MySQL数据系统部署及SQL语句基础

1.数据库的基本概念

2.SQL语句分类:

1、数据库定义语言(DDL) database define language

2、数据库查询语言(DQL) database querry language

3、数据库操纵语言(DML) database manage language

4、数据库控制语言(DCL) database contrl language

3.操作演示:

1.查看当前服务器中有哪些库

2.查看数据库中的数据表信息

3.显示当前操作的数据库

4.查看当前库下有哪些表

5.显示数据表的结构 (字段(列))

6.创建新的数据库

7.创建新的数据表

8.删表

9.删数据库

10.向数据表中插入新的数据记录---往表里面填写一行信息

11.从数据表中查找符合条件的数据记录

12.修改、更新数据表中的数据记录

13.在数据表中删除指定的数据记录

14.设置用户权限(用户不存在时，则新建用户)

MySQL数据系统部署及SQL语句基础

MySQL是一套关系型数据库管理系统

- 在每台MySQL服务器中，均支持运行多个数据库(文件系统中的目录)

（分类，按照产品分）

- 每个库相当于一 个容器，其中存放着许多数据表
- 表中的每一行包含一条具体的数据关系信息， 这些信息被称为数据记录

1.数据库的基本概念

数据

- 描述事物的符号记录称为数据(Data)
- 包括数字，文字、图形、图像、声音、档案记录等
- 以记录"形式按统-的格式进行存储

数据表

- 将不同的记录组织在一起，就形成了“表”
- 是用来存储具体数据的

数据库

- 数据库就是表的集合，是存储数据表的仓库
- 以一定的组织方式存储的相互有关的数据

2.SQL语句分类:

1、数据库定义语言(DDL) database define language

含义：创建、修改或删除数据库中各种对象，包括表、视图、索引等

命令：CREATE TABLE ， CREATE VIEW, CREATE INDEX、ALTER TABLE，DROP TABLE ， DROP VIEW, DROP INDEX

2、数据库查询语言(DQL) database querry language

含义：按照指定的组合、条件表达式或排序检索已存在的数据库中数据，不改变数据库中数据

命令：SELECT...FROM..WHERE...

3、数据库操纵语言(DML) database manage language

含义：对已经存在的数据库进行元组的插入、删除、修改等操作

命令：INSERT、UPDATE、 DELETE

4、数据库控制语言(DCL) database contrl language

含义：用来授予或收回访问数据库的某种特权

控制数据操纵事务的发生时间及效果、对数据库进行监视

命令：GRANT、REVOKE、 COMMIT、ROLLBACK

3.操作演示：

1.查看当前服务器中有哪些库

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
  # 环境兼容性，mysql运行当中位于内存中的信息，关机后是空的     在磁盘上看不到它
| mysql              |
```

初始化 就是在MySQL库下生成了各种各样的表格和各种数据

用户名和密码也保留在MySQL库的一张表里面

这里面的配置是当前数据库软件的运行环境以及一些用户的相关授权信息

包含mysq|数据库软件的用户认证相关表

```
| performance_schema |       与性能相关
| sys                |
+--------------------+
4 rows in set (0.02 sec)
```

2.查看数据库中的数据表信息

```
mysq|> use mysql                切换数据库
Database changed
```

3.显示当前操作的数据库

```
mysq|> select database();
```

4.查看当前库下有哪些表

```
mysql> show tables;
```

MySQL数据库的数据文件存放在/usrlocal/mysql/data中,每个子目录对应一个数据库,

MyISAM

存储引擎时，每个表对应三个文件：

[root@localhost ~]# Is /usrlocal/mysql/data/mysql lgrep ^user

user.frm 表的结构定义

user.MYD 表的数据

user.MYl 表的索引

5.显示数据表的结构 (字段(列))

主键：防止后续查询出问题

DESCRIBE [数据库名.]表名= desc （可以简写）

mysq|> DESCRIBE user;

如果没有进入到库中的时候可以用：

mysql> DESCRIBE mysql.user; --库名.表名

6.创建新的数据库

CREATE DATABASE数据库名;

mysql> create database soifa;

mysql> show databases;

mysql> use soifa;

7.创建新的数据表

GREATE TABLE 表名 (字段1名称 类型，字段2名称 类型，.... primary key (字段名称));

案例:

mysq|> use crushlinux

Database changed

mysql> create table users(user_name char(18) not null,user_passwd char(48) default '',primary key(user_name));

Query OK, 0 rows affected (0.11 sec)

用户名不超过16字节的字符串，不能为空;

密码字串不能超过48个字符，插入记录时使用MySQL函数加密，默认值为空;

考虑到查询的主要用户名不能重复，因此将user_name这列设置为主键

---

mysql> desc users;

