**[sqlite3: SQLite数据库的命令行接口 Command Line Shell For SQLite](http://www.cnblogs.com/gzggyy/archive/2012/08/07/2627026.html)**

2016年8月5日

22:57

[**sqlite3: SQLite数据库的命令行接口 Command Line Shell For SQLite**](http://www.cnblogs.com/gzggyy/archive/2012/08/07/2627026.html)

原文地址：http://blogold.chinaunix.net/u3/90025/showart\_1764687.html

原文地址：http://www.sqlite.org/sqlite.html

    SQLite库包含一个名字叫做**sqlite3**的命令行,它可以让用户手工输入并执行面向SQLite数据库的SQL命令。本文档提供一个样使用sqlite3的简要说明。

**开始**

      启动**sqlite3**程序，仅仅需要敲入带有SQLite数据库名字的"**sqlite3**"命令即可。如果文件不存在，则创建一个新的（数据库）文件。然后**sqlite3**程序将提示你输入SQL。敲入SQL语句（以分号"；"结束），敲回车键之后，SQL语句就会执行。

        例如，创建一个包含一个表"tb11"名字为"ex1"的SQLite数据库，你可以这样做：

$**sqlite3 ex1**

SQLite version 3.3.17

Enter ".help" for instructions

sqlite> **create table tbl1(one varchar(10), two smallint);**

sqlite> **insert into tbl1 values('hello!', 10);**

sqlite> **insert into tbl1 values('goodbye', 20);**

sqlite> **select \* from tbl1;**

hello!|10

goodbye|20

sqlite>

        你可以通过敲你所用系统的文件结束符（通常是Ctrl + D）或者中断字符（通常是Ctrl + C）。来终止sqlite3程序。确定你在每个SQL语句结束敲入分号！sqlite3程序通过查找分号来决定一个SQL语句的结束。如果你省略分号，sqlite3将给你一个连续的命令提示符并等你给当前的SQL命令添加更多的文字。这个特点让你输入多行的多个SQL语句，例如：

sqlite> **create table tbl2(**

   ...>**f1 varchar(30) primary key,**

   ...> **f2 text,**

   ...> **f3 real**

   ...> **);**

sqlite>

**题外话：查询SQLITE\_MASTER表**

    SQLite数据库的框架被保存在一个名叫"sqlite\_master"的特殊的表中。你可以像查询其它表一样通过执行"SELECT"查询这个特殊的表。例如：

$ **sqlite3 ex1**

SQlite vresion 3.3.10

Enter ".help" for instructions

sqlite> **select \* from sqlite\_master;**

    type = table

    name = tbl1

tbl\_name = tbl1

rootpage = 3

     sql = create table tbl1(one varchar(10), two smallint)

sqlite>

    但你不能在sqlite\_master表中执行诸如DROP TABLE, UPDATE, INSERT 或者DELETE命令。sqlite\_master表在你创建、删除和索引数据库时自动更新这个表。你不能手工更改sqlite\_master表。

    TEMPORARY表的结构没有存储在"sqlite\_master"表中，由于TEMPORARY表对应用是不可见的，而不是应用程序创建这个表。TEMPORARY表结构被存储在另外一个名叫"sqlite\_temp\_master"的特定的表中。"sqlite\_temp\_master"表是临时表自身。

**sqlite3的特殊命令**

   大多数候，sqlite3读入输入行，并把它们传递到SQLite库中去运行。但是如果输入行以一个点（"."）开始，那么这行将被sqlite3程序自己截取并解释。这些"点命令"通常被用来改变查询输出的格式，或者执行鞭个预封包（预定义prepackaged）的查询语句。

你可以在任何时候输入".help"，列出可用的点命令。例如

sqlite> **.help**

.bail ON|OFF           Stop after hitting an error.  Default OFF

.databases             List names and files of attached databases

.dump ?TABLE? ...      Dump the database in an SQL text format

.echo ON|OFF           Turn command echo on or off

.exit                  Exit this program

.explain ON|OFF        Turn output mode suitable for EXPLAIN on or off.

