

## Opdracht Gegevensbanken - deel 3: DB creatie en front-end

In de eerste twee delen van de opdracht heb je het ER-model van de database opgesteld voor de luchtvaartsector en dit vertaald naar een relationeel model. Een modeloplossing hiervoor is te vinden op Toledo (Course documents -> Opdracht -> Deel 1/2 - Part 1/2 - Oplossing). In dit derde deel zullen we de database creëren en een front-end webtoepassing ontwikkelen om de gebruikers te laten interageren met de database. Je dient te starten van de modeloplossing voor deel 1 en 2, aangezien je zo direct kan starten met het aangeleverde schema en data.

Op het einde van deze opdracht dient een (kort) **verslag (met een maximum van 2 pagina's)** geschreven te worden over jullie tussentijdse bevindingen over de opdracht. Een template hiervoor met enkele vragen vinden jullie op Toledo.

### Indienen

Zowel het verslag (in PDF) als een ZIP-bestand met de code dienen ingediend te worden **via de indienmodule op Toledo**. De deadline is **maandag 7 april 2014, 9u00**. **Zorg dat de code voldoende gedocumenteerd is om na te gaan wat jullie met een bepaald stuk code willen bereiken**. Slechts één groepslid hoeft in te dienen.

### 1. Web front-end met PHP

Nu gaan we kijken naar het maken van dynamische webpagina's met PHP en een MySQL gegevensbank. Het is de bedoeling je vertrouwd te maken met de basis van HTML en PHP d.m.v. een korte inleiding. Hierna wordt er een connectie opgezet met de MySQL-gegevensbank en worden er een aantal queries uitgevoerd. De resultaten van deze queries worden weergegeven in de HTML-pagina's.

#### 1.1 Inleiding tot HTML

HTML is de taal waarmee webpagina's gemaakt worden. HTML staat voor "Hyper Text Markup Language": het is een serie van codes die toegevoegd worden aan tekst, en met die codes (men noemt ze "tags") wordt aangegeven hoe de layout van de pagina er moet uitzien. We kunnen de broncode van een webpagina ongeveer in elke webbrowser bekijken (in Firefox is dat View > Page Source, of ctrl-U).

```
<html>
<head>
  <title>Welgekomen</title>
</head>
<body>
  <h1>Ga zitten</h1>
  <p>Drink iets. Gemakkelijk gevonden?</p>
  <p>Dit is een webpagina, met een beetje tekst erop. Kijk: <strong>deze tekst staat
in het vet!</strong> En wat nu? <em>Deze tekst staat cursief!</em></p>
</body>
</html>
```

Hierboven vind je een voorbeeld van een HTML-pagina. HTML heeft vaste namen voor zijn elementen. Om met HTML te kunnen werken is het een kwestie te weten welke elementen er zijn, wat ze doen en waar ze gebruikt mogen worden. Een webpagina heeft altijd dezelfde basisstructuur: een html-tag, met daarbinnen een head-tag en een body-tag. In de body-tag staat de eigenlijke inhoud van de pagina die de bezoeker zal zien. In de head-tag staan vooral technische instructies voor de webbrowser, dingen die de gebruiker niet meteen ziet op de pagina (zoals bijvoorbeeld de titel die de browser weergeeft bovenaan in de vensterbalk). Je kan de HTML uit het voorbeeld hierboven ook grafisch voorstellen, zie figuur 1.



**Figuur 1:** Grafische representatie van de HTML pagina.

De gebruikte tags in de body in het stukje hierboven:

- `<h1>`: een hoofding van niveau 1 (het grootste). Je kan ook kleinere subtitels maken door `<h2>`, `<h3>`, `<h4>`, `<h5>` en `<h6>` te gebruiken. Na deze tag begint er een nieuwe regel.
- `<p>`: een paragraaf (verschillende spaties of tabs na elkaar binnen een paragraafelement om "opmaak" te doen, wordt genegeerd door de browser). Na deze tag begint er een nieuwe regel.
- `<strong>`: tekst in het vet
- `<em>`: tekst italiek (em van emphasis)
- `<u>`: onderlijnen, zo: onderlijnd
- `<s>`: doorstrepen, zo: ~~doorstreept~~
- `<tt>`: in "getypte tekst" zetten, zo: getypte tekst

Een link naar een bestand of een andere webpagina kan je maken door het element `<a>...</a>` (anker) te gebruiken. De locatie kan je opgeven m.b.v. het attribuut href. Als ik wil dat "hier" in de paragraaf "Klik zeker ook hier." een link wordt, dan doe je dat zo:

```
<p>Klik zeker ook <a href="http://maddox.xmission.com/">hier</a>.</p>
```

