

Отчёт по лабораторной работе 6

**Установка и настройка системы управления базами данных
MariaDB**

Сейдалиев Тагиетдин Ровшенович

Содержание

1 Цель работы	5
2 Выполнение	6
2.1 Установка MariaDB	6
2.2 Конфигурация кодировки символов UTF-8	10
2.3 Создание базы данных и таблиц	12
2.4 Резервное копирование и восстановление	14
2.5 Подготовка окружения для автоматизации	15
3 Заключение	17
4 Контрольные вопросы	18
4.1 1. Какая команда отвечает за настройки безопасности в MariaDB?	18
4.2 2. Как настроить MariaDB для доступа через сеть?	18
4.3 3. Какая команда позволяет получить обзор доступных баз данных?	18
4.4 4. Какая команда позволяет узнать, какие таблицы доступны в базе данных?	19
4.5 5. Какая команда позволяет узнать, какие поля доступны в таблице?	19
4.6 6. Какая команда позволяет узнать, какие записи доступны в таблице?	19
4.7 7. Как удалить запись из таблицы?	19
4.8 8. Где расположены файлы конфигурации MariaDB? Что можно настроить?	19
4.9 9. Где располагаются файлы с базами данных MariaDB?	20
4.10 10. Как сделать резервную копию базы данных и затем её восстановить?	20

Список иллюстраций

2.1 Содержимое каталога /etc/my.cnf.d	6
2.2 Запуск и проверка прослушивания порта	7
2.3 Процесс mysql_secure_installation	8
2.4 Список клиентских команд mysql	9
2.5 Результат SHOW DATABASES	9
2.6 Статус MariaDB до изменения кодировки	10
2.7 Файл utf8.cnf	10
2.8 Статус MariaDB после применения кодировки UTF-8	11
2.9 Создание таблицы и заполнение данными	12
2.10 Создание пользователя и выдача прав	13
2.11 Просмотр баз данных	14
2.12 Создание и восстановление резервных копий	15
2.13 Скрипт автоматизации mysql.sh	16

Список таблиц

1 Цель работы

Приобретение практических навыков по установке и конфигурированию системы управления базами данных на примере программного обеспечения MariaDB.

2 Выполнение

2.1 Установка MariaDB

После установки пакетов `mariadb` и `mariadb-server` были просмотрены конфигурационные файлы в каталоге `/etc/my.cnf.d`. Содержимое каталога показано на рисунке:

```
Installed:
mariadb-3:10.11.11-1.el10.x86_64
mariadb-client-utils-3:10.11.11-1.el10.x86_64
mariadberrmsg-3:10.11.11-1.el10.noarch
mariadb-server-3:10.11.11-1.el10.x86_64
mysql-selinux-1.0.14-1.el10.noarch
perl-Sys-Hostname-1.25-512.2.el10.x86_64

mariadb-backup-3:10.11.11-1.el10.x86_64
mariadb-common-3:10.11.11-1.el10.noarch
mariadb-gssapi-server-3:10.11.11-1.el10.x86_64
mariadb-server-utils-3:10.11.11-1.el10.x86_64
perl-DBD-MariaDB-1.23-10.el10.x86_64

Complete!
[root@server.trseidaliev.net ~]# ls /etc/my.cnf.d/
auth_gssapi.cnf    enable_encryption.preset  mysql-clients.cnf  provider_lz4.cnf  provider_snappy.cnf
client.cnf          mariadb-server.cnf      provider_bzip2.cnf  provider_lzo.cnf  spider.cnf
[root@server.trseidaliev.net ~]# cat /etc/my.cnf
#
# This group is read both by the client and the server
# use it for options that affect everything
#
[client-server]

#
# include all files from the config directory
#
!includedir /etc/my.cnf.d

[root@server.trseidaliev.net ~]#
```

Рис. 2.1: Содержимое каталога `/etc/my.cnf.d`.

Также был изучен основной конфигурационный файл `/etc/my.cnf`:

Построчно разберём содержание файла `/etc/my.cnf`:

- `#` — строки-комментарии, поясняющие структуру файла.
- Блок `[client-server]` задаёт общие параметры, применяемые одновременно к клиенту `mysql` и серверу `mysqld`.

- Директива `!includedir /etc/my.cnf.d` подключает все дополнительные конфигурационные файлы из соответствующего каталога.

