

MySQL - informacje różne

Położenie: (nie dotyczy)

© 3bird Projects 2020, <http://edukacja.3bird.pl>

Uruchamianie

Uruchamianie serwera:

```
# /etc/init.d/mysql start
```

Testowanie serwera i sprawdzanie wersji (musi być już konto rootMySQL z hasłem):

```
# mysqladmin -u root -p version
```

Inny sposób:

```
mysql> select version(), current_date, now(), user();
```

Użycie serwera mySQL jako kalkulatora:

```
mysql> select (4+5)*7;
```

Wchodzenie do klienta (z konta zwykłego użytkownika):

```
$ mysql
```

Wchodzenie do klienta na konto użytkownika, który ma hasło:

```
$ mysql -u Użytkownik -p
```

To samo, ale od razu otwieramy bazę danych:

```
$ mysql -u Użytkownik -p nazwaBazy
```

Łączenie się ze zdalnym serwerem mySQL:

```
$ mysql -u Użytkownik -p -h 192.168.1.1
```

Bazy

Wyświetlenie listy testowych przykładowych baz danych:

```
mysql> show databases;
```

Załadowanie konkretnej bazy:

```
mysql> use nazwaBazyDanych;
```

Utworzenie nowej bazy (robimy to tylko jako *rootMySQL*):

```
mysql> create database nazwaBazyDanych;
```

Usuwanie bazy danych (robimy to tylko jako *rootMySQL*):

```
mysql> drop database nazwaBazyDanych;
```

Tabele

Wyświetlanie tabel z bazyDanych:

```
mysql> show tables;
```

Tworzenie tabeli:

```
mysql> create table nazwaTabeli (nazwaKolumny1 atrybuty, nazwaKolumny2 strybuty);
```

Usuwanie tabeli z bazyDanych:

```
mysql> drop table nazwaTabeli;
```

Zmiana nazwy tabeli:

```
mysql> alter table nazwaTabeli rename nowaNazwaTabeli;
```

Kolumny

Wyświetlanie kolumn z tabeli:

```
mysql> describe nazwaTabeli;
```

Dodawanie nowej kolumny:

```
mysql> alter table nazwaTabeli add nazwaNowejKolumny varchar(30) after nazwaKolumnyPoprzedniej;
```

Zmiana nazwy kolumny:

```
mysql> alter table nazwaTabeli change nazwaKolumny nazwaNowejKolumny;
```

Zmiana typu danych w kolumnie:

```
mysql> alter table nazwaTabeli modify nazwaKolumny varchar(60);
```

Wyście z klienta:

```
mysql> exit
```

Utworzenie pierwszej bazy (przykład)

```
mysql> create database MoiKlienci;
```

```
mysql> use MoiKlienci;
```

```
mysql> create table DaneOsobowe (
```

```
mysql> ID int not null auto_increment,
```

```
mysql> Imie varchar(25) character set utf8 not null,
```

```
mysql> Nazwisko varchar(25) character set utf8 not null,
```

```
mysql> Email text,
```

```
mysql> primary key(ID));
```

Tabela zawiera 4 kolumny: *ID*, *Imie*, *Nazwisko*, *Email*. Atrybuty kolumn:

int - liczby całkowite;

tinyint - liczby od -128 do 127;

real - liczby rzeczywiste;

date - data, np. 2005-11-07;

year(4) - rok zapisany za pomocą 4 cyfr;

text - typ tekstowy o długości 65538 znaków;

char(25) - typ znakowy (sztywno określona ilość znaków w nawiasach); jeśli kolumny w tabeli są różnego typu, char większy niż 3 zostanie automatycznie zamieniony na varchar;

varchar(25) - typ znakowy(ilośćZnaków) dynamiczny, tzn. Zaniża liczbę znaków, jeśli można (ale dodatkowo obciąża to procesor); jeżeli nie podamy liczby znaków, każda komórka zużyje maksymalną liczbę, czyli 65538 (duży rozmiar pliku będzie);

not null - nie może być puste;

auto_increment - powiększanie numeracji o 1;

primary key(Nazwa) - ustawienie klucza podstawowego na daną kolumnę;

Ustawianie praw do utworzonej bazy danych

Rodzaje praw:

- **select** - prawo do odczytu, sortowania danych;
- **insert** - prawo do wstawiania danych;
- **update** - prawo do poprawiania danych;
- **delete** - prawo do kasowania danych;

```
mysql> grant rodzajPraw on nazwaBazyDanych.nazwaTabeli to nazwaUzytkownika@host identified by hasloUzytkownika;
```

