

Łączenie z bazą danych

Spis treści

Teoria – podstawowe info o funkcjach do łączenia z bazą danych na mysqli	1
BAZA DANYCH - ćwiczenia.....	5
Podstawowa strona.....	8
ĆWICZENIE 1.....	12
Ćwiczenie 2.....	15
Ćwiczenie 3.....	17
Ćwiczenie 4.....	22
Ćwiczenie 5.....	26
Ćwiczenie 6.....	33

Teoria – podstawowe info o funkcjach do łączenia z bazą danych na mysqli

1. Łączenie z bazą danych: mysqli_connect

```
$servername = "localhost";
$username = "nazwa_uzytkownika";
$password = "haslo";
$dbname = "nazwa_bazy";

$conn = mysqli_connect($servername, $username, $password, $dbname);

if (!$conn) {
    die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
}
```

Funkcja `mysqli_connect` tworzy połączenie z bazą danych MySQL. W przypadku błędu, `mysqli_connect_error` zwraca opis błędu.

kod

```
$servername = "localhost";
$username = "nazwa_uzytkownika";
$password = "haslo";
$dbname = "nazwa_bazy";

$conn = mysqli_connect($servername, $username, $password, $dbname);

if (!$conn) {
    die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
}
```

2. Wykonywanie zapytań SQL: `mysqli_query`

```
$sql = "SELECT * FROM tabela";  
$result = mysqli_query($conn, $sql);
```

Funkcja `mysqli_query` wykonuje zapytanie do bazy danych. Parametry to połączenie (`$conn`) i zapytanie SQL (`$sql`). Wynik zapytania jest przechowywany w zmiennej `$result`.

3. Wyświetlanie wszystkich rekordów z bazy danych:

przykłady różnych metod fetchowania rekordów do tablicy jednowymiarowej:

1. **`mysqli_fetch_row()`**: Pobiera następny wiersz z wyniku zapytania i zwraca go jako tablicę indeksowaną liczbami całkowitymi (tablica jednowymiarowa), prościej wyjmowanie liczbowe numerowane od 0
2. **`mysqli_fetch_array()`** Podobnie jak `mysqli_fetch_row()`, ale zwraca tablicę indeksowaną liczbami całkowitymi i asocjacyjną, zarówno po numerach jak i nazwy szuflad
3. **`mysqli_fetch_assoc()`**: Pobiera następny wiersz z wyniku zapytania i zwraca go jako tablicę asocjacyjną (indeksowaną nazwami kolumn), prościej- wyjmowanie asocjacyjne czyli skojarzeniowe, podaj nazwy szufladki tablicy jednowymiarowej, nadając taką samą nazwę jakiej noszą nazwę kolumny w bazie danych

Przykład dla funkcji `mysqli_query` i `mysqli_fetch_assoc`

```
$sql = "SELECT * FROM tabela";  
$result = mysqli_query($conn, $sql);  
  
while ($row = mysqli_fetch_assoc($result)) {  
    echo "ID: " . $row['id'] . " - Kolumna: " . $row['kolumna'] . "<br>";  
}
```

4. Pobieranie danych z bazy danych:

Przykład dla funkcji `mysqli_fetch_assoc`

```
$sql = "SELECT * FROM tabela WHERE id = 1";  
$result = mysqli_query($conn, $sql);  
  
if ($row = mysqli_fetch_assoc($result)) {  
    echo "ID: " . $row['id'] . " - Kolumna: " . $row['kolumna'];  
} else {  
    echo "Brak wyników.";  
}
```

Funkcja `mysqli_fetch_assoc` jest używana do pobierania jednego wiersza wyników.

Funkcje `mysqli_fetch_assoc` i `mysqli_fetch_array` są używane do pobierania wiersza wyników z zapytania SQL wykonanego za pomocą MySQLi w PHP. Oto główne różnice między nimi:

`mysqli_fetch_assoc`:

- Zwraca wiersz wyników jako asocjacyjny array (tablicę, gdzie kluczami są nazwy kolumn).
- Dostarcza bardziej czytelne dane, ponieważ można odwoływać się do wartości za pomocą nazw kolumn.

Przykład:

```
$row = mysqli_fetch_assoc($result);  
echo $row['kolumna1'];
```

`mysqli_fetch_array`:

- Zwraca wiersz wyników jako array indeksowany numerycznie i asocjacyjny (czyli zawiera zarówno indeksy numeryczne, jak i nazwy kolumn jako klucze).
- Daje większą elastyczność, ponieważ można odwoływać się do wartości za pomocą indeksów numerycznych lub nazw kolumn.
- Domyślnie, `mysqli_fetch_array` zwraca obiekt typu array.

Przykład

```
$row = mysqli_fetch_array($result);  
echo $row['kolumna1']; // Odwołanie za pomocą nazwy kolumny  
echo $row[0];          // Odwołanie za pomocą indeksu numerycznego
```

W przypadku, gdy interesują Cię tylko nazwy kolumn i preferujesz bardziej czytelny kod, `mysqli_fetch_assoc` może być bardziej intuicyjny. Jeśli jednak potrzebujesz jednocześnie dostępu do danych za pomocą indeksów numerycznych i nazw kolumn, `mysqli_fetch_array` może być bardziej elastyczny.

5. Wstawianie danych do bazy danych:

mysqli_query z poleceniem INSERT

```
$nowa_wartosc = "Nowa Wartość";  
  
$sql_insert = "INSERT INTO tabela (kolumna) VALUES ('$nowa_wartosc')";  
  
if (mysqli_query($conn, $sql_insert)) {  
    echo "Rekord został dodany pomyślnie.";  
} else {  
    echo "Błąd podczas dodawania rekordu: " . mysqli_error($conn);  
}
```

Funkcja `mysqli_query` może być używana do wykonania polecenia `INSERT` w celu wstawienia nowego rekordu.

6. Zabezpieczanie przed atakami SQL injection:

`mysqli_real_escape_string`

```
$wartosc_do_zabezpieczenia = "Wartość do zabezpieczenia";  
$zabezpieczona_wartosc = mysqli_real_escape_string($conn, $wartosc_do_zabezpieczenia);
```

Funkcja `mysqli_real_escape_string` zabezpiecza dane przed atakami SQL injection, eliminując specjalne znaki.

7. Funkcja `mysqli_num_rows` jest używana w PHP w połączeniu z MySQLi do zwrócenia liczby wierszy (rekordów) w wynikach zapytania SQL. Oto podstawowe informacje o tej funkcji:

`mysqli_num_rows`

- **Opis:**

- Zwraca liczbę wierszy w wynikach zapytania.
- Jest często używana do sprawdzenia, czy wyniki zapytania zawierają jakiekolwiek dane, czy też są puste.
- Najczęściej używana w połączeniu z funkcją `mysqli_query` po wykonaniu zapytania SQL.

```
$sql = "SELECT * FROM tabela";  
$result = mysqli_query($conn, $sql);  
  
if (mysqli_num_rows($result) > 0) {  
    // Jest co najmniej jeden wiersz wyników  
    while ($row = mysqli_fetch_assoc($result)) {  
        // Przetwarzanie wiersza  
    }  
} else {  
    // Brak wyników  
    echo "Brak wyników."  
}
```

- **Zwracana wartość:**

- Liczba całkowita reprezentująca ilość wierszy w wynikach zapytania.

Funkcja ta jest przydatna do sprawdzenia, czy wyniki zapytania zawierają jakiekolwiek dane przed przystąpieniem do przetwarzania wyników. Działa zarówno dla zapytań `SELECT`, jak i innych zapytań, które mogą zwracać dane (np. `UPDATE`, `DELETE`).

- **Funkcja `mysqli_insert_id`**

jest funkcją MySQLi w PHP, która zwraca ID ostatnio wstawionego wiersza do tabeli, które posiada kolumnę z atrybutem `AUTO_INCREMENT`. Funkcja ta jest często używana po wykonaniu zapytania `INSERT`, aby uzyskać wartość ID dla nowego rekordu.

