

# Het ontwerpen van een relationele database

lid	
LEENERSPAS	VARCHAR(6)
NAAM	VARCHAR(30)
STRAAT	VARCHAR(25)
POSTCODE	VARCHAR(7)
WOONPLAATS	VARCHAR(20)
GEB_DATUM	VARCHAR(6)

lening	
EXEMPLAARCODE	VARCHAR(11)
LEENERSPAS	VARCHAR(6)
EINDDATUM	VARCHAR(6)

exemplaar	
EXEMPLAARCODE	VARCHAR(11)
ISBN	VARCHAR(15)
DRUK	VARCHAR(3)
BOEKSOORT	VARCHAR(15)
LOKATIE	VARCHAR(20)
INVOERDATUM	VARCHAR(6)

uitgave	
ISBN	VARCHAR(15)
DRUK	VARCHAR(3)
PLAATS	VARCHAR(25)
JAAR	VARCHAR(4)

boekcode	
ISBN	VARCHAR(15)
INTERN_BOEKNR	VARCHAR(8)

boek	
INTERN_BOEKNR	VARCHAR(8)
TITEL	VARCHAR(50)
AUTEURS	VARCHAR(50)



informatica 5 HAVO en 6 VWO

## Introductie

In deze cursus leer je een database ontwerpen. Een database of gegevensbank is een hoeveelheid gegevens die op een of andere manier bij elkaar horen. Databases spelen een belangrijke rol. Wij zullen ons vooral concentreren op databases die gebruikt worden in samenwerking met een website. Dit kan op vele manieren. Een bekend voorbeeld is marktplaats.nl. De gegevens van de producten staan in een database. In deze cursus relationele database ontwerpen, leer je zelf een database ontwerpen.

Een relationele database bestaat uit een verzameling tabellen die een relatie met elkaar hebben.

We beginnen met een klein beginprobleem. Stel, we willen van mensen het volgende bijhouden: naam, adres, woonplaats en geboortedatum. We willen ook de e-mailadressen en telefoonnummers vastleggen. Je zou de volgende tabel kunnen maken:

naam	adres	woonplaats	Geb_datum	e-mail	telefoon
Mark Rutte	Lijnbaanstraat 67	Den Haag	14-02-1967	markrutte@vvd.nl	070-342323

Maar wat gaan we doen als bijvoorbeeld Mark Rutte meer e-mailadressen heeft en op meerdere telefoonnummers bereikbaar is? Een rij er onder aanmaken?

naam	adres	woonplaats	Geb_datum	e-mail	telefoon
Mark Rutte	Lijnbaanstraat 67	Den Haag	14-02-1967	markrutte@vvd.nl	070-342323
Mark Rutte	Lijnbaanstraat 67	Den Haag	14-02-1967	markrutte@vvd.nl	06-24343434

Wel erg onhandig! Je moet nu de andere gegevens ook weer opnieuw invullen. Je krijgt met het opnieuw invullen een grote kans dat er een intikfout gemaakt wordt, je gegevens raken als het ware vervuild! Dat is strijdig met het principe dat informatie eenduidig vastgelegd moet zijn.

Een andere oplossing zou zijn dat het aantal kolommen verder uitgebreid wordt:

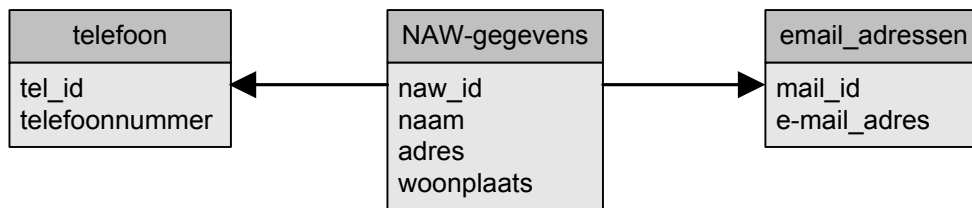
naam	adres	woonplaats	Geb_datum	e-mail	tel1	tel2	tel3
Mark Rutte	Lijnbaanstraat 67	Den Haag	14-02-1967	markrutte@vvd.nl	070-342323	06-24343434	06-53758999

Is ook niet echt handig. Je moet zoveel kolommen aanmaken als de persoon met de meeste telefoonnummers. Misschien heeft er iemand wel 8 en verder bijna iedereen 2. Dat betekent veel lege plekken in je tabel. Dat is zonde van de ruimte. Dit speelt natuurlijk vooral een rol als je veel mensen in je tabel hebt staan.

Het zoeken in zo'n tabel is ook niet erg makkelijk. Met het "blote oog" gaat het nog wel maar voor een computer is het niet makkelijk. Wat wordt gevraagd? tel1, tel2, tel3,...?

Kortom dit is geen goede oplossing, vooral voor computers niet. In de jaren zeventig is hiervoor dan ook een goede oplossing bedacht. Een verzameling tabellen die bij elkaar horen en die met elkaar in verbinding staan, relaties hebben. We noemen zo'n oplossing een relationele gegevensbank of relationele database.

De oplossing van het probleem kan als volgt:



We maken gewoon drie tabellen. In de tabel met de adresgegevens nemen we een extra kolom op: `naw_id`. Dit doen we om een verwijzing te maken naar de andere 2 tabellen. Dat kun je beter niet met de naam doen. Sommige namen komen nu eenmaal vaak voor. Het is beter om elke rij in de tabel met NAW-gegevens te voorzien van een code, bijvoorbeeld gewoon een nummer:

naw_id	naam	adres	woonplaats	Geb_datum
1	Mark Rutte	Lijnbaanstraat 67	Den Haag	14-02-1967
2	Maxime Verhagen	Heerlenseweg 12	Brunssum	14-09-1956

De tabel met telefoongegevens ziet er dan als volgt uit:

tel_id	telefoon
1	070-342323
1	06-24343434
1	06-53758999
2	045-315220
2	06-44231212

De tabel met e-mailadressen kan er dan zo uit zien

mail_id	e-mail
1	markrutte@vvd.nl
1	premier@alegemenezaken.nl
2	verhagen@cda.nl
2	minister@ecomische_zaken.nl

Zo heb je eigenlijk alle problemen opgelost. Je hoeft de gegevens maar één keer in te vullen en je gebruikt precies zo veel ruimte als nodig; tabellen worden langer en niet breder. Tevens kun je zo vrij goed zoeken in je gegevens.

Het beheer van de tabellen gebeurt met speciale programmatuur: een databasemanagementsysteem of, nog beter, een relationeel database management systeem (RDBMS). Een voorbeeld van zo'n programma is Access van Microsoft. Er zijn veel RDBMS's op de markt. Access is een voorbeeld van een RDBMS dat geschikt is voor kleine kantoren. Grote organisaties gebruiken andere systemen. Voorbeelden zijn bijvoorbeeld Oracle, DB2 van IBM en SQL-Server van Microsoft.

In deze cursus gebruiken we MySQL (dat echt uitgesproken moet worden als: mei es cue el). Een RDBMS dat vooral op het Internet zeer populair is. Een betrouwbaar en gratis systeem!

In deze cursus leer je een relationele database ontwerpen. Het bovenstaande voorbeeld is erg eenvoudig. Dat je drie tabellen nodig hebt, ligt nogal voor de hand. Je kunt je voorstellen dat het bij een ingewikkelder probleem lastiger is, om te bepalen wat in welke tabel moet

komen te staan. We gaan daarom gebruik maken van een bepaalde manier van werken. Er zijn ook professionele ontwikkelmethodieken. Het nadeel hiervan is, dat je eerst veel moet leren voordat je het kunt gebruiken. Een veel gebruikte methodiek in de professionele wereld is UML.

We gaan het in deze cursus een “beetje uit het handje en een beetje methodisch” doen.

Een relationele database bestaat dus uit een verzameling tabellen. In deze tabellen zijn gegevens van dezelfde soort in *rijen* of in *records* opgeslagen. In één tabel staan altijd dezelfde soort gegevens. In de tabel over de NAW-gegevens was dat al goed te zien.

We hebben ook een `naw_id` ingevoegd. Een soort label waardoor je steeds de rijen in een tabel van elkaar kunt onderscheiden. Meestal gebruiken we hier een nummer of een code voor. Denk aan je leerlingnummer, je BSN (Burger Service Nummer) of bestelnummer van een artikel bij een webwinkel.

We noemen zo'n label meestal een primaire sleutel of in het Engels, *primary key*.

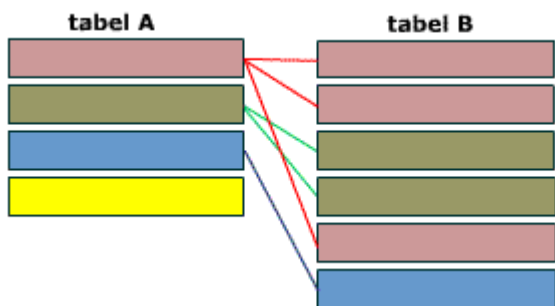
## Relaties

In een relationele database hebben we tabellen die een relatie met elkaar hebben. Bij het ontwerpen is het zaak dat je de relaties tussen de tabellen benoemt. Met benoemen wordt bedoeld dat je aangeeft wat voor soort relatie er tussen twee tabellen bestaat. In principe onderscheiden we de volgende soorten:

- De **één op veel** relatie
- De **veel op veel** relatie
- De **één op één** relatie

- **De één op veel relatie**

De één op veel relatie hebben we al gebruikt. Een rij gegevens (een record) uit de NAW-tabel heeft meerdere verwijzingen naar de tabel met telefoongegevens. Hetzelfde geldt voor de relatie tussen NAW-gegevens en tabel met e-mailadressen. Je kunt het onderstaande plaatje gebruiken om het te illustreren:



De koppeling verloopt eigen via de primaire sleutel van tabel A naar de primaire sleutel van tabel B. In deze relatie wordt de sleutel van B de vreemde sleutel genoemd. Of in het Engels de foreign key

- **De veel op veel relatie**

Stel je hebt de volgende gegevens van een school

lnr	naam	klas
1292	Henk van Gelder	V5a
1545	Milou Santinck	H4q
1189	Timo te Zetten	H4q
1052	Jessica van Slooten	V5a

docentnr	naam	vak
D34	Ruijter	wiskunde
D18	Jansschen	informatica
D35	Plasterk	biologie
D37	Jol	gym

Van de docenten is bekend dat: Ruijter wiskunde geeft in v5a en in h4q, Jansschen informatica in v5a en Jol gym in h4q en v5a

Hoe verwerk je dat nu in een database? Er komen in ieder geval twee tabellen: één tabel voor de leerling gegevens en één tabel met docentgegevens. Nu komt het probleem van het vastleggen van de docenten met de klassen. Je kunt dat niet in één van de bestaande tabellen verwerken.

docentnr	naam	vak	klas
D34	Ruijter	wiskunde	V5a
D34	Ruijter	wiskunde	H4q
D18	Jansschen	informatica	V5a
D35	Plasterk	biologie	
D37	Jol	gym	H4q
D37	Jol	gym	V5a

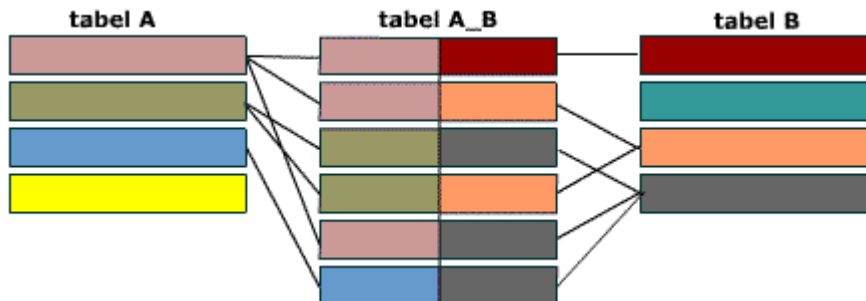
Dit is niet gewenst! We moeten nu ook iedere keer opschrijven dat Ruijter wiskunde geeft. Je moet dan maar één keer vastleggen. Het betekent dus een nieuwe tabel:

docentnr	klas
D34	H4q
D34	V5a
D37	H4q
D37	V5a
D18	V5a

We noemen dit een koppeltabel. Dit komt doordat een klas meerdere docenten kan hebben en een docent meerdere klassen.

Dit is een veel op veel relatie. Dit is een relatie waarbij meerdere rijen uit tabel A gekoppeld kunnen zijn aan meerdere rijen uit tabel B. De veel op veel relatie modelleren we met drie tabellen. Twee zogenaamde brontabellen en één koppeltabel.

We kunnen dat net weer als bij het vorige relatietype, illustreren met een plaatje:



Tabel A en tabel B zijn beide tabellen met een primaire sleutel. Tabel A\_B verbindt records uit tabel A en B en heet een 'koppeltabel'. Een koppeltabel bestaat uit maar twee kolommen die allebei een vreemde sleutel bevatten. De vreemde sleutel in de linkerkolom verwijst naar de primaire sleutel uit tabel A en die in de rechterkolom naar de primaire sleutel uit tabel B. De primaire sleutel in koppeltabel A\_B kun je op twee manieren maken. Je kunt natuurlijk weer kiezen om een extra kolom aan te maken, bijvoorbeeld `les_id`. Bedenk echter ook dat in elke rij de combinatie `docentnr` en `klas` maar één voorkomt, en dus uniek is. De primaire sleutel van de koppeltabel kan dus ook gemaakt worden uit de combinatie van de twee vreemde sleutels.

Het is op dit moment nog niet goed aan te geven welke primaire sleutel de beste is.

### De één op één relatie

Een relatietype dat je in de praktijk niet zo vaak zult zien, is de één op één relatie. Sommige deskundigen zeggen, dat als je dit relatietype gebruikt, je databaseontwerp eigen niet goed is. We gaan dit in deze cursus verder niet bekijken.

## Database normaliseren

We beginnen dit hoofdstukje eerst met het bekijken van een stukje theorie over het ontwerpen van een relationele database. De regels voor goed relationeel database-ontwerp zijn samengevat in 5 'normaalvormen', waarbij de eerste normaalvorm de laagste en de vijfde de hoogste (meest genormaliseerd) is. Deze normaalvormen zijn *richtlijnen* voor het juist ontwerpen van een relationele database.

We bekijken in deze cursus de eerste drie normaalvormen.

Normaliseren gebruiken we om er voor te zorgen dat we een zekere garantie krijgen op een goede database. Bedenk echter dat het geen absolute garantie op succes is!

Wat is het resultaat bij goede normalisatie?

- In een genormaliseerde database worden gegevens maar op 1 plek opgeslagen. Als je data wil invoeren, aanpassen of verwijderen hoeft je dat dus maar op 1 plek te doen.
- Integriteit. In een genormaliseerde database kun je gegevens zeer betrouwbaar opslaan.
- Flexibiliteit. De genormaliseerde structuur van de database zorgt ervoor dat gegevens op veel verschillende manieren opgevraagd en bijgewerkt kunnen worden.

Redundantie is de horror voor elke databaseontwerper. Normalisering probeert er dus voor te zorgen dat je geen redundantie hebt.

Het normaliseren en dus het ontwerpen van een database, komt er eigenlijk op neer dat de volgende zaken gedaan zijn:

- Het verdelen van gegevens in logische samenhangende groepen.
- Het minimaliseren van de hoeveelheid data die dubbel opgeslagen is, ofwel het voorkomen van redundantie.
- De gegevens zo organiseren dat het aanpassen of verwijderen van een gegeven altijd maar op één plek hoeft te gebeuren. Dit wordt de referentiële integriteit genoemd
- Gegevens zo organiseren dat ze snel en efficiënt op te vragen zijn.

Met een beetje oefening en puzzelen is het toch wel mogelijk om een niet al te uitgebreide database te maken. Voor het ontwerpen van grotere databases is echt wel een methodische manier van werken nodig.



## ▪ De eerste normaalvorm (1NF)

**De eerste normaalvorm** is een set basis ontwerpregels die op elke database van toepassing moeten zijn.

Een belangrijk startpunt is dat elke tabel een primaire sleutel heeft. Het makkelijkst is een zogenaamde externe sleutel; een extra kolom in je tabel met een nummer of een code of iets dergelijks. We hebben dit al toegepast op de database met de drie tabellen.

Een ander belangrijk aandachtspunt is dat je let op dat je in een kolom maar één waarde kunt invullen. Scheid daarom voorletters of voornaam van de achternaam. Tussenvoegsels kun je ook beter scheiden van de achternaam. Het maakt het zoeken en sorteren een stuk makkelijker. Het opslaan van slechts één waarde in een kolom wordt atomiciteit genoemd. Het ligt niet altijd eenduidig vast. Een mooi voorbeeld is de postcode. Hier is meestal verstandig om na te denken over het scheiden van de cijfers en de letters. Wat wil je met je database gaan doen? Wil je deze bijvoorbeeld gebruiken om bepaalde wijken te bereiken dan is een scheiding tussen letters en cijfers zelfs een vereiste. Het is daarom altijd van belang om na te denken waarom je bepaalde gegevens wilt vastleggen.

## ▪ De tweede normaalvorm (2NF)

**De tweede normaalvorm:** het verwijderen van dubbele of redundante gegevens

- De database voldoet aan alle regels van de eerste normaalvorm.
- Zo min mogelijk gegevens worden dubbel opgeslagen in de database.
- De velden die geen primaire sleutel zijn, zijn afhankelijk van de primaire sleutel.

Je moet bij de tweede normaalvorm je tabellen goed inspecteren op gegevens die dubbel opgeslagen worden in een kolom. Gegevens die dubbel opgeslagen worden, komen in aanmerking voor afsplitsing in een aparte tabel. Zie bijvoorbeeld nog eens de tabel bij van de docenten met de klassen. Indien je gegevens meerdere keren gaat vastleggen in één tabel dan is dat altijd reden om aan te nemen dat er een tabel bij moet komen.

▪ **De derde normaalvorm (3NF)**

*De derde normaalvorm: transitieve afhankelijkheden.*

- De database voldoet aan alle regels van de tweede normaalvorm.
- Van een tabel die voldoet aan de derde normaalvorm zijn alle kolommen die geen primaire sleutel zijn, uitsluitend afhankelijk van de primaire sleutel.

