

## 0409库与表的操作

## 1.5 SQL分类

- DDL【data definition language】数据定义语言，用来维护存储数据的**结构**

代表指令: `create, drop, alter`

- DML【data manipulation language】数据操纵语言，用来对**数据**进行操作

代表指令: `insert, delete, update`

- DML中又单独分了一个DQL，数据查询语言，代表指令: `select`

- DCL【Data Control Language】数据控制语言，主要负责权限管理和事务

代表指令: `grant, revoke, commit`

### 1.6.2 查看存储引擎

```
show engines;
```

## 库的操作

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| lesson1_db |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> █
```

```
(root)[root@VM-12-12-centos:~]$ cd /var/lib/mysql
(root)[root@VM-12-12-centos:/var/lib/mysql]$ ls
auto.cnf          ib_logfile0      performance_schema
ca-key.pem        ib_logfile1      private_key.pem
ca.pem            ibtmp1           public_key.pem
client-cert.pem   lesson1_db       server-cert.pem
client-key.pem    mysql            server-key.pem
ib_buffer_pool    mysql.sock       sys
ibdata1           mysql.sock.lock
```

## 删除一个数据库

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| lesson1_db |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)
```

```
mysql> drop database lesson1_db;
Query OK, 1 row affected (0.04 sec)
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)
```

```
mysql> █
```

```
mysql> create database lesson2_db;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> show create database lesson2_db;
```

查看数据库的信息

```
+-----+-----+
| Database | Create Database |
+-----+-----+
| lesson2_db | CREATE DATABASE `lesson2_db` /*!40100 DEFAULT CHARACTER SET utf8 */ |
+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> █
```

mysql的名字可以用反引号弄起来

什么时候一定要呢？

当名称和mysql里一些关键字有重复的时候，就需要用反引号

1. mysql大小写不敏感
2. 建库建表的时候，名称一般建议带上``

```
mysql> show variables like 'character_set_database';
```

Variable_name	Value
character_set_database	utf8

1 row in set (0.01 sec)

一般字符集和校验集是匹配的

```
mysql> █
```

```
show variables like 'character_set_database'; 字符集
show variables like 'collation_database'; 校验集
```

## 2.1 创建数据库

语法:

```
CREATE DATABASE [IF NOT EXISTS] db_name [create_specification [, create_specification] ...]
```

create\_specification:

```
[DEFAULT] CHARACTER SET charset_name
```

```
[DEFAULT] COLLATE collation_name
```

说明:

- 大写的表示关键字
- [] 是可选项
- CHARACTER SET: 指定数据库采用的字符集
- COLLATE: 指定数据库字符集的校验规则

```
5 rows in set (0.01 sec)

mysql> create database d1;
Query OK, 1 row affected (0.00 sec)

mysql>
```

```
5 rows in set (0.01 sec)

mysql> create database d1;
Query OK, 1 row affected (0.00 sec)

mysql> create database d1 character set gbk;
ERROR 1007 (HY000): Can't create database 'd1'; database exists
mysql> create database d2 character set gbk;
Query OK, 1 row affected (0.00 sec)

mysql>
```

```
mysql> create database d1 character set gbk;
ERROR 1007 (HY000): Can't create database 'd1'; database exists
mysql> create database d2 character set gbk;
Query OK, 1 row affected (0.00 sec)

mysql> create database d3 charset=cp1256;
Query OK, 1 row affected (0.00 sec)

mysql>
```

```
auto.cnf          ib_logfile0      performance_schema
ca-key.pem        ib_logfile1      private_key.pem
ca.pem           ibtmp1           public_key.pem
client-cert.pem  lesson2_db       server-cert.pem
client-key.pem   mysql            server-key.pem
ib_buffer_pool   mysql.sock       sys
ibdata1          mysql.sock.lock

• (root)[root@VM-12-12-centos:/var/lib/mysql]$ tree lesson2_db/
lesson2_db/
└── db.opt

0 directories, 1 file
• (root)[root@VM-12-12-centos:/var/lib/mysql]$ cat d1/db.opt
default-character-set=utf8
default-collation=utf8_general_ci
• (root)[root@VM-12-12-centos:/var/lib/mysql]$
```

```
• (root)[root@VM-12-12-centos:/var/lib/mysql]$ tree lesson2_db/
lesson2_db/
└── db.opt

0 directories, 1 file
• (root)[root@VM-12-12-centos:/var/lib/mysql]$ cat d1/db.opt
default-character-set=utf8
default-collation=utf8_general_ci
• (root)[root@VM-12-12-centos:/var/lib/mysql]$ cat d2/db.opt
default-character-set=gbk
default-collation=gbk_chinese_ci
• (root)[root@VM-12-12-centos:/var/lib/mysql]$

ib_buffer_pool   mysql.sock       sys
ibdata1          mysql.sock.lock

• (root)[root@VM-12-12-centos:/var/lib/mysql]$ tree lesson2_db/
lesson2_db/
└── db.opt

0 directories, 1 file
• (root)[root@VM-12-12-centos:/var/lib/mysql]$ cat d1/db.opt
default-character-set=utf8
default-collation=utf8_general_ci
• (root)[root@VM-12-12-centos:/var/lib/mysql]$ cat d2/db.opt
default-character-set=gbk
default-collation=gbk_chinese_ci
• (root)[root@VM-12-12-centos:/var/lib/mysql]$ cat d3/db.opt
default-character-set=cp1256
default-collation=cp1256_general_ci
• (root)[root@VM-12-12-centos:/var/lib/mysql]$
```

