一、约束 Constraints  
在上一篇随笔的结尾，我提到了约束， 但是在那里我把它翻译成了限定符，不太准确，这里先更正一下，应该翻译成约束更贴切一点。 那么什么是约束呢？

我们在数据库中存储数据的时候，有一些数据有明显的约束条件。 比如一所学校关于教师的数据表，其中的字段列可能有如下约束：

年龄 - 至少大于20岁。如果你想录入一个小于20岁的教师，系统会报错   
国籍 - 默认中国。所谓默认，就是如果你不填写，系统自动填上默认值   
姓名 - 不能为空。每个人都有名字嘛   
员工号 - 唯一。这个可不能乱，工资发错了就麻烦了   
上面提到的大于、默认、不能为空、唯一等等，就是数据的约束条件。 我们在用 CREATE TABLE 创建表的时候，就应该将每个字段列的约束条件事先说明（如果有的话）， 以后再往表里输入数据的时候，系统会自动为我们检查是否满足约束条件，如果不满足系统会报错。 

SQLite 常用约束如下

NOT NULL - 非空   
UNIQUE - 唯一   
PRIMARY KEY - 主键   
FOREIGN KEY - 外键   
CHECK - 条件检查   
DEFAULT - 默认   
二、主键 PRIMARY KEY  
我们还是进入 SQLite 命令行环境，建立一个 test.db 数据库用来做实验，如下

[myqiao@ubuntu:~/My](mailto:myqiao@ubuntu:~/My) Documents/db$ sqlite3 test.db  
-- Loading resources from /home/myqiao/.sqliterc  
SQLite version 3.7.4  
Enter ".help" for instructions  
Enter SQL statements terminated with a ";"  
sqlite> .tables   
sqlite>    
              
运行 .tables 命令没有返回，说明数据库是空的。如果你的数据库里面有内容并影响到下面的实验， 你可以用我们上一篇学的 DROP TABLE 来删除造成影响的表， 或者用 ALTER TABLE ... RENAME TO ... 来改名。

--------------------------------------------------------------------------------

下面言归正转，我们来说说主键 PRIMARY KEY 。

首先，数据表中每一条记录都有一个主键， 这就像我们每的身份证号码、员工号、银行帐号； 反过来也可以说，每一个主键对应着一条数据记录。 所以，主键必须是唯一的。   
其次，一般情况下主键同时也是一个索引，所以通过主键查找记录速度比较快。   
第三，在关系型数据库中，一个表的主键可以作为另外一个表的外键， 这样，这两个表之间就通过这个键建立了关系。   
最后，主键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。

--------------------------------------------------------------------------------

下面我们来做实验

sqlite>    
sqlite> CREATE TABLE Teachers(Id integer PRIMARY KEY,Name text);  
sqlite> .tables  
Teachers  
sqlite> INSERT INTO Teachers(Name) Values(张三);  
sqlite> INSERT INTO Teachers(Name) Values(李四);  
sqlite> INSERT INTO Teachers(Name) Values(王二麻子);  
sqlite> SELECT \* FROM Teachers;  
Id          Name        
----------  ----------  
1           张三      
2           李四      
3           王二麻   
sqlite> INSERT INTO Teachers(Id,Name) Values(2,孙悟空);  
Error: PRIMARY KEY must be unique  
sqlite>  

--------------------------------------------------------------------------------

我们先新建了一个 Teachers 表，并设置了两个字段列，其中 Id 字段列为主键列。 然后，我们向其中插入三条数据并查询，反馈一切正常。

注意：在插入前三条数据的时候，命令中并没有明确指明 Id 的值，系统自动赋值，并且数值自动增长。

插入第四条数据的时候，我给了一个明确的 Id 编号为 2，因为李四的编号已经是 2 了， 所以系统提示我错误：主键必须唯一。

三、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。

下面我们来做实验

sqlite>    
sqlite> DROP TABLE Teachers;  
sqlite> .tables  
sqlite>   
sqlite> CREATE TABLE Teachers(Id integer PRIMARY KEY,Name text,Country text DEFAULT 中国);  
sqlite> .tables  
Teachers  
sqlite> INSERT INTO Teachers(Name) Values(张三);  
sqlite> INSERT INTO Teachers(Name) Values(李四);  
sqlite> INSERT INTO Teachers(Name) Values(王二麻子);  
sqlite> INSERT INTO Teachers(Name,Country) Values(孙悟空,天庭);  
sqlite> SELECT \* FROM Teachers;  
Id    Name             Country          
----  ---------------  ---------------  
1     张三           中国           
2     李四           中国           
3     王二麻子     中国           
4     孙悟空        天庭   
sqlite>   
              
先把之前的 Teachers 表删除，然后重新创建。这回 Teachers 表多了一个 Country 字段， 并且设置默认值为“中国”，然后我们插入四条数据到 Teachers 表。

