Opleiding Applicatie Ontwikkelaar

Leerlijn Database

Connectie maken met de database

Domein B Level 2

Auteur: Benn Bergmann, Erik Mast

Datum: 13-12-2017

Inhoudsopgave

Overzicht

Level: Domein B Level 2
Duur: 2 weken
Methode: Weekplanning

Voorkennis

Domein B Level 1 Introductie Database

Domein A Level 1 Programmeren PHP

Gecombineerd studeren

Domein A Level 2

Materialen

- Je laptop met;
- Editor, bijvoorbeeld Kladblok, Notepad++, Atom (☺ : aanrader!)
- Webbrowser, bijvoorbeeld Internet Explorer, Firefox, Chrome
- XAMPP

Bronnen

- Zie bijlage Bronnen. Lees voor je begint in ieder geval de bronnen door.

Instructies

- SQL opdrachten maken
- Zoeken in de referenties van SQL
- Als in de tekst staat dat je iets van de bronnen moet doorlezen, moet je dat daadwerkelijk doen.

Beschrijving

In deze opdracht leer je SQL gebruiken om gegevens op te halen en te bewaren in de database. Eerst leer je de verschillende SQL-statements en vervolgens gebruik je deze bij het maken van de eindopdracht.

Ook leer je hoe je regels aan een tabel kunt toevoegen, weggooien van regels en het aanpassen van regels, en een overzicht te maken van alle regels uit een tabel.

Doelen

Na deze periode zal je in staat zijn simpele SQL-statements te maken, en deze toe te passen op de database. Ook zal je in staat zijn om contact te maken met een database te maken en daar gegevens uit te halen en in te bewaren.

Beoordeling

Deze opdracht wordt beoordeeld aan de hand van de eindopdracht en een gesprek over het eindproduct.

Studieonderdelen

SQL-statements (+/- 32 uur)

SELECT, INSERT en DELETE statements.

Theorie

In het gebruik van databases staat het begrip CRUD centraal. Dit is een afkorting van

"Create, Read, Update and Delete"

Vertaald zijn dit de elementaire acties die je op een database kunt doen namelijk:

Wat doet het	Afkorting	SQL opdracht
Een nieuwe regel maken in een tabel.	CREATE	INSERT
Eén of meer regels uit een tabel lezen.	READ	SELECT
Een regel in de tabel aanpassen	UPDATE	UPDATE
Een regel uit de tabel verwijderen	DELETE	DELETE

Elk van deze SQL-Statements heeft zijn eigen syntax (hoe je de opdracht correct geeft aan een database). Deze syntax kun je terugvinden in de bronnen.

Inleiding oefenopdrachten

In onderstaande opdracht wordt steeds gevraagd om de SQL-statements die je gemaakt hebt te bewaren in een tekstbestand. Doe dat ook, want er wordt naar gevraagd.
Er wordt steeds gevraagd om de opdrachten helemaal uit te typen. Dit is noodzakelijk om oefening te krijgen in het maken van SQL-statements. Als je kopiëren en plakken gebruikt zal je dat later een probleem worden, omdat je dan elke keer moet opzoeken hoe het werkt. En daardoor zal je misschien voor gek staan bij je collega's.

Hoe noem je een tabel?

Er is op internet tussen technici een discussie gaande over hoe je je tabellen zou moeten noemen.

1. Meervoud (bv songs)
    a. omdat er meer songs in een tabel zitten.
    b. Het beter leest in een query
2. Enkelvoud (bv song)
    a. Omdat de table een relatie beschrijft en niet een ding (is misschien vaag op dit moment)
    b. Het leest beter in de "where" van de SQL-opdracht
    c. Elk record is een representatie van één ding.

Wat jij zelf ook vindt: zorg altijd dat je naamgeving consequent is. En ga soepel om met de keuze van je team, consequent zijn is belangrijker dan of één iemand gelijk heeft.

Oefenopdracht 1

Maak in PhpMyAdmin een nieuwe database aan en noem die “db_level2_opdr1”.Maak in deze database een tabel (“songs”) waarin je je favoriete muzieknummers bewaren kunt.

