© 2020 CODEMEMBERS - AUTEUR/ TOM VANHOUTTE CODEMEMBERS IS EIGENDOM VAN IN2SITES.BE

CODEMEMBERS - PROFESSIONELE ICT CURSUSSEN

Cursus MYSQL

Inhoudstafel

intro	4
Database	4
Installatie	5
Windows - wampserver	5
PHPmyadmin	g
Openen	
Databases	
Nieuw	
Engine InnoDB	
Verwijderen	
Sample Databases	
Installatie SAKILA database	
Tabellen.	
Datatypes	
Nieuw	
veld instellingen	
Code: CREATE TABLE	
Relaties	
Relatie: één-op-veel	
Code: FOREIGN KEY CONSTRAINTS	
CRUD	
SELECT STATEMENT	
OEFENINGEN	
JOINS	
TYPES	
INNER JOIN	
LEFT JOIN	
RIGHT JOIN	
CROSS JOIN	
INSERT STATEMENT	
UPDATE STATEMENT	34
DELETE STATEMENT	34
VIEWS	
EIGEN FUNCTIES	36
EXECUTE	37
EXECUTE - CODE	
STORED PROCEDURE	39
CREATE PROCEDURE	39
Database normalisatie	42
Voordelen van database normalisatie	42
Doel database normalisatie	42
Voorbeeld	42
Normalisatieproces	43
ODE NORMAALVORM	43
1STE NORMAALVORM	44
2DE NORMAALVORM	45
3DE NORMAALVORM	48

Intro

In deze cursus zal manipulaties op een database zowel lokaal als online leren uitvoeren. De taal die we hiervoor zullen leren is mySQL. SQL = Structured Query Language.

LEGENDE
Belangrijke info
Notitie Notitie
■ Oefening
□ Opdracht

De symbolen die u in de legende hiernaast ziet staan zullen we doorheen deze cursus gebruiken. Belangrijke info en oefeningen worden gezamenlijk aangeleerd en uitgelegd.

Notities kunnen door u als cursist worden toegevoegd. Opdrachten volgen op de oefeningen en dient u individueel op te lossen om de opgedane kennis te toetsen.

Database

en database is een georganiseerde verzameling van gegevens die toegankelijk is en wordt opgeslagen door een computersysteem. Databases worden ontwikkeld d.m.v. modelleertechnieken. Er is een verchil tussen relationele en nietrelationele databases. In deze cursus zullen we het hebben over relationele databases.

Voorbeelden van relationele databases zijn: Mysql, Oracle, Access, Microsoft Sql Server, DB2, ...

Voordelen:

- snel opzoeken van informatie
- verschillende tabel die gerelateerd worden d.m.v. keys (sleutels)
- efficiënt opslaan van data
- rechtenstructuur voor gebruikers
- opvragen via één gestandardiseerde taal: SQL.

Installatie

Pen mySQL database wordt gebruikt voor webtoepassingen. D.w.z. dat we een werkende webserver dienen te hebben met een volledige installatie van mySQL op deze server. De server die we hiervoor zullen installeren is de Apache Webserver die de meest gebruikte webserver is op het internet. Daarnaast installeren we op deze webserver de mySQL omgeving. Er zijn verschillende mogelijkheden voor deze omgeving zoals: mongoDB, mariaDb, phpmyAdmin, ...

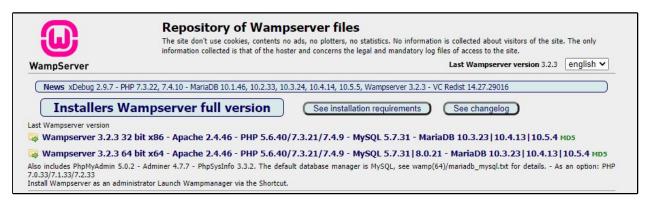
Wij gebruiken phpmyAdmin, daar deze het meest wordt toegepast.

Windows - wampserver

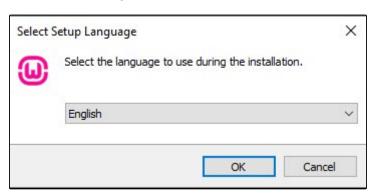
Belangrijke info

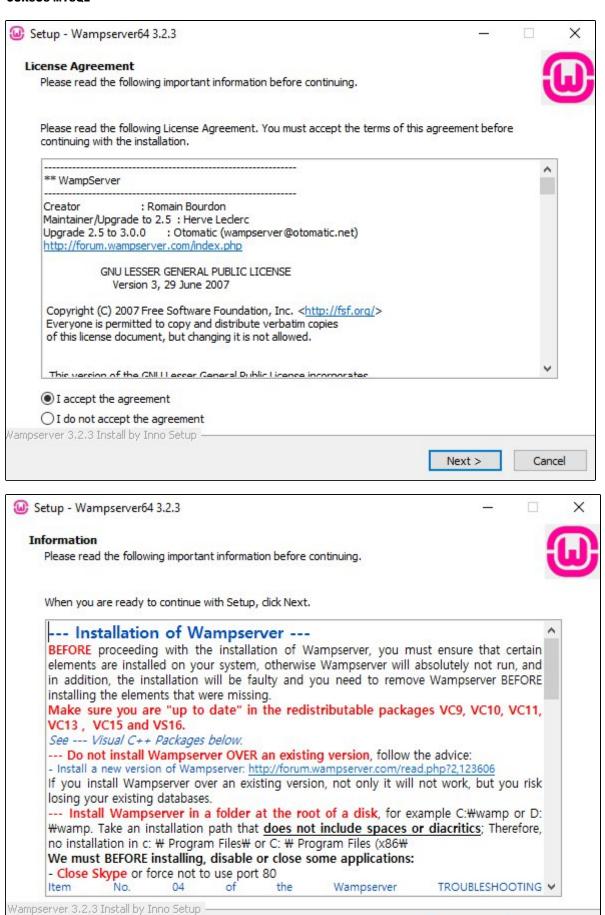
Om de apache webserver en de mysql omgeving te installeren surf je naar de volgende link: http://wampserver.aviatechno.net

Download de gepaste versie (voor de meeste pc's zal dit de x64 versie zijn.



