

Temat: Połączenia z bazą danych,
mysqli -ujęcie proceduralne

PHP – połączenia z bazami danych

W PHP do połączeń z bazami danych MySQL możemy używać

- ✓ **Rozszerzenia MySQLi**
- ✓ **PDO (PHP Data Objects)**

Rozszerzenia mysqli możemy używać zarówno w wersji proceduralnej jak i obiektowej.

Kryteria:

1. Nawiązanie połączenia,
wykorzystanie rozszerzenia **mysqli**

PHP – Nawiązanie połączenia

Nawiązanie połączenia w wersji proceduralnej:

```
mysqli_connect($dbServer,  
$dbUser, $dbPassword, $dbName);
```

Funkcja zwraca identyfikator nawiązanego połączenia, jeśli udało się je nawiązać, lub wartość false w przeciwnym razie.

PHP – Nawiązanie połączenia

```
<?php  
$dbServer = 'localhost';  
$dbUser = 'root';  
$dbPassword = '';  
$dbName = 'wazniak';  
  
$mysqli = mysqli_connect($dbServer, $dbUser, $dbPassword, $dbName);  
print_r($mysqli);  
?>
```

Połączenie z bazą danych

```
mysqli Object ( [affected_rows] => 0 [client_info] => mysqlnd 5.0.12-dev - 20150407 - $Id:  
38fea24f2847fa7519001be390c98ae0acafe387 $ [client_version] => 50012 [connect_errno] => 0  
[connect_error] => [errno] => 0 [error] => [error_list] => Array () [field_count] => 0 [host_info] =>  
localhost via TCP/IP [info] => [insert_id] => 0 [server_info] => 5.5.5-10.1.35-MariaDB  
[server_version] => 50505 [stat] => Uptime: 8017 Threads: 1 Questions: 119 Slow queries: 0 Opens: 20  
Flush tables: 1 Open tables: 14 Queries per second avg: 0.014 [sqlstate] => 00000 [protocol_version]  
=> 10 [thread_id] => 33 [warning_count] => 0 )
```

Kryteria:

2. Komunikaty o błędzie

PHP – komunikat o błędzie

Komunikat o błędzie w przypadku braku połączenia:

```
<?php  
$dbServer = 'localhost';  
$dbUser = 'root';  
$dbPassword = '';  
$dbName = 'wazniak2';  
  
$mysqli = mysqli_connect($dbServer, $dbUser, $dbPassword, $dbName);  
print_r($mysqli);  
?>
```

Warning: mysqli_connect(): (HY000/1049): Unknown database 'wazniak2' in C:\xampp\htdocs\ZSK\l10p2.php on line 18

PHP – komunikat o błędzie

Przyjazny dla użytkownika komunikat o błędzie w przypadku braku połączenia.

Wykorzystanie funkcji:

`mysqli_connect_errno ()`

- Zwraca ostatni numer kodu błędu z połączenia do `mysqli_connect ()`

`mysqli_connect_error ()`

- Zwraca opis ciągu ostatnich błędów połączenia

PHP – komunikat o błędzie

Przyjazny dla użytkownika komunikat o błędzie w przypadku braku połączenia.

Wykorzystanie funkcji:

die() - kończy wykonywanie skryptu,
konstrukcja języka równoważna konstrukcji
exit ()

PHP – Komunikat o błędzie

```
<?php
$dbServer = 'localhost';
$dbUser = 'root';
$dbPassword = '';
$dbName = 'wazniak2';
// ukrywamy komunikat błędu używając @:
$mysqliCon = @mysqli_connect($dbServer,$dbUser,$dbPassword,$dbName);
// tworzymy własny komunikat o błędzie:
if (!$mysqliCon) {
    die("Błąd połączenia: " . mysqli_connect_errno());
}
echo "Nawiązano połączenie z bazą";
?>
```

Błąd połączenia: 1049

PHP – Komunikat o błędzie

```
<?php
$dbServer = 'localhost';
$dbUser = 'root';
$dbPassword = '';
$dbName = 'wazniak3';
// ukrywamy komunikat błędu używając @:
$mysqli = @mysqli_connect($dbServer, $dbUser, $dbPassword, $dbName);
// tworzymy własny komunikat o błędzie:
if (!$mysqli){
    switch (mysqli_connect_errno()){
        case 1049: $err="Niewłaściwa baza danych"; break;
        case 2002: $err="Nie można połączyć się z serwerem MySQL"; break;
        case 1044: $err="Niewłaściwy użytkownik"; break;
        case 1045: $err="Błędne hasło"; break;
        default : $err="Wystąpił nieznany błąd";
    }
    die("Błąd połączenia. " . $err);
}
echo "Nawiązano połączenie z bazą";
?>
```

PHP – Komunikat o błędzie

```
<?php
$dbServer = 'localhost';
$dbUser = 'root';
$dbPassword = 'b';
$dbName = 'wazniak';
// ukrywamy komunikat błędu używając @:
$mysqliCon = @mysqli_connect($dbServer, $dbUser, $dbPassword, $dbName);
// tworzymy własny komunikat o błędzie:
if (!$mysqliCon) {
    die("Błąd połączenia: " . mysqli_connect_error());
}
echo "Nawiązano połączenie z bazą";
?>
```

Błąd połączenia: Access denied for user 'root'@'localhost' (using password: YES)

PHP – Komunikat o błędzie

```
<?php
$dbServer = 'localhost';
$dbUser = 'root';
$dbPassword = 'b';
$dbName = 'wazniak';
// ukrywamy komunikat błędu używając @:
$mysqliCon = @mysqli_connect($dbServer,$dbUser,$dbPassword,$dbName)
// tworzymy własny komunikat o błędzie:
| or die("Brak połączenia z serwerem MySQL lub bazą danych");
echo "Nawiązano połączenie z bazą";
?>
```

| Brak połączenia z serwerem MySQL lub bazą danych

PHP – zamykanie połączenia

Połączenie otwarte za pomocą funkcji **mysqli_connect** zostanie zamknięte automatycznie, gdy skrypt zakończy działanie lub gdy zostanie wywołana funkcja **mysqli_close**.