前三条数据都没有明确指明 Country 字段的值，只有第四条数据指明了“孙悟空”的 Country 为“天庭”。

查询数据，发现前三条数据都填上了默认值，实验成功。

--------------------------------------------------------------------------------  
数据显示有点走样，命令 .width 4 15 15 设置的列宽，可以通过 .show 查看， 可能是因为中文的原因，所以没有对齐。   
四、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。

下面我们来做实验

sqlite>    
sqlite> DROP TABLE Teachers;  
sqlite> .tables  
sqlite>   
sqlite> CREATE TABLE Teachers(Id integer PRIMARY KEY,Name text,Age integer NOT NULL,City text);  
sqlite> .tables  
Teachers  
sqlite> INSERT INTO Teachers(Name,Age) Values(Alice,23);  
sqlite> INSERT INTO Teachers(Name,Age) Values(Bob,29);  
sqlite> INSERT INTO Teachers(id,Name,Age) Values(6,Jhon,36);  
sqlite> SELECT \* FROM Teachers;  
Id    Name             Age              City             
----  ---------------  ---------------  ---------------  
1     Alice            23               NULL             
2     Bob              29               NULL             
6     Jhon             36               NULL             
sqlite> INSERT INTO Teachers(Name) Values(Mary);  
Error: Teachers.Age may not be NULL  
sqlite>   
              
还是先删除旧表，创建新表。

这回 Teachers 表声明了一个 NOT NULL 字段 Age，同时还有一个可以为 NULL 的字段 City

插入前三条数据都没有指定 City 的值，查询可以看到 City 字段全部为空

注意：这里的 NULL 只是对“什么都没有”的一种显示形式， 可以通过 .nullvalue 命令改为别的形式，具体见第一篇

插入第四条数据时没有指定 Age 的值，系统就报错了： Teachers.Age 不能为空

五、 唯一 UNIQUE  
这一约束很好理解，除了主列以为，还有一些列也不能有重复值。不多说，直接看代码

sqlite>    
sqlite> DROP TABLE Teachers;  
sqlite> .tables  
sqlite>   
sqlite> CREATE TABLE Teachers(Id integer PRIMARY KEY,Name text UNIQUE);  
sqlite> .tables  
Teachers  
sqlite> INSERT INTO Teachers(Name) VALUES(Alice);  
sqlite> INSERT INTO Teachers(Name) VALUES(Bob);  
sqlite> INSERT INTO Teachers(Name) VALUES(Jane);  
sqlite> INSERT INTO Teachers(Name) VALUES(Bob);  
Error: column Name is not unique  
sqlite>   
              
这次的 Teachers 表只有 Name 这一列，但是 Name 列不能有重复值。可以看到，到我们第二次插入 Bob 时，系统就报错了。

六、 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。

sqlite>    
sqlite> DROP TABLE Teachers;  
sqlite> .tables  
sqlite>   
sqlite> CREATE TABLE Teachers(Id integer PRIMARY KEY,Age integer CHECK(Age>22));  
sqlite> .tables  
Teachers  
sqlite> INSERT INTO Teachers(Age) VALUES(45);  
sqlite> INSERT INTO Teachers(Age) VALUES(33);  
sqlite> INSERT INTO Teachers(Age) VALUES(23);  
sqlite> INSERT INTO Teachers(Age) VALUES(21);  
Error: constraint failed  
sqlite>   
              
Age 字段要求必须大于 22，当插入的数据小于22时，系统报错。

七、外键 FOREIGN KEY  
现在，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。

sqlite>   
sqlite> .tables  
Teachers  
sqlite> CREATE TABLE Students (Id integer PRIMARY KEY,  TeacherId integer,  FOREIGN KEY(TeacherId) REFERENCES Teachers(id) );  
sqlite> .tables  
Students  Teachers  
sqlite> SELECT \* FROM Teachers;  
Id    Age              
----  ---------------  
1     40               
2     33               
3     23   
sqlite> INSERT INTO Students(TeacherId) VALUES(1);  
sqlite> INSERT INTO Students(TeacherId) VALUES(3);  
sqlite> INSERT INTO Students(TeacherId) VALUES(9);  
sqlite> SELECT \* FROM Students;  
Id    TeacherId        
----  ---------------  
1     1                
2     3                
3     9   
sqlite>   
              
这里建立了 Students 表，并且把 TeacherId 作为外键与 Teachers 表的 Id 列相对应。

问题来了：插入的前两条数据没问题，因为 Id 编号 1、3 都在 Teachers 表中； 但是数字 9 并不在 Teachers 表中，不但没有报错，系统还保存进去了，这是为什么呢？

据说 SQLite 的外键约束默认情况下并不是开启的，如果你需要这个功能，你可能需要[下载](http://www.2cto.com/soft)源代码版本，设置每个编译参数，然后重新编译，这样你就得到支持外键的 SQLite 了。