

# Databases 1

1ste jaars IB studenten introductie RDBMS, phpmyadmin, databases en queries.  
XAMPP, SELECT, COUNT(\*), LIKE, AND/OR, MIN(), MAX(), JOIN, ORDER BY

- [XAMPP Installeren](#)
- [1 Wat is een Database?](#)
- [2 SQL - SELECT-FROM-WHERE](#)
- [3 SELECT count\(\\*\)](#)
- [4 Instructiefilmpjes \(les 1,2,3\)](#)
- [5 Herhalingsoefeningen \(les 1,2,3\)](#)
- [6 Sorteren](#)
- [7 Join](#)
- [8 Aggregaat functies](#)
- [Instructiefilmpjes \(les 6,7,8\)](#)
- [Herhaling \(alles\)](#)
- [Student \(test\)database](#)

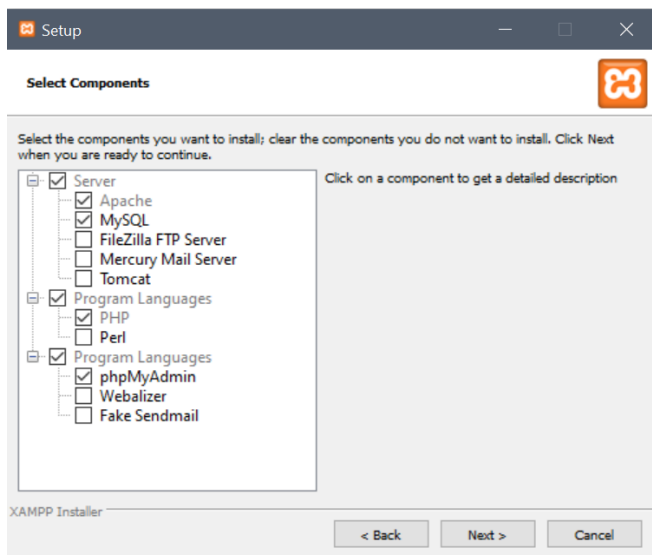
# XAMPP Installeren

In deze les leggen we uit hoe je XAMPP moet installeren. XAMPP is software voor de Mac en voor Windows waarin een RDMS (Database) en een webserver zit. Je kunt met deze software oefenen met het aanmaken en raadplegen van databases.

## Download de software

Google of download de juiste versie van [apachefriends](http://apachefriends.org) en voer de installatie uit.

Kies de volgende opties om te installeren.



Voor de installatie blijven de rest van de opties op de standaard waarden staan.

## Update

Als je al een keer eerder XAMPP had geïnstalleerd dan moet je de folder c:\xampp weggooien of beter nog een andere naam geven. Let op als je websites hebt ontwikkeld of databases hebt aangemaakt dan staan deze waarschijnlijk ergens in de folder c:\xampp. Als je deze folder dan weggooit, dan ben je dus ook jouw web sites en databases kwijt. Als je dit niet wilt dan zul je dus een back-up moeten maken. Kopieer hiervoor jouw web sites (html en php files) naar een andere plaats en maak met *phpmyadmin* een export van je databases.

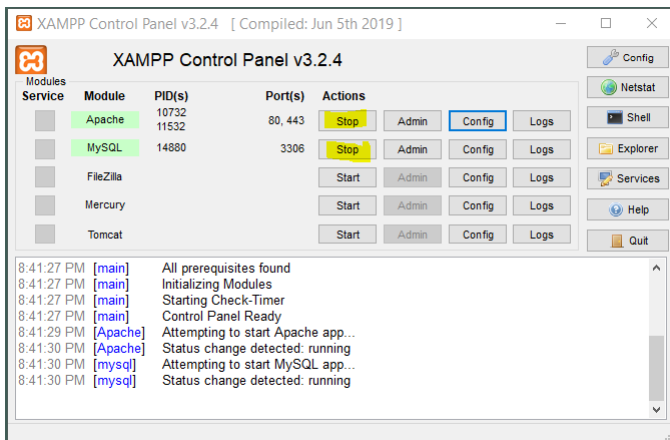
De Installatie spreekt voor zich je kunt overal de standaard opties aanhouden.

## Opstarten XAMPP

Als het goed is, is er tijdens de installatie een short cut gemaakt die heet *XAMPP Control*.

Deze shortcut voert `C:\xampp\xampp-control.exe` uit.

Druk op de eerste twee knopjes start naast Apache en MySQL. Hiermee start je de webserver en de database op. Als alles goed is dan zie je het volgende.



## Port conflict

Het kan zijn dat je foutmeldingen ziet, lees die goed. Zie je iets over port conflict op poort 443? Dat wordt mogelijk door VM player of door Steam veroorzaakt. Je kunt twee dingen doen. Je kunt VM player server (of Steam) stoppen of je kunt het poortnummer van de webserver veranderen. Dit laatste doe je als volgt.