Служба MariaDB была успешно запущена и добавлена в автозагрузку. Проверка прослушивания порта показала, что процесс `mariadb` принимает соединения на стандартном порту 3306:

```
[root@server.trseidaliev.net ~]# systemctl start mariadb
[root@server.trseidaliev.net ~]# systemctl enable mariadb
Created symlink '/etc/systemd/system/mysql.service' → '/usr/lib/systemd/system/mariadb.service'.
Created symlink '/etc/systemd/system/mysqld.service' → '/usr/lib/systemd/system/mariadb.service'.
Created symlink '/etc/systemd/system/multi-user.target.wants/mariadb.service' → '/usr/lib/systemd/system/mariadb.service'.
[root@server.trseidaliev.net ~]# ss -tulpn | grep mysql
[root@server.trseidaliev.net ~]# ss -tulpn | grep maria
tcp    LISTEN  0      0.0.0.0:3306      0.0.0.0:*      users:(("mariadb",pid=16621,fd=18))
                                                uid:27  ino:92920 sk:18 cgroup:/system.slice
/mariadb.service <->
tcp    LISTEN  0      [::]:3306      [::]:*      users:(("mariadb",pid=16621,fd=19))
                                                uid:27  ino:92921 sk:24 cgroup:/system.slice
/mariadb.service v6only:1 <->
[root@server.trseidaliev.net ~]#
```

Рис. 2.2: Запуск и проверка прослушивания порта

Для обеспечения базовой безопасности был запущен скрипт интерактивной настройки `mysql_secure_installation`. В ходе работы были выполнены следующие действия:

- установлен пароль для пользователя `root` MariaDB;
- отключён удалённый доступ пользователя `root`;
- удалены анонимные пользователи;
- удалена тестовая база данных `test`;
- перезагружены таблицы привилегий.

Процесс работы утилиты приведён на скриншоте:

```
production environment.

Remove anonymous users? [Y/n]
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n]
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n]
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n]
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
[root@server.trseidaliev.net ~]#
```

Рис. 2.3: Процесс mysql_secure_installation

После входа в MariaDB был отображён список доступных клиентских команд:

```

General information about MariaDB can be found at
http://mariadb.org

List of all client commands:
Note that all text commands must be first on line and end with ';'
?          (\?) Synonym for 'help'.
charset   (\C) Switch to another charset. Might be needed for processing binlog with multi-byte charsets.
clear     (\c) Clear the current input statement.
connect   (\r) Reconnect to the server. Optional arguments are db and host.
delimiter (\d) Set statement delimiter.
edit      (\e) Edit command with $EDITOR.
ego       (\G) Send command to MariaDB server, display result vertically.
exit      (\q) Exit mysql. Same as quit.
go        (\g) Send command to MariaDB server.
help      (\h) Display this help.
nopager   (\n) Disable pager, print to stdout.
notee     (\t) Don't write into outfile.
nowarning (\w) Don't show warnings after every statement.
pager     (\P) Set PAGER [to_pager]. Print the query results via PAGER.
print     (\p) Print current command.
prompt    (\R) Change your mysql prompt.
quit      (\q) Quit mysql.
rehash    (\#) Rebuild completion hash.
sandbox   (\-) Disallow commands that access the file system (except \P without an argument and \e).
source    (\.) Execute an SQL script file. Takes a file name as an argument.
status    (\s) Get status information from the server.
system    (\!) Execute a system shell command.
tee       (\T) Set outfile [to_outfile]. Append everything into given outfile.
use      (\u) Use another database. Takes database name as argument.
warnings  (\W) Show warnings after every statement.

For server side help, type 'help contents'

```

Рис. 2.4: Список клиентских команд mysql

Для анализа доступных баз данных использована команда SHOW DATABASES, вывод которой показан ниже:

```

MariaDB [(none)]>
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+
4 rows in set (0.000 sec)

MariaDB [(none)]>

```

Рис. 2.5: Результат SHOW DATABASES

В системе присутствуют следующие стандартные базы:

- **information_schema** – метаданные объектов базы данных;
- **mysql** – система управления пользователями и правами;
- **performance_schema** – данные о работе сервера;

- sys – представления для удобного анализа производительности.

