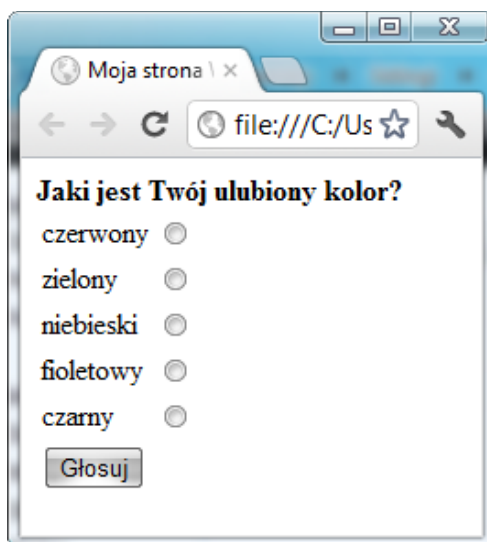


# Bazy danych w praktyce cz. 2

## 1. Ankieta

Często spotykanym elementem witryn są różnego rodzaju sondy i ankiety, które pozwalają na poznanie opinii na różne tematy osób odwiedzających witrynę. Zacznijmy od przygotowania formularza umożliwiającego oddawanie głosów. W tym przykładzie będzie można głosować na ulubione kolory. Formularz głosowania:



Dane będą zapisywane w bazie, w tabeli o nazwie ***kolory***, która będzie miała dwie kolumny – ***Kolor*** i ***Glosy***. W pierwszej będą zapisywane nazwy kolorów, a w drugiej liczba oddanych na nie głosów. Tabela taka generowana jest przez instrukcję SQL w postaci:

```
CREATE TABLE kolory(
```

```
Kolor VARCHAR(15) NOT NULL,
```

```
Glosy INTEGER
```

```
);
```

Po utworzeniu tabeli należy wprowadzić do niej dane początkowe, czyli nazwy kolorów i pierwotną liczbę głosów na każdego z nich. Tę czynność można wykonać za pomocą serii instrukcji SQL:

```
INSERT INTO kolory (Kolor, Glosy) VALUES ('czerwony',0);
```

```
INSERT INTO kolory (Kolor, Glosy) VALUES ('zielony',0);
```

```
INSERT INTO kolory (Kolor, Glosy) VALUES ('niebieski',0);
```

```
INSERT INTO kolory (Kolor, Glosy) VALUES ('fioletowy',0);
```

```
INSERT INTO kolory (Kolor, Glosy) VALUES ('czarny',0);
```

Skrypt powinien sprawdzać, czy zostały do niego przekazane dane określające kolor, na który zagłosował użytkownik. Jeżeli parametr został przekazany (tablica *\$\_GET*) jego wartość powinna być zapisywana w zmiennej pomocniczej *\$kolor*, jeśli nie został – zmiennej tej trzeba przypisać pusty ciąg znaków. Następnie musi nastąpić sprawdzenie, czy w zmiennej *\$kolor* znajdują się poprawne dane, to znaczy, czy jest ona różna od pustego ciągu znaków oraz czy zawiera jeden z ciągów znaków opisujących kolor.

Gdy dane są poprawne, nastąpi wywołanie funkcji, która zajmuje się uaktualnieniem tabeli *kolory* oraz wyświetleniem wyników głosowania.

Funkcja nawiązuje połączenie z bazą danych, a następnie wykonuje zapytanie uaktualniające dane w postaci:

***UPDATE kolory SET Glosy=Glosy+1 WHERE Kolor = '\$kolor'***

Zwiększa ono o jeden wartość pola *Glosy*, które odpowiada wybranemu kolorowi.

W tabeli wynikowej będzie prezentowana zarówno liczba, jak i procent głosów oddanych na dany kolor. W związku z tym po aktualizacji danych następuje wyliczenie procentów. Zaczyna się to od uzyskania informacji o sumie wszystkich oddanych głosów, za co odpowiada zapytanie:

***SELECT SUM(Glosy) FROM kolory***

Jego wynik jest zapisywany w zmiennej *\$suma*. Zmienna ta jest następnie wykorzystywana w kolejnym zapytaniu SQL, które już bezpośrednio wylicza procent głosów oddanych na każdy z kolorów. Wykorzystywana jest w tym celu instrukcja SQL w postaci:

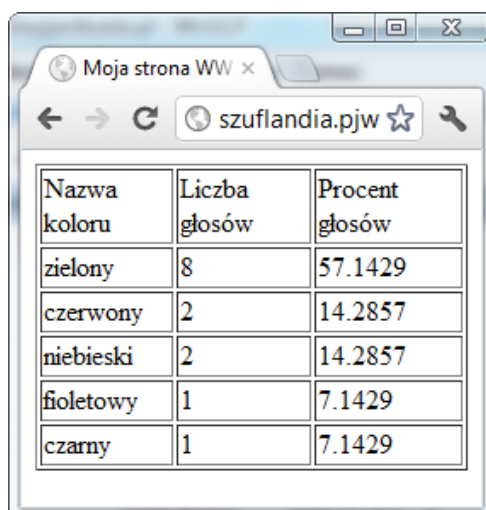
***SELECT Kolor, Glosy, Glosy\*100/\$suma AS Proc FROM kolory ORDER BY Glosy DESC***

Wartość procentowa jest wyliczana z prostego wzoru:

***Procent= liczba głosów na dany kolor \*100 / całkowita liczba głosów***

Całość jest sortowana w porządku malejącym, a zatem na górze tabeli wynikowej znajduje się kolor, na który zostało oddane najwięcej głosów.

Do wyświetlenia tak uzyskanych wyników na ekranie posłuży pętla *while* pobierająca kolejne wiersze tabeli wynikowej za pomocą metody *mysql\_fetch\_row*. W przeglądarce pojawi się więc tabela z wynikami głosowania:



The screenshot shows a web browser window with the title 'Moja strona WW' and the address bar displaying 'szuflandia.pjw'. The main content is a table with three columns: 'Nazwa koloru', 'Liczba głosów', and 'Procent głosów'. The table lists five colors: zielony (8 votes, 57.1429%), czerwony (2 votes, 14.2857%), niebieski (2 votes, 14.2857%), fioletowy (1 vote, 7.1429%), and czarny (1 vote, 7.1429%).

| Nazwa koloru | Liczba głosów | Procent głosów |
|--------------|---------------|----------------|
| zielony      | 8             | 57.1429        |
| czerwony     | 2             | 14.2857        |
| niebieski    | 2             | 14.2857        |
| fioletowy    | 1             | 7.1429         |
| czarny       | 1             | 7.1429         |

## Ćwiczenie

Napisz skrypt, który będzie umożliwiał zarządzanie danymi (dodawanie, modyfikację i usuwanie rekordów) w tabeli *users* używanej w skrypcie logowania.

### Zarządzanie hasłami:

|  |                            |                                 |
|--|----------------------------|---------------------------------|
| Użytkownik:                            | <input type="text"/>       |                                 |
| Hasło:                                 | <input type="text"/>       |                                 |
| <input checked="" type="radio"/> Dodaj | <input type="radio"/> Usuń | <input type="radio"/> Modyfikuj |
| <input type="button" value="Wykonaj"/> |                            |                                 |