Druk op de knop Config op de Apache regel en selecteer de `Apache (HTTP-SSL.conf)` optie. Er opent een window. Zoek het stukje dat er als volgt uitziet:

```
## When we also provide SSL we have to listen to the
# standard HTTP port (see above) and to the HTTPS port
#
Listen 443
```

en verander dat in

```
# When we also provide SSL we have to listen to the
# standard HTTP port (see above) and to the HTTPS port
#
Listen 1443
```

Er wordt dus een 1 voor de 443 geplaatst. Sluit de file en bewaar deze. Start de webserver opnieuw op.

..



# 1 Wat is een Database?

## Verschillende soorten databases

- Flat file (txt of cvs)
- Hierarchisch (bijvoorbeeld file systeem)
- Netwerk
- Relatieveel RDBMS
- Object database

De meest gebruikte vorm is het Relatieve Database Management Systeem, ook wel RDBMS. Voorbeelden van een RDBMS zijn: MySQL, MariaDB, Oracle, Microsoft SQL Server, SQLite en PostgreSQL.

## RDBMS

Een RDBMS is gebouwd op basis van relaties tussen de verschillende data eilandjes (tables).

SQL, Structured Query language is de taal waarmee je met de database praat in een RDBMS

Een RDBMS bestaat uit de volgende fysieke (hoofd)onderdelen:

1. **databases**

Dit is een verzameling tabellen die logisch bij elkaar horen.

2. **tables**

Dit is een groepje gegevens die bij iets horen, bijvoorbeeld bij de table persoon of klaslokaal.

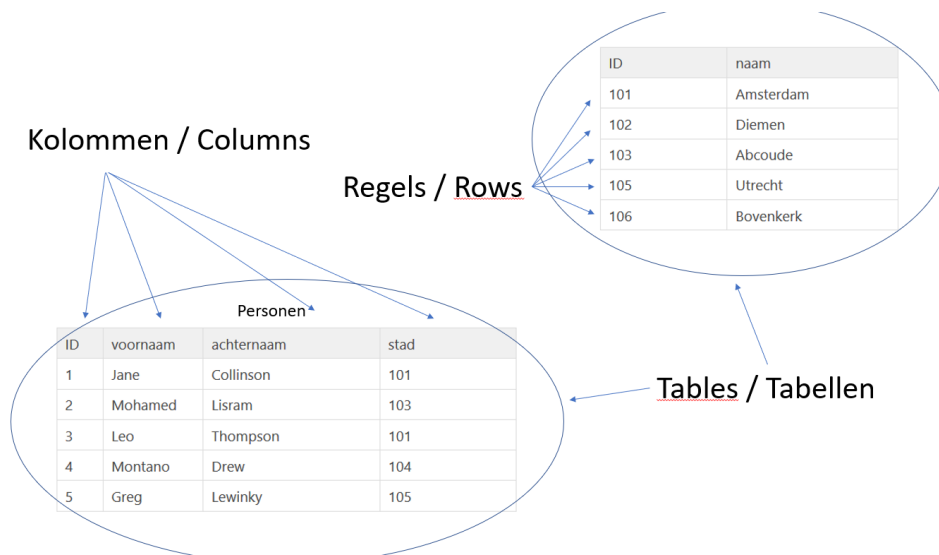
3. **columns** (fields)

Dit zijn de gevens die je van een tabel wilt vastleggen, bijvoorbeeld de naam van een persoon of het nummer van een klaslokaal.

4. **rows** (records)

Dit zijn de gegevens zelf. Bijvoorbeeld alle personen in de table persoon of alle klaslokalen in de table klaslokaal.

Een relatie in een RDBMS wordt gelegd door verschillende tabellen met elkaar te verbinden door middel van keys (veelal ID's).



In het plaatje zien we hoe tabellen, regels en kolommen (tables, rows and columns) in een database zich tot elkaar verhouden. We kunnen ook zien dat de twee (voorbeeld) tabellen een relatie met elkaar hebben. Je kunt bijvoorbeeld zien dat Jane in Amsterdam woont. De plaatsnaam is namelijk via het ID gekoppeld aan de tabel *Personen*.

## Database World

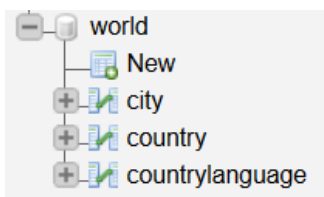
We gaan de database *World* maken en de tabellen en regels importeren.

In XAMPP start Apache en MySQL op en ga in je browser naar 127.0.0.1/phpmyadmin.

