

Praca z bazą danych MySQL
wersja 2.0

Nr ćwiczenia:	1
Temat:	Podstawy pracy z bazą danych MySQL
Cel ćwiczenia:	Celem ćwiczenia jest zapoznanie studenta z podstawami pracy z bazą danych MySQL. Większość z omawianych tu zagadnień jest przedstawiona bardzo pobieżnie. Szczegóły można znaleźć w literaturze lub „dopytać się” u prowadzącego ćwiczenia.
Wymagane przygotowanie teoretyczne:	—
Sposób zaliczenia:	Pozytywna ocena ćwiczenia przez prowadzącego pod koniec zajęć.
Prowadzący zajęcia:	dr inż. Artur Gramacki, dr inż. Jarosław Gramacki

1. Uwagi wstępne

1.1. Oprogramowanie

Wszystkie wykonywane ćwiczenia będą bazowały na wersji 5.0.16 systemu MySQL pracującego w systemie Windows. Jest to jedna z bardziej aktualnych wersji tego oprogramowania, choć prawdopodobnie w chwili czytania tego opracowania będzie dostępna już wersja nowsza¹. Jednak w praktyce zdecydowaną większość ćwiczeń można z powodzeniem wykonać używając wersji wcześniejszych (3.x, 4.x) i najprawdopodobniej późniejszych. Ćwiczenia mają bowiem bardzo podstawowy charakter i przynajmniej na początku nie dotyczą żadnych bardziej zaawansowanych zagadnień. Tam, gdzie numer wersji jest istotny zostanie to wyraźnie zaznaczone.

System MySQL można z powodzeniem zainstalować i używać w środowisku LINUX. Praca w tym środowisku, w porównaniu do środowiska Windows, jest niemal identyczna. Większe różnice występują w procesie instalacji i konfiguracji systemu (w LINUX-ie jest on bardziej złożony niż w Windows-ie), jednak te zagadnienia nie wchodzą w zakres materiału ćwiczeniowego i nie będą omawiane. Zakładamy więc, że na poszczególnych stanowiskach laboratoryjnych (pracujących pod systemem Windows) serwer MySQL jest już prawidłowo zainstalowany, skonfigurowany i gotowy do pracy.

Większość ćwiczeń wykonywana będzie na lokalnych instalacjach MySQL-a, gdzie studenci posiadać będą pełne uprawnienia do bazy danych. Podczas ćwiczeń może się więc zdarzyć przypadkowe uszkodzenie bazy. Oczywiście należy starać się do tego nie dopuszczać, jednak gdy coś takiego się zdarzy należy powiadomić o tym prowadzącego, który postara się odtworzyć uszkodzone elementy korzystając z kopii zapasowej. Hasło administratora (użytkownik root) zostanie podane w czasie zajęć.

Większość ćwiczeń będzie wykonywana z wykorzystaniem *wyłącznie* oprogramowywania klienckiego dostarczanego wraz z serwerem MySQL (chodzi o tzw. *MySQL Monitor*; program *mysql*). Nie będziemy więc korzystali z programów firm trzecich służących do wspomaganie i ułatwiania zarządzania bazą danych (jak np. bardzo popularny program *phpMyAdmin*² dostępny do pobra-

¹Tak to już jest w świecie informatyki, że nowe wersje programów pojawiają się bardzo często.

²Do pracy wymaga on zainstalowania oraz skonfigurowania serwera WWW *Apache* oraz języka *PHP*

nia ze strony <http://www.phpmyadmin.net>). Czynimy tak celowo, gdyż programy te, jakkolwiek bardzo dobre i funkcjonalne, skutecznie ukrywają przez użytkownika wiele istotnych szczegółów dotyczących budowy i działania serwera MySQL. Jednym z podstawowych zadań kursu jest natomiast pokazanie istoty baz danych i systemów służących do ich zarządzania. Oczywiście po zapoznaniu się z „filozofią działania” MySQL-a zachęcamy do korzystania ze wspomnianych tu programów, gdyż z ich pomocą praca z MySQL-em staje się łatwiejsza, przyjemniejsza i bardziej efektywna.

1.2. Nakładki graficzne *MySQL Query Browser* oraz *MySQL Administrator*

Producent serwera MySQL stworzył dwa narzędzia graficzne ułatwiające zarządzanie i bieżącą pracę z systemem. Są to: *MySQL Administrator* oraz *MySQL Query Browser*³. Zgodnie z uwagami z poprzedniego akapitu, mimo iż nie będziemy tych narzędzi używali podczas kursu, zachęcamy do samodzielnego ich zainstalowania i wypróbowania. Programy instalacyjne można pobrać ze strony www.mysql.org.

