

W tym odcinku poznamy sposoby wykonywania typowych zadań administracyjnych, związanych z konfiguracją samego serwera, wykonywaniem i odtwarzaniem kopii zapasowych baz danych i monitorowaniem aktywności użytkowników. **Konfiguracja serwera MySQL**

Administrator serwera powinien umieć skonfigurować, przynajmniej w podstawowym zakresie, serwer, którym zarządza. Kolejne punkty przedstawiają sposób wykonania typowych zadań administracyjnych za pomocą programu MySQL Administrator, zainstalowanego w poprzednim odcinku.

MySQL Administrator jest dostępny zarówno na platformie GNU/Linux, jak i Microsoft Windows. Dzięki temu większość czynności administracyjnych może być przeprowadzana tak samo w obu systemach — polegają one na modyfikacji pliku `my.ini` (w systemach Windows) lub `my.cnf` (w systemach GNU/Linux). Ja używam wersji przeznaczonej dla systemów Windows. **Uruchamianie i zatrzymywanie serwera**

Aby zatrzymać serwer:

1. Uruchom narzędzie MySQL Administrator i połącz się z serwerem jako użytkownik root.
2. Zaznacz sekcję Service Control i kliknij przycisk Stop Service.
3. W okienku dziennika najpierw zobaczysz komunikat Trying to stop the server..., a po chwili drugi — Server was stopped. Jeśli zamiast tego wyświetli się komunikat Server could not be stopped, uruchom program MySQL Administrator z uprawnieniami lokalnego administratora.
4. Uruchomienie serwera (o ile wcześniej uruchomiłeś MySQL Administrator i połączyłeś się z nim) przebiega identycznie. Wystarczy kliknąć przycisk Start Service.

Nieemożliwe jest połączenie się z zatrzymanym serwerem. W takim wypadku najpierw trzeba uruchomić serwer, a dopiero potem program MySQL Administrator. W systemach Windows można to zrobić w sposób pokazany na listingu 2.1.

Listing 2.1. Uruchomienie i zatrzymanie serwera MySQL z wiersza polecenia systemów Windows

```
C:>net start mysql
```

Usługa MySQL jest właśnie uruchamiana.

Pomyślnie uruchomiono usługę MySQL.

```
C:>net stop mysql
```

Usługa MySQL jest właśnie zatrzymywana...

Usługa MySQL została zatrzymana pomyślnie.

Ręczne uruchomienie serwera w systemach GNU/Linux przebiega w sposób pokazany na listingu 2.2.

Listing 2.2. Uruchomienie i zatrzymanie serwera MySQL z konsoli systemów GNU/Linux

```
[root@maszynka ~]# service mysqld start
```

Uruchamianie MySQL: [OK]

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

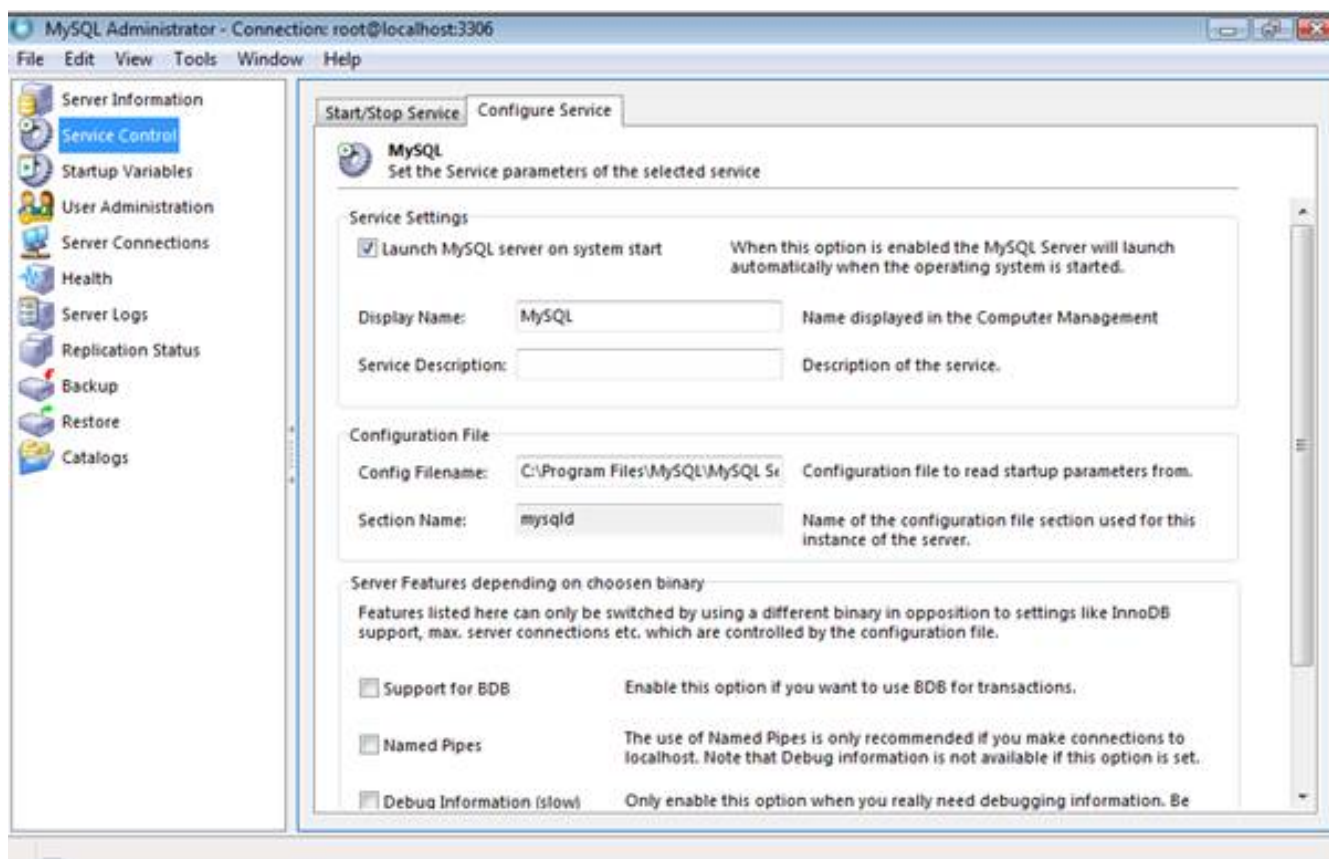
```
[root@maszynka ~]# service mysqld stop
```

Zatrzymywanie MySQL: [OK]

W systemach GNU/Linux można uruchomić serwer MySQL jako działającą w tle usługę za pomocą skryptu `safe_mysql`, znajdującego się w podfolderze `bin`. **Konfiguracja usługi serwera MySQL**

Na poziomie usługi możemy określić sposób uruchamiania, nazwę usługi i wykorzystywane przez nas cechy serwera.

1. Uruchom narzędzie MySQL Administrator i połącz się z serwerem jako użytkownik `root`.
2. Zaznacz sekcję Service Control i przejdź na zakładkę Configure Services (rysunek 2.1).



Rysunek 2.1. Okno konfiguracyjne usługi serwera MySQL

3. Jeżeli chcesz, żeby serwer MySQL był automatycznie uruchamiany przy starcie systemu. Upewnij się, że pole `Launch MySQL server automatically` jest zaznaczone, a jeżeli nie — zaznacz je.

4. Sprawdź lokalizację pliku konfiguracyjnego — to w tym pliku tekstowym zapisane są wszystkie parametry startowe serwera MySQL (listing 2.3).

Listing 2.3. Fragment przykładowego pliku konfiguracyjnego — serwer MySQL może być również konfigurowany przez bezpośrednie modyfikacje tego pliku

```
# MySQL Server Instance Configuration File
```

```
# -----
```

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

```
# Generated by the MySQL Server Instance Configuration Wizard

#

#

# Installation Instructions

# -----

#

# On Linux you can copy this file to /etc/my.cnf to set global options,

# mysql-data-dir/my.cnf to set server-specific options

# (@localstatedir@ for this installation) or to

# ~/.my.cnf to set user-specific options.

#

# On Windows you should keep this file in the installation directory

# of your server (e.g. C:\Program Files\MySQL\MySQL Server X.Y). To

# make sure the server reads the config file use the startup option

# "--defaults-file".

#

# To run the server from the command line, execute this in a

# command line shell, e.g.

# mysqld --defaults-file="C:\Program Files\MySQL\MySQL Server X.Y\my.ini"

#

# To install the server as a Windows service manually, execute this in a

# command line shell, e.g.

# mysqld --install MySQLXY --defaults-file="C:\Program Files\MySQL\MySQL Server X.Y\my.ini"

#

# And then execute this in a command line shell to start the server, e.g.

# net start MySQLXY

#

#
```

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

```
# Guildlines for editing this file

# -----

#

# In this file, you can use all long options that the program supports.

# If you want to know the options a program supports, start the program

# with the "--help" option.

#

# More detailed information about the individual options can also be

# found in the manual.

#

#

# CLIENT SECTION

# -----

#

# The following options will be read by MySQL client applications.

# Note that only client applications shipped by MySQL are guaranteed

# to read this section. If you want your own MySQL client program to

# honor these values, you need to specify it as an option during the

# MySQL client library initialization.

#

[client]

port=3306

[mysql]

default-character-set=utf8

...
```

5. Zaznacz pole Support for BDB (jeżeli to pole wyboru jest niedostępne lub ścieżka do plików binarnych została zaznaczona na czerwono, to znaczy, że serwer MySQL jest zainstalowany bez obsługi tabel BDB) — w ten sposób będziesz mógł tworzyć tabele typu IBDB (ang. BerkeleyDB) i korzystać z nich. Tabele BDB obsługują transakcje i są wyjątkowo odporne na uszkodzenia związane z nieoczekiwanym przerwaniem pracy serwera MySQL.

6. Upewnij się, że zaznaczone jest pole Named Pipes — potoki nazwane to najszybszy mechanizm komunikacji z działającym lokalnie serwerem MySQL.

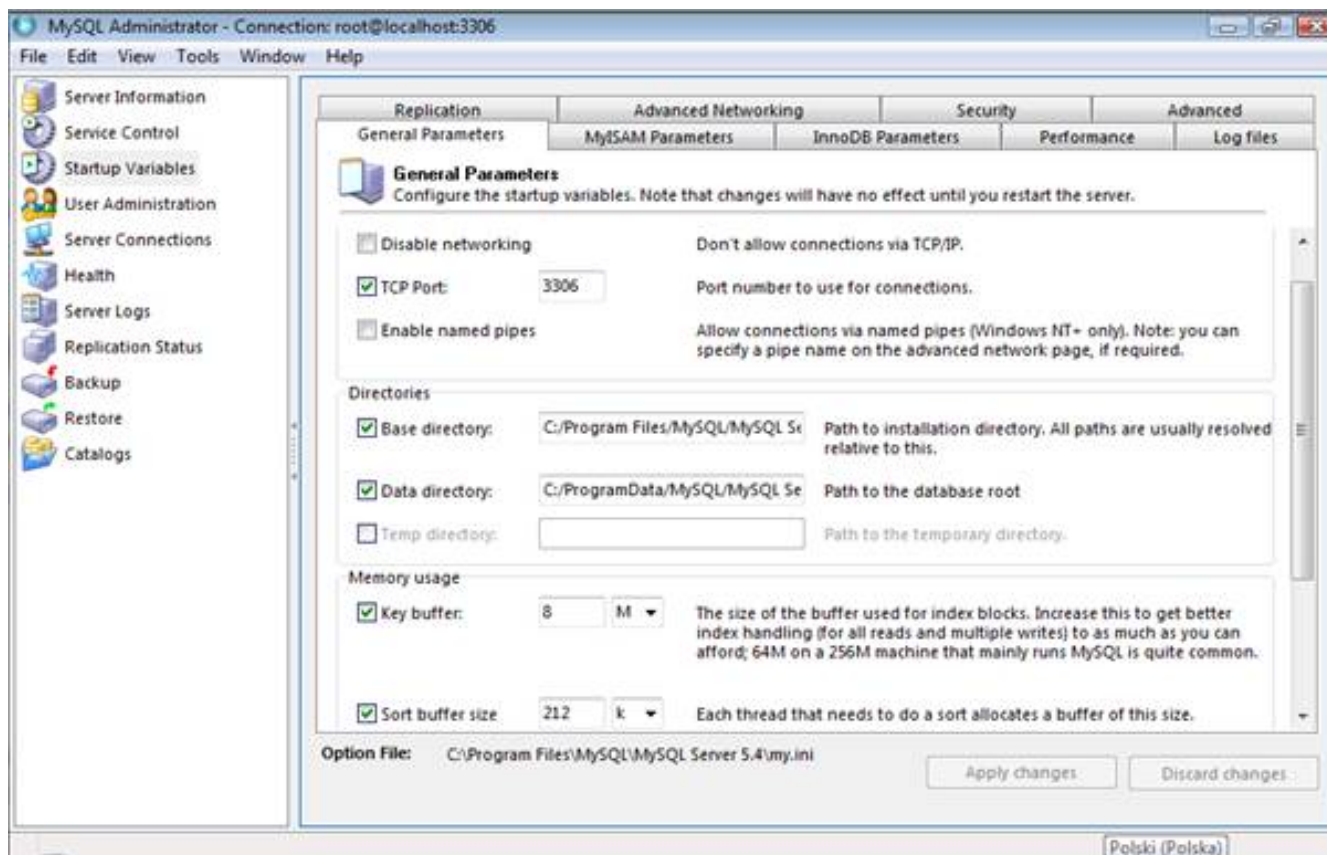
7. Kliknij przycisk Apply Changes, a następnie zatrzymaj i ponownie uruchom serwer.

