

**Instytut Teleinformatyki**

Wydział Fizyki, Matematyki i Informatyki

Politechnika Krakowska



Laboratorium Administrowania Systemami Komputerowymi

---

*„Konfiguracja i administracja  
bazą danych MySQL”*

ćwiczenie numer 7

---

## Spis treści

<b>1. INFORMACJE WSTĘPNE</b> .....	<b>3</b>
1.1. TEMAT ĆWICZENIA .....	4
1.2. ZAGADNIENIA DO PRZYGOTOWANIA.....	4
1.3. CEL ĆWICZENIA .....	4
<b>2. ATUTY MYSQL</b> .....	<b>5</b>
<b>3. PRZEBIEG ĆWICZENIA</b> .....	<b>7</b>
3.1. ZADANIE NR 1 – KONFIGURACJA UŻYTKOWNIKA ROOT I TWORZENIE NOWEJ BAZY	8
3.2. ZADANIE NR 2 - TWORZENIE KONT UŻYTKOWNIKÓW ORAZ NADAWANIE IM PRAW DOSTĘPU.....	9
3.3. ZADANIE NR 3 - TWORZENIE KOPII ZAPASOWEJ BAZY DANYCH.....	10
3.4. ZADANIE NR 4 – MIGRACJA/PRZENOSZENIE BAZY DANYCH W INNE MIEJSCE.....	12
3.5. ZADANIE NR 5 – KLASTROWANIE/REPLIKACJA.....	13

---

# **1. Informacje wstępne**

### 1.1. **Temat ćwiczenia**

Tematem ćwiczenia jest konfiguracja i administracja bazą danych MySQL w środowisku Linux Debian Squeez.

### 1.2. **Zagadnienia do przygotowania**

Przed przystąpieniem do wykonania ćwiczenia należy zapoznać się z następującymi zagadnieniami:

- czym jest MySQL
- jakie są korzyści stosowania tego rodzaju baz danych

### 1.3. **Cel ćwiczenia**

Dzięki temu ćwiczeniu wykonujący pozna:

- jakie pakiety są konieczne do administrowania i użytkowania MySQL
- jak utworzyć nowego użytkownika „root” w MySQL
- jak utworzyć nowych użytkowników/klientów
- jak nadawać użytkownikom określone prawa dostępu
- jak utworzyć nową bazę danych
- jak tworzyć kopię zapasową bazy danych
- jak tworzyć tabele w danej bazie
- jak zrobić kopię zapasową wybranej tabeli
- jak utworzyć skompresowaną kopię zapasową
- jak dokonywać migracji
- jak działa klastrowanie/replikacja, tworzenie serwerów master’a i slave’a gotowych do współpracy

---

## **2. Atuty MySQL**

Wymienione poniżej możliwości MySQL-a sprawiają, iż stanowi on interesującą alternatywę dla podobnych, lecz komercyjnych produktów:

- baza danych zdolna pomieścić nawet kilkadziesiąt milionów rekordów
- interfejsy API dla najważniejszych języków programowania (m.in. C, PHP, Perl)
- wykorzystanie mocy komputerów wieloprocesorowych
- nieograniczona liczba użytkowników mogących jednocześnie korzystać z bazy danych
- duża szybkość działania (wynikająca w pewnym stopniu z braku niektórych funkcji - np. obsługi transakcji)

MySQL może zostać uruchomiony na maszynach pracujących pod kontrolą takich systemów jak Windows, Unix, Mac OS, OS/2.

MySQL jest dostępny na licencji GPL, jednakże można nabyć również wersje licencjonowane komercyjnie. Biblioteki klienckie są również dostępne na licencji GPL lub komercyjnej. Istnieje wymóg nabycia licencji komercyjnej jeśli zamierzamy dystrybuować aplikację komercyjną (niezgodną z licencją GPL), wraz z zintegrowaną bazą MySQL.

---

### **3. Przebieg ćwiczenia**

### 3.1. Zadanie nr 1 – Konfiguracja użytkownika root i tworzenie nowej bazy

Pierwszą konieczną operacją do wykonania jest ustawienie hasła użytkownika root.

```
mysqladmin -u root password hasło
```

gdzie “hasło” zastępujemy naszym własnym hasłem np. “askask”

**WAŻNE:** Po wykonaniu powyższej komendy należy usunąć wpis znajdujący się w pliku `/.bash_history` który zawiera zapis wykonanej przez nas komendy (wraz ustalonym przez nas hasłem).

Kolejną operacją jest utworzenie nowej bazy danych:

```
mysqladmin create mojaBaza -u root -p  
Enter password: hasło (np. askask)
```

**WAŻNE:** Jako, że root posiada nieograniczone prawa do bazy, nie powinno się go używać jako konta dostępu do użytkowych baz danych. Do takich prac powinno się używać kont użytkowników nie mających pełni praw w bazie danych.



### 3.2. Zadanie nr 2 - Tworzenie kont użytkowników oraz nadawanie im praw dostępu

Konta tworzymy z poziomu root'a.

```
mysql -u root -p
Enter password: hasło
```

Wykonując polecenie powyżej przechodzimy do MySQL'a.

Aby utworzyć konto oraz nadać podstawowe uprawnienia do bazy "mojaBaza" użytkownikowi studentASK musimy wykonać następującą komendę:

```
grant select,insert,update,delete,create,drop on mojaBaza.* to
studentASK@localhost identified by 'ask';
```

Aby opuścić znakowego klienta bazy danych należy użyć polecenia *exit*.

W celu sprawdzenia poprawności wykonanej operacji należy spróbować zalogować się na konto użytkownika „studentASK”. Aby to wykonać postępujemy analogicznie jak było to w przypadku użytkownika *root*:

```
mysql -u studentASK -p
Enter password: ask
```

### 3.3. Zadanie nr 3 - Tworzenie kopii zapasowej bazy danych

MySQL posiada wbudowane narzędzie “mysqldump” które służy do tworzenia niezależnych od platformy plików tekstowych, które zawierają pełną kopię tego co znajdowało się w bazie , która przez przypadek została utracona.

Tworzymy katalog w którym będziemy przechowywać dane z dzisiejszego laboratorium. Katalog w którym utworzymy nowy folder to /home.

```
mkdir mysql
```

Należy zacząć od utworzenia folderu na naszą kopię zapasową:

```
mkdir mysql/kopiaZapasowa
```

Kolejnym krokiem jest wykonanie komendy która utworzy plik kopii:

```
mysqldump -u root -p mojaBaza> /home/mysql/kopiaZapasowa/mojaBaza.2012-10-18
```

W wyniku powyższej komendy został utworzony plik mojaBaza.2012-10-18, który jest kopią zapasową naszej bazy 'mojaBaza' w lokalizacji sprecyzowanej przez nas w /home/mysql/kopiaZapasowa/.

Kopia zapasowa może być również zrobiona dla określonych tabel, czyli nie koniecznie dla całej bazy danych.

Aby przetestować takie rozwiązanie musimy najpierw użyć naszą bazę danych.

```
use mojaBaza;
```

następnie tworzymy przykładową tabelę:

```
create table student(  
nazwisko char(22),  
ocena int(7));
```

Teraz możemy zrobić kopię zapasową naszej tabeli. Tworzymy ją w sposób podobny jak robiliśmy to z bazą danych:

```
mysqldump -u root -p mojaBaza student > /home/mysql/kopiaZapasowa/student.2012-10-18
```

Po wykonaniu zadania należy sprawdzić czy nasze dane z tabeli znalazły się w pliku kopii zapasowej.

Aby zaoszczędzić miejsca na dysku, możemy w locie kompresować pliki za pomocą gzipa:

```
mysqldump -u root -p mojaBaza | gzip > /home/mysql/kopiaZapasowa/mojaBaza.2012-10-18.gz
```

Żeby przywrócić bazę z tego przykładowego pliku, musimy najpierw stworzyć pustą bazę.

```
mysql -u root -p  
password: askask
```

i w mysql:

```
CREATE DATABASE mojaBaza1;
```

Aby wypełnić ją tabelkami, a tabelki danymi, musimy wybrać właściwy plik z kopią bazy:

```
mysql -u root -p mojaBaza1 < /home/mysql/kopiaZapasowa/mojaBaza.2012-10-18
```

Należy podać dane niezbędne do autoryzacji i nasza baza może być znowu używana.

### 3.4. Zadanie nr 4 – Migracja/przenoszenie bazy danych w inne miejsce

Przed rozpoczęciem operacji przenoszenia bazy danych należy ją wyłączyć:

```
/etc/init.d/mysql stop
```

Następnie tworzymy katalog do którego będziemy migrować bazę danych:

```
mkdir /home/mysql/migracja
```

Kolejnym krokiem jest skopiowanie naszej bazy używając komendy 'cp' wraz z parametrami p,v,d,R:

```
cp -pvdR /var/lib/mysql /home/mysql/migracja
```

Kolejnym krokiem w przenoszeniu naszej bazy danych jest wskazanie lokalizacji danych z których chcemy korzystać. Lokalizację tą musimy zmienić w pliku: `/etc/mysql/my.cnf` zmieniając wiersz:

```
datadir = /var/lib/mysql
```

na

```
datadir = /home/mysql/migracja/mysql
```

Jeśli powyższe kroki zostały wykonane poprawnie, baza danych korzystająca z przeniesionych plików powinna bez problemów być włączona za pomocą polecenia:

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

### 3.5. Zadanie nr 5 – Klastrowanie/Replikacja

Niezależne i działające w czasie rzeczywistym serwery pozwalają w widoczny sposób wspomagać łatwość dostępu do danych. W ostatnim czasie wyjątkowo rekomendowaną techniką takiego klastrowania jest replikacja serwera głównego. Taką replikację można przeprowadzić w oparciu o schemat master-slave. Serwer master udostępnia dane zapisując je do pliku logowania binarnego (w momencie kiedy cokolwiek zostało zmienione w co najmniej jednej z baz). Serwer slave pobiera dane i zapisuje je w replice. W przypadku awarii mastera mamy w pełni funkcjonalne serwery slave na które możemy przełączyć aplikację. Można również podłączyć system backupów do slave'a aby nie obciążać mastera.

Dla szybszego wykonania ćwiczenia użyjemy bazę danych utworzoną przez nas wcześniej (np. mojaBaza).

#### 3.5.1 Konfiguracja serwera „Master”

Na serwerze master należy włączyć logowanie binarne. Aby to uzyskać należy zmodyfikować plik `/etc/mysql/my.cnf` dodając w części `[mysqld]` poniższą linię:

```
log-bin = /var/log/mysql/mysql-bin.log
```

W tej samej sekcji dodajemy linię, która ustawia unikalny identyfikator serwera:

```
server-id=1
```

Pod dopisaniu powyższych parametrów konfiguracyjnych należy zrestartować serwer. To zadanie powinno być wykonane z poziomu roota systemu. (np. `su root`, a następnie hasło dostępu do roota)

Restart serwera możemy wykonać komendą:

```
/etc/init.d/mysql restart
```

W wyniku tej operacji powstanie nowy plik w `/var/log/mysql/` np. `mysql-bin.00001`

Aby się upewnić, że zadanie zostało wykonane poprawnie należy zalogować się do MySQL:

```
mysql -u root -p  
password: hało (np. askask)
```

a następnie z poziomu komend MySQL wydajemy polecenie, które podaje aktualną nazwę pliku logowania binarnego i bieżącą pozycję w tym pliku:

```
show master status;
```

Wyświetlone parametry File i Position będą potrzebne do konfiguracji serwera slave.

Ostatnim zadaniem do wykonania w tej części jest utworzenie użytkownika `user2` z uprawnieniami do pracy w charakterze serwera slave. Za pomocą tego użytkownika serwer slave będzie łączył się z serwerem master.

```
GRANT REPLICATION SLAVE ON *.* to 'user2'@'%' IDENTIFIED BY 'ask2';
```

### 3.5.2. Konfiguracja serwera „Slave”

Serwer Slave jest konfigurowany częściowo w pliku `/etc/mysql/my.cnf` a częściowo dalej podanymi komendami.

Należy rozpocząć od konfiguracji pliku `my.cnf` a konkretnie sekcji `[mysqld]` dodając linię, która ustawia unikalny identyfikator dla serwera slave:

```
server-id = 2
```

Dalszą część konfiguracji należy przeprowadzić już z poziomu konta MySQL (`root`) stosując dość skomplikowane polecenie:

```
CHANGE MASTER TO MASTER_HOST='127.0.0.1', MASTER_USER='user2',  
MASTER_PASSWORD='ask2', MASTER_LOG_FILE='mysql-bin.000001',  
MASTER_LOG_POS=98;
```

MASTER\_HOST - to adres IP serwera master, w naszym przypadku 127.0.0.1  
MASTER\_USER - nazwa użytkownika utworzonego w poprzednim punkcie  
MASTER\_PASSWORD - hasło użytkownika utworzonego poprzednim punkcie  
MASTER\_LOG\_FILE - parametr File odczytany w punkcie poprzednim  
MASTER\_LOG\_POS - parametr Position odczytany punkcie poprzednim

Aby rozpocząć replikację, z poziomu MySQL wykonujemy polecenie:

```
START SLAVE;
```

Aby na bieżąco sprawdzać stan serwera „Slave” należy użyć komendy:

```
SHOW SLAVE STATUS;
```

W wyniku tej komendy zostaną wyświetlone bieżące parametry użytego serwera Slave. Jeśli zależy nam na bardzo dokładnych statystykach dobrym rozwiązaniem jest użycie

oprogramowania Nagios.

### 3.5.3. Inne możliwości replikacji baz:

Istnieje też możliwość replikacji nie wszystkich dostępnych baz danych, a tylko jednej z nich.

Wymusić taką operację można umieszczając (na serwerze slave) w pliku my.cnf w sekcji [mysqld] następującą opcję:

```
replicate-do-db = mojaBaza
```

(mojaBaza to baza przykładowa)

Aby replikować wszystkie bazy z wyjątkiem jednej należy użyć opcji:

```
replicate-ignore-db = mojaBaza
```

Aby replikować tylko wybraną tabelę można użyć następującej opcji na serwerze slave:

```
replicate-do-table = mojaBaza.student
```

(*student to tabela*)

Aby replikować wszystkie tabele z wyjątkiem wybranej należy użyć następującej opcji na serwerze slave:

```
replicate-ignore-table = mojaBaza.student
```



### 3.6. Zakończenie ćwiczenia

Po zakończeniu ćwiczenia proszę usunąć wszystkie pliki i katalogi które zostały utworzone podczas wykonywania ćwiczeń oraz przywrócić ustawienia w pliku `my.cnf`

Zatrzymać server Slave

```
STOP SLAVE;
```

Zresetować hasło roota

```
mysqladmin -p -u root password ''
```