```
+-------------+----------+------+-----+---------+-------+
| Field       | Type     | Null | Key | Default | Extra |
+-------------+----------+------+-----+---------+-------+
| user_name   | char(18) | NO   | PRI | NULL    |       |
| user_passwd | char(48) | YES  |     |         |       |
+-------------+----------+------+-----+---------+-------+
```

2 rows in set (0.04 sec)

8.删表

```
mysql> drop tables users;
Query OK, 0 rows affected (0.01 sec)
mysql> show tables;
Empty set (0.00 sec)
```

9.删数据库

```
mysql> drop database soifa;
Query OK, 0 rows affected (0.01 sec)
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)
```

再重新创建回来：

```
mysql> create database sofia;
Query OK, 1 row affected (0.00 sec)
mysql> use sofia;
Database changed
mysql> create table users(user_name char(18) not null,user_passwd char(48)
default '',primary key(us
er_name));Query OK, 0 rows affected (0.01 sec)
```

10.向数据表中插入新的数据记录---往表里面填写一行信息

INSERT INTO 表名(字段1,字段2, ……) VALUES(字段1的值，字段2的值,……);

```
mysql> insert into users(user_name,user_passwd) values('zhangsan','123456');
mysql> insert into users(user_name,user_passwd) values('lisi','123456');
mysql> insert into users(user_name,user_passwd)
values('wangwu',password('123456'));
mysql> insert into users values('cc',password('123456'));
```

当只有两列的时候，直接写不用指定也行

```
mysql> insert into users values('bb',password('123456')),
('aa',password('123456'));
```

也可直接创建两个用户

```
mysql> select * from users;
+-----------+-------------------------------------------+
| user_name | user_passwd                               |
+-----------+-------------------------------------------+
| aa        | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| bb        | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| cc        | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| lisi      | 123456                                    |
| wangwu    | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| zhangsan  | 123456                                    |
+-----------+-------------------------------------------+
6 rows in set (0.00 sec)
```

11.从数据表中查找符合条件的数据记录

SELECT 字段名1,字段名2 ..... FROM 表名 WHERE 条件表达式

案例：

```
mysql> select * from users;
+-----------+-------------------------------------------+
| user_name | user_passwd                               |
+-----------+-------------------------------------------+
| aa        | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| bb        | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| cc        | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| lisi      | 123456                                    |
| wangwu    | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| zhangsan  | 123456                                    |
+-----------+-------------------------------------------+
6 rows in set (0.00 sec)
```

```
mysql> select user_name from users;
+-----------+
| user_name |
+-----------+
| aa        |
```

```
| bb        |
| cc        |
| lisi      |
| wangwu    |
| zhangsan  |
+-----------+
6 rows in set (0.00 sec)
```

```
mysql> select user_name from users;where user_name='zhangsan';
+-----------+
| user_name |
+-----------+
| aa        |
| bb        |
| cc        |
| lisi      |
| wangwu    |
| zhangsan  |
+-----------+
6 rows in set (0.00 sec)
```

```
mysql> select user_name from users where user_name='zhangsan';
+-----------+
| user_name |
+-----------+
| zhangsan  |
+-----------+
1 row in set (0.00 sec)
```

```
mysql> select user_name,user_passwd from users where user_name='zhangsan';
+-----------+-------------+
| user_name | user_passwd |
+-----------+-------------+
| zhangsan  | 123456      |
+-----------+-------------+
1 row in set (0.00 sec)
```

12.修改、更新数据表中的数据记录

```
UPDATE 表名 SET 字段名1=新值1[,字段名2=新值2] WHERE 条件表达式;
```

```
mysql> update users set user_passwd=password('123456') where
user_name='zhangsan';
Query OK, 1 row affected, 1 warning (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 1
mysql> select * from users;
+-----------+-------------------------------------------+
| user_name | user_passwd                               |
+-----------+-------------------------------------------+
| aa        | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| bb        | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| cc        | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| lisi      | 123456                                    |
| wangwu    | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| zhangsan  | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
+-----------+-------------------------------------------+
6 rows in set (0.00 sec)
```

```
mysql> select user,authentication_string from mysql.user;
+---------------+-------------------------------------------+
| user          | authentication_string                     |
+---------------+-------------------------------------------+
| root          | *E56A114692FE0DE073F9A1DD68A00EEB9703F3F1 |
| mysql.session | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE |
| mysql.sys     | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE |
+---------------+-------------------------------------------+
3 rows in set (0.00 sec)
mysql> flush privileges;
```

13.在数据表中删除指定的数据记录