这个db.opt里面存的，默认就是字符集和校验集

```
mysql> create database d4 charset=utf8 collate utf8_bin;
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql>
```

显式指明字符集和校验集

```
(root)[root@VM-12-12-centos:/var/lib/mysql]$ cat d1/db.opt
default-character-set=utf8
default-collation=utf8_general_ci
• (root)[root@VM-12-12-centos:/var/lib/mysql]$ cat d2/db.opt
default-character-set=gbk
default-collation=gbk_chinese_ci
• (root)[root@VM-12-12-centos:/var/lib/mysql]$ cat d3/db.opt
default-character-set=cpl256
default-collation=cpl256_general_ci
• (root)[root@VM-12-12-centos:/var/lib/mysql]$ cat d4/db.opt
default-character-set=utf8
default-collation=utf8_bin
• (root)[root@VM-12-12-centos:/var/lib/mysql]$
```

建议这么显式指明吗？

不建议！

除非你需要把这些数据库移植到别的平台，写名能防止出问题

### 2.3.4 校验规则对数据库的影响

- 不区分大小写

设置数据库的编码和校验规则，本质会影响谁？

其实是影响对应的数据库内部的表，所对应的编码和检验规则

```
mysql> use gc_db;
Database changed
mysql> create table `person`(name varchar(20));
Query OK, 0 rows affected (0.03 sec)

mysql> desc person;
```

Field	Type	Null	Key	Default	Extra
name	varchar(20)	YES		NULL	

1 row in set (0.00 sec)

desc相当于是查看表的属性

select \* 这里相当于查看所有表的内容

```
mysql> select * from person where name='a';
```

name
a
A

2 rows in set (0.00 sec)

```
mysql>
```

在终端中执行以下命令查看数据库配置：

```
(root)[root@VM-12-12-centos:/var/lib/mysql]$ cat gc_db/db.opt
default-character-set=utf8
default-collation=utf8_general_ci
```

我们可以发现，如果校验规则是utf8\_general\_ci的时候比较是忽略大小写的

```
mysql> insert into person values('c');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> show tables;
```

Tables_in_gc_db
person

1 row in set (0.00 sec)

```
mysql> select * from person;
```

name
a
A
b
B
C
c

6 rows in set (0.00 sec)

```
mysql>
```

```
mysql> use bin_db;
Database changed
mysql> show tables;
Empty set (0.00 sec)
```

如果换成bin\_db;这个数据库

```
mysql> create table `person`(name varchar(20));
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select * from person where name='a';
```

```
+-----+
| name |
+-----+
| a    |
+-----+
1 row in set (0.00 sec)
```

```
mysql> █
```



```
default-collation=utf8_general_ci
• (root)[root@VM-12-12-centos:/var/lib/mysql]$ cat bin_db/db.o
pt
default-character-set=utf8
default-collation=utf8_bin
○ (root)[root@VM-12-12-centos:/var/lib/mysql]$ █
```

我们发现，在这种校验规则下，是大小写敏感的！

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

```
+-----+
| database() |
+-----+
| bin_db    |
+-----+
1 row in set (0.00 sec)
```

```
mysql> █
```

这个命令是告诉我们在哪个数据库下的

相当于pwd

```
mysql> select * from person order by name;
```

```
+-----+
| name |
+-----+
| A    |
| B    |
| C    |
| a    |
| b    |
| c    |
+-----+
6 rows in set (0.00 sec)
```

```
mysql> █
```