Hieronder een voorbeeld van een tabel. Deze tabel bevat 2 rijen en 2 kolommen. Een rij maak je met het `<tr>`-element en voor elke kolom in een rij plaats je in het rij-element een `<td>`-element. Voor meer informatie kijk op: [http://w3schools.com/tags/tag\\_table.asp](http://w3schools.com/tags/tag_table.asp)

```

<table>
  <tr>
    <td>Cell 1.A</td>
    <td>Cell 1.B</td>
  </tr>
  <tr>
    <td>Cell 2.A</td>
    <td>Cell 2.B</td>
  </tr>
</table>
  
```

```
</tr>
</table>
```

Een voorbeeld van een ongeordende lijst vind je hieronder. <ul> (unordered list) wordt gebruikt voor het maken van een lijst met bullet points. <ol> wordt gebruikt voor een geordende (genummerde lijst). De elementen in een lijst worden weergegeven door lijstelementen <li> (list items). Voor meer informatie over lijsten, kijk op: [http://w3schools.com/tags/tag\\_ul.asp](http://w3schools.com/tags/tag_ul.asp)

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Sake</li>
</ul>
```

Voor meer informatie over alle tags gebruikt in XHTML, kan je deze pagina raadplegen: <http://w3schools.com/tags/default.asp>

## 1.2 Inleiding tot PHP

PHP staat voor "PHP: Hypertext Preprocessor". Het is een open-source, server-sided scripting taal waarmee het mogelijk wordt om dynamische webpagina's te programmeren.

Zo een dynamische pagina krijgt de extensie .php. Dit bestand kan een combinatie van html en php code bevatten. Alle php code komt tussen php tags te staan. Bij het opvragen van een php pagina zal de server de code tussen deze tags uitvoeren. Enkel het resultaat hiervan wordt dan doorgestuurd naar de client. Hieronder staat de "Hello World" van php.

```
<html>
<body>
  <?php
    echo "Hello World";
  ?>
</body>
</html>
```

De php code `echo "Hello World";` geeft als uitvoer een HTML-pagina met "Hello World" op. De echo functie laat dus toe resultaten naar de browser te sturen en is daardoor een van de belangrijkste functies van php. Het resultaat van dit php script staat hieronder.

```
<html>
<body>
  Hello World
</body>
</html>
```

Dit is natuurlijk niet echt een zinvol script. Het resultaat is immers altijd hetzelfde en de pagina is dus niet dynamisch. Door meer gebruik te maken van de php taal kunnen we bepaalde code vereenvoudigen of gegevens opvragen uit een gegevensbank en die dan verwerken in de pagina. In de volgende paragrafen geven we een kort overzicht van de belangrijkste php functies. De syntax is relatief eenvoudig en bestaat uit een mengeling van Java, C en Perl.

### *Variabelen en hun operatoren*

Zoals in elke programmeertaal kunnen we in php waarden toekennen aan variabelen om deze naderhand te gebruiken. Hieronder staan enkele voorbeelden.

```
/*
Dit is commentaar. Commentaar wordt genoteerd zoals in Java.
Tussen /* en */ of voor een enkele regel na //
*/
```

```
<?php
  $hello = "Hello World!";
  echo $hello; //geeft als uitvoer "Hello World!"
  $a_number = 4;
  $another_number = 7;
  $sum = $a_number + $another_number;
  echo $sum; //geeft hetzelfde resultaat als echo 11
```

```
?>
```

Zoals je in het voorbeeld kan zien beginnen alle variabelen in php met een \$ teken, gevolgd door een letter of een underscore.

In tegenstelling tot bijvoorbeeld java moeten de variabelen niet eerst gedeclareerd worden en zijn ze dynamisch getypeerd.

Op numerieke variabelen kunnen de gebruikelijke aritmetische operators toegepast worden: +, -, /, \* en %. In het vorige voorbeeld wordt de + operator gebruikt.

Ook de gecombineerde operators zijn mogelijk. Zo is bijvoorbeeld  $\$x += 2$  gelijk aan  $\$x = \$x + 2$

### Concatenatie van strings

Een veel voorkomende operatie is het concateneren van strings. In php kan dit eenvoudig met de punt operator zoals in het voorbeeld hieronder wordt gedemonstreerd.

