr	Alcohol
7	Aardbeien witbier	56	53	2.50
8	Aarschots kruikenbier (=St. Sebastiaan grand cru)	105	15	7.60
12	Aerts 1900	81	14	7.00
14	Affligem christmas ale (Abdij)	100	36	9.00
15	Affligem dubbel (Abdij)	100	14	7.00
20	Aldegonde brune	72	36	8.50
21	Aldegonde cuvee	72	15	7.50
22	Aldegonde speciale	72	36	8.50
25	Alfa (=Huyghe blond)	56	51	2.00
26	Alfri	93	2	0.80
77	Alken face	7	51	1.30

### 6.2.2. Wanneer gebruik je een subquery en wanneer join?

Soms kan je een opdracht zowel oplossen door gebruik te maken een subquery als door gebruik te maken van joins. In dat geval bestudeer je best het bijhorende execution plan. Dit is een overzicht hoe de database de query stap per stap uitvoert en welke van die stappen veel moeite vragen.

Om het execution plan te tonen, klik je op de knop .

Voorbeeld: Je wil weten welke brouwers bieren brouwen met een alcoholpercentage van 12.

Je kan dit oplossen met een subquery en met een join. We bekijken voor beide oplossingen het execution plan.

#### met subquery

##### Instructie

```

select *
from brouwers
where brouwerNr in (select brouwerNr from bieren where alcohol=12)

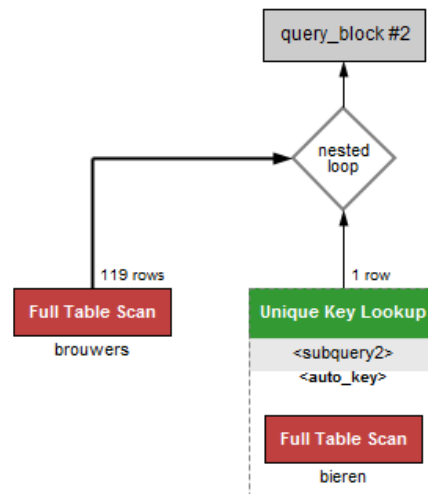
```

##### Execution plan

Dit execution plan bevat twee stappen met een Full Table Scan.

Een Full Table Scan betekent dat alle records uit die table gelezen worden.





### met join

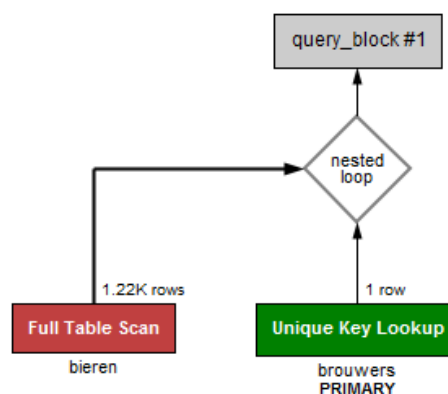
#### Instructie

```

select brouwers.*
from brouwers inner join bieren on brouwers.brouwernr=bieren.brouwernr
where alcohol=12
  
```

#### Execution plan

Dit execution plan bevat maar één Full Table Scan. Deze oplossing krijgt hierdoor de voorkeur.



## 6.3. Opgaven

In dit onderdeel krijg je een hele reeks oefeningen op het gebruik van union en subquery's

Alle oefeningen kan je maken in de database PLANTEN. De oplossingen vind je in de webversie.

### 6.3.1.Opgave 1

Het tuincentrum wil graag een lijst waarop is aangegeven welke bomen in Aalsmeer en welke buiten Aalsmeer verkrijgbaar zijn. Op het overzicht moeten de volgende gegevens verschijnen : plantid, plantennaam, artikelcode van de leverancier, alsmede een aanduiding 'AALSMEER' of 'BUITEN AALSMEER'.

Sorteer de lijst op plantid.

Resultaat: 23 records

Begin van de lijst:

	plantId	plantNaam	artikelLeverancierCode	plaats
▶	6	JENEVERBES	002	buiten Aalsmeer
	10	GOUDEN REGEN	011	buiten Aalsmeer
	11	DWERG CYPRES	103	buiten Aalsmeer
	17	LINDE	B-104	buiten Aalsmeer
	25	ZILVERSPAR	D555	Aalsmeer
	44	TULPEBOOM	B-011	buiten Aalsmeer
	44	TULPEBOOM	078	buiten Aalsmeer
	45	MEIDOORN	114	buiten Aalsmeer
	48	ACACIA	C243	Aalsmeer
	50	TULPEBOOM	B-011	Aalsmeer

### 6.3.2. Opgave 2

Je wil een overzicht waarop is aangegeven welke bestellingen te laat zijn. Voor de bestellingen die te laat zijn moet in een extra kolom de opmerking 'TE LAAT' worden geplaatst; voor de andere bestellingen wordt in die kolom een aantal streepjes geplaatst.

Sorteer het overzicht op bestelid.

Geef bestelid, besteldatum, artikelLeverancierId, leveringsdatum, levertijd, bericht 'TE LAAT' of '-----'.

Resultaat: 93 records

Begin van de lijst:

	bestelId	artikelLeverancierId	bestelDatum	leveringsDatum	levertijd	bericht
▶	1	70	2016-01-17 00:00:00	2016-01-31 00:00:00	10	TE LAAT
	1	75	2016-01-17 00:00:00	2016-01-31 00:00:00	14	-----
	1	50	2016-01-17 00:00:00	2016-01-31 00:00:00	7	TE LAAT
	1	59	2016-01-17 00:00:00	2016-01-31 00:00:00	10	TE LAAT
	1	60	2016-01-17 00:00:00	2016-01-31 00:00:00	10	TE LAAT
	1	56	2016-01-17 00:00:00	2016-01-31 00:00:00	14	-----
	2	6	2016-02-25 00:00:00	2016-03-04 00:00:00	7	TE LAAT
	2	7	2016-02-25 00:00:00	2016-03-04 00:00:00	7	TE LAAT

### 6.3.3. Opgave 3

Welke planten zijn hoger dan de gemiddelde hoogte van alle planten samen? Toon alle gegevens.

Resultaat: 21 records

Begin van de lijst:

	plantId	plantNaam	soort	kleur	hoogte	beginBloeimaand	eindBloeimaand	prijs	categorieId
▶	9	OLIJFWILG	HEESTER	GEEL	400	9	10	10	6
	10	GOUDEN REGEN	BOOM	GEEL	600	5	5	22	6
	11	DWERGCEYPRES	BOOM	NULL	500	0	0	13.5	6
	17	LINDE	BOOM	GEEL	4000	7	8	42.5	6
	25	ZILVERSPAR	BOOM	NULL	3000	0	0	19.5	6
	38	SERING	HEESTER	PAARS	500	5	6	19.5	6
	44	TULPEBOOM	BOOM	GEEL	2000	6	7	22.5	6
	45	MEIDOORN	BOOM	ROZE	700	5	5	19.5	6
	48	ACACIA	BOOM	WIT	2500	6	6	17.5	6
	50	TULPEBOOM	BOOM	GEEL	2000	6	7	22.5	6

### 6.3.4. Opgave 4

Welke planten zijn duurder dan de gemiddelde prijs van de bomen? Toon alle gegevens.

Resultaat: 13 records

	plantId	plantNaam	soort	kleur	hoogte	beginBloeimaand	eindBloeimaand	prijs	categorieId
▶	10	GOUDEN REGEN	BOOM	GEEL	600	5	5	22	6
	17	LINDE	BOOM	GEEL	4000	7	8	42.5	6
	25	ZILVERSPAR	BOOM	NULL	3000	0	0	19.5	6
	38	SERING	HEESTER	PAARS	500	5	6	19.5	6
	39	RODODENDRON	HEESTER	ROOD	125	5	7	19.5	4
	44	TULPEBOOM	BOOM	GEEL	2000	6	7	22.5	6
	45	MEIDOORN	BOOM	ROZE	700	5	5	19.5	6
	48	ACACIA	BOOM	WIT	2500	6	6	17.5	6
	70	HULST	HEESTER	NULL	700	0	0	22.5	6
	71	ESDOORN	BOOM	GROEN	2500	6	6	17.5	6
	107	AZALEA	HEESTER	ORANJE	200	4	5	17.5	5
	112	TOVERHAZELAAR	HEESTER	GEEL	500	1	2	32	6
	118	PAARDEKASTA...	BOOM	WIT	2500	5	5	17.5	6
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### 6.3.5. Opgave 5

Maak een overzicht van de leveranciers (alle gegevens) waar nog bestellingen uitstaan met een leverdatum die vóór 1 april 2016 ligt.

Resultaat: 7 records

	leverancierId	leverancierNaam	adres	woonplaats
▶	1	HOVENIER G.H.	ZANDWEG 50	LISSE
	2	BAUMGARTEN R.	TAKSTRAAT 13	HILLEGOM
	4	SPITMAN EN ZN.	ACHTERTUIN 9	AALSMEER
	5	DEZAAIER L.J.A.	DE GRONDEN 101	LISSE
	7	BLOEM L.Z.H.W.	LINNAEUSHOF 17	HILLEGOM
	9	ERICA BV.	BERKENWEG 87	HEEMSTEDE
	11	FLORA BV.	OEVERSTRAAT 76	AALSMEER
*	NULL	NULL	NULL	NULL

### 6.3.6. Opgave 6

Welke rijen hebben de laagste offerteprijs van alle offertes in de table *artikelsleveranciers*? Geef alle gegevens.

Resultaat: 2 records

	artikelLeverancierId	leverancierId	artikelLeverancierCode	plantId	levertijd	offertePrijs
▶	139	7	263	76	7	0.05
	200	10	WIST	99	14	0.05
*	NULL	NULL	NULL	NULL	NULL	NULL

### 6.3.7. Opgave 7

Welke planten zijn lager dan de laagste vaste plant?

Toon alle gegevens.

Planten waar de hoogte 0 is, worden niet meegerekend.

