

SQLite

SQLite 是一个非常好的“完成工作”的工具，特别是当您达到 Microsoft Excel（或其他电子表格软件）的极限时。

据其作者称，SQLite 以某种形式在每一个 Windows 10、Mac、Android 和 iOS 设备中使用，并且内置于 Firefox、Chrome（包括 Edge）和 Safari 浏览器中：浏览器将其浏览历史记录和 cookie 存储在 SQLite 中-format 数据库，如果您愿意，您可以使用命令行工具手动编辑它们。

目前在版本 3.34.1 中，命令行程序被调用 `sqlite3` 并存在于所有主要操作系统中，您应该能够将其作为单个可执行文件下载，放入您喜欢的任何文件夹中并运行它。在 Alpine 上，您可以 `sqlite` 使用 `apk`。

SQLite 是一个嵌入式数据库，这意味着没有单独的服务器进程。数据库只是文件。当您运行命令行工具时，该工具同时履行客户端和服务角色；当您从程序中使用 SQLite 时，SQLite 驱动程序也充当“服务器”。

人口普查练习

您将需要数据库活动 1 中的人口普查文件：使用该 `setup.sql` 文件和人口普查 csv 文件创建一个文件夹。

下载或安装 `sqlite` 并运行 `sqlite3 census.db` 以创建数据库文件。

为了创建表，该命令 `.read setup.sql` 读取并执行文件中的命令（类似于 `source shell`）。SQLite 系统命令以点开头，末尾不带分号 - 请参阅 `.help` 参考资料 中的列表。（不要运行该 `import.sql` 文件，这是 MariaDB 特定的。）

用于 `.tables` 显示表并检查它们是否已创建。例如，`.schema Ward` 您可以显示单个表的创建语句。

源数据位于简单的 CSV 文件中，例如，如果（返回 shell）您执行了 `a, head -n 5 County.csv` 您将看到以下内容：

```
E090000001,"City of London",E120000007,E920000001
E090000002,"Barking and Dagenham",E120000007,E920000001
E090000003,Barnet,E120000007,E920000001
E090000004,Bexley,E120000007,E920000001
E090000005,Brent,E120000007,E920000001
```

与安装脚本中的以下内容匹配：

```
CREATE TABLE County (  
    code VARCHAR(10) PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    parent VARCHAR(10),  
    country VARCHAR(10),  
  
    FOREIGN KEY (parent) REFERENCES Region(code),  
    FOREIGN KEY (country) REFERENCES Country(code)  
);
```

要导入数据，请在 SQLite 提示符下运行以下命令：

```
.mode csv  
.header off  
.import Country.csv Country  
.import Region.csv Region  
.import County.csv County  
.import Ward.csv Ward  
.import Occupation.csv Occupation  
.import Statistic.csv Statistic
```

您现在应该已导入所有数据，并且可以在此数据库中尝试之前练习中的任何 SQL 查询。SQLite 的 SQL 方言与 MariaDB 并不完全相同：例如，SQLite 可以执行 INNER 和 LEFT OUTER JOIN（以及 CROSS JOIN），但不能执行 RIGHT OUTER 或 FULL OUTER JOIN。

SQLite 在数据类型方面也非常宽容：它或多或少将类型视为注释，因此如果您尝试的话，它可以让您将字符串存储在整数列中。您还可以编写 CREATE TABLE 语句并省略类型，SQLite 不会介意。默认情况下，SQLite *也不强制执行外键约束*，但如果您愿意，可以[将其打开](#)。这些都是 SQLite 开发人员做出的设计选择，对于任何特定用例来说，它们可能是也可能不是最好的。

模式和文件输出

SQLite 有不同的方式来显示数据表。尝试 `.mode csv`，`.header on` 然后 `SELECT * FROM Region`；以带有标题的 CSV 文件形式查看区域（模式和标题设置一直适用，直到您更改它们或退出 SQLite）。如果您想实际生成 CSV 文件，`.once FILENAME` 请告诉 SQLite 仅将以下查询的结果发送到文件，这样您就可以处理查询，直到它执行您想要的操作，然后运行命令 `.once` 并重新运行查询（SQLite 在大多数平台上支持 readline，因此 UP 键可以取回最后一行）并将其发送到文件。

默认模式，`list` 显示由条形字符分隔的列。您还可以尝试 `.mode column` 哪种方式更易于阅读，在排列的列中打印，列之间有空格。或者，如果您想在 HTML 页面中包含表格，`.mode html` 请打印带有 HTML 标记的表格。如果您使用的是主机操作系统（不是 alpine VM）并且安装了 Excel，`.excel` 则在 Excel 中打开以下查询的结果。

您现在已经学会了一种极其强大的方法来分析以一个或多个 CSV 文件或任何可以轻松导出到 CSV 的数据格式（例如大多数 Excel 文件）中提供的数据。

来自 Java 的 SQLite

您可以使用 JDBC 驱动程序从 Java 程序访问 SQLite 数据库。在 Maven 中，声明以下依赖项：

```
<dependency>
  <groupId>org.xerial</groupId>
  <artifactId>sqlite-jdbc</artifactId>
  <version>3.34.0</version>
</dependency>
```

连接字符串采用格式 `jdbc:sqlite:FILEPATH`。文件的绝对路径应该始终有效（您必须将其拼写出来，但不能使用 `~` 快捷方式），或者如果您从与数据库文件相同的文件夹运行 JAR 文件，则只需给出不带路径的文件名即可也工作。

练习：编写一个最小的 Maven/Java 应用程序，用于打印刚刚创建的 sqlite 数据库中的区域名称。

来自 Python 的 SQLite

Python 是一种很棒的数据科学语言，如果您需要从 Python 连接到 SQLite 数据库，那么方法如下：

```
# you may have to 'pip install sqlite3'
import sqlite3
with sqlite3.connect('FILENAME') as connection:
    cursor = connection.cursor()
    for row in cursor.execute('SELECT * FROM Ward WHERE name LIKE ?',
WARDNAME):
        # do something with row
        print(row)
```

- 该 `with` 语句相当于 Java 的 `try-with-resources`。
- 该 `execute` 语句执行准备好的语句。您将语句本身放在第一个参数中，然后将替换占位符的值作为附加参数。
- 该 `execute` 命令返回一个可以循环的迭代器，行本身是 python 元组（您也可以将 `.fetchall()` 它们放入列表中）。

有关更多信息，请参阅[Python sqlite 文档](#)。

这里有一个重要的警告：如果您将任何数据写入数据库，则必须 `connection.commit()` 随后调用，否则数据实际上不会写入磁盘。

来自 C 的 SQLite

这里的最后一个练习是按照幻灯片中的说明编写一个 C 程序，该程序读取 SQLite `census.db`，迭代区域表并打印“Region NAME is code CODE”，其中 NAME/CODE 替换为区域的名称和代码。

当然，您需要处理 `sqlite3_` 函数中的潜在错误：只有它们返回时才可以安全地继续 `SQLITE_OK`。

您将需要安装 `sqlite-dev` 包含头文件的包，并在命令 `sqlite3.h` 中链接到库。 `-lsqlite3 gcc`

如果与回调函数一起使用 `sqlite3_exec`，则回调必须在成功时返回 0，如果遇到错误则必须返回非零，这反过来也会导致 `sqlite_exec` 返回错误。

对于其他方法，请参阅[SQLite C API](#)。例如，`sqlite3_step` 您可以在不使用回调函数的情况下迭代 SQL 结果的行。