

aaudio 范例: sqlite 库 - 预处理命令

```
//sqlite 库 - 预处理命令
import sqlite

//创建测试数据库
var db = sqlite("/testParameters.db")
if( not db.existsTable("film") ){
    db.exec( "CREATE TABLE film(title, length, year, starring);" )
}

/*
下面 SQL 语句中用@表示的命名参数 - 这是 sqlite 提供的原生参数化,
由 cmd.bind.parameterAtNames 函数绑定参数, 这种命名参数
不支持由 sqlite.format 函数参数化查询中的数组、针对一个命名参数的名值对参数化等功能。

如果sqlConnection.prepare() 函数的第2个参数是一个数组,
则调用 sqlite.format 格式化 SQL 语句中的 ?? 参数 (不格式化 ? 或 @ 参数)

如果 db.prepare() 函数的第2个参数表包含名值对成员,
则这些名值对被自动创建为 WHERE 条件语句 ( 等价于格式化 SQL 语句中的 "WHERE ??")
*/
var cmd = db.prepare("INSERT INTO film VALUES (@title,@length,@year, 'Jodie Foster');" )

//绑定命名参数并执行一次预处理命令
cmd.step(
    title = "标题11111";
    length = 4;
    year = time.now();
)

//上面的代码等价于下面的代码
cmd.reset();
cmd.bind.parameterAtNames(
    ["title"] = "标题";
    ["length"] = 4;
    ["year"] = time.now();
);
cmd.step();

/*
aaudio 中的 string 类型字符串存为 sqlite 的 text 类型(UTF-8文本),
aaudio 中的 buffer 类型字节数组存为 sqlite 的 blob 类型。
可在预处理语句的命名参数名中使用 blob,utf8,utf16 前缀指定字符串值的编码、或数据类型。
*/
cmd.prepare("INSERT INTO film VALUES (@blobTitle,@length,@year,'Jodie Foster');" );
//注意上面我们用 cmd.prepare() 更新了预处理命令对象的 SQL 语句。

//预处理 SQL 语句里的问号表示匿名参数 ( 不要和 sqlite.format 的数组参数混淆 )
cmd.prepare("INSERT INTO film VALUES (?, ?, ?, 'Jodie Foster');" )

//按参数顺序绑定多个参数
cmd.step( {
    '用 string.loadBuffer() 读入二进制字符\0串', //设定第一个?号表示的参数
    null, //设定第二个?号表示的参数
    time.now() //设定第三个?号表示的参数
})

//上面的代码等价于下面的代码
cmd.reset();
cmd.bind.blob('用 string.loadBuffer() 读入二进制字符\0串');
cmd.bind.double(123,2/*参数二*/ );
cmd.bind.text( toString( time.now() ),3/*参数三*/ );
cmd.step();

//释放预处理命令对象
cmd.finalize();

//迭代方式查询数据
import console;
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" );
execute("pause")
```

