数据库系统概论

课程项目报告

何钦尧，王凯，刘晓鸿

2016年1月6日

# 概述

我们完全使用Java作为开发语言来实现了这个项目。完成了数据库系统的几个重要模块的编写，实现了对数据库进行管理和操作的基本功能，以及多表连接，聚集查询，GUI界面，复杂where语句等拓展功能。其中包括自己重新用Java实现的页式文件系统。

# 系统结构设计

根据课程设计的阶段性项目提交，我们的系统结构设计分为：系统管理模块、记录管理模块、SQL解析模块、查询模块。我们自己用Java重新实现了页式文件系统，自行设计了页的布局，将页按功能分为库页、表页、数据页。

## 系统管理模块

系统管理模块负责数据库文件的创建、删除、使用，还有表的创建、删除。还其他提供接口，获得当前数据库及其信息，获得当前数据库的表及其信息。 数据库文件的第一页的一半（4096字节）作为GAM(Global Allocation Map)，用于分配页。

## 记录管理模块

记录管理模块负责记录的插入、删除、更新。

未考虑索引前的增删改策略：

插入：从前往后查找，找到空闲的槽就插入。如果页已满，则顺着页链到下一页，如果已经到最后一页了，则申请新的数据页，然后插入。

删除：被删除的记录，如果在该页的最后一条，则直接记录数减一，如果不是该页最后一条，则将该页最后一条记录移到被删除的记录位置，记录数减一。如果该页已空，则回收该页。

更新：就地更新。

我们没有实现索引功能。如果要实现索引功能，则在插入的时候要按索引键，找到相应的位置，要处理页分裂。删除时要处理页合并。更新时，应该考虑为两个动作：删除一条记录，插入新的记录。

增删改的过程检查primary key、列数据类型等等合法性。

## SQL解析模块

SQL解析模块负责将用户输入的SQL语句进行解析、识别和检查，本模块的最终结果为一个简洁并易于后部进行进一步处理的结果类，后面的模块可以据此进行内部的数据库操作。

## 查询模块

查询模块负责记录的查询任务，包括生成查询计划、执行查询。

## 页布局和行结构

用Global.java来全局配置。

|  |  |  |
| --- | --- | --- |
| 页类型 | 库页 | |
| 位置 | 位置文件的第一页 | |
| 功能 | 库的基本信息 | |
| 布局 | 库信息  256字节 | 表个数0 |
| 页管理位图  4096字节 | 相当于GAM的功能，能管理分配4096\*8=32768页 |
| 表信息  3840字节 | 每个表信息128字节，最多可管理30个表  表页位置0（4字节）  表名4（124字节） |

|  |  |  |
| --- | --- | --- |
| 页类型 | 表页 | |
| 位置 | 创建的时候分配 | |
| 功能 | 表的基本信息 | |
| 布局 | 表信息  256字节 | 表名0（124字节）  表属性128（4字节）  表首页索引132（4字节）  数据页首页索引136（4字节）  列数140（2字节）  记录长度142（2字节）  记录总数144（4字节） |
| 列信息  7936字节 | 每条列信息128字节，一个表最多62列  列名称0（120字节）  列属性120（4字节，32位，0是否允许NULL，1是否primary）  列数据类型124（2字节）  列数据长度126（2字节） |

|  |  |  |
| --- | --- | --- |
| 页类型 | 数据页 | |
| 位置 | 创建的时候分配 | |
| 功能 | 存放数据记录 | |
| 布局 | 数据信息  96字节 | 页号0（4字节）  上一个页4（4字节）  下一个页8（4字节）  数据页属性12（2字节，16位，0是否有主索引）  记录的长度14（2字节）  记录的个数16（4字节）  记录的最大数20（4字节） |
| 行记录  8096字节 | 每条记录的长度为：6+定长总和+ ceil(列数/8)  行结构：  状态位（2字节）  定长部分的长度（2字节）  列数（2字节）  定长数据（n字节）  NULL位图（ceil(列数/8)字节） |

# 系统功能

## 数据库管理

实现SQL的DML功能：

CREATE DATABASE, DROP DATABASE, USE DATABASE

CREATE TABLE, DROP TABLE, SHOW TABLES, DESC TABLE

## 记录管理

实现SQL的DDL中的增删改功能：

INSERT INTO, DELETE FROM, UPDATE TABLE

## 查询

实现SQL的DDL中的查询功能：

SELECT FROM（包括多表连接、聚集查询）

# 主要模块设计原理

## LRU页式文件系统

数据库软件与其他软件的不同之处在于，数据库操作对象是文件，数据在硬盘（大容量、低速存储设备）中，以页为单位载入内存，进行相关操作，然后以页为单位写回。

为了提高数据库的IO效率，需要将页在内存中缓存。页式文件系统就是用来实现页的缓存功能。由于课程提供的是c语言的页式文件系统，而我们用Java开发，因此需要自行实现页式文件系统。

我们主要通过三个类来实现：Page、FilePageManager、BufPageManager，具体参加项目代码。

Page，页，管理一个长度8192字节的byte[]。

FilePageManager，页式文件管理，文件的打开关闭，按页读取文件，得到Page。

BufPageManager，页的缓存管理，当页的缓存满了，用LRU算法（最近最少使用）选择页写回和释放。

## 多表连接

实现方法为，通过迭代将多个表的笛卡尔积求出，生成查询计划，然后在笛卡尔积后的表上执行查询计划。由于这样的处理比较粗暴，没有做一些优化，比较大的表的连接占用的内存空间会很大，当笛卡尔积的结果表大于100万的行时，会报内存空间不足异常。