Resultaat: 5 records

	plantId	plantNaam	soort	kleur	hoogte	beginBloeimaand	eindBloeimaand	prijs	categorieId
▶	1	BEGONIA	1-JARIG	ROOD	15	6	9	0.65	1
	68	TIJM	KRUID	PAARS	10	6	6	2.5	1
	76	KROKUS	BOL	WIT	15	2	3	0.1	1
	88	ALYSSUM	1-JARIG	PAARS	10	6	9	0.6	1
	102	VIOOLTJE	2-JARIG	GEMENGD	15	3	8	0.5	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### 6.3.8. Opgave 8

Welke planten zijn hoger dan de gemiddelde hoogte van vaste planten en tevens goedkoper dan de gemiddelde prijs van vaste planten? Geef alle gegevens.

Resultaat: 13 records

Begin van de lijst:

	plantId	plantNaam	soort	kleur	hoogte	beginBloeimaand	eindBloeimaand	prijs	categorieId
▶	5	LUPINE	VAST	GEMENGD	100	6	7	2.5	3
	16	RIDDERSPOOR	VAST	LILA	150	6	7	3	4
	28	MONNIKSKAP	VAST	VIOLET	120	8	9	3	4
	33	ROZEMARIJN	KRUID	BLAUW	150	5	5	2	4
	36	IRIS	BOL	BLAUW	100	5	7	0.14	3
	47	STOKROOS	VAST	ROOD	250	6	9	2	5
	56	LIGUSTER	HEESTER	WIT	200	7	7	0.4	5
	62	KOGELDISTEL	VAST	BLAUW	175	6	7	3	5
	85	LEVERKRUID	VAST	PAARS	175	8	9	2.5	5
	92	SALIE	KRUID	VIOLET	100	6	7	2	3
	106	DRAGON	KRUID	WIT	100	8	9	2	3

### 6.3.9. Opgave 9

Welke planten hebben een prijs die tussen de laagste en hoogste prijs van de klimplanten ligt? Geef alle gegevens.

SQL

Resultaat: 11 records

	plantId	plantNaam	soort	kleur	hoogte	beginBloeimaand	eindBloeimaand	prijs	categorieId
▶	2	KLIMOP	KLIM	NULL	0	0	0	7.5	5
	6	JENEVERBES	BOOM	NULL	250	0	0	6.5	6
	9	OLIJFWILG	HEESTER	GEEL	400	9	10	10	6
	27	PAMPUSGRAS	VAST	WIT	300	9	10	9.5	5
	50	JUDASBOOM	BOOM	ROZE	800	5	5	9.5	6
	66	LIJSTERBES	BOOM	WIT	500	5	5	7.5	6
	69	BOSRANK	KLIM	PAARS	300	7	9	6.5	5
	72	PASSIEBLOEM	KLIM	BLAUW	0	6	9	9.5	1
	78	BERK	BOOM	NULL	2000	0	0	7.5	6
	84	AZIJBLOOM	BOOM	ROOD	0	6	7	9.5	6
	101	WIJNSTOK	KLIM	NULL	600	0	0	10	6
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### 6.3.10. Opgave 10

Maak een overzicht van alle plantIds die een lagere offerteprijs hebben dan de gemiddelde offerteprijs voor de betreffende plantId.

Plaats de gegevens plantId, leveranciersnaam en offerteprijs op het overzicht. Sorteer op plantId.

Resultaat: 71 records

Begin van de lijst:

	plantId	leverancierNaam	offertePrijs
▶	1	DEZAAIER L.J.A.	0.35
	2	DE GROENE KAS BV.	4.65
	3	DEZAAIER L.J.A.	1.45
	5	DE GROENE KAS BV.	1.55
	5	MOOIWEER FA.	1.5
	9	BAUMGARTEN R.	4.5
	9	TRA A.	5
	14	DEZAAIER L.J.A.	1.7
	16	DEZAAIER L.J.A.	1.7
	18	MOOIWEER FA.	1.35
	21	MOOIWEER FA.	2.1

### 6.3.11. Opgave 11

Maak een overzicht van bestelde planten die een bestelprijs hebben welke hoger is dan de maximum offerteprijs voor zo'n plant.

Plaats de volgende gegevens op het overzicht : bestelnummer, artikelcode van de leverancier, plantennaam en bestelprijs.

Resultaat: 34 records

Begin van de lijst:

	bestelid	artikelLeverancierCode	plantNaam	bestelPrijs
▶	2	A421	SLEUTELBLOEM	1.35
	2	B148	KLAPROOS	0.65
	2	B331	JUDASPENNING	0.7
	4	044	BERK	3.6
	4	114	MEIDOORN	9.45
	5	E-003	DOPHEIDE	1.2
	6	162	LEEUWEBEKJE	0.65
	6	281	SIERUI	3.25
	6	471	VLAMBLOEM	1.3
	9	ST2P1	ZONNEBLOEM	1.35
	9	ST3D2	KITMOED	5.05

## 6.4. Opdracht voor de coach 2

Op de volgende pagina krijg je een reeks opdrachten. Gebruik hiervoor de database Video.

Verzamel al je oplossingen in een tekstbestand en stuur dit door ter verbetering aan je coach.

Vermeld als onderwerp "**Extra mogelijkheden**".

Stel de SQL-instructies op om volgende opdrachten uit te voeren. Gebruik hiervoor de database Video.

1. Geef een lijst van de films die in de titel noch het woord "kill" noch het woord "blood" hebben. We willen enkel de films van de genres griezel en thriller.
2. We willen een lijst maken van wie welke films huurt. De lijst wordt oplopend gesorteerd op naam en titel.
3. Wat is de totale voorraad per genre?
4. Op basis van het huuraantal willen we een label toevoegen aan elke klant:  
Als het huuraantal groter is dan 250 krijgt de klant het label 'Zeer goede klant', als het huuraantal kleiner is dan 150 krijgt hij het label 'Slechte klant'. In de andere gevallen komt er als label 'Goede klant'.
5. Geef alle gegevens van de duurste film.

## Hoofdstuk 7. Gegevens aanpassen

### 7.1. Inleiding

Tot nu toe hebben we enkel gegevens geselecteerd uit verschillende tables.

In dit hoofdstuk bekijken we hoe je de inhoud van je tables kan wijzigen:

- gegevens toevoegen,
- gegevens aanpassen,
- gegevens verwijderen.

### 7.2. Insert

In dit onderdeel bekijken we hoe je gegevens aan een table kan **toevoegen**.

We maken een onderscheid tussen het toevoegen van één en van meerdere records:

- waarbij je zelf de waarden mee geeft
- die je haalt uit een één of meerdere tables uit je database

#### 7.2.1. Zelf waarden opgeven

In SQL kan je met de INSERT-instructie rijen toevoegen aan een bestaande table.

Syntax:

```
INSERT INTO doel [(veld1[,veld2[,...]])]
VALUES (waarde1[,waarde2 [,...]])
```

De instructie bevat volgende onderdelen:

Onderdeel	Betekenis
<b>doel</b>	De naam van de table waaraan de records toegevoegd worden.
<b>veld1, veld2</b>	Dit zijn de namen van de velden waaraan de gegevens toegevoegd worden. Deze moeten altijd tussen ronde haakjes ( ) staan.
<b>waarde1, waarde2</b>	De waarden die in de velden van het nieuwe record moeten toegevoegd worden.  Elke waarde wordt ingevoegd in het veld dat overeenkomt met de positie van de waarde in de lijst: waarde1 wordt ingevoegd in veld1, waarde2 in veld2,...  Elke waarde van een niet-numeriek veld moet tussen aanhalingstekens '' staan. De verschillende waarden worden met komma's gescheiden. Rond het geheel van de waarden staan ronde haakjes ( ).

Het meegeven van de kolommen, ook bij een insert van alle velden, is niet verplicht maar wel aan te raden. Het kan zijn dat de velden in een andere volgorde staan. Door het opgeven van de kolomnamen kan hierover geen verwarring ontstaan.

Je kan met één insert-instructie ook meerdere gegevens in één keer toevoegen. Je kan meerdere reeksen values na elkaar plaatsen. Tussen twee values-reeksen plaats je telkens een komma.

### 7.2.2. Uit andere tables

Je kan ook gegevens toevoegen vanuit andere tables. Via een select-instructie selecteer je de juiste records.

Syntax:

```
INSERT INTO doel [(veld1[,veld2[,...]])]
SELECT [bron.] veld1 [, veld2 [,...]]
FROM tableexpressie
[WHERE voorwaarde]
```

De instructie bevat volgende onderdelen:

Onderdeel	Betekenis
<b>doel</b>	De naam van de table waaraan de records toegevoegd worden.
<b>bron</b>	De naam van de table (of query) waaruit de records gekopieerd worden.
<b>veld1, veld2</b>	Dit zijn de namen van de velden waaraan de gegevens toegevoegd worden. Deze moeten altijd tussen ronde haakjes ( ) staan.
<b>tableexpressie</b>	De naam van de table of tables waaruit de toegevoegde records afkomstig zijn. Dit kan <ul style="list-style-type: none"> <li>• één tablenaam zijn</li> <li>• of een samenstelling met inner join, left join of right join</li> <li>• of een opgeslagen query.</li> </ul>

### 7.2.3. Voorbeelden

Probeer onderstaande voorbeelden uit in de database BIEREN.

Je kan enkel onderaan in het venster Output zien of je instructie goed uitgevoerd is.

#### Voorbeeld 1

Voeg een nieuwe soort 'Extra donker' toe aan de table *soorten*.

*Oplossing*

```
INSERT INTO soorten (Soort)
VALUES ('Extra donker')
```

Omdat SoortNr een autonummeringsveld is, moeten we dit niet opnemen in onze SQL-instructie. Dit veld zal automatisch ingevuld worden.

Je krijgt onderaan volgende boodschap:

Output			
Action Output			
	Time	Action	Message
✓	1 20:53:12	INSERT INTO Soorten (Soort) VALUES ('Extra donker')	1 row(s) affected

*Resultaat table soorten*



Om het resultaat te bekijken, voer je een query uit die alle velden toont van de table *soorten*.  
Het nieuwe record heeft als SoortNr 66 gekregen.

1 • `SELECT * FROM soorten |`

SoortNr	Soort
12	Donkerbok
13	Dort
14	Dubbel Donker
15	Edelbier
18	Extra
66	Extra donker
19	Faro
21	Gerstewijn
22	Geuze
25	Helderbok
76	Tiebiaer

soorten 1 x

### Voorbeeld 2

Voeg een nieuwe brouwer toe aan de table *brouwers*.