Składnia:

```
mysqli_insert_id(mysqli $conn);
```

```
// Przykład zapytania INSERT
$sql = "INSERT INTO tabela (kolumna1, kolumna2) VALUES ('wartosc1', 'wartosc2')";
$result = mysqli_query($conn, $sql);

// Sprawdzenie, czy zapytanie INSERT było udane
if ($result) {
    // Pobranie ID ostatnio wstawionej wiersza
    $last_insert_id = mysqli_insert_id($conn);
    echo "ID ostatnio wstawionej wiersza: " . $last_insert_id;
} else {
    echo "Błąd podczas zapytania INSERT: " . mysqli_error($conn);
}
```

W powyższym przykładzie, po wykonaniu zapytania `INSERT`, funkcja `mysqli_insert_id` jest używana do uzyskania ID ostatnio wstawionej wiersza. Pamiętaj, że tę funkcję można używać tylko po udanym zapytaniu `INSERT` do tabeli z atrybutem `AUTO_INCREMENT`.

8. Zamykanie połączenia:

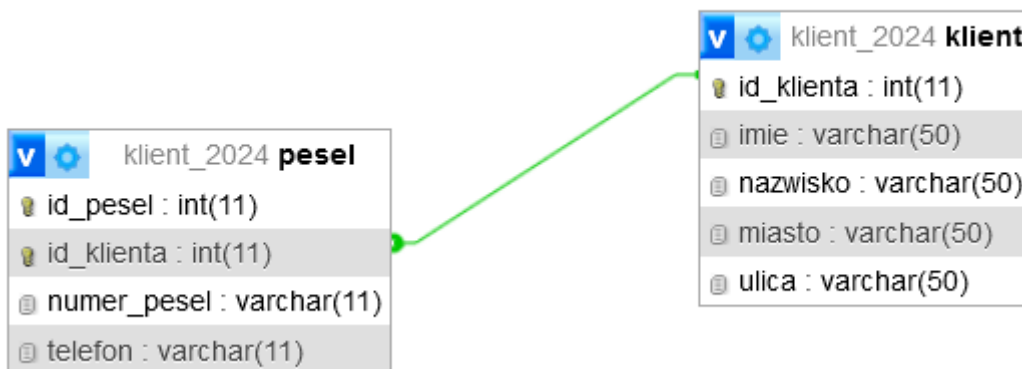
```
mysqli_close
```

```
mysqli_close($conn);
```

Funkcja `mysqli_close` zamyka połączenie z bazą danych.

BAZA DANYCH - ćwiczenia

- 1) Utwórz bazę danych o nazwie `klient_2024` a w nim poniższe tabele:



- 2) Wstaw przykładowe rekordy do obu tabel.

```
-- Tworzenie tabeli Klient
CREATE TABLE Klient (
    id_klienta INT PRIMARY KEY AUTO_INCREMENT,
    imie VARCHAR(50) NOT NULL,
    nazwisko VARCHAR(50) NOT NULL,
    miasto VARCHAR(50) NOT NULL,
    ulica VARCHAR(50) NOT NULL
);

-- Dodawanie 15 rekordów
INSERT INTO Klient (imie, nazwisko, miasto, ulica) VALUES
('Jan', 'Nowak', 'Warszawa', 'Aleje Jerozolimskie 234'),
('Anna', 'Kowalska', 'Kraków', 'ul. Floriańska 56'),
('Piotr', 'Zieliński', 'Gdańsk', 'ul. Długa 78'),
('Katarzyna', 'Wójcik', 'Wrocław', 'pl. Solny 9'),
('Michał', 'Lis', 'Poznań', 'ul. Świętosławska 33'),
('Monika', 'Kowalczyk', 'Łódź', 'ul. Piotrkowska 67'),
('Grzegorz', 'Adamczyk', 'Szczecin', 'ul. Monte Cassino 22'),
('Aleksandra', 'Szymańska', 'Katowice', 'ul. Warszawska 44'),
('Marcin', 'Woźniak', 'Bydgoszcz', 'ul. Dworcowa 10'),
('Kinga', 'Duda', 'Lublin', 'ul. Krakowskie Przedmieście 20'),
('Kamil', 'Mazurek', 'Gdynia', 'ul. Świętojańska 32'),
('Natalia', 'Jankowska', 'Rzeszów', 'ul. 3 Maja 15'),
('Artur', 'Witkowski', 'Olsztyn', 'ul. Pieniężnego 8'),
('Sylwia', 'Michalska', 'Częstochowa', 'ul. Nadrzeczna 35'),
('Tomasz', 'Kaczmarek', 'Radom', 'ul. Żeromskiego 20');

-- Tworzenie tabeli Pesel
CREATE TABLE Pesel (
    id_pesel INT PRIMARY KEY AUTO_INCREMENT,
    id_klienta INT UNIQUE,
    numer_pesel VARCHAR(11) NOT NULL,
    telefon VARCHAR(11),
    FOREIGN KEY (id_klienta) REFERENCES Klient(id_klienta)
);

-- Wypełnianie tabeli Pesel przykładowymi danymi (15 rekordów)
INSERT INTO Pesel (id_klienta, numer_pesel, telefon) VALUES
(1, '92030112345', '123456789'),
(2, '88051256789', '987654321'),
(3, '75063098765', '555666777'),
(4, '91071543210', '111222333'),
(5, '83022487654', '999888777'),
(6, '92050365432', '444555666'),
```

```
(7, '87071010987', '333222111'),
(8, '96021534567', '777888999'),
(9, '88070467890', '123123123'),
(10, '85031598761', '456456456'),
(11, '90052323456', '789789789'),
(12, '95060787654', '321321321'),
(13, '89082634567', '654654654'),
(14, '92041210987', '987987987'),
(15, '80091023456', '654321987');
```

Gotowe skrypty:

-- Tworzenie tabeli Klient

```
CREATE TABLE Klient (
    id_klienta INT PRIMARY KEY AUTO_INCREMENT,
    imie VARCHAR(50) NOT NULL,
    nazwisko VARCHAR(50) NOT NULL,
    miasto VARCHAR(50) NOT NULL,
    ulica VARCHAR(50) NOT NULL
);
```

-- Dodawanie 15 rekordów

```
INSERT INTO Klient (imie, nazwisko, miasto, ulica) VALUES
('Jan', 'Nowak', 'Warszawa', 'Aleje Jerozolimskie 234'),
('Anna', 'Kowalska', 'Kraków', 'ul. Floriańska 56'),
('Piotr', 'Zieliński', 'Gdańsk', 'ul. Długa 78'),
('Katarzyna', 'Wójcik', 'Wrocław', 'pl. Solny 9'),
('Michał', 'Lis', 'Poznań', 'ul. Świętosławska 33'),
('Monika', 'Kowalczyk', 'Łódź', 'ul. Piotrkowska 67'),
('Grzegorz', 'Adamczyk', 'Szczecin', 'ul. Monte Cassino 22'),
('Aleksandra', 'Szymańska', 'Katowice', 'ul. Warszawska 44'),
('Marcin', 'Woźniak', 'Bydgoszcz', 'ul. Dworcowa 10'),
('Kinga', 'Duda', 'Lublin', 'ul. Krakowskie Przedmieście 20'),
('Kamil', 'Mazurek', 'Gdynia', 'ul. Świętojańska 32'),
('Natalia', 'Jankowska', 'Rzeszów', 'ul. 3 Maja 15'),
('Artur', 'Witkowski', 'Olsztyn', 'ul. Pieniężnego 8'),
('Sylwia', 'Michalska', 'Częstochowa', 'ul. Nadrzeczna 35'),
```

```
('Tomasz', 'Kaczmarek', 'Radom', 'ul. Żeromskiego 20');

-- Tworzenie tabeli Pesel

CREATE TABLE Pesel (

    id_pesel INT PRIMARY KEY AUTO_INCREMENT,

    id_klienta INT UNIQUE,

    numer_pesel VARCHAR(11) NOT NULL,

    telefon VARCHAR(11),

    FOREIGN KEY (id_klienta) REFERENCES Klient(id_klienta)

);

-- Wypełnianie tabeli Pesel przykładowymi danymi (15 rekordów)

INSERT INTO Pesel (id_klienta, numer_pesel, telefon) VALUES

(1, '92030112345', '123456789'),

(2, '88051256789', '987654321'),

(3, '75063098765', '555666777'),

(4, '91071543210', '111222333'),

(5, '83022487654', '999888777'),

(6, '92050365432', '444555666'),

(7, '87071010987', '333222111'),

(8, '96021534567', '777888999'),

(9, '88070467890', '123123123'),

(10, '85031598761', '456456456'),

(11, '90052323456', '789789789'),

(12, '95060787654', '321321321'),

(13, '89082634567', '654654654'),

(14, '92041210987', '987987987'),

(15, '80091023456', '654321987');
```

Podstawowa strona

kod:

Ćwiczenia z łączenia z baz danych

1. Ćwiczenie 1
2. Ćwiczenie 2
3. Ćwiczenie 3
4. Ćwiczenie 4
5. Ćwiczenie 5

Wyświetlanie danych z bazy

Tutaj umieść treść danych z bazy.