In de downloadsmap dubbelklik je op het uitvoeringsbestand en start je de installatie: volgt de schermafbeeldingen hieronder.

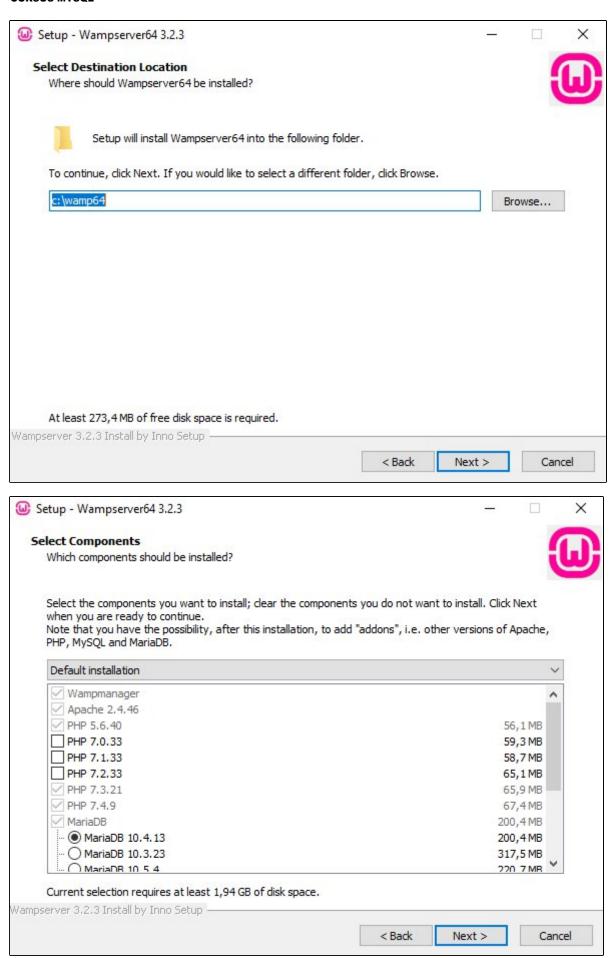


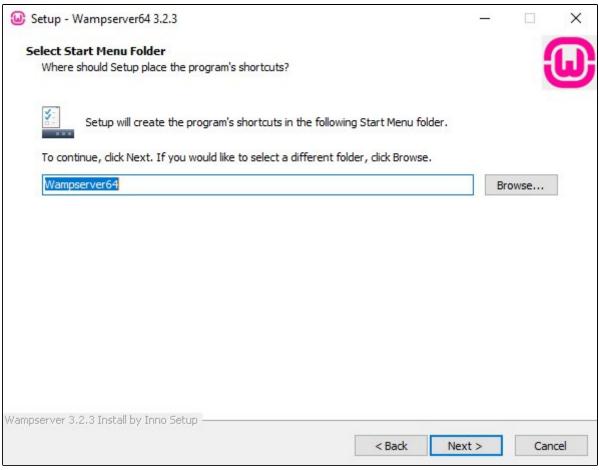


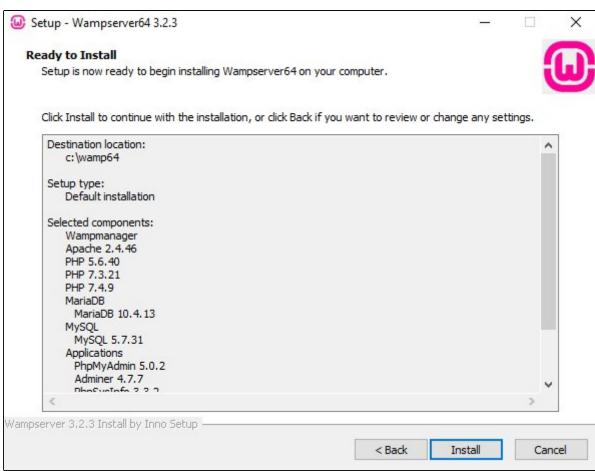
< Back

Next >

Cancel







PHPmyadmin

e database omgeving waarin wij zullen werken is phpmyadmin. We zullen dus de mySQL taal rechtstreeks op verschillende databases uitvoeren. Bij manipulatie van een database kan je dus met deze taal acties uitvoeren die dikwijls niet omkeerbaar zijn. Bijvoorbeeld het verwijderen van een tabel in een dabase of het wijzigen van gebruikers, Heden ten dage draait alles om data die zeer waardevol is voor verschillende mogelijke partijen, waaronder personen met minder goede bedoelingen zoals hackers. Wanneer je bijvoorbeeld later een carrière als ethisch hacker wil uitoefenen, dan dien je dus de SQLtaal al zeer goed te kennen en de omgevingen waarin deze taal wordt uitgevoerd.

Openen

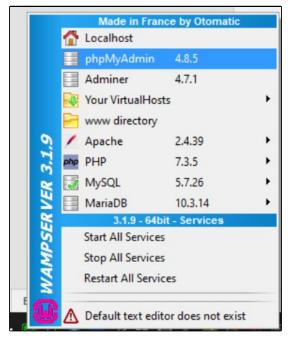
Na de installatie van de wampserver zal je dit icoontje terugvinden op je bureaublad. Dubbelklik op dit icoontje op zowel de apache webserver te starten alsook mySQL. Je pc fungeert nu eigenlijk als een lokale webserver die NIET geconnecteerd is met het web. Een reëele webserver is WEL geconnecteerd met het web.

