

命令行操作数据库

笔记本：MySQL基本使用

创建时间：2018/5/7 23:15

更新时间：2018/5/9 21:35

作者：ly

连接数据库

```
mysql -uroot -p
```

回车后输入密码，当前设置的密码为mysql

退出登录

quit 和 exit或

ctrl+d

登录成功后，输入如下命令查看效果

```
查看版本: select version();
```

```
显示当前时间: select now();
```

修改输入提示符

prompt python>

\D 完整日期

\U 使用用户

数据库操作

查看数据库：

```
show databases;
```

使用数据库：

```
use 数据库名;
```

查看当前使用的数据库：

```
select database();
```

创建数据库：

```
create database 数据库 charset=utf8;
```

删除数据库：

```
drop database 数据库名;
```

数据表操作

查看当前数据库中**所有**表：

```
show tables;
```

查看表结构：

```
desc 表名;
```

创建表：

多行书写：

```
create table 数据表名(  
    字段1 数据类型 约束;  
    字段2 数据类型 约束;  
    fid datatype 约束;  
);
```

单行书写：

```
create table 数据表名字(字段1 类型 约束,字段2 类型 约束.....);
```

修改表

添加字段：

```
alter table 表名 add 列名 类型;
```

修改字段：

重命名版

```
alter table 表名 change 原名 新名 类型或约束;
```

不重命名版

```
alter table 表名 modify 列名 类型或约束;
```

删除表：

```
drop table 表名;
```

查看表的创建语句：

```
show create table 表名;
```

删除字段：

```
alter table 表名 drop 列名;
```

数据增删改查

curd的解释: 代表创建 (Create) 、更新 (Update) 、读取 (Retrieve) 和删除 (Delete)

查询基本

查询所有列(所有记录)

```
select * from 表名;
```

查询指定列 可以使用as为列或表指定别名

```
select 列1, 列2, .....from 表名;
```

增加

全列插入：值的顺序与表中的字段的顺序对应

```
insert into 表名 values(...)
```

部分列插入：值的顺序与给出的列顺序对应

```
insert into 表名(列1, ...) values(值1,...)
```

上面的语句一次可以向表中插入一行数据，还可以一次性插入多行数据，这样可以减少与数据库的通信

全列多行插入：值的顺序与给出的列顺序对应

修改

```
update 表名 set 列1=值1,列2=值2...where 条件;
```

删除

```
delete from 表名 where 条件
```

上叙为 物理删除 真正的删除

逻辑删除 为表添加一个字段is_delete

用一个字段来表示 这条信息是否已经不能再使用了

给数据表添加一个is_delete字段 bit 类型

数据备份

```
运行mysqldump命令  
mysqldump -uroot -p 数据库名 > python.sql;
```

按提示输入mysql的密码

数据恢复

```
连接mysql, 创建新的数据库  
退出连接, 执行如下命令  
mysql -uroot -p 新数据库名 < python.sql
```

根据提示输入mysql密码