# Przetwarzanie danych z bazy mySQL w PHP

**Jak to działa w XAMPP - > mySQL -> ADMIN**

1. Klikamy **ADMIN** 🡪 Wchodzimy na stronę localhost/phpmyadmin
2. Klikamy na bazę: **dziennik**
3. Piszemy zapytanie **SQL** (lub tworzymy je za pomocą myszki i pól wyboru) 🡪 Klikamy wykonaj
4. Zamykamy stronę

**A jak to działa w PHP (IDziennik, Allegro, OLX, Facebook)**

baza.php

**PHP**

1. **Połączenie z bazą**

**.**

**.**

**.**

**<?php**

**…**

**…**

**…**

**?>**

$wiersz[‘Imie‘]

$polaczenie

mySQL

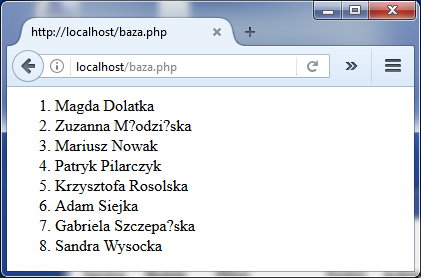
**HOST:** localhost

**USER:** root

**PASS:**

**DB:** dziennik

1. **Wybranie bazy:** dziennik
2. **Zapytanie:** SELECT \* FROM uczen
3. **Zamknięcie połączenia z bazą**



**Ćwiczenie 1. Nawiązywanie połączenia z bazą w PHP**

1. Przepisz poniższy skrypt do pliku baza.php.

<body>

**<?php**

$polaczenie = **mysqli\_connect**('localhost', 'root');

**mysqli\_select\_db**($polaczenie, 'dziennik');

$wynik = **mysqli\_query**($polaczenie, "SELECT \* from uczen");

**echo** "<ul>";

**while**( $wiersz = **mysqli\_fetch\_array**($wynik) )

{

**echo** "<li>" . $wiersz['Imie'] . " " . $wiersz['Nazwisko'];

}

**echo** "</ul>";

**mysqli\_close**($polaczenie);

**?>**

</body>

1. Umieść skrypt w folderze **c:/xampp/htdocs**
2. Uruchom serwer XAMPP: **Apache, mySQL**
3. Wywołaj skrypt w przeglądarce: **localhost/baza.php**
4. Popraw błędy składniowe i testuj do momentu zadziałania programu

**Ćwiczenia dodatkowe:**

1. Dodaj nagłówek do strony:

**<h1>**Dziennik lekcyjny**</h1>**

<**img** **src**=”dziennik.png” **/>**

1. Jak w skrypcie dokonać zmiany listy na numerowaną?
2. Zmień zapytanie SELECT, żeby uczniowie byli wyświetlani alfabetycznie według Imion:
3. Dopisz na początku pliku styl CSS i sprawdź jego wpływ na prezentację wyniku

**<style>**

li { margin: 5px; background: orange; font-size: 20px; }

**</style>**

1. Przetłumacz angielskie słowa

connect -

select -

database (db) -

query -

close -

**Funkcje interfejsu dostępu do bazy w PHP: mysqli\_**

Połączenie z bazą danych w PHP odbywa sięga pomocą funkcji mysqli\_connect(), która przyjmuje trzy parametry i jako wynik zwraca identyfikator połączenia:

**$polaczenie = mysqli\_connect(host, użytkownik, hasło)**

Funkcjamysqli\_connect() zwraca identyfikator połączenia, który jest drugim parametrem wywołania funkcji wybierającej odpowiednią bazę danych mysqli\_select\_db()

**mysqli\_select\_db($polaczenie , nazwa\_bazy)**

Wykonywanie zapytań do bazy realizuje się za pomocą funkcji **mysql\_query()**

**$wynik = mysqli\_query($polaczenie, zapytanie\_sql)**

**mysqli\_query**()  zwraca identyfikator wyniku (lub FALSE w przypadku niepowodzenia) jedynie dla zapytań typu SELECT, SHOW, EXPLAIN i DESCRIBE. Dla innych zapytań SQL  **mysqli\_query()** zwraca  TRUE lub FALSE informując czy zapytanie zakończyło się sukcesem czy też nie. Jeśli nie została zwrócona wartość FALSE to znaczy, że zapytanie było prawidłowe i może być wykonane przez serwer. Nie mówi natomiast nic o liczbie przetworzonych lub zwróconych wierszy. Jest również możliwe, że zapytanie zostanie wykonane poprawnie, nie przetwarzając lub zwracając żadnych wierszy.

Zwrócone wiersze można odczytać przy pomocy funkcji:

**$wiersz = mysqli\_fetch\_row($wynik)**

lub

**$wiersz = mysqli\_fetch\_assoc($wynik)**

lub

**$wiersz = mysqli\_fetch\_array($wynik)**

Funcja **mysqli\_fetch\_row()** zwraca wynik w postaci tablicy numerycznej, gdzie $wiersz[0] oznacza 1 zwrócone pole. Funkcja **mysqli\_fetch\_assoc()** zwraca wiersz w postaci tablicy asocjacyjnej, w której klucze są nazwami pól w bazie np.: wiersz[‘pesel’]. W przypadku funkcji **mysqli\_fetch\_array()**, która jest najbardziej uniwersalna zwrócona tabela może być typu numerycznego, asocjacyjnego bądź obiema jednocześnie (domyślnie). Można zastosować opcjonalny drugi argument, który jest stałą i może przyjmować następujące wartości: MYSQL\_ASSOC, MYSQL\_NUM i MYSQL\_BOTH. Wartością domyślną jest MYSQL\_BOTH.W przypadku gdy zapytanie zwraca kilka wierszy funkcje należy wywoływać w pętli dopóki zmienna wiersz przyjmuje wartość inną niż FALSE.

Do zamknięcia połączenia z bazą służy funkcja

**mysqli\_close($polaczenie)**

**Uwaga:** w celu zbadania poprawności wykonania funkcji bazodanowych często korzysta się z konstrukcji **or die()** występującej zaraz po niej, która w przypadku niepowodzenia (zwrócenia wyniku FASLE) wypisze stosowny komunikat i zatrzyma wykonywanie skryptu.