Deze tabel heeft een kolom met een uniek Id veld (inmiddels weet je hoe dat moet), een veld met artiestnamen(“artist”), en een veld met namen van songs(“title”). Bepaal zelf het datatype van de velden.

a. Vul deze tabel met minimaal 5 songs van minimaal 2 verschillende artiesten. Meer mag natuurlijk ook. Doe dit door de INSERT opdracht te gebruiken, en type de opdracht elke keer helemaal uit.
   Dit is belangrijk omdat je dit moet oefenen.
   Tip: Maak een apart tekstbestand en bewaar daarin de door jou uitgevoerde SQL-opdrachten, zodat je ze later nog een keer kunt gebruiken.
b. Ga bij elke regel iets veranderen aan de naam van de artiest of de naam van de song. Doe dit ook minimaal 5 keer met de UPDATE opdracht, type ook deze opdracht elke keer helemaal uit. Bewaar ook deze opdrachten in het tekstbestand. Controleer na elke opdracht in PhpMyAdmin of het gewenste resultaat bereikt is.
c. Ga met de opdracht SELECT een overzicht maken van alle songs, bewaar ook deze opdracht in je tekstbestand.
d. Gebruik SELECT om alle songs van één artiest te zien. Bewaar deze SQL ook in je tekstbestand.
e. Gebruik SELECT om één song te zien. Daarvoor zijn meer mogelijkheden, probeer te bedenken welke en probeer deze uit. Bewaar ook deze in je tekstbestand.
f. Wat is de handigste manier om één song te laten zien. Bewaar deze query en zet erbij waarom.
g. Gebruik DELETE om twee favoriete nummers te verwijderen. Bewaar ook deze in je tekstbestand.
h. Exporteer de database en bewaar het resultaat onderaan in je tekstbestand.
i. Controleer je export door deze in een **nieuwe** database te importen.

Contact maken met de database (+/- 16 uur)

Als je XAMPP (of een andere "personal" webserver zoals USBWebserver) hebt geinstalleerd, heb je een databaseserver in je computer én een webserver. Als je die combineert kun je vanzelfsprekend veel meer doen. Het is logisch dat dit kan, nu gaan behandelen hoe je zoiets kunt doen.

Theorie

Voor we echt contact kunnen maken met de database, moeten we toch even praten over **Objecten**. Later zullen we daar dieper op gaan, maar nu kijken we er even kort naar:

Objecten in 't kort

Een object is eigenlijk een programmeer-eenheid. De belangrijkste eigenschap is, dat het code en eigenschappen bundelt. Je moet het een beetje zien als een variabele (zoals een integer oid) die zijn eigen zaakjes kan regelen. Zoals jij zelfstandig naar school kan komen, je weet hoe je er moet komen, welke buslijnen je eventueel moet nemen, en welke kant die buslijnen op gaan. Als we tegen jou zeggen "Ga naar huis" dan weet jij waar je huis is, en hoe je daar kunt komen.

Connectie-object

Als je contact wil maken met de database, gebruik je een connectie-object. Voor je connectie moet maken, zul je toch eerst wat **parameters** (gegevens) van je eigen databaseserver moeten achterhalen. We zullen hier stap-voor-stap uitleggen hoe het werkt.

Allereerst maken we een gewoon php-bestand aan

```
<?php
```

Je gebruikt vier variabelen, een voor de server (localhost wat het is immers je eigen computer), een voor de naam van de database, een voor de gebruikersnaam en een voor het wachtwoord. Dit is net als bij allerlei andere systemen, zoals bijvoorbeeld de portal van school.

Deze variabelen moet je natuurlijk wel vullen met de waarden die voor jouw computer van toepassing zijn.

```
$servername = "localhost";
$databasename = "database";
$username = "username";
$password = "password";
```

Uitleg over connectieparameters MySQL.

Voordat je een connectie met de database kunt vaststellen zul je eerst de vier variabelen uit de code de juiste waarden moeten geven.

$servername
Dit is de naam van de server waar de database op staat. In het geval van een door XAMPP gehoste server is de juiste waarde hier "localhost", ofwel de server die lokaal (op je laptop) draait.

