DBMS Functions

Disclaimer: вы смотрите просто запись лекции, это **НЕ** специально подготовленный видеокурс!

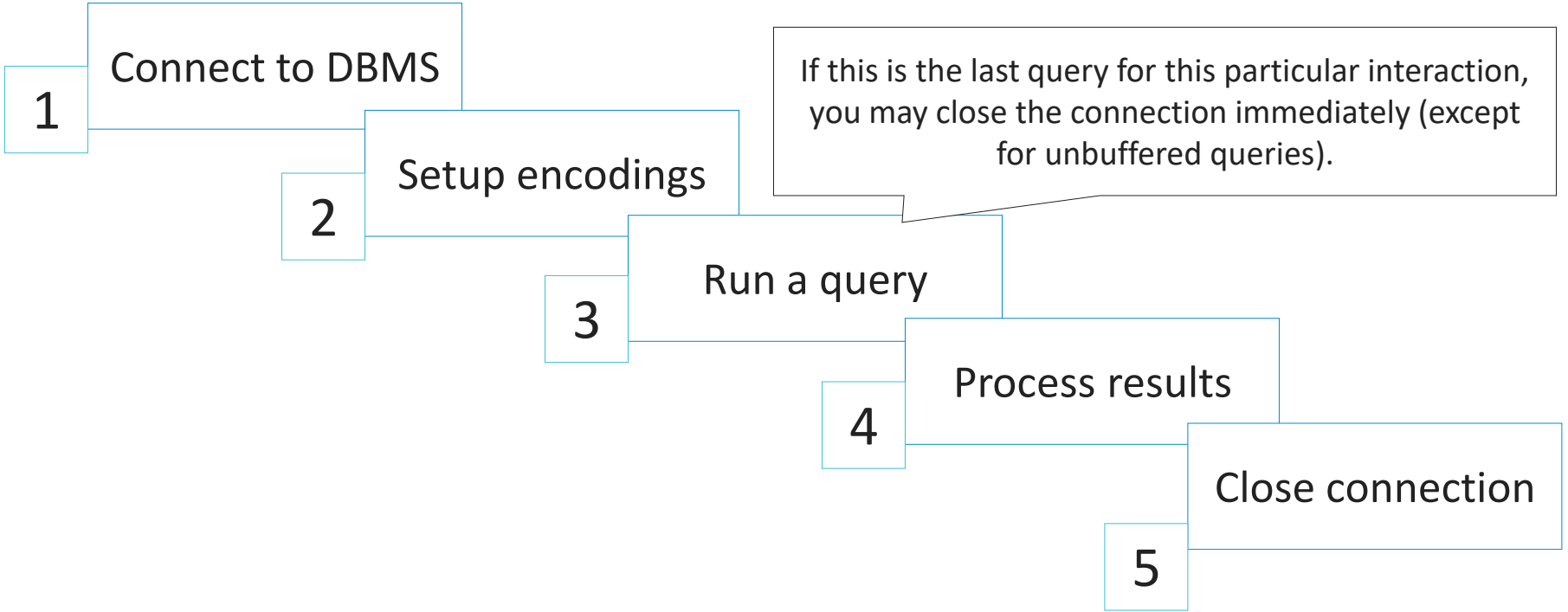
Intro

As majority of applications use DBMS to store and manipulate data, we need to understand at least basics of interaction with DBMSes with PHP.

Full list of functions: https://www.php.net/manual/en/refs.database.php

Warning! When dealing with DBMSes mind encodings, permissions, exceptions and so on!

General algorithm



Most common DBMS extensions in PHP

As we'll use MySQL, for this DBMS there are two powerful PHP extensions:

1. MySQL Improved (mysqli): http://www.php.net/manual/en/book.mysqli.php
2. PHP Data Objects (PDO): http://www.php.net/manual/en/book.pdo.php

If you are not familiar with relational databases and/or SQL, please refer to these online courses:

1. "Relational Databases Basics": https://svyatoslav.biz/urls/rdb-eng
2. "SQL by Examples": https://svyatoslav.biz/urls/sql-eng

Simple **mysqli** samples, establishing connection

To connect to a DBMS with **mysqli** extension we may use the following approach:

```php
<?php

$mysqli = new mysqli('localhost', 'root', '123456', 'site');
```

Simple **mysqli** samples, encodings configuration

To configure encodings for DBMS interaction with **mysqli** extension we may use the following approaches:

```php
<?php

// Usually this is enough:
$mysqli->set_charset('UTF8');

// But sometimes we may need this:
$mysqli->query("SET CHARACTER SET 'UTF8'");
$mysqli->query("SET CHARSET 'UTF8'");
$mysqli->query("SET NAMES 'UTF8'");
```

