

Het SQL Leerboek – zevende editie

De Installatiegids



Auteur: Rick F. van der Lans

Versie: 1.0

Datum: Februari 2012

Alle rechten voorbehouden. Alle auteursrechten en databankrechten ten aanzien van deze uitgave worden uitdrukkelijk voorbehouden. Deze rechten berusten bij de auteur.

Behoudens de in of krachtens de Auteurswet 1912 gestelde uitzonderingen, mag niets uit deze uitgave worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand of openbaar gemaakt in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voorzover het maken van reprografische verveelvoudigingen uit deze uitgave is toegestaan op grond van artikel 16 h Auteurswet 1912, dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht (postbus 3060, 2130 KB Hoofddorp, www.reprorecht.nl). Voor het overnemen van gedeelte(n) uit deze uitgave in bloemlezingen, readers en andere compilatiewerken (artikel 16 Auteurswet 1912) dient men zich te wenden tot de Stichting PRO (Stichting Publicatie- en Reproductierechten Organisatie, Postbus 3060, 2130 KB Hoofddorp, www.cedar.nl/pro). Voor het overnemen van een gedeelte van deze uitgave ten behoeve van commerciële doeleinden dient men zich te wenden tot de uitgever.

Hoewel aan de totstandkoming van deze uitgave de uiterste zorg is besteed, kan voor de afwezigheid van eventuele (druk)fouten en onvolledigheden niet worden ingestaan en aanvaarden de auteur(s), redacteur(en) en uitgever deswege geen aansprakelijkheid voor de gevolgen van eventueel voorkomende fouten en onvolledigheden.

Inleiding

1.1 Inleiding

Deze installatiegids bevat beschrijvingen van hoe de software, benodigd voor *Het SQL leerboek*, geïnstalleerd moet worden en hoe de voorbeelddatabase opgebouwd moet worden. Tevens zijn alle SQL-instructies uit het boek hier opgenomen zodat u deze niet hoeft in te tikken, maar eenvoudig weg kunt kopiëren.

Loop dit document stap voor stap door en controleer goed of u niets vergeet. De volgende stappen worden doorgelopen:

- Downloaden van MySQL
- Installeren van MySQL op Windows
- Downloaden van ODBC-driver
- Installeren van ODBC-driver op Windows
- Downloaden van WinSQL
- Installeren van WinSQL
- Starten van WinSQL
- Aanmaken van de voorbeelddatabase
- Herconstrueren van de voorbeelddatabase

Indien u vragen of opmerkingen op dit document hebt, laat ons dat weten via email adres sql@r20.nl

Downloaden en installeren van de software

2.1 Downloaden van MySQL

Een gratis versie van de MYSQL-databaseserver die in het grootste deel van Het SQL Leerboek gebruikt wordt, kan gedownload worden van de website www.mysql.com. U kunt daar MySQL downloaden voor diverse besturingssystemen, waaronder Windows, enkele Linux varianten, Sun Solaris, Mac OS X, HP-UX en FreeBSD. U kunt ook zelf bepalen welke versie u download. Wij gaan in dit boek uit van MySQL versie 5.5 en wel de *community* versie.

Deze website verandert nog weleens qua opmaak en structuur, daarom zullen we hier niet aangeven naar welk pagina u precies toe moet. Over het algemeen moet u de button volgen genaamd downloads.

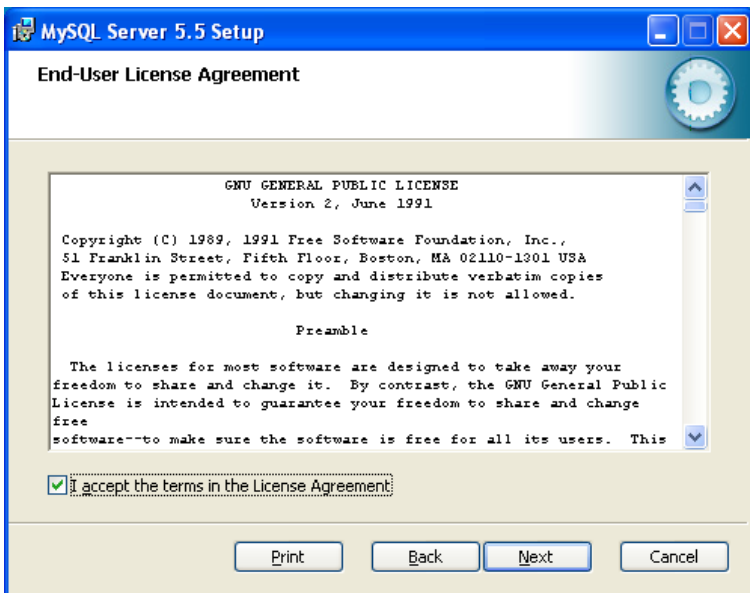
2.2 Installeren van MySQL op Windows

Indien u de software gedownload hebt, kunt u beginnen met het installeren. Voor installeren op Windows staan hier enkele aanwijzingen.

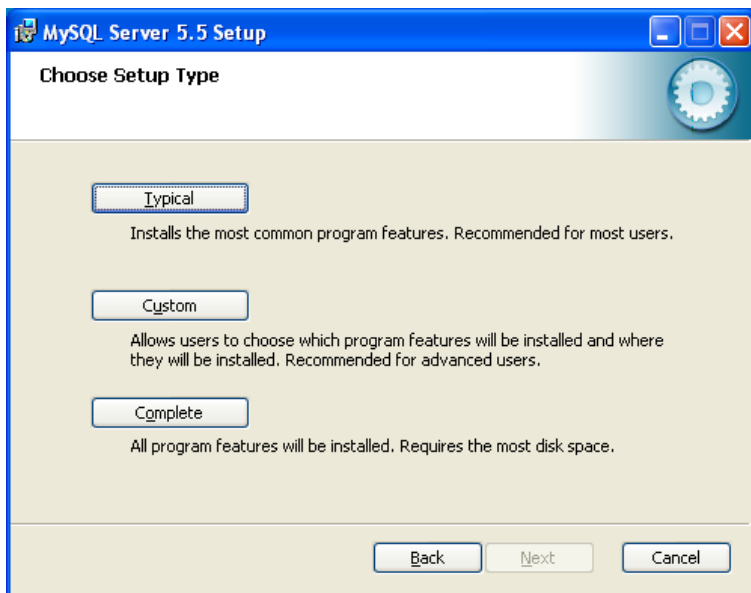
De installatie van MySQL start u door het bestand dat u gedownload hebt uit te voeren. Het volgende scherm verschijnt dan:



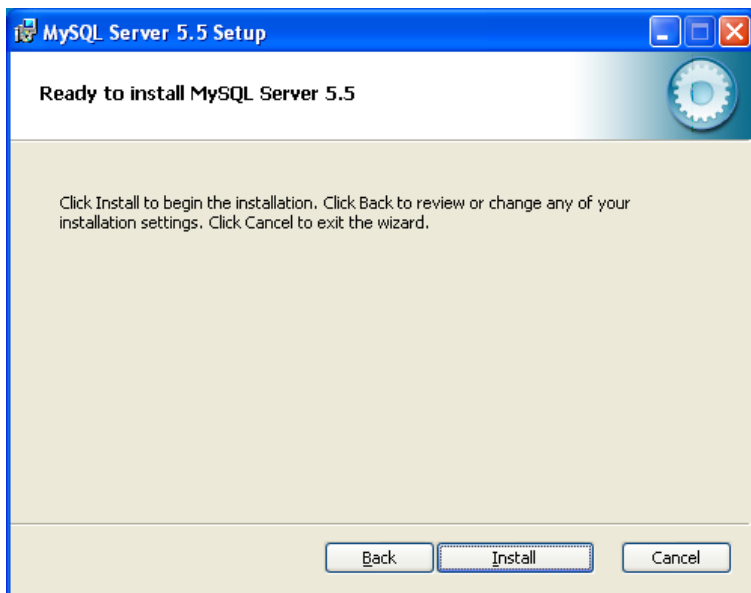
Kies voor *Next*:



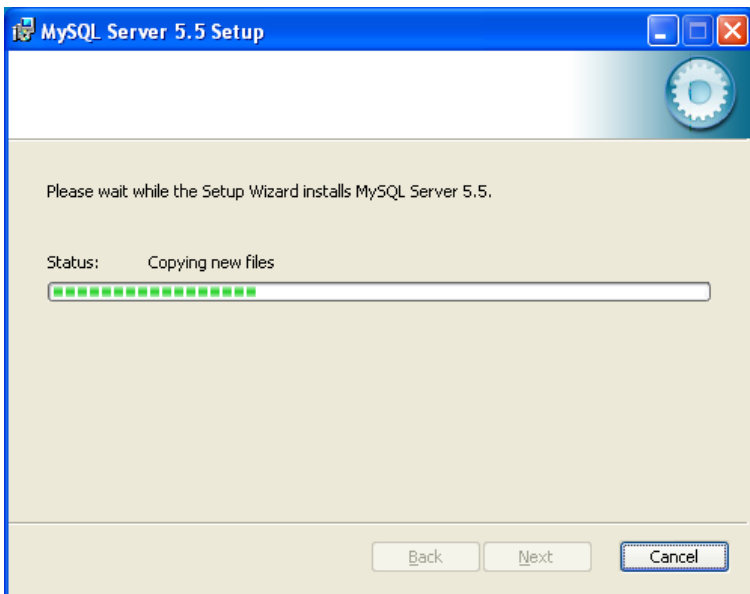
Indien u akkoord bent met de license agreement, kies dan voor *Next*:



Als u nieuw bent met MySQL, kies dan voor een *Typical* installatie. Kies hierna *Next*:



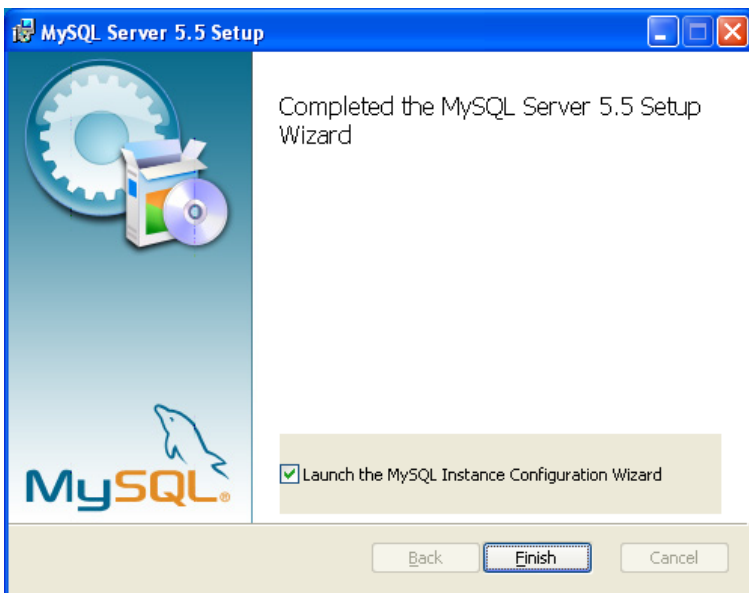
Klik op *Install* om de installatie werkelijk te starten. Het volgende scherm wordt getoond waarin u de voortgang van de installatie kunt volgen:



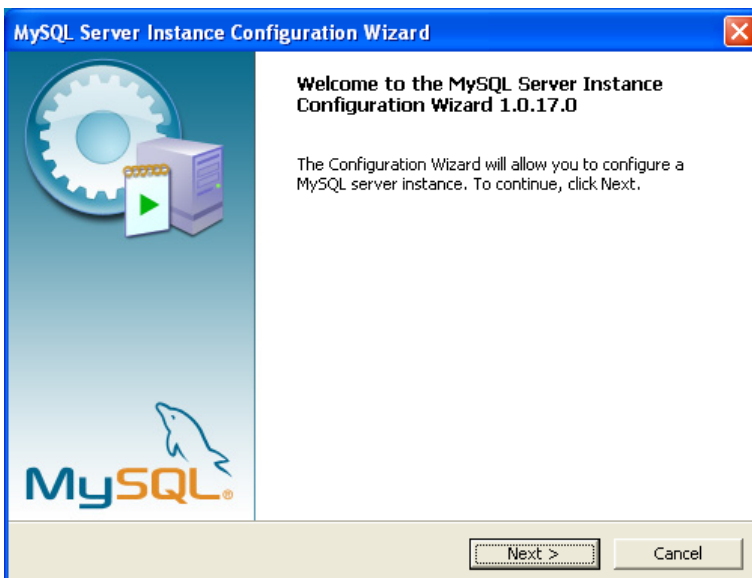
Is de installatie klaar, dan kan het zijn dat enkele reclame-achtige schermen verschijnen, zoals deze:



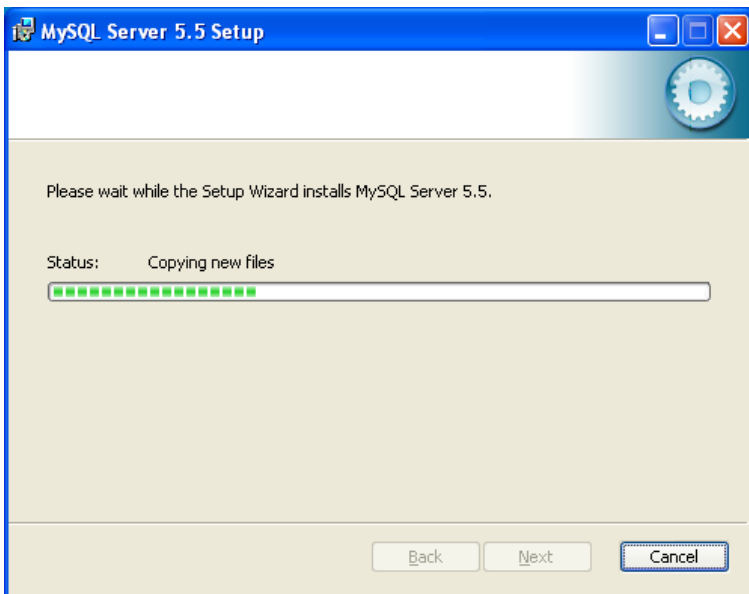
Blijf Next klikken totdat het onderstaande scherm verschijnt:



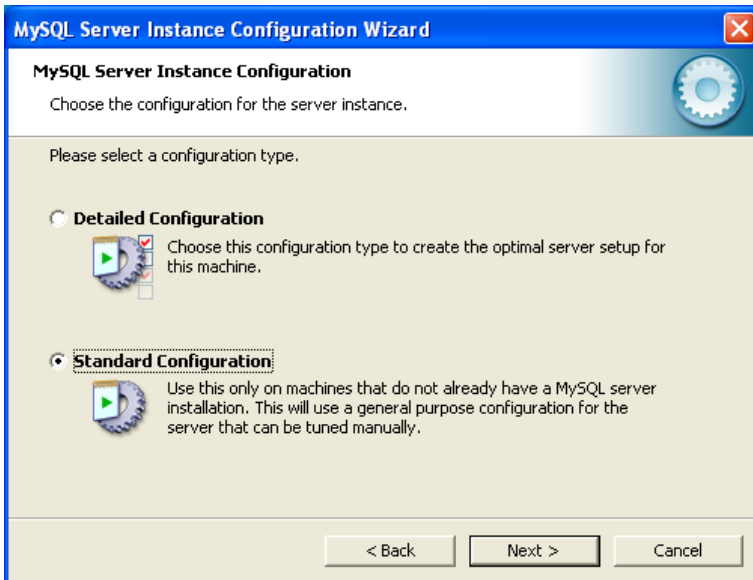
Nu is de installatie van MySQL voor een groot deel klaar. U moet alleen nog de *MySQL Instance Configuration Wizard* starten. Zorg dus dat deze onderaan het scherm aangevinkt is:



Klik op *Next* om de Wizard werkelijk te starten. Er zal getoond worden hoe de installatie vordert:



Tijdens dit proces zullen enkele vragen gestelde worden. De onderstaande vraag is de eerste:



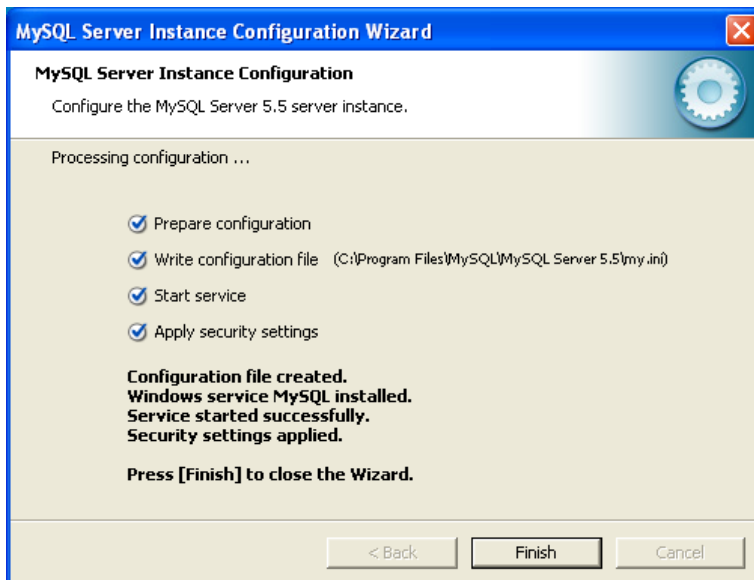
Kies hier voor *Standard Configuration*. Later kunt u dat altijd nog veranderen. Kies *Next* en de tweede vraag verschijnt. Onder Windows is het het eenvoudigst als u het als *Windows Service* installeert.



Plaats dus het correcte vinkje en kies *Next*. De derde vraag betreft de SQL-gebruiker die door MySQL aangeemaakt moet worden genaamd *root*. In dit scherm moet u het wachtwoord voor deze gebruiker bedenken. Wij raden sterk aan hier te kiezen voor het gelijknamige wachtwoord *root*. We gaan hier ook in het boek vanuit.



Vervolgens verschijnt het onderstaande scherm. Als u hier op *Finish* drukt is de installatie van MySQL voltooid.



Het is aan te raden om nu te controleren of inderdaad alles werkt. Start hiervoor de MS/DOS opdrachtprompt en ga naar de directory `c:\program files\mysql\mysql server 5.5\bin`. Tenminste, dat is de standaard directory waar MySQL geïnstalleerd wordt. Als uzelf een andere directory bedacht heeft, dient u daar naar te gaan.

Tik dan de volgende instructie in om de MySQL monitor te starten:

```
mysql -u root -p
```

De monitor vraagt dan om een wachtwoord, waarop u intikt `root` (of het wachtwoord dat u zelf verzonnen heeft). Als alles klopt verschijnt dan

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.5.8 MySQL Community Server (GPL)
```

```
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

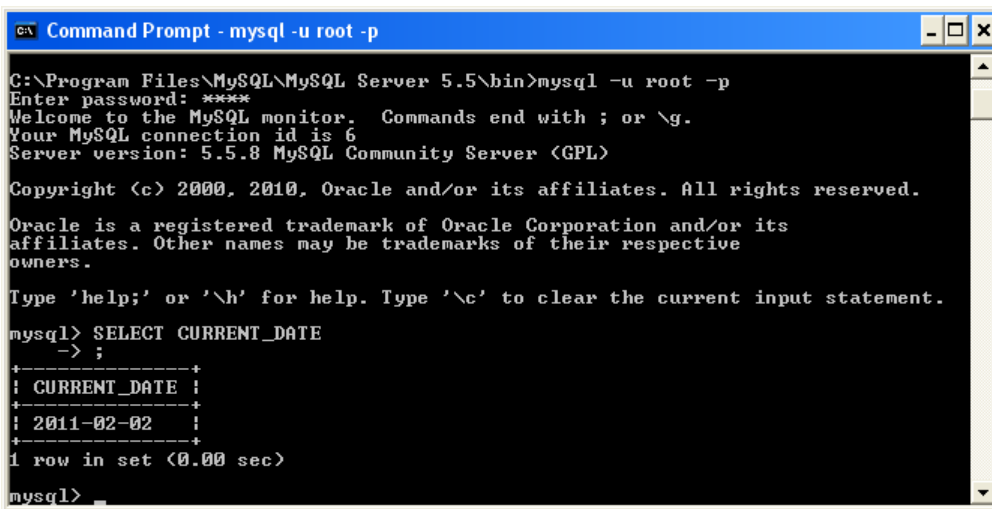
```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
mysql> SELECT CURRENT_DATE
```

Om te kijken of inderdaad SQL-instructies verwerkt kunnen worden, tikt u de volgende SQL-instructie in:

```
SELECT CURRENT_DATE;
```

MySQL toont dan vervolgens een kleine tabel met de dag van vandaag. Dit alles ziet er als volgt uit:



```

C:\Program Files\MySQL\MySQL Server 5.5\bin>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.5.8 MySQL Community Server <GPL>

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SELECT CURRENT_DATE
+-----+
| CURRENT_DATE |
+-----+
| 2011-02-02   |
+-----+
1 row in set (0.00 sec)

mysql>

```

Als dit allemaal werkt, dan kunt u met de instructie QUIT de MySQL minitor weer verlaten.

2.3 Downloaden van de ODBC-driver

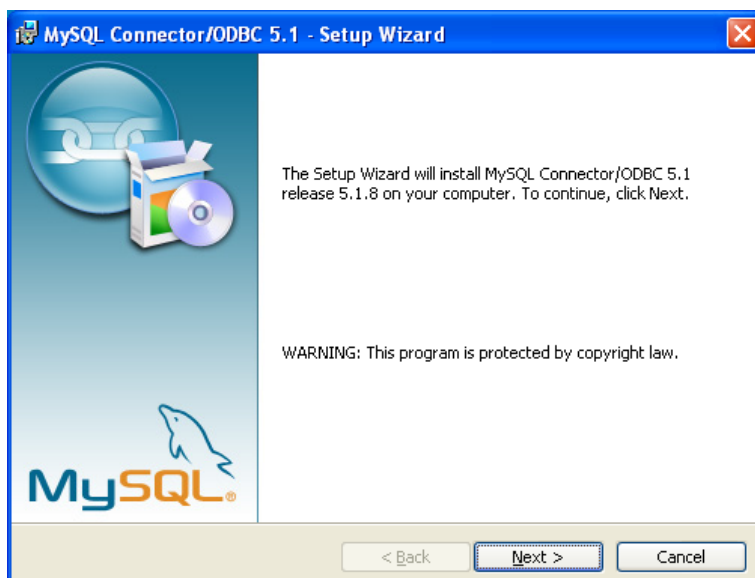
Voor het kunnen werken met diverse opvraagprogramma's zoals WinSQL, dient een speciale ODBC-driver gedownload te worden. Dit gebeurt dus niet automatisch als MySQL zelf geïnstalleerd wordt. Een gratis driver kan ook van de www.mysql.com website gedownload worden. In dit boek gaan we er van uit dat u Connector/ODBC versie 5.1 downloadt.

Deze website verandert nog weleens qua opmaak en structuur, daarom zullen we hier niet aangeven naar welk pagina u precies toe moet. Over het algemeen moet u de button volgen genaamd downloads.

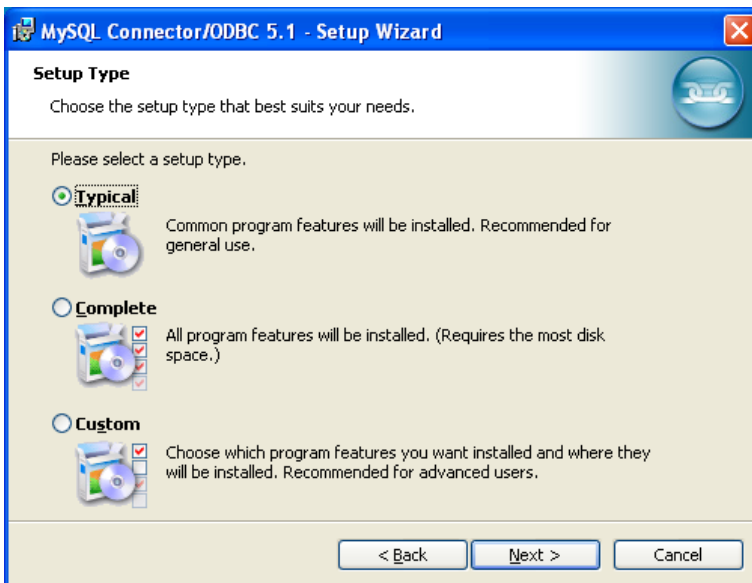
2.4 Installeren van de ODBC-driver op Windows

Indien u de software gedownload hebt, kunt u beginnen met het installeren. Voor installeren op Windows staan hier enkele aanwijzingen.

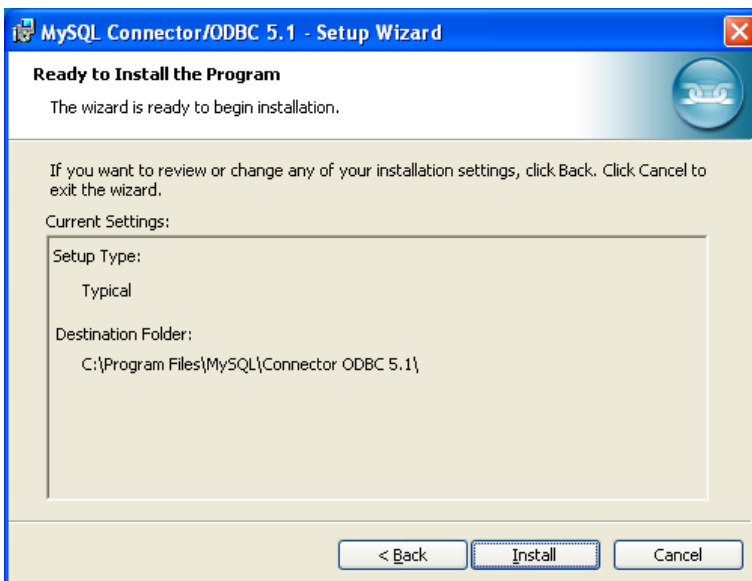
De installatie van MySQL start u door het bestand dat u gedownload hebt uit te voeren. Het volgende scherm verschijnt:



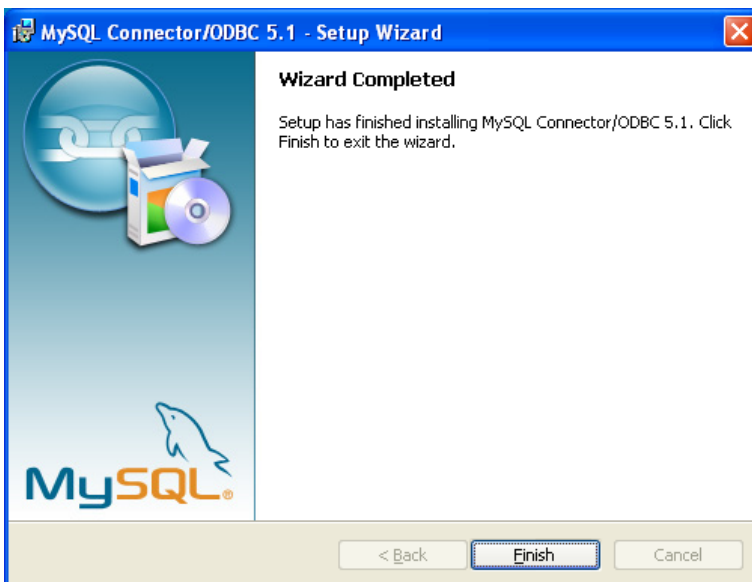
Kies *Next*:



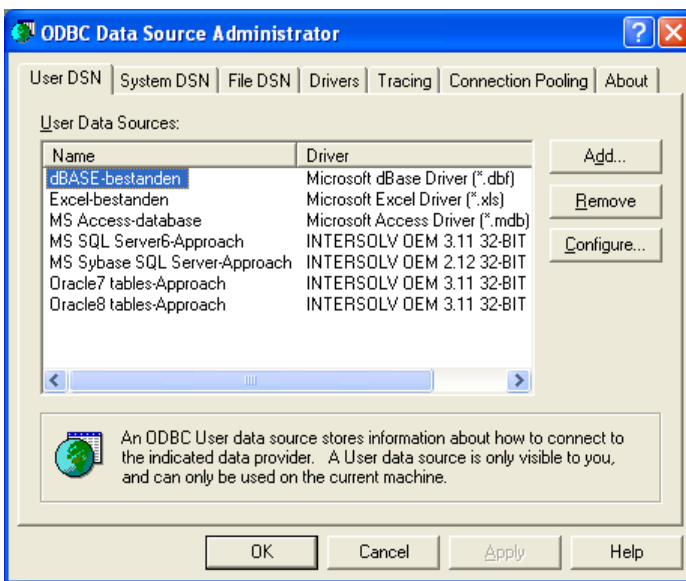
Selecteer *Typical* en kies *Next*:



Als u het met de settings eens bent, kies dan *Install* en de installatie begint. Als alles klaar is verschijnt het volgende scherm:

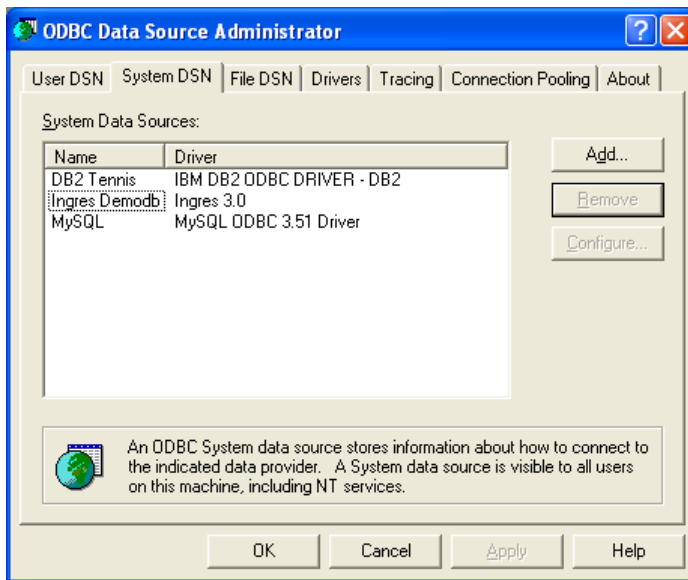


Na deze installatie is de ODBC-driver voor MySQL bekend bij Windows. Wat we wel nog moeten doen is zorgen dat alles gebruikers en programma's de driver kunnen vinden. Hiervoor moeten we de ODBC-driver manager van Windows gebruiken. Ga hiervoor naar het Configuratiescherm van Windows en selecteer Systeembeheer en kies dan Gegevensbronnen (ODBC). Het volgende scherm verschijnt dan:

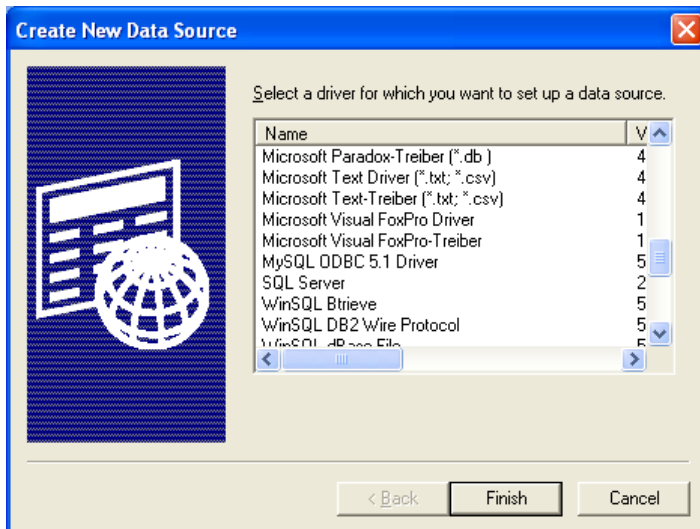


Hier staan alle ODBC-drivers die op uw systeem bekend zijn. Deze lijst in het bovenstaande scherm kan dus afwijken van die op uw machine.

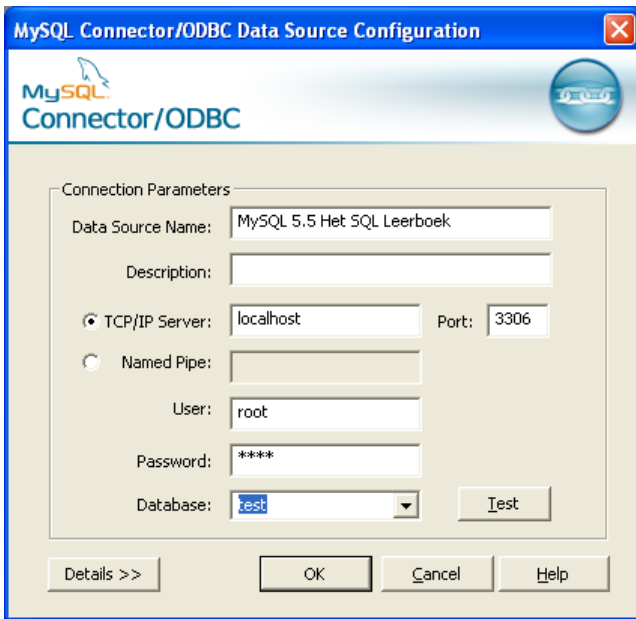
Ga naar het tabblad *System DSN*:



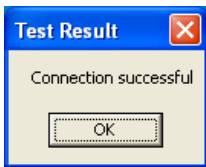
Hier gaan we de ODBC-driver voor MySQL toevoegen. Kies hiervoor *Next*. Alle drivers die dan bekend zijn worden getoond.



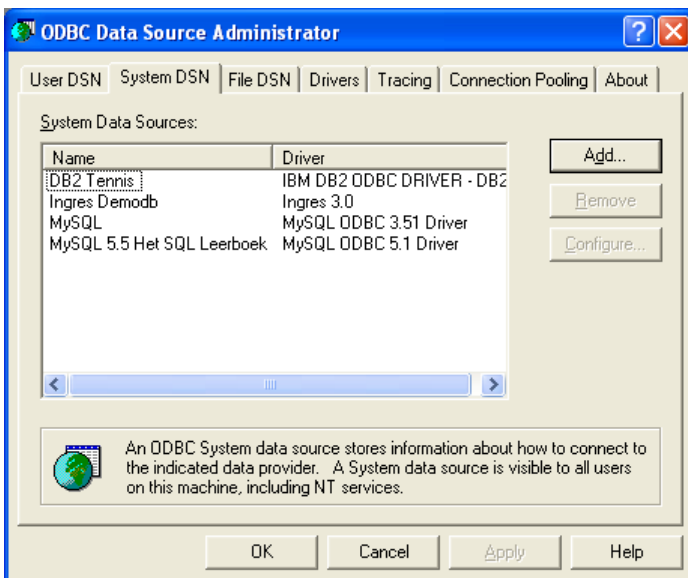
Kies nu MySQL ODBC 5.1 Driver en klik *Finish*. Er zal nu gevraagd worden naar enkele technische gegevens. U mag zelf een Data Source Name bedenken. Hier is gekozen voor MySQL 5.5 Het SQL Leerboek. Vul de velden verder in zoals hieronder. Klik daarna op *OK*.



Om te kijken of u alles correct ingetikt hebt, kunt u op *Test* klikken. Als alles goed werkt, verschijnt het volgende scherm:



Klik op *OK* en vervolgens weer op *OK* en dan wordt de driver toegevoegd aan de lijst van drivers die voor iedereen beschikbaar is:



De installatie van de ODBC-driver voor MySQL is nu klaar.

2.5 Downloaden van WinSQL

Voor het gemakkelijk kunnen invoeren van SQL-instructies is een opvraagprogramma nodig. Wij gaan in dit boek ervanuit dat u WinSQL gebruikt. Een gratis versie van WinSQL kan van de website www.synametrics.com gedownload worden. Download de 30-day trial of lite versie van WinSQL.

Deze website verandert nog weleens qua opmaak en structuur, daarom zullen we hier niet aangeven naar welk pagina u precies toe moet. Over het algemeen moet u de button volgen genaamd downloads.

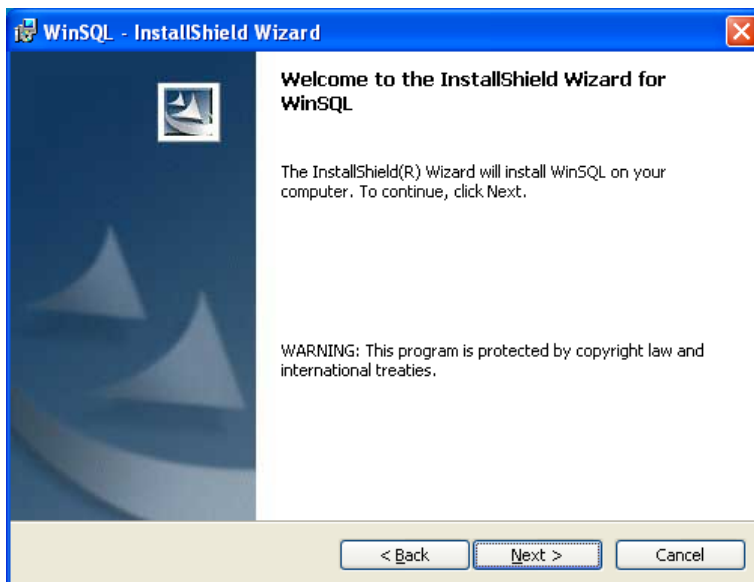
2.6 Installeren van WinSQL

Indien u de software gedownload hebt, kunt u beginnen met het installeren. Voor installeren op Windows staan hier enkele aanwijzingen.

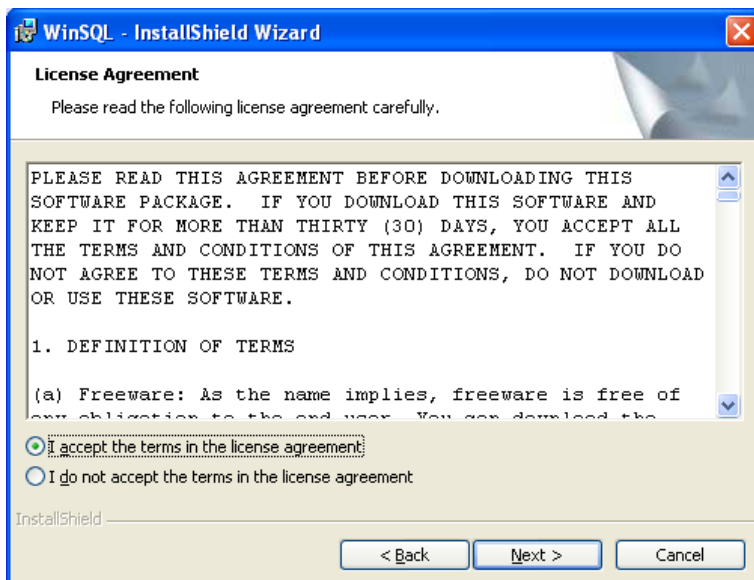
De installatie van WinSQL start u door het bestand dat u gedownload hebt uit te voeren. Het volgende scherm verschijnt dan:



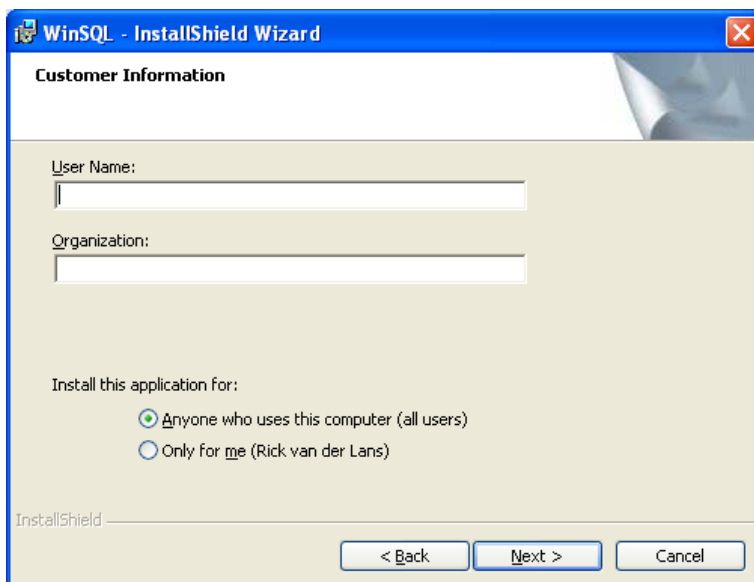
Kies *Next*:



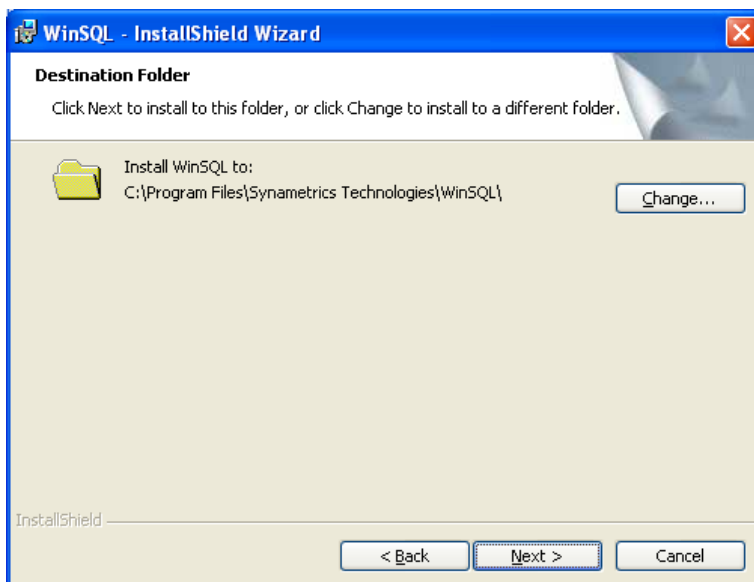
Kies *Next*. Er zal gevraagd worden of u met de license agreement akkoord gaat:



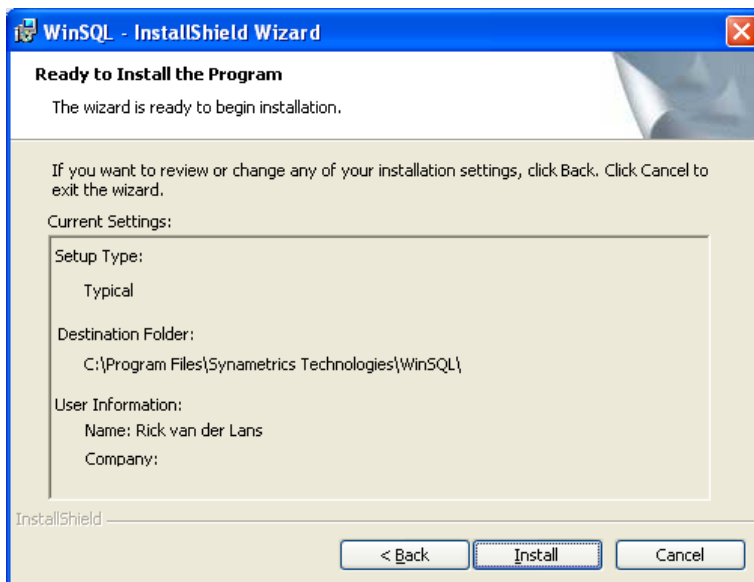
Kies *Next* als dat zo is. Een scherm verschijnt waarin u uw naam en bedrijf kunt invoeren:



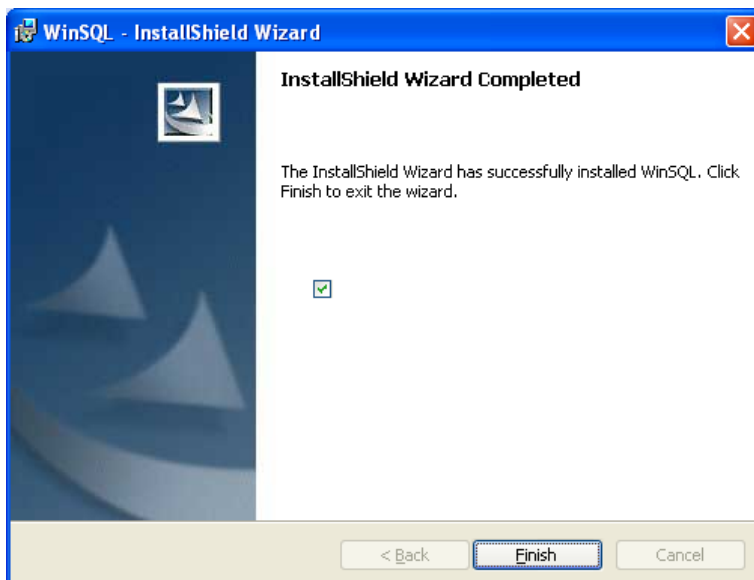
Kies *Next*:



Als u inderdaad WinSQL in de voorgestelde directory wilt installeren, kies dan *Next* en anders *Change*.



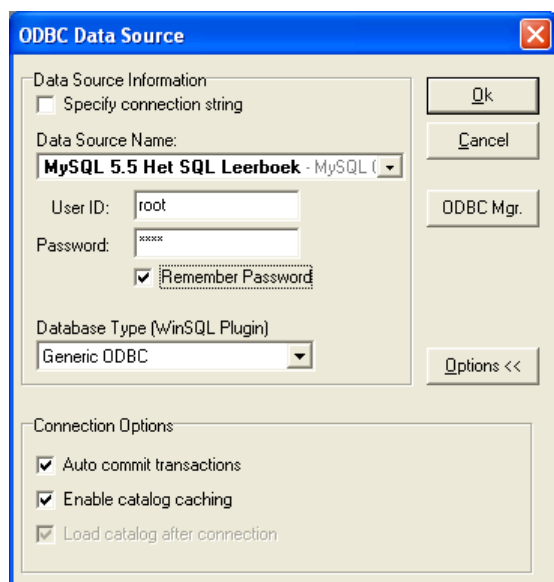
Verloopt de installatie correct, dan ziet u het volgende scherm:



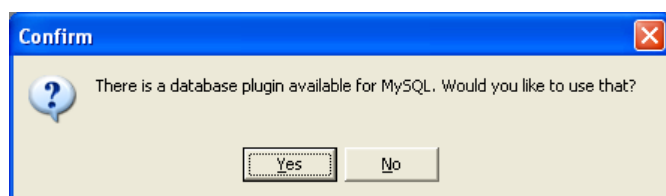
2.7 Starten van WinSQL

We gaan nu kijken of we inderdaad vanuit WinSQL SQL-instructies kunnen uitvoeren op de MySQL-databaseserver.

