

Opracował: Henryk Szantula, Katarzyna Sitko. Źródło: PHP i MySQL , Kewin Yank

Wprowadzenie do systemu MySQL

Język skryptowy PHP już znamy, czas na bazę danych MySQL i język SQL.

Serwer baz danych (MySQL), to program, który potrafi przechowywać wielką ilość informacji w uporządkowanym formacie.

Definicja Bazy Danych - zbiór danych zapisanych zgodnie z określonymi regułami. W węższym znaczeniu obejmuje dane cyfrowe gromadzone zgodnie z zasadami przyjętymi dla danego programu komputerowego specjalizowanego do gromadzenia i przetwarzania tych danych.

Najwcześniejsze znane użycie terminu baza danych miało miejsce w listopadzie 1963, kiedy odbyło się sympozjum pod nazwą "Development and Management of a Computer-centered Data Base

Budowa i Rodzaje Baz Danych

- Baza danych jest złożona z różnych elementów. Najważniejszymi z nich jest rekord podzielony na kilka pól, w których są przechowywane informacje poszczególnych kategorii
- Bazy danych można podzielić według struktur organizacji danych, których używają:
 - Bazy proste:
 - kartotekowe
 - hierarchiczne
 - Bazy złożone:
 - Relacyjne
 - obiektowe
 - relacyjno-obiektowe
 - strumieniowe
 - temporalne
 - nierelacyjne (NoSQL)

Sprawdzamy XAMPP

Z poziomu wiersza poleceń wpisz: mysqladmin -u root (status) W odpowiedzi powinieneś uzyskać odpowiedź serwera:

Heniek@LENOVO-HENIEK c:\xampp # mysqladmin -u root status Uptime: 292 Threads: 1 Questions: 1 Slow queries: 0 Opens: 17 Flush tables: 1 Open tables: 11 Queries per second avg: 0.003

Dla potrzeb edukacyjnych w pracowni nie będziemy zakładać hasła dla użytkownika root. W praktyce i w domu jest to niedopuszczalne. Zadbaj o nadanie hasła dla konta root. <u>KONIECZNIE</u> mysgladmin -u root –p password "nowehasło"

Logowanie na serwerze

Na potrzeby tego kursu utworzymy bazę danych z dowcipami.

Pracę rozpoczynamy od zalogowania się na serwerze na koncie root. Wpisz w wierszu poleceń:

mysql –u root

mysql -u root Welcome to the MariaDB monitor. Commands end with ; or \g. Your MariaDB connection id is 20 Server version: 10.1.25-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>

Logowanie na serwerze

 Po zalogowaniu na serwerze mogę sprawdzić, jakie bazy są przechowywane na serwerze korzystam z polecenia:

SHOW DATABASES;

 Należy pamiętać, aby każde polecenie kończyć średnikiem!!!

Coś więcej o serwerze MySQL

Wykonamy kilka prostych poleceń, aby dowiedzieć się czegoś więcej o serwerze MySQL.

Serwer MySQL może obsługiwać różne bazy danych należące do różnych portali www. Lista baz jakie są zainstalowane na moim serwerze jest następująca:

Information_schema – baza zawierająca dane

o wszystkich innych bazach obsługiwanych przez MySQL.

mysql – zawiera informację o użytkownikach, ich hasłach oraz uprawnieniach.

test – przykładowa baza danych.



Usuwamy Bazę Danych

Przykładowa baza danych "test" nie zawiera żadnych tabel. Możemy ją spokojnie usunąć poleceniem:

mysql-> DROP DATABASE test;

UWAGA !!!

 Image: state with the state with the

System posłusznie usunie bazę danych, nie pytając czy jesteśmy pewni. Ta więc przed naciśnięciem klawisza ENTER pomyśl, czy na pewno usuwasz bazę i wszystkie tabele.

Czas na SQL- strukturalny język zapytań

Zestaw poleceń, którego będziemy używać, by kierować systemem MySQL, to polecenia strukturalnego języka zapytań SQL. Polecenia języka SQL nazywane są zapytaniami (ang. queries)



SQL jest standardowym językiem wykorzystywanym do komunikacji z większością baz danych.

Tworzymy BD

Pamiętamy, że baza danych to zespół tabel połączonych relacjami. Zaczynamy od utworzenia bazy danych, w której umieścimy tabele. Polecenie CREATE DATABASE *nazwa_bazy* tworzy bazę danych. Bazę nazwij wg swojego imienia i np. jeden znak nazwiska.

mysql> CREATE DATABASE henryks;

Baza danych jest gotowa, musimy dodać tabele. mysql>USE henryks;

Tworzenie tabel

Podstawowa składnia polecenia tworzącego tabele:

mysql>CREATE TABLE nazwa_tabeli(

- ->nazwa_kolumny_1 typ_kolumny_1 szczegóły_1,
- ->nazwa_kolumny_1 typ_kolumny_1 szczegóły_1,
- ->...
- ->DEFAULT CHARACTER SET zestaw_znaków;

Tworzymy tabelę z dowcipami więc:

mysql>CREATE TABLE kawal(id INT NOT NULL AUTO_INCREMENT PRIMARY KEY, textkawalu TEXT, datakawalu DATE NOT NULL) DEFAULT CHARACTER SET utf8;

Tworzenie tabel

• Tworzę tabelę kawał:

CREATE TABLE kawal (podaję nazwy pól wraz z typami danych i dodaję na końcu określenie znaków kodowania

DEFAULT CHARACTER SET utf8;

MariaDB [kasiasitko]> CREATE TABLE kawal(

- -> id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
 - -> tekstkawalu TEXT,
 - -> datakawalu DATE NOT NULL)
 - -> DEFAULT CHARACTER SET utf8;

Query OK, 0 rows affected (0.02 sec)

Sprawdzamy

Mogę teraz zobaczyć nowo powstałą tabelę kawal: SHOW TABLES;

DESCRIBE kawal;

Ma	ariaDE	8 [k	asia	sitko))>	SHOW	TABLES	5;
1	Table	es_i	n_ka	siasi	itko	>		
i	kawal					1		
+- 1	row i	in s	et (9.00	sec	:)		

MariaDB [kasiasitko]> DESCRIBE kawal;						
Field	Туре	Null	Key	Default	Extra	
id tekstkawalu datakawalu	int(11) text date	NO YES NO	PRI	NULL NULL NULL	auto_increment	
3 rows in set	(0.00 sec)					

