



**Vlaamse Dienst voor Arbeidsbemiddeling en
Beroepsopleiding**

SQL

Inhoudsopgave

1	WAT IS SQL ?	4
2	BEGRIPPEN	5
3	HET LEERPROGRAMMA.....	7
3.1	Installatie SQLTryOut.....	7
3.2	Gebruiksaanwijzing.....	8
4	DE DATABASES BIEREN EN PLANTV	14
5	GEGEVENS SELECTEREN	16
5.1	Syntax	16
5.2	Lijst opvragen.....	17
5.3	Selectie maken m.b.v. WHERE	17
5.4	Een gesorteerde lijst opvragen m.b.v ORDER BY	19
5.5	Oefeningen SELECT deel 1.....	20
5.6	Aggregate functions	21
5.7	Berekeningen maken	21
5.8	Oefening SELECT deel 2.....	22
5.9	Gegevens groeperen	22
5.10	Oefeningen SELECT deel 3.....	23
5.11	Gegevens selecteren uit meerdere tabellen tegelijkertijd	24
5.12	Outer join.....	26
5.13	Self Join	28
5.14	Select ... into	28
5.15	UNION.....	29
5.16	Oefeningen SELECT deel 4.....	30
5.17	Informatie schrijven naar en lezen uit een externe database	31
5.18	Subqueries.....	31
5.19	Gecorreleerde subqueries	32
5.20	Oefeningen SELECT deel 5.....	32

6	GEGEVENS UIT DE DATABASE AANPASSEN	34
6.1	INSERT	34
6.2	Oefening INSERT	36
6.3	UPDATE.....	36
6.4	Oefeningen UPDATE	38
6.5	DELETE	38
6.6	Oefening DELETE	39
7	BEHEER VAN TABELLEN EN RELATIES	40
7.1	CREATE TABLE	40
7.2	Oefeningen CREATE TABLE	41
7.3	DROP.....	42
7.4	ALTER.....	42
7.5	Oefeningen Alter	44
7.6	CONSTRAINT.....	44
7.7	Oefeningen CONSTRAINT	46
7.8	CREATE INDEX	47
7.9	Oefeningen CREATE INDEX.....	48
8	VIEWS	49
8.1	Create view	49
8.2	Drop view	49
8.3	Oefeningen	49
9	EINDOEFENING	51
10	COLOFON.....	53

1 WAT IS SQL ?

Structured Query Language is taal om een relationele database te beheren en te onderhouden. Databasegegevens worden beheerd door een afzonderlijk systeem : Relational Database Management System. De RDBMS zelf is verantwoordelijk voor de structuur, het bewaren en ophalen van gegevens. Met behulp van SQL geven wij opdrachten aan de RDBMS. De SQL-instructies worden onderverdeeld in drie categoriën : Data Manipulation Language, Data Definition Language, Data Control Language en Transaction Control.

De DML wordt gebruikt om gegevens te selecteren, toe te voegen, te wijzigen en te wissen en omvat de instructies select, insert, update en delete.

De DDL bestaat uit de creatie en onderhoud van databases, de tabellen, views en indexes en omvat de instructies create, drop en alter.

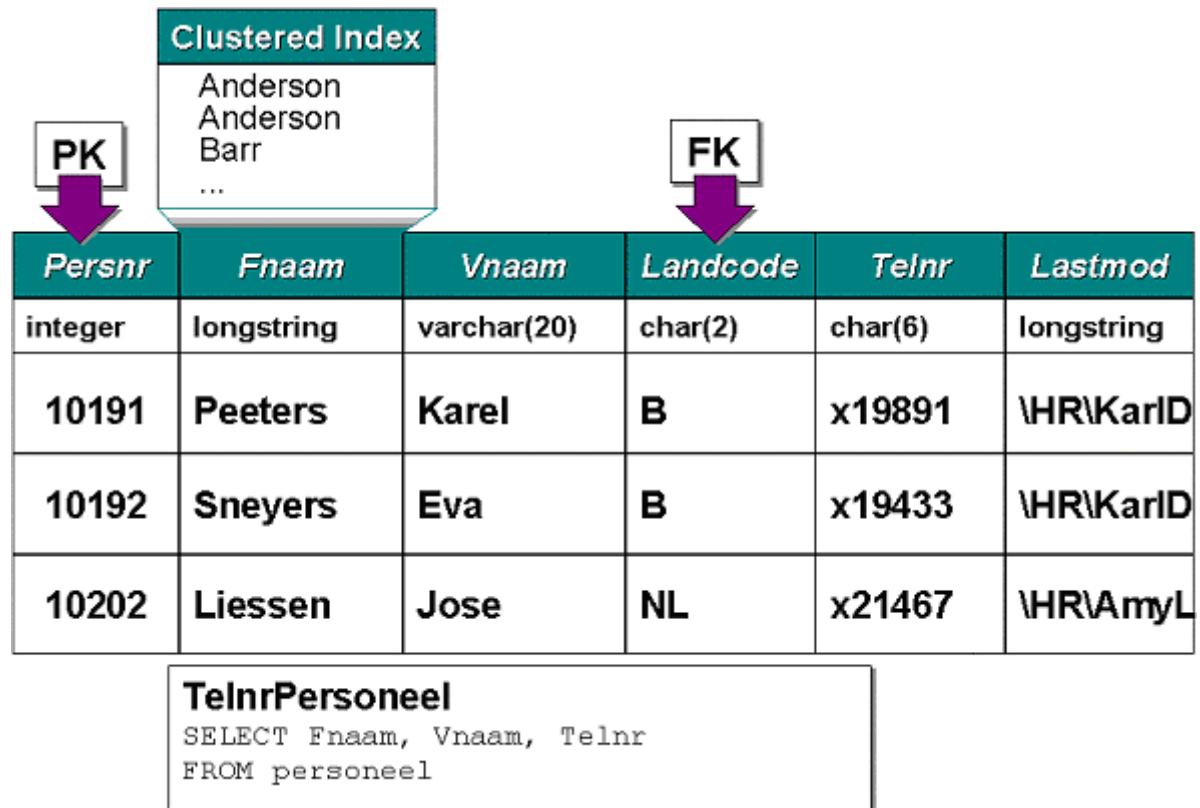
De DCL is verantwoordelijk voor de beveiliging van de gegevens. Voorbeeld : De personeelsdirecteur heeft toegang tot de lonen van de personeelsleden, de secretaresse heeft enkel toegang tot de namen en personeelsnummer. Deze instructies komen niet aanbod.

Transaction control bestuurt de transacties binnen de server. Deze instructies worden niet behandeld in deze cursus.

SQL kan gebruikt worden in programmeertalen (cobol, visual basic, perl, C ...), in tools (report writers, form generators, application programs).

SQL is een gestandaardiseerde taal en dus bruikbaar voor elke soort relationele database. Toch kunnen aan de RDBMS nog extra instructies toegevoegd zijn.

2 BEGRIPPEN



Tabellen (Tables)

Na de normalisatie bekomt u verschillende entiteiten. Deze entiteiten vormen de tabellen van de database.

Attributen (Columns)

De attributen van elke entiteit vormen de kolommen (velden) van de tabel.

Rijen (Tuples)

De gegevens worden in rijen opgeslagen. In de literatuur over relationele databases spreekt men eerder over tuples.

Primary Key (PK)

De sleutel van de tabel, deze bepaalt de rij van de tabel. Elke rij heeft een unieke sleutelwaarde. Vb. personeelsnr, artikelnr, ...

Foreign key (FK)

Dit is een attribuut in de tabel die verwijst naar een andere tabel waar deze attribuut het sleutelveld is. Vb. klantrnr in tabel facturen verwijst naar het klantrnr in de tabel klanten. Het klantrnr van de tabel klantrnr is de PK van de tabel.

Views

Is een select instructie die bewaard wordt. Wanneer we de view gebruiken wordt de select instructie opnieuw uitgevoerd. Op een view kunt u dezelfde bewerkingen uitvoeren als bij een tabel, deze bewerkingen worden dan doorgevoerd naar de onderliggende tabellen waaruit de view samengesteld is.

Voorbeeld :

TelnrPersoneel is een view die een aantal velden selecteert van de tabel Personeel. Wanneer u een rij toevoegt aan de view wordt er automatisch een rij toegevoegd aan de tabel Personeel.

Index

Op de primary key wordt automatisch een index geplaatst. Deze index kunt u het best vergelijken met een inhoudstafel waarbij elke lijn een verwijzing bevat naar de rest van de uitleg. Deze index wordt gebruikt om snel gegevens op te zoeken. U kunt nog bijkomende indexen plaatsen om snel informatie te zoeken op die kolom(men).

3 HET LEERPROGRAMMA

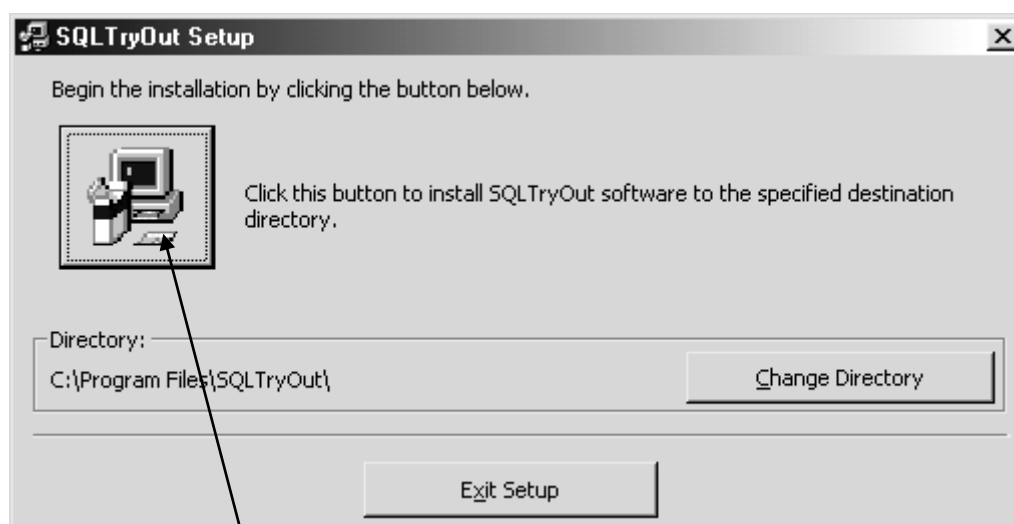
3.1 Installatie SQLTryOut

Het leerprogramma heeft u van uw instructeur mee gekregen op diskettes of cd.

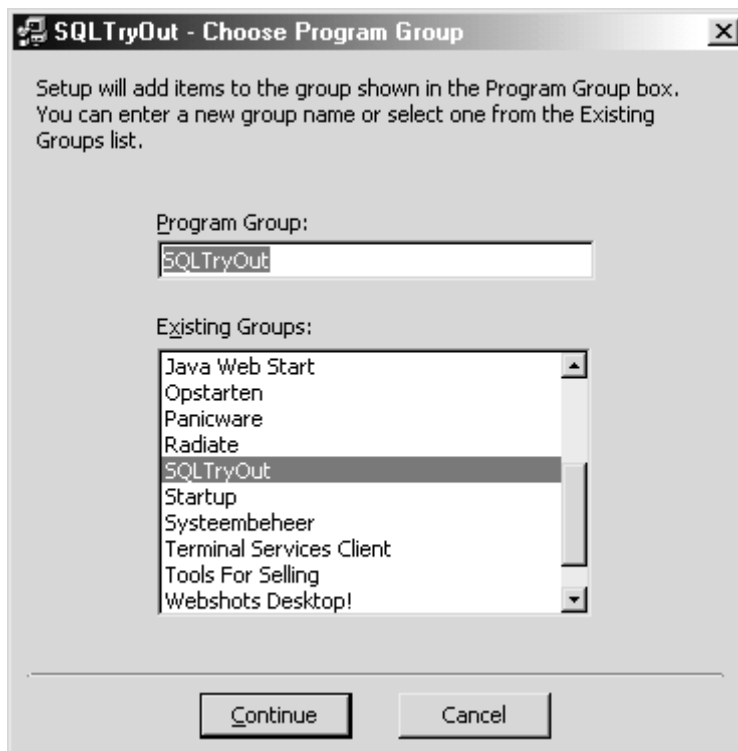
- U start het programma setup.exe van de eerste diskette of cd.
- Het setup-programma kopieert eerst enkele systeembestanden op uw toestel. Afhankelijk van de software die op uw pc geïnstalleerd staat, vraagt het programma al dan niet om de pc te herstarten. Als u die vraag krijgt is het aangewezen om op OK te klikken. Nadien start u terug het programma setup.exe.



- Klik op OK.



- Kies met de knop Change Directory de plaats waar het programma moet geïnstalleerd worden op de computer.
- Klik op de Setup-knop om de installatie te starten.




- Klik op Continue.