Opracował: Imię Nazwisko

Styl. css

```
body {  
    font-family: Arial, sans-serif;  
    margin: 0;  
    padding: 0;  
}  
  
header {  
    background-color: #3498db;  
    color: white;  
    text-align: center;  
    padding: 20px;  
}  
  
header {  
    background-color: #3498db;  
    color: white;  
    text-align: center;  
    padding: 20px;  
}
```

```
main {
  display: flex;
  justify-content: space-between;
  margin: 20px;
}

aside {
  width: 25%;
  background-color: #2ecc71;
  padding: 20px;
  float:left;
}

section {
  width: 70%;
  background-color: #e74c3c;
  padding: 20px;
  float:right;
}

footer {
  background-color: #34495e;
  color: white;
  text-align: center;
  padding: 10px;
  width: 100%;
}

table {
  border-collapse: collapse;
  width: 100%;
}

th, td {
  border: 1px solid #dddddd;
  text-align: left;
  padding: 8px;
}
```

```

    }

    th {
        background-color: #f2f2f2;
    }

```

Strona:

```

<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <title>Podstawowa Strona</title>
    <link rel="stylesheet" type="text/css" href="styl.css">
</head>
<body>

<header>
    <h1>Ćwiczenia z łączenia z baz danych</h1>
</header>

<main>
    <aside>
        <ol>
            <li><a href="1.php">Ćwiczenie 1</a></li>
            <li>Ćwiczenie 2</li>
            <li>Ćwiczenie 3</li>
            <li>Ćwiczenie 4</li>
            <li>Ćwiczenie 5</li>
        </ol>
    </aside>

    <section>
        <h2>Wyświetlanie danych z bazy</h2>
        <p>Tutaj umieść treść danych z bazy.</p>
    </section>
</main>

<footer>
    <h3>Opracował: Imię Nazwisko</h3>
</footer>

```

```
</body>
</html>
```

ĆWICZENIE 1

Utwórz plik o nazwie `cwiczenie1.php` a w nim napisz skrypt, który wyświetli wszystkie rekordy z tabeli klient z bazy klient_2024 w formie tabeli;

Ćwiczenia z łączenia z baz danych

1. Ćwiczenie 1
2. Ćwiczenie 2
3. Ćwiczenie 3
4. Ćwiczenie 4
5. Ćwiczenie 5

Ćwiczenie 1

Wyświetlanie danych w postaci tabeli

ID Klienta	Imię	Nazwisko	Miasto	Ulica
1	Jan	Nowak	Warszawa	Aleje Jerozolimskie 234
2	Anna	Kowalska	Kraków	ul. Floriańska 56
3	Piotr	Zieliński	Gdańsk	ul. Długa 78
4	Katarzyna	Wójcik	Wrocław	pl. Solny 9
5	Michał	Lis	Poznań	ul. Świętosławska 33
6	Monika	Kowalczyk	Łódź	ul. Piotrkowska 67
7	Grzegorz	Adamczyk	Szczecin	ul. Monte Cassino 22
8	Aleksandra	Szymańska	Katowice	ul. Warszawska 44
9	Marcin	Woźniak	Bydgoszcz	ul. Dworcowa 10
10	Kinga	Duda	Lublin	ul. Krakowskie Przedmieście 20
11	Kamil	Mazurek	Gdynia	ul. Świętojańska 32
12	Natalia	Jankowska	Rzeszów	ul. 3 Maja 15
13	Artur	Witkowski	Olsztyn	ul. Pieniężnego 8
14	Sylvia	Michalska	Częstochowa	ul. Nadrzeczna 35
15	Tomasz	Kaczmarek	Radom	ul. Zeromskiego 20

Opracował: Imię Nazwisko

Rysunek 1. wygląd jak powinny wyświetlać się dane

Skrypt na pobieranie danych w formie tabeli:

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "klient_2024";

// Tworzenie połączenia z bazą danych
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Sprawdzanie połączenia
if (!$conn) {
    die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
}

// Zapytanie SQL do pobrania danych z tabeli Klient
```

```

$sql = "SELECT * FROM Klient";
$result = mysqli_query($conn, $sql);

// Sprawdzanie czy są dane do wyświetlenia
if (mysqli_num_rows($result) > 0) {
    echo "<table>
        <tr>
            <th>ID Klienta</th>
            <th>Imię</th>
            <th>Nazwisko</th>
            <th>Miasto</th>
            <th>Ulica</th>
        </tr>";

    // Wyświetlanie danych w tabeli
    while ($row = mysqli_fetch_assoc($result)) {
        echo "<tr>
            <td>" . $row["id_klienta"] . "</td>
            <td>" . $row["imie"] . "</td>
            <td>" . $row["nazwisko"] . "</td>
            <td>" . $row["miasto"] . "</td>
            <td>" . $row["ulica"] . "</td>
        </tr>";
    }

    echo "</table>";
} else {
    echo "Brak danych do wyświetlenia.";
}

// Zamykanie połączenia z bazą danych
mysqli_close($conn);
?>

```

Wyjaśnienie kodu do ćwiczenia 1:

Powyższy kod PHP służy do nawiązania połączenia z bazą danych MySQLi, wykonania zapytania SQL do pobrania danych z tabeli Klient, a następnie wyświetlenia tych danych w formie tabeli HTML. Oto krótkie wyjaśnienie poszczególnych części kodu:

1) Tworzenie połączenia z bazą danych:

```

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "klient_2024";

$conn = mysqli_connect($servername, $username, $password, $dbname);

```

\$servername:

- Adres serwera bazy danych.
- \$username: Nazwa użytkownika bazy danych.
- \$password: Hasło do bazy danych.
- \$dbname: Nazwa konkretnej bazy danych.
- mysqli_connect: Funkcja, która otwiera połączenie z bazą danych.

2) Sprawdzanie połączenia:

```

if (!$conn) {
    die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
}

```

Jeśli połączenie nie powiedzie się, funkcja `mysqli_connect_error()` zwróci opis błędu i skrypt zakończy działanie.

3) Zapytanie SQL do pobrania danych:

```

$sql = "SELECT * FROM Klient";
$result = mysqli_query($conn, $sql);

```

- \$sql: Zmienna przechowująca zapytanie SQL.
- mysqli_query: Funkcja, która wykonuje zapytanie na bazie danych.

4) Sprawdzanie czy są dane do wyświetlenia:

```

if (mysqli_num_rows($result) > 0) {

```

`mysqli_num_rows`: Funkcja zwracająca liczbę wierszy w wyniku zapytania. Jeśli liczba wierszy jest większa od zera, to są dostępne dane do wyświetlenia.

5) Wyświetlanie danych w tabeli HTML:

```

echo "<table>
    <tr>
        <th>ID Klienta</th>
        <th>Imię</th>
        <th>Nazwisko</th>
        <th>Miasto</th>
        <th>Ulica</th>
    </tr>";
while ($row = mysqli_fetch_assoc($result)) {
    echo "<tr>
        <td>" . $row["id_klienta"] . "</td>
        <td>" . $row["imie"] . "</td>
        <td>" . $row["nazwisko"] . "</td>
        <td>" . $row["miasto"] . "</td>
        <td>" . $row["ulica"] . "</td>
    </tr>";
}
echo "</table>";

```

- `mysqli_fetch_assoc`: Funkcja, która pobiera kolejny wiersz z wyniku zapytania jako tablicę asocjacyjną.
- Dane są wyświetlane w formie tabeli HTML.

6) Zamykanie połączenia z bazą danych:

```
mysqli_close($conn);
```

`mysqli_close`: Funkcja zamykająca połączenie z bazą danych.

W rezultacie ten skrypt połączy się z bazą danych, pobierze dane z tabeli Klient, a następnie wyświetli te dane w formie tabeli HTML. Jeśli nie ma danych, wyświetli komunikat "Brak danych do wyświetlenia."

Ćwiczenie 2

Utwórz plik 2.php, w którym wyświetlisz rekordy z bazy danych w postaci listy punktowej, która wyświetla dane z dwóch tabel: imie i nazwisko z tabeli klient i z tabeli pesel: pesel i telefon.

Ćwiczenia z łączenia z baz danych

Ćwiczenie 2

Wyswietlanie danych w postaci listy punktowanej i łączenie tabel w mysql

```
SELECT Klient.imie, Klient.nazwisko, Pesel.numer_pesel, Pesel.telefon FROM Klient INNER JOIN Pesel ON Klient.id_klienta = Pesel.id_klienta
```

- Jan Nowak - PESEL: 92030112345, Telefon: 123456789
- Anna Kowalska - PESEL: 88051256789, Telefon: 987654321
- Piotr Zieliński - PESEL: 75063098765, Telefon: 555666777
- Katarzyna Wójcik - PESEL: 91071543210, Telefon: 11222333
- Michał Lis - PESEL: 83022487654, Telefon: 999888777
- Monika Kowalczyk - PESEL: 92050365432, Telefon: 444555666
- Grzegorz Adamczyk - PESEL: 87071010987, Telefon: 333222111
- Aleksandra Szymańska - PESEL: 96021534567, Telefon: 777888999
- Marcin Woźniak - PESEL: 88070467890, Telefon: 123123123
- Kinga Duda - PESEL: 85031598761, Telefon: 456456456
- Kamil Mazurek - PESEL: 90052323456, Telefon: 789789789
- Natalia Jankowska - PESEL: 95060787654, Telefon: 321321321
- Artur Witkowski - PESEL: 89082634567, Telefon: 654654654
- Sylwia Michalska - PESEL: 92041210987, Telefon: 987987987
- Tomasz Kaczmarek - PESEL: 80091023456, Telefon: 654321987

Opracował: Imię Nazwisko

Kod w php

```
<h2>Ćwiczenie 2</h2>
<p>Wyswietlanie danych w postaci listy punktowanej i łączenie tabel w
mysql</p>
<h3> SELECT Klient.imie, Klient.nazwisko, Pesel.numer_pesel, Pesel.telefon
FROM Klient
INNER JOIN Pesel ON Klient.id_klienta = Pesel.id_klienta</h3>

<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "klient_2024";

$conn = mysqli_connect($servername, $username, $password, $dbname);

if (!$conn) {
    die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
}

$sql = "SELECT Klient.imie, Klient.nazwisko, Pesel.numer_pesel, Pesel.telefon
FROM Klient
INNER JOIN Pesel ON Klient.id_klienta = Pesel.id_klienta";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
```



```

    echo "<ul>";
    while ($row = mysqli_fetch_assoc($result)) {
        echo "<li>" . $row["imie"] . " " . $row["nazwisko"] . " - PESEL: " .
$row["numer_pesel"] . ", Telefon: " . $row["telefon"] . "</li>";
    }
    echo "</ul>";
} else {
    echo "Brak danych do wyświetlenia.";
}

mysqli_close($conn);
?>

```

Wyjaśnienie: while (\$row = mysqli_fetch_assoc(\$result)) {

- Iteruje przez każdy wiersz wyniku zapytania i zwraca go jako tablicę asocjacyjną (indeksowana nazwami kolumn).

Ćwiczenie 3

Wstawianie rekordów do bazy przez formularz.



Ćwiczenie 3

Wstawianie danych

Imię:

Nazwisko:

Miasto:

Ulica:

Numer PESEL:

Telefon:

Skrypt:

```

<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "klient_2024";

```

```

$conn = mysqli_connect($servername, $username, $password, $dbname);

// Sprawdzenie połączenia
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Obsługa formularza
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Pobranie danych z formularza
    $imie = $_POST['imie'];
    $nazwisko = $_POST['nazwisko'];
    $miasto = $_POST['miasto'];
    $ulica = $_POST['ulica'];
    $numer_pesel = $_POST['numer_pesel'];
    $telefon = $_POST['telefon'];

    // Wstawienie danych do tabeli Klient
    $sql_klient = "INSERT INTO Klient (imie, nazwisko, miasto, ulica) VALUES
('$imie', '$nazwisko', '$miasto', '$ulica')";
    if (mysqli_query($conn, $sql_klient)) {
        $last_id_klient = mysqli_insert_id($conn);

        // Wstawienie danych do tabeli Pesel
        $sql_pesel = "INSERT INTO Pesel (id_klienta, numer_pesel, telefon) VALUES
('$last_id_klient', '$numer_pesel', '$telefon')";
        if (!mysqli_query($conn, $sql_pesel)) {
            echo "Error: " . $sql_pesel . "<br>" . mysqli_error($conn);
        }
    } else {
        echo "Error: " . $sql_klient . "<br>" . mysqli_error($conn);
    }
}

mysqli_close($conn);
?>

<!DOCTYPE html>
<html lang="pl">
<head>

```

```

<meta charset="UTF-8">
<title>Podstawowa Strona</title>
<link rel="stylesheet" type="text/css" href="styl.css">
</head>
<body>

<header>
    <h1>Ćwiczenia z łączenia z baz danych</h1>
</header>

<main>
    <aside>
        <ol>
            <li><a href="1.php">Ćwiczenie 1</a></li>
            <li><a href="2.php">Ćwiczenie 2</a></li>
            <li><a href="3.php">Ćwiczenie 3</a></li>
            <li>Ćwiczenie 4</li>
            <li>Ćwiczenie 5</li>
        </ol>
    </aside>

    <section>
        <h2>Ćwiczenie 3</h2>
        <p>Wstawianie danych</p>

        <form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
        Imię: <input type="text" name="imie" required><br>
        Nazwisko: <input type="text" name="nazwisko" required><br>
        Miasto: <input type="text" name="miasto" required><br>
        Ulica: <input type="text" name="ulica" required><br>
        Numer PESEL: <input type="text" name="numer_pesel" maxlength="11"
required><br>
        Telefon: <input type="text" name="telefon" maxlength="11"><br>
        <input type="submit" value="Dodaj">
        </form>

    </section>
</main>

<footer>
    <h3>Opracował: Imię Nazwisko</h3>

```

```
</footer>
```

```
</body>
```

```
</html>
```

Wyjaśnienie kodu:

`action="<?php echo $_SERVER['PHP_SELF']; ?>"` oznacza, że dane z formularza zostaną wysłane do tego samego skryptu, który obsługuje dane wejściowe. `$_SERVER['PHP_SELF']` to zmienna globalna w PHP, która zawiera nazwę bieżącego skryptu, który jest wykonywany. W kontekście formularza HTML, używamy tej zmiennej jako wartości atrybutu `action` w tagu formularza. Działa to w taki sposób, że formularz wysyła dane z powrotem do tego samego skryptu PHP, który obsługuje dane wejściowe.

1. Sprawdzenie metody żądania:

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

Ten warunek sprawdza, czy żądanie wysłane do serwera jest metodą POST. Oznacza to, że formularz został zatwierdzony, a dane zostały wysłane do serwera.

2. Pobranie danych z formularza:

```
$imie = $_POST['imie'];  
$nazwisko = $_POST['nazwisko'];  
$miasto = $_POST['miasto'];  
$ulica = $_POST['ulica'];  
$numer_pesel = $_POST['numer_pesel'];  
$telefon = $_POST['telefon'];
```

Te linie kodu pobierają dane przesłane z formularza za pomocą metody POST. Każda zmienna (`$imie`, `$nazwisko`, itd.) odpowiada polu formularza o określonej nazwie.

3. Wstawienie danych do tabeli Klient:

```
$sql_klient = "INSERT INTO Klient (imie, nazwisko, miasto, ulica) VALUES ('$imie',  
'$nazwisko', '$miasto', '$ulica')";
```

```
if (mysqli_query($conn, $sql_klient)) {
```

Sprawdza się, czy zapytanie zostało wykonane poprawnie (`mysqli_query($conn, $sql_klient)`).

4. Pobranie ostatniego ID klienta:

```
$last_id_klient = mysqli_insert_id($conn);
```

Pobiera ostatnio wstawione ID klienta w tabeli `Klient` za pomocą funkcji `mysqli_insert_id`.

5. Wstawienie danych do tabeli Pesel:

```
$sql_pesel = "INSERT INTO Pesel (id_klienta, numer_pesel, telefon) VALUES ('$last_id_klient', '$numer_pesel', '$telefon')";
```

```
if (!mysqli_query($conn, $sql_pesel)) {  
    echo "Error: " . $sql_pesel . "<br>" . mysqli_error($conn);  
}
```

- Tworzy się zapytanie SQL (`$sql_pesel`), które wstawia dane do tabeli `Pesel`.
- Sprawdzane jest, czy zapytanie zostało wykonane poprawnie (`mysqli_query($conn, $sql_pesel)`).
- W przypadku błędu, wyświetla komunikat z informacją o błędzie.

6. Obsługa błędów w przypadku problemów z zapytaniami SQL:

```
} else {  
    echo "Error: " . $sql_klient . "<br>" . mysqli_error($conn);  
}
```

Jeśli zapytanie do tabeli `Klient` nie zostało wykonane poprawnie, wyświetla komunikat o błędzie.

Dodatkowe ustawienia do walidacji formularza html:

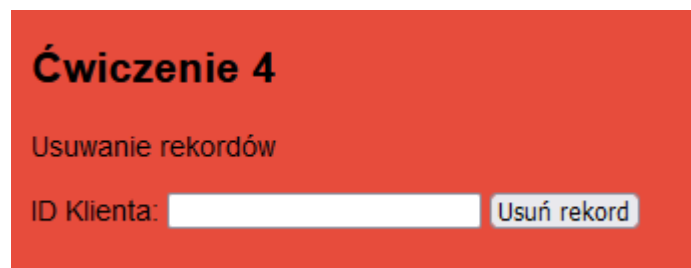
```
<form method="post" action="<?php echo htmlspecialchars($_SERVER['PHP_SELF']);  
?>">  
    Imię: <input type="text" name="imie" required><br>  
    Nazwisko: <input type="text" name="nazwisko" required><br>  
    Miasto: <input type="text" name="miasto" required><br>  
    Ulica: <input type="text" name="ulica" required><br>  
    Numer PESEL: <input type="text" name="numer_pesel" pattern="\d{11}"  
maxlength="11" required title="Wprowadź 11 cyfr PESEL"><br>  
    Telefon: <input type="text" name="telefon" pattern="\d{9,11}" maxlength="11"  
title="Wprowadź 9-11 cyfr telefonu"><br>  
    <input type="submit" value="Dodaj">  
</form>
```

Wyjaśnienie:

1. `htmlspecialchars($_SERVER['PHP_SELF'])` w atrybucie `action` dla bezpieczeństwa, aby uniknąć ataków XSS.
2. W polach "Numer PESEL" i "Telefon" atrybut `pattern` z wyrażeniem regularnym, które definiuje oczekiwany format danych.
3. Atrybut `title`, który pojawi się jako podpowiedź, gdy użytkownik wprowadzi dane w nieprawidłowym formacie.

Ćwiczenie 4

Utwórz formularz, który po podaniu `id_klienta` usunie go z bazy;



Ćwiczenie 4

Usuwanie rekordów

ID Klienta:

Tylko przed wykonaniem tego zadania w bazie danych ustaw kaskadowe usuwanie danych i aktualizacje.

```
ALTER TABLE `pesel` DROP FOREIGN KEY `pesel_ibfk_1`; ALTER TABLE `pesel` ADD CONSTRAINT `pesel_ibfk_1` FOREIGN KEY (`id_klienta`) REFERENCES `klient` (`id_klienta`) ON DELETE CASCADE ON UPDATE CASCADE;
```

Skrypt: 4.php

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <title>Podstawowa Strona</title>
  <link rel="stylesheet" type="text/css" href="styl.css">
</head>
<body>

<header>
  <h1>Ćwiczenia z łączenia z baz danych</h1>
</header>

<main>
  <aside>
    <ol>
```

```

        <li><a href="1.php">Ćwiczenie 1</a></li>
        <li><a href="2.php">Ćwiczenie 2</a></li>
        <li><a href="3.php">Ćwiczenie 3</a></li>
        <li>Ćwiczenie 4</li>
        <li>Ćwiczenie 5</li>
    </ol>
</aside>

<section>
    <h2>Ćwiczenie 4</h2>
    <p>Usuwanie rekordów</p>

<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
    <label for="id_klienta">ID Klienta:</label>
    <input type="text" name="id_klienta" required>
    <button type="submit">Usuń rekord</button>
</form>

</section>
</main>

<footer>
    <h3>Opracował: Imię Nazwisko</h3>
</footer>

</body>
</html>

```

Następnie dodanie php- skryptu

```

<section>
    <h2>Ćwiczenie 4</h2>
    <p>Usuwanie rekordów</p>

<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
    <label for="id_klienta">ID Klienta:</label>
    <input type="text" name="id_klienta" required>

```

```

        <button type="submit">Usuń rekord</button>
    </form>
<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "klient_2024";

$conn = mysqli_connect($servername, $username, $password, $dbname);

// Sprawdzenie połączenia
if (!$conn) {
    die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
}

// Sprawdzenie czy ID klienta zostało przesłane
if (isset($_POST['id_klienta'])) {
    $id_klienta = mysqli_real_escape_string($conn, $_POST['id_klienta']);

    // Zapytanie do bazy danych
    $sql = "DELETE FROM klient WHERE id_klienta = $id_klienta";
    $result = mysqli_query($conn, $sql);

    // Sprawdzenie czy zapytanie zostało wykonane
    if ($result) {
        echo "Rekord o ID Klienta $id_klienta został usunięty.";
    } else {
        echo "Błąd zapytania: " . mysqli_error($conn);
    }
}

// Zamknięcie połączenia
mysqli_close($conn);
?>

```

Kolejny skrypt poprawiony, ma sprawdzić czy w bazie danych znajduje się klient o takim id

Skrypt sprawdza, czy formularz został wysłany, następnie sprawdza, czy klient o podanym ID istnieje w bazie danych. Jeśli tak, usuwa go i wyświetla odpowiednie komunikaty, a jeśli nie, informuje o braku klienta o danym ID.


```

<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
    <label for="id_klienta">ID Klienta:</label>
    <input type="text" name="id_klienta" required>
    <button type="submit">Usuń rekord</button>
</form>
<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "klient_2024";

$conn = mysqli_connect($servername, $username, $password, $dbname);

// Sprawdzenie połączenia
if (!$conn) {
    die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
}

// Sprawdzenie czy ID klienta zostało przesłane
if (isset($_POST['id_klienta'])) {
    $id_klienta = mysqli_real_escape_string($conn, $_POST['id_klienta']);

    // Zapytanie do bazy danych
    $sql = "SELECT * FROM klient WHERE id_klienta = $id_klienta";
    $result = mysqli_query($conn, $sql);

    // Sprawdzenie czy rekord istnieje
    if (mysqli_num_rows($result) > 0) {
        // Usunięcie rekordu
        $deleteSql = "DELETE FROM klient WHERE id_klienta = $id_klienta";
        $deleteResult = mysqli_query($conn, $deleteSql);

        // Sprawdzenie czy rekord został usunięty
        if ($deleteResult) {
            echo "Rekord o ID Klienta $id_klienta został usunięty.";
        } else {
            echo "Błąd podczas usuwania rekordu: " . mysqli_error($conn);
        }
    } else {
        echo "Brak klienta o podanym ID.";
    }
}

```

```

    }
}

// Zamknięcie połączenia
mysqli_close($conn);
?>

```

Wyjaśnienie:

1. `if (isset($_POST['id_klienta']))` {: Sprawdza, czy formularz został przesłany i czy zmienna `$_POST['id_klienta']` istnieje. Jeśli tak, przechodzi do dalszej części kodu.
2. `$id_klienta = mysqli_real_escape_string($conn, $_POST['id_klienta']);` Przypisuje wartość ID klienta z formularza do zmiennej `$id_klienta`. Funkcja `mysqli_real_escape_string` służy do zabezpieczenia przed atakami SQL injection, a także konwertuje wartość na ciąg znaków.
3. `$sql = "SELECT * FROM tabela_klientow WHERE id_klienta = $id_klienta";` Tworzy zapytanie SQL, które sprawdza, czy istnieje rekord w tabeli `tabela_klientow` o podanym ID.
4. `$result = mysqli_query($conn, $sql);` Wykonuje zapytanie do bazy danych i przypisuje wynik do zmiennej `$result`.
5. `if (mysqli_num_rows($result) > 0)` {: Sprawdza, czy istnieje co najmniej jeden rekord pasujący do podanego ID klienta.
6. `$deleteSql = "DELETE FROM tabela_klientow WHERE id_klienta = $id_klienta";` Tworzy zapytanie SQL do usunięcia rekordu o podanym ID z tabeli.
7. `$deleteResult = mysqli_query($conn, $deleteSql);` Wykonuje zapytanie do usunięcia rekordu i przypisuje wynik do zmiennej `$deleteResult`.
8. `if ($deleteResult)` {: Sprawdza, czy usunięcie rekordu zostało pomyślnie zrealizowane.
9. `echo "Rekord o ID Klienta $id_klienta został usunięty."` {: Wyświetla komunikat potwierdzający usunięcie rekordu.
10. `else { echo "Błąd podczas usuwania rekordu: " . mysqli_error($conn); }` Wyświetla komunikat błędu, jeśli usunięcie rekordu nie powiedzie się.
11. `else { echo "Brak klienta o podanym ID.";` {: Jeśli w kroku 5 stwierdzono, że nie istnieje rekord o podanym ID, wyświetla komunikat informujący o braku klienta o danym ID.

Ćwiczenie 5

Dokładnie chodzi o to samo co w 4, tylko zamiast wpisywać `id_klienta` niech się pojawi lista z której mogą wybrać `id_klienta` do usunięcia.

1. [Ćwiczenie 1](#)
2. [Ćwiczenie 2](#)
3. [Ćwiczenie 3](#)
4. [Ćwiczenie 4](#)
5. [Ćwiczenie 5](#)

Ćwiczenie 5

Usuwanie rekordów

Wybierz ID Klienta:

- 1
- 2
- 3
- 4
- 5

Opracował: Imię Nazwisko

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <title>Podstawowa Strona</title>
  <link rel="stylesheet" type="text/css" href="styl.css">
</head>
<body>

<header>
  <h1>Ćwiczenia z łączenia z baz danych</h1>
</header>

<main>
  <aside>
    <ol>
      <li><a href="1.php">Ćwiczenie 1</a></li>
      <li><a href="2.php">Ćwiczenie 2</a></li>
      <li><a href="3.php">Ćwiczenie 3</a></li>
      <li><a href="4.php">Ćwiczenie 4</a></li>
      <li><a href="5.php">Ćwiczenie 5</a></li>
    </ol>
  </aside>

  <section>
    <h2>Ćwiczenie 5</h2>
    <p>Usuwanie rekordów</p>
    <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
```

```

        <label for="id_klienta">Wybierz ID Klienta:</label>
        <select name="id_klienta" required>
            <?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "klient_2024";

$conn = mysqli_connect($servername, $username, $password, $dbname);

        // Sprawdzenie połączenia
        if (!$conn) {
            die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
        }

        // Zapytanie do bazy danych
        $sql = "SELECT id_klienta FROM klient";
        $result = mysqli_query($conn, $sql);

        // Wyświetlanie opcji w polu wyboru
        while ($row = mysqli_fetch_assoc($result)) {
            echo "<option value='" . $row['id_klienta'] . "'>" .
$row['id_klienta'] . "</option>";
        }

        // Zamknięcie połączenia
        mysqli_close($conn);
        ?>
    </select>
    <button type="submit">Usun rekord</button>
</form>

<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "klient_2024";

$conn = mysqli_connect($servername, $username, $password, $dbname);

```

```
// Sprawdzenie połączenia
if (!$conn) {
    die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
}

// Sprawdzenie czy ID klienta zostało przesłane
if (isset($_POST['id_klienta'])) {
    $id_klienta = mysqli_real_escape_string($conn, $_POST['id_klienta']);

    // Zapytanie do bazy danych
    $checkSql = "SELECT * FROM klient WHERE id_klienta = $id_klienta";
    $checkResult = mysqli_query($conn, $checkSql);

    // Sprawdzenie czy rekord istnieje
    if (mysqli_num_rows($checkResult) > 0) {
        // Usunięcie rekordu
        $deleteSql = "DELETE FROM klient WHERE id_klienta = $id_klienta";
        $deleteResult = mysqli_query($conn, $deleteSql);

        // Sprawdzenie czy rekord został usunięty
        if ($deleteResult) {
            echo "Rekord o ID Klienta $id_klienta został usunięty.";
        } else {
            echo "Błąd podczas usuwania rekordu: " . mysqli_error($conn);
        }
    } else {
        echo "Brak klienta o podanym ID.";
    }
}

// Zamknięcie połączenia
mysqli_close($conn);
?>
```

```
</section>
</main>

<footer>
    <h3>Opracował: Imię Nazwisko</h3>
```

```
</footer>
```

```
</body>
```

```
</html>
```

Wyjaśnienie 1 kodu php

```
„// Zapytanie do bazy danych
    $sql = "SELECT id_klienta FROM tabela_klientow";
    $result = mysqli_query($conn, $sql);

    // Wyświetlanie opcji w polu wyboru
    while ($row = mysqli_fetch_assoc($result)) {
        echo "<option value='". $row['id_klienta'] . "'>" . $row['id_klienta'] .
"</option>";
    }

