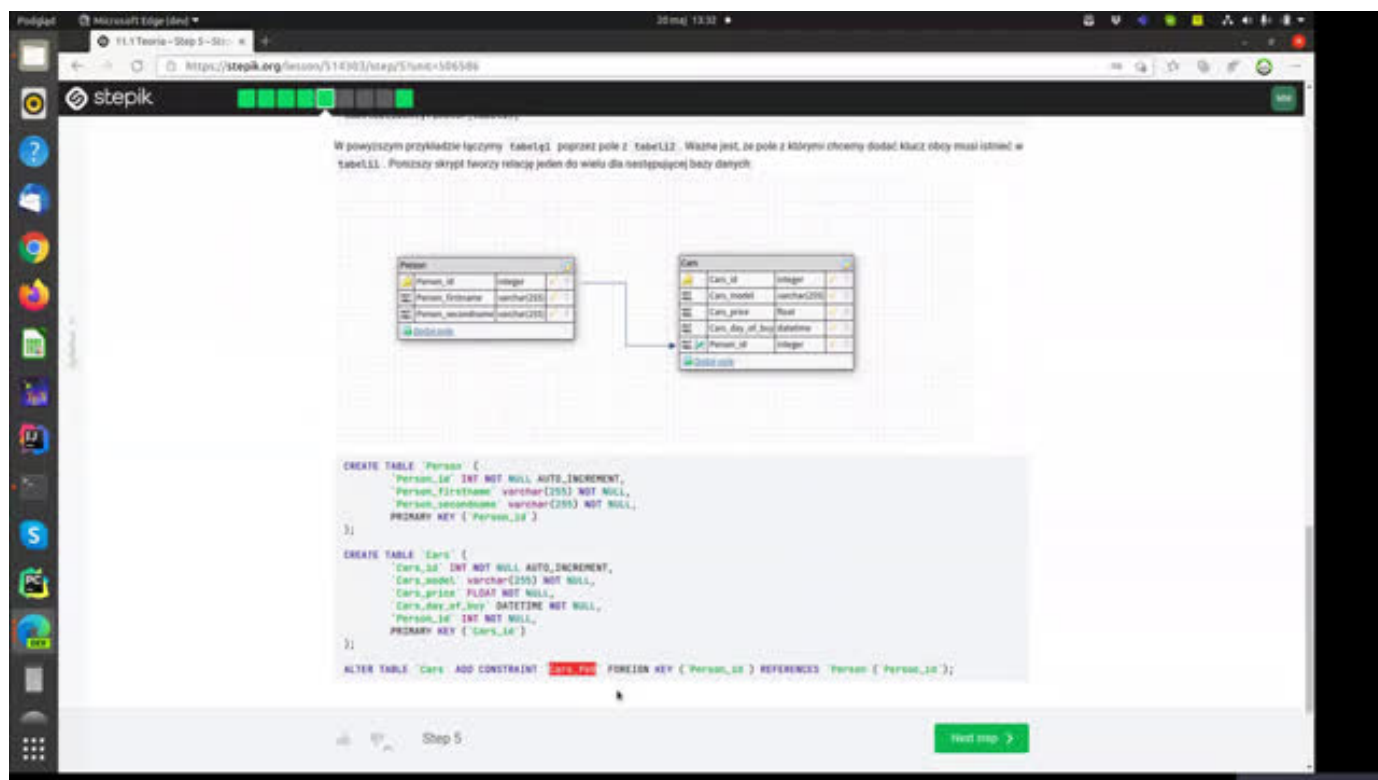


13.1 Teoria - MySQLi

Step 1



To watch this video please visit <https://stepik.org/lesson/514303/step/1>

Step 2

Najpopularniejszą platformą bazodanową używaną w PHP jest baza danych MySQL. Jeśli spojrzymy na witrynę MySQL, zobaczymy, że istnieje kilka różnych wersji MySQL, z których możemy korzystać. PHP ma również wiele różnych interfejsów do tego narzędzia bazy danych, więc przyjrzymy się interfejsowi obiektowemu znanemu jako `MySQLi`, czyli rozszerzenie **MySQL Improved**.