Dobry obyczaj - jawne zamykanie połączenia za pomocą tej funkcji.

mysqli_close([*identyfikator*])

identyfikator - identyfikator połączenia, które ma zostać zamknięte, zwrócony wcześniej przez funkcję **mysqli_connect**

Kryteria:

3. Zapytania do bazy danych –
wyszukiwanie informacji

PHP –Zapytania do bazy danych

Zapytania są wysyłane do bazy za pomocą funkcji

`mysqli_query (identyfikator, "zapytanie")`

Ciąg tworzący zapytanie nie powinien być zakończony znakiem średnika.

PHP –Zapytania do bazy danych

Wartość zwracana przez funkcję jest zależna od typu zapytania.

- ✓ Zapytanie typu SELECT - wartością zwracaną jest identyfikator zasobów, o ile wykonanie zapytania zakończyło się sukcesem, lub false, jeżeli ta operacja się nie powiodła.
- ✓ W przypadku pozostałych typów zapytań zwracaną wartością jest true, jeśli zapytanie było poprawne, lub false w przeciwnym razie.

PHP –Zapytania do bazy danych

```
<?php
// określenie danych dostępu do bazy danych
$dbServer = 'localhost';
$dbUser = 'root';
$dbPassword = '';
$dbName = 'wazniak';
// nawiązanie połączenia, ukrycie komunikatu błędu z użyciem @:
$mysqliCon = @mysqli_connect($dbServer, $dbUser, $dbPassword, $dbName)
// tworzenie komunikatu o błędzie:
| or die("Brak połączenia z serwerem MySQL lub bazą danych");
echo "Nawiązano połączenie z bazą";
// wywołanie zapytania do bazy
$result = mysqli_query ($mysqliCon, "SELECT * FROM pracownicy")
| or
die ("błąd w zapytaniu");
// wyświetlenie wyniku zapytania:
echo '<pre>';
print_r($result);
echo '</pre>';
// zamknięcie połączenia
mysqli_close($mysqliCon);
?>
http://localhost/zsk/l10p7.php
```

Nawiązano połączenie z bazą
mysqli_result Object
(
 [**current_field**] => 0
 [**field_count**] => 9
 [**lengths**] =>
 [**num_rows**] => 14
 [**type**] => 0
)

Kryteria:

4. Odczyt wyników zapytania

PHP – Odczyt wyników zapytania

Funkcje odczytujące dane:

mysqli_fetch_row — Każde wywołanie funkcji zwraca kolejny wiersz z tabeli będącej wynikiem zapytania. Dane zwracane są w tablicy, w której kolejne komórki zawierają dane z kolejnych kolumn tabeli wynikowej. Jeśli zostaną odczytane wszystkie wiersze, funkcja zwraca wartość `false`.

PHP – Odczyt wyników zapytania

Funkcje odczytujące dane:

mysqli_fetch_array — zwraca pojedynczy wiersz wynikowy w postaci tablicy (asocjacyjnej, numerycznej lub jednocześnie asocjacyjnej i numerycznej); rodzaj tablicy dodajemy wykorzystując odpowiedni opcjonalny parametr.

np. **\$row = mysqli_fetch_array(\$result, MYSQLI_ASSOC)**

Możliwe wartości parametru: **MYSQLI_ASSOC**,
MYSQLI_NUM, **MYSQLI_BOTH**

Nazwy pól zwracane przez tę funkcję uwzględniają wielkość *liter*.

PHP – Odczyt wyników zapytania

Funkcje odczytujące dane:

mysqli_fetch_assoc — zwraca pojedynczy wiersz wynikowy w postaci tablicy asocjacyjnej.

Każdy klucz w tablicy reprezentuje nazwę jednej z kolumn zestawu wyników.

Nazwy pól zwracane przez tę funkcję uwzględniają wielkość liter

mysqli_num_rows - pozwala określić, ile wierszy znajduje się w wynikach zapytania

PHP – Odczyt wyników zapytania

wszystkie wyniki zapytania mogą zostać odczytane :

- ✓ w pętli while o schematycznej postaci:

```
while($arr = mysqli_fetch_row($result)){
    // Instrukcje przetwarzające wyniki
}
```

gdzie **\$arr** to tablica, w której będą zapisywane dane z kolejnych wierszy, a **\$result** - zmienna zawierająca identyfikator zasobów zwrócony przez funkcję **mysqli_query**

PHP – Odczyt wyników zapytania

wszystkie wyniki zapytania mogą zostać odczytane :

✓ w pętli typu for z wykorzystaniem funkcji **mysqli_num_rows** do pobrania liczby wierszy z tabeli wynikowej.

