

PHP i MySQL

MySQL jest obecnie najpopularniejszym silnikiem relacyjnych baz danych i zarazem jednym z szybszych, udostępnianym na licencji **GPL**.

W połączeniu z **PHP** daje silne narzędzie do tworzenia zaawansowanych serwisów internetowych.

Połączenie z bazą danych

Aby nasz system miał możliwość łączenia się i współpracy z MySQL w PHP muszą być zainstalowane odpowiednie biblioteki (na szczęście w większości usług hostingowych oraz w środowisku XAMPP tak już jest).

W razie braku należy zadbać o to by owe biblioteki się znalazły i było poprawnie skonfigurowane (tu odsyłam do sieci po dalsze informacje w tej dziedzinie).

A co to są te biblioteki..? Odpowiedź jest prosta – to gotowy kod napisany po to by można było w sposób szybki i łatwy łączyć się i współpracować z MySQL.

Połączenie z bazą danych

dy nasza wersja PHP ma już odpowiednie biblioteki do współpracy z serwerem MySQL pierwszym krokiem, jaki będziemy chcieli wykonać jest połączenie z bazą danych. Aby połączyć się z bazą danych używamy funkcji **mysql_connect** z następującą składnią:

```
mysql_connect([adres_serwera [, nazwa_uzytkownika [, haslo]]]);
```

Pierwszym argumentem, jaki przyjmuje funkcja łączenia z bazą danych jest **adres serwera** - może on być za równo w postaci adresu IP jak i nazwy hosta. Za nazwą hosta po dwukropku możemy podać jeszcze numer portu adresu serwera - nie jest to wymagane, jeśli nie podamy numeru portu nastąpi próba połączenia na standardowym porcie 3306. Kolejnymi argumentami są **nazwa** i **hasło** użytkownika bazy danych MySQL. Funkcja **mysql_connect** przyjmuje również inne argumenty ale jako amatorzy nie będziemy o nich na razie mówili.

Połączenie z serwerem MySQL serwer.com na standardowym porcie:

```
$sql_conn = mysql_connect('serwer.com', 'admin', 'passwd')
```

Wykonanie powyższego kodu połączy nas z serwerem na porcie 3306.

Czasami zdarza się tak, że administrator serwera MySQL zmieni numer portu, połączenie na niestandardowym porcie wygląda w ten sposób:

```
$sql_conn = mysql_connect('serwer.com:27012', 'admin', 'passwd');
```

UWAGA:

Funkcja `mysql_connect` zwróci **identyfikator połączenia** o ile zakończy się sukcesem.
Jeśli połączenie nie powiedzie się funkcja zwróci wartość `false`.

Zamykanie połączenia

Po zakończeniu operacji w bazie należy się rozłączyć i zwolnić zasoby. Za rozłączanie się z bazą danych odpowiedzialna jest funkcja `mysql_close()` o następującej składni:

```
mysql_close([identyfikator_połączenia]);
```

identyfikator połączenia - nie jest wymagany, jeśli go nie podamy zostanie zamknięte ostatnio otwarte połączenie przez użytkownika na którego jesteśmy zalogowani do bazy danych.

Uwaga: jeśli jednak w tym samym czasie utrzymujemy więcej niż jedno połączenie z bazą danych należy kontrolować, to które połączenie zamykamy.

Uwaga 2: Wszystkie otwarte połączenia należy zamknąć (kiedy nie są już potrzebne). To bardzo pożądana sytuacja – nie powinno się zostawiać otwartych połączeń.

Przykład skryptu PHP operującego na połączeniu:

... //wcześniejsze operacje

`$sql_conn = mysql_connect('serwer.com', 'admin', 'passwd')` //nawiązanie poł.

... //operacje po nawiązaniu połączenia

... //np. dodanie danych do bazy, etc.

`mysql_close($sql_conn);` //zakończenie poł.

... //dalsze operacje – już bez operacji bazodanowych – połączenie zostało zamknięte

Zapytania

Aby wykonać jakiegokolwiek operacje na swojej bazie danych należy wysłać do niej informacje o tym co chcemy zrobić. Wszystkie informacje wysyłamy w postaci

zapytań, do czego wykorzystujemy PHPową funkcję

```
mysql_query([zapytanie]);
```

Zapytania do bazy danych wysyłamy w następujący sposób:

```
$zapytanie = "treść zapytania SQL";
```

```
$idzapytania = mysql_query($zapytanie);
```

Po takiej operacji zmienna **\$idzapytania** będzie zawierała identyfikator zapytania o ile zapytanie zostanie wykonane pomyślnie. W przypadku, gdy popełniliśmy błąd w zapytaniu zmienna **\$idzapytania** będzie zawierała wartość false.

Tworzenie / Wybieranie DB

Tworzenie DB

O ile nasz administrator nie utworzył nam wcześniej bazy danych lub sami jesteśmy administratorami należy najpierw utworzyć bazę danych. Do stworzenia nowej bazy używamy funkcji `mysql_create_db()` np.:

```
mysql_create_db('zawodnicy');
```

Funkcja zwróci wartość `true`, w przypadku gdy baza danych została utworzona pomyślnie lub wartość `false` w przypadku gdy baza o takiej nazwie istnieje lub nie mamy uprawnień aby ją utworzyć.

Wybieranie DB

Samo utworzenie bazy danych nie oznacza, że właśnie na niej będziemy wykonywać operacje, dlatego musimy ją wybrać, co robimy za pomocą funkcji

`mysql_select_db(nazwa)` np.:

```
mysql_select_db('zawodnicy');
```

Od tej pory wszelkie zapytania wysyłane do serwera będą wykonywane na bazie danych zawodnicy.