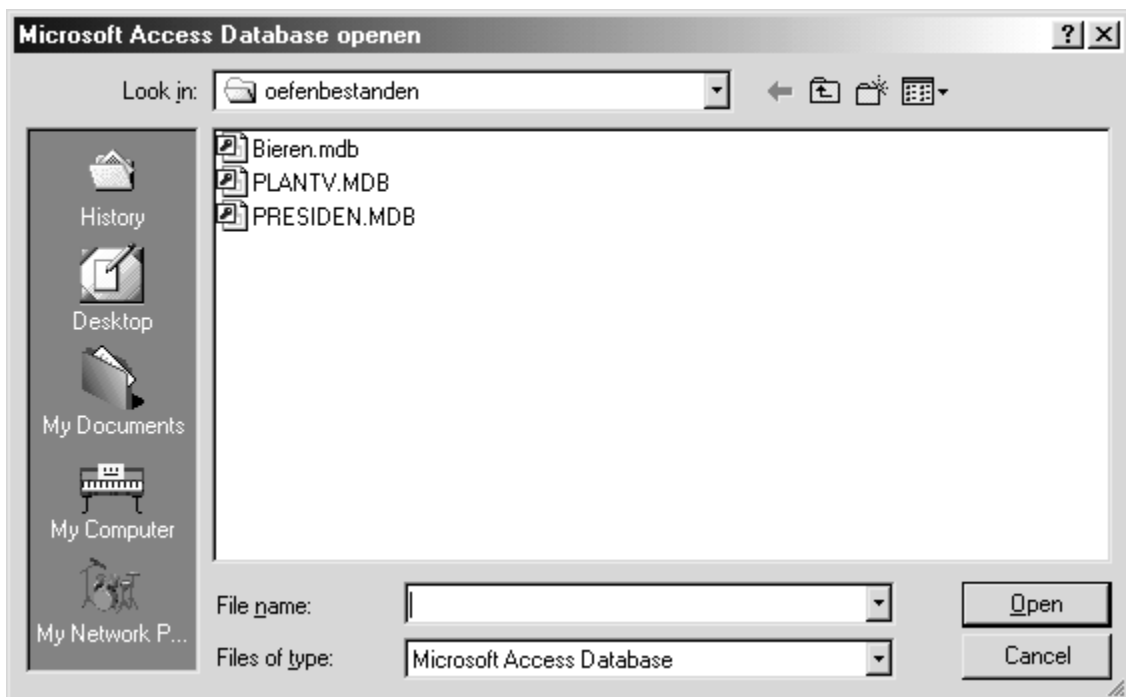
Het programma wordt geïnstalleerd.

3.2 Gebruiksaanwijzing

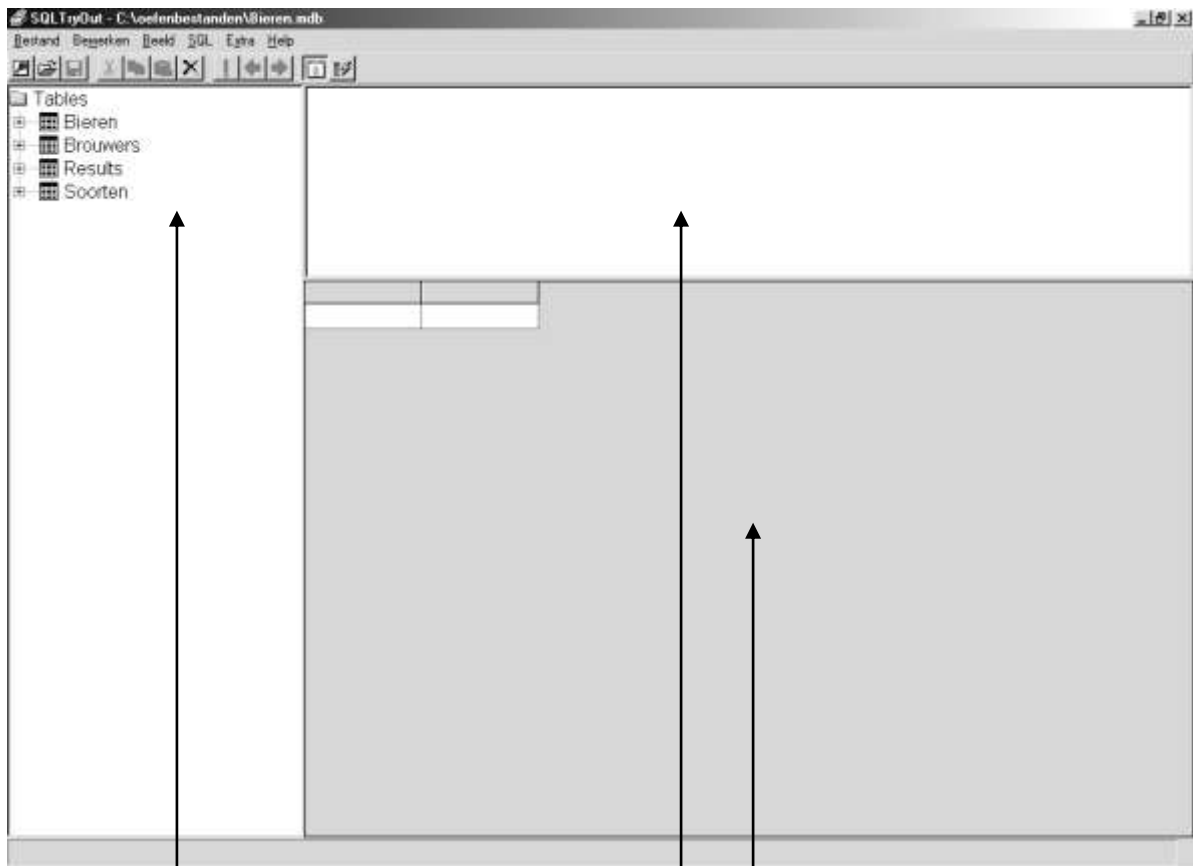
- Open de groep SQLTryOut in het Start-menu.
- Start het programma SQLTryOut.



- Eerst moet u de database selecteren, die u wilt ondervragen. Via Database openen in het menu Bestand of  kunt u de database Bieren openen.



Na de selectie van de database krijgt u dit scherm.

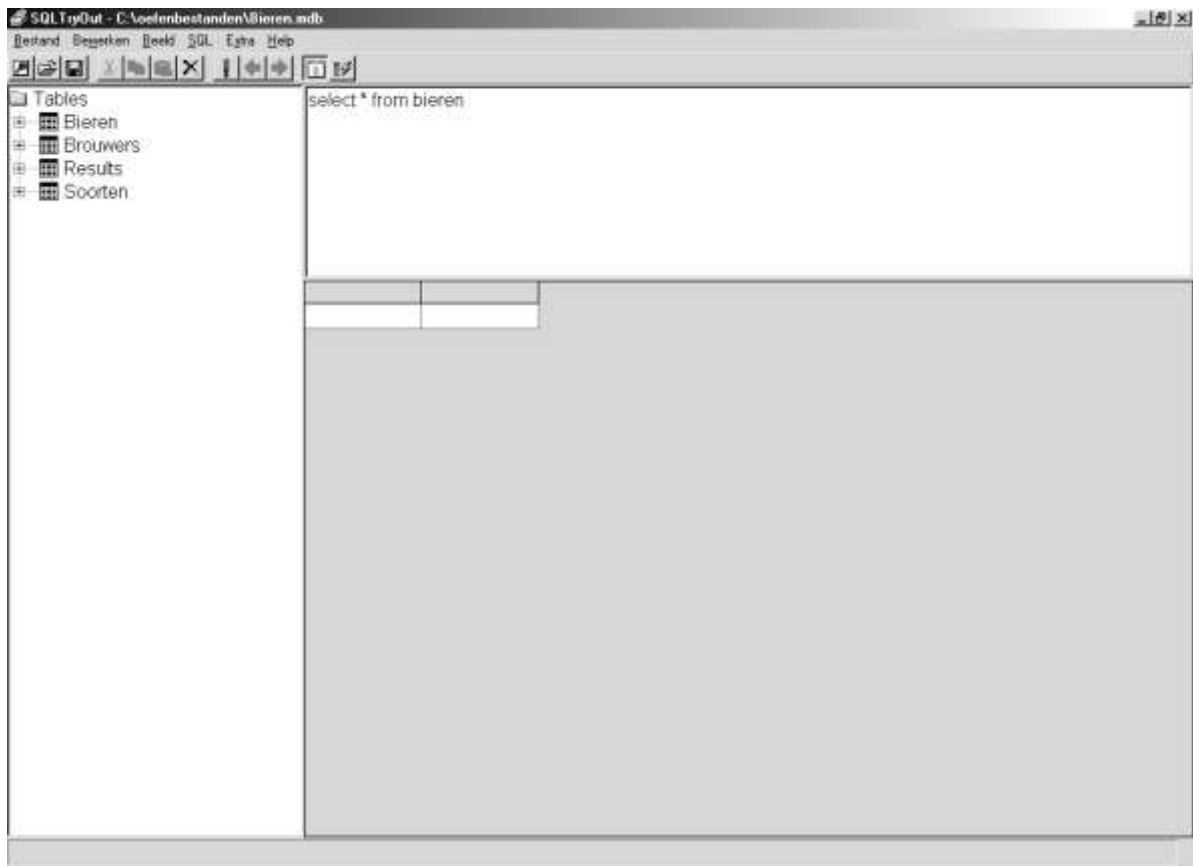



De structuur van de database

klik op het + teken om de bijbehorende velden te zien van de tabel.

In het tekstvak plaatst u de SQL-instructies.

Resultaatsvak: hier ziet u het resultaat van de uitvoering van de instructie



- Klik op de knop  of F5.

Er zijn drie mogelijke resultaatsschermen :

Het tonen van een selectie. Het aantal getoonde records staat naast het verkeerslicht.

SQL TryOut - Bieremdb

Bestand Beveiligen Beeld SQL Extra Help

Tables

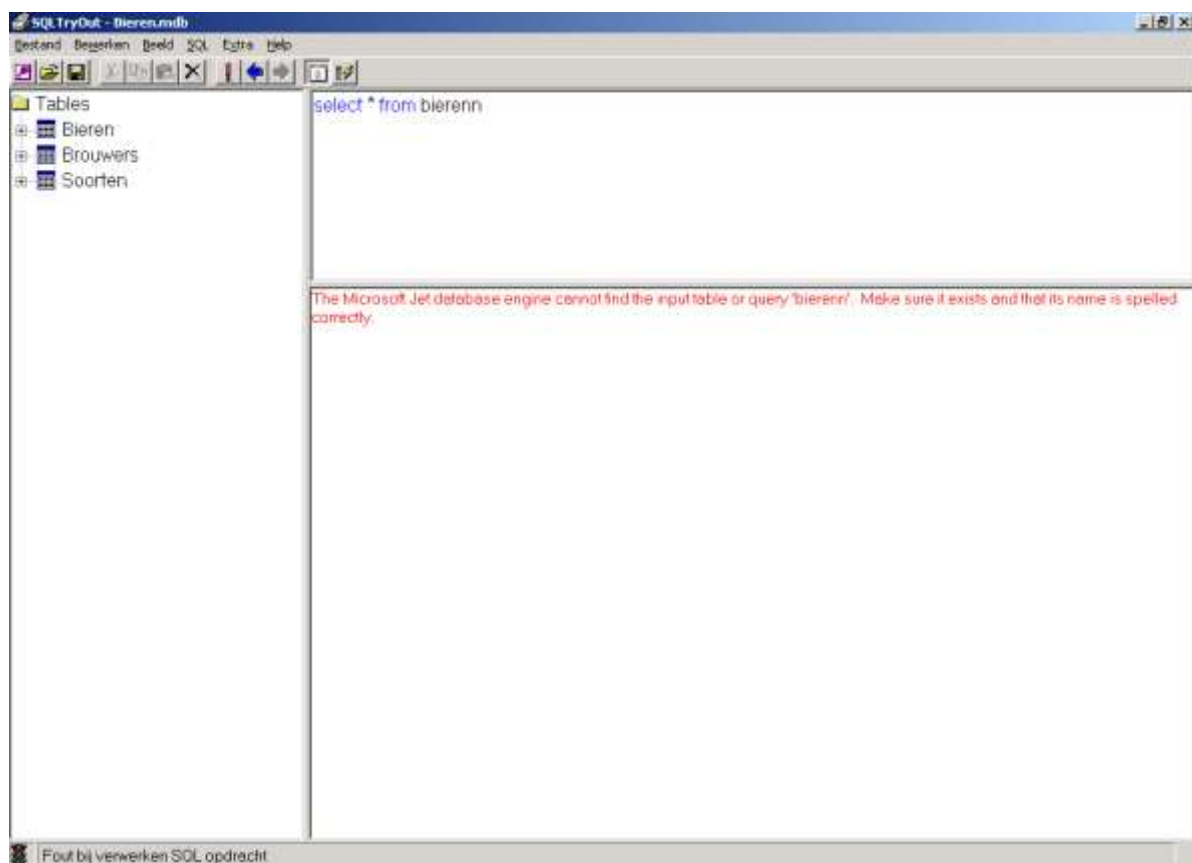
- Bieren
- Brouwers
- Soorten

select * from bieren

BierNr	Naam	BrouwerNr	SoortNr	Alcohol
4	A.C.O.	104	18	7
5	Aalbeeks St. Corneliusbie	113	18	7
7	Aardbeien witbier	56	53	2.5
8	Aarschots kruikenbier (=S	105	15	7.6
10	Abt Bjebier (Nen)	33	18	7
11	Adler	51	42	6.75
12	Aerts 1900	81	14	7
13	Affligem blond (Abdij)	100	33	7
14	Affligem christmas ale (Ab	100	36	9
15	Affligem dubbel (Abdij)	100	14	7
16	Affligem patersvat	100	33	7
17	Affligem tripel (Abdij)	100	59	8.5
18	Akita pilsener	68	42	5
20	Aldegonde brune	72	36	8.5
21	Aldegonde cuvee	72	15	7.5
22	Aldegonde speciale	72	36	8.5
23	Ale brau (=Vega pils)	58	42	5
24	Alexander Rodenbach	92	62	6

1,215 records

Als u een fout heeft gemaakt in de SQL-instructie, ziet u een foutboodschap staan in het resultaatvak. SQLTryOut toont een rood licht.