$databasename
De naam van de database waarmee je een connectie wilt maken. Als je, bijvoorbeeld, de database uit de oefenopdrachten wilt gebruiken, is de juiste waarde hier "db_level2_opdr1".

$username & $password
Om een connectie te maken heb je bovendien een gebruikersaccount nodig dat toegang heeft tot de database.

Accounts
Op de startpagina van phpmyadmin vind je bovenin een link "Gebruikersaccounts". Hier vind je informatie over de bestaande gebruikersaccounts of kun je een nieuw account maken. Zorg dat het account dat je gebruikt een gebruikersnaam en wachtwoord heeft en de rechten heeft om connectie met de database te maken.
Connectie maken met het root-account is **niet wenselijk** en een slechte gewoonte. Het root-account is een speciaal account waar alle rechten op staan. Het is dus ook een paradijs voor **hackers**. Je kunt heel eenvoudig een account aanmaken en een bijbehorende database aanmaken.

Vervolgens maak je een connectie-object (zie hieronder). Wat hier gebeurt is dat in de variabele een stuk code met eigenschappen wordt opgeborgen. Op de achtergrond wordt al contact gezocht met de database. Het leuke is dat je daarna de variabele opdrachten kunt geven. Maar daarover later meer.

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

Omdat op de achtergrond blijkbaar iets gebeurd is, kun je vervolgens direct vragen of het fout gegaan is. En als er inderdaad iets fout is gegaan, kun je php vragen om onmiddellijk te stoppen en de fout te melden. En als er niets fout gaat kun je php laten doorgaan.

