

# 基于 SQLite 的二次封装方法在车载导航系统中的应用

## Application of Secondary Encapsulating Methods Based on SQLite in the Vehicle Navigation System

岑冬梅 陈和平 ( 武汉科技大学 信息科学与工程学院 湖北 武汉 430081 )

张剑波 ( 中国地质大学 信息工程学院 湖北 武汉 430074 )

**摘要:** 嵌入式数据库 SQLite 是一个基于磁盘文件系统的关系数据库,具有占用内存小、速度快、效率高及可移植等特点,非常适合于硬件资源有限的嵌入式系统。本文介绍了 SQLite 的特点,详细分析了 SQLite 的体系结构、工作机制和内部结构,并针对作者所开发的基于 ARM - Windows CE 平台的车载导航系统,重点叙述了如何采用 SQLite 将海量地图数据予以适当的抽象和二次封装,以降低软件开发复杂度的方法。

**关键词:** 嵌入式数据库 SQLite 封装 车载导航系统

虽然 Oracle、SQL Server 及 MySQL 等关系数据库系统代表着当今数据库应用的主流,但它们并不能满足所有应用场合的需要,比如,对于嵌入式系统应用便失去了用武之地。在嵌入式系统中,由于硬件资源有限,不可能安装庞大的数据库服务器,并且嵌入式数据库的实现还须满足系统的可定制性要求,即系统选择的技术路线要面向具体的行业应用,因而,源码开放的嵌入式数据库 SQLite 的优势便显得较为明显,笔者在开发某车载导航系统中采用了 SQLite 嵌入式数据库。

本车载导航系统是一款运行在 Windows CE 操作系统环境下的软件,它具有电子地图显示与浏览、路径分析、模拟导航、GPS 数据导航及周边信息查询等多项功能,地图数据在物理上采用 ArcGIS 的 Shape 文件格式进行存储,辅助 DBF 文件进行属性说明。在逻辑上,则是按照图幅—专题图层—类型文件形式组织的,作者采用 SQLite 将全国海量地图数据进行了二次封装,降低了应用软件的开发难度,提高了程序执行速度并减少了代码的冗余度。其开发环境为:VC4.0,运行环境:Windows CE .NET 4.2、ARM2410。

### 1 SQLite 的特点

SQLite 是 D. Richard Hipp 于 2000 年开发的一个小

型嵌入式数据库,2005 年赢得了美国 O'Reilly Open Source Conference 的最佳开放源代码软件奖。SQLite 完全独立,不具有外部依赖性,不需任何特殊设定就能跨平台编译,小而省,且其源代码完全开放,可以免费用于包括商业目的在内的任何用途。

#### 1.1 SQLite 的优点

- (1) 内存占有量小,编译后低于 250KB,但能支持容量达 2TB 的数据库;
- (2) 零配置,无需安装和管理配置;
- (3) 支持空间数据;
- (4) 快速高效,比 MySQL 快 2 倍,比 PostgreSQL 快 20 倍;
- (5) ACID(原子性、一致性、独立性、可持久性)兼容,支持视图、子查询及触发器;
- (6) 允许为 SQL 命令集动态添加自定义函数(简单函数及聚集函数),而无需重编 SQLite 库;
- (7) 单个库文件中包含数据库引擎与接口,且其运行不依赖其它库,不需要安装数据库服务器软件,一个文件就是一个数据库;
- (8) 为 C/C++、Perl、PHP 等应用提供了接口;
- (9) 完整的 Unicode 支持(没有跨语系的问题)。

#### 1.2 SQLite 的主要不足

- (1) 事务处理并发性较差 SQLite 通过数据库级的

独占性和共享锁来实现独立事务处理,这意味着当多个进程或线程希望在同一时间从数据库读取数据时,只能有一个可以写入,在写入之前,必须获得独占锁,其它的读操作不允许发生;

(2) 用户管理/安全手段不足 数据库的访问是基于操作系统对文件的控制来实现的,不能由用户来区分文件中的不同数据库;

(3) 在创建索引( CREATE INDEX )和删除表( DELETE TABLE )时比其它数据库慢;

