

## Zarządzanie bazami danych MySQL

# Przyjemna praca

Blogi, wiki, galerie obrazów – większość aplikacji z interfejsem WWW potrzebuje bazy danych. Przedstawiamy kilka podstawowych sztuczek, które wystarczą, by stać się administratorem bazy MySQL.

OLIVER FROMMEL



**T**ermin „administrowanie bazami danych” brzmi bardzo poważnie i od razu wzbudza respekt – zwłaszcza, że opisywany dalej MySQL [1] wykorzystywany jest przez duże firmy. Nie ma się jednak czym przejmować, zwykle aplikacje biurowe wymagają często dużej wiedzy i umiejętności, tymczasem zarządzanie MySQL może być całkiem proste.

Niniejszy artykuł koncentruje się na podstawowych czynnościach administracyjnych, wymaganych przez większość aplikacji WWW wykorzystujących bazy danych. Zaczniemy jednak od procedury instalacji (patrz Ramka „Instalacja”).

Istnieje kilka narzędzi administracyjnych wykorzystujących interfejs graficzny (GUI). Pamiętajmy jednak, że programy konsolowe dołączone do MySQL, takie jak np. opisany poniżej *mysql*, są bardzo wygodne i łatwe do opanowania. Korzystanie z klienta znakowego wymaga takiej samej wiedzy jak w przypadku GUI, a przynajmniej nie trzeba się uczyć struktury menu i okien danego narzędzia.

Po opanowaniu korzystania z klientów znakowych MySQL nie będzie problemu z używaniem dowolnego klienta GUI, takiego jak KMySQLAdmin [2], MySQL Navigator oraz *mysqlcc*, który jest programem producenta bazy danych – to tylko część opartych o GUI programów administracyjnych.

Klient *mysql*, uruchomiony bez żadnych parametrów, wykorzysta bieżącą nazwę użytkownika do połączenia się bez hasła z lokalnym serwerem MySQL. Parametr *-u* umożliwia użycie innej nazwy użytkownika, natomiast *-p* wymusza monit o hasło. Może po nim opcjonalnie nastąpić nazwa bazy danych, z którą ma nastąpić połączenie. Jeśli np. chcemy uzyskać dostęp do bazy jako *root* (z hasłem), używamy następującego polecenia:

```
mysql -u root -p mysql
```

Jak na program terminalowy, *mysql* jest łatwy w użyciu. Posiada historię poleceń, którą użytkownicy mogą przeglądać naciskając strzałkę w górę. Polecenia wprowadzane po znaku za-

chęty *mysql>* kończymy średnikiem. Jeśli przed zakończeniem wprowadzania polecenia wciśnięty zostanie klawisz Enter, program wyświetli *->* wskazując, że jeszcze coś należy wpisać. Wpisanie *\q* powoduje zakończenie programu.

## Tworzenie bazy danych

Ściśle rzecz biorąc, MySQL nie jest „bazą danych”, ale systemem zarządzania bazami danych (DBMS). Innymi słowy, MySQL może jednocześnie zarządzać wieloma bazami danych (grupami tabel), np. użytkownicy mogą stworzyć na jednym serwerze MySQL różne bazy danych do zarządzania swoimi kolekcjami CD i listami adresowymi. Wpisując *show databases;* w kliencie *mysql*, otrzymamy listę istniejących baz danych. Do tworzenia nowej bazy danych wykorzystujemy program *mysqladmin*. Jest to standardowe narzędzie służące do administracji MySQL. Bazę możemy utworzyć poleceniem:

```
mysqladmin create pic_db
```

## Ramka 1: Instalacja

Instalacja MySQL jest łatwa, ponieważ niemal wszystkie dystrybucje zawierają tę bazę danych. Jeżeli nie masz jej zainstalowanej, wykorzystaj menadżera pakietów swojej dystrybucji do wyszukania i dodania pakietów MySQL. Wskazana jest instalacja wszystkich pakietów, włącznie z tymi dla programistów, które zwykle mają w nazwie *devel* lub *dev*. Wymagane są one do kompilacji kodu źródłowego wielu programów.

