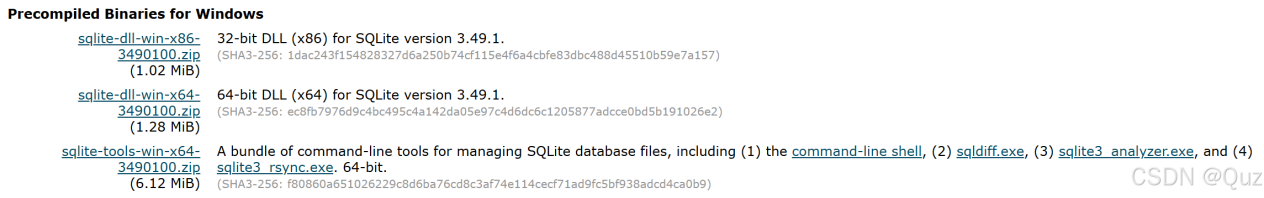
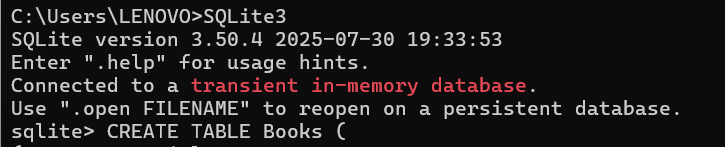
1. **安装配置**
2. **系统：windows11 环境：SQLite**
3. 访问 SQLite 官方下载地址：https://www.sqlite.org/download.html，找到 Windows 区域，下载库文件：sqlite-dll-win-x86-3490100.zip 或者 sqlite-dll-win-x64-3490100.zip,下载操作工具：sqlite-tools-win-x64-3490100.zip。



1. 创建一个文件夹，例如E:\sqlite，将上述两个压缩文件解压到该文件夹中。解压后会得到 sqlite3.def、sqlite3.dll 和 sqlite3.exe 等文件。
2. 将 E:\sqlite 文件夹的路径添加到系统的 PATH 环境变量中。这样可以在命令提示符下直接使用 sqlite3 命令。



**二、实验内容**

1. **按照下面的结构与内容建表。**

表名分别以 T、S 开头，后面是建表人的学号(以下简记为 T\*\*、S\*\* )。要求：用 create table 命令建立表 T\*\*，表中数据使用中英文皆可。

表 T\*\*:

| **Title** | **Author** | **BookID** | **Price** | **Publisher** |
| --- | --- | --- | --- | --- |
| 计算机原理 | 张一平 | S3092 | 20.80 | 暨南大学 |
| C 语言程序设计 | 李华 | H1298 | 15.30 | 电子工业 |
| 数据库原理 | 王家树 | D1007 | 22.70 | 高等教育 |
| 计算机网络 | 高明 | S5690 | 18.90 | 高等教育 |
| Artificial intelligence | P. Winston | D2008 | 20.80 | 电子工业 |
| Expert systems | R. Ullman | H3067 | 17.00 | 清华大学 |
| 软件工程 | 鲁延琦 | S2005 | 35.00 | 暨南大学 |
| Fortran 程序设计 | 顾学峰 | S5006 | 18.00 | 高等教育 |
| Algorithm Design | Cormen | A1001 | 45.00 | MIT |
| Advanced Database Systems | Ozsu | A1002 | 52.00 | Springer |
| Applied Cryptography | Schneier | A1003 | 38.50 | Wiley |

代码（以Sqlite为例）

-- 创建表格 'T2023103896'

CREATE TABLE T2023103896 (

Title TEXT,

Author TEXT,

BookID TEXT PRIMARY KEY,

Price REAL,

Publisher TEXT

);

-- 插入数据

INSERT INTO T2023103896 (Title, Author, BookID, Price, Publisher) VALUES

('计算机原理', '张一平', 'S3092', 20.80, '暨南大学'),

('C 语言程序设计', '李华', 'H1298', 15.30, '电子工业'),

('数据库原理', '王家树', 'D1007', 22.70, '高等教育'),

('计算机网络', '高明', 'S5690', 18.90, '高等教育'),

('Artificial Intelligence', 'P. Winston', 'D2008', 20.80, '电子工业'),

('Expert Systems', 'R. Ullman', 'H3067', 17.00, '清华大学'),

('软件工程', '鲁延琦', 'S2005', 35.00, '暨南大学'),

('Fortran 程序设计', '顾学峰', 'S5006', 18.00, '高等教育'),

('Algorithm Design', 'Cormen', 'A1001', 45.00, 'MIT'),

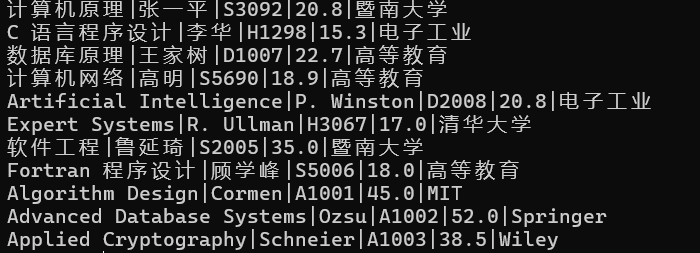
('Advanced Database Systems', 'Ozsu', 'A1002', 52.00, 'Springer'),