Aan de linkerkant kiezen we *new* of *nieuw*. Hiermee kunnen we een nieuwe database aanmaken. Maak de database *World* aan.

Kies het tabje *import* en importeer de file *world.sql*

Links in het menu zien we dat de *World* database uit drie tabellen bestaat:



In de volgende lessen gaan we gebruik maken van deze database.

Hieronder is een ERD (Entiteit Relatie Diagram) van de database weergegeven. In dit ERD kun je de relaties van de verschillende tabellen zien. Deze zijn belangrijk als we gegevens uit twee tabellen willen combineren.

<image-1594373220817.png>



# 2 SQL - SELECT-FROM-WHERE

*In de eerste serie lessen gaan we verschillende queries maken die betrekking hebben op één tabel.*

## SQL

SQL betekent Structured Query Language en is de taal om een database te beheren. Met SQL kan je

- De database structuur aanpassen
- De database inhoud veranderden
- De database raadplegen

## Database raadplegen

De SQL statements waarmee we de database kunnen raadplegen worden het meest gebruikt en de eenvoudige structuur van een raadpleeg statement is:

```
SELECT <kolomnaam<, <kolomnaam>, ....  
FROM <tabelnaam>  
WHERE <vergelijking>
```

Een vergelijking lijkt op JavaScript vergelijkingen, een paar voorbeelden:

WHERE naam = 'Netherlands'	kolom naam moet 'Netherlands' bevatten.
WHERE leeftijd > 18	kolom leeftijd moet groter dan 18 zijn (dus 19,20,21,...)
WHERE prijs < 4.99	kolom prijs is kleiner dan 4.99
WHERE prijs > 5 and prijs <10	kolom prijs is groter dan 5 en kleiner dan 10, en ligt dus tussen 5 en 10 in.
WHERE prijs >=5 and prijs <=10	kolom prijs is minimaal 5 en maximaal 10
WHERE continent='Europe' or continent='Africa'	kolom continent is Europe of Africa

WHERE naam like 'Ned%'	kolom naam begint met een Ned (% betekent <i>any characters</i> )
WHERE naam like '%n%'	kolom naam bevat een n
WHERE naam like '%e'	kolom naam eindigt op een e
WHERE naam like 'A%' or naam like 'B%'	kolom naam begint met een A of met een B
WHERE naame like 'A%' and naam like 'B%'	kolom naam begint met een A en met een B; wat klopt hier niet?

We gaan kijken naar de Country tabel, deze heeft de volgende structuur.

TABEL <b>country</b>		
Kolomnaam	Beschrijving	Voorbeeld
Code	Primary Key (Country Code)	NLD
Name	Naam van het land	Netherlands
Continent	Continent	Europe
Region	Regio	Western Europe
SurfaceArea	Oppervlakte (in KM <sup>2</sup> )	41526.00
IndepYear	Jaar van onafhankelijkheid	1581
Population	Bevolkingsgrootte, inwoneraantal	15864000
LlfeExpectancy	Levensverwachting	78.3
GNP	Bruto Nationaal Product	371362.00
GNPOld	Oud Bruto Nationaal Product	360478.00
LocalName	Lokale naam	Nederland
GovernmentForm	Regeringsvorm	Constitutional Monarchy
HeadOfState	'Baas' van het land	Alexander
Capital	Foreign Key naar City	5
Code2	Officiële landsafkorting (bijv NL)	2

We gaan nu queries uitvoeren op de database world in phpmysql.

Ga naar phpmySQL en selecteer de *World* database (links) kies daarna het tabje SQL. Voer de volgende query uit.

```
SELECT Name, Continent
FROM country
WHERE name like 'F%'
```

Wat zie je?

## Opdrachten

Nu mag jij queries bedenken en uitvoeren. Kopieer de *queries* in een text file en stuur die via Teams op.

1. Maak een query waarmee je alle landennamen laat zien waarvan de naam van het land met een B begint.  
Schrijf de query op je antwoordenblad.
2. Maak een query waarmee je alle landennamen en de continenten laat zien en waarvan de naam van het land met een B begint.  
Schrijf de query op je antwoordenblad.
3. Maak een query waarmee je alle landennamen, de continenten en het aantal inwoners laat zien en waarvan de naam van het land met een B begint.  
Schrijf de query op je antwoordenblad.
4. In welke landen is de levensverwachting (LifeExpectancy) lager dan 40.  
Schrijf de query op je antwoordenblad.
5. Wat denk je zijn er landen waar geen mensen wonen? Maak een query en selecteer alle landen waar geen mensen wonen. Kijk goed in de tabel welke kolommen je kan gebruiken.  
Schrijf de query op je antwoordenblad.
6. Welke landen hebben meer dan 0 en minder dan 100 inwoners?  
Schrijf de query op je antwoordenblad.
7. Maak een query die alle landen (naam en lokale name) laat zien die op het Continent Europa liggen.  
Schrijf de query op je antwoordenblad.