Przykładowo, MySQL w Fedora 2 składa się z następujących pakietów: `mysql-3.23.58-9.i386.rpm` (biblioteka MySQL, klient i inne narzędzia), `mysql-bench-3.23.58-`

`9.i386.rpm` (skrypty i dane wykorzystywane do testów wydajności), `mysql-devel-3.23.58-9.i386.rpm` (pakiety dla programistów), `mysql-jdbc-3.0.8-2.noarch.rpm` (bazodanowy interfejs dla Java), `mysql-jdbc-tomcat-3.0.8-2.noarch.rpm` (sterownik dla serwera Tomcat) i `mysql-server-3.23.58-9.i386.rpm` (właściwy serwer MySQL).

SUSE posiada podobny zestaw pakietów, z tym że w tej dystrybucji monitor znajduje się w pakiecie `mysql-client`, a biblioteka w `mysql-shared`. Najważniejszym plikiem konfiguracyjnym serwera MySQL jest `/etc/my.cnf`.

## Ramka 2: Zapomniane hasła

Jeśli zapomnieliśmy hasła administratora bazy danych (czyli zwykle użytkownika `root`), to nie będzie można wykorzystywać konta administratora do łączenia się z bazami danych. Na szczęście możliwe jest uruchomienie MySQL w trybie, w którym program ignoruje tabele zawierające uprawnienia. Należy uruchomić MySQL w podany poniżej sposób i zmienić hasło bazy danych dla `roota`:

```
safe mysqld --skip-grant-tables
mysqladmin -u root flush-?
privileges password „nowe_haslo”
```

Jeśli MySQL wymaga podania hasła, używamy tej samej opcji co w przypadku klienta, czyli `-p`, np. `mysqladmin -p create pic_db`. Tak naprawdę bazę można utworzyć z poziomu programu `mysql` np.:

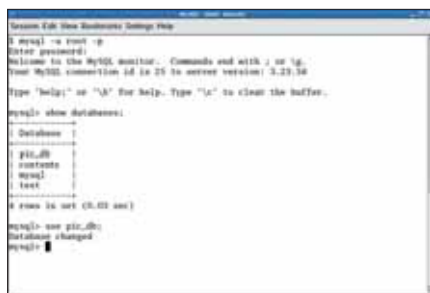
```
mysql> create database pic_db;
```

Usunięcie bazy danych wymaga wprowadzenia analogicznej komendy z parametrem `drop` zamiast `create`.

Aby użyć określonej bazy danych, wpisujemy w kliencie `use baza_danych`. Od tego momentu MySQL będzie pracował z wybraną bazą danych i wszystkie wykonywane od tej pory polecenia i zapytania będą się do niej odnosić. Jeśli chcemy edytować tabele systemowe MySQL (np. po to, by zmodyfikować uprawnienia użytkowników), najpierw wybieramy bazę danych `use mysql`; jeśli ta operacja się nie powiedzie, wyświetlony zostanie komunikat błędny `ERROR 1046: No Database Selected`. W celu wyświetlenia tabeli w wybranej bazie danych wpisujemy `show tables`;

## Zarządzanie użytkownikami

Administrator bazy danych MySQL nie musi mieć w Linuksie uprawnień `roota`. Większość dystrybucji definiuje użytkownika `root` jako



Rysunek 1: Klient `mysql` to program łatwy w obsłudze.

domyślnego administratora serwera baz danych z pełnymi uprawnieniami. Skrypt `mysql_install_db` może posłużyć do stworzenia tabel systemowych (m.in. zawierających uprawnienia), gdy baza danych jest pusta po instalacji lub uległa uszkodzeniu. Oczywiście należy też natychmiast po instalacji ustawić hasło administratora bazy:

```
mysqladmin -u root ?
password 'haselko'
```

Pracując z kilkoma bazami danych, najwygodniej utworzyć dla każdej z nich użytkownika o nazwie takiej samej jak nazwa bazy danych. Wiele aplikacji WWW wymaga podania nazwy i hasła użytkownika, który ma uprawnienia do wydawania zapytań SQL, tworzenia lub usuwania tabel w ramach danej bazy. Jeśli użytkownik, któremu przyznajemy prawa, nie istnieje, MySQL stworzy go automatycznie:

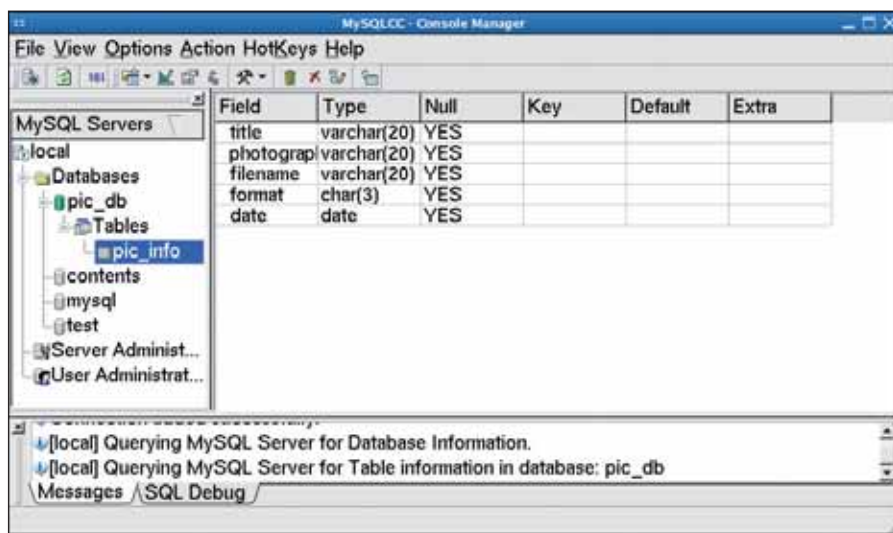
```
mysql> GRANT ALL ON pic_db.* TO ?
pic_user@localhost IDENTIFIED ?
BY 'haselko';
```

```
mysql> flush privileges;
```

Jak już wspomnieliśmy, pierwsze polecenie nadaje wszystkie uprawnienia do bazy danych o nazwie `pic_db` użytkownikowi `pic_user` – będzie on musiał podawać hasło `haselko`, aby połączyć się z bazą. Wpis `@localhost` oznacza, że użytkownik ten może łączyć się z serwerem jedynie z lokalnej maszyny. Aby zezwolić na zdalny dostęp, należy w tym miejscu wprowadzić nazwę hosta lub adres IP. Tematy związane z bezpieczeństwem wykraczają poza zakres materiału przedstawianego w tym artykule, niemniej warto przemyśleć sposób, w jaki będzie wykorzystywana baza danych. Jeśli np. zezwolimy innym maszynom na dostęp do naszego serwera, być może trzeba będzie zabezpieczyć go stosując np. `firewall-a`. Najbezpieczniejszą metodą jest dopuszczenie tylko lokalnego dostępu – wystarczy to dla większości aplikacji WWW, gdyż są one najczęściej uruchamiane na tej samej maszynie. Dwa programy dołączone do pakietu MySQL mogą okazać się bardzo przydatne do konfiguracji zabezpieczeń: `mysql_setpermission`, służący

## Tabela 1: Ważne narzędzia

Narzędzie	Opis
<code>mysql</code>	Klient (zwany też monitorem MySQL)
<code>mysqladmin</code>	Narzędzie administracyjne służące do tworzenia baz danych, zmian haseł itp.
<code>mysql_setpermission</code>	Zarządzanie uprawnieniami
<code>mysqlaccess</code>	Wyświetlanie uprawnień
<code>mysqlcheck</code>	Sprawdza, optymalizuje i naprawia tabele
<code>mysqldump</code>	Generuje pliki zawierające strukturę i zawartość baz danych w formie pliku tekstowego
<code>mysqlhotcopy</code>	Narzędzie służące do tworzenia kopii baz danych „na gorąco”, podczas gdy serwer MySQL jest uruchomiony
<code>mysqlimport</code>	Importuje pliki o określonej strukturze, np. wartości oddzielone przecinkami
<code>mysqlsamcheck</code>	Sprawdza i naprawia tabele
<code>mysqlshow</code>	Wyświetla strukturę bazy danych
<code>mysql_config</code>	Wyświetla położenie plików nagłówkowych, bibliotek, wersję MySQL etc.



