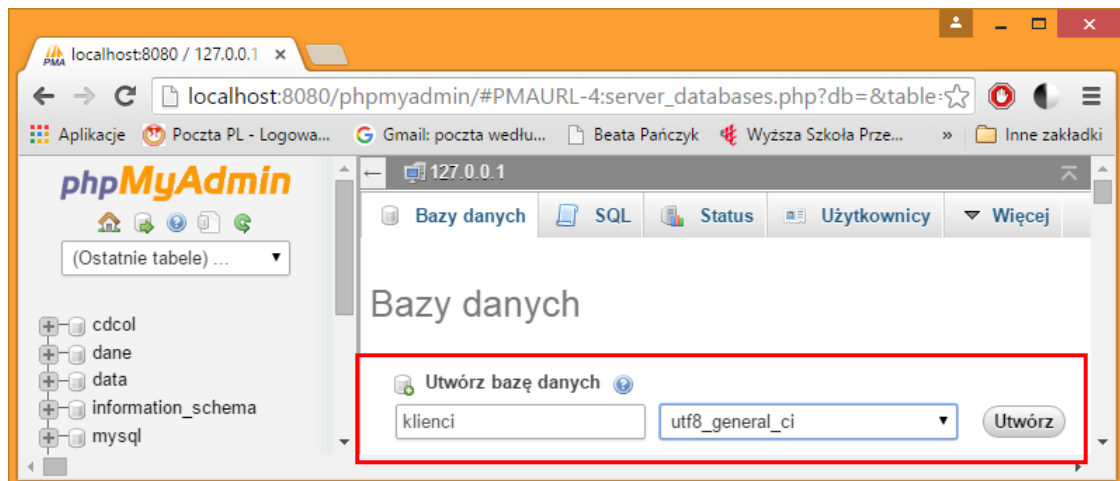
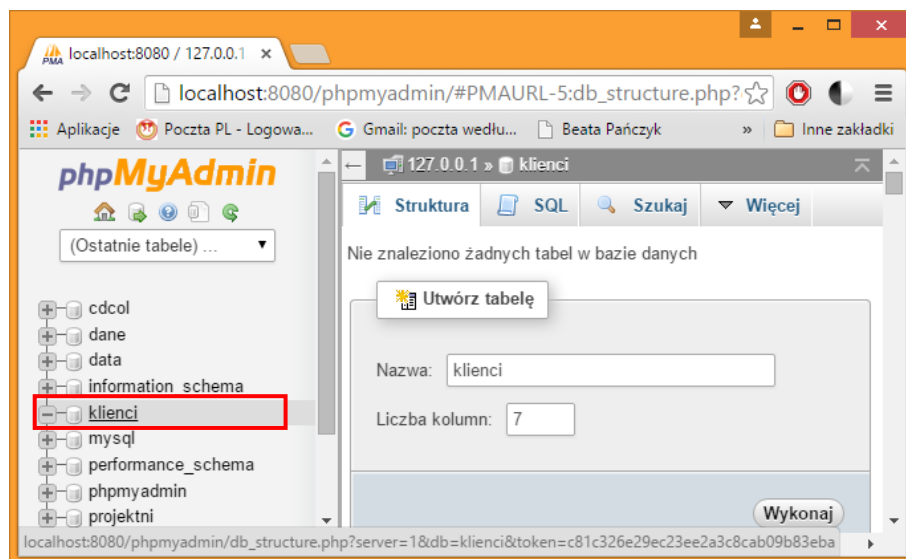


Ćw.1. Tworzenie bazy danych w PHPMyAdmin

- Uruchom serwer **Apache** i serwer **MySQL** (okno *XAMPP Control Panel*). W przeglądarce wpisz adres `localhost:numer_portu/phpmyadmin` (narzędzie wspierające szybkie zarządzanie bazami danych na serwerze **MySQL**).
- Utwórz nową bazę danych o nazwie **klienci** (metodę porównywania napisów ustaw na `utf8_general_ci`) (Rys.1) - po utworzeniu powinieneś ją zobaczyć na liście baz dostępnych na serwerze (pasek z lewej strony okna).
- Wskaż bazę **Klienci** dodaj do niej tabelę **klienci** o 7 polach (Rys.2): **Id** (*INT, PRIMARY KEY, AUTO_INCREMENT, UNSIGNED*), **Nazwisko** (*VARCHAR(40)*), **Wiek** (*TINYINT, UNSIGNED*), **Panstwo** (*ENUM*), **Email** (*VARCHAR(40)*), **Zamowienie** (*SET*), **Platnosc** (*ENUM*). Do definicji typów pól wykorzystaj typy danych **MySQL** (Rys.3) np.:
VARCHAR – pole tekstowe zmiennej długości,
TINYINT – liczby z zakresu 0-256 (jeśli *UNSIGNED*),
ENUM - stosuje się gdy w polu może się znaleźć jedna z podanych na liście wartości,
SET – tak jak *ENUM*, ale może to być kilka wartości równocześnie.
Wartości dla *ENUM* podajemy jako: 'wartosc1', 'wartosc2', 'wartosc3' a wartości domyślne wpisujemy po prostu: wartosc1).
- Zapisz kilka rekordów do stworzonej tabeli (Rys.4). Zwróć uwagę na składnię zapytania SQL (umiejętność sformułowania poprawnego zapytania SQL będzie niezbędna przy pracy z bazą danych z poziomu skryptu PHP).