Zie de onderstaande tabel over het aantal gewerkte uren van werknemers bij een bedrijf. Deze tabel bevat af te leiden informatie. Het totaal aantal gewerkte uren kan met behulp van een berekening ook opgevraagd worden. De kolom totaal is niet uitsluitend afhankelijk van de primaire sleutel. Deze is mede afhankelijk van de overige kolommen en dus niet goed ontworpen.

werknr	ma	di	wo	do	vr	totaal
3456	7	9	9	7	9	41
5427	4	10	7	0	8	29

We kijken in het volgende hoofdstukje naar een eenvoudig praktijkvoorbeeld. We gaan de tot nog toe bekeken theorie daar toepassen.

## Biofriet

Biofriet is een keten van snackbars, oorspronkelijk begonnen in België. Inmiddels zijn er in Nederland ook filialen. Biofriet maakt uitsluitend gebruik van biologische grondstoffen. Het betekent bijvoorbeeld dat de frituurolie, de aardappels waar de friet van gemaakt wordt, het vlees etc. allemaal van biologische oorsprong zijn.

De filiaalhouders bestellen hun producten bij het centrale inkoopcentrum te Halle in België.

Hieronder wat informatie van Biofriet.

- producten (deel van het assortiment)

produkt	prijs
Frikadellen per stuk	0,95
Kroketten per stuk	0,85
Patat per kg	4,50
Slaatje per stuk	1,10
Gehaktbal per stuk	1,05

- filialen (een gedeeltelijk overzicht)

naam	adres	plaats	telnummer
Het Hoekje	Leidsestraat 25	Leiderdorp	070-2346782
De smulpaap	Bergenseweg 1	Schoorl	0226-315778
De biofriteur	Heerlenseweg 2	Heerlen	045-1888541

- bestellingen (een voorbeeld)

Het Hoekje bestelt op 12 december 50 frikadellen, 60 kroketten, 10 slaatjes en 20 gehaktballen.

Het is nu de bedoeling dat we bovenstaande informatie omzetten naar een relationele database zodat we de bestelinformatie goed kunnen beschrijven.

We lopen het database ontwerp door aan de hand van de eerste drie normaalvormen.

### De eerste normaal vorm

We gaan de tabellen voorzien van een primaire sleutel.

productnr	produknaam	eenheid	prijs
200	Frikadel	1	0,95
202	Kroket	1	0,85
101	Patat	kg	4,50
301	Slaatje	1	1,10
203	Gehaktbal	1	1,05

filiaalnr	naam	straat	huisnr	postcode	plaats	telnummer
1	Het Hoekje	Leidsestraat	25	2312 AM	Leiderdorp	070-2346782
6	De smulpaap	Bergenseweg	1	1871 CR	Schoorl	0226-315778
18	De biofriteur	Heerlenseweg	2	6411 NE	Heerlen	045-1888541

We maken ook een tabel voor een bestelling:

bestel_id	filiaalnr	datum
2011-01-12-06	6	12-01-2011

Probleem is nu dat we nog niet in beeld hebben wat er in een bestelling zit. Een bestelling kan namelijk meer dan één product bevatten, zie het voorbeeld.

We kunnen de producten van de bestelling niet in dezelfde tabel opnemen, je gaat dan informatie meerdere keren vastleggen:

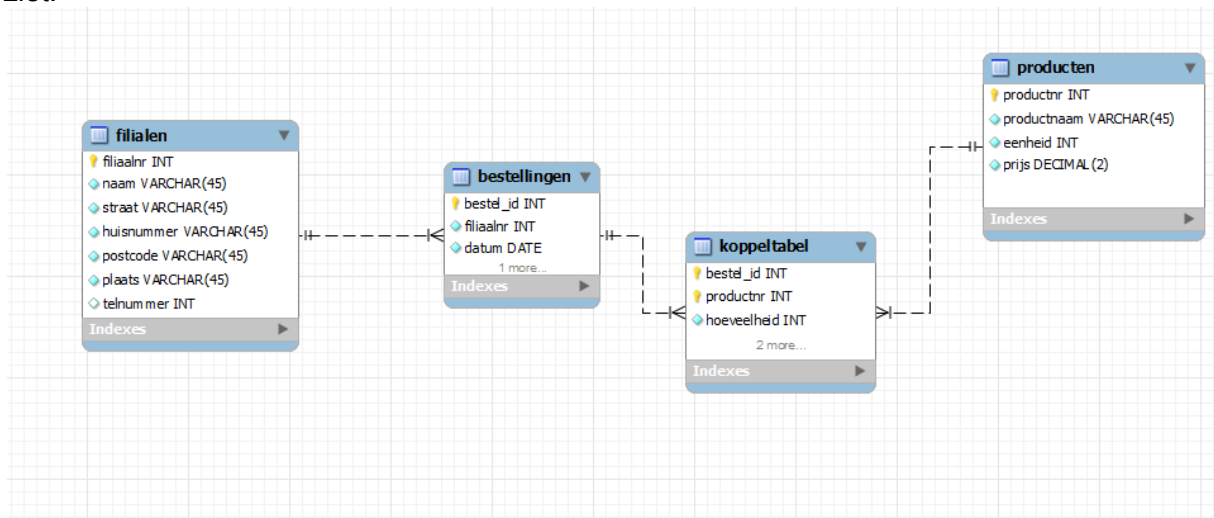
bestel_id	filiaalnr	datum	productnr	hoeveelheid
2011-01-12-06	6	12-01-2011	200	50
2011-01-12-06	6	12-01-2011	202	60
2011-01-12-06	6	12-01-2011	301	10

Dat is dus geen goede oplossing; strijdig met de tweede normaalvorm. Omdat een bestelling uit meerdere producten bestaat, betekent dit weer een nieuwe tabel.

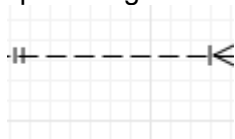
Tussen het filiaal en bestelling is er sprake van een één op veel relatie. Een filiaal zal immers vaker dan één keer bestellen. Dat betekent dat er tussen bestelling en product een veel op veel relatie moet zijn. Dit is zo omdat in een bestelling meerdere producten kunnen zitten en producten in verschillende bestellingen kunnen zitten. Dat houdt dus dat er een koppelingstabel moet komen (zie hoofdstukje over type relaties).

bestel_id	productnr	hoeveelheid
2011-01-12-06	200	50
2011-01-12-06	202	60
2011-01-12-06	301	10

We kunnen dat het volgende schema maken, voorlopig is alleen belangrijk dat je de relaties ziet:



opmerking:



is het teken voor één op veel in het schema.

Goed is te zien dat er eigenlijk tussen bestellingen en producten een veel op veel relatie zit. De koppeltabel is dan nodig.

# MySQL

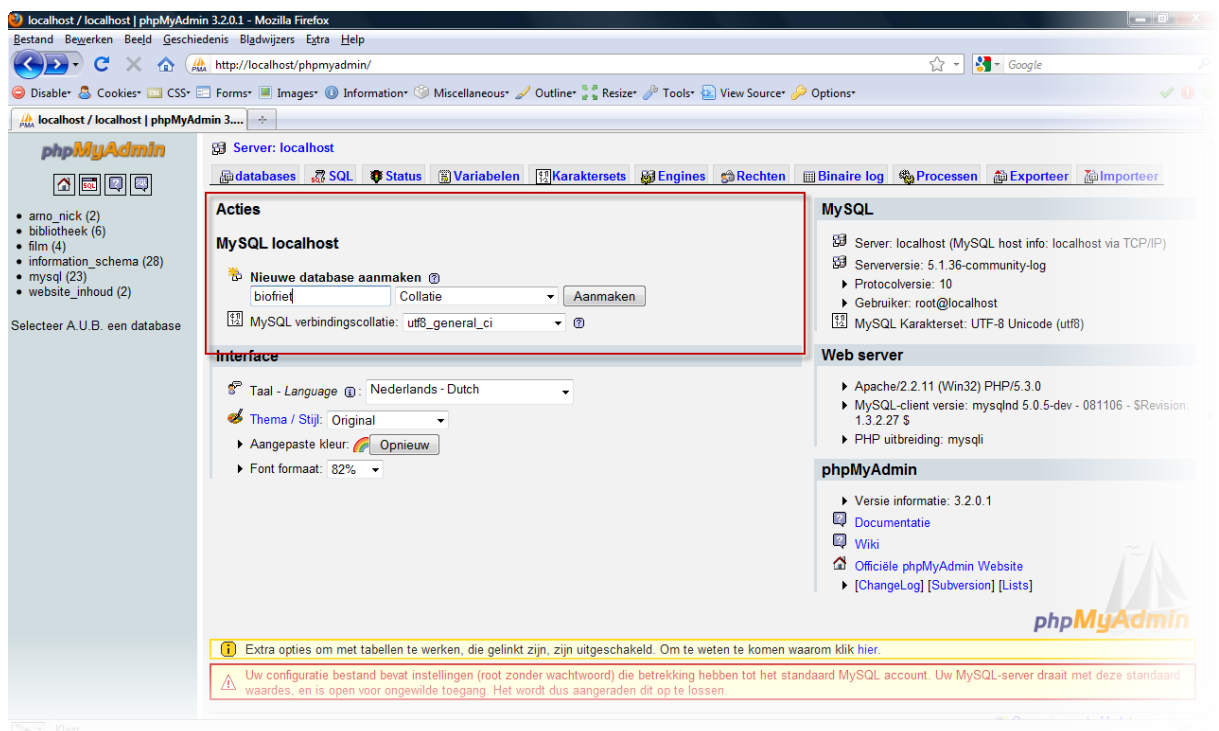
MySQL is een voorbeeld van een relationele database. Het wordt erg veel gebruikt in combinatie met websites die hun gegevens uit een gegevensbank ophalen. Je kunt hierbij denken aan webwinkels, Marktplaats, etc. Het grote voordeel van MySQL is dat het gebruik gratis is en dat je tijdens de ontwikkeling van je systeem dat gewoon op je eigen computer kunt doen. Je kunt MySQL alleen gebruiken maar dat is niet handig. Meestal maakt MySQL deel uit van een zogenaamde bundel. In zo'n bundel zit dan ook al een scripttaal zoals PHP en een programma om MySQL mee te "managen". In deze cursus maken we ook gebruik van zo'n bundel: Wamp (<http://www.wampserver.com/en/index.php>). Er zijn meer van dit soort bundels. Een andere veel gebruikte is XAMPP (<http://www.apachefriends.org/en/xampp.html>). In de bijlage vind je informatie over de installatie van Wamp.

## ▪ Het gebruik van MySQL

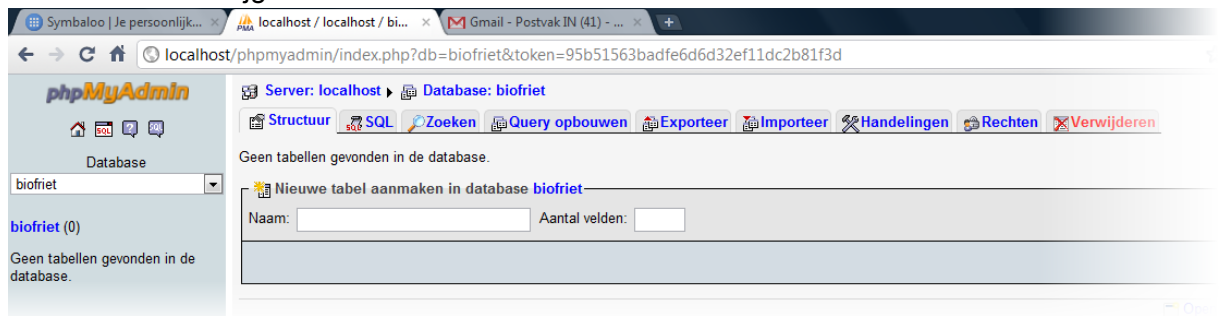
We gaan nu kijken hoe we het databaseontwerp van Biofriet in MySQL kunnen zetten. We lopen de stappen door. We gaan er vanuit dat je Wamp op je computer hebt geïnstalleerd.

1. We gaan in een browser eerst phpmyadmin, het programma waarmee we MySQL kunnen beheren, benaderen:

<http://localhost/phpmyadmin/index.php>

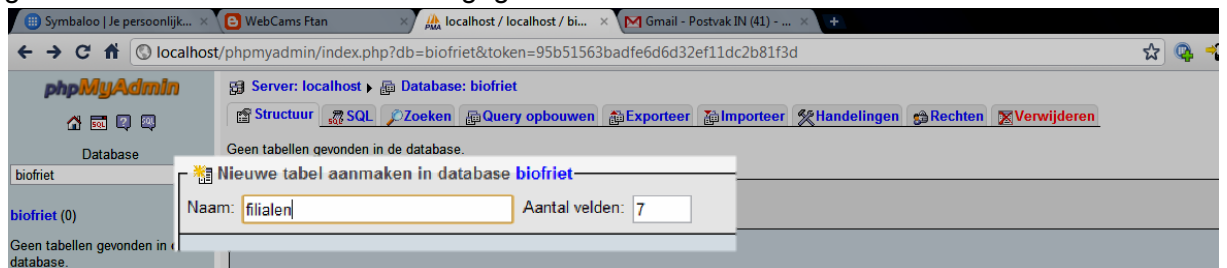


- We gaan de database aanmaken door in het midden van het scherm, bij acties, de naam voor de nieuwe database op te geven. We kiezen voor biofriet, gebruik geen hoofdletters. Je krijgt dan dit:



De database is aangemaakt maar heeft, uiteraard, nog geen tabellen.

- We gaan nu de tabellen aanmaken. We weten al dat het er drie moeten zijn. We beginnen met de tabel filialen. Vul de gegevens voor de eerste tabel in:



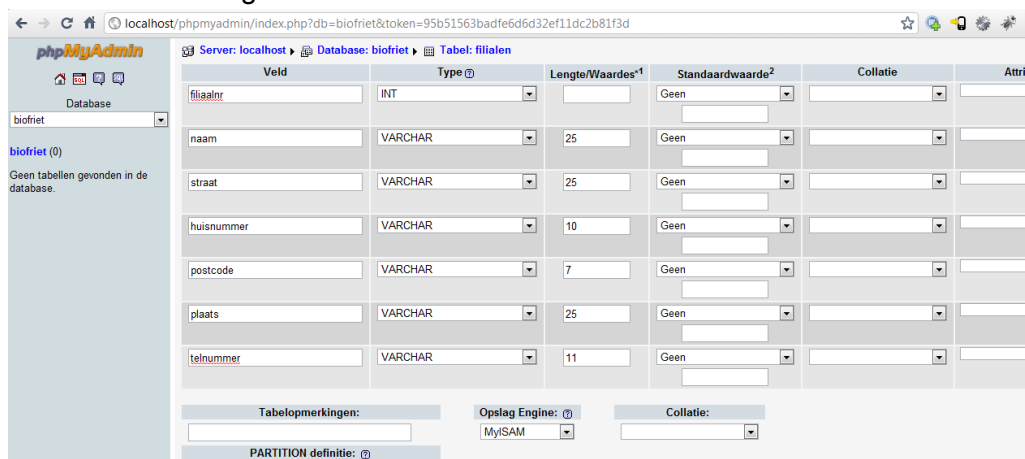
Er zijn 7 velden ofwel kolommen in de eerste tabel. Druk vervolgens rechtsonder op de knop Start.

- In deze stap gaan we de velden, de kolommen, benoemen. We geven het veld eerst een naam en we moeten ook aangeven wat voor soort waarde er in het veld wordt opgeslagen.

Her eerste veld is *filiaalnr*, dit wordt een geheel getal, dus heeft als type INT (een geheel getal)





De tweede veldnaam is *naam*, dit worden tekens. Gebruik hiervoor VARCHAR. Bij deze moet je ook nog aangeven hoeveel tekens er maximaal inzitten, de veldlengte. Meer dan 25 tekens gaan we niet gebruiken.

De rest kan ook ingevuld worden:



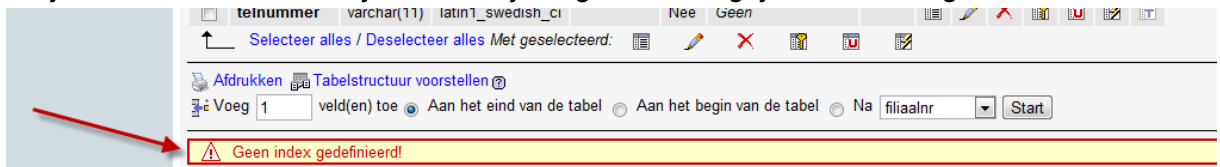
Klik op Opslaan om de gegevens vast te leggen.