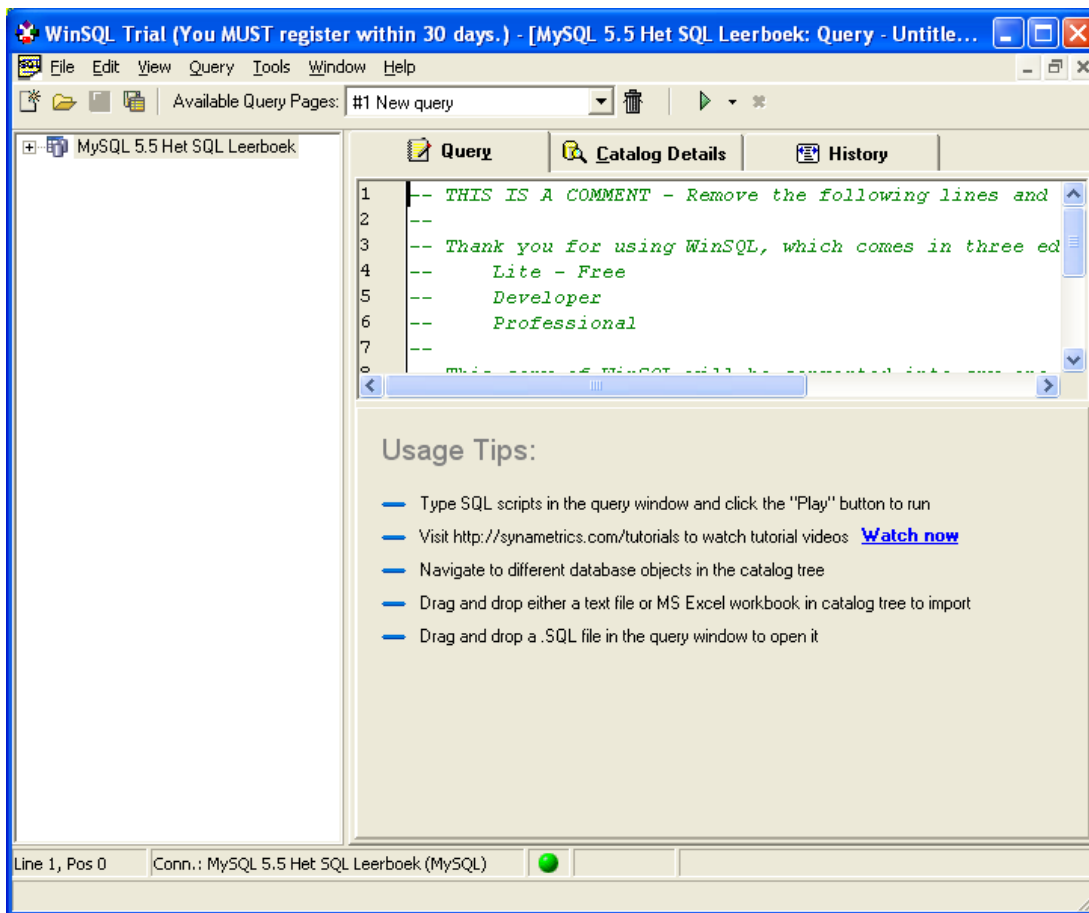
Begin met WinSQL te starten. Het eerste scherm dat verschijnt is het volgende:



Hiermee gaan we de correcte driver kiezen. Dit is uiteraard MySQL 5.5 Het SQL leerboek. Tik tevens in als user *root* en als password *root*. Klik vervolgens op *Ok*. Als het werkelijk de eerste keer is dat u WinSQL opstart verschijnt de volgende vraag:



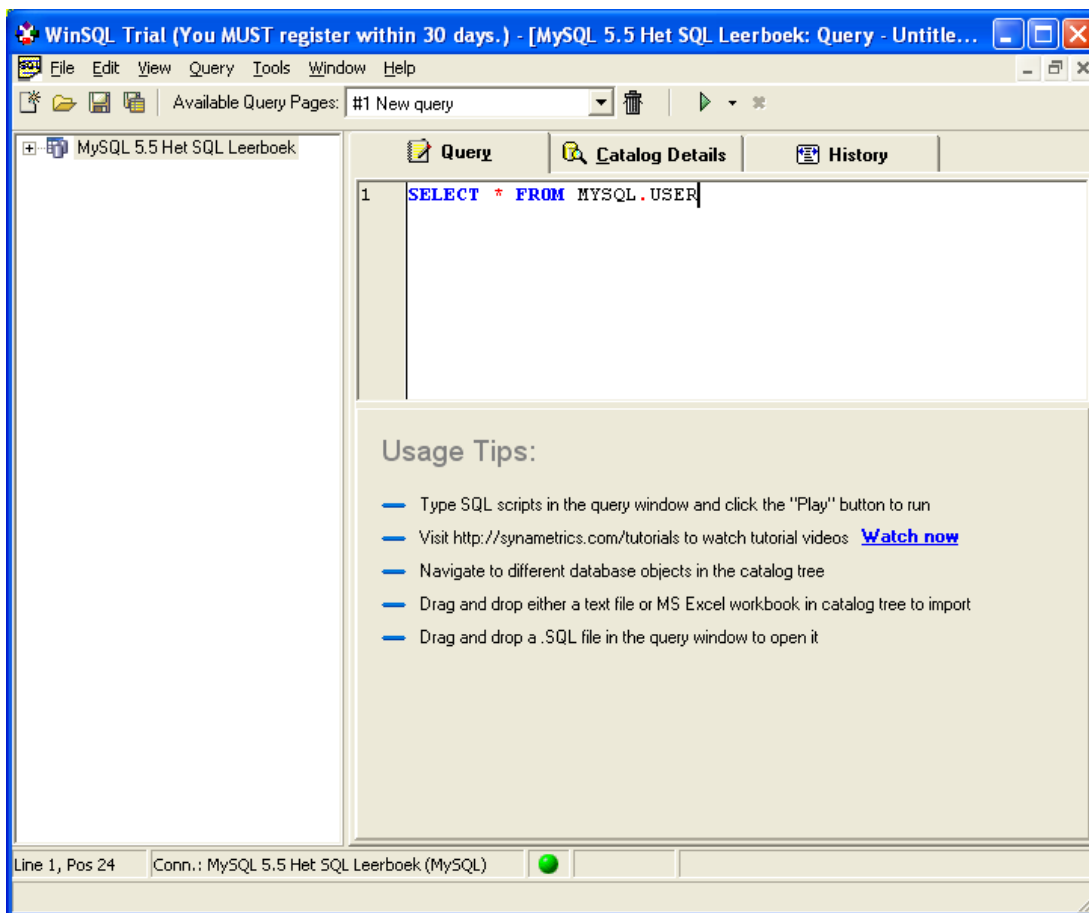
Database plugins komen met WinSQL mee, ze maken dat de toegang tot databaseservers wat sneller verloopt. Kies dus voor *Yes* en het volgende scherm verschijnt:



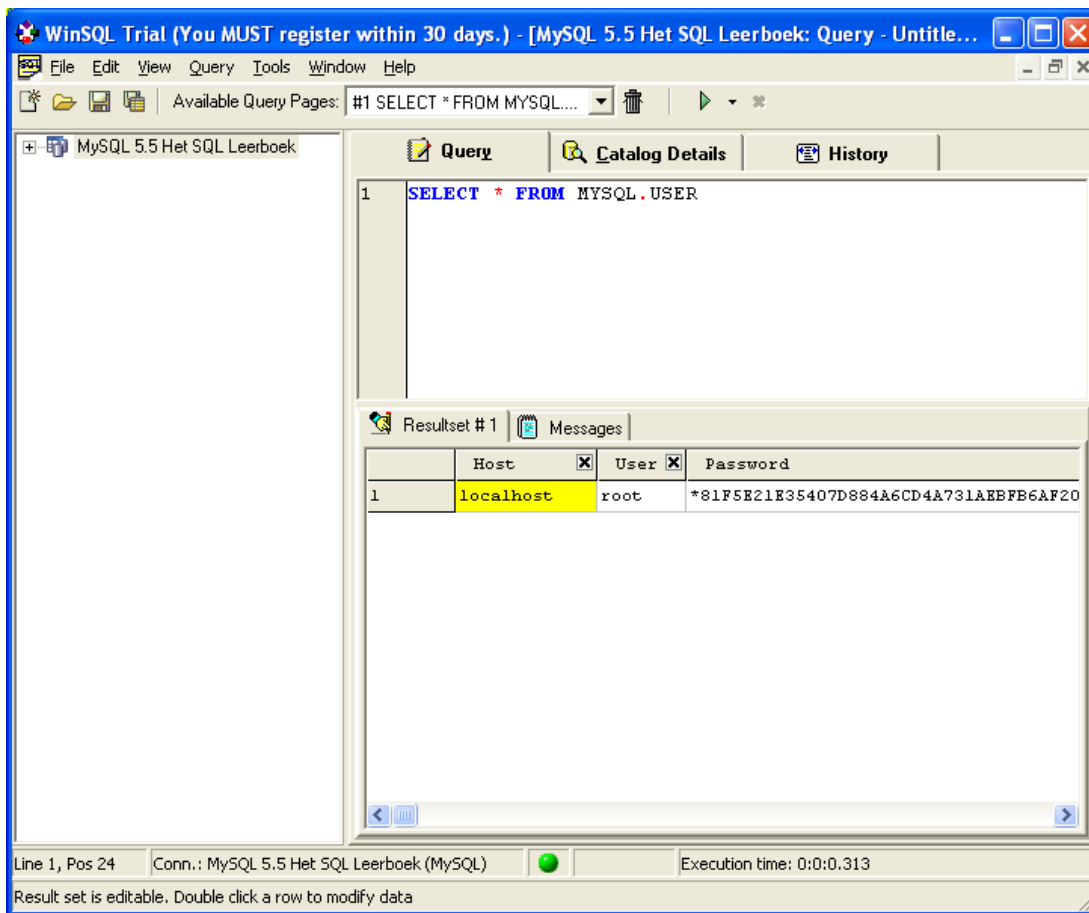
In het grote tekstvlak aan de rechterbovenkant verwijder alle groene letters. Dit kan door in het veld te gaan staan met de cursor, dan Control-A in te tikken en vervolgens de Delete-toets te gebruiken. Tik daarna de volgende SQL-instructie in:

```
SELECT * FROM MYSQL.USER
```

Het scherm ziet er dan als volgt uit:

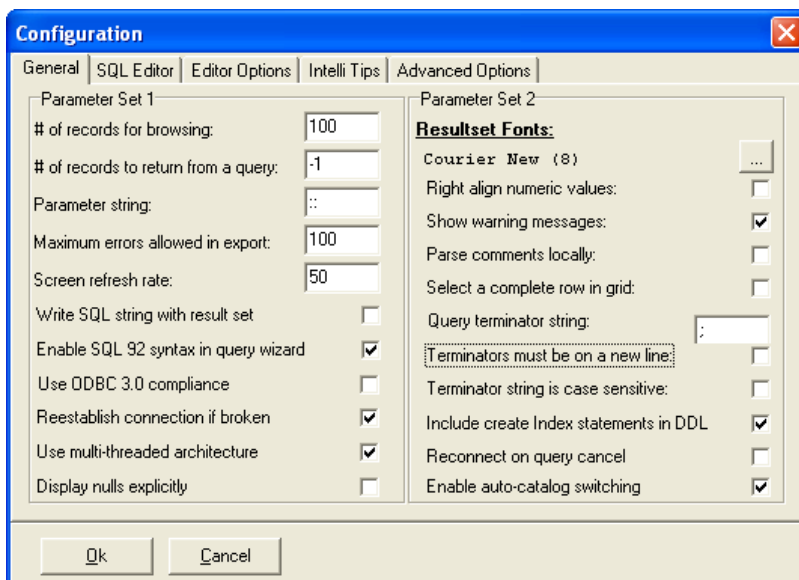


Klik nu op de groene pijl (▶) om de SQL-instructie te starten. WinSQL zal de instructie doorgeven aan MySQL en deze zal de instructie verwerken. Het resultaat zal WinSQL als volgt presenteren:



Als uw scherm er ongeveer uitziet als dit scherm, dan werkt alles!

Wat u nog wel MOET doen is zorgen dat bepaalde instellingen goed staan. Kies hiervoor *Edit* in het menu en kies dan *Options*. Het volgende scherm verschijnt:



Zorg dat in dit scherm de *Query terminator string* op een punt-komma staat en dat zorg dat *Terminators must be on a new line* uit staat. Doet u dit niet dan krijgt u later problemen met het verwerken van bepaalde SQL-instructies uit het boek.

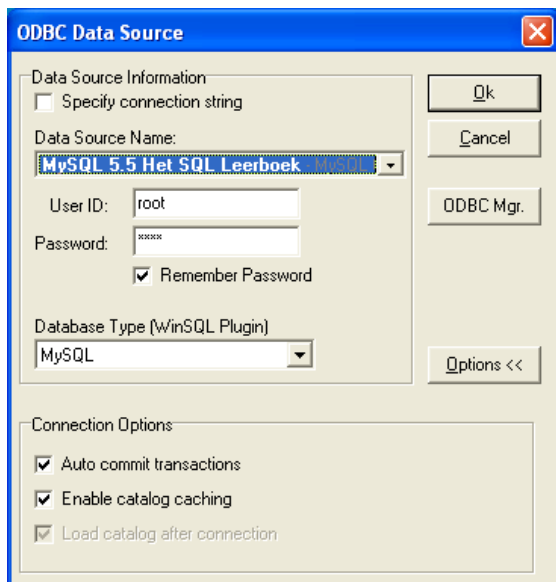
Heefu u beide instellingen aangepast, kies dan *OK*.

Aanmaken en herconstrueren van de voorbeelddatabase

3.1 Aanmaken van de voorbeelddatabase

Deze paragraaf bevat een beschrijving van het voor de eerste keer aanmaken van de voorbeelddatabase. Hiervoor dienen een reeks van SQL-instructies uitgevoerd te worden. In hoofdstuk 4 staat voor een groot deel uitgelegd welke instructies dit moeten zijn. Dit hoofdstuk bevat vergelijkbare instructies.

Als u nog geen gebruiker genaamd BOEKSQL hebt aangemaakt, log dan in met WinSQL onder user root met password root:



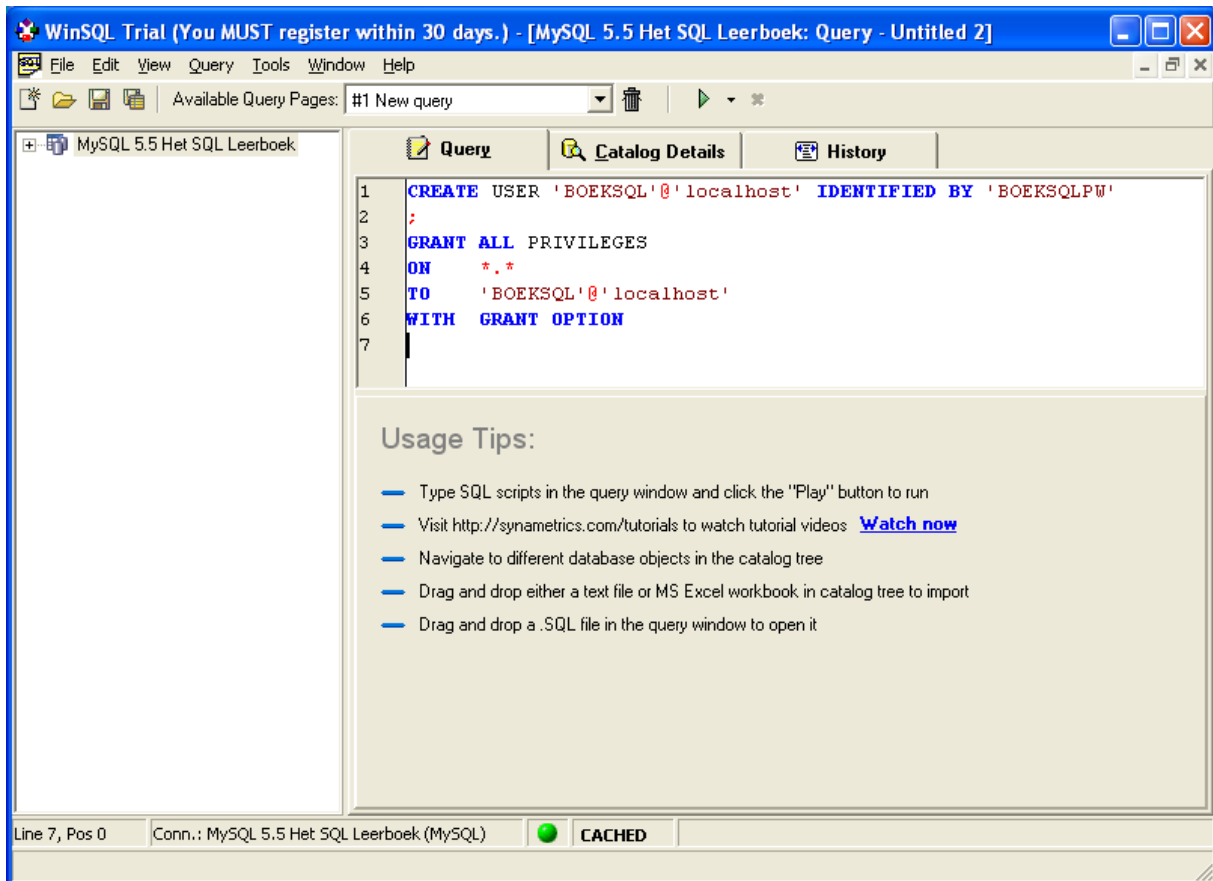
Introduceer vervolgens een nieuwe SQL-gebruiker genaamd BOEKSQL met als wachtwoord BOEKSQLPW, en geef BOEKSQL de mogelijkheden om zelf tabellen te creëren en te manipuleren. Kopieer hiervoor de volgende instructies naar het Query-tabblad van WinSQL:

```
CREATE USER 'BOEKSQL'@'localhost' IDENTIFIED BY 'BOEKSQLPW'  
;
```

```
GRANT ALL PRIVILEGES
ON *.*
TO 'BOEKSQ'@'localhost'
WITH GRANT OPTION
```

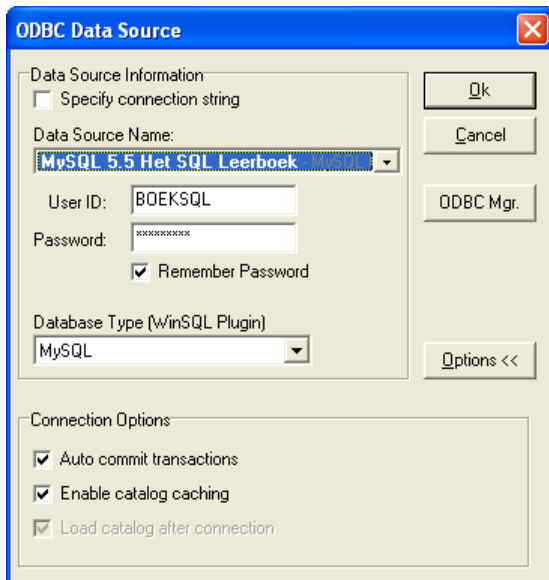
Opmerking: Er staat een puntkammo tussen de twee SQL-instructies. Dit is omdat de SQL-instructies te scheiden. WinSQL weet dan dat een nieuwe instructie begint.

Uw scherm ziet er ongeveer zo uit:



Voer de instructies uit door middel van de ▶ toets.

Log nu uit op WinSQL en log opnieuw in, maar nu als BOEKSQ met wachtwoord BOEKSQ'PW:



Nu kunnen alle tabellen aangemaakt worden en kunnen ze met gegevens gevuld worden. Hieronder staan alle benodigde SQL-instructies. Kopieer ze allemaal naar het Query-tabblad en verwerk ze. Hierna is de voorbeelddatabase gereed.

```

CREATE TABLE SPELERS
(SPELERSNR    INTEGER    NOT NULL,
NAAM         CHAR(15)   NOT NULL,
VOORLETTERS  CHAR(3)   NOT NULL,
GEB_DATUM    DATE
,
GESLACHT     CHAR(1)   NOT NULL,
JAARTOE      SMALLINT  NOT NULL,
STRAAT       VARCHAR(30) NOT NULL,
HUISNR       CHAR(4)
,
POSTCODE     CHAR(6)
,
PLAATS       VARCHAR(30) NOT NULL,
TELEFOON     CHAR(13)
,
BONDSNR      CHAR(4)
,
PRIMARY KEY  (SPELERSNR)
)
;
CREATE TABLE TEAMS
(TEAMNR       INTEGER    NOT NULL,
SPELERSNR    INTEGER    NOT NULL,
DIVISIE      CHAR(6)   NOT NULL,
PRIMARY KEY  (TEAMNR)
)
;
CREATE TABLE WEDSTRIJDEN
(WEDSTRIJDNR INTEGER    NOT NULL,
TEAMNR       INTEGER    NOT NULL,
SPELERSNR    INTEGER    NOT NULL,
GEWONNEN     SMALLINT  NOT NULL,
VERLOREN     SMALLINT  NOT NULL,
PRIMARY KEY  (WEDSTRIJDNR)
)
;
CREATE TABLE BOETES
(BETALINGSNR INTEGER    NOT NULL,
SPELERSNR    INTEGER    NOT NULL,
DATUM        DATE       NOT NULL,
BEDRAG       DECIMAL(7,2) NOT NULL,
PRIMARY KEY  (BETALINGSNR)
)
;
CREATE TABLE BESTUURSLEDEN
(SPELERSNR    INTEGER    NOT NULL,
BEGIN_DATUM   DATE       NOT NULL,
EIND_DATUM    DATE
,

```

```

        FUNCTIE          CHAR(20)
        PRIMARY KEY     (SPELERSNR, BEGIN_DATUM))
;
INSERT INTO SPELERS VALUES (
    2, 'Elfring', 'R', '1948-09-01', 'M', 1975, 'Steden',
    '43', '3575NH', 'Den Haag', '070-237893', '2411')
;
INSERT INTO SPELERS VALUES (
    6, 'Permentier', 'R', '1964-06-25', 'M', 1977, 'HazensteinIn',
    '80', '1234KK', 'Den Haag', '070-476537', '8467')
;
INSERT INTO SPELERS VALUES (
    7, 'Wijers', 'GWS', '1963-05-11', 'M', 1981, 'Erasmusweg',
    '39', '9758VB', 'Den Haag', '070-347689', NULL)
;
INSERT INTO SPELERS VALUES (
    8, 'Niewenburg', 'B', '1962-07-08', 'V', 1980, 'Sporlaan',
    '4', '6584W0', 'Rijswijk', '070-458458', '2983')
;
INSERT INTO SPELERS VALUES (
    27, 'Cools', 'DD', '1964-12-28', 'V', 1983, 'Liespad',
    '804', '8457DK', 'Zoetermeer', '079-234857', '2513')
;
INSERT INTO SPELERS VALUES (
    28, 'Cools', 'C', '1963-06-22', 'V', 1983, 'Oudegracht',
    '10', '1294QK', 'Leiden', '010-659599', NULL)
;
INSERT INTO SPELERS VALUES (
    39, 'Bischoff', 'D', '1956-10-29', 'M', 1980, 'Ericaplein',
    '78', '9629CD', 'Den Haag', '070-393435', NULL)
;
INSERT INTO SPELERS VALUES (
    44, 'Bakker, de', 'E', '1963-01-09', 'M', 1980, 'Lawaaistraat',
    '23', '4444LJ', 'Rijswijk', '070-368753', '1124')
;
INSERT INTO SPELERS VALUES (
    57, 'Bohemen, van', 'M', '1971-08-17', 'M', 1985, 'Erasmusweg',
    '16', '4377CB', 'Den Haag', '070-473458', '6409')
;
INSERT INTO SPELERS VALUES (
    83, 'Hofland', 'PK', '1956-11-11', 'M', 1982, 'Mariakade',
    '16a', '1812UP', 'Den Haag', '070-353548', '1608')
;
INSERT INTO SPELERS VALUES (
    95, 'Meuleman', 'P', '1963-05-14', 'M', 1972, 'Hoofdweg',
    '33a', '57460P', 'Voorburg', '070-867564', NULL)
;
INSERT INTO SPELERS VALUES (
    100, 'Permentier', 'P', '1963-02-28', 'M', 1979, 'HazensteinIn',
    '80', '6494SG', 'Den Haag', '070-494593', '6524')
;
INSERT INTO SPELERS VALUES (
    104, 'Moerman', 'D', '1970-05-10', 'V', 1984, 'Stoutlaan',
    '65', '9437A0', 'Zoetermeer', '079-987571', '7060')
;
INSERT INTO SPELERS VALUES (
    112, 'Baalen, van', 'IP', '1963-10-01', 'V', 1984, 'Vosseweg',
    '8', '6392LK', 'Rotterdam', '010-548745', '1319')
;
INSERT INTO TEAMS VALUES (1, 6, 'ere')
;
INSERT INTO TEAMS VALUES (2, 27, 'tweede')
;
INSERT INTO WEDSTRIJDEN VALUES ( 1, 1, 6, 3, 1)
;
INSERT INTO WEDSTRIJDEN VALUES ( 2, 1, 6, 2, 3)
;

```

```

INSERT INTO WEDSTRIJDEN VALUES ( 3, 1, 6, 3, 0)
;
INSERT INTO WEDSTRIJDEN VALUES ( 4, 1, 44, 3, 2)
;
INSERT INTO WEDSTRIJDEN VALUES ( 5, 1, 83, 0, 3)
;
INSERT INTO WEDSTRIJDEN VALUES ( 6, 1, 2, 1, 3)
;
INSERT INTO WEDSTRIJDEN VALUES ( 7, 1, 57, 3, 0)
;
INSERT INTO WEDSTRIJDEN VALUES ( 8, 1, 8, 0, 3)
;
INSERT INTO WEDSTRIJDEN VALUES ( 9, 2, 27, 3, 2)
;
INSERT INTO WEDSTRIJDEN VALUES (10, 2, 104, 3, 2)
;
INSERT INTO WEDSTRIJDEN VALUES (11, 2, 112, 2, 3)
;
INSERT INTO WEDSTRIJDEN VALUES (12, 2, 112, 1, 3)
;
INSERT INTO WEDSTRIJDEN VALUES (13, 2, 8, 0, 3)
;
INSERT INTO BOETES VALUES (1, 6, '1980-12-08', 100)
;
INSERT INTO BOETES VALUES (2, 44, '1981-05-05', 75)
;
INSERT INTO BOETES VALUES (3, 27, '1983-09-10', 100)
;
INSERT INTO BOETES VALUES (4, 104, '1984-12-08', 50)
;
INSERT INTO BOETES VALUES (5, 44, '1980-12-08', 25)
;
INSERT INTO BOETES VALUES (6, 8, '1980-12-08', 25)
;
INSERT INTO BOETES VALUES (7, 44, '1982-12-30', 30)
;
INSERT INTO BOETES VALUES (8, 27, '1984-11-12', 75)
;
INSERT INTO BESTUURSLEDEN VALUES ( 6, '1990-1-1', '1990-12-31', 'Secretaris')
;
INSERT INTO BESTUURSLEDEN VALUES ( 6, '1991-1-1', '1992-12-31', 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES ( 6, '1992-1-1', '1993-12-31', 'Penningmeester')
;
INSERT INTO BESTUURSLEDEN VALUES ( 6, '1993-1-1', NULL, 'Voorzitter')
;
INSERT INTO BESTUURSLEDEN VALUES ( 2, '1990-1-1', '1992-12-31', 'Voorzitter')
;
INSERT INTO BESTUURSLEDEN VALUES ( 2, '1994-1-1', NULL, 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES (112, '1992-1-1', '1992-12-31', 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES (112, '1994-1-1', NULL, 'Secretaris')
;
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1990-1-1', '1990-12-31', 'Penningmeester')
;
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1991-1-1', '1991-12-31', 'Secretaris')
;
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1993-1-1', '1993-12-31', 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1994-1-1', NULL, 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES ( 57, '1992-1-1', '1992-12-31', 'Secretaris')
;
INSERT INTO BESTUURSLEDEN VALUES ( 27, '1990-1-1', '1990-12-31', 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES ( 27, '1991-1-1', '1991-12-31', 'Penningmeester')

```

```

;
INSERT INTO BESTUURSLEDEN VALUES ( 27, '1993-1-1', '1993-12-31', 'Penningmeester')
;
INSERT INTO BESTUURSLEDEN VALUES ( 95, '1994-1-1', NULL, 'Penningmeester')
;

```

3.2 Herconstrueren van de inhoud van de tabellen van de voorbeelddatabase

Door sommige SQL-instructies in het boek verandert de inhoud van een of meer tabellen. Dit kan later problemen geven in het boek. Om die reden is het soms noodzakelijk de inhoud van elke tabel te verwijderen en weer opnieuw te vullen. Gebruik hiervoor de volgende instructies:

```

DELETE FROM BESTUURSLEDEN
;
DELETE FROM WEDSTRIJDEN
;
DELETE FROM BOETES
;
DELETE FROM TEAMS
;
DELETE FROM SPELERS
;
INSERT INTO TEAMS VALUES (1, 6, 'ere')
;
INSERT INTO TEAMS VALUES (2, 27, 'tweede')
;
INSERT INTO WEDSTRIJDEN VALUES ( 1, 1, 6, 3, 1)
;
INSERT INTO WEDSTRIJDEN VALUES ( 2, 1, 6, 2, 3)
;
INSERT INTO WEDSTRIJDEN VALUES ( 3, 1, 6, 3, 0)
;
INSERT INTO WEDSTRIJDEN VALUES ( 4, 1, 44, 3, 2)
;
INSERT INTO WEDSTRIJDEN VALUES ( 5, 1, 83, 0, 3)
;
INSERT INTO WEDSTRIJDEN VALUES ( 6, 1, 2, 1, 3)
;
INSERT INTO WEDSTRIJDEN VALUES ( 7, 1, 57, 3, 0)
;
INSERT INTO WEDSTRIJDEN VALUES ( 8, 1, 8, 0, 3)
;
INSERT INTO WEDSTRIJDEN VALUES ( 9, 2, 27, 3, 2)
;
INSERT INTO WEDSTRIJDEN VALUES (10, 2, 104, 3, 2)
;
INSERT INTO WEDSTRIJDEN VALUES (11, 2, 112, 2, 3)
;
INSERT INTO WEDSTRIJDEN VALUES (12, 2, 112, 1, 3)
;
INSERT INTO WEDSTRIJDEN VALUES (13, 2, 8, 0, 3)
;
INSERT INTO BOETES VALUES (1, 6, '1980-12-08', 100)
;
INSERT INTO BOETES VALUES (2, 44, '1981-05-05', 75)
;
INSERT INTO BOETES VALUES (3, 27, '1983-09-10', 100)
;
INSERT INTO BOETES VALUES (4, 104, '1984-12-08', 50)
;
INSERT INTO BOETES VALUES (5, 44, '1980-12-08', 25)
;
INSERT INTO BOETES VALUES (6, 8, '1980-12-08', 25)
;

```



```

INSERT INTO BOETES VALUES (7, 44, '1982-12-30', 30)
;
INSERT INTO BOETES VALUES (8, 27, '1984-11-12', 75)
;
INSERT INTO BESTUURSLEDEN VALUES ( 6, '1990-1-1', '1990-12-31', 'Secretaris')
;
INSERT INTO BESTUURSLEDEN VALUES ( 6, '1991-1-1', '1992-12-31', 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES ( 6, '1992-1-1', '1993-12-31', 'Penningmeester')
;
INSERT INTO BESTUURSLEDEN VALUES ( 6, '1993-1-1', NULL, 'Voorzitter')
;
INSERT INTO BESTUURSLEDEN VALUES ( 2, '1990-1-1', '1992-12-31', 'Voorzitter')
;
INSERT INTO BESTUURSLEDEN VALUES ( 2, '1994-1-1', NULL, 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES (112, '1992-1-1', '1992-12-31', 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES (112, '1994-1-1', NULL, 'Secretaris')
;
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1990-1-1', '1990-12-31', 'Penningmeester')
;
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1991-1-1', '1991-12-31', 'Secretaris')
;
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1993-1-1', '1993-12-31', 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1994-1-1', NULL, 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES ( 57, '1992-1-1', '1992-12-31', 'Secretaris')
;
INSERT INTO BESTUURSLEDEN VALUES ( 27, '1990-1-1', '1990-12-31', 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES ( 27, '1991-1-1', '1991-12-31', 'Penningmeester')
;
INSERT INTO BESTUURSLEDEN VALUES ( 27, '1993-1-1', '1993-12-31', 'Penningmeester')
;
INSERT INTO BESTUURSLEDEN VALUES ( 95, '1994-1-1', NULL, 'Penningmeester')
;

```

3.3 Herconstrueren van de complete tabellen van de voorbeelddatabase

Door sommige SQL-instructies in het boek verandert de inhoud plus de structuur van een of meer tabellen. Dit kan later problemen geven in het boek. Om die reden is het soms noodzakelijk elke tabel te verwijderen en geheel opnieuw op te bouwen. Gebruik hiervoor de volgende instructies:

```

DROP TABLE BESTUURSLEDEN
;
DROP TABLE BOETES
;
DROP TABLE WEDSTRIJDEN
;
DROP TABLE TEAMS
;
DROP TABLE SPELERS
;

CREATE TABLE SPELERS
(SPELERSNR INTEGER NOT NULL,
NAAM CHAR(15) NOT NULL,
VOORLETTERS CHAR(3) NOT NULL,
GEB_DATUM DATE ,
GESLACHT CHAR(1) NOT NULL,

```

```

        JAARTOE          SMALLINT    NOT NULL,
        STRAAT          VARCHAR(30) NOT NULL,
        HUISNR          CHAR(4)      ,
        POSTCODE        CHAR(6)      ,
        PLAATS          VARCHAR(30)  NOT NULL,
        TELEFOON        CHAR(13)     ,
        BONDSNR         CHAR(4)      ,
        PRIMARY KEY     (SPELERSNR)  )
;
CREATE TABLE TEAMS
(TEAMNR          INTEGER    NOT NULL,
SPELERSNR       INTEGER    NOT NULL,
DIVISIE         CHAR(6)    NOT NULL,
PRIMARY KEY     (TEAMNR)   )
;
CREATE TABLE WEDSTRIJDEN
(WEDSTRIJDNR    INTEGER    NOT NULL,
TEAMNR         INTEGER    NOT NULL,
SPELERSNR      INTEGER    NOT NULL,
GEWONNEN       SMALLINT   NOT NULL,
VERLOREN       SMALLINT   NOT NULL,
PRIMARY KEY     (WEDSTRIJDNR) )
;
CREATE TABLE BOETES
(BETALINGSNR    INTEGER    NOT NULL,
SPELERSNR      INTEGER    NOT NULL,
DATUM          DATE       NOT NULL,
BEDRAG         DECIMAL(7,2) NOT NULL,
PRIMARY KEY     (BETALINGSNR) )
;
CREATE TABLE BESTUURSLEDEN
(SPELERSNR      INTEGER    NOT NULL,
BEGIN_DATUM    DATE       NOT NULL,
EIND_DATUM     DATE       ,
FUNCTIE        CHAR(20)   ,
PRIMARY KEY     (SPELERSNR, BEGIN_DATUM))
;
INSERT INTO SPELERS VALUES (
    2, 'Elfring', 'R', '1948-09-01', 'M', 1975, 'Steden',
    '43', '3575NH', 'Den Haag', '070-237893', '2411')
;
INSERT INTO SPELERS VALUES (
    6, 'Permentier', 'R', '1964-06-25', 'M', 1977, 'Hazensteinln',
    '80', '1234KK', 'Den Haag', '070-476537', '8467')
;
INSERT INTO SPELERS VALUES (
    7, 'Wijers', 'GWS', '1963-05-11', 'M', 1981, 'Erasmusweg',
    '39', '9758VB', 'Den Haag', '070-347689', NULL)
;
INSERT INTO SPELERS VALUES (
    8, 'Niewenburg', 'B', '1962-07-08', 'V', 1980, 'Sporlaan',
    '4', '6584W0', 'Rijswijk', '070-458458', '2983')
;
INSERT INTO SPELERS VALUES (
    27, 'Cools', 'DD', '1964-12-28', 'V', 1983, 'Liespad',
    '804', '8457DK', 'Zoetermeer', '079-234857', '2513')
;
INSERT INTO SPELERS VALUES (
    28, 'Cools', 'C', '1963-06-22', 'V', 1983, 'Oudegracht',
    '10', '1294QK', 'Leiden', '010-659599', NULL)
;
INSERT INTO SPELERS VALUES (
    39, 'Bischoff', 'D', '1956-10-29', 'M', 1980, 'Ericaplein',
    '78', '9629CD', 'Den Haag', '070-393435', NULL)
;
INSERT INTO SPELERS VALUES (
    44, 'Bakker, de', 'E', '1963-01-09', 'M', 1980, 'Lawaaistraat',

```

```

    '23', '4444LJ', 'Rijswijk', '070-368753', '1124')
;
INSERT INTO SPELERS VALUES (
    57, 'Bohemen, van', 'M', '1971-08-17', 'M', 1985, 'Erasmusweg',
    '16', '4377CB', 'Den Haag', '070-473458', '6409')
;
INSERT INTO SPELERS VALUES (
    83, 'Hofland', 'PK', '1956-11-11', 'M', 1982, 'Mariakade',
    '16a', '1812UP', 'Den Haag', '070-353548', '1608')
;
INSERT INTO SPELERS VALUES (
    95, 'Meuleman', 'P', '1963-05-14', 'M', 1972, 'Hoofdweg',
    '33a', '57460P', 'Voorburg', '070-867564', NULL)
;
INSERT INTO SPELERS VALUES (
    100, 'Permentier', 'P', '1963-02-28', 'M', 1979, 'Hazensteinln',
    '80', '6494SG', 'Den Haag', '070-494593', '6524')
;
INSERT INTO SPELERS VALUES (
    104, 'Moerman', 'D', '1970-05-10', 'V', 1984, 'Stoutlaan',
    '65', '9437A0', 'Zoetermeer', '079-987571', '7060')
;
INSERT INTO SPELERS VALUES (
    112, 'Baaen, van', 'IP', '1963-10-01', 'V', 1984, 'Vosseweg',
    '8', '6392LK', 'Rotterdam', '010-548745', '1319')
;
INSERT INTO TEAMS VALUES (1, 6, 'ere')
;
INSERT INTO TEAMS VALUES (2, 27, 'tweede')
;
INSERT INTO WEDSTRIJDEN VALUES ( 1, 1, 6, 3, 1)
;
INSERT INTO WEDSTRIJDEN VALUES ( 2, 1, 6, 2, 3)
;
INSERT INTO WEDSTRIJDEN VALUES ( 3, 1, 6, 3, 0)
;
INSERT INTO WEDSTRIJDEN VALUES ( 4, 1, 44, 3, 2)
;
INSERT INTO WEDSTRIJDEN VALUES ( 5, 1, 83, 0, 3)
;
INSERT INTO WEDSTRIJDEN VALUES ( 6, 1, 2, 1, 3)
;
INSERT INTO WEDSTRIJDEN VALUES ( 7, 1, 57, 3, 0)
;
INSERT INTO WEDSTRIJDEN VALUES ( 8, 1, 8, 0, 3)
;
INSERT INTO WEDSTRIJDEN VALUES ( 9, 2, 27, 3, 2)
;
INSERT INTO WEDSTRIJDEN VALUES (10, 2, 104, 3, 2)
;
INSERT INTO WEDSTRIJDEN VALUES (11, 2, 112, 2, 3)
;
INSERT INTO WEDSTRIJDEN VALUES (12, 2, 112, 1, 3)
;
INSERT INTO WEDSTRIJDEN VALUES (13, 2, 8, 0, 3)
;
INSERT INTO BOETES VALUES (1, 6, '1980-12-08', 100)
;
INSERT INTO BOETES VALUES (2, 44, '1981-05-05', 75)
;
INSERT INTO BOETES VALUES (3, 27, '1983-09-10', 100)
;
INSERT INTO BOETES VALUES (4, 104, '1984-12-08', 50)
;
INSERT INTO BOETES VALUES (5, 44, '1980-12-08', 25)
;
INSERT INTO BOETES VALUES (6, 8, '1980-12-08', 25)

```

```
;  
INSERT INTO BOETES VALUES (7, 44, '1982-12-30', 30)  
;  
INSERT INTO BOETES VALUES (8, 27, '1984-11-12', 75)  
;  
INSERT INTO BESTUURSLEDEN VALUES ( 6, '1990-1-1', '1990-12-31', 'Secretaris')  
;  
INSERT INTO BESTUURSLEDEN VALUES ( 6, '1991-1-1', '1992-12-31', 'Lid')  
;  
INSERT INTO BESTUURSLEDEN VALUES ( 6, '1992-1-1', '1993-12-31', 'Penningmeester')  
;  
INSERT INTO BESTUURSLEDEN VALUES ( 6, '1993-1-1', NULL, 'Voorzitter')  
;  
INSERT INTO BESTUURSLEDEN VALUES ( 2, '1990-1-1', '1992-12-31', 'Voorzitter')  
;  
INSERT INTO BESTUURSLEDEN VALUES ( 2, '1994-1-1', NULL, 'Lid')  
;  
INSERT INTO BESTUURSLEDEN VALUES (112, '1992-1-1', '1992-12-31', 'Lid')  
;  
INSERT INTO BESTUURSLEDEN VALUES (112, '1994-1-1', NULL, 'Secretaris')  
;  
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1990-1-1', '1990-12-31', 'Penningmeester')  
;  
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1991-1-1', '1991-12-31', 'Secretaris')  
;  
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1993-1-1', '1993-12-31', 'Lid')  
;  
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1994-1-1', NULL, 'Lid')  
;  
INSERT INTO BESTUURSLEDEN VALUES ( 57, '1992-1-1', '1992-12-31', 'Secretaris')  
;  
INSERT INTO BESTUURSLEDEN VALUES ( 27, '1990-1-1', '1990-12-31', 'Lid')  
;  
INSERT INTO BESTUURSLEDEN VALUES ( 27, '1991-1-1', '1991-12-31', 'Penningmeester')  
;  
INSERT INTO BESTUURSLEDEN VALUES ( 27, '1993-1-1', '1993-12-31', 'Penningmeester')  
;  
INSERT INTO BESTUURSLEDEN VALUES ( 95, '1994-1-1', NULL, 'Penningmeester')  
;
```

SQL-instructies uit het boek

4.1 SQL-instructies voor hoofdstuk 4

Voorbeeld 4.1:

```
CREATE USER 'BOEKSQL'@'localhost' IDENTIFIED BY 'BOEKSQLPW'
```

Voorbeeld 4.2:

```
GRANT ALL PRIVILEGES  
ON *.*  
TO 'BOEKSQL'@'localhost'  
WITH GRANT OPTION
```

Voorbeeld 4.3:

```
CREATE DATABASE TENNIS
```

Voorbeeld 4.4:

```
USE TENNIS
```

Voorbeeld 4.5:

```
CREATE TABLE SPELERS (  
  SPELERSNR    INTEGER NOT NULL,  
  NAAM         CHAR(15) NOT NULL,  
  VOORLETTERS CHAR(3) NOT NULL,  
  GEB_DATUM   DATE,  
  GESLACHT    CHAR(1) NOT NULL,  
  JAARTOE     SMALLINT NOT NULL,  
  STRAAT      VARCHAR(30) NOT NULL,  
  HUISNR      CHAR(4),  
  POSTCODE    CHAR(6),  
  PLAATS      VARCHAR(30) NOT NULL,  
  TELEFOON    CHAR(13),  
  BONDSNR     CHAR(4),  
  PRIMARY KEY (SPELERSNR))  
;
```

```

CREATE TABLE TEAMS (
    TEAMNR          INTEGER NOT NULL,
    SPELERSNR       INTEGER NOT NULL,
    DIVISIE         CHAR(6) NOT NULL,
    PRIMARY KEY     (TEAMNR)
);
CREATE TABLE WEDSTRIJDEN (
    WEDSTRIJDNR    INTEGER NOT NULL,
    TEAMNR         INTEGER NOT NULL,
    SPELERSNR       INTEGER NOT NULL,
    GEWONNEN       SMALLINT NOT NULL,
    VERLOREN        SMALLINT NOT NULL,
    PRIMARY KEY     (WEDSTRIJDNR)
);
CREATE TABLE BOETES (
    BETALINGSNR    INTEGER NOT NULL,
    SPELERSNR       INTEGER NOT NULL,
    DATUM          DATE NOT NULL,
    BEDRAG         DECIMAL(7,2) NOT NULL,
    PRIMARY KEY     (BETALINGSNR)
);
CREATE TABLE BESTUURSLEDEN (
    SPELERSNR       INTEGER NOT NULL,
    BEGIN_DATUM    DATE NOT NULL,
    EIND_DATUM     DATE,
    FUNCTIE        CHAR(20),
    PRIMARY KEY     (SPELERSNR, BEGIN_DATUM)
);

```

Voorbeeld 4.6:

```

INSERT INTO SPELERS VALUES (
    2, 'Elfring', 'R', '1948-09-01', 'M', 1975, 'Steden',
    '43', '3575NH', 'Den Haag', '070-237893', '2411')
;
INSERT INTO SPELERS VALUES (
    6, 'Permentier', 'R', '1964-06-25', 'M', 1977, 'Hazensteinln',
    '80', '1234KK', 'Den Haag', '070-476537', '8467')
;
INSERT INTO SPELERS VALUES (
    7, 'Wijers', 'GWS', '1963-05-11', 'M', 1981, 'Erasmusweg',
    '39', '9758VB', 'Den Haag', '070-347689', NULL)
;
INSERT INTO SPELERS VALUES (
    8, 'Niewenburg', 'B', '1962-07-08', 'V', 1980, 'Sporlaan',
    '4', '6584W0', 'Rijswijk', '070-458458', '2983')
;
INSERT INTO SPELERS VALUES (
    27, 'Cools', 'DD', '1964-12-28', 'V', 1983, 'Liespad',
    '804', '8457DK', 'Zoetermeer', '079-234857', '2513')
;
INSERT INTO SPELERS VALUES (
    28, 'Cools', 'C', '1963-06-22', 'V', 1983, 'Oudegracht',
    '10', '1294QK', 'Leiden', '010-659599', NULL)
;
INSERT INTO SPELERS VALUES (
    39, 'Bischoff', 'D', '1956-10-29', 'M', 1980, 'Ericaplein',
    '78', '9629CD', 'Den Haag', '070-393435', NULL)
;
INSERT INTO SPELERS VALUES (
    44, 'Bakker, de', 'E', '1963-01-09', 'M', 1980, 'Lawaaistraat',
    '23', '4444LJ', 'Rijswijk', '070-368753', '1124')
;
INSERT INTO SPELERS VALUES (
    57, 'Bohemen, van', 'M', '1971-08-17', 'M', 1985, 'Erasmusweg',
    '16', '4377CB', 'Den Haag', '070-473458', '6409')
;
INSERT INTO SPELERS VALUES (

```

```

83, 'Hofland', 'PK', '1956-11-11', 'M', 1982, 'Mariakade',
  '16a', '1812UP', 'Den Haag', '070-353548', '1608')
;
INSERT INTO SPELERS VALUES (
  95, 'Meuleman', 'P', '1963-05-14', 'M', 1972, 'Hoofdweg',
  '33a', '57460P', 'Voorburg', '070-867564', NULL)
;
INSERT INTO SPELERS VALUES (
  100, 'Permentier', 'P', '1963-02-28', 'M', 1979, 'HazensteinIn',
  '80', '6494SG', 'Den Haag', '070-494593', '6524')
;
INSERT INTO SPELERS VALUES (
  104, 'Moerman', 'D', '1970-05-10', 'V', 1984, 'Stoutlaan',
  '65', '9437A0', 'Zoetermeer', '079-987571', '7060')
;
INSERT INTO SPELERS VALUES (
  112, 'Baalen, van', 'IP', '1963-10-01', 'V', 1984, 'Vosseweg',
  '8', '6392LK', 'Rotterdam', '010-548745', '1319')
;
INSERT INTO TEAMS VALUES (1, 6, 'ere')
;
INSERT INTO TEAMS VALUES (2, 27, 'tweede')
;
INSERT INTO WEDSTRIJDEN VALUES ( 1, 1, 6, 3, 1)
;
INSERT INTO WEDSTRIJDEN VALUES ( 2, 1, 6, 2, 3)
;
INSERT INTO WEDSTRIJDEN VALUES ( 3, 1, 6, 3, 0)
;
INSERT INTO WEDSTRIJDEN VALUES ( 4, 1, 44, 3, 2)
;
INSERT INTO WEDSTRIJDEN VALUES ( 5, 1, 83, 0, 3)
;
INSERT INTO WEDSTRIJDEN VALUES ( 6, 1, 2, 1, 3)
;
INSERT INTO WEDSTRIJDEN VALUES ( 7, 1, 57, 3, 0)
;
INSERT INTO WEDSTRIJDEN VALUES ( 8, 1, 8, 0, 3)
;
INSERT INTO WEDSTRIJDEN VALUES ( 9, 2, 27, 3, 2)
;
INSERT INTO WEDSTRIJDEN VALUES (10, 2, 104, 3, 2)
;
INSERT INTO WEDSTRIJDEN VALUES (11, 2, 112, 2, 3)
;
INSERT INTO WEDSTRIJDEN VALUES (12, 2, 112, 1, 3)
;
INSERT INTO WEDSTRIJDEN VALUES (13, 2, 8, 0, 3)
;
INSERT INTO BOETES VALUES (1, 6, '1980-12-08', 100)
;
INSERT INTO BOETES VALUES (2, 44, '1981-05-05', 75)
;
INSERT INTO BOETES VALUES (3, 27, '1983-09-10', 100)
;
INSERT INTO BOETES VALUES (4, 104, '1984-12-08', 50)
;
INSERT INTO BOETES VALUES (5, 44, '1980-12-08', 25)
;
INSERT INTO BOETES VALUES (6, 8, '1980-12-08', 25)
;
INSERT INTO BOETES VALUES (7, 44, '1982-12-30', 30)
;
INSERT INTO BOETES VALUES (8, 27, '1984-11-12', 75)
;
INSERT INTO BESTUURSLEDEN VALUES ( 6, '1990-1-1', '1990-12-31', 'Secretaris')
;