Webservers voorzien dus enerzijds hosting om je website of webapplicatie op te laden alsook de phpmyadmin databaseomgeving anderzijds.

Als developer zullen we dus onze webapplicaties eerst lokaal maken alvorens deze op te laden naar een actieve hosting.

Wanneer je alles correct geïnstalleerd hebt en alles opgestart is, zul je rechts onderaan in je taakbalk hetzelfde icoontje zien staan die GROEN dient te zijn. Bij een groen icoontje werd alles correct gestart.

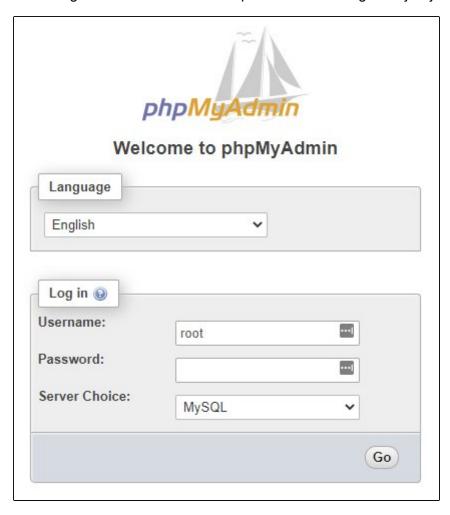
Bij een ORANJE icoontje werd ofwel de apache webserver of mysql niet goed opgestart. Bij een ROOD icoontje werd niets opgestart.



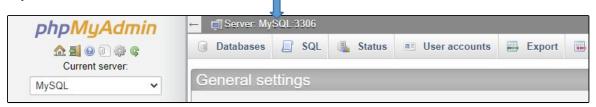
Klik nu met je linker muisknop op dit GROENE icoontje en klik vervolgens op phpmyadmin om de database omgeving te openen. Deze omgeving zal jullie speelplaats worden tijdens deze cursus. Naargelang de evolutie van alle programma's kunnen de versies die in de schermafbeeldingen te zien zijn verschillen van deze cursus.

In het volgende scherm zal je nu kunnen inloggen. In de lokale ontwikkelomgeving dienen we geen speciale username of password te gebruiken. Standaard loggen we op phpmyadmin in met de user **root** en laten we password **blanco** staan. Klik vervoglens op **Go**.

Vanzelfsprekend wanneer we onze afgewerkte database zouden opladen naar een reëele webserver zal dit aangepast worden. Het zou bijzonder leuk zijn voor personen met slechte bedoelingen moest dit ook online op deze manier toegankelijk zijn.



Na het inloggen kom je terecht in de standaard omgeving van phpmyadmin. Hier kan je verschillende databases aanmaken en/of bevragen. Het belangrijkste in onderstaand scherm is de positionering nl. **Server:MySQL:3306**. d.w.z. dat je je nu op het hoogste niveau van de mySQL Server bevindt.



Er dient ook een woordje uitleg te worden gegeven over het onderstaande scherm. Hier zie je dat de **Web Server** die momenteel draaiende is dankzij wamp, de **Apache webserver** is. De **Database Server** die we momenteel via phpmyAdmin bekijken is **mySQL**.

PHPmyAdmin is de huidige omgeving waarin mySQL momenteel draait.

Dit kan dus evengoed een andere omgeving zijn zoals mongoDB, mariaDB die eveneens mySQL draaien, maar worden hier niet besproken.

Database server

- Server: MySQL (127.0.0.1 via TCP/IP)
- · Server type: MySQL
- · Server connection: SSL is not being used (a)
- Server version: 5.7.26 MySQL Community Server (GPL)
- · Protocol version: 10
- User: root@localhost
- · Server charset: UTF-8 Unicode (utf8)

Web server

- Apache/2.4.39 (Win64) PHP/7.3.5
- Database client version: libmysql mysqlnd 5.0.12-dev 20150407 \$Id: 7cc7cc96e675f6d72e5cf0f267f48e167c2abb23 \$
- PHP extension: mysqli curl mbstring mbstring
- PHP version: 7.3.5

phpMyAdmin

- · Version information: 4.8.5, latest stable version: 4.9.5
- Documentation
- Official Homepage
- Contribute
- · Get support
- · List of changes
- License

Databases

Nieuw



Een nieuwe database creëeren via phpMyAdmin is eenvoudig. Aan de linkerkant zie je een menu die alle databases zal bevatten. Je kan dus oneindig veel databases op phpMyAdmin draaien voor verschillende webapplicaties. Klik op **New** om een database aan te maken.



Zoals je hierboven kan zien zie je de characterset latin_swedish_ci. Dit is de standaard. Deze characterset bestaat uit 8 bits in maximale grootte. Standaard is dit meestal meer dan genoeg om tekens (letters, cijfers, speciale tekens) te bewaren.

In deze characterset zitten natuurlijk alle tekens die ook in de ASCII tabel zitten en meer. Waarom wordt dan de ASCII tabel niet gebruikt? ASCII telde initieel 127 symbolen die konden bewaard worden in een 7-bits karakterset. In een 8-bits karakterset kunnen er dus meer symbolen worden gebruikt.

Voor websites zitten we tegenwoordig aan 16-bits in grootte. De characterset die daar zal worden gebruikt is de huidige standaard UTF-8.

