# 2 Database school

Leerdoelen

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

* een database kunt aanmaken
* een database kunt gebruiken
* een tabel kunt aanmaken
* een tabel kunt vullen met gegevens
* weet wat records zijn
* weet wat velden zijn
* weet wat het verschil is tussen waarden en metagegevens

Database aanmaken

Een database aanmaken in MySQL gaat heel eenvoudig. Het enige wat we moeten weten, is welke naam we de database geven. Omdat onze database over een school gaat, noemen we de database school. Achter de prompt in de Command Line Client typen we:

**create database school;**

Het DBMS antwoordt met:

**Query OK, 1 row affected (0.04 sec)**

Query OK betekent dat de query die de database heeft ontvangen in orde is en dus uitgevoerd is. Anders antwoordt de database-server met ERROR.1 row affected betekent dat de opdracht is uitgevoerd en dat er een regel aan een tabel is toegevoegd. Daarachter staat de duur in seconden van de uitvoering van de opdracht.

Opdracht 2.1

Controleer of de database school is aangemaakt.

Database gebruiken

Nu moeten we aan het DBMS vertellen welke database we gaan gebruiken. Dat doen we als volgt:

mysql>**use school;**

De database-server antwoordt met: **Database changed**. Dat wil zeggen dat het DBMS nu weet in welke database wij als gebruiker willen gaan werken.

*Opdracht 2.2*

Zorg ervoor dat je in de database school kunt werken.

Tabel aanmaken

Een school bestaat uit verscheidene eenheden: Entiteiten genaamd. Deze entiteiten zijn onder andere: studenten, docenten, opleidingen, resultaten die studenten hebben behaald. We beginnen met een tabel: studenten.

Een tabel aanmaken doen we met de opdracht **create table**. Daarachter zetten we de naam van de tabel. Deze naam is een zelfstandig naamwoord in het enkelvoud. Omdat in deze tabel de gegevens van studenten komen te staan, noemen we deze **student**. Vervolgens geven we aan welke gegevens er bewaard worden over studenten zoals naam en adres, en wat voor soort gegevens dat zijn:

**create table student (**

**studentnr varchar(5),**

**roepnaam varchar(10),**

**voorletters varchar(10),**

**tussenvoegsels varchar(8),**

**achternaam varchar(25),**

**adres varchar(25),**

**postcode varchar(7),**

**woonplaats varchar(25),**

**geslacht char(1),**

**telefoon varchar(11),**

**geboortedatum date,**

**uitgeschreven date,**

**schoolgeld decimal(6,2),**

**betaald decimal(6,2),**

**primary key(studentnr)**

**) engine = innodb;**

In de opdracht tussen haakjes zie je de naam van het gegeven staan (studentnr, roepnaam, voorletters enz.). Dat wordt in databasetermen een veldnaam genoemd. Een veldnaam is ook een zelfstandig naamwoord in het enkelvoud. Daarachter staat om wat voor soort gegeven het gaat (varchar, date en decimal). Deze worden datatypes genoemd.

Varchar is een reeks willekeurige tekens van een bepaalde maximale lengte. In programmeertermen wordt dit een string genoemd. Achter varchar staat tussen haakjes uit hoeveel tekens een veld maximaal mag bestaan. Minder mag wel, meer mag niet. Dan geeft de database een foutmelding. Bij geslacht staat als datatype char(1). Dat betekent dat ervan uit gegaan wordt dat dit veld altijd uit een teken bestaat. Meer mag niet, minder mag wel maar dat wordt dan aangevuld met spaties. De inhoud van dit veld wordt omsloten door apostrofes (‘) of door aanhalingstekens(”).

Date is een datum. Een database onthoudt de datum in een bepaald formaat en er hoeft dus geen maximale lengte bepaald te worden. De inhoud van dit veld wordt ook omsloten door apostrofes (‘) of door aanhalingstekens(”).

Decimal is een niet geheel getal, dus een getal met cijfers achter de komma. Het eerste getal achter decimal geeft de precisie weer, dat zijn alle cijfers waaruit het getal bestaat. Het tweede cijfer geeft de schaal weer, het aantal cijfers achter de komma. Decimal(6,2) is dus een getal dat in totaal bestaat uit 6 cijfers: 4 cijfers voor de komma en 2 cijfers achter de komma.

Velden worden van elkaar gescheiden door een komma behalve bij de laatste.

De laatste twee regels (**primary key(studentnr)** en **engine = innodb**) worden later behandeld.

Dit is een SQL-opdracht. Elke SQL-opdracht wordt eindigt met een punt-komma (;).

Als je deze opdracht goed invoert, reageert de database met: **Query OK, 0 rows affected (0.28 sec)**

*Opdracht 2.3*

Maak de tabel student aan en controleer met behulp van de opdracht **show tables;** of de tabel is aangemaakt.

