一口Linux

无线传感器网项目实战

公众号:一口Linux

彭老师个人微信号

01

Sqlite3基础

参考文章

- 《嵌入式数据库sqlite3【基础篇】-基本命令操作》
- 《嵌入式数据库sqlite3【进阶篇】-子句和函数的使用》
- 《如何用C语言操作sqlite3，一文搞懂》

常用数据库

- 数据在实际工作中应用非常广泛，数据库的产品也比较多，oracle、DB2、SQL2000、mySQL

- 基于嵌入式linux的数据库主要有SQLite, Firebird, Berkeley DB, eXtremeDB。

SQLite

- 作者D.RichardHipp
- 2000年1月，Hipp开始和一个同事讨论关于创建一个简单的嵌入式SQL数据库的想法，这个数据库将使用GNU DBM哈希库（gdbm）做后台，同时这个数据库将不需要安装和管理支持。
- 后来，一有空闲时间，Hipp就开始实施这项工作，2000年8月，SQLite 1.0版发布了。

SQLite特性

- 1. 零配置—无需安装和管理配置；
- 2. 储存在单一磁盘文件中的一个完整的数据库；
- 3. 数据库文件可以在不同字节顺序的机器间自由共享；
- 4. 支持数据库大小至2TB；
- 5. 足够小，全部源码大致3万行c代码，250KB；
- 6. 比目前流行的大多数数据库对数据的操作要快。

安装

- 源码下载地址
  - https://www.sqlite.org/index.html
- ubuntu安装

```
sudo apt-get install sqlite sqlite3    安装应用程序
sudo apt-get install libsqlite3-dev    安装库+头文件，用代码操作数据库必须安装
```

- 也可以使用下面命令，安装图形化操作工具：

```
sudo apt-get install sqlitebrowser     图形化工具建立数据库
```

查看版本号

```
peng@ubuntu:~/work/sqlite$ sqlite3
SQLite version 3.11.0 2016-02-15 17:29:24
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .version
SQLite 3.11.0 2016-02-15 17:29:24 3d862f207e3adc00f78066799ac5a8c282430a5f
sqlite>
```

02

Sqlite3数据类型和约束

表

sno	sname	ssex	sage	sdept
95001	yikou	m	21	cs
95002	peng	m	21	cs

需要考虑哪些因素：

1. 某列用于唯一区分条目
2. 不能为空
3. 数据类型：整形、字符串、布尔
4. 有的需要有默认值
5. 值必须在一定范围(大小、长度)
6. 对表格的操作：增删改查

主要数据类型

数据类型	定义
数据类型 NULL	表示该值为NULL值。
INTEGER	无符号整型值。
REAL	浮点值。
TEXT	文本字符串，存储使用的编码方式为UTF-8、UTF-16BE、UTF-16LE。
BLOB	存储Blob数据，该类型数据和输入数据完全相同，1表示true，0表示false。

其它数据类型

数据类型	定义
smallint	16位的整数。
interger	32位的整数。
decimal(p,s)	精确值p是指全部有几个十进制数,s是指小数点后可以有几位小数。如果没有特别指定，则系统会默认为p=5 s=0 。
float	32位元的实数。
double	64位元的实数。
char(n)	n 长度的字串，n不能超过 254。
varchar(n)	长度不固定且其最大长度为 n 的字串，n不能超过 4000。
graphic(n)	和 char(n) 一样，不过其单位是两个字节， n不能超过127。这个形态是为了支持两个字节长度的字体，如中文字。
vargraphic(n)	可变长度且其最大长度为n的双字元字串，n不能超过2000
date	包含了 年份、月份、日期。
time	包含了 小时、分钟、秒。
timestamp	包含了 年、月、日、时、分、秒、千分之一秒。

约束

- 表的每一列都有一些限制属性，比如有的列的数据不能重复，有的则限制数据范围等，约束就是用来进一步描述每一列数据属性的。

名称	定义
NOT NULL	- 非空
UNIQUE	唯一
PRIMARY KEY	主键
FOREIGN KEY	外键
CHECK	条件检查
DEFAULT	默认