Taka konstrukcja ma postać:

```
$count = mysqli_num_rows($result);
for($i = 0; $i < $count; $i++){
    // Instrukcje przetwarzające wyniki
}
```

PHP – Odczyt wyników zapytania

```
// wywołanie zapytania do bazy
$result = mysqli_query ($mysqliCon, "SELECT * FROM pracownicy")
or
die ("błąd w zapytaniu");
// wyświetlenie wyniku zapytania:
while( $row=mysqli_fetch_row($result)) {
echo '<pre>';
print_r($row);
echo '</pre>';
}
// zamknięcie połączenia
mysqli_close($mysqliCon);
?>
```

Nawiązano połączenie z bazą

```
Array
(
    [0] => 100
    [1] => Marecki
    [2] => Jan
    [3] => DYREKTOR
    [4] =>
    [5] => 1968-01-01
    [6] => 4730.00
    [7] => 980.50
    [8] => 10
)
```

```
Array
(
    [0] => 110
    [1] => Janicki
    [2] => Karol
    [3] => PROFESOR
    [4] => 100
    [5] => 1973-05-01
    [6] => 3350.00
    [7] => 610.00
    [8] => 40
)
```

PHP – Odczyt wyników zapytania

Zewnętrzny plik połączenia z bazą danych:

```
<?php
// określenie danych dostępu do bazy danych
$dbServer = 'localhost';
$dbUser = 'root';
$dbPassword = '';
$dbName = 'wazniak';
// nawiązanie połączenia, ukrycie komunikatu błędu z użyciem @:
$mysqliCon = @mysqli_connect($dbServer,$dbUser,$dbPassword,$dbName)
// tworzenie komunikatu o błędzie:
    or die("Brak połączenia z serwerem MySQL lub bazą danych");
echo "Nawiązano połączenie z bazą";
```

PHP – Odczyt wyników zapytania

```
<?php
// podpięcie skryptu połączenia z bazą
include('l10_db_connect.php');
echo '<br>';
// wywołanie zapytania do bazy
$sqlQuery='select nazwisko, imie, etat, plac_a_pod from pracownicy
where lower(etat)!="profesor"';
$result = mysqli_query ($mysqliCon, $sqlQuery)
    or
die ("błąd w zapytaniu");
// wyświetlenie wyniku zapytania:
while( $row=mysqli_fetch_array($result, MYSQLI_ASSOC)) {
    foreach($row as $key=>$value) {
        echo $key. ': ' . $value. ' ';
    }
    echo '<br>';
}
// zamknięcie połączenia
mysqli_close($mysqliCon);
?>
```

Nawiązano połączenie z bazą
nazwisko: Janicki; imie: Karol; etat: PROFESOR; plac_a_pod: 3350.00;
nazwisko: Nowicki; imie: Paweł; etat: PROFESOR; plac_a_pod: 3070.00;
nazwisko: Nowak; imie: Piotr; etat: PROFESOR; plac_a_pod: 3960.00;
nazwisko: Kowalski; imie: Krzysztof; etat: PROFESOR; plac_a_pod: 3230.00;

PHP – Odczyt wyników zapytania

```
<?php
// podpięcie skryptu połączenia z bazą
include('l10_db_connect.php');
echo '<br>';
// zapytanie i jego wywołanie
$sqlQuery="SELECT imie, nazwisko FROM pracownicy
where nazwisko like '%ski'";
$result = mysqli_query ($mysqliCon, $sqlQuery)
or die ("błąd w zapytaniu");
// wyświetlenie wyniku zapytania:
if($result){
    $rowCnt=mysqli_num_rows($result);
    printf('Znaleziono %d osób:<br>', $rowCnt);
    echo '<br>';
    while ($row = mysqli_fetch_array($result, MYSQLI_NUM)) {
        printf ("%s %s ", $row[0], $row[1]);
        echo '<br>';
    }
}
// zamknięcie połączenia
mysqli_close($mysqliCon);
?>
```

Nawiązano połączenie z bazą
Znaleziono 5 osób:
Krzysztof Kowalski
Roman Opolski
Marek Makowski
Konrad Kotarski
Mateusz Siekierski

PHP –Zwolnienie pamięci związanej z wynikiem

Gdy obiekt wynikowy nie jest już potrzebny, powinno się zwolnić wynik za pomocą funkcji:

```
mysqli_free_result($result)
```

PHP –Zwolnienie pamięci związanej z wynikiem

```
<?php
// podpięcie skryptu połączenia z bazą
include('l10_db_connect.php');
echo '<br>';
// zapytanie i jego wywołanie
$sqlQuery="SELECT imie, nazwisko FROM pracownicy
where nazwisko like '%ski'";
$result = mysqli_query ($mysqliCon, $sqlQuery)
or die ("błąd w zapytaniu");
// wyświetlenie wyniku zapytania:
if($result){
    $rowCnt=mysqli_num_rows($result);
    printf('Znaleziono %d osób:<br>', $rowCnt);
    echo '<br>';
    while ($row = mysqli_fetch_array($result, MYSQLI_NUM) ) {
        printf ("%s %s ", $row[0], $row[1]);
        echo '<br>';
    }
}
//Usuwamy z pamięci wynik zapytania
mysqli_free_result($result);
// zamknięcie połączenia
mysqli_close($mysqliCon);
?>
```

Nawiązano połączenie z bazą
Znaleziono 5 osób:

Krzysztof Kowalski
Roman Opolski
Marek Makowski
Konrad Kotarski
Mateusz Siekierski



PHP – wyświetlenie danych w tabelce

```
<style>
    table, th, td, tr {
        border: 1px solid black;
        border-collapse: collapse;
        padding: 10px;
    }
</style>
</head>
<body>
    <h3>Połączenie z bazą danych</h3>
    <?php
        // podpięcie skryptu połączenia z bazą
        require_once('l10_db_connect.php');
        echo '<br>';
        // zapytanie i jego wywołanie
        $sqlQuery="SELECT imie, nazwisko, plac_a_pod as plac_a FROM pracownicy
        where nazwisko like '%ski'";
        $result = mysqli_query ($mysqliCon, $sqlQuery)
        or die ("błąd w zapytaniu");
        // wyświetlenie wyniku zapytania:
```