.header(s) ON|OFF      Turn display of headers on or off

.help                  Show this message

.import FILE TABLE     Import data from FILE into TABLE

.indices TABLE         Show names of all indices on TABLE

.load FILE ?ENTRY?     Load an extension library

.mode MODE ?TABLE?     Set output mode where MODE is one of:

                         csv      Comma-separated values

                         column   Left-aligned columns.  (See .width)

                         html     HTML <table> code

                         insert   SQL insert statements for TABLE

                         line     One value per line

                         list     Values delimited by .separator string

                         tabs     Tab-separated values

                         tcl      TCL list elements

.nullvalue STRING      Print STRING in place of NULL values

.output FILENAME       Send output to FILENAME

.output stdout         Send output to the screen

.prompt MAIN CONTINUE  Replace the standard prompts

.quit                  Exit this program

.read FILENAME         Execute SQL in FILENAME

.schema ?TABLE?        Show the CREATE statements

.separator STRING      Change separator used by output mode and .import

.show                  Show the current values for various settings

.tables ?PATTERN?      List names of tables matching a LIKE pattern

.timeout MS            Try opening locked tables for MS milliseconds

.width NUM NUM ...     Set column widths for "column" mode

sqlite>

**改变输出格式**

    sqlite3程序可以以八种不同的格式显示一个查询的结果："csv", "列", "html", "插入", "行", "制表"和"tcl"。你可以用".mode"点命令在这些输出格式之间切换。

    默认的输出格式是"列表"。在列表模式下，每条查询结果记录被写在一行中并且每列之间以一个字符串分割符隔开。默认的分隔符是一个管道符号（"|"）。列表符号在当你输出查询结果到另外一个符加处理的程序（如AWK）中去是尤为有用。

sqlite> **.mode list**

sqlite> **select \* from tbl1;**

hello|10

goodbye|20

sqlite>

    你可以用".separator"点命令来改变分界符。例如，为了把分割符改为一个逗号和一个空格，你可以这样做：

sqlite> **.separator ", "**

sqlite> **select \* from tbl1;**

hello, 10

goodbye, 20

sqlite>

    在"line"模式下，每一个位于条记录中的列在它自己那行显示。每行由列名、一个等号和列数据组成。下一条记录以一个空行隔开。这是一个行模式输出的例子：

sqlite> **.mode line**

sqlite> **select \* from tbl1;**

one = hello

two = 10

one = goodbye

two = 20

sqlite>

    在列模式下，每条记录在一个单独的行中以数据列对齐的方式显示。列如：

sqlite> **.mode column**

sqlite> **select \* from tbl1;**

one         two

----------  ----------

hello       10

goodbye     20

sqlite>

        在默认的情况下，每列至少10个字符宽。太宽的数据将被截取。你可以用".width"命令来调整列宽。如下所示：

sqlite> **.width 12 6**

sqlite> **select \* from tbl1;**

one           two

------------  ------

hello         10

goodbye       20

sqlite>

    上面例子中".width"命令设置第一列宽为12第二列宽为6。其它的列宽不变。你可以指定与你查询结果需要的列数一样多的".width"参数。

    如果你指定一列宽为0，那么这个列宽将自动以下面三个数字中的最大值做为列宽：10、表头宽度和最宽的数据列的宽度。这可以让列自动调整宽度。每列的默认设置为自动调整的0值。

    出现在输出开头两行的列标示可以用".header"点命令关闭。在上面的例子中，列标示是打开的。可以用下面的方法关闭列标示：

sqlite> **.header off**

sqlite> **select \* from tbl1;**

hello         10

goodbye       20

sqlite>

    另外一个有用的输出模式是"insert"。在插入模式下，被子格式化为看起来像SQL INSERT语句的样式。你可以用插入模式来产生文件（便于）以后用于不同数据库的输入。

    当指定插入模式时，你必须给定一个特定参数就是要插入的表名。例如：

sqlite> **.mode insert new\_table**

sqlite> **select \* from tbl1;**

INSERT INTO 'new\_table' VALUES('hello',10);

INSERT INTO 'new\_table' VALUES('goodbye',20);

sqlite>

    最新的输出格式是"html"。在这种模式下，sqlite3把查询的结果写做XHTML表。开始的<TABLE>和结束的</TABLE>（标记）没有写出，但有<TR>、<TH>和<TD>等分界符。html输出对CGI来说是相当有用地。