Sprawdzamy

 Tabela kawal zawiera 3 pola, w phpmyadmin wygląda następująco:

🗕 🗊 Serwer: 127.0.0.1 » 🕤 Baza danych: kasiasitko » 📷 Tabela: kawal 🌾 🧮																	
Przegląd	taj 🥻	Struktura	📙 SQL	🔍 Szukaj	≩ ∎ Wsta	aw	🖶 Eksport	-	Import	I Uprav	vnienia	🌽 Opera	icje 💿	Śledze	nie 3	🖇 Wyzwalacze	
M Strukt	tura tabeli	é 6	Relation view														
# Naz	wa 1	Typ Me nap	toda porówny pisów	wania	Atrybuty N	iuli (Ustawienia domyślne		Dodatko	wo	Działanie						
🗌 1 id 🤞) ii	nt(11)			٨	vie l	Brak		AUTO_IN	NCREMENT	🥜 Zmień	ᇢ Usuń	∲ Podsta ▼Więcej	wowy	j Jednoz	znaczny 🐖 Indeks	3
2 teks	stkawalu t	ext			Т	Tak I	NULL				🥜 Zmień	😂 Usuń		wowy	Jednoz	znaczny 🐖 Indeks	3
□ 3 data	akawalu o	date			Ν	vie l	Brak				🥜 Zmień	🔵 Usuń	∲ Podsta ▼Więcej	wowy	j Jednoz	znaczny 🐖 Indeks	3

Wstawiamy dane do tabeli

Polecenie wstawiania danych do tabeli może przyjmować dwie postacie: **INSERT INTO nazwa_tabeli SET** nazwaKolumny1=wartość1 nazwaKolumny2=wartość2; lub INSERT INTO nazwa_tabeli (nazwaKolumny1, nazwaKolumny2,....) VALUES(wartość 1, wartość 2,);

Praktyka

INSERT INTO kawal SET textkawalu=,Czemu bocian stoi na jednej nodze?. Boj jak ją podniesie to się przewróci', datakawalu="2017-09-01";

MariaDB [henryks]> INSERT INTO kawal SET -> textkawalu=,Czemu bocian stoi na jednej nodze?. Boj jak ją podniesie to się przewróci', -> datakawalu="2017-09-01"; Query OK, 1 row affected (0.07 sec)

Samodzielnie dodaj drugi dowcip korzystając z drugiej składni

Praktyka

• Wstawianie danych to tabeli- dwa sposoby:

```
MariaDB [kasiasitko]> INSERT INTO kawal SET
   -> tekstkawalu='kawał mięsa',
   -> datakawalu='2017-07-25';
Query OK, 1 row affected (0.02 sec)
MariaDB [kasiasitko]> INSERT INTO kawal
   -> (tekstkawalu, datakawalu) VALUES('Czemu bocian stoi na jednej nodze? Bo jak ją podniesie, to
się przewróci.','2017-08-15');
```

Query OK, 1 row affected (0.00 sec)

Przeglądanie danych w tabeli

Polecenie SELECT, Którego będziemy używać należy do najbardziej skomplikowanych poleceń w języku SQL. Na razie skorzystamy z najprostszych postaci tego polecenia.

SELECT * FROM kawal;

MariaDB [henryks]> SELECT * FROM kawal;	
id textkawalu	datakawalu
1 kawał mięsa 2 Czemu bocian stoi na jednej nodze?. Boj jak ją podniesie to się przewró	2017-08-31 ci 2017-09-01
2 rows in set (0.00 sec)	

W odpowiedzi dostaliśmy całość danych z tabeli.

Przeglądanie danych w tabeli

Załóżmy, że interesuje nas tylko kolumn ID i data kawału. **SELECT id, datakawalu FROM kawal;** MariaDB [henryks]> SELECT id, datakawalu FROM kawal;

datakawalu

2017-08-31

Ograniczymy teraz znaki w kolumnie tekstkawalu do 15 znaków w celu zwiększenia czytelności wyświetlanych informacji.

SELECT id, LEFT(textkawalu, 15), datakawalu FROM kawal;

NariaDB [henryks]> SELECT id	d, LEFT(textkawalu,15),	datakawalu	FROM	kawal;
id LEFT(textkawalu,15)	++ datakawalu			
1 kawał mięsa 2 Czemu bocian st	2017-08-31 2017-09-01			
2 rows in set (0.00 sec)	++			

Przeglądanie danych w tabeli

Kolejna użyteczna funkcja COUNT zwracająca liczbę rekordów spełniających dane kryteria

SELECT COUNT(*) FROM kawal;

MariaDB	[henryks]>	SELECT	COUNT(*)	FROM	kawal;
+	+				
COUNT ((*)				
+	+				
	2				

Do definiowania bardziej skomplikowanych ograniczeń stosowana jest klauzula **WHERE**

SELECT COUNT(*) FROM kawal WHERE datakawalu=

'2017-09-01';

MariaDB [henryks]> SELECT COUNT(*) FROM kawal WHERE datakawalu=,2017-09-01';
+-----+
| COUNT(*) |
+-----+
| 1 |
+----+
1 row in set (0.00 sec)

Proste kwerendy wybierające

Poszukamy rekordów, które będą zawierały konkretne fragmenty tekstu.

SELECT textkawalu FROM kawal WHERE textkawalu LIKE '%stoi%';

Słowo kluczowe **LIKE** informuje MySQL, że wymieniona w zapytaniu kolumna musi pasować do podanego wzorca. W tym przypadku "%stoi%" określa ,że przed słowem "stoi" może występować tekst, podobnie jak po nim. W klauzuli WHERE możemy jeszcze bardziej ograniczyć rezultaty.

Proste kwerendy wybierające

 SELECT textkawalu FROM kawal WHERE textkawalu LIKE '%stoi%' AND datakawalu>= '2017-06-01' AND datakawalu< '2017-09-15';

MariaDB [henryks]> SELECT textkawalu Fi luk.2017-09-15':	OM kawal WHERE textkawalu	LIKE ,%stoi%' AND	datakawalu>=,2017-06-01'	AND datakawa
+		++		
textkawalu +		 ++		
Czemu bocian stoi na jednej nodze?. H	oj jak ją podniesie to si	ę przewróci		

Modyfikowanie danych w tabeli

W celu modyfikowania/zaktualizowania danych w tabeli stosujemy polecenie UPDATE.