Gegevens van de nieuwe brouwerij: Brouwerij Vaattappers is gevestigd aan Interleuvenlaan 2 in 3000 Heverlee met een omzet van 1000.

### Oplossing

```
INSERT INTO brouwers (BrNaam, Adres, Postcode, Gemeente, Omzet)
VALUES ('Brouwerij Vaattappers', 'Interleuvenlaan 2', 3000, 'Heverlee', 1000)
```

Ook hier nemen we BrouwerNr niet mee in de SQL-instructie omdat het een autonummeringsveld is.

Output			
Action Output			
	Time	Action	Message
✓	2 20:54:15	SELECT * FROM soorten	39 row(s) returned
✓	3 20:59:01	INSERT INTO Brouwers (Brnaam, Adres, Postcode, Gemeente, Omzet) VALUES ('Brouwerij V...	1 row(s) affected

### Resultaat table *brouwers*

Dit record krijgt als BrouwerNr 127 en wordt onderaan de table toegevoegd.

1 • `SELECT * FROM brouwers`

BrouwerNr	BrNaam	Adres	PostCode	Gemeente	Omzet
119	Vervifontaine	xxx	0	fontaine-Jalhay	NULL
120	Villers	Liezeledorp 37	2870	Liezele-Puurs	1450
121	Walrave	Lepelstraat 36	9270	Laarne	2500
122	Weldebroec	Mechelse Steenweg 53	2830	Willebroek	800
123	Westmalle	Antwerpsesteenweg ...	2390	Malle	125000
124	Westvleteren	Donkerstraat 12	8640	Westvleteren	4000
125	Wieze	Nieuwstraat 1	9280	Wieze	120000
126	Zwingel	Damweg 6	8530	Harelbeke	250
127	Brouwerij Vaattappers	Interleuvenlaan 2	3000	Heverlee	1000
*	NULL	NULL	NULL	NULL	NULL

brouwers 2 x

### Voorbeeld 3

Voeg met één instructie drie nieuwe biersoorten toe aan de table *soorten*: *witbier*, *Ice bier* en *honingbier*.

#### Oplossing

```
INSERT INTO soorten (Soort)
VALUES
('witbier'),
('Ice bier'),
('honing bier')
```

Ook hier nemen we SoortNr weer niet mee omdat dit een autonummeringsveld is.

#### Resultaat table soorten

Om het resultaat te bekijken, voer je een query uit die alle velden toont van de table *soorten*.

1 • `SELECT * FROM soorten ORDER BY soortnr`

SoortNr	Soort
60	Versnijbier
61	Vlaams Bruin
62	Vlaams Rood
64	West-Vlaamse Geuze
65	West-Vlaamse spontane Geuze
66	Extra donker
67	witbier
68	Ice bier
69	honing bier
*	NULL

### Voorbeeld 4

Voeg de bieren met een alcoholpercentage hoger dan 10 toe aan de table *bieren\_oud*.

#### Oplossing

Selecteer eerst de gegevens om je instructie te controleren:

```
SELECT *
FROM bieren
WHERE alcohol > 10
```

Deze select-instructie voeg je in in je insert into-instructie:

```
INSERT INTO bieren_oud
SELECT *
FROM bieren
WHERE alcohol > 10
```

Output			
Action Output			
	Time	Action	Message
✓	5 21:04:56	SELECT * FROM bieren WHERE alcohol > 10	17 row(s) returned
✓	6 21:05:07	INSERT INTO bieren_oud SELECT * FROM bieren WHERE alcohol > 10	17 row(s) affected Records: 17 Duplicates: 0 Warnings: 0

#### Resultaat table *bieren\_oud*

Om het resultaat te controleren vraag je alle velden van de table *bieren\_oud* op.

1 • SELECT \* FROM bieren\_oud

Result Grid					
Filter Rows:					
	BierNr	Naam	BrouwerNr	SoortNr	Alcohol
▶	189	Boucanier (Biere du (=Piraat)	14	21	10.50
	227	Bush beer	36	21	12.00
	229	Bush de Noel	36	21	12.00
	511	Fantome brewery (The best of)	44	36	12.00
	529	Fitt	125	2	15.00
	584	Gaverhopke (t) bruin 12	47	36	12.00
	612	Golden dragon beer (=Gulden Draak)	14	21	10.50
	638	Gulden Draak	14	36	10.50
	748	Kasteel van Ramegnies-Chin (Bier v/h)	43	36	12.00
	750	Kasteelbier Ingelmunster	114	36	11.00
	751	Kasteelbier Ooidonk (=Kasteelbier In	114	36	11.00

bieren\_oud 4 x

## 7.3. Update

In dit onderdeel bekijken we hoe je gegevens kan aanpassen.

### 7.3.1. Syntax

De update-instructie gebruik je als je een groot aantal records wil wijzigen.

Syntax:

SQL

UPDATE tableexpressie  
SET nieuwewaarde  
WHERE criteria

De instructie bevat volgende onderdelen:

Onderdeel	Betekenis
<b>tableexpressie</b>	De naam van de table waarin gegevens gewijzigd moeten worden.
<b>nieuwewaarde</b>	Een expressie die bepaalt welke waarde wordt ingevoegd in een bepaald veld in de bijgewerkte records.
<b>criteria</b>	Een expressie die bepaalt welke records worden bijgewerkt. Alleen de records die voldoen aan de expressie, worden bijgewerkt.

Tableexpressie mag je ook vervangen door meerdere tables verbonden via een inner join. Deze manier van werken wordt afgeraden omdat dit niet in alle databases op dezelfde manier gebruikt kan worden.

Je kan beter werken met een subquery (zoals in voorbeeld 2).



Als je een update instructie gebruikt zonder where clause op de primary key krijg je de error code 1175.

Voer de instructie SET SQL\_SAFE\_UPDATES=0 uit om alle soorten update instructies toe te laten. MySQL onthoudt dit tot je de Workbench sluit.

### 7.3.2. Voorbeelden

Probeer onderstaande voorbeelden uit in de database BIEREN.

Ook bij deze oefeningen krijg je enkel onderaan in het venster Output te zien of je instructie goed uitgevoerd is. Je kan het resultaat enkel controleren door een select-statement te maken.

#### Voorbeeld 1

Verhoog het alcoholpercentage van de bieren (met soortnr 21) met 0,5. Doe deze aanpassing in de table *bieren\_oud*.

#### Oplossing

```
UPDATE bieren_oud
SET alcohol = alcohol+0.5
WHERE soortnr = 21
```

Resultaat table *bieren\_oud*: 7 records gewijzigd

Output		
Action Output		
	Time	Action
✓	7 21:05:43	SELECT * FROM bieren_oud
		17 row(s) returned
✓	8 21:09:06	UPDATE bieren_oud SET alcohol = alcohol+0.5 WHERE soortnr = 21
		7 row(s) affected Rows matched: 7 Changed: 7 Warnings: 0

1 • `SELECT * FROM bieren_oud`

BierNr	Naam	BrouwerNr	SoortNr	Alcohol
189	Boucanier (Biere du) (=Piraat)	14	21	11.00
227	Bush beer	36	21	12.50
229	Bush de Noel	36	21	12.50
511	Fantome brewery (The best of)	44	36	12.00
529	Fitt	125	2	15.00
584	Gaverhopke (t) bruin 12	47	36	12.00
612	Golden dragon beer (=Gulden Draak)	14	21	11.00
638	Gulden Draak	14	36	10.50
748	Kasteel van Ramegnies-Chin (Bier v/h)	43	36	12.00
750	Kasteelbier Ingelmunster	114	36	11.00
751	Kasteelbier Ooidoek (=Kasteelbier In	114	36	11.00

bieren\_oud 5 x

## Voorbeeld 2

Verlaag het alcoholpercentage met 1 van de bieren waarvan de brouwerij een omzet heeft die groter is dan 25000. Doe deze aanpassing in de table *bieren\_oud*.

## Oplossing

```
UPDATE bieren_oud
SET alcohol = alcohol - 1
WHERE brouwerNr in (SELECT brouwerNr FROM brouwers WHERE omzet > 25000)
```

Resultaat table *bieren\_oud*: 10 records gewijzigd

Time	Action	Message
9 21:09:56	SELECT * FROM bieren_oud	17 row(s) returned
10 21:14:36	UPDATE bieren_oud SET alcohol = alcohol - 1 WHERE brouwerNr in (SELECT brouwerNr FR...	10 row(s) affected Rows matched: 10 Changed: 10 Warnings: 0

1 • `SELECT * FROM bieren_oud`

BierNr	Naam	BrouwerNr	SoortNr	Alcohol
189	Boucanier (Biere du) (=Piraat)	14	21	10.00
227	Bush beer	36	21	12.50
229	Bush de Noel	36	21	12.50
511	Fantome brewery (The best of)	44	36	12.00
529	Fitt	125	2	14.00
584	Gaverhopke (t) bruin 12	47	36	12.00
612	Golden dragon beer (=Gulden Draak)	14	21	10.00
638	Gulden Draak	14	36	9.50
748	Kasteel van Ramegnies-Chin (Bier v/h)	43	36	12.00
750	Kasteelbier Ingelmunster	114	36	10.00
751	Kasteelbier Ooidoek (=Kasteelbier In	114	36	10.00

bieren\_oud 6 x

## 7.4. Delete

In dit onderdeel bekijken we tenslotte hoe je gegevens kan verwijderen.

### 7.4.1. Syntax

De DELETE-instructie wordt gebruikt om data uit de database te verwijderen. De geselecteerde rijen worden uit de table verwijderd.

Met een DELETE-instructie verwijder je altijd een **volledig record**. Het is niet mogelijk om alleen de gegevens van bepaalde velden te verwijderen. Dit kan alleen met een UPDATE-instructie.

De DELETE-instructie gebruik je best niet om een volledige table te verwijderen. Dit kan je beter oplossen met een DROP-instructie (zie hoofdstuk Beheer van tables en relaties).

Als je records verwijderd hebt met een DELETE-instructie, kan je dit **niet ongedaan maken**. Het is dus belangrijk om eerst via een SELECT-instructie de juiste records te selecteren voor je ze effectief gaat verwijderen.