”
```

- **\$sql = "SELECT id_klienta FROM tabela_klientow";** Tworzy zapytanie SQL, które wybiera wszystkie wartości z kolumny id_klienta z tabeli klientow.
- **\$result = mysqli_query(\$conn, \$sql);** Wykonuje zapytanie do bazy danych za pomocą funkcji mysqli_query, przy użyciu połączenia \$conn i zapytania SQL \$sql. Wynik zapytania jest przechowywany w zmiennej \$result.
- **while (\$row = mysqli_fetch_assoc(\$result))** {: Pętla while przechodzi przez każdy wiersz wyników zapytania (rekordy w tabeli). Funkcja mysqli_fetch_assoc pobiera kolejny wiersz wyników jako asocjacyjny tablicowy.
- **echo "<option value='". \$row['id_klienta'] . "'>" . \$row['id_klienta'] . "</option>";** W każdym przebiegu pętli, tworzy się element <option> w HTML, który reprezentuje jedną opcję w rozwijanej liście. Wartość opcji (atr. value) jest ustawiana na id_klienta danego wiersza, a tekst opcji jest również id_klienta. W ten sposób tworzymy listę rozwijaną z dostępnymi ID klientów z bazy danych.

Podsumowując, ten fragment kodu wykonuje zapytanie do bazy danych, pobiera dostępne ID klientów, a następnie tworzy rozwijaną listę w formularzu HTML, wyświetlając te ID jako opcje, które użytkownik może wybierać.

Wyjaśnienie 2 kodu php

- **if (isset(\$_POST['id_klienta']))** {: Sprawdza, czy zmienna \$_POST['id_klienta'] została przesłana z formularza. Warunek ten sprawdza, czy użytkownik wypełnił pole z ID klienta przed wysłaniem formularza.
- **\$id_klienta = mysqli_real_escape_string(\$conn, \$_POST['id_klienta']);** Jeśli ID klienta zostało przesłane, używana jest funkcja mysqli_real_escape_string do zabezpieczenia wartości przed atakami SQL injection.
- **\$checkSql = "SELECT * FROM klient WHERE id_klienta = \$id_klienta";** Tworzy zapytanie SQL, które sprawdza, czy istnieje rekord w tabeli tabela_klientow o podanym ID.

- `$checkResult = mysqli_query($conn, $checkSql);` Wykonuje zapytanie do bazy danych i przechowuje wynik w zmiennej `$checkResult`.
- `if (mysqli_num_rows($checkResult) > 0) {` Sprawdza, czy istnieje co najmniej jeden rekord pasujący do podanego ID klienta.
- `$deleteSql = "DELETE FROM klient WHERE id_klienta = $id_klienta";` Tworzy zapytanie SQL do usunięcia rekordu o podanym ID z tabeli `klient`.
- `$deleteResult = mysqli_query($conn, $deleteSql);` Wykonuje zapytanie do bazy danych w celu usunięcia rekordu i przechowuje wynik w zmiennej `$deleteResult`.
- `if ($deleteResult) {` Sprawdza, czy usunięcie rekordu zostało pomyślnie zrealizowane.
- `echo "Rekord o ID Klienta $id_klienta został usunięty.";` Wyświetla komunikat potwierdzający usunięcie rekordu.
- `else { echo "Błąd podczas usuwania rekordu: " . mysqli_error($conn); }` Wyświetla komunikat błędu, jeśli usunięcie rekordu nie powiedzie się.
- `else { echo "Brak klienta o podanym ID.";` Jeśli w kroku 5 stwierdzono, że nie istnieje rekord o podanym ID, wyświetla komunikat informujący o braku klienta o danym ID.

Podsumowując, ten fragment kodu sprawdza, czy ID klienta zostało przesłane, a następnie sprawdza, czy istnieje rekord o podanym ID. Jeśli rekord istnieje, usuwa go i wyświetla odpowiednie komunikaty. Jeśli rekord nie istnieje, informuje o braku klienta o danym ID.

Przykład:

przykład kodu w PHP, który łączy się z bazą danych MySQL za pomocą MySQLi i liczy liczbę rekordów w tabeli o nazwie "klient":

Ćwiczenie 5

Usuwanie rekordów

Wybierz ID Klienta:

Liczba rekordów w tabeli 'klient': 14

```
<hr>
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "klient_2024";

$conn = mysqli_connect($servername, $username, $password, $dbname);

// Sprawdzenie połączenia
if (!$conn) {
```

```

        die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
    }

    // Zapytanie do bazy danych
    $sql = "SELECT COUNT(*) as liczba_rekordow FROM klient";
    $result = mysqli_query($conn, $sql);

    // Sprawdzenie wyników
    if ($result) {
        $row = mysqli_fetch_assoc($result);
        $liczba_rekordow = $row['liczba_rekordow'];

        echo "Liczba rekordów w tabeli 'klient': $liczba_rekordow";
    } else {
        echo "Błąd podczas zapytania do bazy danych: " . mysqli_error($conn);
    }

    // Zamknięcie połączenia
    mysqli_close($conn);
?>

```

Wyświetla ile jest klientów z danego miasta i sortuje od największej wartości do najmniejszej i wyświetla na stronie tylko 5 pierwszych:

Liczba klientów z danego miasta

Miasto	Liczba klientów
Kraków	4
Rzeszów	3
Olsztyn	1
Wrocław	1
Lublin	1

```

<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "klient_2024";

$conn = mysqli_connect($servername, $username, $password, $dbname);

```



```
// Sprawdzenie połączenia
if (!$conn) {
    die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
}

// Zapytanie do bazy danych
$sql = "SELECT miasto, COUNT(*) as liczba_klientow FROM klient GROUP BY miasto
ORDER BY liczba_klientow DESC LIMIT 5";
$result = mysqli_query($conn, $sql);

// Sprawdzenie wyników
if ($result) {
    echo "<table border='1'>";
    echo "<tr><th>Miasto</th><th>Liczba klientów</th></tr>";
    while ($row = mysqli_fetch_assoc($result)) {
        echo
"<tr><td>{$row['miasto']}</td><td>{$row['liczba_klientow']}</td></tr>";
    }
    echo "</table>";
} else {
    echo "Błąd podczas zapytania do bazy danych: " . mysqli_error($conn);
}

// Zamknięcie połączenia
mysqli_close($conn);
?>
```

Ćwiczenie 6

Utwórz bazę danych klasa4, a w niej tabelę osoby a następnie wstaw 15 rekordów:

Kod:

```
CREATE TABLE osoby (
    id_osoba int PRIMARY KEY AUTO_INCREMENT,
    nazwisko varchar(50) NOT NULL,
    imie varchar(50) NOT NULL,
    miasto varchar(50) NOT NULL,
    data_urodzenia DATE,
    plec enum('M','K') NOT NULL
);
```

```

INSERT INTO osoby (nazwisko, imie, miasto, data_urodzenia, plec) VALUES
('Kowalski', 'Jan', 'Warszawa', '1990-05-15', 'M'),
('Nowak', 'Anna', 'Kraków', '1985-08-22', 'K'),
('Wiśniewski', 'Piotr', 'Gdańsk', '1978-12-10', 'M'),
('Lewandowska', 'Katarzyna', 'Poznań', '1995-04-03', 'K'),
('Zieliński', 'Marek', 'Łódź', '1980-06-28', 'M'),
('Szymańska', 'Agnieszka', 'Wrocław', '1992-09-17', 'K'),
('Dąbrowski', 'Michał', 'Szczecin', '1987-02-08', 'M'),
('Kaczmarek', 'Ewa', 'Katowice', '1998-11-25', 'K'),
('Wojciechowski', 'Alicja', 'Białystok', '1983-07-14', 'K'),
('Jankowski', 'Adam', 'Bydgoszcz', '1975-01-20', 'M'),
('Grabowska', 'Patrycja', 'Gdynia', '1993-03-12', 'K'),
('Nowicki', 'Rafał', 'Częstochowa', '1982-10-05', 'M'),
('Mazur', 'Karolina', 'Lublin', '1997-06-18', 'K'),
('Witkowski', 'Grzegorz', 'Rzeszów', '1979-09-30', 'M'),
('Jaworski', 'Magdalena', 'Olsztyn', '1988-04-22', 'K');

```

```

CREATE TABLE osoby (
    id_osoba int PRIMARY KEY AUTO_INCREMENT,
    nazwisko varchar(50) NOT NULL,
    imie varchar(50) NOT NULL,
    miasto varchar(50) NOT NULL,
    data_urodzenia DATE,
    plec enum('M','K') NOT NULL
);

```