5. Als het goed is, krijg je de volgende respons:

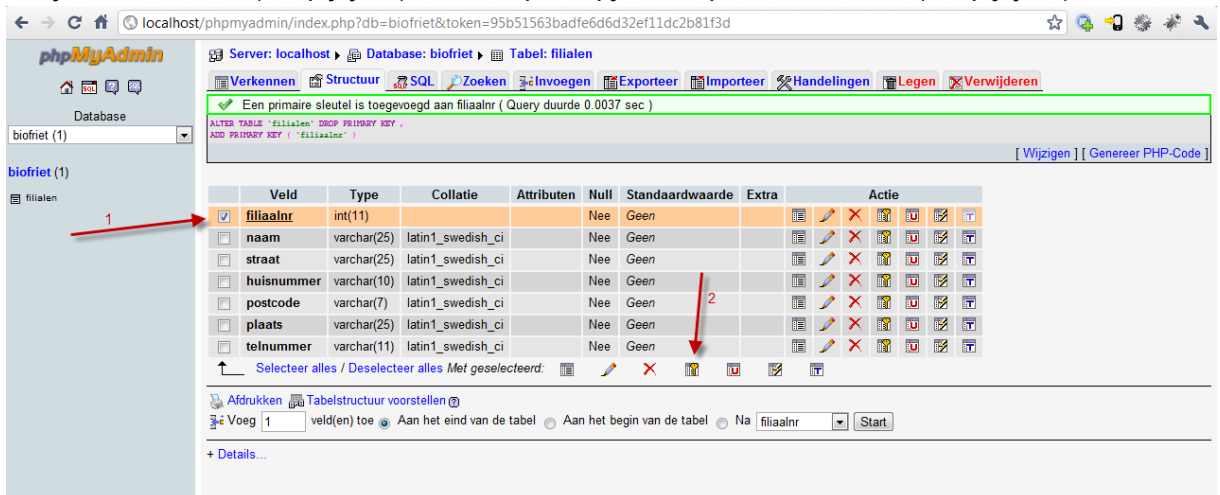
	Veld	Type	Collatie	Attributen	Null	Standaardwaarde	Extra	Actie
<input type="checkbox"/>	filiaalnr	int(11)			Nee	Geen		    
<input type="checkbox"/>	naam	varchar(25)	latin1_swedish_ci		Nee	Geen		    
<input type="checkbox"/>	straat	varchar(25)	latin1_swedish_ci		Nee	Geen		    
<input type="checkbox"/>	huisnummer	varchar(10)	latin1_swedish_ci		Nee	Geen		    
<input type="checkbox"/>	postcode	varchar(7)	latin1_swedish_ci		Nee	Geen		    
<input type="checkbox"/>	plaats	varchar(25)	latin1_swedish_ci		Nee	Geen		    
<input type="checkbox"/>	telnummer	varchar(11)	latin1_swedish_ci		Nee	Geen		    

We noemen nog twee dingen. In de kolom Null staat welke velden leeggelaten mogen worden. Geen enkele kolom mag Null, mag leeg, zijn.

6. Als je onderin het scherm kijkt, dan zie je nog een belangrijke mededeling:



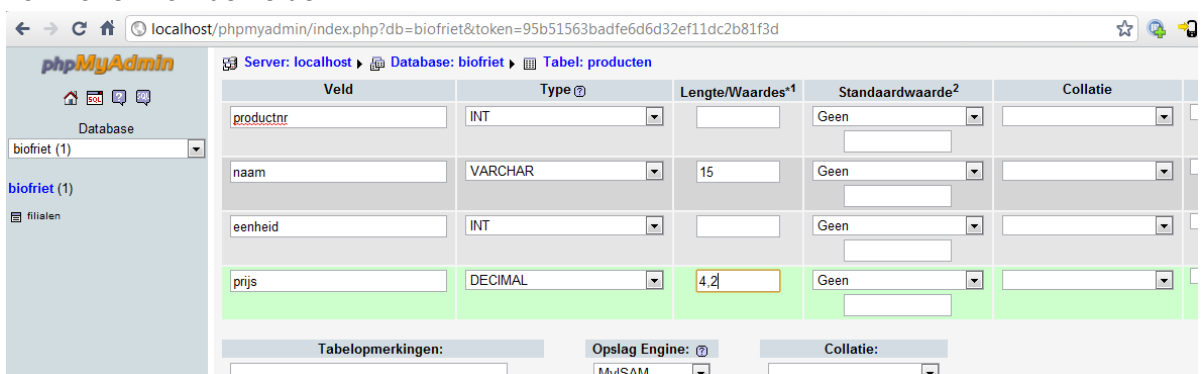
Er is geen index, maar ook geen primaire sleutel! Deze brengen we nu aan. Zet een vinkje voor *filiaalnr* (zie pijltje 1) en klik op knopje voor primaire sleutel (zie pijltje 2).



Er komt nu een streep onder *filiaalnr* te staan, het teken dat dit veld de primaire sleutel heeft.

7. We maken nu de tabel met de producten. Klik hiervoor eerst links in je scherm op biofriet, zo komen we uit de tabel filialen. We gaan nu de stappen 3 t/m 6 herhalen maar nu voor producten. We tonen nog even de belangrijkste schermen.

Aanmaken van de velden:





We geven de prijs Type DECIMAAL. De lengte 4,2 geeft aan dat er in het totaal 4 tekens zijn, met twee achter de “komma”. Het grootste getal dat we dus kunnen maken is 99.99. Let er op dat we de rekenmachine notatie gebruiken, de komma is dus een punt.

Na de velddefiniëring is het nodig om de primaire sleutel aan te brengen:



De tabel met de producten is nu ook klaar.

- We maken de tabel bestellingen. Klik -dus- eerst weer op biofriet(2) om een nieuwe tabel te maken.



- Tot slot komt de koppeltabel.



De primaire sleutel is nu de combinatie van *bestel\_id* en *productnr*. Je maakt deze zowel een vinkje bij de eerste als een vinkje bij de tweede te zetten en dan op het knopje primaire sleutel te klikken.

De database is klaar! We kunnen deze nu gebruiken. Uiteraard staan er nog geen gegevens in. Deze moeten we zelf nog toevoegen.

- **Het toevoegen van gegevens**

We beginnen met het toevoegen van wat producten.

1. Open hiervoor de tabel producten. Kies daarna het tabblad Invoegen:

The screenshot shows the phpMyAdmin interface for the 'biofriet' database, specifically the 'producten' table. The 'Invoegen' (Insert) tab is selected. The interface displays two empty rows for data entry, each with columns for 'productnr', 'naam', 'eenheid', and 'prijs'. The 'Negeer' (Ignore) checkbox is checked, and the 'Start' button is visible at the bottom of each row.

We vullen wat gegevens in:

The screenshot shows the phpMyAdmin interface for the 'biofriet' database, specifically the 'producten' table. The 'Invoegen' (Insert) tab is selected. The interface displays two rows of data entered. A red arrow points to the 'Negeer' (Ignore) checkbox, which is now unchecked. The 'Start' button is visible at the bottom of each row.

Veld	Type	Functie	Null	Waarde
productnr	int(11)			200
naam	varchar(15)			Frikadel
eenheid	int(11)			1
prijs	decimal(4,2)			0.95
Start				
<input type="checkbox"/> Negeer				
Veld	Type	Functie	Null	Waarde
productnr	int(11)			202
naam	varchar(15)			Kroket
eenheid	int(11)			1
prijs	decimal(4,2)			0.85
Start				

Voeg toe als nieuwe rij en dan Terug

1 Start Opnieuw

Herstart het invoegen met 2 rijen

**1** Gebruik de TAB-knop om van waarde naar waarde te navigeren of CTRL+pijltjes om vrijuit te navigeren

Door de checkbox met negeer uit te zetten, kunnen we meteen 2 rijen vullen in deze tabel.

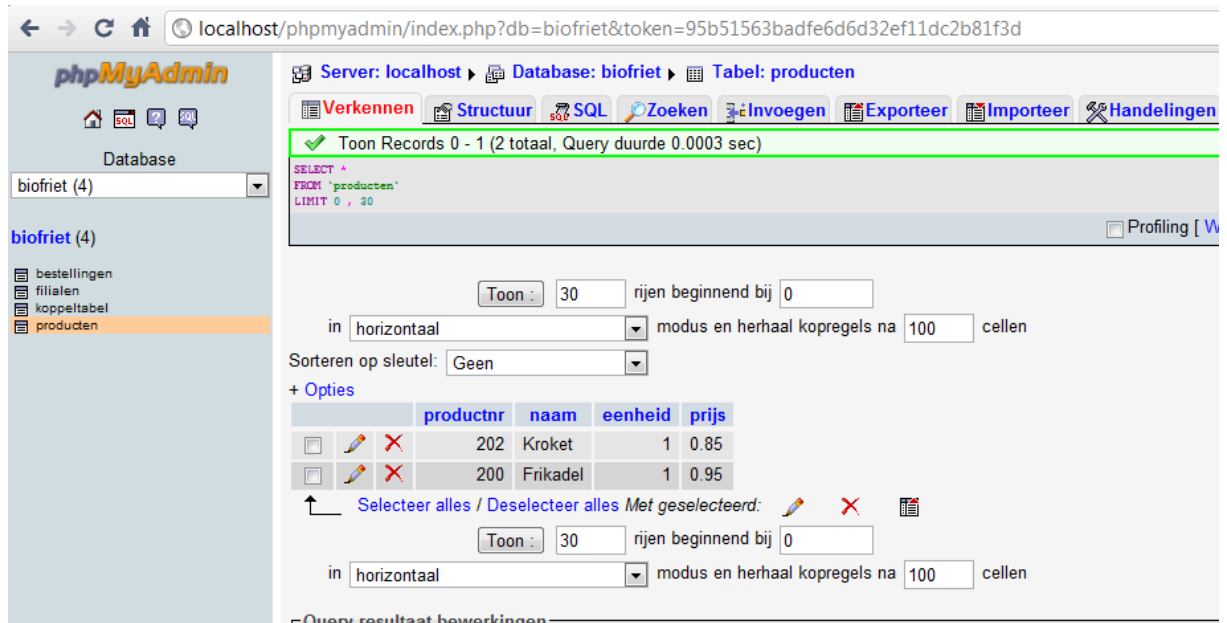
2. Druk daarna op de start button onderaan het scherm; beide invuldelen worden dan verwerkt.

MySQL meldt het resultaat:



The screenshot shows the phpMyAdmin interface for the 'biofriet' database. The 'producten' table is selected. A green message at the top indicates '2 rij(en) toegevoegd.' Below this, the SQL query is displayed: `INSERT INTO 'biofriet'.'producten' ('productnr', 'naam', 'eenheid', 'prijs') VALUES ('200', 'Frikadel', '1', '0.95'), ('202', 'Kroket', '1', '0.85');`. The 'Velden' (Fields) section on the right lists the columns: productnr, naam, eenheid, and prijs. At the bottom, there is a 'Start' button.

3. Door nu het tabblad Verkennen te kiezen kun je zien dat de gegevens er inderdaad in staan:



The screenshot shows the 'Verkennen' (Browse) view of the 'producten' table. A green message at the top indicates 'Toon Records 0 - 1 (2 totaal, Query duurde 0.0003 sec)'. The SQL query is `SELECT * FROM 'producten' LIMIT 0, 20`. Below the query, there are controls for displaying records (Toon: 30, rijen beginnend bij 0) and sorting (Sorteren op sleutel: Geen). The table data is shown below:

	productnr	naam	eenheid	prijs
<input type="checkbox"/>	202	Kroket	1	0.85
<input type="checkbox"/>	200	Frikadel	1	0.95

Below the table, there are controls for displaying records (Toon: 30, rijen beginnend bij 0) and sorting (Sorteren op sleutel: Geen). At the bottom, there is a 'Query resultaat bewerkinen' button.

4. Hetzelfde kunnen we doen voor tabellen bestellingen, filialen en de koppeltabel.

Uiteraard is het vullen van je database op deze manier niet erg gebruikersvriendelijk. In de praktijk zullen we dan ook niet vaak gebruik maken van deze manier van werken. Het was slechts bedoeld om te laten zien hoe je een database kunt vullen.

## Uitgebreidere problemen

In het voorgaande hebben we de theorie en praktijk bekeken om tot een database te komen. We gaan nu een aantal, wat uitgebreidere, problemen bekijken. In de onderstaande opdrachten wordt gevraagd om aan de hand van de beschrijving tot een relationele database te komen. Hierbij wordt gebruik gemaakt van de 3 normaalvormen.

1. We gebruiken een bibliotheek op een school. We lezen in het onderstaande verhaal van het hoofd van de bibliotheek hoe het een en ander in zijn werk gaat.

“In onze bibliotheek houden we van elk boek bij wie het geschreven heeft en in welk jaar het voor het eerst uitgegeven werd.. Voor leerlingen is het ook handig om te weten hoeveel pagina's het boek heeft. We geven aan bij welke taal het boek hoort en wat voor soort boek, bijvoorbeeld roman of novelle.

Van populaire boeken hebben we er meer dan 1. Zo kunnen meerdere leerlingen eenzelfde boek lenen.

Van de schrijvers houden we ook wat gegevens bij. Naast naamgegevens leggen we ook de geboortedatum vast en indien, mogelijk, het jaar van overlijden.

Uiteraard worden ook de uitleningen vastgelegd. Een leerling kan maximaal 5 boeken tegelijk lenen en de leentermijn is hoogstens drie weken.”

De vraag is nu hoe we dit tot een relationele database kunnen omzetten.

Ga voor je zelf na hoe dit gedaan kan worden.

2. Een nieuw verhuursorganisatie verhuurt in Georgië vlak aan de Zwarte Zee, op verschillende plekken vakantiebungalows. Er wordt nu alleen nog onderscheid gemaakt in een hoogseizoen en een laagseizoen. Zie onderstaande tabel voor een voorbeeld van een deel van de prijzen, per week:

plaats	type	hoogseizoen	laagseizoen
Sukhumi	2-persoons	75	50
Sukhumi	4-persoons	85	65
Sukhumi	6-persoons	95	80
Sukhumi	8-persoons	110	90
Ochamchiri	4-persoons	70	45
Ochamchiri	6-persoons	85	55
Gali	2-persoons	55	35
Gali	6-persoon	90	55

Maak een database waarin de verhuur per week kan worden bijgehouden. Er bestaan dus maar twee seizoenen. Uiteraard worden ook de gegevens van de verhuurders, zoals naam, adres, woonplaats etc. vastgelegd en of deze verhuurders wel of niet betaald hebben.

3. Een film liefhebber wil wat gegevens van films bijhouden. Hij wil de naam van de regisseur van de film vastleggen maar ook welke acteurs er in de film spelen. Hij heeft al een begin gemaakt van een soort tabel:

TITEL	JAAR	GENRE	TIJDSDUUR
1941	1979	Satire	146
8 million ways to die	1986	Thriller	115
A bridge too far	1977	Oorlog	158
A Bronx Tale	1993	Drama	122
A Few good men	1992	Drama	137
A river runs through it	1992	Romantiek	122
A.I.: artificial intelligence	2001	Fantasie (avontuur)	0
Absence of Malice	1982	Drama	116

In deze tabel ontbreken nog de namen van de regisseurs en welke spelers er in de film meedoen. Uiteraard wil hij van de spelers ook wat gegevens vastleggen.

Maak van bovenstaande een relationele database.

4. Een keten van apotheken heeft een aantal filialen, die dezelfde medicijnen verstrekken. De medicijnen hebben een unieke medicijnnaam en afkorting en worden in verschillende vormen verstrekt. Bijvoorbeeld in tabletvorm of in een fles of als zalf. Van alle medicijnen die aan een klant zijn verstrekt wordt onder meer bijgehouden of er sprake is van een herhalingsrecept. De klanten zijn vrij in de keus van het filiaal waar zij hun medicijnen halen.

Aan het eind van een dag wordt in elk filiaal door het hoofd een overzicht van verkopen opgemaakt en afgedrukt op het daarvoor bestemde afdrudpapier waar de titel **Overzicht medicijnverkoop apotheek De Gaper** op staat. Zie de schermafdruck voor een voorbeeld van zo'n overzicht.

Een hoofd kan van meer dan één filiaal het hoofd zijn.

Op het hoofdkantoor worden alle afgedrukte overzichten in het archief op volgorde van datum bewaard.

De eigenaar van de keten van apotheken wil voor het aanmaken en afdrucken van de overzichten een geautomatiseerd systeem ontwerpen met als basis een relationele database.

Overzicht medicijnverkoop apotheek De Gaper Levenbeekstraat 4 3 mei 2011

klantnr.	klantnaam	medicijnnaam	afkorting	type	soortmedicijn	herhalings recept	eenheidsprijs medicijn	aantal eenheden	totaal
1245	Jan Pietersen	Aspirine	ASP	tab	pijnstiller	ja	€ 2.50	2	€ 5.00
		Cortison	COR	zal	ontsteking	nee	€ 125.00	1	€ 125.00
		Bricanyl	BRI	inh	astma	nee	€ 98.00	2	€ 196.00
		Selsun	SEL	fles	antiroos	nee	€ 20.00	1	€ 20.00
		Prozac	PRO	tab	kalmering	ja	€ 30.00	1	€ 30.00
					<b>subtotaal</b>			<b>€ 376.00</b>	
2368	Jef Johnson	Aspirine	ASP	tab	pijnstiller	ja	€ 2.50	4	€ 10.00
		Hoestsiroop	HOS	fles	antihoest	ja	€ 15.00	3	€ 45.00
		Antibiotica	ABI	tab	antibiotica	nee	€ 56.00	1	€ 56.00
		Antihistamine	AHI	tab	allergie	nee	€ 34.50	1	€ 34.50
					<b>subtotaal</b>			<b>€ 145.50</b>	
.....	.....	.....	.....	.....	.....	.....	.....	.....	
.....	.....	.....	.....	.....	.....	.....	.....	.....	
					<b>dagtotaal</b>			<b>€ 521.50</b>	

We hebben nu wat oefening opgedaan met het maken van databases. We hebben nog niet gekeken naar het beheer van de databases. We hebben wel gezien hoe je wat beheer met behulp van phpMyAdmin kunt doen. Het is echter niet heel gebruikersvriendelijk. Het is vooral geschikt voor beheer maar niet voor grootschalig gebruik.

We concentreren ons in deze cursus op gebruik van databases via het web. We gaan dan ook gebruik maken van PHP als scriptaal om te communiceren met de database.