('Applied Cryptography', 'Schneier', 'A1003', 38.50, 'Wiley');

1. **无条件查询**

查找表T\*\*的全部信息。

SELECT \* FROM T2023103896;



1. **简单条件查询**

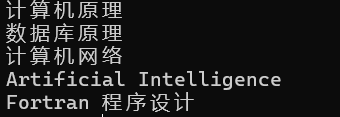
查找T\*\*表中全部书号及书名。

SELECT BookID, Title FROM T2023103896;



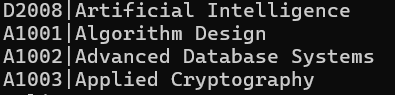
查找T\*\*表中价格在18～25元之间的书的书名。

SELECT Title FROM T2023103896 WHERE Price BETWEEN 18 AND 25;



查找T\*\*表中书名以A开头的书号和书名。

SELECT BookID, Title FROM T2023103896 WHERE Title LIKE 'A%';



1. **多条件查询**

查找书名起始字符为A，价格小于40元的书名及价格。

SELECT Title, Price FROM T2023103896 WHERE Title LIKE 'A%' AND Price < 40;



查找书名起始字符不为A，价格大于30元的书号、书名及价格。

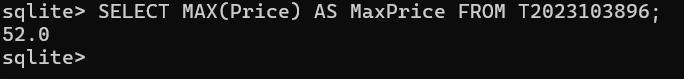
SELECT BookID, Title, Price FROM T2023103896 WHERE Title NOT LIKE 'A%' AND Price > 30;



1. **使用函数进行查找**

查出价格最高的书价。

SELECT MAX(Price) AS MaxPrice FROM T2023103896;



计算这些书籍的最高书价、最低书价及平均书价。

SELECT

...> MAX(Price) AS MaxPrice,

...> MIN(Price) AS MinPrice,

...> AVG(Price) AS AvgPrice

...> FROM T2023103896;

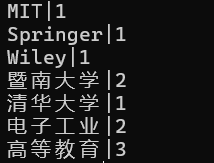


按出版社进行分组，列出各个出版社书籍的总数。

SELECT Publisher, COUNT(\*) AS BookCount

...> FROM T2023103896

...> GROUP BY Publisher;



计算不同的书名有多少种。

SELECT COUNT(DISTINCT Title) AS UniqueTitleCount FROM

T2023103896;



1. **带排序的查询**

查书名和价格，按价格从大到小的顺序排列。

SELECT Title, Price FROM T2023103896

...> ORDER BY Price DESC;

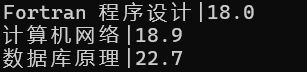


查找出版社为 “高等教育” 的所有书籍的书名及价格，并按价格从低到高排序。

SELECT Title, Price FROM T2023103896

...> WHERE Publisher = '高等教育'

...> ORDER BY Price ASC;



1. **连接查询**

查找价格相同的书籍对，返回每对书的书名和价格（同一书不能与自身匹配）。

SELECT

...> a.Title AS Book1,

...> b.Title AS Book2,

...> a.Price

...> FROM T2023103896 a

...> JOIN T2023103896 b ON a.Price = b.Price

...> WHERE a.BookID < b.BookID; --防止与自身或重复配对



三、个人总结

在简单查询方面，SELECT \* FROM 表名能快速获取全表信息，而指定列查询如SELECT BookID, Title FROM Books，可按需提取关键数据，避免冗余。

条件查询里，WHERE子句的BETWEEN、LIKE等操作符可用于查找像价格区间和特定开头的书名，可精准筛选数据。

多条件查询进一步深化了对逻辑组合的运用，AND、NOT的结合能更细致地定位数据。

函数查询：MAX、MIN、AVG等函数能高效进行统计分析，GROUP BY分组也让数据按类别归纳变得清晰，比如按出版社统计书籍数量。

排序查询的ORDER BY让结果呈现更有条理，升序降序的设置满足不同展示需求。

连接查询尤其是自连接，解决了像查找价格相同书籍对这类复杂关联问题，虽稍显抽象，但能应对更复杂的数据关联场景。