

Fetchowanie danych z użyciem mysqli

Nauczmy się, jak pobierać dane z bazy danych MySQL przy użyciu rozszerzenia mysqli. To jedno z podstawowych narzędzi, jakie daje nam PHP, i bardzo często pojawia się w zadaniach praktycznych na egzaminie INF.03. Angielskie słówko *fetch* oznacza po polsku *aportować, sprowadzić, wydobyć*. I tym właśnie jest fetchowanie informacji z bazy danych – przyniesieniem ich do skryptu PHP, celem dalszego przetworzenia, np. pokazania na stronie instrukcją

```
echo .
```

W tej lekcji poznamy cztery sposoby fetchowania danych z wyniku wykonania zapytania SELECT:

- `mysqli_fetch_row()` – dane jako tablica indeksowana (numerowanie szuflad).
- `mysqli_fetch_assoc()` – dane jako tablica asocjacyjna (nazywanie szuflad).
- `mysqli_fetch_array()` – dane jako tablica indeksowana i asocjacyjna (dostępne zarówno pod numerem szufladki jak i jej nazwą, ale kosztem podwójnego wyjęcia danych – redundancja).
- `mysqli_fetch_object()` – dane wyjęte jako obiekt, a nie tablica.

Jeżeli temat zapytań fetchowania danych nadal okaże się dla nas częściowo niejasny lub potrzebowalibyśmy zrealizowania krok po kroku skryptu współpracującego z bazą danych, to gorąco zachęcam do zajrzenia do tutoriala video: [Współpraca PHP z bazą danych](#).

Połączenie z bazą danych

Przypomnijmy gwoili formalności, że oczywiście do wykonania zapytania SELECT trzeba połączyć się z bazą danych – więcej informacji na temat w lekcji: [Nawiązanie i zamykanie połączenia PHP z bazą danych](#). Połączenie można łatwo zrealizować w mysqli np. w taki sposób:

```
<?php
$connection = mysqli_connect("localhost", "root", "", "vod");
if (!$connection) {
    die("Błąd połączenia: " . mysqli_connect_error());
}
?>
```

Tworzymy połączenie za pomocą funkcji

`mysqli_connect()`, przekazując – przypomnijmy: hosta, nazwę użytkownika, hasło do usługi MariaDB/MySQL i nazwę bazy danych.

Wykonanie zapytania SELECT

To, o czym czasami zapominają ludzie, to fakt, iż samo włożenie tekstu kwerendy do zmiennej to jeszcze nie wykonanie jej w DBMS. Potrzebne jest wywołanie funkcji

`mysqli_query()`, a ponieważ mamy tutaj do czynienia z zapytaniem SELECT, to baza danych “odpowie” nam danymi – stąd konieczne jest włożenie tej odpowiedzi do zmiennej `$result`:

```
$sql = "SELECT * FROM filmy";
$result = mysqli_query($connection, $sql);
```

I teraz fetchowanie – tłumacząc to w kontekście mechaniki PHP – jest tak naprawdę “wycięciem” z całego zestawu wyjętych informacji, poszczególnych wartości pól z tabeli bazy danych do tablicy, której najczęściej nadajemy nazwę

`$row`. Kiedy zaś umieścimy dane w tablicy, to potem możemy je już np. wypisywać instrukcją `echo` albo używać w obliczeniach.

Załóżmy, że chcemy pobrać z bazy danych VOD imię, nazwisko i płeć aktorów urodzonych w USA. Fragment kodu realizujący wykonanie kwerendy może wyglądać następująco:

```
$query = "SELECT imie, nazwisko, plec FROM aktorzy WHERE kraj_ur = 'USA'";  
$result = mysqli_query($connection, $query) or die("Problem z zapytaniem!");
```

Zmienne użyte w kodzie:

- `$connection` – połączenie z bazą danych utworzone wcześniej
- `$query` – treść zapytania SQL jako łańcuch,
- `$result` – odpowiedź bazy danych, którą traktujemy mentalnie jak tabelę w pamięci PHP (tak naprawdę, “pod maską” jest to obiekt klasy `mysqli_result`).

1. Pobieranie wyników funkcją `mysqli_fetch_row()`

Funkcja

`mysqli_fetch_row()` zwraca każdy rekord jako tablicę indeksowaną liczbowo (0, 1, 2...). Wygodne, gdy nie zależy nam na nazwach kolumn albo chcemy iterować po wyjętych szufladach. Kod pętli fetchującej:

```
while ($row = mysqli_fetch_row($result)) {  
    echo $row[0]. " ".$row[1]. " ".$row[2]. "<br>";  
}
```

Odpowiedź na zapytanie stanowi cały zestaw danych, który należy przetworzyć (pociąć na pojedyncze wartości). W praktyce oznacza to więc najczęściej skonstruowanie pętli wykonującej się – jak to mówimy – “dopóki istnieją rekordy”, zaś wewnątrz niej, w ramach każdej iteracji należy:

- wyciąć z całej dwuwymiarowej tablicy bieżący rekord (wiersz),
- z tego rekordu wypisać wartości poszczególnych atrybutów (kolumn).