UPDATE kawal SET datakawalu=,2017-09-02' WHERE id=,2';

Zmieniamy datę kawału na 02-09-2017 w rekordzie o id=2.

MariaDB [smieszna_baza]> UPDATE kawal SET datakawalu='2017-09-02' WHERE id='2'; Query OK, 1 row affected (0.02 sec) Rows matched: 1 Changed: 1 Warnings: 0

Sprawdzę, za pomocą polecenia SELECT, czy data została zmieniona: SELECT * FROM kawal;

ł	id	+ tekstkawalu	datakawalu
	1	kawał mięsa	2017-07-25
	2	Czemu bocian stoi na jednej nodze? Bo jak ją podniesie, to się przewróci	2017-09-02

Modyfikowanie danych w tabeli

Można również seryjnie zmieniać dane dla rekordów spełniających warunki np.

UPDATE kawal SET datakawalu=,2017-09-03' WHERE textkawalu LIKE ,%stoi%';

MariaDB [henryks]> UPDATE kawal SET datakawalu=,2017-09-03' WHERE textkawalu LIKE ,%stoi%'; Query OK, 1 row affected (0.01 sec) Rows matched: 1 Changed: 1 Warnings: 0

Data ponownie została zaktualizowana:

SELECT id, datakawalu FROM kawal;



Ćwiczenie

- Dodaj do bazy 3 kolejne dowcipy, zastosuj wcześniej poznane metody.
- Dlaczego blondynkom nie przysługuje przerwa w pracy? Bo to nieopłacalne- trzeba je ponownie przyuczyć do zawodu.
- 2. Szkoła jest jak Media Markt-powinni tego zabronić.
- 3. Nauczyciel na chemii omawia doświadczenie: teraz patrzcie, jak ten niebieski płyn zazieleni się na czerwono.
- 4. Wyświetl zawartość tabeli kawal.

Ćwiczenie

 Efektem wykonania zadania jest tabela w bazie składająca się z 5 rekordów:

id tekstkawalu	datakawalu
1 kawał mięsa 2 Czemu bocian stoi na jednej nodze? Bo jak ją podniesie, to się przewróci. 3 Dlaczego blondynkom nie przysługuje przerwa w pracy? Bo to nieopłacalne, trzeba je ponownie przyuczyć do zawo 4 Szkoła jest jak MediaMarkt- powinni tego zabronić 5 Nauczyciel na chemii omawia doświadczenie: teraz patrzcie, jak ten niebieski płyn zazieleni się na czerwono	2017-07-25 2017-08-15 2017-08-28 2017-08-28 2017-08-28

 Instrukcja ALTER TABLE modyfikuje strukturę tabelnp. można dodać/usunąć pole w tabeli.
 ALTER TABLE nazwa_tabeli ADD/DROP nazwa_nowego_pola typ_danych;

Do tabeli kawal dodamy nowe pole "rodzaj":

MariaDB [kasiasitko]> ALTER TABLE kawal ADD rodzaj VARCHAR(30); Query OK, 0 rows affected (0.05 sec) Records: 0 Duplicates: 0 Warnings: 0

Nowe pole zostało dodane do tabeli:



Użyj polecenia DROP, aby usunąć pole "rodzaj"

 Ponownie dodamy nowe pole "rodzaj", ale określimy jego miejsce- ma być pomiędzy kolumną "id" oraz "tekstkawalu", czyli PO kolumnie "id":

ALTER TABLE kawal ADD rodzaj VARCHAR(30) AFTER id;

NariaDB [kasiasitko]> SELECT id, rodzaj, tekstkawalu FROM kawal;							
id	rodzaj	tekstkawalu					
1 2 3 4 5	NULL NULL NULL NULL NULL	kawał mięsa Czemu bocian stoi na jednej nodze? Bo jak ją podni Dlaczego blondynkom nie przysługuje przerwa w prac Szkoła jest jak MediaMarkt- powinni tego zabronić Nauczyciel na chemii omawia doświadczenie: teraz p					
5 rows	in set ((0.00 sec)					

- Nowemu polu "rodzaj" przypisaliśmy typ danych VARCHAR. Aby zmienić typ danych w polu należy zastosować polecenie MODIFY:
- ALTER TABLE nazwa_tabeli MODIFY nazwa_pola typ danych; ALTER TABLE kawal MODIFY rodzaj INT;

Aby sprawdzić, czy typ danych w polu rodzaj został zmieniony należy zapisać: DESCRIBE kawal;

MariaDB [kasiasitko]> DESCRIBE kawal;							
Field	Туре	Null	Key	Default	Extra		
id rodzaj tekstkawalu datakawalu	int(11) int(11) text date	NO YES YES NO	PRI	NULL NULL NULL NULL	auto_increment		
4 rows in set	(0.02 sec)		*****				

- Modyfikując strukturę tabeli można również zmienić nazwę danego pola tabeli przy użyciu polecenia CHANGE:
- ALTER TABLE nazwa_tabeli CHANGE bieżąca_nazwa nowa_nazwa_pola typ_danych;

Zmienimy nazwę pola "rodzaj" na "kategoria_kawalu": ALTER TABLE kawal CHANGE rodzaj kategoria_kawalu INT(25);

MariaDB [kasiasitko]> SELECT id, kategoria_kawalu,LEFT(tekstkawalu,12) FROM kawal;							
id	kategoria_kawalu	LEFT(tekstkawalu,12)					
1 2 3 4 5	NULL NULL NULL NULL NULL	kawał mięsa Czemu bocian Dlaczego blo Szkoła jest Nauczyciel n					
t rows	in set (0.00 sec)	++					

 Pole kategoria_kawalu stworzyliśmy tylko na potrzeby ćwiczenia. Nie będzie nam potrzebne, więc usuń go korzystając z właściwego polecenia.

Usuwanie danych w tabeli

Nie ma nic prostszego i niebezpiecznego. PAMIĘTAJ! My SQL nie zapyta, czy jesteś pewien, po prostu po naciśnięciu klawisza ENTER wykona posłusznie polecenie.

DELETE FROM kawal WHERE textkawalu=,%stoi%'; Usuwa wszystkie rekordy ze słowem "stoi".

DELETE FROM kawal;

Usuwa całą tabelę kawał.

