

初识 MySQL

2353113 李阔

2024 年 9 月 15 日

目录

1	MySQL 的安装	2
1.1	配置 MySQL	3
2	MySQL 数据库的基本使用	3
2.1	启动 MySQL 数据库服务	3
2.2	停止 MySQL 数据库服务	3
2.3	查看 MySQL 运行状态	3
3	MySQL 创建用户与授权	3
3.1	使用 root 用户登陆	3
4	使用数据库	3
4.1	显示所有数据库	3
4.2	新建数据库	5
4.3	删除数据库	5
4.4	选择数据库	7
4.5	创建数据表	7
4.6	删除数据表	7
4.7	插入数据	8
4.7.1	插入单个数据	8
4.7.2	插入多组数据	8
4.8	查找数据	8
4.9	UPDATE 更新	9
5	总结	10

1 MySQL 的安装

因为 linux 操作系统对于编程者的优越性，本报告的 mysql 安装部分主要介绍在 linux 虚拟机上安装 MySQL，而 Windows 操作系统的 MySQL 的安装过程则会简要介绍。先介绍 linux 虚拟机的安装，我了解到个人用户安装 linux 操作系统一般会使用 Ubuntu，于是我按照网上的各种教程配置了基于 linux 内核的 Ubuntu 虚拟机，设置如下图：

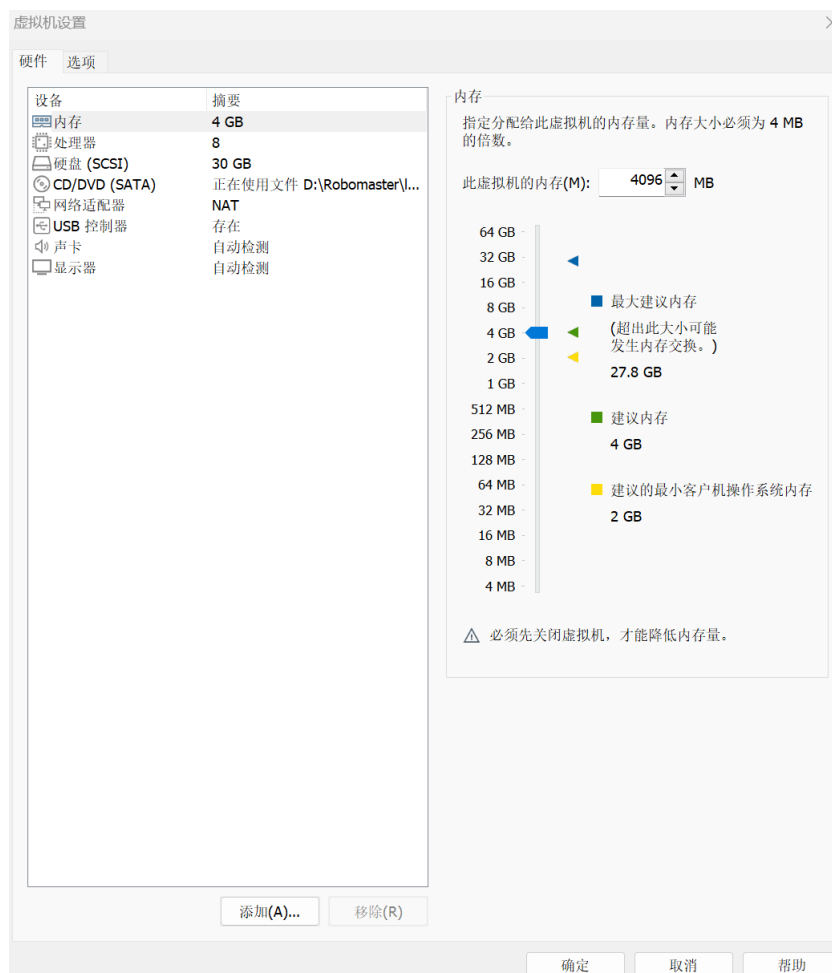


图 1: 虚拟机参数设置

成功安装虚拟机后，在 linux 命令行窗口继续执行以下命令：

```
1      sudo apt-get updat
2      sudo apt-get install mysql-server
3      sudo apt-get install mysql-client
```