Przykład:

Tworzenie nowego użytkownika i dodawanie mu praw do bazy:

```
mysql> grant all on nazwaBazyDanych.* to robert@localhost identified by 'hasloUzytkownika';
```

Dodawanie praw istniejącemu użytkownikowi:

```
mysql> grant delete, update on * to robert@localhost;
```

Usunięcie tych praw:

```
mysql> revoke delete, update on * from robert@localhost;
```

Wprowadzanie danych do bazy

```
mysql> use nazwaBazyDanych;
```

```
mysql> insert nazwaTabeli (Nazwisko, Imie, Email) values ('Surma', 'Robert', 'surma@xyz.pl');
```

To samo, ale bez podawania kolumn:

```
mysql> insert into nazwaTabeli values ('Surma', 'Robert', 'surma@xyz.pl');
```

```
mysql> select * from nazwaTabeli where name='Surma' and wiek<=36;
```

Sortowanie:

```
mysql> select nazwisko, urodziny from nazwaTabeli order by urodziny;
```

To samo, ale malejąco:

```
mysql> select nazwisko, urodziny from nazwaTabeli order by urodziny desc;
```

Użycie dwóch tabel:

```
mysql> select nazwaTabeli1.nazwaKolumny as nowaWirtualnaNazwaKolumny, nazwaTabeli2.nazwaKolumny as nowaWirtualnaNazwaKolumny from nazwaTabeli1 nazwaTabeli2;
```

```
mysql> update nazwaTabeli set Email='nowy@email.pl' where ID=1;
```

```
mysql> delete from nazwaTabeli where ID=1;
```

Można dane do bazy wczytać z pliku tekstowego (wartości oddzielane są tabulatorem, a puste rekordy to “\N”):

```
mysql> LOAD DATA LOCAL INFILE '/home/robert/nazwaPliku.txt' INTO TABLE nazwaTabeli;
```

Konfiguracja

Kolejność odczytywania opcji serwera:

- `/etc/mysql/my.cnf`
- `/var/lib/mysql/my.cnf`
- `$home/.my.cnf`

Pliki baz danych zapisywane są domyślnie w: `/var/lib/mysql` (można to zmienić za pomocą opcji “`datadir=ścieżka`” w pliku konfiguracyjnym).

Ustalanie hasła dla MySQL

Uwaga: administrator serwera o nazwie “root” nie ma nic wspólnego z systemowym administratorem.

```
# /etc/init.d/mysql start
```

(Należy wykonać łącznie te dwa kroki: pierwszy ustawia użytkownika na hoście “localhost”, drugi na “server”):

```
# /usr/bin/mysqladmin -u root password "TreśćHasła"
```

```
# /usr/bin/mysqladmin -u root -h server password "TreśćHasła"
```

Jeśli powyższa nazwa hosta nie zostanie zaakceptowana, należy użyć tej komendy:

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'treśćHasła' WITH GRANT OPTION;
```

Inny sposób (należy wykonać dwa kroki):

```
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('treśćHasła');
```

```
mysql> SET PASSWORD FOR 'root'@'server' = PASSWORD('treśćHasła');
```

Odzyskiwanie utraconego hasła rootMySQL

Pierwszy sposób:

1. Zabić wszystkie procesy MySQL.

2. Utworzyć plik tekstowy, w którym będzie linia: `SET PASSWORD FOR 'root'@'localhost' = PASSWORD('treśćNowegoHasła');`
3. Uruchomić ponownie serwer MySQL z opcją: **mysqld_safe --init-file=~/*nazwaUtworzonegoPliku* &**

Drugi sposób:

1. Ponownie uruchomić serwer MySQL z opcją: **mysqld_safe --skip-grant-tables --user=root &**
2. Zalogować się w nim: **mysql -u root**
3. Wydać polecenia:

```
mysql> UPDATE mysql.user SET Password=PASSWORD('treśćNowegoHasła') WHERE User='root';
```

```
mysql> FLUSH PRIVILEGES;
```

Zarządzanie użytkownikami MySQL

Informacje o użytkownikach przechowywane są w predefiniowanej bazie danych o nazwie "mysql". Nazwy tabel:

- **column_priv** - uprawnienia do kolumn;
- **db** - uprawnienia do baz danych;
- **tables_priv** - uprawnienia do tabel;
- **host** - uprawnienia hostów do baz danych;
- **user** - opis userów (użytkowników MySQL);

Aby wyświetlić zawartość kolumn z danej tabeli:

```
mysql> select nazwyKolumn from nazwaTabeli;
```

Przykład:

```
mysql> use mysql;
```

```
mysql> show tables;
```

```
mysql> describe user;
```

```
mysql> select user, host, select_priv, insert_priv, delete_priv, grant_priv from user;
```

```
mysql> select user, select_priv, insert_priv from user where user = student and user = robert;
```

Uwaga: Aby bez restartowania serwer uwzględnił zmiany, należy wydać polecenie: **flush privileges** (resetuje bufor serwera MySQL).

Naruszenie bezpieczeństwa

1. W pliku `.mysql_history` zapisywana jest historia poleceń, która może zawierać hasła, jeśli logowanie do serwera nastąpiło poprzez polecenie: **mysql -u root -p TreśćHasła;**
2. Jako użytkownik `rootMySQL` tworzymy nazwę bazy danych oraz jej tabelki. Same dane wprowadzamy potem wyłącznie jako zwykły użytkownik.
3. Do konta `rootMySQL` logujemy się z poziomu konta zwykłego użytkownika (nie z poziomu systemowego użytkownika `root`).
4. Zwykłemu `uzytkownikowiMySQL` nigdy nie nadajemy pełni praw (**all**) do bazy danych.
5. Serwer MySQL nie powinien być uruchamiany na prawach `roota`. Jeśli zachodzi potrzeba uruchomienia go z konta `roota`, należy zastosować polecenie: **/bin/mysqld_safe --user=mysql** (w `Gentoo` należy to zmodyfikować w skrypcie `/etc/init.d/mysql`).
6. Należy usunąć domyślne anonimowe konto użytkownika (jest to konto bez nazwy): **DELETE FROM mysql.user WHERE User = '';** Efekty możemy sprawdzić wydając polecenie: **SELECT Host, User FROM mysql.user;**
7. Należy przetestować nasz serwer z Internetu wydając polecenie "**nmap -O naszIP**". Port 3306 nie może być dostępny z Internetu dla każdego hosta.
8. Należy przetestować nasz serwer z Internetu wydając polecenie "**telnet naszIP 3306**". Nie powinno nastąpić połączenie, port musi być zablokowany.

9. Aby dowiedzieć się, czy transmitowane dane przez Internet są szyfrowane (przez SSH lub SSL) należy wydać polecenie: **tcpdump -l -i eth0 -w - src or dst port 3306 | strings**. Jeśli widać odkryty tekst, to znaczy, że nie ma szyfrowania.
10. Nie należy nadawać zwykłym użytkownikom przywilejów: *process, super, file*.

Obsługa mySQL przez PHP

Niezbędne jest uaktywnienie modułu `mysql.so` w pliku `/etc/apache2/conf/php.ini` (jest to zazwyczaj link do `/etc/php/apache2-php4/php.ini`):

```
;UNIX Extensions
```

```
extension=mysql.so
```

Należy także zwrócić uwagę na poprawność ścieżki:

```
extension_dir=/usr/lib/apache
```

Kodowanie znaków

Uwaga: Kodowanie `utf8mb4`, w przeciwieństwie do zwykłego `utf8`, zawiera pełną mapę znaków.

Jeśli mamy prawo do tworzenia bazy danych w `phpMyAdmin`:

```
SQL> CREATE DATABASE nazwaBazy CHARACTER SET utf8mb4 COLLATE utf8mb4_polish_ci;
```

Aby zmienić kodowanie w już istniejącej bazie danych:

```
SQL> ALTER DATABASE nazwaBazy CHARACTER SET utf8mb4 COLLATE utf8mb4_polish_ci;
```

Aby zmienić kodowanie w tabeli:

```
SQL> ALTER TABLE nazwaTabeli CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_polish_ci;
```

Aby zmienić kodowanie w szablonie (nie ma wpływu na już stworzone tabele):

```
SQL> ALTER SCHEMA nazwaBazy DEFAULT CHARACTER SET utf8mb4 DEFAULT COLLATE utf8mb4_unicode_ci;
```

Uwaga: Jeśli baza danych została utworzona w obcym kodowaniu, to późniejsza zmiana tego kodowania i tak nie wpłynie na wyświetlanie polskich znaków na stronie www (wpływa jedynie na wyświetlanie znaków na stronie www). Aby rozwiązać ten problem, należy dodać odpowiednie kodowanie w połączeniu z bazą danych w skrypcie PHP:

```
mysqli_query($daneDoPolaczenia, "SET NAMES utf8mb4");
```

Ostatnia aktualizacja: 10 grudnia 2020.