



Współpraca bazy danych ze stroną internetową, etap II

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Film samouczek](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Witryna internetowa korzystająca z bazy danych bardzo często przetwarza olbrzymie ilości informacji zgromadzonych przez użytkowników. Blogi wyświetlają napisane przez autora artykuły, serwisy społecznościowe prezentują udostępnione posty oraz zdjęcia, zaś dzienniki elektroniczne zapewniają uczniowi wgląd do swoich ocen i frekwencji.

Przetwarzanie dużej ilości danych i wyświetlanie ich w postaci czytelnego raportu to podstawowa funkcjonalność współczesnej witryny. W praktyce więc nie wystarczy wiedzieć, jak nawiązać połączenie z bazą, ustawić poprawne kodowanie znaków diakrytycznych oraz wykonać kwerendę przetwarzającą rekordy. Programista webowy powinien także sprawnie generować raporty, powstałe w oparciu o odpowiedzi na zapytanie wyszukiujące, dostarczone przez bazę danych.

W skład serii wchodzi również e-materiały:

- [Współpraca bazy danych ze stroną internetową, etap I,](#)
- [Współpraca bazy danych ze stroną internetową, etap III,](#)
- [Współpraca bazy danych ze stroną internetową, etap IV.](#)

Twoje cele

- Skonstruujesz swój pierwszy raport z bazy danych na stronie internetowej.

- Ocenisz różnice pomiędzy trzema klasycznymi sposobami przetwarzania informacji zwrotnej, przesłanej do witryny jako rezultat zapytania SELECT.
- Z użyciem znaków sterujących w PHP, wygenerujesz czytelnie sformatowany, wynikowy kod źródłowy HTML raportu.

Przeczytaj

Przeanalizujemy następującą tabelę znajdującą się w bazie danych MySQL:

id	imie	nazwisko	plec	kraj_ur	data_ur
1	Tom	Hanks	m	USA	1956-08-09
2	Jennifer	Lawrence	k	USA	1990-08-15
3	Penelope	Cruz	k	Hiszpania	1974-04-28
4	Sean	Connery	m	Szkocja	1930-08-25
5	Jodie	Foster	k	USA	1962-10-19
6	Christian	Bale	m	Wielka Brytania	1974-01-30
7	Natalie	Portman	k	USA	1981-06-09
8	Nicolas	Cage	m	USA	1964-01-07

Tabela nosi nazwę aktorzy i przechowuje (na potrzeby serwisu recenzującego w internecie filmy) informacje na temat znanych aktorów hollywoodzkich. Cała baza danych nosi nazwę filmoteka. Tabela zawiera kolejno:

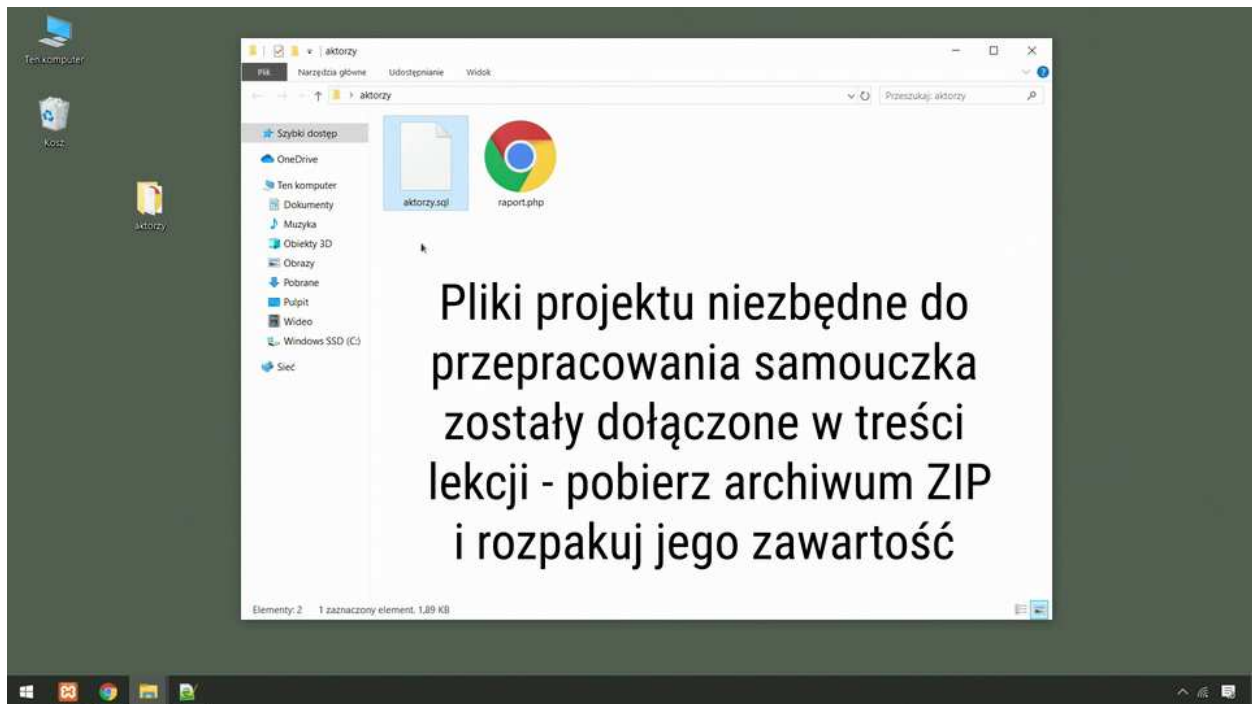
- klucz główny id (typu całkowitego INT, autoinkrementowany),
- atrybuty typu TINYTEXT: imie, nazwisko, plec (płeć), kraj_ur (kraj urodzenia),
- atrybut typu DATE o nazwie: data_ur (data urodzenia).

Archiwum ZIP potrzebne do wykonania zadań w tym e-materiale znajdziesz poniżej:

Plik o rozmiarze 2.16 KB w języku polskim

Skrypt wstawiający rekord do bazy

Założmy, że chcemy wstawić do tabeli kolejny rekord reprezentujący aktora pochodzenia austriackiego – Arnolda Schwarzeneggera.



Film dostępny pod adresem </preview/resource/R1EvZXVdYtn5c>

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Film-samouczek: Współpraca bazy danych ze stroną internetową, etap drugi, część pierwsza.

Skrypt PHP realizujący zadanie dopisania do tabeli kolejnego rekordu prezentuje się następująco:

```
1 $c = mysqli_connect("localhost", "root", "", "filMOTEKA") or die(
2 mysqli_set_charset($c, "utf8");
3 $q = "INSERT INTO aktorzy VALUES (NULL, 'Arnold', 'Schwarzenegger
4 mysqli_query($c, $q);
5 mysqli_close($c);
```

Dla przypomnienia, w liniach o kolejnych numerach zrealizowano:

1. Akt **nawiązania połączenia z bazą danych** lub w razie niepowodzenia **wyświetlenie komunikatu o błędzie** i wstrzymanie dalszego wykonania skryptu.
2. Przypisanie do obsługi komunikacji **zestawu znaków UTF-8**, zawierającego polskie znaki diakrytyczne.
3. Zapisanie **treści zapytania SQL** w specjalnie utworzonej zmiennej PHP – całe zapytanie umieszczono w cudzysłowie, zaś łańcuchy wewnątrz zapytania zamknięto w apostrofach.

4. Rzeczywiste **wykonanie kwerendy** umieszczonej uprzednio w zmiennej `$q`, za pośrednictwem połączenia o identyfikatorze `$c`.
5. **Zamknięcie aktywnego połączenia** o identyfikatorze `$c`.

Okazuje się, że wykonanie kwerendy INSERT, UPDATE lub DELETE na bazie danych, jest zadaniem stosunkowo trywialnym. Wystarczyło bowiem przygotować **treść kwerendy** w pomocniczej zmiennej `$q`, po czym rzeczywiście **wykonać ją** funkcją `mysqli_query()`, używając otwartego połączenia o identyfikatorze `$c`.