PHP – wyświetlenie danych w tabelce

```
// wyświetlenie wyniku zapytania:  
echo <<< TABLE  
    <table>  
        <tr>  
            <th>imię</th>  
            <th>nazwisko</th>  
            <th>płaca podstawowa</th>  
        </tr>  
 TABLE;  
while($row = mysqli_fetch_assoc($result)) {  
    echo <<< ROW  
        <tr>  
            <td>$row[imie]</td>  
            <td>$row[nazwisko]</td>  
            <td>$row[placa]</td>  
        </tr>  
 ROW;  
}  
echo "</table>";  
//Usuwamy z pamięci wynik zapytania  
mysqli_free_result($result);  
// zamknięcie połączenia  
mysqli_close($mysqliCon);  
?>
```

Połączenie z bazą danych

Nawiązano połączenie z bazą

imię	nazwisko	placa podstawowa
Krzysztof	Kowalski	3230.00
Roman	Opolski	1839.70
Marek	Makowski	2610.20
Konrad	Kotarski	1971.00
Mateusz	Siekierski	1889.00

Kryteria:

5. Domyślny zestaw znaków

PHP – polskie znaki

Ustawienie domyślnego zestawu znaków, który będzie używany podczas wysyłania danych z i do serwera bazy danych.

`mysqli_set_charset ($con , $charset)`

\$con - identyfikator odsyłacza zwrócony przez `mysqli_connect ()`

\$charset - zestaw znaków do ustawienia jako domyślny.

Np. **`mysqli_set_charset ($con , "utf8")`**

PHP – polskie znaki

```
<?php
// dane dostępu do bazy
$dbServer = 'localhost';
$dbUser = 'root';
$dbPassword = '';
$dbName = 'blog';
// utworzenie połączenia
$mysqliCon = new mysqli($dbServer, $dbUser, $dbPassword, $dbName);
// Sprawdzenie połączenia
if (mysqli_connect_errno()) {
    echo 'Błąd bazy danych';
}
```

Baza blog – utwórz bazę w phpMyAdmin i zimportuj skrypt tworzący bazę **blog.sql**

PHP – polskie znaki

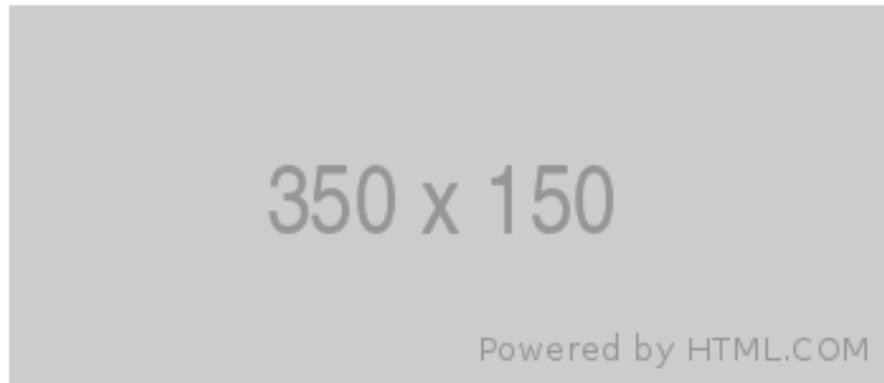
```
<header>
    <h2>Najnowsze artykuły</h2>
</header>
<main>
<!-- lista artykułów -->
<div class="aContainer">
    <?php include('l10p11_db_connect.php') ;
        $sqlQuery="SELECT * FROM article ORDER BY id";
        $result = mysqli_query ($mysqliCon, $sqlQuery)
        or die ("błąd w zapytaniu");
        while ( $article = mysqli_fetch_array($result)) {
            echo '<article>';
            echo '<h3>' . $article['title'] . '</h3>';
            echo '';
            echo '<div>';
            echo '<p>' . $article['content'] . '</p>';
            echo '</div>';
            echo '</article>';
        }
        mysqli_close($mysqliCon);
    ?>
</div>
<aside> reklama
</aside>
</main>
```

Baza blog

PHP – polskie znaki

Najnowsze artykuły

[Jak wys?a? e-mail](#)



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam fermentum et lorem at volutpat. Ut hendrerit fermentum tellus et congue. Proin porta tristique arcu in laoreet. Donec lacinia dictum dolor non cursus. Praesent vitae augue ac turpis porta dapibus. Fusce eleifend ante vel mauris porta

reklama

Baza blog

PHP – polskie znaki

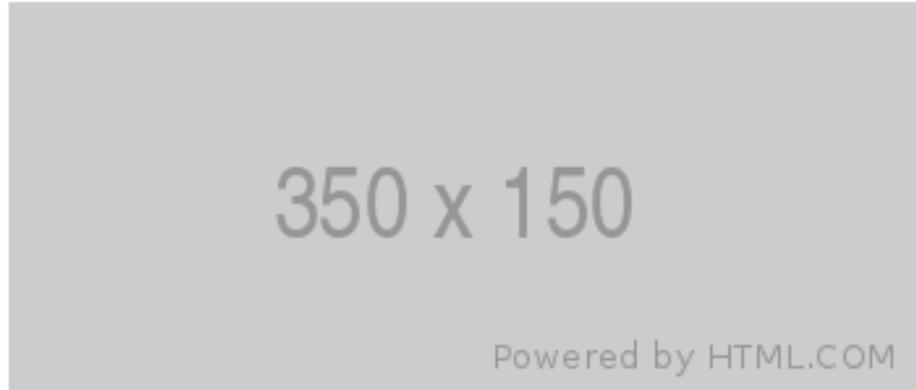
```
<?php
// dane dostępu do bazy
$dbServer = 'localhost';
$dbUser = 'root';
$dbPassword = '';
$dbName = 'blog';
// utworzenie połączenia
$mysqliCon = mysqli_connect($dbServer, $dbUser, $dbPassword, $dbName);
// Sprawdzenie połączenia
if (mysqli_connect_errno()) {
    echo 'Błąd bazy danych';
}
// ustawienie kodowania znaków
mysqli_set_charset($mysqliCon, "utf8");      ←
```

