当前使用的数据库，主要分为两类

文档型，如sqlite，就是一个文件，通过对文件的复制完成数据库的复制

服务型，如mysql、postgre，数据存储在一个物理文件中，但是需要使用终端以tcp/ip协议连接，进行数据库的读写操作

E-R模型

当前物理的数据库都是按照E-R模型进行设计的

E表示entry，实体

R表示relationship，关系

一个实体转换为数据库中的一个表

关系描述两个实体之间的对应规则，包括一对一、一对多、多对多

关系转换为数据库表中的一个列 \*在关系型数据库中一行就是一个对象

三范式

经过研究和对使用中问题的总结，对于设计数据库提出了一些规范，这些规范被称为范式

第一范式（1NF)：列不可拆分

第二范式（2NF)：唯一标识

第三范式（3NF)：引用主键

说明：后一个范式，都是在前一个范式的基础上建立的

SQL常用语法

* use 数据库名称; 使用数据库
* SELECT \* FROM 表名; 表中提取所有列

SELECT 列名 FROM表名; 表中提取制定几列

* UPDATE 表名 更新数据库中的数据

SET列名=值，列名=值…

WHERE列名=值; WHERE规定哪些记录需要更新,若省略，全部更新

* DELETE FROM表名 从数据库中删除表

WHERE 列名=值; 如果省略了WHERE，所有的记录都将被删除

* INSERT INTO表名 向数据库中插入新数据

VALUES (value1,value2 ,...); value可指定在哪一列插入数据

INSERT INTO表名 (column1,column2,...)

VALUES (value1,value2,value3,...);

* CREATE DATABASE 创建新数据库
* ALTER DATABASE 修改数据库
* CREATE TABLE 创建新表
* ALTER TABLE 变更（改变）数据库表
* DROP TABLE 删除表
* CREATE INDEX 创建索引（搜索键）
* DROP INDEX 删除索引

比较运算符

* 等于=
* 大于>
* 大于等于>=
* 小于<
* 小于等于<=
* 不等于!=或<>

a="select \* from people\_data where name ='钟汉良' "  
cursor.execute(a)

逻辑运算符

* and
* or
* not

a="select \* from people\_data where name !='钟汉良' and age<30"  
cursor.execute(a)

模糊查询

* like
* %表示任意多个任意字符
* \_表示一个任意字符

a="select \* from people\_data where name like '钟%'"

范围查询

* in表示在一个非连续的范围内
* between ... and ...表示在一个连续的范围内

a="select \* from people\_data where age between 20 and 30"

空判断

* 注意：null与''是不同的
* 判空is null

a="select \* from people\_data where age is null "

优先级

* 小括号，not，比较运算符，逻辑运算符
* and比or先运算，如果同时出现并希望先算or，需要结合()使用