## 聚集查询

实现方法为，按正常的select的流程将内容取出，然后根据MIN、MAX、AVG、SUM、COUNT的功能做运算，然后返回一个double值。

## SQL解析

SQL语句解析的具体实现框架基本按照LL(1)递归下降分析程序进行设计和编写，首先根据前几个关键词确定语句类型，之后按照SQL标准依次读取其中表名、列名、条件等具体元素，之后转为易于读取的格式并进行存储。

其中条件语句支持AND、OR的优先级和括号，具体实现为依靠栈对条件语句进行分析，之后转换为后缀表达式供后部使用。

UPDATE语句的SET赋值支持四则运算，具体实现与对条件的解析类似，也是依靠栈进行表达式的求值，如果遇到变量则存储并供后端补充内容后进行运算。

SQL的解析鲁棒性很强，在其中具有对于多种错误情况的处理，在分析程序中崩溃会被捕获，并最终将语句的类型置为ERROR供后端处理。对于整个SQL解析模块进行了很细致的测试，对于各种情况都进行了考虑。

## Where的栈式处理

实现更加完备的where，支持括号操作。例如(Table.a>10 and Table.a<15) or (Table.b>10 and Table.b<15)。

where在解析的时候生成逆波兰表达式，存放在栈中，执行查询时，通过栈的push和pop操作求得结果。

## GUI

GUI主要使用Java的swing框架来实现，全部写在me.hqythu.wxydb.ui包中。UIMain类为整个UI程序的入口点，MainWindow类为主界面。主界面上有两个按钮，分别对应新建一个数据库和打开一个已有的数据库文件。这之外页面上主要分为Database Structure，Browse Data，Execute SQL三个Tab页，分别用三个单独的类来实现，然后嵌入到MainWindow里面。

Database Structure用一个树状结构来显示整个数据库的所有表，以及每个表的列的信息。当打开一个数据库之后就可以在这里面看到。

Browse Data有一个Combobox选择栏，可以选择数据库中的任何一个表，然后下面就会显示出这个表中的所有数据。这里由于没有对界面做很好的Block处理，所以对于比较大的数据库而言，选择一项之后可能需要一段的时间才能加载显示出数据，而这之前没有任何提示。

Execute SQL中有一个文本框可以用来输入SQL语句，下方会显示出来执行的结果。对于一般的语句，通常是显示success或者fail。对于select，会和Data的Tab一样的，用一个表格来显示出select出来的数据。

## 单元测试

充分利用Java完善的开发工具链，在开发过程中加入单元测试，使用Java流行的JUnit4单元测试框架。这也是本学期另一门课软件工程所传授的主要思想。

单元测试较大部分用于SQL解析结果的测试，因为SQL是一种语言，具有非常复杂的输入可能，我们通过单元测试发现和修正问题，大大提高了SQL解析的鲁棒性。

单元测试还用于增删查改的检查，直接对比结果，不需要人用眼睛对比输出结果，开发效率大大提高。

我们的单元测试一共含38个测试点，全部可以测试通过（有一个点在单独测试的时候可以通过，在一起测试的时候不通过，是由于主键冲突问题，这个不是功能上的缺陷）。测试的覆盖率为：

类覆盖：56%

方法覆盖：68%

代码行覆盖：73%

其中除了ui包没有进行测试，其他的包如manage，pagefile，object等几个核心的包，都达到了100%的类覆盖率和至少80%的代码行覆盖率。详细的单元测试覆盖率报告附在coverage文件夹中，可以大概index.html查看。

# 实验结果

## 功能实现

实现了课程设计要求的基础功能：

创建数据库、删除数据库、切换数据库、创建表、删除表、列出所有表、列出表的列信息。

插入数据、删除数据、更新数据、查询数据。其中包括主键的检查、null的处理、数据类型的检查。

实现了部分课程设计的拓展功能：

独立完成页式文件系统，GUI界面，多表查询，聚集查询，复杂的where支持。

## 不足之处

最后还是没有完成索引，插入和更新时如果需要检查主键，则为线性复杂度，效率低。

模块阶段划分过细，同时在这个基础之上优化不足，导致多表连接的查询计划一次给出，直接在笛卡尔积的基础之上做查询，而不是先对单个的表查询再做连接。这样无法很好地支持大数据量的表的连接。

SQL语句解析仍然不够完备，鲁棒性有待增强，对语法错误的处理仍然做的不是很好。

GUI还有很多提升空间，如果有更多的时间的话，还可以实现在UI中创建修改表，添加删除数据。而现在UI只能用来查看表结构 和浏览每个表中的数据，这些更丰富的操作只能通过执行SQL语句来实现。

# 小组分工

刘晓鸿：系统管理模块，记录管理模块

王凯：查询解析模块，测试

何钦尧：图形用户界面，测试，开发环境及项目组织结构搭建

# 参考文献和链接

项目Github地址<https://github.com/hqythu/wxyDB>