8. Maak een query die alle landen en hun regeringsvorm laat zien die op het Continent Europa liggen.  
Schrijf de query op je antwoordenblad.
  
9. Hoeveel landen hebben een inwoneraantal meer dan 1 000 000 000 (dat is 1 miljard mensen; dat is 1000-miljoen)? Let op als je dit getal intypt dat je de spaties weglaat.  
Schrijf de query die je hebt gebruikt op je antwoordenblad.
  
10. De oppervlakte van de landen wordt in  $\text{km}^2$  gemeten. Dat is een vierkante kilometer. Dat is een stuk land van 1000 meter in het vierkant. Een land met een oppervlakte van  $1 \text{ km}^2$  is dus heel klein; als je doorloopt, kun je er in 10 minuten dwars doorheen lopen. Zijn er landen met een oppervlakte minder dan  $1 \text{ km}^2$ ? Maak een query om dat te bepalen.  
Schrijf de query die je hebt gebruikt op je antwoordenblad.

--

# 3 SELECT count(\*)

## SELECT count(\*) en SELECT \*

In de select kunnen we ook alleen het aantal rijen dat we terugkrijgen tellen. Stel we willen weten hoeveel lander er in de tabel *Country* staan dan kan met de volgende SQL Query.

```
SELECT count(*)  
FROM Country
```

Als je alle kolommen wilt afdrucken dan kun je ook alleen \* gebruiken:

```
SELECT *  
FROM Country
```

Probeer nu zelf:

```
SELECT count(*)  
FROM country  
WHERE name like 'F%'
```

en

```
SELECT *  
FROM country  
WHERE name like 'F%'
```

Zie je wat het verschil is?

Als je een SELECT count(\*) gebruikt dan wil je dus een aantal weten. Je krijgt dan alleen een getal. Bijvoorbeeld alle leerlingen die een voldoende hebben gehaald.

```
SELECT count(*)  
FROM leerlingen  
WHERE cijfer >= 5.5
```

De uitkomst van deze query is bijvoorbeeld 10. Er zijn in dat geval 10 leerlingen die een 5.5 of hoger hebben gehaald. Na de count(\*) kan je geen andere kolommen namen meer opvragen.

SELECT count(\*), name (is FOUT!)

Kan dus niet want je kan niet alleen het aantal van iets afdrukken en dan de name ook nog afdrukken. Je wilt of de name zien of alleen het aantal.

`count(*)` staat altijd alleen en eenzaam na de `SELECT`.

(Later zul je zien dat er op deze regel uitzonderingen bestaan, maar dat komt pas als we meer ingewikkelde queries gaan maken).

## Opdrachten

(schrijf van elke vraag de query die je hebt gebruikt op en maak gebruik van **count(\*)** )

1. Hoeveel landen zijn waarvan de naam begint met een N?
2. Hoeveel landen hebben een oppervlakte van minder dan 100 km<sup>2</sup>?
3. Hoeveel landen hebben een inwoneraantal meer dan 0 en minder dan 1000 inwoners?
4. Hoeveel landen liggen er op het continent 'Europe'?
5. Hoeveel landen liggen er op het continent 'South America' met meer dan 10 miljoen inwoners?
6. Hoeveel landen liggen er op het continent 'Africa' met meer dan 10 miljoen inwoners?
7. Hoeveel landen liggen er op het continent 'Europe' die kleiner zijn in oppervlakte ('SurfaceArea') dan Nederland?
8. Hoeveel landen liggen er op het continent 'Europe' die groter zijn in inwoneraantal ('Population') dan Nederland?

--

# 4 Instructiefilmpjes (les 1,2,3)

## Instructiefilmpjes op YouTube over les 1, 2 en 3

*Hieronder staan links naar drie korte instructiefilmpjes waarin de belangrijkste zaken van les 1,2 en 3 nog een keer worden uitgelegd.*

### **World Database, hoe werkt PHPMyAdmin en wat is het verschil tussen = en like?**

<https://youtu.be/2ISqv53kP9s>

<https://www.youtube.com/embed/2ISqv53kP9s>

### **Wat is het verschil tussen SELECT count(\*) en SELECT \***

[https://youtu.be/DXN\\_EV6knsI](https://youtu.be/DXN_EV6knsI)

[https://www.youtube.com/embed/DXN\\_EV6knsI](https://www.youtube.com/embed/DXN_EV6knsI)

