

sqliteでセッションという概念が通じるか試してみる

2012/07/14

Androidのsqliteでinsert後にクローズしてからselectしてる、って話を聞いたので、表題のような疑問がわいた。

今回試したのはAndroidのものでなくFreeBSD版なので、同じことをAndroidでも試す必要は当然ある。

sqliteにもセッション、トランザクションの概念はある

とりあえず

用語が正しいかわからんけど、sqliteで一つのデータベースファイルに複数からの接続があった場合、この接続をセッションと呼んでみる。

sqlite3 コマンドラインが使えるそうか試してみる

そういえば家のFreeBSDに導入していたなーという事でコマンドが使えるか試してみる。

termA

```
$ sqlite3 -h
sqlite3: Error: UNKNOWN OPTION: -h
USE -help FOR a list OF options.
$ sqlite3 -help
Usage: sqlite3 [OPTIONS] FILENAME [SQL]
FILENAME IS the name OF an SQLite DATABASE. A NEW DATABASE IS created
IF the file does NOT previously exist.
OPTIONS include:
  -help                SHOW this message
  -init filename       READ/process named file
  -echo                print commands BEFORE execution
  -[no]header          turn headers ON OR off
  -bail                stop after hitting an error
  -interactive         force interactive I/O
  -batch               force batch I/O
  -COLUMN              SET output mode TO 'column'
  -csv                 SET output mode TO 'csv'
  -html                SET output mode TO HTML
  -line                SET output mode TO 'line'
  -list                SET output mode TO 'list'
  -separator 'x'      SET output FIELD separator (|)
  -stats               print memory stats BEFORE each finalize
  -nullvalue 'text'   SET text string FOR NULL VALUES
  -version             SHOW SQLite version
```

```

-vfs NAME          USE NAME AS the DEFAULT VFS
$

```

お。

termA

```

$ sqlite3
SQLite version 3.7.9 2011-11-01 00:52:41
Enter ".help" FOR instructions
Enter SQL statements TERMINATED WITH a ";"
sqlite> .help
.backup ?DB? FILE      Backup DB (DEFAULT "main") TO FILE
.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
                      IF TABLE specified, ONLY dump TABLES matching
                      LIKE pattern TABLE.

.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.
                      WITH no args, it turns EXPLAIN ON.

.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
                      IF TABLE specified, ONLY SHOW indices FOR
TABLES
                      matching LIKE pattern TABLE.

.log FILE|off         Turn logging ON OR off.  FILE can be
stderr/stdout
.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
.restore ?DB? FILE   Restore content OF DB (DEFAULT "main") FROM FILE
.schema ?TABLE?     SHOW the CREATE statements
                      IF TABLE specified, ONLY SHOW TABLES matching
                      LIKE pattern TABLE.

```

```
.separator STRING      CHANGE separator used BY output mode AND .import
.show                  SHOW the CURRENT VALUES FOR various settings
.stats ON|OFF          Turn stats ON OR off
.tables ?TABLE?       List names OF TABLES
                       IF TABLE specified, ONLY list TABLES matching
                       LIKE pattern TABLE.
.timeout MS            Try opening locked TABLES FOR MS milliseconds
.width NUM1 NUM2 ...  SET COLUMN widths FOR "column" mode
.timer ON|OFF         Turn the CPU timer measurement ON OR off
sqlite> .quit
$
```

おお、動くっぽい。コマンドラインの FILENAME でデータベースファイルを指定なのかな。

termA

```
$ sqlite3 testdb.db
SQLite version 3.7.9 2011-11-01 00:52:41
Enter ".help" FOR instructions
Enter SQL statements TERMINATED WITH a ";"
sqlite> CREATE TABLE sample(
...>   id VARCHAR(5)
...>   ,val NUMBER(5)
...> );
sqlite> .database
seq  name                file
-----
-----
0    main                  /home/tj510/work200/testdb.db
sqlite>
```

正解っぽい。ついでなんで、テスト用にテーブルを定義した。

お次はトランザクションの確認

グーグル先生によればトランザクションサポートってことなんだけど、一応確認してみる。

termA

```
sqlite> SELECT * FROM sample;
sqlite> INSERT INTO sample VALUES('abc',100);
sqlite> INSERT INTO sample VALUES('def',200);
sqlite> SELECT * FROM sample;
abc|100
def|200
sqlite> BEGIN;
sqlite> INSERT INTO sample VALUES('ghi',300);
```

```

sqlite> SELECT * FROM sample;
abc|100
def|200
ghi|300
sqlite> ROLLBACK;
sqlite> SELECT * FROM sample;
abc|100
def|200
sqlite>

```

明示的にトランザクションを指定できるのもわかった。

次はセッション(という用語でいいのかわからないけど)

2つのセッションを開いてみる。多分こういう使い方はNGなんだろうと思うけど。だってsqliteは1ファイル1データベースなんで、下手するとデータベースファイルを壊しちゃう可能性がある。

最初の端末をtermA、新たに起動した端末をtermBとする。

termA	termB
	<pre> termB \$ sqlite3 testdb.db SQLite version 3.7.9 2011-11-01 00:52:41 Enter ".help" FOR instructions Enter SQL statements TERMINATED WITH a ";" sqlite> SELECT * FROM sample; abc 100 def 200 sqlite> </pre>

別セッションでデータベースファイルを指定できて、テーブルの内容も読み出せてる。ここでtermAでの更新を行う。トランザクションは明示しない。

```

termA
sqlite> SELECT * FROM sample;
abc|100
def|200
sqlite> INSERT INTO sample
VALUES('jkl',400);
sqlite> SELECT * FROM sample;
abc|100
def|200
jkl|400
sqlite>

```

この直後にtermBでSELECTしてみる。

termA	termB
	<pre>termB sqlite> SELECT * FROM sample; abc 100 def 200 jkl 400 sqlite></pre>
おお、読み出せている。なら次はtermA側の操作をトランザクション付きで実行。	
<pre>termA sqlite> BEGIN; sqlite> INSERT INTO sample VALUES('mno',500); sqlite> SELECT * FROM sample; abc 100 def 200 jkl 400 mno 500 sqlite></pre>	
commitしないでtermBの操作をしてみよう。	
	<pre>termB sqlite> SELECT * FROM sample; abc 100 def 200 jkl 400 sqlite></pre>
うむ。トランザクションは正しく効いている様に見える。ではtermAでcommitしてからもう一度。	
<pre>termA sqlite> commit; sqlite></pre>	
	<pre>termB sqlite> SELECT * FROM sample; abc 100 def 200 jkl 400 mno 500 sqlite></pre>
ばっちり、セッション間でもトランザクションは独立して機能してる。	

INSERTとSELECTを同一セッションにしていないと「INSERTしたのにSELECTできない」様に見えるのかな、と思う。

[技術資料](#), [database](#), [sqlite](#), [SQL](#), [トランザクション](#)

From:

<https://wiki.hgotoh.jp/> - 努力したWiki

Permanent link:

<https://wiki.hgotoh.jp/documents/database/sql-0005>

Last update: **2023/04/14 02:32**