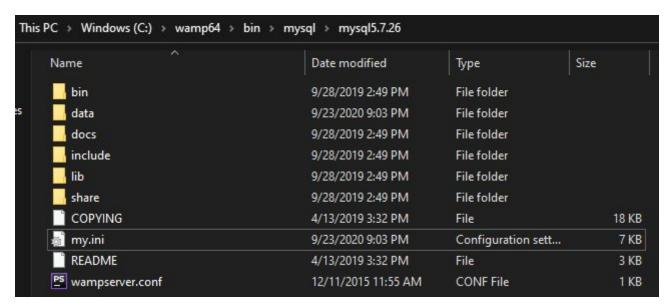
In de meeste gevallen heb je echter genoeg met de swedisch karakterset. Wil je zeker zijn dan kan je de characterset hier ook aanpassen naar UTF-8.

Kies nu utf8_general_ci als characterset en druk create.

De database werd nu aangemaakt. De engine die wordt gebruikt om de database te bevragen in phpMyAdmin is standaard MyISAM. Deze heeft als nadeel dat we geen relaties kunnen leggen tussen tabellen in deze omgeving. We dienen deze om te zetten naar de andere engine van phpMyAdmin, nl. InnoDB

Engine InnoDB

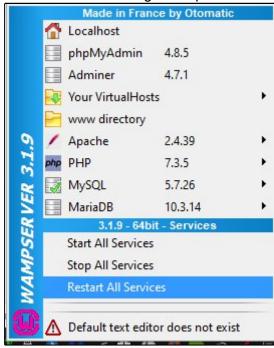
Om de engine te wijzigen dienen we het bestand my.ini te openen op onderstaande locatie (zie schermafbeelding).



In my.ini zoek je achter de mylSAM engine en wijzig je deze naar InnoDB.

- ; The default storage engine that will be used when create new tables default-storage-engine=InnoDB
- ; New for MySQL 5.6 default_tmp_storage_engine if skip-innodb enable
- ; default_tmp_storage_engine=MYISAM

Herstart nu de volledige wampserver als volgt om de engine in te laden:



Verwijderen

Databases kunnen we ook verwijderen uit phpMyAdmin.

- klik bovenaan op Server:MySQL:3306
- klik vervolgens op het tabblad **Databases**
- ▼ ✓ wagens Selecteer de te verwijderen database

Klik op **Drop** en druk **OK**

Sample Databases

Voor mySQL werden er enkel databases aangemaakt speciaal voor studenten die de taal wensen onder de knie te krijgen.

Link: https://dev.mysql.com/doc/index-other.html

De databases die o.a. worden gebruikt zijn employee, world, sakila en menagerie. Wij zullen de **sakila** database gebruiken, maar eerst installeren. **Download** hier het **Zip** bestand van de database. We pakken deze uit op de c:\ schijf.

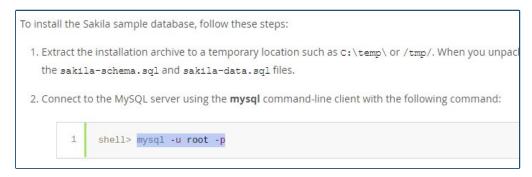
Example Databases			
Title	Download DB	HTML Setup Guide	PDF Setup Guide
employee data (large dataset, includes data and test/verification suite)	GitHub	View	US Ltr A4
world database	Gzip Zip	View	US Ltr A4
world_x database	TGZ Zip	View	US Ltr A4
sakila database	TGZ Zip	View	US Ltr A4
menagerie database	TGZ Zip		

Installatie SAKILA database

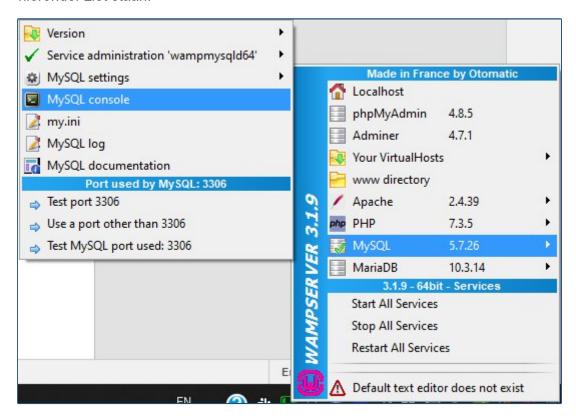
Hiervoor zullen we de **HTML Setup Guide** gebruiken van de sakila database. Druk dan op **Installation**.

Table of Contents 1 Preface and Legal Notices 2 Introduction 3 History 4 Installation 5 Structure 6 Usage Examples 7 Known Issues 8 Acknowledgments 9 License for the Sakila Sample Database 10 Note for Authors 11 Sakila Change History

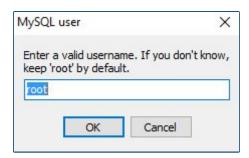
In de volgende pagina zie je nu enkele commando's die we in een console dienen in te geven:



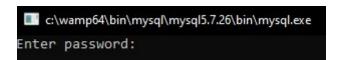
De console die we hiervoor gaan gebruiken is de MySQL console die je in de schermafbeelding hieronder ziet staan.



De standaard username van phpMyAdmin zal worden gevraagd. Druk hier op OK.



Het paswoord die je dient in te vullen is **blanco**. Druk dus gewoon op **ENTER**.



Wanneer alles correct werd uitgevoerd zit je in de console van mySQL.

```
c:\wamp64\bin\mysql\mysql\mysql.exe
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 58
Server version: 5.7.26 MySQL Community Server (GPL)

Copyright (c) 2000, 2019, 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>
```

Nu dienen we het de sakila database die we uitgepakt hebben op de c schijf te installeren. Eerst wordt de structuur (tabellen) geïnstalleerd met het volgende commando. Let op dat de locatie juist is.

```
mysql> SOURCE C:/sakila-db/sakila-schema.sql
```

Vervolgens gaan we de data ook in de tabellen injecteren met het volgende commando.

