

Отчет по лабораторной работе №6

Дисциплина: Администрирование сетевых подсистем

Иванов Сергей Владимирович

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Установка MariaDB	6
3.2	Конфигурация кодировки символов	16
3.3	Создание базы данных	19
3.4	Резервные копии	21
4	Ответы на контрольные вопросы	25
5	Выводы	27

Список иллюстраций

3.1	Запуск server	6
3.2	Установка MariaDB	6
3.3	Файл auth_gssapi.conf	7
3.4	Файл client.cnf	8
3.5	Файл mariadb-server.cnf	8
3.6	Файл mysql-clients.cnf	9
3.7	Файл provider_bzip2.cnf	10
3.8	Файл provider_lz4.cnf	10
3.9	Файл provider_lzo.cnf	11
3.10	Файл provider_snappy.cnf	12
3.11	Файл spider.cnf	12
3.12	Файл my.cnf	13
3.13	Запуск mariaDB	13
3.14	Порт mariaDB	14
3.15	Запуск скрипта безопасности	14
3.16	Вход в базу данных	14
3.17	Список команд	15
3.18	Отображение доступных баз	16
3.19	Статус MariaDB	17
3.20	Создание utf8.cnf	17
3.21	Редактирование utf8.cnf	17
3.22	Перезапуск базы данных	18
3.23	Просмотр статуса	18
3.24	Создание базы данных	19
3.25	Создание и наполнение таблицы	19
3.26	Запрос к базе	20
3.27	Создание пользователя	20
3.28	Обновление привелегий и просмотр базы	21
3.29	Список баз и таблиц	21
3.30	Резервные копии	22
3.31	Восстановление базы данных	22
3.32	Создание каталогов для внесения изменений	23
3.33	Скрипт mysql.sh	23
3.34	Запись в Vagrantfile	24

1 Цель работы

Целью этой работы является приобретение практических навыков по установке и конфигурированию системы управления базами данных на примере программного обеспечения MariaDB.

2 Задание

1. Установите необходимые для работы MariaDB пакеты (см. раздел 6.4.1).
2. Настройте в качестве кодировки символов по умолчанию utf8 в базах данных.
3. В базе данных MariaDB создайте тестовую базу addressbook, содержащую таблицу city с полями name и city, т.е., например, для некоторого сотрудника указан город, в котором он работает (см. раздел 6.4.1).
4. Создайте резервную копию базы данных addressbook и восстановите из неё данные (см. раздел 6.4.1).
5. Напишите скрипт для Vagrant, фиксирующий действия по установке и настройке базы данных MariaDB во внутреннем окружении виртуальной машины server. Соответствующим образом следует внести изменения в Vagrantfile (см. раздел 6.4.5).

3 Выполнение лабораторной работы

3.1 Установка MariaDB

Загрузим операционную систему и перейдем в рабочий каталог с проектом: `cd /var/tmp/user_name/vagrant`. Запустим виртуальную машину `server`: `vagrant up server`. (рис. 1).

```
C:\Users\lserg>cd C:\work_asp\svivanov\vagrant  
  
C:\work_asp\svivanov\vagrant>vagrant up server  
Bringing machine 'server' up with 'virtualbox' provider...  
==> server: You assigned a static IP ending in ".1" or ":1"
```

Рис. 3.1: Запуск server

На виртуальной машине `server` войдем под пользователем и откроем терминал. Перейдем в режим суперпользователя. Установим необходимые для работы с базами данных пакеты: `dnf -y install mariadb mariadb-server` (рис. 2).