Czas na ćwiczenia

Wykorzystując polecenie tworzenia tabeli i wstawiania danych do tabeli wstaw dane zawarte w pliku ćwiczeniowym lektury.txt Powinieneś uzyskać tabelę z 282 rekordami Plik do importu znajdziesz na:

http://www.nasza-szkola.pl/pomoce/informatyka/zspm/lektury.txt

Wklej plik do konsoli dodaj ; (średnik)

Klauzula where

W klauzuli **WHERE** formułuje się warunek, który odpowiada warunkowi wyboru (selekcji) w algebrze relacyjnej i który określa ograniczenia, jakie mają spełniać rekordy, aby zostać wybrane w danym zapytaniu.

Jeżeli rekord spełnia te ograniczenia to zostaje dołączony do tabeli wynikowej.

Postać zapytania:

SELECT * FROM nazwa-tabeli WHERE warunek; Klauzula WHERE pozwala na wybranie z tabeli tych wierszy, które spełniają określone warunki:

SELECT * FROM NAZWISKA WHERE STANOWISKO = 'INFORMATYK'; Dla podanego przykładu z tabeli zostaną wybrane tylko te rekordy, w których w polu STANOWISKO jest wpisane 'INFORMATYK'

Formułowanie warunku

Po słowie kluczowym **WHERE** występuje wyrażenie warunkowe. Do zapisu porównywania wartości w języku SQL służy sześć operatorów: równy = nierówny <> mniejszy < większy > mniejszy lub równy <= większy lub równy >= W wyrażeniu mogą występować stałe oraz nazwy kolumn tabel wymienionych w klauzuli **FROM** Dla wartości numerycznych można budować wyrażenia arytmetyczne korzystając z operatorów + - * / i nawiasów () Stałe tekstowe w SQL są ujmowane w pojedyncze cudzysłowy 'Przykładowe stałe' W wyniku porównania powstaje wartość logiczna TRUE (prawda) lub FALSE (fałsz) Wartości logiczne można łączyć w wyrażenia logiczne za pomocą operatorów logicznych AND, OR i NOT Priorytet operatorów wykorzystywanych w budowie wyrażeń : operatory porównania, NOT, AND, OR


SELECT Imię, Nazwisko, Stanowisko, Pensja FROM NAZWISKA WHERE (Stanowisko ='Informatyk' OR Stanowisko = 'Dyrektor') AND Pensja >= 4900;

WHERE z operatorem LIKE

Działa na kolumnach zawierających wartości łańcuchowe.

Operator LIKE sprawdza czy wartość tekstowa odpowiada podanemu wzorcowi, umożliwia więc wykonywanie częściowych porównań, takich jak "zaczynający się od tekstu", "kończący się na tekście", lub "zawierający tekst"

Tworząc wzorce stosuje się znaki wieloznaczne: **% - zastępuje** sekwencję dowolnych znaków o długości n (gdzie n może być zerem)

<u>- odpowiada jednemu</u> znakowi w przeszukiwanym tekście Ogólna postać polecenia z operatorem LIKE:

WHERE tekst LIKE wzorzec np...

SELECT * FROM lektury1 WHERE AUTOR LIKE 'P%';

Zwróci wszystkich autorów których nazwiska zaczynają się na literę P.

Przykłady

 Załóżmy, że chcesz wyszukać pracowników w tabeli pracownicy, których imię rozpoczyna się znakiem 'a', możemy to zrobić w następujący sposób:

SELECT numer_pracownika, nazwisko, imie FROM pracownicy WHERE imie LIKE 'A%'

 Aby wyszukać wszystkich pracowników, których nazwiska kończą się na 'na' można wykonać zapytanie w następujący sposób:

SELECT numer_pracownika, nazwisko, imie FROM pracownicy WHERE imie LIKE '%on'

Przykłady

 Jeśli chcesz odszukać dany ciąg znaków, który znajduję się w środku słowa, możesz umieścić znak procentu na początku i na końcu, aby znaleźć poszukiwany ciąg znaków.

SELECT numer_pracownika, nazwisko, imie FROM pracownicy WHERE imie LIKE '%on%'

 Aby wyszukać wszystkich pracowników, których nazwiska są jak Byk, Bok. Możesz użyć znaku dolnego podkreślenia (_):

SELECT numer_pracownika, nazwisko, imie FROM pracownicy WHERE nazwisko LIKE 'B_k'

Przykłady

 MySQL wraz z operatorem LIKE umożliwia stosowanie przeczenia NOT, aby zaleźć wszystkie ciągi znaków, które nie są dopasowane do określonego wzorca. Załóżmy, że chcesz wyszukać wszystkich pracowników, których nazwisko nie zaczyna się od znaku 'B', możesz użyć następującego zapytania:

SELECT numer_pracownika, nazwisko, imie FROM pracownicy WHERE nazwisko NOT LIKE 'B%,

<u>Operator LIKE nie rozróżnia wielkości liter, tak więc 'b%'</u> <u>i 'B%' są takie same.</u>

ĆWICZENIA Z KWEREND

Pokaż książki autora którego nazwisko zaczyna się na literę **B** Odpowiedź: 29 pozycji. Pokaż książki droższe niż 5 zł, ale tańsze niż 10 zł. Odpowiedź: 79 pozycji.

Pokaż książki WYDANIE 1 lub 3.

Odpowiedź: 138 pozycji.

Pokaż książki których **TYTUŁ** składa się z 5 liter.

Odpowiedź: np. Diuna

Pokaż książki których TYTUŁ kończy się pojedynczą literą.

Odpowiedź: np. Historia Polski XX w

 Pokaż książki WYDANIE 2 lub 3, po 1990 roku i TYTUŁ zawierający pojedynczą literę i.

SPRAWDZIAN(KARTKÓWKA)

- 1.Pokaż książki, których ceny zawarte są pomiędzy 5 oraz 15 zł (wraz z tymi kwotami) i nie są równe 10 zł.
- 2.Pokaż książki których tytuł zaczyna się na literę M lub N autorów których nazwisko zaczyna się na literę B
- 3.Pokaż książki WYDANIE 1 lub 3, wydane przed 1980 roku i TYTUŁ składający się z 6 liter.
- 4.Pokaż książki WYDANIE 2 lub 3, przed rokiem 1980 i TYTUŁ zawierający pojedynczą literę z.
- 5.Podaj ilości książek (tytułów) wydanych w Warszawie i Poznaniu

Szablony w PHP-?