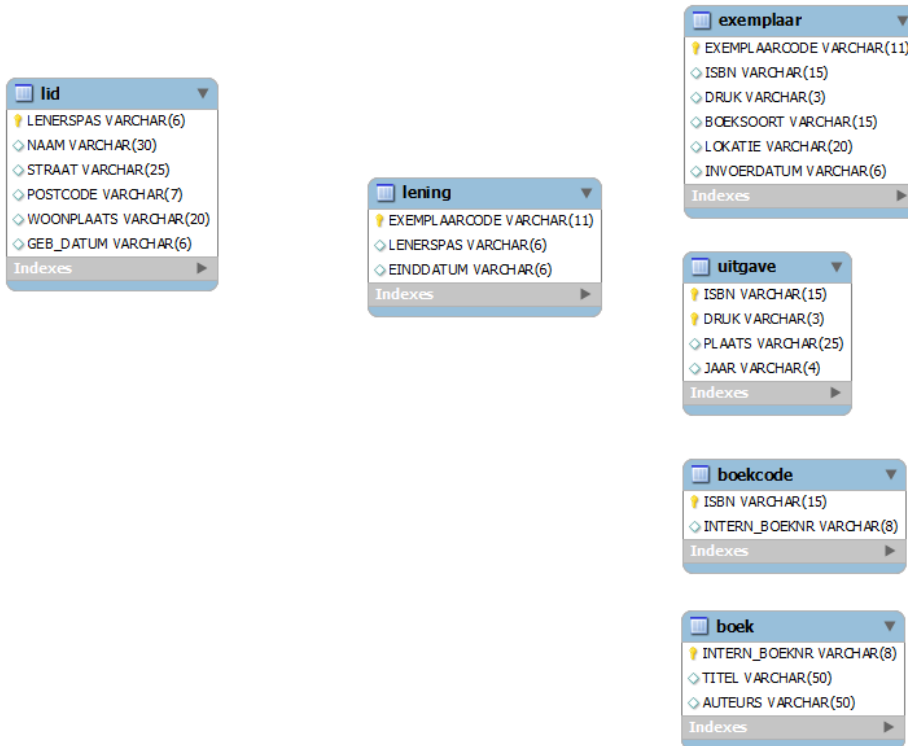
We bouwen de PHP-pagina's met behulp van Dreamweaver. In Dreamweaver zit een goede ondersteuning om snel PHP-pagina's te maken.

We beginnen daarom met een korte introductie van Dreamweaver en PHP.

# Een eerste php-pagina

We gaan nu kijken hoe je php kunt gebruiken in webpagina's om te communiceren met de database van MySQL. We gebruiken in deze paragraaf een al bestaande database van een bibliotheek.

Je ziet hier onder de tabellen die in deze database zitten:



We concentreren ons eerst op de tabel lid. In deze tabel worden de gegevens van de leden bijgehouden. We gaan nu een aantal php-pagina's maken om leden toe te voegen, te verwijderen en gegevens van leden te wijzigen.

We hebben natuurlijk daarvoor de database bibliotheek nodig. Deze gaan we in MySQL importeren. In de bijlage kun je lezen hoe dat moet.

Voordat we met Dreamweaver aan de slag gaan, is het goed om ook te kijken naar de instellingen voor Dreamweaver. Dit vind je ook terug in de bijlage.

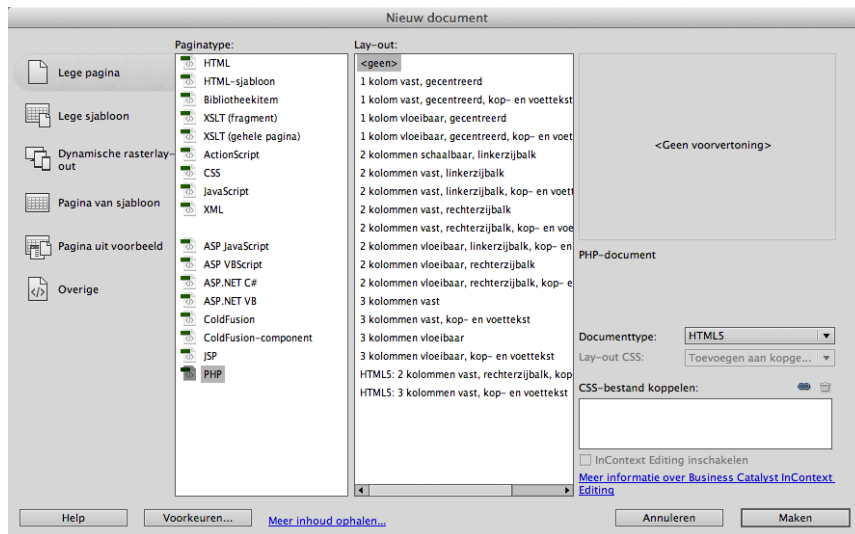
*Vanwege een kleine revisie van deze cursus in januari 2014, is een aantal schermafbeeldingen vernieuwd. Hiervoor is de Nederlandse versie van Dreamweaver CS 6 voor OS X gebruikt.*

*Deze versie is grotendeels gelijk aan de Engelse versie voor Windows.*

## We maken nu gebruik van Dreamweaver CS 6

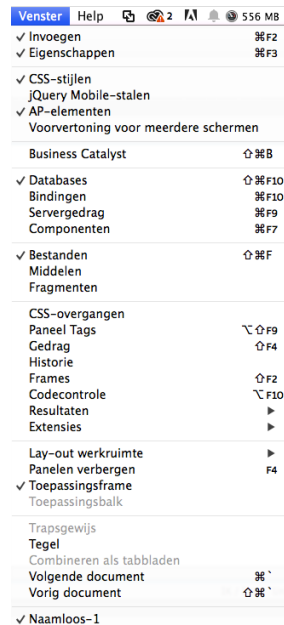
### 1. Kies File>New>Blank Page>PHP

Je kunt eventueel nog een style sheet koppelen.



### 2. Geef de pagina een naam. Noem de pagina leden.php. Deze pagina gaan we verderop gebruiken.

### 3. We moeten nu gaan kiezen met welke database we contact gaan maken. Controleer eerst of je het databasepaneel beschikbaar hebt:

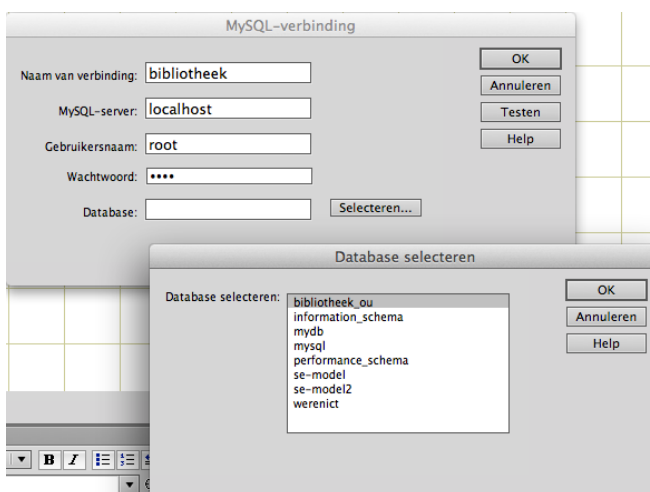




4. We klikken op het plusje om de database te kiezen. We kunnen nu de gegevens invullen. Controleer eerst of je in je panelen het onderstaande hebt:



Let er op dat je sitedefinitie goed staat! Dat zijn de vinkjes voor de stappen 1 t/m 3

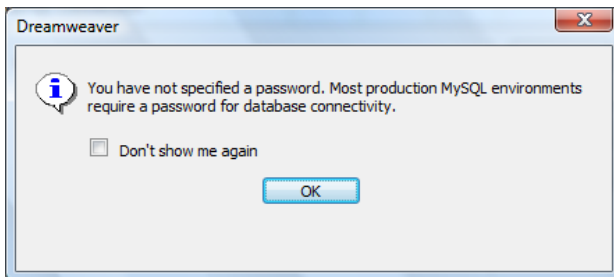


De naam van de connectie (verbinding) mag je vrij kiezen. Soms is het handig om meerdere verbindingen te hebben met dezelfde database. Dit is handig daar de verbinding gekoppeld kan worden aan de rechten die een gebruiker op de database heeft. We kiezen hier voor bibliotheek.

De MySQL server is ALTIJD localhost.

We hebben in de database nu maar 1 gebruiker, dat is de root. Niet verstandig om te gebruiken daar deze alle rechten heeft, dus ook om de database te vernietigen! Afhankelijk van de versie van je omgeving heb je wel of niet een wachtwoord op de gebruiker root. Het is uiteraard zeer verstandig om bij upload naar een openbare webserver deze instellingen te wijzigen!

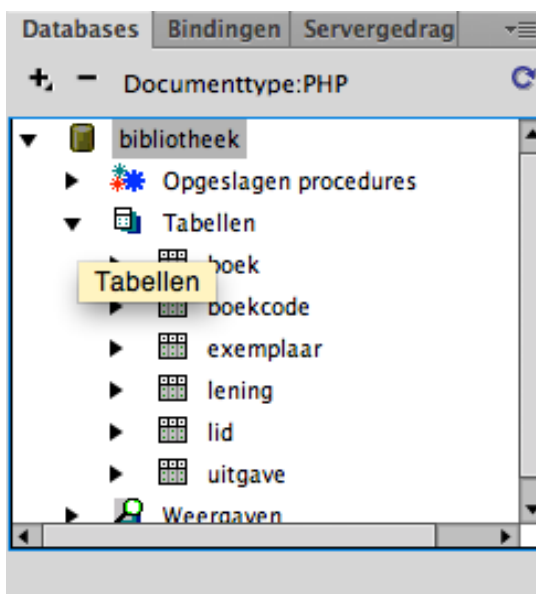
Indien je nu op select klikt, zie je alle databases in beeld die je kunt gebruiken. We kiezen voor de database bibliotheek. Indien je op OK klikt krijg je nog een melding: Inderdaad, een punt van aandacht!



Indien je op OK klikt, krijg je het vorige venster nog een keer. Bij het klikken op OK komen we nu wel terug in de webpagina.

Het beeld is gewijzigd! In het panel Databases is de verbinding bibliotheek te zien!

Indien je op het plusje klikt, kun je de tabelstructuur zien.



Dreamweaver heeft nu een map aangemaakt met de naam Connections. In deze map zit nu één bestand aangemaakt: bibliotheek.php. In dit bestand staan de verbindingsgegevens:

```
<?php
# FileName="Connection_php_mysql.htm"
# Type="MYSQL"
# HTTP="true"
$hostname_bibliotheek = "localhost";
$database_bibliotheek = "bibliotheek_ou";
$username_bibliotheek = "root";
$password_bibliotheek = "root";
$bibliotheek = mysql_pconnect($hostname_bibliotheek, $username_bibliotheek, $password_bibliotheek) or trigger_error(mysql_error(),E_USER_ERROR);
?>
```

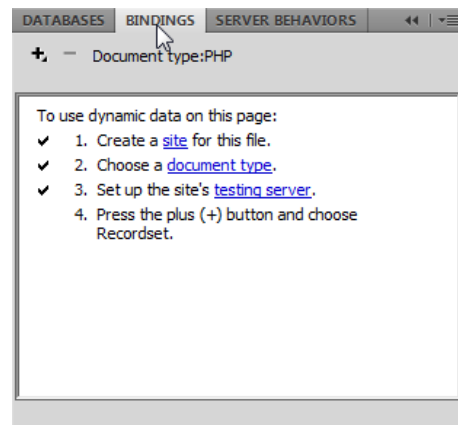
De php-pagina's moeten "weten" welke server er gebruikt wordt (opgeslagen in \$hostname\_bibliotheek), welke database (opgeslagen in \$database\_bibliotheek), welke gebruiker (opgeslagen \$username\_bibliotheek) en welk wachtwoord (opgeslagen in \$password\_bibliotheek). De verbinding, opgeslagen in \$ bibliotheek, geeft de verbindingsgegevens voor MySQL weer en geeft een foutmelding als de verbinding niet slaagt.

Dit bestand wordt ingesloten bij de php-pagina's die deze verbinding gebruiken. Bijvoorbeeld zo iets als:

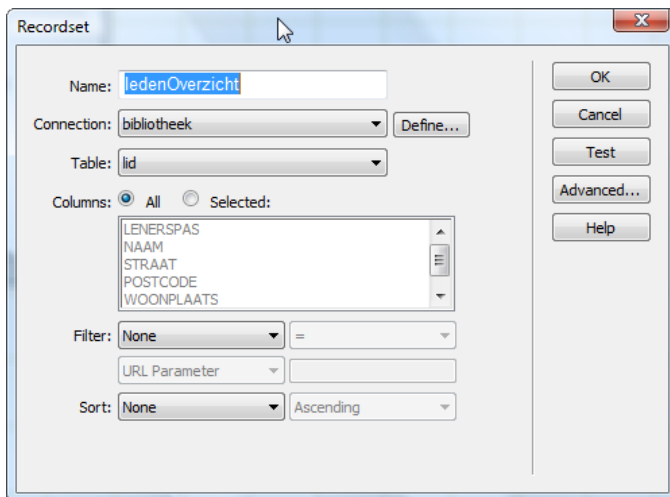
```
<?php require_once('Connections/bibliotheek.php'); ?>
```

Om een query (gegevensvraag) in de pagina te kunnen maken, werkt Dreamweaver met het begrip Recordset. Dit is dus feitelijk de query. Deze maken we nu eerst.

1. Klik in het panel Databases op tab bindings en vervolgens op het plusteken.

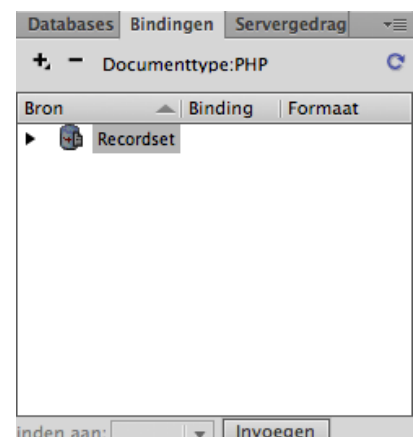


2. Geef de Recordset een passende naam. Bijvoorbeeld ledenOverzicht.



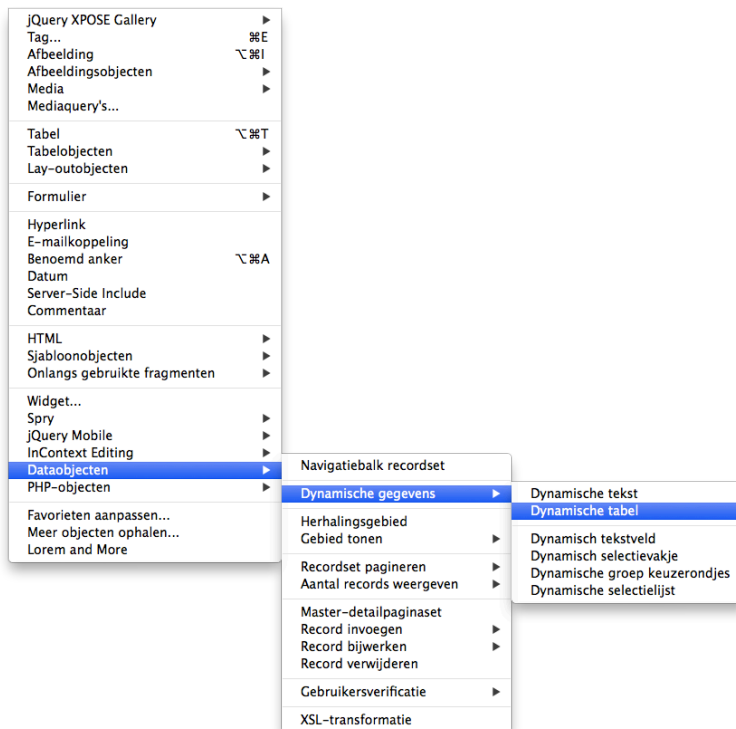
3. Kies de verbinding, bibliotheek. Direct komt bij Database Items de tabel in beeld, je kunt nu een keuze maken voor een tabel. We gaan een overzicht maken van de gegevens van de leden. We kiezen dus voor de tabel lid.  
Je kunt dan nog een keuze maken welke velden je wel of niet wilt zien. We kiezen nu voor All
4. De query is nu klaar. Door op Test te klikken, kun je zien of de query werkt en of je in beeld krijgt wat je in beeld wilde hebben.
5. In het tabblad bindings zie je nu je recordset terug:

Het ziet er goed uit dus we drukken op OK. Bij de Bindings (Bindingen) staat de Recordset nu in beeld.



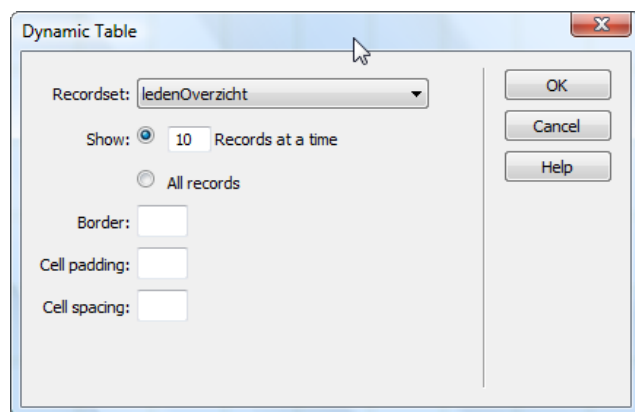
Nu moeten deze gegevens in beeld komen in een php-pagina. Feitelijk plaatsen we nu de query in een php-pagina en maken een tabel waarin de gegevens komen te staan. In Dreamweaver kun je daar een Dynamic table voor gebruiken. Deze wordt precies zo groot, als er gegevens vanuit de query, dus de Recordset, zijn.

1. Kies Insert>Data Objects>Dynamic Data>Dynamic Table.



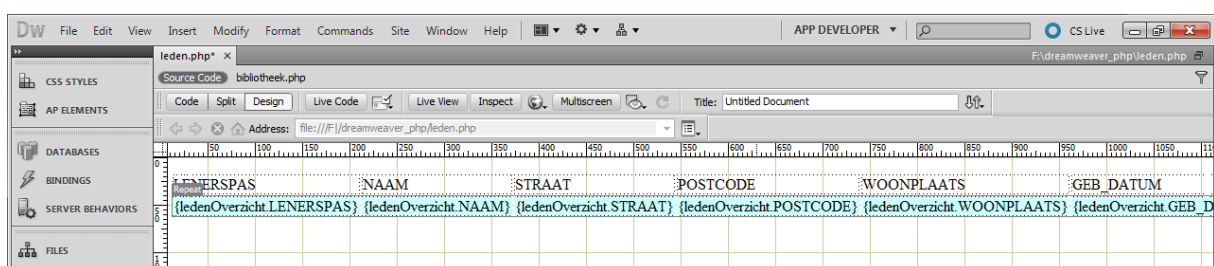
Je krijgt dan het volgende:

Er was maar één Recordset, dus deze wordt gekozen. Je kunt vervolgens wat keuzes maken. Indien je veel records op je scherm hebt, is het verstandig om te kiezen voor een beperkt aantal per keer. De rest moet natuurlijk ook in beeld komen. Verderop zullen we zien hoe je dat kunt oplossen.