UWAGA: wybrać bazę danych należy również w sytuacji, gdy dana baza była już na serwerze utworzona (nie ważne czy przez nas czy przez administratora).

Tworzenie tabel

Tworzenie tabeli

Gdy nasza baza jest już utworzona i mamy z nią połączenie możemy zacząć wprowadzać dane, najpierw jednak trzeba **utworzyć tabelę**, robimy to według następującego schematu składni SQL:

```
CREATE TABLE nazwa_tabeli (nazwa_pola1 typ_pola1 [atrybuty], nazwa_pola2  
nazwa_pola2 [atrybuty], ... nazwa_polaN typ_polaN [atrybuty], PRIMARY  
KEY(nazwa_polaX))
```


Tworzenie tabeli – warunki...

Każda tabela musi posiadać co najmniej jedno pole, dodatkowo musi posiadać co najmniej jedno pole, które jednoznacznie identyfikuje wiersz w tabeli - tak zwany klucz główny. Jeśli w tabeli jest masa danych a Ty znasz klucz główny jednego z wierszy wtedy możesz bez problemu dostać się do tego wiersza.

Szczegóły dot. Tworzenia tabel na przedmiotach związanych z DB...

Dodawanie danych do DB

INSERT

Gdy mamy już utworzoną tabelę przydałoby się dodać do niej jakiś rekord, robimy to według następującej składni:

```
INSERT INTO nazwa_tabeli (nazwa_pola1, nazwa_pola2) VALUES (wartosc_pola1,  
wartosc_pola2)
```

Przykład:

Założmy, że chcemy dodać do naszej tabeli nowego zawodnika, operacja ta wygląda tak:

```
$zapytanie = "INSERT INTO nba (`id`, `nazwisko`, `lata`, `punkty`, `mistrzostwa`)  
VALUES ('', 'Jordan', '13', '32', '6')";
```

```
$idzapytania = mysql_query($zapytanie);
```

UWAGA: apostrofy przy nazwach tabeli oraz nazwach atrybutów (kolumn) to ODWRÓCONY APOSTROF – znajduje się pod znakiem ~ na klawiaturze!

UWAGA: Jako wartość pola id nie podaliśmy nic ponieważ jest to pole typu AUTO_INCREMENT a co za tym idzie serwer baz danych sam nada temu polu kolejną wartość. Jako, że jest to pierwszy rekord w tej tabeli wartość pola id w tym rekordzie będzie wynosiła 1.

Wybieranie danych

SELECT

Jeśli chcemy przeszukać naszą tabelę w poszukiwaniu jakiegoś konkretnego rekordu robimy to za pomocą komendy SELECT według składni:

```
SELECT nazwa_pola1,nazwa_pola2,...nazwa_polaN FROM nazwa_tabeli [WHERE  
warunek]
```

Przykład:

```
$zapytanie = "SELECT `lata`,`punkty` FROM `nba` WHERE `nazwisko`='Pipen';"  
$idzapytania = mysql_query($zapytanie);
```

Po wykonaniu tego zapytania na serwerze baz danych zostaną wybrane wszystkie rekordy, w których wartość pola nazwisko będzie równe Pipen.

SELECT – wyświetlanie wyników

```
$zapytanie = "SELECT `id`,`nazwisko` FROM `osoby`";
```

```
$idzapytania = mysql_query($zapytanie);
```

```
while ($wiersz = mysql_fetch_row($idzapytania))
```

```
{
```

```
    echo $wiersz[0] . ' : ' . $wiersz[1] . '<br />';
```

```
}
```

Istota tego kodu ukryta jest w funkcji `mysql_fetch_row`, która przy każdej iteracji pętli `while` wyciąga dany wiersz zapytania i zapisuje go w tablicy `$wiersz` indeksowanej od zera, wypełnionej elementami rekordu wybranymi za pomocą zapytania. Czyli element o indeksie zero będzie równy wartości danego rekordu pola `id`, element o indeksie jeden będzie równy wartości danego rekordu pola `nazwisko`...

SELECT – wyświetlanie wyników (po raz drugi)

mysql_fetch_assoc — Zapisuje wiersz wyniku w tablicy asocjacyjnej.

Różnica w stosunku do poprzedniej wersji będzie polegała na innym sposobie dostania się do zawartości pojedynczego wiersza:

```
$zapytanie = "SELECT `id`,`nazwisko` FROM `osoby`";
```

```
$idzapytania = mysql_query($zapytanie);
```

```
while ($wiersz = mysql_fetch_assoc($idzapytania))
```

```
{
```

```
    echo $wiersz["id"] . ' : '. $wiersz["nazwisko"] . '<br />';
```

```
}
```

Jak widać tym razem do pola id, nazwisko itd., odwołamy się poprzez nazwę pola a nie jak poprzednio przez kolejny numer porządkowy.

Która metoda jest lepsza? Obie są dobre – kwestia gustu i upodobań...

AUTO_INCREMENT – jakie ID...?

`mysql_insert_id()` zwraca ID wygenerowane dla pola z własnością `AUTO_INCREMENT`.

Funkcja zwróci ID wygenerowane automatycznie przez ostatnią operację `INSERT` używającą podanego identyfikatora połączenia. Jeżeli identyfikator połączenia nie został podany, wykorzystywane jest ostatnio otwarte połączenie.

Przykład:

```
mysql_connect('localhost', 'uzytkownik', 'haslo') ;  
mysql_select_db('baza');  
mysql_query("INSERT INTO mytable (product) values ('kossu')");  
echo "Ostatnio dodany rekord ma id %d\n".mysql_insert_id();
```

Więcej przydatnych funkcji do operowania z poziomu PHP w MySQL można znaleźć w manualu PHP.

KONIEC