Na poprzednich zajęciach, poznaliśmy język PHP. Każdy program pisaliśmy w oddzielnym pliku, z indywidualnym formatowaniem.

Przy dużych projektach jest to prawie niewykonalne, dlatego stosujemy szablony, czyli odseparujemy kod PHP tak, by znajdował się w oddzielnym pliku, a kod HTML pozostał wolny od wszelkich elementów języka PHP.

Do tego celu służy instrukcja *include*. Za jej pomocą do kodu PHP można wstawić zawartość innego pliku dokładnie w tym miejscu, w którym instrukcja *include* się znajduje.

Przykład szablonu

Wcześniej poznałeś zasady pracy z poziomu konsoli z bazą danych MySQL. Połączenie i korzystanie z tabel wymaga :

- Zalogowania do MySQL
- Wybór kodowania znaków
- Połączenie z bazą

Każdą operację konsola kontrolowała i dostałeś odpowiednie komunikaty. W PHP musisz to sam oprogramować i zabezpieczyć przed ewentualnymi błędami (np.: Błąd logowania, Błąd kodowania, Błąd, Baza nie istnieje, itp.).

Przykład szablonu ,output.html.php'

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Struct//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-struct.dtd"> <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">

<head>

<title>Dane wynikowe PHP</title>

<meta http-equiv="content-type" content="text/html: charset=utf-8"/>

</head><body>

<?php echo \$output; ?> (tutaj postawimy odpowiedni komunikat)

</body></html>

PHP i dane w bazie MySQL Połączenie z bazą.

Zanim wyświetlimy dane z tabel bazy danych musimy nawiązać połączenie:

mysqli_conect(nazwa_komputera, użytkownik, hasło) Funkcja zwraca tzw. Identyfikator połączenia, który pozwala odróżnić ustanowione połączenie.

```
$link=mysqli_connect('localhost','root');
if (!$link)
```

```
{

$output='Nie można połączyć się z serwerem bazy

danych';

include 'output.html.php'; ← wywołanie szablonu(s39)

exit();}
```

Zestaw znaków

Jeżeli uda się nawiązać połączenie z bazą danych, warto sprawdzić system kodowania. Najlepiej korzystać z UTF-8.

mysqli_set_charset(\$link,'utf-8');

Praktycznie:

if(!mysqli_set_charset(\$link,'utf-8'))

```
$output='Nie można ustanowić kodowania dla
połączeń z bazą danych';
```

}

Łączymy się z BD

Jeżeli poprzednie dwa punkty powiodły się, łączymy się z BD. **mysqli_select_db(\$link, ,nazwa BD');**

```
Praktycznie:
if(!mysqli_select_db($link,'henryks'))
```

\$output='Nie znaleziono bazy danych henryks';
include 'output.html.php';
exit();

Pełny kod

```
<?php
$link=mysqli connect('localhost','root');
if (!$link)
$output='Nie można płączyć się z serwerem bazy danych';
include 'output.html.php';
exit();}
if(!mysqli_set_charset($link,'utf-8'))
{
         $output='Nie można ustanowić kodowania dla połączeń z bazą danych';
         include 'output.html.php';
         exit();}
if(!mysqli select_db($link, 'henryks'))
{
                                                                    Dane wynikowe PHP
                                                                 ខា
         Soutput='Nie znaleziono bazy danych henrys';
         include 'output.html.php';
                                                                      localhost/kawaly/
         exit();}
$output='Nawiązano połączenie z bazą danych';
                                                               Nawiązano połączenie z bazą danych
include 'output.html.php';
```

?>

Zapytania SQL w PHP

W języku PHP istnieje podobny mechanizm do tego, który stosowaliśmy z poziomu konsoli.

mysqli_query(identyfikator połączenia, zapytanie)

gdzie zapytanie będzie tekstem zawierającym polecenie SQL.

Funkcja **mysqli_query** zwraca prawdę jeżeli udało się wykonać zapytanie lub fałsz jeżeli wystąpiły błędy.

Tworzymy tabelę za pomocą PHP

 \times

```
$sql='CREATE TABLE kawal1(
  INT NOT NULL AUTO INCREMENT PRIMARY KEY,
  tekstkawalu TEXT,
                                                    Dane wynikowe PHP
  datakawalu DATE NOT NULL)
                                                    (i) localhost/kawaly/
DEFAULT CHARACTER SET utf8';
                                                 Nawiązano połączenie z bazą danych
if (!mysqli_query($link,$sql))
                                                 Udało się utworzyć tabelę kawał 1.
```

\$output='Nie udało się utworzyć tabeli'.mysqli_error(\$link); include'output.html.php'; exit();

\$output='Udało się utworzyć tabelę kawał 1.'; include'output.html.php'

Tworzymy tabelę za pomocą PHP

W przypadku wystąpienia błędu funkcja mysqli_error(\$link);

wyświetli szczegółowy opis blędu np. jeżeli tabela już istnieje

\$output='Nie udało się utworzyć tabeli'.mysqli_error(\$link); include'output.html.php'; exit();

) (i) localhost/kawaly/

Nawiązano połączenie z bazą danych

Nie udało się utworzyć tabeliTable 'kawal1' already exists

Modyfikowanie danych w tabeli

W przypadku zapytań DELETE, INSERT i UPDATE system MySQL notuje również liczbę wierszy tabeli (pozycji), które zostały zmienione przez zapytanie. Spróbujemy zmienić datę kawałów z tekstem zawierającym słowo "bocian".

\$sql='UPDATE kawal SET datakawalu="2017-09-10" WHERE textkawalu LIKE "%bocian%"";

```
if(!mysqli_query($link,$sql))
```

\$output='Błąd w trakcie wykonywania uaktualnienia:'.mysqli_error(\$link);
include 'output.html.php';
exit();

```
}
$output='Uaktualniono'.mysqli_affected_rows($link).'wierszy';
include'output.html.php';
```

Wyświetlanie wyników zapytania SELECT

Funkcja **mysqli_query** zwraca wartość false w razie niepowodzenia lub true gdy pojawią się wyniki zapytania. W połączeniu z poleceniem SELECT funkcja zwraca zbiór wyników w postaci rekordów spełniających warunki. Fragment kodu **\$result=mysqli_query(\$link, 'SELECT textkawalu FROM kawal');**

if (!\$result)

{

```
$error='Błąd odczytu kawałów'.mysqli_error($link);
include 'error.html.php';
exit();
```

Szablon obsługi błędów error.html.php

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Struct//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-struct.dtd"> <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl"> <head>

<title>Błąd PHP</title>

<meta http-equiv="content-type" content="text/html:charset=utf-8"/> </head>

```
<body>
<?php echo $error; ?>
</body>
</html>
```

W przypadku wystąpienia błędu wyświetlony będzie opis błędu

Wyświetlanie wyników zapytania SELECT

Przy założeniu, że w trakcie przetwarzania nie pojawi się żaden błąd to zmienna **\$result** zawierać będzie zbiór wyników, czyli w praktyce to może być duża baza kawałów.