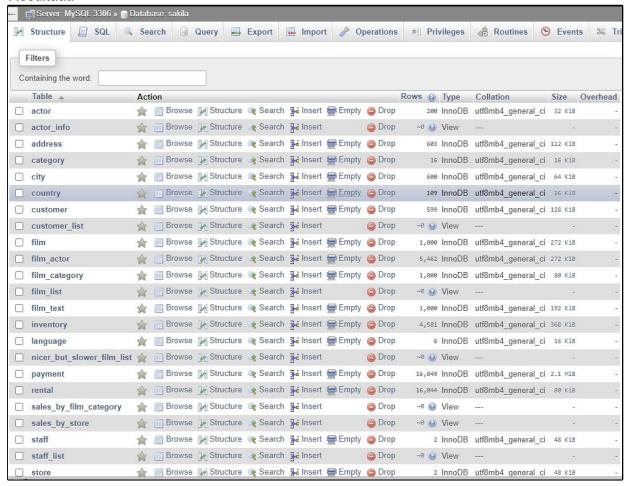
```
mysql> SOURCE C:/sakila-db/sakila-data.sql
```

□Opdracht

Probeer ook de world database te installeren.

Sluit nu de consoletoepassing af en open terug phpMyAdmin.

Resultaat:



Tabellen

Een database bestaat uit genormaliseerde tabellen. Normalisatie is een onderwerp die we later zullen bekijken. Het komt er op neer dat iedere tabel zijn eigen specifieke data heeft en gerelateerd kan zijn met andere tabellen in de database.

Voorbeeld:

De tabel met gebruikers kan gerelateerd zijn met de tabel van adressen.

Een gebruiker kan dus één of meerdere adressen hebben.

Datatypes

Tabellen bestaan uit velden. Ieder veld heeft zijn eigen veldgebonden data en zijn datatype. Een huisnummer zal bijvoorbeeld van het type integer zijn.

De mogelijke datatypes die we kunnen gebruiken in mySQL zijn:

Data Type=INTEGERS	grootte	unsigned
INT	-2.1 billioen tot 2.1 billioen	0 to 4.2 billioen
TINYINT	-128 tot 127	0 tot 255
SMALLINT	-32768 tot 32767	0 tot 65535
MEDIUMINT	-8.3 millioen tot 8.3 millioen	0 tot 16.7 millioen
BIGINT	-263 tot -263-1	0 tot264-1

Data Type=FLOATING POINT	gebruik
FLOAT	Decimalen (single precisie)
DOUBLE	Grote Decimalen (double precisie)
DECIMAL	Waarden waar afrondingserrors niet worden aanvaard.Bijv. geld, 65 digits voor de komma en 30 na de komma precisie. Dit wordt gebruikt om o.a. geld te berekenen.

Voorbeeld van gebruik van FLOATING POINT:

De gebruiker tik respectievelijk in:

Rij 1:

0.6, 0.6, 0.6

Rij 2:

0.00006, 0.00006, 0.00006

FLOAT	DOUBLE	DECIMAL
0.6	0.6	0.60000
0.00006	0.00006	0.00006

Wanneer we nu met deze getallen rekenen dan krijgen we het volgende resultaat: we vermenigvuldigen alles met 1000

FLOAT	DOUBLE	DECIMAL
600.0000238418579	600	600.00000
0.05999999848427251	0.060000000000000005	0.06000

Conclusie: Decimals zijn het meest accuraat te gebruiken.

Data Type= STRINGS en BINAIRE DATA	gebruik
VARCHAR	Korte, variable lengte van tekst
CHAR	Korte,vaste lengte van tekst,bijv. Encrypted passwords
TEXT	Lange tekst, bijv. artikelen
ENUM	1 waarde van een voorgedefinieerde lijst. Voorbeeld van lijst: zomer, herfst, lente, winter
BLOB	Opslag van beelden en audiobestanden als gecompresseerde files.

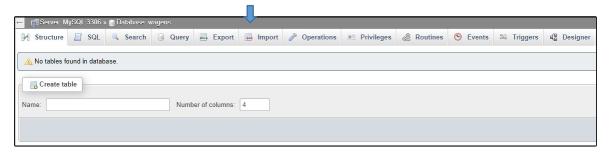
Data Type=DATUM EN TIJD	format
DATE	YYYY-MM-DD
TIME	hh:mm:ss
DATETIME	YYYY-MM-DD hh:mm:ss
TIMESTAMP	YYYY-MM-DD hh:mm:ss
JAAR	YYYY

De meeste gebruikte datatypes zijn:

Data Type	GEBRUIK
INT	Integers
TINYINT	Kleine Integers zoals leeftijd
DECIMAL	Geld en afmetingen
VARCHAR	Korte tekst
TEXT	Lange tekst
DATETIME	Datum

Nieuw

Een nieuwe tabel aanmaken is eenvoudig. Je dient eerst je database te selecteren. Je kan steeds controleren of je in de correcte database zit bovenaan phpMyAdmin. Net zoals in windows verkenner is dit een folder structuur die vertrekt van het server niveau.



In ons voorbeeld hebben we het over wagens. Wagens bestaan uit merken en modellen. Hiervoor creëeren we dus 2 tabellen.

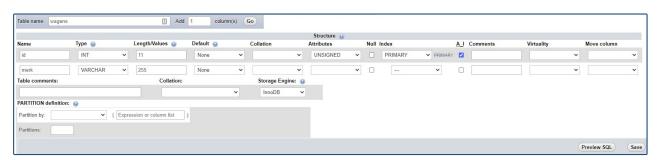
De eerste tabel wagens zal 2 velden bevatten: id, merk.