(4) SQLite 只提供了 B 树( B\_tree )数据存储模式,对二进制数据 SQLite 不能直接保存,只能先将二进制数据转换成 ASCII 编码,然后再予以保存。

## 2 SQLite 体系结构和内部结构

### 2.1 SQLite 体系结构

SQLite 的体系结构自底向上可分为操作系统接口层、存储管理层、虚拟机层及用户接口层,如图 1 所示<sup>[1]</sup>。

用户接口层	用户API
	SQL词法、语法及语法分析器
虚拟机层	关系代数引擎
	虚拟机代码编译器
	虚拟机指令执行器
存储管理层	B树管理及存储调度层
	外存页面及缓冲区管理
	内存页面及缓冲区管理
操作系统接口层	可移植的系统调用统一接口层

图 1 SQLite 体系结构

### 2.2 SQLite 内部结构及其工作机制

在内部,SQLite 主要由 4 个组件组成:SQL 编译器( SQL Compiler )、内核( Core )、后端( Backend )以及附件( Accessories )。SQLite 通过利用虚拟机和虚拟数据库引擎( VDBE )使调试、修改和扩展 SQLite 的内核变得更加方便;所有 SQL 语句都被编译成易读的,可以在 SQLite 虚拟机中执行的程序集,SQL 语句通过接口被送到高效的 SQL 命令编译器,由标记处理器( Tokenizer )分解成各个标志符给分析器( Parser ),分析器重新组合标志符并调用代码生成器( Code Generator )生成虚拟机器码,交由给虚拟机( Virtual Machine )执行,完成 SQL 语句指定的任务;在后端,数据库按照 B 树( B\_Tree )的形式存储在磁盘上,通过可调整的页面缓冲( Pager )得到对数据的快速查找和存储。SQLite 还使

用了一个抽象层接口( OS Interface )与不同操作系统进行对接。附件主要是代码测试,内存分配和字符串比较程序位于 util. c 中,如果计算回归测试脚本,多于一半的 SQLite 代码数据库的代码将被测试,其内部结构如图 2 所示<sup>[2]</sup>。

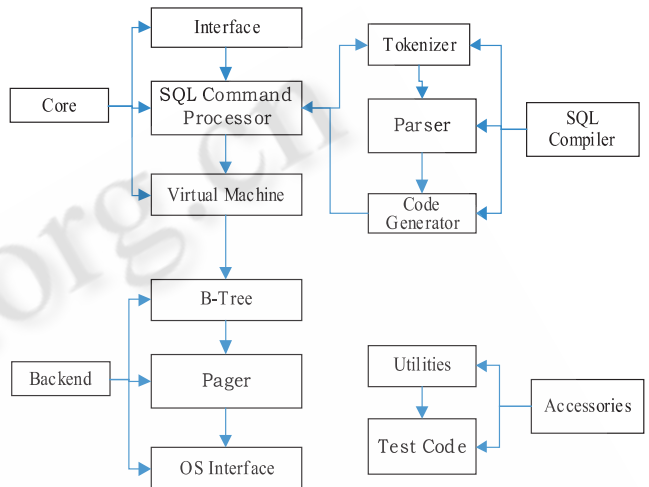


图 2 SQLite 的内部结构

## 3 SQLite 封装

在应用程序开发过程中,如果按常规直接调用 SQLite 提供的底层 C 语言 API 接口,将会导致程序可读性较差,代码维护难度也会相应增大。针对这一问题,作者结合车载导航系统数据管理的需要,在分析 SQLite 接口函数的基础上,对 SQLite 进行了二次封装,把源码和封装的类都生成动态编译库 dll,开发了一套统一的导出接口提供给界面操作调用,较好地达到了降低程序开发复杂度的目的。

### 3.1 SQLite 一次封装

在对 SQLite 进行第一次封装时,作者设计了二个类:嵌入式数据库类( CEDataBase )和 SQL 语句类( CESqlObject ),这两个类相辅相成。CEDataBase 类针对嵌入式数据库的相关操作,主要有创建、连接、关闭数据库、生成 SQL 语句类对象、执行 SQL 语句及批量查询等接口方法,ESqlObject 类则针对 SQL 语句类对象的相关操作,主要的方法有:

(1) 参数传入绑定操作 Bind( )多用于 Insert 语句,例如:INSERT INTO 表名( 字段 1, 字段 2 ) VALUES ( ?, ? ),由于要绑定的数据有不同的类型,所以重载了

四个 Bind ( ) 函数 , 分别为 Integer 型、double 型、Text 型和 BLOB 型 , 参数绑定的索引号从 0 开始 ;

( 2 ) 执行绑定操作 ExecuteBind ( ) 执行顺序为先 Bind , 然后 ExecuteBind , 目前绑定语句只能执行一次 ;

( 3 ) 执行取数据操作 ExecuteFetch ( ) 多用于 Select 语句 , 而且通常是返回多行查询结果 ;

( 4 ) 参数传出绑定操作 GetValue ( ) 由于 Bind ( ) 函数重载了四个函数 , 所以 GetValue ( ) 也对应重载了四个函数 , 调用顺序是先 ExecuteFetch ( ) , 然后 GetValue ( ) 。

### 3.2 SQLite 二次封装

为了更好地利用一次封装后的类管理界面程序数据 ( 界面数据是对用户使用本系统过程中产生的临时数据 , 比如 : 存储地点、目的地履历及经由点等 ) , 作者在 CEdaTase 类和 CESqlObject 类的基础上再次封装了三个具体应用类 : 存储地点类 CStoreSpotOper、目的地履历类 CDestSpotOper 及经由点类 CPassSpotOper , 使数据库的操作更加简单明了 ( 一条语句即可完成操作 ) , 上述五个类的关系图如图 3 所示 :

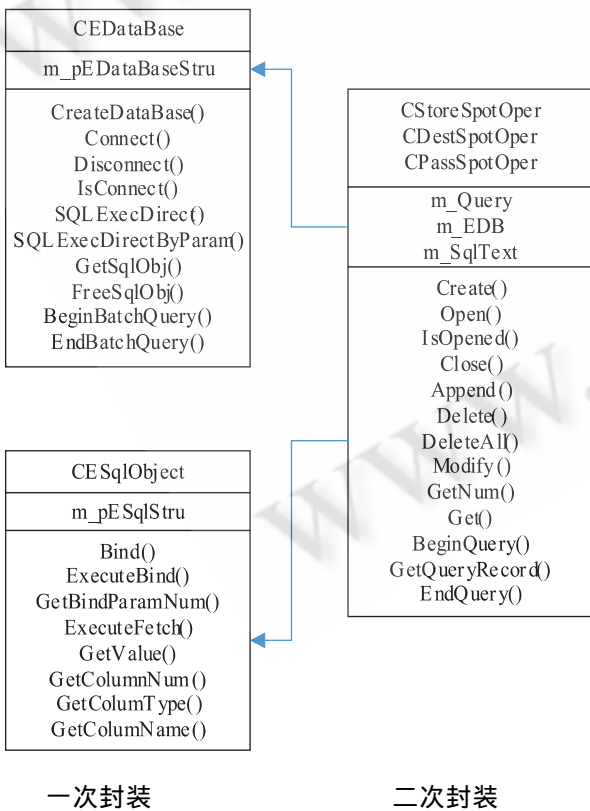


图 3 关系图

图 3 中 , 三个具体应用类的类图类似 , 只是存储的数据不同而已 , 调用 CEdaTase 类方法来进行创建 ( Create )、打开 ( Open )、关闭 ( Close ) 数据库及更新 ( Modify ) 和删除 ( Delete ) 表的操作 , 调用 CESqlObject 类方法来进行插入 ( Append )、查询 ( Get ) 及批处理表的操作。数据库基本操作的流程图如图 4 所示 :

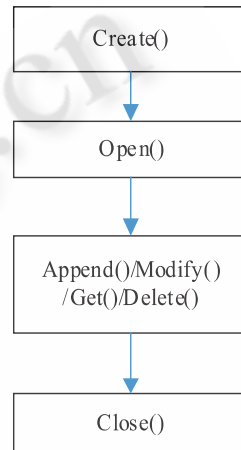


图 4 基本操作流程图

值得注意的是 :