Dokumentacja do interfejsu `MySQLi` : <https://www.php.net/manual/en/book.mysqli.php>
(<https://www.php.net/manual/en/book.mysqli.php>)

Uwaga. Aby działał interfejs `MySQLi` należy w pliku `php.ini` w zakładce `extensions` odkomentować linię `extension=mysqli` (wcześniej należy zainstalować wsparcie MySQL dla PHP - w systemie Linux poprzez instalację pakietu `php*-mysql`, gdzie `*` oznacza wersję PHP).

MySQL udostępnia interfejs obiektowy. Zapewnia zarówno zorientowane obiektowo, jak i proceduralne podejście do obsługi operacji bazy danych.

```
<?php  
$mysqli = mysqli_connect("localhost", "root", "silneHaslo123@", "example");  
  
$result = mysqli_query($mysqli, "SELECT 'Wprowadzenie do MySQLi' AS _message FROM DUAL");  
$row = mysqli_fetch_assoc($result);  
echo $row['_message'];  
?>
```

```
<?php
    $mysqli = new mysqli("localhost", "root", "silneHaslo123@", "example");

    $result = $mysqli->query("SELECT 'Wprowadzenie do MySQLi' AS _message FROM DUAL");
    $row = $result->fetch_assoc();
    echo $row['_message'];
?>
```

PHP zapewnia różne funkcje dostępu do bazy danych MySQL i manipulowania rekordami danych wewnątrz bazy danych MySQL. Wymagane jest wywołanie funkcji PHP w taki sam sposób, jak każda inna funkcja PHP. Funkcje PHP do użytku z MySQL mają następujący ogólny format.

```
mysqli function(value,value,...);
```

PHP zapewnia **utworzenie obiektu poprzez konstruktor** `mysqli` lub użycie funkcji `mysqli_connect()` w celu otwarcia połączenia z bazą danych. Ta funkcja przyjmuje sześć parametrów i zwraca identyfikator łączy MySQL na sukces lub FALSE w przypadku niepowodzenia.

```
$mysqli = new mysqli($host, $username, $passwd, $dbName, $port, $socket);
```

Wszystkie parametry są opcjonalne. Ich znaczenie przedstawia poniższa tabela:

Parametr	Znaczenie
<code>\$host</code>	Nazwa hosta z uruchomionym serwerem bazy danych. Jeśli nie zostanie określony, wartość domyślna będzie <code>localhost:3306</code> .
<code>\$username</code>	Nazwa użytkownika uzyskująca dostęp do bazy danych. Jeśli nie zostanie określony, wartość domyślna będzie nazwa użytkownika, który jest właścicielem procesu serwera.
<code>\$passwd</code>	Hasło użytkownika uzyskującego dostęp do bazy danych. Jeśli nie zostanie określony, domyślnym będzie puste hasło.
<code>\$dbname</code>	Nazwa bazy danych, na której ma być wykonywana kwerenda.
<code>\$port</code>	numer portu, aby spróbować połączyć się z serwerem MySQL
<code>\$socket</code>	gniazdo lub potok, który powinien być używany do połączenia z serwerem

W każdej chwili można odłączyć się od bazy danych MySQL za pomocą metody PHP `close()`.

```
$mysqli->close();
```

```

<html>
<head>
    <title>Connecting MySQL Server</title>
</head>
<body>
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = 'silneHaslo123@';
    $mysqli = new mysqli($dbhost, $dbuser, $dbpass);

    if($mysqli->connect_errno ) {
        printf("Connect failed: %s<br />", $mysqli->connect_error);
        exit();
    }
    printf('Connected successfully.<br />');
    $mysqli->close();
?>
</body>
</html>

```

Uwaga: W przypadku występowania wyjątków typu `mysqli_sql_exception` możemy albo przechwytywać je w klauzulach `try ... catch`, lub możemy automatycznie przechwycić (zignorować) wszystkie wyjątki za pomocą polecenia w kodzie `PHP`

```
mysqli_report(MYSQLI_REPORT_OFF);
```

