

1. Tworzenie nowego użytkownika.

```
CREATE USER nazwa_uzytkownika;
```

```
CREATE USER nazwa_uzytkownika IDENTIFIED BY 'haslo';
```

```
CREATE USER nazwa_uzytkownika IDENTIFIED BY PASSWORD 'tajne_haslo';
```

● Tworzenie i usuwanie użytkowników w bazie MySQL

W celu tworzenia nowych użytkowników w bazie MySQL wykorzystujemy nieco bardziej rozbudowaną postać polecenia `CREATE USER`:

```
CREATE USER 'uzytkownik'@'localhost' IDENTIFIED BY 'haslo';
```

Hasła użytkowników w tej bazie są przechowywane w postaci zaszyfrowanej. Nie ma możliwości podejrzenia hasła w postaci jawnej. Chcąc przenieść użytkownika na inną maszynę (serwer lub bazę) lub przekazać prośbę o utworzenie konta, można wykorzystać inną postać tego polecenia:

```
CREATE USER 'uzytkownik'@'localhost'
IDENTIFIED BY PASSWORD 'zaszyfrowane_haslo';
```

Zaszyfrowane hasło można uzyskać, wydając polecenie (rys. 8.4):

```
SELECT PASSWORD('haslo');
```

Konstrukcja polecenia `uzytkownik@maszyna` (czyt. użytkownik *at* maszyna) pozwala na nadawanie różnych uprawnień użytkownikom w zależności od miejsca logowania się do bazy. Dzięki temu możemy zezwolić na dostęp do danych i ich modyfikację jedynie osobom logującym się z lokalnej maszyny. Ta konstrukcja oznacza też, że musimy jawnie pozwolić użytkownikom na logowanie się z konkretnych domen (komputerów położonych w danej domenie). Jeżeli chcemy,

Ćwiczenie

Wykonaj następujące polecenia:

- Zaloguj się na „superużytkownika” -> `mysql -u root`
- Wyświetl dostępne dla niego bazy danych -> `show databases;`
- Wybierz bazę danych o nazwie `mysql` -> `use mysql;`
- Wyświetl zawartość pól `user`, `host`, `password` w tabeli `user` -> `SELECT user, host, password FROM user;`
- Utwórz użytkownika o nazwie `gosc1` na `localhost`, z hasłem `'12345'` dodając wpis do tabeli `user`.

```
# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.6.21 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SELECT PASSWORD('hasło');
+-----+
| PASSWORD('hasło') |
+-----+
| *05A8A1A73083F816772592F13D11C8AA5CCD9681 |
+-----+
1 row in set (0.00 sec)
```

Rys. 8.4. Otrzymywanie zaszyfrowanego hasła w kliencie mysql w MS Windows

aby użytkownik mógł logować się z dowolnej maszyny w wybranej domenie, to używamy symbolu %, np.:

```
CREATE USER użytkownik@'%.mojadomena.pl' IDENTIFIED BY
'hasło';
```

Użycie apostrofów nie jest konieczne, o ile w nazwie maszyny (domeny) bądź w loginie użytkownika nie występują znaki specjalne, takie jak – czy %. W szczególności użycie jedynie symbolu % oznacza dowolny komputer:

```
CREATE USER 'użytkownik'@'%' IDENTIFIED BY 'hasło';
```

Ten sam efekt uzyskamy przy pominięciu symbolu @ i nazwy maszyny, pozostawiając sam login nowego użytkownika.

W bazie MySQL można dodać użytkownika przez ręczne dopisanie go do tabeli user. Musimy pamiętać przy tym o użyciu wbudowanej funkcji PASSWORD szyfrującej podane hasło:

```
INSERT INTO user (host, user, password) VALUES
('localhost', 'użytkownik', PASSWORD('hasło'));
```

Dodawanie użytkowników za pomocą panelu phpMyAdmin jest intuicyjnie proste. Wystarczy jedynie wypełnić odpowiednie pola. Można również zaznaczyć odpowiednie uprawnienia, kierując się ich nazwami (rys. 8.5).

