Internet Applications

Maciej Zakrzewicz

Institute of Computing Science, Poznan University of Technology, http://zakrzewicz.pl

Database functions for PHP

- SQL-based functions
  - MySQLi
    - MySQL only, procedural or object-oriented
  - PDO
    - universal, object-oriented
  - etc.
- Object-Relational Mapping services (PHP Frameworks)
  - Doctrine
  - Propel
  - Xyster
  - RedBean
  - etc.

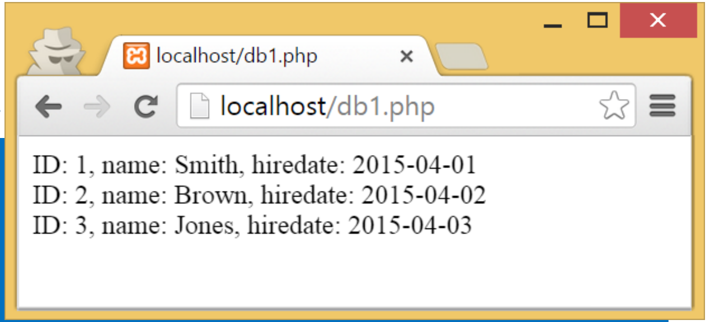
Reading:

„PHP MySQL Database”

http://www.w3schools.com/php/php_mysql_intro.asp

MySQLi Example (Select)

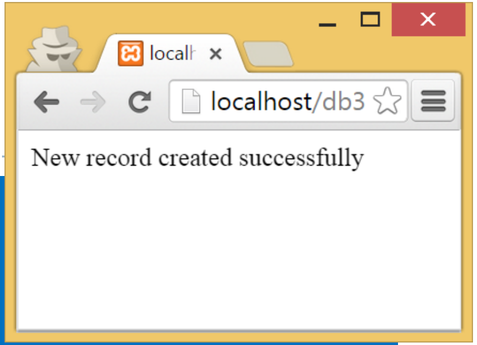
```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {die("Connection failed: " . $conn->connect_error);}
$sql = "SELECT id, name, hiredate, salary FROM emp";
$result = $conn->query($sql);

while($row = $result->fetch_assoc()) {
  echo "ID: " . $row["id"]. ", name: " . $row["name"]. ", hiredate: " . $row["hiredate"]. "<br>";
}
$conn->close();
?>
```



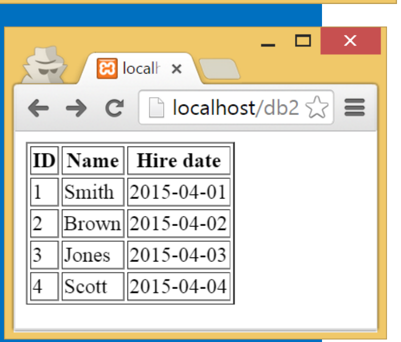
```
ID: 1, name: Smith, hiredate: 2015-04-01
ID: 2, name: Brown, hiredate: 2015-04-02
ID: 3, name: Jones, hiredate: 2015-04-03
```

PDO Example (non-Select)

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test";
try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "INSERT INTO emp (id, name, hiredate, salary)
            VALUES (4, 'Scott', '2015-04-04', 10000)";
    $conn->exec($sql);
    echo "New record created successfully";
    }
catch(PDOException $e)
    {echo $sql . "<br>" . $e->getMessage();}
$conn = null; ?>
```

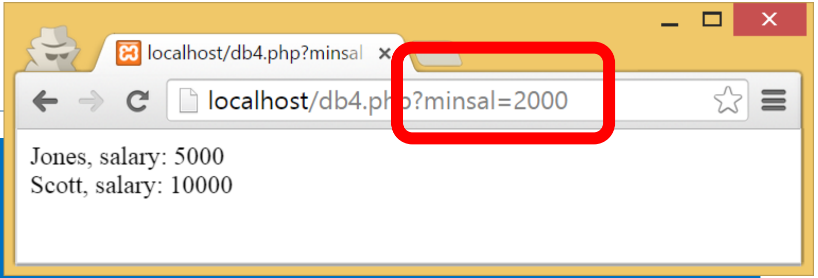


New record created successfully



ID	Name	Hire date
1	Smith	2015-04-01
2	Brown	2015-04-02
3	Jones	2015-04-03
4	Scott	2015-04-04

SQL Parameters