非空 NOT NULL

- 有一些字段我们可能一时不知到该填些什么，同时它也没设定默认值， 当添加数据时，我们把这样的字段空着不填，系统认为他是 NULL 值。
- 但是还有另外一类字段，必须被填上数据，如果不填，系统就会报错。 这样的字段被称为 NOT NULL 非空字段，
- 需要在定义表的时候事先声明。

唯一 UNIQUE

- 除了主列以为，还有一些列也不能有重复值。

主键 PRIMARY KEY

- 一般是整数或者字符串，只要保证唯一就行。
- 在 SQLite 中，主键如果是整数类型，该列的值可以自动增长。

默认值 DEFAULT

- 有一些特别的字段列，在每一条记录中，他的值基本上都是一样的。只是在个别情况下才改为别的值，这样的字段列我们可以给他设一个默认值。

条件检查 CHECK

- 某些值必须符合一定的条件才允许存入，这是就需要用到这个 CHECK 约束。

外键 FOREIGN KEY

- 我们的数据库中已经有 Teachers 表了，假如我们再建立一个 Students 表， 要求 Students 表中的每一个学生都对应一个 Teachers 表中的教师。
  很简单，只需要在 Students 表中建立一个 TeacherId 字段，保存对应教师的 Id 号， 这样，学生和教师之间就建立了关系。
  问题是：我们有可能给学生存入一个不在 Teachers 表中的 TeacherId 值， 而且发现不了这个错误。
  这种情况下，可以把 Students 表中 TeacherId 字段声明为一个外键， 让它的值对应到 Teachers 表中的 Id 字段上。
  这样，一旦在 Students 表中存入一个不存在的教师 Id ， 系统就会报错。

03

表操作：增删改查

表

假设我们要创建一个教学管理的数据库jxgl.db，数据库中要保存学生表stu。

sno	sname	ssex	sage	sdept
95001	yikou	m	21	cs
95002	peng	m	21	cs

sno 学号：
整型值，每个人学号是唯一的，学校一般用学号来区分所有的学生，而且一般学号是递增的，所以我们设置sno为primary key；
sname 姓名：
一般是字符串，可以重复，但是不能为空；
ssex 性别：
字符串，可以为空；
sage 年龄：
整型值，假定年龄要大于14；
sdept 专业：
字符串，可以为空,此处我们默认为’CS’。

常用命令

命令	功能
.help	可显示shell模式中可使用的所有命令列表
.database	显示数据库信息；包含当前数据库的位置
.mode column	使得SQL语句处理的结果以列对齐的方式显示
.mode list	column
.headers on/off	打开关闭列标题显示开关，以使得查询结果在屏幕显示时具有列标题
.tables	列出当前打开的数据库中共有多少张表
.exit	退出SQLite环境
.schema foods	显示表foods 创建时的SQL语句
.schema	显示所有表被创建时的语句
.nullvalue STRING	查询时用指定的串代替输出的NULL串 默认为.nullvalue “
.show	显示shell模式中定义的与输出相关的一些设置
.output file.csv	设置输出文件格式为CSV，文件名为file.csv
.separator	设置select语句输出的列数据间以“，”分隔
.output stdout	恢复输出内容到标准输出设备(屏幕)

创建教学管理“jxgl”数据库

- sqlite3 jxgl.db
- .quit

```
peng@ubuntu:~/work/sqlite$ sqlite3 jxgl.db
SQLite version 3.11.0 2016-02-15 17:29:24
Enter ".help" for usage hints.
sqlite> .quit
peng@ubuntu:~/work/sqlite$
```

创建表：

- CREATE TABLE IF NOT EXISTS *stu* (Sno integer primary key, Sname text not null, Ssex text,Sage integer check(Sage>14),Sdept text default 'CS');

```
peng@ubuntu:~/work/sqlite$ sqlite3 jxgl.db
SQLite version 3.11.0 2016-02-15 17:29:24
Enter ".help" for usage hints.
sqlite> CREATE TABLE IF NOT EXISTS stu(Sno integer primary key,   Sname text n
ot null,   Ssex text,Sage integer check(Sage>14),Sdept text default 'CS');
sqlite> .table
stu
```

