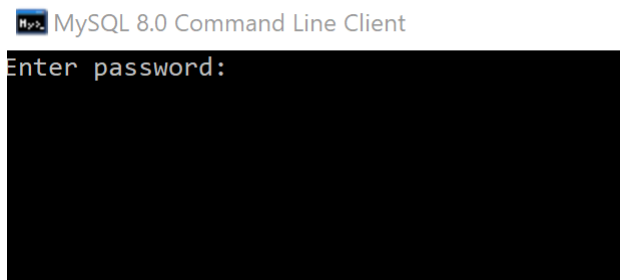


De lessen over MySQL gaan voornamelijk alleen over het kunnen werken met MySQL. De theorie over databases staat in een andere lessen serie.

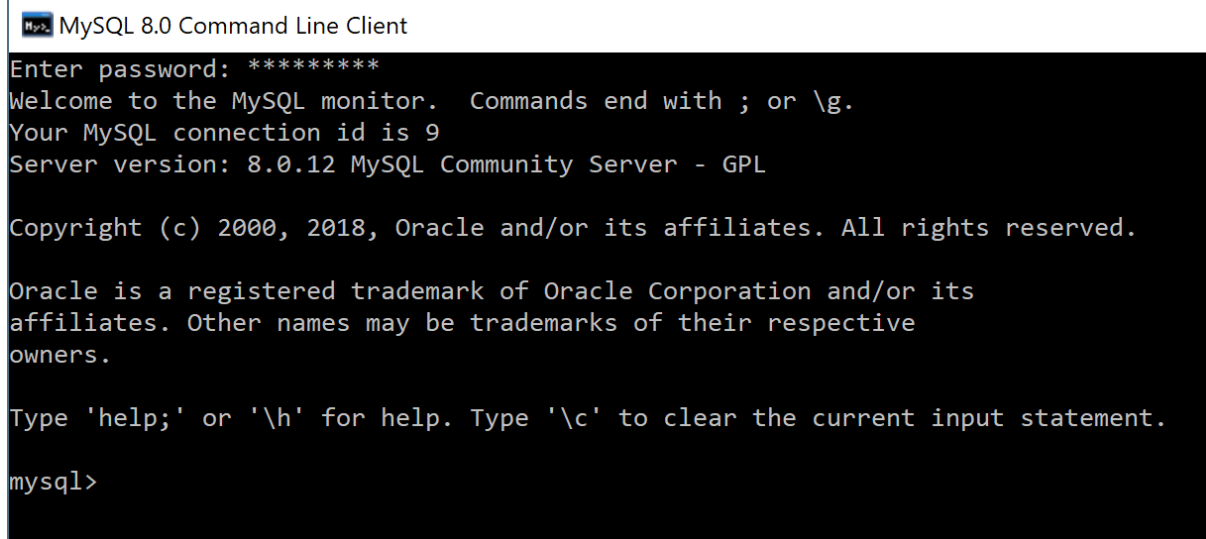
Voor deze lessen wordt gebruik gemaakt van de MySQL Command Line Client (MSCLC).

Tabellen

Start de MSCLC. Door in de takenbalk er naar te zoeken of in het Start menu naar de MySQL map te gaan.



Tik vervolgens het administrator wachtwoord in. Als je de installatie gebruikt hebt zoals in deze lessen is aangegeven dan is dit 'root'.



Om een overzicht te krijgen van alle beschikbare databases type de volgende MySQL instructie in het venster van de MSCLC:

```
SHOW DATABASES;
```

Let op! Vergeet vooral niet de ';' (**puntkomma**) aan het einde van de instructie. Elke MySQL instructie moet eindigen met een puntkomma.

```
MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.12 MySQL Community Server - GPL

Copyright (c) 2000, 2018, 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> show databases;
+-----+
| Database                |
+-----+
| information_schema       |
| mysql                   |
| performance_schema       |
| sakila                   |
| site_db                  |
| sys                      |
| world                    |
+-----+
7 rows in set (0.03 sec)

mysql> _
```

Om te kunnen werken met een van de databases moet je die aan de MySQL Server vertellen met de instructie:

```
USE database naam;
```

Wanneer een database is geselecteerd is het mogelijk om een lijst te krijgen van alle tabellen in deze database met de instructie:

```
SHOW TABLES;
```

In het vervolg van deze lessen gaan we gebruik maken van de databases 'site_db' die je aangemaakt tijdens de installatie procedure van MySQL. Indien dit niet gemaakt is kijk dan terug naar die stappen om te zien wat gedaan moet worden.

1

Start MSCLS en gebruik het root wachtwoord van de installatie.