Syntax:

```
DELETE [table]
FROM tableexpressie
WHERE criteria
```

De instructie bevat volgende onderdelen:

Onderdeel	Betekenis
<b>table</b>	De optionele naam van de table waaruit de records verwijderd worden.
<b>tableexpressie</b>	De naam van de table waaruit records verwijderd moeten worden.
<b>criteria</b>	Een expressie die bepaalt welke records verwijderd worden. Alleen de records die voldoen aan de expressie, worden verwijderd.

Tableexpressie mag je ook vervangen door meerdere tables verbonden via een inner join. Deze manier van werken wordt afgeraden omdat dit niet in alle databases op dezelfde manier gebruikt kan worden.

Je kan beter werken met een subquery (zoals in voorbeeld 2).



Als je een update of delete instructie gebruikt zonder where clause op de primary key krijg je de error code 1175.

Voer de instructie SET SQL\_SAFE\_UPDATES=0 uit om alle soorten update en delete instructies toe te laten. MySQL onthoudt dit tot je Workbench sluit.

### 7.4.2. Voorbeelden

Probeer onderstaande voorbeelden uit in de database BIEREN.

Ook met de DELETE-instructie kan je enkel onderaan in het venster Output zien of je instructie goed uitgevoerd is.

**Voorbeeld 1**

Verwijder het bier met biernr 750 uit de table *bieren\_oud*.

**Oplossing**

```
DELETE
FROM bieren_oud
WHERE biernr=750
```

Output

Action Output

	Time	Action	Message
✓	11 21:15:14	SELECT * FROM bieren_oud	17 row(s) returned
✓	12 21:25:44	DELETE FROM bieren_oud WHERE biernr=750	1 row(s) affected

1 • SELECT \* FROM bieren\_oud

Result Grid

BierNr	Naam	BrouwerNr	SoortNr	Alcohol
189	Boucanier (Biere du) (=Piraat)	14	21	10.00
227	Bush beer	36	21	12.50
229	Bush de Noel	36	21	12.50
511	Fantome brewery (The best of)	44	36	12.00
529	Fitt	125	2	14.00
584	Gaverhopke (t) bruin 12	47	36	12.00
612	Golden dragon beer (=Gulden Draak)	14	21	10.00
638	Gulden Draak	14	36	9.50
748	Kasteel van Ramegnies-Chin (Bier v/h)	43	36	12.00
751	Kasteelbier Ooidonk (=Kasteelbier Ingelmunster)	114	36	10.00
1048	Draak	14	21	10.00

bieren\_oud 7 x

**Voorbeeld 2**

Verwijder uit de table *bieren\_oud* de bieren die in Soy gebrouwen worden.

**Oplossing**

```
DELETE
FROM bieren_oud
WHERE brouwerNr in (SELECT brouwerNr FROM brouwers WHERE gemeente='Soy')
```

Output

Action Output

	Time	Action	Message
✓	13 21:26:28	SELECT * FROM bieren_oud	16 row(s) returned
✓	14 21:28:08	DELETE FROM bieren_oud WHERE brouwerNr in (SELECT brouwerNr FROM brouwers WHE...	1 row(s) affected

1 • `SELECT * FROM bieren_oud`

BierNr	Naam	BrouwerNr	SoortNr	Alcohol
189	Boucanier (Biere du) (=Piraat)	14	21	10.00
227	Bush beer	36	21	12.50
229	Bush de Noel	36	21	12.50
529	Fitt	125	2	14.00
584	Gaverhopke ('t) bruin 12	47	36	12.00
612	Golden dragon beer (=Gulden Draak)	14	21	10.00
638	Gulden Draak	14	36	9.50
748	Kasteel van Ramegnies-Chin (Bier v/h)	43	36	12.00
751	Kasteelbier Ooidonk (=Kasteelbier Ingelmunster)	114	36	10.00
1048	Piraat	14	21	10.00
1115	Pirel triple beer ambree (=Piraat)	14	21	10.00

bieren\_oud 8 x

### 7.4.3. Opgaven

In dit onderdeel maken we een reeks oefeningen op op het aanpassen van de gegevens in je tables.

Alle oefeningen kan je maken in de database PLANTEN. De oplossingen vind je in de webversie.

### 7.4.4. Opgave 1

Het tuincentrum breidt zijn leverancierskring uit met GROEN BV. De volgende gegevens zijn bekend :

<b>Naam</b>	Groen BV.
<b>Adres</b>	Onder de Linde 234
<b>Woonplaats</b>	Aalsmeer

### 7.4.5. Opgave 2

Wijzig in de table *bestellingen* de leverdatum in 5 april 2016 voor de bestelling met nummer 8.

### 7.4.6. Opgave 3

Verwijder alle rijen uit de table *artikelsleveranciers* die betrekking hebben op de heesters van leverancier 8.

*Resultaat:* 14 records verwijderd

### 7.4.7. Opgave 4

Op 23 april 2016 is een nieuwe bestelling geplaatst bij leverancier 4. Dit zijn de details:



ArtikelLeverancierId	Aantal	Prijs
62	10	8.15
58	200	0.40
59	25	2.30
74	50	1.30

De leverancier geeft 8% korting op het bruto bestelbedrag. De leverdatum is één week na de bestelling.

Doe de nodige aanpassingen in de tables *bestellingen* en *bestellijnen*. Denk aan de volgorde!

#### 7.4.8.Opgave 5

Verhoog alle offerteprijzen van de bolgewassen in de table *artikelsleveranciers* met 10%.

*Resultaat:* 7 records aangepast

#### 7.4.9.Opgave 6

Voeg de informatie van de planten met een hoogte groter of gelijk aan 1000 toe aan de table *planten\_oud*. Zorg dat de hoogste planten bovenaan staan.

*Resultaat:* 11 records toegevoegd

## Hoofdstuk 8. Beheer van tables en relaties

### 8.1. Inleiding

Tot nu toe hebben we alleen gewerkt met bestaande tables en met de gegevens in deze tables.

In dit hoofdstuk bekijken we hoe je de tablestructuur kan aanpassen:

- een table kan maken,
- een table kan verwijderen,
- een table kan aanpassen,
- een index maakt.

Omdat er in het vorige hoofdstuk misschien een aantal dingen misgelopen zijn, kan je het eindresultaat van het vorige hoofdstuk opnieuw downloaden. Op deze manier ben je zeker dat je terug met de juiste gegevens begint.

Importeer dit bestand zoals je dit in het begin van de cursus ook gedaan hebt.

### 8.2. Table maken

In dit onderdeel bekijken we hoe je een table kan maken.

We bespreken 2 manieren:

- een lege table,
- een nieuwe table maken op basis van geselecteerde records.

#### 8.2.1. Een lege table maken

Met de instructie Create table kan je een nieuwe table, inclusief velden en beperkende voorwaarden voor velden, maken.

Syntax:

```
CREATE TABLE table  
(veld1 type [(grootte)] [not null] [index1]  
[, veld2 type [(grootte)] [not null] [index2] [...]  
[CONSTRAINT meervoudigeindex [...]])
```

De instructie bevat volgende onderdelen:

Onderdeel	Betekenis
<b>table</b>	De naam van de table die je wil maken.
<b>veld1, veld2</b>	Dit zijn de namen van de velden die je aan de table wil toevoegen. Er moet minstens één veld gemaakt worden. Deze moeten altijd tussen ronde haakjes () staan.
<b>type</b>	Het datatype van het veld in de nieuwe table. Op de volgende pagina krijg je een overzicht van de mogelijke types die je kan gebruiken.
<b>grootte</b>	De veldgrootte in tekens (enkel voor tekst en binaire velden)
<b>index1, index2 meervoudige index</b>	Maakt een index. Meer uitleg in de onderdelen Sleutels en Indexen.

Als **not null** is opgegeven voor een veld moet de inhoud van het veld steeds opgevuld zijn.

### 8.2.2. Datatypes

In onderstaande tabel krijg je een overzicht van de mogelijke datatypes. Voor elk veld van je nieuwe table moet je een type kiezen uit onderstaande lijst. De types die je in deze cursus nodig hebt, staan in het groen.

Naam	MySQL	Opslagwijze	Betekenis
Boolean	Boolean	1 byte	Kan enkel de waarde 0 of 1 bevatten.
Byte TinyInt	TinyInt	1 byte	Kan een getal bevatten tussen -128 en 127.
<b>Datetime</b>	Datetime	8 bytes	Kan een datum en een tijdswaarde bevatten tussen 100 en 9999.
Float	Float	4 bytes	Kan een getal met enkele precisie en floating point bevatten.
<b>Decimal</b>	Decimal(M,D)	4 bytes per 9 tekens	bevat een decimaal getal M = maximaal aantal cijfers (ligt tussen 1 en 65) D = aantal cijfers na de komma (ligt tussen 0 en 30 en D<=M)
Real Double precision	Real Double precision	8 bytes	Kan een getal met dubbele precisie en floating point bevatten. Dit geeft een betere nauwkeurigheid dan het type Float.
SmallInt	SmallInt	2 bytes	Kan een getal bevatten tussen -32768 en 32767.
<b>Int Integer</b>	Integer	4 bytes	Kan een getal tussen -2147483648 en 2147473647 bevatten.
<b>Char Varchar</b>	Char Varchar	1 byte per karakter	Kan een string bevatten. Voorbeelden: <ul style="list-style-type: none"> <li>voornaam varchar(100) → voornaam bevat maximaal 100 karakters.</li> <li>artikelcode char(6) → artikelcode bevat steeds 6 karakters.</li> </ul> De maximale toegelaten waarde als lengte is afhankelijk van de databaseserver.
Binary Varbinary	Binary Varbinary	1 byte per karakter	Kan eender welke informatie bevatten. Het gebruik van binary en varbinary is vergelijkbaar als bij Char en Varchar.

### 8.2.3. Gegevenstype float versus decimal

Voor een kommagetal kan je kiezen voor het gegevenstype **decimal**, dus een decimaal getal.

Je zou hiervoor ook een **float** kunnen gebruiken, maar er zijn nadelen verbonden aan dit gegevenstype.

Eens je berekeningen gaat maken met een dergelijk veld, ontstaan er **afrondingsproblemen**.

