



Web advanced

PDO

DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Dep. PXL-IT – Elfde-Liniestraat 26 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook



Software

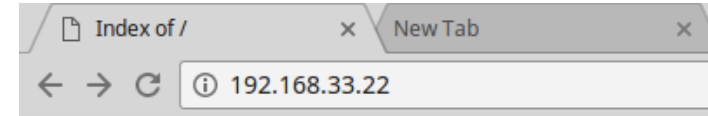
Vagrant & Virtual box

`vagrant up`

Test




`http://192.168.33.22/`

`http://192.168.33.22/phpmyadmin`



Index of /

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
-------------	----------------------	-------------	--------------------

 database.php	2017-09-19 18:34	553	
 fout.php	2017-09-19 11:55	21	
 phpinfo.php	2017-09-16 18:29	17	

Apache/2.4.18 (Ubuntu) Server at 192.168.33.22 Port 80

phpmyadmin



Welcome to phpMyAdmin

Language

English

Log in ?

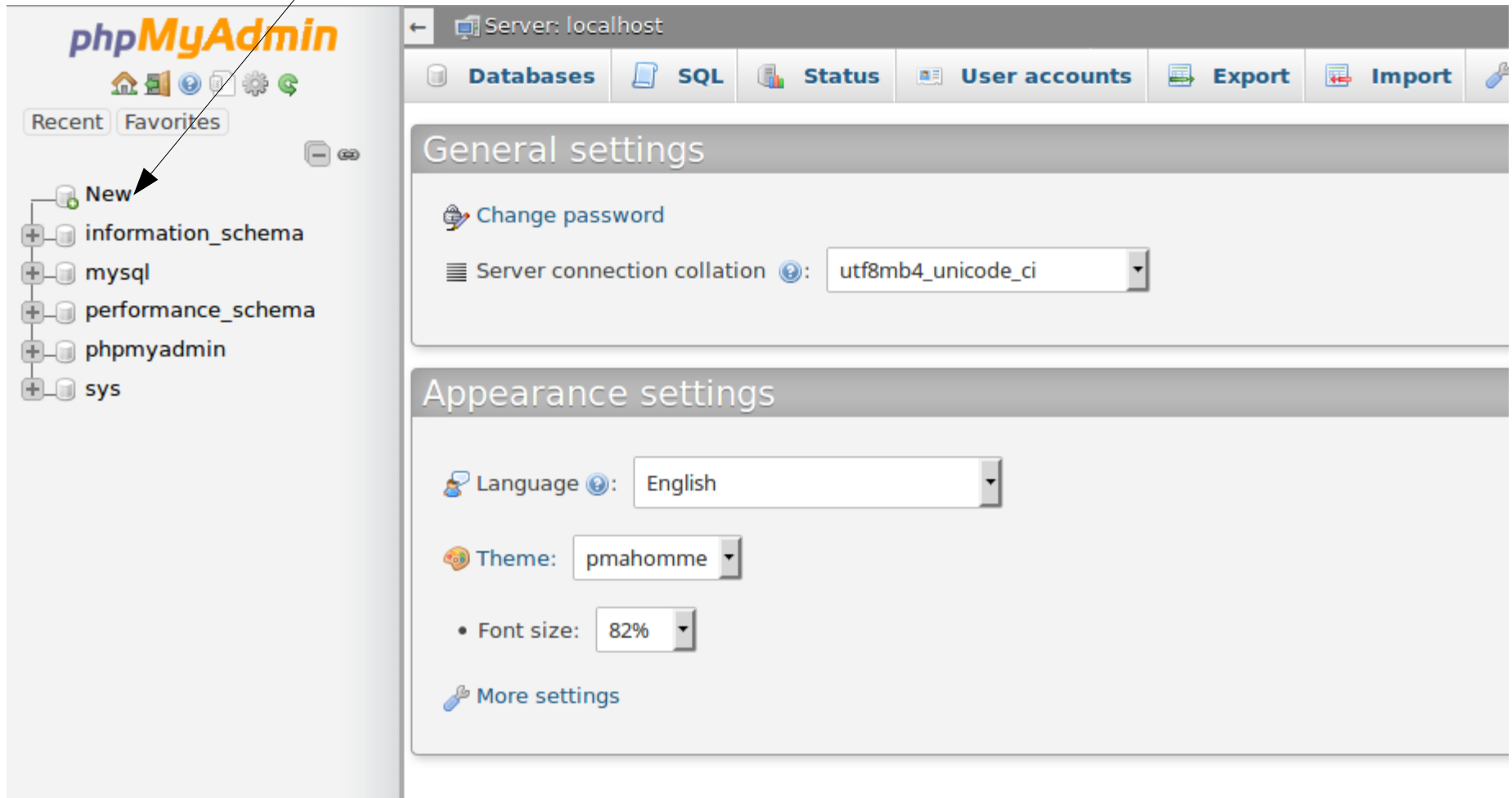
Username:

Password:

Go

root
root

phpmyadmin



The screenshot displays the phpMyAdmin web interface. On the left sidebar, the 'New' button is highlighted with a black arrow. The main content area is divided into two sections: 'General settings' and 'Appearance settings'.

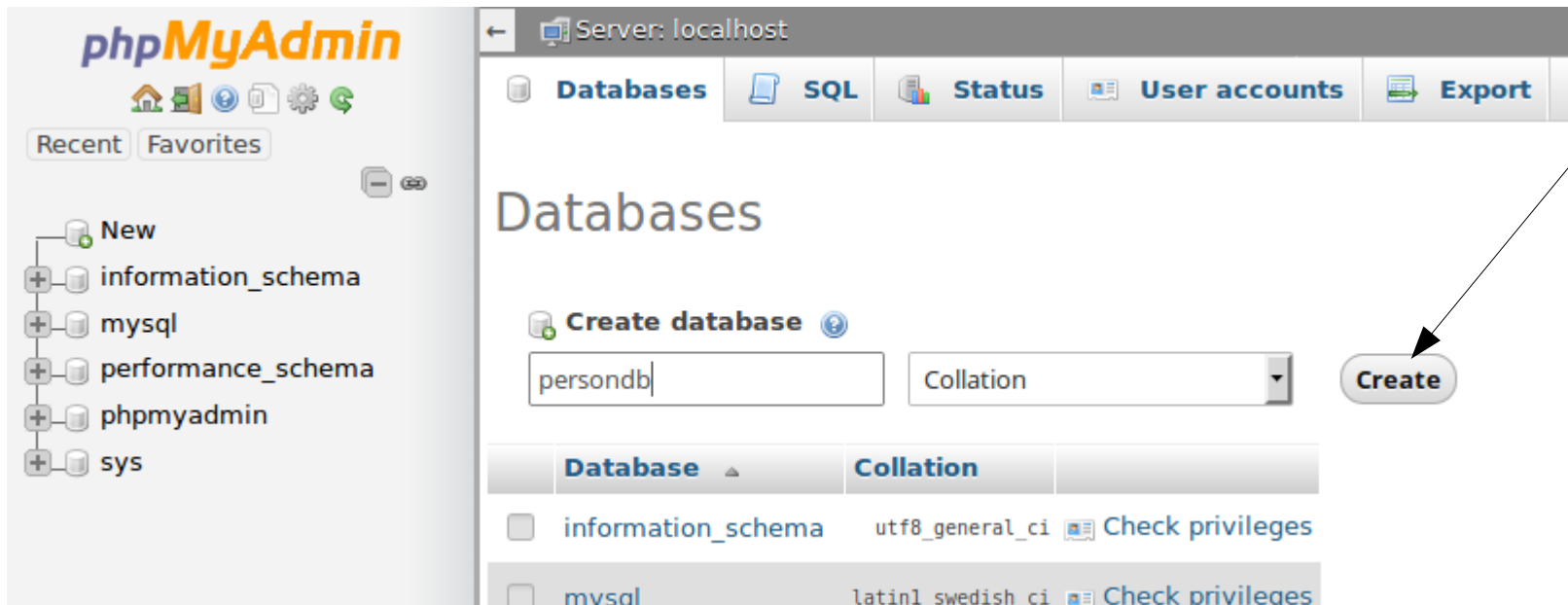
General settings

- [Change password](#)
- Server connection collation:

Appearance settings

- Language:
- Theme:
- Font size:
- [More settings](#)

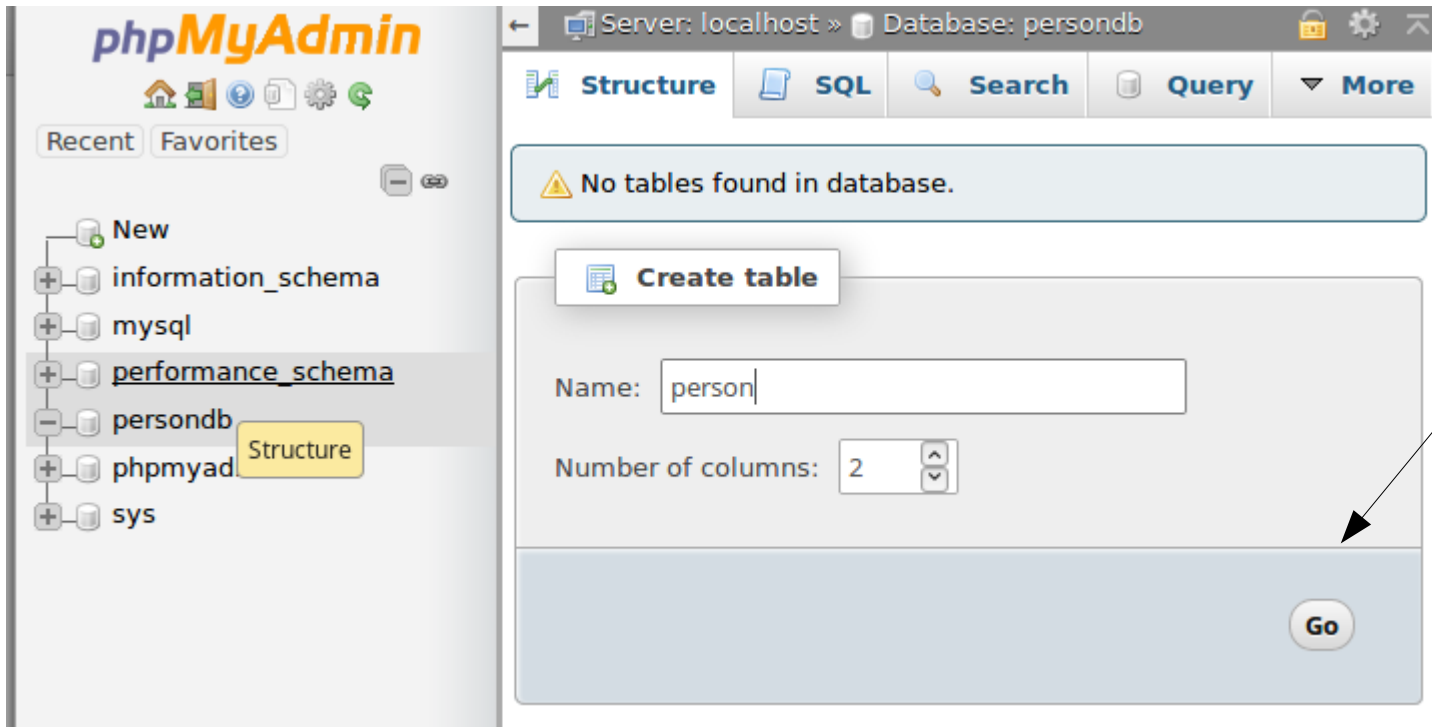
phpmyadmin



The screenshot shows the phpMyAdmin interface for a local MySQL server. The left sidebar displays a tree view of databases, including 'information_schema', 'mysql', 'performance_schema', 'phpmyadmin', and 'sys'. The main panel is titled 'Databases' and contains a 'Create database' form. The form has a text input field with 'persondb' and a dropdown menu for 'Collation'. A 'Create' button is located to the right of the form, with an arrow pointing to it. Below the form is a table listing existing databases and their collations.