Wykorzystamy pętlę **while** w celu uporządkowania danych w zmiennej **\$result**.

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

//zapełniamy tablicę; \$jokes[]=\$row['textkawalu'];

Wykonuj dopóki istnieją wiersze w \$result

Szablon wyświetlania kawałów jokes.html.php

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Struct//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-struct.dtd"> <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl"> <head>

<title>Dane wynikowe PHP</title>

<meta http-equiv="content-type" content="text/html:charset=utf-8"/></head>

<body>

Vszystkie kawały przechowywane w bazie danych :

<blockquote>

<?php //echo htmlspecialchars(\$joke,ENT_QUOTES,'UTF-8'); ?>

//Lub po prostu <?php echo \$joke;?> </blockquote>

<?php endforeach; ?>

</body>

</html>

Kompletny Kod wyświetlania

```
<?php
$link=mysqli_connect('localhost','root');
if (!$link)
{
$cutout='Nic moting placetuá sic z
```

```
$output='Nie można płączyć się z
serwerem bazy danych';
include 'output.html.php';
exit();
```

```
if(!mysqli_select_db($link,'henryks'))
{
```

```
$output='Nie znaleziono bazy
danych henrys';
```

```
include 'output.html.php';
exit();}
$result=mysqli_query($link, 'SELECT
textkawalu FROM kawal');
```

```
if (!$result)
```

```
$error='Błąd odczytu
kawałów'.mysqli_error($link);
include 'error.html.php';
exit();
```

```
while($row=mysqli_fetch_array($result))
{
     $jokes[]=$row['textkawalu'];
}
include'jokes.html.php';
?>
```

Wstawianie danych do bazy

Opracujemy teraz formularz do wprowadzania danych do tabeli. (form.html.php)

<head>

<title>Dodawanie kawału</title>

<meta http-equiv="content-type" content="text/html:charset=utf-8"/> <style type="text/css">

<textarea{display:block; width:100% }> </style>

</head>

<body>

```
<form action="?" method="post">
```

<div> <label for="joketext">Wpisz treść kawału:</label> <textarea id="tekstkawalu" name="tekstkawalu" rows="3" cols="40"></textarea>

</div>

<div><input type="submit" value="dodaj"/></div>

</form>

</body> </html>

Wstawianie danych do bazy

Efekt działania:

Dodawanie kawału	× +	
Iocalhost/kawaly/form.html.php?		
W7 1 4 771 1		
dodaj		.::

Atrybut action="?,, posłuży nam do wywołania odpowiedniego zapytania z adresu URL W tym celu musimy zmodyfikować plik ,,**jokes.html.php**'' (slajd 51) dopisując: Dodaj kawał Wszystkie kawały przechowywane w bazie danych :

Wstawianie danych do bazy

Teraz w kodzie pliku index.php(tzw. kontroler) możemy sprawdzić czy zmienna **addjoke** istnieje. *if(isset(\$_GET['addjoke']))*

include 'form.html.php';
exit();

} Po wypełnieniu formularza i zatwierdzeniu przyciskiem DODAJ do kontrolera zostanie wysłane kolejne żądanie, które zostanie wykryte przez element \$_POST[,joketext']. Wstawianie danych do bazy – to nie może być takie proste

Zatwierdzony tekst kawału wstawiamy poprzez wykonanie zapytania INSERT. Kod może wyglądać następująco: if(isset(\$_POST['joketext']))

\$sql='INSERT INTO kawal SET textkawalu="'.\$joketext."',
 datakawalu=,,dzisiejsza data"';
 if(!mysqli_query(\$link,\$sql)) {
 \$error='Błąd w trakcie dodawania'.mysqli_error(\$link);
 include 'error.html.php';
 exit();

} Taki kod ma jednak bardzo istotną wadę. Użytkownik zamiast nowego kawału może napisać odpowiednie zapytanie języka SQL, które zostanie wykonane na serwerze bazy danych MySQL. Taki atak zwany przez "wstrzyknięcie kodu SQL" (ang. SQL injection attack)

Wstawianie danych do bazy-zabezpieczenia

\$sql='INSERT INTO kawal SET
textkawalu="".\$joketext."',
datakawalu=CURDATE()';

Tak skonstruowany kod posiada bardzo istotną wadę, wartość zmiennej \$_POST[`textkawalu'] nie musi zawierać treści kawału, a zawierać może złośliwe zapytanie SQL wykonane przez serwer.

Od wersji PHP 5 język ma kilka wbudowanych mechanizmów obrony zwanych **magicznymi apostrofami**. Mechanizm ten ma poważną wadę, tak jak chroni tak i przeszkadza.

<u>Przetestuj działanie skryptu z wcześniejszego slajdu. Zwróć uwagę, że najprawdopodobniej dodany będzie nowy rekord, ale bez treści.</u>

```
Zabezpieczenie-nie wiem jak ale działa.
if(get_magic_quotes_gpc())
function stripslashes_deep($value)
     $value=is_array($value) ?
             array_map('stripslashes_deep'. $value);
            stripslashes($value);
            return $value;
$_POST=array_map('stripslashes_deep'.$_POST);
$_GET=array_map('stripslashes_deep'.$_GET);
$_COOKIE=array_map('stripslashes_deep'.$_COOKIE);
$REQUEST=array_map('stripslashes_deep'.$_REQUEST);
```

Funkcja likwiduje ewentualne poczynione szkody przez wbudowane zabezpieczenia PHP. Przygotowanie tekstu SQL

PHP udostępnia funkcję przygotowującą dane od użytkownika w taki sposób by bezpiecznie można było je umieścić w zapytaniu SQL

\$joketext=mysqli_real_escape_string(\$link, \$_POST['joketext']);
\$sql='INSERT INTO kawal SET
textkawalu='''.\$joketext''',
datakawalu=CURDATE()';

Przekierowanie HTTP

W naszym przykładzie po dodaniu kawału, powinniśmy wyświetlić od razu dodany kawał. Wykorzystamy funkcję **header**. header(`Location: adres URL');

Nasza strona ma zostać jeszcze raz wczytana (odświeżona). Katalog bieżący wskazuje się za pomocą kropki (.).