2

Bij de **mysql>** command prompt tik in **SHOW DATABASES;** Controleer dat in de lijst 'site_db' staat.

3

Tik in de instructie `USE site_db;`

4

Tik in de instructie `SHOW TABLES;` om een overzicht van de tabellen in 'site_db' te krijgen.

MySQL 8.0 Command Line Client

```
Server version: 8.0.12 MySQL Community Server - GPL

Copyright (c) 2000, 2018, 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> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sakila |
| site_db |
| sys |
| world |
+-----+
7 rows in set (0.00 sec)

mysql> USE site_db;
Database changed
mysql> SHOW TABLES;
Empty set (0.01 sec)

mysql>
```

Je ziet dat site_db nog geen tabellen bevat. Om een tabel te maken wordt de volgende instructie gebruikt:

```
CREATE TABLE IF NOT EXISTS tabelnaam ( kolom type, kolom type );
```

De structuur van een tabel is op te vragen met de instructie:

```
EXPLAIN tabelnaam;
```

Een tabel kan ook verwijderd worden met de instructie:

```
DROP TABLE IF EXISTS tabelnaam;
```

Het invoeren van een tabel met veel kolommen is een lastige taak in de **mysql>** command prompt van het MSCLC. Daarom wordt een SQL instructie vaak in een SQL tekst bestand aangemaakt die daarna in MSCLC wordt uitgevoerd met de instructie **SOURCE**.

- 1 Start een tekst editor, zoals 'notepad' of 'notepad++' en type daar de onder staande query in.

```
CREATE TABLE IF NOT EXISTS fruit  
(  
    id INT,  
    naam TEXT,  
    kleur TEXT  
);
```

- 2 Sla het bestand op de Desktop op als **create_table.sql**. Start MSCLC en login als admin.
- 3 Gebruik de instructie **USE site_db**; om de database te gebruiken.
- 4 Gebruik de volgende instructie met het juiste pad om het SQL bestand uit te voeren.

SOURCE C:\Users\<Gebruikersnaam>\Desktop\create_table.sql

Waar staat <Gebruikersnaam> vul in de gebruikersnaam die bij de computer is geïnstalleerd.

- 5 Gebruik de instructie **EXPLAIN fruit**; om de tabelstructuur te bekijken.
- 6 Verwijder de tabel en controleer of dat het verwijderd is.
DROP TABLE IF EXISTS fruit;
SHOW TABLES;

De EXPLAIN instructie toont de structuur van de tabel fruit. Waar blijkt dat het 'id' veld een integer getal van 11 cijfers lang kan bevatten. Dit komt overeen met een getal tussen - 2147483648 tot 2147483647. De 'YES' geeft aan dat alle drie de velden de waarde **NULL** kan bevatten. NULL is niet hetzelfde als een lege string "".

Data typen definiëren

```
MySQL 8.0 Command Line Client
Copyright (c) 2000, 2018, 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> USE site_db;
Database changed
mysql> SOURCE c:\Users\Robert\Desktop\create_table.sql
Query OK, 0 rows affected (0.14 sec)

mysql> EXPLAIN fruit;
+----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| id    | int(11)| YES  |     | NULL    |       |
| naam  | text   | YES  |     | NULL    |       |
| kleur | text   | YES  |     | NULL    |       |
+----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> DROP TABLE IF EXISTS fruit;
Query OK, 0 rows affected (0.16 sec)

mysql> SHOW TABLES;
Empty set (0.00 sec)

mysql>
```

De onderstaande tabel beschrijft de verschillende gegevenstypes die kunnen worden gebruikt om kolommen van een tabel te definiëren.

Type	Waarde
INT	Een geheel getal van -2.147.483.648 tot 2.147.483.647
DECIMAL	Een zwevende kommagetal dat het aantal toegestane cijfers kan specificeren. DECIMAL (5,2) staat bijvoorbeeld -999,99 tot 999,99 toe.
DOUBLE	Een lang zwevende-kommagetal met dubbele precisie.
DATE	Een datum in JJJJ-MM-DD-formaat.
TIME	Een tijd in het UU:MM:SS-formaat.
DATETIME	Een gecombineerde datum en tijd in de notatie JJJJ-MM-DD UU:MM:SS
YEAR	Een jaar 1901-2155 in YY- of YYYY-indeling.
TIMESTAMP	Automatische datum en tijd van de laatste recordinvoer.
CHAR()	Een reeks gedefinieerde vaste lengte van maximaal 255 tekens. Met CHAR (100) wordt een kleinere tekenreeks opgemaakt om deze 100 tekens lang te maken.
VARCHAR()	Een reeks met een variabele lengte van maximaal 255 tekens lang die is opgeslagen zonder opvulling.
TEXT	Een reeks van tekens met een maximaal lengte 65.535 tekens.
BLOB	Een binair type variabele gegevens.
ENUM	Een enkele reekswaarde uit een gedefinieerde lijst. Met ENUM ("rood", "groen", "blauw") kan bijvoorbeeld alleen een van deze drie kleuren worden ingevoerd.
SET	Een reeks van meerdere strings uit een gedefinieerde lijst. Met SET ("rood", "groen", "blauw") kan bijvoorbeeld een of meer van deze kleuren worden ingevoerd.

