Login do admina: admin

Hasło do admina: admin

Db\_connect.php

**Opis funkcji:**

Funkcja **connectToDatabase()** służy do nawiązywania połączenia z bazą danych MySQL. Przyjmuje dane dostępowe do serwera bazy danych (serwer, nazwa użytkownika, hasło) oraz nazwę docelowej bazy danych. Funkcja zwraca obiekt reprezentujący połączenie z bazą danych.

**Parametry wejściowe:**

1. **$servername** - Adres serwera bazy danych.
2. **$username** - Nazwa użytkownika do logowania do bazy danych.
3. **$password** - Hasło użytkownika do logowania do bazy danych.
4. **$dbname** - Nazwa docelowej bazy danych. Domyślnie ustawiony na "twitter". Zakomentowana linia z nazwą bazy danych "db\_92584" może być odkomentowana, a domyślne wartości zostaną zastąpione.

**Zwracana wartość:**

Funkcja zwraca obiekt **mysqli** reprezentujący nawiązane połączenie z bazą danych.

**Obsługa błędów:**

Funkcja zawiera mechanizm obsługi błędów połączenia. W przypadku nieudanego połączenia, wywołuje funkcję **die()**, wyświetlając komunikat o błędzie połączenia i kończąc działanie skryptu.

**Przykład użycia:**

<?php

$conn = connectToDatabase();

$conn->close();

?>

QueryBuilder.php

**Opis klasy:**

Klasa **SQLQueryBuilder** służy do konstruowania i wykonywania zapytań SQL na podstawie różnych kryteriów, takich jak warunki, sortowanie, limity i offsety. Klasa obsługuje operacje SELECT, UPDATE oraz COUNT.

**Atrybuty klasy:**

1. **$table** - Nazwa tabeli, na której będą wykonywane zapytania.
2. **$conditions** - Tablica warunków używanych w klauzuli WHERE zapytania SQL.
3. **$orderBy** - Tablica określająca kolumny i kierunki sortowania w klauzuli ORDER BY zapytania SQL.
4. **$limit** - Limit wyników w zapytaniu SELECT.
5. **$offset** - Offset wyników w zapytaniu SELECT.
6. **$conn** - Obiekt reprezentujący połączenie z bazą danych.
7. **$type** - Typ operatora logicznego używanego między warunkami (AND lub OR).

**Metody klasy:**

1. **\_\_construct($table, $conn)**

Konstruktor klasy. Inicjalizuje obiekt klasy i ustawia nazwę tabeli oraz obiekt połączenia z bazą danych.

1. **addCondition($column, $value, $operator = '=', $type = "AND")**

Dodaje warunek do tablicy warunków. Domyślnie używany jest operator równości, a typ logiczny to AND.

1. **orderBy($column, $direction = 'ASC')**

Dodaje kolumnę i kierunek sortowania do tablicy sortowania.

1. **setLimit($limit)**

Ustawia limit wyników w zapytaniu SELECT.

1. **setOffset($offset)**

Ustawia offset wyników w zapytaniu SELECT.

1. **executeQuery()**

Wykonuje zbudowane zapytanie SELECT i zwraca wyniki w formie tablicy asocjacyjnej.

1. **limit($limit, $offset = null)**

Ustawia limit i offset wyników w zapytaniu SELECT (metoda ta jest zduplikowana).

1. **count($customCondition = "")**

Wykonuje zapytanie COUNT z opcjonalnym dodatkowym warunkiem i zwraca liczbę wyników.

1. **update($data, $condition)**

Wykonuje zapytanie UPDATE, aktualizując kolumny z przekazanymi danymi i dodatkowym warunkiem.

1. **buildQuery() (metoda prywatna)**

Buduje pełne zapytanie SQL na podstawie ustawionych kryteriów.

**Przykłady użycia:**

<?php

$conn = connectToDatabase();

$queryBuilder = new SQLQueryBuilder('example\_table', $conn);

$queryBuilder->addCondition('column\_name', 'value', '>');

$queryBuilder->orderBy('column\_name', 'DESC');

$queryBuilder->setLimit(10);

$queryBuilder->setOffset(5);

$results = $queryBuilder->executeQuery();

print\_r($results);

$totalRows = $queryBuilder->count();

echo "Total Rows: " . $totalRows;

?>

DeleteRows.php

**Opis klasy:**

Klasa **DeleteRows** służy do usuwania wierszy z tabeli w bazie danych. Udostępnia metodę **deleteRow()**, która umożliwia wykonanie zapytania DELETE, usuwając wiersze z określonej tabeli spełniające podany warunek.

**Atrybuty klasy:**

1. **$db** - Obiekt reprezentujący połączenie z bazą danych.

**Metody klasy:**

1. **\_\_construct($conn)**

Konstruktor klasy. Inicjalizuje obiekt klasy i ustawia obiekt połączenia z bazą danych.

1. **deleteRow($tableName, $condition)**

Metoda do usuwania wierszy z określonej tabeli na podstawie podanego warunku. Wykorzystuje zapytanie DELETE.

**Przykład użycia:**

<?php

$conn = connectToDatabase();

$deleteRows = new DeleteRows($conn);

$deleteRows->deleteRow('example\_table', 'column\_name = "value"');

$conn->close();

?>

DataInsert.php

**Opis klasy:**

Klasa **DataInserter** służy do wstawiania danych do tabeli w bazie danych. Udostępnia metodę **insertData()**, która umożliwia wstawienie nowego wiersza do określonej tabeli na podstawie przekazanych danych.

**Atrybuty klasy:**

1. **$conn** - Obiekt reprezentujący połączenie z bazą danych.

**Metody klasy:**

1. **\_\_construct($conn)**

Konstruktor klasy. Inicjalizuje obiekt klasy i ustawia obiekt połączenia z bazą danych.

1. **insertData($tableName, $data)**

Metoda do wstawiania danych do określonej tabeli. Akceptuje nazwę tabeli i asocjacyjną tablicę danych. Wstawia nowy wiersz na podstawie przekazanych danych.

**Przykład użycia:**

<?php

$conn = connectToDatabase();

$dataInserter = new DataInserter($conn);

$dataToInsert = [ 'column1' => 'value1', 'column2' => 'value2', 'column3' => 123 ];

$dataInserter->insertData('example\_table', $dataToInsert);

$conn->close();

?>

CountTop.php

**Opis funkcji:**

Funkcja **getTopElementsWithHighestCount** służy do pobierania najczęściej występujących elementów z określonej kolumny w tabeli bazy danych. Zwraca tablicę asocjacyjną z danymi, gdzie każdy element to wartość kolumny oraz ilość jej wystąpień, posortowane malejąco, z ograniczeniem do 5 najczęściej występujących elementów.

**Parametry wejściowe:**

1. **$conn** - Obiekt reprezentujący połączenie z bazą danych.
2. **$tableName** - Nazwa tabeli, z której pobierane są dane.
3. **$columnName** - Nazwa kolumny, dla której mają być pobrane najczęściej występujące elementy.

**Zwracana wartość:**

Tablica asocjacyjna, gdzie klucz to wartość kolumny, a wartość to ilość wystąpień, posortowane malejąco, z ograniczeniem do 5 najczęściej występujących elementów.

**Obsługa błędów:**

W przypadku nieudanego wykonania zapytania, funkcja przechwytuje wyjątek typu **Exception** i kończy działanie skryptu, wyświetlając komunikat z treścią błędu.

**Przykład użycia:**

<?php

$conn = connectToDatabase();

$topElements = getTopElementsWithHighestCount($conn, 'example\_table', 'example\_column');

print\_r($topElements);

$conn->close();

?>