### **Wanneer gebruik je AND en wanneer OR bij de WHERE in een Query?**

<https://youtu.be/tz1BSi-j0yw>

<https://www.youtube.com/embed/tz1BSi-j0yw>

### **Sorteren in SQL**

<https://youtu.be/bB28xwLecAk>

<https://www.youtube.com/embed/bB28xwLecAk>

## **JOIN...ON in SQL**

<https://youtu.be/mpRqK12KNVw>

<https://www.youtube.com/embed/mpRqK12KNVw>

## **Aggregate functies - min() en max()**

<https://youtu.be/SP9ATZor3kl>

<https://www.youtube.com/embed/SP9ATZor3kl>

<https://www.maxdata.ovh/cnt.php?id=DBvIB-filmpjes>

# 5 Herhalingsoefeningen (les 1,2,3)

Hieronder ter voorbereiding op de eerste test nog enkele oefeningen. Kijk nog een keer naar de filmpjes uit de vorige les. Daarin worden de belangrijkste- en moeilijkste dingen nog een keer kort uitgelegd.

Kijk voor je hier aan begint de instructiefilmpjes uit de [les 4](#) door!

Lees goed wanneer je alleen het aantal moet afdrukken gebruik dan count(\*), dit wordt ook in de filmpjes nog een keer uitgelegd.

De vragen gaan in deze oefening over de *country* tabel uit de *world* database.

1. Maak één query waarin je het inwoneraantal van Frankrijk en Spanje laat zien.  
Test de query uit en zet de juiste query op je antwoordenblad.
2. Maak één query waarin je het aantal landen laat zien (dus alleen een getal) dat tussen de 1 en 2 miljoen inwoners heeft. Dus meer dan 1 miljoen en minder dan 2 miljoen.  
Test de query uit en zet de juiste query op je antwoordenblad.
3. Maak één query waarin je het aantal landen laat zien (dus alleen een getal) dat tussen de 1 en 2 miljoen inwoners heeft en dat op het continent 'Africa' ligt.  
Test de query uit en zet de juiste query op je antwoordenblad.
4. Maak een query waarin je alle landen waarvan de naam met een 'C' begint, afdruckt.  
Test de query uit en zet de juiste query op je antwoordenblad.
5. Maak een query waarin je het aantal landen (dus alleen getal) laat zien waarvan de naam met een A, of een B of een C begint.  
Test de query uit en zet de juiste query op je antwoordenblad.
6. Welk continent heeft de meeste landen, het continent 'Africa' of 'Asia'?  
Schrijf het antwoord en de query's die je hebt gebruikt om dit te bepalen op je antwoordenblad.

7. Welk continent heeft meer landen waarvan de eerste letter een A is, 'Europe' of 'Africa'? Schrijf het antwoord en de query's die je hebt gebruikt om dit te bepalen op je antwoordenblad.

--

# 6 Sorteren

In deze les leren we hoe we het resultaat van een query kunnen sorteren. Tot nu toe hebben we geoefend met de table *country* (land). We gaan nu ook naar de table *city* (stad) kijken.

In les 1 hebben we de World database geïmporteerd. In deze World database zitten drie tabellen. In *phpmyadmin* kun je zien welke tabellen dit zijn.

In deze les gaan we kijken naar de tabel *country*.

TABEL <b>city</b>		
Kolomnaam	Beschrijving	Voorbeeld
CountryCode	Foreign Key naar table country	NLD
District	Provincie	Gelderland
ID	Primary key	1
Name	Naam van de stad	Amsterdam
Population	Bevolkingsgrootte, inwoneraantal	153491

Met de foreign key Country Code wordt de table *city* gekoppeld aan de table *country*. Hierover gaat de volgende les. We gaan nu eerst kijken naar de tabel *city*.

## Sorteren

<https://www.youtube.com/embed/bB28xwLecAk>

Als laatste regel in een query kunnen we aangeven dat de regels die een query teruggeeft worden gesorteerd. Dat gaat als volgt:

**ORDER BY <kolomnaam> ASC**

Dit sorteert op de kolomnaam <kolomnaam> in oplopende volgorde, dus van laag naar hoog.

**ORDER BY <kolomnaam> DESC**

Dit sorteert op de kolomnaam <kolomnaam> in aflopende volgorde, dus van hoog naar laag.

Voorbeeld:

```
SELECT Name, Population
FROM country
ORDER BY Population DESC
```

Deze query drukt alle landen af gesorteerd op inwoneraantal (population) van hoog naar laag.

*Vergeet niet de video (die hierboven staat) te bekijken voordat je aan de opgave begint. Veel antwoorden staan namelijk in de video.*

## Opgave 1

Met behulp van de query hierboven kun je bepalen welk land de meeste inwoners heeft. Voer de query uit en schrijf op je antwoordenblad de naam van het land met de meeste inwoners.

## Opgave 2

Welk land heeft de minste inwoners (0 niet meegeteld).

Tip: moet je oplopend of aflopend sorteren om te bepalen? Als je dit niet weet probeer dan beide methodes.

