

Laboratoria z administracji MySQL na Ubuntu Server

MODUŁ 1: Instalacja i podstawowa konfiguracja MySQL

Krok 1: Przygotowanie do pracy

Sprawdzamy adres IP na systemie Windows poleceniem:

```
ipconfig
```

Otrzymamy coś jak poniżej:

Wireless LAN adapter Wi-Fi:

```
Connection-specific DNS Suffix  . :  
Link-local IPv6 Address . . . . . : fe80::ae7b:9e6b:7739:6258%18  
IPv4 Address. . . . . : 192.168.100.27  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . : 192.168.100.1
```

Przechodzimy do systemu Ubuntu i wydajemy polecenie, np.:

```
ip a
```

Otrzymamy coś jak poniżej:

```
enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel  
state UP group default qlen 1000 link/ether 08:00:27:12:34:56 brd
```

```
ff:ff:ff:ff:ff:ff inet 192.168.199.28/24 brd 192.168.100.255 scope  
global dynamic enp0s3 valid_lft 86399sec preferred_lft 86399sec
```

Logujemy się poprzez SSH na systemie Windows do naszego systemu Linux przy pomocy polecenia:

```
ssh nazwauzytkownika@adresip
```

Krok 2: Aktualizacja systemu

Zanim zainstalujemy MySQL, **zaktualizujemy system**, aby mieć najnowsze pakiety:

```
sudo apt update
```

Co robi to polecenie?

- `apt update` – sprawdza nowe wersje pakietów

```
sudo apt upgrade
```

Co robi to polecenie?

- `apt upgrade` – instaluje aktualizacje

Krok 3: Instalacja MySQL

Teraz zainstalujemy MySQL:

```
sudo apt install mysql-server -y
```

Co robi to polecenie?

- Pobiera i instaluje **serwer MySQL**

Sprawdzenie wersji MySQL:

```
mysql --version
```

Krok 4: Sprawdzenie statusu MySQL

Po instalacji MySQL powinien działać automatycznie. Sprawdźmy to:

```
sudo systemctl status mysql
```

Jeśli wyświetli się:

- `mysql.service` - MySQL Community Server
Loaded: loaded (/lib/systemd/system/mysql.service; enabled;
vendor preset: enabled)
Active: active (running)

To znaczy, że MySQL działa.

Jeśli MySQL nie działa, uruchom go ręcznie:

```
sudo systemctl start mysql
```

Aby uruchamiał się automatycznie po restarcie serwera:

```
sudo systemctl enable mysql
```

Ćwiczenie:

- Zatrzymaj MySQL
- Sprawdź, czy MySQL działa
- Ponownie uruchom MySQL

Krok 5: Konfiguracja bezpieczeństwa

MySQL zawiera domyślnie niebezpieczne ustawienia (np. anonimowi użytkownicy).

Uruchamiamy kreatora konfiguracji:

```
sudo mysql_secure_installation
```

Pojawią się pytania:

1. **Czy chcesz skonfigurować walidację haseł?** → **ENTER** (można pominąć)
2. **Ustawić hasło roota?** → **Tak** (ustaw mocne hasło!)
3. **Usunąć anonimowych użytkowników?** → **Tak**
4. **Zablokować zdalne logowanie roota?** → **Tak**
5. **Usunąć testową bazę danych?** → **Tak**
6. **Przeładować uprawnienia?** → **Tak**

Ćwiczenie:

Spróbuj połączyć się do MySQL BEZ hasła:

```
mysql -u root
```

Jeśli dostaniesz błąd, oznacza to, że konfiguracja działa poprawnie!

Krok 6: Logowanie do MySQL

Teraz zalogujmy się do MySQL:

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

(Wpisz hasło, które ustawiłeś wcześniej!)

Aby sprawdzić, czy MySQL działa i jaka jest wersja:

```
SELECT version();
```

Aby wyjść:

```
EXIT; lub quit
```

Ćwiczenie:

- Sprawdź aktualnych użytkowników MySQL (znajdują się w tabeli `mysql.user`):

```
SELECT user, host FROM mysql.user;
```

```
+-----+-----+
| User           | Host       |
+-----+-----+
| debian-sys-maint | localhost |
| mysql.infoschema | localhost |
| mysql.session   | localhost |
| mysql.sys       | localhost |
| root            | localhost |
+-----+-----+
```

MODUŁ 2: Zarządzanie użytkownikami i uprawnieniami

Krok 1: Tworzenie nowego użytkownika

Wchodzimy do MySQL:

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

Tworzymy użytkownika **faber** z hasłem **passw0rd**:

```
CREATE USER 'faber'@'localhost' IDENTIFIED BY 'passw0rd';
```

Sprawdź czy użytkownik został poprawnie dodany:

```
SELECT User FROM mysql.user;
```

Ćwiczenie:

- Stwórz użytkownika devuser z hasłem DevPass123.

Krok 2: Zmiana hasła użytkownika

Zmieńmy hasło użytkownika **faber** na **admin123**:

```
ALTER USER 'faber'@'localhost' IDENTIFIED BY 'admin123';
```

Krok 3: Nadanie uprawnień użytkownikowi

Nadajemy użytkownikowi faber uprawnienia do wszystkiego:

```
GRANT ALL PRIVILEGES ON *.* TO 'faber'@'localhost' WITH GRANT OPTION;
```

W tym poleceniu słowo kluczowe **GRANT** oznacza nadawanie uprawnień użytkownikowi na serwerze MySQL

ALL PRIVILEGES oznacza, że nadajemy mu prawa do wykonywania na bazach danych takich poleceń jak **SELECT**, **INSERT**, **DROP**, **ALTER**, **CREATE**, **UPDATE**.

ON *.* - oznacza wszystkie bazy danych (*) i wszystkie tabele (*) na serwerze.

Gdybyśmy mieli przypadek testowa.* wtedy nadajemy uprawnienia do wszystkich operacji jedynie na bazie o nazwie testowa

TO 'faber'@'localhost' oznacza, że przypisujemy to tylko do użytkownika faber

'faber' to nazwa użytkownika, dla którego nadajemy uprawnienia.

'localhost' oznacza, że użytkownik admin może logować się tylko z lokalnej maszyny (serwera, na którym działa MySQL).

WITH GRANT OPTION pozwala nadawać użytkownikowi te same uprawnienia innym użytkownikom

Krok 4: Usuwanie użytkownika

```
DROP USER 'faber'@'localhost';
```

Krok 5: Tworzenie użytkownika dla zdalnego dostępu

Jeśli chcesz, aby użytkownik mógł logować się **zdalnie**, musisz dodać dostęp na dowolny adres (%):

```
CREATE USER 'remote_admin'@'%' IDENTIFIED BY 'SecurePassword'; GRANT ALL PRIVILEGES ON *.* TO 'remote_admin'@'%';
```

Edytuj plik konfiguracyjny:

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

Znajdź linię:

```
bind-address = 127.0.0.1
```

Zmień na:

```
bind-address = 0.0.0.0
```

Zapisz (CTRL+X → Y → ENTER) i uruchom ponownie MySQL:

```
sudo systemctl restart mysql
```

MODUŁ 3: Tworzenie i zarządzanie bazami danych

Krok 1: Tworzenie nowej bazy danych

1. Zaloguj się do MySQL:

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

Utwórz nową bazę danych:

```
CREATE DATABASE firma_db;
```

Sprawdzenie, czy baza została utworzona:

```
SHOW DATABASES;
```

Powinna pojawić się lista baz, m.in.:

```
+-----+
| Database          |
+-----+
| firma_db          |
| information_schema |
| mysql             |
| performance_schema |
| sys               |
+-----+
```

Ćwiczenie:

- Stwórz bazę danych `testowa_db`.
- Sprawdź listę baz danych.

Krok 2: Usunięcie bazy danych

Jeśli baza nie jest już potrzebna, można ją usunąć:

```
DROP DATABASE testowa_db;
```

Ostrzeżenie: Usunięcie bazy jest nieodwracalne!

Krok 3: Tworzenie użytkownika z dostępem do konkretnej bazy

Jeśli nie chcemy, aby użytkownik miał dostęp do wszystkiego, możemy nadać mu uprawnienia tylko do jednej bazy.

```
CREATE USER 'developer'@'localhost' IDENTIFIED BY  
'BezpieczneHaslo123';
```

```
GRANT ALL PRIVILEGES ON firma_db.* TO 'developer'@'localhost';  
FLUSH PRIVILEGES;
```

Sprawdzenie uprawnień użytkownika:

```
SHOW GRANTS FOR 'developer'@'localhost';
```

Ćwiczenie:

- Stwórz użytkownika tester, bazę danych test_db oraz przypisz użytkownikowi dostęp tylko do test_db.

MODUŁ 4: Tworzenie tabel i operacje CRUD

Aby pracować z bazą `firma_db`, wpisujemy:

```
USE firma_db;
```

Sprawdzenie, na jakiej bazie pracujemy:

```
SELECT DATABASE();
```

Tworzymy tabelę pracownicy:

```
CREATE TABLE pracownicy (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    imie VARCHAR(50),  
    nazwisko VARCHAR(50),  
    stanowisko VARCHAR(50),  
    pensja DECIMAL(10,2)  
);
```

Sprawdzenie tabel w bazie:

```
SHOW TABLES;
```

Podgląd struktury tabeli:

```
DESC pracownicy;
```

Ćwiczenie:

- Utwórz tabelę klienci z polami: id, imie, nazwisko, email.

Krok 3: Wstawianie danych

Dodajmy pracownika:

```
INSERT INTO pracownicy (imie, nazwisko, stanowisko, pensja)  
VALUES ('Jan', 'Kowalski', 'Administrator', 5000.00);
```

Sprawdzenie danych w tabeli:

```
SELECT * FROM pracownicy;
```

Ćwiczenie:

- Dodaj trzech pracowników do tabeli pracownicy.

Krok 4: Pobieranie danych

Wszystkie rekordy:

```
SELECT * FROM pracownicy;
```

Tylko nazwiska i stanowiska:

```
SELECT nazwisko, stanowisko FROM pracownicy;
```

Ćwiczenie:

- Wyszukaj pracowników zarabiających więcej niż 3200.

Krok 5: Aktualizacja danych

Zwiększenie pensji dla pracownika o id=1:

```
UPDATE pracownicy SET pensja = 8000.00 WHERE id = 1;
```

Sprawdzenie:

```
SELECT * FROM pracownicy WHERE id = 1;
```

Ćwiczenie:

- Zmień stanowisko jednego z pracowników.

Krok 6: Usuwanie danych

Usunięcie pracownika o `id=1`:

```
DELETE FROM pracownicy WHERE id = 1;
```

Usunięcie wszystkich pracowników

```
DELETE FROM pracownicy;
```

MODUŁ 5: Backup i przywracanie danych

Krok 1: Wykonanie kopii zapasowej

Backup bazy `firma_db`:

```
sudo mysqldump -u root -p firma_db > firma_db_backup.sql
```

Backup wszystkich baz:

```
sudo mysqldump -u root -p --all-databases > all_databases_backup.sql
```

Przywrócenie bazy `firma_db`:

```
sudo mysql -u root -p firma_db < firma_db_backup.sql
```

Przywrócenie wszystkich baz:

```
sudo mysql -u root -p < all_databases_backup.sql
```