```
Jones, salary: 5000
Scott, salary: 10000
```

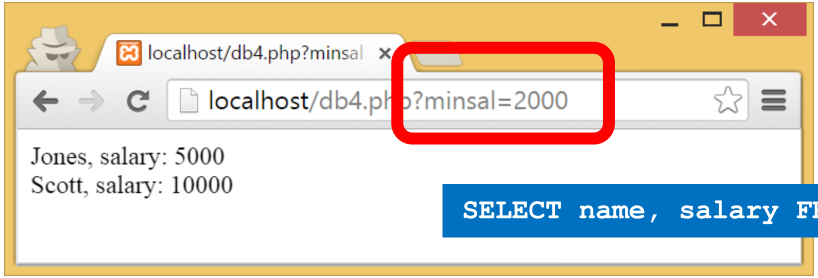
```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {die("Connection failed: " . $conn->connect_error);}
$sql = "SELECT name, salary FROM emp WHERE salary>".$_GET['minsal'];
$result = $conn->query($sql);

while($row = $result->fetch_assoc()) {
  echo $row["name"]. ", salary: " . $row["salary"]. "<br>";
}
$conn->close();
?>
```

Threat: SQL Injection

Reading:

„SQL Injection”

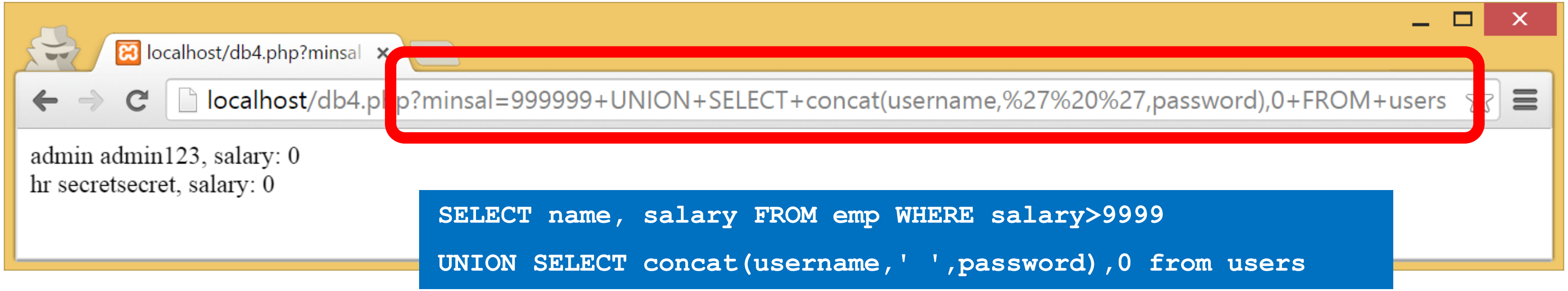
http://www.w3schools.com/sql/sql_injection.asp



```
localhost/db4.php?minsal=2000
```

```
Jones, salary: 5000
Scott, salary: 10000
```

```
SELECT name, salary FROM emp WHERE salary>2000
```



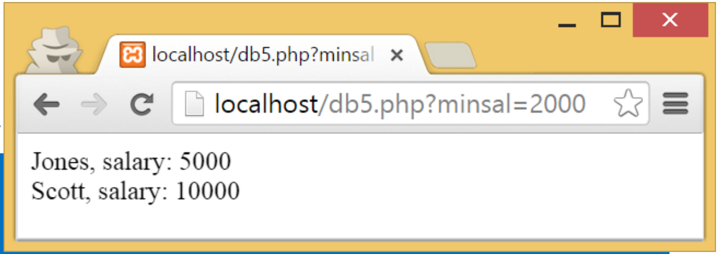
```
localhost/db4.php?minsal=999999+UNION+SELECT+concat(username,%27%20%27,password),0+FROM+users
```