```

```

INSERT INTO BESTUURSLEDEN VALUES ( 6, '1991-1-1', '1992-12-31', 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES ( 6, '1992-1-1', '1993-12-31', 'Penningmeester')
;
INSERT INTO BESTUURSLEDEN VALUES ( 6, '1993-1-1', NULL, 'Voorzitter')
;
INSERT INTO BESTUURSLEDEN VALUES ( 2, '1990-1-1', '1992-12-31', 'Voorzitter')
;
INSERT INTO BESTUURSLEDEN VALUES ( 2, '1994-1-1', NULL, 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES (112, '1992-1-1', '1992-12-31', 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES (112, '1994-1-1', NULL, 'Secretaris')
;
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1990-1-1', '1990-12-31', 'Penningmeester')
;
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1991-1-1', '1991-12-31', 'Secretaris')
;
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1993-1-1', '1993-12-31', 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES ( 8, '1994-1-1', NULL, 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES ( 57, '1992-1-1', '1992-12-31', 'Secretaris')
;
INSERT INTO BESTUURSLEDEN VALUES ( 27, '1990-1-1', '1990-12-31', 'Lid')
;
INSERT INTO BESTUURSLEDEN VALUES ( 27, '1991-1-1', '1991-12-31', 'Penningmeester')
;
INSERT INTO BESTUURSLEDEN VALUES ( 27, '1993-1-1', '1993-12-31', 'Penningmeester')
;
INSERT INTO BESTUURSLEDEN VALUES ( 95, '1994-1-1', NULL, 'Penningmeester')

```

Voorbeeld 4.7:

```

SELECT SPELERSNR, NAAM, GEB_DATUM
FROM SPELERS
WHERE PLAATS = 'Den Haag'
ORDER BY NAAM

```

Voorbeeld 4.8:

```

SELECT SPELERSNR
FROM SPELERS
WHERE JAARTOE > 1980
AND PLAATS = 'Den Haag'
ORDER BY SPELERSNR

```

Voorbeeld 4.9:

```

SELECT *
FROM BOETES

```

Voorbeeld 4.10:

```

SELECT 33 * 121

```

Voorbeeld 4.11:

```

UPDATE BOETES
SET BEDRAG = 200
WHERE SPELERSNR = 44
;
SELECT SPELERSNR, BEDRAG
FROM BOETES
WHERE SPELERSNR = 44

```

Voorbeeld 4.12:


```
DELETE
FROM   BOETES
WHERE  BEDRAG > 100
;
SELECT *
FROM   BOETES
WHERE  BEDRAG = 25
```

Voorbeeld 4.13:

```
CREATE INDEX BOETES_BEDRAG
ON BOETES (BEDRAG)
```

Voorbeeld 4.14:

```
CREATE VIEW AANTAL_SETS (WEDSTRIJDNR, VERSCHIL) AS
SELECT WEDSTRIJDNR, ABS(GEWONNEN - VERLOREN)
FROM   WEDSTRIJDEN
```

Voorbeeld 4.15:

```
GRANT SELECT
ON     SPELERS
TO     DIANE
;
GRANT SELECT, UPDATE
ON     SPELERS
TO     PAUL
;
GRANT SELECT, UPDATE
ON     TEAMS
TO     PAUL
;
SELECT *
FROM   BOEKSQL.TEAMS
```

Voorbeeld 4.16:

```
DROP TABLE BESTUURSLEDEN
```

Voorbeeld 4.17:

```
DROP VIEW AANTAL_SETS
```

Voorbeeld 4.18:

```
DROP INDEX BOETES_BEDRAG
```

Voorbeeld 4.19:

```
DROP DATABASE TENNIS
```

Voorbeeld 4.20:

```
SELECT @@VERSION
```

Voorbeeld 4.21:

```
SET @@SQL_MODE = 'PIPES_AS_CONCAT'
```

Voorbeeld 4.22:

```
CREATE OR REPLACE VIEW USERS
(USER_NAME) AS
SELECT DISTINCT UPPER(CONCAT('',USER,' '@',HOST,''))
FROM   MYSQL.USER
```

```

;
CREATE OR REPLACE VIEW TABLES
(TABLE_CREATOR, TABLE_NAME,
CREATE_TIMESTAMP, COMMENT) AS
SELECT UPPER(TABLE_SCHEMA), UPPER(TABLE_NAME),
CREATE_TIME, TABLE_COMMENT
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE IN ('BASE TABLE','TEMPORARY')
;
CREATE OR REPLACE VIEW COLUMNS
(TABLE_CREATOR, TABLE_NAME, COLUMN_NAME,
COLUMN_NO, DATA_TYPE, CHAR_LENGTH,
'PRECISION', SCALE, NULLABLE, COMMENT) AS
SELECT UPPER(TABLE_SCHEMA), UPPER(TABLE_NAME),
UPPER(COLUMN_NAME), ORDINAL_POSITION,
UPPER(DATA_TYPE), CHARACTER_MAXIMUM_LENGTH,
NUMERIC_PRECISION, NUMERIC_SCALE, IS_NULLABLE,
COLUMN_COMMENT
FROM INFORMATION_SCHEMA.COLUMNS
;
CREATE OR REPLACE VIEW VIEWS
(VIEW_CREATOR, VIEW_NAME, CREATE_TIMESTAMP,
WITHCHECKOPT, IS_UPDATABLE, VIEWFORMULA, COMMENT) AS
SELECT UPPER(V.TABLE_SCHEMA), UPPER(V.TABLE_NAME),
T.CREATE_TIME,
CASE
WHEN V.CHECK_OPTION = 'None' THEN 'NO'
WHEN V.CHECK_OPTION = 'Cascaded' THEN 'CASCADED'
WHEN V.CHECK_OPTION = 'Local' THEN 'LOCAL'
ELSE 'Yes'
END,
V.IS_UPDATABLE, V.VIEW_DEFINITION, T.TABLE_COMMENT
FROM INFORMATION_SCHEMA.VIEWS AS V,
INFORMATION_SCHEMA.TABLES AS T
WHERE V.TABLE_NAME = T.TABLE_NAME
AND V.TABLE_SCHEMA = T.TABLE_SCHEMA
;
CREATE OR REPLACE VIEW INDEXES
(INDEX_CREATOR, INDEX_NAME, CREATE_TIMESTAMP,
TABLE_CREATOR, TABLE_NAME, UNIQUE_ID, INDEX_TYPE) AS
SELECT DISTINCT UPPER(I.INDEX_SCHEMA), UPPER(I.INDEX_NAME),
T.CREATE_TIME, UPPER(I.TABLE_SCHEMA),
UPPER(I.TABLE_NAME),
CASE
WHEN I.NON_UNIQUE = 0 THEN 'YES'
ELSE 'NO'
END,
I.INDEX_TYPE
FROM INFORMATION_SCHEMA.STATISTICS AS I,
INFORMATION_SCHEMA.TABLES AS T
WHERE I.TABLE_NAME = T.TABLE_NAME
AND I.TABLE_SCHEMA = T.TABLE_SCHEMA
;
CREATE OR REPLACE VIEW COLUMNS_IN_INDEX
(INDEX_CREATOR, INDEX_NAME,
TABLE_CREATOR, TABLE_NAME, COLUMN_NAME,
COLUMN_SEQ, ORDERING) AS
SELECT UPPER(INDEX_SCHEMA), UPPER(INDEX_NAME),
UPPER(TABLE_SCHEMA), UPPER(TABLE_NAME),
UPPER(COLUMN_NAME), SEQ_IN_INDEX,
CASE
WHEN COLLATION = 'A' THEN 'ASCENDING'
WHEN COLLATION = 'D' THEN 'DESCENDING'
ELSE 'OTHER'
END
FROM INFORMATION_SCHEMA.STATISTICS
;

```

```

CREATE OR REPLACE VIEW USER_AUTHS
(GRANTOR, GRANTEE, PRIVILEGE, WITHGRANTOPTION) AS
SELECT 'UNKNOWN', UPPER(GRANTEE), PRIVILEGE_TYPE, IS_GRANTABLE
FROM INFORMATION_SCHEMA.USER_PRIVILEGES
;
CREATE OR REPLACE VIEW DATABASE_AUTHS
(GRANTOR, GRANTEE, DATABASE_NAME, PRIVILEGE,
WITHGRANTOPTION) AS
SELECT 'UNKNOWN', UPPER(GRANTEE), UPPER(TABLE_SCHEMA),
PRIVILEGE_TYPE, IS_GRANTABLE
FROM INFORMATION_SCHEMA.SCHEMA_PRIVILEGES
;
CREATE OR REPLACE VIEW TABLE_AUTHS
(GRANTOR, GRANTEE, TABLE_CREATOR, TABLE_NAME,
PRIVILEGE, WITHGRANTOPTION) AS
SELECT 'UNKNOWN', UPPER(GRANTEE), UPPER(TABLE_SCHEMA),
UPPER(TABLE_NAME), PRIVILEGE_TYPE, IS_GRANTABLE
FROM INFORMATION_SCHEMA.TABLE_PRIVILEGES
;
CREATE OR REPLACE VIEW COLUMN_AUTHS
(GRANTOR, GRANTEE, TABLE_CREATOR, TABLE_NAME,
COLUMN_NAME, PRIVILEGE, WITHGRANTOPTION) AS
SELECT 'UNKNOWN', UPPER(GRANTEE), UPPER(TABLE_SCHEMA),
UPPER(TABLE_NAME), UPPER(COLUMN_NAME),
PRIVILEGE_TYPE, IS_GRANTABLE
FROM INFORMATION_SCHEMA.COLUMN_PRIVILEGES

```

Voorbeeld 4.23:

```

SELECT COLUMN_NAME, DATA_TYPE, COLUMN_NO
FROM COLUMNS
WHERE TABLE_NAME = 'SPELERS'
AND TABLE_CREATOR = 'TENNIS'
ORDER BY COLUMN_NO

```

Voorbeeld 4.24:

```

SELECT INDEX_NAME
FROM INDEXES
WHERE TABLE_NAME = 'BOETES'
AND TABLE_CREATOR = 'TENNIS'

```

4.2 SQL-instructies voor hoofdstuk 5

Voorbeeld 5.1:

```

SELECT TRUE, FALSE

```

Voorbeeld 5.2:

```

SELECT WEDSTRIJDNR, GEWONNEN - VERLOREN
FROM WEDSTRIJDEN
WHERE GEWONNEN = VERLOREN + 2

```

Voorbeeld 5.3:

```

SELECT TEAMNR, DIVISIE
FROM TEAMS

```

Voorbeeld 5.4:

```

SELECT TEAMNR AS TEAMNUMMER, DIVISIE AS DIVISIE_VAN_TEAM
FROM TEAMS

```

Voorbeeld 5.5:

```
SELECT  BETALINGSNR, BEDRAG * 100 AS CENTEN
FROM    BOETES
```

Voorbeeld 5.6:

```
SELECT  WEDSTRIJDNR AS PRIMSLEUTEL,
        80 AS TACHTIG,
        GEWONNEN - VERLOREN AS VERSCHIL,
        TIME('23:59:59') AS BIJNA_MIDDERNACHT,
        'TEKST' AS TEKST
FROM    WEDSTRIJDEN
WHERE   WEDSTRIJDNR <= 4
```

Voorbeeld 5.7:

```
SELECT  BETALINGSNR, BEDRAG * 100 AS CENTEN
FROM    BOETES
ORDER BY CENTEN
```

Voorbeeld 5.8:

```
SET @SPELERSNR = 7
```

Voorbeeld 5.9:

```
SELECT  NAAM, PLAATS, POSTCODE
FROM    SPELERS
WHERE   SPELERSNR < @SPELERSNR
```

Voorbeeld 5.10:

```
SELECT  @SPELERSNR
```

Voorbeeld 5.11:

```
SELECT  *
FROM    USER_AUTHS
WHERE   GRANTEE = CURRENT_USER
```

Voorbeeld 5.12:

```
SELECT  CURRENT_USER
```

Voorbeeld 5.13:

```
SELECT  *
FROM    BOETES
WHERE   DATUM = CURRENT_DATE
```

Voorbeeld 5.14:

```
SELECT  SPELERSNR,
        CASE GESLACHT
          WHEN 'V' THEN 'Vrouw'
          ELSE 'Man'
        END AS GESLACHT,
        NAAM
FROM    SPELERS
WHERE   JAARTOE > 1980
;
SELECT  SPELERSNR,
        CASE GESLACHT
          WHEN 'V' THEN 'Vrouw'
        END AS VROUWEN,
        NAAM
```

```
FROM SPELERS
WHERE JAARTOE > 1980
```

Voorbeeld 5.15:

```
SELECT SPELERSNR, PLAATS, GEB_DATUM,
CASE PLAATS
  WHEN 'Den Haag' THEN 0
  WHEN 'Rotterdam' THEN 1
  WHEN 'Rijswijk' THEN 2
  ELSE 3
END AS P,
CASE PLAATS
  WHEN 'Den Haag' THEN
CASE GEB_DATUM
  WHEN '1948-09-01' THEN 'Oude Hagenees'
  ELSE 'Jonge Hagenees'
END
  WHEN 'Rijswijk' THEN
CASE GEB_DATUM
  WHEN '1962-07-08' THEN 'Oude Rijswijker'
  ELSE 'Jonge Rijswijker'
END
  ELSE 'Rest'
END AS SOORT
FROM SPELERS
```

Voorbeeld 5.16:

```
SELECT SPELERSNR, JAARTOE,
CASE
  WHEN JAARTOE < 1980 THEN 'Ouderen'
  WHEN JAARTOE < 1983 THEN 'Jongeren'
  ELSE 'Kinderen'
END AS GROEP
FROM SPELERS
ORDER BY JAARTOE
```

Voorbeeld 5.17:

```
SELECT SPELERSNR, JAARTOE, PLAATS,
CASE
  WHEN JAARTOE >= 1980 AND JAARTOE <= 1982
  THEN 'Ouderen'
  WHEN PLAATS = 'Zoetermeer'
  THEN 'Zoetermeeders'
  WHEN SPELERSNR < 10
  THEN 'Eerste leden'
  ELSE 'Rest'
END
FROM SPELERS
```

Voorbeeld 5.18:

```
SELECT (SPELERSNR), (((NAAM)))
FROM SPELERS
```

Voorbeeld 5.19:

```
SELECT BETALINGSNR, YEAR(DATUM) AS JAARTAL
FROM BOETES
WHERE YEAR(DATUM) > 1980
```

Voorbeeld 5.20:

```
SELECT SPELERSNR, CONCAT(LEFT(VOORLETTERS, 1), '. ', NAAM) AS VOLLE_NAAM
FROM SPELERS
```

```
WHERE LEFT(NAAM, 1) = 'B'
```

Voorbeeld 5.21:

```
SELECT VOORLETTERS, NAAM, COALESCE(BONDSNR, '1')
FROM SPELERS
WHERE PLAATS = 'Den Haag'
```

Voorbeeld 5.22:

```
SELECT SPELERSNR, DAYNAME(GEB_DATUM) AS DAGNAAM,
        MONTHNAME(GEB_DATUM) AS MAANDNAAM,
        DAYOFYEAR(GEB_DATUM) AS DAGNUMMER
FROM SPELERS
WHERE SPELERSNR < 10
```

Voorbeeld 5.23:

```
SELECT SPELERSNR, GEB_DATUM,
        ADDDATE(GEB_DATUM, INTERVAL 7 DAY) AS GEB_DATUM_PLUS_7
FROM SPELERS
WHERE DAYNAME(GEB_DATUM) = 'Saturday'
```

Voorbeeld 5.24:

```
SELECT SPELERSNR, BEGIN_DATUM, EIND_DATUM,
        DATEDIFF(EIND_DATUM, BEGIN_DATUM) AS AANTALDAGEN
FROM BESTUURSLEDEN
WHERE DATEDIFF(EIND_DATUM, BEGIN_DATUM) > 500
OR (EIND_DATUM IS NULL AND DATEDIFF(CURRENT_DATE, BEGIN_DATUM) > 500)
ORDER BY SPELERSNR, BEGIN_DATUM
;
SELECT SPELERSNR, BEGIN_DATUM, EIND_DATUM,
        DATEDIFF(COALESCE(EIND_DATUM, CURRENT_DATE), BEGIN_DATUM)
FROM BESTUURSLEDEN
WHERE DATEDIFF(COALESCE(EIND_DATUM, CURRENT_DATE), BEGIN_DATUM) > 500
ORDER BY SPELERSNR
```

Voorbeeld 5.25:

```
SELECT BETALINGSNR
FROM BOETES
WHERE BEDRAG > 50
```

Voorbeeld 5.26:

```
SELECT CONCAT(RTRIM(NAAM), CAST(GEB_DATUM AS CHAR(10))) AS NAAM_PLUS_DATUM
FROM SPELERS
WHERE PLAATS = 'Rijswijk'
```

Voorbeeld 5.27:

```
UPDATE SPELERS
SET BONDSNR = NULL
WHERE SPELERSNR = 2
```

Voorbeeld 5.28:

```
SELECT TEAMNR, CAST(NULL AS CHAR)
FROM TEAMS
```

Voorbeeld 5.29:

```
SELECT WEDSTRIJDNR, GEWONNEN, VERLOREN
FROM WEDSTRIJDEN
WHERE GEWONNEN >= VERLOREN * 2
```

Voorbeeld 5.30:

```
SELECT SPELERSNR, PLAATS || ' ' || STRAAT || ' ' || HUISNR AS ADRES
FROM SPELERS
WHERE PLAATS = 'Den Haag'
```

Voorbeeld 5.31:

```
SELECT BETALINGSNR, DATUM, DATUM + INTERVAL 7 DAY AS DATUM_PLUS_7
FROM BOETES
WHERE BETALINGSNR > 5
```

Voorbeeld 5.32:

```
SELECT BETALINGSNR, DATUM
FROM BOETES
WHERE DATUM >= '1982-12-25'
AND DATUM <= '1982-12-25' + INTERVAL 6 DAY
```

Voorbeeld 5.35:

```
CREATE TABLE WEDSTRIJDEN_SPECIAAL (
    WEDSTRIJDNR INTEGER NOT NULL,
    TEAMNR      INTEGER NOT NULL,
    SPELERSNR   INTEGER NOT NULL,
    GEWONNEN   SMALLINT NOT NULL,
    VERLOREN   SMALLINT NOT NULL,
    START_DATUM DATE NOT NULL,
    START_TIJD  TIME NOT NULL,
    EIND_TIJD   TIME NOT NULL,
    PRIMARY KEY (WEDSTRIJDNR))
;
INSERT INTO WEDSTRIJDEN_SPECIAAL VALUES
    (1, 1, 6, 3, 1, '2004-10-25', '14:10:12', '16:50:09')
;
INSERT INTO WEDSTRIJDEN_SPECIAAL VALUES
    (2, 1, 44, 3, 2, '2004-10-25', '17:00:00', '17:55:48')
```

Voorbeeld 5.36:

```
SELECT WEDSTRIJDNR, START_TIJD,
    ADDTIME(START_TIJD, '08:00:00') AS START_TIJD_PLUS_8
FROM WEDSTRIJDEN_SPECIAAL
```

Voorbeeld 5.37:

```
SELECT WEDSTRIJDNR, EIND_TIJD
FROM WEDSTRIJDEN_SPECIAAL
WHERE ADDTIME(EIND_TIJD, '06:30:00') <= '24:00:00'
```

Voorbeeld 5.38:

```
CREATE TABLE TSTAMP (KOL TIMESTAMP)
;
SET @TIJD = TIMESTAMP('1980-12-08 23:59:59.59')
;
INSERT INTO TSTAMP VALUES (@TIJD + INTERVAL 3 MICROSECOND)
;
SELECT KOL, KOL + INTERVAL 3 MICROSECOND FROM TSTAMP
```

Voorbeeld 5.39:

```
INSERT INTO BESTUURSLEDEN
VALUES (2 + 4, CURRENT_DATE, CURRENT_DATE + INTERVAL 17 DAY, 'Lid')
```

Voorbeeld 5.40:

```
SELECT SPELERSNR
FROM SPELERS
WHERE (PLAATS, STRAAT) = ('Den Haag', 'Hazensteinln')
```

Voorbeeld 5.41:

```
UPDATE SPELERS
SET (PLAATS, STRAAT) = ('Den Haag', 'Hazensteinln')
WHERE SPELERSNR = 27
```

Voorbeeld 5.42:

```
INSERT INTO BOETES VALUES
(1, 6, '1980-12-08', 100),
(2, 44, '1981-05-05', 75),
(3, 27, '1983-09-10', 100),
(4, 104, '1984-12-08', 50),
(5, 44, '1980-12-08', 25),
(6, 8, '1980-12-08', 25),
(7, 44, '1982-12-30', 30),
(8, 27, '1984-11-12', 75)
```

4.3 SQL-instructies voor hoofdstuk 6

Voorbeeld 6.1:

```
SELECT SPELERSNR
FROM BOETES
WHERE BEDRAG > 25
GROUP BY SPELERSNR
HAVING COUNT(*) > 1
ORDER BY SPELERSNR
```

Voorbeeld 6.2:

```
SELECT SPELERSNR, BONDSNR
FROM SPELERS
WHERE PLAATS = 'Den Haag'
ORDER BY BONDSNR
```

Voorbeeld 6.3:

```
SELECT 89 * 73
```

Voorbeeld 6.4:

```
(SELECT *
FROM TEAMS)
;
((((SELECT *
FROM TEAMS))))
```

Voorbeeld 6.5:

```
SELECT SPELERSNR
FROM TEAMS
UNION
SELECT SPELERSNR
FROM BOETES
;
SELECT SPELERSNR
FROM TEAMS
ORDER BY SPELERSNR
UNION
```



```

SELECT SPELERSNR
FROM BOETES
;
SELECT SPELERSNR
FROM TEAMS
UNION
SELECT SPELERSNR
FROM BOETES
ORDER BY SPELERSNR
;
(SELECT SPELERSNR
FROM TEAMS
ORDER BY SPELERSNR)
UNION
(SELECT SPELERSNR
FROM BOETES)
ORDER BY SPELERSNR

```

Voorbeeld 6.6:

```

SELECT SPELERSNR
FROM (SELECT SPELERSNR, GESLACHT
FROM SPELERS
WHERE SPELERSNR < 10) AS TIJDELIJK
WHERE GESLACHT = 'M'

```

Voorbeeld 6.7:

```

SELECT SPELERSNR
FROM (SELECT SPELERSNR, GESLACHT
FROM (SELECT SPELERSNR, GESLACHT, JAARTOE
FROM (SELECT SPELERSNR, GESLACHT, JAARTOE
FROM SPELERS
WHERE SPELERSNR > 10) AS GROTER10
WHERE SPELERSNR < 100) AS KLEINER100
WHERE JAARTOE > 1980) AS JAARTOE1980
WHERE GESLACHT = 'M'

```

Voorbeeld 6.8:

```

SELECT SPELERSNR, JAARTOE -
      (SELECT JAARTOE
FROM SPELERS
WHERE SPELERSNR = 100)
FROM SPELERS
WHERE SPELERSNR < 60
;
SELECT SPELERSNR, JAARTOE - 1979
FROM SPELERS
WHERE SPELERSNR < 60
;
SELECT TEAMNR
FROM TEAMS
WHERE SPELERSNR =
      (SELECT SPELERSNR
FROM SPELERS)

```

Voorbeeld 6.9:

```

SELECT SPELERSNR
FROM SPELERS
WHERE YEAR(GEB_DATUM) =
      (SELECT YEAR(GEB_DATUM)
FROM SPELERS
WHERE SPELERSNR = 27)
;
SELECT SPELERSNR

```

```
FROM   SPELERS
WHERE  YEAR(GEB_DATUM) = 1964
```

Voorbeeld 6.10:

```
SELECT (SELECT GEB_DATUM
        FROM   SPELERS
        WHERE  SPELERSNR = 27) AS GB27,
       (SELECT GEB_DATUM
        FROM   SPELERS
        WHERE  SPELERSNR = 44) AS GB44,
       (SELECT GEB_DATUM
        FROM   SPELERS
        WHERE  SPELERSNR = 100) AS GB100
```

Voorbeeld 6.11:

```
SELECT SPELERSNR
FROM   SPELERS
WHERE  (GESLACHT, PLAATS) = (SELECT GESLACHT, PLAATS
                             FROM   SPELERS
                             WHERE  SPELERSNR = 100)
```

4.4 SQL-instructies voor hoofdstuk 7

Voorbeeld 7.1:

```
CREATE DATABASE EXTRA
;
USE EXTRA
;
CREATE TABLE WOONPLAATSEN (
    PLAATSNR      INTEGER NOT NULL PRIMARY KEY,
    PLAATSNAAM   CHAR(20) NOT NULL)
;
INSERT INTO WOONPLAATSEN VALUES
    (1, 'Den Haag')
;
INSERT INTO WOONPLAATSEN VALUES
    (2, 'Rijswijk')
```

Voorbeeld 7.2:

```
SELECT *
FROM   EXTRA.WOONPLAATSEN
```

Voorbeeld 7.3:

```
SELECT *
FROM   TENNIS.TEAMS
```

Voorbeeld 7.4:

```
SELECT *
FROM   BOB.BOETES
;
SELECT *
FROM   BOETES
```

Voorbeeld 7.5:

```
SELECT TEAMNR
FROM   TEAMS
;
```

```

SELECT  TEAMS.TEAMNR
FROM    TEAMS
;
SELECT  TENNIS.TEAMS.TEAMNR
FROM    TENNIS.TEAMS

```

Voorbeeld 7.6:

```

SELECT  TEAMNR, NAAM
FROM    TEAMS, SPELERS
WHERE   TEAMS.SPELERSNR = SPELERS.SPELERSNR

```

Voorbeeld 7.7:

```

SELECT  BETALINGSNR, BOETES.SPELERSNR, BEDRAG, NAAM, VOORLETTERS
FROM    BOETES, SPELERS
WHERE   BOETES.SPELERSNR = SPELERS.SPELERSNR
;
SELECT  SPELERS.SPELERSNR
FROM    SPELERS, TEAMS
WHERE   SPELERS.SPELERSNR = TEAMS.SPELERSNR
;
SELECT  SPELERS.SPELERSNR
FROM    TEAMS, SPELERS
WHERE   SPELERS.SPELERSNR = TEAMS.SPELERSNR

```

Voorbeeld 7.8:

```

SELECT  BETALINGSNR, BT.SPELERSNR, BEDRAG, NAAM, VOORLETTERS
FROM    BOETES AS BT, SPELERS AS S
WHERE   BT.SPELERSNR = S.SPELERSNR
;
SELECT  BETALINGSNR, BT.SPELERSNR, BEDRAG, NAAM, VOORLETTERS
FROM    BOETES BT, SPELERS S
WHERE   BT.SPELERSNR = S.SPELERSNR

```

Voorbeeld 7.9:

```

SELECT  T.SPELERSNR
FROM    TEAMS AS T, BOETES AS BT
WHERE   T.SPELERSNR = BT.SPELERSNR

```

Voorbeeld 7.10:

```

SELECT  DISTINCT T.SPELERSNR
FROM    TEAMS AS T, BOETES AS BT
WHERE   T.SPELERSNR = BT.SPELERSNR

```

Voorbeeld 7.11:

```

SELECT  DISTINCT S.NAAM, S.VOORLETTERS
FROM    SPELERS AS S, WEDSTRIJDEN AS W
WHERE   S.SPELERSNR = W.SPELERSNR

```

Voorbeeld 7.12:

```

SELECT  W.WEDSTRIJDNR, W.SPELERSNR, W.TEAMNR, S.NAAM, T.DIVISIE
FROM    WEDSTRIJDEN AS W, SPELERS AS S, TEAMS AS T
WHERE   W.SPELERSNR = S.SPELERSNR
AND     W.TEAMNR = T.TEAMNR

```

Voorbeeld 7.13:

```

SELECT  B.BETALINGSNR, B.SPELERSNR, B.DATUM
FROM    BOETES AS B, SPELERS AS S
WHERE   B.SPELERSNR = S.SPELERSNR
AND     YEAR(B.DATUM) = S.JAARTOE

```

Voorbeeld 7.14:

```
SELECT S.SPELERSNR
FROM SPELERS AS S, SPELERS AS P
WHERE P.NAAM = 'Permentier'
AND P.VOORLETTERS = 'R'
AND S.GEB_DATUM < P.GEB_DATUM
;
SELECT S.SPELERSNR
FROM SPELERS AS S, SPELERS
WHERE SPELERS.NAAM = 'Permentier'
AND SPELERS.VOORLETTERS = 'R'
AND S.GEB_DATUM < SPELERS.GEB_DATUM
```

Voorbeeld 7.15:

```
SELECT SPELERS.SPELERSNR, NAAM, BEDRAG
FROM SPELERS, BOETES
WHERE SPELERS.SPELERSNR = BOETES.SPELERSNR
AND GEB_DATUM > '1920-06-30'
;
SELECT SPELERS.SPELERSNR, NAAM, BEDRAG
FROM SPELERS INNER JOIN BOETES
ON SPELERS.SPELERSNR = BOETES.SPELERSNR
WHERE GEB_DATUM > '1920-06-30'
```

Voorbeeld 7.16:

```
SELECT TEAMNR, NAAM
FROM TEAMS, SPELERS
WHERE TEAMS.SPELERSNR = SPELERS.SPELERSNR
;
SELECT TEAMNR, NAAM
FROM TEAMS INNER JOIN SPELERS
ON TEAMS.SPELERSNR = SPELERS.SPELERSNR
;
SELECT TEAMNR, NAAM
FROM TEAMS JOIN SPELERS
ON TEAMS.SPELERSNR = SPELERS.SPELERSNR
```

Voorbeeld 7.17:

```
SELECT SPELERS.SPELERSNR, NAAM, BEDRAG
FROM SPELERS, BOETES
WHERE SPELERS.SPELERSNR = BOETES.SPELERSNR
ORDER BY SPELERS.SPELERSNR
;
SELECT SPELERS.SPELERSNR, NAAM, BEDRAG
FROM SPELERS LEFT OUTER JOIN BOETES
ON SPELERS.SPELERSNR = BOETES.SPELERSNR
ORDER BY SPELERS.SPELERSNR
```

Voorbeeld 7.18:

```
SELECT BETALINGSNR, NAAM
FROM BOETES LEFT OUTER JOIN SPELERS
ON BOETES.SPELERSNR = SPELERS.SPELERSNR
ORDER BY BETALINGSNR
```

Voorbeeld 7.19:

```
SELECT S.SPELERSNR, NAAM, TEAMNR, DIVISIE
FROM SPELERS AS S LEFT OUTER JOIN TEAMS AS T
ON S.SPELERSNR = T.SPELERSNR
ORDER BY S.SPELERSNR
```

Voorbeeld 7.20:

```
SELECT SPELERS.SPELERSNR, NAAM, BEDRAG, TEAMNR
FROM   SPELERS LEFT OUTER JOIN BOETES
       ON SPELERS.SPELERSNR = BOETES.SPELERSNR
       LEFT OUTER JOIN WEDSTRIJDEN
       ON SPELERS.SPELERSNR = WEDSTRIJDEN.SPELERSNR
WHERE  PLAATS = 'Rijswijk'
```

Voorbeeld 7.21:

```
SELECT SPELERS.SPELERSNR
FROM   SPELERS INNER JOIN BOETES
       ON SPELERS.SPELERSNR = BOETES.SPELERSNR
       LEFT OUTER JOIN TEAMS
       ON SPELERS.SPELERSNR = TEAMS.SPELERSNR
;
SELECT SPELERS.SPELERSNR
FROM   (SPELERS INNER JOIN BOETES
       ON SPELERS.SPELERSNR = BOETES.SPELERSNR)
       LEFT OUTER JOIN TEAMS
       ON SPELERS.SPELERSNR = TEAMS.SPELERSNR
```

Voorbeeld 7.22:

```
SELECT SPELERS.SPELERSNR, NAAM, TEAMNR
FROM   TEAMS RIGHT OUTER JOIN SPELERS
       ON TEAMS.SPELERSNR = SPELERS.SPELERSNR
```

Voorbeeld 7.23:

```
SELECT DISTINCT WEDSTRIJDEN.WEDSTRIJDNR,
               WEDSTRIJDEN.SPELERSNR AS WED_SNR,
               BESTUURSLEDEN.SPELERSNR AS BESTUUR_SNR
FROM   WEDSTRIJDEN FULL OUTER JOIN BESTUURSLEDEN
       ON WEDSTRIJDEN.SPELERSNR = BESTUURSLEDEN.SPELERSNR
ORDER BY WEDSTRIJDEN.WEDSTRIJDNR,
         WEDSTRIJDEN.SPELERSNR,
         BESTUURSLEDEN.SPELERSNR
```

Voorbeeld 7.24:

```
SELECT TEAMS.SPELERSNR, TEAMS.TEAMNR, BOETES.BETALINGSNR
FROM   TEAMS LEFT OUTER JOIN BOETES
       ON TEAMS.SPELERSNR = BOETES.SPELERSNR
WHERE  DIVISIE = 'tweede'
;
SELECT TEAMS.SPELERSNR, TEAMS.TEAMNR, BOETES.BETALINGSNR
FROM   TEAMS LEFT OUTER JOIN BOETES
       ON TEAMS.SPELERSNR = BOETES.SPELERSNR
       AND DIVISIE = 'tweede'
```

Voorbeeld 7.25:

```
SELECT TEAMS.SPELERSNR, TEAMS.TEAMNR, BOETES.BETALINGSNR
FROM   TEAMS FULL OUTER JOIN BOETES
       ON TEAMS.SPELERSNR = BOETES.SPELERSNR
       AND TEAMS.SPELERSNR > 1000
```

Voorbeeld 7.26:

```
SELECT *
FROM   TEAMS UNION JOIN BOETES
;
SELECT T.SPELERSNR, T.TEAMNR, T.DIVISIE,
       B.BETALINGSNR, B.DATUM, B.BEDRAG
FROM   TEAMS AS T INNER JOIN BOETES AS B
```

```

        ON T.SPELERSNR = B.SPELERSNR
WHERE   DIVISIE = 'ere'
;
SELECT *
FROM   TEAMS NATURAL INNER JOIN BOETES
WHERE  DIVISIE = 'ere'
;
SELECT *
FROM   SPELERS, TEAMS
WHERE  SPELERS.SPELERSNR = TEAMS.SPELERSNR
;
SELECT *
FROM   SPELERS LEFT OUTER JOIN TEAMS
        ON SPELERS.SPELERSNR = TEAMS.SPELERSNR
;
SELECT *
FROM   SPELERS, TEAMS
WHERE  SPELERS.SPELERSNR > TEAMS.SPELERSNR
;
SELECT *
FROM   SPELERS, TEAMS
WHERE  SPELERS.SPELERSNR ? TEAMS.SPELERSNR

```

Voorbeeld 7.27:

```

SELECT *
FROM   BOETES LEFT OUTER JOIN TEAMS
        USING (SPELERSNR)

```

Voorbeeld 7.28:

```

SELECT  SPELERSNR
FROM    (SELECT *
        FROM    SPELERS
        WHERE   PLAATS = 'Den Haag') AS HAGENEZEN

```

Voorbeeld 7.29:

```

SELECT  SMALLE_TEAMS.SPELERSNR
FROM    (SELECT  SPELERSNR, DIVISIE
        FROM    TEAMS) AS SMALLE_TEAMS
WHERE   SMALLE_TEAMS.DIVISIE = 'ere'

```

Voorbeeld 7.30:

```

SELECT  WEDSTRIJDNR, VERSCHIL
FROM    (SELECT  WEDSTRIJDNR,
        ABS(GEWONNEN - VERLOREN) AS VERSCHIL
        FROM    WEDSTRIJDEN) AS W
WHERE   VERSCHIL > 2

```

Voorbeeld 7.31:

```

SELECT  *
FROM    (SELECT  'Den Haag' AS PLAATS, 4 AS AANTAL
        UNION
        SELECT  'Rotterdam', 6
        UNION
        SELECT  'Rijswijk', 1
        UNION
        SELECT  'Voorburg', 2) AS PLAATSEN
ORDER  BY  PLAATS

```

Voorbeeld 7.32:

```

SELECT  SPELERSNR, NAAM, SPELERS.PLAATS, AANTAL * 1000
FROM    SPELERS,

```

```

        (SELECT 'Den Haag' AS PLAATS, 4 AS AANTAL
        UNION
        SELECT 'Rotterdam', 6
        UNION
        SELECT 'Rijswijk', 1
        UNION
        SELECT 'Voorburg', 2) AS PLAATSEN
WHERE   SPELERS.PLAATS = PLAATSEN.PLAATS
ORDER BY SPELERSNR
;

```

```

SELECT   SPELERSNR, NAAM, SPELERS.PLAATS, AANTAL * 1000
FROM     SPELERS LEFT OUTER JOIN
        (SELECT 'Den Haag' AS PLAATS, 4 AS AANTAL
        UNION
        SELECT 'Rotterdam', 6
        UNION
        SELECT 'Rijswijk', 1
        UNION
        SELECT 'Voorburg', 2) AS PLAATSEN
        ON SPELERS.PLAATS = PLAATSEN.PLAATS
ORDER BY SPELERSNR

```

Voorbeeld 7.33:

```

SELECT   SPELERSNR
FROM     SPELERS LEFT OUTER JOIN
        (SELECT 'Den Haag' AS PLAATS, 4 AS AANTAL
        UNION
        SELECT 'Rotterdam', 6
        UNION
        SELECT 'Rijswijk', 1
        UNION
        SELECT 'Voorburg', 2) AS PLAATSEN
        ON SPELERS.PLAATS = PLAATSEN.PLAATS
WHERE    PLAATSEN.AANTAL > 2

```

Voorbeeld 7.34:

```

SELECT   *
FROM     (SELECT 'John' AS VOORNAAM
        UNION
        SELECT 'Mark'
        UNION
        SELECT 'Arnold') AS VOORNAMEN,
        (SELECT 'Berg' AS ACHTERNAAM
        UNION
        SELECT 'Johnson'
        UNION
        SELECT 'Willem') AS ACHTERNAMEN

```

Voorbeeld 7.35:

```

SELECT   GETAL, POWER(GETAL,3) AS DERDEMACHT
FROM     (SELECT 10 AS GETAL UNION SELECT 11 UNION SELECT 12
        UNION
        SELECT 13 UNION SELECT 14 UNION SELECT 15
        UNION
        SELECT 16 UNION SELECT 17 UNION SELECT 18
        UNION
        SELECT 19) AS GETALLEN
WHERE    POWER(GETAL,3) <= 4000

```

Voorbeeld 7.36:

```

SELECT   GETAL
FROM     (SELECT (CIJFER1.CIJFER * 100) + (CIJFER2.CIJFER * 10) +

```

```

        CIJFER3.CIJFER AS GETAL
FROM    (SELECT 0 AS CIJFER UNION SELECT 1 UNION
        SELECT 2 UNION SELECT 3 UNION
        SELECT 4 UNION SELECT 5 UNION
        SELECT 6 UNION SELECT 7 UNION
        SELECT 8 UNION SELECT 9) AS CIJFER1,
        (SELECT 0 AS CIJFER UNION SELECT 1 UNION
        SELECT 2 UNION SELECT 3 UNION
        SELECT 4 UNION SELECT 5 UNION
        SELECT 6 UNION SELECT 7 UNION
        SELECT 8 UNION SELECT 9) AS CIJFER2,
        (SELECT 0 AS CIJFER UNION SELECT 1 UNION
        SELECT 2 UNION SELECT 3 UNION
        SELECT 4 UNION SELECT 5 UNION
        SELECT 6 UNION SELECT 7 UNION
        SELECT 8 UNION SELECT 9) AS CIJFER3)
        AS GETALLEN

ORDER BY GETAL

```

Voorbeeld 7.37:

```

SELECT  GETAL AS KWADRAAT, ROUND(SQRT(GETAL)) AS BASIS
FROM    (SELECT (CIJFER1.CIJFER * 100) + (CIJFER2.CIJFER * 10) +
        CIJFER3.CIJFER AS GETAL
        FROM    (SELECT 0 AS CIJFER UNION SELECT 1 UNION
        SELECT 2 UNION SELECT 3 UNION
        SELECT 4 UNION SELECT 5 UNION
        SELECT 6 UNION SELECT 7 UNION
        SELECT 8 UNION SELECT 9) AS CIJFER1,
        (SELECT 0 AS CIJFER UNION SELECT 1 UNION
        SELECT 2 UNION SELECT 3 UNION
        SELECT 4 UNION SELECT 5 UNION
        SELECT 6 UNION SELECT 7 UNION
        SELECT 8 UNION SELECT 9) AS CIJFER2,
        (SELECT 0 AS CIJFER UNION SELECT 1 UNION
        SELECT 2 UNION SELECT 3 UNION
        SELECT 4 UNION SELECT 5 UNION
        SELECT 6 UNION SELECT 7 UNION
        SELECT 8 UNION SELECT 9) AS CIJFER3)
        AS GETALLEN
WHERE   SQRT(GETAL) = ROUND(SQRT(GETAL))
ORDER BY GETAL

```

4.5 SQL-instructies voor hoofdstuk 8

Voorbeeld 8.1:

```

SELECT  SPELERSNR
FROM    SPELERS
WHERE   PLAATS = 'Den Haag'

```

Voorbeeld 8.2:

```

SELECT  SPELERSNR, GEB_DATUM, JAARTOE
FROM    SPELERS
WHERE   YEAR(GEB_DATUM) + 17 = JAARTOE

```

Voorbeeld 8.3:

```

SELECT  SPELERSNR
FROM    SPELERS
WHERE   BONDSNR = '7060'

```

Voorbeeld 8.4:


```

SELECT  SPELERSNR, BONDSNR
FROM    SPELERS
WHERE   BONDSNR = BONDSNR

```

Voorbeeld 8.5:

```

SELECT  WEDSTRIJDNR
FROM    WEDSTRIJDEN
WHERE   (GEWONNEN, VERLOREN) = (2, 3)

```

Voorbeeld 8.6:

```

SELECT  SPELERSNR, NAAM
FROM    SPELERS
WHERE   SPELERSNR =
        (SELECT  SPELERSNR
         FROM    TEAMS
         WHERE   TEAMNR = 1)
;
SELECT  SPELERSNR, NAAM
FROM    SPELERS
WHERE   SPELERSNR = 6
;
SELECT  *
FROM    SPELERS
WHERE   GEB_DATUM <
        (SELECT  GEB_DATUM
         FROM    SPELERS)

```

Voorbeeld 8.7:

```

SELECT  SPELERSNR, NAAM, VOORLETTERS
FROM    SPELERS
WHERE   GEB_DATUM <
        (SELECT  GEB_DATUM
         FROM    SPELERS
         WHERE   BONDSNR = '8467')
;
SELECT  SPELERSNR, NAAM, VOORLETTERS
FROM    SPELERS
WHERE   GEB_DATUM <
        (SELECT  GEB_DATUM
         FROM    SPELERS
         WHERE   BONDSNR = '9999')

```

Voorbeeld 8.8:

```

SELECT  WEDSTRIJDNR
FROM    WEDSTRIJDEN
WHERE   TEAMNR =
        (SELECT  TEAMNR
         FROM    TEAMS
         WHERE   SPELERSNR = 27)

```

Voorbeeld 8.9:

```

SELECT  SPELERSNR, PLAATS, GESLACHT
FROM    SPELERS
WHERE   (PLAATS, GESLACHT) =
        ((SELECT  PLAATS
         FROM    SPELERS
         WHERE   SPELERSNR = 7),
         (SELECT  GESLACHT
         FROM    SPELERS
         WHERE   SPELERSNR = 2))

```

Voorbeeld 8.10:

```
SELECT DISTINCT SPELERSNR
FROM BESTUURSLEDEN
WHERE (BEGIN_DATUM, EIND_DATUM) =
      (SELECT BEGIN_DATUM, EIND_DATUM
       FROM BESTUURSLEDEN
       WHERE SPELERSNR = 6
       AND FUNCTIE = 'Secretaris'
       AND BEGIN_DATUM = '1990-01-01')
```

Voorbeeld 8.11:

```
SELECT SPELERSNR, NAAM, VOORLETTERS
FROM SPELERS
WHERE (NAAM, VOORLETTERS) <
      (SELECT NAAM, VOORLETTERS
       FROM SPELERS
       WHERE SPELERSNR = 6)
ORDER BY NAAM, VOORLETTERS
```

Voorbeeld 8.12:

```
SELECT WEDSTRIJDNR
FROM WEDSTRIJDEN_SPECIAAL
WHERE (START_DATUM, START_TIJD) >
      (SELECT START_DATUM, START_TIJD
       FROM WEDSTRIJDEN_SPECIAAL
       WHERE WEDSTRIJDNR = 1)
```

Voorbeeld 8.13:

```
SELECT WEDSTRIJDNR
FROM WEDSTRIJDEN
WHERE 'Rijswijk' =
      (SELECT PLAATS
       FROM SPELERS
       WHERE SPELERS.SPELERSNR = WEDSTRIJDEN.SPELERSNR)
;
SELECT PLAATS
FROM SPELERS
WHERE SPELERS.SPELERSNR = 6
;
SELECT PLAATS
FROM SPELERS
WHERE SPELERS.SPELERSNR = 44
```

Voorbeeld 8.14:

```
SELECT WEDSTRIJDNR, SPELERSNR, TEAMNR
FROM WEDSTRIJDEN
WHERE SPELERSNR =
      (SELECT SPELERSNR
       FROM TEAMS
       WHERE TEAMS.SPELERSNR = WEDSTRIJDEN.SPELERSNR)
```

Voorbeeld 8.15:

```
SELECT WEDSTRIJDNR
FROM WEDSTRIJDEN
WHERE SUBSTR((SELECT DIVISIE
              FROM TEAMS
              WHERE TEAMS.TEAMNR =
                    WEDSTRIJDEN.TEAMNR),3,1)
      =
      SUBSTR((SELECT NAAM
              FROM SPELERS
```

```
WHERE SPELERS.SPELERSNR =
      WEDSTRIJDEN.SPELERSNR),3,1)
```

Voorbeeld 8.16:

```
SELECT SPELERSNR, NAAM, GESLACHT, GEB_DATUM
FROM   SPELERS
WHERE  GESLACHT = 'M'
AND    GEB_DATUM > '1970-12-31'
```

Voorbeeld 8.17:

```
SELECT SPELERSNR, NAAM, PLAATS
FROM   SPELERS
WHERE  PLAATS = 'Rotterdam'
OR     PLAATS = 'Zoetermeer'
```

Voorbeeld 8.18:

```
SELECT SPELERSNR, NAAM, PLAATS
FROM   SPELERS
WHERE  PLAATS <> 'Den Haag'
;
SELECT SPELERSNR, NAAM, PLAATS
FROM   SPELERS
WHERE  NOT (PLAATS = 'Den Haag')
```

Voorbeeld 8.19:

```
SELECT SPELERSNR, PLAATS, GEB_DATUM
FROM   SPELERS
WHERE  (PLAATS = 'Den Haag' OR YEAR(GEB_DATUM) = 1963)
AND NOT (PLAATS = 'Den Haag' AND YEAR(GEB_DATUM) = 1963)
```

Voorbeeld 8.20:

```
SELECT SPELERSNR, NAAM, PLAATS
FROM   SPELERS
WHERE  PLAATS = 'Rijswijk'
OR     PLAATS = 'Rotterdam'
OR     PLAATS = 'Leiden'
OR     PLAATS = 'Voorburg'
;
SELECT SPELERSNR, NAAM, PLAATS
FROM   SPELERS
WHERE  PLAATS IN ('Rijswijk', 'Rotterdam', 'Leiden', 'Voorburg')
```

Voorbeeld 8.21:

```
SELECT SPELERSNR, YEAR(GEB_DATUM) AS GEBOORTEJAAR
FROM   SPELERS
WHERE  YEAR(GEB_DATUM) IN (1962, 1963, 1970)
```

Voorbeeld 8.22:

```
SELECT WEDSTRIJDNR, GEWONNEN, VERLOREN
FROM   WEDSTRIJDEN
WHERE  2 IN (GEWONNEN, VERLOREN)
```

Voorbeeld 8.23:

```
SELECT SPELERSNR
FROM   SPELERS
WHERE  SPELERSNR IN
      (100,
      (SELECT SPELERSNR
       FROM   BOETES
```

```

WHERE    BETALINGSNR = 1),
(SELECT  SPELERSNR
FROM     TEAMS
WHERE    TEAMNR = 2))

```

Voorbeeld 8.24:

```

SELECT  WEDSTRIJDNR, GEWONNEN, VERLOREN
FROM    WEDSTRIJDEN
WHERE   GEWONNEN IN
        (TRUNCATE(WEDSTRIJDNR / 2,0), VERLOREN,
        (SELECT  VERLOREN
FROM        WEDSTRIJDEN
WHERE      WEDSTRIJDNR = 1))