Pomocniczy pojemnik

`$row` (*ang. row = wiersz*) zawiera zawsze jeden rekord wyjęty ze wszystkich wierszy zwróconych przez kwerendę (czyli jest to tablica już tylko jednowymiarowa). Oczywiście to wyjmowanie do `$row` musi być realizowane w pętli, gdyż rekordów przesłanych w wyniku zapytania może być dużo. W przykładzie wybrano sterowaną warunkiem pętlę `while`, gdyż automatycznie przerwie ona działanie, gdy wszystkie rekordy zostaną wypisane. Indeksy w `$row` oznaczają tu kolejno: 0 – imię, 1 – nazwisko, 2 – płeć. A zatem o

2. Pobieranie wyników funkcją `mysqli_fetch_assoc()`

Funkcja

`mysqli_fetch_assoc()` tworzy tablicę asocjacyjną, gdzie indeksami są nazwy kolumn wyjętych z bazy danych. To najczytelniejszy sposób pracy z wynikami, bo w kodzie “widać” co wypisujemy:

```
while ($row = mysqli_fetch_assoc($result)) {
```

```
echo $row["imie"]." ".$row["nazwisko"]." ".$row["plec"]."<br>";  
}
```

Ten zapis jest wygodny w dużych projektach – jeśli kolejność kolumn się zmieni, oczywiście nadal zadziała poprawnie.

3. Pobieranie wyników funkcją `mysqli_fetch_array()`

Funkcja

`mysqli_fetch_array()` umożliwia dostęp zarówno po indeksach liczbowych, jak i asocjacyjnych. Domyślnie zawiera przez to duplikaty danych – dostępne są oba więc typy indeksów – zarówno słowne, jak i numeryczne:

```
while ($row = mysqli_fetch_array($result)) {  
    echo $row[0]." ".$row["nazwisko"]." ".$row["plec"]."<br>";  
}
```

Można też jednak dodać drugi argument do funkcji, np.

`MYSQLI_ASSOC`, aby ograniczyć się tylko do indeksów asocjacyjnych – wtedy każda kolumna dostępna będzie wyłącznie pod swoją nazwą (bez indeksów liczbowych), a to zapobiega powtarzaniu danych w pamięci:

```
while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {  
    echo $row["imie"]." ".$row["nazwisko"]." ".$row["plec"]."<br>";  
}
```

Alternatywnie można użyć

`MYSQLI_NUM`, aby ograniczyć się tylko do indeksów liczbowych, jeśli ktoś preferuje operowanie na numerach kolumn:

```
while ($row = mysqli_fetch_array($result, MYSQLI_NUM)) {  
    echo $row[0]." ".$row[1]." ".$row[2]."<br>";  
}
```

Jednak domyślnie (bez drugiego argumentu)

`mysqli_fetch_array()` zwraca tablicę łączoną (mieszaną) – zawiera zarówno indeksy liczbowe, jak i asocjacyjne. To jednak oznacza, że każda wartość przechowywana jest w tablicy dwukrotnie – co może zwiększyć zużycie pamięci przy dużych wynikach kwerend.

4. Pobieranie wyników obiektowo: `mysqli_fetch_object()`

Jeśli wolimy używać składni obiektowej, możemy skorzystać z funkcji

`mysqli_fetch_object()`. Każdy rekord zwracany jest jako obiekt z właściwościami o nazwach kolumn:

```
while ($row = mysqli_fetch_object($result)) {  
    echo $row->imie." ".$row->nazwisko." ".$row->plec."<br>";  
}
```

Ten sposób bywa wygodny, jeśli chcemy przekazać rekord dalej jako obiekt lub stosować notację obiektową np. w szablonach HTML. Zamiast dostępu "tablicowego" do

`$row` – na przykład `$row[1]` używamy zapisu `obiekt->atrybut` – na przykład: `$row->nazwisko`.

Liczba zwróconych rekordów: `mysqli_num_rows()`

Po wykonaniu zapytania SELECT możemy także odczytać jaką liczbę rekordów zwróciła nam w zapytaniu baza:

```
$ile = mysqli_num_rows($result);  
echo "Liczba rekordów: ".$ile;
```

Funkcja ta przydaje się także np. do wyświetlenia komunikatu “Brak wyników”, jeśli nie znaleziono żadnych danych – wystarczy skonstruować instrukcję warunkową sprawdzającą, czy wartość

`$ile` jest równa 0.

Podsumowanie metod fetchujących

W praktyce wybór metody pobierania wartości zależy od programisty. Indeksy słowne czy atrybuty obiektowe są czytelne, natomiast indeksy numeryczne pozwalają nam iterować po wyjętych wynikach. Reasumując:

- Fetchowanie to proces pobierania (wydobywania) danych z bazy, a mówiąc dokładniej: z odpowiedzi na zapytanie SELECT, której udziela baza danych.
- Najpierw należy rzecz jasna nawiązać połączenie z bazą danych przy pomocy `mysqli_connect()`.
- Zapytanie SQL wykonujemy funkcją `mysqli_query()` i wynik zapisujemy do zmiennej, np. `$result`.
- Do przetwarzania wyniku (czyli „fetchowania”) służą cztery funkcje:
 - `mysqli_fetch_row()` – zwraca rekord jako tablicę z indeksami numerycznymi (0, 1, 2...).
 - `mysqli_fetch_assoc()` – tworzy tablicę asocjacyjną z nazwami kolumn jako kluczami.
 - `mysqli_fetch_array()` – łączy oba podejścia, ale może dublować dane w pamięci.
 - `mysqli_fetch_object()` – rekordy zwracane są jako obiekty (dostęp przez `->`).
- Do każdej z metod stosujemy zwykle pętlę `while`, aby przetworzyć wszystkie rekordy.
- Dodatkowo możemy sprawdzić, ile rekordów zwróciło zapytanie SELECT – za pomocą `mysqli_num_rows()`.

Dzięki znajomości fetchowania możemy wygodnie pobierać i wyświetlać dane z bazy – to jeden z filarów dynamicznych aplikacji webowych, zarówno na egzaminie INF.03, jak i w realnych projektach.