```
admin admin123, salary: 0
hr secretsecret, salary: 0
```

```
SELECT name, salary FROM emp WHERE salary>9999
UNION SELECT concat(username,' ',password),0 from users
```

!

SQL Bound Parameters



```
Jones, salary: 5000
Scott, salary: 10000
```

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {die("Connection failed: " . $conn->connect_error);}
$sql = "SELECT name, salary FROM emp WHERE salary>?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("d",$_GET['minsal']);
$stmt->execute();
$stmt->bind_result($name, $salary);
while($row = $stmt->fetch()) {
  echo $name. ", salary: " . $salary.
$conn->close();?>
```

Reading:

„Prepared Statements”

http://www.w3schools.com/php/php_mysql_prepared_statements.asp

Why Bound Parameters?

- Database server will compile the SQL query only once
  - instead of once for each value combination of the parameters
  - improved database server performance
- When running a query multiple times, only parameter values are transmitted, not the whole query
  - improved network performance
- Bound parameters cannot be used for SQL syntactical elements (keywords, table names, etc.)
  - protection against SQL Injection attacks

Connection Pooling

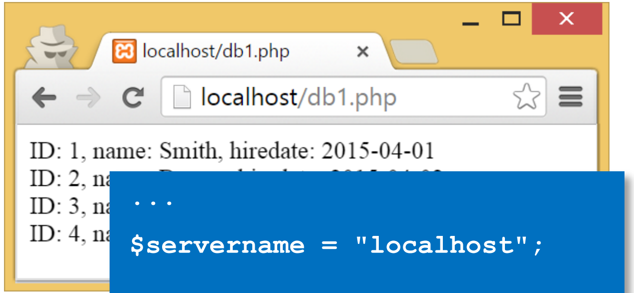
- A database connection between a PHP page and a database server can be reused by a PHP page, rather than being created and destroyed multiple times
  - overhead of creating fresh connections
  - improved database server performance

Reading:

„Connections"

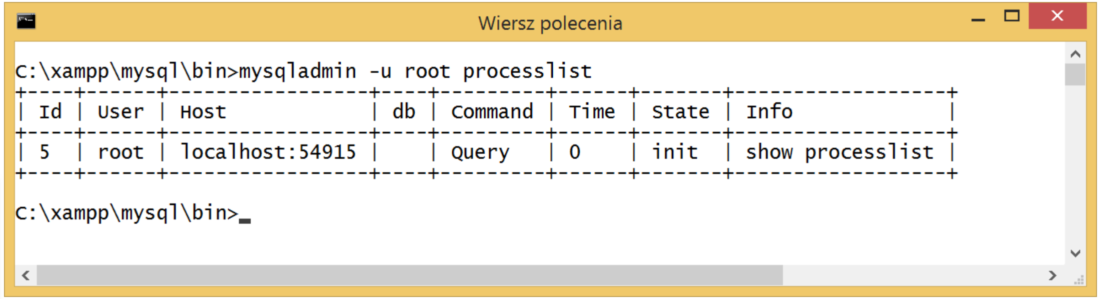
http://php.net/manual/en/mysqli.quickstart.connections.php

Connection Pooling in Action

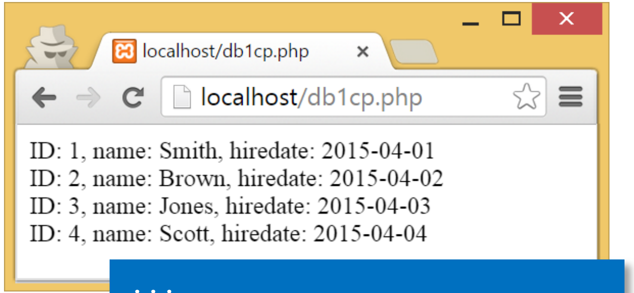


ID: 1, name: Smith, hiredate: 2015-04-01
ID: 2, na
ID: 3, na
ID: 4, na

```
...
$servername = "localhost";
...
```

