

Big data lesson 6 SQL进阶

Python DB API 规范 支持 MySQL, Oracle, SQL Server
对DBMS steps: ①引入API模块 ②与数据库建立连接
③执行SQL语句 ④关闭数据库连接

mysql-connector => MySQL 的官方驱动器, 用于后端语言 ex: python 连接

- 1) 通过指定 host, user, passwd 和 port 等参数来创建数据库连接, 分别对应数据库IP地址、用户名、密码和端口号
- 2) db.close() 关闭数据库连接;
- 3) db.cursor() 创建游标, 操作 database 的数据
- 4) db.begin() 开启事务;
- 5) db.commit() 和 db.rollback() 对事务进行提交及回滚

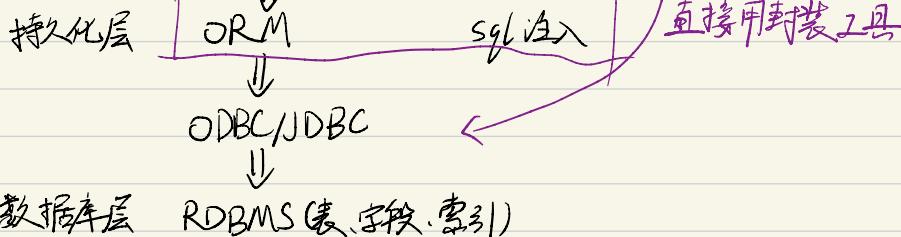
- (1) cursor.execute(query_sql), 执行数据库查询
- (2) cursor.fetchone(), 读取 dataset 中的一条数据
- (3) cursor.fetchall(), 取出 dataset 中的所有行, 返回一个元组 tuples 类型
- (4) cursor.fetchmany(n), 取出数据集中 n 多条数据, 同样返回一个元组 tuples
- (5) cursor.rowcount, 返回查询结果集中的行数, 未设置到 data or 还没查询, 结果为 -1
- (6) cursor.close(), 关闭游标

auth_plugin → 明文 输入的字段显示出来

增删改查中出现异常: try ... except 捕获异常 SQL 注入安全性不能保障

所以我们用封装好的工具

业务逻辑层：业务对象(对象、属性、继承)



sqlalchemy \Rightarrow python 中最有名的 ORM 工具

原理：将数据表转换为 Python 类，其中数据列为属性

优点：简洁易读，将 data table 抽象为对象

可移植：封装多种数据库引擎，面对多个数据库，操作基本一致，代码易维护

更安全：有效避免 SQL 注入

sqlalchemy 使用：from sqlalchemy import Column, String, Integer, Float

ORM

class(类) $\leftarrow \dots \rightarrow$ 数据表

实例对象 $\leftarrow \dots \rightarrow$ 数据行

属性 $\leftarrow \dots \rightarrow$ 字段

建议用 decimal 而不是 float 类型

MVC

Excel中使用SQL:

- ① 通过OLE DB 接口获取外部数据源
- ② 通过Microsoft Query 导入外部数据源

Excel 可与 MySQL交互, need use MySQL for Excel 插件

WebSQL:

WebSQL是一种操作本地数据库的网页API接口, 通过它就能操作客户端的本地存储

Cookies是最早的本地存储, 但容量有限, 只有4KB, 只要超过这个限制就会忽略, 无法保存

Web Storage } local Storage 本地存储, 除非主动删数据否则会一直存储在本地
Session Storage 只存在Session会话中, 当会话结束, 会自动释放
→与Cookie类似, 但有更大容量的存储

WebSQL & IndexedDB 是HTML5本地缓存的新技术

比 web storage 存储能力更强, 支持的数据 type更多如图片, 视频

SQLite 在 Chrome & Safari 上用来实现本地化
Wechat

WebSQL方便我们对DBMS进行操作

IndexedDB: IndexedDB 是一种NoSQL方式, 存储的是 key-value类型的数据
支持事务, 可通过事务进行增删改查(CRUD)

WebSQL 使用

`openDatabase()` ⇒ 打开一个已经存在的数据库, 如果不存在则会创建

`var db = window.openDatabase(dbname, version, dbdesc, dbsize, function() {});`

↓ ↓ ↓ ↓ ↓
数据库名 版本号 描述 大小 创建回调

事务操作 transaction(callback, errorCallback, successCallback);

Callback: 处理事务的回调函数(必选), 在回调函数中可以执行SQL语句
会使用 executeSQL方法。

error callback: 执行失败时的回调函数 (可选)

successCallback: 执行成功时的回调函数

Web SQL = Chrome's SQLite

SQL执行: executeSql()

SQLite 用的是 C语言编写的, 提升代码底层的执行效率

C++ 面对对象 Object C是底层语言 所以C快

Pro: 1) 轻量级, 存储高效, 查询操作方便

2) 不需安装和配置, 有很好的迁移性, 能减少 server压力

Con: 1) 用于中小型 data store, 不适用于高并发

微信 server 不能用 SQLite 的原因? SQLite 同一时间只允许一个写操作, 吞吐量非常有限。

SQLite 使用: 没有用户管理功能; 在语法上也有自己的“方言”

比如在 SQL 中的 SELECT 语句我们想拼接两个列

MySQL 拼接两列用 Concat, 而在 SQLite, PostgreSQL, Oracle 和 DB2 中使用 || 号
在 SQLite 中不支持 RIGHT JOIN

SQLite 是世界上使用最多的数据库

Python 中使用 SQLite3: import sqlite3

cur 会指向查询结果集的第一个位置:

cur.fetchone() 方法, 获取一条记录;

cur.fetchall() 获取全部记录

cur.fetchmany(n) 方法, 获取 n 条记录

result = cur.fetchall()

对事务操作完了，可提交事务，使用 Conn.commit()

关闭游标 cur.close() 关闭 database connect: conn.close()

通过 SQLite 查看 wechat 聊天记录

wechat => MM.sqlite sahre_master 是 SQLite 的系统表，数据表是只读的，里面保存了数据库中数据表的名称

聊天记录的数据表都以 chat_ 开头的

因为 _ 属于特殊字符，在 like 语句中会将 _ 作为通配符

如果想要对开头为 chat_ 的文件名进行匹配，need 用 escape 对这个特殊字符做转义

WITH a AS b AS