#### Een voorbeeld:

Hieronder zie je een table met als tweede kolom een floatgetal en als derde kolom een decimaal getal. Elke rij krijgt dezelfde waarde, nl. 0,1.

	id	floatgetal	decimalgetal
▶	1	0.1	0.10
	2	0.1	0.10
	3	0.1	0.10
	4	0.1	0.10
	5	0.1	0.10
	6	0.1	0.10
	7	0.1	0.10
	8	0.1	0.10
	9	0.1	0.10
	10	0.1	0.10

Als je voor beide kolommen een totaal gaat berekenen zie je al een probleem. In de kolom met het floatgetal ontstaat er een afrondingsfout.

	sum(floatgetal)	sum(decimalgetal)
▶	1.0000000149011612	1.00

Naarmate de table meer rijen bevat, wordt je afrondingsprobleem steeds groter.

**Je kiest dus best altijd voor het gegevenstype decimal.**

### 8.2.4. Een nieuwe table op basis van geselecteerde records

Je kan ook een nieuwe table creëren op basis van gegevens uit een andere table. Hiervoor gebruik je de `as select ...` instructie binnen je `Create table`.

De velden in de nieuwe table krijgen automatisch hetzelfde gegevenstype en dezelfde veldlengte als in de onderliggende table(s).

Met deze instructie kan je records archiveren, reservekopieën maken,...

Syntax:

```
CREATE TABLE nieuwetable
AS SELECT veld1[,veld2[,...]]
FROM bron
```

De instructie bevat volgende onderdelen:

Onderdeel	Betekenis
<b>veld1, veld2</b>	Dit zijn de namen van de velden die naar de nieuwe table gekopieerd moeten worden.
<b>nieuwetable</b>	De naam van de table die gemaakt moet worden. Als de naam van de nieuwe table gelijk is aan de naam van een bestaande table, krijg je een fout.
<b>bron</b>	De naam van een bestaande table waaruit records geselecteerd worden. Dit kan <ul style="list-style-type: none"> <li>• één tablenaam zijn</li> <li>• of een samenstelling met <code>inner join</code>, <code>left join</code> of <code>right join</code></li> <li>• of een opgeslagen query.</li> </ul>

### 8.2.5. Voorbeelden

Probeer onderstaande voorbeelden uit in de database BIEREN.

Als de aanpassing gelukt is, krijg je onderaan het scherm weer een boodschap.

#### Voorbeeld 1

Maak een nieuwe table *klanten* met de velden *klantnr*, *knaam*, *kladres*, *klpost* en *klgemeente*. De velden *klntnr* en *knaam* moeten een waarde bevatten.

## Oplossing

```
CREATE TABLE klanten
(klantnr integer not null, klnaam varchar(30) not null, kladres varchar(40), klpost char(4),
klgemeente varchar(40))
```

Output			
Action Output			
	Time	Action	Message
✓	1 14:51:16	CREATE TABLE klanten (klantnr integer not null, klnaam char(30) not null, kladres char(40), klpo...	0 row(s) affected

## Voorbeeld 2

Maak een table *alcoholarm*. Plaats hierin de gegevens van de alcoholarme bieren. Neem naam van de brouwer en naam van het bier mee.

## Oplossing

```
CREATE TABLE alcoholarm
AS
SELECT naam, brnaam
FROM bieren
INNER JOIN brouwers ON bieren.brouwernr=brouwers.brouwernr
INNER JOIN soorten ON bieren.soortnr=soorten.soortnr
WHERE soort='Alcoholarm'
```

Output			
Action Output			
	Time	Action	Message
✓	1 15:04:23	CREATE TABLE klanten (klantnr integer not null, klnaam char(30) not null, kladres char(40), klpo...	0 row(s) affected
✓	2 15:05:22	CREATE TABLE alcoholarm AS SELECT naam, brnaam FROM (bieren inner join brouwers on bie...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

Aan de linkerkant zie je nu dat er een table *alcoholarm* bij gekomen is:

The screenshot shows a database management interface. On the left, under 'SCHEMAS', the 'bieren' schema is expanded, showing a list of tables: 'alcoholarm', 'bieren', 'bieren\_oud', 'brouwers', 'klanten', and 'soorten'. The 'alcoholarm' table is highlighted. In the main area, a SQL query is entered: 'SELECT \* FROM alcoholarm'. Below the query, a 'Result Grid' is displayed with the following data:

naam	brnaam
Alfri	Roman
Brouwmeester alcohol vrij bier	Alken
Fitt	Wieze
Star blond	Haacht
Star bruin	Haacht

At the bottom, a tab labeled 'alcoholarm 1' is visible.

### 8.3. Table verwijderen

In dit onderdeel bekijken we hoe je een bestaande table kan verwijderen.

### 8.3.1. Syntax

De DELETE-instructie kan je, zoals reeds gezegd, gebruiken om geselecteerde rijen uit een table te verwijderen. Hiermee is de table nog niet weg.

Met de **drop**-instructie kan je een bestaande table volledig uit de database verwijderen.

Syntax:

DROP TABLE [table]

De instructie bevat volgende onderdelen:

Onderdeel	Betekenis
<b>table</b>	De naam van de table die je wil verwijderen.

### 8.3.2. Voorbeelden

Probeer onderstaand voorbeeld uit in de database BIEREN.

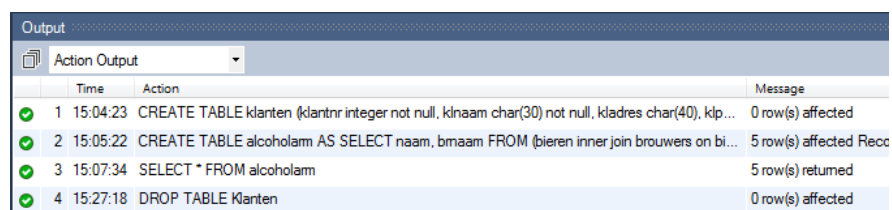
Ook met de DROP-instructie kan je enkel onderaan in het venster Output zien of je instructie goed uitgevoerd is. De table zal verwijderd worden aan de linkerkant van het scherm (misschien moet je je scherm even refreshen).

#### Voorbeeld 1

Verwijder de table *klanten*.

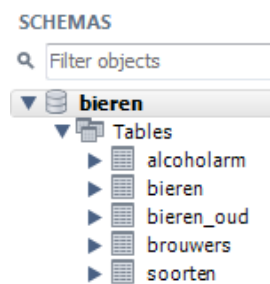
*Oplossing*

DROP TABLE klanten



Output			
Action Output			
	Time	Action	Message
✓	1 15:04:23	CREATE TABLE klanten (klantnr integer not null, klnaam char(30) not null, kladres char(40), klp...	0 row(s) affected
✓	2 15:05:22	CREATE TABLE alcoholarm AS SELECT naam, bmaam FROM (bieren inner join brouwers on bi...	5 row(s) affected Recd
✓	3 15:07:34	SELECT * FROM alcoholarm	5 row(s) returned
✓	4 15:27:18	DROP TABLE klanten	0 row(s) affected

Aan de linkerkant zie je nu dat de table *klanten* verdwenen is:



## 8.4. Table wijzigen

Hoe je de structuur van de table kan aanpassen, leggen we in dit onderdeel uit.

### 8.4.1. Syntax

Wanneer een table gemaakt is, zijn er veel scenario's mogelijk waardoor je de structuur van je tables wil **wijzigen**.

We beginnen met het **toevoegen** en **verwijderen** van een kolom. Het maken van een primaire sleutel en indexen komt verder in deze cursus aan bod.

#### Veld toevoegen

```
ALTER TABLE table
ADD [COLUMN] veld type [(grootte)] [NOT NULL], veld2 type [(grootte)] [NOT NULL],..., veldn type
[(grootte)] [NOT NULL]
```

Deze instructie bevat volgende onderdelen:

Onderdeel	Betekenis
<b>table</b>	Naam van de table die je wil wijzigen
<b>veld</b>	Naam van het nieuwe veld
<b>type</b>	Gegevenstype van het nieuwe veld
<b>grootte</b>	De veldgrootte in tekens. Enkel voor tekst- en binaire velden De grootte staat altijd tussen ronde haakjes ( )
<b>not null</b>	Bij het invoeren van nieuwe records mogen enkel geldige gegevens ingevoerd worden in dit veld

#### Veld verwijderen

```
ALTER TABLE table
DROP [COLUMN] veld1, DROP veld2,... veldn
```

De betekenis van de verschillende onderdelen is hetzelfde als bij het toevoegen van een veld.

### 8.4.2. Voorbeelden

Probeer onderstaande voorbeelden uit in de database BIEREN.

Ook met de ALTER-instructie kan je enkel onderaan in het venster Output zien of je instructie goed uitgevoerd is.

#### Voorbeeld 1

Voeg een veld Opmerkingen toe aan de table *brouwers*.

##### Oplossing

```
ALTER TABLE brouwers
ADD Opmerkingen varchar(25)
```

#### Voorbeeld 2

Voeg een veld Contactpersoon toe aan de table *brouwers*. Zorg dat hier altijd iets ingevuld moet worden.



*Oplossing*

```
ALTER TABLE brouwers
ADD Contactpersoon varchar(25) NOT NULL
```

**Voorbeeld 3**

Verwijder de velden Opmerkingen en Contactpersoon uit de table *brouwers*.

*Oplossing*

```
ALTER TABLE brouwers
DROP Opmerkingen, DROP Contactpersoon
```

**8.5. Sleutels**

Naast toevoegen en verwijderen van velden, kan je ook beperkingen opleggen aan je table zoals het gebruik van sleutels.

In dit onderdeel bekijken we hoe je sleutels en relaties maakt en verwijdert. We maken hiervoor gebruik van de component **CONSTRAINT**.

**8.5.1.Constraint**

De **CONSTRAINT**-component laat je toe om

- primaire sleutels te maken,
- vreemde sleutels te maken,
- relaties te definiëren,
- referentiële integriteit af te dwingen.

Deze **CONSTRAINT** gebruik je altijd binnen een **CREATE TABLE**- of **ALTER TABLE**-instructie.

Er zijn **2 types** **CONSTRAINT**-component:

- één om een beperkende voorwaarde in te stellen op één veld.  
Deze plaats je in de velddefinitie van de **ALTER TABLE** of direct na het gegevenstype bij **CREATE TABLE**.
- één om een beperkende voorwaarde in te stellen op meerdere velden.