- 查看表：
  - .table

注意事项

- 1. 命令前面要加. ，操作语句不加
- 2. 操作语句后面一定要分号；结尾如果漏掉了，一定要补上分号；
- 3. 操作语句对字母的全角半角很敏感，所有的符号都要用半角。
- 4. 语句不区分大小写

```
sqlite> CREATE TABLE IF NOT EXISTS stu(Sno integer primary key,   Sname text n
ot null,   Ssex text,Sage integer check(Sage>14),Sdept text default 'CS');
sqlite> .table
stu
sqlite> .schema
CREATE TABLE stu(Sno integer primary key,   Sname text not null,   Ssex text,S
age integer check(Sage>14),Sdept text default 'CS');
```

插入数据

- 插入数据采用insert into语句：
  - INSERT INTO *STU* VALUES('95001','李勇','M',20,'CS');
  - INSERT INTO STU VALUES('95002','刘晨','F',19,'IS');
  - INSERT INTO STU VALUES('95003','王敏','F',18,'MA');
  - INSERT INTO STU VALUES('95004','张立','M',18,'IS');

```
sqlite> INSERT INTO STU  VALUES('95001','李勇','M',20,'CS');
sqlite> INSERT INTO STU VALUES('95002','刘晨','F',19,'IS');
sqlite> INSERT INTO STU VALUES('95003','王敏','F',18,'MA');
sqlite> INSERT INTO STU VALUES('95004','张立','M',18,'IS');
```

```
sqlite> .dump
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE stu(Sno integer primary key,   Sname text not null,   Ssex text,S
age integer check(Sage>14),Sdept text default 'CS');
INSERT INTO "stu" VALUES(95001,'李勇','M',20,'CS');
INSERT INTO "stu" VALUES(95002,'刘晨','F',19,'IS');
INSERT INTO "stu" VALUES(95003,'王敏','F',18,'MA');
INSERT INTO "stu" VALUES(95004,'张立','M',18,'IS');
```

可以用.dump查看所有记录对应的语句

查看表

- 用SELECT语句查看表中的内容，显示标题和列对齐用下面语句：
  - sqlite> .headers on　　显示列名
  - sqlite> .mode column　　列对齐

```
sqlite> .headers on
sqlite> .mode column
sqlite> select * from stu;
Sno         Sname       Ssex        Sage        Sdept
----------  ----------  ----------  ----------  ----------
95001       李勇          M           20          CS
95002       刘晨          F           19          IS
95003       王敏          F           18          MA
95004       张立          M           18          IS
```

删除一行信息

- 删除条目用delete语句，通常会配合where子句一起使用
  - delete from stu where sname = '王敏';

```
sqlite> delete from stu where sname = '王敏';
sqlite>
sqlite> .dump
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE stu(Sno integer primary key,   Sname text not null,   Ssex text,S
age integer check(Sage>14),Sdept text default 'CS');
INSERT INTO "stu" VALUES(95001,'李勇','M',20,'CS');
INSERT INTO "stu" VALUES(95002,'刘晨','F',19,'IS');
INSERT INTO "stu" VALUES(95004,'张立','M',18,'IS');
```

修改一条记录的某个内容

- 修改记录用update set命令
  - UPDATE stu SET sage=29 WHERE sname='张立';

```
sqlite> UPDATE stu SET sage=29 WHERE sname='张立';
sqlite>
sqlite> .dump
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE stu(Sno integer primary key,   Sname text not null,   Ssex text,S
age integer check(Sage>14),Sdept text default 'CS');
INSERT INTO "stu" VALUES(95001,'李勇','M',20,'CS');
INSERT INTO "stu" VALUES(95002,'刘晨','F',19,'IS');
INSERT INTO "stu" VALUES(95004,'张立','M',29,'IS');
COMMIT;
```