Większość zmian w konfiguracji zostanie zastosowana dopiero po ponownym uruchomieniu serwera.

Najważniejsze opcje startowe

Po zaznaczeniu sekcji Startup Variables będziesz mógł wygodnie zmodyfikować pliki z opcjami startowymi serwera.

1. Na zakładce General Parameters możesz m.in.:
1. zablokować zdalny dostęp i określić port serwera,
2. przydzielić mu określoną ilość pamięci operacyjnej,
3. wskazać domyślny typ tabel — upewnij się, że jest nim InnoDB (rysunek 2.2).



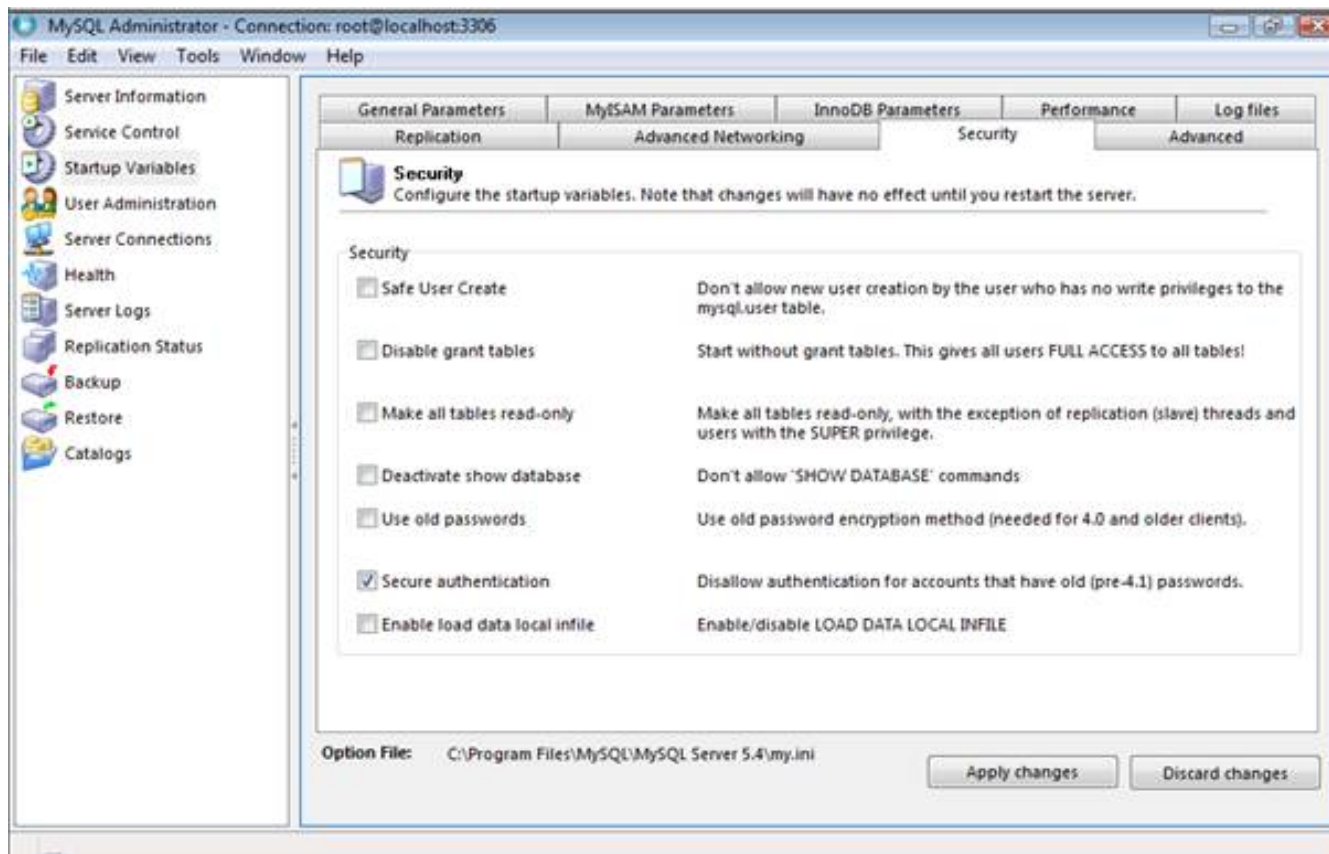
Rysunek 2.2. W przypadku lokalnych serwerów developerskich lub testowych zaznaczenie opcji Disable networking znacznie

4. Przejdź na zakładkę InnoDB Parameters i upewnij się, że zaznaczone jest pole Activate InnoDB.
5. Zakładka Performance pozwoli skonfigurować mechanizm buforowania danych. Jeżeli na przykład nie

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

chcesz buforować małych zbiorów danych, uaktywnij opcję Query cache limit i ustal rozmiar najmniejszego buforowanego wyniku zapytania.

6. Przejdź na zakładkę Security. Jeżeli nie używasz wersji serwera MySQL starszej niż piąta, zaznacz pole wyboru Secure authentication. W ten sposób uniemożliwisz zalogowanie się na podstawie kont i haseł założonych użytkownikom wcześniejszych wersji (rysunek 2.3).



Rysunek 2.3. Hasła zaszyfrowane przez starsze wersje serwera można było łatwo złamać

7. Zastosuj wprowadzone zmiany.

Typy tabel serwera MySQL

Serwer MySQL obsługuje kilka różnych typów tabel, w tym typy tabel umożliwiające transakcyjne przetwarzanie danych. Chociaż w ramach kursu będziemy wykorzystywać przede wszystkim typ InnoDB, warto wiedzieć, czym charakteryzują się wszystkie typy tabel. Żeby wyświetlić podstawowe informacje na temat obsługiwanych przez Twoją kopię serwera MySQL typów tabel, wykonaj poniższą instrukcję (listing 2.4).

Listing 2.4. W tym przypadku instrukcję wykonano w tekstowym monitorze MySQL, a opcja G została użyta, żeby poprawić czytelność wyniku

```
mysql> SHOW ENGINESG;
```

```
***** 1. row *****
```

```
Engine: MEMORY
```

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

Support: YES

Comment: Hash based, stored in memory, useful for temporary tables

Transactions: NO

XA: NO

Savepoints: NO

***** 2. row *****

Engine: FEDERATED

Support: NO

Comment: Federated MySQL storage engine

Transactions: NULL

XA: NULL

Savepoints: NULL

***** 3. row *****

Engine: MyISAM

Support: YES

Comment: Default engine as of MySQL 3.23 with great performance

Transactions: NO

XA: NO

Savepoints: NO

***** 4. row *****

Engine: BLACKHOLE

Support: YES

Comment: /dev/null storage engine (anything you write to it disappears)

Transactions: NO

XA: NO

Savepoints: NO

***** 5. row *****

Engine: MRG_MYISAM

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

Support: YES

Comment: Collection of identical MyISAM tables

Transactions: NO

XA: NO

Savepoints: NO

***** 6. row *****

Engine: CSV

Support: YES

Comment: CSV storage engine

Transactions: NO

XA: NO

Savepoints: NO

***** 7. row *****

Engine: ARCHIVE

Support: YES

Comment: Archive storage engine

Transactions: NO

XA: NO

Savepoints: NO

***** 8. row *****

Engine: InnoDB

Support: DEFAULT

Comment: Supports transactions, row-level locking, and foreign keys

Transactions: YES

XA: YES

Savepoints: YES

8 rows in set (0.00 sec) **Tabele typu MEMORY**

Zarządzamy serwerem baz danych

Dodał Administrator

poniedziałek, 19 kwiecień 2010 18:41

Każda tabela tego typu przechowywana jest w postaci pojedynczego pliku z rozszerzeniem .frm. Jak sugeruje nazwa (poprzednio tabele tego typu nazywane były tabelami HASH), dane tego typu tabel przechowywane są w pamięci operacyjnej i są tracone podczas wyłączania serwera (plik .frm zawiera jedynie definicję tabeli).

Domyślnie w tabelach typu MEMORY wykorzystywane są indeksy mieszane (ang. Hash), dzięki czemu modyfikacje przechowywanych w nich danych są bardzo szybkie. Tego typu tabele powinny być używane jako tabele tymczasowe. **Tabele typu FEDERATED**

Tego typu tabele nie są domyślnie obsługiwane przez serwery MySQL w wersji 5.1 lub wyższej. Żeby włączyć ich obsługę, należy uruchomić serwer z parametrem --federated. W przeciwieństwie do tabel innych typów tabele FEDERATED są bezpośrednio dostępne ze zdalnych serwerów MySQL — zapytania adresowane do tego typu tabel są automatycznie wykonywane przez docelowy serwer MySQL. **Tabele typu MyISAM**

Każda tabela tego typu przechowywana jest w postaci trzech plików:

1. plik .frm zawiera definicję tabeli,
2. plik .MYD zawiera dane (wiersze) tabeli,
3. plik .MYI zawiera indeksy tabeli.

Tabele typu MyISAM pozwalają na szybkie odczytywanie danych, ale nie umożliwiają ich transakcyjnego przetwarzania (w dodatku modyfikacje danych są dość wolne). Z tego powodu powinny być wykorzystywane w do przechowywania rzadko zmienianych (albo zmienianych przez pojedynczych użytkowników) danych, np. zawartości stron WWW lub słowników. **Tabele typu BLACKHOLE**

Tego typu tabele nie przechowują żadnych danych — tak jak w przypadku tabel typu MEMORY, plik .frm przechowuje jedynie definicję tabeli typu BLACKHOLE, natomiast w przeciwieństwie do tabel innego typu zapisywane w nich wiersze są automatycznie usuwane. Dzięki temu, że informacje o operacji są zapisywane w dzienniku zdarzeń serwera MySQL, tabele typu BLACKHOLE są przydatne podczas diagnozowania i testowania baz danych. **Tabele typu MRG_MyISAM**

Ta specjalna odmiana tabel typu MyISAM może być współdzielona pomiędzy różnymi serwerami MySQL. **Tabele typu CSV**

Każda tabela tego typu przechowywana jest w postaci trzech plików:

1. plik .frm zawiera definicję tabeli,
2. plik .csv zawiera dane (wiersze) tabeli w postaci danych tekstowych rozdzielanych przecinkiem,
3. plik .csm zawiera metadane.

Tabele typu CSV są używane przede wszystkim do wymieniaania (importowania i eksportowania) danych pomiędzy serwerem MySQL a innymi serwerami lub programami. **Tabele typu ARCHIVE**

Każda tabela tego typu przechowywana jest w postaci trzech plików:

1. plik .frm zawiera definicję tabeli,
2. plik .ARZ zawiera skompresowane dane (wiersze) tabeli,
3. tymczasowy plik .ARN jest tworzony podczas optymalizacji tabeli.

Tabele typu ARCHIVE umożliwiają wyłącznie odczytywanie (instrukcją SELECT) oraz wstawianie (instrukcją

Zarządzamy serwerem baz danych

Dodał Administrator

poniedziałek, 19 kwiecień 2010 18:41

INSERT) danych — zmienianie czy usuwanie danych jest niedopuszczalne. Z tego powodu używane są głównie do przechowywania danych historycznych lub diagnostycznych. **Tabele typu InnoDB**

Tego typu tabele są domyślne na serwerach MySQL działających w środowisku systemu Windows i coraz częściej zastępują tabele typu MyISAM (w tabelach tego typu przechowywane są między dane serwisu <http://slashdot.org/>). Tabele InnoDB obsługują transakcyjne przetwarzanie danych, gwarantując atomowość, spójność, niezależność i trwałość poszczególnych transakcji (serwer MySQL automatycznie blokuje odczytywane i modyfikowane wiersze). Tego typu tabele pozwalają również definiować i sprawdzać ograniczenia klucza obcego.

Wiersze tabel typu InnoDB przechowywane są w postaci drzew zrównoważonych (a więc tworzą indeks zgrupowany) — między innymi dzięki temu ten typ tabel pozwala szybko odczytywać i modyfikować dane.