```
<?php
$hello = "Hello";
$world = "World";

echo $hello." ".$world."!";
?>
```

### Include en Require

Vaak willen we bepaalde delen code opnieuw gebruiken op verschillende pagina's. Denk bijvoorbeeld aan een navigatiemenu of een header. PHP voorziet hiervoor de include functie. Deze functie plakt alle code in de included file op de plaats waar de functie wordt opgeroepen.

Bijvoorbeeld: het menu.php-bestand ziet er als volgt uit

```
<html>
<body>
<a href="default.php">Home</a>
<a href="pag1.php">Pagina 1</a>
<a href="pag2.php">Pagina 2</a>
```

De andere pagina's van de website kunnen dan deze pagina "includen":

```
<?php include("menu.php"); ?>
<h1>Welkom</h1>
</body>
</html>
```

de client krijgt dan volgende code toegestuurd:

```
<html>
<body>
<a href="default.php">Home</a>
<a href="pag1.php">Pagina 1</a>
<a href="pag2.php">Pagina 2</a>
<h1>Welkom</h1>
</body>
</html>
```

require() werkt net als include maar geeft een "fatal error" als de pagina niet gevonden kan worden, i.t.t. include() dat enkel een waarschuwing geeft.

### If/Elseif/Else

If en else statements werken op de gebruikelijke manier:

```
<html>
<body>
<?php
    $a=10;
    if ($a==20)
        echo "a is 20";
    elseif ($a>20)
        echo "a is larger then 20";
```

```

        else{
            echo "a is smaller then 20"
            echo "-----"
        }
    ?>
</body>
</html>

```

### Lussen

PHP kent while en for lussen. Een voorbeeld, de getallen 1 tot 10 afdrukken met een while lus:

```

<?php
    $i=1;
    while($i<=10)
    {
        echo $i."<br>";
        $i++;
    }
?>

```

en hetzelfde script met een for-lus:

```

<?php
    for ($i=1; $i<=10; $i++)
    {
        echo $i."<br>";
    }
?>

```

Itereren over een array kan eenvoudig met de foreach methode:

```

<?php
    $arr=array("een", "twee", "drie");
    foreach ($arr as $waarde)
    {
        echo "Waarde: ".$waarde."<br>";
    }
?>

```

### Funcities

Een voorbeeld:

```

<?php
    function echo_square($number)
    {
        return $number*$number;
    }

    echo "Het kwadraat van 12 is ".echo_square(12);
?>

```

### Formulieren

Om te communiceren met de client kan men gebruik maken van formulieren.

De velden van het formulier zijn automatisch beschikbaar in het php-script d.m.v. de "super global" \$\_POST variabele. Een "super global" variabele is altijd beschikbaar voor alle PHP-bestanden of -klassen zonder dat deze moet gedefinieerd worden. We demonstreren dit a.d.h.v een voorbeeld.

```

<html>
<body>
    <form action="formHandler.php" method="post">
        Voornaam: <input type="text" name="voornaam" />
        Achternaam: <input type="text" name="naam" />
        <input type="submit" />
    </form>
</body>

```

```
</html>
```

Wanneer de gebruiker nu dit formulier verstuurt dan worden de gegevens doorgestuurd naar het `formHandler.php`-script. Daar kunnen dan de gegevens gebruikt worden. `formHandler.php` kan er bijvoorbeeld als volgt uitzien:

```
<html>
<body>
    Hallo <?php echo $_POST["voornaam"]." ". $_POST["naam"]."!" ; ?>
</body>
</html>
```

Indien je na het lezen van deze korte inleiding tot PHP meer info zou nodig hebben, kan je terecht bij de PHP manual, die te vinden is op <http://www.php.net/manual/en/index.php>

### 1.3 Database creatie

Voor het ontwikkelen van de front-end toepassing zullen we opnieuw gebruikmaken van XAMPP. De creatie en het laden van de database gebeurt op dezelfde manier als in oefenzitting 3, met dat verschil dat we nu 2 afzonderlijke bestanden gebruiken. Eén bestand, `database_schema.sql`, bevat het schema van de database (i.e. de specificatie van de tabellen en kolommen uit het relationeel model in SQL), en het andere bestand, `database_data.sql` (of `database_data_DEVEL.sql.zip`), bevat de data. Gebruik dezelfde stappen als in oefenzitting 3 om eerst het schema en vervolgens de data te laden. Bemerkt echter dat je de database nu niet meer expliciet moet aanmaken, maar dit automatisch gebeurt, tijdens het creëren van het schema.

**\*\*Opmerking:** de grootte van bestanden die geupload worden in PHPMyAdmin is in XAMPP standaard gelimiteerd op 2 MB. Grotere bestanden inladen kan door de variabele `upload_max_filesize` in het bestand '`php.ini`' onder `[XAMPP install dir]/php/` aan te passen (naar bv. 128 MB). Merk op dat het inladen van grotere bestanden heel wat tijd kan vragen. Daarom raden we aan om tijdens de oefenzitting te werken met de DEVEL dataset. Alternatief kan je grotere bestanden ook inladen met het `LOAD DATA INFILE` statement in SQL (maar hebben we hier vermeden door de wijzigende bestandspaden op de computers in de PC-klassen).

### 1.4 Connecteren met een gegevensbank

Een gegevensbank kan aangesproken worden vanuit PHP. Hier voor moet je eerst een connectie maken op de gegevensbank. Dit kan je doen met de methode `mysql_connect`.