```

Voorbeeld 8.25:

```

SELECT  WEDSTRIJDNR
FROM    WEDSTRIJDEN
WHERE   (SELECT  SUBSTR(NAAM,1,1)
FROM     SPELERS
WHERE    SPELERS.SPELERSNR = WEDSTRIJDEN.SPELERSNR)
        IN ('B','C','E')

```

Voorbeeld 8.26:

```

SELECT  WEDSTRIJDNR, GEWONNEN, VERLOREN
FROM    WEDSTRIJDEN
WHERE   (GEWONNEN, VERLOREN) IN ((3,1),(3,2))

```

Voorbeeld 8.27:

```

SELECT  SPELERSNR, NAAM, VOORLETTERS
FROM    SPELERS
WHERE   (NAAM, VOORLETTERS) IN
        ((SELECT  NAAM, VOORLETTERS
FROM        SPELERS
WHERE      SPELERSNR = 6),
        (SELECT  NAAM, VOORLETTERS
FROM        SPELERS
WHERE      SPELERSNR = 27))

```

Voorbeeld 8.28:

```

SELECT  SPELERSNR
FROM    WEDSTRIJDEN
;
SELECT  SPELERSNR, NAAM, VOORLETTERS
FROM    SPELERS
WHERE   SPELERSNR IN (6, 6, 6, 44, 83, 2, 57, 8, 27, 104, 112, 112, 8)
;
SELECT  SPELERSNR, NAAM, VOORLETTERS
FROM    SPELERS
WHERE   SPELERSNR IN
        (SELECT  SPELERSNR
FROM        WEDSTRIJDEN)
;
SELECT  SPELERSNR, NAAM, VOORLETTERS
FROM    SPELERS
WHERE   SPELERSNR IN (6, 6, 6, 44, 83, 2, 57, 8, 27, 104, 112, 112, 8)

```

Voorbeeld 8.29:

```

SELECT  SPELERSNR, NAAM
FROM    SPELERS
WHERE   SPELERSNR IN
        (SELECT  SPELERSNR

```

```

FROM WEDSTRIJDEN
WHERE TEAMNR = 1)

```

Voorbeeld 8.30:

```

SELECT SPELERSNR, NAAM
FROM SPELERS
WHERE SPELERSNR IN
  (SELECT SPELERSNR
   FROM WEDSTRIJDEN
   WHERE TEAMNR NOT IN
     (SELECT TEAMNR
      FROM TEAMS
      WHERE SPELERSNR = 6))

```

Voorbeeld 8.31:

```

SELECT *
FROM BESTUURSLEDEN
WHERE SPELERSNR = 95
AND EIND_DATUM IN
  (SELECT EIND_DATUM
   FROM BESTUURSLEDEN
   WHERE SPELERSNR = 27)

```

Voorbeeld 8.32:

```

SELECT *
FROM BESTUURSLEDEN
WHERE SPELERSNR = 57
AND EIND_DATUM IN
  (SELECT EIND_DATUM
   FROM BESTUURSLEDEN
   WHERE SPELERSNR = 7)

```

Voorbeeld 8.33:

```

SELECT *
FROM BESTUURSLEDEN
WHERE SPELERSNR = 95
AND EIND_DATUM NOT IN
  (SELECT EIND_DATUM
   FROM BESTUURSLEDEN
   WHERE SPELERSNR = 7)

```

Voorbeeld 8.34:

```

SELECT *
FROM BESTUURSLEDEN
WHERE (BEGIN_DATUM, EIND_DATUM) IN
  (SELECT BEGIN_DATUM, EIND_DATUM
   FROM BESTUURSLEDEN
   WHERE FUNCTIE = 'Secretaris')

```

Voorbeeld 8.35:

```

CREATE TABLE SPELERS_NV (
  NAAM          CHAR(10) NOT NULL,
  VOORLETTERS   CHAR(3)  NOT NULL,
  PLAATS        VARCHAR(30) NOT NULL,
  PRIMARY KEY   (NAAM, VOORLETTERS))
;
INSERT INTO SPELERS_NV VALUES ('Permentier', 'R', 'Den Haag')
;
INSERT INTO SPELERS_NV VALUES ('Permentier', 'P', 'Den Haag')
;
INSERT INTO SPELERS_NV VALUES ('Meuleman', 'P', 'Voorburg')

```

```

;
CREATE TABLE BOETES_NV (
    BETALINGSNR    INTEGER NOT NULL,
    NAAM           CHAR(10) NOT NULL,
    VOORLETTERS    CHAR(3) NOT NULL,
    BEDRAG         DECIMAL(7,2) NOT NULL,
    PRIMARY KEY    (BETALINGSNR),
    FOREIGN KEY    (NAAM, VOORLETTERS)
                  REFERENCES SPELERS_NV (NAAM, VOORLETTERS))
;
INSERT INTO BOETES_NV VALUES (1, 'Permentier', 'R', 100.00)
;
INSERT INTO BOETES_NV VALUES (2, 'Meuleman', 'P', 200.00)

```

Voorbeeld 8.36:

```

SELECT  NAAM, VOORLETTERS, PLAATS
FROM    SPELERS_NV
WHERE   NAAM IN
        (SELECT  NAAM
         FROM    BOETES_NV)
AND     VOORLETTERS IN
        (SELECT  VOORLETTERS
         FROM    BOETES_NV)
;
SELECT  NAAM, VOORLETTERS, PLAATS
FROM    SPELERS_NV
WHERE   (NAAM, VOORLETTERS) IN
        (SELECT  NAAM, VOORLETTERS
         FROM    BOETES_NV)
;
SELECT  NAAM, VOORLETTERS, PLAATS
FROM    SPELERS_NV
WHERE   NAAM IN
        (SELECT  NAAM
         FROM    BOETES_NV
         WHERE   SPELERS_NV.VOORLETTERS = BOETES_NV.VOORLETTERS)

```

Voorbeeld 8.37:

```

SELECT  NAAM, VOORLETTERS, PLAATS
FROM    SPELERS_NV
WHERE   (NAAM, VOORLETTERS) NOT IN
        (SELECT  NAAM, VOORLETTERS
         FROM    BOETES_NV)

```

Voorbeeld 8.38:

```

SELECT  SPELERSNR, GEB_DATUM
FROM    SPELERS
WHERE   GEB_DATUM >= '1962-01-01'
AND     GEB_DATUM <= '1964-12-31'
;
SELECT  SPELERSNR, GEB_DATUM
FROM    SPELERS
WHERE   GEB_DATUM BETWEEN '1962-01-01' AND '1964-12-31'

```

Voorbeeld 8.39:

```

SELECT  WEDSTRIJDNR, GEWONNEN + VERLOREN
FROM    WEDSTRIJDEN
WHERE   GEWONNEN + VERLOREN BETWEEN 2 AND 4

```

Voorbeeld 8.40:

```

SELECT  SPELERSNR, GEB_DATUM, NAAM, VOORLETTERS
FROM    SPELERS

```

```

WHERE     GEB_DATUM BETWEEN
          (SELECT  GEB_DATUM
           FROM    SPELERS
           WHERE   NAAM = 'Niewenburg'
           AND     VOORLETTERS = 'B')
          AND
          (SELECT  GEB_DATUM
           FROM    SPELERS
           WHERE   NAAM = 'Meuleman'
           AND     VOORLETTERS = 'P')

```

Voorbeeld 8.41:

```

SELECT  NAAM, SPELERSNR
FROM    SPELERS
WHERE   NAAM LIKE 'B%'

```

Voorbeeld 8.42:

```

SELECT  NAAM, SPELERSNR
FROM    SPELERS
WHERE   NAAM LIKE '%n'

```

Voorbeeld 8.43:

```

SELECT  NAAM, SPELERSNR
FROM    SPELERS
WHERE   NAAM LIKE '%a_'

```

Voorbeeld 8.44:

```

SELECT  NAAM, PLAATS, SPELERSNR
FROM    SPELERS
WHERE   NAAM LIKE CONCAT('%', SUBSTR(PLAATS,3,1))

```

Voorbeeld 8.45:

```

SELECT  NAAM, SPELERSNR
FROM    SPELERS
WHERE   NAAM LIKE '%#_%' ESCAPE '#'

```

Voorbeeld 8.46:

```

SELECT  SPELERSNR, NAAM
FROM    SPELERS
WHERE   NAAM REGEXP 'b'

```

Voorbeeld 8.47:

```

SELECT  NAAM, SPELERSNR
FROM    SPELERS
WHERE   NAAM REGEXP '^ba'

```

Voorbeeld 8.48:

```

SELECT  NAAM, PLAATS, SPELERSNR
FROM    SPELERS
WHERE   NAAM REGEXP CONCAT(SUBSTR(PLAATS,1,1), '$')

```

Voorbeeld 8.49:

```

SELECT  NAAM, SPELERSNR
FROM    SPELERS
WHERE   NAAM REGEXP '[abc]'

```

Voorbeeld 8.50:

```
SELECT NAAM, SPELERSNR
FROM SPELERS
WHERE NAAM REGEXP 'm.n'
```

Voorbeeld 8.51:

```
SELECT NAAM, SPELERSNR
FROM SPELERS
WHERE NAAM REGEXP '[men][men]'
```

Voorbeeld 8.52:

```
SELECT SPELERSNR, POSTCODE
FROM SPELERS
WHERE POSTCODE REGEXP '^[0-9][0-9]3'
```

Voorbeeld 8.53:

```
SELECT NAAM, SPELERSNR
FROM SPELERS
WHERE NAAM REGEXP '^bo.*van$'
```

Voorbeeld 8.54:

```
SELECT SPELERSNR, POSTCODE
FROM SPELERS
WHERE POSTCODE REGEXP '[0-9][0-9]*[a-z][a-z]*'
```

Voorbeeld 8.55:

```
SELECT NAAM, SPELERSNR
FROM SPELERS
WHERE NAAM REGEXP '^[^A-M]'
```

Voorbeeld 8.56:

```
SELECT SPELERSNR, NAAM
FROM SPELERS
WHERE NAAM REGEXP '^[a-z]{8}'
```

Voorbeeld 8.57:

```
SELECT SPELERSNR, NAAM
FROM SPELERS
WHERE NAAM REGEXP '^[a-z]{5,7}$'
```

Voorbeeld 8.58:

```
SELECT SPELERSNR, POSTCODE
FROM SPELERS
WHERE POSTCODE REGEXP '4{4}'
```

Voorbeeld 8.59:

```
SELECT SPELERSNR, NAAM
FROM SPELERS
WHERE NAAM REGEXP 'man|van'
```

Voorbeeld 8.60:

```
SELECT SPELERSNR, NAAM
FROM SPELERS
WHERE NAAM REGEXP '[[.space.]]'
```

Voorbeeld 8.61:

```
SELECT SPELERSNR, NAAM
```



```
FROM SPELERS
WHERE NAAM REGEXP '[:<:]]van[:>:]'
```

Voorbeeld 8.62:

```
CREATE TABLE BOEKEN (
  BOEKNR          INTEGER NOT NULL PRIMARY KEY,
  AUTEURS         TEXT NOT NULL,
  TITEL          TEXT NOT NULL,
  JAAR UITGIFTE  YEAR NOT NULL,
  SAMENVATTING   TEXT NOT NULL)
ENGINE = MyISAM
```

Voorbeeld 8.63:

```
SET @@SQL_MODE = 'PIPES_AS_CONCAT'
;
INSERT INTO BOEKEN VALUES (1,
  'Ramez Elmasri and Shamkant B. Navathe',
  'Fundamentals of Database Systems', 2007,
  'This market-leading text serves as a valued resource for '||
  'those who will interact with databases in future courses '||
  'and careers. Renowned for its accessible, comprehensive '||
  'coverage of models and real systems, it provides an '||
  'up-to-date introduction to modern database technologies.')
;
INSERT INTO BOEKEN VALUES (2,
  'George Coulouris, Jean Dollimore and Tim Kindberg',
  'Distributed Systems: Concepts and Design', 2005,
  'This book provides broad and up-to-date coverage of the '||
  'principles and practice in the fast moving area of '||
  'distributed systems. It includes the key issues in the '||
  'debate between components and web services as the way '||
  'forward for industry. The depth of coverage will enable '||
  'students to evaluate existing distributed systems and '||
  'design new ones.')
;
INSERT INTO BOEKEN VALUES (3,
  'Rick van der Lans',
  'Introduction to SQL: Mastering the Relational Database '||
  'Language', 2007,
  'This book provides a technical introduction to the '||
  'features of SQL. Aimed at those new to SQL, but not new '||
  'to programming, it gives the reader the essential skills '||
  'required to start programming with this language.')
;
INSERT INTO BOEKEN VALUES (4,
  'Chris Date',
  'An Introduction to Database Systems', 2004,
  'Continuing in the eighth edition, this book provides a '||
  'comprehensive introduction to the now very large field of '||
  'database systems by providing a solid grounding in the '||
  'foundations of database technology. This new edition has '||
  'been rewritten and expanded to stay current with database '||
  'system trends.')
;
INSERT INTO BOEKEN VALUES (5,
  'Thomas M. Connolly and Carolyn E. Begg',
  'DataBase Systems: A Practical Approach to Design, '||
  'Implementation and Management', 2005,
  'A clear introduction to design implementation and management '||
  'issues, as well as an extensive treatment of database '||
  'languages and standards, make this book an indispensable '||
  'complete reference for database students and professionals.')
```

Voorbeeld 8.64:

```
CREATE FULLTEXT INDEX INDEX_TITEL
  ON BOEKEN (TITEL)
;
CREATE FULLTEXT INDEX INDEX_SAMENVATTING
  ON BOEKEN (SAMENVATTING)
```

Voorbeeld 8.65:

```
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL) AGAINST ('design')
;
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL) AGAINST ('design' IN NATURAL LANGUAGE MODE)
```

Voorbeeld 8.66:

```
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL) AGAINST ('to')
```

Voorbeeld 8.67:

```
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL) AGAINST ('database')
```

Voorbeeld 8.68:

```
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL) AGAINST ('practical')
```

Voorbeeld 8.69:

```
SELECT  BOEKNR, MATCH(SAMENVATTING) AGAINST ('distributed')
FROM    BOEKEN
```

Voorbeeld 8.70:

```
SELECT  BOEKNR, MATCH(TITEL) AGAINST ('introduction')
FROM    BOEKEN
WHERE   MATCH(TITEL) AGAINST ('introduction')
```

Voorbeeld 8.71:

```
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL) AGAINST ('practical distributed')
```

Voorbeeld 8.72:

```
CREATE FULLTEXT INDEX INDEX_TITEL_SAMENVATTING
  ON BOEKEN (TITEL, SAMENVATTING)
```

Voorbeeld 8.73:

```
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL, SAMENVATTING) AGAINST ('careers')
```

Voorbeeld 8.74:

```
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL) AGAINST ('database' IN BOOLEAN MODE)
```

Voorbeeld 8.75:

```
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL, SAMENVATTING) AGAINST ('introduction' IN BOOLEAN MODE)
```

Voorbeeld 8.76:

```
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL) AGAINST ('database design' IN BOOLEAN MODE)
```

Voorbeeld 8.77:

```
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL) AGAINST ('+database +design' IN BOOLEAN MODE)
```

Voorbeeld 8.78:

```
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL) AGAINST ('+database -design' IN BOOLEAN MODE)
```

Voorbeeld 8.79:

```
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL)
        AGAINST ("design implementation" IN BOOLEAN MODE)
```

Voorbeeld 8.80:

```
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL) AGAINST ('data*' IN BOOLEAN MODE)
```

Voorbeeld 8.81:

```
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL) AGAINST ('practical'
                              IN NATURAL LANGUAGE MODE WITH QUERY EXPANSION)
;
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL) AGAINST ('practical')
;
SELECT  BOEKNR, TITEL
FROM    BOEKEN
WHERE   MATCH(TITEL) AGAINST ('DataBase Systems: A Practical
                              Approach to Design, Implementation and Management')
```

Voorbeeld 8.82:

```
SELECT  SPELERSNR, BONDSNR
FROM    SPELERS
WHERE   BONDSNR IS NOT NULL
```

Voorbeeld 8.83:

```
SELECT  NAAM, SPELERSNR, BONDSNR
FROM    SPELERS
WHERE   BONDSNR <> '8467'
OR      BONDSNR IS NULL
```

Voorbeeld 8.84:

```

SELECT  NAAM, VOORLETTERS
FROM    SPELERS
WHERE   SPELERSNR IN
        (SELECT  SPELERSNR
         FROM    BOETES)
;
SELECT  NAAM, VOORLETTERS
FROM    SPELERS
WHERE   EXISTS
        (SELECT  *
         FROM    BOETES
         WHERE   SPELERSNR = SPELERS.SPELERSNR)
;
SELECT  *
FROM    BOETES
WHERE   SPELERSNR = 6

```

Voorbeeld 8.85:

```

SELECT  NAAM, VOORLETTERS
FROM    SPELERS
WHERE   NOT EXISTS
        (SELECT  *
         FROM    TEAMS
         WHERE   SPELERSNR = SPELERS.SPELERSNR)
;
SELECT  NAAM, VOORLETTERS
FROM    SPELERS
WHERE   NOT EXISTS
        (SELECT  'niets'
         FROM    TEAMS
         WHERE   SPELERSNR = SPELERS.SPELERSNR)

```

Voorbeeld 8.86:

```

SELECT  SPELERSNR, NAAM, GEB_DATUM
FROM    SPELERS
WHERE   GEB_DATUM <= ALL
        (SELECT  GEB_DATUM
         FROM    SPELERS)

```

Voorbeeld 8.87:

```

SELECT  SPELERSNR, GEB_DATUM
FROM    SPELERS
WHERE   GEB_DATUM < ALL
        (SELECT  GEB_DATUM
         FROM    SPELERS AS S INNER JOIN WEDSTRIJDEN AS W
         ON S.SPELERSNR = W.SPELERSNR
         WHERE   W.TEAMNR = 2)

```

Voorbeeld 8.88:

```

SELECT  DISTINCT TEAMNR, SPELERSNR
FROM    WEDSTRIJDEN AS W1
WHERE   GEWONNEN <= ALL
        (SELECT  GEWONNEN
         FROM    WEDSTRIJDEN AS W2
         WHERE   W1.TEAMNR = W2.TEAMNR)

```

Voorbeeld 8.89:

```

SELECT  BONDSNR, SPELERSNR
FROM    SPELERS
WHERE   BONDSNR >= ALL

```

```

        (SELECT  BONDSNR
         FROM    SPELERS)
;
SELECT  BONDSNR, SPELERSNR
FROM    SPELERS
WHERE   BONDSNR >= ALL
        (SELECT  BONDSNR
         FROM    SPELERS
         WHERE   BONDSNR IS NOT NULL)

```

Voorbeeld 8.90:

```

SELECT  SPELERSNR, PLAATS, BONDSNR
FROM    SPELERS AS S1
WHERE   BONDSNR <= ALL
        (SELECT  S2.BONDSNR
         FROM    SPELERS AS S2
         WHERE   S1.PLAATS = S2.PLAATS)
;
SELECT  SPELERSNR, PLAATS, BONDSNR
FROM    SPELERS AS S1
WHERE   BONDSNR <= ALL
        (SELECT  S2.BONDSNR
         FROM    SPELERS AS S2
         WHERE   S1.PLAATS = S2.PLAATS
         AND     BONDSNR IS NOT NULL)
;
SELECT  SPELERSNR, PLAATS, BONDSNR
FROM    SPELERS AS S1
WHERE   BONDSNR <= ALL
        (SELECT  S2.BONDSNR
         FROM    SPELERS AS S2
         WHERE   S1.PLAATS = S2.PLAATS
         AND     BONDSNR IS NOT NULL)
AND     PLAATS IN
        (SELECT  PLAATS
         FROM    SPELERS
         WHERE   BONDSNR IS NOT NULL)

```

Voorbeeld 8.91:

```

SELECT  SPELERSNR, NAAM, GEB_DATUM
FROM    SPELERS
WHERE   GEB_DATUM > ANY
        (SELECT  GEB_DATUM
         FROM    SPELERS)

```

Voorbeeld 8.92:

```

SELECT  DISTINCT SPELERSNR
FROM    BOETES
WHERE   SPELERSNR <> 27
AND     BEDRAG > ANY
        (SELECT  BEDRAG
         FROM    BOETES
         WHERE   SPELERSNR = 27)

```

Voorbeeld 8.93:

```

SELECT  SPELERSNR, GEB_DATUM, PLAATS
FROM    SPELERS AS S1
WHERE   GEB_DATUM > ANY
        (SELECT  GEB_DATUM
         FROM    SPELERS AS S2
         WHERE   S1.PLAATS = S2.PLAATS)

```

Voorbeeld 8.94:

```

SELECT  SPELERSNR, NAAM
FROM    SPELERS
WHERE   EXISTS
        (SELECT  *
         FROM    BOETES
         WHERE   SPELERS.SPELERSNR = SPELERSNR)
;
SELECT  SPELERSNR, NAAM
FROM    SPELERS
WHERE   EXISTS
        (SELECT  *
         FROM    BOETES
         WHERE   SPELERS.SPELERSNR = SPELERS.SPELERSNR)
;
SELECT  SPELERSNR, NAAM
FROM    SPELERS AS S
WHERE   EXISTS
        (SELECT  *
         FROM    BOETES AS BT
         WHERE   S.SPELERSNR = BT.SPELERSNR)

```

Voorbeeld 8.95:

```

SELECT  TEAMNR, DIVISIE
FROM    TEAMS
WHERE   EXISTS
        (SELECT  *
         FROM    WEDSTRIJDEN
         WHERE   SPELERSNR = 44
         AND    TEAMNR = TEAMS.TEAMNR)

```

Voorbeeld 8.96:

```

SELECT  DISTINCT SPELERSNR
FROM    BOETES AS BT
WHERE   SPELERSNR IN
        (SELECT  SPELERSNR
         FROM    BOETES
         WHERE   BETALINGSNR <> BT.BETALINGSNR)

```

Voorbeeld 8.97:

```

SELECT  SPELERSNR, NAAM
FROM    SPELERS
WHERE   1 <> ALL
        (SELECT  TEAMNR
         FROM    WEDSTRIJDEN
         WHERE   SPELERSNR = SPELERS.SPELERSNR)

```

Voorbeeld 8.98:

```

SELECT  TEAMNR
FROM    TEAMS
WHERE   NOT EXISTS
        (SELECT  *
         FROM    WEDSTRIJDEN
         WHERE   SPELERSNR = 57
         AND    TEAMNR = TEAMS.TEAMNR)

```

Voorbeeld 8.99:

```

SELECT  SPELERSNR
FROM    SPELERS AS S
WHERE   NOT EXISTS
        (SELECT  *
         FROM    TEAMS AS T

```

```

WHERE NOT EXISTS
      (SELECT *
        FROM WEDSTRIJDEN AS W
        WHERE T.TEAMNR = W.TEAMNR
        AND S.SPELERSNR = W.SPELERSNR))
;
SELECT *
FROM TEAMS AS T
WHERE NOT EXISTS
      (SELECT *
        FROM WEDSTRIJDEN AS W
        WHERE T.TEAMNR = W.TEAMNR
        AND W.SPELERSNR = 27)

```

Voorbeeld 8.100:

```

SELECT SPELERSNR
FROM SPELERS
WHERE NOT EXISTS
      (SELECT *
        FROM WEDSTRIJDEN AS W1
        WHERE SPELERSNR = 57
        AND NOT EXISTS
              (SELECT *
                FROM WEDSTRIJDEN AS W2
                WHERE W1.TEAMNR = W2.TEAMNR
                AND SPELERS.SPELERSNR =
                  W2.SPELERSNR))

```

Voorbeeld 8.101:

```

SELECT SPELERSNR
FROM WEDSTRIJDEN
WHERE TEAMNR IN
      (SELECT TEAMNR
        FROM TEAMS
        WHERE TEAMNR NOT IN
              (SELECT TEAMNR
                FROM WEDSTRIJDEN
                WHERE SPELERSNR = 57))
;
SELECT SPELERSNR
FROM SPELERS AS S
WHERE NOT EXISTS
      (SELECT *
        FROM WEDSTRIJDEN AS W1
        WHERE SPELERSNR = 57
        AND NOT EXISTS
              (SELECT *
                FROM WEDSTRIJDEN AS W2
                WHERE W1.TEAMNR = W2.TEAMNR
                AND S.SPELERSNR = W2.SPELERSNR))
AND SPELERSNR NOT IN
      (SELECT SPELERSNR
        FROM WEDSTRIJDEN
        WHERE TEAMNR IN
              (SELECT TEAMNR
                FROM TEAMS
                WHERE TEAMNR NOT IN
                      (SELECT TEAMNR
                        FROM WEDSTRIJDEN
                        WHERE SPELERSNR = 57))))

```

Voorbeeld 8.102:

```

SELECT SPELERSNR
FROM SPELERS

```

```

WHERE    PLAATS = 'Den Haag'
;
SELECT  SPELERSNR
FROM    SPELERS
WHERE   NOT (PLAATS = 'Den Haag')
;
SELECT  SPELERSNR
FROM    SPELERS
WHERE   PLAATS <> 'Den Haag'

```

Voorbeeld 8.103:

```

SELECT  SPELERSNR
FROM    BOETES
WHERE   BEDRAG = 25
;
SELECT  SPELERSNR
FROM    BOETES
WHERE   BEDRAG <> 25
;
SELECT  SPELERSNR
FROM    SPELERS
WHERE   SPELERSNR NOT IN
      (SELECT  SPELERSNR
       FROM    BOETES
       WHERE   BEDRAG = 25)
;
SELECT  SPELERSNR
FROM    SPELERS
WHERE   SPELERSNR IN
      (SELECT  SPELERSNR
       FROM    BOETES
       WHERE   BEDRAG = 25)

```

4.6 SQL-instructies voor hoofdstuk 9

Voorbeeld 9.1:

```

SELECT  *
FROM    BOETES
;
SELECT  BETALINGSNR, SPELERSNR, DATUM, BEDRAG
FROM    BOETES
;
SELECT  BOETES.*
FROM    BOETES

```

Voorbeeld 9.2:

```

SELECT  BOETES.*
FROM    BOETES INNER JOIN TEAMS
      ON BOETES.SPELERSNR = TEAMS.SPELERSNR
;
SELECT  BOETES.BETALINGSNR, BOETES.SPELERSNR, BOETES.DATUM, BOETES.BEDRAG
FROM    BOETES INNER JOIN TEAMS
      ON BOETES.SPELERSNR = TEAMS.SPELERSNR
;
SELECT  BT.*
FROM    BOETES AS BT INNER JOIN TEAMS
      ON BT.SPELERSNR = TEAMS.SPELERSNR

```

Voorbeeld 9.3:

```

SELECT  WEDSTRIJDNR, 'Sa1do', GEWONNEN - VERLOREN, GEWONNEN * 10

```



```
FROM WEDSTRIJDEN
```

Voorbeeld 9.4:

```
SELECT PLAATS
FROM SPELERS
;
SELECT DISTINCT PLAATS
FROM SPELERS
```

Voorbeeld 9.5:

```
SELECT STRAAT, PLAATS
FROM SPELERS
;
SELECT DISTINCT STRAAT, PLAATS
FROM SPELERS
;
SELECT PLAATS
FROM SPELERS
;
SELECT ALL PLAATS
FROM SPELERS
```

Voorbeeld 9.6:

```
SELECT DISTINCT BONDSNR
FROM SPELERS
```

Voorbeeld 9.7:

```
SELECT DISTINCT *
FROM (SELECT 1 AS A, 'Hello' AS B, 4 AS C UNION
      SELECT 1, 'Hello', NULL UNION
      SELECT 1, 'Hello', NULL UNION
      SELECT 1, NULL, NULL) AS X
```

Voorbeeld 9.8:

```
SELECT COUNT(*)
FROM SPELERS
```

Voorbeeld 9.9:

```
SELECT COUNT(*)
FROM SPELERS
WHERE PLAATS = 'Den Haag'
;
SELECT COUNT(*), SPELERSNR
FROM SPELERS
;
SELECT 'Het aantal spelers is', COUNT(*)
FROM SPELERS
```

Voorbeeld 9.10:

```
SELECT COUNT(BONDSNR)
FROM SPELERS
;
SELECT COUNT(ALL BONDSNR)
FROM SPELERS
```

Voorbeeld 9.11:

```
SELECT COUNT(DISTINCT PLAATS)
FROM SPELERS
```

Voorbeeld 9.12:

```
SELECT COUNT(DISTINCT SUBSTR(NAAM, 1, 1)) AS LETTERS
FROM SPELERS
```

Voorbeeld 9.13:

```
SELECT COUNT(DISTINCT YEAR(DATUM)) AS JAREN
FROM BOETES
```

Voorbeeld 9.14:

```
SELECT COUNT(DISTINCT PLAATS), COUNT(DISTINCT GESLACHT)
FROM SPELERS
```

Voorbeeld 9.15:

```
SELECT SPELERSNR, NAAM
FROM SPELERS AS S
WHERE (SELECT COUNT(*)
       FROM BOETES AS BT
       WHERE S.SPELERSNR = BT.SPELERSNR)
      >
      (SELECT COUNT(*)
       FROM WEDSTRIJDEN AS W
       WHERE S.SPELERSNR = W.SPELERSNR)
```

Voorbeeld 9.16:

```
SELECT SPELERSNR, NAAM,
       (SELECT COUNT(*)
        FROM BOETES
        WHERE BOETES.SPELERSNR = SPELERS.SPELERSNR)
       AS AANTAL
FROM SPELERS
WHERE (SELECT COUNT(*)
       FROM BOETES
       WHERE BOETES.SPELERSNR = SPELERS.SPELERSNR) >= 2
;
SELECT SPELERSNR, NAAM, AANTAL
FROM (SELECT SPELERSNR, NAAM,
            (SELECT COUNT(*)
             FROM BOETES
             WHERE BOETES.SPELERSNR = SPELERS.SPELERSNR)
            AS AANTAL
      FROM SPELERS) AS SA
WHERE AANTAL >= 2
```

Voorbeeld 9.17:

```
SELECT (SELECT COUNT(*)
       FROM BOETES) AS AANTAL_BOETES,
       (SELECT COUNT(*)
       FROM WEDSTRIJDEN) AS AANTAL_WEDSTRIJDEN
```

Voorbeeld 9.18:

```
SELECT MAX(BEDRAG)
FROM BOETES
```

Voorbeeld 9.19:

```
SELECT MIN(BEDRAG)
FROM BOETES
WHERE SPELERSNR IN
      (SELECT SPELERSNR
       FROM SPELERS)
```

```
WHERE PLAATS = 'Den Haag')
```

Voorbeeld 9.20:

```
SELECT COUNT(*)
FROM BOETES
WHERE BEDRAG =
      (SELECT MIN(BEDRAG)
       FROM BOETES)
```

Voorbeeld 9.21:

```
SELECT DISTINCT TEAMNR, SPELERSNR
FROM WEDSTRIJDEN AS W1
WHERE GEWONNEN =
      (SELECT MAX(GEWONNEN)
       FROM WEDSTRIJDEN AS W2
       WHERE W1.TEAMNR = W2.TEAMNR)
```

Voorbeeld 9.22:

```
SELECT (MAX(BEDRAG) - MIN(BEDRAG)) * 100
FROM BOETES
```

Voorbeeld 9.23:

```
SELECT SUBSTR(MAX(NAAM), 1, 1)
FROM SPELERS
```

Voorbeeld 9.24:

```
SELECT MAX(BONDSNR)
FROM SPELERS
WHERE PLAATS = 'Leiden'
```

Voorbeeld 9.25:

```
SELECT CASE WHEN MIN(BONDSNR) IS NULL
           THEN 'Onbekend'
           ELSE MIN(BONDSNR)
        END
FROM SPELERS
WHERE PLAATS = 'Amsterdam'
```

Voorbeeld 9.26:

```
SELECT SPELERSNR, BEDRAG, DATUM
FROM BOETES AS BT1
WHERE BEDRAG =
      (SELECT MAX(BEDRAG)
       FROM BOETES AS BT2
       WHERE BT2.SPELERSNR = BT1.SPELERSNR)
```

Voorbeeld 9.27:

```
SELECT SPELERSNR,
      (SELECT MAX(BEDRAG)
       FROM BOETES
       WHERE BOETES.SPELERSNR = SPELERS.SPELERSNR)
      AS HOOGSTEBEETE,
      (SELECT MAX(GEWONNEN)
       FROM WEDSTRIJDEN
       WHERE WEDSTRIJDEN.SPELERSNR = SPELERS.SPELERSNR)
      AS AANTALSETS
FROM SPELERS
```

Voorbeeld 9.28:

```

SELECT  SPELERSNR
FROM    SPELERS
WHERE   (SELECT  MIN(BEDRAG)
        FROM    BOETES
        WHERE   BOETES.SPELERSNR = SPELERS.SPELERSNR) =
        (SELECT  MAX(BEDRAG)
        FROM    BOETES
        WHERE   BOETES.SPELERSNR = SPELERS.SPELERSNR)

```

Voorbeeld 9.29:

```

SELECT  SUM(BEDRAG)
FROM    BOETES
WHERE   SPELERSNR IN
        (SELECT  SPELERSNR
         FROM    SPELERS
         WHERE   PLAATS = 'Rijswijk')
;
SELECT  SUM(DISTINCT BEDRAG)
FROM    BOETES
WHERE   SPELERSNR IN
        (SELECT  SPELERSNR
         FROM    SPELERS
         WHERE   PLAATS = 'Rijswijk')

```

Voorbeeld 9.30:

```

SELECT  AVG(BEDRAG)
FROM    BOETES
WHERE   SPELERSNR = 44

```

Voorbeeld 9.31:

```

SELECT  DISTINCT SPELERSNR
FROM    BOETES
WHERE   BEDRAG >
        (SELECT  AVG(BEDRAG)
         FROM    BOETES)

```

Voorbeeld 9.32:

```

SELECT  AVG(DISTINCT BEDRAG)
FROM    BOETES

```

Voorbeeld 9.33:

```

SELECT  AVG(LENGTH(RTRIM(NAAM))), MAX(LENGTH(RTRIM(NAAM)))
FROM    SPELERS

```

Voorbeeld 9.34:

```

SELECT  BETALINGSNR, BEDRAG,
        ABS(BEDRAG - (SELECT  AVG(BEDRAG)
                     FROM    BOETES)) AS VERSCHIL
FROM    BOETES AS B

```

Voorbeeld 9.35:

```

SELECT  VARIANCE(BEDRAG)
FROM    BOETES
WHERE   SPELERSNR = 44
SELECT  BEDRAG -
        (SELECT  AVG(BEDRAG)
         FROM    BOETES
         WHERE   SPELERSNR = 44)
FROM    BOETES

```

```

WHERE   SPELERSNR = 44
;
SELECT  SUM(P)
FROM    (SELECT  POWER(BEDRAG -
                  (SELECT  AVG(BEDRAG)
                   FROM    BOETES
                   WHERE   SPELERSNR = 44),2) AS P
         FROM    BOETES
         WHERE   SPELERSNR = 44) AS POWERS
;
SELECT  SUM(P) /
        (SELECT  COUNT(*) FROM BOETES WHERE SPELERSNR = 44)
FROM    (SELECT  POWER(BEDRAG -
                  (SELECT  AVG(BEDRAG)
                   FROM    BOETES
                   WHERE   SPELERSNR = 44),2) AS P
         FROM    BOETES
         WHERE   SPELERSNR = 44) AS POWERS

```

Voorbeeld 9.36:

```

SELECT  STDDEV(BEDRAG)
FROM    BOETES
WHERE   SPELERSNR = 44

```

Voorbeeld 9.37:

```

SELECT  ROW_NUMBER() OVER(), SPELERSNR
FROM    SPELERS
WHERE   PLAATS = 'Den Haag'

```

Voorbeeld 9.38:

```

SELECT  ROW_NUMBER() OVER(), SPELERSNR
FROM    SPELERS
WHERE   PLAATS = 'Den Haag'
ORDER BY SPELERSNR DESC

```

Voorbeeld 9.39:

```

SELECT  ROW_NUMBER() OVER(), SPELERSNR, PLAATS
FROM    SPELERS
WHERE   YEAR(GEB_DATUM) > 1962
ORDER BY PLAATS

```

Voorbeeld 9.40:

```

SELECT  ROW_NUMBER() OVER(ORDER BY SPELERSNR ASC), SPELERSNR, PLAATS
FROM    SPELERS
WHERE   YEAR(GEB_DATUM) > 1962
ORDER BY PLAATS DESC

```

Voorbeeld 9.41:

```

SELECT  ROW_NUMBER() OVER(ORDER BY SPELERSNR ASC),
        ROW_NUMBER() OVER(ORDER BY SPELERSNR DESC),
        SPELERSNR, PLAATS
FROM    SPELERS
WHERE   YEAR(GEB_DATUM) > 1962
ORDER BY PLAATS

```

Voorbeeld 9.42:

```

SELECT  VOLGNR, SPELERSNR
FROM    (SELECT  ROW_NUMBER()
         OVER(ORDER BY SPELERSNR ASC) AS VOLGNR,
         SPELERSNR

```

```

        FROM SPELERS) AS T
WHERE VOLGNR <= 5
ORDER BY VOLGNR

```

Voorbeeld 9.43:

```

SELECT ROW_NUMBER() OVER(ORDER BY BONDSNR ASC NULLS FIRST)
       AS VOLGNR, BONDSNR
FROM   SPELERS

```

Voorbeeld 9.44:

```

SELECT PLAATS,
       ROW_NUMBER() OVER(ORDER BY PLAATS ASC),
       RANK() OVER(ORDER BY PLAATS ASC),
       DENSE_RANK() OVER(ORDER BY PLAATS ASC)
FROM   SPELERS
ORDER BY PLAATS

```

Voorbeeld 9.45:

```

SELECT AVG(BEDRAG) AS MEDIAAN
FROM   (SELECT BEDRAG,
              ROW_NUMBER() OVER(ORDER BY BEDRAG ASC) AS VOLGNR
        FROM   BOETES
        ORDER BY BEDRAG) AS T
WHERE  T.VOLGNR =
       ROUND(CAST((SELECT COUNT(*) FROM BOETES)
                 AS DECIMAL(8,1)) / 2, 0)
OR     T.VOLGNR =
       TRUNCATE((SELECT COUNT(*) FROM BOETES) / 2, 0) + 1

```

Voorbeeld 9.46:

```

SELECT SPELERSNR, SUM(BEDRAG),
       ROW_NUMBER() OVER(ORDER BY SUM(BEDRAG) ASC)
FROM   BOETES
GROUP BY SPELERSNR
ORDER BY SPELERSNR

```

Voorbeeld 9.47:

```

SELECT SPELERSNR, PLAATS,
       ROW_NUMBER() OVER(PARTITION BY PLAATS)
FROM   SPELERS
ORDER BY PLAATS
;
SELECT SPELERSNR, PLAATS,
       ROW_NUMBER() OVER(PARTITION BY PLAATS)
FROM   SPELERS
ORDER BY SPELERSNR

```

Voorbeeld 9.48:

```

SELECT SPELERSNR, PLAATS, YEAR(GEB_DATUM), GESLACHT,
       ROW_NUMBER() OVER (PARTITION BY YEAR(GEB_DATUM), GESLACHT)
FROM   SPELERS

```

Voorbeeld 9.49:

```

SELECT SPELERSNR, PLAATS, YEAR(GEB_DATUM), GESLACHT,
       ROW_NUMBER() OVER (PARTITION BY YEAR(GEB_DATUM), GESLACHT
                          ORDER BY SPELERSNR DESC)
FROM   SPELERS

```

Voorbeeld 9.50:

```
SELECT  GEB_DATUM, SPELERSNR,
        ROW_NUMBER() OVER (PARTITION BY CAST(YEAR(GEB_DATUM) / 10 AS INTEGER)
                          ORDER BY GEB_DATUM ASC)
FROM    SPELERS
ORDER BY GEB_DATUM, SPELERSNR
```

Voorbeeld 9.51:

```
SELECT  BETALINGSNR, BEDRAG,
        BEDRAG * 100 / (SELECT SUM(BEDRAG) FROM BOETES)
FROM    BOETES
ORDER BY BETALINGSNR
;
SELECT  BETALINGSNR, BEDRAG, SUM(BEDRAG) OVER (),
        BEDRAG * 100 / SUM(BEDRAG) OVER () AS PERCENTAGE
FROM    BOETES
ORDER BY BETALINGSNR
```

Voorbeeld 9.52:

```
SELECT  SPELERSNR, BEDRAG,
        SUM(BEDRAG) OVER (PARTITION BY SPELERSNR)
FROM    BOETES
ORDER BY SPELERSNR
```

Voorbeeld 9.53:

```
SELECT  SPELERSNR, BEDRAG,
        SUM(BEDRAG) OVER (ORDER BY SPELERSNR),
        COUNT(*) OVER (ORDER BY SPELERSNR)
FROM    BOETES
ORDER BY SPELERSNR
;
SELECT  SPELERSNR, BEDRAG,
        (SELECT SUM(BEDRAG)
         FROM   BOETES AS B2
         WHERE  B2.SPELERSNR <= B1.SPELERSNR)
FROM    BOETES AS B1
ORDER BY SPELERSNR
```

Voorbeeld 9.54:

```
SELECT  SPELERSNR, BEDRAG,
        SUM(BEDRAG) OVER (ORDER BY SPELERSNR
                          ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
FROM    BOETES
ORDER BY SPELERSNR
```

Voorbeeld 9.55:

```
SELECT  TEAMNR, SPELERSNR, GEWONNEN,
        SUM(GEWONNEN) OVER (PARTITION BY TEAMNR ORDER BY SPELERSNR
                          ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
FROM    WEDSTRIJDEN
ORDER BY TEAMNR, SPELERSNR
```

Voorbeeld 9.56:

```
SELECT  TEAMNR, SPELERSNR, GEWONNEN,
        SUM(GEWONNEN) OVER (PARTITION BY TEAMNR ORDER BY SPELERSNR
                          ROWS BETWEEN 2 PRECEDING AND CURRENT ROW)
FROM    WEDSTRIJDEN
ORDER BY TEAMNR, SPELERSNR
```

Voorbeeld 9.57:

```
SELECT  TEAMNR, SPELERSNR, GEWONNEN,
```

```

        SUM(GEWONNEN) OVER (PARTITION BY TEAMNR ORDER BY SPELERSNR
        ROWS BETWEEN 2 PRECEDING AND 1 FOLLOWING)
FROM    WEDSTRIJDEN
ORDER BY TEAMNR, SPELERSNR

```

4.7 SQL-instructies voor hoofdstuk 10

Voorbeeld 10.1:

```

SELECT  PLAATS
FROM    SPELERS
GROUP BY PLAATS

```

Voorbeeld 10.2:

```

SELECT  PLAATS, COUNT(*)
FROM    SPELERS
GROUP BY PLAATS

```

Voorbeeld 10.3:

```

SELECT  TEAMNR, COUNT(*), SUM(GEWONNEN)
FROM    WEDSTRIJDEN
GROUP BY TEAMNR

```

Voorbeeld 10.4:

```

SELECT  TEAMNR, COUNT(*)
FROM    WEDSTRIJDEN
WHERE   TEAMNR IN
        (SELECT  TEAMNR
         FROM    TEAMS INNER JOIN SPELERS
              ON TEAMS.SPELERSNR = SPELERS.SPELERSNR
         WHERE   PLAATS = 'Zoetermeer')
GROUP BY TEAMNR

```

Voorbeeld 10.5:

```

SELECT  BEDRAG, COUNT(*), SUM(BEDRAG)
FROM    BOETES
GROUP BY BEDRAG

```

Voorbeeld 10.6:

```

SELECT  TEAMNR, SPELERSNR
FROM    WEDSTRIJDEN
GROUP BY TEAMNR, SPELERSNR
;
SELECT  TEAMNR, SPELERSNR
FROM    WEDSTRIJDEN
GROUP BY SPELERSNR, TEAMNR
;
SELECT  TEAMNR, SPELERSNR, SUM(GEWONNEN),
        COUNT(*), MIN(VERLOREN)
FROM    WEDSTRIJDEN
GROUP BY TEAMNR, SPELERSNR

```

Voorbeeld 10.7:

```

SELECT  S.SPELERSNR, NAAM, SUM(BEDRAG)
FROM    SPELERS AS S INNER JOIN BOETES AS BT
        ON S.SPELERSNR = BT.SPELERSNR
GROUP BY S.SPELERSNR, NAAM

```


Voorbeeld 10.8:

```
SELECT YEAR(DATUM) AS JAARTAL, COUNT(*)
FROM BOETES
GROUP BY YEAR(DATUM)
```

Voorbeeld 10.9:

```
SELECT TRUNCATE(SPELERSNR/25,0) AS SPELERSGROEP,
COUNT(*), MAX(SPELERSNR)
FROM SPELERS
GROUP BY TRUNCATE(SPELERSNR/25,0)
```

Voorbeeld 10.10:

```
SELECT BONDSNR
FROM SPELERS
GROUP BY BONDSNR
;
SELECT PLAATS, COUNT(*)
FROM SPELERS
GROUP BY GESLACHT
```

Voorbeeld 10.11:

```
SELECT CAST(BEDRAG * 100 AS SIGNED INTEGER) AS BEDRAG_IN_CENTEN
FROM BOETES
GROUP BY BEDRAG
```

Voorbeeld 10.12:

```
SELECT AVG(TOTAAL)
FROM (SELECT SPELERSNR, SUM(BEDRAG) AS TOTAAL
FROM BOETES
GROUP BY SPELERSNR) AS TOTALEN
WHERE SPELERSNR IN
(SELECT SPELERSNR
FROM SPELERS
WHERE PLAATS = 'Den Haag' OR PLAATS = 'Rijswijk')
```

Voorbeeld 10.13:

```
SELECT SPELERS.SPELERSNR, NAAM, AANTALBOETES, AANTALTEAMS
FROM SPELERS,
(SELECT SPELERSNR, COUNT(*) AS AANTALBOETES
FROM BOETES
GROUP BY SPELERSNR) AS AANTAL_BOETES,
(SELECT SPELERSNR, COUNT(*) AS AANTALTEAMS
FROM TEAMS
GROUP BY SPELERSNR) AS AANTAL_TEAMS
WHERE SPELERS.SPELERSNR = AANTAL_BOETES.SPELERSNR
AND SPELERS.SPELERSNR = AANTAL_TEAMS.SPELERSNR
;
SELECT SPELERS.SPELERSNR, NAAM,
(SELECT COUNT(*)
FROM BOETES
WHERE SPELERS.SPELERSNR = BOETES.SPELERSNR) AS AANTALBOETES,
(SELECT COUNT(*)
FROM TEAMS
WHERE SPELERS.SPELERSNR = TEAMS.SPELERSNR) AS AANTALTEAMS
FROM SPELERS
```

Voorbeeld 10.14:

```
SELECT DISTINCT W.SPELERSNR, AANTALB
FROM WEDSTRIJDEN AS W LEFT OUTER JOIN
(SELECT SPELERSNR, COUNT(*) AS AANTALB
```

```

FROM BOETES
GROUP BY SPELERSNR) AS AB
ON W.SPELERSNR = AB.SPELERSNR

```

Voorbeeld 10.15:

```

SELECT GROEPEN.GROEP, SUM(B.BEDRAG)
FROM BOETES AS B,
(SELECT 1 AS GROEP, '1980-01-01' AS STARTDATUM, '1981-06-30' AS EINDDATUM
UNION
SELECT 2, '1981-07-01', '1982-12-31'
UNION
SELECT 3, '1983-01-01', '1984-12-31') AS GROEPEN
WHERE B.DATUM BETWEEN STARTDATUM AND EINDDATUM
GROUP BY GROEPEN.GROEP
ORDER BY GROEPEN.GROEP

```

Voorbeeld 10.16:

```

SELECT CASE
WHEN BEDRAG > 70.00 THEN 'SERIEUS'
WHEN BEDRAG > 35.00 THEN 'GEMIDDELD'
ELSE 'LAAG'
END AS SERIEUSHEID,
SUM(BEDRAG) AS TOTAAL_BOETE_BEDRAG
FROM BOETES
GROUP BY CASE
WHEN BEDRAG > 70.00 THEN 'SERIEUS'
WHEN BEDRAG > 35.00 THEN 'GEMIDDELD'
ELSE 'LAAG'
END
ORDER BY TOTAAL_BOETE_BEDRAG

```

Voorbeeld 10.17:

```

SELECT B1.BETALINGSNR, B1.BEDRAG, SUM(B2.BEDRAG)
FROM BOETES AS B1, BOETES AS B2
WHERE B1.BETALINGSNR >= B2.BETALINGSNR
GROUP BY B1.BETALINGSNR, B1.BEDRAG
ORDER BY B1.BETALINGSNR