Raport wyszukiujący informacje w bazie

Jednak zadanie to staje się trudniejsze, jeżeli chcemy **wyszukać informacje**, czyli wykonać zapytanie SELECT. Baza danych musi wówczas także **odpowiedzieć na kwerendę** – zwracając do skryptu PHP mnóstwo wyszukanych (zgodnie z podanymi kryteriami) informacji. Proces przetwarzania takiej odpowiedzi określamy w nomenklaturze bazodanowej **fetchowaniem**.

Ciekawostka

Angielskie słówko *fetch* oznacza: sprowadzić, przynieść, wydobyć. Doskonale oddaje to naturę procesu **wyjmowania pojedynczych rekordów** z całej przesłanej odpowiedzi bazy danych na zapytanie wyszukiujące SELECT.

Założmy, że mamy wyszukać w tabeli aktorzy informacje o aktorach (imię, nazwisko, płeć) urodzonych w Stanach Zjednoczonych. Wykonanie takiej kwerendy wyszukiwanej będzie zatem wymagało **przechowania odpowiedzi bazy danych** w dodatkowej, pomocniczej zmiennej PHP:

```
1 $q = "SELECT imie, nazwisko, plec FROM aktorzy WHERE kraj_ur = 'U  
2 $result = mysqli_query($c, $q) or die("Problem z kwerendą!");
```

Odpowiedź bazy na zapytanie zapisane w zmiennej `$q` zostanie umieszczona w pojemniku o nazwie `$result`. Dodatkowo, z użyciem znanej nam klauzuli `or die()` przerwiemy dalsze przetwarzanie skryptu, jeśli wystąpi problem z wykonaniem kwerendy. Opisane przekształcenie skryptu PHP w raport prześledźmy dokładnie w filmie samouczku:

```
C:\xampp\htdocs\aktorzy\raport.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

raport.php
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="utf-8">
  <title>Raport</title>
</head>
<body>
<?php
    $c = mysqli_connect("localhost", "root", "", "filmoteka") or die("Problem z serwerem!");
    mysqli_set_charset($c, "utf8");

    $q = "INSERT INTO aktorzy VALUES (NULL, 'Arnold', 'Schwarzenegger', 'm', 'Austria',
    '1947-07-30')";

    mysqli_query($c, $q);
    mysqli_close($c);
?>
</body>
</html>
```

Film dostępny pod adresem </preview/resource/ROEHvI7RpjN8>

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Film-samouczek: Współpraca bazy danych ze stroną internetową, etap drugi, część druga.

Zawartość zmiennej `$result` nie jest pojedynczą wartością jakiegokolwiek atrybutu. Jest to obiekt klasy `mysqli_result`, który jednak możesz roboczo traktować jako **dwuwymiarową tablicę** (tabelę złożoną z wierszy i kolumn) składająca się z **pięciu wyjętych rekordów** (aktorów), z których każdy posiada po **trzy atrybuty** (imię, nazwisko, płeć):

imię	nazwisko	plec
Tom	Hanks	m
Jennifer	Lawrence	k
Jodie	Foster	k
Natalie	Portman	k
Nicolas	Cage	m

Zatem odpowiedź na zapytanie stanowi cały **zestaw danych**, który należy **przetworzyć** (pociąć na pojedyncze wartości). W praktyce oznacza to 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).

Pętla, wypisująca rekord po rekordzie, cały zestaw zwróconych w `$result` prezentuje się następująco:

```
1 while ($row = mysqli_fetch_assoc($result)) {  
2     echo $row['imie']." ".$row['nazwisko']." ".$row['plec']."<br>  
3 }
```

Pomocnicza zmienna `$row` (ang. *row* = wiersz) zawiera zawsze **jeden rekord** wyjęty ze wszystkich wierszy zwróconych przez kwerendę (czyli jest to tablica już tylko jednowymiarowa). Oczywiście to wyjmowanie do `$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.

Natomiast do poszczególnych **wartości pól** w rekordzie dostajemy się, podając **indeks w tablicy** – inny dla każdej szufladki. W metodzie `mysqli_fetch_assoc()` indeksy są **asocjacyjne**, czyli skojarzeniowe. Jest to **słowna reprezentacja**, będąca po prostu nazwą kolumny występującej w bazie danych.