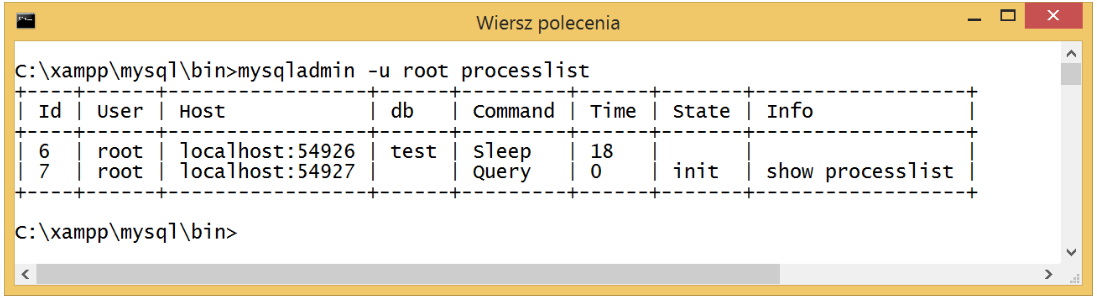


```
C:\xampp\mysql\bin>mysqladmin -u root processlist
+----+------+-----------------+----+---------+------+-------+------------------+
| Id | User | Host            | db | Command | Time | State | Info             |
+----+------+-----------------+----+---------+------+-------+------------------+
| 5  | root | localhost:54915 |    | Query   | 0    | init  | show processlist |
+----+------+-----------------+----+---------+------+-------+------------------+

C:\xampp\mysql\bin>_
```



ID: 1, name: Smith, hiredate: 2015-04-01
ID: 2, name: Brown, hiredate: 2015-04-02
ID: 3, name: Jones, hiredate: 2015-04-03
ID: 4, name: Scott, hiredate: 2015-04-04

