

Bazy danych w praktyce.

1. Licznik

Pierwszym najprostszym przykładem będzie oparty na bazie danych licznik odwiedzin strony. Ten skrypt potraktujemy jako rozgrzewkę przed kolejnymi projektami. Oczywiście uruchamianie serwera baz danych i tworzenie bazy tylko dla licznika nie ma większego sensu. Jeśli jednak cała witryna jest oparta na systemie bazodanowym, nie stoi na przeszkodzie, aby umieścić w bazie i taką informację.

Zacznijmy od utworzenia tabeli licznik przechowującej liczbę odwiedzin. Będzie ona zawierała tylko jedną kolumnę typu *integer* o nazwie *Hits*.

Do tabeli wprowadzimy jeden wiersz odpowiadający początkowemu stanowi licznika.

Do pobrania aktualnej wartości licznika oraz jego uaktualnienie stwórzmy funkcję *getAndUpdateCounter*. Jej działanie powinno być takie, że zwracać ona ma aktualną liczbę odwiedzin odczytaną z bazy lub wartość *false*, jeśli z jakichś powodów odczyt lub aktualizacja danych zakończyły się niepowodzeniem.

Wewnątrz funkcji *getAndUpdateCounter* jest otwierane połączenie z bazą, a następnie wykonywane jest zapytanie zwracające zawartość tabeli *licznik*.

Po pobraniu aktualnej zawartości licznika niezbędne jest zwiększenie jego wartości o jeden. Czynność ta może być wykonana przez zapytanie w postaci:

UPDATE licznik SET Hits = Hits+1

Za kodem funkcji należy umieścić właściwą treść witryny, która może wyglądać tak:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
<title>Moja strona www</title>
</head>
<body>
<p>Treść strony</p>
<p>
<?php
if(($counter = getAndUpdateCounter()) == false){
    echo 'Licznik czasowo niedostępny';
}
else if($counter == 1){
    echo 'Ta strona została odwiedzona 1 raz od 20 maja 2012 roku.';
}
else{
    echo "Ta strona została odwiedzona $counter razy od 20 maja 2012 roku.";
}
?>
</p>
</body>
</html>
```

2. Liczba osób na stronie

Informacją często spotykaną na stronach jest liczba osób aktualnie przeglądających daną stronę. Skrypty realizujące takie zadanie nie są wcale skomplikowane, gdyż zazwyczaj przyjmują upraszczające założenie, że każdy użytkownik jest obecny na stronie przez taki sam czas. Jeśli będzie to na przykład **120** sekund, należy wtedy wykonać następujące czynności:

- Zapisanie czasu wejścia użytkownika na stronę w bazie danych,
- Usunięcie z bazy wszystkich wpisów starszych niż 120 sekund,
- Zsumowanie wpisów znajdujących się w bazie,
- Wyświetlenie uzyskanej wartości na ekranie.

Dane powinny być przechowywane w tabeli bazy danych o nazwie *czas_wizyt*. Tabela ta posiada tylko jedną kolumnę o nazwie *czas* typu *integer*

Każde wywołanie skryptu powinno nawiązywać połączenie z bazą i dodanie do tabeli nowego wpisu.

Po uzyskaniu aktualnego czasu usuwane są wszystkie wpisy starsze niż 120 sekund.

Po wykonaniu instrukcji usunięcia wpisów w tabeli pozostaną jedynie wpisy nie starsze niż 120 sekund. Wystarczy je zatem policzyć, aby stwierdzić ile osób w ostatnim czasie było obecnych na stronie. Tego obliczenia dokonuje instrukcja:

SELECT COUNT(*) FROM czas_wizyt

Przykładowy fragment kodu wyświetlający liczbę osób obecnych na stronie:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
<title>Moja strona WWW</title>
</head>
<body>
<div>
<?php
if (($file = visitsCount($timeout)) !== false) {
    echo "Liczba osób obecnych na stronie w ciągu ostatnich ";
    echo "$timeout sekund: $file.";
}
else{
    echo "Licznik ostatnich wizyt nie jest dostępny.";
}
?>
</div>
</body>
</html>
```