# 修改数据库

## 2.4.2 修改数据库

语法:

```
ALTER DATABASE db_name
[alter_specification [,alter_specification]...]

alter_specification:
[DEFAULT] CHARACTER SET charset_name
[DEFAULT] COLLATE collation_name
```

```
mysql> select database();
+-----+
| database() |
+-----+
| gc_db      |
+-----+
1 row in set (0.00 sec)
```

```
mysql> alter database gc_db charset=gbk;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> 
```

```
• (root)[root@VM-12-12-centos:/var/lib/mysql]$ ls
auto.cnf          ibdata1           performance_schema
bin_db            ib_logfile0       private_key.pem
ca-key.pem        ib_logfile1       public_key.pem
ca.pem            ibtmp1            server-cert.pem
client-cert.pem   lesson2_db        server-key.pem
client-key.pem    mysql             sys
gc_db             mysql.sock
ib_buffer_pool    mysql.sock.lock
• (root)[root@VM-12-12-centos:/var/lib/mysql]$ cat gc_db/db.opt
+
default-character-set=utf8
default-collation=utf8_general_ci
• (root)[root@VM-12-12-centos:/var/lib/mysql]$ cat gc_db/db.opt
+
default-character-set=gbk
default-collation=gbk_chinese_ci
• (root)[root@VM-12-12-centos:/var/lib/mysql]$
```

# 数据库的备份

## 2.4.5 备份和恢复 -- 放在最后

### 2.4.5.1 备份

语法：

```
# mysqldump -P3306 -u root -p 密码 -B 数据库名 > 数据库备份存储的文件路径
```

示例：将mytest库备份到文件（退出连接）

```
# mysqldump -P3306 -u root -p123456 -B mytest > D:/mytest.sql
```

这时，可以打开看看 `mytest.sql` 文件里的内容，其实把我们整个创建数据库，建表，导入数据的语句都装载这个文件中。

数据库备份：

1. 对数据作备份
2. 或对操作语句作备份

mysqldump是把所有的操作语句作备份

# 数据库表的操作

## 3.1 创建表

语法:

```
CREATE TABLE table_name (  
    field1 datatype,  
    field2 datatype,  
    field3 datatype  
) character set 字符集 collate 校验规则 engine 存储引擎;
```

说明:

- field 表示列名
- datatype 表示列的类型
- character set 字符集, 如果没有指定字符集, 则以所在数据库的字符集为准
- collate 校验规则, 如果没有指定校验规则, 则以所在数据库的校验规则为准

这些东西, 在创建库的时候已经弄好了  
创建表的时候不建议在这写

```
mysql> create table if not exists `users`(  
-> id int comment '用户的ID',  
-> name varchar(20) comment '用户的姓名',  
-> password varchar(32) comment '密码',  
-> birthday date comment '用户的生日'  
-> )character set utf8 collate utf8_general_ci engine MyI  
SAM;  
Query OK, 0 rows affected (0.01 sec)  
  
mysql>
```

```
mysql> show tables;
```

```
+-----+
| Tables_in_lesson2 |
+-----+
| users              |
+-----+
1 row in set (0.00 sec)
```

```
mysql> desc users;
```

```
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)   | YES  |     | NULL    |       |
| name  | varchar(20) | YES  |     | NULL    |       |
| password | varchar(32) | YES  |     | NULL    |       |
| birthday | date      | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> show create table users;
```

```
+-----+-----+
| Table | Create Table
+-----+-----+
| users | CREATE TABLE `users` (
  `id` int(11) DEFAULT NULL COMMENT '用户的ID',
  `name` varchar(20) DEFAULT NULL COMMENT '用户的姓名',
  `password` varchar(32) DEFAULT NULL COMMENT '密码',
  `birthday` date DEFAULT NULL COMMENT '用户的生日'
) ENGINE=MyISAM DEFAULT CHARSET=utf8
```

```
mysql> show create table users \G;
```

```
***** 1. row *****
      Table: users
Create Table: CREATE TABLE `users` (
  `id` int(11) DEFAULT NULL COMMENT '用户的ID',
  `name` varchar(20) DEFAULT NULL COMMENT '用户的姓名',
  `password` varchar(32) DEFAULT NULL COMMENT '密码',
  `birthday` date DEFAULT NULL COMMENT '用户的生日'
) ENGINE=MyISAM DEFAULT CHARSET=utf8
1 row in set (0.00 sec)
```

带上 \G 好看些