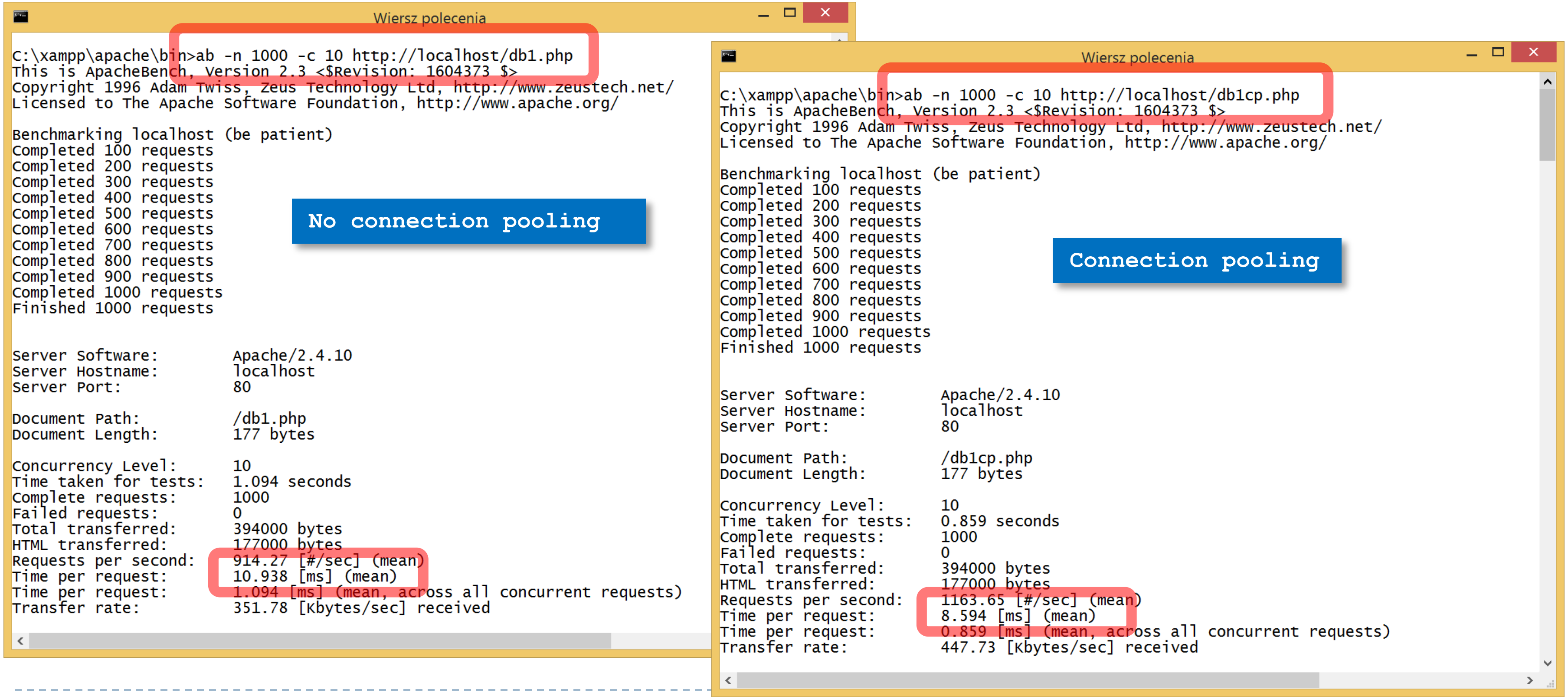
```
...
$servername = "p:localhost";
...
```



```
C:\xampp\mysql\bin>mysqladmin -u root processlist
+----+------+-----------------+------+---------+------+-------+------------------+
| Id | User | Host            | db   | Command | Time | State | Info             |
+----+------+-----------------+------+---------+------+-------+------------------+
| 6  | root | localhost:54926 | test | Sleep   | 18   |       |                  |
| 7  | root | localhost:54927 |      | Query   | 0    | init  | show processlist |
+----+------+-----------------+------+---------+------+-------+------------------+

C:\xampp\mysql\bin>
```