```

Voorbeeld 10.18:

```

SELECT B1.BETALINGSNR, B1.BEDRAG,
(B1.BEDRAG * 100) / SUM(B2.BEDRAG)
FROM BOETES AS B1, BOETES AS B2
GROUP BY B1.BETALINGSNR, B1.BEDRAG
ORDER BY B1.BETALINGSNR

```

Voorbeeld 10.19:

```

SELECT SPELERSNR, SUM(BEDRAG)
FROM BOETES
GROUP BY SPELERSNR
UNION
SELECT CAST(NULL AS SIGNED INTEGER), SUM(BEDRAG)
FROM BOETES
;
SELECT SPELERSNR, SUM(BEDRAG)
FROM BOETES
GROUP BY SPELERSNR WITH ROLLUP

```

Voorbeeld 10.20:

```

SELECT GESLACHT, PLAATS, COUNT(*)
FROM SPELERS
GROUP BY GESLACHT, PLAATS WITH ROLLUP

```

Voorbeeld 10.21:

```
SELECT  ROW_NUMBER() OVER ( ) AS VOLGNR,
        GESLACHT, PLAATS, COUNT(*)
FROM    SPELERS
GROUP BY GESLACHT, PLAATS WITH CUBE
ORDER BY GESLACHT, PLAATS
```

Voorbeeld 10.22:

```
SELECT  PLAATS, MIN(GEB_DATUM)
FROM    SPELERS
GROUP BY PLAATS
;
SELECT  PLAATS, MIN(GEB_DATUM)
FROM    SPELERS
GROUP BY GROUPING SETS ((PLAATS))
```

Voorbeeld 10.23:

```
SELECT  CAST(NULL AS CHAR) AS GESLACHT, PLAATS, COUNT(*)
FROM    SPELERS
GROUP BY PLAATS
UNION
SELECT  GESLACHT, CAST(NULL AS CHAR), COUNT(*)
FROM    SPELERS
GROUP BY GESLACHT
ORDER BY PLAATS, GESLACHT
;
SELECT  GESLACHT, PLAATS, COUNT(*)
FROM    SPELERS
GROUP BY GROUPING SETS ((PLAATS), (GESLACHT))
ORDER BY PLAATS, GESLACHT
```

Voorbeeld 10.24:

```
SELECT  GESLACHT, PLAATS, COUNT(*)
FROM    SPELERS
GROUP BY GROUPING SETS ((GESLACHT, PLAATS), (GESLACHT), ())
ORDER BY PLAATS, GESLACHT
```

Voorbeeld 10.25:

```
SELECT  TEAMNR, SPELERSNR, COUNT(*)
FROM    WEDSTRIJDEN
GROUP BY GROUPING SETS (TEAMNR, SPELERSNR)
ORDER BY SPELERSNR, TEAMNR
```

Voorbeeld 10.26:

```
SELECT  GESLACHT, PLAATS, COUNT(*)
FROM    SPELERS
GROUP BY ROLLUP (GESLACHT, PLAATS)
ORDER BY GESLACHT, PLAATS
```

Voorbeeld 10.27:

```
SELECT  ROW_NUMBER() OVER ( ) AS VOLGNR,
        GESLACHT, PLAATS, YEAR(GEB_DATUM), COUNT(*)
FROM    SPELERS
GROUP BY ROLLUP (GESLACHT, PLAATS, YEAR(GEB_DATUM))
ORDER BY GESLACHT, PLAATS, YEAR(GEB_DATUM)
```

Voorbeeld 10.28:

```
SELECT  ROW_NUMBER() OVER ( ) AS VOLGNR,
```

```

        GESLACHT, PLAATS, YEAR(GEB_DATUM), COUNT(*)
FROM    SPELERS
GROUP BY ROLLUP (GESLACHT, (PLAATS, YEAR(GEB_DATUM)))
ORDER BY GESLACHT, PLAATS, YEAR(GEB_DATUM)

```

Voorbeeld 10.29:

```

SELECT  ROW_NUMBER() OVER () AS VOLGNR,
        GESLACHT, PLAATS, COUNT(*)
FROM    SPELERS
GROUP BY CUBE (GESLACHT, PLAATS)
ORDER BY GESLACHT, PLAATS

```

4.8 SQL-instructies voor hoofdstuk 11

Voorbeeld 11.1:

```

SELECT  SPELERSNR
FROM    BOETES
GROUP BY SPELERSNR
HAVING  COUNT(*) > 1

```

Voorbeeld 11.2:

```

SELECT  SPELERSNR
FROM    BOETES
GROUP BY SPELERSNR
HAVING  MAX(YEAR(DATUM)) = 1984

```

Voorbeeld 11.3:

```

SELECT  SPELERSNR, SUM(BEDRAG)
FROM    BOETES
GROUP BY SPELERSNR
HAVING  SUM(BEDRAG) > 150

```

Voorbeeld 11.4:

```

SELECT  SPELERSNR, SUM(BEDRAG)
FROM    BOETES
WHERE   SPELERSNR IN
        (SELECT  SPELERSNR
         FROM    TEAMS)
GROUP BY SPELERSNR
HAVING  SUM(BEDRAG) > 80

```

Voorbeeld 11.5:

```

SELECT  SPELERSNR, SUM(BEDRAG)
FROM    BOETES
GROUP BY SPELERSNR
HAVING  SUM(BEDRAG) >= ALL
        (SELECT  SUM(BEDRAG)
         FROM    BOETES
         GROUP BY SPELERSNR)

```

Voorbeeld 11.6:

```

SELECT  SPELERSNR, NAAM
FROM    SPELERS
WHERE   SPELERSNR IN
        (SELECT  SPELERSNR
         FROM    WEDSTRIJDEN
         GROUP BY SPELERSNR)

```

```

        HAVING SUM(GEWONNEN) >= ALL
              (SELECT SUM(GEWONNEN)
               FROM WEDSTRIJDEN
               GROUP BY SPELERSNR)
;
SELECT PLAATS, COUNT(*)
FROM SPELERS
GROUP BY PLAATS
HAVING GEB_DATUM > '1970-01-01'

```

4.9 SQL-instructies voor hoofdstuk 12

Voorbeeld 12.1:

```

SELECT BETALINGSNR, SPELERSNR
FROM BOETES
ORDER BY SPELERSNR

```

Voorbeeld 12.2:

```

SELECT SPELERSNR, BEDRAG
FROM BOETES
ORDER BY SPELERSNR, BEDRAG

```

Voorbeeld 12.3:

```

SELECT BEDRAG
FROM BOETES
ORDER BY SPELERSNR, BEDRAG

```

Voorbeeld 12.4:

```

SELECT NAAM, VOORLETTERS, SPELERSNR
FROM SPELERS
ORDER BY SUBSTR(NAAM, 1, 1)

```

Voorbeeld 12.5:

```

SELECT SPELERSNR, BEDRAG
FROM BOETES
ORDER BY ABS(BEDRAG - (SELECT AVG(BEDRAG) FROM BOETES))

```

Voorbeeld 12.6:

```

SELECT SPELERSNR, BEDRAG
FROM BOETES AS B1
ORDER BY (SELECT AVG(BEDRAG)
          FROM BOETES AS B2
          WHERE B1.SPELERSNR = B2.SPELERSNR)
;
SELECT BETALINGSNR, SPELERSNR
FROM BOETES
ORDER BY SPELERSNR
;
SELECT BETALINGSNR, SPELERSNR
FROM BOETES
ORDER BY 2

```

Voorbeeld 12.7:

```

SELECT SPELERSNR, SUM(BEDRAG)
FROM BOETES
GROUP BY SPELERSNR
ORDER BY 2

```

Voorbeeld 12.8:

```
SELECT  SPELERSNR, NAAM,
        (SELECT  SUM(BEDRAG)
         FROM    BOETES AS B
         WHERE   B.SPELERSNR = S.SPELERSNR)
FROM    SPELERS AS S
ORDER BY 3
;
SELECT  SPELERSNR, NAAM,
        (SELECT  SUM(BEDRAG)
         FROM    BOETES AS B
         WHERE   B.SPELERSNR = S.SPELERSNR) AS TOTAAL
FROM    SPELERS AS S
ORDER BY TOTAAL
```

Voorbeeld 12.9:

```
SELECT  SPELERSNR, BEDRAG
FROM    BOETES
ORDER BY SPELERSNR DESC, BEDRAG ASC
```

Voorbeeld 12.10:

```
CREATE TABLE CODES (
    CODE CHAR(4) NOT NULL)
;
INSERT INTO CODES VALUES ('abc')
;
INSERT INTO CODES VALUES ('ABC')
;
INSERT INTO CODES VALUES ('-abc')
;
INSERT INTO CODES VALUES ('a bc')
;
INSERT INTO CODES VALUES ('ab')
;
INSERT INTO CODES VALUES ('9abc')
;
SELECT  *
FROM    CODES
ORDER BY CODE
```

Voorbeeld 12.11:

```
SELECT  DISTINCT BONDSNR
FROM    SPELERS
ORDER BY BONDSNR DESC
```

4.10 SQL-instructies voor hoofdstuk 13

Voorbeeld 13.1:

```
SELECT  MAX(SPELERSNR)
FROM    SPELERS
;
SELECT  SPELERSNR, NAAM
FROM    SPELERS AS S1
WHERE   4 >
        (SELECT  COUNT(*)
         FROM    SPELERS AS S2
         WHERE   S1.SPELERSNR < S2.SPELERSNR)
ORDER BY SPELERSNR DESC
```

```

;
SELECT SPELERSNR, NAAM
FROM SPELERS
ORDER BY SPELERSNR DESC
LIMIT 4

```

Voorbeeld 13.2:

```

SELECT BONDSNR, SPELERSNR, NAAM
FROM SPELERS
ORDER BY BONDSNR ASC
LIMIT 5

```

Voorbeeld 13.3:

```

SELECT SPELERSNR, COUNT(*) AS AANTAL
FROM WEDSTRIJDEN
WHERE GEWONNEN > VERLOREN
GROUP BY SPELERSNR
ORDER BY AANTAL DESC
LIMIT 3

```

Voorbeeld 13.4:

```

SELECT SPELERSNR, COUNT(*) AS AANTAL
FROM WEDSTRIJDEN
WHERE GEWONNEN > VERLOREN
GROUP BY SPELERSNR
ORDER BY AANTAL DESC, SPELERSNR DESC
LIMIT 3

```

Voorbeeld 13.5:

```

SELECT *
FROM (SELECT SPELERSNR, COUNT(*) AS AANTAL
      FROM WEDSTRIJDEN
      WHERE GEWONNEN > VERLOREN
      GROUP BY SPELERSNR
      ORDER BY AANTAL DESC, SPELERSNR DESC
      LIMIT 3) AS T
ORDER BY 1

```

Voorbeeld 13.6:

```

SELECT AVG(BEDRAG)
FROM (SELECT BEDRAG
      FROM BOETES
      ORDER BY BEDRAG
      LIMIT 4) AS T

```

Voorbeeld 13.7:

```

SELECT MIN(BEDRAG)
FROM (SELECT BEDRAG
      FROM BOETES
      ORDER BY BEDRAG DESC
      LIMIT 3) AS T

```

Voorbeeld 13.8:

```

SELECT DISTINCT BEDRAG
FROM BOETES
ORDER BY BEDRAG DESC
LIMIT 3

```

Voorbeeld 13.9:

```

SELECT  SPELERSNR
FROM    (SELECT  SPELERSNR
        FROM    SPELERS
        WHERE   BONDSNR IS NOT NULL
        ORDER BY BONDSNR DESC
        LIMIT   6) AS T
ORDER BY SPELERSNR
LIMIT   3

```

Voorbeeld 13.10:

```

SELECT  SPELERSNR, NAAM
FROM    SPELERS
WHERE   SPELERSNR IN
        (SELECT  SPELERSNR
        FROM    (SELECT  SPELERSNR, SUM(BEDRAG) AS TOTAAL
                FROM    BOETES
                GROUP BY SPELERSNR
                ORDER BY TOTAAL DESC
                LIMIT   3) AS T)

```

Voorbeeld 13.11:

```

SELECT  SPELERSNR, NAAM
FROM    SPELERS
WHERE   SPELERSNR IN
        (SELECT  SPELERSNR
        FROM    BOETES)
AND     SPELERSNR NOT IN
        (SELECT  SPELERSNR
        FROM    BOETES
        ORDER BY BEDRAG DESC
        LIMIT   2)
AND     SPELERSNR NOT IN
        (SELECT  SPELERSNR
        FROM    BOETES
        ORDER BY BEDRAG ASC
        LIMIT   2)
;
SELECT  SPELERSNR, NAAM
FROM    SPELERS
WHERE   SPELERSNR IN
        (SELECT  SPELERSNR
        FROM    BOETES
        WHERE   SPELERSNR NOT IN
                (SELECT  SPELERSNR
                FROM    BOETES
                ORDER BY BEDRAG DESC
                LIMIT   2)
        AND     SPELERSNR NOT IN
                (SELECT  SPELERSNR
                FROM    BOETES
                ORDER BY BEDRAG ASC
                LIMIT   2))

```

Voorbeeld 13.12:

```

SELECT  SPELERSNR, NAAM
FROM    SPELERS
ORDER BY SPELERSNR ASC
LIMIT   5 OFFSET 3

```

4.11 SQL-instructies voor hoofdstuk 14

Voorbeeld 14.1:

```
SELECT SPELERSNR, PLAATS
FROM SPELERS
WHERE PLAATS = 'Rijswijk'
UNION
SELECT SPELERSNR, PLAATS
FROM SPELERS
WHERE PLAATS = 'Rotterdam'
;
SELECT SPELERSNR, PLAATS
FROM SPELERS
WHERE PLAATS = 'Rijswijk'
OR PLAATS = 'Rotterdam'
```

Voorbeeld 14.2:

```
SELECT GEB_DATUM AS DATUMS
FROM SPELERS
UNION
SELECT DATUM
FROM BOETES
```

Voorbeeld 14.3:

```
SELECT SPELERSNR
FROM BOETES
UNION
SELECT SPELERSNR
FROM TEAMS
```

Voorbeeld 14.4:

```
SELECT SPELERSNR
FROM BOETES
UNION
SELECT SPELERSNR
FROM TEAMS
UNION
SELECT SPELERSNR
FROM SPELERS
WHERE PLAATS = 'Den Haag'
;
SELECT *
FROM SPELERS
UNION
SELECT *
FROM BOETES
;
SELECT SPELERSNR
FROM SPELERS
WHERE PLAATS = 'Den Haag'
ORDER BY SPELERSNR
UNION
SELECT SPELERSNR
FROM TEAMS
ORDER BY SPELERSNR
```

Voorbeeld 14.5:

```
SELECT CAST(TEAMNR AS CHAR(4)) AS TEAMNR,
       CAST(SPELERSNR AS CHAR(4)) AS SPELERSNR,
       SUM(GEWONNEN + VERLOREN) AS TOTAAL
FROM WEDSTRIJDEN
GROUP BY TEAMNR, SPELERSNR
UNION
SELECT CAST(TEAMNR AS CHAR(4)),
```

```

        'subtotaal',
        SUM(GEWONNEN + VERLOREN)
FROM    WEDSTRIJDEN
GROUP BY TEAMNR
UNION
SELECT  'totaal', 'totaal', SUM(GEWONNEN + VERLOREN)
FROM    WEDSTRIJDEN
ORDER BY TEAMNR, SPELERSNR

```

Voorbeeld 14.6:

```

SELECT  SPELERSNR, GEB_DATUM
FROM    SPELERS
WHERE   PLAATS = 'Den Haag'
INTERSECT
SELECT  SPELERSNR, GEB_DATUM
FROM    SPELERS
WHERE   GEB_DATUM > '1960-12-31'
ORDER BY SPELERSNR
;
SELECT  SPELERSNR, GEB_DATUM
FROM    SPELERS
WHERE   PLAATS = 'Den Haag'
AND     GEB_DATUM > '1960-12-31'
ORDER BY SPELERSNR

```

Voorbeeld 14.7:

```

SELECT  SPELERSNR
FROM    TEAMS
INTERSECT
SELECT  SPELERSNR
FROM    BOETES

```

Voorbeeld 14.8:

```

SELECT  SPELERSNR, NAAM
FROM    SPELERS
WHERE   SPELERSNR IN
        (SELECT  SPELERSNR
         FROM    TEAMS
         INTERSECT
         SELECT  SPELERSNR
         FROM    BOETES)

```

Voorbeeld 14.9:

```

SELECT  POWER(GETAL,2) AS MACHTEN
FROM    (SELECT  CAST(CIJFER1.CIJFER || CIJFER2.CIJFER
                 AS UNSIGNED INTEGER) AS GETAL
        FROM    (SELECT '0' AS CIJFER UNION SELECT '1' UNION
                 SELECT '2' UNION SELECT '3' UNION
                 SELECT '4' UNION SELECT '5' UNION
                 SELECT '6' UNION SELECT '7' UNION
                 SELECT '8' UNION SELECT '9') AS CIJFER1,
        (SELECT '0' AS CIJFER UNION SELECT '1' UNION
                 SELECT '2' UNION SELECT '3' UNION
                 SELECT '4' UNION SELECT '5' UNION
                 SELECT '6' UNION SELECT '7' UNION
                 SELECT '8' UNION SELECT '9') AS CIJFER2)
        AS GETALLEN
WHERE   POWER(GETAL,2) < 5000
INTERSECT
SELECT  POWER(GETAL,3) AS MACHTEN
FROM    (SELECT  CAST(CIJFER1.CIJFER || CIJFER2.CIJFER
                 AS UNSIGNED INTEGER) AS GETAL
        FROM    (SELECT '0' AS CIJFER UNION SELECT '1' UNION

```

```

        SELECT '2' UNION SELECT '3' UNION
        SELECT '4' UNION SELECT '5' UNION
        SELECT '6' UNION SELECT '7' UNION
        SELECT '8' UNION SELECT '9') AS CIJFER1,
    (SELECT '0' AS CIJFER UNION SELECT '1' UNION
    SELECT '2' UNION SELECT '3' UNION
    SELECT '4' UNION SELECT '5' UNION
    SELECT '6' UNION SELECT '7' UNION
    SELECT '8' UNION SELECT '9') AS CIJFER2)
    AS GETALLEN
WHERE     POWER(GETAL,3) < 5000

```

Voorbeeld 14.10:

```

SELECT  SPELERSNR, GEB_DATUM
FROM    SPELERS
WHERE   PLAATS = 'Den Haag'
EXCEPT
SELECT  SPELERSNR, GEB_DATUM
FROM    SPELERS
WHERE   GEB_DATUM > '1960-12-31'
ORDER BY SPELERSNR
;
SELECT  SPELERSNR, GEB_DATUM
FROM    SPELERS
WHERE   PLAATS = 'Den Haag'
AND     NOT(GEB_DATUM > '1960-12-31')
ORDER BY SPELERSNR

```

Voorbeeld 14.11:

```

SELECT  SPELERSNR, NAAM
FROM    SPELERS
WHERE   SPELERSNR IN
        (SELECT  SPELERSNR
         FROM    BOETES
         EXCEPT
         SELECT  SPELERSNR
         FROM    TEAMS)
;
SELECT  SPELERSNR
FROM    TEAMS
INTERSECT
SELECT  SPELERSNR
FROM    BOETES
;
SELECT  SPELERSNR
FROM    TEAMS
EXCEPT (SELECT  SPELERSNR
         FROM    TEAMS
         EXCEPT
         SELECT  SPELERSNR
         FROM    BOETES)

```

Voorbeeld 14.12:

```

SELECT  SPELERSNR
FROM    BOETES
EXCEPT
(SELECT  6
 UNION
 SELECT  27
 UNION
 SELECT  58)

```

Voorbeeld 14.13:

```

SELECT  SPELERSNR
FROM    BOETES
UNION ALL
SELECT  SPELERSNR
FROM    TEAMS

```

Voorbeeld 14.14:

```

SELECT  SPELERSNR
FROM    BOETES
EXCEPT ALL
SELECT  SPELERSNR
FROM    TEAMS
;
SELECT  SPELERSNR, BONDSNR
FROM    SPELERS
WHERE   SPELERSNR = 27
UNION
SELECT  SPELERSNR, BONDSNR
FROM    SPELERS
WHERE   SPELERSNR = 27

```

Voorbeeld 14.15:

```

SELECT  SPELERSNR
FROM    BOETES
EXCEPT
SELECT  SPELERSNR
FROM    TEAMS
UNION
SELECT  SPELERSNR
FROM    SPELERS
WHERE   PLAATS = 'Zoetermeer'
;
SELECT  SPELERSNR
FROM    BOETES
EXCEPT
(SELECT  SPELERSNR
FROM    TEAMS
UNION
SELECT  SPELERSNR
FROM    SPELERS
WHERE   PLAATS = 'Zoetermeer')

```

4.12 SQL-instructies voor hoofdstuk 15

Voorbeeld 15.1:

```

WITH    AANTALLEN (AANTAL) AS
        (SELECT  COUNT(*)
         FROM    BOETES
         GROUP BY SPELERSNR)
SELECT  AVG(AANTAL*1.0)
FROM    AANTALLEN
;
SELECT  COUNT(*) AS AANTAL
FROM    BOETES
GROUP BY SPELERSNR
;
SELECT  AVG(AANTAL*1.0)
FROM    AANTALLEN
;
SELECT  AVG(AANTAL*1.0)
FROM    (SELECT  COUNT(*)

```

```

FROM    BOETES
GROUP BY SPELERSNR) AS AANTALLEN (AANTAL)

```

Voorbeeld 15.2:

```

SELECT  BETALINGSNR, BEDRAG,
        (SELECT  MIN(BEDRAG) AS MIN_BEDRAG
         FROM    BOETES
         WHERE   SPELERSNR = 44),
        (SELECT  MAX(BEDRAG) AS MAX_BEDRAG
         FROM    BOETES
         WHERE   SPELERSNR = 44)
FROM    BOETES
WHERE   BEDRAG >
        (SELECT  MIN(BEDRAG)
         FROM    BOETES
         WHERE   SPELERSNR = 44)
AND     BEDRAG <
        (SELECT  MAX(BEDRAG)
         FROM    BOETES
         WHERE   SPELERSNR = 44)
;
WITH    MAX_MIN (MAX_BEDRAG, MIN_BEDRAG) AS
        (SELECT  MAX(BEDRAG), MIN(BEDRAG)
         FROM    BOETES
         WHERE   SPELERSNR = 44)
SELECT  B.BETALINGSNR, B.BEDRAG, MM.MIN_BEDRAG, MM.MAX_BEDRAG
FROM    BOETES AS B, MAX_MIN AS MM
WHERE   B.BEDRAG > MM.MIN_BEDRAG
AND     B.BEDRAG < MM.MAX_BEDRAG

```

Voorbeeld 15.3:

```

WITH    CIJFERS (CIJFER) AS
        (SELECT 0 AS CIJFER UNION SELECT 1 UNION
         SELECT 2 UNION SELECT 3 UNION
         SELECT 4 UNION SELECT 5 UNION
         SELECT 6 UNION SELECT 7 UNION
         SELECT 8 UNION SELECT 9)
SELECT  GETAL
FROM    (SELECT  (CIJFER1.CIJFER * 100) + (CIJFER2.CIJFER * 10) + CIJFER3.CIJFER
         FROM    CIJFERS AS CIJFER1,
                CIJFERS AS CIJFER2,
                CIJFERS AS CIJFER3) AS GETALLEN (GETAL)
ORDER BY GETAL

```

Voorbeeld 15.4:

```

WITH    AANTAL_BOETES (AANTAL) AS
        (SELECT  COUNT(*)
         FROM    BOETES),
        AANTAL_WEDSTRIJDEN (AANTAL) AS
        (SELECT  COUNT(*)
         FROM    WEDSTRIJDEN)
VALUES ((SELECT AANTAL FROM AANTAL_BOETES) +
        (SELECT AANTAL FROM AANTAL_WEDSTRIJDEN))

```

Voorbeeld 15.5:

```

WITH    AANTALLEN (AANTAL) AS
        (SELECT  COUNT(*)
         FROM    BOETES
         GROUP BY SPELERSNR),
        GEMIDDELDE (WAARDE) AS
        (SELECT  AVG(AANTAL*1.0)
         FROM    AANTALLEN)
SELECT  *

```

```
FROM GEMIDDELDE
```

Voorbeeld 15.6:

```
SELECT SPELERSNR, YEAR(GEB_DATUM),
       CAST(CAST(CURRENT_DATE AS CHAR(4)) || '-01-01' AS DATE) - 20 YEARS AS LANG_GELEDEN
FROM   SPELERS
WHERE  YEAR(GEB_DATUM) =
       CAST(CAST(CURRENT_DATE AS CHAR(4)) || '-01-01' AS DATE) - 20 YEARS
;
WITH   TWINTIGJAAR (LANG_GELEDEN) AS
      (SELECT CAST(CAST(CURRENT_DATE AS CHAR(4)) || '-01-01' AS DATE) - INTERVAL 20 YEAR)
SELECT SPELERSNR, YEAR(GEB_DATUM), TWINTIGJAAR.LANG_GELEDEN
FROM   SPELERS, TWINTIGJAAR
WHERE  YEAR(GEB_DATUM) = TWINTIGJAAR.LANG_GELEDEN
```

Voorbeeld 15.7:

```
WITH   GETALLEN (GETAL) AS
      (SELECT 1
       UNION ALL
       SELECT GETAL + 1
       FROM   GETALLEN
       WHERE  GETAL < 5)
SELECT *
FROM   GETALLEN
```

Voorbeeld 15.8:

```
WITH   GETALLEN (GETAL) AS
      (SELECT 1
       UNION ALL
       SELECT GETAL + 1
       FROM   GETALLEN
       WHERE  GETAL < 1000)
SELECT MAX(GETAL) AS GROOTSTE_KWADRAAT
FROM   GETALLEN
WHERE  GETAL = POWER(INTEGER(SQRT(GETAL)),2)
;
WITH   GETALLEN (GETAL) AS
      (SELECT 1000
       UNION ALL
       SELECT GETAL - 1
       FROM   GETALLEN
       WHERE  GETAL > 1
       AND   GETAL <> POWER(INTEGER(SQRT(GETAL)),2))
SELECT MIN(GETAL) AS GROOTSTE_KWADRAAT
FROM   GETALLEN
```

Voorbeeld 15.9:

```
WITH   GETALLEN (SOORT, GETAL) AS
      (SELECT 'G1', 1
       FROM
       UNION ALL
       SELECT 'G2', 1001
       UNION ALL
       SELECT SOORT, GETAL + 1
       FROM   GETALLEN
       WHERE  GETAL < 5
       AND   SOORT = 'G1'
       UNION ALL
       SELECT SOORT, GETAL + 1
       FROM   GETALLEN
       WHERE  GETAL < 1005
       AND   SOORT = 'G2')
SELECT SUM(GETAL) AS SOM
```

```
FROM   GETALLEN
```

Voorbeeld 15.10:

```
CREATE TABLE ONDERDELEN (
  SUPER      CHAR(3) NOT NULL,
  SUB        CHAR(3) NOT NULL,
  AANTAL     INTEGER NOT NULL,
  PRIMARY KEY (SUPER, SUB)
);
INSERT INTO ONDERDELEN VALUES
('01', '02', 10),
('01', '03', 5),
('01', '04', 10),
('02', '05', 25),
('02', '06', 5),
('03', '07', 10),
('06', '08', 15),
('08', '011', 5),
('09', '010', 20),
('010', '011', 25)
```

Voorbeeld 15.11:

```
SELECT  04.SUB
FROM    ONDERDELEN AS 01,
        ONDERDELEN AS 02,
        ONDERDELEN AS 03,
        ONDERDELEN AS 04
WHERE   01.SUPER = '02'
AND     01.SUB = 02.SUPER
AND     02.SUB = 03.SUPER
AND     03.SUB = 04.SUPER
;
WITH    RELATIES (SUPER, SUB) AS
        (SELECT SUPER, SUB
         FROM   ONDERDELEN
         WHERE  SUPER = '02'
         UNION ALL
         SELECT 0.SUPER, 0.SUB
         FROM   ONDERDELEN AS 0, RELATIES AS R
         WHERE  0.SUPER = R.SUB)
SELECT  *
FROM    RELATIES
```

Voorbeeld 15.12:

```
WITH    RELATIES (SUPER, SUB, AANTAL) AS
        (SELECT *
         FROM   ONDERDELEN
         WHERE  SUPER NOT IN
              (SELECT SUB
               FROM ONDERDELEN)
         UNION ALL
         SELECT 0.*
         FROM   ONDERDELEN AS 0, RELATIES AS R
         WHERE  0.SUPER = R.SUB)
SELECT  *
FROM    RELATIES
```

Voorbeeld 15.13:

```
WITH    RELATIES (TOP, SUPER, SUB, AANTAL) AS
        (SELECT SUPER, SUPER, SUB, AANTAL
         FROM   ONDERDELEN
         WHERE  SUPER NOT IN
              (SELECT SUB
```

```

        FROM ONDERDELEN)
    UNION ALL
    SELECT R.TOP, O.SUPER, O.SUB, O.AANTAL
    FROM ONDERDELEN AS O, RELATIES AS R
    WHERE O.SUPER = R.SUB)
SELECT *
FROM RELATIES

```

Voorbeeld 15.14:

```

WITH RELATIES (SUPER, SUB, PAD) AS
(SELECT SUPER, SUB, CAST(RTRIM(SUPER)||'<-'||RTRIM(SUB) AS VARCHAR(100))
FROM ONDERDELEN
WHERE SUPER = '01'
UNION ALL
SELECT O.SUPER, O.SUB, RTRIM(R.PAD)||'<-'||RTRIM(O.SUB)
FROM ONDERDELEN AS O, RELATIES AS R
WHERE O.SUPER = R.SUB)
SELECT *
FROM RELATIES

```

Voorbeeld 15.15:

```

WITH RELATIES (SUPER, SUB, STAP) AS
(SELECT SUPER, SUB, 1
FROM ONDERDELEN
WHERE SUPER = '01'
UNION ALL
SELECT O.SUPER, O.SUB, R.STAP + 1
FROM ONDERDELEN AS O, RELATIES AS R
WHERE O.SUPER = R.SUB
AND R.STAP < 4)
SELECT *
FROM RELATIES

```

Voorbeeld 15.16:

```

WITH RELATIES (TOP, SUPER, SUB, STAP) AS
(SELECT SUPER, SUPER, SUB, 1
FROM ONDERDELEN
WHERE SUPER = '01'
UNION ALL
SELECT R.TOP, O.SUPER, O.SUB, R.STAP + 1
FROM ONDERDELEN AS O, RELATIES AS R
WHERE O.SUPER = R.SUB
AND STAP < 10)
SELECT STAP
FROM RELATIES
WHERE TOP = '01'
AND SUB = '011'

```

Voorbeeld 15.17:

```

WITH RELATIES (TOP, SUB, STAP, KLAAR) AS
(SELECT SUPER, SUB, 1,
CASE WHEN SUB IN (SELECT SUPER FROM ONDERDELEN)
THEN 'NEE' ELSE 'JA'
END
FROM ONDERDELEN
WHERE SUPER = '01'
UNION ALL
SELECT R.TOP, O.SUB, R.STAP + 1,
CASE WHEN O.SUB IN (SELECT SUPER FROM ONDERDELEN)
THEN 'NEE' ELSE 'JA'
END
FROM ONDERDELEN AS O, RELATIES AS R
WHERE O.SUPER = R.SUB)

```



```

        AND      R.KLAAR = 'NEE'
        AND      R.STAP < 10)
SELECT  TOP, SUB, STAP
FROM    RELATIES
WHERE   KLAAR = 'JA'

```

Voorbeeld 15.18:

```

CREATE TABLE ACTIVITEITEN (
  ACTIVITEIT  VARCHAR(2) NOT NULL PRIMARY KEY,
  START       CHAR(1) NOT NULL,
  EINDE       CHAR(1) NOT NULL,
  TIJDSDUUR   INTEGER NOT NULL)
;
INSERT INTO ACTIVITEITEN VALUES
( '1', 'A', 'B', 10),
( '2', 'A', 'C', 20),
( '3', 'B', 'D', 30),
( '4', 'B', 'D', 20),
( '5', 'C', 'E', 30),
( '6', 'D', 'F', 40),
( '7', 'D', 'G', 10),
( '8', 'E', 'G', 20),
( '9', 'E', 'H', 70),
( '10', 'F', 'I', 10),
( '11', 'G', 'I', 10),
( '12', 'G', 'K', 20),
( '13', 'H', 'J', 30),
( '14', 'I', 'K', 20),
( '15', 'J', 'K', 10),
( '16', 'C', 'K', 80)

```

Voorbeeld 15.19:

```

WITH    RELATIES (TOP, START, EINDE, PAD, STAP) AS
        (SELECT  START, START, EINDE,
                CAST(START||'-'||ACTIVITEIT||')-'||EINDE AS VARCHAR(100)), 1
        FROM    ACTIVITEITEN
        WHERE   START = 'A'
        UNION ALL
        SELECT  R.TOP, A.START, A.EINDE,
                R.PAD||'-'||A.ACTIVITEIT||')-'||A.EINDE, R.STAP+1
        FROM    ACTIVITEITEN AS A, RELATIES AS R
        WHERE   A.START = R.EINDE
        AND     R.STAP < 20)
SELECT  *
FROM    RELATIES
WHERE   EINDE = 'K'

```

Voorbeeld 15.20:

```

WITH    RELATIES (TOP, START, EINDE, PAD, TOTALE_TIJDSDUUR, STAP) AS
        (SELECT  START, START, EINDE,
                CAST(START||'-'||ACTIVITEIT||')-'||EINDE AS VARCHAR(100)),
                TIJDSDUUR, 1
        FROM    ACTIVITEITEN
        WHERE   START = 'A'
        UNION ALL
        SELECT  R.TOP, A.START, A.EINDE, R.PAD||'-'||A.ACTIVITEIT||')-'||A.EINDE,
                R.TOTALE_TIJDSDUUR + A.TIJDSDUUR, R.STAP+1
        FROM    ACTIVITEITEN AS A, RELATIES AS R
        WHERE   A.START = R.EINDE
        AND     R.STAP < 20)
SELECT  PAD, TOTALE_TIJDSDUUR
FROM    RELATIES
WHERE   EINDE = 'K'
AND     TOTALE_TIJDSDUUR =

```

```
(SELECT MIN(TOTALE_TIJDSDUUR)
FROM RELATIES
WHERE EINDE = 'K')
```

Voorbeeld 15.21:

```
CREATE TABLE VLUCHTEN (
  VLUCHTNR          INTEGER NOT NULL PRIMARY KEY,
  VLIEGTUIGMIJ     VARCHAR(3) NOT NULL,
  VERTREK_VLIEGVELD VARCHAR(3) NOT NULL,
  AANKOMST_VLIEGVELD VARCHAR(3) NOT NULL,
  VERTREKTIJD      TIMESTAMP NOT NULL,
  AANKOMSTTIJD     TIMESTAMP NOT NULL,
  VLUCHTDUUR       SMALLINT NOT NULL,
  PRIJS            DECIMAL(8,2))
;
INSERT INTO VLUCHTEN (VLUCHTNR, VLIEGTUIGMIJ, VERTREK_VLIEGVELD, AANKOMST_VLIEGVELD,
  VERTREKTIJD, AANKOMSTTIJD, VLUCHTDUUR, PRIJS) VALUES
(0, 'KL', 'AMS', 'LHR', '2007-03-01-11.30.00.000000', '2007-03-01-12.30.00.000000', 1, 100.18),
(1, 'KL', 'LHR', 'ORD', '2007-03-01-13.30.00.000000', '2007-03-01-19.30.00.000000', 6, 919.01),
(2, 'DL', 'ORD', 'LAX', '2007-03-01-20.30.00.000000', '2007-03-02-01.30.00.000000', 5, 732.16),
(3, 'DL', 'LAX', 'SYD', '2007-03-02-02.30.00.000000', '2007-03-02-12.30.00.000000', 10, 1214.85),
(4, 'SQ', 'AMS', 'TYO', '2007-03-01-11.00.00.000000', '2007-03-01-22.00.00.000000', 11, 1621.04),
(5, 'SQ', 'TYO', 'SYD', '2007-03-02-03.00.00.000000', '2007-03-02-14.00.00.000000', 11, 1318.74),
(6, 'KL', 'AMS', 'LAX', '2007-03-01-18.00.00.000000', '2007-03-02-07.00.00.000000', 13, 2496.25),
(7, 'KL', 'AMS', 'JFK', '2007-03-01-10.00.00.000000', '2007-03-01-16.00.00.000000', 6, 815.96),
(8, 'CO', 'JFK', 'PHX', '2007-03-01-19.00.00.000000', '2007-03-02-01.00.00.000000', 6, 1107.71),
(9, 'KL', 'AMS', 'LGA', '2007-03-01-10.00.00.000000', '2007-03-01-16.00.00.000000', 6, 972.52),
(10, 'CO', 'LGA', 'PHX', '2007-03-01-20.00.00.000000', '2007-03-02-02.00.00.000000', 6, 688.22),
(11, 'KL', 'AMS', 'EWR', '2007-03-01-10.00.00.000000', '2007-03-01-17.00.00.000000', 7, 1306.90),
(12, 'KL', 'EWR', 'PHX', '2007-03-01-19.00.00.000000', '2007-03-02-00.00.00.000000', 5, 882.12),
(13, 'KL', 'AMS', 'CAI', '2007-03-01-09.00.00.000000', '2007-03-01-16.00.00.000000', 7, 1080.28),
(14, 'KL', 'CAI', 'TYO', '2007-03-01-19.00.00.000000', '2007-03-02-00.00.00.000000', 5, 683.75),
(15, 'KL', 'AMS', 'JFK', '2007-03-01-15.00.00.000000', '2007-03-01-21.00.00.000000', 6, 713.80),
(16, 'KL', 'AMS', 'LGA', '2007-03-01-12.00.00.000000', '2007-03-01-18.00.00.000000', 6, 601.15),
(17, 'KL', 'AMS', 'LHR', '2007-03-01-15.00.00.000000', '2007-03-01-16.00.00.000000', 1, 173.82),
(18, 'KL', 'AMS', 'LHR', '2007-03-01-18.00.00.000000', '2007-03-01-19.00.00.000000', 1, 148.25),
(19, 'KL', 'AMS', 'LHR', '2007-03-01-21.00.00.000000', '2007-03-01-22.00.00.000000', 1, 147.34),
(20, 'KL', 'PHX', 'LAX', '2007-03-01-21.00.00.000000', '2007-03-01-23.00.00.000000', 2, 215.74),
(21, 'KL', 'LHR', 'AMS', '2007-03-02-09.30.00.000000', '2007-03-02-10.30.00.000000', 1, 128.03),
(22, 'KL', 'ORD', 'LHR', '2007-03-02-02.30.00.000000', '2007-03-02-08.30.00.000000', 6, 1121.04),
(23, 'DL', 'LAX', 'ORD', '2007-03-01-20.30.00.000000', '2007-03-02-01.30.00.000000', 5, 644.53),
(24, 'DL', 'SYD', 'LAX', '2007-03-01-09.30.00.000000', '2007-03-01-19.30.00.000000', 10, 1682.82),
(25, 'SQ', 'TYO', 'AMS', '2007-03-01-16.00.00.000000', '2007-03-02-03.00.00.000000', 11, 1543.06),
(26, 'SQ', 'SYD', 'TYO', '2007-03-01-10.00.00.000000', '2007-03-01-21.00.00.000000', 11, 1576.73),
(27, 'KL', 'LAX', 'AMS', '2007-03-01-18.00.00.000000', '2007-03-02-07.00.00.000000', 13, 2064.36),
(28, 'KL', 'JFK', 'AMS', '2007-03-01-21.00.00.000000', '2007-03-02-03.00.00.000000', 6, 858.91),
(29, 'CO', 'PHX', 'JFK', '2007-03-01-12.00.00.000000', '2007-03-01-18.00.00.000000', 6, 803.85),
(30, 'KL', 'LGA', 'AMS', '2007-03-01-20.00.00.000000', '2007-03-02-02.00.00.000000', 6, 1029.45),
(31, 'CO', 'PHX', 'LGA', '2007-03-01-12.00.00.000000', '2007-03-01-18.00.00.000000', 6, 919.98),
(32, 'KL', 'EWR', 'AMS', '2007-03-01-19.00.00.000000', '2007-03-02-02.00.00.000000', 7, 701.15),
(33, 'KL', 'PHX', 'EWR', '2007-03-01-13.00.00.000000', '2007-03-01-18.00.00.000000', 5, 530.71),
(34, 'KL', 'CAI', 'AMS', '2007-03-01-17.00.00.000000', '2007-03-02-00.00.00.000000', 7, 912.45),
(35, 'KL', 'TYO', 'CAI', '2007-03-01-10.00.00.000000', '2007-03-01-15.00.00.000000', 5, 682.59),
(36, 'KL', 'JFK', 'AMS', '2007-03-01-15.00.00.000000', '2007-03-01-21.00.00.000000', 6, 695.84),
(37, 'KL', 'LGA', 'AMS', '2007-03-01-12.00.00.000000', '2007-03-01-18.00.00.000000', 6, 784.61),
(38, 'KL', 'LHR', 'AMS', '2007-03-01-15.00.00.000000', '2007-03-01-16.00.00.000000', 1, 135.83),
(39, 'KL', 'LHR', 'AMS', '2007-03-01-18.00.00.000000', '2007-03-01-19.00.00.000000', 1, 195.48),
(40, 'KL', 'LHR', 'AMS', '2007-03-01-21.00.00.000000', '2007-03-01-22.00.00.000000', 1, 172.41),
(41, 'KL', 'LAX', 'PHX', '2007-03-01-21.00.00.000000', '2007-03-01-23.00.00.000000', 2, 365.42),
(42, 'KL', 'AMS', 'LHR', '2007-03-02-11.30.00.000000', '2007-03-02-12.30.00.000000', 1, 152.34),
(43, 'KL', 'LHR', 'ORD', '2007-03-02-13.30.00.000000', '2007-03-02-19.30.00.000000', 6, 722.88),
(44, 'DL', 'ORD', 'LAX', '2007-03-02-20.30.00.000000', '2007-03-03-01.30.00.000000', 5, 994.93),
(45, 'DL', 'LAX', 'SYD', '2007-03-03-02.30.00.000000', '2007-03-03-12.30.00.000000', 10, 1227.27),
(46, 'SQ', 'AMS', 'TYO', '2007-03-02-11.00.00.000000', '2007-03-02-22.00.00.000000', 11, 2198.38),
(47, 'SQ', 'TYO', 'SYD', '2007-03-03-03.00.00.000000', '2007-03-03-14.00.00.000000', 11, 2145.75),
(48, 'KL', 'AMS', 'LAX', '2007-03-02-18.00.00.000000', '2007-03-03-07.00.00.000000', 13, 2142.16),
```

(49, 'KL', 'AMS', 'JFK', '2007-03-02-10.00.00.000000', '2007-03-02-16.00.00.000000', 6,805.39),
(50, 'CO', 'JFK', 'PHX', '2007-03-02-19.00.00.000000', '2007-03-03-01.00.00.000000', 6,636.34),
(51, 'KL', 'AMS', 'LGA', '2007-03-02-10.00.00.000000', '2007-03-02-16.00.00.000000', 6,912.84),
(52, 'CO', 'LGA', 'PHX', '2007-03-02-20.00.00.000000', '2007-03-03-02.00.00.000000', 6,1037.45),
(53, 'KL', 'AMS', 'EWR', '2007-03-02-10.00.00.000000', '2007-03-02-17.00.00.000000', 7,1067.31),
(54, 'KL', 'EWR', 'PHX', '2007-03-02-19.00.00.000000', '2007-03-03-00.00.00.000000', 5,907.40),
(55, 'KL', 'AMS', 'CAI', '2007-03-02-09.00.00.000000', '2007-03-02-16.00.00.000000', 7,1129.80),
(56, 'KL', 'CAI', 'TYO', '2007-03-02-19.00.00.000000', '2007-03-03-00.00.00.000000', 5,803.03),
(57, 'KL', 'AMS', 'JFK', '2007-03-02-15.00.00.000000', '2007-03-02-21.00.00.000000', 6,1060.24),
(58, 'KL', 'AMS', 'LGA', '2007-03-02-12.00.00.000000', '2007-03-02-18.00.00.000000', 6,872.39),
(59, 'KL', 'AMS', 'LHR', '2007-03-02-15.00.00.000000', '2007-03-02-16.00.00.000000', 1,163.88),
(60, 'KL', 'AMS', 'LHR', '2007-03-02-18.00.00.000000', '2007-03-02-19.00.00.000000', 1,158.16),
(61, 'KL', 'AMS', 'LHR', '2007-03-02-21.00.00.000000', '2007-03-02-22.00.00.000000', 1,105.99),
(62, 'KL', 'PHX', 'LAX', '2007-03-02-21.00.00.000000', '2007-03-02-23.00.00.000000', 2,361.93),
(63, 'KL', 'LHR', 'AMS', '2007-03-03-09.30.00.000000', '2007-03-03-10.30.00.000000', 1,119.16),
(64, 'KL', 'ORD', 'LHR', '2007-03-03-02.30.00.000000', '2007-03-03-08.30.00.000000', 6,658.15),
(65, 'DL', 'LAX', 'ORD', '2007-03-02-20.30.00.000000', '2007-03-03-01.30.00.000000', 5,946.27),
(66, 'DL', 'SYD', 'LAX', '2007-03-02-09.30.00.000000', '2007-03-02-19.30.00.000000', 10,1474.99),
(67, 'SQ', 'TYO', 'AMS', '2007-03-02-16.00.00.000000', '2007-03-03-03.00.00.000000', 11,1990.65),
(68, 'SQ', 'SYD', 'TYO', '2007-03-02-10.00.00.000000', '2007-03-02-21.00.00.000000', 11,1199.26),
(69, 'KL', 'LAX', 'AMS', '2007-03-02-18.00.00.000000', '2007-03-03-07.00.00.000000', 13,1692.89),
(70, 'KL', 'JFK', 'AMS', '2007-03-02-21.00.00.000000', '2007-03-03-03.00.00.000000', 6,769.10),
(71, 'CO', 'PHX', 'JFK', '2007-03-02-12.00.00.000000', '2007-03-02-18.00.00.000000', 6,999.53),
(72, 'KL', 'LGA', 'AMS', '2007-03-02-20.00.00.000000', '2007-03-03-02.00.00.000000', 6,922.20),
(73, 'CO', 'PHX', 'LGA', '2007-03-02-12.00.00.000000', '2007-03-02-18.00.00.000000', 6,827.40),
(74, 'KL', 'EWR', 'AMS', '2007-03-02-19.00.00.000000', '2007-03-03-02.00.00.000000', 7,988.37),
(75, 'KL', 'PHX', 'EWR', '2007-03-02-13.00.00.000000', '2007-03-02-18.00.00.000000', 5,680.02),
(76, 'KL', 'CAI', 'AMS', '2007-03-02-17.00.00.000000', '2007-03-03-00.00.00.000000', 7,1191.88),
(77, 'KL', 'TYO', 'CAI', '2007-03-02-10.00.00.000000', '2007-03-02-15.00.00.000000', 5,815.66),
(78, 'KL', 'JFK', 'AMS', '2007-03-02-15.00.00.000000', '2007-03-02-21.00.00.000000', 6,1154.71),
(79, 'KL', 'LGA', 'AMS', '2007-03-02-12.00.00.000000', '2007-03-02-18.00.00.000000', 6,876.79),
(80, 'KL', 'LHR', 'AMS', '2007-03-02-15.00.00.000000', '2007-03-02-16.00.00.000000', 1,114.02),
(81, 'KL', 'LHR', 'AMS', '2007-03-02-18.00.00.000000', '2007-03-02-19.00.00.000000', 1,111.94),
(82, 'KL', 'LHR', 'AMS', '2007-03-02-21.00.00.000000', '2007-03-02-22.00.00.000000', 1,198.08),
(83, 'KL', 'LAX', 'PHX', '2007-03-02-21.00.00.000000', '2007-03-02-23.00.00.000000', 2,391.77),
(84, 'KL', 'AMS', 'LHR', '2007-03-03-11.30.00.000000', '2007-03-03-12.30.00.000000', 1,196.28),
(85, 'KL', 'LHR', 'ORD', '2007-03-03-13.30.00.000000', '2007-03-03-19.30.00.000000', 6,1057.74),
(86, 'DL', 'ORD', 'LAX', '2007-03-03-20.30.00.000000', '2007-03-04-01.30.00.000000', 5,773.76),
(87, 'DL', 'LAX', 'SYD', '2007-03-04-02.30.00.000000', '2007-03-04-12.30.00.000000', 10,1277.32),
(88, 'SQ', 'AMS', 'TYO', '2007-03-03-11.00.00.000000', '2007-03-03-22.00.00.000000', 11,1564.61),
(89, 'SQ', 'TYO', 'SYD', '2007-03-04-03.00.00.000000', '2007-03-04-14.00.00.000000', 11,1900.35),
(90, 'KL', 'AMS', 'LAX', '2007-03-03-18.00.00.000000', '2007-03-04-07.00.00.000000', 13,2417.77),
(91, 'KL', 'AMS', 'JFK', '2007-03-03-10.00.00.000000', '2007-03-03-16.00.00.000000', 6,607.50),
(92, 'CO', 'JFK', 'PHX', '2007-03-03-19.00.00.000000', '2007-03-04-01.00.00.000000', 6,796.45),
(93, 'KL', 'AMS', 'LGA', '2007-03-03-10.00.00.000000', '2007-03-03-16.00.00.000000', 6,779.72),
(94, 'CO', 'LGA', 'PHX', '2007-03-03-20.00.00.000000', '2007-03-04-02.00.00.000000', 6,855.14),
(95, 'KL', 'AMS', 'EWR', '2007-03-03-10.00.00.000000', '2007-03-03-17.00.00.000000', 7,731.27),
(96, 'KL', 'EWR', 'PHX', '2007-03-03-19.00.00.000000', '2007-03-04-00.00.00.000000', 5,664.49),
(97, 'KL', 'AMS', 'CAI', '2007-03-03-09.00.00.000000', '2007-03-03-16.00.00.000000', 7,805.83),
(98, 'KL', 'CAI', 'TYO', '2007-03-03-19.00.00.000000', '2007-03-04-00.00.00.000000', 5,925.04),
(99, 'KL', 'AMS', 'JFK', '2007-03-03-15.00.00.000000', '2007-03-03-21.00.00.000000', 6,710.58),
(100, 'KL', 'AMS', 'LGA', '2007-03-03-12.00.00.000000', '2007-03-03-18.00.00.000000', 6,964.39),
(101, 'KL', 'AMS', 'LHR', '2007-03-03-15.00.00.000000', '2007-03-03-16.00.00.000000', 1,140.63),
(102, 'KL', 'AMS', 'LHR', '2007-03-03-18.00.00.000000', '2007-03-03-19.00.00.000000', 1,126.91),
(103, 'KL', 'AMS', 'LHR', '2007-03-03-21.00.00.000000', '2007-03-03-22.00.00.000000', 1,100.78),
(104, 'KL', 'PHX', 'LAX', '2007-03-03-21.00.00.000000', '2007-03-03-23.00.00.000000', 2,398.02),
(105, 'KL', 'LHR', 'AMS', '2007-03-04-09.30.00.000000', '2007-03-04-10.30.00.000000', 1,141.05),
(106, 'KL', 'ORD', 'LHR', '2007-03-04-02.30.00.000000', '2007-03-04-08.30.00.000000', 6,996.23),
(107, 'DL', 'LAX', 'ORD', '2007-03-03-20.30.00.000000', '2007-03-04-01.30.00.000000', 5,519.50),
(108, 'DL', 'SYD', 'LAX', '2007-03-03-09.30.00.000000', '2007-03-03-19.30.00.000000', 10,1033.78),
(109, 'SQ', 'TYO', 'AMS', '2007-03-03-16.00.00.000000', '2007-03-04-03.00.00.000000', 11,1773.28),
(110, 'SQ', 'SYD', 'TYO', '2007-03-03-10.00.00.000000', '2007-03-03-21.00.00.000000', 11,2008.04),
(111, 'KL', 'LAX', 'AMS', '2007-03-03-18.00.00.000000', '2007-03-04-07.00.00.000000', 13,2511.48),
(112, 'KL', 'JFK', 'AMS', '2007-03-03-21.00.00.000000', '2007-03-04-03.00.00.000000', 6,659.27),
(113, 'CO', 'PHX', 'JFK', '2007-03-03-12.00.00.000000', '2007-03-03-18.00.00.000000', 6,876.90),
(114, 'KL', 'LGA', 'AMS', '2007-03-03-20.00.00.000000', '2007-03-04-02.00.00.000000', 6,790.60),
(115, 'CO', 'PHX', 'LGA', '2007-03-03-12.00.00.000000', '2007-03-03-18.00.00.000000', 6,1087.49),