2.2 Конфигурация кодировки символов UTF-8

Команда `status` отобразила текущие параметры сервера. Все значения кодировок (`Server`, `Db`, `Client`) указывали использование `latin1`:

```
mariadb [(none)]>
MariaDB [(none)]> status
-----
mysql Ver 15.1 Distrib 10.11.11-MariaDB, for Linux (x86_64) using EditLine wrapper

Connection id: 13
Current database:
Current user: root@localhost
SSL: Not in use
Current pager: stdout
Using outfile: ''
Using delimiter: ;
Server: MariaDB
Server version: 10.11.11-MariaDB MariaDB Server
Protocol version: 10
Connection: Localhost via UNIX socket
Server characterset: latin1
Db characterset: latin1
Client characterset: utf8mb3
Conn. characterset: utf8mb3
UNIX socket: /var/lib/mysql/mysql.sock
Uptime: 4 min 16 sec

Threads: 1 Questions: 25 Slow queries: 0 Opens: 20 Open tables: 13 Queries per second avg: 0.097
-----
MariaDB [(none)]>
```

Рис. 2.6: Статус MariaDB до изменения кодировки

В каталоге `/etc/my.cnf.d` был создан файл `utf8.cnf` следующего содержания:

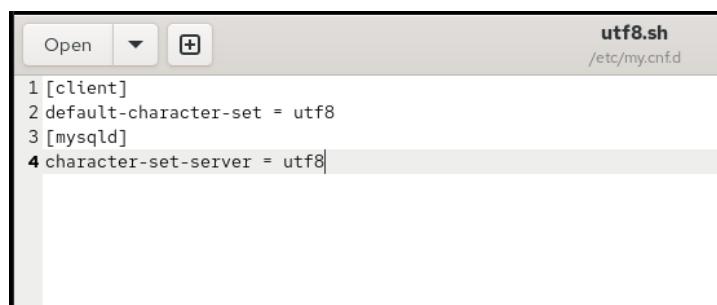


Рис. 2.7: Файл utf8.cnf

Файл включает параметры:

- для клиента: `default-character-set = utf8;`
- для сервера: `character-set-server = utf8.`

Это гарантирует использование UTF-8 на всех уровнях взаимодействия.

После перезапуска MariaDB повторный вызов команды `status` показал изменение всех кодировок на `utf8mb3`, что подтверждает успешное применение настроек:

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

MariaDB [(none)]> status
-----
mysql Ver 15.1 Distrib 10.11.11-MariaDB, for Linux (x86_64) using EditLine wrapper

Connection id:          3
Current database:
Current user:           root@localhost
SSL:                   Not in use
Current pager:          stdout
Using outfile:
Using delimiter:
Server:                 MariaDB
Server version:         10.11.11-MariaDB MariaDB Server
Protocol version:       10
Connection:             Localhost via UNIX socket
Server characterset:    utf8mb3
Db     characterset:    utf8mb3
Client characterset:   utf8mb3
Conn.  characterset:   utf8mb3
UNIX socket:            /var/lib/mysql/mysql.sock
Uptime:                 6 sec

Threads: 1  Questions: 4  Slow queries: 0  Opens: 17  Open tables: 10  Queries per second avg: 0.666
-----
MariaDB [(none)]>
```

Рис. 2.8: Статус MariaDB после применения кодировки UTF-8

Изменились следующие параметры:

- Server characterset;
- Db characterset;
- Client characterset;
- Connection characterset.

Все они теперь используют UTF-8, что обеспечивает корректную работу с многоязычными данными.

2.3 Создание базы данных и таблиц

После входа в окружение MariaDB с административными правами была создана база данных addressbook с кодировкой UTF-8. На рисунке показан процесс её создания, выбор активной базы и вывод списка таблиц:

Так как база только что создана, она не содержит таблиц, что подтверждается выводом SHOW TABLES.

Далее создана таблица city с двумя текстовыми полями: name и city. После создания таблица была заполнена тремя строками со сведениями о пользователях:

```
MariaDB [(none)]> CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> USE addressbook;
Database changed
MariaDB [addressbook]> SHOW TABLES;
Empty set (0.000 sec)

MariaDB [addressbook]> CREATE TABLE city(name VARCHAR(40), city VARCHAR(40));
Query OK, 0 rows affected (0.004 sec)

MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Иванов','Москва');
Query OK, 1 row affected (0.001 sec)

MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Петров','Сочи');
Query OK, 1 row affected (0.001 sec)

MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Сидоров','Дубна');
Query OK, 1 row affected (0.003 sec)

