SQLite

# 简介

SQLite，是一款轻型的数据库，是遵守[ACID](https://baike.baidu.com/item/ACID/10738)的关系型[数据库管理系统](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F)。它的设计目标是[嵌入式](https://baike.baidu.com/item/%E5%B5%8C%E5%85%A5%E5%BC%8F/575465)的，而且已经在很多[嵌入式产品](https://baike.baidu.com/item/%E5%B5%8C%E5%85%A5%E5%BC%8F%E4%BA%A7%E5%93%81/9245765)中使用了它，它占用资源非常的低，在[嵌入式设备](https://baike.baidu.com/item/%E5%B5%8C%E5%85%A5%E5%BC%8F%E8%AE%BE%E5%A4%87/10055189)中，可能只需要几百K的内存就够了。它能够支持Windows/Linux/Unix等等主流的[操作系统](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F/192" \t "_blank)，同时能够跟很多程序语言相结合，比如 Tcl、C#、PHP、Java等，还有ODBC接口，同样比起Mysql、PostgreSQL这两款开源的世界著名[数据库管理系统](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/1239101" \t "_blank)来讲，它的处理速度比他们都快。

## 工作原理

SQLite[引擎](https://baike.baidu.com/item/%E5%BC%95%E6%93%8E)不是个[程序](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F)与之通信的独立进程，而是连接到程序中成为它的一个主要部分。所以主要的通信协议是在[编程](https://baike.baidu.com/item/%E7%BC%96%E7%A8%8B)语言内的直接API调用。这在消耗总量、延迟时间和整体简单性上有积极的作用。整个数据库(定义、表、索引和数据本身)都在宿主主机上存储在一个单一的文件中。它的简单的设计是通过在开始一个事务的时候锁定整个[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE)文件而完成的。

## 功能特性

1. ACID事务[1]

2. 零配置 – 无需安装和管理配置

3.[储存](https://baike.baidu.com/item/%E5%82%A8%E5%AD%98)在单一[磁盘](https://baike.baidu.com/item/%E7%A3%81%E7%9B%98)文件中的一个完整的数据库

4.[数据库文件](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E6%96%87%E4%BB%B6)可以在不同[字节顺序](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82%E9%A1%BA%E5%BA%8F)的机器间自由的共享

5.[支持数据库](https://baike.baidu.com/item/%E6%94%AF%E6%8C%81%E6%95%B0%E6%8D%AE%E5%BA%93)大小至2TB

6. 足够小, 大致13万行C代码, 4.43M

7. 比一些流行的数据库在大部分普通数据库操作要快

8. 简单, 轻松的API

9. 包含TCL绑定, 同时通过Wrapper支持其他语言的绑定

10. 良好注释的[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81" \t "_blank), 并且有着90%以上的测试覆盖率

11. 独立: 没有额外依赖

12. 源码完全的开源, 你可以用于任何用途, 包括出售它

## 数据类型

CREATE TABLE ex2

a VARCHAR(10),

b NVARCHAR(15),

c TEXT,

d INTEGER,

e FLOAT,

f BOOLEAN,

g CLOB,

h BLOB,

i TIMESTAMP,

j NUMERIC(10,5)

k VARYING CHARACTER (24),

l NATIONAL VARYING CHARACTER(16)

## 常用函数

SQLite 有许多内置函数用于处理字符串或数字数据。下面列出了一些有用的 SQLite 内置函数，且所有函数都是大小写不敏感，这意味着您可以使用这些函数的小写形式或大写形式或混合形式。欲了解更多详情，请查看 SQLite 的官方文档：

序号 函数 & 描述

1 SQLite COUNT 函数　SQLite COUNT 聚集函数是用来计算一个数据库表中的行数。

2 SQLite MAX 函数　SQLite MAX 聚合函数允许我们选择某列的最大值。

3 SQLite MIN 函数　SQLite MIN 聚合函数允许我们选择某列的最小值。

4 SQLite AVG 函数　SQLite AVG 聚合函数计算某列的平均值。

5 SQLite SUM 函数　SQLite SUM 聚合函数允许为一个数值列计算总和。

6 SQLite RANDOM 函数　SQLite RANDOM 函数返回一个介于 -9223372036854775808 和 +9223372036854775807 之间的伪随机整数。

7 SQLite ABS 函数　SQLite ABS 函数返回数值参数的绝对值。

8 SQLite UPPER 函数　SQLite UPPER 函数把字符串转换为大写字母。

9 SQLite LOWER 函数　SQLite LOWER 函数把字符串转换为小写字母。

10 SQLite LENGTH 函数　SQLite LENGTH 函数返回字符串的长度。

11 SQLite sqlite\_version 函数　SQLite sqlite\_version 函数返回 SQLite 库的版本。

## 官网地址

<https://www.sqlite.org/index.html>

## 自己测试

### 免费版参考链接：

#### 重点：

https://www.cnblogs.com/desireyang/p/13234985.html

<https://www.jianshu.com/p/0f87799fe5ce>

1. 建项目
2. 配置build.gradle文件

# 加密方式

对数据库加密的思路有两种：

## 1. 将内容加密后再写入数据库

这种方式使用简单，在入库/出库只需要将字段做对应的加解密操作即可，一定程度上解决了将数据赤裸裸暴露的问题。

不过这种方式并不是彻底的加密，因为数据库的表结构等信息还是能被查看到。另外写入数据库的内容加密后，搜索也是个问题。

## 2. 对数据库文件加密

将整个数据库整个文件加密，这种方式基本上能解决数据库的信息安全问题。目前已有的SQLite加密基本都是通过这种方式实现的。

## SQLite Encryption Extension (SEE)

事实上SQLite有加解密接口，只是免费版本没有实现而已。而SQLite Encryption Extension (SEE)是SQLite的加密版本，提供以下加密方式：