Po zainstalowaniu w środowisku Windows oba programy dostępne są bezpośrednio z menu systemowego. Jeżeli nie można ich tam znaleźć, spytaj się prowadzącego jak do nich dotrzeć.

Pierwszego programu (*MySQL Administrator*) raczej na początku nie będziesz potrzebował. Służy on, jak wskazuje nazwa, do *administrowania* serwerem. Aby to robić świadomie, wymagana jest jednak dość duża wiedza o szczegółach budowy i działania serwera MySQL. Podczas tego kursu tylko w bardzo ograniczonym zakresie będziemy poznawali te zagadnienia.

Natomiast drugi program (*MySQL Query Browser*) warto sobie uruchomić i zapoznać się z jego możliwościami. Ma on w zasadzie podobną funkcjonalność jak *MySQL Monitor*, jednak z racji jego „okienkowości” może on na początku być bardziej przyjazny dla początkującego użytkownika.

Warto zwrócić uwagę na bardzo przejrzyste napisane i łatwo dostępne pliki pomocy. W prawym dolnym rogu znajduje się panel, gdzie mamy bardzo szybki dostęp do materiałów referencyjnych dotyczących języka SQL (zakładka *Syntax*) oraz wszystkich dostępnych funkcji, operatorów itd. (zakładka *Functions*). Na uwagę zasługuje też funkcjonalny moduł zapamiętujący poprzednio wydawane polecenia (górny panel z prawej strony, zakładka *History*). Reszta opcji jest na tyle intuicyjna, że z łatwością sam szybko je poznasz metodą ”kliknij i sprawdź”.

Na koniec wyraźnie jeszcze raz podkreślmy: *we wszystkich instrukcjach zakładamy, że praca odbywa się z wykorzystaniem konsoli tekstowej MySQL Monitor, która jest integralną częścią serwera MySQL i jest zawsze dostępna (instaluje się automatycznie w czasie instalowania serwera). To środowisko przyjmujemy za obowiązujące. Wszystkie bez wyjątku przykłady zamieszczone w instrukcjach wykonano właśnie w tym środowisku. Jeżeli jednak zechcesz pracować w środowisku okienkowym, możesz tak oczywiście robić i nie będziemy z tego powodu robili problemów.*

1.3. Literatura

Podaną na końcu instrukcji listę publikacji na temat MySQL należy traktować jako subiektywny i bardzo niekompletny wybór autora. Możesz wybrać sobie takie źródło wiedzy, które Ci najbardziej odpowiada. Pozycję [3] należy jednak potraktować szczególnie. Jest to najbardziej aktualne i kompletne opracowanie i można tam znaleźć wyczerpujące informacje niemal na każdy temat związany z MySQL-em. Jest to tekst w języku angielskim.

³We wcześniejszych wersjach oprogramowanie to było udostępniane oddzielnie. Obecnie (połowa roku 2007) rozprowadzane jest w postaci jednego programu o nazwie *MySQL GUI Tools*.

2. Rozpoczęcie pracy w środowisku MySQL

2.1. Logowanie i kończenie pracy

Aby rozpocząć pracę z systemem MySQL należy połączyć się z serwerem. Używamy do tego programu klienckiego o nazwie *MySQL Monitor* (program *mysql*, który zwykle znajduje się w podkatalogu *bin* katalogu domowego serwera). Oczywiście warunkiem koniecznym jest, aby serwer był prawidłowo uruchomiony i skonfigurowany⁴. Zwykle może to robić tylko administrator danego komputera lub ktoś, kto posiada odpowiednie do tego uprawnienia. Gdy na ekranie otrzymamy komunikat podobny do poniższego oznacza to, że serwer MySQL nie działa i dalsze próby skomunikowania się z nim są bezcelowe:

```
shell> mysql
ERROR 2003 (HY000): Can't connect to MySQL server on 'localhost' (10061)
```

Aby prawidłowo połączyć się z serwerem MySQL musimy znać nazwę użytkownika, hasło oraz nazwę komputera, na którym działa serwer MySQL⁵. Gdy nie podamy tych danych⁶ system odmówi nam połączenia:

```
shell> mysql
ERROR 1045 (28000): Access denied for user 'ODBC'@'localhost'
(using password: NO)
```

Uwaga: w systemie *LINUX* zamiast `'ODBC'@'localhost'` zobaczymy `'xxxx'@'localhost'`, gdzie `xxxx` będzie nazwą aktualnie zalogowanego użytkownika.