Rysunek 2: Nowy program producenta bazy MySQL z graficznym interfejsem użytkownika mysqlcc.

do zarządzania uprawnieniami, oraz mysql\_access – do przeglądania zaprezentowanych w czytelnej formie bieżących uprawnień.

## Tworzenie tabel

Tworzenie tabel jest podobne do tworzenia baz danych czy użytkowników, jednakże w tym wypadku należy dodatkowo opisać strukturę tabeli. Każda tabela składa się z pewnej liczby pól, w których będą przechowywane później dane. Typy danych określają rodzaj przechowywanych przez pole danych: np. tekst, pojedyncze litery, liczby dziesiętne, całkowite itd. Aplikacje WWW najczęściej tworzą tabele samodzielnie, dlatego jedyne co należy zrobić, to stworzyć bazę danych, użytkownika oraz określić hasło. Spójrzmy na prosty przykład bazy danych z fotografiami:

```
mysql> CREATE TABLE
-> pic_info (
-> file VARCHAR(20),
-> format CHAR(3),
-> date DATE);
```

Zmienna VARCHAR oznacza łańcuch (litery, cyfry,..) o zmiennej długości. W tym przykładzie jego długość została ograniczona do 20 znaków. Zmienna CHAR(3) oznacza dokładnie trzy znaki. DATE jest formatem daty, który MySQL wyświetla jako 'RRRR-MM-DD', czyli 2004-08-31 dla 31 sierpnia 2004. MySQL oferuje duży wybór typów danych, zostały one dokładnie opisane w dokumentacji [3]. Bardzo wygodne w zarządzaniu tabelami są narzędzia GUI – korzystając z nich użytkownicy nie muszą się martwić o odpowiednie przecinki i nawiasy. Także klient znakowy posiada alternatywny tryb wprowadzania poleceń: wpi-

sanie edit uruchamia edytor tekstowy, w którym można wpisywać serie zapytań SQL. Zmienna środowiskowa EDITOR pozwala na określenie używanego edytora.

Aby wyświetlić strukturę tabeli, można użyć w kliencie znakowym polecenia describe (Rysunek 3):

```
mysql> describe pic_info;
```

Pliki tekstowe ułatwiają wprowadzanie rozbudowanych definicji tabel. Klient MySQL wykorzystuje takie pliki na wiele sposobów. Po pierwsze, jeśli klient jest uruchomiony do załadowania pliku zewnętrznego, można użyć polecenia source nazwa pliku (albo krócej \.). Można też użyć tradycyjnego przekierowania z linii poleceń, tak aby klient pobrał dane z pliku.

```
mysql < sqldata.txt
```

W ten sposób można wczytywać do MySQL pliki wygenerowane przez mysqldump (Tabela 1).

Zwykle każda tabela posiada tzw. klucz podstawowy (primary key). Jest to pole, które nie może pozostać puste, ponieważ umożliwia ono dostęp do każdego wiersza tabeli. Na ogół zawiera ono numer seryjny, który nie może być zmieniany, w przeciwieństwie do innych pól w tabeli. MySQL wykorzystuje atrybut AUTO\_INCREMENT, aby automatycznie wstawiać numer dla każdego kolejnego wiersza. Dodanie do siebie tych dwóch elementów spowoduje, że nasz przykład będzie wyglądał jak poniżej:

## Listing 1: Przykład użycia języka SQL

```
mysql> INSERT INTO pic_info (file, format) VALUES ('heaven.jpg', 'JPG');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO pic_info (file, format) VALUES ('earth.gif', 'GIF');
Query OK, 1 row affected (0.00 sec)
```