**把结果写到文件中**

    默认情况下，sqlte3把结送到标准输出。你可以用".output"命令改变它。只须把输出文件名做为.output命令的输出参数然后所有后续查询结果将被写到那个文件中。用".output stdout"再一次改为标准输出。例如：

sqlite> **.mode list**

sqlite> **.separator |**

sqlite> **.output test\_file\_1.txt**

sqlite> **select \* from tbl1;**

sqlite> **.exit**

$ **cat test\_file\_1.txt**

hello|10

goodbye|20

$

**查询数据库结构**

    sqlite3程序提供几个有用的用于查询数据库结构的快捷命令。这些不是不可以用别的方式来实现。这些命令仅仅是一个快捷方式而已。

    例如，为了查看数据库的表列表，你可以敲入".tables"。

sqlite> **.tables**

tbl1

tbl2

sqlite>

    ".tables"命令相似于设置列表模式然后执行接下来的查询：

SELECT name FROM sqlite\_master

WHERE type IN ('table','view') AND name NOT LIKE 'sqlite\_%'

UNION ALL

SELECT name FROM sqlite\_temp\_master

WHERE type IN ('table','view')

ORDER BY 1

事实上，你可以查看sqlite3的源代码（可以在源文件树的src/shell.c中），你可找到上面的具体的查询。

".indices"命令作用类似的方式是列出特定表的所有的索引。".indics"命令须一个参数即所要索引表的表名。最后，但不是至少，是".schema"命令。不带任何参数，".schema"命令显示原始的用于创建当前数据库的CREATE TABLE和CREATE INDEX语句。如果你给".schema"命令一个表名，它显示原始的创建该表和它所有索引的CREATE语句。我们可以：

sqlite> **.schema**

create table tbl1(one varchar(10), two smallint)

CREATE TABLE tbl2 (

  f1 varchar(30) primary key,

  f2 text,

  f3 real

)

sqlite> **.schema tbl2**

CREATE TABLE tbl2 (

  f1 varchar(30) primary key,

  f2 text,

  f3 real

)

sqlite>

".schema"命令可以用设置列表然后执行以下查询来实现：

SELECT sql FROM

(SELECT \* FROM sqlite\_master UNION ALL

SELECT \* FROM sqlite\_temp\_master)

WHERE type!='meta'

ORDER BY tbl\_name, type DESC, name

或者，如果你给".schema"命令一个参数，由于你只想得到一个表的结构，查询可以是这样：

SELECT sql FROM

(SELECT \* FROM sqlite\_master UNION ALL

SELECT \* FROM sqlite\_temp\_master)

WHERE type!='meta' AND sql NOT NULL AND name NOT LIKE 'sqlite\_%'

ORDER BY substr(type,2,1), name

你可以担供一个参数给.schema命令。如果这橛，查询可以是这样的：

SELECT sql FROM

(SELECT \* FROM sqlite\_master UNION ALL

SELECT \* FROM sqlite\_temp\_master)

WHERE tbl\_name LIKE '%s'

AND type!='meta' AND sql NOT NULL AND name NOT LIKE 'sqlite\_%'

ORDER BY substr(type,2,1), name

在查询中"%S"为你的参数所取代。这使你可以询数据库结构的某个子集。

sqlite> **.schema %abc%**

与这些一起，".table"命令也接受一个模式作为他的参数。如果你给".table"一个参数，"%"将被前后扩展并且一个LIKE子句被附加到查询上。这让你可以列出只与特定模式相匹配的的表。

".datebasae"命令显示所有当前连接打开的数据库的一个列表。将允许一次到少两个。第一个是"main"，最初打开的那个数据库。第二个是"temp"，用于临时表的数据库。对于用ATTACH语句附加的数据也许有附加数据库列表。输出的第一列与之相联的数据库名，第二列是外部文件名。