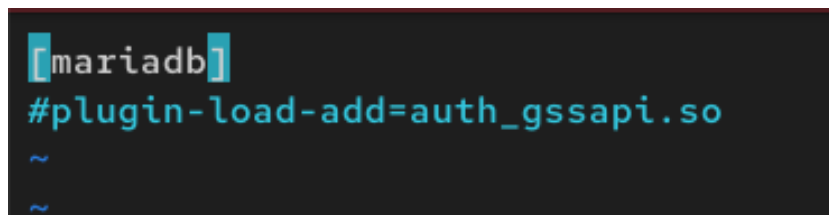
```
[root@server.svivanov.net ~]# dnf -y install mariadb mariadb-server  
Extra Packages for Enterprise Linux 10 - x86_64          508 B/s | 43 kB  01:26  
Extra Packages for Enterprise Linux 10 - x86_64          237 kB/s | 4.8 MB 00:20  
Rocky Linux 10 - BaseOS                                  260 B/s | 3.9 kB  00:15  
Rocky Linux 10 - BaseOS                                  422 kB/s | 20 MB  00:48  
Rocky Linux 10 - AppStream                               1.6 kB/s | 3.9 kB  00:02  
Rocky Linux 10 - AppStream                               435 kB/s | 2.1 MB 00:04  
Rocky Linux 10 - Extras                                  298 B/s | 3.1 kB  00:10  
Rocky Linux 10 - Extras                                  1.9 kB/s | 5.4 kB  00:02  
Dependencies resolved.  
=====
```

Package	Architecture	Version	Repository	Size
Installing:				
mariadb	x86_64	3:10.11.11-1.el10	appstream	1.6 M

Рис. 3.2: Установка MariaDB

Просмотрим конфигурационные файлы `mariadb` в каталоге `/etc/my.cnf.d` и в файле `/etc/my.cnf`.

Файл `auth_gssapi.conf`. В данном конфигурационном файле `auth_gssapi.conf` директива, отвечающая за загрузку плагина аутентификации через Kerberos (`auth_gssapi.so`), закомментирована. Это означает, что на данном сервере MariaDB механизм аутентификации GSSAPI отключен и не используется. Сервер будет работать со стандартными методами аутентификации (рис. 3)



```
[mariadb]
#plugin-load-add=auth_gssapi.so
~
~
```

Рис. 3.3: Файл `auth_gssapi.conf`

Файл `client.cnf`. Данный конфигурационный файл является заготовкой для настройки клиентских подключений. В его текущем состоянии он не содержит активных параметров (таких как хост, пользователь, пароль и т.д.), а лишь определяет логическую структуру:

Секция `client` предназначена для универсальных настроек, которые применяются ко всем клиентским программам (как MySQL, так и MariaDB).

Секция `client-mariadb` предназначена для специфичных настроек, которые будут применены только к клиентским утилитам MariaDB, что обеспечивает гибкость и совместимость.

Файл подготовлен для добавления конкретных параметров подключения, но в настоящее время он пуст от активных директив. Его функция — организовать будущие настройки клиентской части СУБД. (рис. 4)

```
#
# These two groups are read by the client library
# Use it for options that affect all clients, but not the server
#

[client]

# This group is not read by mysql client library,
# If you use the same .cnf file for MySQL and MariaDB,
# use it for MariaDB-only client options
[client-mariadb]
```

Рис. 3.4: Файл client.cnf

Файл mariadb-server.cnf. Конфигурационный файл содержит базовые, но критически важные настройки сервера MariaDB. Он определяет “рабочее пространство” сервера: Расположение данных (datadir), Точку для локальных подключений (socket), Файл для диагностики ошибок (log-error), Файл идентификации процесса (pid-file).

Файл задает стандартную конфигурацию, которая обеспечивает базовую функциональность сервера, определяя правильные и общепринятые пути для его ключевых компонентов. (рис. 5)

```
# this is only for the mysqld standalone daemon
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mysqld/mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd

[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
log-error=/var/log/mariadb/mariadb.log
pid-file=/run/mariadb/mariadb.pid

"mariadb-server.cnf" 55L, 1460B
```

Рис. 3.5: Файл mariadb-server.cnf

Файл mysql-clients.cnf. Конфигурационный файл представляет собой структурированную заготовку для тонкой настройки клиентских утилит MariaDB. В его текущем состоянии он не содержит активных параметров, а лишь определяет логические секции для различных программ.

