aardio 范例: sqlite 库 - 多线程读写

```
//sqlite 库 - 多线程读写
import console;
import sqlite;

console.log(`尽量不要在客户端软件中多线程同时写sqlite数据库。
把多线程的程序比喻成一家公司，公司只要一个人负责记账就可以了，没必要每个线程都去做这件事。`)
var db = sqlite("/test-sqlite-thread.db")

//创建表
if( not db.existsTable("film") ){
    db.exec( "CREATE TABLE [film](title, length, year, starring);")
}

//创建线程
var func = function(){

    import sqlite;
    var db = sqlite("/test-sqlite-thread.db")

    //多线程冲突锁定时的重试次数
    db.busyTimeout(10000);
    thread.lock("PRINT",λ() io.print("正在写数据库,线程ID:",thread.getId()) )

    var command = db.prepare("REPLACE INTO film VALUES (@title,@length,@year, 'Jodie Foster');" )
    for(i=1;10;1){
        command.step(
            title = "标题";
            length = 4;
            year = thread.getId();
        )
    }

    command.finalize();
    db.close();
}

var t1 = thread.create( func )
var t2 = thread.create( func )
var t3 = thread.create( func )
var t4 = thread.create( func )
var t5 = thread.create( func )
var t6 = thread.create( func )

thread.waitClose(t1,t2,t3,t4,t5,t6)

for title, length, year, starring in db.each("SELECT * FROM film") {
    console.log( title, length, year, starring  )
}

//删除表
db.exec("DROP TABLE film" );

console.pause(true);
```

Markdown 格式