Database	Collation	
<input type="checkbox"/> information_schema	utf8_general_ci	Check privileges
<input type="checkbox"/> mysql	latin1_swedish_ci	Check privileges

phpmyadmin



phpmyadmin

Server: localhost » Database: persondb » Table: person

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Privileges](#) [Operations](#) [Tracking](#) [Triggers](#)

Table name: Add column(s)

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Comments	Virtuality	MIME type	Browser display transformation
<input type="text" value="id"/> <small>Pick from Central Columns</small>	<input type="text" value="INT"/>	<input type="text"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="PRIMARY"/> <small>PRIMARY</small>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="name"/> <small>Pick from Central Columns</small>	<input type="text" value="VARCHAR"/>	<input type="text" value="25"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="---"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Table comments:

Collation:

Storage Engine:

PARTITION definition:

phpmyadmin

Server: localhost » Database: persondb » Table: person

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#)

[Table structure](#) [Relation view](#)

	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1	id	int(11)			No	None		Change Drop Key
<input type="checkbox"/>	2	name	varchar(25)	latin1_swedish_ci		No	None		Change Drop Key

[↑](#) ☐ Check all With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#)

[Print view](#) [Propose table structure](#) [Track table](#) [Move columns](#) [Improve table structure](#)

[Add](#) column(s) [Go](#)

[+ Indexes](#)

Information

Space usage	
Data	16 KiB
Index	0 B
Total	16 KiB

Row statistics	
Format	dynamic
Collation	latin1_swedish_ci
Creation	Oct 02, 2017 at 10:40 AM

← Server: localhost » Database: persondb » Table: person

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Column	Type	Function	Null	Value
id	int(11)	<div></div>		<div>1</div>
name	varchar(25)	<div></div>		<div>jan</div>

Go

☐ Ignore

Column	Type	Function	Null	Value
id	int(11)	<div></div>		<div>2</div>
name	varchar(25)	<div></div>		<div>sofie</div>

Go

Insert as new row

and then

Go back to previous page

?

Go

Preview SQL

Reset

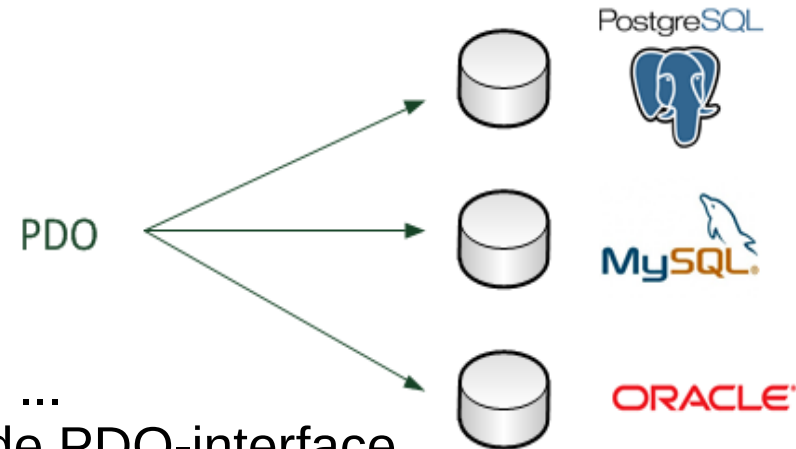


PDO



- PHP Data Objects
- PHP extension sinds PHP5 (2004)
(geschreven in C++)
- Data access abstraction layer:

dezelfde code voor de interactie
met PostgreSQL, MySQL, Oracle, ...
(op voorwaarde dat de driver die de PDO-interface
implementeert geïnstalleerd is)



alle configuratie in de [connection-string](#)

```
$pdo=new PDO ("mysql:host=localhost;dbname=...",  
              $user,$pass) ;
```

alternatief = aparte methodes voor elke DB

- | | | | |
|-------------------|-----------------|---------------|-----|
| - pg_connect, | pg_execute, | pg_fetch_all, | ... |
| - mysqli_connect, | mysqli_execute, | mysqli_fetch | ... |
| - oci_connect, | oci_execute, | oci_fetch | ... |