Файл демонстрирует модульный подход к конфигурации MariaDB, где разные

аспекты системы настраиваются в отдельных специализированных файлах. В данном случае этот файл отвечает исключительно за настройку поведения клиентских утилит командной строки. (рис. 6)

```
# These groups are read by MariaDB command-line tools
# Use it for options that affect only one utility
#

[mysql]

[mysql_upgrade]

[mysqladmin]

[mysqlbinlog]

[mysqlcheck]

[mysqldump]

[mysqlimport]

[mysqlshow]

[mysqlslap]
```

Рис. 3.6: Файл mysql-clients.cnf

Файл provider_bzip2.cnf. Конфигурационный файл отвечает за подключение и настройку плагина сжатия bzip2 в сервере MariaDB.

Загрузка плагина: Директива `plugin_load_add` гарантирует, что плагин `provider_bzip2` будет загружен при каждом запуске сервера.

Параметр `force_plus_permanent` обеспечивает отказоустойчивость (сервер не “упадет” из-за отсутствия плагина) и сохраняет настройку постоянной.

После загрузки этого плагина MariaDB получает возможность использовать алгоритм bzip2 для операций сжатия. Это может быть полезно в различных сценариях, например: Сжатие данных при использовании функций типа `COMPRESS()`, Настройка сжатия для механизма хранения InnoDB (если плагин это поддерживает). (рис. 7)

```
[server]
plugin_load_add=provider_bzip2
provider_bzip2=force_plus_permanent
~
~
~
```

Рис. 3.7: Файл provider_bzip2.cnf

Файл provider_lz4.cnf. Конфигурационный файл предназначен для подключения и настройки плагина сжатия LZ4 в сервере MariaDB.

Загрузка плагина: Директива plugin_load_add должна гарантировать, что плагин provider_lz4 будет загружен при каждом запуске сервера.

Параметр force_plus_permanent обеспечивает отказоустойчивость (сервер не “упадет” из-за отсутствия плагина) и сохраняет настройку постоянной.

Практическое применение: После загрузки этого плагина MariaDB получает возможность использовать алгоритм LZ4 для операций сжатия. LZ4 известен своим очень высоким скоростью сжатия и распаковки, что делает его особенно полезным для: Сжатия данных в реальном времени, в приложениях, где важна низкая задержка (low latency, использования в функциях сжатия MariaDB и различных механизмах хранения (рис. 8)

```
[server]
plugin_load_add=provider_lz4
provider_lz4=force_plus_permanent
~
~
~
```

Рис. 3.8: Файл provider_lz4.cnf

Файл provider_lzo.cnf. Конфигурационный файл отвечает за подключение и настройку плагина сжатия LZO в сервере MariaDB.

Загрузка плагина: Директива plugin_load_add гарантирует, что плагин provider_lzo будет загружен при каждом запуске сервера.

Параметр `force_plus_permanent` обеспечивает отказоустойчивость (сервер не “упадет” из-за отсутствия плагина) и сохраняет настройку постоянной.

Практическое применение: После загрузки этого плагина MariaDB получает возможность использовать алгоритм LZO для операций сжатия. LZO — это алгоритм сжатия, который обеспечивает хороший баланс между скоростью и степенью сжатия. Он может использоваться для: Сжатия данных в функциях типа `COMPRESS()`, настройки сжатия для различных механизмов хранения, операций бэкапа и восстановления данных, любых других сценариев, где требуется эффективное сжатие данных (рис. 9)



```
[server]
plugin_load_add=provider_lzo
provider_lzo=force_plus_permanent
~
~
```

Рис. 3.9: Файл `provider_lzo.cnf`

Файл `provider_snappy.cnf`. Конфигурационный файл предназначен для подключения и настройки плагина сжатия Snappy в сервере MariaDB.