3. Logowanie

Na laboratorium 7 powstał prosty system kontroli dostępu do witryny, w którym nazwy i hasła użytkowników były zapisane w pliku tekstowym. Spróbujmy więc teraz wykonać to zadanie, opierając się na bazie danych *MySQL*. W tym celu w bazie należy utworzyć tabelę *users*, do czego można wykorzystać instrukcje *SQL*:

```
CREATE TABLE users(  
  
id INTEGER AUTO_INCREMENT PRIMARY KEY,  
  
nazwa VARCHAR(40) UNIQUE.  
  
haslo VARCHAR(40),  
  
email VARCHAR(40)  
  
);
```

Od razu warto też wprowadzić do niej przykładowy wiersz, który pozwoli na testowanie skryptu:

```
INSERT INTO users (nazwa,haslo) VALUES('user1', 'pass1', 'user1@mojastrona.pl');
```

Całość będzie się składała z trzech plików:

- *index.php* – zawiera procedury i formularz logowania
- *glowna.php* – zawiera główną część serwisu,
- *logout.php* – zawiera procedury wylogowania.

Kod strony *glowna.php*:

```
1 <?php  
2 session_start();  
3 if(!isset($_SESSION['zalogowany'])) {  
4     header("Location: index.php");  
5     exit();  
6 }  
7 ?>  
8 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">  
9 <html>  
10 <head>  
11 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">  
12 <title>Moja strona WWW</title>  
13 </head>  
14 <body>  
15 <div>  
16 Witamy na stronie.  
17 </div>  
18 <p>Pamiętaj o wylogowaniu przed opuszczeniem strony.</p>  
19 <p><a href="http://localhost/logout.php">Wylogowanie</a></p>  
20 </body>  
21 </html>
```

Kod strony logout.php

```
1 <?php
2 session_start();
3 if(isset($_SESSION['zalogowany'])) {
4     unset($_SESSION['zalogowany']);
5 }
6 else{
7     header("Location: index.php");
8     exit();
9 }
10 if(isset($_COOKIE[session_name()])){
11     setcookie(session_name(), '', time() - 360);
12 }
13 session_destroy();
14 ?>
15 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
16 <html>
17 <head>
18 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
19 <title>Moja strona WWW</title>
20 </head>
21 <body>
22 <p>Wylogowanie prawidłowe</p>
23 <p><a href='http://localhost/index.php'>Powrót do strony logowania</a></p>
24 </body>
25 </html>
```

Za całą kontrolę dostępu w pliku *index.php* odpowiada funkcja *checkPass*. W jej treści najpierw następuje nawiązanie połączenia z serwerem *MySQL* oraz wybranie bazy. Następnie wykonywane jest zapytanie, którego zdaniem jest pobranie hasła użytkownika o nazwie przekazanej w postaci argumentu.

W kolejnym kroku jest sprawdzane, czy hasło odczytane z bazy jest takie samo jak hasło otrzymane z przeglądarki. Jeśli tak jest, użytkownik może być zalogowany i funkcja zwraca wartość *true*. W przeciwnym razie zwracana jest wartość *false*.

Fragment kodu *index.php*:

```
if(isset($_SESSION['zalogowany'])) {  
    header("Location: glowna.php");  
    exit();  
}  
else if(isset($_POST['user']) && isset($_POST['haslo'])) {  
    if(checkPass($_POST['user'], $_POST['haslo'])) {  
        $_SESSION['zalogowany'] = $_POST['user'];  
        header("Location: glowna.php");  
        exit();  
    }  
    else {  
        $komunikat = "Niepoprawne dane logowania.";  
    }  
}  
?>
```

Ćwiczenia do samodzielnego wykonania

Ćwiczenie 1

Do serwisu logowania dodaj stronę rejestracja.php, na której użytkownik będzie mógł się zarejestrować w naszym serwisie. Strona zawierać ma formularz z wymaganymi polami do rejestracji (login, email, hasło, hasło2).