**8.5.2.Beperkingen**

Met een **CONSTRAINT** kan je verschillende beperkingen instellen op één of meerdere velden:

**Unique**

Hiermee kan je een veld aanwijzen als **een unieke index**. Dit kan voor elk veld van je table toegepast worden.

Unique voorkomt dat twee records in de table dezelfde waarde hebben voor een bepaald veld.

Als de unieke index toegepast wordt op meerdere velden, betekent dit dat de combinatie van de waarden in die velden uniek moet zijn.

Ga uit van volgende table:

Veld 1	Veld 2	Veld 3
1	2	1
1	1	2
2	3	2

### Situatie 1: unique op veld 2

Het record met veld 1=1, veld 2=4 en veld 3=2 kan toegevoegd worden aan de table.

Een record met veld 1=1, veld 2=1 en veld 3=2 kan niet toegevoegd worden aan de table omdat er reeds een record is met 1 als waarde in veld 2.

### Situatie 2: unique op veld 1 en veld 2

Het record met veld 1=1, veld 2=3 en veld 3=1 kan toegevoegd worden aan de table.

Een record met veld 1=1, veld 2=1 en veld 3=3 kan niet toegevoegd worden omdat deze combinatie van veld 1 en veld 2 reeds bestaat in de table.

### Primary key

Met de gereserveerde woorden **PRIMARY KEY** kan je één veld of een groep van velden in een table aanwijzen als primaire sleutel.

Alle waarden in de primaire sleutel moeten **uniek** zijn en **niet Null** zijn.

Per table kan je slechts één primaire sleutel definiëren.

Het veld met de primaire sleutel is ook vaak een autonummeringsveld. Je kan dit realiseren door de optie **auto\_increment** toe te voegen aan de definitie van je veld.

Let op: Dit kan alleen als je van dit veld ook een primaire sleutel maakt!

### Foreign key

Een **FOREIGN KEY** is een veld (of een combinatie van velden) dat verwijst naar de primaire sleutel van een andere table.

Deze foreign key dient om de **referentiële integriteit** van de gegevens te garanderen. Met andere woorden: enkel waarden die in de primaire sleutel bepaald zijn kan je als waarde gebruiken in de foreign key.

Stel je hebt twee tables: een table Klanten met de klantgegevens en een table Orders met hun bestellingen.

Table Klanten		Table Orders	
Veld	Kenmerk	Veld	Kenmerk
KlantId	Primaire sleutel	Ordernr	Primaire sleutel
Voornaam		OrderDatum	
Familienaam		Klantnr	Foreign key
Adres		Bedrag	

Het veld Klantnr in de table Orders is een foreign key. Deze is gelinkt aan de primaire sleutel KlantId in de table Klanten.

### 8.5.3. Syntax

Om sleutels en relatie te definiëren heb je volgende syntax nodig:

#### Basisstructuur

```
CREATE TABLE table
(
  veldbeschrijving1
  [, veldbeschrijving2 [,...]]
  [, tableconstraintbeschrijving[,...]]
)
```

#### Veldbeschrijving:

```
Veldnaam type [(grootte)] [NOT NULL] [INDEX [indexnaam]][AUTO_INCREMENT]
[ [CONSTRAINT constraintnaam]
{UNIQUE | PRIMARY KEY | FOREIGN KEY REFERENCES tablenaam (primaryKeyNaam)} ]
```

#### Tableconstraint:

```
CONSTRAINT constraintnaam
{ UNIQUE (veldnaam1 [, veldnaam2 [,...]])
| PRIMARY KEY (veldnaam1 [, veldnaam2 [,...]])
| FOREIGN KEY (veldnaam1 [, veldnaam2 [,...]])
REFERENCES tablenaam (primaryKeyNaam1 [,primaryKeyNaam2 [,...]])
}
```

Deze deelinstructions bevatten volgende onderdelen:

Onderdeel	Betekenis
<b>constraintnaam</b>	De naam van de beperkende voorwaarde
<b>veldnaam</b>	De naam van het veld waaraan de constraint van toepassing is
<b>indexnaam</b>	Naam van de index. Meer hierover in het onderdeel Indexen.
<b>tablenaam</b>	De naam van de table waarmee je een relatie wilt leggen. Dit noemen we de refererende table.
<b>primaryKeyNaam</b>	De naam van het veld in de refererende table. Dit veld is de primaire sleutel of een deel van de primaire sleutel in de referende table.

### 8.5.4. Voorbeelden

Probeer onderstaande voorbeelden uit in de database Bieren.

Je kan enkel onderaan in het venster Output zien of je instructie goed uitgevoerd is.

#### Voorbeeld 1

Maak een table *klanten* met de velden *klantId* en *klantnaam*.

*KlantId* is de primaire sleutel.

De naam mag maximaal 30 tekens lang zijn en moet altijd ingevuld worden.

#### Oplossing

```
CREATE TABLE klanten (
  klantId integer not null auto_increment,
  klantnaam varchar(30) not null,
  CONSTRAINT pk_klantId primary key (klantId))
```

De beperkende voorwaarde krijgt hier de **naam** *pk\_klantId*:

- *pk* want het gaat om een primaire sleutel
- op het veld *klantId*

Na de naam komt het sleutelwoord **primary key** omdat we een primaire sleutel willen maken.

### Voorbeeld 2

Maak een table *gebruikers* met de velden nummer, naam en userId.

De inhoud van het veld userId moet een unieke waarde hebben maar is toch niet de primaire sleutel.

Het veld nummer moet de primaire sleutel zijn.

#### Oplossing

```
CREATE TABLE gebruikers (
  nummer integer not null,
  naam varchar(30),
  userId varchar(8),
  CONSTRAINT pk_nr primary key (nummer),
  CONSTRAINT u_userid unique (userId))
```

Hier hebben we 2 constraint-instructies:

- CONSTRAINT *pk\_nr* primary key  
Hiermee maken we een **primaire sleutel** met de naam *pk\_nr* voor het veld nr.
- CONSTRAINT *u\_userid* unique  
Met deze instructie maken we een **unieke index** met de naam *u\_userid* op het veld userId.

### Voorbeeld 3

Maak een table *bestellingen* met de velden bestelId, klantnummer, besteldatum.

De primaire sleutel is het veld bestelId.

Tussen het veld klantnummer van deze table en het veld klantId van de table *klanten* wordt een 1-op-veel-relatie gedefinieerd.

#### Oplossing

```
CREATE TABLE bestellingen (
  bestelId integer,
  klantnummer integer,
  besteldatum datetime,
  CONSTRAINT pk_bestelId primary key (bestelId),
  CONSTRAINT f_klantnummer foreign key (klantnummer) references klanten (klantId))
```

Ook hier hebben we 2 constraint-instructies:

- *CONSTRAINT pk\_besteld primary key*  
Hiermee maken we een **primaire sleutel** met de naam `pk_besteld` voor het veld `besteld`.
- *CONSTRAINT f\_klantnummer references klanten (klantId)*  
Met deze instructie maken we een **externe sleutel (foreign key)** met de naam `f_klantnummer` op het veld `klantnummer`. Dit veld wordt gelinkt aan `klantId` (de primaire sleutel) in de table *klanten*.

#### Voorbeeld 4

Maak een table *bestellijnen* met de velden: `bestelnummer`, `biernummer` en `aantal`.  
Definieer een samengestelde sleutel met de velden `bestelnummer` en `biernummer`.

*Oplossing*

```
CREATE TABLE bestellijnen
(bestelnummer integer, biernummer integer, aantal integer, CONSTRAINT pk_bestbier
primary key (bestelnummer, biernummer))
```

#### Voorbeeld 5

Definieer een 1-op-veel-relatie tussen de tables *bestellingen* (`bestelId`) en *bestellijnen* (`bestelnummer`).

*Oplossing*

```
ALTER TABLE bestellijnen
ADD CONSTRAINT f_bestelnummer foreign key (bestelnummer) references bestellingen
(bestelId)
```

#### Voorbeeld 6

Verwijder de beperkende voorwaarde `f_bestelnummer`.

*Oplossing*

```
ALTER TABLE bestellijnen
DROP FOREIGN KEY f_bestelnummer
```

#### Voorbeeld 7

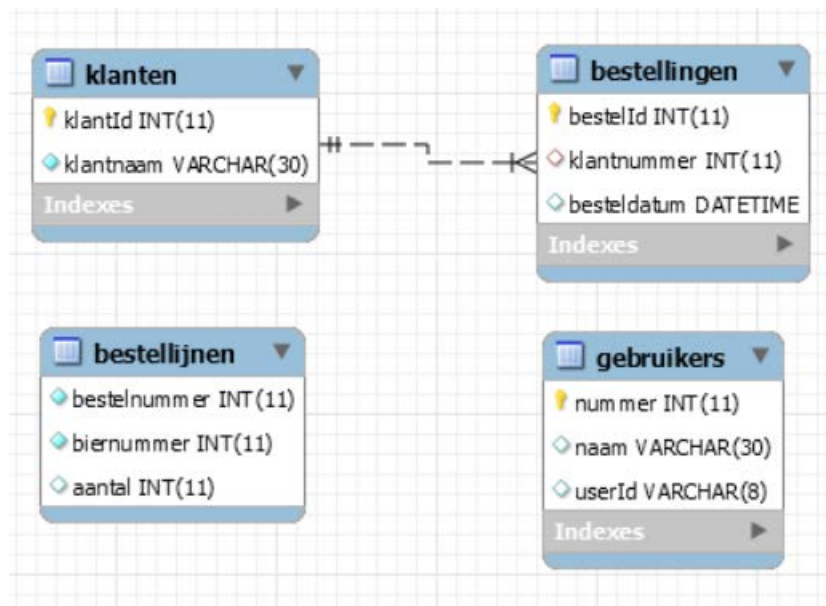
Verwijder de primaire sleutel van de table *bestellijnen*.

*Oplossing*

```
ALTER TABLE bestellijnen
DROP PRIMARY KEY
```

#### Eindresultaat

Als je het relatieschema opvraagt na deze 7 voorbeelden, krijg je dit resultaat:



### 8.5.5. Indexen

Tenslotte leggen we in dit onderdeel uit

- wat indexen zijn
- hoe je een index kan maken
- hoe je een index kan verwijderen.