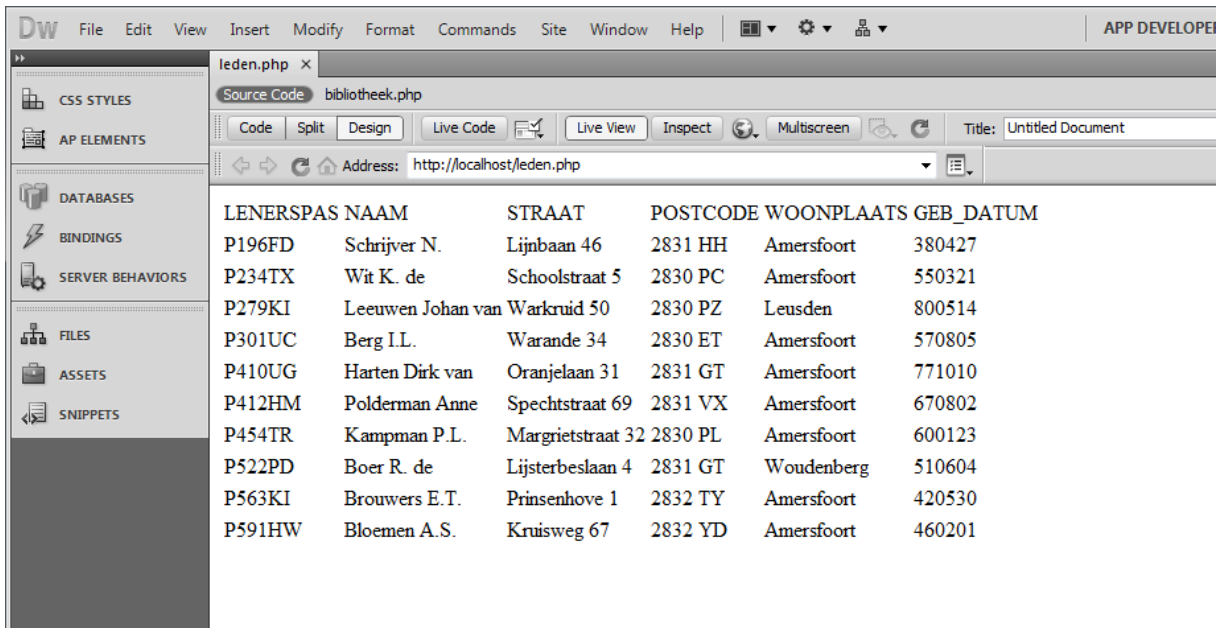


Klik op OK. Je hebt nu alles gedaan

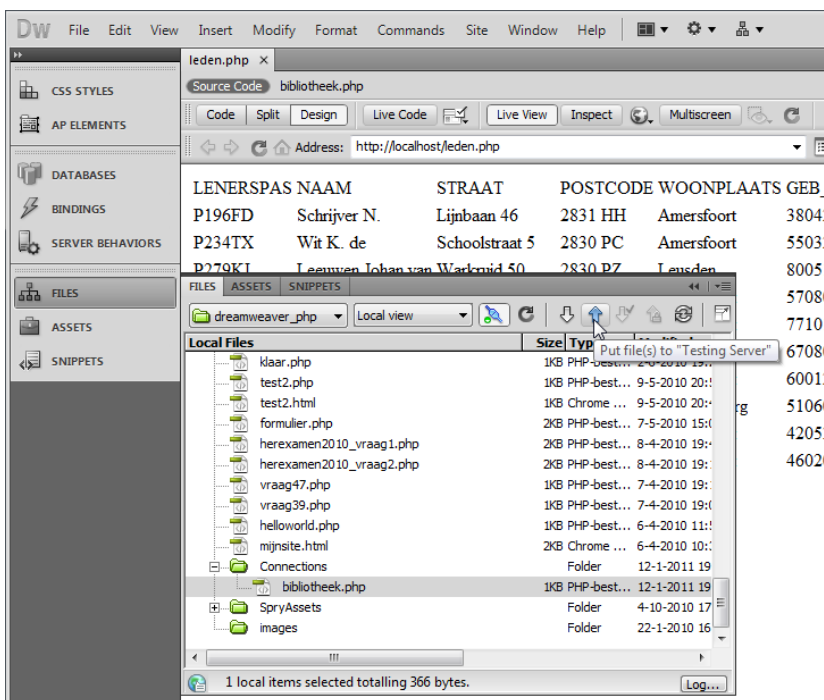
om de gegevens die met een query uit de database komen, in een php-pagina met een tabel te zetten.



2. Je kunt nu met de knop Live View zien hoe de pagina er uit komt te zien.



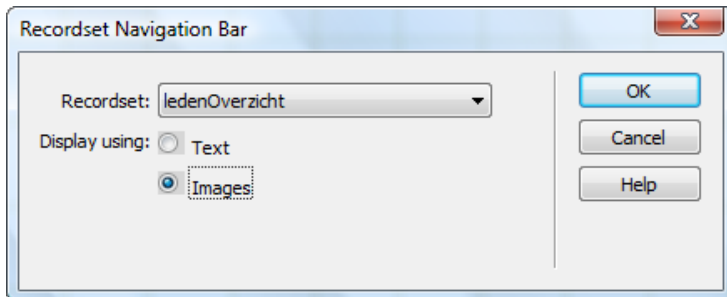
Indien je foutmeldingen krijgt, is meestal het probleem dat je verbindingsbestand niet op de webserver staat. Upload dan nog even handmatig het bestand:



Je kunt uiteraard ook een Preview in browser doen (F12). Ook daar moet je er op letten dat je het verbindingsbestand upload!

Uiteraard kun je de opmaak aanpassen met behulp van CSS. Indien je met Code view kijkt, dan zie je dat er veel code is aangemaakt. De eerste 30 regels zijn bedoeld om je pagina te beveiligen tegen zogenaamde SQL-injecties. Een systeem dat het mogelijk maakt om via je pagina allerlei ellende op je webserver en databaseserver te plaatsen. We gaan niet in op waarom de code er zo uitziet.

Indien je te maken hebt met veel records is een navigatie wel handig. Dat gaat heel snel. Kies In-  
sert>Data Objects>Recordset Paging>Recordset Navigation Bar.



Indien je voor Images kiest krijg je keurige navigatiepijltjes onder je tabel. Vergeet de vier gif-afbeeldingen niet te uploaden!

Je kunt de Recordset ook gewoon met SQL maken. In de bijlage wordt dat nog verder toegelicht

## Records toevoegen

Een record toevoegen gaat erg gemakkelijk. Met de Record Insertion Wizard wordt er een eenvoudig formulier gemaakt en worden de gegevens m.b.v. een query naar de database verstuurd. We gaan nu een pagina maken waarmee nieuwe leden aan de database toegevoegd kunnen worden.

1. Maak een nieuwe pagina aan: leden\_toevoegen.php.  
Kies Insert>Data Object>Insert Record>Record Insertion Wizard.

Record Insertion Form

Connection: bibliotheek

Table: lid

After inserting, go to:  Browse...

Form fields:

Column	Label	Display As	Submit As
LENERSPAS	LENERSPAS:	Text field	Text
NAAM	NAAM:	Text field	Text
STRAAT	STRAAT:	Text field	Text
POSTCODE	POSTCODE:	Text field	Text
WOONPLAATS	WOONPLAATS:	Text field	Text
GFR_DATIIM	GFR_DATIIM:	Text field	Text

Label: LENERSPAS:

Display as: Text field Submit as: Text

Default value:

OK  
Cancel  
Help

Kies bij Connection weer voor bibliotheek. Kies voor de tabel lid en je ziet dat dan ogenblikkelijk de tabelstructuur weer in beeld komt.

Indien je een database hebt waarbij een van de velden een auto\_increment bevat (een kolom die je automatisch laat ophogen), dan moet je deze natuurlijk niet door een bezoeker laten invullen. Je kunt dan dat veld in Form fields met de min-knop verwijderen.

Alles ziet er goed uit en we drukken daarom op OK. De pagina is klaar!

De opmaak is nog niet geweldig maar met wat HTML en CSS is dat gemakkelijk te veranderen. Het label van de button bijvoorbeeld, moet echt anders!

Een ander probleem is dat het formulier niet beveiligd is tegen onjuiste invoer. Dat kun je handmatig doen, met Javascript of PHP. We gaan daar nu niet op in.

Deze invoer is natuurlijk niet erg flexibel. Dreamweaver heeft veel mogelijkheden om formulieren te maken waardoor je meer greep krijgt op de wijze waarop gebruikers gegevens kunnen toevoegen.

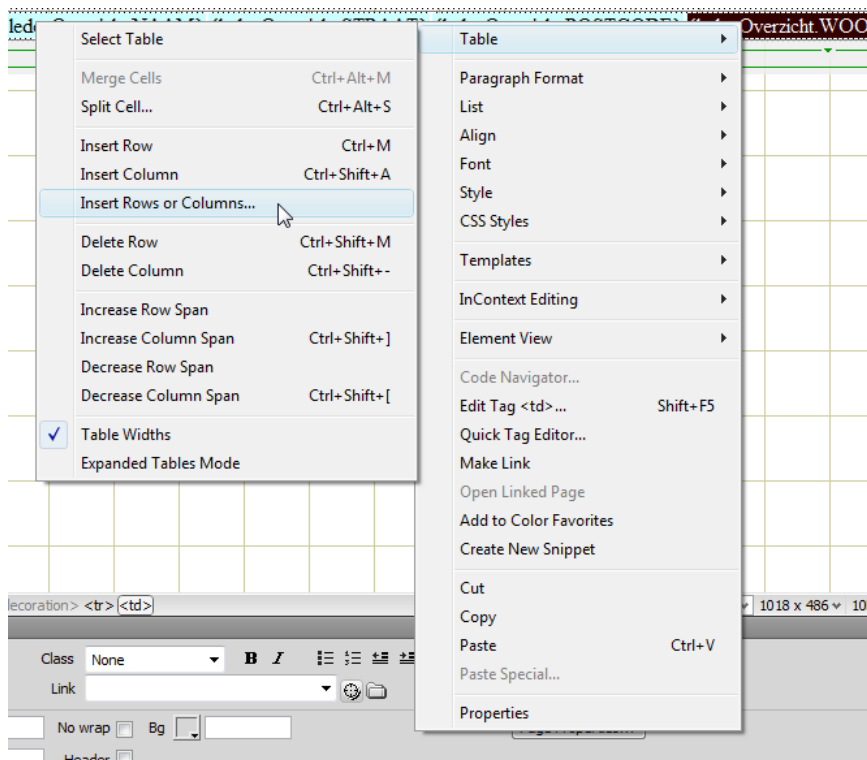


## Wijzigen van records

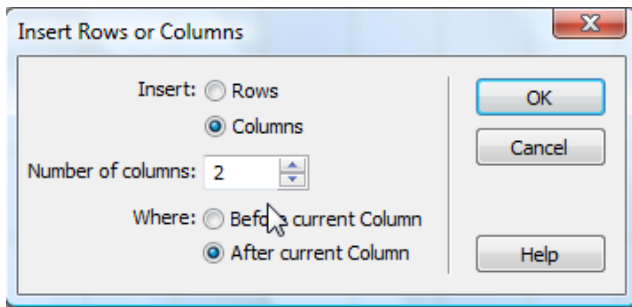
Het wijzigen (updaten) gaat iets gecompliceerder. Dit heeft te maken met het feit dat je de primaire sleutel nodig hebt. Met behulp van de primaire sleutel kunnen de records uniek vastgelegd worden en ook dus op basis van de primaire sleutel gewijzigd worden. Toch wil je eigenlijk niet dat de bezoekers van je webpagina daar iets mee te maken krijgen. We gaan dus iets maken waarbij op de achtergrond de primaire sleutel gebruikt wordt, zonder dat een bezoeker daar iets van merkt.

We gebruiken weer de tabel lid. Hierin is lenerspas de primaire sleutel. We willen niet dat leners hun pasgegevens kunnen wijzigen.

1. We beginnen met het maken van de nieuwe pagina: open de pagina leden.php en wijzig deze in leden\_wijzigen.php.  
Zo hebben we meteen weer de beschikking over de Recordset!
2. We zetten twee extra kolommen naast de al bestaande kolommen van de tabel. Ga in de laatste kolom staan en klik met de rechter muisknop. Kies Table>Insert Rows or Columns:

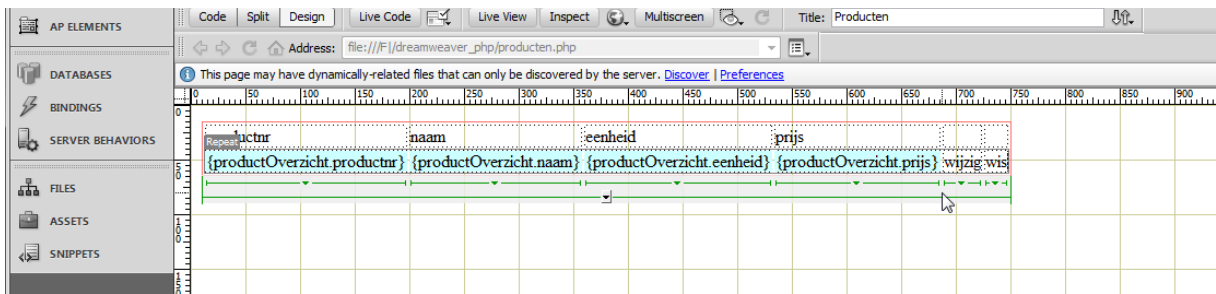


Vul vervolgens het volgende venster als volgt in:



We willen twee kolommen naast de al bestaande kolommen.

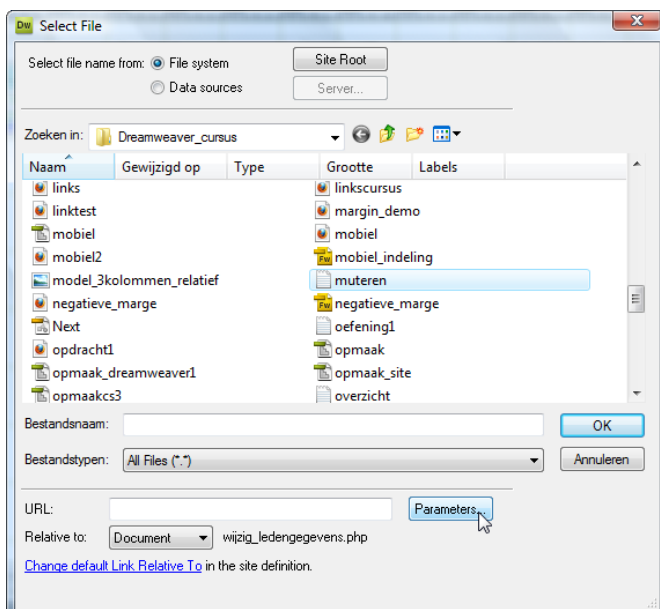
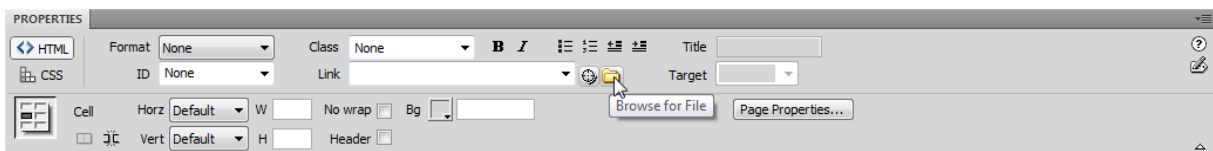
Tik vervolgens in de tweede rij van de eerste aangemaakte kolom wijzig en daarnaast wis. Je krijgt dan:



We hebben nog een pagina nodig waarin we de gegevens daadwerkelijk kunnen wijzigen. Maak een nieuwe pagina aan: muteren.php

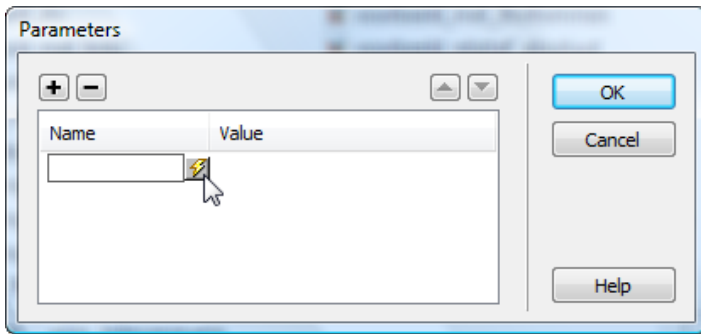
We gaan nu van `wijzig` een link maken. Deze moet er voor zorgen dat van het betreffende record de primaire sleutel, hier dus de lenerspas, wordt overgedragen naar de wijzig-pagina. We noemen dit parameteroverdracht. We gaan dat nu instellen.

Selecteer de tekst `wijzig`. Kies voor `Browse for File` in het Properties Panel:



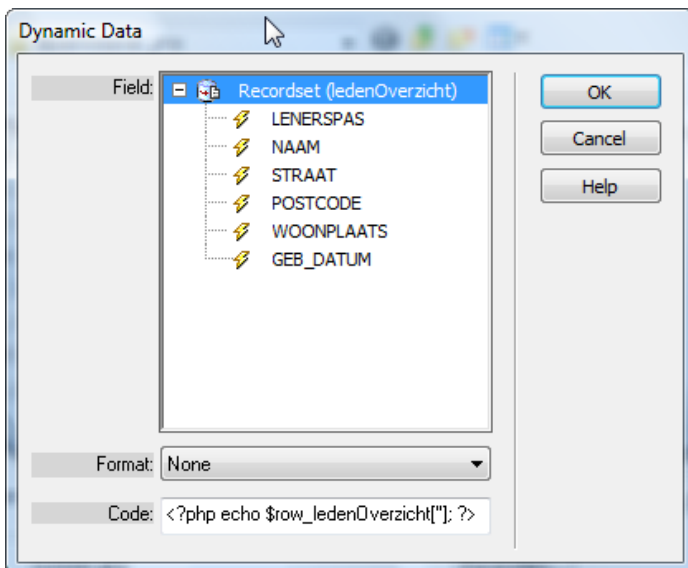
Selecteer vervolgens het juiste bestand en klik op `Parameter`. We kunnen nu kiezen welke parameter overgedragen moet worden.