Detailed configuration in php.ini

Connection Pooling in Action



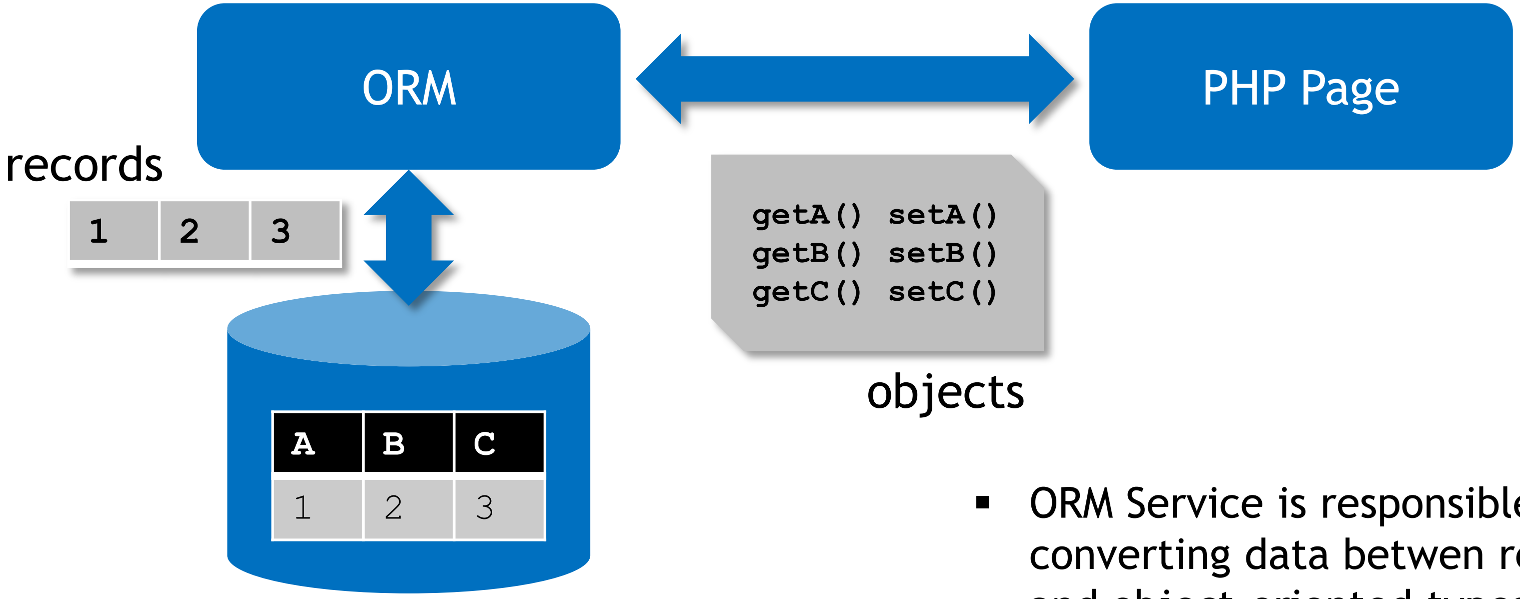
```
C:\xampp\apache\bin>ab -n 1000 -c 10 http://localhost/db1.php
This is ApacheBench, Version 2.3 <$Revision: 1604373 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests


Server Software:        Apache/2.4.10
Server Hostname:        localhost
Server Port:            80

Document Path:          /db1.php
Document Length:        177 bytes

Concurrency Level:      10
Time taken for tests:   1.094 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      394000 bytes
HTML transferred:       177000 bytes
Requests per second:    914.27 [#/sec] (mean)
Time per request:       10.938 [ms] (mean)
Time per request:       1.094 [ms] (mean, across all concurrent requests)
Transfer rate:          351.78 [Kbytes/sec] received
```

No connection pooling

```
C:\xampp\apache\bin>ab -n 1000 -c 10 http://localhost/db1cp.php
This is ApacheBench, Version 2.3 <$Revision: 1604373 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests


Server Software:        Apache/2.4.10
Server Hostname:        localhost
Server Port:            80

Document Path:          /db1cp.php
Document Length:        177 bytes

Concurrency Level:      10
Time taken for tests:   0.859 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      394000 bytes
HTML transferred:       177000 bytes
Requests per second:    1163.65 [#/sec] (mean)
Time per request:       8.594 [ms] (mean)
Time per request:       0.859 [ms] (mean, across all concurrent requests)
Transfer rate:          447.73 [Kbytes/sec] received
```

Connection pooling

Object-Relational Mapping



ORM

PHP Page

records

1 2 3

```
getA()  setA()
getB()  setB()
getC()  setC()
```

objects

A	B	C
1	2	3

- ORM Service is responsible for converting data betwen relational and object-oriented types
- ORM Service replaces SQL with object method calls

ORM Example (Doctrine)

```
<?php
require_once "bootstrap.php";

$empRepository = $entityManager->getRepository('emp');
$emps = $empRepository->findAll();
foreach ($emps as $emp) {
    echo "ID: ".$emp->getId().", name: ".$emp->getName();."<br>";} ?>
```

Select all records from EMP table

```
<?php
require_once "bootstrap.php";
$newEmpId = 6;
$newEmpName = 'Johnson';
$emp = new Emp();
$emp->setId($newEmpId);
$emp->setName($newEmpName);
$entityManager->persist($emp);
$entityManager->flush();?>
```

Insert a new record into EMP table

Reading:

„Getting Started with Doctrine”

http://doctrine-orm.readthedocs.org/en/latest/tutorials/getting-started.html