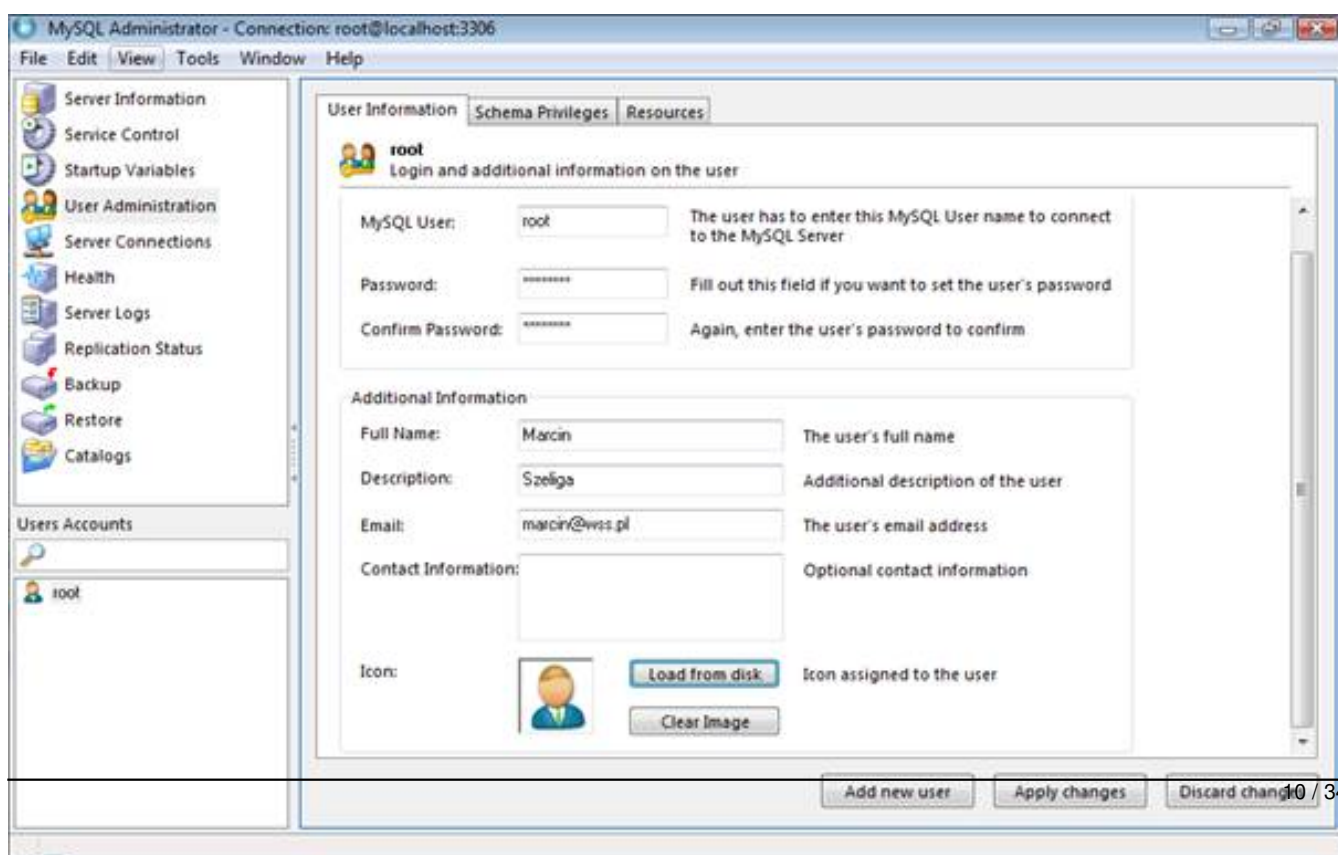
Zabezpieczanie serwera MySQL

Model bezpieczeństwa MySQL-a jest następujący: żeby połączyć się z serwerem, użytkownik musi potwierdzić (za pomocą loginu i hasła) swoją tożsamość. Proces ten nazywa się uwierzytelnianiem. Następnie, przed wykonaniem przez uwierzytelnionego użytkownika jakiegokolwiek operacji (odczytania lub zmodyfikowania danych, utworzenia tabeli itd.), serwer sprawdza, czy ma on wystarczające uprawnienia, i jeżeli ma — zezwala na wykonanie operacji. Brak wymaganych uprawnień spowoduje zgłoszenie błędu i przerwanie operacji. Ten proces nazywa się autoryzacją.

Uprawnienia mogą być nadawane użytkownikom na trzech poziomach: całego serwera, konkretnego schematu (wybranej bazy danych) i konkretnego obiektu bazy danych (na przykład tabeli). Domyślnie uprawnienia są dziedziczone, czyli użytkownik, który ma nadane jakieś uprawnienia na poziomie serwera, automatycznie może wykonać daną operację we wszystkich bazach danych i w odniesieniu do każdego obiektu bazodanowego.

Sprawdzamy uprawnienia użytkownika root

1. Uruchom program MySQL Administrator i połącz się z serwerem jako użytkownik root.
2. Zaznacz sekcję User Administration i wybierz widoczne w lewym dolnym okienku konto użytkownika root (po zainstalowaniu serwera będzie to jedyne konto).
3. Na zakładce User Information zobaczysz podstawowe informacje o użytkowniku (rysunek 2.4).

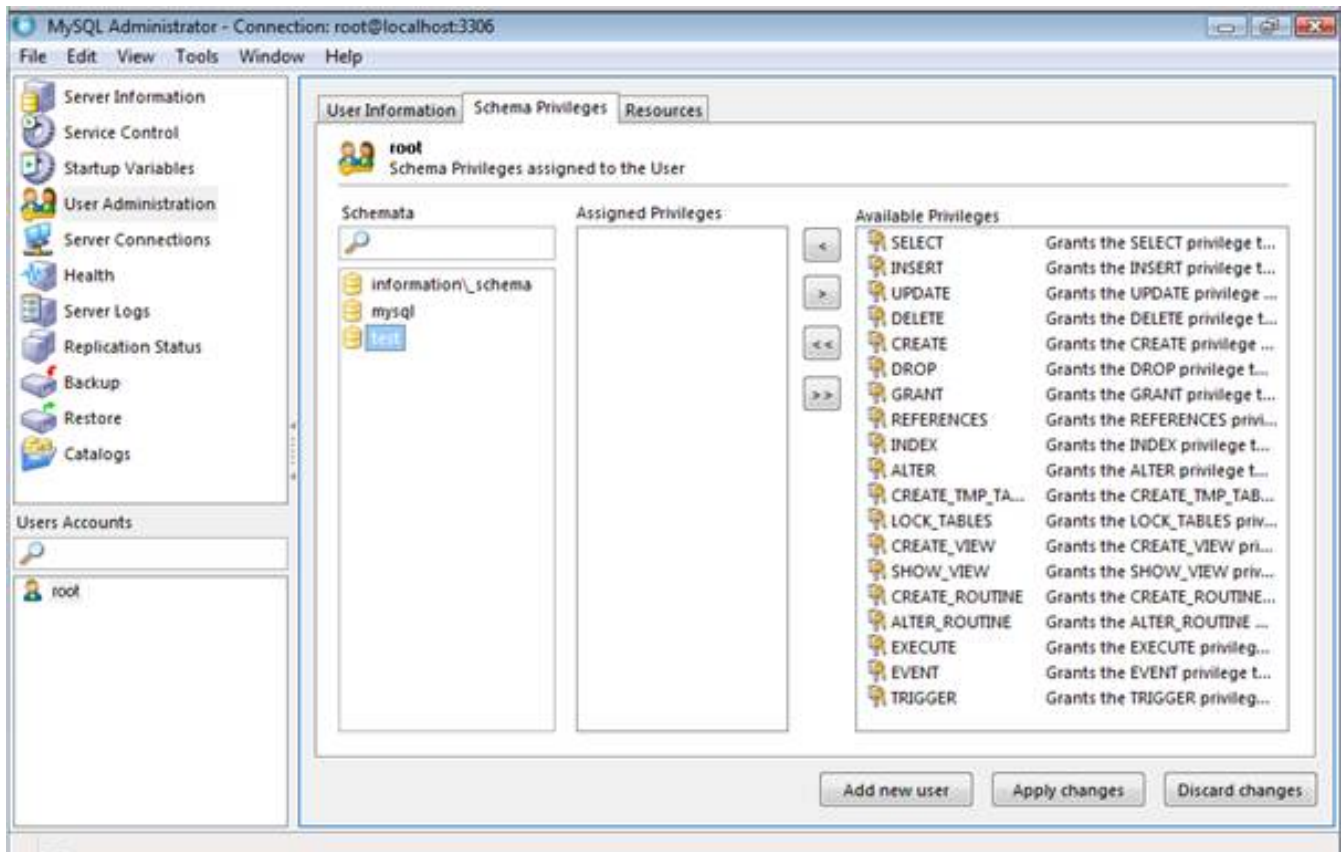


Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

Rysunek 2.4. We wcześniejszych wersjach MySQL-a nazwa użytkownika składała się z dwóch części: nazwy użytkownika i

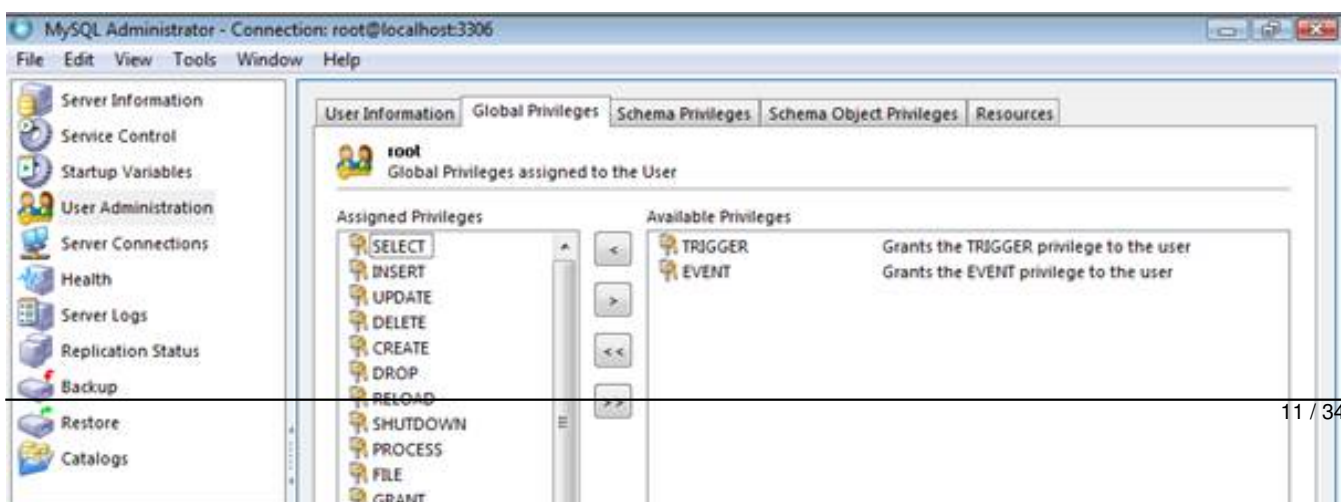
4. Zmień hasło użytkownika root i kliknij przycisk Apply changes.
5. Przejdź na zakładkę Schema Privileges i wybierz utworzoną w poprzednim odcinku bazę test. Zwróć uwagę, że root nie ma nadanych w niej żadnych uprawnień (zresztą tak samo jak w pozostałych bazach) (rysunek 2.5).



Rysunek 2.5. Skoro bazę test utworzył użytkownik root, który w dodatku jest administratorem całego serwera, to dlaczego nie

6. Skoro root nie ma jawnie nadanych uprawnień do bazy, a mimo to może wykonać w niej dowolne operacje, to musiał te uprawnienia odziedziczyć. Żeby się o tym przekonać, z menu Tools wybierz Options i zaznacz pola wyboru Show Global Privileges i Show Schema Object Privileges.

7. Zaakceptuj zmiany i zamknij okienko opcji. W głównym oknie programu pojawiły się dwie nowe zakładki: Global Privileges i Schema Object Privileges. Zaznacz tę pierwszą (rysunek 2.6).



Rysunek 2.6. Jak należało się spodziewać, użytkownik root (administrator serwera MySQL) ma nadane wszystkie uprawnienia

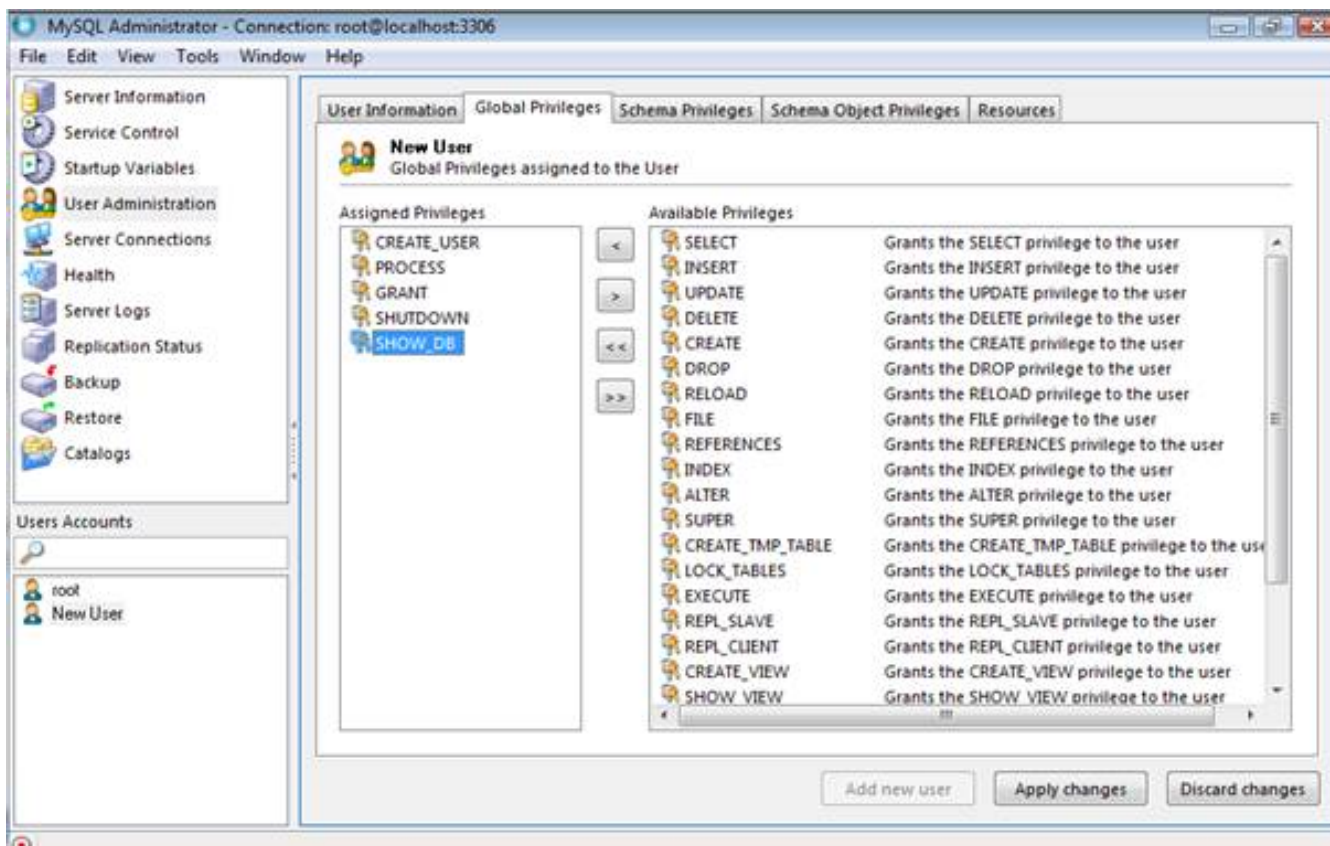
Zakładamy konto użytkownika

Konto root ma niczym nieograniczone uprawnienia i nie powinno być na co dzień używane nawet przez administratora, a już w żadnym wypadku przez program kliencki (np. stronę WWW). Korzystając z konta root, drastycznie obniżamy poziom bezpieczeństwa serwera — właściwie wyłączamy cały rozbudowany mechanizm autoryzacji (sprawdzania, czy dana osoba ma uprawnienia do wykonania żądanej operacji). Skoro łączymy się jako „wszechmogący” administrator, to na pewno będziemy mogli wykonać każdą, w tym niebezpieczną, błędną czy przypadkową instrukcję.

Jedną z podstawowych zasad bezpieczeństwa komputerowego jest zasada minimalnych uprawnień — użytkownik powinien mieć nadane tylko takie uprawnienia, jakie są mu potrzebne do wykonania jego pracy, i żadnych innych.

Aby założyć konto administratora testowej bazy danych:

1. Kliknij prawym przyciskiem myszy wolny obszar okienka User Accounts i z menu kontekstowego wybierz Add new User.
2. Podaj nazwę użytkownika i hasło i przejdź na zakładkę Global Privileges.
3. Nadaj tworzonemu użytkownikowi następujące uprawnienia administracyjne: CREATE USER, PROCESS, SHOW DB, SHUTDOWN i GRANT (rysunek 2.7).



Rysunek 2.7. Ten użytkownik też będzie w stanie na bieżąco zarządzać serwerem, a skutki ewentualnego zdobycia przez niego

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

4. Przejdź na zakładkę Schema Privileges, zaznacz bazę test i nadaj użytkownikowi wszystkie uprawnienia do niej.
5. Kliknij przycisk Apply Changes.
6. W wierszu polecenia połącz się z serwerem jako nowo utworzony użytkownik i odczytaj dane o jednym z klientów (listing 2.5).

Listing 2.5. Udane uwierzytelnienie i autoryzacja nowego użytkownika

```
C:\Users\Marcin.VistaPC>mysql -u szel -p
```

```
Enter password: *****
```

```
Welcome to the MySQL monitor. Commands end with ; or g.
```

```
Your MySQL connection id is 10
```

```
Server version: 5.4.1-beta-community MySQL Community Server (GPL)
```

```
Type 'help;' or 'h' for help. Type 'c' to clear the buffer.
```

```
mysql> USE test;
```

```
Database changed
```

```
mysql> SELECT * FROM customer WHERE customer_id=2;
```

```
+-----+-----+-----+-----+-----+-----+-----+
-----+
| customer_id | title | fname | lname | addressline | town | zipcode | phone |
+-----+-----+-----+-----+-----+-----+-----+
|          2 | Mr   | Andrew | Stones | 52 The Willows | Lowtown | LT5 7RA | 876 3527 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.10 sec)

1 row in set (0.00 sec)
```


Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

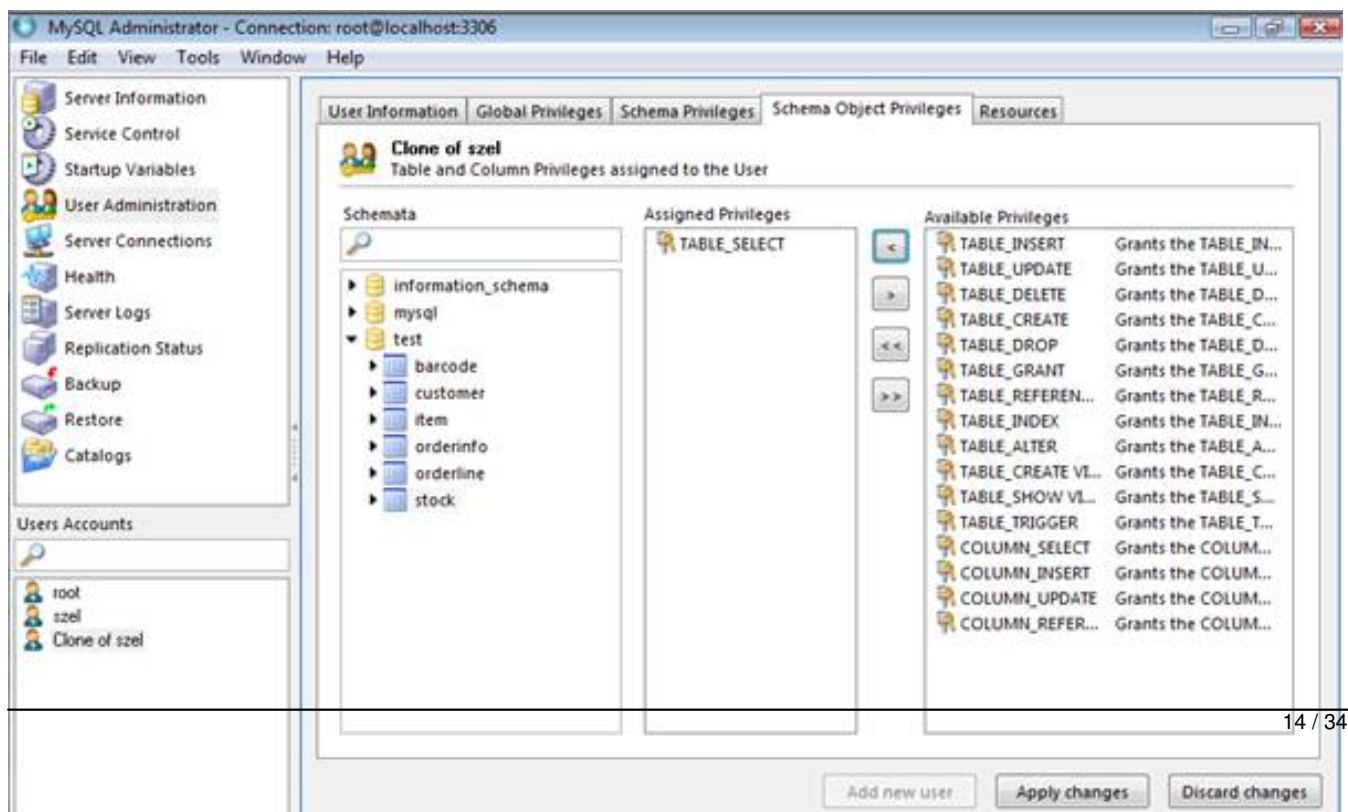
mysql> quit

Bye

Kopiujemy i usuwamy konto użytkownika

Gdyby za każdym razem podczas dodawania nowego użytkownika trzeba było na nowo nadawać mu odpowiednie uprawnienia, administrator spędzałby większość czasu, robiąc tylko to. Najprostszym (i w przypadku zarządzania niewielką liczbą kont użytkowników wystarczająco funkcjonalnym) rozwiązaniem jest kopiowanie kont.

1. Wybierz sekcję User Administration i kliknij prawym przyciskiem myszy utworzone w poprzednim punkcie konto użytkownika.
2. Z menu kontekstowego wybierz opcję Clone user.
3. Podaj nazwę i hasło nowego użytkownika.
4. Sprawdź (zakładka Global Privileges), czy nowo utworzony użytkownik ma nadane takie same uprawnienia jak użytkownik wzorcowy.
5. Przejdź na zakładkę Schema Privileges i odbierz użytkownikowi prawo wykonywania instrukcji SELECT (a więc odczytywania danych).
6. Na zakładce Schema Object Privileges rozwiń bazę test, zaznacz tabelę Stock i nadaj użytkownikowi uprawnienie TABLE SELECT.
7. Zastosuj wprowadzone zmiany przyciskiem Apply changes (rysunek 2.8).



Rysunek 2.8. Nowy użytkownik będzie mógł odczytywać dane tylko z jednej tabeli bazy testowej

8. W wierszu polecenia połącz się z serwerem jako nowo utworzony użytkownik i wykonaj poniższe instrukcje (listing 2.6).

Listing 2.6. Sprawdzamy uprawnienia użytkownika na podstawie istniejącego konta

```
C:\Users\Marcin.VistaPC>mysql -u test -p
```

```
Enter password: *
```

```
Welcome to the MySQL monitor.  Commands end with ; or g.
```

```
Your MySQL connection id is 11
```

```
Server version: 5.4.1-beta-community MySQL Community Server (GPL)
```

```
Type 'help;' or 'h' for help. Type 'c' to clear the buffer.
```

```
mysql> USE test;
```

```
Database changed
```

```
mysql> SELECT * FROM customer;
```

```
ERROR 1142 (42000): SELECT command denied to user 'test'@'localhost' for table 'customer'
```

```
mysql> SELECT * FROM stock;
```

```
+-----+-----+  
| item_id | quantity |  
+-----+-----+  
| 1 | 12 |  
| 2 | 2 |  
| 4 | 8 |  
| 5 | 3 |  
| 7 | 8 |
```

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

```
| 8 | 18 |
```

```
| 10 | 1 |
```

```
+-----+-----+
```

7 rows in set (0.00 sec)

```
mysql> quit
```

Bye

9. W konsoli MySQL Administrator kliknij prawym przyciskiem myszy ostatnio dodane konto użytkownika i z menu kontekstowego wybierz Delete User.

10. Potwierdź swoją decyzję i raz jeszcze spróbuj połączyć się z serwerem jako właśnie usunięty użytkownik (listing 2.7).

Listing 2.7. Nieudane uwierzytelnienie — na serwerze nie ma takiego konta użytkownika

```
C:\Users\Marcin.VistaPC>mysql -u test -p
```

```
Enter password: *
```

```
ERROR 1045 (28000): Access denied for user 'test'@'localhost' (using password: Y
```

```
ES)
```

Kopie zapasowe baz danych

Zabezpieczenie przed utratą danych jest jednym z podstawowych obowiązków administratora baz danych. Niezależnie od odporności na awarie używanego sprzętu komputerowego (redundantne macierze dyskowe, zasilacze awaryjne itp.), jedynie posiadanie aktualnej kopii zapasowej gwarantuje odzyskanie danych utraconych w wyniku:

1. przypadkowego lub złośliwego wykonania instrukcji DELETE,
2. przypadkowego lub złośliwego wykonania instrukcji UPDATE z brakującym albo błędnie sformułowanym warunkiem logicznym w klauzuli WHERE,
3. przypadkowego lub złośliwego zastąpienia istniejących danych — bądź całej bazy danych importowanymi danymi albo odtwarzaną kopią zapasową,
4. kradzieży dysku lub serwera,
5. działania wirusów.

Ponieważ bazy danych z reguły zawierają niezbędne dla funkcjonowania firmy informacje, każdy administrator powinien przygotować i przetestować strategię wykonywania i odtwarzania kopii zapasowych. W ramach strategii należy określić m.in.:

1. Dopuszczalny koszt wykonywania kopii zapasowych, z uwzględnieniem czasu potrzebnego na wykonanie kopii i kosztu nośników niezbędnych do przechowywania danych archiwalnych.
2. Dopuszczalny czas potrzebny do przywrócenia sprawności systemu po awarii.
3. Czy dopuszczalna jest utrata danych (np. danych z ostatniego dnia roboczego), a jeżeli tak, to ilu?

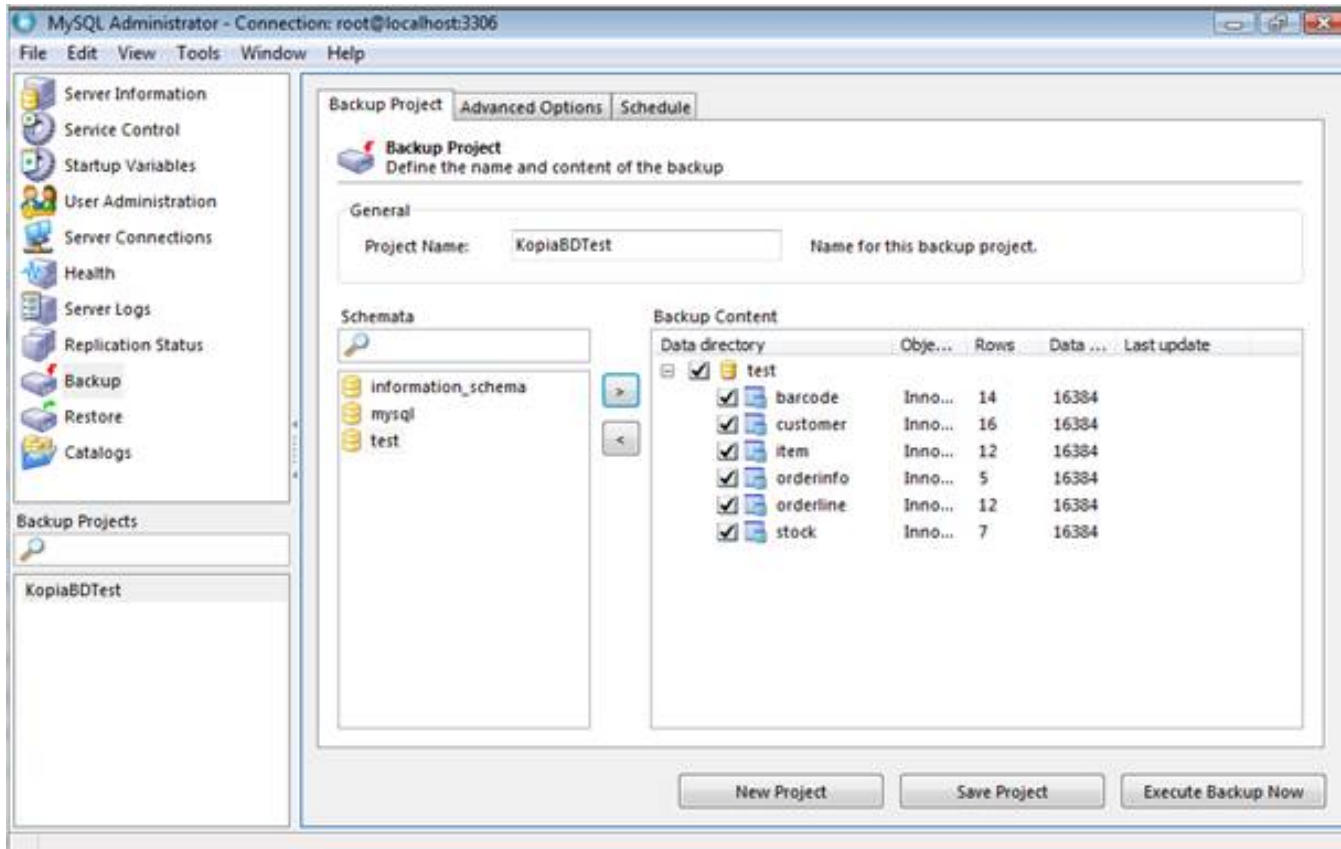
Zarządzamy serwerem baz danych

Dodał Administrator

poniedziałek, 19 kwiecień 2010 18:41

Innym sposobem zabezpieczenia się przed skutkami ewentualnej awarii czy utraty danych jest replikowanie (powielanie) danych pomiędzy kilkoma serwerami bazodanowymi. **Tworzymy kopię zapasową bazy**

1. Uruchom narzędzie MySQL Administrator i połącz się z serwerem jako użytkownik root.
2. Przejdź do sekcji Backup kliknij przycisk New project.
3. Nazwij projekt KopiaBDTest i zaznacz kopiowaną bazę danych test (rysunek 2.9).



Rysunek 2.9. W większości przypadków najlepszą strategią jest wykonywanie kopii całej bazy danych

4. Przejdź na zakładkę Advanced Options. W pierwszej sekcji możesz wybrać sposób tworzenia kopii zapasowych. W przypadku kopiowania tabel typu InnoDB należy wybrać opcję InnoDB Online Backup — w ten sposób cała kopia będzie wykonana jako jedna, duża transakcja, dzięki czemu będzie zawierać spójny obraz danych z momentu rozpoczęcia jej wykonywania. Niestety, w czasie tworzenia kopii dostęp do bazy będzie ograniczony.

5. W dolnej części okienka możemy określić format i zawartość pliku kopii zapasowej. Kopie baz zapisywane są w postaci plików tekstowych zawierających potrzebne do ich odtworzenia instrukcje języka SQL.

6. Kliknij Execute Backup Now, podaj lokalizację pliku kopii zapasowej i kliknij przycisk Zapisz.

7. Otwórz utworzony plik w edytorze tekstu. Powinieneś zobaczyć dane pokazane na listingu 2.8.

Listing 2.8. Analizując plik kopii zapasowej, dowiesz się, jak efektywnie odtwarzać tabele i zapisane w nich informacje. Zwróć uwagę na przykład na wyłączanie na czas wstawiania danych zawężeń

```
-- MySQL Administrator dump 1.4
```

```
--
```

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

-- Server version 5.4.1-beta-community

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

/*!40101 SET NAMES utf8 */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;

/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;

/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

--

-- Create schema test

--

CREATE DATABASE IF NOT EXISTS test;

USE test;

--

-- Definition of table `barcode`

--

DROP TABLE IF EXISTS `barcode`;

CREATE TABLE `barcode` (

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

```
`barcode_ean` char(13) NOT NULL,  
`item_id` int(11) NOT NULL,  
PRIMARY KEY (`barcode_ean`),  
KEY `item_id` (`item_id`),  
CONSTRAINT `barcode_item_id_fk` FOREIGN KEY (`item_id`) REFERENCES `item` (`item_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

--

-- Dumping data for table `barcode`

--

```
/*!40000 ALTER TABLE `barcode` DISABLE KEYS */;
```

```
INSERT INTO `barcode` (`barcode_ean`,`item_id`) VALUES
```

```
('6241527836173',1),
```

```
('6241574635234',2),
```

```
('6241527746363',3),
```

```
('6264537836173',3),
```

```
('7465743843764',4),
```

```
('3453458677628',5),
```

```
('6434564564544',6),
```

```
('8476736836876',7),
```

```
('6241234586487',8),
```

```
('9473625532534',8),
```

```
('9473627464543',8),
```

```
('4587263646878',9),
```

```
('2239872376872',11),
```

```
('9879879837489',11);
```

```
/*!40000 ALTER TABLE `barcode` ENABLE KEYS */;
```

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

```
--  
  
-- Definition of table `customer`  
  
--  
  
DROP TABLE IF EXISTS `customer`;  
  
CREATE TABLE `customer` (  
  `customer_id` int(11) NOT NULL AUTO_INCREMENT,  
  `title` char(4) DEFAULT NULL,  
  `fname` varchar(32) DEFAULT NULL,  
  `lname` varchar(32) NOT NULL,  
  `addressline` varchar(64) DEFAULT NULL,  
  `town` varchar(32) DEFAULT NULL,  
  `zipcode` char(10) NOT NULL,  
  `phone` varchar(16) DEFAULT NULL,  
  PRIMARY KEY (`customer_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8;  
  
--  
  
-- Dumping data for table `customer`  
  
--  
  
/*!40000 ALTER TABLE `customer` DISABLE KEYS */;  
  
INSERT INTO `customer` (`customer_id`,`title`,`fname`,`lname`,`addressline`,`town`,`zipcode`,`phone`) VALUES  
(1,'Miss','Jenny','Stones','27 Rowan Avenue','Hightown','NT2 1AQ','023 9876'),  
(2,'Mr','Andrew','Stones','52 The Willows','Lowtown','LT5 7RA','876 3527'),  
(3,'Miss','Alex','Matthew','4 The Street','Niketown','NT2 2TX','010 4567'),
```

Zarządzamy serwerem baz danych

Dodał Administrator

poniedziałek, 19 kwiecień 2010 18:41

```
(4,'Mr','Adrian','Matthew','The Barn','Yuleville','YV67 2WR','487 3871'),
(5,'Mr','Simon','Cozens','7 Shady Lane','Oahenham','OA3 6QW','514 5926'),
(6,'Mr','Neil','Matthew','5 Pasture Lane','Nicetown','NT3 7RT','267 1232'),
(7,'Mr','Richard','Stones','34 Holly Way','Bingham','BG4 2WE','342 5982'),
(8,'Mrs','Anna','Stones','34 Holly Way','Bingham','BG4 2WE','342 5982'),
(9,'Mrs','Christine','Hickman','36 Queen Street','Histon','HT3 5EM','342 5432'),
(10,'Mr','Mike','Howard','86 Dysart Street','Tibsville','TB3 7FG','505 5482'),
(11,'Mr','Dave','Jones','54 Vale Rise','Bingham','BG3 8GD','342 8264'),
(12,'Mr','Richard','Neill','42 Thached way','Winersby','WB3 6GQ','505 6482'),
(13,'Mrs','Laura','Hendy','73 Margeritta Way','Oxbridge','OX2 3HX','821 2335'),
(14,'Mr','Bill','Neill','2 Beamer Street','Welltown','WT3 8GM','435 1234'),
(15,'Mr','David','Hudson','4 The Square','Milltown','MT2 6RT','961 4526'),
(16,NULL,NULL,'Wolski',NULL,NULL,'MX2 6RV',NULL);
```

```
/*!40000 ALTER TABLE `customer` ENABLE KEYS */;
```

```
--
```

```
-- Definition of table `item`
```

```
--
```

```
DROP TABLE IF EXISTS `item`;
```

```
CREATE TABLE `item` (
```

```
  `item_id` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `description` varchar(64) NOT NULL,
```

```
  `cost_price` decimal(7,2) DEFAULT NULL,
```

```
  `sell_price` decimal(7,2) DEFAULT NULL,
```

```
  PRIMARY KEY (`item_id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8;
```

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

--

-- Dumping data for table `item`

--

```
/*!40000 ALTER TABLE `item` DISABLE KEYS */;
```

```
INSERT INTO `item` (`item_id`,`description`,`cost_price`,`sell_price`) VALUES
```

```
(1,'Wood Puzzle','15.23','21.95'),
```

```
(2,'Rubik Cube','7.45','11.49'),
```

```
(3,'Linux CD','1.99','2.49'),
```

```
(4,'Tissues','2.11','3.99'),
```

```
(5,'Picture Frame','7.54','9.95'),
```

```
(6,'Fan Small','9.23','15.75'),
```

```
(7,'Fan Large','13.36','19.95'),
```

```
(8,'Toothbrush','0.75','1.45'),
```

```
(9,'Roman Coin','2.34','2.45'),
```

```
(10,'Carrier Bag','0.01','0.00'),
```

```
(11,'Speakers','19.73','25.32'),
```

```
(12,'SQL Server 2005',NULL,NULL);
```

```
/*!40000 ALTER TABLE `item` ENABLE KEYS */;
```

--

-- Definition of table `orderinfo`

--

```
DROP TABLE IF EXISTS `orderinfo`;
```

```
CREATE TABLE `orderinfo` (
```

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

```
`orderinfo_id` int(11) NOT NULL AUTO_INCREMENT,  
  
`customer_id` int(11) NOT NULL,  
  
`date_placed` date NOT NULL,  
  
`date_shipped` date DEFAULT NULL,  
  
`shipping` decimal(7,2) DEFAULT NULL,  
  
PRIMARY KEY (`orderinfo_id`),  
  
KEY `customer_id` (`customer_id`),  
  
CONSTRAINT `orderinfo_customer_id_fk` FOREIGN KEY (`customer_id`) REFERENCES `customer`  
(`customer_id`)  
  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;  
  
--  
-- Dumping data for table `orderinfo`  
--  
  
/*!40000 ALTER TABLE `orderinfo` DISABLE KEYS */;  
  
INSERT INTO `orderinfo` (`orderinfo_id`,`customer_id`,`date_placed`,`date_shipped`,`shipping`) VALUES  
  
(1,3,'2000-03-13','2000-03-17','2.99'),  
  
(2,8,'2000-06-23','2000-06-23','0.00'),  
  
(3,15,'2000-09-02','2000-09-12','3.99'),  
  
(4,13,'2000-09-03','2000-09-10','2.99'),  
  
(5,8,'2000-07-21','2000-07-24','0.00');  
  
/*!40000 ALTER TABLE `orderinfo` ENABLE KEYS */;  
  
--  
-- Definition of table `orderline`  
--
```

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

```
DROP TABLE IF EXISTS `orderline`;

CREATE TABLE `orderline` (
  `orderinfo_id` int(11) NOT NULL,
  `item_id` int(11) NOT NULL,
  `quantity` int(11) NOT NULL,
  PRIMARY KEY (`orderinfo_id`,`item_id`),
  KEY `item_id` (`item_id`),
  CONSTRAINT `orderline_item_id_fk` FOREIGN KEY (`item_id`) REFERENCES `item` (`item_id`),
  CONSTRAINT `orderline_orderinfo_id_fk` FOREIGN KEY (`orderinfo_id`) REFERENCES `orderinfo`
(`orderinfo_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Dumping data for table `orderline`

--

/*!40000 ALTER TABLE `orderline` DISABLE KEYS */;

INSERT INTO `orderline` (`orderinfo_id`,`item_id`,`quantity`) VALUES
(1,4,1),
(1,7,1),
(1,9,1),
(2,1,1),
(2,4,2),
(2,7,2),
(2,10,1),
(3,1,1),
(3,2,1),
```


Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

```
(4,5,2),  
  
(5,1,1),  
  
(5,3,1);  
  
/*!40000 ALTER TABLE `orderline` ENABLE KEYS */;  
  
  
--  
  
-- Definition of table `stock`  
  
--  
  
DROP TABLE IF EXISTS `stock`;  
  
CREATE TABLE `stock` (  
  `item_id` int(11) NOT NULL AUTO_INCREMENT,  
  `quantity` int(11) NOT NULL,  
  PRIMARY KEY (`item_id`),  
  CONSTRAINT `stock_item_id_fk` FOREIGN KEY (`item_id`) REFERENCES `item` (`item_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8;  
  
--  
  
-- Dumping data for table `stock`  
  
--  
  
/*!40000 ALTER TABLE `stock` DISABLE KEYS */;  
  
INSERT INTO `stock` (`item_id`,`quantity`) VALUES  
  
(1,12),  
  
(2,2),  
  
(4,8),  
  
(5,3),
```

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

(7,8),

(8,18),

(10,1);

```
/*!40000 ALTER TABLE `stock` ENABLE KEYS */;
```

```
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
```

```
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
```

```
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

```
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
```

```
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

Odtwarzamy bazę po awarii

Przed wykonaniem tego ćwiczenia upewnij się, że masz aktualną kopię zapasową bazy test.

1. Uruchom narzędzie MySQL Query Browser i połącz się z serwerem jako root.
2. Kolejno wykonaj instrukcje z listingu 2.9. (W języku SQL koniec instrukcji oznacza średnik. W MySQL Query Browser wystarczy, że klikniesz instrukcję lewym przyciskiem myszy — zostanie ona podświetlona. Kliknięcie przycisku Execute wykona tylko zaznaczoną instrukcję).

Listing 2.9. Przypadkowo usuwamy informacje magazynowe i zamiast zmienić cenę jednego elementu, pomyłkowo zmieniamy ceny wszystkich. Dwie ostatnie instrukcje pokazują przeprowadzone zmiany

```
DELETE FROM test.stock;
```

```
UPDATE test.item i
```

```
SET i.sell_price = 10;
```

```
SELECT * FROM test.stock s;
```

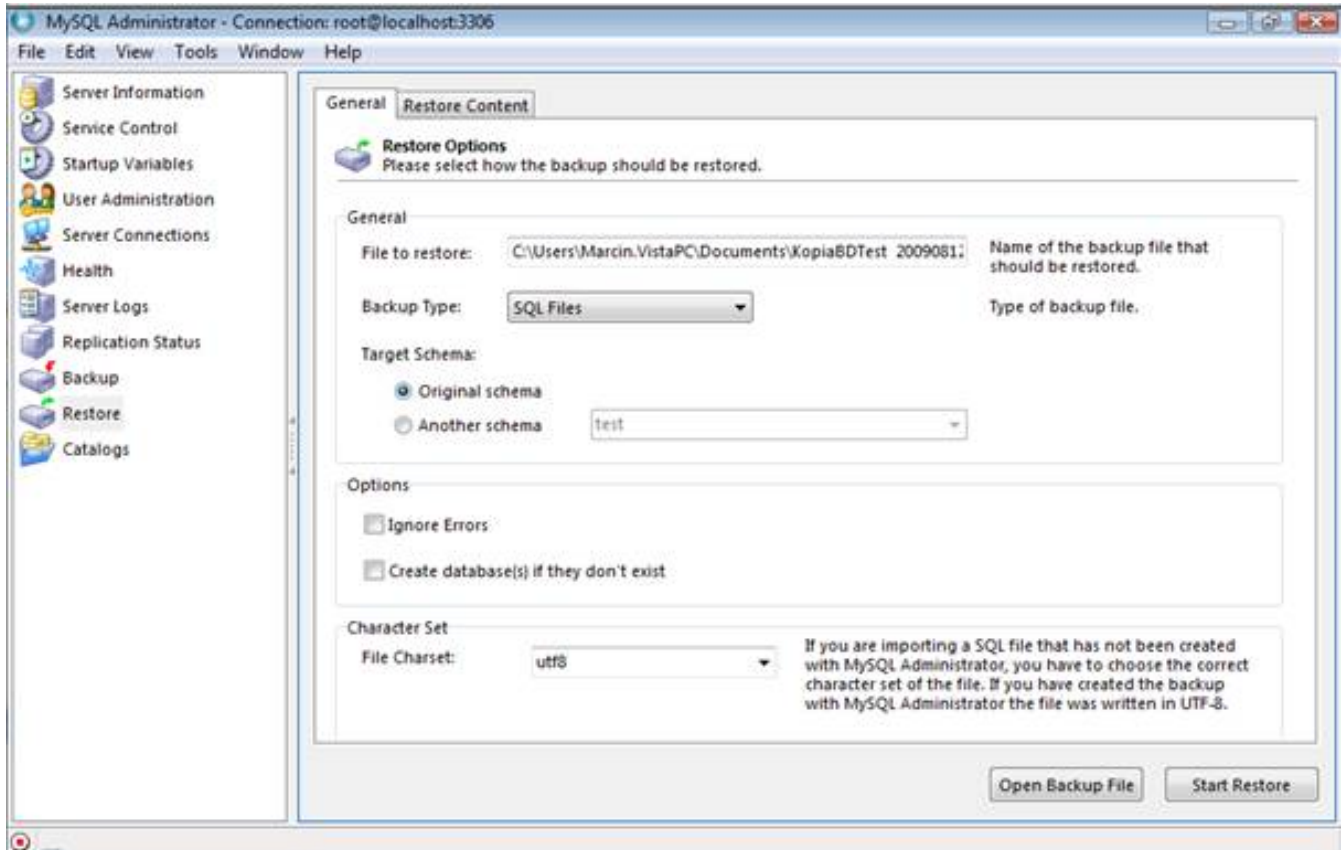
```
SELECT * FROM test.item i;
```

3. Wykonując instrukcje SELECT, sprawdziłeś, że w tabeli stock nie ma już żadnych danych i że cena sprzedaży wszystkich towarów wynosi 10. Na szczęście mamy aktualną kopię zapasową tych danych.
4. Przełącz się do programu MySQL Administrator i wybierz sekcję Restore.
5. Kliknij przycisk Open Backup File i wskaż utworzony w poprzednim ćwiczeniu plik kopii.
6. W sekcji Target Schema znajdują się dwa pola wyboru:

Zarządzamy serwerem baz danych

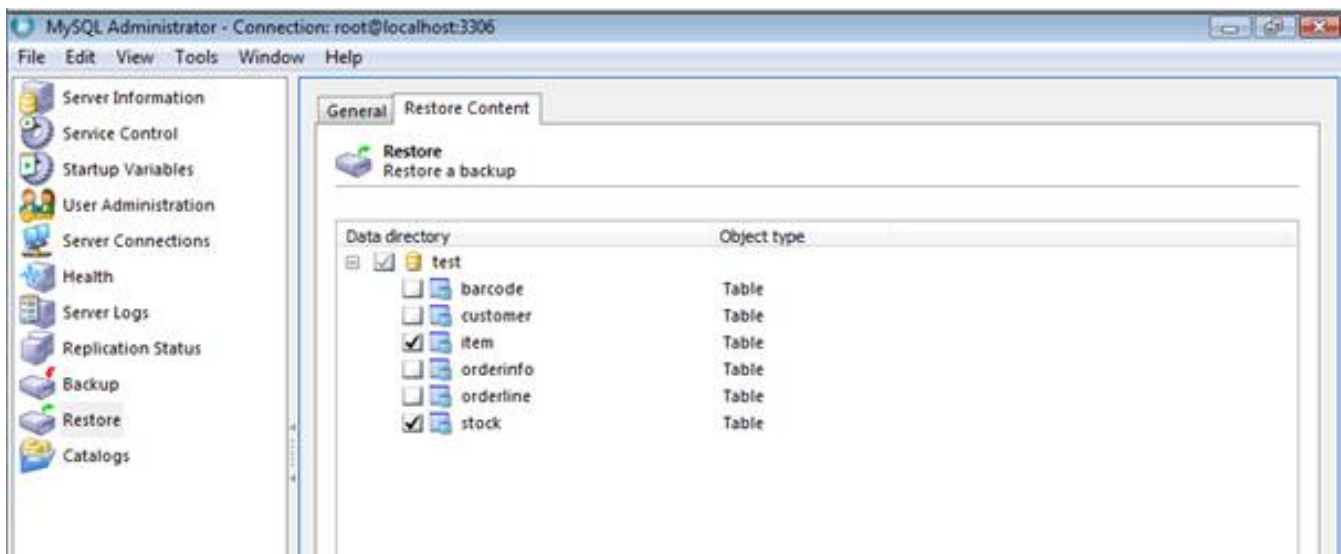
Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

1. Domyślnie odtwarzane z kopii zapasowej obiekty (takie jak tabele) tworzone są w tej samej bazie (schemacie), w której powstały.
2. Odtwarzając plik kopii zapasowej, możemy również utworzyć kopię bazy albo zastąpić inną bazę danych. Upewnij się, że wybrana jest opcja Original Schema (rysunek 2.10).



Rysunek 2.10. Wykonanie i odtworzenie kopii zapasowej to bezpieczny i łatwy sposób przeniesienia bazy na inny serwer

3. Przejdź na zakładkę Restore Content i kliknij Analize Backup File Content. Po chwili zobaczysz zawartość pliku kopii zapasowej.
4. Rozwiń drzewko przedstawiające zawartość pliku kopii i zaznacz tylko te dwie tabele, które należy odtworzyć (rysunek 2.11).



Rysunek 2.11. W tym wypadku chcemy odtworzyć tylko dwie tabele. Możemy to zrobić, nie tracąc zmian, które zostały wprowadzone.

5. Kliknij Start Restore.
6. Przełącz się do programu MySQL Query Browser i raz jeszcze wykonaj obie instrukcje SELECT — tym razem zwróciły one prawidłowe dane. Nie wykonuj ponownie instrukcji DELETE i UPDATE.

Automatyzujemy wykonywanie kopii zapasowych

Mam nadzieję, że poprzedni prosty przykład przekonał wszystkich, że warto regularnie wykonywać kopie baz danych. A skoro jakaś operacja ma być wykonywana regularnie, to warto ją zautomatyzować. Niestety, za taką wygodę zapłacimy bardzo wysoką cenę — znacznie obniżymy poziom bezpieczeństwa serwera.

1. Uruchom program MySQL Administrator, połącz się z serwerem jako root i wybierz sekcję Backup.
2. Zaznacz utworzony wcześniej projekt kopii bazy test.
3. Przejdź na zakładkę Schedule i zaznacz opcję Schedule this backup project.
4. Zobaczysz ostrzeżenie — automatyczne uruchamianie zadania wymaga zapisania w programie haseł użytkowników.
5. Otwórz (klikając wyświetlony razem z komunikatem ostrzeżenia przycisk Open Options) okienko opcji i przejdź do sekcji General Options. Zaznacz pole Store passwords jako metodę, wybierając Obscured. Od tego momentu niepowołane osoby będą w stanie odczytać zapisane w programie zakodowane hasła.
6. Zatwierdź zmiany i przejdź do sekcji Connections — ponieważ kopia będzie wykonywana o określonych godzinach, program MySQL Administrator będzie musiał automatycznie połączyć się z serwerem MySQL. Wymaga to zapisania w nim odpowiedniego ciągu połączenia, z którego będzie mógł skorzystać:
 1. Kliknij New Connection.
 2. Podaj nazwę tworzonego połączenia, nazwę użytkownika (root), jego hasło oraz nazwę komputera, na którym działa serwer MySQL.
 3. Zastosuj wprowadzone zmiany i zamknij okienko opcji.
 4. Z menu File wybierz Reconnect.
 5. Rozwiń znajdujące się w okienku logowania pole Stored Connection i wybierz utworzone wcześniej połączenie.
 6. Połącz się z serwerem MySQL, podając te same nazwę i hasło użytkownika, które są zapisane w ciągu połączenia.

7. Kopia bazy powinna być tworzona codziennie. Wybierz opcję Execute backup daily — wtedy najmniej użytkowników korzysta z serwera — i wpisz godzinę 2:00.
8. Określ lokalizację plików kopii zapasowej i zapisz projekt. Zostaniesz poproszony o podanie nazwy użytkownika i hasła. Wpisz swoje hasło systemowe (nie nazwę i hasło użytkownika serwera MySQL) i kliknij OK.

Monitorowanie serwera MySQL

Tylko monitorując na bieżąco działanie serwera baz danych, administrator może zapobiegać sytuacjom problemowym lub przynajmniej szybko reagować, gdy się pojawią. Absolutne minimum to regularne (codzienne) sprawdzanie dzienników serwera, monitorowanie wydajności i stopnia wykorzystania takich zasobów, jak pamięć czy procesor i zarządzanie sesjami użytkowników. **Konfigurujemy i odczytujemy dzienniki serwera**

MySQL może zapisywać informacje diagnostyczne w kilku różnych dziennikach. Domyślnie włączony jest dziennik błędów — znajdziemy w nim informacje o stanie usługi. Żeby je przejrzeć:

1. Uruchom narzędzie MySQL Administrator i połącz się z serwerem jako użytkownik root.
2. Przejdź do sekcji Serwer Logs. Na zakładce Error Log zobaczysz zgłoszone przez MySQL informacje o uruchamianiu i zatrzymywaniu głównej usługi serwera i usługi wsparcia dla tabel typu InnoDB (listing 2.10).

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

Listing 2.10. Fragment dziennika błędów serwera MySQL z włączoną obsługą tabel typu InnoDB

```
090811 12:30:02 [Warning] option 'innodb-log-buffer-size': signed value 1048576 adjusted to 2097152
090811 12:30:03 [Warning] option 'innodb-buffer-pool-size': signed value 8388608 adjusted to 67108864
090811 12:30:03 [Warning] option 'innodb-log-file-size': signed value 10485760 adjusted to 33554432
InnoDB: The InnoDB memory heap has been disabled.
InnoDB: Mutex and rw_lock use atomics.
InnoDB: ios_per_array 32 read threads 1 write threads 1
InnoDB: The first specified data file .ibdata1 did not exist:
InnoDB: a new database to be created!
090811 12:30:04 InnoDB: Setting file .ibdata1 size to 10 MB
InnoDB: Database physically writes the file full: wait...
090811 12:30:05 InnoDB: Log file .ib_logfile0 did not exist: new to be created
InnoDB: Setting log file .ib_logfile0 size to 32 MB
InnoDB: Database physically writes the file full: wait...
090811 12:30:07 InnoDB: Log file .ib_logfile1 did not exist: new to be created
InnoDB: Setting log file .ib_logfile1 size to 32 MB
InnoDB: Database physically writes the file full: wait...
090811 12:30:08 InnoDB: Log file .ib_logfile2 did not exist: new to be created
InnoDB: Setting log file .ib_logfile2 size to 32 MB
InnoDB: Database physically writes the file full: wait...
InnoDB: Doublewrite buffer not found: creating new
InnoDB: Doublewrite buffer created
InnoDB: Creating foreign key constraint system tables
InnoDB: Foreign key constraint system tables created
InnoDB master thread running with io_capacity 200
090811 12:30:10 InnoDB: Started; log sequence number 0 0
090811 12:30:11 [Note] Event Scheduler: Loaded 0 events
090811 12:30:11 [Note] C:\Program Files\MySQL\MySQL Server 5.4\bin\mysqld: ready for connections.
```

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

Version: '5.4.1-beta-community' socket: " port: 3306 MySQL Community Server (GPL)

090811 12:31:19 [Note] C:\Program Files\MySQL\MySQL Server 5.4\bin\mysqld: Normal shutdown

090811 12:31:19 [Note] Event Scheduler: Purging the queue. 0 events

090811 12:31:19 InnoDB: Starting shutdown...

090811 12:31:20 InnoDB: Shutdown completed; log sequence number 0 46409

090811 12:31:20 [Warning] Forcing shutdown of 1 plugins

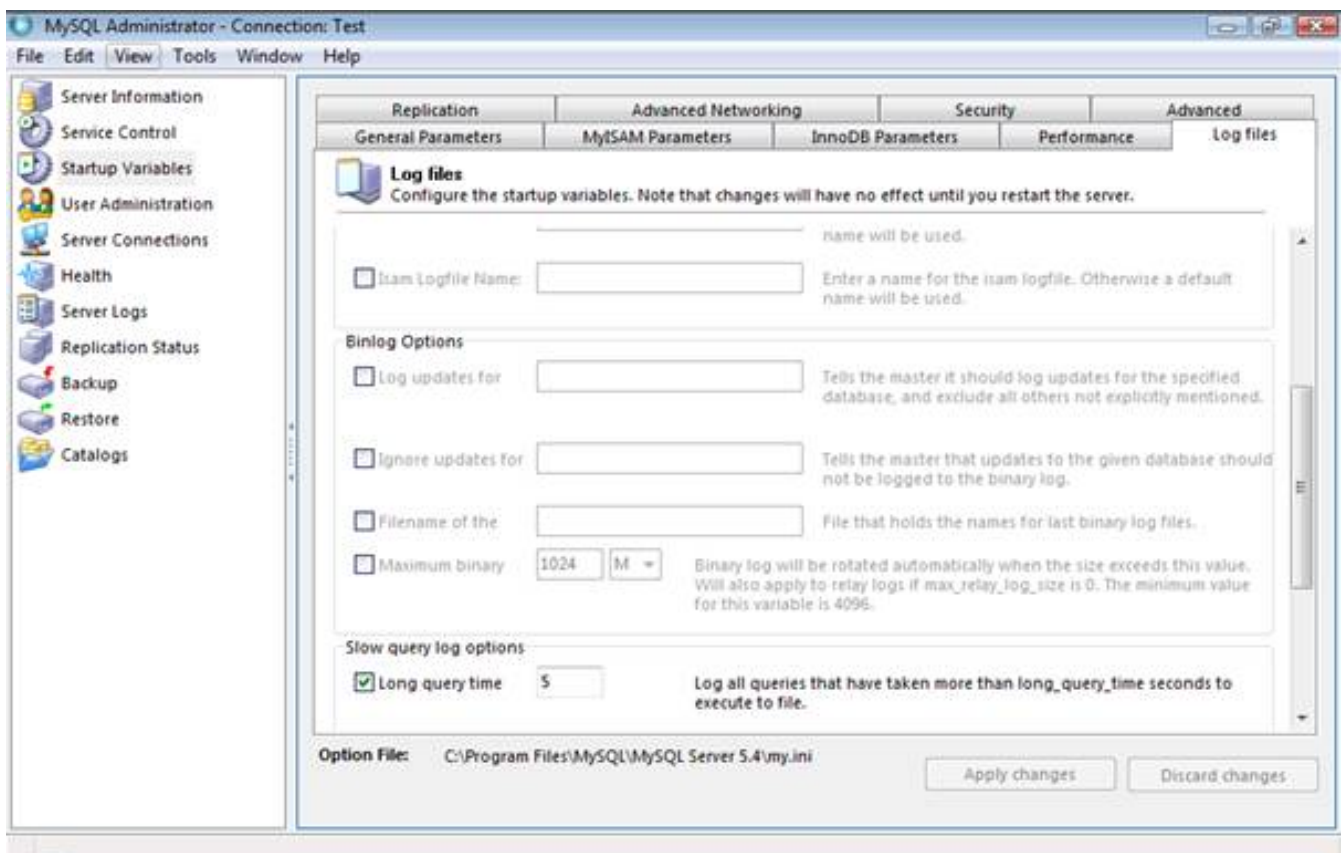
090811 12:31:20 [Note] C:\Program Files\MySQL\MySQL Server 5.4\bin\mysqld: Shutdown complete

3. Gdy wybierzesz w okienku po lewej stronie konkretne zdarzenie, w okienku po prawej zobaczysz jego szczegółowy opis.

4. Drugim przydatnym dziennikiem jest dziennik zapisujący informacje o długo wykonywanych, prawdopodobnie wymagających sprawdzenia lub optymalizacji, instrukcjach użytkowników. Ten dziennik musimy najpierw włączyć.

5. Przejdź do sekcji Startup Variables i wybierz zakładkę Log files.

6. Włącz dziennik Slow Queries Log i podaj jego nazwę, a następnie uaktywnij sekcję Long Query time i określ czas wykonania (5 sekund), po którego przekroczeniu zapytanie będzie uważane za wolne (rysunek 2.12).



Rysunek 2.12. W przypadku testowej bazy danych, liczącej tylko sześć niewielkich tabel, zapytanie wykonywane ponad pięć

Zarządzamy serwerem baz danych

Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

7. Zaakceptuj zmiany i uruchom ponownie serwer MySQL.
8. Uruchom (wybierając schemat test) MySQL Query Browser i wykonaj instrukcję z listingu 2.11.

Listing 2.11. Źle napisana instrukcja, której wykonanie będzie trwało dłużej niż 5 sekund

```
USE test;
```

```
SELECT *
```

```
FROM barcode, customer, item, orderinfo, orderline;
```

9. Wróć do konsoli administracyjnej, wybierz sekcję Server Logs i przejdź na zakładkę Slow Log. Znajdziesz tam następujące informacje o jednej, długo wykonywanej instrukcji SELECT (listing 2.12).

Listing 2.12. W tym wypadku odczytanie 138 tysięcy wierszy zajęło 48 sekund

```
Time Id Command Argument
```

```
# Time: 090812 14:32:34
```

```
# User@Host: root[root] @ localhost [127.0.0.1]
```

```
# Query_time: 48 Lock_time: 0 Rows_sent: 138600 Rows_examined: 5
```

```
use test;
```

```
SELECT * FROM barcode, customer, item, orderinfo, orderline;
```

10. Zapisz plik dziennika, klikając przycisk Save Log Page i podając nazwę pliku kopii.

Monitorowanie bieżącej aktywności i wykorzystania zasobów serwera

MySQL Administrator graficznie i w czasie rzeczywistym przedstawia najważniejsze statystyki serwera. W dodatku możemy dodawać do niej własne zakładki zawierające dodatkowe statystyki. Przeanalizujemy tylko interpretowanie wyników standardowych liczników:

1. Zaznacz sekcję Health.

2. Na zakładce Connection Health znajdują się trzy liczniki:

1. Pierwszy, Connection Usage, pokazuje procentową liczbę jednoczesnych połączeń względem ustalonego maksimum. Jeżeli wartości licznika przekraczają 90, zwiększ (w sekcji Startup Variables) maksymalną liczbę jednoczesnych połączeń.

2. Drugi, Traffic, pokazuje liczbę bajtów wysyłanych przez serwer w odpowiedzi na żądania użytkowników. Obserwując jego wskazania, będziemy wiedzieli, kiedy serwer jest silnie obciążony i ile danych może wysłać w jednostce czasu.

3. Trzeci, Number of SQL Queries, pokazuje liczbę właśnie wykonywanych instrukcji użytkowników. Należy wykorzystywać go w połączeniu z poprzednim — w ten sposób lepiej można ocenić wydajność serwera.

4. Przejdź na zakładkę Memory Health. Tutaj znajdują się dwa liczniki bardzo istotne dla optymalizacji wydajności:

1. Żeby rozumieć wskazania pierwszego licznika, Query Cache Hitrate, musimy pamiętać o dwóch rzeczach:

- i. że pamięci RAM są średnio 1000 razy szybsze niż dyski twarde; dlatego serwery bazodanowe buforują w pamięci odczytane wcześniej dane;

Zarządzamy serwerem baz danych

Dodał Administrator

poniedziałek, 19 kwiecień 2010 18:41

ii. że SQL jest językiem interpretowanym. Wykonanie zapytania przebiega następująco: najpierw sprawdzana jest poprawność zapytania, potem następuje optymalizacja (wybór indeksów, sposobów złączeń itd.), a na końcu zostaje zrealizowany najlepszy znaleziony plan wykonania. Operacje te zajmują sporo czasu, dlatego serwer buforuje plany wykonanych zapytań, i jeżeli takie samo zapytanie będzie po raz kolejny wysłane przez użytkownika, korzysta z wcześniej przygotowanych planów. Jeżeli wskazania tego licznika stale utrzymują się około 100%, należy zwiększyć rozmiar bufora zapytań. Jeżeli nie przekraczają kilkunastu procent, należy zoptymalizować program klienta.

1.

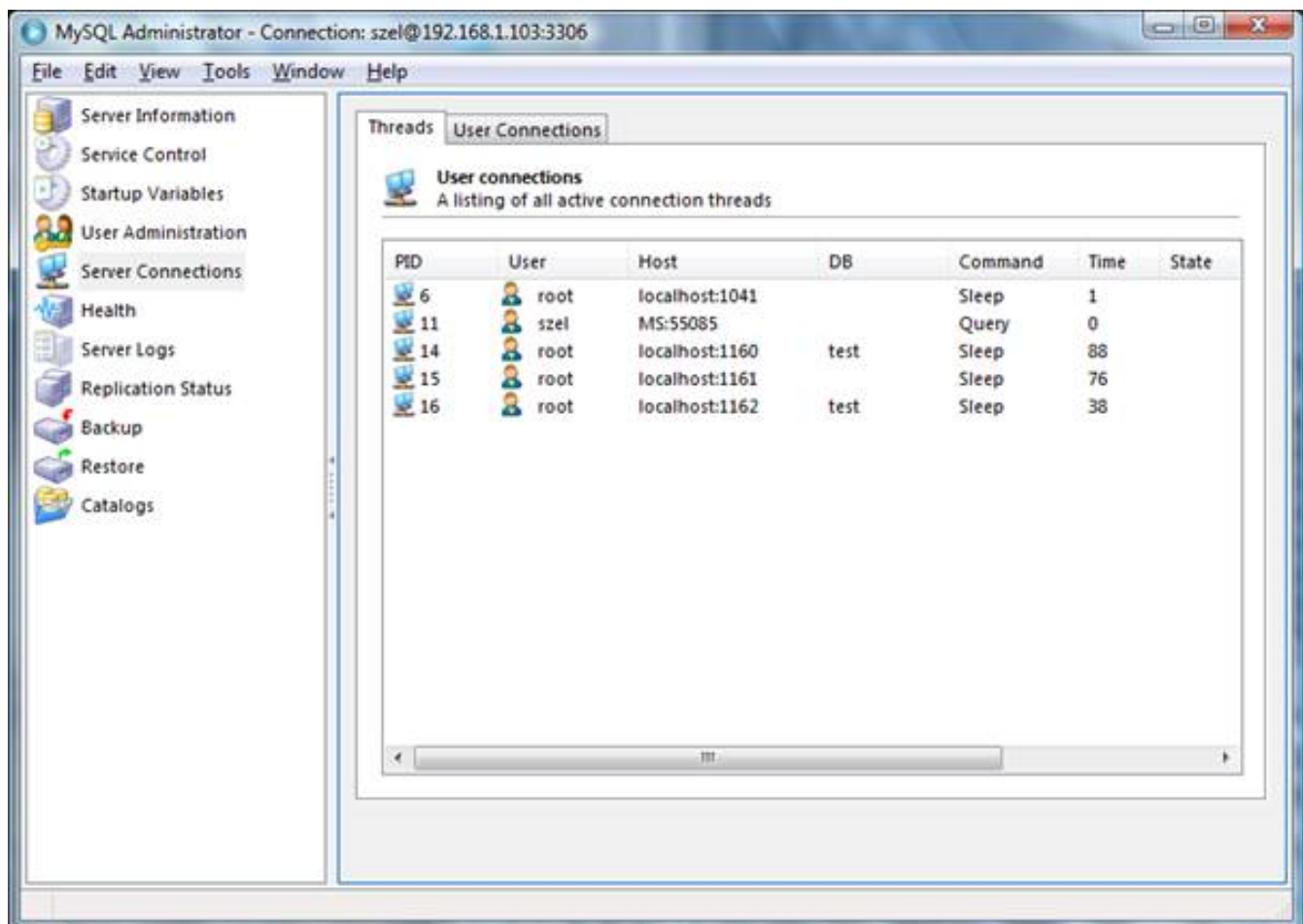
1. Drugi, Key Efficiency, pokazuje procent wykorzystania bufora. Jeżeli ten procent jest wysoki (ponad 90%), należy (w sekcji Startup Variables) zwiększyć jego rozmiar.

Zarządzanie sesjami użytkowników

Czasami administrator musi monitorować i zakończyć sesje użytkowników. Jeżeli na przykład pewna tabela jest zablokowana pomimo tego, że w danej chwili nikt nie pracuje z bazą, może to oznaczać awarię lub błąd programu klienckiego. W takiej sytuacji zablokowane zasoby mogą nie zostać zwolnione aż do momentu ponownego uruchomienia serwera.

1. Przejdź na zakładkę Server Connections.

2. Na zakładce Threads zobaczysz aktywne wątki — MySQL jest programem wielowątkowym, czyli (na wieloprocesorowym komputerze) może jednocześnie wykonywać wiele operacji. W uproszczeniu każde żądanie klienta powoduje uruchomienie kolejnego wątku, w ramach którego serwer zrealizuje to żądanie (rysunek 2.13).



Zarządzamy serwerem baz danych

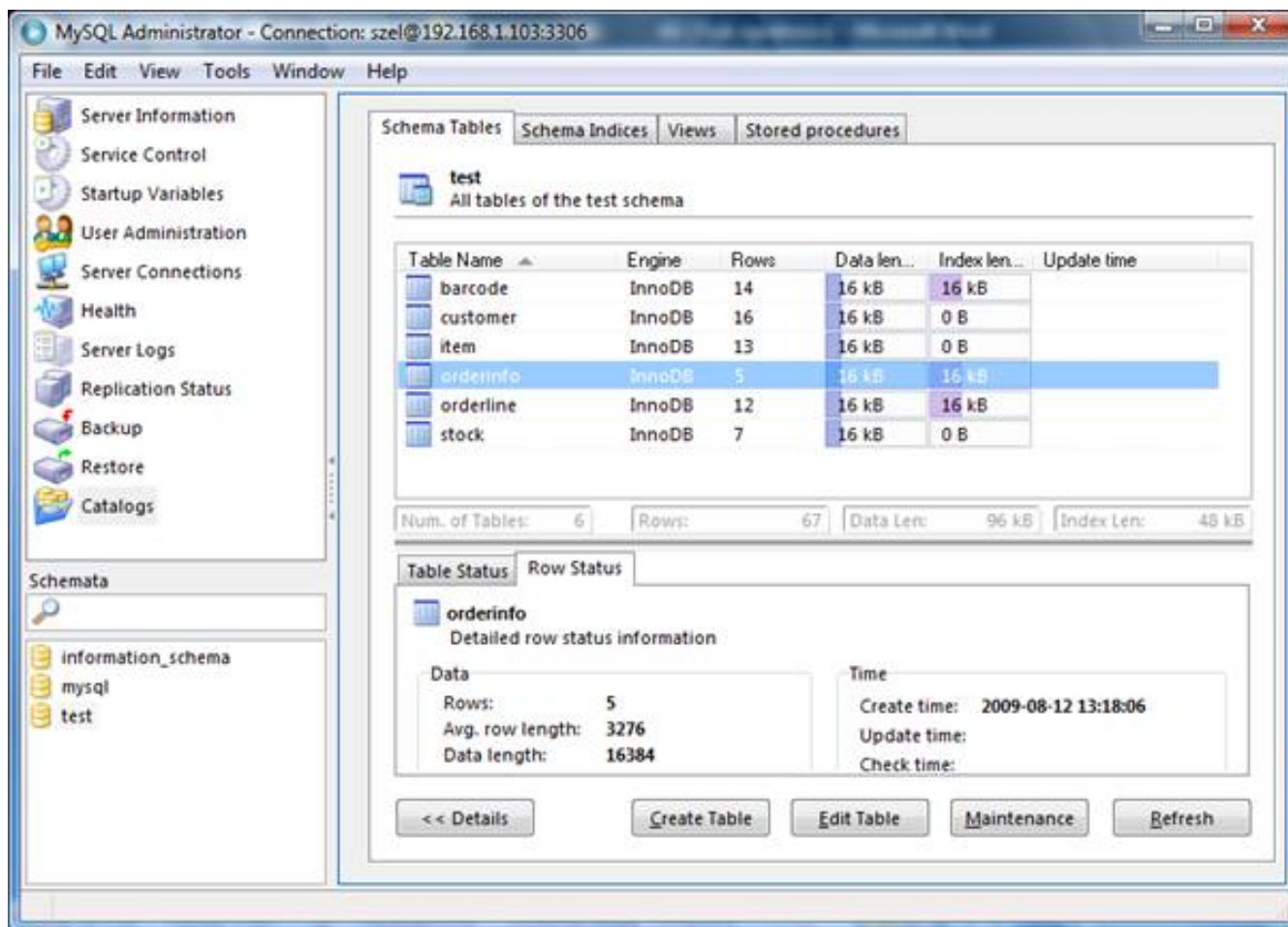
Dodał Administrator
poniedziałek, 19 kwiecień 2010 18:41

Rysunek 2.13. Każda sesja kliencka ma swój identyfikator (PID). MySQL Administrator pokazuje dodatkowo nazwę użytkownika

3. Klikając prawym przyciskiem myszy wybrany wątek, będziesz mógł natychmiast go zakończyć.
4. Zakładka User Connections pokazuje te same dane, ale w nieco inny sposób — zamiast informacji o pojedynczych wątkach zobaczysz tu informacje o połączonych z serwerem użytkownikach. Dopiero po wybraniu użytkownika w dolnej części ekranu zobaczysz szczegółowy opis każdej nawiązanej przez niego sesji.

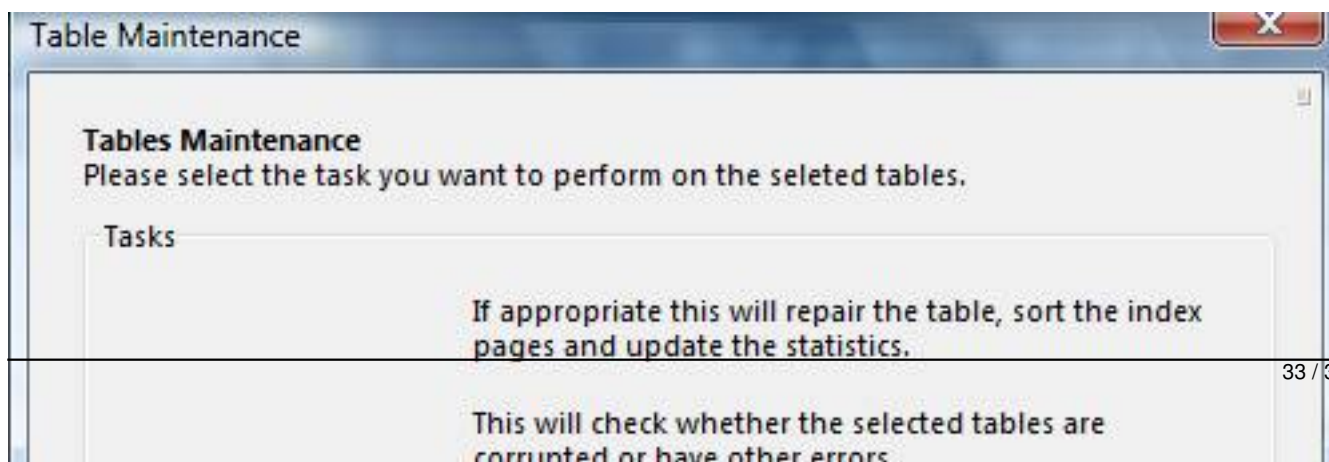
Sprawdzanie spójności i optymalizacja bazy danych

1. Ważną dla administratora informacją jest rozmiar bazy danych i wielkość poszczególnych tabel. Dane te poznamy, wybierając sekcję Catalogs i zaznaczając interesującą nas bazę danych (rysunek 2.14).



Rysunek 2.14. Graficzne przedstawienie wielu informacji o fizycznej strukturze tabel i zapisanych w nich danych

2. Zaznacz wszystkie tabele (na przykład zaznacz jedną, a potem naciśnij kombinację klawiszy Ctrl+A), a następnie kliknij przycisk Maintenance. Wyświetli się okno dialogowe pokazane na rysunku 2.15.



Rysunek 2.15. Do typowych zadań administratora należy również sprawdzanie spójności danych, ich porządkowanie i naprawa.

3. Wybierz opcję Check Table i przekonaj się, ile trwa szybkie, a ile dokładne (ang. Extended) sprawdzenie tych sześciu tabel. Za każdym razem powinieneś otrzymać komunikat: Status checked for table test.<nazwa tabeli>. OK.

Przed ewentualną naprawą tabel zrób kopię bazy danych.

4. W następnej kolejności zoptymalizuj (poprzez uporządkowanie) tabele. Optymalizacja każdej tabeli powinna zakończyć się komunikatem: Table test. <nazwa tabeli>. optimized. OK.