Vul de **naam** van de tabel in en duid hiervoor **2** kolommen aan voor het aantal velden. Klik dan op **Go**.



Conventie bij het schrijven van veldnamen: ALLEMAAL KLEINE LETTERS! GEEN CAMELCASE zoals variabelen in PROGRAMMEERTALEN.

Vul in zoals onderstaande afbeelding en klik op SAVE. Je hebt nu een lege tabel zonder data aangemaakt in de database van wagens.



Veld instellingen

Hier zullen we kort alle mogelijke instellingen bespreken tijdens het maken van velden in een tabel.

Name

Hier worden de kolomkoppen of veldnamen ingegeven. De eerste **veldnaam** die wij bijna altijd zullen invullen is het **id** van de tabel. Het **id** is ook meestal een oplopende nummering die start van 1 met een **auto_increment**. Auto_increment zorgt dus m.a.w. automatisch dat een volgend **record** automatisch verder wordt genummerd. Dit id is **UNIEK** en wordt de **PRIMARY KEY** van een tabel genoemd.

Een **record** is één enkele data-lijn in de database met ingevulde data per veld.

Type

Het type per veld is het datatype die we dienen te bepalen. Aangezien een id een oplopende nummering is zal dit van het type integer zijn. Hier kiezen we **INT.**

Een VARCHAR kan bijvoorbeeld gebruikt worden voor een string, DATETIME voor een datum, ...

Length/values

Aangezien het datatype bijvoorbeeld INT is kunnen we tot ca 2 billioen in cijfers laten oplopen. Dankzij lengte beperken we het aantal karakters toch. Veelal wordt er bij een id 11 karakters aan lengte toegevoegd die voor de meeste database meer dan voldoende is.

Het laatste id zou dus in principe 9999999999 kunnen zijn.

Default values

Hiermee kan je een standaard waarde meegeven wanneer een veld niet zou worden ingevuld door een gebruiker.

Er zijn 4 opties:

none = is default en blijft dus leeg

NULL = hier wordt een nullable in het veld geschreven. Je zal dan het woord NULL zien staan in het veld.

As defined: hier verschijnt een extra veld waar je zelf een default waarde kan schrijven. Deze default waarde wordt in het veld van de database weggeschreven wanneer de gebruiker niets zou hebben geschreven.

Current time stamp = Zal de datumtijdstempel wegschrijven van de pc.

Collation

Voor ieder veld kan je een aparte character set toevoegen indien je dit nodig zou hebben. In de meeste gevallen is de initiële character set van de database voldoende en blijft dit veld leeg.

Attributes

BINARY

De opslag van een ingetikte waarde zal binair gebeuren (1 en 0)

UNSIGNED

Een unsigned value kan enkel positieve getallen opslaan in de database.

Zie tabel datatypes (voorgaand).

UNSIGNED ZERO FILL

Zero fill zorgt ervoor dat alle ontbrekende characters met een 0 worden opgevuld.

Bijvoorbeeld: id(11)

id(11) kan een lengte van 11 getallen opslaan. Wanneer we nu het getal 1 opslaan dan wordt dit met zero fill: 00000000001

on update CURRENT_TIMESTAMP

Wanneer er een veld van het datatype datetime wordt gebruikt dan kunnen we hier ervoor zorgen dat automatisch de huidige datumtijdstempel van de pc wordt gebruikt wanneer een gebruiker een wijziging aanbrengt op dit record.

Null

Het veld mag leeg zijn wanneer aangevinkt.

Index

PRIMARY

Met deze optie kan je een veld de primaire sleutel maken van je tabel (PK = Primary KEY) In combinatie met A-I (auto increment), maak je een primaire sleutel dus aan die daarnaast ook uniek is

UNIQUE

Wanneer je een veld UNIQUE zet, dan zal je dubbele waarden vermijden in een database. Dit veld kan wel een NULL value bevatten.

INDEX

MySQL gebruikt indexen om snel rijen met specifieke kolomwaarden te vinden. Zonder index moet MySQL de hele tabel scannen om de relevante rijen te lokaliseren. Hoe groter de tafel, hoe langzamer hij zoekt.

A I

Wanneer aangevinkt zorgt A_I voor een oplopende nummering van een veld met een datatype integer.

COMMENTS

Hier kan je als developer een beschrijving geven van de bedoeling van een veld.

Code: CREATE TABLE

Een tabel kan je ook met code aanmaken. We maken dus hier gebruik van de mySQL-taal.

■ Oefening

Voorbeeld:

```
CREATE TABLE Klanten (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
voornaam VARCHAR(30) NOT NULL,
familienaam VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
)
```

Na het gegevenstype kunt u voor elke kolom andere optionele attributen specificeren:

- NOT NULL Elke rij moet een waarde voor die kolom bevatten, null-waarden zijn niet toegestaan
- DEFAULT-waarde Stel een standaardwaarde in die wordt toegevoegd als er geen andere waarde wordt doorgegeven
- UNSIGNED Gebruikt voor number datatypes , beperkt de opgeslagen gegevens tot positieve getallen en nul
- AUTO INCREMENT MySQL verhoogt automatisch de waarde van het veld met 1 telkens wanneer een nieuw record wordt toegevoegd
- PRIMARY KEY Wordt gebruikt om de rijen in een tabel uniek te identificeren. De kolom met PRIMARY KEY-instelling is vaak een ID-nummer en wordt vaak gebruikt met AUTO INCREMENT

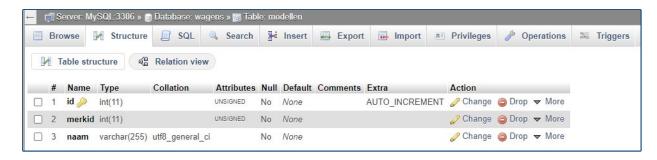
□Opdracht

Maak een nieuwe tabel **modellen** aan in de tabel wagens. Zorg dat er 3 velden aanwezig zijn. Het eerste veld **id** is een integer die positieve primaire sleutel is met autonummering en een grootte van 11 heeft. Het tweede veld is **merkid** en is een positieve integer met een grootte van 11 en kan geen null bevatten.