Links onderaan staat een verkeerslicht. Dit aanduidt of de uitvoering van laatste instructie gelukt of mislukt is.





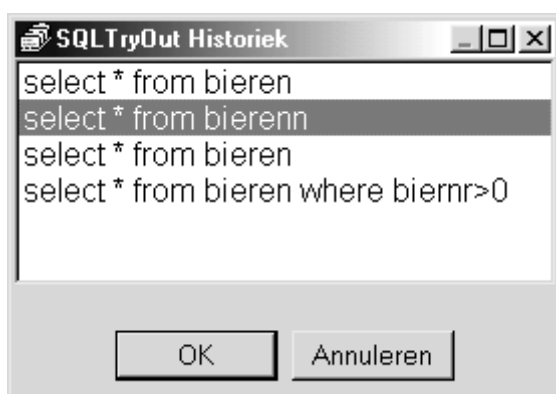
De laatste instructie is gelukt.





De laatste instructie bevat fouten en werd niet uitgevoerd.

Als u een wijziging aan de database aangebracht heeft, krijgt u een boodschap dat het gelukt is.

De instructies worden in een buffer geplaatst. U kunt met de toetsen Page Up en Page Down door de ingetypte instructies. U kunt ook gebruik maken van de knoppen  en  om te bladeren. Met de optie Historiek uit het menu SQL kunt u een overzicht opvragen van de SQL-instructies.



Met de opdracht Historiek leegmaken uit het menu SQL kunt u de historiek verwijderen.

Via SQL opdracht in tekstbestand opslaan in het menu Bestand of  kunt u de ingetypte instructie bewaren. Het bestand word bewaard in tekstformaat en kan met Notepad/Kladblok aangepast worden. De opgeslagen instructie kunt u terug openen met Tekstbestand met SQL bestand openen uit het menu Bestand of .

Het lettertype kunt u wijzigen met Letterttype uit het menu Extra.

- Als u het programma wilt afsluiten, kiest u Afsluiten uit het menu Bestand.

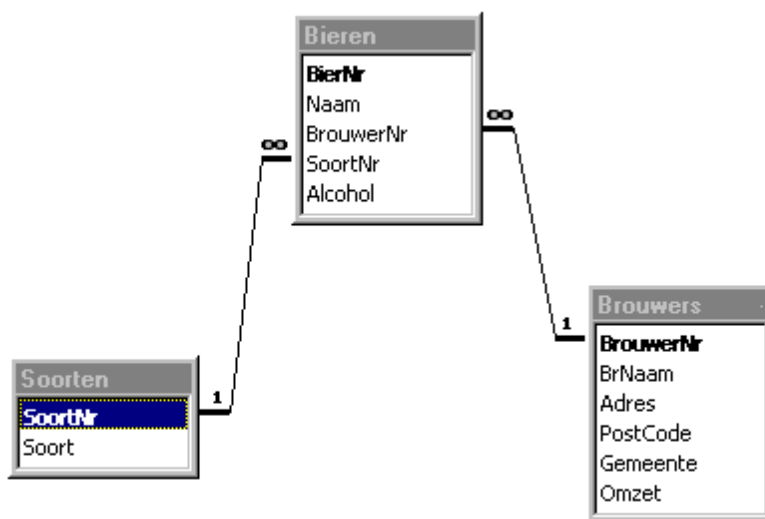
4 DE DATABASES BIEREN EN PLANTV

In deze cursus wordt er gebruik gemaakt van twee databases: bieren en plantv. Alle voorbeelden uit de cursus zijn gebaseerd op de database bieren. Voor de oefeningen maak je gebruik van de database plantv.

De database Bieren bestaat uit drie tabellen, nl. :

Bieren	Bevat een lijst van alle bieren, met de velden Biernr, Naam, BrouwerNr, Soortnr, Alcohol. Biernr is de Primary Key. Het veld BrouwerNr verwijst naar de brouwerij uit de tabel Brouwers. Het veld Soortnr verwijst naar de biersoort uit de tabel Soorten.
Brouwers	Bevat een lijst van alle brouwerijen, met de velden BrouwerNr, Brnaam, Adres, Postcode, Gemeente, Omzet. BrouwerNr is de Primary Key.
Soorten	Bevat een lijst van alle biersoorten, met de velden Soortnr, Soorten. Soortnr is de Primary Key.

Het schema van de database Bieren :

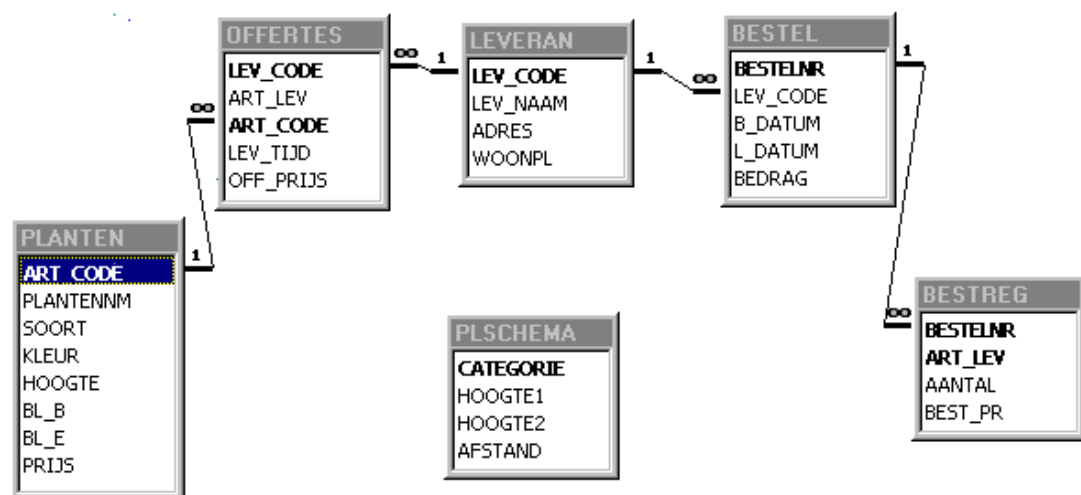


De database plantv bestaat uit 6 tabellen, nl:

Planten	Bevat de gegevens van de planten waarbij het veld art_code de primary key, plantennm bevat de naam van de plant, bl_b en bl_e bevat de maanden wanneer de bloeiperiode van de plant begint en eindigt.
Offertes	Bevat de catalogusgegevens van onze leveranciers.

	lev_code en art_code vormen samen de primary key. art_lev is de code die de leveranciers gegeven aan de plant.
Leveran	Bevat de gegevens van de leveranciers. Lev_code is de primary key.
Bestel	Bevat de bestellingen die wij geplaatst hebben bij de leveranciers. B_datum is de besteldatum en l_datum is de gewenste leveringsdatum.
Bestreg	Bevat de detaillijnen van de bestelling. De primary key wordt bepaald door de velden bestelnr en art_lev.
Plschema	Bevat de gegevens van de verschillende soorten categorieën van planten.

De prijzen in deze tabel zijn uitgedrukt in €. De hoogte is uitgedrukt in cm.



Deze database wordt gebruikt in de oefeningen.

5 GEGEVENS SELECTEREN

5.1 Syntax

Met deze instructie geeft u de RDBMS opdracht om gegevens uit de database in de vorm van een set records als resultaat te geven.

```
SELECT [predikaat] { * | tabel.* | [tabel.]veld1 [AS alias1]
[, [tabel.]veld2 [AS alias2] [, ...]] }
FROM tabelexpressie [, ...] [IN externedatabase]
[WHERE... ]
[GROUP BY... ]
[HAVING... ]
[ORDER BY... ]
```

De instructie SELECT bevat de volgende onderdelen :

Onderdeel	Beschrijving
predikaat	Een van de volgende predikaten: ALL, DISTINCT of TOP. Met een predikaat kunt u het aantal records in het resultaat beperken. De standaardinstelling is ALL als geen predikaat is opgegeven. TOP wordt niet door alle relationele databases ondersteund.
*	Bepaalt dat alle velden uit de opgegeven tabel of tabellen worden geselecteerd.
tabel	De naam van de tabel die de velden bevat waaruit records worden geselecteerd.
veld1, veld2	De namen van de velden waaruit gegevens worden opgehaald. Als u meer dan één veld opgeeft, worden deze opgehaald in de opgegeven volgorde.
alias1, alias2	De namen die u wilt gebruiken als kolomkoppen in plaats van de oorspronkelijke kolomnamen in tabel.
tabelexpressie	De naam van de tabel of de namen van de tabellen waaruit u gegevens wilt ophalen.
externedatabase	De naam van de database met de tabellen in tabelexpressie, als deze zich niet in de actieve database bevinden.

Voor het uitvoeren van deze bewerking doorzoekt de RDBMS de opgegeven tabel of tabellen, licht de opgegeven kolommen uit, selecteert de rijen die aan de voorwaarde(n) voldoen en sorteert of groepeert de resultaatrijen in de opgegeven volgorde.

Instructies met SELECT wijzigen de gegevens in de database niet.

5.2 Lijst opvragen

De minimale syntax voor een instructie SELECT luidt:

```
SELECT velden FROM tabel
```

```
SELECT naam, alcohol from bieren
```

geeft als resultaat een lijst met de naam en alcoholpercentage van alle bieren

```
SELECT brnaam from brouwers
```

geeft als resultaat een lijst met alle namen van de brouwerijen

```
SELECT gemeente from brouwers
```

geeft een lijst van alle woonplaatsen van de brouwerijen. Sommige gemeenten komen meerdere keren voor omdat er in deze gemeenten meerdere brouwerijen gevestigd zijn. Om elke gemeente maar één keer te krijgen, gebruiken we DISTINCT. DISTINCT zorgt ervoor dat elke rij van het resultaat uniek is. De SQL-instructie wordt dan :

```
SELECT DISTINCT gemeente FROM brouwers
```

Met een sterretje (*) kunt u alle velden in een tabel selecteren. In het volgende voorbeeld worden alle velden in de tabel bieren geselecteerd :

```
SELECT * from bieren
```

5.3 Selectie maken m.b.v. WHERE

De WHERE clause wordt gevolgd door een conditie. Een conditie is samengesteld uit :

```
attribuut operator constante | attribuut
```

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

Een constante kan een getal zijn of een karakterreeks. Een karakterreeks moet altijd beginnen en eindigen met een enkele quote (').

```
SELECT naam from bieren where alcohol < 5
```

geeft als resultaat een lijst met de naam van alle bieren met een alcoholpercentage lager dan 5%

```
SELECT brnaam from brouwers where gemeente = 'Brussel'
```

geeft als resultaat een lijst van alle brouwerijen gelegen in Brussel

```
select naam from bieren where naam like '%ale%'
```

geeft als resultaat een lijst van alle bieren waar het woord `ale` voorkomt in de naam. De asterisk duidt aan dat op die plaats een willekeurig aantal letters kunnen voorkomen. vb. als de naam moet beginnen met Wit dan luidt de selectie als volgt: `naam like 'wit%'`.

Aard van selectie	Patroon	Waarden die in het patroon passen	Waarden die niet voldoen aan de voorwaarden
Meerdere karakters	<code>a%a</code> <code>%ab%</code> <code>ab%</code>	<code>aa, aBa, aBBBa</code> <code>abc, AABB, Xab</code> <code>abcdefg, abc</code>	<code>aBC</code> <code>aZb, bac</code> <code>cab, aab</code>
Special teken	<code>a[@]a</code>	<code>a@a</code>	<code>aaa</code>
Eén karakter	<code>a_a</code>	<code>aaa, a3a, aBa</code>	<code>aBBBa</code>
Karakter moet voorkomen in de reeks	<code>[a-z]</code>	<code>f, p, j</code>	<code>2, &</code>
Karakter moet voorkomen in de benoemde lijst	<code>[agm]</code>	<code>a, g, m</code>	<code>b, c, n, 4</code>
Combinatie van hiervoor vermelde formaten	<code>a[b-m]_</code>	<code>Ab9, af0</code>	<code>aacfd, a90</code>

Opmerking

```
select naam from bieren where alcohol between 5 and 7
```

geeft een lijst van alle bieren met een alcoholpercentage dat gelegen is tussen 5% en 7%.

```
select naam from bieren where alcohol in (0, 5, 8)
```

geeft een lijst van alle bieren met een alcoholpercentage van 0%, 5% of 8%.

```
select brnaam from brouwers
where gemeente
in ('Leuven', 'Genk', 'Antwerpen', 'Dendermonde', 'Wevelgem')
```

geeft een lijst van alle brouwerijen gevestigd in de gemeenten Leuven, Hasselt, Genk, Antwerpen, Dendermonde en Wevelgem.

```
select naam from bieren where alcohol is Null
```

geeft een lijst van alle bieren waarvan de kolom alcohol niet ingevuld is. Om de kolommen te controleren die niet leeg zijn gebruikt u "is not null".