Operator kropki: `.` oznacza w PHP tzw. **konkatenację** łańcuchów, która umożliwia wplecenie wartości zmiennych PHP prosto w generowany wynikowy kod źródłowy HTML.

Trzy sposoby przetwarzania danych

W języku PHP wyróżniamy trzy podstawowe odmiany operacji **fetch**, realizującej wkładanie pojedynczych rekordów do pomocniczej tablicy jednowymiarowej `$row`:

- `mysqli_fetch_row()` – **przetwarzanie liczbowe** – ponumeruj (od zera) szufladki tablicy jednowymiarowej;
- `mysqli_fetch_assoc()` – **wyjmowanie asocjacyjne (skojarzeniowe)** – ponazywaj słownie szufladki tablicy jednowymiarowej, nadając im zawsze taki sam identyfikator, jaki posiadają kolumny w bazie danych;
- `mysqli_fetch_array()` – **przetwarzanie równocześnie numeryczne i asocjacyjne** (dostępne są oba rodzaje indeksów); wadą tej metody jest redundancja (powtarzanie) danych.

W zależności od wyboru metody przetwarzania, nieco inny wygląd przyjmie pętla wypisująca wartości odczytanych pól. W poprzednim przykładzie, w którym użyto funkcji `mysqli_fetch_assoc()`, indeksy tablicy były słowne. Natomiast w przypadku użycia metody `mysqli_fetch_row()` kolumny indeksujemy numerycznie:



```
1 while ($row = mysqli_fetch_assoc($result)) {  
2     echo $row[0]." ".$row[1]." ".$row[2]."<br>";  
3 }
```

Wynik wypisania wartości w przeglądarce jest identyczny jak ostatnio, zmieniły się jedynie **indeksy tablicy** jednowymiarowej `$row` – zamiast wartości słownych, zastosowano teraz kolejne liczby (inkrementowane od zera). Ostateczny wybór metody przetwarzania rezultatu zależy najczęściej tylko od osobistej preferencji programisty. Indeksy numeryczne łatwiej można przetwarzać w pętlach, zaś słowne szybciej pozwalają zorientować się, co dokładnie w sposób iteracyjny wypisujemy.

Pobranie liczby zwróconych rekordów

Dzięki zastosowaniu funkcji o nazwie `mysqli_num_rows()` istnieje dodatkowa możliwość odczytania **liczby rekordów** zwróconych przez zapytanie `SELECT`:

```
1 $q = "SELECT imie, nazwisko, plec FROM aktorzy WHERE kraj_ur = 'U  
2 $result = mysqli_query($c, $q) or die("Problem z kwerendą!");  
3  
4 $ile = mysqli_num_rows($result);  
5 echo "Liczba aktorów amerykańskich w bazie: ".$ile;
```

Jedynym argumentem funkcji zastosowanej w linii czwartej jest identyfikator zmiennej, w której umieszczono wyniki zapytania `SELECT` – w naszym przykładzie jest to `$result`. W pomocniczej zmiennej `$ile` znajdzie się zatem liczba zwróconych tym zapytaniem rekordów.

Ciekawostka

Należy bezwzględnie pamiętać, aby funkcję `mysqli_num_rows()` umiejscowić w kodzie pod wywołaniem metody `mysqli_query()`, nigdy powyżej, tak aby zmienna `$result` przechowywała już w sobie zwrócony wynik zapytania.

Słownik

fetchowanie

operacja wyjmowania wartości pojedynczych rekordów z całej przesłanej odpowiedzi bazy danych na zapytanie wyszukujące `SELECT`