Het derde veld is naam en zal alle modelnamen bevatten. Dit is een string met een grootte van 255 en kan geen null bevatten.

Oplossing:

```
CREATE TABLE merken (
id INT(11) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
merkid INT(11) UNSIGNED NOT NULL,
naam VARCHAR(255) NOT NULL
)
```



Momenteel hebben we dus een database met de naam wagens. In deze database hebben we 2 tabellen, nl. merken en modellen. Wat we nog NIET hebben is een relatie tussen deze 2 tabellen.

Relaties

We sommen eerst de meest voorkomende relaties op die mogelijk zijn in een database tussen 2 of meerdere tabellen.

- één-op-één
- één-op-veel
- veel-op-veel

Relatie: één-op-veel

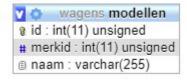
In ons voorbeeld hebben we bijvoorbeeld een model: AUDI Audi heeft meerdere modellen in zijn gamma: A3, A4, A5,...

Dit noemen we een één-op-veel-relatie, want het merk audi (één) heeft meerdere modellen (veel).

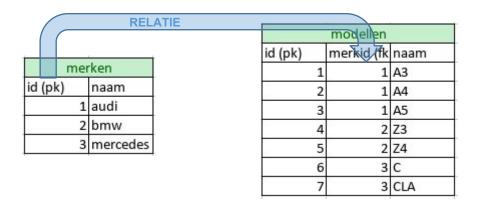
Een relatie dienen we dus ook te definiëren. Momenteel hebben we reeds alle velden in de tabellen gedefinieerd, maar nog niet de relatie.

Wanneer we de tabellen even schematisch bekijken in de designer tab op het niveau van de database dan kan je dit zien. Er is duidelijk geen link dus deze 2 tabellen. Deze lijken te 'zweven' naast elkaar



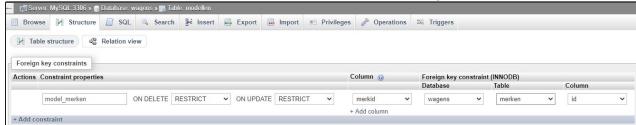


Hieronder zie je wat de bedoeling zou moeten zijn in een één-op-veel relatie datagewijs.



Zoals je kan zien hebben beide tabellen een unieke primary key. Het merkid echter is een sleutel die vreemd is aan de tabel modellen en die werd geërfd van de tabel merken. Deze sleutel noemen we een FOREIGN KEY.

Om nu de relatie te bewerkstelligen in phpMyAdmin dienen we de TWEEDE tabel modellen te selecteren en het tabblad STRUCTURE aan te klikken. Klik vervolgens op RELATION VIEW.



De opties die mogelijk zijn bij "on delete en on update" zijn:

RESTRICT

Wanneer je data die verbonden is in 2 tabellen zou willen wissen, dan zal dit niet lukken.

Wanneer je bijvoorbeeld audi in de tabel merken zou willen zal de database een restriction error geven omdat je modellen van audi in de tabel modellen zitten hebt.

CASCADE

Cascade zal wel wissen. Wanneer je audi zou wissen in de tabel merken dan zullen al zijn modellen in de tabel modellen ook gewist worden.

SET NULL

Wanneer je audi zou wissen in de tabel merken dan zullen de rijen in de tabel modellen niet gewist worden maar zal de inhoud van de rij op NULL worden geplaatst.

Wanneer we nu kijken naar de tabel modellen dan zie je een grijze sleutel staan die de foreign key weergeeft. Een gouden sleutel is de primary key van de tabel zelf.



Op het database niveau kan je terug naar het designer tabblad gaan en zie je het uiteindelijke resultaat van een één-op-veel relatie.



Code: FOREIGN KEY CONSTRAINTS

Wanneer je alle bovenstaande acties in code zou willen uitvoeren dan kan dit natuurlijk ook. WIS de tabel modellen uit de database.

Open nu het SQL tabblad op database niveau en voeg onderstaande code toe:

```
CREATE TABLE modellen (
id INT(11) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
merkid INT(11) UNSIGNED NOT NULL,
naam VARCHAR(255) NOT NULL,
FOREIGN KEY (merkid) REFERENCES merken(id)
ON UPDATE RESTRICT ON DELETE RESTRICT
```

Bovenstaande geeft hetzelfde resultaat terug en maakt de tabel modellen aan met de foreign key en zijn restriction.

CRUD

CRUD = CREATE READ UPDATE DELETE

Wanneer we over CRUD spreken, spreken we over alle mogelijk manipulaties die we op één of meerdere tabellen van een database kunnen uitvoeren.

De SQL syntax is de taal die we nodig zullen hebben om deze handelingen/manipulaties uit te voeren.

SELECT STATEMENT

We starten eerst met het READ gedeelte van de CRUD. Deze wordt in SQL uitgevoerd door het SELECT STATEMENT

Hieronder zie je de voorstelling van het volledige select statement die gebruikt KAN worden.

```
SELECT field references
```

```
[ALL | DISTINCT | DISTINCTROW ]
[HIGH PRIORITY]
[STRAIGHT JOIN]
[SQL SMALL RESULT] [SQL BIG RESULT] [SQL BUFFER RESULT]
[SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
select_expr [, select_expr] ...
[into_option]
[FROM table references
  [PARTITION partition list]]
[WHERE where condition]
[GROUP BY {col name | expr | position}
  [ASC | DESC], ... [WITH ROLLUP]]
[HAVING where_condition]
[ORDER BY {col name | expr | position}
  [ASC | DESC], ...]
[LIMIT {[offset,] row count | row count OFFSET offset}]
```

We hebben de SAKILA database samen geïnstalleerd. Deze gaan we nu gebruiken om het select statement in detail aan te leren.