Je kunt voorwaarden combineren met elkaar met AND, OR en NOT. Bij AND moeten beide voorwaarden waar zijn omdat het resultaat waar zou zijn. Bij OR is het voldoende dat één van de voorwaarden waar is. Bij NOT gebeurt de negatie: waar wordt onwaar en omgekeerd.

Let op

alcohol is null	wordt alcohol is not null
alcohol between 5 and 7	wordt alcohol not between 5 and 7
naam like '%wit'	wordt naam not like '%wit%'
gemeente in ('genk','leuven')	wordt gemeente not in ('genk','leuven')

```
select * from bieren
where naam like '%wit%' and alcohol > 5
```

geeft een lijst van alle bieren met een naam die het woord wit bevat en alcoholpercentage hoger dan 5

```
select * from bieren
where naam not like '%wit%'
```

geeft een lijst van alle bieren met een naam die het woord wit niet bevat.

Let op

Let op het aantal bieren dat getoond wordt:

instructie1	select * from bieren where alcohol > 5
instructie2	select * from bieren where alcohol <= 5

De som van het aantal bieren dat getoond wordt door de instructie 1 en 2 is niet gelijk aan het totale aantal bieren. De rijen met een alcoholwaarde null worden genegeerd.

5.4 Een gesorteerde lijst opvragen m.b.v ORDER BY

```
select naam, alcohol from bieren order by naam asc
```

geeft een lijst van alle bieren (naam en alcoholpercentage) gesorteerd op alfabetisch op naam.

```
select naam, alcohol from bieren order by brouwnr desc,
naam asc
```

geeft een lijst van alle bieren (naam en alcoholpercentage) gesorteerd op brouwnr (dalend) en vervolgens alfabetisch op naam (stijgend).

```
select naam, alcohol from bieren order by 1 asc
```

geeft een lijst van alle bieren (naam en alcoholpercentage) gesorteerd op alfabetisch op naam. Maar hier wordt het kolomnummer gebruikt. De naam staat als eerste in de select, gevolgd door de kolom alcohol.

5.5 Oefeningen SELECT deel 1 (database plantv)

1. Geef een overzicht, met alle gegevens, van de leveranciers uit AALSMEER.
2. Geef een alfabetisch overzicht op plantennaam, met de artikelcode, plantennaam en prijs, van alle planten uit de tabel PLANTEN.
3. Welke planten beginnen in de maand maart te bloeien? Druk artikelcode, plantennaam, en begin bloeimaand af.
4. Maak een overzicht uit de tabel OFFERTES, gesorteerd op artikelcode en binnen artikelcode een sortering op artikelcode van de leverancier. Alleen de gegevens artikelcode, artikelcode van de leverancier en leverancierscode tonen.
5. Maak een gesorteerd overzicht van alle waterplanten. Sorteer op HOOGTE, grootste voorop.
6. Maak een lijst van de verschillende kleuren die bij de planten uit de tabel PLANTEN horen.
7. Maak een lijst van alle planten waarvan de kolom KLEUR niet ingevuld is.
8. Toon de verschillende soorten planten in de tabel PLANTEN.
9. Geef een overzicht van alle vaste planten welke geelkleurige bloemen geven. Van iedere plant toont u de volgende gegevens : artikelcode, plantennaam, hoogte en bl_b.
10. Geef een overzicht van alle planten met een prijs boven de 10 € die niet tot de soort bomen behoren.
11. Maak een lijst van alle planten die in juni beginnen te bloeien en gele bloemen geven en tevens van alle planten die in augustus voor het eerst bloeien en een rode bloemen hebben. Plaats alle beschikbare gegevens in het overzicht.
12. Welke planten met gemengde bloekleuren worden maximum 60 cm hoog? Druk artikelcode, plantennaam en hoogte af.
13. Geef een overzicht van alle leveranciers die niet in Hillegom wonen.
14. Van welke planten is zowel de kleur als de hoogte onbekend? Geef artikelcode, plantennaam, kleur en hoogte.
15. Welke planten bloeien in ieder geval in de periode augustus tot en met oktober? Geef artikelcode en plantennaam.
16. Welke planten bloeien in ieder geval in de maand september? Geef artikelcode en plantennaam.

17. Geef een overzicht van alle vaste planten met een prijs tussen 3 € en 5 €.
18. Geef een overzicht van alle planten die in maart, april, september of oktober beginnen te bloeien.
19. Bij welke planten komt het woord BOOM voor? Geef artikelcode en plantennaam.
20. Geef de artikelcode en plantennaam van alle planten die als derde letter een N hebben.
21. Welke 1- en 2-jarige planten staan er in de tabel PLANTEN? Probeer deze vraag met en zonder de operator LIKE op te lossen.
22. 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.
23. Bij welke planten komt het woord KRUID voor in hun plantennaam, maar behoren niet tot de soort KRUID? Geef artikelcode en plantennaam.
24. Geef de artikelcode en de plantennaam van alle planten die beginnen met de letter L en eindigen met de letter E.
25. Geef de artikelcode en plantennaam van alle planten die of met de letter A, of met de letter M beginnen en eindigen.
26. Welke planten hebben een plantennaam van precies 5 letters lang? Geef artikelcode en plantennaam.
27. Welke planten hebben een plantennaam van minimum 5 letters lang? Geef artikelcode en plantennaam.

5.6 Aggregate functions

```
select count(*) from bieren
```

geeft het totaal aantal bieren uit de tabel. In het resultaat zien we geen kolomnamen. Om toch een kolomnaam te geven aan het resultaat, gebruiken we de as om een alias toe te kennen zoals in het volgende voorbeeld :

```
select count(*) as aantal from bieren
```

```
select avg(alcohol) as gemidd, max(alcohol) as maxi,  
min(alcohol) as mini, sum(alcohol) as som from bieren
```

geeft het gemiddelde, maximum , minimum, totaal alcoholpercentage uit de tabel bieren

5.7 Berekeningen maken

```
select brnaam, omzet * 0.9118 as omzet_dollar, omzet * 116.6 as  
omzet_yen from brouwers
```

geeft een lijst van alle brouwerijen met hun omzet in dollar en yen. De mogelijke werkingen zijn optellen(+), aftrekken(-), vermenigvuldigen(*), delen(/) en machtsverheffing(^). Het gebruik van haakjes () in de bewerking is toegelaten.

5.8 Oefening SELECT deel 2

1. Hoeveel leveranciers telt ons tuincentrum?
2. Geef de gemiddelde prijs van alle waterplanten.
3. Wat is de maximale hoogte van de bomen in de tabel PLANTEN?
4. Geef de gemiddelde, de laagste en de hoogste offerteprijs van leverancier 013.
5. Wat is de laagste offerteprijs voor artikelcode 123?
6. Geef bestelnummer, artikelcode van de leverancier en het totale bestelbedrag per bestelrij uit de tabel BESTREG.
7. Maak een overzicht van de heesters uit de tabel PLANTEN, waarbij de prijzen met 5% zijn verhoogd.
8. Wat is het totale bedrag aan bestellingen bij leverancier 004?
9. Hoeveel stuks van de leveranciers-artikelcode B111 zijn besteld?
10. Wat is het totale bestelbedrag (exclusief korting) voor de leveranciers-artikelcode B331?

5.9 Gegevens groeperen

```
select brouwnr, avg(alcohol) as gemidd
from bieren
group by brouwnr
```

berekent het gemiddelde alcoholpercentage per brouwnr

In de lijst van de te tonen kolommen mogen enkel bewerkingen met een aggregate functie staan en kolommen die vermeld staan na de group by.

```
select art_code, art_lev, avg(off_prijs)
from offertes group by art_code
```

is niet toegelaten omdat bij art_lev geen aggregaat functie gebruikt wordt of omdat art_lev niet na de group by staat

```
select left(naam,1), avg(alcohol)
from bieren
group by left(naam,1)
```

toont ons de gemiddelde alcoholpercentage gegroepeerd volgens de eerste letter van de biernaam. Als je wilt groeperen op de beginletter, dan moet u de bewerking in het group by clause herhalen.

```
select brouwnr, min(alcohol) as mini  
from bieren  
group by brouwnr  
having min(alcohol)<5
```

bepaalt het minimum alcoholpercentage per brouwnr, de lijst toont enkel de brouwnr's en percentages die kleiner zijn dan 5%. U gebruikt "having" indien de selectie gebaseerd is op het resultaat van een bewerking met een aggregaat functie. In alle andere gevallen gebruikt u "where".

```
select brouwnr, avg(alcohol) as mini  
from bieren  
group by brouwnr  
having count(*)>10
```

toont het gemiddelde alcoholpercentage per brouwnr voor alle brouwers die minimum 10 bieren produceren.

Let op

```
select brouwnr, avg(alcohol) as mini  
from bieren  
group by 1
```

is FOUT. Je mag niet verwijzen met een nummer zoals bij ORDER BY.

5.10 Oefeningen SELECT deel 3

1. Tel het aantal planten per plantensoort uit de tabel PLANTEN.
2. Tel het aantal bestelregels per bestelling.
3. Wat is de gemiddelde prijs per plantensoort.
4. Hoeveel planten zijn er per plantensoort-kleurgroep in de tabel PLANTEN?
5. Vervaardig een overzicht waaruit visueel kan worden afgeleid welke kleur van de vaste planten de hoogste gemiddelde prijs heeft.
6. Laat per leverancierscode het aantal artikelen zien dat de leverancier aanbiedt onder voorwaarde dat de levertijd van het artikel minder dan 18 dagen bedraagt.
7. Wat is de gemiddelde prijs per plantensoort, exclusief de geelbloemige planten?
8. Maak een overzicht met de laagste en de hoogste offerteprijs per artikelcode.
9. Wat is de gemiddelde prijs per plantensoort, voor soorten met minstens 10 exemplaren in de tabel PLANTEN.
10. Hebben de planten met korte levertijden in het algemeen een hogere gemiddelde offerteprijs?

11. Maak een overzicht met de laagste en de hoogste bestelprijs per artikelcode van de leverancier.
12. Geef een overzicht van het aantal beschikbare planten per bloeibeginmaand/hoogte/kleur groep.
13. Wat is de laagste prijs per plantensoort van de planten die in ieder geval bloeien in de periode mei t/m juni?
14. Tel het aantal planten per aantal maanden dat de planten bloeien.
 Bijvoorbeeld

Duurtijd	Aantal
0	29
1	50
2	24
3	13
...	
15. De functie ROUND(getal, 0) rondt het getal af naar een geheel getal. Tel het aantal planten per prijs (na afronding).

5.11 Gegevens selecteren uit meerdere tabellen tegelijkertijd

Er zijn op dit ogenblik twee gangbare methodes. De eerste is de clause INNER JOIN, die wordt niet erkend door alle soorten relationele databases.

Met deze bewerking kunt u records uit twee tabellen combineren wanneer een gemeenschappelijk veld overeenkomstige waarden bevat.

```
FROM tabel1 INNER JOIN tabel2
ON tabel1.veld1 operator tabel2.veld2
```

De instructie INNER JOIN bevat de volgende onderdelen:

Onderdeel	Beschrijving
tabel1, tabel2	De namen van de tabellen waaruit records worden gecombineerd.
veld1, veld2	De namen van de velden die worden gekoppeld. Als het geen numerieke velden zijn, moet het gegevenstype van de velden gelijk zijn en moeten ze dezelfde soort gegevens bevatten. Ze hoeven echter niet dezelfde naam te hebben. Indien dit wel zo moet u wel een onderscheid maken door de tabelnamen te gebruiken op de volgende wijze : tabel.veld
operator	Een willekeurige relationele vergelijkingsoperator: "=", "<", ">", "<=", ">=" of "<>".

U kunt een bewerking INNER JOIN toepassen in elke component FROM. Dit is het meest gebruikte type koppeling. Met INNER JOIN worden records uit twee tabellen gecombineerd op basis van gelijke waarden in een gemeenschappelijk veld.

U kunt INNER JOIN gebruiken met de tabellen Brouwers en Bieren om alle van alle bieren hun brouwerij te selecteren. Wilt u daarentegen alle brouwerijen (met inbegrip van brouwerijen zonder bieren), dan kunt u een bewerking LEFT JOIN of RIGHT JOIN gebruiken om een outer join (koppeling aan de buitenkant) te maken.

In het volgende voorbeeld worden de tabellen bieren en brouwers gekoppeld via het veld brouwnr:

```
SELECT Naam, Brnaam
FROM Bieren INNER JOIN Brouwers
ON Bieren.Brouwnr = Brouwers.Brouwnr
```

Hoewel Brouwnr in dit voorbeeld het gekoppelde veld is, wordt het niet opgenomen in de query-uitvoer omdat het niet is opgenomen in de instructie SELECT. Als u het gekoppelde veld wel wilt opnemen, geeft u de veldnaam op in de instructie SELECT (in dit geval: Bieren.Brouwnr).

Met de volgende syntax kunt u ook verschillende componenten ON koppelen in een instructie JOIN:

```
SELECT velden
FROM tabel1 INNER JOIN tabel2
ON tabel1.veld1 operator tabel2.veld1 AND
ON tabel1.veld2 operator tabel2.veld2) OR
ON tabel1.veld3 operator tabel2.veld3]
```

Deze syntax heeft u nodig wanneer u twee tabellen wilt koppelen met een samengestelde sleutel (dit is een sleutel gebaseerd op de meerdere velden).

