2. 库的操作

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

2.2 创建数据库案例

- 创建名为`db1`的数据库

  ```
  create database db1;
  ```

> 说明：当我们创建数据库没有指定字符集和校验规则时，系统使用默认字符集：utf8，校验规则是：utf8_ general_ ci

- 创建一个使用utf8字符集的 db2 数据库

  ```
  create database db2 charset=utf8;
  ```

- 创建一个使用utf字符集，并带校对规则的`db3`数据库。

  ```
  create database db3 charset=utf8 collate utf8_general_ci;
  ```

2.3 字符集和校验规则

2.3.1 查看系统默认字符集以及校验规则

```
show variables like 'character_set_database';
show variables like 'collation_database';
```

2.3.2 查看数据库支持的字符集

```
show charset;
```

字符集主要是控制用什么语言。比如utf8就可以使用中文。

2.3.3 查看数据库支持的字符集校验规则

```
show collation;
```

2.3.4 校验规则对数据库的影响

- 不区分大小写

创建一个数据库，校验规则使用utf8_ general_ ci[不区分大小写]

```
create database test1 collate utf8_general_ci;
```

```
use test1;
```

```
create table person(name varchar(20));
```

```
insert into person values('a');
insert into person values('A');
insert into person values('b');
insert into person values('B');
```

- 区分大小写

创建一个数据库，校验规则使用utf8_ bin[区分大小写]

```
create database test2 collate utf8_bin;
```

```
use test2
```

```
create table person(name varchar(20));
```

```
insert into person values('a');
insert into person values('A');
insert into person values('b');
insert into person values('B');
```

- 进行查询

不区分大小写的查询以及结果

```
mysql> use test1;
mysql> select * from person where name='a';
+------+
| name |
+------+
| a    |
| A    |
+------+
2 rows in set (0.01 sec)
```

区分大小写的查询以及结果

```
mysql> use test2;
mysql> select * from person where name='a';
+------+
| name |
+------+
| a    |
+------+
2 rows in set (0.01 sec)
```

- 结果排序
  不区分大小写排序以及结果：

```
mysql> use test1;
mysql> select * from person order by name;
+------+
| name |
+------+
| a    |
| A    |
| b    |
| B    |
+------+
```

区分大小写排序以及结果：

```
mysql> use test2;
mysql> select * from person order by name;
+------+
| name |
+------+
| A    |
| B    |
| a    |
| b    |
+------+
```

2.4 操纵数据库

2.4.1 查看数据库

```
show databases;
```

2.4.2 显示创建语句

```
show create database 数据库名;
```

示例：

```
mysql> show create database mytest;
+----------+-----------------------------------------------------------------+
| Database | Create Database                                                 |
+----------+-----------------------------------------------------------------+
| mysql    | CREATE DATABASE `mytest` /*!40100 DEFAULT CHARACTER SET utf8 */ |
+----------+-----------------------------------------------------------------+
```

说明：

- MySQL 建议我们关键字使用大写，但是不是必须的。
- 数据库名字的反引号``,是为了防止使用的数据库名刚好是关键字
- `/*!40100 default.... */`这个不是注释，表示当前mysql版本大于4.01版本，就执行这句话

2.4.2 修改数据库

语法：

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

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

说明：

- 对数据库的修改主要指的是修改数据库的字符集，校验规则

实例： 将`mytest`数据库字符集改成`gbk`

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

mysql> show create database mytest;
+----------+-----------------------------------------------------------------+
| Database | Create Database                                                 |
+----------+-----------------------------------------------------------------+
| mytest   | CREATE DATABASE `mytest` /*!40100 DEFAULT CHARACTER SET gbk */ |
+----------+-----------------------------------------------------------------+
```

2.4.4 数据库删除

```
DROP DATABASE [IF EXISTS] db_ name;
```

执行删除之后的结果:

- 数据库内部看不到对应的数据库
- 对应的数据库文件夹被删除，级联删除，里面的数据表全部被删

> 注意：不要随意删除数据库

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。

2.4.5.2 还原

```
mysql> source D:/mysql-5.7.22/mytest.sql;
```

2.4.5.3 注意事项

- 如果备份的不是整个数据库，而是其中的一张表，怎么做?

```
   # mysqldump -u root -p 数据库名 表名1 表名2  > D:/mytest.sql
```

- 同时备份多个数据库

```
# mysqldump -u root -p -B 数据库名1  数据库名2 ... > 数据库存放路径
```

- 如果备份一个数据库时，没有带上-B参数， 在恢复数据库时，需要先创建空数据库，然后使用数据库，再使用source来还原。

2.4.6 查看连接情况

语法：

```
show processlist
```

示例：

```
mysql> show processlist;
+----+------+-----------+------+---------+------+-------+------------------+
| Id | User | Host      | db   | Command | Time | State | Info             |
+----+------+-----------+------+---------+------+-------+------------------+
|  2 | root | localhost | test | Sleep   | 1386 |       | NULL             |
|  3 | root | localhost | NULL | Query   |    0 | NULL  | show processlist |
+----+------+-----------+------+---------+------+-------+------------------+
```

可以告诉我们当前有哪些用户连接到我们的MySQL，如果查出某个用户不是你正常登陆的，很有可能你的数据库被人入侵了。以后大家发现自己数据库比较慢时，可以用这个指令来查看数据库连接情况。