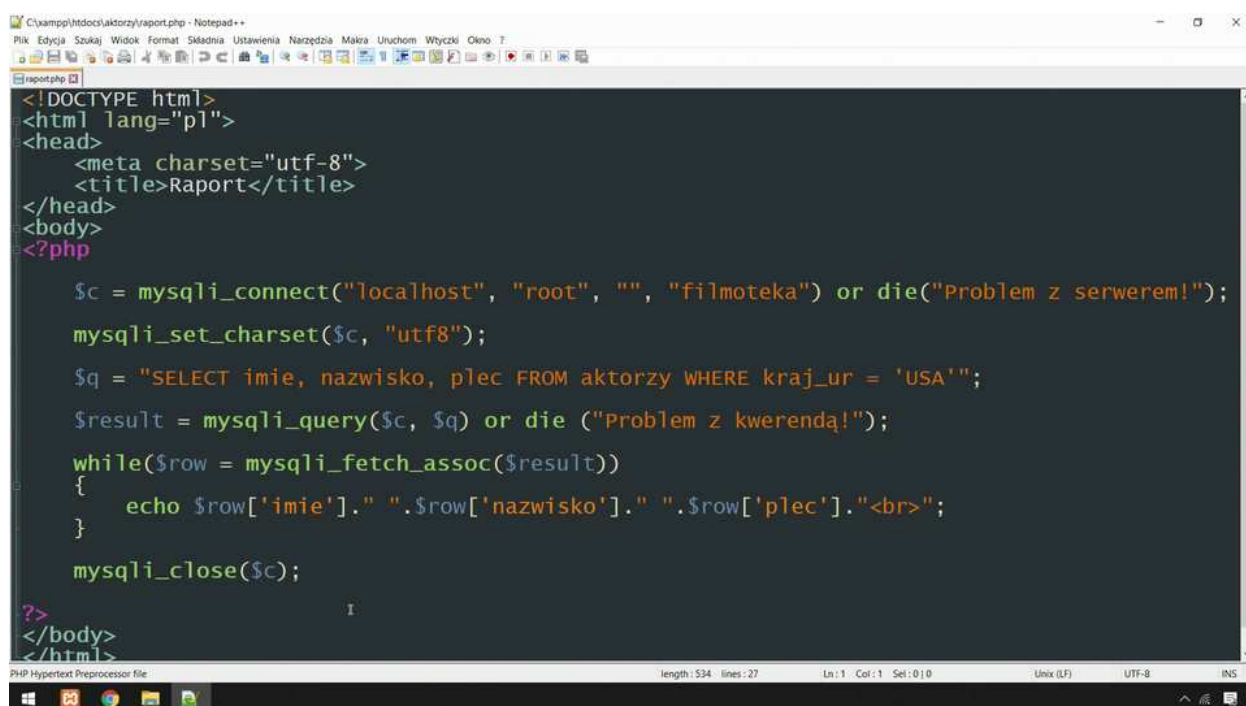
konkatenacja

proces łączenia (sklejania ze sobą) dwóch lub więcej ciągów znaków

Film samouczek

Znaki HTML i znaki sterujące w PHP

Zrealizujmy w praktyce **czytelne formatowanie** generowanego w skryptach kodu źródłowego HTML. Z filmu samouczka dowiemy się, jak zamykać surowe wartości **wyjęte z bazy danych**, wewnątrz odpowiednich **znaczników HTML** oraz jak stosować tzw. **znaki sterujące** (formatujące wyjście) w PHP:



```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="utf-8">
  <title>Raport</title>
</head>
<body>
<?php
    $c = mysqli_connect("localhost", "root", "", "filmoteka") or die("Problem z serwerem!");
    mysqli_set_charset($c, "utf8");
    $q = "SELECT imie, nazwisko, plec FROM aktorzy WHERE kraj_ur = 'USA'";
    $result = mysqli_query($c, $q) or die ("Problem z kwerendą!");
    while($row = mysqli_fetch_assoc($result))
    {
        echo $row['imie']." ".$row['nazwisko']." ".$row['plec']."<br>";
    }
    mysqli_close($c);
?>
</body>
</html>
```

Film dostępny pod adresem </preview/resource/RzNA8gCHAKqRi>

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Film nawiązujący do treści materiału dotyczącej surowych wartości wyjętych z bazy danych, wewnątrz odpowiednich znaczników HTML oraz stosowania tak zwanych znaków sterujących.

Plik o rozmiarze 793.00 B w języku polskim

Polecenie 1

Stwórz w HTML raport z bazy danych wykorzystujący w strukturze HTML tabelę, czyli znacznik `<table>`. W raporcie wypisz z bazy danych wszystkie dane (poza id) dotyczące aktorek urodzonych po roku 1980.