Step 3

PHP używa funkcji `mysqli->query()` lub `mysql_query()` do tworzenia lub usuwania bazy danych MySQL. Ta funkcja przyjmuje dwa parametry i zwraca wartość PRAWDA w przypadku powodzenia lub FAŁSZ w przypadku awarii.

```
$mysqli->query($sql,$resultmode)
```

Parametr `$sql` jest wymagany i jest to zapytanie SQL do utworzenia bazy danych. Parametr `$resultmode` jest opcjonalny i jest to stała `MYSQLI_USE_RESULT` lub `MYSQLI_STORE_RESULT`. Domyślnie używany jest `MYSQLI_STORE_RESULT`.

```

<html>
<head><title>Creating MySQL Database</title></head>
<body>
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = 'silneHaslo123@';
    $mysqli = new mysqli($dbhost, $dbuser, $dbpass, $dbname);

    if($mysqli->connect_errno) {
        printf("Connect failed: %s<br />", $mysqli->connect_error);
        exit();
    }
    printf('Connected successfully.<br />');

    $createDatabaseQuery = "CREATE DATABASE IF NOT EXISTS TUTORIALS";
    if ($mysqli->query($createDatabaseQuery)) {
        if ($mysqli->affected_rows > 0) {
            printf("Database TUTORIALS created successfully.<br />");
        } else {
            printf("Database TUTORIALS already exists.<br />");
        }
    } else {
        printf("Could not create database: %s<br />", $mysqli->error);
    }

    $mysqli->close();
?>
</body>
</html>

```

Możemy zmodyfikować powyższy skrypt tak, że w przypadku istnienia bazy `TUTORIALS` zostanie ona skasowana i utworzona na nowo:

```

<html>
<head><title>Creating MySQL Database</title></head>
<body>
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = 'siłneHasło123@';
    $mysqli = new mysqli($dbhost, $dbuser, $dbpass);

    if($mysqli->connect_errno ) {
        printf("Connect failed: %s<br />", $mysqli->connect_error);
        exit();
    }
    printf('Connected successfully.<br />');

    if ($mysqli->query("CREATE DATABASE TUTORIALS")) {
        printf("Database TUTORIALS created successfully.<br />");
    }
    if ($mysqli->error) { // Corrected line
        if ($mysqli->query("DROP DATABASE TUTORIALS")) {
            printf("Database TUTORIALS dropped successfully.<br />"); // Corrected line
        }
        else {
            printf("Could not create database: %s<br />", $mysqli->error);
        }
    }

    $mysqli->close();
?>
</body>
</html>

```

PHP używa **funkcji** `mysqli_select_db`, aby wybrać bazę danych, na której mają być wykonywane kwerendy. Ta funkcja przyjmuje dwa parametry i zwraca wartość PRAWDA w przypadku powodzenia lub FAŁSZ w przypadku awarii.

```
mysqli_select_db ( mysqli $link , string $dbname ) : bool
```

Parametr `$link` jest wymagany i jest to identyfikator łącza zwrócony przez `mysqli_connect()`. Parametr `$dbname` jest wymagany i jest to nazwa bazy danych do połączenia.

```

<html>
<head>
    <title>Selecting MySQL Database</title>
</head>
<body>
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = 'siłneHasło123@';
    $conn = mysqli_connect($dbhost, $dbuser, $dbpass);

    if(! $conn ) {
        die('Could not connect: ' . mysqli_error($conn));
    }
    echo 'Connected successfully<br />';

    $retval = mysqli_select_db( $conn, 'TUTORIALS' );

    if(! $retval ) {
        die('Could not select database: ' . mysqli_error($conn));
    }
    echo "Database TUTORIALS selected successfully\n";
    mysqli_close($conn);
?>
</body>
</html>