Algemene code

```
1 <?php
2 $user='root';
3 $password='root';
4 $database='persondb';
5 $pdo=null;
6 try {
7     $pdo = new PDO( "mysql:host=localhost;dbname=$database",
8         $user, $password );
9     $pdo->setAttribute( PDO::ATTR_ERRMODE,
10         PDO::ERRMODE_EXCEPTION );
11
12     /* ... */ exec, query zie volgende slides
13
14 } catch ( PDOException $e ) {
15     print 'Exception!: ' . $e->getMessage();
16 }
17 $pdo = null;
18
19
```

exec: gegevens wijzigen

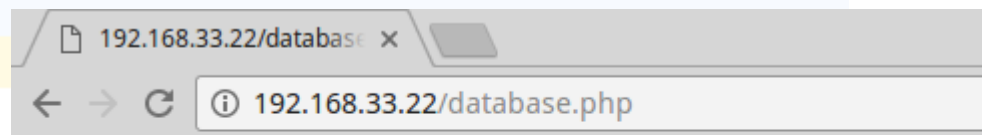
```
1  <?php
2  $user='root';
3  $password='root';
4  $database='persondb';
5  $pdo=null;
6  try {
7      $pdo = new PDO( "mysql:host=localhost;dbname=$database",
8                      $user, $password );
9      $pdo->setAttribute( PDO::ATTR_ERRMODE,
10                          PDO::ERRMODE_EXCEPTION );
11      $numberOfRows = $pdo->exec("DELETE FROM person WHERE ".
12                                "name LIKE 's%'");
13      print("$numberOfRows rows modified");
14
15  } catch ( PDOException $e ) {
16      print 'Exception!: ' . $e->getMessage();
17  }
18  $pdo = null;
19
```

← → ↻ ⓘ 192.168.33.22/database.php

1 rows modified

query: gegevens opvragen

```
1 <?php
2 $user='root';
3 $password='root';
4 $database='persondb';
5 $pdo=null;
6 try {
7     $pdo = new PDO( "mysql:host=localhost;dbname=$database",
8                     $user, $password );
9     $pdo->setAttribute( PDO::ATTR_ERRMODE,
10                        PDO::ERRMODE_EXCEPTION );
11     $statement = $pdo->query('SELECT * from person');
12     $statement->setFetchMode(PDO::FETCH_ASSOC);
13     while($row = $statement->fetch()) {
14         print_r($row);
15     }
16 } catch ( PDOException $e ) {
17     print 'Exception!: ' . $e->getMessage();
18 }
19 $pdo = null;
20
```



Array ([id] => 1 [name] => jan) Array ([id] => 2 [name] => sofie)

PDO (1. Query)

```
$statement = $pdo->query('SELECT * from person');
```

resultset, **cursor** duidt 1 rij aan



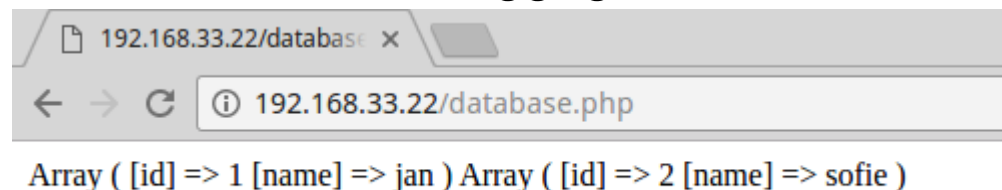
id	name
1	jan
2	sofie

```
while($row = $statement->fetch()) {  
    print_r($row);  
}
```

fetch schuift de cursor 1 positie op.

- de geselecteerde rij wordt teruggegeven
- voorbij de laatste rij return-value = false

fetchAll: alle gegevens uit het resultset worden teruggegeven als 2D array



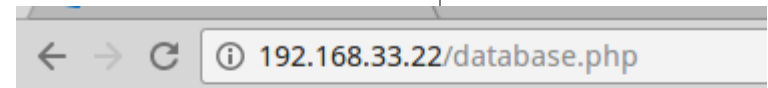
PDO (1. Query)

```
$statement->setFetchMode(PDO::FETCH_ASSOC);
```

associative array: `$row['id'], $row['name']`

id	name
1	jan
2	sofie

```
print("<table>");
while($row = $statement->fetch()) {
    print('<tr><td>'.$row['id'].'</td><td>'
        . $row['name'].'</td></tr>');
}
print("</table>");
```