Przekierowanie przeglądarki z powrotem do kontrolera ma następującą treść.

header(`Location:.;')
exit();

Gotowy kod

<?php

if(get magic quotes gpc())

{ function stripslashes deep(\$value)

```
{
          $value=is_array($value)?
                       array map('stripslashes deep'. $value):
                       stripslashes($value);
                       return Svalue; }
$ POST=array map('stripslashes deep'.$ POST);
$_GET=array_map('stripslashes_deep'.$_GET);
$_COOKIE=array_map('stripslashes_deep'.$_COOKIE);
SREQUEST=array map('stripslashes deep'.S REQUEST);
}
if(isset($_GET['addjoke']))
          include 'form.html.php';
{
          exit();
                   }
$link=mysqli_connect('localhost','root');
if (!$link)
    Soutput='Nie można płączyć się z serwerem bazy danych';
include 'output.html.php';
exit();
          }
if(!mysgli select db($link,'henryks'))
{
          $output='Nie znaleziono bazy danych henrys';
          include 'error.html.php';
          exit();
                     }
```

Dodaj kawał

Wszystkie kawały przechowywane w bazie danych :

kawa? mi?sa

Czemu bocian stoi na jednej nodze?. Bo jak j? podniesie to si? przewr�ci

Szkoła jest jak Media Markt- powinni tego zabronić.

Nauczyciel na chemii omawia doświadczenie: teraz patrzcie, jak ten niebieski płyn :

Dlaczego blondynkom nie przysługuje przerwa w pracy? Bo to nieopłacalne- trzeba

```
Gotowy kod cd.
```

```
if(isset($_POST['joketext']))
{
           $joketext=mysqli_real_escape_string($link, $_POST['joketext']);
           $sql='INSERT INTO kawal SET
           textkawalu="".$joketext."",
           datakawalu=CURDATE()';
           if(!mysqli_query($link,$sql))
                        $error='Blad w trakcie dodawania Kawalu'.mysqli_error($link);
                       include 'error.html.php';
                        exit();
           }
           header('Location:.');
           exit(): }
$result=mysqli_query($link, 'SELECT textkawalu FROM kawal');
if (!$result)
{
           $error='Blad odczytu kawałów'.mysqli error($link);
           include 'error.html.php';
           exit(): }
while($row=mysqli_fetch_array($result))
           $jokes[]=$row['textkawalu'];
{
                                            }
include'jokes.html.php';
```

?>

http://www.nasza-szkola.pl/pomoce/informatyka/zspm/php/jokes_php1

Usuwanie danych z bazy

Obok każdego dowcipu dodamy przycisk o nazwie **Usuń**, którego kliknięcie spowoduje usunięcie kawału z bazy danych i wyświetlenie uaktualnionej listy dowcipów.

Zrobimy to w tym samym pliku, który wyświetla kawały i ma możliwość dodania.

Będzie nam potrzebna instrukcja DELETE języka SQL, która wymaga jednoznacznego określenia usuwanego rekordu czyli ID.

Zmodyfikujemy kilka szczegółów.

Zmieniamy

Modyfikujemy polecenie SELECT tak by pobierać oprócz tekstu kawału również **id**.

\$result=mysqli_query(\$link, 'SELECT id,textkawalu FROM kawal');
if (!\$result)

\$error='Błąd odczytu kawałów'.mysqli_error(\$link); include 'error.html.php'; exit();

Musimy również zmodyfikować pętlę while while(\$row=mysqli_fetch_array(\$result))

}

\$jokes[]=array('id'=>\$row['id'], 'text'=>\$row['textkawalu']);

Zmieniamy jokes.html.php

```
<?php foreach ($jokes as $joke): ?>
     <form action="?deletejoke" method="post"> //1
     <blockguote>
     <?php echo
htmlspecialchars($joke['text'],ENT_QUOTES,'UTF-8'); ?>//2
     <input type="hidden" name="id" value=
                 "<?php echo $joke['id'];?>"/>
                                                //3
     <input type="submit" value="Usuń"/> //4
     </blockquote>
     </form>
     <?php endforeach; ?>
```
Objaśnienia do zmian

- //1 w formularzu wyświetlone są wszystkie kawały.
 Sygnałem do usunięcia będzie ?deletejoke w atrybucie action
- //2 Obecnie każdy kawał jest reprezentowany w tablicy dwuelementowej (id, textkawlu)
- //3 Do usunięcia konieczny jest id, ale w formularzu nie jest potrzebny, dlatego użyłem
 <input type="hidden" name="id" value="<?php echo \$joke['id'];?>"/>
- //4 Zatwierdzenie formularza i wysłanie żądania usunięcia.

Efekt widoczny

Dodaj kawał Wszystkie kawały przechowywane w bazie danych : kawa? mi?sa Usuń Usuń Szkoła jest jak Media Markt- powinni tego zabronić. Usuń Nauczyciel na chemii omawia doświadczenie: teraz patrzcie, jak ten niebieski płyn zazieleni się na czerwono. Usuń Dlaczego blondynkom nie przysługuje przerwa w pracy? Bo to nieopłacalne- trzeba je ponownie przyuczyć do zawodu. Usuń

Może nadszedł czas by trochę "upiększyć" naszą stronę. Mam na myśli CSS.

Ostatni krok to oprogramowanie przycisku **usuń**

Usuwanie rekordów

```
if(isset($_GET['deletejoke']))
```

```
$id=mysqli_real_escape_string($link,$_POST['id']);
$sql="DELETE FROM kawal WHERE id='$id";
if(!mysqli_query($link,$sql))
```

```
$error='Błąd w trakcie usuwania kawału'.mysqli_error($link);
include 'error.html.php';
```

```
exit();
```

```
header('Location:.');
```

```
exit();
```

Relacyjne bazy danych

Rozbudujemy naszą bazę kawałów a nazwisko i imię autora dodającego dowcip do bazy. Zgodnie z zasadą:

"Każdemu rodzajowi informacji(lub rzeczy albo encji), które chcemy przechowywać w bazie danych, powinniśmy przydzielić własną tabelę."

