

Отчёт по лабораторной работе 6

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

Чилеше Лупупа

Содержание

1	Цель работы	5
2	Выполнение работы	6
2.1	Установка и первичная настройка MariaDB	6
2.1.1	Установка необходимых пакетов	6
2.1.2	Просмотр конфигурационных файлов MariaDB	7
2.1.3	Запуск и включение службы MariaDB	7
2.1.4	Проверка прослушивания порта MariaDB	8
2.2	Настройка безопасности MariaDB	8
2.3	Проверка доступных команд и системных баз данных	9
2.4	Конфигурация кодировки символов	11
2.4.1	Просмотр состояния MariaDB до изменений	11
2.4.2	Создание конфигурации кодировки	12
2.4.3	Перезапуск и проверка изменений	12
3	Создание базы данных и управление данными в MariaDB	14
3.0.1	Создание базы данных addressbook	14
3.0.2	Создание таблицы city и заполнение данными	14
3.0.3	Создание пользователя и назначение прав	15
3.0.4	Просмотр списка баз данных и таблиц	16
3.1	Создание резервных копий базы данных	17
3.2	Добавление конфигурации MariaDB в Vagrant-окружение	17
4	Вывод	19
5	Контрольные вопросы	20

Список иллюстраций

2.1	Установка MariaDB	6
2.2	mysql_secure_installation	9
2.3	Список баз данных	10
2.4	Статус MariaDB до изменения кодировки	11
2.5	Создание файла utf8.cnf	12
2.6	Статус MariaDB после изменения кодировки	13
3.1	Результат SELECT * FROM city	15
3.2	Создание пользователя	16
3.3	mysqlshow -u root -p	16
3.4	Создание каталога /var/backup	17
3.5	Содержимое mysql.sh	18

Список таблиц

1 Цель работы

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

2 Выполнение работы

2.1 Установка и первичная настройка MariaDB

2.1.1 Установка необходимых пакетов

Для работы с системой управления базами данных были установлены пакеты mariadb и mariadb-server. После установки система вывела перечень компонентов, включая серверную часть, клиентские утилиты и дополнительные модули.

```
Installed:
mariadb-3:10.11.11-1.el10.x86_64
mariadb-client-utils-3:10.11.11-1.el10.x86_64
mariadb-errmsg-3:10.11.11-1.el10.noarch
mariadb-server-3:10.11.11-1.el10.x86_64
mysql-selinux-1.0.14-1.el10_0.noarch
perl-Sys-Hostname-1.25-512.2.el10_0.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.chileshe.net ~]#
[root@server.chileshe.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.chileshe.net ~]# cat /etc/my.cnf
#
# This group is read both 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.chileshe.net ~]# systemctl start mariadb
[root@server.chileshe.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.servi
ce'.
[root@server.chileshe.net ~]# ss -tulpen | grep maria
tcp LISTEN 0 80 0.0.0.0:3306 0.0.0.0:* users:(("mariabdb",pid=10990,fd=16))
uid:27 ino:54644 sk:14 cgroup:/system.slice
/mariadb.service <->
tcp LISTEN 0 80 [::]:3306 [::]:* users:(("mariabdb",pid=10990,fd=17))
uid:27 ino:54645 sk:21 cgroup:/system.slice
/mariadb.service v6only:1 <->
[root@server.chileshe.net ~]#
```

Рис. 2.1: Установка MariaDB

2.1.2 Просмотр конфигурационных файлов MariaDB

В каталоге `/etc/my.cnf.d` находятся конфигурационные файлы службы MariaDB, которые включаются основным файлом `/etc/my.cnf`. На изображении продемонстрировано содержимое каталога конфигурации:

Основной файл `my.cnf` включает общие настройки клиента и сервера:

Построчный комментарий файла `/etc/my.cnf`:

- `#` — строки комментариев, используемые для пояснений.
- `This group is read both by the client and the server` — параметры из этой группы читаются и клиентами, и сервером.
- `use it for options that affect everything` — рекомендуется помещать сюда глобальные настройки.
- `[client-server]` — секция общих параметров.
- `# include all files from the config directory` — пояснение о подключении дополнительных конфигов.
- `!includedir /etc/my.cnf.d` — директива, подключающая содержимое каталога `/etc/my.cnf.d`.