(116, 'KL', 'EWR', 'AMS', '2007-03-03-19.00.00.000000', '2007-03-04-02.00.00.000000', 7,1365.17),
(117, 'KL', 'PHX', 'EWR', '2007-03-03-13.00.00.000000', '2007-03-03-18.00.00.000000', 5,867.16),
(118, 'KL', 'CAI', 'AMS', '2007-03-03-17.00.00.000000', '2007-03-04-00.00.00.000000', 7,768.61),
(119, 'KL', 'TYO', 'CAI', '2007-03-03-10.00.00.000000', '2007-03-03-15.00.00.000000', 5,717.26),
(120, 'KL', 'JFK', 'AMS', '2007-03-03-15.00.00.000000', '2007-03-03-21.00.00.000000', 6,1149.13),
(121, 'KL', 'LGA', 'AMS', '2007-03-03-12.00.00.000000', '2007-03-03-18.00.00.000000', 6,710.36),
(122, 'KL', 'LHR', 'AMS', '2007-03-03-15.00.00.000000', '2007-03-03-16.00.00.000000', 1,120.18),
(123, 'KL', 'LHR', 'AMS', '2007-03-03-18.00.00.000000', '2007-03-03-19.00.00.000000', 1,130.61),
(124, 'KL', 'LHR', 'AMS', '2007-03-03-21.00.00.000000', '2007-03-03-22.00.00.000000', 1,112.82),
(125, 'KL', 'LAX', 'PHX', '2007-03-03-21.00.00.000000', '2007-03-03-23.00.00.000000', 2,386.18),
(126, 'KL', 'AMS', 'LHR', '2007-03-04-11.30.00.000000', '2007-03-04-12.30.00.000000', 1,119.56),
(127, 'KL', 'LHR', 'ORD', '2007-03-04-13.30.00.000000', '2007-03-04-19.30.00.000000', 6,865.89),
(128, 'DL', 'ORD', 'LAX', '2007-03-04-20.30.00.000000', '2007-03-05-01.30.00.000000', 5,947.67),
(129, 'DL', 'LAX', 'SYD', '2007-03-05-02.30.00.000000', '2007-03-05-12.30.00.000000', 10,1494.88),
(130, 'SQ', 'AMS', 'TYO', '2007-03-04-11.00.00.000000', '2007-03-04-22.00.00.000000', 11,1877.01),
(131, 'SQ', 'TYO', 'SYD', '2007-03-05-03.00.00.000000', '2007-03-05-14.00.00.000000', 11,2096.97),
(132, 'KL', 'AMS', 'LAX', '2007-03-04-18.00.00.000000', '2007-03-05-07.00.00.000000', 13,2221.90),
(133, 'KL', 'AMS', 'JFK', '2007-03-04-10.00.00.000000', '2007-03-04-16.00.00.000000', 6,1125.18),
(134, 'CO', 'JFK', 'PHX', '2007-03-04-19.00.00.000000', '2007-03-05-01.00.00.000000', 6,1082.88),
(135, 'KL', 'AMS', 'LGA', '2007-03-04-10.00.00.000000', '2007-03-04-16.00.00.000000', 6,1066.18),
(136, 'CO', 'LGA', 'PHX', '2007-03-04-20.00.00.000000', '2007-03-05-02.00.00.000000', 6,1167.04),
(137, 'KL', 'AMS', 'EWR', '2007-03-04-10.00.00.000000', '2007-03-04-17.00.00.000000', 7,1110.89),
(138, 'KL', 'EWR', 'PHX', '2007-03-04-19.00.00.000000', '2007-03-05-00.00.00.000000', 5,761.45),
(139, 'KL', 'AMS', 'CAI', '2007-03-04-09.00.00.000000', '2007-03-04-16.00.00.000000', 7,1210.46),
(140, 'KL', 'CAI', 'TYO', '2007-03-04-19.00.00.000000', '2007-03-05-00.00.00.000000', 5,885.31),
(141, 'KL', 'AMS', 'JFK', '2007-03-04-15.00.00.000000', '2007-03-04-21.00.00.000000', 6,634.47),
(142, 'KL', 'AMS', 'LGA', '2007-03-04-12.00.00.000000', '2007-03-04-18.00.00.000000', 6,772.56),
(143, 'KL', 'AMS', 'LHR', '2007-03-04-15.00.00.000000', '2007-03-04-16.00.00.000000', 1,185.38),
(144, 'KL', 'AMS', 'LHR', '2007-03-04-18.00.00.000000', '2007-03-04-19.00.00.000000', 1,129.89),
(145, 'KL', 'AMS', 'LHR', '2007-03-04-21.00.00.000000', '2007-03-04-22.00.00.000000', 1,129.15),
(146, 'KL', 'PHX', 'LAX', '2007-03-04-21.00.00.000000', '2007-03-04-23.00.00.000000', 2,310.37),
(147, 'KL', 'LHR', 'AMS', '2007-03-05-09.30.00.000000', '2007-03-05-10.30.00.000000', 1,189.00),
(148, 'KL', 'ORD', 'LHR', '2007-03-05-02.30.00.000000', '2007-03-05-08.30.00.000000', 6,910.59),
(149, 'DL', 'LAX', 'ORD', '2007-03-04-20.30.00.000000', '2007-03-05-01.30.00.000000', 5,604.89),
(150, 'DL', 'SYD', 'LAX', '2007-03-04-09.30.00.000000', '2007-03-04-19.30.00.000000', 10,1404.09),
(151, 'SQ', 'TYO', 'AMS', '2007-03-04-16.00.00.000000', '2007-03-05-03.00.00.000000', 11,1674.62),
(152, 'SQ', 'SYD', 'TYO', '2007-03-04-10.00.00.000000', '2007-03-04-21.00.00.000000', 11,1126.65),
(153, 'KL', 'LAX', 'AMS', '2007-03-04-18.00.00.000000', '2007-03-05-07.00.00.000000', 13,2037.18),
(154, 'KL', 'JFK', 'AMS', '2007-03-04-21.00.00.000000', '2007-03-05-03.00.00.000000', 6,862.45),
(155, 'CO', 'PHX', 'JFK', '2007-03-04-12.00.00.000000', '2007-03-04-18.00.00.000000', 6,1138.60),
(156, 'KL', 'LGA', 'AMS', '2007-03-04-20.00.00.000000', '2007-03-05-02.00.00.000000', 6,659.65),
(157, 'CO', 'PHX', 'LGA', '2007-03-04-12.00.00.000000', '2007-03-04-18.00.00.000000', 6,1114.57),
(158, 'KL', 'EWR', 'AMS', '2007-03-04-19.00.00.000000', '2007-03-05-02.00.00.000000', 7,1162.05),
(159, 'KL', 'PHX', 'EWR', '2007-03-04-13.00.00.000000', '2007-03-04-18.00.00.000000', 5,509.59),
(160, 'KL', 'CAI', 'AMS', '2007-03-04-17.00.00.000000', '2007-03-05-00.00.00.000000', 7,804.80),
(161, 'KL', 'TYO', 'CAI', '2007-03-04-10.00.00.000000', '2007-03-04-15.00.00.000000', 5,556.16),
(162, 'KL', 'JFK', 'AMS', '2007-03-04-15.00.00.000000', '2007-03-04-21.00.00.000000', 6,1036.62),
(163, 'KL', 'LGA', 'AMS', '2007-03-04-12.00.00.000000', '2007-03-04-18.00.00.000000', 6,704.51),
(164, 'KL', 'LHR', 'AMS', '2007-03-04-15.00.00.000000', '2007-03-04-16.00.00.000000', 1,105.03),
(165, 'KL', 'LHR', 'AMS', '2007-03-04-18.00.00.000000', '2007-03-04-19.00.00.000000', 1,199.53),
(166, 'KL', 'LHR', 'AMS', '2007-03-04-21.00.00.000000', '2007-03-04-22.00.00.000000', 1,127.97),
(167, 'KL', 'LAX', 'PHX', '2007-03-04-21.00.00.000000', '2007-03-04-23.00.00.000000', 2,210.11),
(168, 'KL', 'AMS', 'LHR', '2007-03-05-11.30.00.000000', '2007-03-05-12.30.00.000000', 1,141.96),
(169, 'KL', 'LHR', 'ORD', '2007-03-05-13.30.00.000000', '2007-03-05-19.30.00.000000', 6,662.89),
(170, 'DL', 'ORD', 'LAX', '2007-03-05-20.30.00.000000', '2007-03-06-01.30.00.000000', 5,829.95),
(171, 'DL', 'LAX', 'SYD', '2007-03-06-02.30.00.000000', '2007-03-06-12.30.00.000000', 10,1699.51),
(172, 'SQ', 'AMS', 'TYO', '2007-03-05-11.00.00.000000', '2007-03-05-22.00.00.000000', 11,1492.33),
(173, 'SQ', 'TYO', 'SYD', '2007-03-06-03.00.00.000000', '2007-03-06-14.00.00.000000', 11,2075.38),
(174, 'KL', 'AMS', 'LAX', '2007-03-05-18.00.00.000000', '2007-03-06-07.00.00.000000', 13,2344.26),
(175, 'KL', 'AMS', 'JFK', '2007-03-05-10.00.00.000000', '2007-03-05-16.00.00.000000', 6,1199.89),
(176, 'CO', 'JFK', 'PHX', '2007-03-05-19.00.00.000000', '2007-03-06-01.00.00.000000', 6,1003.55),
(177, 'KL', 'AMS', 'LGA', '2007-03-05-10.00.00.000000', '2007-03-05-16.00.00.000000', 6,638.34),
(178, 'CO', 'LGA', 'PHX', '2007-03-05-20.00.00.000000', '2007-03-06-02.00.00.000000', 6,942.23),
(179, 'KL', 'AMS', 'EWR', '2007-03-05-10.00.00.000000', '2007-03-05-17.00.00.000000', 7,1246.48),
(180, 'KL', 'EWR', 'PHX', '2007-03-05-19.00.00.000000', '2007-03-06-00.00.00.000000', 5,573.90),
(181, 'KL', 'AMS', 'CAI', '2007-03-05-09.00.00.000000', '2007-03-05-16.00.00.000000', 7,1393.86),
(182, 'KL', 'CAI', 'TYO', '2007-03-05-19.00.00.000000', '2007-03-06-00.00.00.000000', 5,654.82),

```
(183,'KL','AMS','JFK','2007-03-05-15.00.00.000000','2007-03-05-21.00.00.000000',6,755.35),
(184,'KL','AMS','LGA','2007-03-05-12.00.00.000000','2007-03-05-18.00.00.000000',6,1107.25),
(185,'KL','AMS','LHR','2007-03-05-15.00.00.000000','2007-03-05-16.00.00.000000',1,183.91),
(186,'KL','AMS','LHR','2007-03-05-18.00.00.000000','2007-03-05-19.00.00.000000',1,193.00),
(187,'KL','AMS','LHR','2007-03-05-21.00.00.000000','2007-03-05-22.00.00.000000',1,194.66),
(188,'KL','PHX','LAX','2007-03-05-21.00.00.000000','2007-03-05-23.00.00.000000',2,202.55),
(189,'KL','LHR','AMS','2007-03-06-09.30.00.000000','2007-03-06-10.30.00.000000',1,101.44),
(190,'KL','ORD','LHR','2007-03-06-02.30.00.000000','2007-03-06-08.30.00.000000',6,758.48),
(191,'DL','LAX','ORD','2007-03-05-20.30.00.000000','2007-03-06-01.30.00.000000',5,843.05),
(192,'DL','SYD','LAX','2007-03-05-09.30.00.000000','2007-03-05-19.30.00.000000',10,1602.19),
(193,'SQ','TYO','AMS','2007-03-05-16.00.00.000000','2007-03-06-03.00.00.000000',11,1895.91),
(194,'SQ','SYD','TYO','2007-03-05-10.00.00.000000','2007-03-05-21.00.00.000000',11,1104.59),
(195,'KL','LAX','AMS','2007-03-05-18.00.00.000000','2007-03-06-07.00.00.000000',13,2558.02),
(196,'KL','JFK','AMS','2007-03-05-21.00.00.000000','2007-03-06-03.00.00.000000',6,1165.00),
(197,'CO','PHX','JFK','2007-03-05-12.00.00.000000','2007-03-05-18.00.00.000000',6,650.41),
(198,'KL','LGA','AMS','2007-03-05-20.00.00.000000','2007-03-06-02.00.00.000000',6,1194.61),
(199,'CO','PHX','LGA','2007-03-05-12.00.00.000000','2007-03-05-18.00.00.000000',6,712.43),
(200,'KL','EWR','AMS','2007-03-05-19.00.00.000000','2007-03-06-02.00.00.000000',7,1111.57),
(201,'KL','PHX','EWR','2007-03-05-13.00.00.000000','2007-03-05-18.00.00.000000',5,845.98),
(202,'KL','CAI','AMS','2007-03-05-17.00.00.000000','2007-03-06-00.00.00.000000',7,796.60),
(203,'KL','TYO','CAI','2007-03-05-10.00.00.000000','2007-03-05-15.00.00.000000',5,999.95),
(204,'KL','JFK','AMS','2007-03-05-15.00.00.000000','2007-03-05-21.00.00.000000',6,1135.12),
(205,'KL','LGA','AMS','2007-03-05-12.00.00.000000','2007-03-05-18.00.00.000000',6,630.12),
(206,'KL','LHR','AMS','2007-03-05-15.00.00.000000','2007-03-05-16.00.00.000000',1,177.12),
(207,'KL','LHR','AMS','2007-03-05-18.00.00.000000','2007-03-05-19.00.00.000000',1,142.37),
(208,'KL','LHR','AMS','2007-03-05-21.00.00.000000','2007-03-05-22.00.00.000000',1,130.79),
(209,'KL','LAX','PHX','2007-03-05-21.00.00.000000','2007-03-05-23.00.00.000000',2,352.84)
```

Voorbeeld 15.22:

```
WITH VLUCHTPLAN(VLUCHTNR, PLAN_VLIEGVELDEN, PLAN_VLUCHTEN, START_VLIEGVELD,
               EIND_VLIEGVELD, STARTTIJD, EINDTIJD, VERTREK_VLIEGVELD,
               AANKOMST_VLIEGVELD, VERTREKTIJD, AANKOMSTTIJD) AS
(SELECT VLUCHTNR, CAST(VERTREK_VLIEGVELD || '->' ||
                    AANKOMST_VLIEGVELD AS VARCHAR(100)),
        CAST(RTRIM(CHAR(VLUCHTNR)) AS VARCHAR(100)),
        VERTREK_VLIEGVELD, AANKOMST_VLIEGVELD, VERTREKTIJD, AANKOMSTTIJD,
        VERTREK_VLIEGVELD, AANKOMST_VLIEGVELD, VERTREKTIJD, AANKOMSTTIJD
FROM VLUCHTEN
WHERE VERTREK_VLIEGVELD='AMS'
AND CAST(VERTREKTIJD AS DATE) = '2007-03-01'
UNION ALL
SELECT P.VLUCHTNR, P.PLAN_VLIEGVELDEN || '->' || F.AANKOMST_VLIEGVELD,
        P.PLAN_VLUCHTEN || '->' || RTRIM(CHAR(F.VLUCHTNR)),
        P.START_VLIEGVELD, F.AANKOMST_VLIEGVELD, P.STARTTIJD, F.AANKOMSTTIJD,
        P.VERTREK_VLIEGVELD, P.AANKOMST_VLIEGVELD, P.VERTREKTIJD, P.AANKOMSTTIJD
FROM VLUCHTPLAN AS P, VLUCHTEN AS F
WHERE P.AANKOMST_VLIEGVELD = F.VERTREK_VLIEGVELD
AND P.AANKOMSTTIJD < F.VERTREKTIJD
AND F.VERTREK_VLIEGVELD <> 'PHX'
AND LOCATE(F.AANKOMST_VLIEGVELD, P.PLAN_VLIEGVELDEN) = 0)
SELECT PLAN_VLIEGVELDEN, PLAN_VLUCHTEN, START_VLIEGVELD, EIND_VLIEGVELD,
        STARTTIJD, EINDTIJD
FROM VLUCHTPLAN
WHERE EIND_VLIEGVELD = 'PHX'
```

Voorbeeld 15.23:

```
WITH VLUCHTPLAN(VLUCHTNR, PLAN_VLIEGVELDEN, PLAN_VLUCHTEN, START_VLIEGVELD,
               EIND_VLIEGVELD, STARTTIJD, EINDTIJD, VERTREK_VLIEGVELD,
               AANKOMST_VLIEGVELD, VERTREKTIJD, AANKOMSTTIJD, STOPS) AS
(SELECT VLUCHTNR,
        CAST(VERTREK_VLIEGVELD || '->' || AANKOMST_VLIEGVELD AS VARCHAR(100)),
        CAST(RTRIM(CHAR(VLUCHTNR)) AS VARCHAR(100)),
        VERTREK_VLIEGVELD, AANKOMST_VLIEGVELD, VERTREKTIJD, AANKOMSTTIJD,
        VERTREK_VLIEGVELD, AANKOMST_VLIEGVELD, VERTREKTIJD, AANKOMSTTIJD, 0
FROM VLUCHTEN
```

```

WHERE   VERTREK_VLIEGVELD='AMS'
AND     CAST(VERTREKTIJD AS DATE) = '2007-03-01'
UNION ALL
SELECT  P.VLUCHTNR, P.PLAN_VLIEGVELDEN || '->' || F.AANKOMST_VLIEGVELD,
        P.PLAN_VLUCHTEN || '->' || RTRIM(CHAR(F.VLUCHTNR)),
        P.START_VLIEGVELD, F.AANKOMST_VLIEGVELD, P.STARTTIJD, F.AANKOMSTTIJD,
        P.VERTREK_VLIEGVELD, P.AANKOMST_VLIEGVELD,
        P.VERTREKTIJD, P.AANKOMSTTIJD, STOPS+1
FROM    VLUCHTPLAN AS P, VLUCHTEN AS F
WHERE   P.AANKOMST_VLIEGVELD = F.VERTREK_VLIEGVELD
AND     P.AANKOMSTTIJD < F.VERTREKTIJD
AND     F.VERTREK_VLIEGVELD <> 'PHX'
AND     LOCATE(F.AANKOMST_VLIEGVELD, P.PLAN_VLIEGVELDEN) = 0
AND     STOPS < 1
AND     P.AANKOMSTTIJD + 4 HOURS > F.VERTREKTIJD)
SELECT  PLAN_VLIEGVELDEN, PLAN_VLUCHTEN, START_VLIEGVELD, EIND_VLIEGVELD,
        STARTTIJD, EINDTIJD
FROM    VLUCHTPLAN
WHERE   EIND_VLIEGVELD = 'PHX'

```

4.13 SQL-instructies voor hoofdstuk 16

Voorbeeld 16.1:

```

INSERT INTO TEAMS (TEAMNR, SPELERSNR, DIVISIE)
VALUES (3, 100, 'derde')
;
INSERT INTO TEAMS
VALUES (3, 100, 'derde')
;
INSERT INTO TEAMS (SPELERSNR, DIVISIE, TEAMNR)
VALUES (100, 'derde', 3)
;
INSERT INTO TEAMS (TEAMNR, DIVISIE)
VALUES (3, 'derde')

```

Voorbeeld 16.2:

```

INSERT INTO SPELERS
(SPELERSNR, NAAM, VOORLETTERS, GESLACHT, JAARTOE, STRAAT, PLAATS)
VALUES (611, 'Jones', 'GG', 'M', 1977, 'Herenlaan', 'Den Haag')
;
INSERT INTO SPELERS
(SPELERSNR, NAAM, VOORLETTERS, GEB_DATUM, GESLACHT, JAARTOE,
 STRAAT, HUISNR, POSTCODE, PLAATS, TELEFOON, BONDSNR)
VALUES (611, 'Jones', 'GG', NULL, 'M', 1977,
 'Herenlaan', NULL, NULL, 'Den Haag', NULL, NULL)

```

Voorbeeld 16.3:

```

INSERT INTO TEAMS (TEAMNR, SPELERSNR, DIVISIE)
VALUES (6, 100, 'derde'),
       (7, 27, 'vierde'),
       (8, 39, 'vierde'),
       (9, 112, 'zesde')

```

Voorbeeld 16.4:

```

CREATE TABLE TOTALEN (
  AANTALSPELERS  INTEGER NOT NULL,
  SOMBOETES      DECIMAL(9,2) NOT NULL)
;
INSERT INTO TOTALEN (AANTALSPELERS, SOMBOETES)
VALUES ((SELECT COUNT(*) FROM SPELERS),

```

```
(SELECT SUM(BEDRAG) FROM BOETES))
```

Voorbeeld 16.5:

```
CREATE TABLE RECREANTEN (
  SPELERSNR    SMALLINT NOT NULL,
  NAAM         CHAR(15) NOT NULL,
  PLAATS       CHAR(10) NOT NULL,
  TELEFOON     CHAR(13),
  PRIMARY KEY (SPELERSNR))
;
INSERT INTO RECREANTEN
(SPELERSNR, NAAM, PLAATS, TELEFOON)
SELECT SPELERSNR, NAAM, PLAATS, TELEFOON
FROM SPELERS
WHERE BONDSNR IS NULL
;
INSERT INTO RECREANTEN
SELECT SPELERSNR, NAAM, PLAATS, TELEFOON
FROM SPELERS
WHERE BONDSNR IS NULL
;
INSERT INTO RECREANTEN
(PLAATS, TELEFOON, NAAM, SPELERSNR)
SELECT PLAATS, TELEFOON, NAAM, SPELERSNR
FROM SPELERS
WHERE BONDSNR IS NULL
```

Voorbeeld 16.6:

```
INSERT INTO RECREANTEN
(SPELERSNR, NAAM, PLAATS, TELEFOON)
SELECT SPELERSNR + 1000, NAAM, PLAATS, TELEFOON
FROM RECREANTEN
```

Voorbeeld 16.7:

```
INSERT INTO BOETES
SELECT BETALINGSNR + 100, SPELERSNR, DATUM, BEDRAG
FROM BOETES
WHERE BEDRAG >
      (SELECT AVG(BEDRAG)
       FROM BOETES)
```

Voorbeeld 16.8:

```
UPDATE SPELERS
SET BONDSNR = '2000'
WHERE SPELERSNR = 95
;
UPDATE SPELERS AS S
SET BONDSNR = '2000'
WHERE S.SPELERSNR = 95
```

Voorbeeld 16.9:

```
UPDATE BOETES
SET BEDRAG = BEDRAG * 1.05
```

Voorbeeld 16.10:

```
UPDATE WEDSTRIJDEN
SET GEWONNEN = 0
WHERE SPELERSNR IN
      (SELECT SPELERSNR
       FROM SPELERS
       WHERE PLAATS = 'Den Haag')
```

Voorbeeld 16.11:

```

UPDATE SPELERS
SET   STRAAT = 'Pisuissestraat',
      HUISNR = '83',
      PLAATS = 'Rijswijk',
      POSTCODE = '1234UU',
      TELEFOON = NULL
WHERE NAAM = 'Permentier'
;
UPDATE SPELERS
SET   STRAAT = PLAATS,
      PLAATS = STRAAT
WHERE SPELERSNR = 44
;
UPDATE SPELERS
SET   STRAAT = PLAATS
WHERE SPELERSNR = 44
;
UPDATE SPELERS
SET   PLAATS = STRAAT
WHERE SPELERSNR = 44
;
UPDATE SPELERS
SET   STRAAT = PLAATS,
      PLAATS = STRAAT
WHERE SPELERSNR = 44

```

Voorbeeld 16.12:

```

CREATE TABLE SPELERS_GEGEVENS (
  SPELERSNR  INTEGER NOT NULL PRIMARY KEY,
  AANTAL_WED INTEGER,
  SOM_BOETES DECIMAL(7,2))
;
INSERT INTO SPELERS_GEGEVENS(SPELERSNR)
SELECT SPELERSNR FROM SPELERS
;
UPDATE SPELERS_GEGEVENS AS SG
SET   AANTAL_WED =
      (SELECT COUNT(*)
       FROM WEDSTRIJDEN AS W
       WHERE W.SPELERSNR = SG.SPELERSNR),
      SOM_BOETES =
      (SELECT SUM(BEDRAG)
       FROM BOETES AS B
       WHERE B.SPELERSNR = SG.SPELERSNR)

```

Voorbeeld 16.13:

```

UPDATE BOETES
SET   BEDRAG = BEDRAG + (SELECT AVG(BEDRAG)
                        FROM BOETES)

```

Voorbeeld 16.14:

```

DELETE
FROM BOETES
WHERE SPELERSNR = 44
;
DELETE
FROM BOETES AS B
WHERE B.SPELERSNR = 44

```

Voorbeeld 16.15:


```
DELETE
FROM   WEDSTRIJDEN
WHERE  (SELECT AVG(GEWONNEN)
        FROM   WEDSTRIJDEN
        WHERE  SPELERSNR = 83)
```

4.14 SQL-instructies voor hoofdstuk 17

Voorbeeld 17.1:

```
CREATE TABLE XML_WEDSTRIJDEN (
  WEDSTRIJDNR   INTEGER NOT NULL PRIMARY KEY,
  WEDSTRIJD_INFO TEXT)
```

Voorbeeld 17.2:

```
INSERT INTO XML_WEDSTRIJDEN VALUES (1,
'<wedstrijd nummer="1">Wedstrijdgegevens van 1
  <team>Teamgegevens van 1
    <nummer>1</nummer>
    <divisie>ere</divisie>
  </team>
  <speler>Spelersgegevens van 6
    <nummer>6</nummer>
    <naam>De naam van 6
      <achternaam>Permentier</achternaam>
      <voorletters>R</voorletters>
    </naam>
    <adres>Het adres van 6
      <straat>Hazensteinln</straat>
      <huisnr>80</huisnr>
      <postcode>1234KK</postcode>
      <plaats>Den Haag</plaats>
    </adres>
  </speler>
  <sets>Gegevens over sets van 1
    <gewonnen>3</gewonnen>
    <verloren>1</verloren>
  </sets>
</wedstrijd>')
```

```
;
```

```
INSERT INTO XML_WEDSTRIJDEN VALUES (9,
'<wedstrijd nummer="9">Wedstrijdgegevens van 9
  <team>Teamgegevens van 2
    <nummer>2</nummer>
    <divisie>tweede</divisie>
  </team>
  <speler>Spelersgegevens van 27
    <nummer>27</nummer>
    <naam>De naam van 27
      <achternaam>Cools</achternaam>
      <voorletters>DD</voorletters>
    </naam>
    <adres>Het adres van 27
      <straat>Liespad</straat>
      <huisnr>804</huisnr>
      <postcode>8457DK</postcode>
      <plaats>Zoetermeer</plaats>
    </adres>
  <telefoons>Telefoonnummers van 27
    <nummer>1234567</nummer>
    <nummer>3468346</nummer>
    <nummer>6236984</nummer>
```

```

        <nummer>6587437</nummer>
    </telefoons>
</speler>
<sets>Gegevens over sets van 9
    <gewonnen>3</gewonnen>
    <verloren>2</verloren>
</sets>
</wedstrijd>')
;
INSERT INTO XML_WEDSTRIJDEN VALUES (12,
'<wedstrijd nummer="12">Wedstrijdgegevens van 12
    <team>Teamgegevens van 2
        <nummer>2</nummer>
        <divisie>tweede</divisie>
    </team>
    <speler>Spelersgegevens van 8
        <nummer>8</nummer>
        <naam>De naam van 8
            <achternaam>Niewenburg</achternaam>
            <voorletters>B</voorletters>
        </naam>
        <adres>Het eerste adres van 8
            <straat>Spoorlaan</straat>
            <huisnr>4</huisnr>
            <postcode>6584R0</postcode>
            <plaats>Rijswijk</plaats>
        </adres>
        <adres>Het tweede adres van 8
            <straat>Tramlaan</straat>
            <huisnr>14</huisnr>
            <postcode>2728YG</postcode>
            <plaats>Voorburg</plaats>
        </adres>
    </speler>
    <sets>Gegevens over sets van 12
        <gewonnen>1</gewonnen>
        <verloren>3</verloren>
    </sets>
</wedstrijd>')
```

Voorbeeld 17.3:

```

SELECT  WEDSTRIJDNR,
        EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/team/divisie') AS DIVISIE
FROM    XML_WEDSTRIJDEN
```

Voorbeeld 17.4:

```

SELECT  WEDSTRIJDNR,
        EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/speler/naam/achternaam')
        AS SPELER
FROM    XML_WEDSTRIJDEN
WHERE   EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/sets/gewonnen') = 3
```

Voorbeeld 17.5:

```

SELECT  EXTRACTVALUE('
    <team>
        <nummer>2</nummer>
        <divisie>tweede</divisie>
    </team>'
, '/team') = '' AS TEAM
```

Voorbeeld 17.6:

```

SELECT  WEDSTRIJDNR,
        EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/speler/telefoons/nummer')
```

```

        AS TELEFOONS
FROM    XML_WEDSTRIJDEN
WHERE   WEDSTRIJDNR = 9

```

Voorbeeld 17.7:

```

SELECT  WEDSTRIJDNR,
        EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/speler') AS SPELERS
FROM    XML_WEDSTRIJDEN
;
SELECT  REPLACE(EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/speler'), ' ', '#')
        AS SPELERSGEGEVENS
FROM    XML_WEDSTRIJDEN

```

Voorbeeld 17.8:

```

SELECT  WEDSTRIJDNR,
        EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/*/nummer') AS NUMMERS
FROM    XML_WEDSTRIJDEN

```

Voorbeeld 17.9:

```

SELECT  WEDSTRIJDNR,
        EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd//nummer') AS NUMMERS
FROM    XML_WEDSTRIJDEN
;
SELECT  WEDSTRIJDNR,
        EXTRACTVALUE(WEDSTRIJD_INFO, '//nummer') AS NUMMERS
FROM    XML_WEDSTRIJDEN

```

Voorbeeld 17.10:

```

SELECT  EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd//*') AS ALLES
FROM    XML_WEDSTRIJDEN
WHERE   WEDSTRIJDNR = 1

```

Voorbeeld 17.11:

```

SELECT  WEDSTRIJDNR,
        EXTRACTVALUE(WEDSTRIJD_INFO, '//plaats|//gewonnen') AS PLAATS_GEWONNEN
FROM    XML_WEDSTRIJDEN
;
SELECT  WEDSTRIJDNR,
        CONCAT(EXTRACTVALUE(WEDSTRIJD_INFO, '//plaats'),
              ' ',
              EXTRACTVALUE(WEDSTRIJD_INFO, '//gewonnen'))
        AS PLAATS_GEWONNEN
FROM    XML_WEDSTRIJDEN

```

Voorbeeld 17.12:

```

SELECT  WEDSTRIJDNR,
        EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/@nummer') AS XML_WEDSTRIJDNR
FROM    XML_WEDSTRIJDEN

```

Voorbeeld 17.13:

```

SELECT  WEDSTRIJDNR,
        EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/sets/gewonnen+10')
        AS GEWONNEN_PLUS_10
FROM    XML_WEDSTRIJDEN

```

Voorbeeld 17.14:

```

SELECT  WEDSTRIJDNR,
        EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/speler/adres[1]/plaats')
        AS PLAATS

```

```
FROM XML_WEDSTRIJDEN
```

Voorbeeld 17.15:

```
SELECT WEDSTRIJDNR,
       EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/speler/*[1]') AS EEN_WAARDE
FROM XML_WEDSTRIJDEN
```

Voorbeeld 17.16:

```
SELECT WEDSTRIJDNR,
       EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/speler/telefoons/nummer[last()]')
       AS LAATSTE
FROM XML_WEDSTRIJDEN
```

Voorbeeld 17.17:

```
SELECT WEDSTRIJDNR,
       EXTRACTVALUE(WEDSTRIJD_INFO,
                    '/child::wedstrijd/child::team/child::nummer ') AS NUMMERS
FROM XML_WEDSTRIJDEN
```

Voorbeeld 17.18:

```
SELECT EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/speler/adres/descendant::* ')
       AS ADRESGEGEVENS
FROM XML_WEDSTRIJDEN
```

Voorbeeld 17.19:

```
SELECT EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/speler/descendant::* ')
       AS SPELERSGEGEVENS
FROM XML_WEDSTRIJDEN
```

Voorbeeld 17.20:

```
SELECT EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/speler/descendant::* ')
       AS SPELERSGEGEVENS
FROM XML_WEDSTRIJDEN
```

Voorbeeld 17.21:

```
SELECT WEDSTRIJDNR,
       EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/speler[nummer=8]')
       AS SPELER8
FROM XML_WEDSTRIJDEN
```

Voorbeeld 17.22:

```
SELECT WEDSTRIJDNR,
       EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/speler') AS SPELER8
FROM XML_WEDSTRIJDEN
WHERE EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/speler[nummer=8]') <> ''
```

Voorbeeld 17.23:

```
SELECT WEDSTRIJDNR,
       EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/sets') AS DRIE_EN_EEN
FROM XML_WEDSTRIJDEN
WHERE EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/sets[gewonnen=3 and verloren=1]') <> ''
```

Voorbeeld 17.24:

```
UPDATE XML_WEDSTRIJDEN
SET WEDSTRIJD_INFO =
    UPDATEXML(WEDSTRIJD_INFO, '/wedstrijd/sets/verloren',
              '<verloren>2</verloren>')
```

```

WHERE   WEDSTRIJDNR = 1
;
SELECT  EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/sets/verloren') AS VERLOREN
FROM    XML_WEDSTRIJDEN
WHERE   WEDSTRIJDNR = 1

```

Voorbeeld 17.25:

```

UPDATE  XML_WEDSTRIJDEN
SET     WEDSTRIJD_INFO =
        UPDATEXML(WEDSTRIJD_INFO,
        '/wedstrijd/speler/adres',
        '<adres>Het nieuwe adres van 8
        <straat>Mooilaan</straat>
        <huisnr>30</huisnr>
        <postcode>5383GH</postcode>
        <plaats>Voorburg</plaats>
        </adres>')
WHERE   WEDSTRIJDNR = 1
;
SELECT  EXTRACTVALUE(WEDSTRIJD_INFO, '/wedstrijd/speler/adres/*') AS NIEUW_ADRES
FROM    XML_WEDSTRIJDEN
WHERE   WEDSTRIJDNR = 1

```

4.15 SQL-instructies voor hoofdstuk 18

Voorbeeld 18.1:

```

CREATE TABLE SPELERS (
  SPELERSNR   INTEGER NOT NULL PRIMARY KEY,
  NAAM        CHAR(15) NOT NULL,
  VOORLETTERS CHAR(3) NOT NULL,
  GEB_DATUM   DATE,
  GESLACHT   CHAR(1) NOT NULL,
  JAARTOE     SMALLINT NOT NULL,
  STRAAT      VARCHAR(30) NOT NULL,
  HUISNR      CHAR(4),
  POSTCODE    CHAR(6),
  PLAATS      VARCHAR(30) NOT NULL,
  TELEFOON    CHAR(13),
  BONDSNR     CHAR(4))

```

Voorbeeld 18.2:

```

CREATE TABLE TEST.BOETES (
  BETALINGSNR INTEGER NOT NULL PRIMARY KEY,
  SPELERSNR   INTEGER NOT NULL,
  DATUM       DATE NOT NULL,
  BEDRAG      DECIMAL(7,2) NOT NULL)

```

Voorbeeld 18.3:

```

CREATE TABLE MEETGEGEVENS (
  NR           INTEGER,
  MEETWAARDE  FLOAT(1))
;
INSERT INTO MEETGEGEVENS VALUES
(1, 99.99),
(2, 99999.99),
(3, 99999999.99),
(4, 9999999999.99),
(5, 999999999999.99),
(6, 0.999999),
(7, 0.999999),

```

```
(8, 99999999.9999),
(9, (1.0/3))
;
SELECT * FROM MEETGEGEVENS
```

Voorbeeld 18.4:

```
CREATE TEMPORARY TABLE SOMBOETES (
  TOTAAL DECIMAL(10,2))
;
INSERT INTO SOMBOETES
SELECT SUM(BEDRAG)
FROM BOETES
```

Voorbeeld 18.5:

```
CREATE TABLE TESTTABEL (
  KOL1 INTEGER)
;
INSERT INTO TESTTABEL VALUES (1)
;
CREATE TEMPORARY TABLE TESTTABEL (
  KOL1 INTEGER,
  KOL2 INTEGER)
;
INSERT INTO TESTTABEL VALUES (2, 3)
;
SELECT * FROM TESTTABEL
;
SELECT * FROM MAIN.TESTTABEL
```

Voorbeeld 18.6:

```
CREATE TABLE TEAMS_KOPIE LIKE TEAMS
```

Voorbeeld 18.7:

```
CREATE TABLE TEAMS_KOPIE1 AS
(SELECT *
FROM TEAMS)
```

Voorbeeld 18.8:

```
CREATE TABLE TEAMS_KOPIE2 AS
(SELECT TEAMNR AS TNR, SPELERSNR AS SNR, DIVISIE
FROM TEAMS)
;
SELECT *
FROM TEAMS_KOPIE2
```

Voorbeeld 18.9:

```
CREATE TABLE TEAMS_KOPIE3 AS
(SELECT TEAMNR, SPELERSNR
FROM TEAMS
WHERE SPELERSNR = 27)
```

Voorbeeld 18.10:

```
CREATE TEMPORARY TABLE TEAMS AS
(SELECT *
FROM TEAMS)
```

Voorbeeld 18.11:

```
CREATE TABLE TEAMS_KOPIE4 (
  TEAMNR INTEGER NOT NULL PRIMARY KEY,
```

```

    SPELERSNR  INTEGER NULL,
    DIVISIE    CHAR(10) NOT NULL) AS
(SELECT *
FROM TEAMS)
;
CREATE TABLE TEAMS_KOPIE4 (
    SPELERSNR  INTEGER NULL,
    DIVISIE    CHAR(10) NOT NULL) AS
(SELECT *
FROM TEAMS)

```

Voorbeeld 18.12:

```

CREATE TABLE TEAMS_KOPIE5 (
    SPELERSNR  INTEGER NULL,
    COMMENTAAR VARCHAR(100)) AS
(SELECT *
FROM TEAMS)
;
SELECT * FROM TEAMS_KOPIE5

```

Voorbeeld 18.13:

```

CREATE TABLE BOETES (
    BETALINGSNR  INTEGER NOT NULL PRIMARY KEY,
    SPELERSNR    INTEGER NOT NULL,
    DATUM        DATE NOT NULL DEFAULT '2007-01-01',
    BEDRAG       DECIMAL(7,2) NOT NULL DEFAULT 50.00)
;
INSERT INTO BOETES
    (BETALINGSNR, SPELERSNR)
VALUES (15, 27)

```

Voorbeeld 18.14:

```

CREATE TABLE BOETES (
    BETALINGSNR  INTEGER NOT NULL PRIMARY KEY,
    SPELERSNR    INTEGER NOT NULL,
    DATUM        DATE NOT NULL DEFAULT CURRENT_DATE,
    BEDRAG       DECIMAL(7,2) NOT NULL DEFAULT 50.00)
;
INSERT INTO BOETES
    (BETALINGSNR, SPELERSNR, DATUM, BEDRAG)
VALUES (15, 27, DEFAULT, DEFAULT)

```

Voorbeeld 18.15:

```

UPDATE BOETES
SET    BEDRAG = DEFAULT

```

Voorbeeld 18.16:

```

UPDATE BOETES
SET    BEDRAG = YEAR(DEFAULT(DATUM))*10

```

Voorbeeld 18.17:

```

CREATE TABLE BOETES (
    BETALINGSNR  INTEGER NOT NULL PRIMARY KEY
                    COMMENT 'Primaire sleutel van de tabel',
    SPELERSNR    INTEGER NOT NULL
                    COMMENT 'Speler voor wie de boete betaald is',
    DATUM        DATE NOT NULL
                    COMMENT 'Datum waarop de boete betaald is',
    BEDRAG       DECIMAL(7,2) NOT NULL COMMENT
                    'Hoogte van het boetebedrag in euro''s')
;

```

```
SELECT COLUMN_NAME, COLUMN_COMMENT
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'BOETES'
```

Voorbeeld 18.18:

```
CREATE TABLE WEDSTRIJDEN (
  WEDSTRIJDNR INTEGER NOT NULL PRIMARY KEY,
  TEAMNR      INTEGER NOT NULL,
  SPELERSNR  INTEGER NOT NULL,
  GEWONNEN   SMALLINT NOT NULL,
  VERLOREN   SMALLINT NOT NULL,
  SALDO      AS ABS(GEWONNEN - VERLOREN))
```

Voorbeeld 18.19:

```
SELECT WEDSTRIJDNR, SALDO
FROM WEDSTRIJDEN
WHERE SALDO > 1
```

Voorbeeld 18.20:

```
SELECT COLUMN_NAME, DATA_TYPE, CHAR_LENGTH, NULLABLE
FROM COLUMNS
WHERE TABLE_NAME = 'SPELERS'
AND TABLE_CREATOR = 'TENNIS'
ORDER BY COLUMN_NO
```

Voorbeeld 18.21:

```
SELECT 'SPELERS' AS TABEL_NAAM, COUNT(*) AS AANTAL_RIJEN,
(SELECT COUNT(*)
 FROM COLUMNS
 WHERE TABLE_NAME = 'SPELERS'
 AND TABLE_CREATOR = 'TENNIS') AS S
FROM SPELERS
UNION
SELECT 'TEAMS', COUNT(*),
(SELECT COUNT(*)
 FROM COLUMNS
 WHERE TABLE_NAME = 'TEAMS'
 AND TABLE_CREATOR = 'TENNIS') AS T
FROM TEAMS
UNION
SELECT 'BOETES', COUNT(*),
(SELECT COUNT(*)
 FROM COLUMNS
 WHERE TABLE_NAME = 'BOETES'
 AND TABLE_CREATOR = 'TENNIS') AS B
FROM BOETES
UNION
SELECT 'WEDSTRIJDEN', COUNT(*),
(SELECT COUNT(*)
 FROM COLUMNS
 WHERE TABLE_NAME = 'WEDSTRIJDEN'
 AND TABLE_CREATOR = 'TENNIS') AS W
FROM WEDSTRIJDEN
UNION
SELECT 'BESTUURSLEDEN', COUNT(*),
(SELECT COUNT(*)
 FROM COLUMNS
 WHERE TABLE_NAME = 'BESTUURSLEDEN'
 AND TABLE_CREATOR = 'TENNIS') AS BL
FROM BESTUURSLEDEN
ORDER BY TABEL_NAAM
```


4.16 SQL-instructies voor hoofdstuk 19

Voorbeeld 19.1:

```
CREATE TABLE SPELERS (
  SPELERSNR    INTEGER NOT NULL PRIMARY KEY,
  NAAM         CHAR(15) NOT NULL,
  VOORLETTERS CHAR(3) NOT NULL,
  GEB_DATUM   DATE,
  GESLACHT    CHAR(1) NOT NULL,
  JAARTOE     SMALLINT NOT NULL,
  STRAAT      VARCHAR(30) NOT NULL,
  HUISNR      CHAR(4),
  POSTCODE   CHAR(6),
  PLAATS      VARCHAR(30) NOT NULL,
  TELEFOON   CHAR(13),
  BONDSNR    CHAR(4))
;
CREATE TABLE SPELERS (
  SPELERSNR    INTEGER NOT NULL,
  NAAM         CHAR(15) NOT NULL,
  VOORLETTERS CHAR(3) NOT NULL,
  GEB_DATUM   DATE,
  GESLACHT    CHAR(1) NOT NULL,
  JAARTOE     SMALLINT NOT NULL,
  STRAAT      VARCHAR(30) NOT NULL,
  HUISNR      CHAR(4),
  POSTCODE   CHAR(6),
  PLAATS      VARCHAR(30) NOT NULL,
  TELEFOON   CHAR(13),
  BONDSNR    CHAR(4),
  PRIMARY KEY (SPELERSNR))
```