Dodamy do tabeli kawał pole **idautora** i utworzymy nową tabelę **autor** gdzie znajdą się **pola id, nazwa, email.**

Tabela autorów

Wszystko oczywiście robimy z poziomu konsoli.

Tworzymy tabelę autor

CREATE TABLE autor (id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,nazwa VARCHAR(255),email VARCHAR(255)) DEFAULT CHARACTER SET uff8;

Dodajemy pole idautora do tabeli kawal

ALTER TABLE kawal ADD COLUMN idautora INT;

Dodajemy autorów

INSERT INTO autor SET id=1,nazwa=,lwona Kowalska',email='henryk@mail.com'; INSERT INTO autor SET id=2,nazwa=,lmię Nazwisko',email=`imie@mail.com';

Dodajmy jeszcze numery autorów do tabeli kawal. Możesz to zrobić z poziomu phpmyadmin

+ Opcje								
ר≁	→		\bigtriangledown	id	textkawalu	datakawalu	idautora	
	🥜 Edytuj	👫 Kopiuj	🤤 Usuń	1	kawał mięsa	2017-08-31	1	
	🥜 Edytuj	📑 Kopiuj	🤤 Usuń	8	SzkoÅ,a jest jak Media Markt- powinni tego zabroni	2017-10-08	2	
	🥜 Edytuj	👫 Kopiuj	🥥 Usuń	9	Nauczyciel na chemii omawia doÂ>wiadczenie: teraz	2017-10-08	2	
	🥜 Edytuj	📑 Kopiuj	🤤 Usuń	10	Dlaczego blondynkom nie przysÂ,uguje przerwa w pr	2017-10-08	1	
	🥜 Edytuj	📲 Kopiuj	🤤 Usuń	12	Wpada facet na piwo i mówi, Auć.	2017-10-08	1	

Zakładamy relacje

Skorzystamy z nowej formy instrukcji **SELECT**, zwanej złączeniem (ang. join)

<u>SELECT kolumny FROM tabela1 INNER JOIN tabela2</u> <u>ON warunek(ki) definiujące relację między</u> <u>wybieranymi danymi</u>.

SELECT kawal.id, LEFT(textkawalu,20),nazwa,email FROM kawal INNER JOIN autor WHERE idautora=autor.id;

MariaDB [henryks]> SELECT kawal.id, LEFT(textkawalu,20),nazwa,email FROM kaw								
+ id	LEFT(textkawalu,20)	+ nazwa +	+ email +	+ +				
1 8 9 10 12	kawał mięsa Szko??a jest jak Med Nauczyciel na chemii Dlaczego blondynkom Wpada facet na piwo	Iwona Kowalska Henryk Szantula Henryk Szantula Iwona Kowalska Iwona Kowalska	iwona@mail.com henryk@mail.com henryk@mail.com iwona@mail.com iwona@mail.com					
5 rows	s in set (0.00 sec)							

Kilka zmian w index.php

//\$result=mysqli_query(\$link, 'SELECT id,textkawalu FROM kawal'); bez autorów

\$result=mysqli_query(\$link, 'SELECT
kawal.id,textkawalu,nazwa,email FROM kawal INNER JOIN autor
ON idautora=autor.id;');

if (!\$result)

{ \$error='Błąd odczytu kawałów'.mysqli_error(\$link); include 'error.html.php'; exit(); }

```
while($row=mysqli_fetch_array($result))
{
//$jokes[]=array('id'=>$row['id'],'text'=>$row['textkawalu']); bez
autorów
```

\$jokes[]=array('id'=>\$row['id'],'text'=>\$row['textkawalu'],
'name'=>\$row['nazwa'],'email'=>\$row['email']);

```
include'jokes.html.php';
```

Wyświetlamy autorów

fragment jokes.html.php <?php echo htmlspecialchars(\$joke['text'],ENT_QUOTES,'UTF-8'); ?> <input type="hidden" name="id" value="<?php echo \$joke['id'];?>"/> <input type="submit" value="Usuń"/> (autor<a href="mailto:<?php echo htmlspecialchars(\$joke['email'],ENT_QUOTES, 'UTF-8');?>"> <?php echo htmlspecialchars(\$joke['email'],ENT_QUOTES,'UTF-8');?>)

</n>

NA KONIEC

System oparty na SQL i PHP działa.

$\leftrightarrow \rightarrow \mathbb{C}$ () localhost/kawaly/						
📙 SSL 🚺 Galeria obiektów Wel: 🎦 Sugerowane witryny 💪 Google 🦲 LIBRUS - e-Dziennik 🕥 LIBRUS Synergia 🍾 Jest już darmowy Offi 💪 Google 🛝 INTERIA.PL - Polska						
<u>Dodaj kawał</u>						
Wszystkie kawały przechowywane w bazie danych :						
Szkoła jest jak Media Markt- powinni tego zabronić. Usuń (autor <u>henryk@mail.com</u>)						
Nauczyciel na chemii omawia doświadczenie: teraz patrzcie, jak ten niebieski płyn zazieleni się na czerwono. Usuń (autor henryk@mail.com)						
Dlaczego blondynkom nie przysługuje przerwa 🗆 w pracy? Bo to nieopłacalne- trzeba je ponownie przyuczyć do zawodu. Usuń (autor <u>iwona@mail.com</u>)						
Wpada facet na piwo i mówi, Auć. Usuń (autor <u>iwona@mail.com</u>)						

Prawdopodobnie pojawił się problem z dodawaniem kawałów. Zapytanie skierowane do kolumny bez jednoznacznego ID generuje błąd i dodany rekord nie jest wyświetlany. **Możesz to potraktować jako** wyzwanie do zmodyfikowania dodawania kawałów.

Koniec projektu

Poznaliśmy możliwości współpracy php z serwerem MySQL i zastosowaliśmy język SQL do tworzenia, modyfikowania i usuwania rekordów z bazy danych. Pamiętamy jednak, że baza danych wg definicji jest "zespołem tabel połączonych relacjami" Na następnych slajdach wzbogacimy nasz projekt o możliwość podpisywania dodanych kawałów przez autorów zapisanych w oddzielnej tabeli.