Schrijf de query op die je hebt gebruikt om dit te bepalen op je antwoordenblad.

## Opgave 3

Deze vraag gaat over steden en niet meer over landen. We gaan gebruik maken van de tabel *city*.

Hoeveel steden staan er in de database World?

Maak een query waarbij de alleen het aantal steden laat zien.

Schrijf de query op je antwoordenblad.

## Opgave 4

Hoeveel steden staan er in de database waarvan de naam van de stad begint met een 'X' ?

Maak een query waarbij je alleen het aantal steden laat zien die met een 'X' beginnen.

Schrijf de query op je antwoordenblad.

## Opgave 5

Alle steden in de database uit Nederland hebben een CountryCode 'NLD'. Maak een query met een lijst van alle Nederlandse steden.

Schrijf de query op je antwoordenblad.

## Opgave 6

Hoeveel inwoners (population) heeft de stad Eindhoven?

Schrijf de query op je antwoordenblad.

## Opgave 7

Welke stad uit Nederland in de database heeft de *meeste* inwoners?

Tip: denk aan de sorteerfunctie.

Schrijf de query op je antwoordenblad.

## Opgave 8

Welke stad uit Nederland in de database heeft de *minste* inwoners?

Tip: denk aan de sorteerfunctie.

Schrijf de query op je antwoordenblad.

## Opgave 9

We hebben nu kennis gemaakt met de tabellen country en city. Er zit nog een derde table in de World database. Zoek met phpmyadmin op wat deze derde table is.

Schrijf op je antwoordenblad op welke drie tabellen er in de World database zitten.

## Opgave 10

Wat komt er in een query na een FROM?

Schrijf je antwoord op je antwoordenblad.

--



# 7 Join

In deze les leer je hoe je de gegevens uit twee tabellen uit de database kunt combineren

## Uitleg

In een relationele database zitten meerdere tabellen en met behulp van relaties kun je deze gegevens combineren. In deze les leren we op een hele eenvoudige manier hoe je dat kunt doen.

Stel je hebt twee tabellen: *persoon* en *stad*.

De tabel *persoon* bestaat uit 4 rows en 5 kolommen.

ID	voornaam	achternaam	stad
1	Jane	Collinson	101
2	Mohamed	Lisram	103
3	Leo	Thompson	101
4	Montano	Drew	104
5	Greg	Lewinky	105

De tabel *stad* bestaat uit 5 rows en 2 kolommen

ID	naam
101	Amsterdam
102	Diemen
103	Abcoude
105	Utrecht
106	Bovenkerk

Vraag 1: weet jij waar Jane Collinson woont (goed kijken)?

Als je ziet waar Jane woont dan heb je waarschijnlijk gezien dat de kolom *stad* uit de tabel *persoon* verwijst naar de kolom *ID* uit de tabel *stad*. Of in het kort:

<https://www.youtube.com/embed/mpRqK12KNVw>

## Primary Key en Foreign Key

De kolom `persoon.stad` verwijst naar een andere tabel dat heet een foreign key (vreemde key, een key van een andere tabel).

De kolom `ID` in de tabel `stad` is een primary key. Een hoofd sleutel. Een primary key heeft voor elke regel een unieke waarde en je kan er dus altijd precies één rij mee aanwijzen.

Een foreign key verwijst naar een aan andere primary key, maar hoeft niet uniek te zijn. Kijk maar in het voorbeeld, Jane en Leo wonen allebij in stad 101 (Amsterdam) en hebben dus dezelfde foreigen key.

## Opdracht (in de les)

Maak een nieuwe database aan in phpmyadmin en noem de database `persoon`.

We gaan nu een database en twee tabellen maken in phpmyadmin.

1. Download het bestandje [persoon.sql](#) aan de linkerkant van deze pagina.
2. Start XAMPP (Apache en MySQL).
3. Kies in het linker gedeelte 'new' om een nieuwe database aan te maken.
4. Maak een database met de naam `persoon` aan.
5. Ga naar het tabje import en selecteer browse; ga naar het bestand `persoon.sql` en importeer deze.

(*optioneel*) Als je wilt, kun je de tabellen ook handmatig aanmaken. Je moet daarvoor de volgende twee tabellen maken:

1. **persoon** met de kolommen **ID** (integer), **voornaam** (varchar(80)), **achternaam** (varchar(80)), en **stad** (integer)
2. **stad** met de kolommen **ID** (integer) en **naam** (varchar(80)).
3. vul de tabellen nu met de waarden zoals hierboven aangegeven.

Voer nu de volgende query uit:

```
SELECT *  
FROM persoon  
JOIN stad  
ON persoon.stad = stad.id
```

## Opdracht (huiswerk)

Start XAMPP (Apache en SQL server) en ga naar phpmySQL <http://localhost/phpmyadmin/>. Ga naar de database persoon (linkerkant van scherm) en ga naar het tabje SQL.