04

Sqlite C编程接口

sqlite3_open

```
int   sqlite3_open(char  *path,   sqlite3 **db);
功能:
    打开sqlite数据库
参数:
    path:  数据库文件路径
    db:  指向sqlite句柄的指针，后面对数据库所有的操作都要依赖这个句柄
返回值:
    成功返回0，失败返回错误码(非零值)
```

sqlite3_close sqlite3_errmsg

```
int   sqlite3_close(sqlite3 *db);
功能:
    关闭sqlite数据库
返回值:
    成功返回0，失败返回错误码
```

```
const  char  *sqlite3_errmsg(sqlite3 *db);
功能:
    打印错误信息
返回值:
    返回错误信息
```

sqlite3_exec

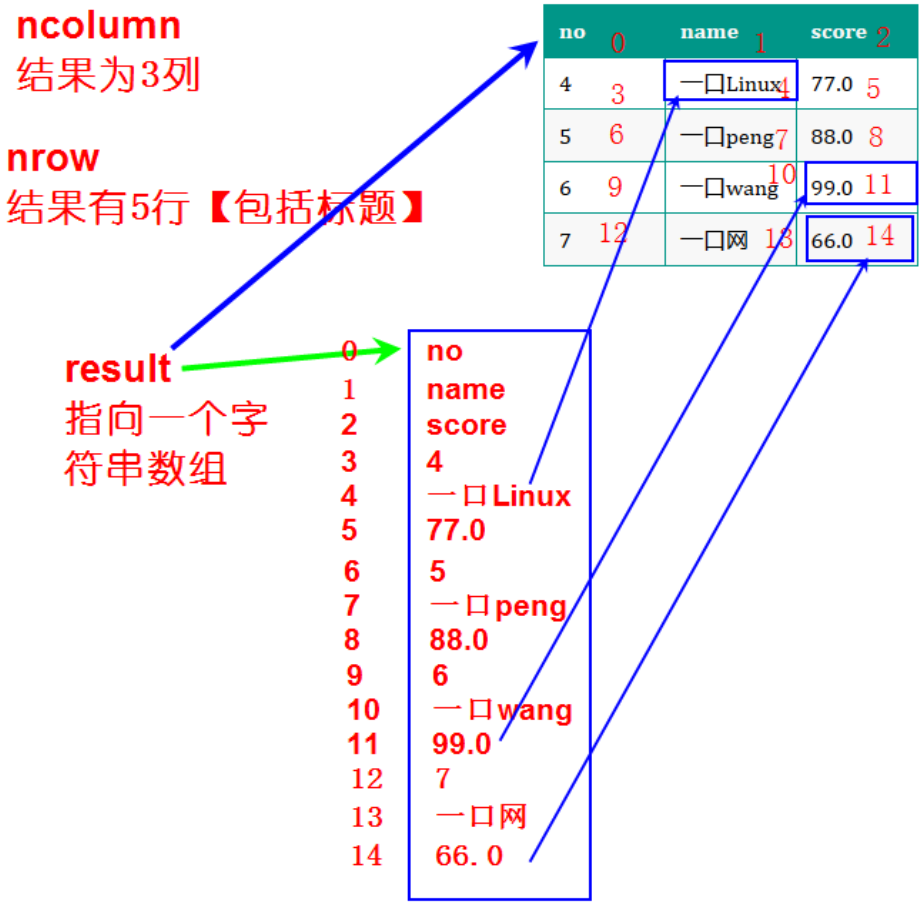
```
typedef  int (*sqlite3_callback)(void *, int, char **, char **);

int   sqlite3_exec(sqlite3 *db, const  char  *sql,  sqlite3_callback callback, void *,  char **errmsg);
功能:
    执行SQL操作
参数:
    db: 数据库句柄
    sql: SQL语句，就是我们前面两章用于操作表的增删改查语句
    callback: 回调函数
    errmsg: 错误信息指针的地址
返回值:
    成功返回0，失败返回错误码
```