- 创建数据库时先创建一个 .db 文件 , 然后创建一张空表 ;
- 对表的操作须在数据库已经被打开的前提下进行 ;
- 联合批处理操作须成对出现 : BeginQuery ( ) 开始查询、GetQueryRecord ( ) 获取一条记录、EndQuery ( ) 结束查询 , 其中 BeginQuery ( ) 函数传入要查询的 SQL 语句 , 把查询得到的结果放在记录集数组 RecordSet [ ] 中 , 再循环调用 GetQueryRecord ( ) 函数把相应的记录存放到结构体中。

通过对 SQLite 所进行的二次封装 , 可为应用系统开发带来以下好处 :

( 1 ) 系统模块化 ;

( 2 ) 节省内存 , 易于维护 , 当需要功能升级时 , 只需升级底层 dll , 而不需要升级整个系统 ;

( 3 ) 编译好的 dll 可以直接加载到移动设备中 , 不需再次编译 , 节省时间 ;

( 4 ) 有助于资源的共享。

#### 4 应用举例

以上从软件开发的角 度讲述了 SQLite 的特点、结 构和对底层 API 的封装,下面以存储地点类的批处理 查询操作为例,介绍如何应用上述二次封装的类对数 据库进行操作。

```
void CStoreSpotOper : :OnSearch( )
{
    CStoreSpotOper StoreSpot ;//存储地点类
    StoreSpotStrcT MyStore ;//存储地点结构体
    char PathName[ 32 ]=" MySPOT. db " ;//数据库
    名
    CString SqlText ;//输入 SQL 语句
    long RecordRowNum = 0 ;//保存结果数
    int m_tNum = SPOT_SPEC_2 ;//特别记录的宏
    定义
    StoreSpot. Create( PathName ) ;//创建数据库
    short Isopen = StoreSpot. Open( PathName ) ;//
    打开数据库
    if( ! Isopen )//判断数据库是否已经打开
    {
        AfxMessageBox( _T( " 打开数据库错误" ) ) ;
        return ;
    }
    //批量查询存储地点中特别记录 ,TBL_SPOT_INFO
    为表名
    SqlText. Format( _T( " SELECT * FROM TBL_SPOT_
    INFO where SpotLabel = % d " ) ,m_tNum ) ;
    //开始批量查询 ,RecordRowNum 返回结果数
    StoreSpot. BeginQuery( SqlText ,RecordRowNum ) ;
    for( int i = 0 ; i < RecordRowNum ; i + + )
```

```
{
    StoreSpot. GetQueryRecord( i ,MyStore ) ;
    //依次取结果集数据 ,存放到结构体中
    ..... //数据的显示
}
StoreSpot. EndQuery( ) ;//结束批量查询
StoreSpot. Close( ) ;//关闭数据库
}
```

#### 5 结语

嵌入式数据库应用将随着各种移动设备发展而 迅速普及,在车载导航系统中对数据管理的要求程 度较高,笔者利用 SQLite 来进行数据管理,简化了应 用程序的复杂度,提高了程序的运行效率,减少了内 存的消耗,增强了软件的可靠性。SQLite 因其体积小、快速高效、稳定可靠、易移植以及没有版权限制 等优势,将在嵌入式数据库应用中扮演越来越重要 的角色。

#### 参考文献

- 1 王沛汉,国强,周毅等. SQLite 在电话网管系统中的 应用. 计算机工程与设计, 2005, 26( 11 ): 3152 - 3157.
- 2 SQLite 's architecture. [http://www.sqlite.com.cn/MySqlite/5/179. Html](http://www.sqlite.com.cn/MySqlite/5/179.Html). [ 2006 - 7 - 3 ].
- 3 庄宗辉,薛毓强,连瑞红等. 嵌入式数据库 SQLite 在 远程监控系统中的应用. 现代电子技术, 2007, ( 8 ): 62 - 64.
- 4 SQLite homepage. <http://www.sqlite.org>.
- 5 孙鑫,余安萍. VC + + 深入详解. 北京: 电子工业出 版社, 2006. 701 - 731.