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

?>
```

Oefenopdracht 2

Gebruik de code in de theorie hierboven, en probeer deze werkend te krijgen voor jouw laptop. Gebruik daarvoor de database “databaselevel2opdr1” die je in opdracht 1 hebt gemaakt.

Theorie

Het object wat je gemaakt hebt door contact te maken met de database kun je allerlei opdrachten geven:

$conn->query($sql)

Met deze opdracht voer je een SQL opdracht uit op de databaseserver. Het resultaat is ook weer een **object** waar je dus ook weer opdrachten aan kunt geven!

$conn->close()

Deze opdracht is om de connectie af te sluiten, als je klaar bent. Dat is wel zo netjes.

Er zijn nog veel meer opdrachten, die kun je terugvinden in de bronnen.

Het volgende object

Het resultaat van **$conn-query($sql)** is ook weer een object zoals hierboven al vermeld is.

Dat object kun je ook weer allerlei interessante opdrachten geven zoals **$result->fetch_assoc()**. En de uitkomst daarvan is een **associatief array**. Een array waarin je de verschillende waarden terug kunt vinden op basis van een naam en niet van een index, zoals dat bij simpele arrays gaat.

Oefenopdracht 3

Laat je favoriete songs zien

Lees “MySQLi for beginners” uit de bronnen goed door tot en met “Close that connection”, en maak een webpagina, die de inhoud van de “favoritesongs” tabel laat zien. Opmaak is niet belangrijk, maar als je extra wilt oefenen met <TABLE> tags, ga dat vooral doen. Het zal je docenten erg blij maken.

Verfraai dit bestand ook met enige opmaak in CSS, en zorg dat het een net HTML bestand wordt.

Theorie

Je hebt eerder al wat kunnen oefenen met het maken van **INSERT** query’s. Het laatste wat de gebruiker wil, is dit soort query’s intypen en hopen dat het in de database terecht komt.

Een INSERT query kun je op dezelfde manier uitvoeren als de overzichtsquery, alleen zal er natuurlijk geen resultaat teruggeven.

Het is handiger dat je daarvoor een invoerformulier hebt. Kijk eens terug naar de module Programmeren Level 2 hoe je daar een HTML form gemaakt hebt. Deze kennis ga je combineren met de kennis hierboven.

Oefenopdracht 4

Maak een nieuw php-bestand, waarin je een favoriete song kunt invullen, en zorg dat deze verwerkt wordt in de database. Als de verwerking geslaagd is, spring je terug naar het bestand waar je het overzicht kunt zien, dat is het bestand wat je in de vorige oefenopdracht gemaakt hebt. Zorg ook dat je in dat bestand een knop hebt waarmee je naar je toevoeg formulier springt. Voor het formulier geldt ook dat je die opmaakt met CSS en dat hij past bij je vorige scherm.

Oefenopdracht 5

Maak een kopie van de form uit opdracht 4 (voor de zekerheid)

Plaats op het overzicht scherm voor elke song een knop, waarmee je die song zou kunnen wijzigen. Kijk of je de form uit opdracht 4 kunt aanpassen zodanig dat de gegevens van je song ingevuld worden in die form. Je kunt bijvoorbeeld via een sessie variabele of parameters in de adresregel (beide technieken moet je uiteindelijk kennen dus verken beide mogelijkheden) of op een andere manier. Beantwoord de vraag: **Hoe zou dat ook kunnen?**

Oefenopdracht 6

Je kunt gegevens toevoegen en een overzicht maken, en je kunt een scherm aanroepen met de gegevens al ingevuld. Ga nu dit zodanig aanpassen dat de gewijzigde gegevens vanuit de form ook in de database terecht komt. Hint: je moet een **UPDATE** query gebruiken, ga uitzoeken hoe dat werkt voor zover je dat nog niet gedaan hebt. Die query moet je programmeren en de wijzigingen die je met de form aanbrengt in de database moeten zichtbaar worden in het overzichtsscherm.

Oefenopdracht 7

Plaats naast elke knop in het overzichtsscherm nog een knop. Zorg dat die knop steeds de geselecteerde song weggooit en dat je daarna terugkeert naar het overzicht scherm.

Samengevat

Nu heb je alle elementen van een CRUD een keer gezien:

C: het toevoegen van een regel

R: het lezen van één of meer regels

U: het wijzigen van een regel

D: het verwijderen van een regel.

Nu is het tijd om daar nog wat mee te oefenen

Oefenopdracht 8

Maak nu een totaal set van bestanden voor CRUD met een tabel waarin je je verlanglijstje voor je verjaardag kunt bijhouden. Zorg dat je een veld hebt voor prijs, omschrijving, en waar het te koop is, en een veld voor een webadres van de betreffende winkel. Denk goed na wat voor velden je gaat gebruiken.

Eindopdracht (+/- 16 uur)

Maak een CRUD set voor de verjaardagen van je vrienden en familie. Zorg dat je de geboortedatum vastlegt en ook dat je uitrekent hoe oud iemand dan wordt. Je zult velden voor voor- en achternaam nodig hebben en geboortedatum. Uitdaging hierbij is vanzelfsprekend de leeftijd, je zult veel ontzag oogsten als je ook de huidige leeftijd in dagen en/of maanden kunt afdrukken in het overzichtsscherm, maar let op! Dat is niet iets wat je in je database bewaart.

De opdrachten (zowel oefenopdrachten als eindopdracht) lever je in in Magister. Zorg altijd de website ingepakt verstuurd wordt (zipfile) en vergeet niet om ook de SQL te exporteren en mee te sturen. Als je niet weet hoe je een zipfile maakt: Klik rechts op de map waar jouw bestanden staan, en kies daarna voor “Kopiëren naar”->Gecomprimeerde (gezipte) map. Er wordt dan een zipfile gemaakt die je kunt opsturen.

Bronnen

SQL

- ✓ **Create, Read, Update and Delete**
  https://en.wikipedia.org/wiki/Create,_read,_update_and_delete

- ✓ **SQL Tutorial**
  Lees deze door tot en met het hoofdstuk “SQL Delete”

  https://www.w3schools.com/sql/

- ✓ **PHP: MySQL Database**
  https://www.w3schools.com/PhP/php_mysql_intro.asp

Connectie maken

- ✓ **PHP Connect to MySQL**
  https://www.w3schools.com/php/php_mysql_connect.asp

MySQLi referentie

- ✓ **PHP5 MySQLi functions**
  https://www.w3schools.com/php/php_ref_mysqli.asp
- ✓ **MySQLi for Beginners**
  http://codular.com/php-mysqli

Enkelvoud of meervoud in de tabelnaam

- ✓ **Use singular names nouns for table names**
  https://www.teamten.com/lawrence/programming/use-singular-nouns-for-database-table-names.html