注：mysql-server 是 MySQL 核心程序将安装 MySQL 数据库服务器，用于生成管理多个数据库实例，持久保存数据并将其提供查询接口（SQL），供不同客户端调用。

mysql-client 是操作数据库实例的工具，允许连接到 MySQL 服务器使用该查询接口。它将为您提供 MySQL 命程序。

如果只需要连接到远程服务器并运行查询，只安装 mysql-client 就可以了。如果是服务器

只提供连接服务的只需要安装 mysql-server

1.1 配置 MySQL

运行初始化安全脚本

```
1 sudo mysql_secure_installation
```

2 MySQL 数据库的基本使用

2.1 启动 MySQL 数据库服务

```
1 sudo service mysql start
```

2.2 停止 MySQL 数据库服务

```
1 sudo service mysql stop
```

2.3 查看 MySQL 运行状态

```
1 sudo service mysql status
```

3 MySQL 创建用户与授权

3.1 使用 root 用户登陆

第一次以 root 用户在虚拟机登陆 mysql 时已设置了默认密码，需要手动查看，即在 linux 命令行窗口输入一下指令：

```
1 sudo cat /etc/mysql/debian.cnf
```

记住密码后再以以下指令登陆：（输入在指令后会要求输入密码）

```
1 mysql -u root -p
```

可以将密码修改为个人常用密码

4 使用数据库

4.1 显示所有数据库

```
1 SHOW DATABASES;
```

```
rodebiau@rodebiau-virtual-machine:~$ sudo cat /etc/mysql/debian.cnf
[sudo] rodebiau 的密码:
# Automatically generated for Debian scripts. DO NOT TOUCH!
[client]
host      = localhost
user      = debian-sys-maint
password  = H99IBJRuW7tu3VLT
socket    = /var/run/mysqld/mysqld.sock
[mysql_upgrade]
host      = localhost
user      = debian-sys-maint
password  = H99IBJRuW7tu3VLT
socket    = /var/run/mysqld/mysqld.sock
```

图 2: 查找默认密码

```
rodebiau@rodebiau-virtual-machine:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.39-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

图 3: 登陆成功

```
mysql> SHOW DATABASES;
+-----+
| Database          |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+-----+
4 rows in set (0.01 sec)
```

图 4: Enter Caption

4.2 新建数据库

```
1 CREATE DATABASE TEST1;
```

```
mysql> CREATE DATABASE TEST1;
Query OK, 1 row affected (0.01 sec)
```

图 5: 数据库列表

再输入显示所有数据库命令，就可以看到我们新建的数据库出现在列表中：

4.3 删除数据库

```
1 DROP DATABASE TEST1;
```

再输入 SHOW DATABASES; 的语句就可以发现，我们刚刚创建的数据库已经被删除了

```
mysql> SHOW DATABASES;
+-----+
| Database          |
+-----+
| TEST1             |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+-----+
5 rows in set (0.00 sec)
```

图 6: 数据库列表

```
mysql> DROP DATABASE TEST1;
Query OK, 0 rows affected (0.04 sec)
```

图 7: 删除数据库

4.4 选择数据库

```
1 create DATABASE demo1;
2 use demo1;
```

```
mysql> create DATABASE demo1;
Query OK, 1 row affected (0.01 sec)

mysql> use demo1;
Database changed
mysql>
```

图 8: 创建并选择数据库

4.5 创建数据表

```
1 CREATE TABLE users (
2     id INT AUTO_INCREMENT PRIMARY KEY,
3     username VARCHAR(50) NOT NULL,
4     birthdate DATE,
5     is_active BOOLEAN DEFAULT TRUE
6 );
```

```
mysql> create TABLE users( id INT AUTO_INCREMENT PRIMARY KEY, username VARCHAR(50) NOT NULL, birthdate DATE, is_active BOOLEAN DEFAULT TRUE);
Query OK, 0 rows affected (0.07 sec)
```

图 9: 创建数据表

id: 用户 id, 整数类型, 自增长, 作为主键

username: 用户名, 变长字符串, 不允许为空

birthdate: 用户的生日, 日期类型

is_active: 用户是否已经激活, 布尔类型, 默认值为 true

4.6 删除数据表

```
1 drop TABLE users;
```

4.7 插入数据

4.7.1 插入单个数据

```
1 INSERT INTO users (username, birthdate, is_active)
2 VALUES ('test1', '1990-01-01', true);
```

```
mysql> INSERT INTO users(username,birthdate,is_active)
-> VALUES('test1','1990-01-01',true);
Query OK, 1 row affected (0.05 sec)
```

图 10: 插入单个数据

4.7.2 插入多组数据