Tevens is het mogelijk met de volgende syntax meerdere tabellen met JOIN te nesten:

```
SELECT velden
FROM tabel1 INNER JOIN
(tabel2 INNER JOIN [( ]tabel3
[INNER JOIN [( ]tabelx [INNER JOIN ...)]
ON tabel3.veld3 operator tabelx.veldx])
ON tabel2.veld2 operator tabel3.veld3)
ON tabel1.veld1 operator tabel2.veld2
```

```
SELECT Bieren.Naam, Brouwers.BrNaam, Soorten.Soort
FROM Soorten
INNER JOIN (Brouwers INNER JOIN Bieren
ON Brouwers.BrouwerNr = Bieren.BrouwerNr) ON
Soorten.SoortNr = Bieren.SoortNr
```

Geeft een lijst van alle bieren met de velden naam, brouwerij en soort uit de tabellen bieren, brouwers en soorten.

De voorgaande methode wordt niet door alle relationele databases ondersteund. De tweede methode gaat als volgt :

```
SELECT velden
FROM tabel1, tabel2
WHERE tabel1.veld1=tabel2.veld2
```

Onderdeel	Beschrijving
tabel1, tabel2	De namen van de tabellen waaruit records worden gecombineerd.
veld1, veld2	De namen van de velden die worden gekoppeld. Als het geen numerieke velden zijn, moet het gegevenstype van de velden gelijk zijn en moeten ze dezelfde soort gegevens bevatten. Ze hoeven echter niet dezelfde naam te hebben. Indien dit wel zo moet u wel een onderscheid maken door de tabelnamen te gebruiken op de volgende wijze : tabel.veld

In het volgende voorbeeld worden de tabellen bieren en brouwers gekoppeld via het veld brouwnr:

```
SELECT Naam, Brnaam
FROM Bieren, Brouwers
WHERE Bieren.Brouwnr = Brouwers.Brouwnr;
```

Algemene syntax voor meer dan twee tabellen wordt :

```
SELECT velden
FROM tabel1, tabel2, ...tabelN
WHERE tabel1.veld1 = tabel2.veld2 and
      tabel2.veld2 = ... and
      tabel...veld... = tabeln.veldn
```

```
SELECT Naam, Brnaam, Soort
FROM Bieren, Brouwers, Soorten
WHERE bieren.brouwnr=brouwers.brouwnr and
      bieren.soortnr=soorten.soortnr
```

Geeft een lijst van alle bieren met de velden naam, brouwerij en soort uit de tabellen bieren, brouwers en soorten.

5.12 Outer join

Met deze bewerking combineert u records van de brontabel in een component FROM.

```
FROM tabel1 [ LEFT | RIGHT ] JOIN tabel2
ON tabel1.veld1 vergopr tabel2.veld2
```

De bewerkingen LEFT JOIN en RIGHT JOIN bevatten de volgende onderdelen:

Onderdeel	Beschrijving
tabel1, tabel2	De namen van de tabellen waaruit records worden gecombineerd.
veld1, veld2	De namen van de velden die worden gekoppeld. De velden moeten hetzelfde gegevenstype hebben en dezelfde soort gegevens bevatten. Ze hoeven echter niet dezelfde naam te hebben.
operator	Een willekeurige relationele vergelijkingsoperator: "=", "<", ">", "<=", ">=" of "<>".

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

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

U kunt bijvoorbeeld in de tabellen Afdelingen (linkertabel) en Werknemers (rechtertabel) met LEFT JOIN alle afdelingen selecteren, inclusief afdelingen zonder personeelsleden. Als u alle personeelsleden wilt selecteren, dus met inbegrip van personeelsleden die niet werkzaam zijn op een bepaalde afdeling, gebruikt u RIGHT JOIN.

In het volgende voorbeeld worden de tabellen Soorten en Bieren gekoppeld op het veld Soortnr. De select-instructie genereert een lijst van alle Soorten, inclusief biersoorten waarvan er geen bieren aanwezig zijn in de tabel Bieren :

```
SELECT Naam, Soort
FROM Soorten LEFT JOIN Bieren
ON Soorten.Soortnr = Bieren.Soortnr
```

In het volgende voorbeeld worden de tabellen Brouwers en Bieren gekoppeld op het veld Brouwnr. De select-instructie genereert een lijst van alle Brouwers, inclusief de brouwerijen waarvan er geen bieren aanwezig zijn in de tabel Bieren :

```
SELECT BrNaam, Naam
FROM Bieren RIGHT JOIN Brouwers
ON Brouwers.Brouwnr = Bieren.Brouwnr
```

Opmerkingen

- Als u een query wilt maken waarin alleen records met identieke gegevens in de gekoppelde velden worden opgenomen, gebruikt u een bewerking INNER JOIN.
- LEFT JOIN of RIGHT JOIN kan worden genest binnen INNER JOIN, maar INNER JOIN kan niet worden genest binnen LEFT JOIN of RIGHT JOIN.

- U kunt meerdere componenten ON koppelen.

5.13 Self Join

Een self join is een inner join of een outer join waarbij je twee keer dezelfde tabel gebruikt. Om dit goed te laten verlopen moet je de tabellen een aliasnaam geven. Als je de tabellen een aliasnaam geeft, moet je bovendien in de select, on, group by en where clauses verwijzen naar de tabel met de aliasnaam ipv van de tabelnaam.

```
select b1.brnaam, b2.brnaam
from brouwers as b1 inner join brouwers as b2
on b1.gemeente=b2.gemeente and b1.brouwernr<b2.brouwernr
```

geeft een lijst van brouwers die in dezelfde gemeente wonen. Aan de tabel brouwers wordt telkens een aliasnaam gegeven. Als je een kolomnaam gebruikt, ben je verplicht om de aliasnaam te gebruiken.

5.14 Select ... into ...

Hiermee maakt u een nieuwe tabel op basis van de geselecteerde records.

```
SELECT veld1[, veld2[, ...]] INTO nieuwetabel [IN externedatabase]
FROM bron
```

De instructie SELECT...INTO bevat de volgende onderdelen:

Onderdeel	Beschrijving
veld1, veld2	De namen van de velden die naar de nieuwe tabel moeten worden gekopieerd.
nieuwetabel	De naam van de tabel die moet worden gemaakt. De naam moet voldoen aan de standaardconventies voor naamgeving. Als nieuwetabel gelijk is aan de naam van een bestaande tabel, treedt een ondervangbare fout op.
externedatabase	Het pad naar een externe database. Zie de component IN voor een beschrijving van het pad.
bron	De naam van een bestaande tabel waaruit records worden geselecteerd. Dit kan één tabelnaam of een samengestelde tabelnaam of query zijn.

Met een SELECT ... INTO kunt u records archiveren, reservekopieën maken van uw tabellen en kopieën maken die u naar een andere database kunt exporteren of als basis kunt gebruiken voor rapporten met gegevens over een bepaald biersoort.

```
select naam, brnaam
into alcoholarm
from bieren, brouwers, soorten
```

```
where bieren.brouwernr=brouwers.brouwernr and  
      bieren.soortnr=soorten.soortnr and  
      soorten.soort='Alcoholarm'
```

Opmerkingen

- U kunt een primaire sleutel definiëren voor de nieuwe tabel. Als u de tabel maakt, krijgen de velden in de nieuwe tabel hetzelfde gegevenstype en dezelfde veldlengte als de velden in de onderliggende tabellen in de query, maar worden er verder geen veld- en tabeleigenschappen overgenomen.
- Als u gegevens wilt toevoegen aan een tabel, kunt u de instructie INSERT INTO gebruiken.

5.15 UNION

Hiermee maakt u een samenvoegquery, waarmee de resultaten van twee of meer onafhankelijke query's of tabellen worden gecombineerd.

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

De UNION-bewerking bevat de volgende onderdelen:

Onderdeel	Beschrijving
query1-n	Een SELECT-instructie, de naam van een opgeslagen query of de naam van een opgeslagen tabel die vooraf wordt gegaan door het sleutelwoord TABLE.

Nadere informatie

In een enkele UNION-bewerking kunt u in elke willekeurige combinatie de resultaten samenvoegen van twee of meer query's, tabellen of SELECT-instructies.

```
select * from bieren where soortnr = 1  
union  
select * from bieren where soortnr = 5
```

selecteer alle bieren met soortnr 1 of 5

```
select * from bieren  
union  
select * from bieren
```

toont alle gegevens van de tabel bieren slechts éénmaal. Standaard worden geen dubbele records als resultaat gegeven als u een UNION-bewerking gebruikt. Met UNION ALL wordt wel alle records getoond, ook de dubbele. De query wordt dan ook sneller uitgevoerd.

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.

Aliassen kunt u alleen gebruiken in de eerste SELECT-component, omdat ze worden genegeerd in alle overige. In de ORDER BY-component moet u naar velden verwijzen met dezelfde namen die worden gebruikt in de eerste SELECT-component.

Opmerkingen

- U kunt een GROUP BY- en/of HAVING-component opnemen in elk query-argument om de als resultaat gegeven gegevens te groeperen.
- Aan het eind van het laatste query-argument kunt u een ORDER BY-component opnemen om de gegevens in het resultaat in een bepaalde volgorde weer te geven.

5.16 Oefeningen SELECT deel 4

1. Maak een overzicht van de bestellingen met het bestelnummer, de naam van de leverancier, de leverdatum en het bestelde bedrag.
2. Welke planten hebben een levertijd van maximum 10 dagen ? Plaats in het overzicht de gegevens : artikelcode, leveranciers-artikelcode, plantennaam en levertijd.
3. Geef een overzicht waaruit blijkt bij welke leverancier(s) de planten besteld kunnen worden. In het overzicht moeten de volgende gegevens staan : artikelcode, plantennaam, leverancierscode, naam van de leverancier, levertijd en offerteprijs. Sorteer het overzicht op plantennaam.
4. Maak een overzicht van die planten die een prijs hebben welke minstens 50% boven de offerteprijs ligt. Geef artikelcode, plantennaam, prijs van de plant, leverancierscode en offerteprijs.
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.
6. In welke plaatsen kan het tuincentrum vaste planten bestellen?
7. Maak een overzicht van alle rode planten geleverd door leveranciers die niet in Aalsmeer wonen. Vermeld de gegevens : artikelcode, plantennaam, soort, naam van de leverancier, woonplaats. Sorteer op soort en plantennaam.
8. Bepaal voor iedere artikelcode de laagste offerteprijs. Vermeld in het overzicht de volgende gegevens : artikelcode, plantennaam en de bijbehorende offerteprijs. Sorteer het overzicht op plantennaam.
9. Zoek de bestellingen welke een besteldatum hebben die gelijk is aan de leverdatum van één of meer andere bestellingen.

- Geef bestelnummer 1^e bestelling, besteldatum 1^e bestelling, bestelnummer 2^e bestelling, leveringsdatum 2^e bestelling.
10. Welke artikelcodes van het tuincentrum en artikelcodes van een leveranciers zijn hetzelfde ongeacht de plant?
Geef leverancierscode 1^e artikel, artikelcode, leverancierscode 2^e artikel, artikelcode van de leverancier.
 11. Op een gegeven moment (1 april 1999) moet een overzicht worden vervaardigd, 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 bestelnummer.
Geef bestelnummer, leveringsdatum, bedrag, bericht 'TE LAAT' of '-----'.
(In SQLtryout gebruik je #1999-4-1# als getalnotatie)
 12. 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 : artikelcode, plantennaam, artikelcode van de leverancier, alsmede een aanduiding 'AALSMEER' of 'BUITEN AALSMEER'.

5.17 Informatie schrijven naar en lezen uit een externe database

Om een andere database aan te spreken gebruiken we de `IN` clausule.

In het programma hebben we de database bieren geopend. Maar we willen toch alle planten lezen uit de databank plantv.

```
select * from planten in 'c:\databank\plantv.mdb'
```

Een nieuwe tabel maken vanuit de databank bieren in de plantv

```
select * into test in 'c:\databank\plantv.mdb' from bieren
```

5.18 Subqueries

Bij een subquery wordt het resultaat van een select-instructie gebruikt in een andere sql-instructie.

```
select naam from bieren where alcohol = (select max(alcohol) from bieren)
```

geeft een lijst van alle bieren met het hoogste alcoholpercentage. In dit voorbeeld mag de subquery slechts een waarde als resultaat geven. Indien u de operatoren `=`, `>`, `<`, `>=`, `<=` gebruikt mag de subquery slechts **één** waarde als resultaat geven. Alleen de operator `in` kan met een set van waarden werken zoals in het volgende voorbeeld getoond wordt.

```
select naam from bieren where brouwnr in (select brouwnr from brouwers where gemeente like 'brussel')
```

geeft een lijst van alle bieren die in Brussel gebrouwen zijn.

In deze voorbeelden worden de subquery gebruikt in de where-clausule van de select-instructie. U kunt ook subqueries combineren met de INSERT en UPDATE instructies.