Het SQL-script dat hier onder wordt vermeld, maakt een tabel met specificatietypes voor gegevenstypen in de kolomdefinities. De kolom 'id' accepteert alleen geheel getalwaarden. De datum en tijd van elk item worden automatisch vastgelegd in de kolom "datum". De kolommen "voornaam" en "achternaam" kunnen elk maximaal 20 tekens bevatten.

- 1 Start een teksteditor, zoals Notepad, en maak een query die een tabel met kolommen met verschillende gegevenstypen definieert.

```
CREATE TABLE IF NOT EXISTS user_log
(
    id            INT,
    datum        TIMESTAMP,
    voornaam     VARCHAR(20),
    achternaam   VARCHAR(20)
);
```

- 2 Sla het bestand op het bureaublad op als **data_types.sql**. Start de MSCLC en log in als een gebruiker met volledige toegangsrechten, zoals root.
- 3 Voer vervolgens een query uit om de gevestigde database **USE site_db**; te kunnen gebruiken.
- 4 Voer nu de SQL-query uit, door het pad naar het SQL-bestand op uw computer te vermelden, zoals:

```
SOURCE C:\Users\<Gebruikersnaam>\Desktop\data_types.sql
```

- 5 Voer de query uit om het bestaan en formaat van de tabel te bevestigen. **EXPLAIN user_log**;

```
MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.12 MySQL Community Server - GPL

Copyright (c) 2000, 2018, 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> USE site_db;
Database changed
mysql> SOURCE c:\Users\Robert\Desktop\data_types.sql
Query OK, 0 rows affected (0.11 sec)

mysql> EXPLAIN user_log;
+----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| id    | int(11) | YES |     | NULL |       |
| datum | timestamp | YES |     | NULL |       |
| voornaam | varchar(20) | YES |     | NULL |       |
| achternaam | varchar(20) | YES |     | NULL |       |
+----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

Modifiers toevoegen

Naast het opgeven van toegestane gegevenstypen, met sleutelwoorden in de vorige tabel, kunnen de modificatietrefwoorden in de volgende tabel optioneel worden gebruikt om de kolominhoud verder te beheren:

Type	Waarde
NOT NULL	Verplicht dat elke record een gegevenswaarde moet bevatten in deze kolom
UNIQUE	Verplicht dat records geen enkel item in deze kolom mogen dupliceren
AUTO_INCREMENT	Alleen beschikbaar voor numerieke kolommen, om automatisch een getal te genereren dat één meer is dan de vorige waarde in die kolom
DEFAULT	Geeft een waarde aan die moet worden gebruikt als er geen waarde wordt opgegeven voor deze kolom wanneer een record wordt ingevoegd.
PRIMARY KEY	Geeft de kolom of kolommen aan die moeten worden gebruikt als primaire sleutel voor die tabel.

Het onderstaande SQL-script maakt een tabel die *modificatoren* (**modifiers**) gebruikt in de kolomdefinities om de toegestane inhoud te beheren.

- 1 Start een teksteditor, zoals Notepad, en maak een query die een tabel met kolommen met verschillende modificatoren definieert.

```
CREATE TABLE IF NOT EXISTS producten
(
    id            INT UNIQUE      AUTO_INCREMENT,
    code          INT            NOT NULL,
    naam          VARCHAR(20)    NOT NULL,
    hoeveelheid  INT            DEFAULT 1 NULL,
    prijs         DECIMAL(6,2)   NOT NULL
);
```

- 2 Sla het bestand op het bureaublad op als **modifiers.sql**. Start de MSCLC en log in als een gebruiker met volledige toegangsrechten, zoals root.
- 3 Voer vervolgens een query uit om de gevestigde database **USE site_db**; te kunnen gebruiken.
- 4 Voer nu de SQL-query uit, door het pad naar het SQL-bestand op uw computer te vermelden, zoals:

SOURCE C:\Users\<Gebruikersnaam>\Desktop\modifiers.sql

- 5 Voer de query uit om het bestaan en formaat van de tabel te bevestigen.
EXPLAIN producten;