```

INSERT INTO osoby (nazwisko, imie, miasto, data_urodzenia, plec) VALUES
('Kowalski', 'Jan', 'Warszawa', '1990-05-15', 'M'),
('Nowak', 'Anna', 'Kraków', '1985-08-22', 'K'),
('Wiśniewski', 'Piotr', 'Gdańsk', '1978-12-10', 'M'),
('Lewandowska', 'Katarzyna', 'Poznań', '1995-04-03', 'K'),
('Zieliński', 'Marek', 'Łódź', '1980-06-28', 'M'),
('Szymańska', 'Agnieszka', 'Wrocław', '1992-09-17', 'K'),
('Dąbrowski', 'Michał', 'Szczecin', '1987-02-08', 'M'),

```

('Kaczmarek', 'Ewa', 'Katowice', '1998-11-25', 'K'),
('Wojciechowski', 'Alicja', 'Białystok', '1983-07-14', 'K'),
('Jankowski', 'Adam', 'Bydgoszcz', '1975-01-20', 'M'),
('Grabowska', 'Patrycja', 'Gdynia', '1993-03-12', 'K'),
('Nowicki', 'Rafał', 'Częstochowa', '1982-10-05', 'M'),
('Mazur', 'Karolina', 'Lublin', '1997-06-18', 'K'),
('Witkowski', 'Grzegorz', 'Rzeszów', '1979-09-30', 'M'),
('Jaworski', 'Magdalena', 'Olsztyn', '1988-04-22', 'K');

Ćwiczenie 6

Wyswietlanie danych w postaci tabeli

Wybierz płeć i miasto lub miasto:

Kobieta

Męczyzna

Miasto:

Wyświetl dane

Użytkownik wybrał następujące kryteria pobierania danych:

Płeć: (K).

Miasto: nie określono (%).

Nazwisko	Imię	Miasto	Data urodzenia	Płeć
Nowak	Anna	Kraków	1985-08-22	K
Lewandowska	Katarzyna	Poznań	1995-04-03	K
Szymańska	Agnieszka	Wrocław	1992-09-17	K
Kaczmarek	Ewa	Katowice	1998-11-25	K
Wojciechowski	Alicja	Białystok	1983-07-14	K
Grabowska	Patrycja	Gdynia	1993-03-12	K
Mazur	Karolina	Lublin	1997-06-18	K
Jaworski	Magdalena	Olsztyn	1988-04-22	K

Kod strony:

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <title>Podstawowa Strona</title>
  <link rel="stylesheet" type="text/css" href="styl.css">
</head>
<body>

<header>
  <h1>Ćwiczenia z łączenia z baz danych</h1>
</header>

<main>
  <aside>
```

```

<ol>
  <li><a href="6.php">Ćwiczenie 6</a></li>
  <li><a href="7.php">Ćwiczenie 7</a></li>
  <li><a href="8.php">Ćwiczenie 8</a></li>
  <li><a href="9.php">Ćwiczenie 9</a></li>
  <li><a href="10.php">Ćwiczenie 10</a></li>
</ol>
</aside>

<section>
  <h2>Ćwiczenie 6</h2>
  <p>Wyswietlanie danych w postaci tabeli</p>

  <p>Wybierz płeć i miasto lub miasto:</p>

<form action="6.php" method="post">
  <input type="radio" id="kobieta" name="plec" value="K" />
  <label for="kobieta">Kobieta</label><br>
  <input type="radio" id="mezczyzna" name="plec" value="M" />
  <label for="mezczyzna">Mężczyzna</label><br><br><br>
  <label for="miasto">Miasto: </label><br><br>
  <input type="text" id="miasto" name="miasto" /><br>
  <input type="submit" value="Wyświetl dane" />
</form>

<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "klasa4";

$conn = mysqli_connect($servername, $username, $password, $dbname);

// Sprawdzanie połączenia
if (!$conn) {
    die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
}

if (isset($_POST['plec'])) {
    $plec = $_POST['plec'];

```

```

}
else {
// Znak % jest to dowolny znak:
$plec = "%";
}
if ( isset($_POST['miasto']) and !empty($_POST['miasto']) ) {
$miasto = $_POST['miasto'];
}

else {
$miasto = "%";
}

echo "<p>Użytkownik wybrał następujące kryteria pobierania danych:</p>";
echo "<ul>";
echo "<li><span>Płeć</span>: ";
if ( $plec == "%" ) {
echo "nie określono";
}
echo " (<b>$plec</b>).</li>";
echo "<li><span>Miasto</span>: ";
if ( $miasto == "%" ) {
echo "nie określono";
}
echo " (<b>$miasto</b>).</li>";
echo "</ul>";
// Definiujemy, co chcemy wyszukać w tabeli osoby:
$zapytanie1 = "SELECT * FROM osoby WHERE Plec LIKE
'$plec' AND Miasto LIKE '$miasto'";

if ($rezultat = mysqli_query($conn,
$zapytanie1)) {
// Jeśli został znaleziony chociaż jeden rekord spełniający kryteria, to:
if (mysqli_num_rows($rezultat) > 0) {
echo "<table>";
echo "<tr>";
echo "<th>Nazwisko</th>";
echo "<th>Imię</th>";
echo "<th>Miasto</th>";
echo "<th>Data urodzenia</th>";
echo "<th>Płeć</th>";

```

```

echo "</tr>";
// Tak długo, jak udaje się wydobyć dane z kolejnych rekordów,
// przypisz je do zmiennej $wiersz, a następnie wypisz każdą komórkę w wierszu:
while($wiersz = mysqli_fetch_array($rezultat)) {
echo "<tr>";

echo "<td>" . $wiersz['nazwisko'] . "</td>";
echo "<td>" . $wiersz['imie'] . "</td>";
echo "<td>" . $wiersz['miasto'] . "</td>";
echo "<td>" . $wiersz['data_urodzenia'] . "</td>";
echo "<td>" . $wiersz['plec'] . "</td>";
echo "</tr>";
}
echo "</table>";
// Czyści zmienną $rezultat:
mysqli_free_result($rezultat);
}
}
else {
echo "<p style='color: red;'>Wystąpiły błędy:</p> <ul
style='color: green;'><li>tabela lub kolumny o tych nazwach nie
istnieją;</li><li>lub problem z nawiasami</li></ul>";
}

mysqli_close($conn);
?>

</section>
</main>

<footer>
<h3>Opracował: Imię Nazwisko</h3>
</footer>

</body>
</html>

```

Wyjaśnienie:

Ten fragment kodu PHP odpowiada za pobranie i przetworzenie danych przesłanych przez formularz HTML, a następnie wykonanie zapytania do bazy danych MySQL i wyświetlenie wyników w postaci tabeli.

1. Pobranie wartości z formularza dotyczącego płci:

```
if (isset($_POST['plec'])) {  
    $plec = $_POST['plec'];  
} else {  
    // Jeśli wartość nie została przesłana, ustaw wartość domyślną na '%'  
    $plec = "%";  
}
```

- Sprawdza, czy w formularzu została przesłana wartość dla pola 'plec'. Jeśli tak, przypisuje ją do zmiennej \$plec, w przeciwnym razie ustawia wartość domyślną na '%'.

2. Pobranie wartości z formularza dotyczącego miasta:

```
if (isset($_POST['miasto']) and !empty($_POST['miasto'])) {  
    $miasto = $_POST['miasto'];  
} else {  
    // Jeśli wartość nie została przesłana lub jest pusta, ustaw wartość domyślną na  
    $miasto = "%";  
}
```

- Sprawdza, czy w formularzu została przesłana wartość dla pola 'miasto' i czy nie jest pusta. Jeśli warunki są spełnione, przypisuje wartość do zmiennej \$miasto, w przeciwnym razie ustawia wartość domyślną na '%'.

3. Wyświetlenie wybranych przez użytkownika kryteriów:

```
echo "<p>Użytkownik wybrał następujące kryteria pobierania danych:</p>";  
echo "<ul>";  
echo "<li><span>Płeć</span>: ";  
if ($plec == "%") {  
    echo "nie określono";  
}  
echo " (<b>$plec</b>).</li>";  
echo "<li><span>Miasto</span>: ";  
if ($miasto == "%") {  
    echo "nie określono";  
}  
echo " (<b>$miasto</b>).</li>";  
echo "</ul>";
```

- Wyświetla informacje o wybranych kryteriach pobierania danych na podstawie płci i miasta.

4. Definiowanie zapytania SQL na podstawie wybranych kryteriów:

```
$zapytanie1 = "SELECT * FROM osoby WHERE Plec LIKE '$plec' AND Miasto LIKE '$miasto'";
```

- Tworzy zapytanie SQL, które będzie pobierać dane z tabeli `osoby` na podstawie wybranych przez użytkownika kryteriów płci i miasta.

5. Wykonanie zapytania do bazy danych i wyświetlenie wyników:

```
if ($rezultat = mysqli_query($conn, $zapytanie1)) {  
    if (mysqli_num_rows($rezultat) > 0) {  
        // ... (kod do wyświetlania tabeli)  
    }  
} else {  
    // Wyświetlenie komunikatu o błędzie, jeśli zapytanie nie powiodło się  
    echo "<p style='color: red;'>Wystąpiły błędy:</p> <ul style='color: green;'><li>
```

Wykonuje zapytanie do bazy danych, a następnie sprawdza, czy znaleziono jakieś rekordy spełniające kryteria. Jeśli tak, przechodzi do wyświetlania wyników w postaci tabeli. W przypadku błędu, wyświetla komunikat o błędzie.

Zadanie.

Na podstawie powyższych przykładów przygotuj dowolną stronę w php5, w stopce umieść swoje dane, wymyśl dowolne 3 tabele w mysql a następnie wykorzystaj powyższe przykłady do realizacji własnego projektu.