```

Metoda `query` może stanowić również podstawę tworzenia tabeli za pomocą języka PHP:

```

<html>
<head>
    <title>Creating MySQL Table</title>
</head>
<body>
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = 'silneHaslo123@';
    $dbname = 'TUTORIALS';
    $mysqli = new mysqli($dbhost, $dbuser, $dbpass, $dbname);

    if($mysqli->connect_errno ) {
        printf("Connect failed: %s<br />", $mysqli->connect_error);
        exit();
    }
    printf('Connected successfully.<br />');

    $sql = "CREATE TABLE tutorials_tbl( ".
        "tutorial_id INT NOT NULL AUTO_INCREMENT, ".
        "tutorial_title VARCHAR(100) NOT NULL, ".
        "tutorial_author VARCHAR(40) NOT NULL, ".
        "submission_date DATE, ".
        "PRIMARY KEY ( tutorial_id )); ";
    if ($mysqli->query($sql)) {
        printf("Table tutorials_tbl created successfully.<br />");
    }
    if ($mysqli->errno) {
        if ($mysqli->query("DROP TABLE tutorials_tbl")){
            printf("Table tutorials_tbl dropped succesfully");
        }
        else {
            printf("Could not create table: %s<br />", $mysqli->error);
        }
    }
    $mysqli->close();
?>
</body>
</html>

```

Step 4

Kolejny przykład korzysta z metody query w celu dodania rekordów do bazy danych:

```

<html>
<head>
    <title>Add New Record in MySQL Database</title>
</head>
<body>
<?php
    if(isset($_POST['add'])) {
        $dbhost = 'localhost';
        $dbuser = 'root';
        $dbpass = 'silneHaslo123@';
        $dbname = 'TUTORIALS';
        $mysqli = new mysqli($dbhost, $dbuser, $dbpass, $dbname);

        if($mysqli->connect_errno ) {
            printf("Connect failed: %s<br />", $mysqli->connect_error);
            exit();
        }
        printf('Connected successfully.<br />');

        $tutorial_title = $_POST['tutorial_title'];
        $tutorial_author = $_POST['tutorial_author'];

        $submission_date = $_POST['submission_date'];
        $sql = "INSERT INTO tutorials_tbl ".
            "(tutorial_title,tutorial_author, submission_date) ". "VALUES ".
            "('$tutorial_title','$tutorial_author','$submission_date')";

        if ($mysqli->query($sql)) {
            printf("Record inserted successfully.<br />");
        }
        if ($mysqli->errno) {
            printf("Could not insert record into table: %s<br />", $mysqli->error);
        }
        $mysqli->close();
    } else {
        //$_PHP_SELF oznacza wykonanie skryptu będącego w pliku .php (w tym przypadku kodu powyżej)
        ?>
        <form method = "post" action = "<?php $_PHP_SELF ?>">
            <table width = "600" border = "0" cellpadding = "1" cellspacing = "2">
                <tr>
                    <td width = "250">Tutorial Title</td>
                    <td><input name = "tutorial_title" type = "text" id = "tutorial_title"></td>
                </tr>
                <tr>
                    <td width = "250">Tutorial Author</td>
                    <td><input name = "tutorial_author" type = "text" id = "tutorial_author"></td>
                </tr>
                <tr>
                    <td width = "250">Submission Date [   yyyy-mm-dd ]</td>
                    <td><input name = "submission_date" type = "text" id = "submission_date"></td>
                </tr>
                <tr>
                    <td width = "250"> </td>
                    <td></td>
                </tr>
                <tr>
                    <td width = "250"> </td>
                    <td><input name = "add" type = "submit" id = "add" value = "Add Tutorial"></td>
                </tr>
            </table>
        </form>
        <?php
    }
?>
</body>
</html>