sqlite> **.databases**

**将整个数据库转换为ASCII文本文件**

".dump"命令成一个单一的ASCII文本文件。这个文件可以被用作管道传递给sqlite3命令来转换回数据库。

一个最好的制作一个数据库档案拷贝的命令是：

$ **echo '.dump' | sqlite3 ex1 | gzip -c >ex1.dump.gz**

它产生一个一个名为**ex1.dump.gz**的文件，它包含了你以后或在其它机器上重构数据库的所有的信息。要重构数据库，只须敲入：

$ **zcat ex1.dump.gz | sqlite3 ex2**

这个文本格式是纯粹的SQL语句所以你可以用.dump命令来导出一个SQLite数据库到另一个常用的SQL数据库引擎。比如：

$ **createdb ex2**

$ **sqlite3 ex1 .dump | psql ex2**

**其它的点命令**

".explain"命令可以被用来设置输出格式为"column" 并设置列宽为EXPLAIN命令看起来比较合理的宽度。EXPLAIN命令是SQLite特有的SQL扩展，它是对调试有用。如果任何常规的SQL被EXPLAIN执行，那么SQL命令被分解并分析但并不执行。取而代之的是，虚拟机指令序列将被用于执行SQL命令并返回一个类似的查询结果。如：

sqlite> **.explain**

sqlite> **explain delete from tbl1 where two<20;**

addr  opcode        p1     p2     p3

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

0     ListOpen      0      0

1     Open          0      1      tbl1

2     Next          0      9

3     Field         0      1

4     Integer       20     0

5     Ge            0      2

6     Key           0      0

7     ListWrite     0      0

8     Goto          0      2

9     Noop          0      0

10    ListRewind    0      0

11    ListRead      0      14

12    Delete        0      0

13    Goto          0      11

14    ListClose     0      0

".timeout"命令设置sqlite3等待一个试图存储文件锁定请除直到错误返回的总时间。默认的超时值是0因此如果任何需要的数据库表或序列列被锁定时它将立即返回一个错误。

最后，我们提到".exit"命令它交导致sqlite3退出。

**在命令和脚本中使用sqlite3**

一个在脚本命令中使用sqlite3的方式是用"echo"或"cat"来产生一个命令序列在一个文件中，然后当从一个产生的命令行中重定向输入时调用sqlite3。它有用并且适应许多环境。但作为一附加的便利是，sqlite3允许一个单一的SQL语句在命令行中作为数据库名后的第二个参数输入。当sqlite3程序带着两个参数启动时，第二个参数被传递给SQLite库处理，查询以列表模式打印到标准输出，然后程序退出。这个机制被设计用于让sqlite3容易于用于连接诸如"AWK"的程序。例如：

$ **sqlite3 ex1 'select \* from tbl1' |**

> **awk '{printf "<tr><td>%s<td>%s\n",$1,$2 }'**

<tr><td>hello<td>10

<tr><td>goodbye<td>20

$

**结束命令行命令**

SQLite命令通常以一个分号结束。在一个命令行中你也可以用"GO"单词（大小写敏感）或者一个"/"斜线在它所在好行结束一个命令。这常被SQL Server和Oracle使用。这些将不在sqlite3\_exec()中有用，因为命令行在传递它们到函数之前把这些翻译为分号。

**从源文件中编译sqlite3**

sqlite3程序当你编译SQLite库时自动被建立。只须取得一个源文件树的拷贝，运行"configure"然后"make"即可。

SQLite3的Shell(1)

sqlite3.exe是SQLite访问数据库的一个工具，使用命令行的形式。

Shell的命令分为两种，第一是Shell本身的命令，以点开头。第二是用来管理当前数据库的SQL命令，以分号结束。

其中Shell本身的命令分以下几种：控制数据的显示模式；设置Shell读取和显示数据的方式；其他系统控制命令

常用的显示模式控制命令：

.header ON|OFF：设置是否显示数据表格的列名

.indices TABLE：显示表格的所有列名

.mode MODE：设置数据表格的显示模式。默认是list，用.separator命令设置的字符串分隔字符(默认是|)。我们比较习惯的是column格式

.nullvalue STRING：当数据为空时，用STRING来填充。默认是空字符串，也就是什么也不显示。

.tables PATTERN：使用PATTERN模式来显示表格

.width NUM NUM...：设置当以column模式显示时，数据显示的宽度

.output FILENAME|stdout：设置数据输出的形式：文件或屏幕的标准输出。

.read FILENAME：从文件读取SQL语句

其他系统控制命令：

.databases：显示目前已经匹配的数据库名称

.table：显示当前数据库中的所有表

.help：显示shell的帮助信息

.show：显示shell目前的设置参数

.quit 和.exit：退出shell

来自 <[*http://www.cnblogs.com/gzggyy/archive/2012/08/07/2627026.html*](http://www.cnblogs.com/gzggyy/archive/2012/08/07/2627026.html)>