Spróbujmy więc ponownie, tym razem podając poprawną nazwę komputera, nazwę użytkownika i jego hasło:

```
shell> mysql -h localhost -u root -pmysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 5.0.16

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

mysql>
```

Tym razem udało się połączyć z serwerem. Uruchomiliśmy właśnie program klienta, który przedstawia nam się tekstem powitalnym oraz wyświetla znak zachęty (`mysql>`). Podaliśmy nazwę komputera (tu: `localhost`), nazwę użytkownika (tu: `root` – jest to użytkownik z uprawnieniami administratora!) oraz hasło (tu: `mysql`). Można również użyć bardziej opisowej formy. Poniższe polecenie jest więc równoważne temu, które podano wyżej:

⁴W systemie Windows zalecanym sposobem uruchamiania serwera MySQL jest zainstalowanie go jako usługi. Windows będzie wówczas automatycznie uruchamiać i zatrzymywać serwer podczas rozruchu i zamykania systemu. W systemie UNIX serwer MySQL (program o nazwie *mysqld*) uruchamiany jest jako tzw. demon. System można tak skonfigurować, aby demon ten uruchamiał się automatycznie podczas rozruchu systemu.

⁵Server MySQL może zostać zainstalowany i skonfigurowany w taki sposób, że będą istniały tzw. konta anonimowe, które umożliwiają zalogowanie się bez potrzeby podawania nazwy użytkownika oraz hasła. Sytuacja taka stwarza poważne zagrożenia dla bezpieczeństwa. Administrator (lub Ty) powinien jak najszybciej takie „otwarte” konta usunąć. My zakładamy, że w naszych instalacjach MySQL-a takie konta nie istnieją.

⁶Mogą one też być zapisane „na sztywno” w pliku konfiguracyjnym o standardowej nazwie *my.ini* lub *my.cnf*. Wówczas nie trzeba ich podawać jawnie. Jest to jednak sytuacja zupełnie inna niż ta pokazana w poprzednim przypisie.

```
mysql --host=localhost --user=root --password=mysql
```

Zwróćmy uwagę, że po przełączniku `-p` nie ma spacji. Spacji nie ma też po znaku `=` w przełączniku `--password`.

Ponieważ łączymy się do serwera na maszynie lokalnej, więc nie musimy podawać jawnie przełącznika `-h` oraz nazwy komputera. Ponadto lepiej jest nie podawać hasła bezpośrednio podczas wpisywania parametrów dla programu *mysql*. Lepiej więc będzie napisać tak jak poniżej. Zostaniemy wówczas poproszeni o podanie hasła (wpisywane hasło nie pojawia się na ekranie; zamiast niego widać tylko gwiazdki):

```
shell> mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11 to server version: 5.0.16

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

mysql>
```

Gdy któryś z wymaganych parametrów (np. hasło) zostanie podany błędnie system odmówi połączenia do serwera MySQL. Przykładowe komunikaty, jakie możemy wówczas zobaczyć na ekranie pokazano niżej:

```
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
ERROR 2005 (HY000): Unknown MySQL server host 'localhosta' (11001)
```

Aby zakończyć pracę z serwerem MySQL powinniśmy wydać polecenie `QUIT` (wielkość liter nie jest istotna) lub jego skróconą wersję `\q`:

```
mysql> quit
Bye
```

Pamiętajmy, że zakończyliśmy tylko *lokalną sesję* z serwerem MySQL. Sam serwer oczywiście nadal pracuje i być może obsługuje w tym momencie innych użytkowników (MySQL jest serwerem wielodostępnym). Aby go zamknąć należy wydać inne polecenie⁷ (może to zrobić tylko administrator).

2.2. Przykładowe polecenia

Będąc już podłączonym do serwera MySQL możemy wydać kilka poleceń, które po pierwsze upewnią nas, że całość działa poprawnie a po drugie pozwolą nieco dokładniej poznać system MySQL. Postaraj się zrozumieć co wykonują poszczególne polecenia.