MariaDB [addressbook]> SELECT * FROM city
-> ;
+-----+-----+
| name | city |
+-----+-----+
| Иванов | Москва |
| Петров | Сочи |
| Сидоров | Дубна |
+-----+-----+
3 rows in set (0.000 sec)

MariaDB [addressbook]>
```

Рис. 2.9: Создание таблицы и заполнение данными

Запрос SELECT * FROM city; вывел все записи таблицы:

- Иванов — Москва
- Петров — Сочи

- Сидоров – Дубна

Это подтверждает корректное создание структуры таблицы и успешное добавление данных.

Для работы с базой был создан отдельный пользователь. На рисунке показано создание пользователя, выдача ему прав на базу и обновление привилегий:

```
mariadb [addressbook]>
MariaDB [addressbook]> CREATE USER trseidaliev@'%' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.004 sec)

MariaDB [addressbook]> GRANT SELECT,INSERT,UPDATE,DELETE ON addressbook.* TO trseidaliev@'%';
Query OK, 0 rows affected (0.001 sec)

MariaDB [addressbook]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [addressbook]> DESCRIBE city;
+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| name  | varchar(40) | YES  |     | NULL    |       |
| city   | varchar(40) | YES  |     | NULL    |       |
+-----+-----+-----+-----+
2 rows in set (0.001 sec)

MariaDB [addressbook]>
```

Рис. 2.10: Создание пользователя и выдача прав

Права **SELECT, INSERT, UPDATE, DELETE** позволяют пользователю выполнять базовые операции с данными.

Команда **DESCRIBE city**; отобразила структуру таблицы:

- поле **name** – тип **varchar(40)**
- поле **city** – тип **varchar(40)**

Это подтверждает корректность схемы данных.

Проверен общий список баз данных:

```
[root@server.trseidaliev.net my.cnf.d]# mysqlshow -u root -p
Enter password:
+-----+
| Databases |
+-----+
| addressbook
| information_schema
| mysql
| performance_schema
| sys
+-----+
[root@server.trseidaliev.net my.cnf.d]# mysqlshow -u trseidaliev -p addressbook
Enter password:
Database: addressbook
+-----+
| Tables |
+-----+
| city |
+-----+
[root@server.trseidaliev.net my.cnf.d]#
```

Рис. 2.11: Просмотр баз данных

В списке присутствуют:

- addressbook
- information_schema
- mysql
- performance_schema
- sys

Также были просмотрены таблицы в базе addressbook как от имени root, так и от имени созданного пользователя. Оба варианта показывают таблицу city, что подтверждает корректность прав доступа.

2.4 Резервное копирование и восстановление

В каталоге /var/backup были созданы резервные копии базы данных:

- обычная SQL-копия;
- сжатая gzip-копия;

- сжатая копия с указанием текущей даты и времени в имени файла.

Результаты выполнения представлены на скриншоте:

```
[root@server.trseidaliev.net my.cnf.d]# 
[root@server.trseidaliev.net my.cnf.d]# mkdir -p /var/backup
[root@server.trseidaliev.net my.cnf.d]# cd /var/backup/
[root@server.trseidaliev.net backup]# mysqldump -u root -p addressbook > addressbook.sql
Enter password:
[root@server.trseidaliev.net backup]# mysqldump -u root -p addressbook | gzip > addressbook.sql.gz
Enter password:
[root@server.trseidaliev.net backup]# mysqldump -u root -p addressbook | gzip > $(date +addressbook.%Y%m%d.%H%M%S.sql.gz)
Enter password:
[root@server.trseidaliev.net backup]# ls
addressbook.202511.1750.sql.gz addressbook.sql addressbook.sql.gz
[root@server.trseidaliev.net backup]#
[root@server.trseidaliev.net backup]# mysql -u root -p addressbook < addressbook.sql
Enter password:
[root@server.trseidaliev.net backup]# zcat addressbook.sql.gz | mysql -u root -p addressbook
Enter password:
[root@server.trseidaliev.net backup]#
```

Рис. 2.12: Создание и восстановление резервных копий

Созданные файлы:

- addressbook.sql
- addressbook.sql.gz
- addressbook.YYYYMMDD.HHMMSS.sql.gz

Также выполнено восстановление базы из обычной и сжатой копии. Обе операции завершились успешно.

2.5 Подготовка окружения для автоматизации

Для последующего использования в Vagrant конфигурации создан каталог структуры /vagrant/provision/server/mysql, в который были скопированы:

- конфигурационные файлы MariaDB;
- резервные копии базы addressbook.

Следом создан исполняемый скрипт `mysql.sh`, выполняющий автоматическую установку и конфигурирование MariaDB. Его фрагмент приведён на рисунке:

```
1 #!/bin/bash
2 echo "Provisioning script $0"
3 systemctl restart named
4 echo "Install needed packages"
5 dnf -y install mariadb mariadb-server
6 echo "Copy configuration files"
7 cp -R /vagrant/provision/server/mysql/etc/* /etc
8 mkdir -p /var/backup
9 cp -R /vagrant/provision/server/mysql/var/backup/* /var/backup
10 echo "Start mysql service"
11 systemctl enable mariadb
12 systemctl start mariadb
13 if [[ ! -d /var/lib/mysql/mysql ]]
14 then
15 echo "Securing mariadb"
16 mysql_secure_installation <<EOF
17 y
18 123456
19 123456
20 Y
21 Y
22 Y
23 Y
24 EOF
25 echo "Create database"
26 mysql -u root -p123456 <<EOF
27 CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE utf8_general_ci;
28 EOF
29 mysql -u root -p123456 addressbook < /var/backup/addressbook.sql
30 fi
```

Рис. 2.13: Скрипт автоматизации `mysql.sh`

3 Заключение

В процессе работы:

- создана база данных `addressbook` и таблица `city`;
- добавлены записи с данными о трёх пользователях;
- создан пользователь с правами на операции в базе `addressbook`;
- проверены структура таблицы и доступность базы через `mysqlshow`;
- выполнено резервное копирование базы в обычном и сжатом формате, а также восстановление данных;
- подготовлена структура каталогов и скрипт `mysql.sh` для автоматизации установки и конфигурации MariaDB.

4 Контрольные вопросы

4.1 1. Какая команда отвечает за настройки безопасности в MariaDB?

За базовую настройку безопасности отвечает команда:

`mysql_secure_installation`

Она позволяет задать пароль root, удалить тестовые базы, отключить анонимных пользователей и запретить удалённый вход root.

4.2 2. Как настроить MariaDB для доступа через сеть?

Для сетевого доступа требуется:

- в файле конфигурации (`/etc/my.cnf` или `/etc/my.cnf.d/*.cnf`) задать параметр `bind-address = 0.0.0.0` или указать нужный IP;
- обеспечить открытие порта 3306 в firewall;
- создать пользователя с доступом по сети, например: `CREATE USER user@'%' IDENTIFIED BY 'password';`

4.3 3. Какая команда позволяет получить обзор доступных баз данных?

`SHOW DATABASES;`

4.4 4. Какая команда позволяет узнать, какие таблицы доступны в базе данных?

SHOW TABLES;

4.5 5. Какая команда позволяет узнать, какие поля доступны в таблице?

DESCRIBE table_name; или SHOW COLUMNS FROM table_name;

4.6 6. Какая команда позволяет узнать, какие записи доступны в таблице?

SELECT * FROM table_name;

4.7 7. Как удалить запись из таблицы?

Используется команда:

DELETE FROM table_name WHERE условие;

Например:

DELETE FROM city WHERE name='Иванов';

4.8 8. Где расположены файлы конфигурации MariaDB? Что можно настроить?

Конфигурационные файлы находятся в:

- /etc/my.cnf

- `/etc/my.cnf.d/`

С помощью конфигурации можно задавать:

- сетевые настройки;
- кодировки и локаль;
- параметры хранения данных;
- поведение сервера (кэш, буферы, ограничения);
- пути к файлам журналов и сокетам.

4.9 9. Где располагаются файлы с базами данных

MariaDB?

Все базы данных находятся в каталоге:

`/var/lib/mysql/`

В нём для каждой базы создаётся отдельная директория.

4.10 10. Как сделать резервную копию базы данных и затем её восстановить?

Резервная копия:

- в SQL-файл: `mysqldump -u root -p database > backup.sql`
- в сжатом виде: `mysqldump -u root -p database | gzip > backup.sql.gz`

Восстановление:

- из обычного файла: `mysql -u root -p database < backup.sql`
- из сжатого файла: `zcat backup.sql.gz | mysql -u root -p database`