Загрузка плагина: Директива `plugin_load_add` должна гарантировать, что плагин `provider_snappy` будет загружен при каждом запуске сервера.

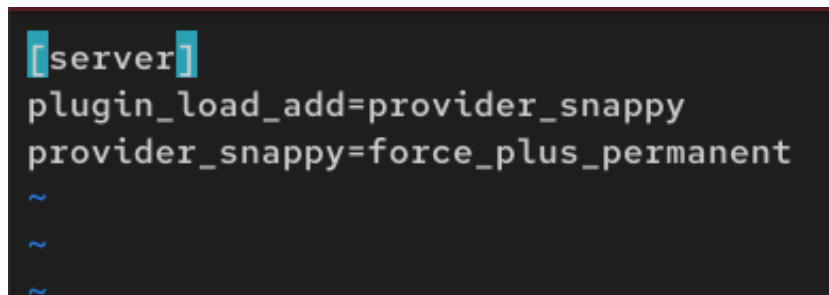
Параметр `force_plus_permanent` обеспечивает отказоустойчивость (сервер не “упадет” из-за отсутствия плагина) и сохраняет настройку постоянной.

Практическое применение: после загрузки этого плагина MariaDB получает возможность использовать алгоритм Snappy для операций сжатия. Snappy — это алгоритм сжатия, разработанный Google, который характеризуется:

- Очень высокой скоростью сжатия и распаковки
- Умеренной степенью сжатия
- Оптимизацией для 64-битных систем

Snappy особенно полезен для:

- Сжатия данных в реальном времени
- Приложений, где критична скорость обработки данных
- Распределенных систем и баз данных
- Использования в различных механизмах хранения MariaDB (рис. 10)



```
[server]
plugin_load_add=provider_snappy
provider_snappy=force_plus_permanent
~
~
~
```

Рис. 3.10: Файл provider_snappy.cnf

Файл spider.cnf. Конфигурационный файл содержит закомментированную директиву для загрузки плагина распределенного механизма хранения Spider. В текущем состоянии плагин отключен и не загружается сервером MariaDB. Для его активации необходимо раскомментировать строку plugin-load-add = ha_spider и перезапустить сервер. Spider предоставляет возможности для создания шардированных и распределенных баз данных, что полезно для масштабирования систем (рис. 11)



```
[mariadb]
#
# Uncomment line to enable
#
#plugin-load-add = ha_spider

# Read more at https://mariadb.com/kb/en/spider/
~
~
```

Рис. 3.11: Файл spider.cnf

Файл my.cnf. Главный конфигурационный файл my.cnf в его текущем состоянии не функционален. Критически важная директива includedir /etc/my.cnf.d закомментирована, что предотвращает загрузку всех модульных конфигурационных файлов из каталога /etc/my.cnf.d. Это означает, что настройки сервера, клиентов и плагинов, определенные в отдельных файлах, не будут применены. Для нормальной работы MariaDB необходимо раскомментировать строку includedir /etc/my.cnf.d. Файл демонстрирует структуру для централизованных настроек через секцию client-server, но в текущем виде не содержит активных конфигурационных параметров (рис. 12)

```
##
# 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
```

Рис. 3.12: Файл my.cnf

Для запуска и включения программного обеспечения mariadb используем:

systemctl start mariadb

systemctl enable mariadb (рис. 13)

```
[root@server.svivanov.net my.cnf.d]# systemctl start mariadb
[root@server.svivanov.net my.cnf.d]# 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.svivanov.net my.cnf.d]#
```

Рис. 3.13: Запуск mariaDB

Убедимся, что mariadb прослушивает порт, используя ss -tulpen | grep 3306 (рис. 14)

```
[root@server.svivanov.net my.cnf.d]# ss -tulpen | grep 3306
tcp    LISTEN 0      80          0.0.0.0:3306  0.0.0.0:*    users:(("mariadb",pid=17651,fd=18))
                                uid:27 ino:91802 sk:1001 cgroup:/system.slice/
tcp    LISTEN 0      80          [::]:3306   [::]:*       users:(("mariadb",pid=17651,fd=19))
                                uid:27 ino:91803 sk:1002 cgroup:/system.slice/
1 <->
[root@server.svivanov.net my.cnf.d]#
```