De subqueries kunt u ook gebruiken in de from clausule van de select. Eerst wordt dan de subquery uitgevoerd en dan pas de andere select-instructie.

```
select soortnr
from (select distinct soortnr, brouwnr from bieren)

group by soortnr having count(*) = 1
```

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

```
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 gebruikt wordt als kolomnaam in de buitenste query. Om het resultaat van de subquery te kunnen joinen met een andere tabel moet je aan aliasnaam geven aan de subquery.

5.19 Gecorreleerde subqueries

Een gecorreleerde subquery is een subquery waarin een kolom gebruikt wordt die tot een tabel behoort die in een ander queryblok gespecificeerd is.

De opdracht “Maak een lijst van alle bieren die een lager alcoholpercentage dan die van zijn eigen soort” luidt als volgt :

```
Select b1.* from bieren b1 where b1.alcohol < (select
avg(b2.alcohol) from bieren b2 where b2.soortnr=b1.soortnr)
```

Hier wordt wel tweemaal gebruik gemaakt van de tabel Bieren. Daarom wordt er aan de tabel een alias toegekend, om aan te duiden tot welke tabel de velden horen gebruiken we hier de aliasnaam i.p.v. de naam Bieren.

5.20 Oefeningen SELECT deel 5

1. Welke planten zijn hoger dan de gemiddelde hoogte van alle planten tezamen? Druk alle gegevens af.
2. Welke planten zijn duurder dan de gemiddelde prijs van de bomen? Druk alle gegevens ervan af.
3. Maak een overzicht van de leveranciers (alle gegevens) waar nog bestellingen uitstaan met een leverdatum die vóór 1 april 1999 ligt. (In SQLtryout gebruik je #1999-4-1# als getalnotatie)

4. Welke rijen hebben, de laagste offerteprijs van alle offertes in de tabel OFFERTES? Geef alle gegevens.
5. Welke planten zijn lager dan de laagste vaste plant? (alle gegevens geven)
Planten waar de hoogte = 0 worden niet meegerekend.
6. Welke planten zijn hoger dan de gemiddelde hoogte van vaste planten en tevens goedkoper dan de gemiddelde prijs van vaste planten? Geef alle gegevens.
7. Welke planten hebben een prijs die tussen de laagste en hoogste prijs van de klimplanten ligt? Geef alle gegevens.
8. Welke bestellingen hebben meer dan 5% korting gekregen. Betrek de tabel BESTREG bij de oplossing.
Geef alle gegevens.
9. Maak een overzicht van alle artikelcodes die een lagere offerteprijs hebben dan de gemiddelde offerteprijs voor de betreffende artikelcode. Plaats de gegevens artikelcode, leveranciersnaam en offerteprijs op het overzicht, en sorteer op artikelcode.
10. 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.

6 GEGEVENS UIT DE DATABASE AANPASSEN

6.1 INSERT

Met deze instructie kunt u een of meer records toevoegen aan een tabel.

Meerdere records:

```
INSERT INTO doel [IN externedatabase] [(veld1[, veld2[, ...]])]  
SELECT [bron.]veld1[, veld2[, ...]]  
FROM tabelexpressie
```

Eén record:

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

De instructie INSERT INTO bevat de volgende onderdelen:

Onderdeel	Beschrijving
doel	De naam van de tabel of query waaraan records worden toegevoegd.
externedatabase	Het pad naar een externe database. Zie de component IN voor een beschrijving van het pad.
bron	De naam van de tabel of query waaruit de records worden gekopieerd.
veld1, veld2	Bij het argument doel de namen van de velden waaraan gegevens worden toegevoegd en bij het argument bron de namen van de velden waaruit de gegevens afkomstig zijn.
tabelexpressie	De naam van de tabel of de namen van de tabellen waaruit de toegevoegde records afkomstig zijn. Dit argument kan één tabelnaam zijn of een samenstelling die het resultaat is van een bewerking INNER JOIN, LEFT JOIN of RIGHT JOIN, of een opgeslagen query.
waarde1, waarde2	De waarden die in bepaalde velden van de nieuwe record worden ingevoegd. Elke waarde wordt ingevoegd in het veld dat overeenkomt met de positie van de waarde in de lijst: waarde1 wordt ingevoegd in veld1 van de nieuwe record, waarde2 in veld2, enzovoort. U moet elke waarde tussen aanhalingstekens (' ') plaatsen en de afzonderlijke waarden scheiden met komma's.

U kunt met de instructie INSERT INTO één record toevoegen aan een tabel met de eerder beschreven syntax voor het toevoegen van één record. In dit geval geeft de

code de naam en de waarde voor elk veld in de record op. U moet elk veld in de record opgeven waaraan een waarde moet worden toegewezen en tevens de waarde zelf. Als u niet elk veld opgeeft, wordt in de ontbrekende kolommen de standaardwaarde of Null ingevoegd. Records worden aan het einde van de tabel toegevoegd.

U kunt met INSERT INTO ook een set records uit een andere tabel of view toevoegen door de component SELECT ... FROM te gebruiken met de hierboven beschreven syntax voor meerdere records. In dat geval geeft de component SELECT aan welke velden worden toegevoegd aan de tabel die wordt aangegeven door doel.

De bron- of doeltabel kan een tabel of een view zijn. Als een view is opgegeven, worden de records door de RDBMS toegevoegd aan alle tabellen die zijn opgegeven in de query.

INSERT INTO is optioneel, maar als de instructie wordt gebruikt, moet deze voorafgaan aan de instructie SELECT.

Als de doeltabel een primaire sleutel bevat, moet u ervoor zorgen unieke, niet-Null-waarden toe te voegen aan het primaire sleutelveld of de primaire sleutelvelden. Als u dat niet doet, zal de RDBMS geen records toevoegen.

Met de component IN kunt u records toevoegen aan een tabel in een andere database.

Een INSERT INTO ... SELECT kopieert records uit een of meer tabellen naar een andere tabel. De tabellen met de oorspronkelijke records worden door de SQL-instructie niet beïnvloed.

In plaats van bestaande records uit een andere tabel toe te voegen, kunt u ook met de component VALUES een waarde opgeven voor elk veld in één nieuwe record. Als u de veldenlijst achterwege laat, moet de component VALUES een waarde voor elk veld in de tabel bevatten, want anders zal de bewerking INSERT niet slagen. Gebruik voor elke record die u wilt toevoegen, een afzonderlijke instructie INSERT INTO met een component VALUES.

Indien u records toevoegt aan de tabel wordt er steeds gecontroleerd of de referentiële integriteit niet aangetast wordt. U kan geen records toevoegen aan de tabel Bieren met een soortnr of brouwnr dat niet voorkomt in de respectievelijke tabellen.

```
INSERT INTO Soorten (Soortnr, Soort)
VALUES (30, 'Extra donker')
```

voegt een nieuwe soort toe aan de tabel Soorten

```
INSERT INTO Brouwers
VALUES (99, 'Brouwerij Vaattappers', 'Interleuvenlaan 2',
3000, 'Heverlee', 1000)
```

voegt een nieuwe brouwerij toe aan de tabel brouwers. U hoeft de veldnamen niet te vermelden indien u de waarden (values) opgeeft in de volgorde zoals de velden gedefinieerd zijn.

6.2 Oefening INSERT

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

Lev_code	045
Lev_naam	GROEN BV.
Adres	ONDER DE LINDE 234
WOONPL	AALSMEER

Voeg deze nieuwe leverancier toe aan de tabel LEVERAN.

2. Er is behoefte aan een tabel (naam tabel = AANBIED) waaruit snel de goedkoopste leveranciers van de planten uit de tabel PLANTEN kunnen worden opgezocht. De tabel heeft de volgende kolommen : ART_CODE, PLANTENNM, LEV_CODE en OFF_PRIJS. Vul deze tabel vanuit de PLANTEN en OFFERTES tabellen.
3. Op 23 april 1999 is een nieuwe bestelling geplaatst bij leverancier 013. De volgende rijen worden in de tabel BESTREG opgenomen :

BESTELNR	ART_LEV	AANTAL	PRIJS
0205	C051	10	8.15
0205	B101	200	0.40
0205	B111	25	2.30
0205	G001	50	1.30

Voer deze toevoeging uit.

De leverancier geeft 8% korting op het bruto bestelbedrag. Maak een toevoeging aan de BESTEL-tabel met als leverdatum één week na de bestelling.

6.3 UPDATE

Met deze instructie wijzigt u de waarden in velden in een opgegeven tabel op basis van opgegeven criteria.

```
UPDATE tabelexpressie
SET nieuwewaarde
WHERE criteria;
```

De instructie UPDATE bevat de volgende onderdelen:

Onderdeel	Beschrijving
Tabelexpressie	De naam van de tabellen (met inner join) met de gegevens die worden gewijzigd en uit geselecteerd worden.
Nieuwewaarde	Een expressie die bepaalt welke waarde wordt ingevoegd in een bepaald veld in de bijgewerkte records.
Criteria	Een expressie die bepaalt welke records worden bijgewerkt. Alleen records die voldoen aan de expressie, worden bijgewerkt.

UPDATE is vooral bruikbaar als u een groot aantal records wilt wijzigen of als de te wijzigen records zich in meerdere tabellen bevinden.

U kunt diverse velden in één keer wijzigen. In het volgende voorbeeld worden het adres, postcode en gemeente van de brouwerij met het nummer 99 gewijzigd :

```
UPDATE Brouwers
SET Adres='Keizerslaan 111', Postcode=1000,
Gemeente='Brussel'
WHERE Brouwernr = 99
```

Het alcoholpercentage wordt verhoogd met 0.5 voor bieren die gebrouwen worden in een brouwerij met een omzet > 1000000.

Access :

```
UPDATE Bieren INNER JOIN Brouwers
ON Bieren.BrouwerNr = Brouwers.BrouwerNr
SET alcohol=alcohol+0.5
WHERE omzet>1000000
```

Standaard SQL :

```
UPDATE Bieren
SET alcohol=alcohol+0.5
FROM Bieren INNER JOIN Brouwers
ON Bieren.BrouwerNr = Brouwers.BrouwerNr
WHERE omzet>1000000
```

Ons programma volgt de schrijfwijze van access.

Je kan deze update instructie met een subquery oplossen:

```
UPDATE Bieren
SET alcohol=alcohol+0.5
WHERE BrouwerNr in
(SELECT BrouwerNr FROM Brouwers WHERE omzet>1000000)
```

Deze laatste werkwijze is de veiligste en werkt op alle database server correct.

Belangrijk

UPDATE genereert geen resultaatset. Als u records hebt bijgewerkt met de UPDATE-instructie, kunt u de bewerking niet meer ongedaan maken. Als u van tevoren wilt weten welke records zullen worden gewijzigd, kunt u eerst de resultaten van een SELECT-instructie met dezelfde criteria onderzoeken en vervolgens de UPDATE-instructie uitvoeren.

6.4 Oefeningen UPDATE

1. Wijzig in de tabel BESTEL de leverdatum in 5 april 1999 voor het bestelnummer 0191.
2. Verhoog alle offerteprijzen van de bolgewassen in de tabel OFFERTES met 10%.

6.5 DELETE

Met deze instructie verwijdert u de records uit een of meer tabellen die zijn opgegeven in de component FROM en die voldoen aan de component WHERE.

```
DELETE [tabel]
FROM tabelexpressie
WHERE criteria
```

De instructie DELETE bevat de volgende onderdelen:

Onderdeel	Beschrijving
tabel	De optionele naam van de tabel waaruit records worden verwijderd.
tabel	De naam van de tabel waaruit records worden verwijderd.
criteria	Een expressie die aangeeft welke records worden verwijderd.

DELETE is met name bruikbaar als u een groot aantal records wilt verwijderen.

Gebruik de instructie DROP als u een volledige tabel wilt verwijderen uit de database. De structuur gaat echter verloren als u de tabel verwijdert. Als u daarentegen DELETE gebruikt, worden alleen de gegevens verwijderd. De tabelstructuur en alle tabeleigenschappen, zoals veldkenmerken en indexen, blijven bestaan.

U kunt met DELETE records verwijderen uit tabellen die een één-op-veel-relatie hebben met andere tabellen. Ook hier wordt de referentiële integriteit bewaakt. U kunt geen records verwijderen uit de tabel Brouwers als er nog records voorkomen in de tabel Bieren met dat Brouwernr.

Met een DELETE worden volledige records verwijderd en niet alleen de gegevens in bepaalde velden. Als u waarden in een bepaald veld wilt verwijderen, kunt u een UPDATE instructie geven waarmee de waarden worden gewijzigd in Null.

```
delete from bieren where biernr = 52
```

verwijdert het record met het nummer 52 uit de tabel Bieren

```
DELETE brouwers.*  
FROM brouwers  
WHERE brouwnr IN  
(SELECT brouwnr from BIEREN WHERE alcohol > 20)
```

verwijdert alle brouwerijen die een bier produceren met een alcoholpercentage die hoger is dan 20%. Mbv de subquery

De volgende instructies werken niet in SQLTryout en Access maar wel in SQL Server, Oracle en andere professionele databases.