1 jan
2 sofie

metadata

rowCount

aantal rijen in resultset

columnCount

aantal kolommen in resultset

getColumnMeta

metadata over kolom

```
$statement = $pdo->query('SELECT * from person');
$statement->setFetchMode(PDO::FETCH_ASSOC);

if($statement->rowCount() > 0){
    $columnNames=[];
    for ($i = 0; $i < $statement->columnCount(); $i++) {
        $columnData = $statement->getColumnMeta($i);
        $columnName = $columnData['name'];
        $columnNames[] = $columnName;
    }

    print("<table>");
    print('<tr><th>'.implode('</th><th>',$columnNames).
        '</th></tr>');

    while($row = $statement->fetch()) {
        print('<tr><td>'.implode('</td><td>',$row).
            '</td></tr>');
    }
    print("</table>");
```

← → ↻ ⓘ 192.168.33.22/database.php

id name

1 jan

2 sofie

PDO (2. Prepared statements)



Prepared statement

- via de methode **prepare** wordt de prepared statement doorgestuurd naar de databank (precompiled)
- via de methode **execute** wordt de prepared statement uitgevoerd bij de executie kunnen parameters meegegeven worden
- named en unnamed parameters
- voordelen van prepared statements:
 - efficiëntie wanneer query meerdere keren uitgevoerd moet worden
 - parameters worden geëscaped (**sql-injection**)

PDO (2. Prepared statements)



Unnamed parameters

- **prepare** maakt de prepared statement klaar
- binnen de query worden parameters aangeduid als **?**
- via **bindParam** wordt het eerste **?** verbonden met \$titel, ...
- **execute** voert de preparedstatement uit

```
$statement = $pdo->prepare("INSERT INTO cds ".  
    "(titel, interpret, year) VALUES (?, ?, ?);");  
$statement->bindParam(1, $titel, PDO::PARAM_STR);  
$statement->bindParam(2, $interpret, PDO::PARAM_STR);  
$statement->bindParam(3, $year, PDO::PARAM_INT);  
$nr=$statement->execute();  
print("$nr rows modified");
```

PDO (2. Prepared statements)



Named parameters

- **prepare** maakt de prepared statement klaar
- binnen de query worden parameters aangeduid als **:naam**
- via **bindParam** wordt de parameter :id verbonden met een variabele
- **execute** voert de preparedstatement uit

```
$id = 1;  
$statement=$pdo->prepare('SELECT * FROM cds WHERE id = :id');  
$statement->bindParam(':id', $id, PDO::PARAM_INT);  
$statement->setFetchMode(PDO::FETCH_ASSOC);  
$statement->execute();  
var_dump($statement->fetch());
```

SQL injection



SQL injection



invoer.html

user:

Password:



id en password worden
rechtstreeks in
commando geplaatst

verwerk.php

```
SELECT * FROM users WHERE id = 1 # AND password = MD5('')
```

Ingelogd als 1 jan

SQL injection



invoer.html

user:

Password:

OK



id en password worden
rechtstreeks in
commando geplaatst

verwerk.php

```
SELECT * FROM users WHERE id = 1 OR 1 = 1 AND password = MD5('')
```

true

false

false

Ingelogd als 1 jan

voor id=1: true OR false=true

PDO (2. SQL-injection)






Bekijk

- Injection, Cross-site-scripting (XSS), Cross-site Request Forgery (CSRF)

op

<https://app.pluralsight.com/library/courses/web-security-owasp-top10-big-picture/table-of-contents>

Expand all		
▶ Introduction	🔖	7m 47s ▼
▶ Injection 	🔖	14m 21s ▼
▶ Broken Authentication and Session Management	🔖	14m 19s ▼
▶ Cross-Site Scripting (XSS) 	🔖	12m 29s ▼
▶ Insecure Direct Object References	🔖	11m 16s ▼
▶ Security Misconfiguration	🔖	9m 46s ▼
▶ Sensitive Data Exposure	🔖	12m 9s ▼
▶ Missing Function Level Access Control	🔖	11m 44s ▼
▶ Cross-Site Request Forgery (CSRF) 	🔖	11m 34s ▼
▶ Using Components with Known Vulnerabilities	🔖	9m 8s ▼
▶ Unvalidated Redirects and Forwards	🔖	9m 5s ▼