Рис. 3.14: Порт mariaDB

Запустим скрипт конфигурации безопасности mariadb, используя: `mysql_secure_installation`. С помощью запустившегося диалога и путём выбора Y/n установим пароль для пользователя root базы данных (обратите внимание, что это не пользователь root операционной системы), отключим удалённый корневой доступ и удалим тестовую базу данных и любых анонимных пользователей. (рис. 15)

```
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] y
... 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] y
- 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] y
... 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.svivanov.net my.cnf.d]#
```

Рис. 3.15: Запуск скрипта безопасности

Для входа в базу данных с правами администратора базы данных введем `mysql -u root -p`. (рис. 16)

```
[root@server.svivanov.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 14
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)]>
```

Рис. 3.16: Вход в базу данных

Просмотрим список команд MySQL, введя h. (рис. 17)

```
MariaDB [(none)]> \h

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 mu
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.
```

Рис. 3.17: Список команд

Из приглашения интерактивной оболочки MariaDB для отображения доступных в настоящее время баз данных введем MySQL-запрос SHOW DATABASES;

Были обнаружены четыре системные базы данных:

- information_schema - содержит метаданные всех объектов СУБД
- mysql - хранит информацию о пользователях, привилегиях и системных настройках
- performance_schema - предоставляет инструменты для мониторинга производительности
- sys - содержит упрощенные представления для анализа метрик производительности

Все перечисленные базы данных являются служебными и создаются автоматически при установке MariaDB. Пользовательских баз данных в системе не существует. (рис. 18)

```

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database           |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+-----+
4 rows in set (0,003 sec)

```

Рис. 3.18: Отображение доступных баз

3.2 Конфигурация кодировки символов

Войдем в базу данных с правами администратора: `mysql -u root -p`. Для отображения статуса MariaDB введем из приглашения интерактивной оболочки MariaDB: `status`

Команда `status` предоставляет информацию о текущем состоянии соединения и сервера MariaDB. Из вывода видно, что:

- Используется MariaDB версии 10.11.11
- Подключение выполнено локально через UNIX-сокеты
- Сервер работает 10 минут 38 секунд
- Наблюдается несоответствие кодировок (сервер использует `latin1`, а клиент - `utf8mb3`)
- Нагрузка на сервер минимальная (в среднем 0.045 запросов в секунду) (рис. 19)


```

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

Connection id:          15
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:                 10 min 38 sec

Threads: 1  Questions: 29  Slow queries: 0  Opens: 20  Open tables: 13  Queries per second avg: 0.045

```

Рис. 3.19: Статус MariaDB

В каталоге /etc/my.cnf.d создадим файл utf8.cnf:

cd /etc/my.cnf.d

touch utf8.cnf (рис. 20)

```

[root@server.svivanov.net my.cnf.d]# cd /etc/my.cnf.d
[root@server.svivanov.net my.cnf.d]# touch utf8.cnf
[root@server.svivanov.net my.cnf.d]# vim utf8.cnf

```

Рис. 3.20: Создание utf8.cnf

Откроем его на редактирование и укажем в нём следующую конфигурацию:
(рис. 21)

```

[client]
default-character-set = utf8
[mysqld]
character-set-server = utf8
~

```

Рис. 3.21: Редактирование utf8.cnf

Перезапустим MariaDB: systemctl restart mariadb. (рис. 22)

```
[root@server.svivanov.net my.cnf.d]# systemctl restart mariadb
[root@server.svivanov.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
```

Рис. 3.22: Перезапуск базы данных

Войдем в базу данных с правами администратора и посмотрим статус MariaDB.