```
DELETE brouwers.* FROM brouwers INNER JOIN bieren ON  
brouwers.brouwnr=bieren.brouwnr  
WHERE alcohol > 20
```

verwijdert alle brouwerijen die een bier produceren met een alcoholpercentage die hoger is dan 20%

```
DELETE brouwers.*, bieren.*  
FROM brouwers INNER JOIN bieren  
ON brouwers.brouwnr=bieren.brouwnr  
WHERE brnaam = 'DECA'
```

verwijdert de brouwerij 'DECA' en alle bieren die in deze brouwerij geproduceerd worden. In SQLTryout en Access voert u dit in twee stappen uit. U verwijdert eerst de bieren en daarna de brouwer.

Belangrijk

Als u records hebt verwijderd met DELETE, kunt u de bewerking niet meer ongedaan maken. Als u wilt weten welke records zullen worden verwijderd, kunt u eerst het resultaat van een SELECT instructie met dezelfde voorwaarden onderzoeken en vervolgens de DELETE instructie uitvoeren.

6.6 Oefening DELETE

1. Verwijder alle rijen uit de tabel OFFERTES die betrekking hebben op de heesters van leverancier 021
2. Verwijder alle gegevens over de waterplanten in zowel de tabel PLANTEN als OFFERTES. Let op de volgorde van de behandeling.

7 BEHEER VAN TABELLEN EN RELATIES

7.1 CREATE TABLE

```
CREATE TABLE tabel  
(veld1 type [(grootte)] [NOT NULL] [index1]  
[, veld2 type [(grootte)] [NOT NULL] [index2] [, ...]]  
[, CONSTRAINT meervoudigeindex [, ...]])
```

De instructie CREATE TABLE bevat de volgende onderdelen:

Onderdeel	Beschrijving
tabel	De naam van de tabel die u wilt maken.
veld1, veld2	De naam van het veld of de velden die u in de tabel wilt opnemen. Er moet minstens één veld worden gemaakt.
type	Het gegevenstype van veld in de nieuwe tabel.
grootte	De veldgrootte in tekens (alleen voor tekst- en binaire velden).
index1, index2	Een component CONSTRAINT die een index met een enkelvoudig veld definieert.
Meervoudigeindex	Een component CONSTRAINT die een index met meervoudige velden opgeeft.

Met de instructie CREATE TABLE kunt u een nieuwe tabel inclusief velden en beperkende voorwaarden voor velden opgeven. Als NOT NULL is opgegeven voor een veld moet de inhoud van het veld steeds opgevuld zijn..

Met een component CONSTRAINT kunt u een aantal beperkingen opleggen aan een veld en kunt u de primaire sleutel bepalen. Het component CONSTRAINT komt in het volgende onderdeel aan bod. U kunt de primaire sleutel of overige indexen van bestaande tabellen ook bepalen met de instructie CREATE INDEX.

U kunt NOT NULL gebruiken in een enkelvoudig veld of binnen een benoemde component CONSTRAINT die geldt voor een benoemde CONSTRAINT met ofwel een enkelvoudig veld, danwel met meervoudige velden. De beperking NOT NULL kunt u echter slechts één keer toepassen op een veld, anders treden er runtime-fouten op.

De mogelijke data types

Naam	Opslagwijze	Betekenis
BINARY	1 byte per	Kan eender welke informatie bevatten, er gebeurt

	karakter	geen enkele vorm van conversie
BIT	1 byte	Kan enkel de waarde 0 of 1 bevatten
BYTE	1 byte	Kan een getal bevatten met een waarde tussen 0 en 255
MONEY	8 bytes	Kan een getal bevatten tussen -2^{63} en $2^{63}-1$. Getal wordt getoond met een muntcode.
DATETIME	8 bytes	Kan een datum en tijdswaarde bevatten tussen de jaren 100 en 9999.
SINGLE REAL	4 bytes	Kan een single-precision floating-point getal bevatten
DOUBLE FLOAT	8 bytes	Kan een double-precision floating-point getal bevatten
SHORT SMALLINT	2 bytes	Kan een getal bevatten met een waarde tussen -32768 en 32767
LONG INTEGER	4 bytes	Kan een getal bevatten met een waarde tussen -2147483648 en 2147483647
CHAR VARCHAR	1 byte per karakter	Kan een string bevatten. De opgegeven lengte mag tot 255 karakters bevatten.

```
Create table klanten (klantnr integer not null, klnaam
char(30) not null, kladres char(40), klpost char(4),
klgemeente char(40))
```

Creëert een nieuwe tabel klanten met de velden klantnr, klnaam, kladres, klpost, klgemeente. De velden klantnr en klnaam moeten een waarde bevatten.

7.2 Oefeningen CREATE TABLE

1. Creëer de tabel klachten met de volgende velden :

veld	data-type	lengte
klachtnr	numeriek	4
art_code	karakter	3
datum	datum	
klacht	karakter	100
Status	karakter	2

Vul de tabel KLACHTEN met de klacht nr. 1 die betrekking heeft op artikel-code 316. De klacht luidt : "PLANTEN (LEV_CODE 019) VERKOCHT OP 14-1-1999 VERTOONDEN BRUINE PLEKKEN NA CIRCA 2 MAANDEN".
De status is GL (gemeld aan leverancier) en de klacht wordt ingebracht op 15-3-1999.

7.3 DROP

Deze instructie verwijdert een bestaande tabel uit een database of verwijdert een bestaande index voor een tabel.

```
DROP {TABLE tabel | INDEX index ON tabel}
```

De instructie DROP bevat de volgende onderdelen:

Onderdeel	Beschrijving
Tabel	De naam van de tabel die u wilt verwijderen of de naam van de tabel waarvoor u een index wilt verwijderen.
index	De naam van de index die u wilt verwijderen voor tabel.

Voordat u een tabel of een index voor een tabel kunt verwijderen, moet u de tabel sluiten.

U kunt ook met ALTER TABLE een index voor een tabel verwijderen.

Met CREATE TABLE kunt u een tabel maken en met CREATE INDEX of ALTER TABLE kunt u een index maken. U wijzigt een tabel met ALTER TABLE.

```
drop table klanten
```

verwijdert de tabel klanten uit database.

7.4 ALTER

Met deze instructie wijzigt u het ontwerp van een tabel nadat deze is gemaakt met de instructie CREATE TABLE.

```
ALTER TABLE tabel  
{ADD {veldtype[(grootte)] [NOT NULL][CONSTRAINT index] |  
CONSTRAINT meervoudigeindex} |  
DROP {COLUMN veld | CONSTRAINT indexnaam} }
```

De instructie ALTER TABLE bevat de volgende onderdelen:

Onderdeel	Beschrijving
Tabel	De naam van de tabel die u wilt wijzigen.
Veld	De naam van het veld dat u wilt toevoegen aan of

	verwijderen uit de tabel.
Type	Het gegevenstype van veld.
grootte	De veldgrootte in tekens (alleen voor tekst- en binaire velden).
index	De index voor veld. Zie de component CONSTRAINT voor meer informatie over het maken van deze index.
Meervoudigeindex	De definitie van een index met meervoudige velden die u wilt toevoegen aan tabel. Zie het onderwerp CONSTRAINT (Component) voor meer informatie over het gebruik van deze component.
Indexnaam	De naam van de index met meervoudige velden die u wilt verwijderen.

Met de instructie ALTER TABLE kunt u een bestaande tabel op verschillende manieren wijzigen. U kunt de volgende dingen doen:

Een nieuw veld toevoegen aan de tabel met ADD COLUMN. U geeft de veldnaam, het gegevenstype en (voor tekst- en binaire velden) een optionele grootte op. De volgende instructie voegt bijvoorbeeld aan de tabel Brouwers een veld van 25 karakters met de naam Opmerkingen toe:

```
ALTER TABLE Brouwers ADD COLUMN Opmerkingen CHAR(25)
```

U kunt ook een index definiëren op dat veld. Zie CONSTRAINT pagina 44.

Als u NOT NULL opgeeft voor een veld, dan moeten bij nieuwe records geldige gegevens in dat veld worden ingevoerd.

Een index met meervoudige velden toevoegen met ADD CONSTRAINT. Zie CONSTRAINT pagina 44.

Een veld verwijderen met DROP COLUMN. U hoeft alleen de naam van het veld op te geven.

Een index met meervoudige velden verwijderen met DROP CONSTRAINT. U geeft alleen het gereserveerde woord CONSTRAINT op, gevolgd door de indexnaam.

Opmerkingen

- U kunt per keer slechts één veld of index toevoegen of verwijderen.
- Met de instructie CREATE INDEX kunt u index met een enkelvoudig veld of met meervoudige velden toevoegen aan een tabel. Met de instructie ALTER TABLE of DROP kunt u een index verwijderen die is gemaakt met ALTER TABLE of CREATE INDEX.

- U kunt NOT NULL gebruiken in een enkelvoudig veld of binnen een benoemde component CONSTRAINT die geldt voor een benoemde CONSTRAINT met ofwel een enkelvoudig veld, dan wel met meervoudige velden. De beperking NOT NULL kunt u echter slechts één keer toepassen op een veld, anders treden er runtime-fouten op.

```
alter table brouwers drop column opmerkingen
```

verwijdert de kolom opmerkingen uit de tabel Brouwers

7.5 Oefeningen Alter

1. Breid de tabel PLANTEN uit met een kolom VOORRAAD (numeriek 4 cijfers) voor het aantal stuks dat het tuincentrum van de desbetreffende plant nog in voorraad heeft.

7.6 CONSTRAINT

Een beperkende voorwaarde (constraint) komt overeen met een index maar kan ook worden gebruikt om een relatie met een andere tabel tot stand te brengen.

De component CONSTRAINT gebruikt u in de instructies ALTER TABLE en CREATE TABLE voor het maken of verwijderen van beperkende voorwaarden. Er zijn twee typen CONSTRAINT-componenten: één om een beperkende voorwaarde in te stellen op een enkelvoudig veld, en één om een beperkende voorwaarde in te stellen op meerdere velden.

Beperkende voorwaarde voor één veld:

```
CONSTRAINT naam {PRIMARY KEY | UNIQUE | NOT NULL | REFERENCES  
refererendetabel [(referendveld1)]}
```

Beperkende voorwaarde voor meerdere velden:

```
CONSTRAINT naam  
{PRIMARY KEY (primair1[, primair2 [, ...]]) |  
UNIQUE (uniek1[, uniek2 [, ...]]) |  
FOREIGN KEY (verw1[, verw2 [, ...]])  
REFERENCES refererendetabel [(referendveld1[,  
referendveld2[, ...]])]}
```

De component CONSTRAINT bevat de volgende onderdelen:

Onderdeel	Beschrijving
Naam	De naam van de beperkende voorwaarde die u wilt maken.
Primair1, primair2	De naam van het veld of de velden die u wilt aanwijzen als de primaire sleutel(s).
Uniek1, uniek2	De naam van het veld of de velden die u wilt

	aanwijzen als unieke sleutel(s).
Verw1, verw2	De naam van een of meer refererende sleutelvelden die verwijzen naar velden in een andere tabel.
Refererendetabel	De naam van de refererende tabel met het veld of de velden die worden aangegeven door refererendveld.
Refererendveld1, refererendveld2	De naam van het veld of de velden in refererendetabel die worden aangegeven door verw1, verw2. U kunt deze component weglaten als het veld waarnaar wordt verwezen, de primaire sleutel is van refererendetabel.

De syntax voor een beperkende voorwaarde voor één veld gebruikt u in de velddefinitie van een instructie ALTER TABLE of CREATE TABLE, direct na de opgave van het gegevenstype voor het veld.

De syntax voor een beperkende voorwaarde voor meerdere velden gebruikt u telkens wanneer u het gereserveerde woord CONSTRAINT buiten de velddefinitie in een instructie ALTER TABLE of CREATE TABLE toepast.

Met CONSTRAINT kunt u een veld aanwijzen als een veld met een van de volgende typen beperkende voorwaarden:

- Met het gereserveerde woord UNIQUE kunt u een veld aanwijzen als een unieke sleutel. Hiermee kunt u voorkomen dat twee records in de tabel dezelfde waarde hebben binnen hetzelfde veld. U kunt elk willekeurig veld of elke lijst met velden beperken als uniek. Als een beperkende voorwaarde voor meerdere velden is aangewezen als een unieke sleutel, moet de combinatie van waarden in alle velden van de index uniek zijn, zelfs als twee of meer records dezelfde waarde in slechts één van de velden hebben.
- Met de gereserveerde woorden PRIMARY KEY kunt u één veld of een groep velden in een tabel aanwijzen als de primaire sleutel. Alle waarden in de primaire sleutel moeten uniek en niet Null zijn. Voor een tabel kan slechts één primaire sleutel bestaan.

Opmerking

- U kunt geen beperkende voorwaarde PRIMARY KEY instellen voor een tabel die reeds een primaire sleutel heeft.
- Met de gereserveerde woorden FOREIGN KEY kunt u een veld aanwijzen als refererende sleutel. Als de primaire sleutel van de refererende tabel uit meerdere velden bestaat, moet u een beperkende voorwaarde voor meerdere velden definiëren, waarbij u de volgende zaken opgeeft: alle velden waarnaar wordt verwezen, de naam van de refererende tabel en de namen van de velden waarnaar wordt verwezen in de refererende tabel in dezelfde volgorde waarin deze velden in de tabel zijn opgenomen (REFERENCES). Als het veld of de velden waarnaar wordt verwezen, de primaire sleutel van de refererende tabel

vormen, hoeft u de velden waarnaar wordt verwezen, niet op te geven. De RDBMS neemt standaard aan dat de velden waarnaar wordt verwezen, de primaire sleutel van de refererende tabel vormen.