```
mysql> SOURCE c:\Users\Robert\Desktop\modifiers.sql
Query OK, 0 rows affected (0.18 sec)

mysql> EXPLAIN producten;
+----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| id    | int(11) | NO | PRI | NULL | auto_increment |
| code  | int(11) | NO |     | NULL |                 |
| naam  | varchar(20) | NO |     | NULL |                 |
| hoeveelheid | int(11) | YES |     | 1 |                 |
| prijs | decimal(6,2) | NO |     | NULL |                 |
+----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Primaire sleutels instellen

Een **PRIMAIRE SLEUTEL** is een "*beperking*" die wordt toegepast op een kolom om elke rij van de databasetabel eenduidig te identificeren. Het zorgt ervoor dat de waarden in elke rij van die kolom uniek zijn en nooit veranderen, dus die waarden kunnen worden gebruikt om naar een specifieke rij te verwijzen.

Door de **PRIMAIRE SLEUTEL**-voorwaarde in te stellen, is het mogelijk om gegevens op specifieke rijen van de databasetabel te manipuleren.

Elke kolom kan worden ingesteld als een **PRIMAIRE SLEUTEL**, maar het is vaak de eerste kolom die wordt gebruikt om een uniek identificatienummer op te geven.

Elke kolom die als **PRIMAIRE SLEUTEL** is ingesteld, moet aan deze criteria voldoen:

- Elk veld in die kolom moet een waarde hebben
 - Het mag niet leeg zijn of een **NULL**-waarde hebben.
- Elke waarde in die kolom moet uniek zijn
 - Er mogen geen duplicaties zijn.
- Elke waarde in die kolom kan *nooit* worden gewijzigd of bijgewerkt.
- Elke waarde in die kolom kan *niet opnieuw* worden gebruikt
 - Wanneer een rij wordt verwijderd, kan de **PRIMAIRE SLEUTEL**-waarde ervan niet opnieuw worden toegewezen als de **PRIMAIRE SLEUTEL**-waarde van een nieuwe rij.

Merk op dat in het vorige voorbeeld MySQL automatisch de "id" -kolom als de **PRIMAIRE SLEUTEL** instelt vanwege de **UNIQUE**-modifier die in de definitie van die kolom was opgenomen.

De **PRIMAIRE SLEUTEL**-voorwaarde kan worden ingesteld door de **PRIMARY SLEUTEL** sleutelwoorden toe te voegen aan de kolomdefinitie in de **CREATE TABLE** query.

Als alternatief kan een **PRIMAIRE SLEUTEL** elders in de **CREATE TABLE** query worden ingesteld door de naam van de kolom tussen haakjes na de **PRIMARY SLEUTEL** sleutelwoorden te vermelden.

Het SQL-script op de tegenovergestelde pagina maakt twee tabellen met een **PRIMAIRE SLEUTEL**-kolom die op verschillende manieren is ingesteld. Elke tabel kan naar elke specifieke rij verwijzen door zijn "id" -waarde.

1

Start een teksteditor, zoals Notepad, en maak een query die een tabel met kolommen met verschillende modificatoren definieert.

```
CREATE TABLE IF NOT EXISTS bekers
(
    id          INT AUTO_INCREMENT PRIMARY KEY,
    beker_patroon VARCHAR(25)
);
```

```
CREATE TABLE IF NOT EXISTS potten
(
    id          INT AUTO_INCREMENT,
    pot_patroon VARCHAR(25), PRIMARY KEY( id )
);
```

2

Sla het bestand op het bureaublad op als **primary_key.sql**. Start de MSCLC en log in als een gebruiker met volledige toegangsrechten, zoals root.

3

Voer vervolgens een query uit om de gevestigde database **USE site_db**; te kunnen gebruiken.

4

Voer nu de SQL-query uit, door het pad naar het SQL-bestand op uw computer te vermelden, zoals:

SOURCE C:\Users\<Gebruikersnaam>\Desktop\modifiers.sql

5

Voer de query uit om het bestaan en formaat van de tabel te bevestigen.