```

Mając możliwość wstawiania danych, możemy również sprawdzić czy zostały one dodane poprzez użycie zapytania SELECT:

```

<html>
<head>
    <title>Creating MySQL Table</title>
</head>
<body>
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = 'silneHaslo123@';
    $dbname = 'TUTORIALS';
    $mysqli = new mysqli($dbhost, $dbuser, $dbpass, $dbname);

    if($mysqli->connect_errno ) {
        printf("Connect failed: %s<br />", $mysqli->connect_error);
        exit();
    }
    printf('Connected successfully.<br />');

    $sql = "SELECT tutorial_id, tutorial_title, tutorial_author, submission_date FROM tutorials_tbl";

    $result = $mysqli->query($sql);

    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            printf("Id: %s, Title: %s, Author: %s, Date: %d <br />",
                $row["tutorial_id"],
                $row["tutorial_title"],
                $row["tutorial_author"],
                $row["submission_date"]);
        }
    } else {
        printf('No record found.<br />');
    }
    mysqli_free_result($result);
    $mysqli->close();
?>
</body>
</html>

```

Używana jest tutaj metoda obiektu o nazwie `fetch_assoc()` , aby dostarczyć jeden wiersz danych naraz i przechowywać ten pojedynczy wiersz w zmiennej o nazwie `$row` . Pętla `while` trwa tak długo, jak długo istnieją wiersze do przetworzenia. W ramach pętli `while` zrzucana jest zawartość zmiennej `$row` do okna przeglądarki. Na koniec zamykany jest zarówno wynik, jak i obiekty bazy danych.

Kolejny przykład aktualizuje rekordy w bazie danych korzystając z metody `query` :

```

<html>
<head>
    <title>Updating MySQL Table</title>
</head>
<body>
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = 'silneHaslo123@';
    $dbname = 'TUTORIALS';
    $mysqli = new mysqli($dbhost, $dbuser, $dbpass, $dbname);

    if($mysqli->connect_errno ) {
        printf("Connect failed: %s<br />", $mysqli->connect_error);
        exit();
    }
    printf('Connected successfully.<br />');

    if ($mysqli->query('UPDATE tutorials_tbl set tutorial_title = "Learning Java" where tutorial_id = 2'))
    {
        printf("Table tutorials_tbl updated successfully.<br />");
    }
    if ($mysqli->errno) {
        printf("Could not update table: %s<br />", $mysqli->error);
    }
    $sql = "SELECT tutorial_id, tutorial_title, tutorial_author, submission_date FROM tutorials_tbl";

    $result = $mysqli->query($sql);

    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            printf("Id: %s, Title: %s, Author: %s, Date: %d <br />",
                $row["tutorial_id"],
                $row["tutorial_title"],
                $row["tutorial_author"],
                $row["submission_date"]);
        }
    } else {
        printf('No record found.<br />');
    }
    mysqli_free_result($result);
    $mysqli->close();
?>
</body>
</html>

```

Ostatni w tej sekcji przykład usuwa dane z tabeli za pomocą metody query:


```

<html>
<head>
    <title>Deleting MySQL Table record</title>
</head>
<body>
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = 'silneHaslo123@';
    $dbname = 'TUTORIALS';
    $mysqli = new mysqli($dbhost, $dbuser, $dbpass, $dbname);

    if($mysqli->connect_errno ) {
        printf("Connect failed: %s<br />", $mysqli->connect_error);
        exit();
    }
    printf('Connected successfully.<br />');

    if ($mysqli->query('DELETE FROM tutorials_tbl where tutorial_id = 1')) {
        printf("Table tutorials_tbl record deleted successfully.<br />");
    }
    if ($mysqli->errno) {
        printf("Could not delete record from table: %s<br />", $mysqli->error);
    }

    $sql = "SELECT tutorial_id, tutorial_title, tutorial_author, submission_date FROM tutorials_tbl";

    $result = $mysqli->query($sql);

    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            printf("Id: %s, Title: %s, Author: %s, Date: %d <br />",
                $row["tutorial_id"],
                $row["tutorial_title"],
                $row["tutorial_author"],
                $row["submission_date"]);
        }
    } else {
        printf('No record found.<br />');
    }
    mysqli_free_result($result);
    $mysqli->close();
?>
</body>
</html>

```

Step 5

Rozważmy tabelę `tcount_tbl`, której zawartość wygląda następująco:

tutorial_author	tutorial_count
Mahesh	3
Suresh	1
Mateusz Miotk	20
Mateusz	NULL
Kazik	NULL