Baza blog

PHP – polskie znaki

Najnowsze artykuły

[Jak wysłać e-mail](#)



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam fermentum et lorem at volutpat. Ut hendrerit fermentum tellus et congue. Proin porta tristique arcu in laoreet. Donec lacinia dictum dolor non cursus. Praesent vitae augue ac turpis porta dapibus. Fusce eleifend ante vel mauris porta

reklama

Baza blog

Kryteria:

6. Tworzenie bazy danych

PHP – Tworzenie bazy danych

```
<?php
    // dane dostępu do serwera
    $dbServer = 'localhost';
    $dbUser = 'root';
    $dbPassword = '';
    // nawiązanie połączenia, ukrycie komunikatu błędu z użyciem @:
    $mysqliCon = @mysqli_connect($dbServer, $dbUser, $dbPassword)
    // tworzenie komunikatu o błędzie:
    or die("Brak połączenia z serwerem");
    //utworzenie bazy
    $sql = "CREATE DATABASE dziennik DEFAULT CHARACTER SET utf8
    COLLATE utf8_general_ci";
    if (mysqli_query($mysqliCon, $sql)) {
        echo "Baza została utworzona";
    } else {
        echo "Błąd przy tworzeniu bazy: " . mysqli_error($mysqliCon);
    }
    // zamykamy połączenie
    mysqli_close($mysqliCon);
?>
```

Kryteria:

7. Tworzenie tabel

PHP – Tworzenie tabel

```
<?php
// dane dostępu do bazy
$dbServer = 'localhost';
$dbUser = 'root';
$dbPassword = '';
$dbName = 'dziennik';
// utworzenie połączenia
$mysqliCon = mysqli_connect($dbServer, $dbUser, $dbPassword, $dbName);
// Sprawdzenie połączenia
if (mysqli_connect_errno()) {
    echo 'Błąd bazy danych';
}
// ustawienie kodowania znaków
mysqli_set_charset($mysqliCon, "utf8");
```

Baza dziennik

PHP – Tworzenie tabel

```
<h3>Tworzenie nowej tabeli - mysqli proceduralne</h3>
<?php
    include('l10p13_db_connect.php');
    //utworzenie tabeli
    $sql = "CREATE TABLE uczniowie (
        id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
        imie VARCHAR(30) NOT NULL,
        nazwisko VARCHAR(30) NOT NULL,
        email VARCHAR(50)
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8";
    if (mysqli_query($mysqliCon, $sql)) {
        echo "Tabela została utworzona";
    } else {
        echo "Błąd tworzenia tabeli: " . mysqli_error($mysqliCon);
    }
    // zamykamy połaczenie
    mysqli_close($mysqliCon);
?>
```

Baza dziennik

Kryteria:

8. Wstawianie danych

PHP – Wstawianie danych

```
<?php
    include('l10p13_db_connect.php');
    //wstawianie danych
    $sql = "INSERT INTO uczniowie (imie, nazwisko, email)
VALUES ('Anna', 'Nowacka', 'anna@example.com')";
    if (mysqli_query($mysqliCon, $sql)) {
        echo "Nowy rekord został wstawiony";
    } else {
        echo "Błąd: ". mysqli_error($mysqliCon);
    }
    // zamykamy połaczenie
    mysqli_close($mysqliCon);
?> |
```

Baza dziennik

PHP – Wstawianie danych, formularz

```
<?php
    include('l10p15_val.php');
?>
<form method="post" action="" class="form1">
<fieldset>
    <label for="fname">Imię:</label>
    <input type="text" name="fname" value='<?php echo $fname;?>'>
    <p class="red_label"><?php echo $fname_err;?></p>
    <br>
    <label for="sname">Nazwisko:</label>
    <input type="text" name="sname" value='<?php echo $sname;?>'>
    <p class="red_label"><?php echo $sname_err;?></p>
    <br>
    <label for="kod_post">Adres email:</label>
    <input type="text" name="email" value='<?php echo $email;?>'>
    <p class="red_label"><?php echo $email_err;?></p>
    <br>
    <input type="submit" name="send" value="wyślij">
</fieldset>
</form>
```

Baza dziennik

PHP – Wstawianie danych, formularz

Validacja:

```
<?php
    $sname="";
    $sname_err="";
    $fname="";
    $fname_err="";
    $email="";
    $email_err="";
    $regex1="/^ [A-Z] [a-z]+$/";
    $valid_form="TRUE";
    if ( isset( $_POST['send'] ) ) {
        if( isset( $_POST['sname'] ) ){
            $sname = strip_tags(trim($_POST['sname']));
            if ( ! $sname ) {
                $sname_err = 'Uzupełnij pole';
                $valid_form="FALSE";
            } elseif ( !preg_match($regex1, $sname) ) {
                $sname_err= 'Nazwisko powinno zaczynać się wielką literą';
                $valid_form="FALSE";
            }
        }
    }
}
```

PHP – Wstawianie danych, formularz

```
if( isset( $_POST['fname'] ) ) {
    $fname = strip_tags(trim($_POST['fname']));
    if ( ! $fname ) {
        $fname_err = 'Uzupełnij pole';
        $valid_form="FALSE";
    } elseif (!preg_match($regex1, $fname)) {
        $fname_err= 'Imię powinno zaczynać się wielką literą';
        $valid_form="FALSE";
    }
}
if( isset( $_POST['email'] ) ) {
    $email = strip_tags(trim($_POST['email']));
    if ( ! $email ) {
        $email_err = 'Uzupełnij pole email';
        $valid_form="FALSE";
    } elseif ($email && !filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $email_err = 'Upewnij się, że adres email ma prawidłowy format';
        $valid_form="FALSE";
    }
}
```