### 8.5.6. Algemeen

#### Waarom een index?

Sommige SQL-instructies hebben een redelijk **constante verwerkingstijd**. Het maakt niet uit onder welke omstandigheden deze uitgevoerd worden, het zal altijd even lang duren. Je kan de verwerkingstijd op geen enkele manier beïnvloeden. De CREATE TABLE instructie is hier een voorbeeld van.

Dit geldt niet voor alle instructies. De verwerkingstijd van SELECT, UPDATE en DELETE-instructies kan erg **variëren**: van enkele seconden tot enkele minuten. Deze tijd kan je wel beïnvloeden.

Het aan- of afwezig zijn van **indexen** kan een grote invloed hebben op die verwerkingstijden.

#### Werking van een index

SQL heeft verschillende manieren om tables te benaderen.

De **sequentiële zoekmethode** is het 'rij per rij doorlopen van een table'. Als je slechts één rij zoekt in een table met enkele duizenden rijen, zal dit erg lang duren.

Met de **geïndexeerde zoekmethode** benadert SQL alleen die rijen die bepaalde kenmerken hebben. Om dit te kunnen doen heeft SQL indexen nodig.

### 8.5.7. Waarom? - Voorbeeld

Stel je wil weten hoe je groene pepers kweekt. Je kan dit opzoeken in een boek over tuinieren. In plaats van het volledige boek te doorbladeren tot je een deel vindt dat over pepers gaat, is het veel makkelijker naar de inhoudstafel te gaan aan het einde van het boek, de pagina's te vinden met informatie over pepers, en rechtstreeks naar deze pagina's te gaan.

Je bespaart tijd door eerst de index te raadplegen. Het is een doeltreffende methode om de gewenste informatie te vinden.

Een ander voorbeeld is het opzoeken van een telefoonnummer. Je gaat het telefoonboek niet pagina per pagina doorlopen. Je gaat bijvoorbeeld ineens naar de juiste gemeente en de juiste beginletter.

Hetzelfde principe geldt om gegevens op te halen in een databasetabel.

Zonder index neemt het databasesysteem de volledige table door (dit proces wordt een 'tablescan' genoemd) om de gewenste informatie te vinden.

Dankzij de juiste index kan het databasesysteem eerst de index doornemen om te zien waar het de gegevens kan ophalen, om deze vervolgens rechtstreeks op de juiste plaatsen te vinden. Dit is veel sneller.

### 8.5.8. Syntax

#### Index definiëren

```
CREATE [UNIQUE] INDEX naam
ON table (veld [asc | desc] [, veld [asc | desc] [...]])
```

#### Index verwijderen

```
DROP INDEX naam ON table
```

Deze instructies bevatten volgende onderdelen:

Onderdeel	Betekenis
<b>naam</b>	De naam van de index
<b>table</b>	De naam van de (bestaande) table waarvoor je een index maakt
<b>veld</b>	De naam van het veld of de velden die je in de index wil opnemen Voor elk veld kan je aangeven of er oplopend (ASC) of aflopend (DESC) gesorteerd moet worden. Standaard zal dit oplopend zijn.

**Dubbele waarden** in het geïndexeerde veld kan je voorkomen met de het gereserveerde woord **unique**.

### 8.5.9. Effect van een index

Om het effect van een index aan te tonen, gaan we een SQL-statement uitvoeren. Als we een index maken, dan zal hetzelfde SQL-statement nadien minder tijd vragen.

Om dit uit te testen hebben we een grote table nodig. Voer onderstaand script uit (dit vraagt wel wat tijd). Je krijgt dan een database METINDEXES met daarin een table *GroteTabel*. Deze table bevat 5 miljoen records.

De table GroteTabel bevat 3 kolommen:

- a - primaire sleutel, dus geïndexeerd
- b - geïndexeerd
- c - niet geïndexeerd

Probeer de drie volgende instructies uit. Kijk goed naar de tijd nodig om deze uit te voeren. Wat stel je vast?

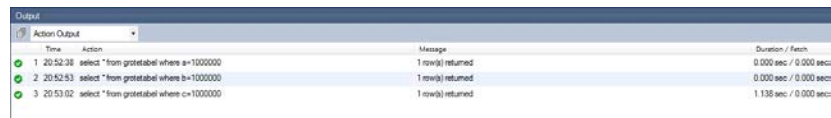
```
select * from grotetabel where a=1000000
```

```
select * from grotetabel where b=1000000
```

```
select * from grotetabel where c=1000000
```

**Antwoord**

De derde instructie neemt beduidend meer tijd in beslag omdat kolom c niet geïndexeerd is.



Time	Action	Message	Duration / Fetch
1 20:52:38	select * from grotetabel where a=1000000	1 row(s) returned	0.000 sec / 0.000 sec
2 20:52:53	select * from grotetabel where b=1000000	1 row(s) returned	0.000 sec / 0.000 sec
3 20:53:02	select * from grotetabel where c=1000000	1 row(s) returned	1.136 sec / 0.000 sec

## 8.5.10. Voorbeelden

Probeer onderstaande voorbeelden uit in de database BIEREN.

Je kan enkel onderaan in het venster Output zien of je instructie goed uitgevoerd is.

### Voorbeeld 1

Maak een index voor het veld *knaam* van de table *klanten*.

*Oplossing*

```
CREATE INDEX i_naam
ON klanten (klientnaam)
```

### Voorbeeld 2

Verwijder de index *i\_naam* van de table *klanten*.

*Oplossing*

```
DROP INDEX i_naam ON klanten
```

## 8.6. Opgaven

In dit onderdeel krijg je een hele reeks oefeningen op het aanpassen van je tables.

Ook deze kan je uittesten in de database PLANTEN. De oplossingen vind je in de webversie.

### 8.6.1. Opgave 1

Creëer een table *klachten* met de volgende velden:



veld	data-type	lengte
klachtId	numeriek	
plantId	numeriek	
datum	datum	
klacht	karakter	100
status	karakter	2

Voeg nadien volgende klacht toe aan deze table:

Klacht nr. 1 heeft betrekking op plantid 10. De klacht luidt : “Planten (leveranciercode 2) verkocht op 14-1-2016 vertoonden bruine vlekken na circa 2 maanden”.

De status is GL (gemeld aan leverancier) en de klacht wordt ingebracht op 15-3-2016.

### 8.6.2.Opgave 2

Er is behoefte aan een table *aanbied* waaruit snel de goedkoopste leveranciers van de planten uit de table *planten* kunnen worden opgezocht.

De table heeft de volgende kolommen : plantId, plantNaam, leverancierid en offerteprijs.

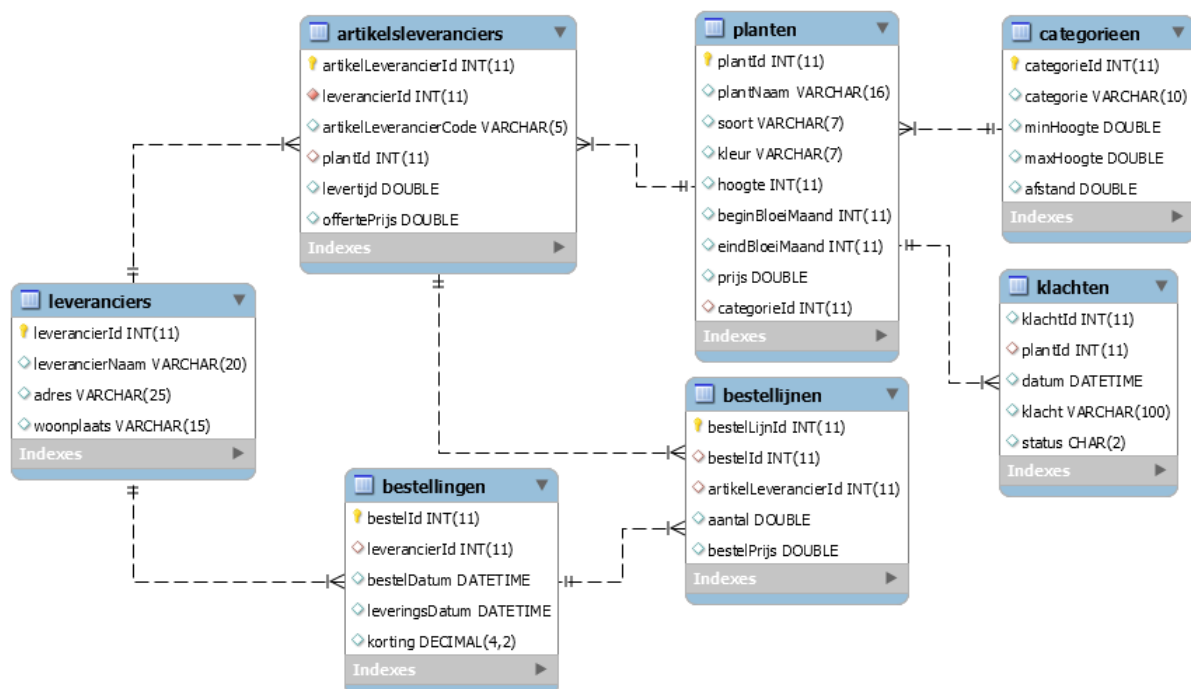
Vul deze table vanuit de tables *planten* en *artikelsleveranciers*.

### 8.6.3.Opgave 3

Breid de table *planten* uit met een kolom voorraad om het aantal stuks dat het tuincentrum nog in voorraad heeft bij te houden.

### 8.6.4.Opgave 4

Definieer alle relaties tussen de tables van de database Planten.



### 8.6.5. Opgave 5

Maak een index BSRIDX01 op de samengestelde sleutel besteld en artikelLeverancierId van de table *bestellijnen*.

### 8.6.6. Opgave 6

De table artikelsleveranciers wordt regelmatig gejoined met de tables *planten*, *leveranciers* en *bestellijnen*.

Welke indexen zijn waardevol voor de table *artikelsleveranciers*?

## Hoofdstuk 9. Views

### 9.1. Wat zijn views?

In dit hoofdstuk bekijken we wat views zijn en hoe je ze kan gebruiken.

#### 9.1.1. Definitie

SQL kent twee soorten tables: echte tables, meestal de basistables genoemd, en afgeleide tables of views.