```
DELETE FROM 表名 WHERE 条件表达式;
mysql> delete from users where user_name='zhangsan';
Query OK, 1 row affected (0.00 sec)
mysql> delete from users where user_name='lisi';
Query OK, 1 row affected (0.00 sec)
mysql> select * from users;
+-----------+-------------------------------------------+
| user_name | user_passwd                               |
```

```
+----------+-------------------------------------------+
| aa       | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| bb       | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| cc       | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| wangwu   | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
+----------+-------------------------------------------+
4 rows in set (0.00 sec)
```

14.设置用户权限(用户不存在时，则新建用户)

MySQL数据库的root用户账户拥有对所有库，表的全部权限

频繁使用root账号会给数据库服务器带来一定的安全风险（root密码还有可能会泄露）

实际工作中，通常会建立一些低权限的用户，只负责一部分库， 表的管理和维护操作，甚至可以对查询，修改,删除记录等各种操作做进一步的细化限制，从而降低数据库的风险

GRANT 权限列表 ON 数据库名.表名 TO 用户名@来源地址[IDENTIFIED BY ’ 密码 ‘];

权限列表:用于列出授权的各种数据库操作，通过逗号进行分割，如:

select,insert,update等， all表示所有权限，可以执行任意操作

库名.表名:用于指定授权操作的数据库和表的名称，可以使用通配符(*)表示所有

用户名@来源地址:用于指定用户和允许访问的客户机地址;来源地址可以是IP地址,域名，%通配符表示所有(但不能表示localhost)

MySQL通配符:

_ : 任意单个字符192.168.1._

%: 任意长度的任意字符192.168.1.%

```
mysql> select database();
+------------+
| database() |
+------------+
| sofia      |
+------------+
1 row in set (0.00 sec)
mysql> show tables;
+----------------+
| Tables_in_sofia |
+----------------+
| users          |
+----------------+
1 row in set (0.00 sec)
```

授权这个操作只有管理员可以给别的用户做

```
mysql> grant select on sofia.* to 'teacher'@'localhost' identified by '123456';
Query OK, 0 rows affected, 1 warning (0.01 sec)
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
[root@localhost ~]# mysql -uteacher -p123456            登录
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| sofia              |
+--------------------+
2 rows in set (0.00 sec)

mysql> select * from sofia.users;
+-----------+-------------------------------------------+
| user_name | user_passwd                               |
+-----------+-------------------------------------------+
| aa        | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| bb        | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| cc        | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| wangwu    | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
+-----------+-------------------------------------------+
4 rows in set (0.00 sec)
[root@localhost ~]# mysql -uteacher -p123456
mysql> show grants for 'teacher'@'localhost';
+-------------------------------------------------+
| Grants for teacher@localhost                    |
+-------------------------------------------------+
| GRANT USAGE ON *.* TO 'teacher'@'localhost'     |
| GRANT SELECT ON `sofia`.* TO 'teacher'@'localhost' |
+-------------------------------------------------+
2 rows in set (0.00 sec)
mysql> revoke select on `sofia`.* to 'teacher'@'localhost'   撤销授权
```

```
mysql> select user,host from mysql.user;
+---------------+-----------+
| user          | host      |
+---------------+-----------+
| mysql.session | localhost |
| mysql.sys     | localhost |
| root          | localhost |
| teacher       | localhost |
+---------------+-----------+
4 rows in set (0.00 sec)
```

授权的时候创建的用户， 保存到这里了

使用说明书：

```
mysql> help create database;
mysql> help create table;
mysql> show grants;
```

查看哪些用户得到授权

显示服务器错误或警告信息

```
show errors;
show warnings;
mysql> select now();
```

查看当前时间

```
mysql> select user();
mysql> select database();
```

```
mysql> show processlist;
```

查看目前有哪些任务在执行的：关乎性能的语句

```
+----+---------+-----------+-------+---------+------+----------+-------------
-----+
| Id | User    | Host      | db    | Command | Time | State    | Info
|
+----+---------+-----------+-------+---------+------+----------+-------------
-----+
|  7 | root    | localhost | sofia | Query   |    0 | starting | show
processlist |
|  9 | teacher | localhost | NULL  | Sleep   | 1698 |          | NULL
|
```

```
+----+---------+-----------+-------+---------+------+----------+-------------
-----+
2 rows in set (0.00 sec)
```