Изменение конфигурации:

- Кодировка сервера изменилась с latin1 на utf8mb3 - это указывает на то, что после перезапуска сервер начал использовать корректные настройки кодировки из конфигурационных файлов
- Сервер работает стабильно, время работы - 23 секунды
- Активно 1 подключение
- Медленных запросов не зафиксировано
- Нагрузка на сервер минимальная
- Перезапуск службы успешно применен и сервер MariaDB работает с обновленными настройками, включая корректную UTF-8 кодировку. (рис. 23)

```
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:                  23 sec

Threads: 1  Questions: 4  Slow queries: 0  Opens: 17  Open tables: 10  Queries per second avg: 0.173
```

Рис. 3.23: Просмотр статуса

3.3 Создание базы данных

Войдем в базу данных с правами администратора, Создадим базу данных с именем addressbook:

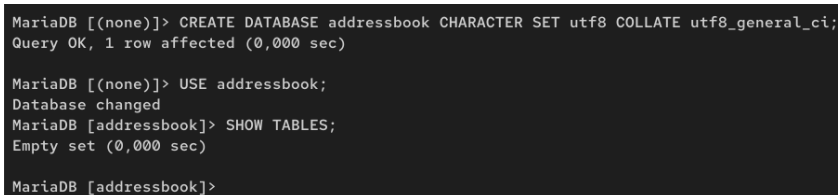
```
CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Перейдем к базе данных addressbook

```
USE addressbook;
```

Отообразим имеющиеся в базе данных addressbook таблицы:

```
SHOW TABLES;(рис. 24)
```



```
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]>
```

Рис. 3.24: Создание базы данных

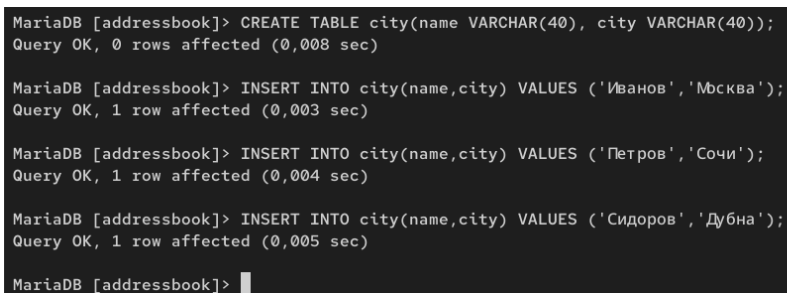
Создадим таблицу city с полями name и city:

```
CREATE TABLE city(name VARCHAR(40), city VARCHAR(40));
```

Заполним несколько строк таблицы некоторыми данными по аналогии в соответствии с синтаксисом MySQL:

```
INSERT INTO city(name,city) VALUES ('Иванов','Москва');
```

В частности, добавим в базу сведения о Петрове и Сидорове: (Петров, Сочи), (Сидоров, Дубна) (рис. 25)



```
MariaDB [addressbook]> CREATE TABLE city(name VARCHAR(40), city VARCHAR(40));
Query OK, 0 rows affected (0,008 sec)

MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Иванов','Москва');
Query OK, 1 row affected (0,003 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 (0,005 sec)

MariaDB [addressbook]> █
```

Рис. 3.25: Создание и наполнение таблицы

Сделаем следующий MySQL-запрос: `SELECT * FROM city;`

Запрос `SELECT * FROM city` успешно выполнен и показал содержимое таблицы `city` из базы данных `addressbook`.

Результат демонстрирует:

- Таблица содержит справочник “Фамилия - Город проживания”
- В текущий момент в таблице зарегистрировано 3 человека
- Все данные отображаются корректно, включая кириллические символы (что подтверждает правильную настройку кодировки UTF-8 после перезапуска сервера) (рис. 26)

```
MariaDB [addressbook]> SELECT * FROM city;
+-----+-----+
| name   | city   |
+-----+-----+
| Иванов | Москва |
| Петров | Сочи   |
| Сидоров| Дубна  |
+-----+-----+
3 rows in set (0,001 sec)