De **basistable** is de enige tabelvorm waarin gegevens werkelijk opgeslagen kunnen worden. In onze database Bieren zijn dit bijvoorbeeld *bieren*, *brouwers* en *soorten*.

Een afgeleide table of **view** bevat zelf geen rijen. Het is een soort voorschrift om gegevens uit de basistables in een 'virtuele' table samen te voegen. 'Virtueel' omdat de view enkel bestaat als deze opgeroepen wordt in een instructie (bijvoorbeeld in Access of in een webapplicatie).

Je kan een view ook bekijken als een SELECT-instructie die bewaard wordt.

Als je de gegevens van een view wil tonen, wordt de SELECT-instructie terug opnieuw uitgevoerd.

#### 9.1.2. Toepassing

Een veel voorkomende toepassing van views is het afschermen van bepaalde kolommen om zo **beveiliging** te implementeren.

Zo bestaat een personeelstable uit allerlei velden waaronder:

- privéadres
- rijksregisternummer
- salaris
- ...

Door verschillende views te maken die deze gegevens wel of juist niet bevatten en deze te beveiligen kan:

- de loonadministratie alle gegevens zien;
- personeelszaken de gegevens zien en bewerken;
- het management alleen de salarissen zien;
- kunnen overige medewerkers geen van deze gegevens zien.

Voor de vier groepen worden aparte views gedefinieerd. Geen van de groepen heeft toegang tot de basistable.

### 9.2. Maken en verwijderen

In dit onderdeel bekijken we hoe je een view kan maken en verwijderen.

### 9.2.1. Maken

Om een view te maken, heb je volgende syntax nodig:

```
CREATE VIEW viewnaam
AS
SELECT-instructie
```

### 9.2.2. Verwijderen

Om een view te verwijderen, heb je volgende syntax nodig:

```
DROP VIEW viewnaam
```

### 9.2.3. Voorbeelden

Probeer onderstaande voorbeelden uit in de database BIEREN.

Ook hier zie je in het venster Output of je instructie goed uitgevoerd werd.

Aan de linkerkant kan je de view ook zien.

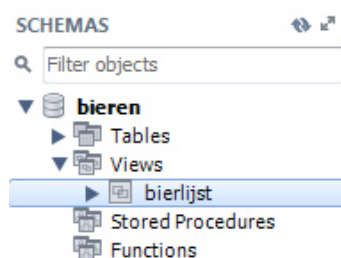
#### Voorbeeld 1: View maken

Maak een view *bierlijst*. Op deze lijst staat de naam van het bier, de naam van de brouwer en het soort bier.

*Oplossing*

```
CREATE VIEW bierlijst
AS
SELECT naam, brnaam, soort
FROM bieren
INNER JOIN brouwers ON bieren.brouwernr = brouwers.brouwernr
INNER JOIN soorten ON bieren.soortnr=soorten.soortnr
```

Aan de linkerkant zie je bij Views je nieuwe view:



#### Voorbeeld 2: View bekijken

Bekijk de inhoud van de view *bierlijst*.

*Oplossing*

```
SELECT *
FROM bierlijst
```

```

1 • SELECT *
2 FROM bierlijst

```

naam	brnaam	soort
A.C.O.	Steedje	Extra
Aalbeeks St. Corneliusbier (=Kapittel pater (Het))	Van Eecke	Extra
Aardbeien witbier	Huyghe	Tarwebier of witbier
Aarschots kruikenbier (=St. Sebastiaan grand cru)	Sterkens	Edelbier
Abt Bijbier (Nen)	Domus	Extra
Adler	Haacht	Pils
Aerts 1900	Palm	Dubbel Donker
Affigem blond (Abdij)	Smedt (De)	Lichtblond
Affigem christmas ale (Abdij)	Smedt (De)	Massieve Ale
Affigem dubbel (Abdij)	Smedt (De)	Dubbel Donker

bierlijst 1 x

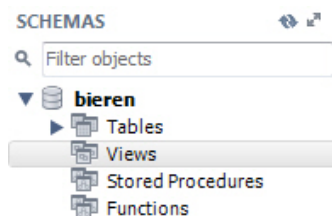
### Voorbeeld 3: View verwijderen

Verwijder de view *bierlijst*.

*Oplossing*

DROP VIEW bierlijst

Aan de linkerkant zie je dat je view verdwenen is:



## 9.3. Opgaven

In dit onderdeel krijg je een hele reeks oefeningen op het gebruik van views. De oplossingen vind je in de webversie.

Ook nu kan uittesten in de database PLANTEN.

### 9.3.1.Opgave 1

Definieer een view *vastlaag* waarin alle gegevens van alle vaste planten uit de table *planten* voorkomen met een hoogte van maximaal 15 cm.

### 9.3.2.Opgave 2

Definieer een view *offerteprijzen* met de kolommen plantId, minOff, maxOff en gemOff waarin respectievelijk plantid, laagste, hoogste en gemiddelde offerteprijs vermeld is.

### 9.3.3. Opgave 3

Maak een view *zomerplanten* waarmee de gegevens plantid, plantennaam, soort en prijs zijn te benaderen van alle planten die in de maanden juni, juli en augustus beginnen te bloeien.

### 9.3.4. Opgave 4

Definieer een view *bomen* met de gegevens plantId, plantennaam, hoogte en prijs van alle bomen.

### 9.3.5. Opgave 5

Definieer een view *leverancier5* waarin alleen van leverancier 5 de volgende gegevens staan: plantId, plantNaam, artikelLeverancierCode, offertePrijs, prijs

### 9.3.6. Opgave 6

Maak een view waardoor alleen de offertegegevens van de leveranciers uit Lisse zijn te selecteren.

### 9.3.7. Opgave 7

Maak een view *besteldeplanten* die een overzicht geeft van bestelid, artikelcode van de leverancier en de plantnaam.

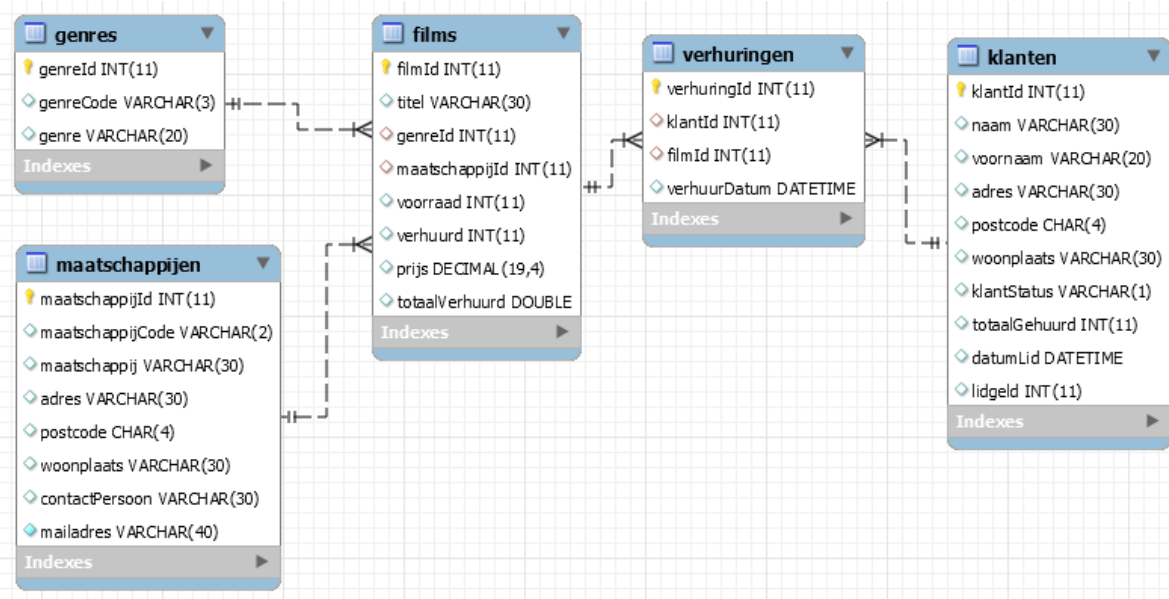
## 9.4. Opdracht voor de coach 3

Op de volgende pagina krijg je een reeks opdrachten. Gebruik hiervoor de database VIDEO.

Verzamel al je oplossingen in een tekstbestand en stuur dit door ter verbetering aan je **coach**. Vermeld als onderwerp "**Aanpassen en beheer**".

Stel de SQL-instructies op om volgende opdrachten uit te voeren. Gebruik hiervoor de database Video.

1. Voeg jezelf toe als klant.
2. Geef alle films met maatschappijCode VH een prijsverhoging van 10%.
3. Maak een table met enkel de films van het genre Thriller.
4. Voeg aan de table maatschappijen een veld mailadres toe. Dit veld mag maximaal 40 tekens lang zijn en mag niet leeg zijn.
5. Definieer alle nodige primary en foreign keys.



## Hoofdstuk 10. Einde cursus

In dit laatste hoofdstuk overlopen we nog de volgende onderdelen:

- de eindoefening
- hoe je de afdrukbare versie van deze cursus kan downloaden;
- hoe je je attest kan ontvangen

### 10.1.1. Eindoefening

Bij deze cursus hoort ook een eindoefening. In die eindoefening komen de belangrijkste zaken van deze cursus terug aan bod.

Stuur een bericht aan je **coach** met als onderwerp "**Opdracht eindoefening**".

### 10.1.2. Wat nu?

Beste cursist,

Je bent nu aan het einde gekomen van de webcursus SQL. Proficiat!

#### Enquête

We horen graag wat je van deze cursus vindt. Vul het [enquêteformulier](#) zeker in!

#### Afdrukbare cursus

Wil je een afdrukbare versie van deze cursus? Stuur dan een bericht met als onderwerp "Afdrukbare versie" naar de coach.

Deze zal je vervolgens een downloadlink bezorgen waarmee je de cursus in pdf-formaat kan downloaden.

#### Attest

Wat de administratieve afhandeling betreft: je hoeft niets te doen om de cursus af te sluiten. Dit gebeurt automatisch na het verlopen van je leerperiode. Wanneer je precies je deelnameattest ontvangt, lees je [hier](#). We wensen je nog veel succes en tot een volgende keer.

De coaches.