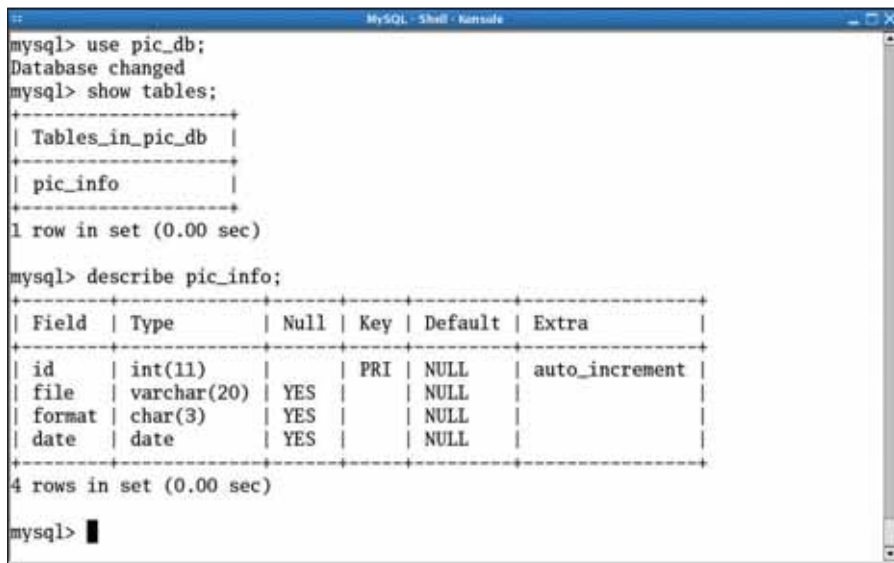
```
mysql> select * from pic_info;
+----+-----+-----+-----+
| id | file | format | date |
+----+-----+-----+-----+
| 1 | heaven.jpg | JPG | NULL |
| 2 | earth.gif | GIF | NULL |
+----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM pic_info WHERE format = 'JPG';
```

```
+----+-----+-----+-----+
| id | file | format | date |
+----+-----+-----+-----+
| 1 | heaven.jpg | JPG | NULL |
+----+-----+-----+-----+
1 row in set (0.03 sec)
```

```
mysql> UPDATE pic_info SET date = '2004-07-22' WHERE file = 'heaven.jpg';
```

```
mysql> DELETE FROM pic_info WHERE id = 1;
```



Rysunek 3: Klient mysql wyświetla strukturę tabeli pic\_info w bazie danych pic\_db.

```
mysql> CREATE TABLE
-> pic_info (
-> id INT AUTO_INCREMENT,
-> file VARCHAR(20),
-> format CHAR(3),
-> date DATE,
-> PRIMARY KEY (id));
```

Stosowanie bazy danych zaczyna mieć sens, gdy łączymy wiele tabel, pozwalając w ten sposób na efektywne przechowywanie danych. W naszym przykładzie mogłoby to oznaczać stworzenie tabel przechowujących dane fotografa, opisy fotografii, a nawet same fotografie. Zamiast przechowywać szczegóły dotyczące fotografa w tabeli, której się obecnie przyglądamy, do tablicy zawierającej dane fotogra-

fów możemy dołączyć numery ID fotografii danego fotografa. Mimo pozorów, projektowanie baz danych nie jest trywialnym zadaniem i wykracza poza zakres tego artykułu.

### Zapytania

SQL – Structured Query Language – standardyzowany język zapytań, umożliwiający zapisywanie, manipulowanie oraz odczyt danych w tabelach. MySQL oczywiście wspiera ten język. Mimo że inne języki programowania lub programy GUI umożliwiają pracę bez znajomości SQL, warto poznać jego podstawy. Głównymi poleceniami SQL są: SELECT, INSERT, UPDATE i DELETE. Listing 1 pokazuje ich wykorzystanie na przykładzie wcześniej omawianej tabeli.