EXPLAIN bekers;

EXPLAIN potten;

```
mysql> SOURCE c:\Users\Robert\Desktop\primary_key.sql
Query OK, 0 rows affected (0.19 sec)

Query OK, 0 rows affected (0.07 sec)

mysql> EXPLAIN BEKERS;
+----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| beker_patroon | varchar(25) | YES | | NULL | |
+----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> EXPLAIN POTTEN;
+----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| pot_patroon | varchar(25) | YES | | NULL | |
+----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Merk op dat de **EXPLAIN** query's beide **PRIMAIRE SLEUTEL** instellingen bevestigen.

Tabellen wijzigen

Het formaat van een bestaande databasetabel kan worden gewijzigd met een **ALTER TABLE**-instructie. Met deze instructie kan een enkele wijziging aangebracht worden of een aantal wijzigingen kunnen worden opgeven door een komma's gescheiden lijst.

Een **ALTER TABLE**-instructie kan een volledige nieuwe **COLUMN** aan een bestaande tabel toevoegen met een **ADD**-instructie, zoals deze:

```
ALTER TABLE tabelnaam;  
ADD COLUMN kolomnaam datatype optionele modificatoren;
```

Het kan ook een **PRIMAIRE SLEUTEL** toevoegen met de **ADD**-instructie aan een bestaande kolomdefinitie met behulp van deze syntaxis:

```
ALTER TABLE tabelnaam;  
ADD PRIMARY KEY ( kolomnaam );
```

Een **ALTER TABLE**-instructie kan de naam van een bestaande kolom veranderen met de **CHANGE** instructie. De nieuwe kolom neemt geen gegevenstype of modifiers over die zijn opgegeven in de oorspronkelijke kolom - deze moeten opnieuw worden ingesteld in de instructie **ALTER TABLE**, zoals hieronder:

```
ALTER TABLE tabelnaam;  
CHANGE oude-kolomnaam nieuwe-kolomnaam datatype optionele modificatoren;
```

Een **ALTER TABLE**-instructie kan ook een volledige kolom permanent van de tabel verwijderen met behulp van de instructie **DROP COLUMN**.

```
ALTER TABLE tabelnaam;  
DROP COLUMN kolomnaam ;
```

Het volgende SQL-script toont al deze mogelijkheden:

- 1 Start een teksteditor, zoals Notepad, en maak een query die een tabel met drie kolommen maakt.

```
CREATE TABLE IF NOT EXISTS pannen
(
    id            INT            NOT NULL,
    patroon       VARCHAR(25)    NOT NULL,
    prijs         DECIMAL(6,2)
);
```

```
EXPLAIN pannen;
```

```
ALTER TABLE pannen
ADD PRIMARY KEY ( ID ),
ADD COLUMN code INT UNIQUE NOT NULL,
CHANGE patroon pan_patroon VARCHAR(25) NOT NULL,
DROP COLUMN price;
```

```
EXPLAIN pannen;
```

- 2 Sla het bestand op het bureaublad op als **alter_table.sql**. Start de MSCLC en log in als een gebruiker met volledige toegangsrechten, zoals root.
- 3 Voer vervolgens een query uit om de gevestigde database **USE site_db**; te kunnen gebruiken.
- 4 Voer nu de SQL-query uit, door het pad naar het SQL-bestand op uw computer te vermelden, zoals:

SOURCE C:\Users\<Gebruikersnaam>\Desktop\alter_table.sql

```
mysql> SOURCE c:\Users\Robert\Desktop\alter_table.sql
Query OK, 0 rows affected, 1 warning (0.01 sec)

+----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| id | int(11) | NO |  | NULL |  |
| patroon | varchar(25) | NO |  | NULL |  |
| PRIJS | decimal(6,2) | YES |  | NULL |  |
+----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.25 sec)
Records: 0 Duplicates: 0 Warnings: 0

+----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL |  |
| pan_patroon | varchar(25) | NO |  | NULL |  |
| code | int(11) | NO | UNI | NULL |  |
+----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

MySQLi methode

Naast **.sql** tekstbestanden is het ook mogelijk om met behulp van **MySQLi** (de 'i' staat voor "improved") of **PDO** dezelfde handelingen uit te voeren in een PHP-script. MySQLi is een vervanging voor de mysql-functies, met object-georiënteerde en procedurele versies. Het heeft ondersteuning voor voorbereide verklaringen. PDO (PHP Data Objects) is een algemene database-abstractielaag met ondersteuning voor MySQL tussen vele andere databases.

Script voor een verbinding met MySQL database:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

Script voor het aanmaken van een database met MySQLi:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . $conn->error;
}

$conn->close();
?>
```

Script voor het aanmaken van een tabel in een DB met MySQLi:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// sql to create table
$sql = "CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP
)";

if ($conn->query($sql) === TRUE) {
    echo "Table MyGuests created successfully";
} else {
    echo "Error creating table: " . $conn->error;
}

$conn->close();
?>
```