Rys.1. Tworzenie nowej bazy danych *testowa*



Rys.2. Tworzenie tabeli **Klienci** w bazie *Testowa*

#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Dodatkowo
1	<u>Id</u>	int(10)		UNSIGNED	Nie	Brak	AUTO_INCREMENT
2	Nazwisko	varchar(40)	utf8_general_ci		Nie	Brak	
3	Wiek	tinyint(3)		UNSIGNED	Nie	Brak	
4	Panstwo	enum('Polska', 'Niemcy', 'Wielka Brytania', 'Czech')	utf8_general_ci		Nie	Polska	
5	Email	varchar(40)	utf8_general_ci		Nie	Brak	
6	Zamowienie	set('PHP', 'Java', 'CPP')	utf8_general_ci		Nie	PHP	
7	Platnosc	set('Visa', 'Master Card', 'Przelew')	utf32_general_ci		Nie	Visa	

Rys.3. Struktura tabeli *klienci*

Ćw.2. Praca z PHP i MySQL

Do zadania wykorzystaj formularz utworzony na poprzednich laboratoriach, tyle że tym razem wszystkie operacje zamiast na pliku wykonaj na bazie danych (przycisk *dodaj* – dodaje rekord z zamówieniem do tabeli *klienci*, przycisk *pokaz* – wyświetla wszystkie rekordy z tabeli *klienci* itd.) Do operacji na BD skorzystaj z pomocniczej klasy (Listing 1). Definicję klasy zapisz w pliku **Baza.php**. Na początek napisz skrypt, który korzystając z metod klasy **Baza** wyświetli wszystkie dane pobrane z tabeli *klienci*.

Schemat skryptu do pracy z bazą pokazuje Listing 2. Zauważ ponadto, że zapytanie SELECT można wykonać również z wykorzystaniem dopasowania do wyrażenia regularnego. Ma ono wtedy postać np.:

SELECT kolumna1,kolumna2 FROM tabela WHERE kolumna2 REGEXP 'Java'

Pamiętaj, żeby przed dodaniem danych do bazy sprawdzić ich poprawność i zastosować odpowiednie funkcje zabezpieczające (*addslashes*, *strip_tags*, *trim*).

Listing 1. Klasa Baza.php

```
<?php
class Baza {

    private $mysqli; //uchwyt do BD

    public function __construct($serwer, $user, $pass, $baza) {
        $this->mysqli = new mysqli($serwer, $user, $pass, $baza);
        /* sprawdź połączenie */
        if ($this->mysqli->connect_errno) {
            printf("Nie udało się połączenie z serwerem: %s\n",
                $mysqli->connect_error);
            exit();
        }
        /* zmień kodowanie na utf8 */
        if ($this->mysqli->set_charset("utf8")) {
            //udało się zmienić kodowanie
        }
    } //koniec funkcji konstruktora

    function __destruct() {
        $this->mysqli->close();
    }
}
```

```
public function select($sql, $pola) {
    //parametr $sql - łańcuch zapytania select
    //parametr $pola - tablica z nazwami pol w bazie
    //Wynik funkcji - kod HTML tabeli z rekordami (String)
    $tresc = "";
    if ($result = $this->mysqli->query($sql)) {
        $ilepol = count($pola); //ile pól
        $ile = $result->num_rows; //ile wierszy
        // pętla po wyniku zapytania $results
        $tresc."<table><tbody>";
        while ($row = $result->fetch_object()) {
            $tresc."<tr>";
            for ($i = 0; $i < $ilepol; $i++) {
                $p = $pola[$i];
                $tresc."<td>" . $row->$p . "</td>";
            }
            $tresc."</tr>";
        }
        $tresc."</table></tbody>";
        $result->close(); /* zwolnij pamięć */
    }
    return $tresc;
}

public function insert($sql) {
    // uzupełnij zapytanie - i zwróć true lub false
}

public function delete($sql) {
    // uzupełnij zapytanie - i zwróć true lub false
}
} //koniec klasy Baza

?>
```

Listing 2. Schemat skryptu

```
<?php
    //tworzymy uchwyt do bazy klienci:
    include_once "lib/Baza.php";
    $ob = new Baza("localhost", "root", "", "klienci");
    if (isset($_POST['pokaz'])) {
        echo $ob->select("select Nazwisko,Zamowienie from klienci",
                        array("Nazwisko","Zamowienie"));
    } else
    if (isset($_POST['dodaj'])) {
        //pobierz dane z formularza, dokonaj ich walidacji
        //Skorzystaj w tym celu z pomocniczej funkcji
        //sformułuj zapytanie insert i wywołaj metodę:
        //$ob->insert($sql);
    }
    //else ...

?>
```