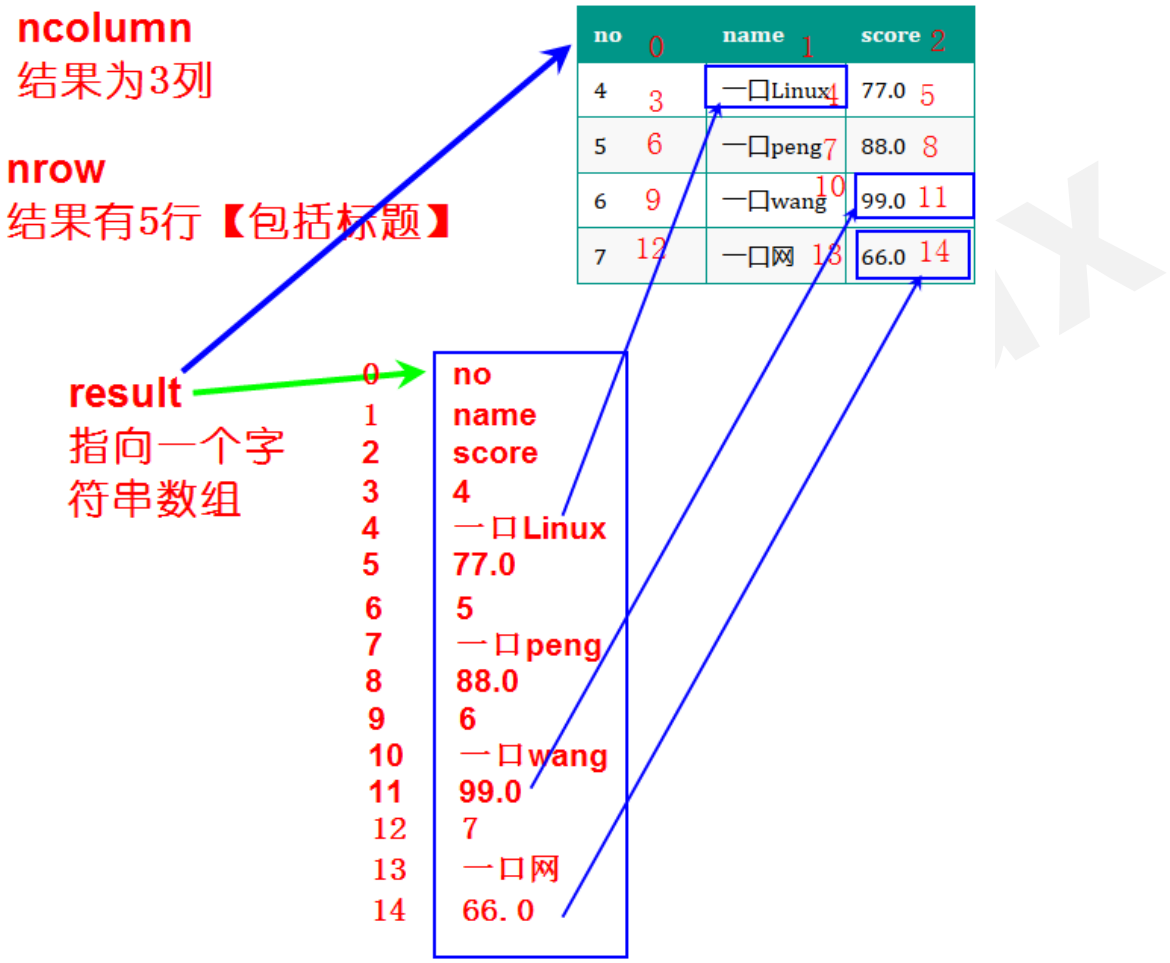
sqlite3_get_table

Select * from user;

```
int   sqlite3_get_table(sqlite3 *db, const  char  *sql,
             char ***resultp,  int*nrow,  int *ncolumn, char **errmsg);
功能:
    执行SQL操作
参数:
    db：数据库句柄
    sql：SQL语句
    resultp：用来指向sql执行结果的指针
    nrow：满足条件的记录的数目
    ncolumn：每条记录包含的字段数目
    errmsg：错误信息指针的地址
返回值:
    成功返回0，失败返回错误码
```



sqlite3_get_table参数含义



实例：

- 本项目中数据库为user.db,有数据表user。

name	password
text	text

相关操作语句

- 创建表
  - CREATE TABLE IF NOT EXISTS user(name text NOT NULL,password text NOT NULL);
- 插入
  - insert into user values(‘root', ‘123456');
- 查找
  - select * from user where name= 'root' and password= '123456';
- 修改
  - Update user where name= 'root ‘;

编译

- gcc user.c -o run -lsqlite3
- sudo chmod 777 user.db

user.db

user.c

更多嵌入式Linux知识
请关注一口君的公众号：一口Linux