OEFENINGEN

BASIS SELECT EN/OF ORDER BY

Belangrijke info

ORDER BY: zorgt voor het alfabetisch sorteren van velden. Opties zijn ASC en DESC DISTINCT: zorgt dat een VOLLEDIG record (rij) UNIEK is in het resultaat.

• Selecteer alle velden uit de tabel actor

```
SELECT *
FROM actor
```

- Selecteer enkel familienaam en voornaam uit de tabel actor SELECT actor.last name, actor.first name
 - FROM actor
- Selecteer enkel familienaam en voornaam uit de tabel actor en sorteer deze eerst volgens familienaam (A-Z) en vervolgens volgens voornaam (A-Z)

```
SELECT actor.last_name, actor.first_name
FROM actor
OPDER By actor last name ASC actor first name DESC
```

ORDER BY actor.last_name ASC, actor.first_name DESC

 Selecteer enkel de unieke familienaam en voornaam uit de tabel actor en sorteer Z-A SELECT DISTINCT actor.last_name, actor.first_name FROM actor
 ORDER BY actor.last name DESC, actor.first name DESC

WHERE CLAUSE

Belangrijke info

LOGISCHE OPERATOREN: and, or, IN, BETWEEN

Selecteer alle velden uit de tabel actor waar de familienaam gelijk is aan 'ZELLWEGER'
 SELECT *

```
FROM actor
WHERE actor.last_name = 'ZELLWEGER'
```

 Selecteer alle velden uit de tabel actor waar de familienaam gelijk is aan 'ZELLWEGER' of 'GUINESS'

```
oplossing 1:
SELECT *
FROM actor
WHERE actor.last_name = 'ZELLWEGER' or actor.last_name ='GUINESS'
oplossing 2:
SELECT *
FROM actor
WHERE actor.last_name IN ('ZELLWEGER', 'GUINESS')
```

 Selecteer alle velden uit de tabel actor waar de familienaam begint met Z of met D SELECT *

FROM actor

WHERE actor.last_name LIKE ('Z%') OR actor.last_name LIKE ('D%')

 Selecteer alle velden uit de tabel actor waar de familienaam DAVIS, DUKAKIS of ZELLWEGER is.

```
SELECT *
FROM actor
WHERE actor.last_name IN ('DAVIS','DUKAKIS','ZELLWEGER')
```

• Selecteer alle acteurs waar het id groter of gelijk is dan 5 en kleiner of gelijk is aan 10.

```
oplossing 1:
SELECT *
FROM actor
WHERE actor.actor_id >= 5 and actor.actor_id <= 10
oplossing 2:
SELECT *
FROM actor
WHERE actor.actor_id BETWEEN 5 and 10</pre>
```

FUNCTIONS

Belangrijke info

AS: het keyword AS wordt gebruikt om een veld een anderen naam te geven in een weergave of een nieuw samengesteld veld een naam te geven. AS is dus wat noemt een **alias** van een bestaand of nieuw veld.

In mySQL kunnen we daarnaast ook functies toepassen. Functies worden toegepast op bestaande velden in. Functies worden gekenmerkt door de naamgeving en ronde haakjes. Bijvoorbeeld: de functie MIN() haalt de laagste waarde uit een kolom.

Hieronder zie je een lijst van veel gebruikte functies

COUNT	De MySQL COUNT-aggregatiefunctie wordt gebruikt om het
	aantal rijen in een databasetabel te tellen.
MAX	Met de MySQL MAX-aggregatiefunctie kunnen we de hoogste
	(maximale) waarde voor een bepaalde kolom selecteren.
MIN	Met de MySQL MIN-aggregatiefunctie kunnen we de laagste
	(minimum) waarde voor een bepaalde kolom selecteren.
AVG	De MySQL AVG-aggregatiefunctie selecteert de gemiddelde
	waarde voor een bepaalde tabelkolom.
SUM	Met de MySQL SUM-aggregatiefunctie kunt u het totaal voor
	een numerieke kolom selecteren.
SQRT	Dit wordt gebruikt om een vierkantswortel van een bepaald getal
	te genereren.
RAND	Dit wordt gebruikt om een willekeurig getal te genereren met
	behulp van de MySQL-opdracht.
CONCAT	Dit wordt gebruikt om elke tekenreeks binnen een MySQL-
	opdracht samen te voegen .
LENGTH	Geeft de lengte van een strin gerug
CURDATE	Retourneert de huidige datum
CURTIME	Retourneert de huidige tijd
ADDDATE	Telt dagen bij een datum op
ADDTIME	Telt tijd bij tijd op
DATEFORMAT	Hier kan je de weergave van een datum mee wijzigen
DATE_SUB	Hier kan je 2 data van elkaar aftrekken
DATE	Geeft de datum van vandaag terug
WEEKDAY	Retourneert het weekdag nummer (index)
WEEK	Retourneert het weeknummer (index)

mySQL telt dus heel wat BUILT-IN functions. Er zijn er een paar honderd. Alle functies kun je hieronder terugvinden via de volgende link:

https://dev.mysql.com/doc/refman/8.0/en/udf-reference.html

- Selecteer het hoogste bedrag uit de tabel payment SELECT max(amount) from payment
- Hoeveel rijen telt de tabel payment SELECT max(amount) from payment