Voorbeeld 19.2:

```
CREATE TABLE DIPLOMAS (
  CURSIST     INTEGER NOT NULL,
  CURSUS     INTEGER NOT NULL,
  DATUM      DATE NOT NULL,
  SUCCESVOL CHAR(1),
  LOCATIE    VARCHAR(50),
  PRIMARY KEY (CURSIST, CURSUS, DATUM))
```

Voorbeeld 19.3:

```
CREATE TABLE TEAMS (
  TEAMNR     INTEGER NOT NULL,
  SPELERSNR INTEGER NOT NULL UNIQUE,
  DIVISIE    CHAR(6) NOT NULL,
  PRIMARY KEY (TEAMNR))
;
CREATE TABLE TEAMS (
  TEAMNR     INTEGER NOT NULL,
  SPELERSNR INTEGER NOT NULL,
  DIVISIE    CHAR(6) NOT NULL,
  PRIMARY KEY (TEAMNR),
  UNIQUE     (SPELERSNR))
```

Voorbeeld 19.4:

```
CREATE TABLE SPELERS (
  SPELERSNR    INTEGER NOT NULL,
  NAAM         CHAR(15) NOT NULL,
  VOORLETTERS CHAR(3) NOT NULL,
  GEB_DATUM   DATE,
```

```

GESLACHT      CHAR(1) NOT NULL,
JAARTOE       SMALLINT NOT NULL,
STRAAT        VARCHAR(30) NOT NULL,
HUISNR        CHAR(4),
POSTCODE      CHAR(6),
PLAATS        VARCHAR(30) NOT NULL,
TELEFOON      CHAR(13),
BONDSNR       CHAR(4) UNIQUE,
PRIMARY KEY   (SPELERSNR)

```

Voorbeeld 19.5:

```

CREATE TABLE TEAMS (
  TEAMNR      INTEGER NOT NULL,
  SPELERSNR   INTEGER NOT NULL,
  DIVISIE     CHAR(6) NOT NULL,
  PRIMARY KEY (TEAMNR),
  FOREIGN KEY (SPELERSNR) REFERENCES SPELERS (SPELERSNR))
;
CREATE TABLE TEAMS (
  TEAMNR      INTEGER NOT NULL,
  SPELERSNR   INTEGER NOT NULL,
  DIVISIE     CHAR(6) NOT NULL,
  PRIMARY KEY (TEAMNR),
  FOREIGN KEY (SPELERSNR) REFERENCES SPELERS (SPELERSNR))
ENGINE=INNODB
;
SELECT *
FROM   TEAMS
WHERE  SPELERSNR NOT IN
      (SELECT SPELERSNR
       FROM   SPELERS)

```

Voorbeeld 19.6:

```

CREATE TABLE TEAMS (
  TEAMNR      INTEGER NOT NULL,
  SPELERSNR   INTEGER NOT NULL,
  DIVISIE     CHAR(6) NOT NULL,
  PRIMARY KEY (TEAMNR),
  FOREIGN KEY (SPELERSNR) REFERENCES SPELERS (SPELERSNR))

```

Voorbeeld 19.7:

```

CREATE TABLE WEDSTRIJDEN (
  WEDSTRIJDNR INTEGER NOT NULL,
  TEAMNR      INTEGER NOT NULL,
  SPELERSNR   INTEGER NOT NULL,
  GEWONNEN    INTEGER NOT NULL,
  VERLOREN    INTEGER NOT NULL,
  PRIMARY KEY (WEDSTRIJDNR),
  FOREIGN KEY (TEAMNR) REFERENCES TEAMS (TEAMNR),
  FOREIGN KEY (SPELERSNR) REFERENCES SPELERS (SPELERSNR))

```

Voorbeeld 19.8:

```

CREATE TABLE BOETES (
  BETALINGSNR INTEGER NOT NULL,
  SPELERSNR   INTEGER NOT NULL,
  DATUM       DATE NOT NULL,
  BEDRAG      DECIMAL(7,2) NOT NULL,
  PRIMARY KEY (BETALINGSNR),
  FOREIGN KEY (SPELERSNR) REFERENCES SPELERS (SPELERSNR))
;
CREATE TABLE WERKNEMERS (
  WERK_NR     CHAR(10) NOT NULL,
  BAAS_NR     CHAR(10),

```

```

PRIMARY KEY (WERK_NR),
FOREIGN KEY (BAAS_NR) REFERENCES WERKNEMERS (WERK_NR))

```

Voorbeeld 19.9:

```

CREATE TABLE SPELERS_MET_PASPOORT (
  SPELERSNR      INTEGER NOT NULL PRIMARY KEY,
  PASPOORTNR     CHAR(10) NOT NULL UNIQUE,
  NAAM           CHAR(15) NOT NULL,
  VOORLETTERS    CHAR(3) NOT NULL,
  GEB_DATUM      DATE,
  GESLACHT       CHAR(1) NOT NULL,
  JAARTOE        SMALLINT NOT NULL,
  STRAAT         VARCHAR(30) NOT NULL,
  HUISNR         CHAR(4),
  POSTCODE       CHAR(6),
  PLAATS         VARCHAR(30) NOT NULL,
  TELEFOON       CHAR(13),
  BONDSNR        CHAR(4))
;
CREATE TABLE PASPOORT_VISA (
  PASPOORTNR     CHAR(10) NOT NULL,
  VISA_CODE       CHAR(10) NOT NULL,
  VERLOOPDATUME  DATE NOT NULL,
  PRIMARY KEY     (PASPOORTNR, VISA_CODE),
  FOREIGN KEY     (PASPOORTNR) REFERENCES SPELERS_MET_PASPOORT (PASPOORTNR))

```

Voorbeeld 19.10:

```

CREATE TABLE BOETES (
  BETALINGSNR    INTEGER NOT NULL,
  SPELERSNR      INTEGER NOT NULL,
  DATUM          DATE NOT NULL,
  BEDRAG         DECIMAL(7,2) NOT NULL,
  PRIMARY KEY     (BETALINGSNR),
  FOREIGN KEY     (SPELERSNR) REFERENCES SPELERS (SPELERSNR)
                  ON UPDATE RESTRICT
                  ON DELETE RESTRICT)

```

Voorbeeld 19.11:

```

CREATE TABLE BOETES (
  BETALINGSNR    INTEGER NOT NULL,
  SPELERSNR      INTEGER NOT NULL,
  DATUM          DATE NOT NULL,
  BEDRAG         DECIMAL(7,2) NOT NULL,
  PRIMARY KEY     (BETALINGSNR),
  FOREIGN KEY     (SPELERSNR) REFERENCES SPELERS (SPELERSNR)
                  ON DELETE CASCADE)
;
DELETE
FROM   SPELERS
WHERE  SPELERSNR = 127
;
DELETE
FROM   BOETES
WHERE  SPELERSNR = 127

```

Voorbeeld 19.12:

```

CREATE TABLE BOETES (
  BETALINGSNR    INTEGER NOT NULL,
  SPELERSNR      INTEGER,
  DATUM          DATE NOT NULL,
  BEDRAG         DECIMAL(7,2) NOT NULL,
  PRIMARY KEY     (BETALINGSNR),
  FOREIGN KEY     (SPELERSNR) REFERENCES SPELERS (SPELERSNR))

```

```
ON DELETE SET NULL)
```

Voorbeeld 19.13:

```
CREATE TABLE SPELERS_X (
  SPELERSNR  INTEGER NOT NULL,
  GESLACHT   CHAR(1) NOT NULL CHECK(GESLACHT IN ('M', 'V'))
);
CREATE TABLE SPELERS_MET_NULL (
  SPELERSNR  INTEGER NOT NULL,
  GESLACHT   CHAR(1) CHECK(GESLACHT IN ('M', 'V'))
```

Voorbeeld 19.14:

```
CREATE TABLE SPELERS_Y (
  SPELERSNR  INTEGER NOT NULL,
  GEB_DATUM  DATE NOT NULL CHECK(GEB_DATUM > '1920-01-01')
```

Voorbeeld 19.15:

```
CREATE TABLE SPELERS_Z (
  SPELERSNR  INTEGER NOT NULL,
  GEB_DATUM  DATE,
  JAARTOE    INTEGER NOT NULL,
  CHECK(YEAR(GEB_DATUM) < JAARTOE))
;
CREATE TABLE SPELERS_W (
  SPELERSNR  INTEGER,
  GEB_DATUM  DATE NOT NULL,
  JAARTOE    INTEGER NOT NULL,
  CHECK(YEAR(GEB_DATUM) < JAARTOE),
  CHECK(GEB_DATUM > '1920-01-01'),
  CHECK(JAARTOE < 1880))
```

Voorbeeld 19.16:

```
CREATE TABLE SPELERS_V
  SPELERSNR  SMALLINT NOT NULL,
  GESLACHT   CHAR(1) NOT NULL
  CHECK(GESLACHT IN (SELECT GESLACHT FROM SPELERS)))
```

Voorbeeld 19.17:

```
CREATE TABLE DIPLOMAS (
  CURSIST    INTEGER NOT NULL,
  CURSUS     INTEGER NOT NULL,
  DATUM      DATE NOT NULL,
  SUCCESVOL  CHAR(1),
  LOCATIE    VARCHAR(50),
  CONSTRAINT PRIMARY_KEY_DIPLOMAS
  PRIMARY KEY (CURSIST, CURSUS, DATUM))
```

Voorbeeld 19.18:

```
CREATE TABLE SPELERS (
  SPELERSNR  INTEGER NOT NULL,
  NAAM       CHAR(15) NOT NULL,
  VOORLETTERS CHAR(3) NOT NULL,
  GEB_DATUM  DATE,
  GESLACHT   CHAR(1) NOT NULL,
  JAARTOE    SMALLINT NOT NULL,
  STRAAT     VARCHAR(30) NOT NULL,
  HUISNR     CHAR(4),
  POSTCODE   CHAR(6),
  PLAATS     VARCHAR(30) NOT NULL,
  TELEFOON   CHAR(13),
  BONDSNR    CHAR(4),
```

```

CONSTRAINT  PRIMAIRE_SLEUTEL_SPELERS PRIMARY KEY(SPELERSNR),
CONSTRAINT  JAARTOETREDING CHECK(JAARTOE > 1969),
CONSTRAINT  POSTCODE_ZES_TEKENS_LANG CHECK(POSTCODE LIKE '_____' ),
CONSTRAINT  TOEGESTANE_WAARDEN_GESLACHT CHECK(GESLACHT IN ('M', 'V'))

```

4.17 SQL-instructies voor hoofdstuk 20

Voorbeeld 20.1:

```

SHOW CHARACTER SET
;
SELECT  CHARACTER_SET_NAME, DESCRIPTION, DEFAULT_COLLATE_NAME, MAXLEN
FROM    INFORMATION_SCHEMA.CHARACTER_SETS

```

Voorbeeld 20.2:

```

SHOW COLLATION LIKE 'utf8%'
;
SELECT  *
FROM    INFORMATION_SCHEMA.COLLATIONS
WHERE   COLLATION_NAME LIKE 'utf8%'

```

Voorbeeld 20.3:

```

CREATE TABLE TABUCS2 (
  KOL1 CHAR(10) CHARACTER SET ucs2 NOT NULL PRIMARY KEY,
  KOL2 VARCHAR(10) CHARACTER SET ucs2)

```

Voorbeeld 20.4:

```

CREATE TABLE TABDEFKARSET (
  KOL1 CHAR(10) NOT NULL,
  KOL2 VARCHAR(10))
;
SELECT  COLUMN_NAME, CHARACTER_SET_NAME
FROM    INFORMATION_SCHEMA.COLUMNS
WHERE   TABLE_NAME = 'TABDEFKARSET'

```

Voorbeeld 20.5:

```

CREATE TABLE TABUTF8 (
  KOL1 CHAR(10) NOT NULL,
  KOL2 VARCHAR(10))
DEFAULT CHARACTER SET utf8
;
SELECT  COLUMN_NAME, CHARACTER_SET_NAME
FROM    INFORMATION_SCHEMA.COLUMNS
WHERE   TABLE_NAME = 'TABUTF8'

```

Voorbeeld 20.6:

```

SELECT  TABLE_NAME, COLUMN_NAME, COLLATION_NAME
FROM    INFORMATION_SCHEMA.COLUMNS
WHERE   TABLE_NAME IN ('TABUCS2', 'TABDEFKARSET')

```

Voorbeeld 20.7:

```

CREATE TABLE TABCOLLATE (
  KOL1 CHAR(10)
      CHARACTER SET utf8
      COLLATE utf8_romanian_ci NOT NULL,
  KOL2 VARCHAR(10)
      CHARACTER SET utf8
      COLLATE utf8_spanish_ci)

```

```

;
SELECT COLUMN_NAME, CHARACTER_SET_NAME, COLLATION_NAME
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'TABCOLLATE'

```

Voorbeeld 20.8:

```

CREATE TABLE TABDEFCOL (
    KOL1 CHAR(10) NOT NULL,
    KOL2 VARCHAR(10))
CHARACTER SET utf8
COLLATE utf8_romanian_ci
;
SELECT COLUMN_NAME, CHARACTER_SET_NAME, COLLATION_NAME
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'TABDEFCOL'

```

Voorbeeld 20.9:

```

CREATE TABLE TWEELARSETS (
    KOL1 CHAR(10) CHARACTER SET 'latin1' NOT NULL,
    KOL2 VARCHAR(10) CHARACTER SET 'hp8')
;
INSERT INTO TWEELARSETS VALUES ('A', 'A')
;
SELECT *
FROM TWEELARSETS
WHERE KOL1 = KOL2

```

Voorbeeld 20.10:

```

CREATE TABLE TWEELCOLL (
    KOL1 CHAR(10) COLLATE 'latin1_general_ci' NOT NULL,
    KOL2 VARCHAR(10) COLLATE 'latin1_danish_ci')
;
INSERT INTO TWEELCOLL VALUES ('A', 'A')
;
SELECT *
FROM TWEELCOLL
WHERE KOL1 = KOL2
;
SELECT *
FROM TWEELCOLL
WHERE KOL1 COLLATE latin1_danish_ci = KOL2

SELECT *
FROM TWEELCOLL
WHERE KOL1 COLLATE utf8_general_ci = KOL2

```

Voorbeeld 20.11:

```

SELECT _utf8'database'

```

Voorbeeld 20.12:

```

SELECT COLLATION(_utf8'database'),
    COLLATION(_utf8'database' COLLATE utf8_bin),
    COLLATION((SELECT MAX(NAAM) FROM SPELERS))

```

Voorbeeld 20.13:

```

SELECT CHARSET(_utf8'database'),
    CHARSET((SELECT MAX(NAAM) FROM SPELERS))

```

Voorbeeld 20.14:

```

SELECT _latin1'Muller' AS NAAM

```

```
UNION
SELECT CONCAT('M', _latin1 x'FC', 'ller')
ORDER BY NAAM COLLATE latin1_swedish_ci
```

Voorbeeld 20.15:

```
CREATE TABLE LETTERS (
  VOLGNR  INTEGER NOT NULL PRIMARY KEY,
  LETTER  CHAR(1) CHARACTER SET latin2 NOT NULL)
;
INSERT INTO LETTERS VALUES (1, 'e'), (2, 'é'),(3, 'ě')
;
SELECT  LETTER, COUNT(*)
FROM    (SELECT  LETTER COLLATE latin2_czech_cs AS LETTER
        FROM    LETTERS) AS LATIN2_CZECH_LETTERS
GROUP BY LETTER
;
SELECT  LETTER, COUNT(*)
FROM    (SELECT  LETTER COLLATE latin2_croatian_ci AS LETTER
        FROM    LETTERS) AS LATIN2_CROATIAN_LETTERS
GROUP BY LETTER
```

Voorbeeld 20.16:

```
SELECT  LETTER
FROM    LETTERS
ORDER BY LETTER
```

Voorbeeld 20.17:

```
SELECT  COERCIBILITY('Rick' COLLATE latin1_general_ci) AS C0,
        COERCIBILITY(Teamnr) AS C2,
        COERCIBILITY(USER()) AS C3,
        COERCIBILITY('Rick') AS C4,
        COERCIBILITY(NULL) AS C5
FROM    TEAMS
WHERE   Teamnr = 1
```

Voorbeeld 20.18:

```
SELECT @@COLLATION_DATABASE
```

Voorbeeld 20.19:

```
SHOW VARIABLES LIKE 'CHARACTER_SET%'
```

4.18 SQL-instructies voor hoofdstuk 21

Voorbeeld 21.1:

```
DROP TABLE SPELERS
```

Voorbeeld 21.2:

```
RENAME TABLE SPELERS TO TENNISERS
```

Voorbeeld 21.3:

```
ALTER TABLE SPELERS RENAME TO TENNISERS
```

Voorbeeld 21.4:

```
ALTER TABLE SPELERS
  CONVERT TO CHARACTER SET utf8 COLLATE utf8_general_ci
```

Voorbeeld 21.5:

```
ALTER TABLE TEAMS
  ADD COLUMN SOORT CHAR(1)
```

Voorbeeld 21.6:

```
ALTER TABLE TEAMS
  ADD COLUMN SOORT CHAR(1) AFTER TEAMNR
```

Voorbeeld 21.7:

```
ALTER TABLE TEAMS
  ADD (CATEGORIE  VARCHAR(20) NOT NULL,
       IMAGO       INTEGER DEFAULT 10)
```

Voorbeeld 21.8:

```
ALTER TABLE SPELERS
  CHANGE GEB_DATUM GEBOORTEDATUM DATE
```

Voorbeeld 21.9:

```
ALTER TABLE SPELERS
  CHANGE PLAATS PLAATS VARCHAR(40) NOT NULL
```

Voorbeeld 21.10:

```
ALTER TABLE SPELERS
  CHANGE PLAATS PLAATS VARCHAR(5) NOT NULL
```

Voorbeeld 21.11:

```
ALTER TABLE WEDSTRIJDEN
  CHANGE GEWONNEN GEWONNEN SMALLINT
```

Voorbeeld 21.12:

```
ALTER TABLE SPELERS
  CHANGE PLAATS PLAATS VARCHAR(5) NOT NULL AFTER SPELERSNR
```

Voorbeeld 21.13:

```
ALTER TABLE SPELERS
  MODIFY PLAATS VARCHAR(5) NOT NULL AFTER SPELERSNR
```

Voorbeeld 21.14:

```
ALTER TABLE BESTUURSLEDEN
  ALTER FUNCTIE SET DEFAULT 'Lid'
;
ALTER TABLE BESTUURSLEDEN
  MODIFY FUNCTIE CHAR(20) DEFAULT 'Lid'
```

Voorbeeld 21.15:

```
ALTER TABLE BESTUURSLEDEN
  ALTER FUNCTIE DROP DEFAULT
```

Voorbeeld 21.16:

```
ALTER TABLE TEAMS
  DROP COLUMN SOORT
```

Voorbeeld 21.17:


```

CREATE TABLE T1 (
  A INTEGER NOT NULL PRIMARY KEY,
  B INTEGER NOT NULL)
;
CREATE TABLE T2 (
  A INTEGER NOT NULL PRIMARY KEY,
  B INTEGER NOT NULL,
  CONSTRAINT C1 CHECK (B > 0),
  CONSTRAINT FK1 FOREIGN KEY (A) REFERENCES T1 (A))
;
ALTER TABLE T1
  ADD CONSTRAINT FK2 FOREIGN KEY (A) REFERENCES T2 (A)

```

Voorbeeld 21.18:

```

ALTER TABLE BOETES
  DROP PRIMARY KEY

```

Voorbeeld 21.19:

```

ALTER TABLE T2
  DROP CONSTRAINT FK1

```

Voorbeeld 21.20:

```

ALTER TABLE T2
  DROP CONSTRAINT C1

```

4.19 SQL-instructies voor hoofdstuk 22

Voorbeeld 22.1:

```

CREATE TABLE KINDEREN (
  SPELERSNR INTEGER NOT NULL PRIMARY KEY,
  VNAAM1 CHAR(20),
  VNAAM2 CHAR(20),
  VNAAM3 CHAR(20))
;
INSERT INTO KINDEREN VALUES
  ( 6, 'Mieke', 'Diana', 'Joop'),
  (44, NULL, NULL, NULL),
  (83, 'Willem-Tel', 'Jan-Olaf', NULL)

```

Voorbeeld 22.2:

```

SELECT  SPELERSNR, 0 AS AANTAL_KINDEREN
FROM    KINDEREN
WHERE   VNAAM1 IS NULL
AND     VNAAM2 IS NULL
AND     VNAAM3 IS NULL
UNION
SELECT  SPELERSNR, 1
FROM    KINDEREN
WHERE   VNAAM1 IS NOT NULL
AND     VNAAM2 IS NULL
AND     VNAAM3 IS NULL
UNION
SELECT  SPELERSNR, 2
FROM    KINDEREN
WHERE   VNAAM1 IS NOT NULL
AND     VNAAM2 IS NOT NULL
AND     VNAAM3 IS NULL
UNION
SELECT  SPELERSNR, 3

```

```

FROM    KINDEREN
WHERE   VNAAM1 IS NOT NULL
AND     VNAAM2 IS NOT NULL
AND     VNAAM3 IS NOT NULL
UNION
SELECT  SPELERSNR, 0
FROM    SPELERS
WHERE   SPELERSNR NOT IN
        (SELECT  SPELERSNR
         FROM    KINDEREN)
ORDER BY AANTAL_KINDEREN DESC, SPELERSNR ASC

```

Voorbeeld 22.3:

```

SELECT  *
FROM    KINDEREN
WHERE   SPELERSNR = 6
;
UPDATE  KINDEREN
SET     VNAAM2 = 'Diane'
WHERE   SPELERSNR = 6
;
SELECT  SPELERSNR, COUNT(*)
FROM    SPELERS LEFT OUTER JOIN KINDEREN
        ON SPELERS.SPELERSNR = KINDEREN.SPELERSNR
GROUP BY SPELERSNR
;
UPDATE  KINDEREN
SET     VNAAM = 'Diane'
WHERE   SPELERSNR = 6
AND     VNAAM = 'Diana'

```

Voorbeeld 22.4:

```

SELECT  ADRES
FROM    SPELERS
WHERE   SPELERSNR = 44

```

Voorbeeld 22.5:

```

SELECT  NAAM, BEDRAG
FROM    BOETES AS BT, SPELERS AS S
WHERE   BT.SPELERSNR = S.SPELERSNR
;
SELECT  NAAM, BEDRAG
FROM    BOETES

```

Voorbeeld 22.6:

```

SELECT  SPELERSNR, SUM(BEDRAG) AS TOTAAL
FROM    SPELERS LEFT OUTER JOIN BOETES
        ON SPELERS.SPELERSNR = BOETES.SPELERSNR
GROUP BY SPELERSNR
ORDER BY SPELERSNR
;
SELECT  SPELERSNR, TOT_BEDRAG
FROM    SPELERS
;
SELECT  WEDSTRIJDENR
FROM    WEDSTRIJDEN
WHERE   GEWONNEN - 2 > VERLOREN
;
SELECT  NAAM
FROM    SPELERS INNER JOIN BOETES
        ON SPELERS.SPELERSNR = BOETES.SPELERSNR

```

4.20 SQL-instructies voor hoofdstuk 23

Voorbeeld 23.4:

```
SELECT *
FROM SPELERS
WHERE SPELERSNR = 44
```

Voorbeeld 23.5:

```
SELECT SPELERSNR, PLAATS
FROM SPELERS
WHERE SPELERSNR < 10
AND PLAATS = 'Den Haag'
ORDER BY SPELERSNR
```

Voorbeeld 23.6:

```
SELECT NAAM, VOORLETTERS
FROM SPELERS
WHERE PLAATS =
  (SELECT PLAATS
   FROM SPELERS
   WHERE SPELERSNR = 44)
```

Voorbeeld 23.7:

```
CREATE INDEX SPEL_PC
ON SPELERS (POSTCODE ASC)
```

Voorbeeld 23.8:

```
CREATE INDEX WED_GV
ON WEDSTRIJDEN (GEWONNEN, VERLOREN)
```

Voorbeeld 23.9:

```
CREATE UNIQUE INDEX NAAMVOOR
ON SPELERS (NAAM, VOORLETTERS)
```

Voorbeeld 23.10:

```
CREATE UNIQUE CLUSTERED INDEX SPELERS_CLUSTERED
ON SPELERS (SPELERSNR)
```

Voorbeeld 23.11:

```
ALTER TABLE TEAMS
  ADD INDEX TEAMS_DIVISIE USING BTREE (DIVISIE)
```

Voorbeeld 23.12:

```
DROP INDEX SPEL_PC
;
DROP INDEX WED_GV
;
DROP INDEX NAAMVOOR
```

Voorbeeld 23.13:

```
CREATE TABLE T1 (
  KOL1 INTEGER NOT NULL,
  KOL2 DATE NOT NULL UNIQUE,
  KOL3 INTEGER NOT NULL,
  KOL4 INTEGER NOT NULL,
```

```

    PRIMARY KEY (KOL1, KOL4),
    UNIQUE (KOL3, KOL4),
    UNIQUE (KOL3, KOL1))
;
CREATE UNIQUE INDEX "PRIMARY" USING BTREE
ON    T1 (KOL1, KOL4)
;
CREATE UNIQUE INDEX KOL2 USING BTREE
ON    T1 (KOL2)
;
CREATE UNIQUE INDEX KOL3 USING BTREE
ON    T1 (KOL3, KOL4)
;
CREATE UNIQUE INDEX KOL3_2 USING BTREE
ON    T1 (KOL3, KOL1)

```

Voorbeeld 23.14:

```

CREATE TABLE SPELERS_XXL (
    SPELERSNR    INTEGER NOT NULL PRIMARY KEY,
    NAAM         CHAR(15) NOT NULL,
    VOORLETTERS CHAR(3) NOT NULL,
    GEB_DATUM   DATE,
    GESLACHT    CHAR(1) NOT NULL,
    JAARTOE     SMALLINT NOT NULL,
    STRAAT      VARCHAR(30) NOT NULL,
    HUISNR      CHAR(4),
    POSTCODE    CHAR(6),
    PLAATS      VARCHAR(30) NOT NULL,
    TELEFOON    CHAR(13),
    BONDSNR     CHAR(8))

```

Voorbeeld 23.15:

```

CREATE PROCEDURE VUL_SPELERS_XXL
    (IN AANTAL_SPELERS INTEGER)
BEGIN
    DECLARE TELLER INTEGER;
    TRUNCATE TABLE SPELERS_XXL;
    COMMIT WORK;
    SET TELLER = 1;
    WHILE TELLER <= AANTAL_SPELERS DO
        INSERT INTO SPELERS_XXL VALUES(
            TELLER,
            CONCAT('naam',CAST(TELLER AS CHAR(10))),
            CASE MOD(TELLER,2) WHEN 0 THEN 'v11' ELSE 'v12' END,
            DATE('1960-01-01') + INTERVAL (MOD(TELLER,300)) MONTH,
            CASE MOD(TELLER,20) WHEN 0 THEN 'V' ELSE 'M' END,
            1980 + MOD(TELLER,20),
            CONCAT('straat',CAST(TELLER/10 AS UNSIGNED INTEGER)),
            CAST(CAST(TELLER/10 AS UNSIGNED INTEGER)+1 AS CHAR(4)),
            CONCAT('p',MOD(TELLER,50)),
            CONCAT('plaats',MOD(TELLER,10)),
            '070-6868689',
            CASE MOD(TELLER,3) WHEN 0
                THEN NULL ELSE CAST(TELLER AS CHAR(8)) END);
        IF MOD(TELLER,1000) = 0 THEN
            COMMIT WORK;
        END IF;
        SET TELLER = TELLER + 1;
    END WHILE;
    COMMIT WORK;
END

```

Voorbeeld 23.16:

```

CALL VUL_SPELERS_XXL(100000)

```

Voorbeeld 23.17:

```

CREATE INDEX SPELERS_XXL_VOORLETTERS
  ON SPELERS_XXL(VOORLETTERS)
;
CREATE INDEX SPELERS_XXL_POSTCODE
  ON SPELERS_XXL(POSTCODE)
;
CREATE INDEX SPELERS_XXL_STRAAT
  ON SPELERS_XXL(STRAAT)
;
SELECT *
FROM   SPELERS
WHERE  PLAATS = 'Den Haag'
;
SELECT COUNT(*)
FROM   SPELERS_XXL
WHERE  VOORLETTERS = 'v11'
;
SELECT COUNT(*)
FROM   SPELERS_XXL
WHERE  POSTCODE = 'p25'
;
SELECT COUNT(*)
FROM   SPELERS_XXL
WHERE  STRAAT = 'straat164'
;
SELECT COUNT(*)
FROM   SPELERS_XXL
WHERE  GESLACHT = 'M'
;
SELECT COUNT(*)
FROM   SPELERS_XXL
WHERE  GESLACHT = 'V'
;
SELECT *
FROM   SPELERS
WHERE  NAAM = 'Cools'
AND    VOORLETTERS = 'DD'
;
CREATE INDEX NAAMVOOR
  ON   SPELERS (NAAM, VOORLETTERS)

```

Voorbeeld 23.18:

```

CREATE INDEX SPEL_WEDS
  ON   SPELERS(SPELERSNR), WEDSTRIJDEN(SPELERSNR)

```

Voorbeeld 23.19:

```

CREATE INDEX WED_HALVESALDO
  ON   WEDSTRIJDEN((GEWONNEN - VERLOREN)/2)
;
SELECT *
FROM   WEDSTRIJDEN
WHERE  (GEWONNEN - VERLOREN)/2 > 1

```

Voorbeeld 23.20:

```

CREATE INDEX BOETES_DATUM
  ON   BOETES
WHERE  DATUM > '1996-12-31'

```

Voorbeeld 23.21:

```
CREATE HASH INDEX SPELERSNR_HASH
ON   SPELERS (SPELERSNR)
WITH PAGES = 100
```

Voorbeeld 23.22:

```
CREATE BITMAP INDEX SPELERS_GESLACHT
ON   SPELERS(GESLACHT)
```

Voorbeeld 23.23:

```
SELECT TABLE_CREATOR, TABLE_NAME, COUNT(*)
FROM   INDEXES
GROUP BY TABLE_CREATOR, TABLE_NAME
HAVING COUNT(*) > 1
```

Voorbeeld 23.24:

```
SELECT TABLE_CREATOR, TABLE_NAME
FROM   TABLES AS TAB
WHERE  NOT EXISTS
      (SELECT *
       FROM INDEXES AS IDX
       WHERE TAB.TABLE_CREATOR = IDX.TABLE_CREATOR
            AND TAB.TABLE_NAME = TAB.TABLE_NAME
            AND IDX.UNIQUE_ID = 'YES')
```

4.21 SQL-instructies voor hoofdstuk 24

Voorbeeld 24.1:

```
CREATE VIEW WOONPLAATSEN AS
SELECT DISTINCT PLAATS
FROM   SPELERS
```

Voorbeeld 24.2:

```
SELECT *
FROM   WOONPLAATSEN
```

Voorbeeld 24.3:

```
CREATE VIEW WSPELERS AS
SELECT SPELERSNR, BONDSNR
FROM   SPELERS
WHERE  BONDSNR IS NOT NULL
;
SELECT *
FROM   WSPELERS
```

Voorbeeld 24.4:

```
SELECT *
FROM   WSPELERS
WHERE  SPELERSNR BETWEEN 6 AND 44
;
SELECT SPELERSNR, BONDSNR
FROM   SPELERS
WHERE  BONDSNR IS NOT NULL
AND    SPELERSNR BETWEEN 6 AND 44
```

Voorbeeld 24.5:

```
DELETE
```

```
FROM    WSPELERS
WHERE   BONDSNR = '7060'
```

Voorbeeld 24.6:

```
CREATE  VIEW SOMMIGEN AS
SELECT  *
FROM    WSPELERS
WHERE   SPELERSNR BETWEEN 6 AND 27
;
SELECT  *
FROM    SOMMIGEN
```

Voorbeeld 24.7:

```
CREATE VIEW CIJFERS AS
SELECT 0 CIJFER UNION SELECT 1 UNION
SELECT 2 UNION SELECT 3 UNION
SELECT 4 UNION SELECT 5 UNION
SELECT 6 UNION SELECT 7 UNION
SELECT 8 UNION SELECT 9
;
SELECT * FROM CIJFERS
```

Voorbeeld 24.8:

```
CREATE  VIEW HAGENEZEN (SPELERSNR, NAAM, VLS, GEBOORTE) AS
SELECT  SPELERSNR, NAAM, VOORLETTERS, GEB_DATUM
FROM    SPELERS
WHERE   PLAATS = 'Den Haag'
;
SELECT  *
FROM    HAGENEZEN
WHERE   SPELERSNR > 90
```

Voorbeeld 24.9:

```
CREATE  VIEW INWONERS (PLAATS, AANTAL) AS
SELECT  PLAATS, COUNT(*)
FROM    SPELERS
GROUP BY PLAATS
```

Voorbeeld 24.10:

```
CREATE  VIEW VETERANEN AS
SELECT  *
FROM    SPELERS
WHERE   GEB_DATUM < '1960-01-01'
;
UPDATE  VETERANEN
SET     GEB_DATUM = '1970-09-01'
WHERE   SPELERSNR = 2
;
CREATE  VIEW VETERANEN AS
SELECT  *
FROM    SPELERS
WHERE   GEB_DATUM < '1960-01-01'
WITH    CHECK OPTION
```

Voorbeeld 24.11:

```
CREATE  VIEW RIJSWIJKSE_VETERANEN AS
SELECT  *
FROM    VETERANEN
WHERE   PLAATS = 'Rijswijk'
WITH    CASCADED CHECK OPTION
```

Voorbeeld 24.12:

```
DROP VIEW WSPELERS
```

Voorbeeld 24.13:

```
SELECT TABLE_NAME
FROM TABLES
WHERE TABLE_NAME = 'VOORRAAD'
AND TABLE_CREATOR = 'TENNIS'
UNION
SELECT VIEW_NAME
FROM VIEWS
WHERE VIEW_NAME = 'VOORRAAD'
AND VIEW_CREATOR = 'TENNIS'
;
CREATE VIEW TOTALEN
(SPELERSNR, TOT_BEDRAG) AS
SELECT SPELERSNR, SUM(BEDRAG)
FROM BOETES
GROUP BY SPELERSNR
;
SELECT *
FROM TOTALEN
WHERE TOT_BEDRAG > 100
;
SELECT MAX(TOT_BEDRAG)
FROM TOTALEN
;
SELECT NAAM, TOT_BEDRAG
FROM SPELERS, TOTALEN
WHERE SPELERS.SPELERSNR = TOTALEN.SPELERSNR
;
CREATE VIEW LEEFTIJDEN (SPELERSNR, BEGINLEEF) AS
SELECT SPELERSNR, JAARTOE – YEAR(GEB_DATUM)
FROM SPELERS
;
CREATE VIEW SPELERS_NAMEN AS
SELECT SPELERSNR, NAAM, VOORLETTERS
FROM SPELERS
```

Voorbeeld 24.14:

```
CREATE VIEW DUREN AS
SELECT *
FROM SPELERS
WHERE SPELERSNR IN
(SELECT SPELERSNR
FROM BOETES)
;
SELECT SPELERSNR
FROM DUREN
WHERE PLAATS = 'Den Haag'
;
SELECT SPELERSNR
FROM (SELECT *
FROM SPELERS
WHERE SPELERSNR IN
(SELECT SPELERSNR
FROM BOETES)) AS VIEWFORMULE
WHERE PLAATS = 'Den Haag'
```

Voorbeeld 24.15:

```
DELETE
FROM HAGENEZEN
WHERE GEBORTE > '1965-12-31'
```



```

;
DELETE
FROM   SPELERS
WHERE  GEB_DATUM > '1965-12-31'
AND    PLAATS = 'Den Haag'
;
SELECT *
FROM   SPELERS
WHERE  SPELERSNR IN
      (SELECT SPELERSNR
       FROM   BOETES)

```

Voorbeeld 24.16:

```

SELECT *
FROM   SPELERS
WHERE  SPELERSNR IN
      (SELECT SPELERSNR
       FROM   BOETES)
AND    PLAATS = 'Den Haag'
;
SELECT PLAATS, COUNT(*)
FROM   SPELERS
WHERE  SPELERSNR IN
      (SELECT SPELERSNR
       FROM   BOETES)
GROUP BY PLAATS
;
CREATE VIEW BSPELERS AS
SELECT *
FROM   SPELERS
WHERE  SPELERSNR IN
      (SELECT SPELERSNR
       FROM   BOETES)
;
SELECT *
FROM   BSPELERS
WHERE  PLAATS = 'Den Haag'
;
SELECT PLAATS, COUNT(*)
FROM   BSPELERS
GROUP BY PLAATS

```

Voorbeeld 24.18:

```

SELECT DISTINCT NAAM, VOORLETTERS, DIVISIE
FROM   SPELERS AS S, WEDSTRIJDEN AS W, TEAMS AS T
WHERE  S.SPELERSNR = W.SPELERSNR
AND    W.TEAMNR = T.TEAMNR
;
CREATE VIEW TEAMS (TEAMNR, SPELERSNR, DIVISIE) AS
SELECT DISTINCT TEAMNR, AANVOERDER, DIVISIE
FROM   RESULTAAT
;
CREATE VIEW WEDSTRIJDEN AS
SELECT WEDSTRIJDNR, TEAMNR, SPELERSNR,
       GEWONNEN, VERLOREN
FROM   RESULTAAT
;
CREATE VIEW GROTER AS
SELECT DISTINCT SPELERSNR
FROM   BOETES
WHERE  BEDRAG >
      (SELECT AVG(BEDRAG)
       FROM   BOETES
       WHERE  SPELERSNR IN
            (SELECT SPELERSNR

```

```

        FROM    WEDSTRIJDEN
        WHERE   TEAMNR = 2))
;
CREATE VIEW ERE AS
SELECT DISTINCT SPELERSNR
FROM WEDSTRIJDEN
WHERE TEAMNR IN
      (SELECT TEAMNR
       FROM TEAMS
       WHERE DIVISIE = 'ere')
;
SELECT NAAM, VOORLETTERS
FROM SPELERS
WHERE PLAATS = 'Den Haag'
AND SPELERSNR IN
     (SELECT SPELERSNR
      FROM GROTER)
AND SPELERSNR IN
     (SELECT SPELERSNR
      FROM ERE)

```

Voorbeeld 24.19:

```

CREATE VIEW SPELERSG AS
SELECT *
FROM SPELERS
WHERE GESLACHT IN ('M', 'V')
WITH CHECK OPTION

```

4.22 SQL-instructies voor hoofdstuk 25

Voorbeeld 25.1:

```

SELECT SCHEMA_NAME
FROM INFORMATION_SCHEMA.SCHEMATA

```

Voorbeeld 25.2:

```

SELECT TABLE_NAME
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'TENNIS'
ORDER BY TABLE_NAME

```

Voorbeeld 25.3:

```

CREATE DATABASE TENNIS2
  DEFAULT CHARACTER SET utf8
  DEFAULT COLLATE utf8_general_ci

```

Voorbeeld 25.4:

```

SELECT SCHEMA_NAME, DEFAULT_CHARACTER_SET_NAME,
       DEFAULT_COLLATION_NAME
FROM INFORMATION_SCHEMA.SCHEMATA

```

Voorbeeld 25.5:

```

ALTER DATABASE TENNIS2
  DEFAULT CHARACTER SET sjis
  DEFAULT COLLATE sjis_japanese_ci

```

Voorbeeld 25.6:

```

ALTER DATABASE TENNIS CHARACTER SET hp8

```

```

;
CREATE TABLE KARSETHP8 (
  KOL1 CHAR(10) NOT NULL,
  KOL2 VARCHAR(10))
;
SELECT COLUMN_NAME, CHARACTER_SET_NAME, COLLATION_NAME
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'KARSETHP8'

```

Voorbeeld 25.7:

```
ALTER DATABASE TENNIS COLLATE hp8_bin
```

Voorbeeld 25.8:

```
DROP DATABASE TENNIS2
```

4.23 SQL-instructies voor hoofdstuk 26

Voorbeeld 26.1:

```

CREATE USER CHRIS IDENTIFIED BY CHRISSEC
;
CREATE USER PAUL IDENTIFIED BY LUAP
;
CREATE USER 'CHRIS'@'%' IDENTIFIED BY 'CHRISSEC'
;
CREATE USER 'SAM'@'TEST' IDENTIFIED BY 'SAMSEC'

```

Voorbeeld 26.2:

```
DROP USER JAN
```

Voorbeeld 26.3:

```

CREATE USER JOHN
;
ALTER USER JOHN IDENTIFIED BY JOHN1

```

Voorbeeld 26.4:

```

GRANT SELECT
ON SPELERS
TO JAMIE

```

Voorbeeld 26.5:

```

GRANT INSERT, UPDATE
ON TEAMS
TO JAMIE, PIET

```

Voorbeeld 26.6:

```

GRANT SELECT, INSERT
ON BOETES
TO PUBLIC
;
SELECT GRANTOR, GRANTEE, TABLE_NAME, PRIVILEGE
FROM TABLE_AUTHS
WHERE GRANTEE = 'PUBLIC'

```

Voorbeeld 26.7:

```
GRANT UPDATE (SPELERSNR, DIVISIE)
```

```

ON      TEAMS
TO      PAUL
;
SELECT  GRANTOR, GRANTEE, TABLE_NAME,
        COLUMN_NAME, PRIVILEGE
FROM    COLUMN_AUTHS
WHERE   GRANTEE = 'PAUL'

```

Voorbeeld 26.8:

```

GRANT  SELECT
ON     TENNIS.*
TO     PAUL

```

Voorbeeld 26.9:

```

GRANT  CREATE, ALTER, DROP, CREATE VIEW
ON     TENNIS.*
TO     JOHN

```

Voorbeeld 26.10:

```

GRANT  SELECT
ON     INFORMATION_SCHEMA.*
TO     PAUL

```

Voorbeeld 26.11:

```

GRANT  SELECT, INSERT
ON     *
TO     ALYSSA

```

Voorbeeld 26.12:

```

GRANT  CREATE, ALTER, DROP
ON     *.*
TO     MAX

```

Voorbeeld 26.13:

```

GRANT  CREATE USER
ON     *.*
TO     ALYSSA
;
GRANT  ALL PRIVILEGES
ON     *.*
TO     ROOT

```

Voorbeeld 26.14:

```

GRANT  REFERENCES
ON     TEAMS
TO     JIM
WITH   GRANT OPTION
;
GRANT  REFERENCES
ON     JIM
TO     PAUL

```

Voorbeeld 26.15:

```

CREATE ROLE VERKOOP
;
GRANT  SELECT, INSERT
ON     BOETES
TO     VERKOOP
;

```

```
GRANT VERKOOP TO IRENE, KELLY, JIM, MARK
```

Voorbeeld 26.16:

```
DROP ROLE VERKOOP
```

Voorbeeld 26.17:

```
SELECT  GRANTEE
FROM    USER_AUTHS
WHERE   PRIVILEGE = 'SELECT'
UNION
SELECT  GRANTEE
FROM    DATABASE_AUTHS
WHERE   DATABASE_NAME = 'TENNIS'
AND     PRIVILEGE = 'SELECT'
UNION
SELECT  GRANTEE
FROM    TABLE_AUTHS
WHERE   TABLE_CREATOR = 'TENNIS'
AND     PRIVILEGE = 'SELECT'
AND     TABLE_NAME = 'SPELERS'
```

Voorbeeld 26.18:

```
CREATE USER 'JONI'@'localhost' IDENTIFIED BY 'JONIPW'
;
GRANT  SELECT
ON     SPELERS
TO     JONI
;
REVOKE SELECT
ON     SPELERS
FROM   JONI
```

Voorbeeld 26.19:

```
REVOKE REFERENCES
ON     TEAMS
FROM   JOHN
```

Voorbeeld 26.20:

```
REVOKE VERKOOP FROM IRENE
```

Voorbeeld 26.21:

```
REVOKE SELECT
ON     BOETES
FROM   VERKOOP
```

Voorbeeld 26.22:

```
CREATE USER DIANE IDENTIFIED BY 'GEHEIM'
;
CREATE VIEW NAW AS
SELECT NAAM, VOORLETTERS, STRAAT, HUISNR, PLAATS
FROM   SPELERS
WHERE  BONDSNR IS NULL
;
GRANT  SELECT
ON     NAW
TO     DIANE
```

Voorbeeld 26.23:

```
CREATE USER GERARD IDENTIFIED BY 'XYZ1234'
```

```

;
CREATE VIEW INWONERS (PLAATS, AANTAL) AS
SELECT PLAATS, COUNT(*)
FROM SPELERS
GROUP BY PLAATS
;
GRANT SELECT
ON INWONERS
TO GERARD

```

4.24 SQL-instructies voor hoofdstuk 27

```

SELECT CASE
        WHEN MAX(TEAMNR) IS NULL THEN 0
        ELSE MAX(TEAMNR) + 1
      END
FROM TEAMS

```

Voorbeeld 27.1:

```

CREATE TABLE LANDEN (
  LANDNR INTEGER NOT NULL PRIMARY KEY,
  LANDNAAM VARCHAR(30) NOT NULL)
;
CREATE SEQUENCE LANDNUMMERS
;
INSERT INTO LANDEN
  (LANDNR, LANDNAAM)
VALUES (NEXT VALUE FOR LANDNUMMERS, 'China')
;
UPDATE LANDEN
SET LANDNR = NEXT VALUE FOR LANDNUMMERS
WHERE LANDNR = 1

```

Voorbeeld 27.2:

```

SELECT SPELERSNR, NEXT VALUE FOR LANDNUMMERS
FROM SPELERS

```

Voorbeeld 27.3:

```

CREATE SEQUENCE HONDERD START WITH 100
;
SELECT NEXT VALUE FOR HONDERD AS NUMMERS
FROM WEDSTRIJDEN
WHERE WEDSTRIJDNR <= 5

```

Voorbeeld 27.4:

```

CREATE SEQUENCE GROTE_STAPPEN
  INCREMENT BY 100
;
SELECT NEXT VALUE FOR GROTE_STAPPEN AS NUMMERS
FROM WEDSTRIJDEN
WHERE WEDSTRIJDNR <= 5

```

Voorbeeld 27.5:

```

CREATE SEQUENCE ACHTERUIT
  INCREMENT BY -10
;
SELECT NEXT VALUE FOR ACHTERUIT AS NUMMERS
FROM WEDSTRIJDEN
WHERE WEDSTRIJDNR <= 5

```

Voorbeeld 27.6:

```
CREATE SEQUENCE MIN98
  START WITH 100
  INCREMENT BY -1
  MINVALUE 98
;
SELECT  NEXT VALUE FOR MIN98 AS NUMMERS
FROM    WEDSTRIJDEN
WHERE   WEDSTRIJDNR <= 5
```

Voorbeeld 27.7:

```
CREATE SEQUENCE ROND3
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 3
  CYCLE
;
SELECT  NEXT VALUE FOR ROND3 AS NUMMERS
FROM    WEDSTRIJDEN
```

Voorbeeld 27.8:

```
CREATE SEQUENCE STANDAARD
  MINVALUE 1
  MAXVALUE 2147483647
  START WITH 1
  INCREMENT BY 1
  NOCYCLE
  CACHE 20
  ORDER
```

Voorbeeld 27.9:

```
CREATE SEQUENCE BONDSNUMMERS
  START WITH 1000
;
UPDATE  SPELERS
SET     BONDSNR = CHAR(NEXT VALUE FOR BONDSNUMMERS,4)
WHERE   BONDSNR IS NOT NULL
```

Voorbeeld 27.10:

```
CREATE SEQUENCE DRIEVOUDIG
;
SELECT  NEXT VALUE FOR DRIEVOUDIG AS NUMMER1,
        NEXT VALUE FOR DRIEVOUDIG AS NUMMER2,
        NEXT VALUE FOR DRIEVOUDIG AS NUMMER3
FROM    WEDSTRIJDEN
WHERE   WEDSTRIJDNR <= 5
```

Voorbeeld 27.11:

```
CREATE SEQUENCE SBEREKENINGEN
;
SELECT  1000 * (NEXT VALUE FOR SBEREKENINGEN) AS NUMMER1,
        NEXT VALUE FOR SBEREKENINGEN *
        NEXT VALUE FOR SBEREKENINGEN AS NUMMER2,
        MOD(NEXT VALUE FOR SBEREKENINGEN,4) AS NUMMER3
FROM    WEDSTRIJDEN
WHERE   WEDSTRIJDNR <= 5
```

Voorbeeld 27.12:

```
CREATE SEQUENCE TEAMNUMMERS
```

```

;
INSERT INTO TEAMS
      (TEAMNR, SPELERSNR, DIVISIE)
VALUES (NEXT VALUE FOR TEAMNUMMERS, 6, 'ere')
;
UPDATE WEDSTRIJDEN
SET    TEAMNR = PREVIOUS VALUE FOR TEAMNUMMERS
WHERE  WEDSTRIJDNR = 10