2.1.3 Запуск и включение службы MariaDB

После установки сервер был запущен и добавлен в автозагрузку. Завершение команд сопровождалось созданием симлинков на соответствующие unit-файлы.

2.1.4 Проверка прослушивания порта MariaDB

Командой для анализа активных сетевых соединений было подтверждено, что процесс `mariadb` слушает порт 3306.

2.2 Настройка безопасности MariaDB

Для выполнения базовой конфигурации безопасности был запущен скрипт `mysql_secure_installation`. В ходе работы были выполнены следующие действия:

- установлен пароль пользователя `root` (MariaDB);
- отключён удалённый доступ `root`;
- удалены анонимные пользователи;
- удалена тестовая база данных;
- обновлены таблицы привилегий.


```

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
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.chileshe.net ~]# █

```

Рис. 2.2: mysql_secure_installation

2.3 Проверка доступных команд и системных баз данных

После входа в MariaDB (`mysql -u root -p`) был просмотрен список доступных команд (`\h`), а затем выполнен SQL-запрос для отображения доступных баз данных

Результат:

```

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'

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

MariaDB [(none)]>

```

Рис. 2.3: Список баз данных

В системе доступны следующие базы данных:

- `information_schema` — служебная база метаданных;
- `mysql` — таблицы пользователей, прав и системных параметров;
- `performance_schema` — статистика о работе сервера;
- `sys` — представления для анализа производительности.

2.4 Конфигурация кодировки символов

2.4.1 Просмотр состояния MariaDB до изменений

Для анализа текущей конфигурации был выполнен просмотр статуса сервера:

```
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:                  3 min 45 sec

Threads: 1 Questions: 25 Slow queries: 0 Opens: 20 Open tables: 13 Queries per second avg: 0.111

MariaDB [(none)]>
```

Рис. 2.4: Статус MariaDB до изменения кодировки

Пояснение ключевых параметров:

- *Server version* — версия MariaDB.
- *Current user* — пользователь, от имени которого осуществляется соединение.
- *SSL* — состояние SSL-соединения.
- *Connection* — локальное соединение через UNIX-сокеты.
- *Server/Db/Client/Conn character set* — используемые кодировки.
- *Uptime* — время работы службы.

- *Threads, Questions, Slow queries* — статистика активности.

2.4.2 Создание конфигурации кодировки

В каталоге `/etc/my.cnf.d` был создан файл `utf8.cnf` со следующим содержанием:

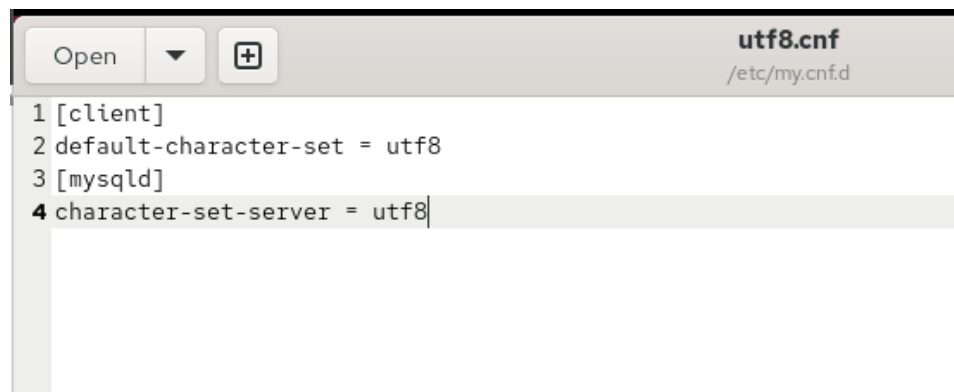


Рис. 2.5: Создание файла `utf8.cnf`

2.4.3 Перезапуск и проверка изменений

После перезапуска MariaDB был повторно просмотрен статус системы:

```

[root@server.chileshe.net my.cnf.d]# systemctl restart mariadb
[root@server.chileshe.net my.cnf.d]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.11.11-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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:                 14 sec

Threads: 1  Questions: 4  Slow queries: 0  Opens: 17  Open tables: 10  Queries per second avg: 0.285
-----
MariaDB [(none)]>

```

Рис. 2.6: Статус MariaDB после изменения кодировки

Изменения после применения конфигурации:

- параметры Server character set, Db character set, Client character set и Conn character set изменились на utf8mb3;
- сервер начал работу в корректной UTF-8-совместимой кодировке;
- обновлённые параметры подтвердили успешное применение настроек.

