

MySQL 数据库

使用 MySQL

MySQL 是 Web 世界中使用最广泛的数据库服务器。

MySQL 内部有多种数据库引擎,最常用的引擎是支持数据库事务的 InnoDB。

安装MySQL

- 一定要进行设置密码，确保数据库安全；
- 配置文件为/etc/my.cnf；
- 修改数据库默认的编码全部改为 UTF-8；
- 查看编码格式:show variables like '%char%';
- 安装MySQL-python，封装了 MySQL C 驱动的 Python 驱动。

建立MySQL连接

```
conn = MySQLdb.connect(host='localhost', \
                        user='root',passwd='123456', \
                        db='testpython', \
                        port=3306,charset='utf8')
```

- host : MySQL 数据库地址
- user: 数据库登陆用户名
- passwd: 数据库登陆密码
- db: 登陆数据库后,需要操作的库名
- port: 数据库监听端口,默认为 3306
- charset: 数据库编码

MySQL连接操作

- `commit()` 如果数据库表进行了修改,提交保存当前的数据。
- `rollback()` 如果有权限,就取消当前的操作,否则报错
- `cursor()` 游标指针。

游标操作

创建一个数据库游标

```
In [3]: cur = conn.cursor()
```

cursor() 提供的方法来进行操作,方法主要是 :

1. 执行命令
2. 接收结果

游标操作

cursor 执行命令的方法:

- `execute(query, args)`: 执行单条 sql 语句。 `query` 为 sql 语句本身, `args` 为参数值的列表。执行后返回值为受影响的行数。
- `executemany(query, args)`: 执行单条 sql 语句, 但是重复执行参数列表里的参数, 返回值为受影响的行数

`conn.commit()`

提交事务

插入数据

日常工作中,大多数会先拼接一个 SQL 语句 然后再执行,实现插入数据的操作:

```
sql = "INSERT INTO member (username,password,email)
values ('%s','%s','%s')" % ('walter.liu','123456','zheshiz2@163.com')

cur.execute(sql)

conn.commit()
```


查询数据

- `fetchall(self)`: 接收全部的返回结果行。
- `fetchmany(size=None)`: 接收 `size` 条返回结果行。如果 `size` 的值大于返回的结果行的数量，则会返回 `cursor.arraysize` 条数据。
- `fetchone()`: 返回一条结果行。
- `scroll(value, mode='relative')`: 移动指针到某一行。

如果 `mode='relative'`, 则表示从当前所在行移动 `value` 条；

如果 `mode='absolute'`, 则表示从结果集的第一行移动 `value` 条；

over !