**MiniSQL 总体设计报告**

作者：07级计算机科学与技术专业

陈矫彦 3071102293

1. **引言**

## 1.1编写目的

为了更好地介绍我们的MiniSQL数据库管理系统，这里先对系统进行一个总体的分析，对其层次的设计进行说明，以便读者对MiniSQL的总体规划有一个了解。

## 1.2项目背景

本次“数据库系统设计”的课程设计由陈根才老师指导，由陈矫彦、郭舟东、聂森、吴世奇共同合作完成。

本课程设计主要运用了本课程的知识和“数据库概念”这门课的知识，结合C++编程技术，通过合作，由四位同学共同完成的。

## 1.3定义

### 1．3．1 专业术语

MiniSQL：微型数据库管理系统。

C++：面向对象编程语言。

SQL：数据库管理系统访问管理语言，和标准定义的SQL语言略有不同。

### 1．3．2 缩写

SQL：Structured Query Language（结构化查询语言）。

## 1.4参考资料

《数据库系统概念》 Abraham Silberschatz

Henry F.Korth

S. Sudarshan

1. **任务概述**

## 2.1功能概述

设计并实现了一个精简型单用户SQL引擎(DBMS)MiniSQL，允许用户通过字符界面输入SQL语句实现表的建立/删除；索引的建立/删除以及记录的插入/删除/查找等最基本的数据库操作。

## 2.2运行环境

Windows XP操作系统上完成测试。

## 2.3具体分析

数据类型：

只要求支持三种基本数据类型：int，char(n)，float，其中char(n)满足 1 <= n <= 255 。

表定义：

表中的每个属性可以指定是否为unique；支持单属性的主键定义。

索引操作：

对于表的主属性自动建立B+树索引，对于声明为unique的属性可以通过SQL语句由用户指定建立/删除B+树索引，也就是说所有的B+树索引都是单属性单值的。

查找记录：

可以表全部查找，也可以通过指定用and 或者 or逻辑操作，连接一个或者两个where条件进行查询，支持等值查询和区间查询。

记录增删：

支持每次一条记录的插入操作；支持每次一条或多条记录的删除操作，可以通过类似查找类似的where条件来选择删除的语句。

**SQL语句说明：**

参见文档“MiniSQL使用文档”。

1. **MiniSQL系统结构设计**

**3.1、总体设计：**

MiniSQL系统的结构很大程度上是参考课程实验样本给定的方案进行设计的，但也做了一定的改进。设计结构图如下：

**Main函数**

Interpreter

API

CatalogManager

IndexManager

RecordManager

BufferManager

DB F

DB File

**3.2、Interpreter 模块：**

这个模块的主要功能是接受用户输入的SQL语句，并检验用户输入的SQL语句及的格式，检验其正确性。同时将符合格式要求的语句转化为内部形式，供主函数使用；而对不符合格式要求的语句，输出错信息，供用户参考。具体的命令解释实现可以参考Interpreter模块设计说明。

**3.3、API模块：**

是整个系统的核心，功能主要是调用CatalogManager提供的信息进行进一步的处理执行语句，并调用RecordManager、IndexManager和CatalogManager提供的相应接口执行各SQL语句，一般SQL语句的执行包括对字典信息的操作和对数据的操作。最终这个模块为Main函数直接执行SQL语句提供接口。

API是RecordManager、IndexManager、CatalogManager和Main函数交互的核心，并且完成数据操作和字典信息操作的结合。因此，虽然API模块所实现的功能十分有限，但是该模块却是整个系统的核心。具体实现可以参考文档“API模块设计说明”。

**3.4、CatalogManager模块：**

负责管理数据库的所有字典信息，包括：

* + 1. 数据库中所有表的定义信息，包括表的名称、表中字段（列）数、主键、定义在该表上的索引。
    2. 表中每个字段的定义信息，包括字段类型、是否唯一、是否有索引。
    3. 数据库中所有索引的定义，包括所属表、索引建立在那个字段上等。
    4. 数据表中的记录条数。

3.4.5、数据库内已建的表、索引的数目。

CatalogManager还提供了访问及操作上述信息的接口，供BufferManager模块和API模块使用。

出于方便实现和MiniSQL字典信息并不庞大的事实，Catalog模块采用直接访问磁盘文件的形式，不通过BufferManager，CatalogManager对象实例中的数据也不要求分块存储。在具体使用CatalogManager时，系统一启动时就读入字典信息，直接形成CatalogManager的全局实例，在系统退出或者commit命令时，将全局实例的内容写回文件。

具体的实现参考文档“CatalogManager设计说明”。

**3.5、RecordManager模块：**

负责管理记录表中的数据。主要功能为记录的插入、删除与查找操作，并对API模块提供相应的接口。其中记录的查找和删除操作能够支持不带条件和带条件（包括等值查找、不等值查找和区间查找）。

具体内容可以参考文档“RecordManager模块设计说明”。

**3.6、Indexmanager模块：**

是程序的索引部分,直接对BufferManager提供的内存索引块操作，主要是构建B＋树，负责实现RecordManager这一模块（提供函数接口）所需要的函数并向API提供接口。功能有：存储\删除记录的索引，查找记录在table表中的相对位置，向索引中插入入一条记录。

**3.7、BufferManager模块：**

负责缓冲区的管理，主要功能有：

* + 1. 根据需要，读取指定的数据到系统缓冲区或将缓冲区中的数据写出到文件；
    2. 实现缓冲区的替换算法，当缓冲区满时选择合适的块进行替换；
    3. 记录缓冲区中各块的状态，如是否被修改过等；

具体可以参考文档“BufferManager模块设计说明”。

1. **出错处理设计**

本系统出错主要在两方面，一个是Interpreter的语法分析，另一个是后台程序时运行出错，比如文件打开失败等等。

Interpreter模块只检查SQL语句结构，若出现错误则重新输入SQL语句，并不关闭程序。其它模块出现的错误，一般在底层出错提示，然后传到主函数，重新执行SQL语句，并不终止程序。

1. **设计分工**

本系统的分工如下：

Interpreter模块………………………………………………郭舟东

CatalogManager模块…………………………………………陈矫彦

RecordManager模块…………………………………………陈矫彦

IndexManager模块…………………………………………聂森，吴世奇

Buffer Manager模块…………………………………………陈矫彦

总体设计…………………………………………………………陈矫彦

总体测试……………………………………………陈矫彦，聂森，吴世奇，郭舟东