Wykonanie w kliencie WWW pakietu XAMPP operacji utworzenia w bazie *Komis* użytkownika o nazwie *kupiec* z pełnymi uprawnieniami sprowadza się do wykonania następującego kodu w języku SQL:

```
CREATE USER 'kupiec'@'%' IDENTIFIED BY '***';
```

2. Usuwanie użytkownika.

```
DROP USER nazwa_uzytkownika;
```

3. Rozpoznanie nazwy bieżącego użytkownika.

```
SELECT current_user();
```

4. Zmiana hasła.

a) dla bieżącego użytkownika

```
SET PASSWORD = PASSWORD('hasło');
```

b) dla innego konta (wymaga odpowiednich uprawnień)

```
SET PASSWORD FOR nazwa_uzytkownika = PASSWORD('hasło');
```

5. Prawa dostępu do serwera.

Prawa dostępu serwera MySQL mogą być nadawane użytkownikom na poziomie:

- całego serwera,
- bazy danych (np. CREATE, ALTER DROP),
- obiektów bazy danych (np. SELECT, INSERT, UPDATE, DELETE)

Informacje na temat praw dostępu do serwera MySQL są przechowywane w tabelach słownikowych bazy danych mysql. Są to m. in. tabele: host, db, user, tables_priv, columns_priv.

- user przechowuje inf. o prawach dostępu użytkownika niezależnie od bazy danych,
- db przechowuje inf. o prawach dostępu użytkownika w zależności od bazy danych
- host przechowuje inf. w kontekście komputera, a którego łączy się użytkownik

6. Nadawanie praw.

Prawa są nadawane użytkownikom instrukcją GRANT. Składania polecenia ma postać:

```
GRANT [prawo] ON nazwa_bazy.* TO '[nazwa_uzytkownika]'@'[nazwa_hosta]' IDENTIFIED BY '[hasło]';
```

Prawa mogą być nadawane użytkownikom na różnych poziomach:

- nadawane globalnie, np. GRANT UPDATE ON *.* TO 'nazwa_uzytkownika'@'localhost';
- na poziomie domyślnej bazy danych, np.

```
GRANT UPDATE ON ksiegarnia_internetowa.* TO 'nazwa_uzytkownika'@'localhost';
```

lub

```
GRANT UPDATE ON * TO 'nazwa_uzytkownika'@'localhost';
```

- na poziomie obiektów bazy danych, np. do wybranej tabeli

GRANT INSERT ON ksiegarnia_internetowa.ksiazki TO 'nazwa_uzytkownika'@'localhost';

- na poziomie wybranych kolumn tabeli

GRANT UPDATE(ksiazki.tytul), INSERT ON ksiegarnia_internetowa.ksiazki TO 'nazwa_uzytkownika'@'localhost';

Do przyznania wszystkich przywilejów służy w poleceniu GRANT opcja ALL PRIVILEGES lub ALL.

Przykład 6.24

```
GRANT ALL PRIVILEGES ON *.* TO Marcin;
```

Polecenie GRANT może zawierać dodatkowe klauzule:

- MAX_QUERIES_PER_HOUR — ogranicza liczbę zapytań;
- MAX_UPDATES_PER_HOUR — ogranicza liczbę zmian wprowadzanych do bazy;
- MAX_CONNECTIONS_PER_HOUR — ogranicza liczbę logowań użytkownika w ciągu godziny;
- MAX_USER_CONNECTIONS — ogranicza liczbę jednoczesnych połączeń uzyskiwanych z jednego konta.

Przykład 6.25

```
GRANT USAGE ON *.* TO Marcin WITH MAX_QUERIES_PER_HOUR 1;
```

Prawo USAGE oznacza, że użytkownikowi nie zostały nadane żadne prawa.

7. Odbieranie praw.

Odbieranie praw

Instrukcją REVOKE można usunąć wcześniej nadane prawa.

Przykład 6.26

```
REVOKE UPDATE ON ksiegarnia_internetowa.Ksiazki FROM Marcin;
```

W podanym przykładzie użytkownikowi *Marcin* zostały odebrane uprawnienia do modyfikowania danych w tabeli *Ksiazki*.

Lista uprawnień

- ALL — użytkownik otrzymuje wszystkie prawa poza GRANT OPTION.
- CREATE — użytkownik może tworzyć tabele.
- SELECT — użytkownik może wyświetlać zawartość tabel.
- INSERT — użytkownik może dodawać nowe dane do tabel.
- SHOW DATABASES — użytkownik może przeglądać listy dostępnych baz danych.
- USAGE — umożliwia tworzenie użytkownika bez uprawnień.
- GRANT OPTION — użytkownik może nadawać uprawnienia innym użytkownikom.

wyszukiwania i modyfikowania danych (tab. 8.3). Podstawowe prawa dostępu są zdefiniowane w taki sam sposób w systemach Oracle, MySQL i PostgreSQL.

Tab. 8.3. Prawa dostępu do danych w tabelach dla systemów Oracle, MySQL i PostgreSQL

Nazwa uprawnień	Opis
SELECT	prawo wykonania funkcji SELECT; wymagane do wykonania funkcji UPDATE lub DELETE, gdy odwołują się do wartości kolumn
INSERT	tworzenie nowych wierszy tabeli, może być ograniczane do ustawiania nowych wartości jedynie wybranym kolumnom (pozostałe będą ustawione domyślnie)
UPDATE	modyfikacja i aktualizacja wierszy w dowolnych lub tylko wskazanych kolumnach w tabeli; bez prawa SELECT można nadać jedną globalną wartość dla całej kolumny w tabeli
DELETE	prawo do usunięcia wybranego wiersza (rekordu) z tabeli; każde bardziej złożone użycie funkcji DELETE wymaga ukrytego użycia funkcji SELECT w celu odwołania się do wartości pól w kolumnach tabeli
REFERENCES	prawo do tworzenie kluczy obcych; w bazie PostgreSQL musi być nadane do tabeli zawierającej klucz obcy oraz tabeli wskazywanej; w bazie Oracle wystarczy prawo do tabeli wskazującej; w bazie MySQL obecnie nie jest wspierane
ALL	wszystkie uprawnienia

Upewnienia mogą być na poziomie obiektów w bazie danych lub w schemacie. W tabeli 8.4 zebrano typy dostępów do obiektów wraz

Tab. 8.4. Prawa dostępu do obiektów w schematach i bazach dla systemów Oracle, MySQL i PostgreSQL

Nazwa uprawnień	Opis
CREATE	prawo do tworzenia odpowiednich obiektów, jak schematy, tabele, kolumny, indeksy; w bazie Oracle po słowie CREATE należy jawnie podać typy obiektów; w bazie MySQL samo słowo CREATE daje prawo do tworzenia baz i użytkowników, może też być użyte z parametrami USER, VIEW; w bazie PostgreSQL – tworzenie obiektów we wskazanej bazie

cd. tab. 8.4. Prawa dostępu do obiektów w schematach i bazach dla systemów Oracle, MySQL i PostgreSQL

Nazwa uprawnień	Opis
ALTER	prawo do modyfikacji struktury obiektu; w Oracle po słowie ALTER należy jawnie podać typy obiektów, np. ALTER TABLE, ALTER USER; w MySQL wymaga dodatkowo uprawnień CREATE i INSERT; w PostgreSQL nie występuje jako osobne uprawnienie
DROP	prawo do usuwania baz danych, tabel i perspektyw; w bazie Oracle po słowie DROP należy jawnie podać typy obiektów; w bazie MySQL wystarczy użycie samego słowa DROP; w bazie PostgreSQL nie występuje jako osobne uprawnienie
CONNECT	prawo do łączenia się z bazą danych; nie występuje w MySQL
ALL	wszystkie uprawnienia

z ich opisami. Rozwiązania dotyczące zarządzania użytkownikami, które oferują producenci systemów zarządzania bazami danych, ciągle się zmieniają, dlatego zawsze musimy sprawdzić, czy dane polecenie jest wspierane przez wersję bazy, w której pracujemy. Dokumentacja MySQL oraz PostgreSQL pozwala na przełączanie się między opisami danego polecenia w różnych wersjach SZBD dzięki wygodnym linkom nawigacyjnym (rys. 8.11). Dokumentacja Oracle niestety nie oferuje takich możliwości.