PHP –Wstawianie danych, formularz

```
<?php  
if ( isset( $_POST['send'] ) && $valid_form=="TRUE" ) {  
include('l10p13_db_connect.php');  
//wstawianie danych  
$sql = "INSERT INTO uczniowie (imie, nazwisko, email)  
VALUES ('$fname', '$sname', '$email')";  
if (mysqli_query($mysqlCon, $sql)) {  
    echo "Nowy rekord został wstawiony";  
} else {  
    echo "Błąd: ". mysqli_error($mysqlCon);  
}  
// zamykamy połaczenie  
mysqli_close($mysqlCon);  
?  
?>
```

Baza dziennik

Kryteria:

9. PHP Prepared Statements

PHP –MySQLi Prepared Statements

MySQLi Prepared Statements - to konstrukcja używana do wykonywania tych samych (lub podobnych) instrukcji SQL wielokrotnie z wysoką wydajnością.

Składa się z dwóch etapów: przygotowania i wykonania.

- ✓ Na etapie przygotowania szablon wyciągu jest wysyłany do serwera bazy danych. Serwer wykonuje sprawdzanie składni i inicjuje wewnętrzne zasoby serwera do późniejszego wykorzystania.
- ✓ Po przygotowaniu następuje wykonanie. Podczas wykonywania klient wiąże wartości parametrów i wysyła je do serwera. Serwer tworzy instrukcję z szablonu oświadczenia i powiązanych wartości, aby wykonać ją przy użyciu wcześniej utworzonych zasobów wewnętrznych.

PHP –MySQLi Prepared Statements

- ✓ Tworzymy szablon instrukcji SQL . Niektóre wartości są nieokreślone, zwane parametrami (oznaczone "?").
- ✓ Baza danych analizuje, kompliluje i wykonuje optymalizację zapytań w szablonie instrukcji SQL i przechowuje wynik bez jego wykonywania
- ✓ Następnie aplikacja wiąże wartości z parametrami, a baza danych wykonuje instrukcję.
- ✓ Aplikacja może wykonać instrukcję tyle razy, ile chce, z różnymi wartościami

Wykorzystanie funkcji :

mysqli_prepare

mysqli_stmt_bind_param

mysqli_stmt_execute

i inne

PHP –MySQLi Prepared Statements

Wykorzystanie funkcji .

mysqli_prepare (\$con, \$sqlquery) - Przygotowuje instrukcję SQL do wykonania ;
zwraca uchwyt instrukcji, który będzie używany do dalszych operacji. Zapytanie musi składać się z pojedynczej instrukcji SQL.

\$con - identyfikator odsyłacza zwrócony przez **mysqli_connect ()**
\$sqlquery - zapytanie sql, może zawierać jeden lub więcej znaczników parametrów poprzez osadzanie znaków zapytania (?) na odpowiednich pozycjach.

Np. **INSERT article (title,image,content) VALUES (?,?,?)**

PHP –MySQLi Prepared Statements

Wykorzystanie funkcji .

mysqli_prepare (\$con, \$sqlquery)

Np. **INSERT article (title,image,content) VALUES (?,?,?,?)**

? są dopuszczalne tylko w niektórych miejscach w instrukcjach SQL. Np. są one dozwolone w *values ()* w instrukcji *INSERT* (aby określić wartości dla kolumn z rzędu), lub w stosunku do kolumny w klauzuli *WHERE* , aby określić wartość porównawczą.

Znaczniki parametrów muszą być powiązane ze zmiennymi aplikacji za pomocą **mysqli_stmt_bind_param ()** przed wykonaniem instrukcji lub pobraniem wierszy.

PHP –MySQLi Prepared Statements

`mysqli_stmt_bind_param ($stmt, $types,&$var1 [,&$var2...])` - Powiązanie zmiennych dla znaczników parametrów w instrukcji SQL przekazanej do mysqli_prepare

\$stmt - identyfikator odsyłacza zwrócony przez `mysqli_prepare()`

\$types - Ciąg zawierający jeden lub więcej znaków określających typy dla odpowiednich zmiennych wiążących:

i - zmienna ma typ integer; **d** -zmienna ma typ double

s- zmienna ma ciąg znaków

Liczba zmiennych i długość łańcucha types musi być zgodna z parametrami w instrukcji.

`mysqli_stmt_bind_param` pobiera wartość zmiennej przez REFERENCJE

PHP –MySQLi Prepared Statements

mysqli_stmt_execute (\$stmt) - Wykonuje przygotowane zapytanie

Po uruchomieniu wszelkie istniejące znaczniki parametrów zostaną automatycznie zastąpione odpowiednimi danymi.

mysqli_stmt_get_result (\$stmt) - Pobiera zestaw wyników z przygotowanej instrukcji

mysqli_stmt_close (\$stmt)

Zamyka przygotowaną instrukcję

PHP –MySQLi Prepared Statements

Korzyści:

- ✓ szkielet zapytania możemy wykorzystać wielokrotnie, podmieniając jedynie wartości zmiennych.
- ✓ informujemy jakiego typu zmienną przesyłamy, więc zwalnia to jednocześnie z ujmowania odpowiednich wartości w cudzysłowy.
- ✓ daje prosty system walidacji, gdyż mechanizm nie przyjmie w parametrze wartości innej, niż określona - nie wyślemy np. stringa do pola zdefiniowanego jako integer.
- ✓ mechanizm ten zabezpiecza przed atakami typu SQL Injection, polegającymi na wstrzykiwaniu do zapytania spreparowanych wartości. Tutaj to nie zadziała, gdyż zmienne nie są wysyłane razem z zapytaniem - są przetwarzane oddzielnie.