Zwróć uwagę, że każde z nich jest zakończone średnikiem (za wyjątkiem dwóch pierwszych, gdyż są to polecenia programu *mysql*). Wielkość liter nie ma znaczenia, jednak sugerujemy, aby wszystkie słowa kluczowe języka SQL oraz polecenia programu *mysql* wpisywać dużymi literami a pozostałe małymi (lub też odwrotnie. Najważniejsze jest w sumie konsekwentne trzymanie się cały czas tej samej konwencji).

W ostatnim poleceniu widać znaki `->`. Program *mysql* wykona wpisane polecenie dopiero po napotkaniu średnika (średnik pełni rolę *separatora* instrukcji). Każde więc wciśnięcie klawisza

⁷`mysqladmin -h nazwa-komputera -u uzytkownik -phaslo shutdown`

Enter przenosi nas do linii poniżej i umożliwia kontynuowanie wpisywania polecenia. Znaki `->` wyświetlane są automatycznie i mają one przypominać nam, że wpisywanie polecenie jeszcze się nie zakończyło. Taka funkcjonalność przydaje się wówczas, gdy polecenia są długie i nie mieszczą się w jednej linii.

```
mysql> help
mysql> status
mysql> show databases;
mysql> SELECT database();
mysql> use mysql;
mysql> show tables;
mysql> SELECT current_user();
mysql> SELECT user, host, password FROM user;
mysql> SELECT current_date();
mysql> SELECT VERSION(), CURRENT_DATE;
mysql> select version(), current_date;
mysql> SeLeCt vErSiOn(), current_DATE;
mysql> SELECT
    -> user()
    -> ,
    -> current_date;
```

2.3. „Bezpieczne” wykonywanie ćwiczeń

Jak już wspomniano wcześniej większość ćwiczeń wykonywana będzie na lokalnych instalacjach MySQL-a, gdzie będziesz posiadać pełne uprawnienia do bazy danych (będziesz znał hasło użytkownika `root`). Jednak na początku poznawania MySQL-a zdecydowanie bezpieczniej będzie pracować na koncie użytkownika, który będzie posiadał zdecydowanie mniej uprawnień i dzięki temu trudniej będzie przypadkowo uszkodzić serwer MySQL. Dlatego też zakładamy, że wszystkie dalsze ćwiczenia będą wykonywane po zalogowaniu się na konto użytkownika `ulab` (hasło `hlab`). Podczas ćwiczeń będziemy używali bazy o nazwie `blab`. Gdy baza i użytkownik nie istnieją trzeba je utworzyć. Jak to zrobić pokazano w kolejnym podrozdziale (2.4).

Rozpoczynając wykonywanie dalszych ćwiczeń powinniśmy więc albo od razu podczas łączenia się do bazy MySQL wybrać bazę `blab` (przypominamy, że po przełączniku `-p` jest spacja. Gdyby jej nie było, to fraza `blab` zostałaby potraktowana jako hasło:

```
shell> mysql -u ulab -p blab
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 5.0.16-nt

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

mysql>
```

Bazę danych można wybrać też zaraz po udanym logowaniu (polecenie `use blab`):

```
shell> mysql -u ulab -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5 to server version: 5.0.16-nt

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

```
mysql> use blab
Database changed
```

Aby upewnić się, jaki użytkownik jest zalogowany oraz do jakiej bazy jest on podłączony wykonaj dwa poniższe polecenia. Powinieneś na ekranie zobaczyć taki wynik:

```
mysql> SELECT DATABASE();
+-----+
| database() |
+-----+
| blab      |
+-----+
1 row in set (0.00 sec)
```

oraz

```
mysql> SELECT CURRENT_USER();
+-----+
| current_user() |
+-----+
| ulab@localhost |
+-----+
1 row in set (0.00 sec)
```

Użytkownik `ulab` ma taki zestaw uprawnień, który daje mu nieograniczony dostęp *tylko* do obiektów (głównie chodzi o tabele) w bazie `blab`. Próba przełączenia się na inną bazę danych zakończy się niepowodzeniem:

```
mysql> use mysql
ERROR 1044 (42000): Access denied for user 'ulab'@'localhost' to database 'mysql'
```

2.4. Tworzenie nowej bazy danych i nowego użytkownika

W tym miejscu pokażemy jak utworzyć nową bazę `blab` oraz nowego użytkownika o nazwie `ulab` z hasłem `hlab`. Polecenia te podajemy bez szczegółowego ich objaśniania. Czynimy tak, abyś mógł na swoim domowym komputerze wykonać analogiczne ćwiczenia jak te powyżej. Tworzenie oraz kasowanie baz danych oraz użytkowników to czynność zarezerwowana *wyłącznie* dla administratora MySQL-a. Trzeba to robić *bardzo ostrożnie*, gdyż bardzo łatwo jest spowodować w bazie danych ogromne spustoszenia! Aby wykonać poniższe ćwiczenia musisz zalogować się na użytkownika `root`, czyli:

```
shell> mysql -h localhost -u root -p
```

Dla pewności najpierw wydamy polecenia kasujące istniejącą bazę `blab` oraz użytkownika `ulab`⁸ (gdyby okazało się, że w Twojej instalacji MySQL-a taka baza i taki użytkownik już istnieją). Zauważmy, że nazwa użytkownika oraz nazwa komputera (`ulab`, `localhost` oraz `%`) są umieszczone w apostrofach. Apostrofy są konieczne w zasadzie tylko wtedy, gdy nazwy zawierają znaki specjalne, które są w szczególności sposób traktowane przez system operacyjny. W naszym przypadku można by apostrofy pominąć, jednak ich obecność w niczym nie przeszkadza.

⁸W wersjach serwera MySQL *wcześniejszych* niż 5.0.1 kasowanie użytkownika odbywa się w nieco bardziej skomplikowany sposób. Szczegóły patrz [3].

```
mysql> DROP DATABASE blab;
Query OK, 0 rows affected (0.00 sec)

mysql> DROP USER 'ulab'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> DROP USER 'ulab'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE DATABASE blab;
Query OK, 1 row affected (0.03 sec)
```

Gdybyś w czasie wykonywania powyższych poleceń zobaczył poniższe komunikaty o błędach, będzie to oznaczało, że baza danych `blab` oraz użytkownik `ulab` jeszcze nie istnieją w systemie (więc operacja kasowanie czegoś nieistniejącego nie może się udać). Z taką sytuacją spotkasz się, gdy polecenia uruchamiasz po raz pierwszy:

```
mysql> DROP DATABASE blab;
ERROR 1008 (HY000): Can't drop database 'blab'; database doesn't exist
mysql> DROP USER 'ulab'@'localhost';
ERROR 1396 (HY000): Operation DROP USER failed for 'ulab'@'localhost'
mysql> DROP USER 'ulab'@'%';
ERROR 1396 (HY000): Operation DROP USER failed for 'ulab'@'%'
```

Teraz możemy wpisać polecenia tworzące nową bazę danych oraz konto użytkownika. Uwaga: w zasadzie musimy utworzyć dwa konta. Jedno (`'ulab'@'localhost'`) będzie używane *tylko* do łączenia się lokalnego komputera. Drugie (`'ulab'@'%'`) można używać do łączenia się z dowolnego komputera. Nie wnikając na razie w szczegóły zapamiętaj, że musisz utworzyć oba konta.

```
mysql> GRANT ALL PRIVILEGES ON blab.* TO 'ulab'@'localhost'
-> IDENTIFIED BY 'hlab';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON blab.* TO 'ulab'@'%'
-> IDENTIFIED BY 'hlab';
Query OK, 0 rows affected (0.00 sec)
```

Po utworzeniu kont warto sprawdzić, czy wszystko jest w porządku. Wydajemy więc polecenia „sprawdzające”. Najpierw musimy jednak przełączyć na bazę „systemową” (polecenie `use mysql`). Powinniśmy zobaczyć wynik taki jak poniżej:

```
mysql> use mysql;
Database changed
mysql> SELECT host, user, password FROM user WHERE user LIKE 'ulab';
+-----+-----+-----+-----+
| host      | user | password                                     |
+-----+-----+-----+-----+
| localhost | ulab | *BD7E2F4D848AC59F76E03A900C20B94ED0902D78 |
| %         | ulab | *BD7E2F4D848AC59F76E03A900C20B94ED0902D78 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

oraz

```
mysql> SELECT host, db, user FROM db;
+-----+-----+-----+
| host      | db    | user  |
+-----+-----+-----+
| %         | blab  | ulab  |
| localhost | blab  | ulab  |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

Aby za każdym razem nie wpisywać mozolnie dość długich poleceń można wszystkie je zapisać w dowolnym pliku tekstowym i uruchomić w tzw. trybie wsadowym. Niech więc nasz plik wygląda jak poniżej:

```
DROP DATABASE blab;
DROP USER 'ulab'@'localhost';
DROP USER 'ulab'@'%';
CREATE DATABASE blab;
GRANT ALL PRIVILEGES ON blab.* TO 'ulab'@'localhost' IDENTIFIED BY 'hlab';
GRANT ALL PRIVILEGES ON blab.* TO 'ulab'@'%' IDENTIFIED BY 'hlab';
USE mysql;
SELECT host, user, password FROM user WHERE user LIKE 'ulab';
SELECT host, db, user FROM db;
```

Następnie z poziomu programu klienckiego *mysql* wydajemy polecenie *source* (lub jego skróconą wersję *\.* – ukośnik oraz kropka), które odczyta ten pliku i wykona zapisane tam polecenia (ścieżkę dostępu oraz nazwę pliku musisz sam ustalić). Gdy wszystkie polecenia wykonają się poprawnie, na ekranie zobaczysz wynik podobny do poniższego:

```
mysql> source c:\temp\mysql_01.sql
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.00 sec)

Database changed
+-----+-----+-----+
| host      | user          | password          |
+-----+-----+-----+
| localhost | ulab         | *BD7E2F4D848AC59F76E03A900C20B94ED0902D78 |
| %         | ulab         | *BD7E2F4D848AC59F76E03A900C20B94ED0902D78 |
+-----+-----+-----+
3 rows in set (0.01 sec)

+-----+-----+-----+
| host      | db    | user  |
+-----+-----+-----+
| %         | blab  | ulab  |
| localhost | blab  | ulab  |
+-----+-----+-----+
2 rows in set (0.00 sec)
```


2.5. Zmiana hasła użytkownika

Hasła użytkowników oraz adekwatny do wykonywanych zadań zestaw uprawnień są podstawą bezpiecznej pracy z systemem MySQL (oraz z każdą inną bazą danych). W ramach bieżącej instrukcji nie będziemy poznawali zagadnień (dość trudnych) dotyczących systemu uprawnień w MySQL-u. Poznamy natomiast polecenia służące do zmiany hasła użytkownika.

Na początek zobaczmy na jakiego użytkownika jesteśmy zalogowani:

```
mysql> SELECT USER();
+-----+
| user()          |
+-----+
| ulab@localhost  |
+-----+
1 row in set (0.00 sec)
```

Następnie zmienimy hasło temu użytkownikowi:

```
mysql> SET PASSWORD = PASSWORD('nowe-haslo');
Query OK, 0 rows affected (0.00 sec)
```

lub też używając nieco „nadmiarowej” składni:

```
mysql> SET PASSWORD FOR 'ulab'@'localhost'= PASSWORD('nowe-haslo');
Query OK, 0 rows affected (0.00 sec)
```

Od tego momentu logując się do bazy MySQL na konto ulab musimy podawać hasło nowe-haslo.

Zwykły użytkownik (nie administrator) może zmienić hasło tylko sobie. Administrator może zmieniać natomiast hasła dowolnych użytkowników. Robi to w następujący sposób:

```
mysql> SET PASSWORD FOR 'ulab'@'localhost'= PASSWORD('nowe-haslo');
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> SET PASSWORD FOR 'ulab'@'%'= PASSWORD('mysql');
Query OK, 0 rows affected (0.00 sec)
```

Jeżeli zwykły użytkownik (nie administrator) będzie chciał zmienić nie swoje hasło otrzyma następujący komunikat o błędzie:

```
mysql> SET PASSWORD FOR 'root'@'localhost'= PASSWORD('nowe-haslo-root');
ERROR 1044 (42000): Access denied for user 'ulab'@'localhost' to database 'mysql'
```

Literatura

- [1] Lech Banachowski (tłum.). *SQL. Język relacyjnych baz danych*. WNT Warszawa, 1995.
- [2] Paul Dubios. *MySQL. Podręcznik administratora*. Wydawnictwo HELION, 2005.
- [3] *MySQL 5.0 Reference Manual*, 2005. (jest to najbardziej aktualne opracowanie na temat bazy MySQL stworzone i na bieżąco aktualizowane przez jej twórców. Książka dostępna w wersji elektronicznej pod adresem <http://dev.mysql.com/doc/>).
- [4] Richard Stones and Neil Matthew. *Od podstaw. Bazy danych i MySQL*. Wydawnictwo HELION, 2003.
- [5] Luke Welling and Laura Thomson. *MySQL. Podstawy*. Wydawnictwo HELION, 2005.