MariaDB [addressbook]> 
```

Рис. 3.26: Запрос к базе

Создадим пользователя для работы с базой данных `addressbook` и зададим для него пароль:

```
CREATE USER user@'%' IDENTIFIED BY 'password';
```

Предоставим права доступа созданному пользователю `user` на действия с базой данных `addressbook` (просмотр, добавление, обновление, удаление данных):

```
GRANT SELECT,INSERT,UPDATE,DELETE ON addressbook.* TO user@'%'; (рис. 27)
```

```
MariaDB [addressbook]> CREATE USER svivanov@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0,004 sec)

MariaDB [addressbook]> GRANT SELECT,INSERT,UPDATE,DELETE ON addressbook.* TO svivanov@'%';
Query OK, 0 rows affected (0,002 sec)
```

Рис. 3.27: Создание пользователя

Обновим привилегии (права доступа) базы данных addressbook: FLUSH PRIVILEGES; Посмотрим общую информацию о таблице city базы данных addressbook: DESCRIBE city; (рис. 28)

```
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 (0,002 sec)
```

Рис. 3.28: Обновление привелегий и просмотр базы

Посмотрим список баз данных: mysqlshow -u root -p. Посмотрим список таблиц базы данных addressbook: mysqlshow -u root -p addressbook (рис. 29)

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

Рис. 3.29: Список баз и таблиц

3.4 Резервные копии

На виртуальной машине server создадим каталог для резервных копий:

```
mkdir -p /var/backup
```

Сделаем резервную копию базы данных addressbook:

```
mysqldump -u root -p addressbook > /var/backup/addressbook.sql
```

Сделаем сжатую резервную копию базы данных addressbook:

```
mysqldump -u root -p addressbook | gzip > /var/backup/addressbook.sql.gz
```

Сделаем сжатую резервную копию базы данных addressbook с указанием даты создания копии:

```
mysqldump -u root -p addressbook | gzip > $(date+var/backup/addressbook.%Y%m%d.%H%M%S
```

(рис. 30)

```
[root@server svivanov.net my.cnf.d]# mkdir -p /var/backup
[root@server svivanov.net my.cnf.d]# mysqldump -u root -p addressbook > /var/backup/addressbook.sql
Enter password:
[root@server svivanov.net my.cnf.d]# mysqldump -u root -p addressbook | gzip > /var/backup/addressbook.sql.gz
Enter password:
[root@server svivanov.net my.cnf.d]# mysqldump -u root -p addressbook | gzip > $(date+var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz)
-bash: date+var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz: Нет такого файла или каталога
-bash: $(date+var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz): ambiguous redirect
Enter password:
mysqldump: Got errno 32 on write
```

Рис. 3.30: Резервные копии

Восстановим базу данных addressbook из резервной копии:

```
mysql -u root -p addressbook < /var/backup/addressbook.sql
```

Восстановим базу данных addressbook из сжатой резервной копии:

```
zcat /var/backup/addressbook.sql.gz | mysql -u root -p addressbook (рис. 31)
```

```
[root@server svivanov.net my.cnf.d]# mysql -u root -p addressbook < /var/backup/addressbook.sql
Enter password:
[root@server svivanov.net my.cnf.d]# zcat /var/backup/addressbook.sql.gz | mysql -u root -p addressbook
Enter password:
[root@server svivanov.net my.cnf.d]#
```

Рис. 3.31: Восстановление базы данных

На виртуальной машине server перейдем в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создадим в нём каталог mysql, в который поместим в соответствующие подкаталоги конфигурационные файлы MariaDB и резервную копию базы данных addressbook:

```
cd /vagrant/provision/server
```

```
mkdir -p /vagrant/provision/server/mysql/etc/my.cnf.d
```

```
mkdir -p /vagrant/provision/server/mysql/var/backup
```

```
cp -R /etc/my.cnf.d/utf8.cnf /vagrant/provision/server/mysql/etc/my.cnf.d/
```