PHP –MySQLi Prepared Statements

```
<h2>Dodaj nowy artykuł</h2>
<form method="post" action="php echo $_SERVER['PHP_SELF']; ?" class="form1">
    <fieldset>
        <div class="field">
            <label for="title">Tytuł</label>
            <input type="text" name="title" id="title">
            <label for="content">Treść artykułu</label>
            <textarea name="content" id="content" cols="30" rows="10"></textarea>
            <label for="image">Obrazek</label>
            <input type="text" name="image" id="image">
        </div>
        <input type="submit" class="ui primary button" id="add" name="add" value="Dodaj artykuł">
    </fieldset>
</form>
```

Dodaj nowy artykuł

The screenshot shows a web page with a form for adding a new article. The form consists of three main input fields: 'Tytuł' (Title), 'Treść artykułu' (Article content, represented by a large text area), and 'Obrazek' (Image, represented by a text input). Below the form is a submit button labeled 'Dodaj artykuł'.

Baza blog

PHP –MySQLi Prepared Statements

```
/* dodawanie nowego artykułu */
if ( isset($_POST['add']) ) {
    $title = strip_tags($_POST['title']);
    $content = strip_tags($_POST['content']);
    $image = strip_tags($_POST['image']);
    // przygotowujemy zapytanie
    $sqlQuery2="INSERT article (title,image,content) VALUES (?, ?, ?)";
    $statement = mysqli_prepare($mysqliCon,$sqlQuery2 );
    //przypisujemy wartości parametrów do zapytania
    mysqli_stmt_bind_param($statement,"sss",$title,$image,$content);
    // wykonujemy zapytanie
    mysqli_stmt_execute($statement);
    // zamknięcie zapytania
    mysqli_stmt_close($statement);
    // przeładowanie strony
    header("Location: l10p16.php");
}
```

Baza blog

PHP –MySQLi Prepared Statements w zapytaniach select

```
<?php include('l10p11_2_db_connect.php') ;  
$sqlQuery="SELECT * FROM article ORDER BY id desc";  
$statement = mysqli_prepare($mysqliCon,$sqlQuery);  
// wykonujemy zapytanie  
mysqli_stmt_execute($statement);  
//wyświetlamy wynik  
$result = mysqli_stmt_get_result($statement)  
or die ("błąd w zapytaniu");  
while ( $article = mysqli_fetch_array($result) ) {  
    echo '<article>';  
    echo '<h3>' . $article['title'] . '</h3>';  
    echo '<img src="" . $article['image'] . "" alt="">';  
    echo '<div>';  
    echo '<p>' . $article['content'] . '</p>';  
    echo '</div>';  
    echo '</article>';  
}  
?>
```

Baza blog

Kryteria:

9. Usuwanie danych

PHP – Usuwanie danych

```
<?php
    include('l10p13_db_connect.php');
    // usunięcie rekordu
    $sql = "DELETE FROM uczniowie WHERE id=3";
    if (mysqli_query($mysqliCon, $sql)) {
        echo "Rekord usunięto";
    } else {
        echo "Błąd usuwania: " . mysqli_error($mysqliCon);
    }
    // zamykamy połaczenie
    mysqli_close($mysqliCon);
?>
```

Baza dziennik

PHP – Usuwanie danych

```
<?php
include('l10p13_db_connect.php');
$sql = "SELECT id, imie, nazwisko, email FROM uczniowie;";
$result = mysqli_query($mysqliCon, $sql);
$count = mysqli_num_rows($result);
if ($count > 0) {
    // Dane wyjściowe każdego wiersza
    echo "<table><tr><th>Id</th><th>Imię</th><th>Nazwisko</th>
<th>Email</th><th>Usuwanie</th></tr>";
    while($row = mysqli_fetch_assoc($result)) {
        echo "<tr><td>".$row["id"]."</td><td>".$row["imie"]."</td>";
        echo "<td>".$row["nazwisko"]."</td><td>".$row["email"]."</td>";
        echo "<td><a href='l10p18_del.php?id=".$row["id"].">Usuń</a></td></tr>";
    }
    echo "</table>";
} else {
    echo "0 wyników zapytania";
}
// Usuwamy z pamięci wynik zapytania
mysqli_free_result($result);
// Zamkamy połączenie
mysqli_close($mysqliCon);
?>
```



Id	Imię	Nazwisko	Email	Usuwanie
2	Maria	Dolna	dd@gmail.com	Usuń
4	Anka	Narozna	an@gmail.com	Usuń
6	Marek	Powolny	mp@gmail.com	Usuń

Baza dziennik

PHP – Usuwanie danych

```
<?php
    include('110p13_db_connect.php');
    // usunięcie rekordu

    if (isset($_GET['id']) && is_numeric($_GET['id'])) {
        $id=$_GET['id'];
        $sql = "DELETE FROM uczniowie WHERE id=$id";
        if (mysqli_query($mysqliCon, $sql)) {
            echo "Rekord usunięto";

        } else {
            echo "Błąd usuwania: " . mysqli_error($mysqliCon);
        }
        mysqli_close($mysqliCon);
        header("Location: 110p18.php");
    } else{
        header("Location: 110p18.php");
    }
}
```

Baza dziennik

http://localhost/zsk/l10p18_del.php

PHP – Usuwanie danych

```
<?php include('l10p11_2_db_connect.php');  
$sqlQuery="SELECT * FROM article ORDER BY id desc";  
$result = mysqli_query ($mysqliCon, $sqlQuery)  
or die ("błąd w zapytaniu");  
while ( $article = mysqli_fetch_array($result)) {  
    echo '<article>';  
    echo '<h3>' . $article['title'] . '</h3>';  
    echo '<img src="" . $article['image'] . '" alt="">';  
    echo '<div>';  
    echo '<p>' . $article['content'] . '</p>';  
    echo '</div>';  
    echo '<a href="l10p19_del.php?id=' . $article['id'] . '">';  
    echo '<div class="delLabel">Usuń</div>';  
    echo '</a>';  
    echo '</article>';  
}  
}
```