W przedstawionej na filmie wersji skryptu, informację o liczbie aktorów **zamknięto w paragrafie** `<p></p>`, zaś **wypisanie aktorów** zrealizowano **z użyciem listy** nienumerowanej `<ul></ul>`.

```
1 echo "<p>Liczba aktorów amerykańskich w bazie: ".$ile."</p>";
2 echo "\r\n<ul>\r\n";
3
4 while ($row = mysqli_fetch_assoc($result)) {
5     echo "\t<li>".$row['imie']." ".$row['nazwisko']." ( ".$row['pl
6 }
7
8 echo "</ul>\r\n";
```

Ciekawostka

Otwarcie listy `<ul>` zostało wstawione **przed pętlą**, ponieważ umieszczenie rozpoczęcia listy wewnątrz instrukcji iteracyjnej stworzyłoby efektywnie aż **pięć list** zamiast **jednej**. To właśnie elementów listy `<li></li>` ma być pięć. Zgodnie z tą logiką, również zamknięcie `</ul>` całej listy zostało zaimplementowane już po zakończeniu ciała pętli `while`.

Dodatkowo, w zaprezentowanym fragmencie pojawiły się specjalne **znaki sterujące PHP**, określające sposób formatowania generowanego źródła HTML przez przeglądarkę:

- `\r\n` – zrealizowanie przejścia do nowej linii,
- `\t` – wstawienie tabulacji (odstępu poziomego).

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Wskaż, która z poniższych instrukcji w PHP jest konkatencją.

☐ `echo "Imię: " + $imie;`

☐ `echo "Imię: " ++ $imie;`

☐ `echo "Imię: " . $imie;`

☐ `echo "Imię: " % $imie;`

Ćwiczenie 2



Wybierz właściwą odpowiedź.

W języku PHP pobrano z bazy danych wyniki działania kwerendy za pomocą polecenia `mysqli_query()`. Wskaż, które polecenie zastosujesz, aby otrzymać liczbę wierszy danych zwróconych przez kwerendę.

☐ `mysqli_num_rowcount()`

☐ `mysqli_num_rows()`

☐ `mysqli_num_fetch_row()`

☐ `mysqli_num_field_length()`

Ćwiczenie 3



Uporządkuj etapy przetwarzania odpowiedzi bazy danych na zapytanie SELECT.

mysqli_set_charset()



while(\$row = mysqli_fetch_assoc(\$result))



mysqli_connect() or die()



\$result = mysqli_query() or die()



mysqli_close()



Ćwiczenie 4



Wstaw w odpowiednie miejsca fragmenty algorytmu współpracy bazy danych z witryną.

\$result = (\$c, \$q) or die("Problem z kwerendą!");

```
while ($row = ()) {  
     $row[0]." ".$row[1]." ".$row[2];  
}
```

echo

mysqli_fetch_assoc

mysqli_query

mysqli_fetch_row

\$result

Ćwiczenie 5



Połącz w pary rodzaje instrukcji fetch z opisem sposobu ich działania.

mysqli_fetch_row()

wyjmowanie pól zarówno numeryczne,
jak i słowne

mysqli_fetch_assoc()

indeksowanie rezultatów zapytania
tylko numerycznie

mysqli_fetch_array()

przetwarzanie wartości z użyciem
słownych identyfikatorów

Ćwiczenie 6



Wskaż, co należy zrobić, aby wyświetlić wszystkie wyniki zwrócone przez bazę danych wskutek wykonania instrukcji: `$result = mysqli_query($c, "SELECT imie FROM pracownicy WHERE wiek>30");`.

- ☐ wyświetlić zawartość zmiennej `$result["imie"]`
- ☐ wyświetlić zawartość zmiennej `$result[0]`
- ☐ zastosować pętlę `while` zawierającą funkcję `mysqli_num_rows()`
- ☐ zastosować pętlę `while` zawierającą funkcję `mysqli_fetch_row()`

Ćwiczenie 7



Wstaw w odpowiednie miejsca fragmenty algorytmu przetwarzania odpowiedzi bazy danych na zapytanie `SELECT`.

```
$q = "SELECT imie, nazwisko FROM aktorzy ORDER BY nazwisko";
```

```
$result = mysqli_($c, $q) or die("Problem z kwerendą!");  
echo "<p>Nazwiska aktorów ułożone alfabetycznie A-Z</p>";
```

```
while ( = mysqli_fetch_($result)) {  
echo "<p>". ".</p>";  
}
```

row

DESC

ASC

query

assoc

\$row[1]