Je krijgt het volgende scherm:



Bij de Name mag je in principe een eigen naam invoeren maar het is eigenlijk wel zo handig om gewoon de gegevens uit de Recordset te gebruiken. Klik op het bliksemteken. Je kies daar voor lenerspas.

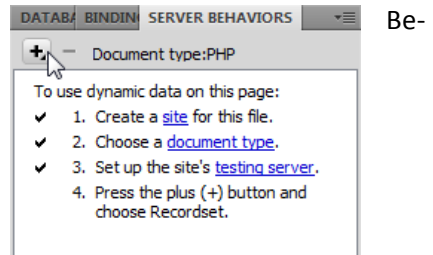
Nu komt de belangrijkste stap! Klik in Value, kies weer voor het bliksemteken en kies nu lenerspas. Dit is de waarde die overgedragen wordt! Het is erg belangrijk dat je nu kiest voor een waarde uit de Recordset. Hier mag je absoluut niet iets zelf typen!!



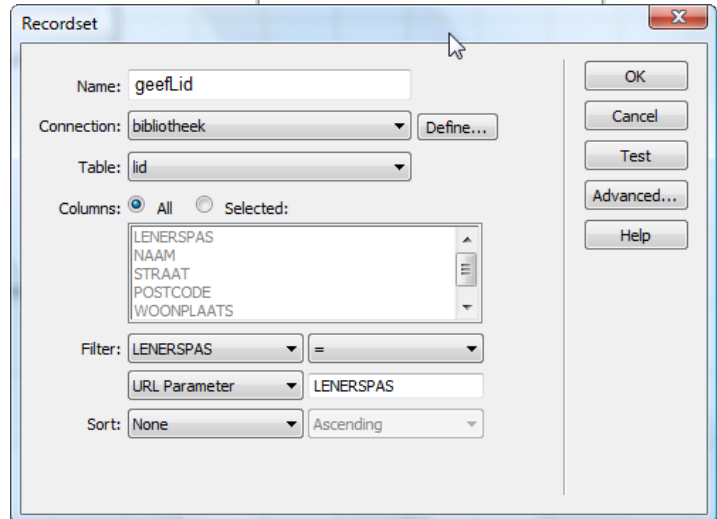
Klik drie keer op OK en de link is klaar.

De volgende stap is het maken van de wijzig-pagina. Deze pagina ontvangt de waarde van het lenerspas, waardoor de gegevens die bij deze lenerspas horen, gewijzigd kunnen worden.

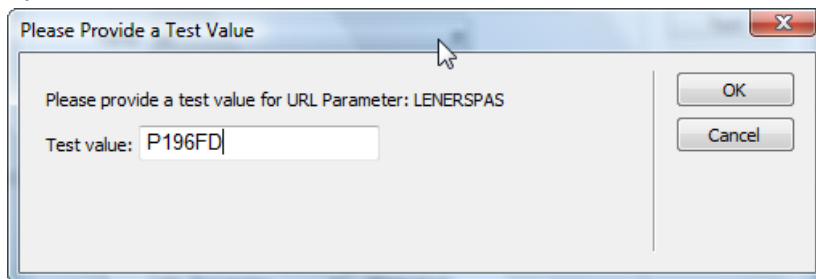
1. Open het bestand muteren.php. Klik op Server Behaviors om een nieuwe Recordset aan te maken:



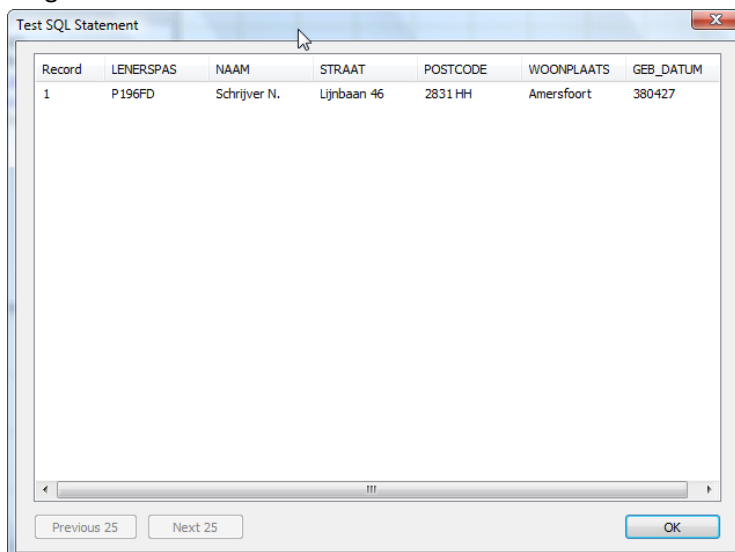
2. Maak de instellingen zoals op het scherm te zien zijn. Bij Filter kies je voor lenerspas, dit is het veld waarop geselecteerd wordt. Vandaar dat je kiest voor het = teken. Je bent ook verplicht om aan te geven hoe de overdracht gedaan is. We kiezen voor overdracht via een link (URL).



3. Je kunt nog kiezen voor Test, je moet dan wel een bestaande lenerspas invoeren, je "filtert" tenslotte op lenerspas. Bijvoorbeeld:



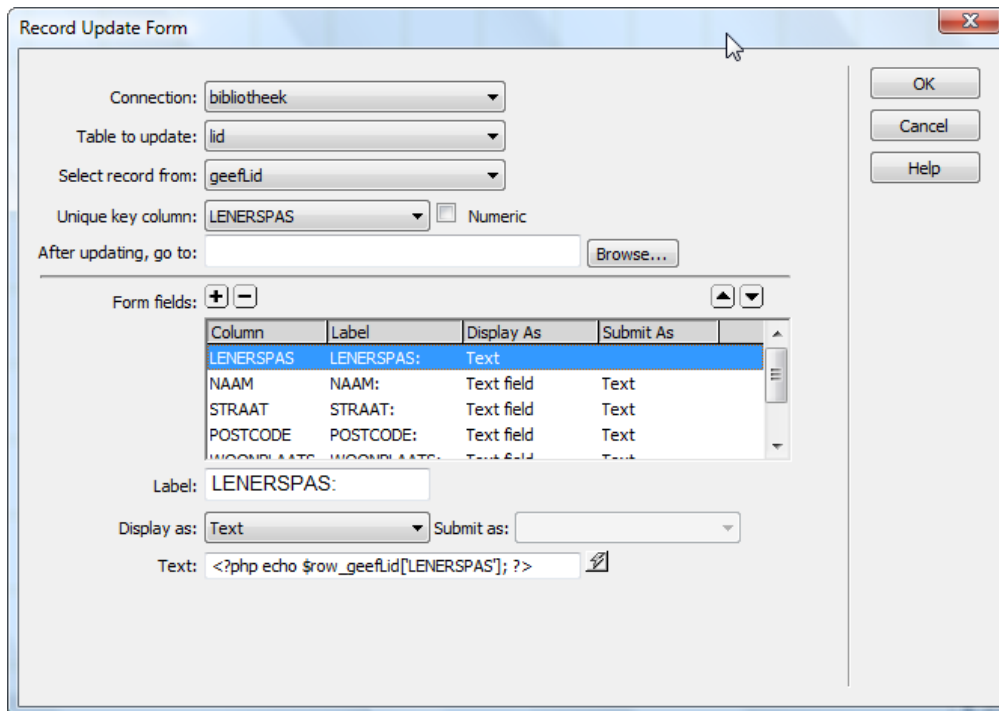
Dit geeft dan:



4. Druk nu twee keer op OK.

We kunnen nu het formulier maken om het wijzigen echt uit te voeren.

5. Zorg er voor dat de cursor nu ergens in de body staat. Het is anders niet mogelijk om het wijzig formulier te laten zien! Kijk dus even in Code view waar je cursor staat.  
We kunnen nu gebruik gaan maken van de update wizard van Dreamweaver. Kies In-  
sert>Data Objects>Update Record>Record Update Form Wizard, je krijgt dan:

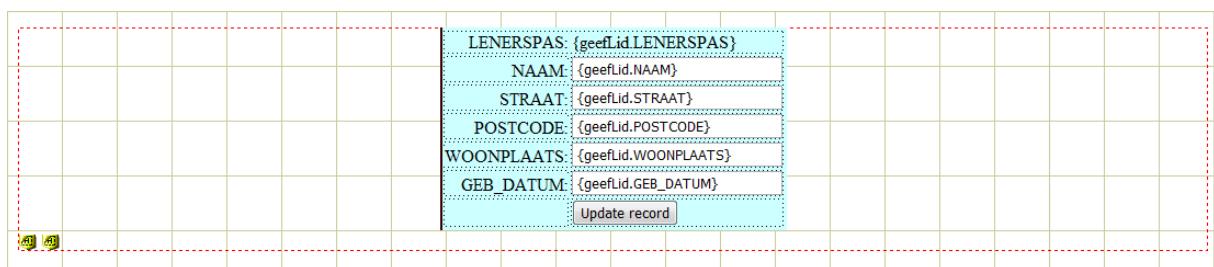


We maken nu gebruik van de Recordset geeflid die we hiervoor gemaakt hebben. Deze gegevens komen dus in de velden van de pagina muteren.php

Wat opvalt is dat Lenerspas niet vermeld wordt in de kolom Submit As. Dit is de primaire sleutel en daar wilden we vanaf blijven!

6. In het veld After updating, go to kun je de pagina invoeren die je wilt laten zien nadat de wijzigen zijn doorgevoerd. Het is verstandig om weer terug te gaan naar wijzig\_ledengegevens.php

De pagina moet nu dit bevatten:



De wijzig pagina is klaar!

## Het wissen van records

Het laatste dat we bekijken is het wissen van gegevens uit een tabel. We nemen weer als voorbeeld het verwijderen van een lid uit de tabel lid.

Om een ander probleem, bij het wissen, ook duidelijk te maken hebben we de tabel leden gewijzigd. We hebben een nieuwe primaire sleutel aangebracht: lid\_id en het betekent dus dat lenerspas geen primaire sleutel meer is.

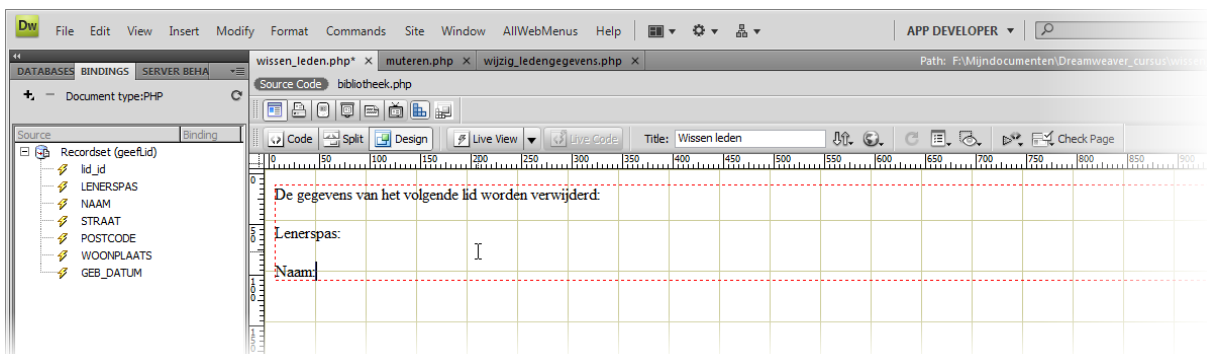
Importeer nu eerst de nieuwe database bibliotheek\_extra.sql

We lopen de noodzakelijke stappen door.

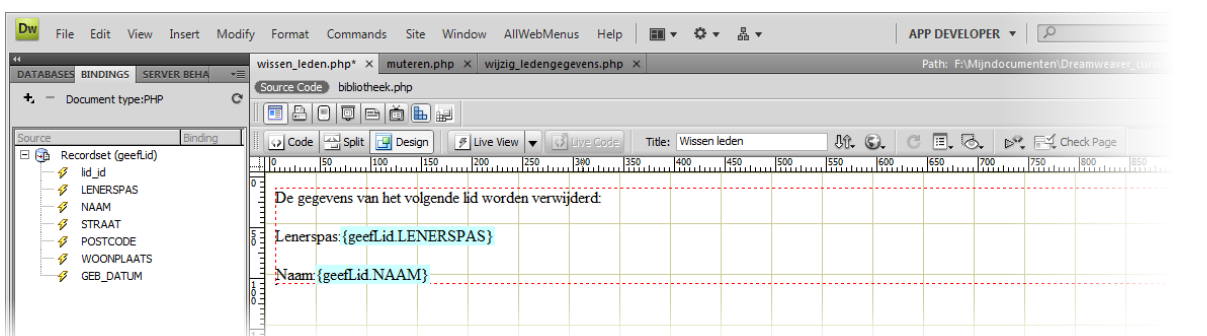
1. Maak eerst de pagina verwijder\_lid.php aan. We hebben nog een keer de Recordset GeefLid nodig. Deze kunnen we uit de pagina muteren.php halen. Open deze en kopieer de Recordset (selecteer deze en klik met rechter muisknop en kies Copy). Plak deze nu in het Panel van verwijder\_lid.php. Zo hebben we daar de Recordset ook tot onze beschikking.

Het is goed om te bedenken dat we op deze pagina komen door een link vanaf wijzig\_ledengegevens.php, deze link gaan we straks nog maken. Op de verwijderpagina moet dus in beeld komen wat we gaan verwijderen. Tevens moet er een knop komen die het verwijderen daadwerkelijk uitvoert. Dit gaan we nu maken.

2. Zet een formulier op het scherm, controleer of je cursor in de body staat! Insert>Form>Form. Je krijgt nu een rood kader op je scherm. Tik in het formulier de volgende tekst:



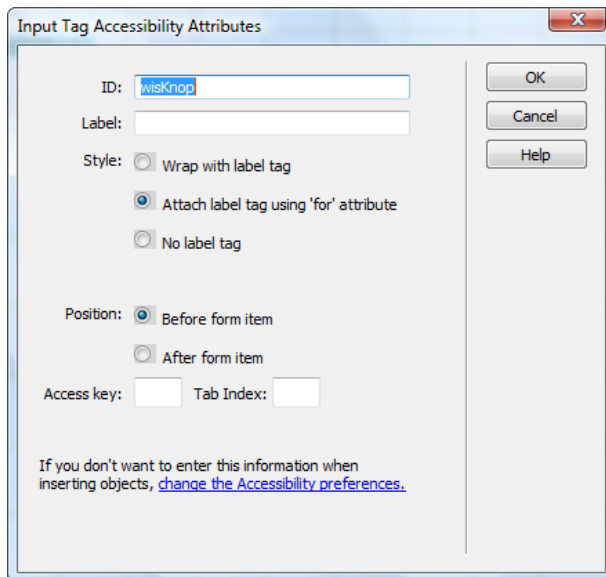
3. Zo ziet je wie er verwijderd wordt. Je hebt alleen nog geen gegevens. Deze kunnen we uit de Recordset halen: sleep LENERSPAS uit de Recordset naast de tekst Lenerspas: Doe hetzelfde met NAAM, zet deze naast Naam:  
Je krijgt dan het volgende:



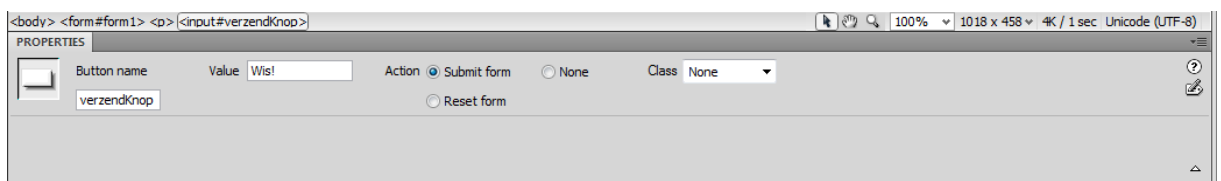
- Nu moet er nog een knop (submit button) in gezet worden. Deze komt uiteraard onder de naam.


Kies Insert>Form>Button.

Vul het volgende venster als volgt in:

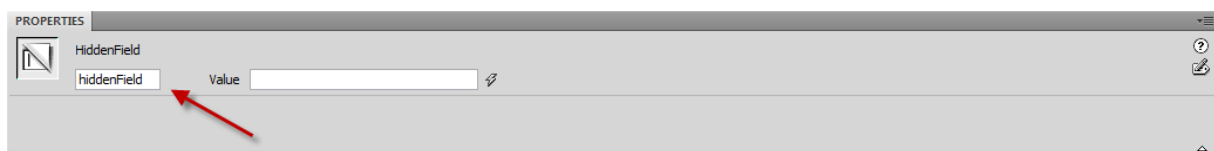


We wijzigen de eigenschappen van de knop nog even met behulp van het Properties panel: selecteer de knop en wijzig de eigenschap als volgt:



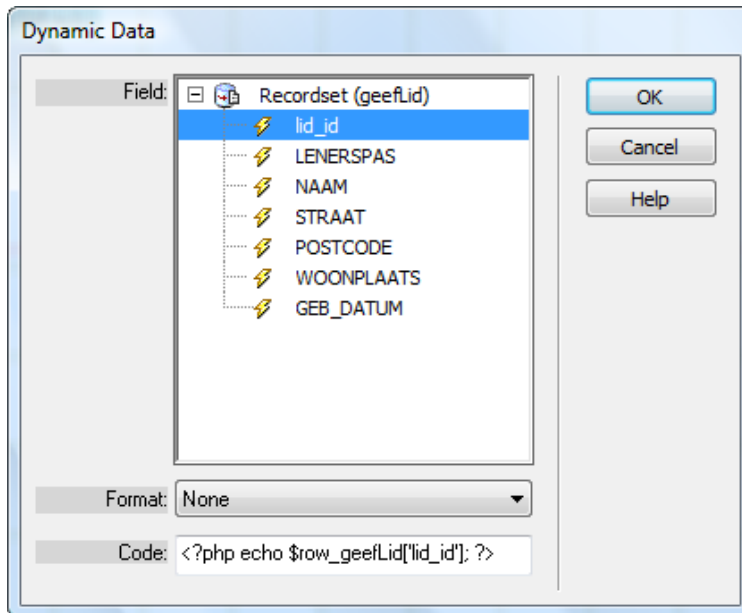
- Nu moeten we er voor zorgen dat we wel gebruiken maken van lid\_id maar dat een gebruiker van de pagina deze niet ziet. Deze is nodig omdat de primaire sleutel nodig is om een record te verwijderen. We doen dat met behulp van een verborgen tekstveld (hidden Field). Deze zetten we onder Naam:{geefLid.NAAM}; klik op Insert>Form>Hidden Field. Je krijgt dan een symbooltje in je pagina: . In dit tekstveld plaatsen we lid\_id van het geselecteerde lid. Dit doen we weer met behulp van de Recordset.