Simple **mysqli** samples, queries execution

To execute a query to DBMS with **mysqli** extension we may use the following approach:

```php
<?php

// Inserting data
$result = $mysqli->query("INSERT INTO `site_users`
                          (`su_fio`, `su_email`, `su_login`, `su_password`)
                          VALUES ('Smith J.', 'smith@gmail.com', 'smith', '" . sha1('smith') . "')");

// Selecting data
$result = $mysqli->query("SELECT * FROM `site_users`");
```

Simple **mysqli** samples, results processing

To process a query execution result with **mysqli** extension we may use the following approach:

```php
<?php

$result = $mysqli->query("SELECT * FROM `site_users`");

if ($result !== false) {
    while ($row = $result->fetch_assoc()) {
        echo $row['su_fio'] . ' (' . $row['su_email'] . ')';
    }
    $result->free();
}
```

Simple **mysqli** samples, closing connection

To close a connection to a DBMS with **mysqli** extension we may use the following approach:

```php
<?php

$mysqli->close();
```

All **mysqli** actions in one sample

Now let's combine all these **mysqli** actions in one simple script that demonstrates all necessary operations at once:

```php
<?php

// 1. Connect to DBMS
$mysqli = new mysqli('localhost', 'root', '123456', 'site');

// 2. Setup encodings
$mysqli->set_charset('UTF8');

// 3. Run a query
$result = $mysqli->query("SELECT * FROM `site_users`");

// 4. Process results
if ($result !== false) {
    while ($row = $result->fetch_assoc()) {
        echo $row['su_fio'] . ' (' .
             $row['su_email'] . ')' . "\n";
    }
    $result->free();
}

// 5. Close connection
$mysqli->close();
```

```sql
/* SQL script to test this sample:

CREATE TABLE IF NOT EXISTS `site_users` (
  `su_uid` int(11) NOT NULL AUTO_INCREMENT,
  `su_login` varchar(200) NOT NULL DEFAULT '',
  `su_password` char(40) NOT NULL DEFAULT '',
  `su_fio` varchar(255) NOT NULL DEFAULT '',
  `su_email` varchar(255) NOT NULL DEFAULT '',
  PRIMARY KEY (`su_uid`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;

INSERT INTO `site_users` (`su_login`, `su_password`, `su_fio`, `su_email`) VALUES
('user1', 'e9d71f5ee7c92d6dc9e92ffdad17b8bd49418f98', 'John Smith', 'jh.sm@gmail.com'),
('user2', 'da4b9237bacccdf19c0760cab7aec4a8359010b0', 'Jane Smith', 'jn.sm@gmail.com'),
('user3', 'ff4b9cc7baccc1219c0760cab7aec4a8359010b0', 'Some User', 'su@gmail.com');
*/
```

And what about PDO?!

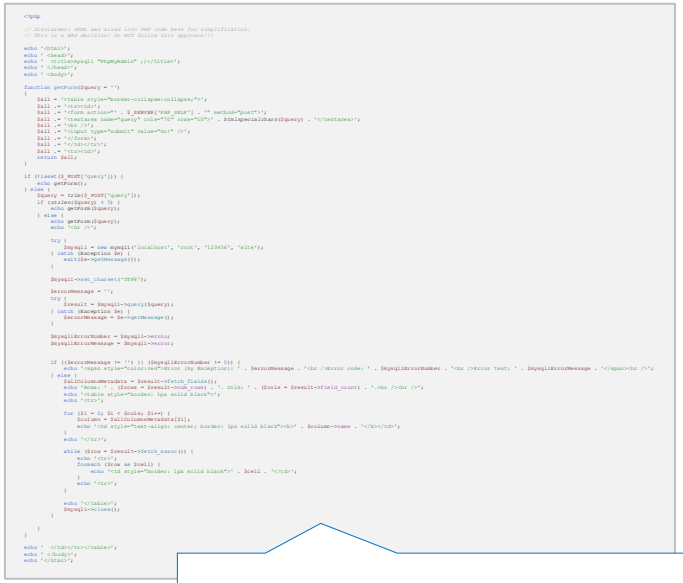
Now let’s review all the same actions (we’ve just seen with **mysqli**), but using PDO extension:

```php
<?php

// 1. Connect to DBMS
$pdo = new PDO('mysql:dbname=site;host=127.0.0.1;charset=utf8’,
               'root', '123456');

// 2. Setup encodings (not needed, as charset
// was set in the connection string)
// $pdo->exec('set names utf8');

// 3. Run a query
$result = $pdo->query("SELECT * FROM `site_users`");

// 4. Process results
while ($row = $result->fetch()) {
    echo $row['su_fio'] . ' (' .
         $row['su_email'] . ')' . "\n";
}

// 5. Close connection
unset($pdo);
```