Tip: voor opgave 4 t/m 8 moet je een join gebruiken. Dit wordt in het [filmpje](#) (hierboven) uitgelegd.

1. Maak een query die alle plaatsnamen uit de database (persoon) laat zien.  
Schrijf de query op je antwoordenblad.
2. Maak een query die alle voornamen en achternamen van alle personen uit de database afdrukt.  
Schrijf de query op je antwoordenblad.
3. Maak een query die alle voornamen en achternamen laat zien van alle personen waarvan de voornaam met een M begint.  
Schrijf de query op je antwoordenblad.
4. Maak een query van alle voornamen en achternamen en woonplaatsen van alle personen uit de database.  
Schrijf de query op je antwoordenblad.
5. Maak een query die alle voornamen afdrukt van alle personen uit Amsterdam.  
*Zoek op op welke kolomnaam je moet zoeken; waar vind je de plaatsnaam en hoe heet die? Let ook op dat Amsterdam in de database met een kleine letter begint.*  
Schrijf de query op je antwoordenblad.
6. Maak een query die alle voornamen en achternamen afdrukt van alle personen uit Abcoude.  
Schrijf de query op je antwoordenblad.
7. Maak een query die telt hoeveel personen er uit Amsterdam komen.  
Schrijf de query op je antwoordenblad.

8. Maak een query die telt hoeveel personen er uit Amsterdam of Utrecht komen. Schrijf de query op je antwoordenblad.

<https://www.maxdata.ovh/cnt.php?id=DBvIB-join>

# 8 Aggregaat functies

Aggregaat functies zijn functies die op de hele data-set betrekking hebben. Zo kun je alle waarden van een kolom optellen, de kleinste of de grootste waarde bepalen.

## Aggregaat functies.

Met de functie `sum(<kolomnaam>` kunnen we het totaal van die kolom berekenen. Bijvoorbeeld:

```
SELECT sum(Population)
FROM Country
WHERE Continent = 'Europe'
```

Hieronder worden de belangrijkste aggregaat functies beschreven:

Aggregaat	functie	voorbeeld	wat toont dit voorbeeld?
sum	optellen	<code>sum(population)</code>	totaal aantal inwoners
min	minimale waarde	<code>min(population)</code>	bepaal kleinste aantal inwoners
max	maximale waarde	<code>max(population)</code>	bepaal grootste aantal inwoners
count	aantal	<code>count(*)</code>	bepaal aantal rijen
avg	gemiddelde	<code>avg(population)</code>	gemiddeld aantal inwoners

Stel je wilt weten welk land in West Europa de meeste inwoners heeft dan kun je met een query bepalen wat het grootste aantal inwoners is. Maar dan weet je nog niet welk land dit is. Dit kan in een ingewikkelde query, maar het kan ook eenvoudiger met twee queries:

```
SELECT max(Population)
FROM Country
WHERE Region = 'Western Europe'
```

Het antwoord op deze query is: 82164700

Nu kunnen we het land opzoeken:

```
SELECT *
FROM Country
WHERE Region = 'Western Europe'
```

```
and Population = 82164700
```

We kunnen nu deze twee queries combineren. We vullen dan in plaats van het getal de eerste query in en gebruiken dan wel haakjes om de tweede query heen.

```
SELECT *  
FROM Country  
WHERE Region = 'Western Europe'  
and Population = (SELECT max(Population) FROM Country WHERE Region = 'Western Europe')
```

## Voorbeeld filmpje - we gaan op zoek naar het grootste en kleinste land van west-europa.

<https://www.youtube.com/embed/SP9ATZor3kl>

## Opgaven

1. Wat is het grootste land in oppervlakte (SurfaceArea) in de regio 'West Europe'?  
Schrijf de query(s) die je hebt gebruikt op.
2. Wat is het kleinste land in oppervlakte op het continent 'South America'?  
Schrijf de query(s) die je hebt gebruikt op.
3. Wat is het grootste land in oppervlakte op het continent 'Africa'?  
Schrijf de query(s) die je hebt gebruikt op.
4. Hoeveel mensen wonen er in totaal in de regio 'West Europe'?  
Schrijf de query(s) die je hebt gebruikt op.
5. Hoeveel mensen wonen er in totaal in de regio 'North America'?  
Schrijf de query(s) die je hebt gebruikt op.