```

Voorbeeld 27.13:

```

CREATE SEQUENCE SEQNR1
;
CREATE SEQUENCE SEQNR2
;
SELECT NEXT VALUE FOR SEQNR1 AS NUMMER1,
       CURRENT VALUE FOR SEQNR1 AS NUMMER2,
       CURRENT VALUE FOR SEQNR2 AS NUMMER3,
       NEXT VALUE FOR SEQNR2 AS NUMMER4
FROM   WEDSTRIJDEN
WHERE  WEDSTRIJDNR <= 5

```

Voorbeeld 27.14:

```

ALTER SEQUENCE LANDNUMMERS
RESTART

```

Voorbeeld 27.15:

```

ALTER SEQUENCE LANDNUMMERS
RESTART WITH 100

```

Voorbeeld 27.16:

```

ALTER SEQUENCE VIJFJES
MAXVALUE 800

```

Voorbeeld 27.17:

```

DROP SEQUENCE AFLOPER

```

Voorbeeld 27.18:

```

GRANT ALTER, USAGE
ON     SEQUENCE LANDNUMMERS
TO     BEN

```

4.25 SQL-instructies voor hoofdstuk 28

Voorbeeld 28.1:

```

CREATE TABLE BART.TEST (
      KOL1 INTEGER)

```

Voorbeeld 28.2:

```

CREATE INDEX BART.INDEXA ON TEST (KOL1)

```

Voorbeeld 28.3:

```

CREATE SCHEMA TENNIS_SCHEMA

```

Voorbeeld 28.4:


```
CREATE VIEW TENNIS_SCHEMA.ENKELE_WEDSTRIJDEN AS
SELECT *
FROM WEDSTRIJDEN
WHERE WEDSTRIJDNR < 5
```

Voorbeeld 28.5:

```
CREATE SCHEMA TWEE_TABELLEN
CREATE TABLE TABEL1 (KOL1 INTEGER)
CREATE TABLE TABEL2 (KOL1 INTEGER)
CREATE INDEX INDEX1 ON TABEL1(KOL1)
```

4.26 SQL-instructies voor hoofdstuk 29

Voorbeeld 29.1:

```
CREATE PROCEDURE DELETE_WEDSTRIJDEN
(IN P_SPELERSNR INTEGER)
BEGIN
DELETE
FROM WEDSTRIJDEN
WHERE SPELERSNR = P_SPELERSNR;
END
```

Voorbeeld 29.2:

```
CALL DELETE_WEDSTRIJDEN(8)
```

Voorbeeld 29.5:

```
CREATE PROCEDURE TEST
(OUT GETAL1 INTEGER)
BEGIN
DECLARE GETAL2 INTEGER DEFAULT 100;
SET GETAL1 = GETAL2;
END
;
CALL TEST (@GETAL)
;
SELECT @GETAL
```

Voorbeeld 29.6:

```
CREATE PROCEDURE TEST
(OUT GETAL1 INTEGER)
BEGIN
DECLARE GETAL2 INTEGER
DEFAULT (SELECT COUNT(*) FROM SPELERS);
SET GETAL1 = GETAL2;
END
```

Voorbeeld 29.7:

```
CREATE PROCEDURE HOOGSTE
(IN P1 INTEGER,
IN P2 INTEGER,
OUT P3 INTEGER)
BEGIN
IF P1 > P2 THEN
SET P3 = 1;
ELSEIF P1 = P2 THEN
SET P3 = 2;
ELSE
SET P3 = 3;
```

```
END IF;
END
```

Voorbeeld 29.8:

```
CREATE PROCEDURE FIBONACCI
  (INOUT GETAL1 INTEGER,
   INOUT GETAL2 INTEGER,
   INOUT GETAL3 INTEGER)
BEGIN
  SET GETAL3 = GETAL1 + GETAL2;
  IF GETAL3 > 10000 THEN
    SET GETAL3 = GETAL3 - 10000;
  END IF;
  SET GETAL1 = GETAL2;
  SET GETAL2 = GETAL3;
END
;
SET @A = 16, @B = 27
;
CALL FIBONACCI(@A,@B,@C)
;
SELECT @C
```

Voorbeeld 29.9:

```
CREATE PROCEDURE GROTER
  (OUT T CHAR(10))
BEGIN
  IF (SELECT COUNT(*) FROM SPELERS) >
     (SELECT COUNT(*) FROM BOETES) THEN
    SET T = 'SPELERS';
  ELSEIF (SELECT COUNT(*) FROM SPELERS) =
         (SELECT COUNT(*) FROM BOETES) THEN
    SET T = 'GELIJK';
  ELSE
    SET T = 'BOETES';
  END IF;
END
```

Voorbeeld 29.10:

```
CREATE PROCEDURE LEEFTIJD
  (IN  START_DATUM DATE,
   IN  EIND_DATUM  DATE,
   OUT JAREN       INTEGER,
   OUT MAANDEN     INTEGER,
   OUT DAGEN       INTEGER)
BEGIN
  DECLARE VOLGENDE_DATUM, VORIGE_DATUM DATE;
  SET JAREN = 0;
  SET VORIGE_DATUM = START_DATUM;
  SET VOLGENDE_DATUM = START_DATUM + INTERVAL 1 YEAR;
  WHILE VOLGENDE_DATUM <= EIND_DATUM DO
    SET JAREN = JAREN + 1;
    SET VORIGE_DATUM = VOLGENDE_DATUM;
    SET VOLGENDE_DATUM = VOLGENDE_DATUM + INTERVAL 1 YEAR;
  END WHILE;
  SET MAANDEN = 0;
  SET VOLGENDE_DATUM = VORIGE_DATUM + INTERVAL 1 MONTH;
  WHILE VOLGENDE_DATUM <= EIND_DATUM DO
    SET MAANDEN = MAANDEN + 1;
    SET VORIGE_DATUM = VOLGENDE_DATUM;
    SET VOLGENDE_DATUM = VOLGENDE_DATUM + INTERVAL 1 MONTH;
  END WHILE;
  SET DAGEN = 0;
  SET VOLGENDE_DATUM = VORIGE_DATUM + INTERVAL 1 DAY;
```

```

    WHILE VOLGENDE_DATUM <= EIND_DATUM DO
        SET DAGEN = DAGEN + 1;
        SET VORIGE_DATUM = VOLGENDE_DATUM;
        SET VOLGENDE_DATUM = VOLGENDE_DATUM + INTERVAL 1 DAY;
    END WHILE;
END
;
SET @START = '1991-01-12'
;
SET @EIND = '1999-07-09'
;
CALL LEEFTIJD (@START, @EIND, @JAAR, @MAAND, @DAG)
;
SELECT @START, @EIND, @JAAR, @MAAND, @DAG

```

Voorbeeld 29.11:

```

CREATE PROCEDURE ERUIT
    (OUT P1 INTEGER, OUT P2 INTEGER)
BEGIN
    SET P1 = 1;
    SET P2 = 1;
    BLOK1 : BEGIN
        LEAVE BLOK1;
        SET P2 = 3;
    END;
    SET P1 = 4;
END

```

Voorbeeld 29.12:

```

CREATE PROCEDURE WACHTEN
    (IN WACHT_SECONDS INTEGER)
BEGIN
    DECLARE EIND_TIJD INTEGER
        DEFAULT NOW() + INTERVAL WACHT_SECONDS SECOND;
    WACHT_LOOP: LOOP
        IF NOW() > EIND_TIJD THEN
            LEAVE WACHT_LOOP;
        END IF;
    END LOOP WACHT_LOOP;
END

```

Voorbeeld 29.13:

```

CREATE PROCEDURE OPNIEUW
    (OUT RESULTAAT INTEGER)
BEGIN
    DECLARE TELLER INTEGER DEFAULT 1;
    SET RESULTAAT = 0;
    LOOP1: WHILE TELLER <= 1000 DO
        SET TELLER = TELLER + 1;
        IF TELLER > 100 THEN
            LEAVE LOOP1;
        ELSE
            ITERATE LOOP1;
        END IF;
        SET RESULTAAT = TELLER * 10;
    END WHILE LOOP1;
END

```

Voorbeeld 29.14:

```

CALL WACHTEN ((SELECT COUNT(*) FROM BOETES))
;
CREATE TABLE SPELERS_MET_OUDERS (
    SPELERSNR          INTEGER NOT NULL PRIMARY KEY,

```

```

VADER_SPELERSNR    INTEGER,
MOEDER_SPELERSNR  INTEGER)
;
ALTER TABLE SPELERS_MET_OUDERS ADD
  FOREIGN KEY (VADER_SPELERSNR)
    REFERENCES SPELERS_MET_OUDERS (SPELERSNR)
;
ALTER TABLE SPELERS_MET_OUDERS ADD
  FOREIGN KEY (MOEDER_SPELERSNR)
    REFERENCES SPELERS_MET_OUDERS (SPELERSNR)
;
INSERT INTO SPELERS_MET_OUDERS VALUES
  (9,NULL,NULL), (8,NULL,NULL), (7,NULL,NULL), (6,NULL,NULL),
  (5,NULL,NULL), (4,8,9), (3,6,7), (2,4,5), (1,2,3)

```

Voorbeeld 29.15:

```

CREATE PROCEDURE SOM_AANTAL_OUDERS
  (IN P_SPELERSNR INTEGER,
   INOUT AANTAL INTEGER)
BEGIN
  DECLARE V_VADER, V_MOEDER INTEGER;
  SET V_VADER =
    (SELECT  VADER_SPELERSNR
     FROM    SPELERS_MET_OUDERS
     WHERE   SPELERSNR = P_SPELERSNR);
  SET V_MOEDER =
    (SELECT  MOEDER_SPELERSNR
     FROM    SPELERS_MET_OUDERS
     WHERE   SPELERSNR = P_SPELERSNR);
  IF V_VADER IS NOT NULL THEN
    CALL SOM_AANTAL_OUDERS(V_VADER, AANTAL);
    SET AANTAL = AANTAL + 1;
  END IF;
  IF V_MOEDER IS NOT NULL THEN
    CALL SOM_AANTAL_OUDERS(V_MOEDER, AANTAL);
    SET AANTAL = AANTAL + 1;
  END IF;
END
;
SET @AANTAL = 0
;
CALL SOM_AANTAL_OUDERS (1, @AANTAL)
;
SELECT @AANTAL

```

Voorbeeld 29.16:

```

CREATE PROCEDURE SOM_BOETES_SPELER
  (IN P_SPELERSNR INTEGER,
   OUT SOM_BOETES DECIMAL(8,2))
BEGIN
  SELECT SUM(BEDRAG)
  INTO   SOM_BOETES
  FROM   BOETES
  WHERE  SPELERSNR = P_SPELERSNR;
END
;
CALL SOM_BOETES_SPELER(27, @SOM)
;
SELECT @SOM
;
SELECT  VADER_SPELERSNR, MOEDER_SPELERSNR
INTO    V_VADER, V_MOEDER
FROM    SPELERS_MET_OUDERS
WHERE   SPELERSNR = P_SPELERSNR

```

Voorbeeld 29.17:

```
CREATE PROCEDURE GEEF_ADRES
  (IN P_SPELERSNR INTEGER,
   OUT P_STRAAT   VARCHAR(30),
   OUT P_HUISNR   CHAR(4),
   OUT P_PLAATS   VARCHAR(30),
   OUT P_POSTCODE CHAR(6))
BEGIN
  SELECT PLAATS, STRAAT, HUISNR, POSTCODE
  INTO   P_PLAATS, P_STRAAT, P_HUISNR, P_POSTCODE
  FROM   SPELERS
  WHERE  SPELERSNR = P_SPELERSNR;
END
```

Voorbeeld 29.18:

```
CREATE TABLE FIBON (
  GETAL1 INTEGER NOT NULL PRIMARY KEY,
  GETAL2 INTEGER NOT NULL)
;
CREATE PROCEDURE FIBONACCI_START()
BEGIN
  DELETE FROM FIBON;
  INSERT INTO FIBON (GETAL1, GETAL2) VALUES (16, 27);
END
;
CREATE PROCEDURE FIBONACCI_GEEF
  (INOUT GETAL INTEGER)
BEGIN
  DECLARE G1, G2 INTEGER;
  SELECT GETAL1, GETAL2
  INTO   G1, G2
  FROM   FIBON;
  SET GETAL = G1 + G2;
  IF GETAL > 10000 THEN
    SET GETAL = GETAL - 10000;
  END IF;
  SET G1 = G2;
  SET G2 = GETAL;
  UPDATE FIBON
  SET   GETAL1 = G1,
        GETAL2 = G2;
END
;
CALL FIBONACCI_START()
;
CALL FIBONACCI_GEEF(@C)
;
SELECT @C
;
CALL FIBONACCI_GEEF(@C)
;
SELECT @C
;
CALL FIBONACCI_GEEF(@C)
;
SELECT @C
```

Voorbeeld 29.19:

```
CREATE PROCEDURE DELETE_SPELER
  (IN P_SPELERSNR INTEGER)
BEGIN
  DECLARE AANTAL_SPELERS INTEGER;
  DECLARE AANTAL_BOETES INTEGER;
  DECLARE AANTAL_TEAMS  INTEGER;
```

```

DECLARE AANTAL_LEDEN INTEGER;
SELECT COUNT(*)
INTO AANTAL_SPELERS
FROM SPELERS
WHERE SPELERSNR = P_SPELERSNR;
IF AANTAL_SPELERS = 0 THEN
  SELECT COUNT(*)
  INTO AANTAL_BOETES
  FROM BOETES
  WHERE SPELERSNR = P_SPELERSNR;
  SELECT COUNT(*)
  INTO AANTAL_TEAMS
  FROM TEAMS
  WHERE SPELERSNR = P_SPELERSNR;
  SELECT COUNT(*)
  INTO AANTAL_LEDEN
  FROM BESTUURSLEDEN
  WHERE SPELERSNR = P_SPELERSNR;
  IF AANTAL_BOETES = 0 AND
     AANTAL_TEAMS = 0 AND
     AANTAL_LEDEN = 0 THEN
    CALL DELETE_WEDSTRIJDEN(P_SPELERSNR);
    DELETE FROM SPELERS
    WHERE SPELERSNR = P_SPELERSNR;
  END IF;
END IF;
END

```

Voorbeeld 29.20:

```

CREATE PROCEDURE DUBBEL
  (OUT P_VERWERKT SMALLINT)
BEGIN
  SET P_VERWERKT = 1;
  INSERT INTO TEAMS VALUES (2,27,'derde');
  SET P_VERWERKT = 2;
END
;
CALL DUBBEL(VERWERKT)

```

Voorbeeld 29.21:

```

CREATE PROCEDURE FOUTJE1
  (OUT ERROR CHAR(5))
BEGIN
  DECLARE CONTINUE HANDLER FOR SQLSTATE '23000'
    SET ERROR = '23000';
  SET ERROR = '00000';
  INSERT INTO TEAMS VALUES (2,27,'derde');
END

```

Voorbeeld 29.22:

```

CREATE PROCEDURE FOUTJE2
  (OUT ERROR CHAR(5))
BEGIN
  DECLARE CONTINUE HANDLER FOR SQLSTATE '23000'
    SET ERROR = '23000';
  DECLARE CONTINUE HANDLER FOR SQLSTATE '21S01'
    SET ERROR = '21S01';
  SET ERROR = '00000';
  INSERT INTO TEAMS VALUES (2,27,'derde',5);
END

```

Voorbeeld 29.23:

```

CREATE PROCEDURE FOUTJE3

```

```

(OUT ERROR CHAR(5))
BEGIN
  DECLARE CONTINUE HANDLER FOR SQLWARNING, NOT FOUND,
    SQLEXCEPTION SET ERROR = 'XXXXX';
  SET ERROR = '00000';
  INSERT INTO TEAMS VALUES (2,27,'derde');
END

```

Voorbeeld 29.24:

```

CREATE PROCEDURE FOUTJE4
  (OUT ERROR CHAR(5))
BEGIN
  DECLARE NIET_UNIEK CONDITION FOR SQLSTATE '23000';
  DECLARE CONTINUE HANDLER FOR NIET_UNIEK
    SET ERROR = '23000';
  SET ERROR = '00000';
  INSERT INTO TEAMS VALUES (2,27,'derde');
END

```

Voorbeeld 29.25:

```

CREATE PROCEDURE FOUTJE5
  (OUT ERROR CHAR(5))
BEGIN
  DECLARE NIET_UNIEK CONDITION FOR SQLSTATE '23000';
  DECLARE CONTINUE HANDLER FOR NIET_UNIEK
    SET ERROR = '23000';
  BEGIN
    DECLARE CONTINUE HANDLER FOR NIET_UNIEK
      SET ERROR = '23000';
  END;
  BEGIN
    DECLARE CONTINUE HANDLER FOR NIET_UNIEK
      SET ERROR = '00000';
    INSERT INTO TEAMS VALUES (2,27,'derde');
  END;
END
;
CREATE PROCEDURE FOUTJE6 ()
BEGIN
  DECLARE CONTINUE HANDLER FOR SQLSTATE '23000'
    SET @VERWERKT = 100;
  BEGIN
    DECLARE CONTINUE HANDLER FOR SQLSTATE '23000'
      SET @VERWERKT = 200;
    INSERT INTO TEAMS VALUES (2,27,'derde');
  END;
END

```

Voorbeeld 29.26:

```

CREATE PROCEDURE AANTAL_SPELERS
  (OUT AANTAL INTEGER)
BEGIN
  DECLARE EEN_SPELERSNR INTEGER;
  DECLARE FOUND BOOLEAN DEFAULT TRUE;
  DECLARE C_SPELERS CURSOR FOR
    SELECT SPELERSNR FROM SPELERS;
  DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET FOUND = FALSE;
  SET AANTAL = 0;
  OPEN C_SPELERS;
  FETCH C_SPELERS INTO EEN_SPELERSNR;
  WHILE FOUND DO
    SET AANTAL = AANTAL + 1;
    FETCH C_SPELERS INTO EEN_SPELERSNR;
  END

```

```

    END WHILE;
    CLOSE C_SPELERS;
END

```

Voorbeeld 29.27:

```

CREATE PROCEDURE DELETE_OUDER_DAN_30()
BEGIN
    DECLARE V_LEEFTIJD, V_SPELERSNR, V_JAREN,
            V_MAANDEN, V_DAGEN INTEGER;
    DECLARE V_GEB_DATUM DATE;
    DECLARE FOUND BOOLEAN DEFAULT TRUE;
    DECLARE C_SPELERS CURSOR FOR
        SELECT SPELERSNR, GEB_DATUM
        FROM SPELERS;
    DECLARE CONTINUE HANDLER FOR NOT FOUND
        SET FOUND = FALSE;
    OPEN C_SPELERS;
    FETCH C_SPELERS INTO V_SPELERSNR, V_GEB_DATUM;
    WHILE FOUND DO
        CALL LEEFTIJD(V_GEB_DATUM, NOW(), V_JAREN,
                    V_MAANDEN, V_DAGEN);
        IF V_JAREN > 30 THEN
            DELETE FROM BOETES WHERE SPELERSNR = V_SPELERSNR;
        END IF;
        FETCH C_SPELERS INTO V_SPELERSNR, V_GEB_DATUM;
    END WHILE;
    CLOSE C_SPELERS;
END

```

Voorbeeld 29.28:

```

CREATE PROCEDURE TOP_DRIE
    (IN P_SPELERSNR INTEGER,
    OUT OK BOOLEAN)
BEGIN
    DECLARE EEN_SPELERSNR, SALDO, VOLGNR INTEGER;
    DECLARE FOUND BOOLEAN;
    DECLARE SALDO_SPELERS CURSOR FOR
        SELECT SPELERSNR, SUM(GEWONNEN) - SUM(VERLOREN)
        FROM WEDSTRIJDEN
        GROUP BY SPELERSNR
        ORDER BY 2;
    DECLARE CONTINUE HANDLER FOR NOT FOUND
        SET FOUND = FALSE;
    SET VOLGNR = 0;
    SET FOUND = TRUE;
    SET OK = FALSE;
    OPEN SALDO_SPELERS;
    FETCH SALDO_SPELERS INTO EEN_SPELERSNR, SALDO;
    WHILE FOUND AND VOLGNR < 3 AND OK = FALSE DO
        SET VOLGNR = VOLGNR + 1;
        IF EEN_SPELERSNR = P_SPELERSNR THEN
            SET OK = TRUE;
        END IF;
        FETCH SALDO_SPELERS INTO EEN_SPELERSNR, SALDO;
    END WHILE;
    CLOSE SALDO_SPELERS;
END

```

Voorbeeld 29.29:

```

CREATE PROCEDURE AANTAL_BOETES
    (IN V_SPELERSNR INTEGER,
    OUT AANTAL INTEGER)
BEGIN
    DECLARE EEN_SPELERSNR INTEGER;

```



```

DECLARE FOUND BOOLEAN DEFAULT TRUE;
DECLARE C_SPELERS CURSOR FOR
  SELECT SPELERSNR
  FROM   BOETES
  WHERE  SPELERSNR = V_SPELERSNR;
DECLARE CONTINUE HANDLER FOR NOT FOUND
  SET FOUND = FALSE;
SET AANTAL = 0;
OPEN C_SPELERS;
FETCH C_SPELERS INTO EEN_SPELERSNR;
WHILE FOUND DO
  SET AANTAL = AANTAL + 1;
  FETCH C_SPELERS INTO EEN_SPELERSNR;
END WHILE;
CLOSE C_SPELERS;
END

```

Voorbeeld 29.30:

```

SELECT COLUMN_NAME
FROM   INFORMATION_SCHEMA.COLUMNS
WHERE  TABLE_SCHEMA = 'INFORMATION_SCHEMA'
AND    TABLE_NAME = 'ROUTINES'
ORDER BY ORDINAL_POSITION

```

Voorbeeld 29.31:

```

DROP PROCEDURE DELETE_SPELER

```

Voorbeeld 29.33:

```

ALTER PROCEDURE DELETE_WEDSTRIJDEN COMPILE

```

Voorbeeld 29.34:

```

GRANT EXECUTE
ON   DELETE_WEDSTRIJDEN
TO   JOHN

```

4.27 SQL-instructies voor hoofdstuk 30

Voorbeeld 30.1:

```

CREATE FUNCTION DOLLARS(BEDRAG DECIMAL(7,2))
  RETURNS DECIMAL(7,2)
BEGIN
  RETURN BEDRAG * (1 / 0.8);
END
;
SELECT BETALINGSNR, BEDRAG, DOLLARS(BEDRAG)
FROM   BOETES
WHERE  BETALINGSNR <= 3

```

Voorbeeld 30.2:

```

CREATE FUNCTION AANTAL_SPELERS()
  RETURNS INTEGER
BEGIN
  RETURN (SELECT COUNT(*) FROM SPELERS);
END
;
SELECT AANTAL_SPELERS()

```

Voorbeeld 30.3:

```

CREATE FUNCTION AANTAL_BOETES
(P_SPELERSNR INTEGER)
RETURNS INTEGER
BEGIN
    RETURN (SELECT    COUNT(*)
            FROM      BOETES
            WHERE     SPELERSNR = P_SPELERSNR);
END
;
CREATE FUNCTION AANTAL_WEDSTRIJDEN
(P_SPELERSNR INTEGER)
RETURNS INTEGER
BEGIN
    RETURN (SELECT    COUNT(*)
            FROM      WEDSTRIJDEN
            WHERE     SPELERSNR = P_SPELERSNR);
END
;
SELECT  SPELERSNR, NAAM, VOORLETTERS
FROM    SPELERS
WHERE   AANTAL_BOETES(SPELERSNR) >
        AANTAL_WEDSTRIJDEN(SPELERSNR)

```

Voorbeeld 30.4:

```

CREATE FUNCTION AANTAL_DAGEN
(START_DATUM DATE,
 EIND_DATUM DATE)
RETURNS INTEGER
BEGIN
    DECLARE DAGEN INTEGER;
    DECLARE VOLGENDE_DATUM, VORIGE_DATUM DATE;
    SET DAGEN = 0;
    SET VOLGENDE_DATUM = START_DATUM + INTERVAL 1 DAY;
    WHILE VOLGENDE_DATUM <= EIND_DATUM DO
        SET DAGEN = DAGEN + 1;
        SET VORIGE_DATUM = VOLGENDE_DATUM;
        SET VOLGENDE_DATUM = VOLGENDE_DATUM + INTERVAL 1 DAY;
    END WHILE;
    RETURN DAGEN;
END

```

Voorbeeld 30.5:

```

CREATE FUNCTION DELETE_SPELER
(P_SPELERSNR INTEGER)
RETURNS BOOLEAN
BEGIN
    DECLARE AANTAL_SPELERS INTEGER;
    DECLARE AANTAL_BOETES INTEGER;
    DECLARE AANTAL_TEAMS INTEGER;
    DECLARE AANTAL_LEDEN INTEGER;
    DECLARE EXIT HANDLER FOR SQLWARNING RETURN FALSE;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION RETURN FALSE;
    SELECT COUNT(*)
    INTO AANTAL_SPELERS
    FROM SPELERS
    WHERE SPELERSNR = P_SPELERSNR;
    IF AANTAL_SPELERS = 0 THEN
        SELECT COUNT(*)
        INTO AANTAL_BOETES
        FROM BOETES
        WHERE SPELERSNR = P_SPELERSNR;
        SELECT COUNT(*)
        INTO AANTAL_TEAMS
        FROM TEAMS

```

```

WHERE SPELERSNR = P_SPELERSNR;
SELECT COUNT(*)
INTO AANTAL_LEDEN
FROM BESTUURSLEDEN
WHERE SPELERSNR = P_SPELERSNR;
IF AANTAL_BOETES = 0 AND
   AANTAL_TEAMS = 0 AND
   AANTAL_LEDEN = 0 THEN
   CALL DELETE_WEDSTRIJDEN(P_SPELERSNR);
   DELETE FROM SPELERS
   WHERE SPELERSNR = P_SPELERSNR;
END IF;
END IF;
RETURN TRUE;
END

```

Voorbeeld 30.6:

```

CREATE FUNCTION GEEF_AANTAL_SPELERS()
  RETURNS INTEGER
BEGIN
  DECLARE AANTAL INTEGER;
  CALL AANTAL_SPELERS(AANTAL);
  RETURN AANTAL;
END

```

Voorbeeld 30.7:

```

CREATE FUNCTION OVERLAP_TUSSEN_PERIODES
  (PERIODE1_START DATETIME,
   PERIODE1_EIND DATETIME,
   PERIODE2_START DATETIME,
   PERIODE2_EIND DATETIME)
  RETURNS BOOLEAN
BEGIN
  DECLARE TIJDELIJKE_DATUM DATETIME;
  IF PERIODE1_START > PERIODE1_EIND THEN
    SET TIJDELIJKE_DATUM = PERIODE1_START;
    SET PERIODE1_START = PERIODE1_EIND;
    SET PERIODE1_EIND = TIJDELIJKE_DATUM;
  END IF;
  IF PERIODE2_START > PERIODE2_EIND THEN
    SET TIJDELIJKE_DATUM = PERIODE2_START;
    SET PERIODE2_START = PERIODE2_EIND;
    SET PERIODE2_EIND = TIJDELIJKE_DATUM;
  END IF;
  RETURN NOT(PERIODE1_EIND < PERIODE2_START OR
             PERIODE2_EIND < PERIODE1_START);
END

```

Voorbeeld 30.8:

```

SELECT *
FROM BESTUURSLEDEN
WHERE OVERLAP_TUSSEN_PERIODES(BEGIN_DATUM,EIND_DATUM,
                              '1991-06-30','1992-06-30')
ORDER BY 1, 2

```

Voorbeeld 30.9:

```

DROP FUNCTION AANTAL_SPELERS

```

4.28 SQL-instructies voor hoofdstuk 31

Voorbeeld 31.1:

```
CREATE TABLE MUTATIES (
  MUT_VOLGNR          INTEGER UNSIGNED AUTO_INCREMENT
                    NOT NULL PRIMARY KEY,
  MUT_GEBRUIKER      CHAR(30) NOT NULL,
  MUT_TIJDSTIP       TIMESTAMP NOT NULL,
  MUT_SPELERSNR      INTEGER NOT NULL,
  MUT_TYPE           CHAR(1) NOT NULL,
  MUT_BONDSNR_NEW    CHAR(4),
  MUT_BONDSNR_OLD    CHAR(4))
```

Voorbeeld 31.2:

```
CREATE TRIGGER INSERT_SPELER
  AFTER
  INSERT ON SPELERS FOR EACH ROW
  BEGIN
    INSERT INTO MUTATIES
      (MUT_GEBRUIKER, MUT_TIJDSTIP, MUT_SPELERSNR, MUT_TYPE, MUT_SPELERSNR_NEW)
    VALUES (CURRENT_USER, CURRENT_DATE, NEW.SPELERSNR, 'I', NEW.BONDSNR, NULL);
  END
;
INSERT INTO SPELERS VALUES
  (2000, 'Brant', 'K', '1959-06-25', 'M', 1978,
   'Moskoulaan', '80', '5674BF', 'Den Haag', '070-346734', '6377')
;
SELECT  MUT_VOLGNR AS VOLGNR, MUT_GEBRUIKER AS GEBRUIKER,
        MUT_SPELERSNR AS SPELERSNR, MUT_TYPE AS TYPE,
        MUT_BONDSNR_NEW AS BNEW,
        MUT_BONDSNR_OLD AS BOLD
FROM    MUTATIES
WHERE   MUT_GEBRUIKER = CURRENT_USER
AND     MUT_SPELERSNR = 2000
AND     MUT_TIJDSTIP =
        (SELECT  MAX(MUT_TIJDSTIP)
         FROM    MUTATIES
         WHERE   MUT_GEBRUIKER = CURRENT_USER
         AND     MUT_SPELERSNR = 2000)
;
CREATE PROCEDURE INSERT_MUTATIE
  (IN MSNR      INTEGER,
   IN MTYPE     CHAR(1),
   IN MBOND_NEW CHAR(4),
   IN MBOND_OLD CHAR(4))
BEGIN
  INSERT INTO MUTATIES
    (MUT_GEBRUIKER, MUT_TIJDSTIP, MUT_SPELERSNR,
     MUT_TYPE, MUT_BONDSNR_NEW, MUT_BONDSNR_OLD)
  VALUES (CURRENT_USER, CURRENT_DATE, MSNR, MTYPE,
          MBOND_NEW, MBOND_OLD);
END
;
CREATE TRIGGER INSERT_SPELER
  AFTER INSERT ON SPELERS FOR EACH ROW
  BEGIN
    CALL INSERT_MUTATIE (NEW.SPELERSNR, 'I',
                       NEW.BONDSNR, NULL);
  END
```

Voorbeeld 31.3:

```
CREATE TRIGGER DELETE_SPELERS
  AFTER DELETE ON SPELERS FOR EACH ROW
  BEGIN
    CALL INSERT_MUTATIE (OLD.SPELERSNR, 'D',
                       NULL, OLD.BONDSNR);
```

END

Voorbeeld 31.4:

```
DELETE
FROM   SPELERS
WHERE  SPELERSNR = 2000
;
SELECT MUT_VOLGNER AS VOLGNER, MUT_GEBRUIKER AS GEBRUIKER,
       MUT_SPELERSNR AS SPELERSNR, MUT_TYPE AS TYPE,
       MUT_BONDSNR_NEW AS BNEW,
       MUT_BONDSNR_OLD AS BOLD
FROM   MUTATIES
WHERE  MUT_GEBRUIKER = CURRENT_USER
AND    MUT_SPELERSNR = 2000
AND    MUT_TIJDSTIP =
      (SELECT MAX(MUT_TIJDSTIP)
       FROM   MUTATIES
       WHERE  MUT_GEBRUIKER = CURRENT_USER
       AND    MUT_SPELERSNR = 2000)
```

Voorbeeld 31.5:

```
CREATE TRIGGER UPDATE_SPELERS
AFTER UPDATE ON SPELERS FOR EACH ROW
BEGIN
  CALL INSERT_MUTATIE
    (OLD.SPELERSNR, 'U', NEW.BONDSNR, OLD.BONDSNR);
END
```

Voorbeeld 31.6:

```
UPDATE SPELERS
SET    BONDSNR = '4444'
WHERE  SPELERSNR = 6
;
SELECT MUT_VOLGNER AS VOLGNER, MUT_GEBRUIKER AS GEBRUIKER,
       MUT_SPELERSNR AS SPELERSNR, MUT_TYPE AS TYPE,
       MUT_BONDSNR_NEW AS BNEW,
       MUT_BONDSNR_OLD AS BOLD
FROM   MUTATIES
WHERE  MUT_GEBRUIKER = CURRENT_USER
AND    MUT_SPELERSNR = 6
AND    MUT_TIJDSTIP =
      (SELECT MAX(MUT_TIJDSTIP)
       FROM   MUTATIES
       WHERE  MUT_GEBRUIKER = CURRENT_USER
       AND    MUT_SPELERSNR = 6)
```

Voorbeeld 31.7:

```
CREATE TRIGGER UPDATE_SPELERS2
AFTER UPDATE(BONDSNR) ON SPELERS FOR EACH ROW
BEGIN
  CALL INSERT_MUTATIE
    (OLD.SPELERSNR, 'U', NEW.BONDSNR, OLD.BONDSNR);
END
```

Voorbeeld 31.8:

```
CREATE TRIGGER UPDATE_SPELERS3
AFTER UPDATE(BONDSNR) ON SPELERS FOR EACH ROW
WHEN ( OLD.SPELERSNR > 100 )
BEGIN
  CALL INSERT_MUTATIE
    (OLD.SPELERSNR, 'U', NEW.BONDSNR, OLD.BONDSNR);
END
```

Voorbeeld 31.9:

```
CREATE TABLE SPELERS_WED (
    SPELERSNR          INTEGER NOT NULL PRIMARY KEY,
    AANTAL_WEDSTRIJDEN INTEGER NOT NULL)
;
INSERT INTO SPELERS_WED (SPELERSNR, AANTAL_WEDSTRIJDEN)
SELECT  SPELERSNR,
        (SELECT  COUNT(*)
         FROM    WEDSTRIJDEN AS W
         WHERE   S.SPELERSNR = W.SPELERSNR)
FROM    SPELERS AS S
```

Voorbeeld 31.10:

```
CREATE TRIGGER INSERT_SPELERS_WED
AFTER INSERT ON SPELERS FOR EACH ROW
BEGIN
    INSERT INTO SPELERS_WED
    VALUES(NEW.SPELERSNR, 0);
END
```

Voorbeeld 31.11:

```
INSERT INTO SPELERS VALUES
(16, 'Permentier', 'S', '1964-06-25', 'F', 1977, 'Hazensteinln',
'80', '1234KK', 'Den Haag', '070-476537', '8467')
;
SELECT *
FROM    SPELERS_WED
WHERE   SPELERSNR = 16
```

Voorbeeld 31.12:

```
CREATE TRIGGER DELETE_SPELERS
AFTER DELETE ON SPELERS FOR EACH ROW
BEGIN
    DELETE FROM SPELERS_WED
    WHERE SPELERSNR = OLD.SPELERSNR;
END
```

Voorbeeld 31.13:

```
CREATE TRIGGER INSERT_WEDSTRIJDEN
AFTER INSERT ON WEDSTRIJDEN FOR EACH ROW
BEGIN
    UPDATE SPELERS_WED
    SET    AANTAL_WEDSTRIJDEN = AANTAL_WEDSTRIJDEN + 1
    WHERE SPELERSNR = NEW.SPELERSNR;
END
```

Voorbeeld 31.14:

```
CREATE TRIGGER DELETE_WEDSTRIJDEN
AFTER DELETE ON WEDSTRIJDEN FOR EACH ROW
BEGIN
    UPDATE SPELERS_WED
    SET    AANTAL_WEDSTRIJDEN = AANTAL_WEDSTRIJDEN - 1
    WHERE SPELERSNR = OLD.SPELERSNR;
END
```

Voorbeeld 31.15:

```
CREATE TRIGGER SOM_BOETES_INSERT
AFTER INSERT, UPDATE ON BOETES FOR EACH ROW
BEGIN
```

```

    DECLARE SOM DECIMAL(8,2);
    SELECT    SUM(BEDRAG)
    INTO      SOM
    FROM      BOETES
    WHERE     SPELERSNR = NEW.SPELERSNR;
    UPDATE    SPELERS
    SET       SOM_BOETES = SOM
    WHERE     SPELERSNR = NEW.SPELERSNR
END
;
CREATE TRIGGER SOM_BOETES_DELETE
AFTER DELETE, UPDATE ON BOETES FOR EACH ROW
BEGIN
    DECLARE SOM DECIMAL(8,2);
    SELECT    SUM(BEDRAG)
    INTO      SOM
    FROM      BOETES
    WHERE     SPELERSNR = OLD.SPELERSNR;
    UPDATE    SPELERS
    SET       SOM_BOETES = SOM
    WHERE     SPELERSNR = OLD.SPELERSNR
END
;
UPDATE    SPELERS
SET       SOM_BOETES = (SELECT    SUM(BEDRAG)
                        FROM      BOETES
                        WHERE     SPELERSNR = NEW.SPELERSNR)
WHERE     SPELERSNR = NEW.SPELERSNR

```

Voorbeeld 31.16:

```

CREATE TRIGGER GEBJAARTOE
BEFORE INSERT, UPDATE ON SPELERS FOR EACH ROW
BEGIN
    IF YEAR(NEW.GEB_DATUM) >= NEW.JAARTOE) THEN
        ROLLBACK WORK;
    END IF;
END

```

Voorbeeld 31.17:

```

CREATE TRIGGER REF_SLEUTEL1
BEFORE INSERT, UPDATE ON BOETES FOR EACH ROW
BEGIN
    IF (SELECT COUNT(*) FROM SPELERS
        WHERE SPELERSNR = NEW.SPELERSNR) = 0 THEN
        ROLLBACK WORK;
    END IF;
END
;
CREATE TRIGGER REF_SLEUTEL2
BEFORE DELETE, UPDATE ON SPELERS FOR EACH ROW
BEGIN
    DELETE
    FROM    BOETES
    WHERE   SPELERSNR = OLD.SPELERSNR;
END

```

Voorbeeld 31.18:

```
DROP TRIGGER GEBJAARTOE
```

4.29 SQL-instructies voor hoofdstuk 33

```
SET @@AUTOCOMMIT = 1
;
SET @@AUTOCOMMIT = 0
```

Voorbeeld 33.1:

```
DELETE
FROM   BOETES
WHERE  SPELERSNR = 44
;
SELECT *
FROM   BOETES
;
ROLLBACK WORK
;
COMMIT WORK
```

Voorbeeld 33.3:

```
DELETE FROM SPELERS WHERE SPELERSNR = 6
;
DELETE FROM BOETES WHERE SPELERSNR = 6
;
DELETE FROM WEDSTRIJDEN WHERE SPELERSNR = 6
;
DELETE FROM BESTUURSLEDEN WHERE SPELERSNR = 6
;
UPDATE TEAMS SET SPELERSNR = 83 WHERE SPELERSNR = 6
```

Voorbeeld 33.6:

```
CREATE PROCEDURE NIEUW_TEAM ()
BEGIN
    INSERT INTO TEAMS VALUES (100,27,'ere');
END
;
SET AUTOCOMMIT = 1
;
START TRANSACTION
;
INSERT INTO TEAMS VALUES (200,27,'ere')
;
CALL NIEUW_TEAM()
;
ROLLBACK WORK
```

Voorbeeld 33.7:

```
UPDATE BOETES
SET    BEDRAG = BEDRAG + 25
WHERE  BETALINGSNR = 4
;
SELECT *
FROM   BOETES
WHERE  BETALINGSNR = 4
```

Voorbeeld 33.8:

```
SELECT SPELERSNR
FROM   SPELERS
WHERE  PLAATS = 'Den Haag'
;
UPDATE SPELERS
SET    PLAATS = 'Zoetermeer'
WHERE  SPELERSNR = 7
;
SELECT SPELERSNR, NAAM, VOORLETTERS, STRAAT,
```



```

    HUISNR, POSTCODE, PLAATS
FROM   SPELERS
WHERE  SPELERSNR IN (6, 83, 2, 7, 57, 39, 100)

```

Voorbeeld 33.9:

```

SELECT  SPELERSNR
FROM    SPELERS
WHERE   PLAATS = 'Den Haag'

```

Voorbeeld 33.10:

```

UPDATE  BOETES
SET     BEDRAG = BEDRAG + 25
WHERE   BETALINGSNR = 4
;
UPDATE  BOETES
SET     BEDRAG = BEDRAG + 30
WHERE   BETALINGSNR = 4

```

Voorbeeld 33.11:

```

ALTER TABLE BOETES LOCKSIZE ROW

```

Voorbeeld 33.12:

```

LOCK TABLE SPELERS IN SHARE MODE
;
LOCK TABLE SPELERS READ

```

4.30 SQL-instructies voor hoofdstuk 34

Voorbeeld 34.1:

```

SELECT  NAAM, VOORLETTERS
FROM    SPELERS
WHERE   SPELERSNR = 6
OR      SPELERSNR = 83
OR      SPELERSNR = 44
;
SELECT  NAAM, VOORLETTERS
FROM    SPELERS
WHERE   SPELERSNR IN (6, 83, 44)

```

Voorbeeld 34.2:

```

SELECT  *
FROM    SPELERS
WHERE   JAARTOE = 1980
OR      PLAATS = 'Den Haag'
;
SELECT  *
FROM    SPELERS
WHERE   JAARTOE = 1980
UNION
SELECT  *
FROM    SPELERS
WHERE   PLAATS = 'Den Haag'

```

Voorbeeld 34.3:

```

UPDATE  BOETES
SET     BEDRAG = 150
WHERE   BEDRAG = 100

```

```

OR      DATUM = '1980-12-01'
;
UPDATE  BOETES
SET     BEDRAG = 150
WHERE   BEDRAG = 100
;
UPDATE  BOETES
SET     BEDRAG = 150
WHERE   DATUM = '1980-12-01'
;
SELECT  NAAM
FROM    SPELERS
WHERE   JAARTOE = 1980
OR      PLAATS = 'Den Haag'

```

Voorbeeld 34.4:

```

SELECT  WEDSTRIJDNR, GEWONNEN - VERLOREN
FROM    WEDSTRIJDEN
WHERE   GEWONNEN >= VERLOREN
UNION
SELECT  WEDSTRIJDNR, VERLOREN - VERLOREN
FROM    WEDSTRIJDEN
WHERE   GEWONNEN < VERLOREN
;
SELECT  WEDSTRIJDNR, ABS(GEWONNEN - VERLOREN)
FROM    WEDSTRIJDEN

```

Voorbeeld 34.5:

```

SELECT  *
FROM    SPELERS
WHERE   NOT (JAARTOE > 1980)

```

Voorbeeld 34.6:

```

SELECT  *
FROM    SPELERS
WHERE   NOT (GESLACHT = 'M')
;
SELECT  *
FROM    SPELERS
WHERE   GESLACHT = 'V'

```

Voorbeeld 34.7:

```

SELECT  *
FROM    SPELERS
WHERE   JAARTOE + 10 = 1990
;
SELECT  *
FROM    SPELERS
WHERE   JAARTOE = 1980

```

Voorbeeld 34.8:

```

SELECT  SPELERSNR
FROM    SPELERS
WHERE   GEB_DATUM >= '1962-01-01'
AND     GEB_DATUM <= '1965-12-31'
;
SELECT  SPELERSNR
FROM    SPELERS
WHERE   GEB_DATUM BETWEEN '1962-01-01' AND '1965-12-31'

```

Voorbeeld 34.9:

```
SELECT *
FROM SPELERS
WHERE NAAM LIKE '%n'
```

Voorbeeld 34.10:

```
SELECT BETALINGSNR, NAAM
FROM BOETES AS BT, SPELERS AS S
WHERE BT.SPELERSNR = S.SPELERSNR
AND BT.SPELERSNR = 44
;
SELECT BETALINGSNR, NAAM
FROM BOETES AS BT, SPELERS AS S
WHERE BT.SPELERSNR = S.SPELERSNR
AND BT.SPELERSNR = 44
AND S.SPELERSNR = 44
```

Voorbeeld 34.11:

```
SELECT SPELERSNR, COUNT(*)
FROM BOETES
GROUP BY SPELERSNR
HAVING SPELERSNR >= 40
;
SELECT SPELERSNR, COUNT(*)
FROM BOETES
WHERE SPELERSNR >= 40
GROUP BY SPELERSNR
```

Voorbeeld 34.12:

```
SELECT SPELERSNR, NAAM
FROM SPELERS
WHERE EXISTS
  (SELECT '1'
   FROM BOETES
   WHERE BOETES.SPELERSNR = SPELERS.SPELERSNR)
```

Voorbeeld 34.13:

```
SELECT DISTINCT WEDSTRIJDNR, NAAM
FROM WEDSTRIJDEN, SPELERS
WHERE WEDSTRIJDEN.SPELERSNR = SPELERS.SPELERSNR
```

Voorbeeld 34.14:

```
SELECT NAAM, VOORLETTERS
FROM SPELERS
WHERE PLAATS = 'Den Haag'
UNION ALL
SELECT NAAM, VOORLETTERS
FROM SPELERS
WHERE PLAATS = 'Voorburg'
```

Voorbeeld 34.15:

```
SELECT SPELERS.SPELERSNR, NAAM, BEDRAG
FROM SPELERS, BOETES
WHERE SPELERS.SPELERSNR = BOETES.SPELERSNR
UNION
SELECT SPELERSNR, NAAM, NULL
FROM SPELERS
WHERE SPELERSNR NOT IN
  (SELECT SPELERSNR
   FROM BOETES)
ORDER BY SPELERSNR
;
```

```

SELECT  SPELERSNR, NAAM, BEDRAG
FROM    SPELERS LEFT OUTER JOIN BOETES
        USING (SPELERSNR)
ORDER BY SPELERSNR

```

Voorbeeld 34.16:

```

SELECT  SPELERSNR, NAAM, GEB_DATUM
FROM    SPELERS
WHERE   GEB_DATUM <= ALL
        (SELECT  GEB_DATUM
         FROM    SPELERS)
;
SELECT  SPELERSNR, NAAM, GEB_DATUM
FROM    SPELERS
WHERE   GEB_DATUM =
        (SELECT  MIN(GEB_DATUM)
         FROM    SPELERS)

```

Voorbeeld 34.17:

```

SELECT  SPELERSNR, NAAM, GEB_DATUM
FROM    SPELERS
WHERE   GEB_DATUM > ANY
        (SELECT  GEB_DATUM
         FROM    SPELERS)
;
SELECT  SPELERSNR, NAAM, GEB_DATUM
FROM    SPELERS
WHERE   GEB_DATUM >
        (SELECT  MIN(GEB_DATUM)
         FROM    SPELERS)

```

De Auteur

Rick F. van der Lans is auteur van vele boeken over SQL. Naast dit SQL Leerboek dat in diverse talen vertaald is, waaronder Engels, Duits, Chinees en Italiaans, heeft hij SQL boeken geschreven voor producten als MySQL, Oracle, SQLite, Ingres en Pervasive PSQL.



Hij is onafhankelijk adviesur, auteur en docent gespecialiseerd in databasetechnologie, datawarehousing en applicatie-integratie. Hij is oprichter en directeur van R20/Consultancy. Door de jaren heen heeft hij veel organisaties geadviseerd op het gebied van IT-architecturen.

Als spreker op conferenties en seminars wordt hij internationaal gerespecteerd. Al meer dan vijftig jaar geeft hij over de gehele wereld lezingen, inclusief in de meeste Europese landen, Noord- en Zuid-Amerika en Australië. Hij is voorzitter van het jaarlijkse European Data Warehouse and Business Intelligence Conference. Hij schrijft een column voor Database Magazine en voor het internationale Beye-Network.com. Zeven jaar lang was hij lid van de Nederlandse ISO commissie verantwoordelijk voor ISO SQL Standaard.

Rick kan via de volgende kanalen bereikt worden:

Email: rick@r20.nl
Twitter: http://twitter.com/Rick_vanderlans
LinkedIn: <http://www.linkedin.com/pub/rick-van-der-lans/9/207/223>

Cursussen over de volgende onderwerpen kunnen door Rick F. van der Lans verzorgd worden

- Database-ontwerp en informatiemodellering
- De basis van SQL
- Het ontwikkelen van geavanceerde SQL queries
- Datawarehousing en business intelligence
- Data virtualisatie

Andere boeken geschreven door Rick F. van der Lans