\$row[0]

\$row

Ćwiczenie 8



Przypisz instrukcje biblioteki `mysqli` do odpowiedniej grupy.

obsługa połączenia z bazą danych

`mysqli_fetch_array()`

`mysqli_set_charset()`

`mysqli_num_rows()`

przetwarzanie zwróconych rekordów

`mysqli_fetch_assoc()`

`mysqli_close()`

`mysqli_fetch_row()`

informacja o ilości zwróconych rekordów

`mysqli_connect()`

Dla nauczyciela

Autor: Mirosław Zelent

Przedmiot: Informatyka

Temat: Współpraca bazy danych ze stroną internetową, etap II

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

4) przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym:

d) projektuje i tworzy relacyjną bazę złożoną z wielu tabel oraz sieciową aplikację bazodanową dla danych związanych z rozwiązywanym problemem, formułuje kwerendy, tworzy i modyfikuje formularze oraz raporty, stosuje język SQL do wyszukiwania informacji w bazie i do jej modyfikacji, uwzględnia kwestie integralności danych, bezpieczeństwa i ochrony danych w bazie,

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Skonstruujesz swój pierwszy raport z bazy danych na stronie internetowej.
- Ocenisz różnice pomiędzy trzema klasycznymi sposobami przetwarzania informacji zwrotnej, przesłanej do witryny jako rezultat zapytania SELECT.
- Z użyciem znaków sterujących w PHP, wygenerujesz czytelnie sformatowany, wynikowy kod źródłowy HTML raportu.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- metody aktywizujące.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- telefony z dostępem do internetu.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Współpraca bazy danych ze stroną internetową, etap II”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj”.
2. Chętny lub wybrany uczeń przygotowuje rozwiązanie polecenia nr 1: „Stwórz w HTML raport z bazy danych wykorzystujący w strukturze HTML tabelę, czyli znacznik

Faza wstępna:

1. Nauczyciel wyświetla temat oraz cele zajęć, omawiając lub ustalając razem z uczniami kryteria sukcesu.

2. Prowadzący prosi uczniów, aby zgłaszali swoje propozycje pytań do tematu. Jedna osoba może zapisywać je na tablicy. Gdy uczniowie wyczerpią swoje pomysły, a pozostały jakieś ważne kwestie do poruszenia, nauczyciel je dopowiada.

Faza realizacyjna:

1. **Praca z tekstem.** Uczniowie analizują treści z sekcji „Przeczytaj” wyświetlone na tablicy.
2. Chętni lub wybrani uczniowie przedstawiają swoje rozwiązanie polecenia nr 1 i wyjaśniają otrzymany rezultat. Pozostali uczniowie weryfikują poprawność rozwiązań lub przedstawiają alternatywne sposoby.
3. **Praca z multimediami.** Nauczyciel wyświetla zawartość sekcji „Audiobook”. Uczniowie wspólnie zapoznają się z treścią zawartego w niej multimediu. Zapisują ewentualne problemy i pytania. Po czym następuje dyskusja, w trakcie której nauczyciel wyjaśnia niezrozumiałe treści.
4. W kolejnym etapie uczniowie dobierają się w pary i wykonują ćwiczenia nr 1-5. Następnie konsultują swoje rozwiązania z inną parą uczniów i ustalają jedną wersję odpowiedzi.

Faza podsumowująca:

1. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności, omawia ewentualne problemy podczas rozwiązywania ćwiczeń.

Praca domowa:

1. Uczniowie opracowują FAQ (minimum 3 pytania i odpowiedzi) do tematu lekcji („Współpraca bazy danych ze stroną internetową, etap II”).
2. Uczniowie wykonują ćwiczenia 6-8 z sekcji „Sprawdź się”.

Wskazówki metodyczne:

- Treści w sekcji „Przeczytaj” można wykorzystać jako podsumowanie i utrwalenie wiedzy uczniów.