```
<?php
    //the credentials
    $host = 'mysql.oracle.com';
    $user = 'johndoe';
    $pass = '1234isaverybadpassword';

    //the database name to connect to
    $database = 'muziek';

    //connect to the database
    $conn = mysql_connect($host, $user, $pass) or die("Cannot connect to the
database");
    mysql_select_db($database);
?>
```

De `or die` clause zal een exception werpen als de connectie gefaald is. Eenmaal de connectie opgezet is, is het belangrijk om de naam van de gegevensbank aan te geven. Dit kan met `mysql_select_db`.

Je kan de gegevensbank connectie afsluiten wanneer je de gegevensbank niet meer nodig hebt door de instructie `mysql_close($conn)` uit te voeren.

```
<?php
    //make connection to the database, see example above
    ...

    //make the query
    $query = "SELECT title as titel FROM SONGS;";

    //query the database
```

```

$result = mysql_query($query) or die("Database error: " . mysql_error());

while( $entry = mysql_fetch_array($result, MYSQL_ASSOC) )
{
    echo $entry['titel'];
}
?>

```

In bovenstaand voorbeeld wordt een query uitgevoerd. Dit geeft een resultaat terug, waar we een associatieve tabel op basis van de kolomnamen (gebruik makend van de aliasnamen die je meegeeft in je query) aan kunnen opvragen, mbv. `mysql_fetch_array($result, MYSQL_ASSOC)`. Hierover kan men dan itereren. Je kan in deze methode verschillende parameters meegeven: `MYSQL_ASSOC` geeft een associatieve tabel terug, `MYSQL_NUM` geeft een genummerde lijst terug en `MYSQL_BOTH` geeft een tabel terug die zowel genummerd als associatief is.

### 1.5 Een web-based front-end voor de database

Nu je een basiskennis hebt van PHP en HTML, kan je van start gaan met het ontwikkelen van een web-based front-end voor de database. Download daarvoor de starter code van Toledo, decomprimeer en plaats de map 'luchtvaart' onder 'C:\Workdir\xampp\_183\htdocs\', indien je op een PC in de PC-klas werkt, of ga op zoek naar de htdocs map in je XAMPP installatiemap. De indexpagina kan je nu in een browser bekijken via 'localhost/luchtvaart'.

#### Oefening 1: Experimenteren met PHP

Als opwarmertje gaan we de PHP API aanspreken en de resultaten afprinten en formatteren in een HTML-pagina. Deze opgave los je op in `index.php`. Geef elke keer de pagina herlaadt, een willekeurig getal terug tussen 1 en 20 en geef dit weer in de HTML-pagina. De functie om willekeurige getallen te genereren in PHP is `rand()`, zie ook <http://be.php.net/rand>.

PHP laat ook verschillende wiskundige operaties toe. Zoek ook uit hoe je de datum van vandaag kan weergeven, en bereken op basis hiervan de datum van de dag een week voor en een week na vandaag. Geef ook dit weer op de voorpagina.

#### Oefening 2: Connectie maken met de gegevensbank

Wanneer je in de index naar "Lijst van klanten" gaat, dan zie je dat er een gegevensbankfout is. We gaan nu de gegevensbankconnectie initialiseren. Hiervoor voegen we code toe aan het bestand `database.inc`. Je slaat de gegevensbankconnectie op in een variabele `$conn`. Dit is noodzakelijk omdat deze variabele in andere bestanden gebruikt wordt. Vergeet niet het juiste gegevensbankschema te selecteren mbv. `mysql_select_db`. Nu zou de lijst van klanten ter beschikking moeten zijn. In het `klanten.php`-bestand kan je bekijken hoe dit gedaan wordt.

Genereer nu ook een lijst van luchtvaartmaatschappijen in `luchtvaartmaatschappijen.php`, maar zorg dat je enkel de luchtvaartmaatschappijen met de Verenigde States (United States) als thuisbasis toont, en orden ze op naam en vervolgens op call sign.