RC4

AES-128 in OFB mode

AES-128 in CCM mode

AES-256 in OFB mode

<https://blog.csdn.net/ccm_oliver/article/details/46346127>

C中有一个可用的加密库（sqlite-crypt）

# [Android数据存储之SQLCipher数据库加密](https://www.cnblogs.com/whoislcj/p/5511522.html)

## java demo

<https://stackoverflow.com/questions/23170836/jdbc-for-sqlcipher>

源码：

<https://github.com/Willena/sqlite-jdbc-crypt#encryption_config_cipher>

SQLiteMCWxAES256Config

AES 256位CBC-无HMAC（wxSQLite3） aes256cbc

## 测试

用SQLiteStudio\_4828 测试 ， Wxsqlite ok 输入密码解锁

SQLCipher 代码可以通，用SQLiteStudio\_4828 测试 都填写正确程序就崩了

## 集成mybits

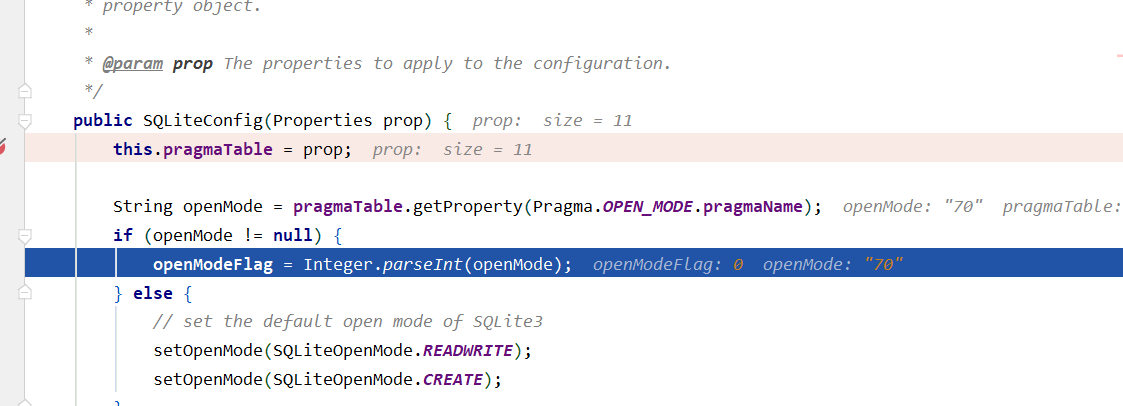
**public static** Connection getConnection***(***String url,  
 java.util.Properties info***)* throws** SQLException ***{* return *(****getConnection****(***url, info, Reflection.*getCallerClass****()))***;  
***}***

SQLiteMCWxAES256Config.getDefault().withKey("1234567").toProperties()

107

102

121



## 分析mybits

### 最基本的使用MyBatis的代码

public static void main(String[] args) throws Exception {

String resource = "mybatis-config.xml";

InputStream inputStream = Resources.getResourceAsStream(resource);

//创建SqlSessionFacory

SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*分割线\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

SqlSession sqlSession = sqlSessionFactory.openSession();

//获取Mapper

DemoMapper mapper = sqlSession.getMapper(DemoMapper.class);

Map<String,Object> map = new HashMap<>();

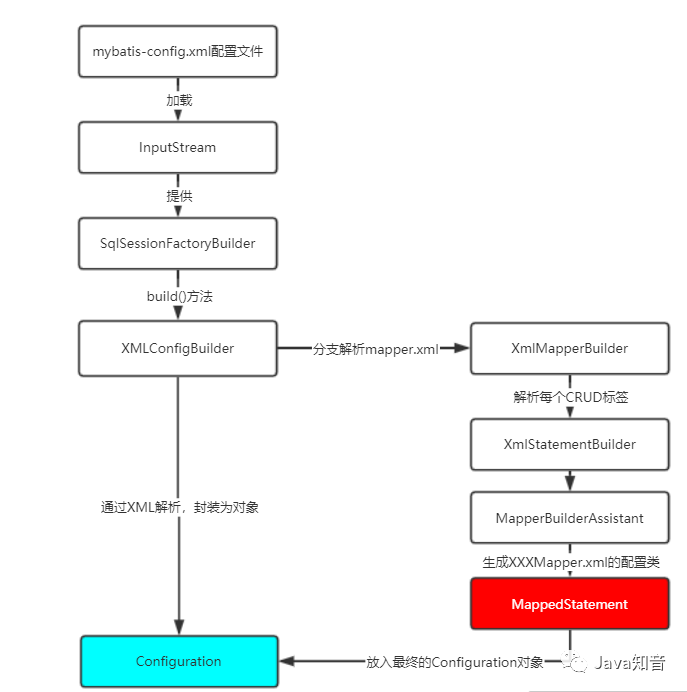
map.put("id","123");

System.out.println(mapper.selectAll(map));

sqlSession.close();

sqlSession.commit();

}



https://www.cnblogs.com/javazhiyin/p/12340498.html

### 改造参考

https://blog.csdn.net/java\_magicsun/article/details/80825133

# 收费

**SQLite加密扩展（SEE）。** 公共领域SQLite源代码的直接替代品，具有读取/写入AES加密数据库的附加功能。 $ 2000一次性费用 13061.4人民币元

SQLite 3.7.13的加密解密（四）—— 挂接加密解密函数

https://developer.aliyun.com/article/432176