--

# Instructiefilmpjes (les 6,7,8)

## Instructiefilmpjes op YouTube over les 6, 7 en 8

*Hieronder staan links naar drie korte instructiefilmpjes waarin de belangrijkste zaken van les 6, 7 en 8 nog een keer worden uitgelegd.*

### **Sorteren in SQL**

<https://youtu.be/bB28xwLecAk>

<https://www.youtube.com/embed/bB28xwLecAk>

### **JOIN...ON in SQL**

<https://youtu.be/mpRqK12KNVw>

<https://www.youtube.com/embed/mpRqK12KNVw>

### **Aggregate functies - min() en max()**

<https://youtu.be/SP9ATZor3kl>

<https://www.youtube.com/embed/SP9ATZor3kl>

Instructie

# Herhaling (alles)

We gaan alles wat we tot nu toe hebben geleerd nog een keer oefenen.

## Opgaven

1. Je hebt een restaurant en wil je van al je menu's vastleggen hoe ze heten en wat ze kosten en wat de voorraad is. Je wilt per menu ook een categorie vastleggen. Er zijn vijf categorieën: voorgerecht, hoofdgerecht, nagerecht, drank, overig. Maak een tabel waarin je deze gegevens kan vastleggen. Benoem de naam van de tabel en de naam van de kolommen.

Tabelnaam: .....

Kolomnaam: .....

Kolomnaam: .....

Kolomnaam: .....

Kolomnaam: .....

2. Je hebt een tabel en die heet *Persoon*. De volgende query geeft als resultaat 12:

```
SELECT count(*)  
FROM Persoon
```

Kun je op basis van deze informatie vaststellen hoeveel kolommen de tabel *Persoon* heeft?

Kun je op basis van deze informatie vaststellen hoeveel rijen deze tabel *Persoon* heeft?

3. Beschrijf in eigen woorden wat de volgende query laat zien:

```
SELECT voornaam, achternaam, schoolvak  
FROM student  
WHERE cijfer < 5.5
```

4. Gebruik de world table *Country* uit de *world* database. Wat is de levensverwachting in België (Belgium)?
5. Hoeveel landen zijn er in oppervlakte kleiner dan Nederland? Schrijf de queries die je hebt gebruikt op.

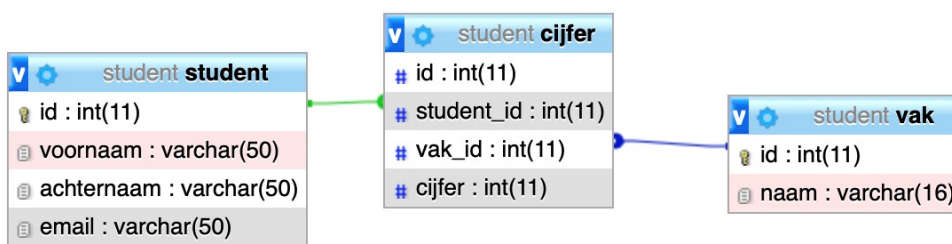
6. Het GNP geeft aan hoeveel een land verdient. Het geeft op een bepaalde manier aan hoe rijk een land is. Wat is het rijkste land uit de regio 'Western Europe'?
7. Life Expectancy, oftewel levensverwachting is de leeftijd dat een gemiddeld persoon wordt.  
In welke landen in West Euroep is de levensverwachting (LifeExpectancy) hoger dan die in Nederland?  
Schrijf de queries die je hebt gebruikt op.
8. In welk land is de levensverwachting het hoogste?  
Schrijf de queries die je hebt gebruikt op.
9. Hoeveel landen zijn er waarvan de naam met een A begint?  
Schrijf de queries die je hebt gebruikt op.
10. Waarvan denk je dat er de meeste landen bestaan. Landen waarvan de naam met een A, een B of een C begint? Van welke beginletter (A, B of C) bestaan de meeste landen en hoeveel zijn dat er?
11. Hoeveel landen zijn er waarvan de naam een X bevat.  
Schrijf de queries die je hebt gebruikt op.
12. Welk(e) land(en) zijn er waarvan de naam minimaal 6 a's bevat?  
Schrijf de queries die je hebt gebruikt op.

# Student (test)database

De student database is een eenvoudige database met een N:M relatie.

De database heeft een tabel student die een N:M relatie heeft met de tabel vak. Een student heeft 0,1 of meer vakken en een vak wordt door 0,1, of meer studenten gevolgd. In het voorbeeld staan 100 studenten die allen precies 9 vakken hebben gevolgd. De cijfers voor de vakken staan in de koppeltabel.

De database ziet er als volgt uit:



Aanmaken student database.

1. [Download de file student.sql](#)
2. [ga naar 127.0.0.1/phpmysql](#)
3. klik in het linker gedeelte new (of nieuw) en vul daar student in



4. Druk op Create
5. Ga naar het tabje import

6. Importeer de file: druk op browse (of Choose file) en selecteer de file student.sql (uit stap 1).
7. Druk op go en de database is geïmporteerd.

--