Najpierw utwórzmy ją za pomocą zapytania SQL (wraz z dodaniem powyższych danych):

```

create table tcount_tbl(
    tutorial_author VARCHAR(40) NOT NULL,
    tutorial_count int
);
insert into tcount_tbl values('Mahesh', 3);
insert into tcount_tbl values('Suresh', 1);
insert into tcount_tbl values('Mateusz Miotk', 20);
insert into tcount_tbl values('Mateusz', NULL);
insert into tcount_tbl values('Kazik', NULL);

```

Teraz połączmy te tabelę za pomocą `JOIN` przy użyciu metody `query` w PHP:

```
<html>
<head>
  <title>Handling NULL</title>
</head>
<body>
<?php
  $dbhost = 'localhost';
  $dbuser = 'root';
  $dbpass = 'silneHaslo123@';
  $dbname = 'TUTORIALS';
  $mysqli = new mysqli($dbhost, $dbuser, $dbpass, $dbname);

  if($mysqli->connect_errno ) {
    printf("Connect failed: %s<br />", $mysqli->connect_error);
    exit();
  }
  printf('Connected successfully.<br />');

  $sql = 'SELECT a.tutorial_id, a.tutorial_author, b.tutorial_count
        FROM tutorials_tbl a, tcount_tbl b
        WHERE a.tutorial_author = b.tutorial_author';

  $result = $mysqli->query($sql);

  if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
      printf("Id: %s, Author: %s, Count: %d <br />",
        $row["tutorial_id"],
        $row["tutorial_author"],
        $row["tutorial_count"]);
    }
  } else {
    printf('No record found.<br />');
  }
  mysqli_free_result($result);
  $mysqli->close();
?>
</body>
</html>
```

Korzystając z konstrukcji `if... warunek else`, możemy przygotować obsłużyć zapytanie, które będzie posiadało wartości NULL.

```

<html>
<head>
    <title>Handling NULL</title>
</head>
<body>
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = 'silneHaslo123@';
    $dbname = 'TUTORIALS';
    $mysqli = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
    $tutorial_count = null;
    if($mysqli->connect_errno ) {
        printf("Connect failed: %s<br />", $mysqli->connect_error);
        exit();
    }
    printf('Connected successfully.<br />');
    if( isset($tutorial_count) ) {
        $sql = 'SELECT tutorial_author, tutorial_count
                FROM tcount_tbl
                WHERE tutorial_count = ' + $tutorial_count;
    } else {
        $sql = 'SELECT tutorial_author, tutorial_count
                FROM tcount_tbl
                WHERE tutorial_count IS NULL';
    }
    $result = $mysqli->query($sql);

    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            printf("Author: %s, Count: %d <br />",
                $row["tutorial_author"],
                $row["tutorial_count"]);
        }
    } else {
        printf('No record found.<br />');
    }
    $mysqli->close();
?>
</body>
</html>

```

Z MySQL możemy uzyskać trzy rodzaje informacji: o wynikach zapytań (co było pokazane dotychczas), o tabelach i bazach danych oraz o serwerze MySQL. Aby dowiedzieć się jaka jest domyślna użyta baza danych możemy skorzystać z następującego kodu:

```

<html>
<head>
    <title>Getting MySQL Database Info</title>
</head>
<body>
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = 'silneHaslo123@';
    $dbname = 'TUTORIALS';
    $mysqli = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
    $tutorial_count = null;

    if($mysqli->connect_errno ) {
        printf("Connect failed: %s<br />", $mysqli->connect_error);
        exit();
    }
    printf('Connected successfully.<br />');

    if ($result = mysqli_query($mysqli, "SELECT DATABASE()")) {
        $row = mysqli_fetch_row($result);
        printf("Default database is %s<br />", $row[0]);
        mysqli_free_result($result);
    }
    $mysqli->close();
?>
</body>
</html>

```

Oprócz `SELECT DATABASE()` mamy również do dyspozycji `SELECT VERSION()`, `SELECT USER()`, `SHOW STATUS` oraz `SHOW VARIABLES`.