```
cp -R /var/backup/* /vagrant/provision/server/mysql/var/backup/ (рис. 32)
```

```
[root@server.svivanov.net my.cnf.d]# cd /vagrant/provision/server
[root@server.svivanov.net server]# mkdir -p /vagrant/provision/server/mysql/etc/my.cnf.d
[root@server.svivanov.net server]# mkdir -p /vagrant/provision/server/mysql/var/backup
[root@server.svivanov.net server]# cp -R /etc/my.cnf.d/utf8.cnf/vagrant/provision/server/mysql/etc/my.cnf.d/
cp: missing destination file operand after '/etc/my.cnf.d/utf8.cnf/vagrant/provision/server/mysql/etc/my.cnf.d/'
Try 'cp --help' for more information.
[root@server.svivanov.net server]# cp -R /etc/my.cnf.d/utf8.cnf /vagrant/provision/server/mysql/etc/my.cnf.d/
[root@server.svivanov.net server]# cp -R /var/backup/* /vagrant/provision/server/mysql/var/backup/
[root@server.svivanov.net server]#
```

Рис. 3.32: Создание каталогов для внесения изменений

В каталоге /vagrant/provision/server создадим исполняемый файл mysql.sh:

```
cd /vagrant/provision/server
```

```
touch mysql.sh
```

```
chmod +x mysql.sh
```

Открыв его на редактирование, пропишем в нём следующий скрипт: (рис. 33)

```
#!/bin/bash
echo "Provisioning script $0"
systemctl restart named
echo "Install needed packages"
dnf -y install mariadb mariadb-server
echo "Copy configuration files"
cp -R /vagrant/provision/server/mysql/etc/* /etc
mkdir -p /var/backup
cp -R /vagrant/provision/server/mysql/var/backup/* /var/backup
echo "Start mysql service"
systemctl enable mariadb
systemctl start mariadb
if [[ ! -d /var/lib/mysql/mysql ]]
then
echo "Securing mariadb"
mysql_secure_installation <<EOF
y
123456
123456
y
y
y
y
EOF
-- РЕЖИМ ВСТАВКИ --
```

Рис. 3.33: Скрипт mysql.sh

Для отработки созданного скрипта во время загрузки виртуальных машин в конфигурационном файле Vagrantfile необходимо добавить в конфигурации сервера следующую запись (рис. 34)

```
server.vm.provision "server mysql",  
    type: "shell",  
    preserve_order: true,  
    path: "provision/server/mysql.sh"
```

Рис. 3.34: Запись в Vagrantfile

4 Ответы на контрольные вопросы

1. Какая команда отвечает за настройки безопасности в MariaDB?

`mysql_secure_installation`

2. Как настроить MariaDB для доступа через сеть?

В конфигурационном файле `/etc/my.cnf.d/mariadb-server.cnf`:

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

`SHOW DATABASES;`

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

`USE database_name;`

`SHOW TABLES;`

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

`DESCRIBE table_name;`

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

`SELECT * FROM table_name;`

7. Как удалить запись из таблицы?

`DELETE FROM table_name WHERE condition;`

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

`/etc/my.cnf` - главный конфигурационный файл

`/etc/my.cnf.d/` - каталог с модульными конфигурационными файлами

9. Где располагаются файлы с базами данных MariaDB?

/var/lib/mysql/

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

Создание резервной копии:

```
mysqldump -u username -p database_name > backup.sql
```

```
mysqldump -u username -p --all-databases > full_backup.sql
```

Восстановление из резервной копии:

```
mysql -u username -p database_name < backup.sql
```

```
mysql -u username -p < full_backup.sql
```

5 Выводы

В ходе выполнения лабораторной работы мы приобрели практические навыки по установке и конфигурированию системы управления базами данных на примере программного обеспечения MariaDB.