We kunnen deze niet zoals eerder er in slepen; het is immers een verborgen tekstveld! Het kan wel op de volgende manier: selecteer het symbool en met behulp van het Properties panel kunnen we dit wel doen. Na selectie geeft het Properties panel het volgende:



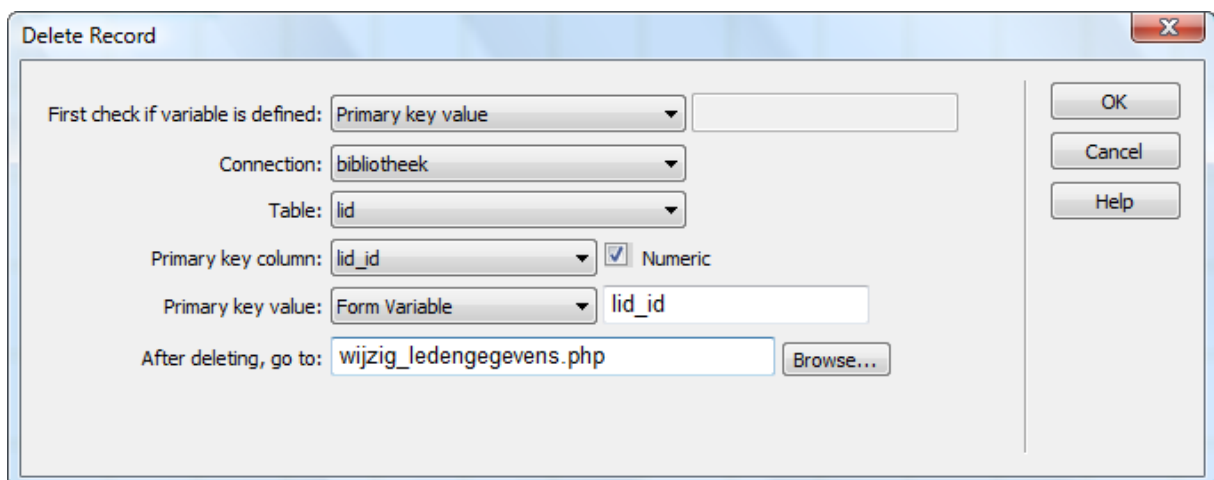
Omdat lid\_id met zijn waarde doorgegeven moet worden, geven we het veld de naam lid\_id. Verander dus de naam van Hidden Field in lid\_id (zie rode pijl).

De bijbehorende waarde (Value) koppelen, doen we met de waarde uit de Recordset. Klik hiervoor op Bind to Dynamic Source (het bliksemtekentje naast het veld Value) De Recordset wordt getoond en selecteer dan lid\_id:



Klik op OK.

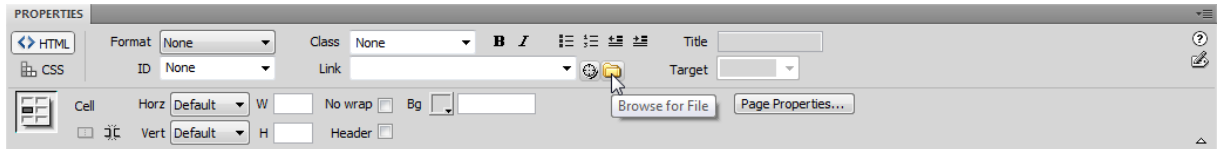
- Alle voorbereiding zijn nu gedaan om het wissen met behulp van de primaire sleutel (lid\_id) goed te laten verlopen. We kunnen nu Delete Record van Dreamweaver gebruiken. Kies In-  
sert>Data Objects>Delete Record:



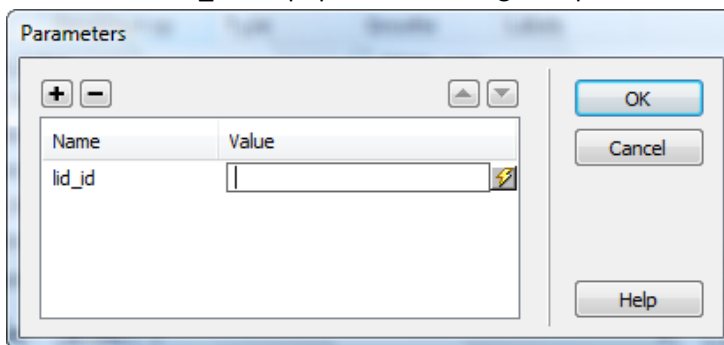
Een belangrijke opmerking: de primaire sleutel komt niet van de URL Parameter maar komt van de formulier variabele, het verborgen tekstveld. Daarom Stellen we bij Primary key value de waarde in op Form Variabele met lid\_id.



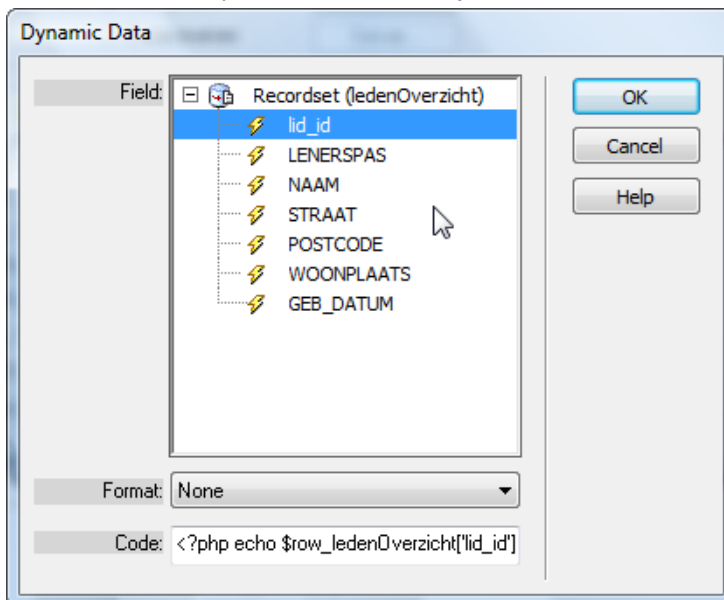
7. Indien je nu op OK klikt is de pagina in principe af! We moeten alleen de pagina nog dynamisch linken vanuit `wijzig_ledengegevens.php`.  
 Open deze en schrijf in de kolom naast `wijzig_nu_wis`. Hier maken we ook weer een link waarbij er parameter overdracht plaatsvindt: we sturen de primaire sleute, `lid_id`, van het te verwijderen lid door naar de wispagina.  
 Selecteer de tekst en maak met behulp van het Properties panel de link aan, door Browse for file te kiezen bij de link:



Kies voor `wissen_leden.php` en klik vervolgens op Parameter:



Wees weer bedacht op het goed invullen. Bij Value moet je de waarde uit de Recordset gebruiken. Klik dus op het bliksemtekentje:



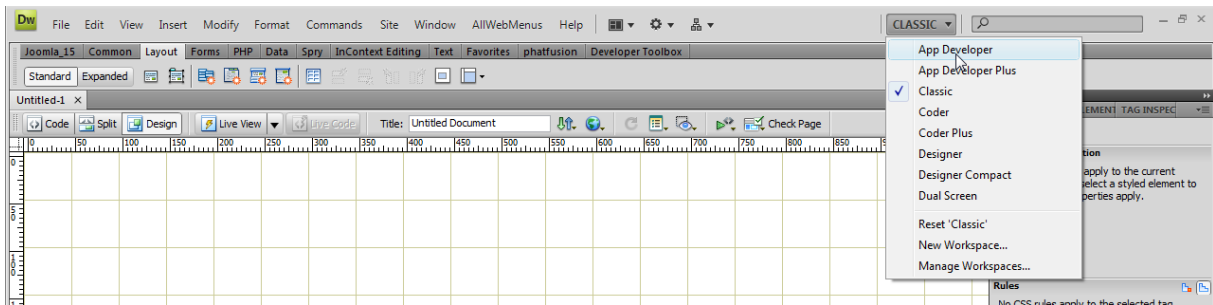
en vervolgens drie keer op OK om alle vensters weer te sluiten. Hiermee is de link klaar en zou het wissen moeten werken. Je kunt dat controleren door vanuit de pagina `wijzig_ledengegevens.php` een willekeurig lid te verwijderen.

We hebben nu alles aangemaakt om voor de tabel Lid alles te kunnen administreren met behulp van php-pagina's. Er zijn uiteraard allerlei verbeteringen mogelijk. Deze bekijken we in het hoofdstuk verdieping.

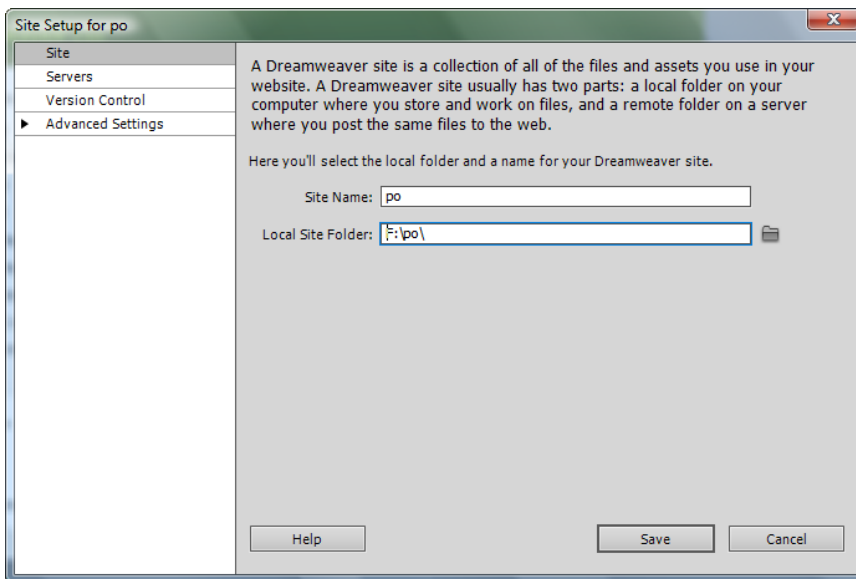
# Instellingen van Dreamweaver

Het is erg belangrijk om eerst een site te definiëren! Dreamweaver weet dan op welke wijze er met de webserver en de databases communiceert moet worden. Het is belangrijk dat je dat eerst uitvoert voordat je je php-pagina's gaat maken!

Het is bovendien handig om eerst de werkrumte anders in te stellen. Kies voor App Developer:



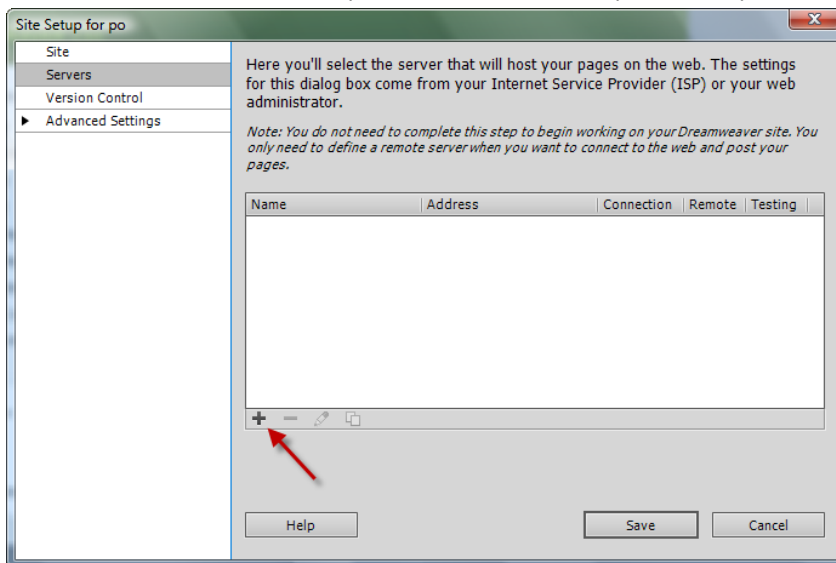
## 1. Kies Site>New Site:



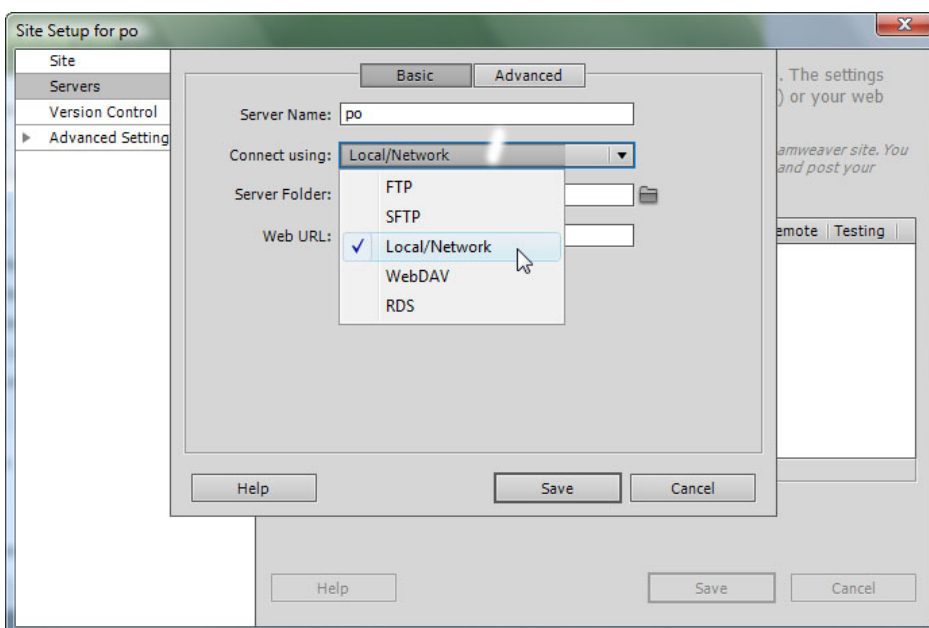
2. Geef de site een passende naam. Kies ook de juiste map voor de webdocumenten. Kies hiervoor je eigen map Mijn documenten. Het werkt het beste als je voor elk project een aparte map aanmaakt. Zo heb je goed overzicht over de bestanden die bij het project horen.

Voor het uitvoeren van de queries in een php-pagina is het noodzakelijk dat je de beschikking hebt over een webserver met MySQL en PHP. Zo'n webserver kun je heel gemakkelijk op je eigen computer installeren. Er is een groot aantal pakketten verkrijgbaar die dat allemaal hebben. In deze handleiding gebruiken we Wamp. Zie de bijlage voor de installatie van Wamp. Zorg er dus voor dat je eerst een webserver geïnstalleerd hebt. De webserver wordt bij Dreamweaver de Server genoemd. Naar deze locatie worden de bestanden geupload. Het uploaden is dus gewoon je eigen computer. Uiteraard kun je ook een externe webserver gebruiken. Hiervoor moet je de ftp-gegevens instellen.

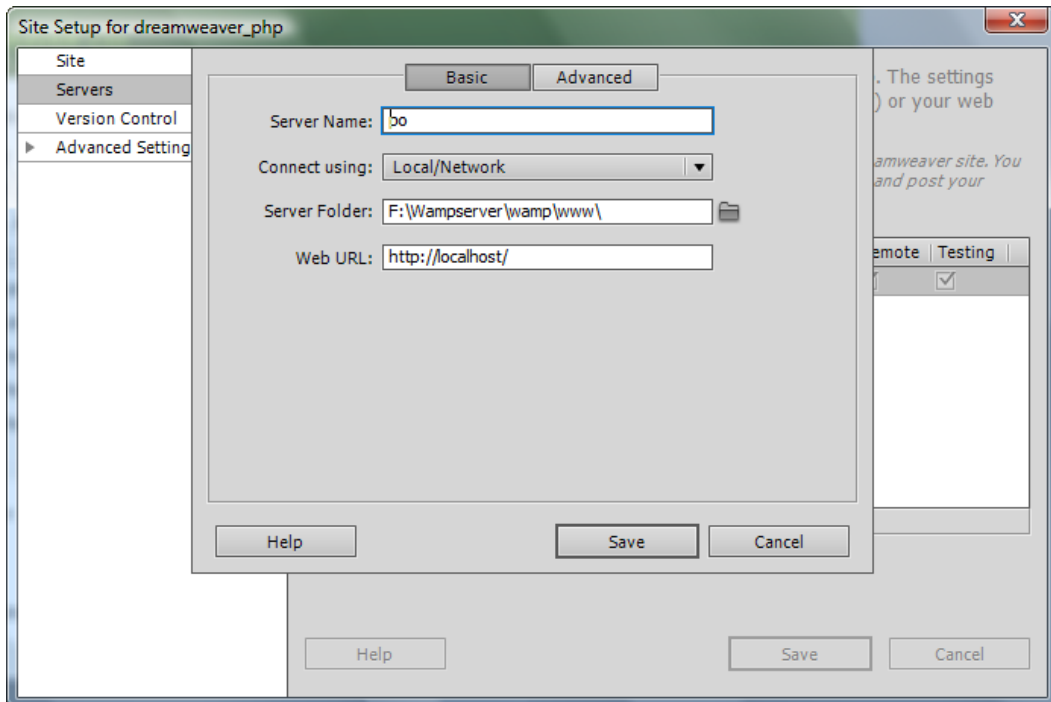
3. Klik nu in hetzelfde venster op Servers en klik dan op de + knop:



4. Geef de server een naam en klik vervolgens op Connect using. Kies hier voor Local/Network. We gaan immers de lokale webserver gebruiken.