```
1 INSERT INTO users (username, birthdate, is_active)
2 VALUES
3     ('test2', '1985-07-10', true),
4     ('test3', '1988-11-25', false),
5     ('test4', '1993-05-03', true);
```

```
mysql> INSERT INTO users(username,birthdate,is_active)
-> VALUES
-> ('test2','1985-07-10',true),
-> ('test3','1988-11-25',false),
-> ('test4','1993-05-03',true);
Query OK, 3 rows affected (0.02 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

图 11: 插入多组数据

4.8 查找数据

```
1 -- 使用 where 子句
2 SELECT * FROM users WHERE is_active = true;
```



```
mysql> SELECT * FROM users WHERE is_active=true;
+----+-----+-----+-----+
| id | username | birthdate | is_active |
+----+-----+-----+-----+
| 1 | test1 | 1990-01-01 | 1 |
| 2 | test2 | 1985-07-10 | 1 |
| 4 | test4 | 1993-05-03 | 1 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)
```

图 12: 查找激活状态的用户

```
1 -- 使用 or 逻辑符
2 SELECT * FROM users WHERE is_active = TRUE OR birthdate < '1990-01-01';
```

```
mysql> SELECT * FROM users WHERE is_active=true OR birthdate<'1990-01-01';
+----+-----+-----+-----+
| id | username | birthdate | is_active |
+----+-----+-----+-----+
| 1 | test1 | 1990-01-01 | 1 |
| 2 | test2 | 1985-07-10 | 1 |
| 3 | test3 | 1988-11-25 | 0 |
| 4 | test4 | 1993-05-03 | 1 |
+----+-----+-----+-----+
4 rows in set (0.00 sec)
```

图 13: 使用 or 查询

4.9 UPDATE 更新

```
1 UPDATE users
2 SET is_active=false
3 WHERE username='test1';
```

然后我们再查询激活状态为 false 的用户，则包括了 test1，说明我们的 update 生效了

```
mysql> UPDATE users
      -> SET is_active=false
      -> WHERE username='test1';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

图 14: 将 test1 用户的激活状态更新为 false

```
mysql> SELECT * FROM users WHERE is_active = false;
+----+-----+-----+-----+
| id | username | birthdate | is_active |
+----+-----+-----+-----+
|  1 | test1    | 1990-01-01 |         0 |
|  3 | test3    | 1988-11-25 |         0 |
+----+-----+-----+-----+
2 rows in set (0.00 sec)
```

图 15: 显示激活状态为 false 的用户

5 总结

坦白说，这次报告的前期还是比较波折的，涉及到操作系统的选择，以及排版工具的选择。考虑到之前没有怎么接触过 linux 也没有装过虚拟机，便想以此次作业为契机，在我的电脑上安装 linux 内核的虚拟机，而操作系统的选择，我曾问过学长，自己也在网上查询过资料，最终在 centOS 和 Ubuntu 之间选择了 Ubuntu，因为我了解到 centOS 较多用于企业服务器端，而初 linux 初学者以及个人用户一般更倾向于使用 Ubuntu，因为 Ubuntu 有比较简单的安装过程和图形化界面以及社区论坛，对新手更加友好。

而排版工具的选择也是摇摆不定，一开始选择 latex，因为其美观并且我之前曾使用过这两大原因我开始着手在本地的 latex 编译器 Texstudio 上进行 latex 的编译工作，而由于很长时间不使用 latex，发现我在有一些排版想法时却不能实现，出现各种报错，于是我开始逐渐放弃使用 latex 排版，进而选择学习成本更低更加轻量的 markdown 语言进行排版，而上手后却发现版面实在是不太美观，而我意识到语气花费一定的学习成本去使 markdown 排版达到一定的美观程度，不如使用 latex，如果能比较熟练地掌握 latex，其在以后的其他课的报告撰写上将受益无穷，所以我又转而使用了在线的 latex 编译器 overleaf 内嵌了 XeLatex。

而基本的 MySQL 语句的学习到没有遇到太大的障碍，就是第一次登陆时查找初始默认密码时看了不少教程，花费了我不少时间。

而目前，以我的个人理解，我了解到类似于 MySQL 这样数据库的应用要与前端的接口进行匹配，比如作为一个网站或者一个小程序的后端，而这些都是我的盲区，所以我现在的课余学习目标就暂时为学习一定简单的前端技术，并着重学习前后端的接口以及后端的数据处理。

这次报告只是载体，而他真正地让我在撰写报告中学会很多东西，比如 linux 虚拟机的基本知识，MySQL 的基本语句等等。