3 Создание базы данных и управление данными в MariaDB

3.0.1 Создание базы данных addressbook

Для начала был выполнен вход в MariaDB с правами администратора. Затем была создана база данных addressbook с использованием кодировки UTF-8:

После создания база была выбрана для дальнейшей работы:

Так как база данных только создана, список таблиц оказался пустым.

3.0.2 Создание таблицы city и заполнение данными

В базе данных была создана таблица city с двумя текстовыми полями:

После этого таблица была заполнена тремя строками:

- Иванов — Москва
- Петров — Сочи
- Сидоров — Дубна

Для проверки данных был выполнен запрос:

```

MariaDB [(none)]> CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> USE addressbook;
Database changed
MariaDB [addressbook]> SHOW TABLES;
Empty set (0.001 sec)

MariaDB [addressbook]> CREATE TABLE city(name VARCHAR(40), city VARCHAR(40));
Query OK, 0 rows affected (0.007 sec)

MariaDB [addressbook]> INSERT INTO city(name, city) VALUES ('Иванов', 'Москва');
Query OK, 1 row affected (0.005 sec)

MariaDB [addressbook]> INSERT INTO city(name, city) VALUES ('Петров', 'Сочи');
Query OK, 1 row affected (0.004 sec)

MariaDB [addressbook]> INSERT INTO city(name, city) VALUES ('Сидоров', 'Дубна');
Query OK, 1 row affected (0.008 sec)

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

MariaDB [addressbook]> █

```

Рис. 3.1: Результат SELECT * FROM city

Пояснение результата:

Запрос выводит все строки таблицы city. Таблица содержит три записи, каждая из которых включает имя и соответствующий город.

3.0.3 Создание пользователя и назначение прав

Был создан пользователь chileshe@'%' и установлен пароль:

```

MariaDB [addressbook]> CREATE USER chileshe@%' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.001 sec)

MariaDB [addressbook]> GRANT SELECT,CREATE,INSERT,UPDATE<DELETE on addressbook.* TO chileshe@%';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '<DELETE on addressbook.* TO chileshe@%' at line 1
MariaDB [addressbook]> GRANT SELECT,CREATE,INSERT,UPDATE,DELETE on addressbook.* TO chileshe@%';
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. 3.2: Создание пользователя

3.0.4 Просмотр списка баз данных и таблиц

После выхода из MariaDB были просмотрены все базы данных:

```

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

```

Рис. 3.3: mysqlshow -u root -p

Затем был выполнен просмотр таблиц базы addressbook

Обе команды подтвердили наличие таблицы city.

3.1 Создание резервных копий базы данных

В системе был создан каталог для резервных копий:

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

Рис. 3.4: Создание каталога /var/backup

Далее были выполнены следующие действия:

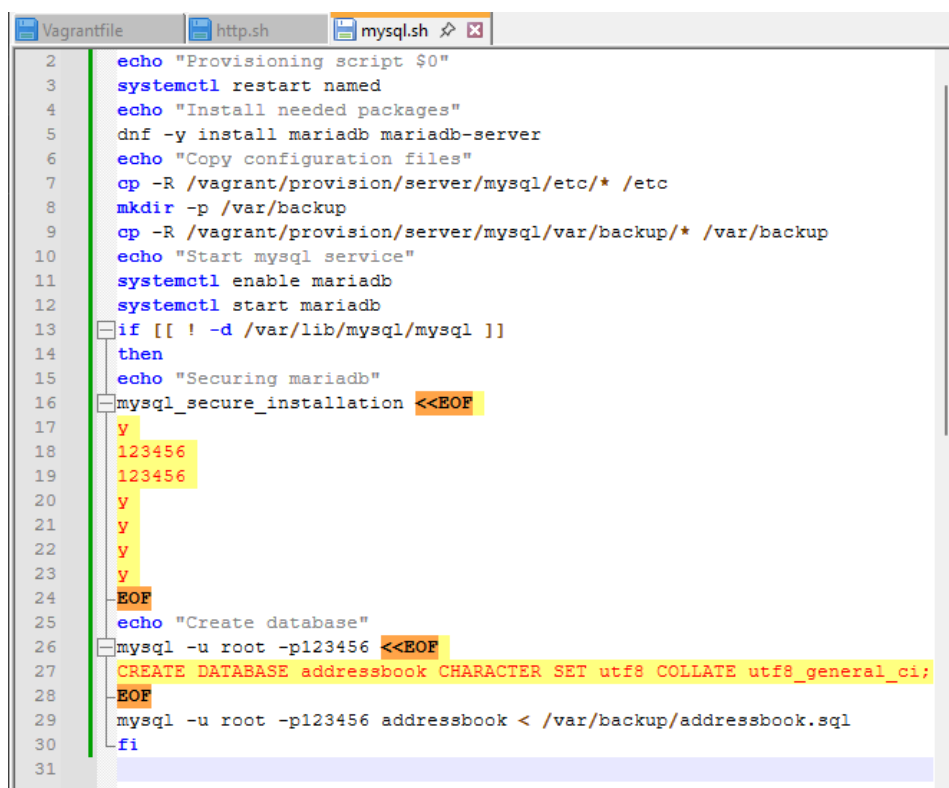
1. Обычная резервная копия
2. Сжатая резервная копия
3. Сжатая копия с датой в имени файла
4. Восстановление из обычной копии
5. Восстановление из gzip-копии

3.2 Добавление конфигурации MariaDB в Vagrant-окружение

В каталоге /vagrant/provision/server/ был создан подкаталог mysql и в него были помещены:

- конфигурационные файлы MariaDB;
- резервные копии базы данных.

Также был создан и сделан исполняемым файл `mysql.sh`, предназначенный для автоматизации установки и настройки MariaDB при развёртывании виртуальной машины:



```
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
31
```

Рис. 3.5: Содержимое `mysql.sh`

4 Вывод

В ходе работы была создана и настроена база данных MariaDB, включающая таблицу `city` и записи пользователей. Были продемонстрированы операции создания базы, таблиц, добавления данных, управления пользователями и привилегиями. Выполнены резервные копии базы данных в разных форматах и проверено корректное восстановление. Конфигурационные файлы и резервные копии интегрированы в инфраструктуру Vagrant для автоматизации последующих развертываний. Работа подтверждает корректность настройки MariaDB и выполнение всех требуемых операций.

5 Контрольные вопросы

1. Какая команда отвечает за настройки безопасности в MariaDB?

За выполнение первичной настройки безопасности отвечает команда `mysql_secure_installation`.

Она позволяет задать пароль root, удалить анонимных пользователей, отключить удалённый вход root и удалить тестовую базу данных.

2. Как настроить MariaDB для доступа через сеть?

Необходимо изменить параметр `bind-address` в конфигурации MariaDB:

- указать `0.0.0.0`, чтобы разрешить подключение с любых адресов;
- либо указать конкретный IP.

После изменения — перезапустить службу MariaDB. Также пользователям должны быть выданы права вида `user@'%'` или `user@'<ip>'`.

3. Какая команда позволяет получить обзор доступных баз данных после входа в среду оболочки MariaDB?

Команда:

```
SHOW DATABASES;
```

Она выводит перечень всех доступных пользователю баз данных.

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

Команда:

```
SHOW TABLES;
```

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

Команда:

```
DESCRIBE имя_таблицы;
```

или короче:

```
DESC имя_таблицы;
```

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

Команда:

```
SELECT * FROM имя_таблицы;
```

7. Как удалить запись из таблицы?

Используется команда:

```
DELETE FROM имя_таблицы WHERE условие;
```

Например:

```
DELETE FROM city WHERE name='Петров';
```

8. Где расположены файлы конфигурации MariaDB? Что можно настроить с их помощью?

Основные конфигурации находятся в каталоге:

```
/etc/my.cnf
```

и в директории:

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

С их помощью можно настроить: - параметры кодировки; - пути к данным; - сетевые параметры (порт, bind-address); - настройки буферов и кешей; - параметры клиентов и сервера.

9. Где располагаются файлы с базами данных MariaDB?

Физические файлы баз данных по умолчанию находятся по пути:

`/var/lib/mysql/`

В этом каталоге каждая база представлена собственной директорией.

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

Резервная копия создаётся командой:

```
mysqldump -u root -p база > файл.sql
```

Сжатая копия:

```
mysqldump -u root -p база | gzip > файл.sql.gz
```

Восстановление из обычного файла:

```
mysql -u root -p база < файл.sql
```

Восстановление из gzip-архива:

```
zcat файл.sql.gz | mysql -u root -p база
```