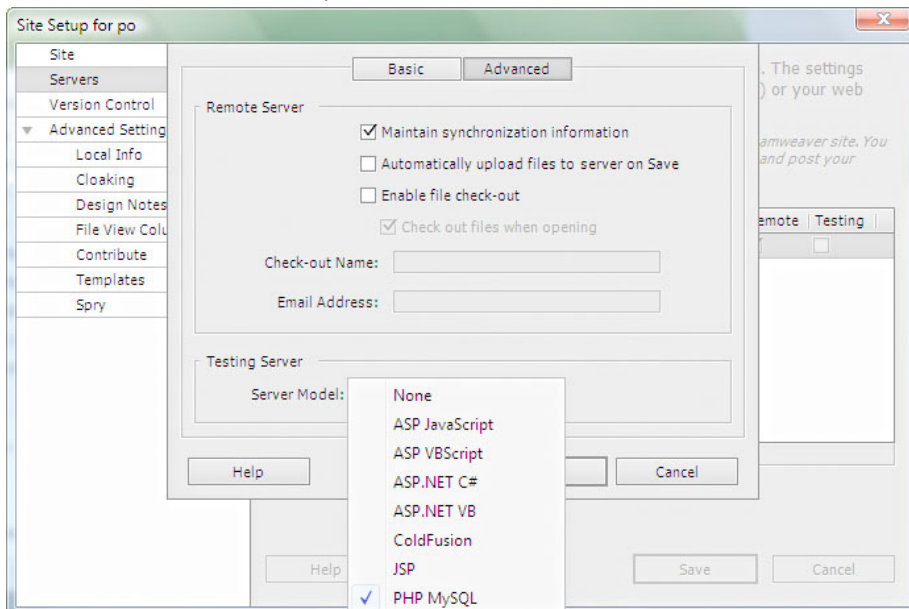


5. Geef daarna op waar de Server Folder op je computer is, dat is dus de map waar Wampserver in geïnstalleerd is. In deze map kies je de submap www. Geef daarna aan dat we de localhost gebruiken:



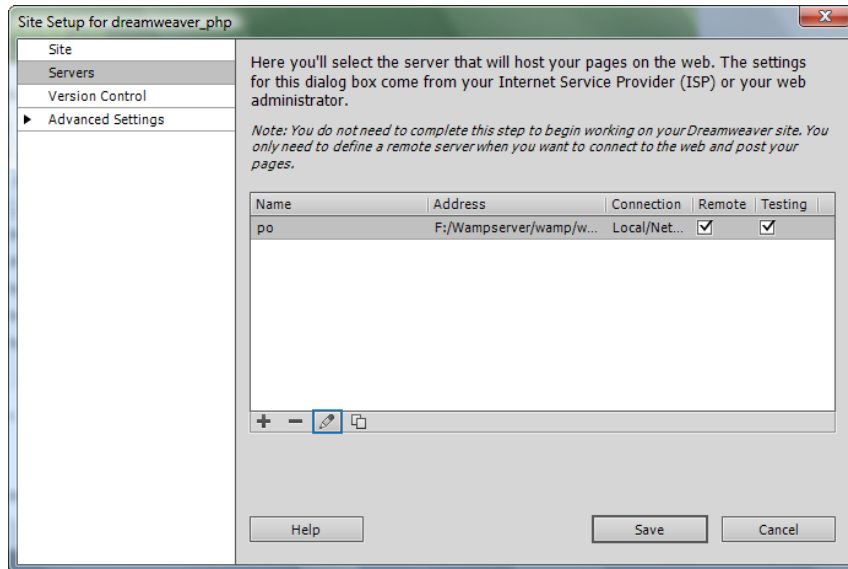
We kunnen de instellingen nu gebruiken. Als laatste stellen we nog in dat we MySQL met PHP willen gebruiken.

6. Klik nu op de button Advanced. In het nieuwe scherm klik je op Server Model. Je kiest daar voor de combinatie PHP MySQL.



Vergeet niet op Save te klikken.

Je krijgt dan onderstaand beeld:



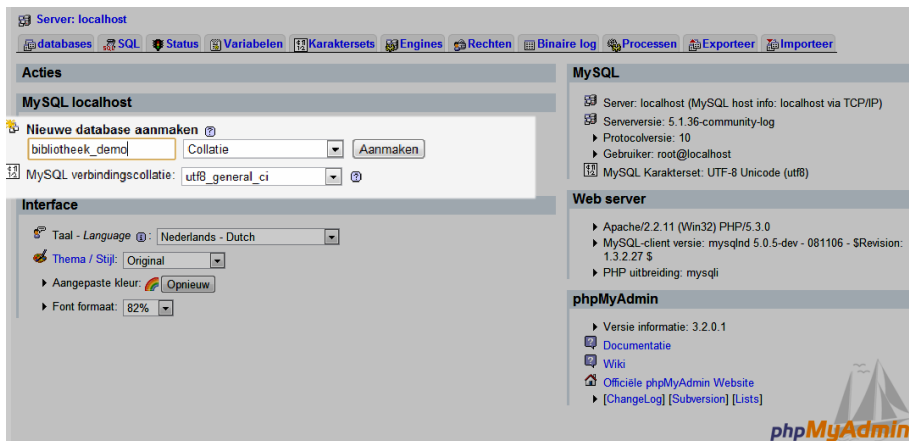
Als laatste zetten we nog even een vinkje bij Testing.

# Verplaatsen van een database

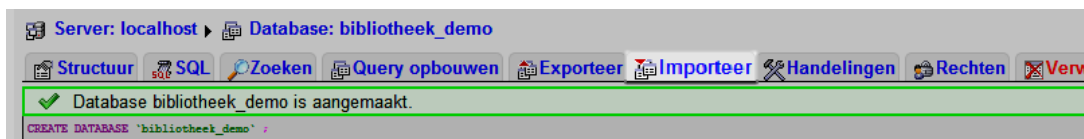
Het is mogelijk om een database te importeren in MySQL. Hiervoor heb je een bestand nodig met de extensie sql, bijvoorbeeld bibliotheek.sql. Het is ook mogelijk om de database te exporteren, zo kun je je database makkelijk verplaatsen.

We lopen de stappen door die je moet uitvoeren om een database te importeren.

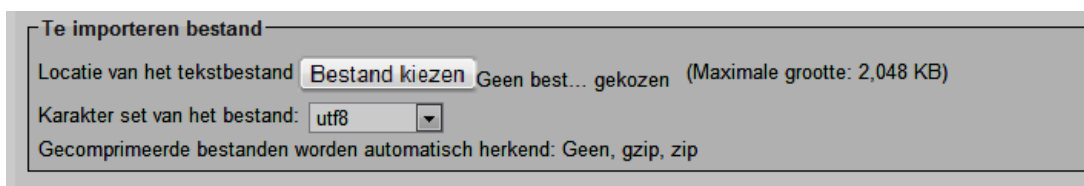
1. Start phpmyadmin en kies voor het aanmaken van een nieuwe database



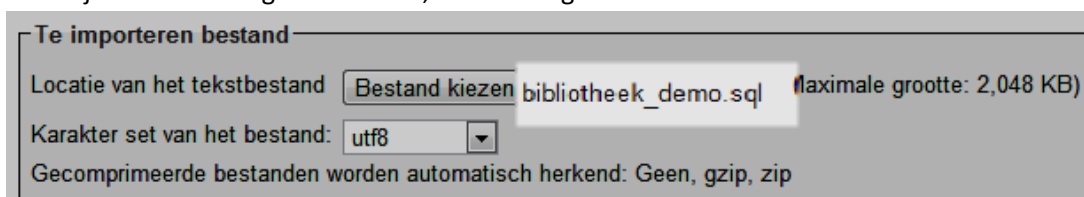
2. Kies dan in het tabblad voor importeer



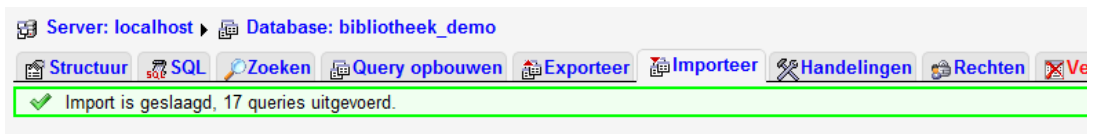
3. Je moet dan aangeven welk bestand je moet inlezen. Kies hier altijd een bestand met de extensie .sql



4. Indien je het bestand gekozen hebt, wordt dat gemeld:

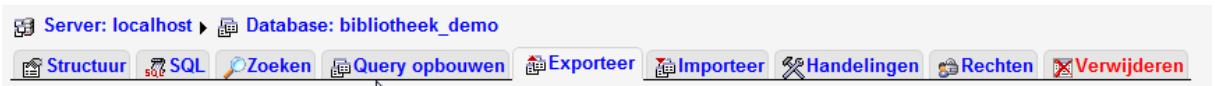


5. Klik vervolgens op Start, soms moet je dan even wachten maar MySQL meldt dan dat de import geslaagd is:

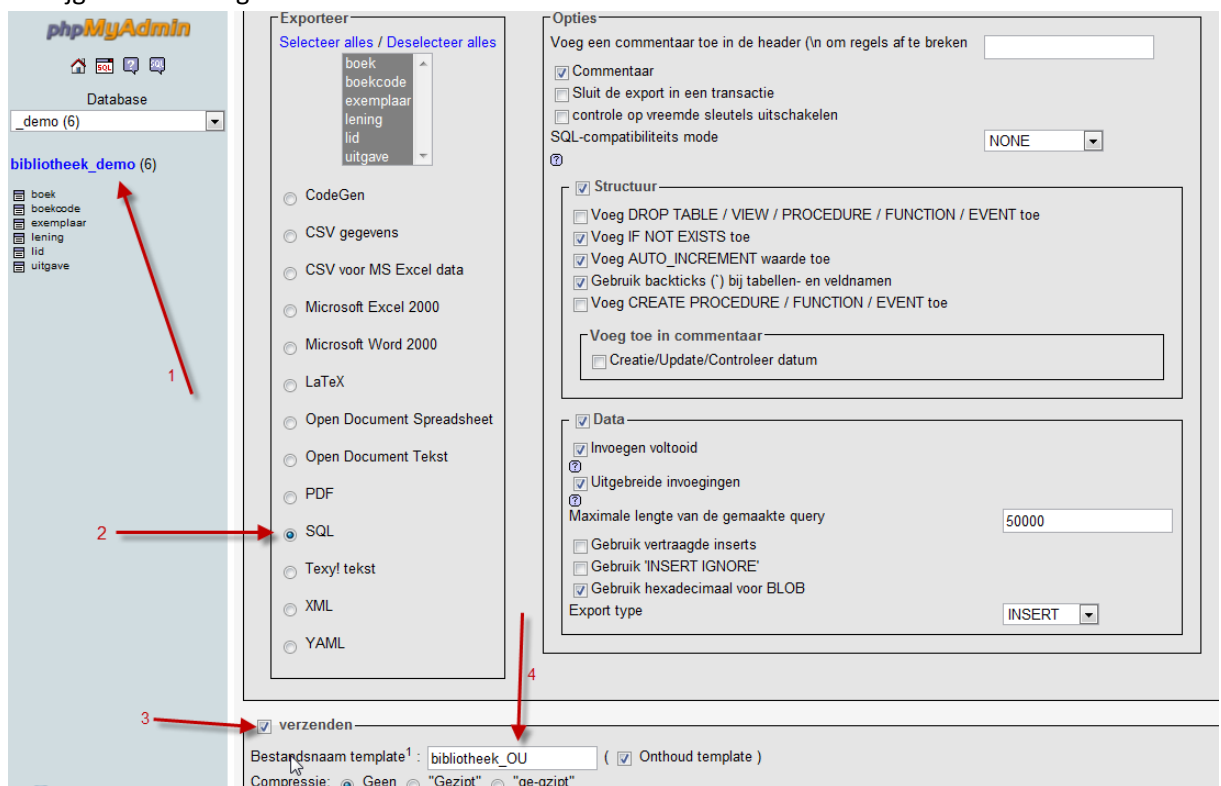


Het exporteren gaat op vergelijkbare wijze. We lopen de stappen door

1. Kies nu voor het tabblad exporteer:



2. Je krijgt dan het volgende scherm:



Let op de volgende instellingen.

Zie pijl 1: zorg dat je alle tabellen van je database ziet.

Zie pijl 2: zorg ervoor dat je als uitvoertype SQL kiest.

Zie pijl 3: zorg ervoor dat verzenden is aangevinkt.

Zie pijl 4: zorg ervoor dat je een goede bestandsnaam kiest, gebruik geen spaties in deze bestandsnaam.

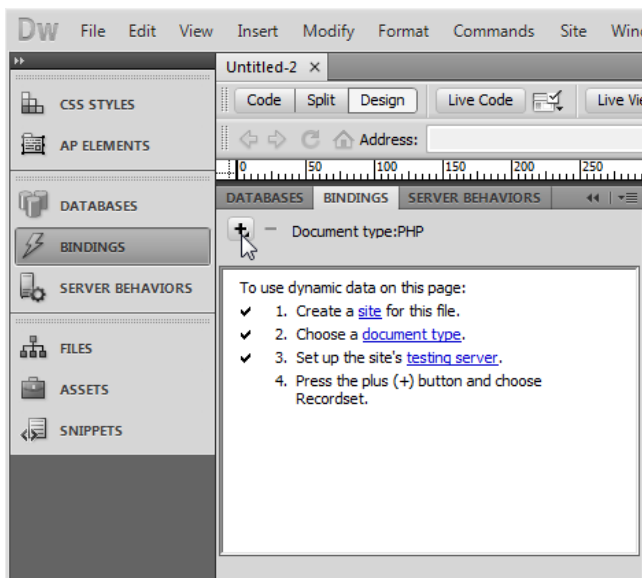
## Een Recordset met SQL

Je kunt met Dreamweaver je recordset ook gewoon met SQL maken. De recordset wordt zo flexibeler en je kunt uit meerdere tabellen tegelijk gegevens opvragen.

Stel we willen uit de bibliotheek alle namen hebben die het boek *Beginselen der Astrologie* van Snijders geleend hebben. Dat levert een forse query op. Deze is echter met SQL vrij snel gemaakt. in SQL levert dit het volgende op:

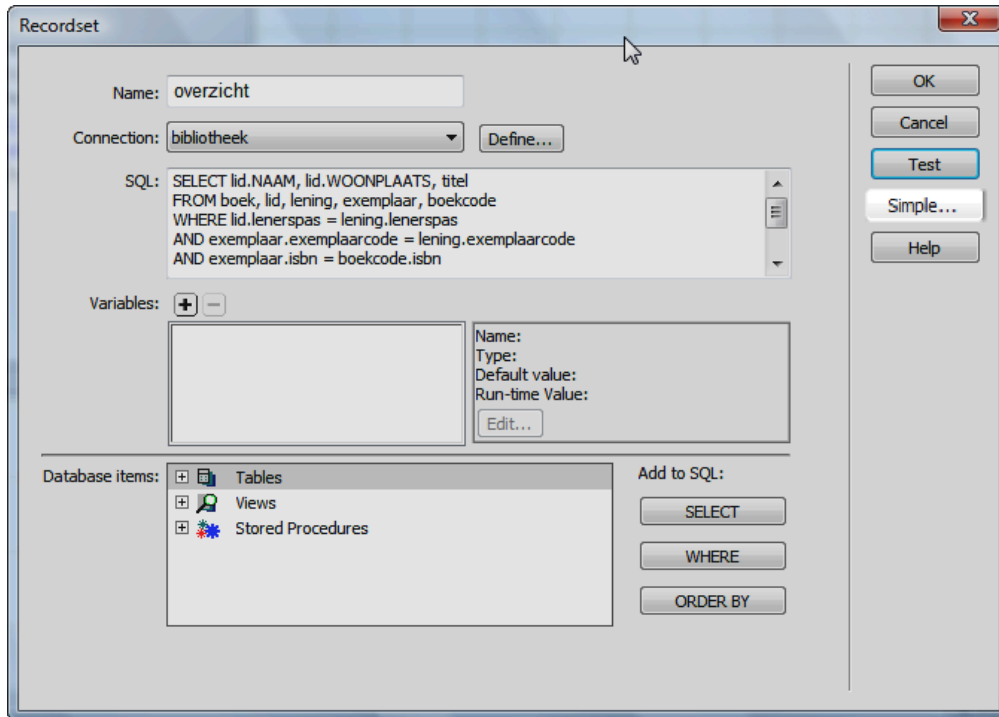
```
SELECT lid.NAAM, lid.WOONPLAATS, titel
FROM boek, lid, lening, exemplaar, boekcode
WHERE lid.lenerspas = lening.lenerspas
AND exemplaar.exemplaarcode = lening.exemplaarcode
AND exemplaar.isbn = boekcode.isbn
AND boekcode.intern_boeknr = boek.intern_boeknr
AND auteurs LIKE "Snij%"
```

In Dreamweaver maken we de recordset:





We krijgen dan:



Zorg er wel voor dat je in Advanced mode staat; je ziet dat doordat de button de mogelijkheid geeft voor simple (zie de figuur).

Je geeft de recordset weer een naam en je geeft in het SQL veld de benodigde SQL-code. Met test kun je controleren of je query werkt.

## De installatie van Wamp.

1. Download Wampserver van <http://www.wampserver.com/en/>
2. Dubbelklik op het installatiebestand.  
De installatie wordt uitgevoerd. Je krijgt na afloop een snelkoppeling op je bureaublad
3. Het starten van Wamp doe je op de gebruikelijke wijze. Je kunt zien dat het programma gestart is omdat je een icoontje rechtsonder in de taakbalk krijgt.



4. Je kunt nu met een browser via <http://localhost/phpmyadmin/> met MySQL aan de slag.