Tot nu toe hebben we alleen gesproken over gegevens die over gegevens gaan. Die noemen we metagegevens: Gegevens over gegevens. Het deel van SQL dat over metagegevens gaat wordt ook wel data definition language (DDL) genoemd.

Het wordt tijd om de ‘echte’ gegevens in de velden in te voeren: de waarden in de velden. Daarvoor gebruiken we de SQL-opdracht **insert into**. Daarachter wordt de naam van de tabel genoemd waar de gegevens in bewaard moeten worden. Vervolgens komt er tussen haakjes een lijst met veldnamen, daarna volgt **values** en dan ook tussen haakjes in dezelfde volgorde als de veldnamen de waarden in de velden:

**insert into student**

**(studentnr, roepnaam, voorletters, tussenvoegsels, achternaam, adres, postcode, woonplaats, geboortedatumschoolgeld, betaald)**

**values**

**('1111', 'Piet', 'P.', '', 'Pietersen', 'Pietstraat 2', '1111 PP', 'Almere', ‘m’, '1995-2-20'500, 500);**

We hebben in bovenstaande query niet de waarden voor telefoon en geslacht ingevuld.

We hebben nu een record (dat zijn de gegevens van een student) gevuld. Met de volgende opdracht, waarin we de records met komma’s scheiden, kunnen we meerdere records tegelijk invoeren:

**insert into student**

**(studentnr, roepnaam, voorletters, tussenvoegsels, achternaam, adres, postcode, woonplaats, geslacht, telefoon, geboortedatumschoolgeld, betaald)**

**values**

**('2222', 'Jan', 'J.', null, 'Jansen', 'Janstraat 34', '2222 JJ', 'Almere', ‘m’, ‘036-9876543’, '1995-11-20', 500, 500);**

Bij het veld tussenvoegsel bij student 1111 zien we dat er niets staat tussen de aanhalingtekens. Dat betekent dat de inhoud van het veld leeg blijft. De emmer blijft leeg. Bij student 2222 staat in het veld tussenvoegsels null. Dat betekent dat er geen waarde wordt toegekend aan dat veld. Met andere woorden: de emmer bestaat niet.

Bij het veld geboortedatum staat eerst het jaartal, dan het maandnummer en dan het dagnummer (yyyy-mm-dd). Dat is de manier waarop een datum in MySQL wordt ingevoerd. Daar komen we later nog op terug.

*Opdracht 2.4*

Vul de tabel met de volgende gegevens van studenten:

1111, Piet, P. Pietersen, Pietstraat 2, 1111 PP, Almere, man, 036-1234567, 20-2-1995, € 500,-, € 500,-

2222, Jan, J. Jansen, Janstraat 34, 2222 JJ, Almere, man, 036-9876543,20-11-1995, € 500,-, € 500,-

6666, Angelique, A de Hoed, Hoedenstraat 12, 1663 LW, Lelystad, vrouw, 0320-654378, 6-6-1998, € 200,-, € 100,-

3333, Willem, W. Willemsen, Willemstraat 45, 3333 WW, Almere, 2-10-1998, € 200,-, € 200,-

4444, Berend, B. van der Tol, Tolweg 10, 4444 WE Lelystad, man, 3-4-1994, € 500,-, € 0,-

5555, Cynthia, C de Bruin, Almerestraat 10, 1553 CW, Zeewolde, vrouw, 06-65789431, 1996-5-5, € 500,-, € 300,-

SQL-instructies waarmee gegevens in een record geraadpleegd of gewijzigd kunnen worden, wordt Data Manipulation Language (DML) genoemd.

Vaktaal

* **Tabel**: onderdeel van een database bestaande uit kolommen en regels waarin gegevens worden opgeslagen.
* **Record**: een regel in een tabel.
* **Veld:** kruising van regel en kolom in een tabel, waarin een ondeelbaar gegeven bewaard wordt.
* **Veldnaam**: de naam van een kolom in een tabel.
* **DDL** (Data Definition Language): SQL-instructies die invloed hebben op de structuur van database-objecten. (Bijvoorbeeld: de SQL-instructie: create table)
* **DML** (Data Manipulation Language): SQL-instructies waarmee de inhoud van tabellen geraadpleegd en gewijzigd kan worden. (Bijvoorbeeld de SQL-instructie: insert into)
* **create table**:SQL-instructie waarmee een tabel wordt aangemaakt.
* **insert into**: SQL-instructie waarmee gegevens aan een tabel toegevoegd worden.
* **null**: veld heeft geen waarde
* **Datatype**: Soort waarde die een veld kan bevatten. Voorbeelden van datatypes zijn: varchar, char, date en decimal.