Baza blog

PHP – Usuwanie danych

```
<?php include('110p11_2_db_connect.php');  
/* usuwanie artykułu */  
if (isset($_GET['id']) && is_numeric($_GET['id'])) {  
    $id=$_GET['id'];  
    // przygotowujemy zapytanie  
    $sqlQuery="DELETE FROM article WHERE id=? LIMIT 1";  
    $statement = mysqli_prepare($mysqliCon,$sqlQuery);  
    // przypisujemy wartości parametrów do zapytania  
    mysqli_stmt_bind_param($statement,"i",$id);  
    // wykonujemy zapytanie  
    mysqli_stmt_execute($statement);  
    // zamknięcie zapytania  
    mysqli_stmt_close($statement);  
    // przeładowanie strony  
    header("Location: 110p19.php");  
}
```

Kryteria:

10. Aktualizacja danych

PHP –Aktualizacja danych

```
<?php
    include('l10p13_db_connect.php');
    //aktualizacja danych
    $sql = "UPDATE uczniowie SET nazwisko='Dolina' WHERE id=1";
    if (mysqli_query($mysqliCon, $sql)) {
        echo "Dane zaktualizowano";
    } else {
        echo "Błąd aktualizacji: " . mysqli_error($mysqliCon);
    }
    // zamykamy połączenie
    mysqli_close($mysqliCon);
?>
```

Baza dziennik

PHP – Aktualizacja danych

```
<?php
include ('110p13_db_connect.php');
$sql = "SELECT id,imie, nazwisko, email FROM uczniowie";
$result = mysqli_query ($mysqliCon, $sql);
$count = mysqli_num_rows ($result);
if ($count > 0) {
    // Dane wyjściowe każdego wiersza
    echo "<table><tr><th>Id</th><th>Imię</th><th>Nazwisko</th>
<th>Email</th><th>Usuwanie</th><th>Edycja</th></tr>";
    while ($row = mysqli_fetch_assoc ($result)) {
        echo "<tr><td>".$row["id"]."</td><td>".$row["imie"]."</td>";
        echo "<td>".$row["nazwisko"]."</td><td>".$row["email"]."</td>";
        echo "<td><a href='110p21_del.php?id=".$row["id"].">Usuń</a></td>";
        echo "<td><a href='110p21_edit.php?id=".$row["id"].">Edytuj</a></td></tr>";
    }
}
```

Id	Imię	Nazwisko	Email	Usuwanie	Edycja
2	Maria	Dolna	dd@gmail.com	Usuń	Edytuj
4	Anka	Narozna	an@gmail.com	Usuń	Edytuj
6	Marek	Pawlak	mpaw@gmail.com	Usuń	Edytuj

Baza dziennik

PHP –Aktualizacja danych

```
<?php
include ('l10p13_db_connect.php') ;
// aktualizacja rekordu
if (isset($_GET['id']) && is_numeric($_GET['id'])) {
    $id=$_GET['id'];
    $sql = "SELECT * FROM uczniowie WHERE id=$id";
    $result = mysqli_query($mysqliCon, $sql);
    $count = mysqli_num_rows($result);
    if ($count > 0) {
        while ($row=mysqli_fetch_assoc($result)) {
            ?>
```

Baza dziennik

PHP – Aktualizacja danych

```
<form method="post" action="l10p21_apd.php" class="form1">
<fieldset>
    <label for="idF">Id: <?php echo $row['id'];?></label>
    <input type="hidden" name="idF" value="<?php echo $row['id'];?>">
    <br><br>
    <label for="fname">Imię:</label>
    <input type="text" name="fname" value="<?php echo $row['imie'];?>">
    <br><br>
    <label for="sname">Nazwisko:</label>
    <input type="text" name="sname" value="<?php echo $row['nazwisko'];?>">
    <br><br>
    <label for="kod_post">Adres email:</label>
    <input type="text" name="email" value="<?php echo $row['email'];?>">
    <br><br>
    <input type="submit" name="send" value="zaktualizuj dane">
</fieldset>
</form>
```

PHP – Aktualizacja danych

```
<?php  
    }  
} else {  
    echo "Brak danych do wyświetlenia.";  
}  
mysqli_close($mysqliCon);  
else {  
    header("Location: l10p21.php");  
}  
?>
```

Id: 2

Imię:

Nazwisko:

Adres email:

Baza dziennik

PHP –Aktualizacja danych

```
<?php
if ( isset( $_POST['send'])) {
    include('l10p13_db_connect.php');
    $idU=$_POST['idF'];
    $imieU=strip_tags($_POST['fname']);
    $nazwiskoU=strip_tags($_POST['surname']);
    $emailU=strip_tags($_POST['email']);
    //modyfikacja danych
    $sql = "UPDATE uczniowie SET imie='$imieU',
    nazwisko='$nazwiskoU', email='$emailU' WHERE id=$idU";
    mysqli_query($mysqliCon, $sql);
    mysqli_close($mysqliCon);
}
header("Location: l10p21.php");
```

Baza dziennik

PHP – połączenia z bazą danych, podsumowanie

Zadanie:

Wykorzystując przykłady l10p21.php , l10p16_2.php i l10p19.php przygotuj skrypty pozwalające na aktualizację wpisów blogowych z przykładu 19. Zastosuj MySQLi Prepared Statements