Instrukcja INSERT wstawia rekord do tabeli. Wstawienie rekordów zwykle nie wymaga określenia wszystkich pól, niemniej potrzebny będzie klucz podstawowy (primary key). W tym przypadku MySQL sam zajmie się kluczem głównym, o ile istnieje pole AUTO\_INCREMENT. Instrukcja SELECT, wykorzystująca maskę (\*), wyświetla wszystkie wiersze tabeli, WHERE wprowadza ograniczenie liczby wyświetlanych wyników. W tym przypadku SELECT wyświetli tylko te wiersze, które mają JPG w polu format. UPDATE zmienia istniejące pola w wierszach i, podobnie jak SELECT, jest ograniczany przez WHERE. Analogicznie DELETE usuwa całe wiersze, w tym przypadku wiersz z ID równym 1. Wartość ta została przyznana przez funkcję AUTO\_INCREMENT.

Bazy danych to temat rzeka, książki o tej

## Ramka 4: Bezpieczeństwo w SUSE

SUSE dodaje do MySQL narzędzie TCP wrapper tcpd. Umożliwia to użytkownikom określenie adresów IP lub nazwy hostów, które mają możliwość łączenia z bazą danych. Hosty z uprawnieniami dostępu muszą zostać dodane do /etc/hosts.allow, podczas gdy /etc/hosts.deny zawiera listę hostów pozbawionych uprawnień. W celu umożliwienia maszynie o adresie IP 192.168.1.1 uzyskania połączenia, dodajemy do /etc/hosts.allow następujący wiersz:

```
mysql!d: 192.168.1.1
```

Kolejnym krokiem jest dodanie do pliku /etc/hosts.deny wpisu uniemożliwiającego dostęp ze wszystkich innych adresów:

```
mysql!d: ALL
```

Wykonanie polecenia man 5 hosts\_access spowoduje wyświetlenie dokumentacji na ten temat. Pamiętajmy, że MySQL wymaga wprowadzenia zezwolenia na dostęp z innych hostów przy wykorzystaniu instrukcji GRANT, tak jak opisałyśmy to powyżej w sekcji „Zarządzanie użytkownikami”.

tematyce mogą z łatwością wypełnić całe biblioteki. Na szczęście pierwsze kroki nie są trudne – zarządzanie bazą danych nie wymaga ukończenia studiów informatycznych, ponieważ MySQL i dołączone go niego narzędzia są funkcjonalne i łatwe w użyciu. Uruchamianie gotowych aplikacji WWW jest łatwe – stąd już tylko mały krok do przygotowania własnej aplikacji bazodanowej. Bliższe informacje na ten temat można znaleźć w Internecie i innych artykułach Linux Magazine [6].

## Ramka 3: Kłopoty licencyjne

Fedora jest jedną z niewielu dystrybucji, które nie zawierają bieżącej wersji MySQL 4. Przyczyną tego jest fakt zmiany przez firmę MySQL licencji przy przejściu z wersji 3 do 4. Zamiast liberalnej licencji LGPL, baza danych jest obecnie rozprowadzana na licencji GPL (GNU Public License) [4]. GPL jest bardziej restrykcyjna, nie pozwala ona programom wydawanym na innych licencjach wykorzystywania jakiegokolwiek części MySQL. Programy te również muszą być wolnym oprogramowaniem w rozumieniu definicji terminu GPL, w odróżnieniu od LGPL. W ten sposób firma MySQL próbuje zmusić komercyjnych użytkowników do kupowania komercyjnych licencji na bazy danych. Zamieszczenie MySQL 4 spowodowałoby dodatkowo konflikt z licencją pakietu PHP [5].

## INFO

- [1] Strona domowa MySQL:  
<http://www.mysql.com>
- [2] KMySQLAdmin:  
<http://www.alwings-world.de/programs/kmysqladmin/>
- [3] Typy danych MySQL:  
[http://dev.mysql.com/doc/mysql/en/Column\\_types.html](http://dev.mysql.com/doc/mysql/en/Column_types.html)
- [4] Licencja GNU:  
<http://www.gnu.org/copyleft/copyleft.html>
- [5] Zagadnienia związane z licencją Fedory:  
<http://www.internetnews.com/dev-news/article.php/3358061>
- [6] Wykonywanie kopii zapasowej baz danych MySQL, Thomas Wolfer, Linux Magazine 6/2004