

aardio 范例: sqlite 库 - 全文搜索

```
//sqlite 库 - 全文搜索
import sqlite;
import console.int;

var db = sqlite("/test-fts.db");
if(!db.existsTable("ftstest")){
    //tokenize 指定分词器, fts5 改用 unicode61 (需要改用 sqlite.latest 库)
    db.exec("CREATE VIRTUAL TABLE ftstest USING fts4 (data,body,tokenize=simple)");
}

import mmseg
var str = /*
一些普通名词作为专有名词使用时, 通常需要大写首字母以示区分。
例如 Python, python 分别表示不同的意思。

但有些名词本就是专有名词, 并不需要大写首字母。
例如 eBay, iPhone, adidas , aria2 ..... 以及 aardio 。

logo 里使用小写首字母品牌或产品名称就更多了。
小写首字母体现友好、接地气、非传统、简洁、务实与创新的精神。

aardio 在编码中也保持小写首字母的驼峰式命名风格。
仅仅在调用其他语言编写的接口或组件时, 会保持原来的大写首字母以示区分。
*/

var data = "";
for word,attr in mmseg.each(str){
    //SQLite 自带的 simple 分词器用空格分词
    if(attr) data = data + " " + word;
}

//创建预处理命令,用 @ 字符作为 SQL 命名参数的前缀
var cmd = db.prepare("INSERT INTO [ftstest] VALUES ( @data,@body );");

//执行命令语句,插入测试数据,并指定 SQL 命名参数
cmd.step(
    data = data;
    body = str;
);

cmd.finalize();

//MATCH 用于全文检索, simple 或 unicode61 分词器在查询时忽略大小写。
var result = db.stepQuery("SELECT body FROM [ftstest] WHERE data MATCH '专有名词'");

//在词前面加^ 表示该词必须是某列的第一个词
var result = db.stepQuery("SELECT body FROM [ftstest] WHERE data MATCH '^一些'");

//在词后面加* 表示匹配所有以该词为前缀的词
var result = db.stepQuery("SELECT body FROM [ftstest] WHERE data MATCH '专有*');

//可以空格分隔多个必须包含的查询词, 相当于使用隐式 AND 操作符
var result = db.stepQuery(`SELECT body FROM [ftstest] WHERE data MATCH '小写 首字母'`);

//包含多个词的短语首尾要加双引号
var result = db.stepQuery(`SELECT body FROM [ftstest] WHERE data MATCH '"小写 首字母"'`);

//支持逻辑操作符 OR 分隔查询词 (找到其中一个即可)。
var result = db.stepQuery(`SELECT body FROM [ftstest] WHERE data MATCH '小写 OR 首字母'`);
//改用库 sqlite.latest, 并改用 FTS5 可支持逻辑操作符 NOT AND 。

//查看结果
console.dump(result[["body"]]);

//删除表
db.exec("DROP TABLE ftstest");
```