```
Create table klanten (klantnr integer not null constraint  
pk_klantnr primary key, klnaam char(30) not null)
```

Creëert een tabel klanten met de velden klantnr, die wij als primary key definiëren, en klnaam.

```
Create table gebruiker (nr integer not null constraint pk_nr  
primary key, naam char(30), uid char(8) constraint u_uid  
unique)
```

Creëert een tabel gebruikers met de velden nr (primary key), naam en uid. De inhoud van veld uid moet een unieke waarde hebben, toch is dit veld geen primary key.

De CONSTRAINT wordt ook gebruikt om relaties te definiëren tussen de tabellen.

Met de CREATE TABLE definiëren we de tabel bestellingen met de velden bestelnr (primary key), klantnr, besteldatum.

```
Create table bestellingen (bestelnr integer constraint  
pk_bestelnr primary key, klantnr integer constraint f_klantnr  
references klanten (klantnr), besteldatum datetime)
```

Tussen het klantnr van deze tabel en het klantnr van de tabel klanten wordt een 1 op n relatie gedefinieerd. Het data type van beide velden moeten hetzelfde zijn.

```
Create table bestellijn (bestelnr integer, biernr integer,  
aantal integer, constraint pk_bestbier primary key (bestelnr,  
biernr))
```

Creëert een tabel bestellijn bestaande uit velden bestelnr, biernr en aantal. Er wordt een samengestelde sleutel gedefinieerd op basis van de velden bestelnr en biernr.

```
Alter table bestellijn add constraint f_bestelnr foreign key  
(bestelnr) references bestellingen (bestelnr)
```

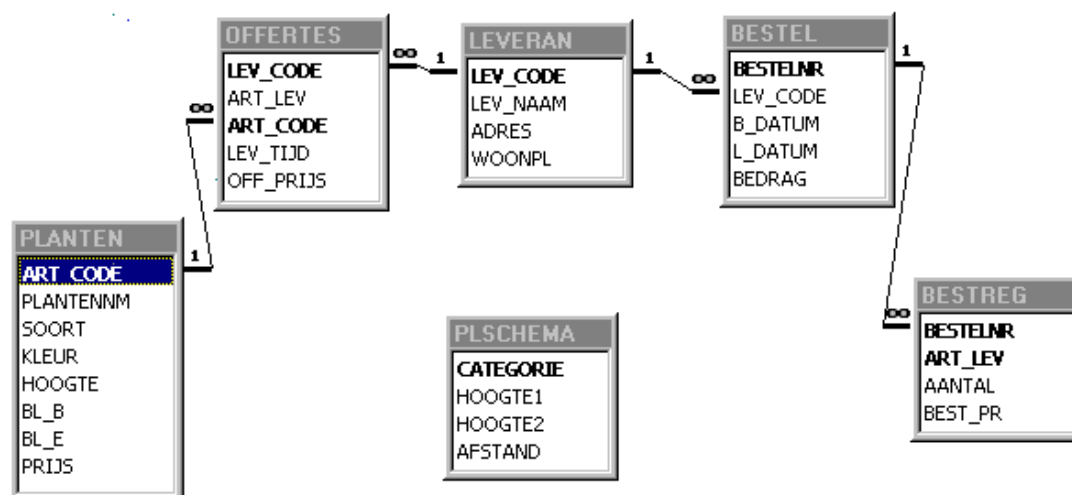
Definieert een 1-n relatie tussen de velden bestelnr van de tabel bestellingen en bestelnr van de tabel bestellijn.

```
Alter table bestellijn drop constraint f_bestelnr
```

Verwijdert de beperkende voorwaarde f_bestelnr

7.7 Oefeningen CONSTRAINT

1. Definieer de nodige relaties tussen de tabellen van de database PLANTV zoals de tekening dit aangeeft.



7.8 CREATE INDEX

Met deze instructie maakt u een nieuwe index voor een bestaande tabel. Indien u vaak gegevens zoekt op basis van een bepaald veld(en), dan is het aangewezen om op het veld(en) een index te plaatsen. Dit verhoogt aanzienlijk de responstijd.

```
CREATE [ UNIQUE ] INDEX index
ON tabel (veld [ASC|DESC][, veld [ASC|DESC], ...])
[WITH { PRIMARY | DISALLOW NULL | IGNORE NULL }]
```

De instructie CREATE INDEX bevat de volgende onderdelen:

Onderdeel	Beschrijving
index	De naam van de index die u wilt maken.
tabel	De naam van de bestaande tabel waarvoor de index wordt gemaakt.
veld	De naam van het veld of de velden die u in de index wilt opnemen. Als u een enkelvoudige index wilt maken, plaatst u de veldnaam tussen haakjes achter de tabelnaam. Als u een meervoudige index wilt maken, geeft u de naam op van elk veld dat in de index moet worden opgenomen. Als u een aflopende index wilt maken, neemt u het gereserveerde woord DESC op, anders wordt een oplopende index gemaakt.

Dubbele waarden in het geïndexeerde veld of de geïndexeerde velden van verschillende records kunt u voorkomen met het gereserveerde woord UNIQUE.

In de optionele WITH-component kunt u regels voor gegevensvalidatie opgeven:

Null-waarden in de geïndexeerde velden van nieuwe records voorkomt u met DISALLOW NULL

U kunt voorkomen dat records met Null-waarden in de geïndexeerde velden worden opgenomen in de index als u IGNORE NULL gebruikt.

Met PRIMARY kunt u het geïndexeerde veld of de geïndexeerde velden aanwijzen als de primaire sleutel. Dit impliceert een unieke sleutel, dus kunt u het gereserveerde woord UNIQUE dan weglaten.

U kunt ook met ALTER TABLE een enkelvoudige of meervoudige index toevoegen aan een tabel. Een index die is gemaakt met ALTER TABLE of CREATE INDEX, kunt u verwijderen met ALTER TABLE of met DROP.

Opmerking

Gebruik het gereserveerde woord PRIMARY niet als u een nieuwe index maakt voor een tabel die al over een primaire sleutel beschikt.

```
Create index i_naam on klanten (klnaam)
```

Creëert een index op het veld klnaam van de tabel klanten

Met de instructie DROP INDEX kunt u de index verwijderen.

```
Drop index i_naam on klanten
```

Verwijdert de index i_naam van de tabel klanten.

7.9 Oefeningen CREATE INDEX

1. Specificeer een index BSRIDX01 op de samengestelde sleutel bestelnr en art_lev van de tabel BESTREG.
2. De tabel OFFERTES wordt regelmatig geJOINED met de tabellen PLANTEN, LEVERAN en BESTREG. Welke indexen zijn waardevol voor deze tabel OFFERTES.

8 VIEWS

8.1 Create view

```
create view viewnaam  
as  
select-instructie
```

Een view is een select-instructie die bewaard wordt. Indien u gegevens wilt tonen van de view, wordt de overeenkomstige select-instructie terug opnieuw uitgevoerd. U kunt via de view ook gegevens toevoegen en wijzigen, dan worden deze gegevens in de overeenkomstige tabellen aangepast. Deze mogelijkheid wel afhankelijk van de opgegeven select-instructie.

```
create view toptien  
as  
select top 10 brouwernr, brnaam, omzet from brouwers  
order by omzet desc
```

maakt een view die een lijst van de 10 brouwers met de meeste omzet selecteert

Opmerkingen

De clause ORDER BY is enkel toegelaten in combinatie met de predikaten TOP n en TOP n PERCENT. Alle andere clauses zijn wel toegelaten, en ook het gebruik van aggregate functies en bewerkingen zijn toegelaten.

Access en SQLTryOut ondersteunen de clauses TOP n PERCENT, TOP n en ORDER BY niet bij views.

8.2 Drop view

Verwijdert de view uit de database.

```
drop view viewnaam  
  
drop view toptien
```

verwijdert de view toptien uit de database

Opmerking

Access kent de drop view instructie niet. Views verwijdert u met de instructie DROP TABLE viewnaam.

8.3 Oefeningen

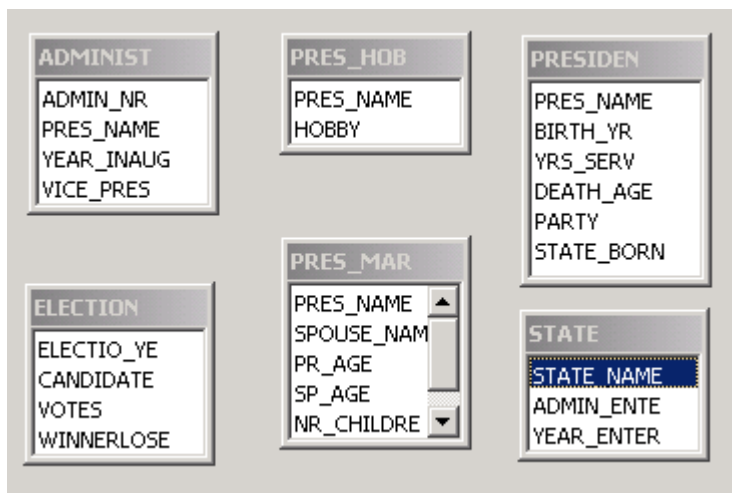
1. Definieer een view VASTLAAG waarin alle gegevens van alle vaste planten uit de tabel PLANTEN voorkomen met een hoogte van maximaal 15 cm.

2. Definieer een view GEM_O_PR met de kolommen art_code, MIN_OFF, MAX_OFF en GEM_OFF, waarin respectievelijk de artikelcode en per artikelcode de laagste, de hoogste en gemiddelde offerteprijs is vermeld.
3. Definieer een view LEV014 waarin alleen van leverancier 014 de volgende gegevens staan : ART_CODE, PLANTENNM, ART_LEV, OFF_PRIJS, PRIJS.
4. Maak een view BOMEN met de volgende gegevens : ART_CODE, PLANTENNM, HOOGTE, PRIJS. Uiteraard alleen de bomen selecteren.
5. Maak een view PLANTEN_IB waarin de volgende gegevens komen : BESTELNR, ART_LEV en PLANTENNM. Per samengestelde sleutel van de BESTREG tabel moet dus de plantennaam worden toegevoegd.
6. Maak een view ZOMERPL waarmee de gegevens ART_CODE, PLANTENNM, SOORT en PRIJS zijn te benaderen van alle planten die in de maanden juni, juli en augustus beginnen te bloeien.
7. Maak een view waardoor alleen de offertegegevens van de leveranciers uit LISSE zijn te selecteren.

9 EINDOEFENING

De database PRESIDEN bevat 6 tabellen:

ADMINIST	Bevat de lijst van presidenten en hun vice-president met het beginjaar van hun ambtstermijn.
ELECTION	Bevat de presidentskandidaten.
PRES_HOB	Bevat de hobby's van de presidenten
PRES_MAR	Bevat de huwelijken van de presidenten
PRESIDEN	Bevat de lijst van presidenten en de duurtijd van hun ambtstermijn
STATE	Bevat een lijst van alle staten en wanneer zij tot de USA toegetreden zijn.



1. Druk de namen van alle presidenten af met hun geboortjaar.
2. Druk de namen van de staten af waarin minstens 1 president geboren is.
3. Druk de namen van de staten af met het aantal presidenten die daar geboren werden.
4. Druk de namen af van de presidenten, huwelijksja(a)r(en), echtgenote(n) en aantal kinderen per huwelijk.
5. Druk de namen af van de presidenten die de meeste kinderen hebben.
6. Druk de namen af van de presidenten en het aantal maal dat zij gehuwd zijn/waren, en dit alleen voor die meer dan 1 maal gehuwd zijn/waren.
7. Druk de naam en de levensperiode (geboortjaar - sterfjaar) af van de presidenten(en) die vrijgezel bleven.
8. Druk de presidenten af gesorteerd per hobby.
9. Maak een view van de namen en dienstjaren van de presidenten die langer in dienst bleven dan de gemiddelde diensttijd van hun partijgenoten.
10. Druk de namen van de staten af die opgenomen werden in de unie tijdens het geboortjaar van de eerste echtgenote van de president(en) die op de meest jonge leeftijd in het huwelijksbootje trad(en).
11. Druk een lijst af van de gebeurtenissen :
jaar geboorte naam state

jaar	overlijden	naam	state
jaar	huwelijk	naam	spouse
jaar	winnaar verkiezingen	naam	party
jaar	verliezen verkiezingen	naam	party
jaar	toetreding state bij	state	huidige president
jaar	geboorte echtgenote	naam	echtgenoot

De lijst is gesorteerd volgens jaar.

12. Maak een view van de president(en) die nooit werd(en) verkozen.
13. Druk de namen af van de staten en het aantal presidenten af, van de staten waarin het meest aantal presidenten geboren werd.
14. Geef een lijst van alle presidenten, toon de naam, hoe vaak hij/zij gehuwd was, en hoeveel kinderen hij/zij had.
15. Geef een lijst van alle presidenten die gehuwd zijn tijdens hun ambtstermijn.
16. Geef een lijst van alle presidenten die een staat geboren zijn waar meer dan twee presidenten geboren zijn...

10COLOFON

Sectorverantwoordelijke: Ortaire Uyttersprot

Cursusverantwoordelijke: Jean Smits

Didactiek: Werkgroep Informatica Basis

Lay-out: Leyman Eugène

Medewerkers: Brigitte Van Ceulebroeck

Versie: 7/5/2010

Nummer dotatielijst: