

MariaDB

Autorzy: Marcin Kostrzemiński i Michał Puchawski

Spis treści

Omówienie MariaDB.....	3
Czym jest MariaDB ?.....	3
Idea relacyjnej bazy danych.....	3
Do czego służy ?.....	3
Porównanie z innymi systemami.....	4
SQLite.....	4
PostgreSQL.....	4
MariaDB.....	5
Rozwiązanie zadania.....	5
Instalacja MariaDB na Fedorze 5.8.....	5
Zakładanie hasła administratora.....	6
Konfiguracja połączeń zdalnych.....	6
Przyznawanie uprawnień użytkownikom zdalnym.....	7
Integracja z serwerem WWW.....	7
Testowanie zdalnego połączenia.....	9
Najważniejsze pliki.....	9
Bibliografia:.....	9

Omówienie MariaDB

Czym jest MariaDB ?

MariaDB jest to system do zarządzania relacyjną bazą danych. Jest prowadzony głównie przez byłych pracowników MySQL, którzy stworzyli ten system z powodu obaw o przejęcie firmy przez Oracle Corporation .

Celem tego projektu jest współpraca ze społecznością na zasadzie open-source na licencji GPL. Kod źródłowy MariaDB jest oparty na starszych wersjach MySQL i zachowuje wysoką kompatybilność, umożliwiając zastępowanie bazy danych oraz dopasowanie do interfejsów API i poleceń.

MariaDB jest używany przez wiele korporacji m.in. ServiceNow, DBS Bank, Google, Mozilla. Kilka dystrybucji Linuxa jak Arch Linux, Manjaro czy Debian domyślnie zawiera ten system.

Idea relacyjnej bazy danych

Relacyjna baza danych to opisany i zorganizowany zbiór tabel połączonych relacjami – związkami między sobą. Ten sposób przechowywania informacji pozwala na uniknięcie redundancji oraz przeprowadzanie analiz na podstawie wielu tabel. Każda tabela składa się z rekordów (tak nazywamy pojedyncze wiersze). Poszczególne rekordy składają się z pól (komórek), przechowujących jedną daną.

Ideą relacyjnej bazy danych jest przechowywanie danych za pomocą tabel połączonych relacjami między sobą. Każda tabela ma swoje właściwości. Przykładowo osoba ma swoje imię nazwisko itp. Te elementy nazywamy atrybutami.

Zbiór atrybutów nazywamy krotką, a zbiór krotek tworzy tabelę. W modelu programowania obiektowego krotka odpowiada instancji obiektu.

Zbiór atrybutów tworzy klucz główny. Jest on unikalnym identyfikatorem dla każdej z krotek. Może składać się z jednego lub wielu pól. Najczęściej tworzona jest dodatkowa kolumna np. „ID”, w której przechowujemy numer krotki.

Tabele możemy powiązać za pomocą klucza obcego. Zazwyczaj jest to dodatkowy atrybut, w którym znajduje się klucz główny do tabeli którą chcemy powiązać. Istnieją trzy rodzaje zależności między tabelami:

- jeden do jednego
- jeden do wielu
- wiele do wielu

Do czego służy ?

MariaDB w znaczący sposób ułatwia zarządzanie danymi w bazie. Za pomocą języka SQL konstruując odpowiednie zapytanie możemy szybko wyszukać określone informacje z bazy, szczególnie w przypadku rozbudowanych baz. Dlatego bardzo często jest używana przez duże firmy. System ten również pozwala na tworzenie i zarządzanie bazą użytkowników. Ograniczając odpowiedzialność zwiększamy bezpieczeństwo naszej bazy.

Porównanie z innymi systemami

Na rynku istnieje wiele systemów rozwiązujących problem zarządzania relacyjną bazą danych. Do najpopularniejszych należą SQLite, PostgreSQL. Każdy z systemów ma swoje wady i zalety. Główne różnice:

Pamiętajmy też o komercyjnych bazach danych, np. Oracle

SQLite

Zalety

- przenośność (cała baza danych przechowywana jest w jednym pliku)
- skalowalność (duża ilość funkcji oferuje łatwą modyfikację bazy)

Wady

- brak zarządzania użytkownikami

Zastosowanie

- wbudowane aplikacje
- testowanie aplikacji

PostgreSQL

Zalety

- kod open source (a zarazem silna społeczność)
- wspierana przez inne firmy

Wady

- mało wydajny przy skomplikowanych operacjach
- słaby wybór firm oferujących zarządzanie PostgreSQL

Zastosowanie

- złożone projekty (duża funkcjonalność)
- niezawodność oraz integralność danych jest jego silną stroną

MariaDB

Zalety

- open source
- bezpieczeństwo
- szybkość
- łatwość w obsłudze (proste GUI)

Wady

- brak wbudowanych narzędzi zarządzania
- brak mocnej (jak kiedyś) kompatybilność z system MySQL

Zastosowanie

- aplikacje internetowe
- kiedy potrzebny jest wysoki poziom bezpieczeństwa

Rozwiązanie zadania

Instalacja MariaDB na Fedorze 5.8

Aby zainstalować MariaDB na Fedorze uruchamiamy polecenie

```
sudo dnf install mariadb mariadb-server
```

Uruchamiamy MariaDB

```
systemctl start mariadb
```

Jeżeli chcemy aby MariaDB było uruchamiane wraz z systemem wpisujemy

```
systemctl enable mariadb
```

Uruchamiamy bezpieczną instalację. Jeżeli wcześniej ustawiliśmy hasło roota to będziemy musieli je tutaj wpisać.

```
mysql_secure_installation
```

Po uruchomieniu instalacji będziemy mieli do wyboru kilka opcji, takich jak:

- zmiana hasła roota
- możliwość usunięcia przykładowych baz i użytkowników
- wyłączenie możliwości zdalnego łączenia się z rootem
- natychmiastowe odświeżenie dokonanych zmian

Opcjonalnie możemy sprawdzić wersję zainstalowanej MariaDB za pomocą komendy `mysql -v`

Zakładanie hasła administratora

Aby zmienić hasło roota musimy najpierw upewnić się, że usługa MariaDB jest uruchomiona. Następnie używamy polecenia, gdzie jako pass wpisujemy wybrane przez nas hasło

```
mysqladmin --user=root password "pass"
```

Alternatywnie, jeżeli root posiada już hasło musimy użyć polecenia:

```
mysqladmin --user=root -p password "pass"
```

W tym przypadku zostaniemy poproszeni o podanie poprzedniego hasła.

Konfiguracja połączeń zdalnych

Zależnie od tego jak skonfigurujemy MariaDB podczas instalacji połączenia zdalne mogą być zablokowane lub nie. Domyślnie MariaDB zezwala na połączenie tylko z hosta lokalnego, a wszystkie połączenia ze zdalnego serwera są zabronione. Co jeżeli chcielibyśmy zezwolić innym maszynom na łączenie się z naszym serwerem?

W tym przypadku musimy zedytować plik `/etc/my.cnf`, użyjemy do tego polecenia:

```
sudo vi /etc/my.cnf
```

pod fragmentem `[client-server]` dopisujemy:

```
bind-address = 0.0.0.0
```

i zapisujemy plik. Następnie używamy polecenia

```
systemctl restart mariadb
```

W ten sposób zrestartujemy MariaDB. Następnie aby upewnić się czy MariaDB nasłuchuje wszystkie adresy ip wpisujemy:

```
netstat -ant | grep 3306
```

Jeżeli wszystko przebiegło poprawnie powinniśmy otrzymać coś takiego:

```
tcp      0      0 0.0.0.0:3306      0.0.0.0:*        LISTEN
```

Uwaga! Musimy upewnić się, że firewall nie blokuje portu 3306. Aby odblokować go w systemie Fedora wpisujemy:

```
sudo firewall-cmd --zone=public --permanent --add-port=3306/tcp
```

Przyznawanie uprawnień użytkownikom zdalnym

Jeżeli użytkownik któremu chcemy nadać odpowiednie uprawnienia nie istnieje to uruchamiamy konsolę MariaDB i tworzymy go następującym poleceniem:

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

Gdzie oczywiście za `user` podajemy nazwę naszego użytkownika. Następnie nadajemy mu odpowiednie uprawnienia za pomocą polecenia

```
GRANT ALL ON nazwa_bazy.* to 'user'@'ip_maszyny' IDENTIFIED BY 'haslo' WITH  
GRANT OPTION;
```

- `nazwa_bazy` określa nazwę bazy do której chcemy nadać uprawnienia. Możemy podać `*` jeżeli chcemy nadać uprawnienia do wszystkich baz.
- `user` określa użytkownika któremu nadajemy uprawnienia
- `ip_maszyny` określa z jakiego ip użytkownik może się połączyć. Możemy wpisać `%` jeżeli chcemy zezwolić na dowolne ip
- `haslo` określa hasło użytkownika

Opcja `WITH GRANT OPTION` jest opcjonalna. Możemy usunąć ten fragment polecenia jeżeli nie chcemy aby nowy użytkownik również mógł nadawać uprawnienia.

Integracja z serwerem WWW

Co jeżeli chcemy wyświetlać zawartość bazy danych za pomocą strony www? Z pomocą przyjdzie nam PHP.

Założmy, że mamy już utworzoną bazę danych (o nazwie `reviews`). Utworzymy w niej nową tabelę przechowującą recenzje.

```
CREATE TABLE user_review (  
  id MEDIUMINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  reviewer_name CHAR(100),  
  star_rating TINYINT,  
  details VARCHAR(4000)  
);
```

Do tabeli wstawimy przykładowe dane:

```
INSERT INTO user_review (reviewer_name, star_rating, details) VALUES ('Ben', '5', 'Love the calzone!');
```

```
INSERT INTO user_review (reviewer_name, star_rating, details) VALUES ('Leslie', '1', 'Calzones are the worst.');
```

Tworzymy usera który będzie mógł przeglądać zawartość bazy:

```
GRANT ALL ON reviews.* to review_site@localhost IDENTIFIED BY 'JxSLRkdutW';
```

Teraz tworzymy plik php w /var/www/html/

Przykładowy skrypt który wyświetli nam zawartość tabeli user_review:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<body>
<?php
$hostname = "localhost";
$username = "review_site";
$password = "JxSLRkdutW";
$db = "reviews";
$dbconnect=mysqli_connect($hostname,$username,$password,$db);
if ($dbconnect->connect_error) {
    die("Database connection failed: " . $dbconnect->connect_error);
}
?>
```

```
<table border="1" align="center">
<tr>
<td>Reviewer Name</td>
<td>Stars</td>
<td>Details</td>
</tr>
<?php
$query = mysqli_query($dbconnect, "SELECT * FROM user_review")
    or die (mysqli_error($dbconnect));
while ($row = mysqli_fetch_array($query)) {
    echo
    "<tr>
    <td>{$row['reviewer_name']}</td>
    <td>{$row['star_rating']}</td>
    <td>{$row['details']}</td>
    </tr>\n";
}
?>
```



```
</table>
</body>
</html>
```

Aby przetestować wynik skryptu musimy uruchomić plik php w przeglądarce.

(http://localhost/nazwa_skryptu.php)

Testowanie zdalnego połączenia

Najpierw trzeba zainstalować klient MariaDB na systemie zdalnym. Użyjemy do tego polecenia:

```
apt-get install mariadb-client -y
```

następnie łączymy się z naszym głównym serwerem używając

```
mysql -u nazwa_uzytkownika -h ip_serwera -p
```

gdzie oczywiście w miejsce nazwy użytkownika wpisujemy nazwę uprawnionego do połączenia usera, a w miejsce ip podajemy ip serwera bazy danych.

Aby połączyć się z bazą lokalną używamy prostego polecenia

```
mysql -u nazwa_uzytkownika -p
```

Najważniejsze pliki

Pliki konfiguracyjne są przechowywane w `/etc/my.cnf`

Domyślny plik z logami `/var/log/mariadb/mariadb.log`

Domyślny folder instalacji `/var/lib/mysql`

Plik PID `/var/run/mariadb/mariadb.pid`

Domyślny plik gniazda unixowego `/var/lib/mysql/mysql.sock`

Bibliografia:

- <https://pl.wikipedia.org/wiki/MariaDB>
- <https://en.wikipedia.org/wiki/MariaDB>
- <https://www.samouczekprogramisty.pl/wstep-do-relacyjnych-baz-danych/>
- <https://excelbi.pl/czym-jest-relacyjna-baza-danych-jak-tworzyc-relacje-w-powerpivot/>
- <https://hostovita.pl/blog/porownanie-relacyjnych-systemow-zarzadzania-bazami-danych-sqlite-mysql-postgresql/>
- <https://webdock.io/en/docs/how-guides/how-enable-remote-access-your-mariadbmysql-database>
- <https://fedoraproject.org/wiki/MariaDB>
- <https://www.ionos.com/community/hosting/mysql/use-php-to-retrieve-information-from-a-mysqldb-mariadb-database/>