#### Oefening 3: Zoeken op aantal stops, vertrektijd, luchthaven van oorsprong

Bekijk "Vluchtzoeker" in de index. Zoeken op aantal stops is al geïmplementeerd, te bekijken in `zoekopaantalstops.php`. Test dit uit door alle vluchten met 1 stop te zoeken.

Voeg nu ook een functie toe om te zoeken op de vertrektijd (een bepaalde dag), en geef als resultaat het vluchtnummer van de vlucht terug. Doe dit in `zoekopvertrektijd.php`. Het toevoegen van het invoerveld en de knop kan in `zoeken.php`. Hoe worden de ingevoerde data behandeld? Hoe dien je de query in SQL te formuleren om alle vluchten die vertrekken op een bepaalde dag te krijgen (meerdere oplossingen mogelijk)?

Vervolgens zou het ook handig zijn om te kunnen zoeken op de luchthaven van oorsprong. Geef als resultaat het vluchtnummer en de luchthaven van bestemming. Implementeer dit in `zoekopoorsprong.php`. Hoe dien je deze query te formuleren?

PHP voorziet ook de mogelijkheid om methodes uit te voeren op teruggekregen resultaten. Voeg bij de resultatenlijst na het zoeken op luchthaven van oorsprong ook de vluchttijd toe (in aantal dagen). Hiervoor zijn verschillende mogelijke oplossingen. **Tip:** <http://www.php.net/manual/en/datetime.diff.php>

#### *Oefening 4: Verander het adres van een klant*

Bekijk "Stel een nieuw adres in" in de index. Het is de bedoeling dat je de functionaliteit om het adres van een klant aan te passen programmeert. Bekijk `nieuwadres.php`; daarna kan je de functionaliteit toevoegen aan `nieuwadres_uitvoer.php`.

**Tip:** een nuttige PHP functie voor het splitsen van een string is 'explode' (<http://php.net/manual/en/function.explode.php>).

**Tip:** een nuttige PHP functionaliteit is `$_SESSION`, een associatieve lijst waarin je variabelen kan opslaan die beschikbaar zijn in de huidige sessie. Het doorgeven van variabelen tussen twee scripts kan dus door middel van:

```
$_SESSION["variabelenaam"] = $variableToStore;
```

en het opvragen via

```
$_SESSION["variabelenaam"];
```

#### *Oefening 5: Boeken van een vlucht*

Bekijk "Boek vlucht" in de index. Het is de bedoeling dat een werknemer van een reisbureau hier eerst een lijst van reisbureaus te zien krijgt, waaruit hij zijn/haar reisbureau kan selecteren. Implementeer dit in `boeking_selecteerreisbureau.php`.

Op de volgende pagina (`boeking_selecteerklant.php`), moet de werknemer een klant kunnen kiezen. Voor deze klant moet nu een vlucht geboekt worden (`boeking_selecteervlucht.php`). Hiervoor moet de werknemer een lijst te zien krijgen van alle vluchten (met vermelding van luchthavens van vertrek en luchthaven van aankomst) waarop nog vrije (nog niet geboekte) zitplaatsen beschikbaar zijn (samen met hun klasse en het ID van de luchtvaartmaatschappij).

Als ook de vrije zitplaats (met een klasse en bij een bepaalde maatschappij) geselecteerd is, moet de boeking in de database opgenomen worden. Doe dit in `'boeking_uitvoer.php'`. Toon de gebruiker hierna de prijs van de geboekte vlucht (de som van de taksen op luchthaven van vertrek en aankomst en de prijs van de klasse).

## **2. Bronnen**

- Michel Vuijlsteke's Blog, Tales of Drudgery and Boredom, in het artikel: "Iets over HTML...", <http://blog.zog.org/2008/04/iets-over-html.html>
- W3 Schools, HTML tutorial, <http://w3schools.com/html/default.asp>
- W3 Schools, XHTML tutorial, <http://w3schools.com/xhtml/default.asp>
- PHP manual, <http://www.php.net/manual/en/index.php>