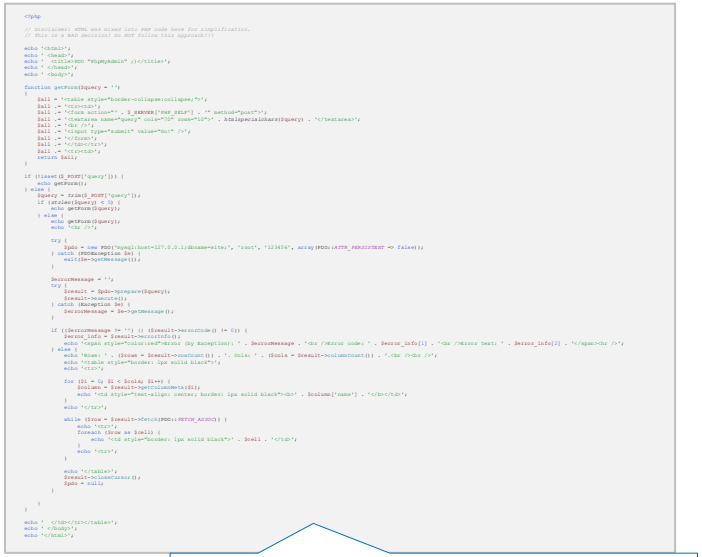
```sql
/* SQL script to test this sample:

CREATE TABLE IF NOT EXISTS `site_users` (
  `su_uid` int(11) NOT NULL AUTO_INCREMENT,
  `su_login` varchar(200) NOT NULL DEFAULT '',
  `su_password` char(40) NOT NULL DEFAULT '',
  `su_fio` varchar(255) NOT NULL DEFAULT '',
  `su_email` varchar(255) NOT NULL DEFAULT '',
  PRIMARY KEY (`su_uid`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;

INSERT INTO `site_users` (`su_login`, `su_password`, `su_fio`, `su_email`) VALUES
('user1', 'e9d71f5ee7c92d6dc9e92ffdad17b8bd49418f98', 'John Smith', 'jh.sm@gmail.com'),
('user2', 'da4b9237bacccdf19c0760cab7aec4a8359010b0', 'Jane Smith', 'jn.sm@gmail.com'),
('user3', 'ff4b9cc7baccc1219c0760cab7aec4a8359010b0', 'Some User', 'su@gmail.com');
*/
```

Finally…

To explore the power of **mysqli** and **PDO** please scrutinize these two scripts (see handouts, as these scripts are too large to fit the screen):



See “08_mysqli_all_in_one.php”



See “09_pdo_all_in_one.php”

And what about security?

Let’s take a quick look at a perfect way to protect your queries from malicious user input – prepared statements:

```php
<?php

try {
    $pdo = new PDO('mysql:host=127.0.0.1;dbname=site', 'root', '123456', array(PDO::ATTR_PERSISTENT => false));
} catch (PDOException $e) {
    exit($e->getMessage());
}

$pdo->exec("SET CHARACTER SET 'UTF8'");
$pdo->exec("SET CHARSET 'UTF8'");
$pdo->exec("SET NAMES 'UTF8'");

// Binding columns for data selection
$result = $pdo->prepare("SELECT * FROM `site_users`");
$result->execute();
$result->bindColumn('su_fio', $fio);
$result->bindColumn('su_email', $email);

while ($row = $result->fetch(PDO::FETCH_BOUND)) {
    echo $fio . ' = ' . $email . "\n";
}

$result->closeCursor();

// Parameterization with prepared statements
$result = $pdo->prepare("SELECT * FROM `site_users` WHERE `su_uid`=:su_uid OR `su_email`=:su_email");
$result->bindValue(':su_uid', 1, PDO::PARAM_INT);
$result->bindValue(':su_email', 'jh.sm@gmail.com', PDO::PARAM_STR);
$result->execute();

// Print the whole result at once
print_r($result->fetchAll());

$result->closeCursor();
unset($pdo);
```

Here we use special mechanism to bind query parameters without risking to break the query by malicious parameters values.

DBMS Functions

Disclaimer: вы смотрите просто запись лекции, это **НЕ** специально подготовленный видеокурс!