

# MiniSQL 总体设计报告

彭子帆、刘圣源、陈宇威

## 1. MiniSQL 设计概述

### 1.1 任务目标

1. 设计并实现一个精简型单用户 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/删除以及表记录的插入/删除/查找。
2. 通过对 MiniSQL 的设计与实现，提高系统编程能力，加深对数据库管理系统事先技术的理解。

### 1.2 系统需求

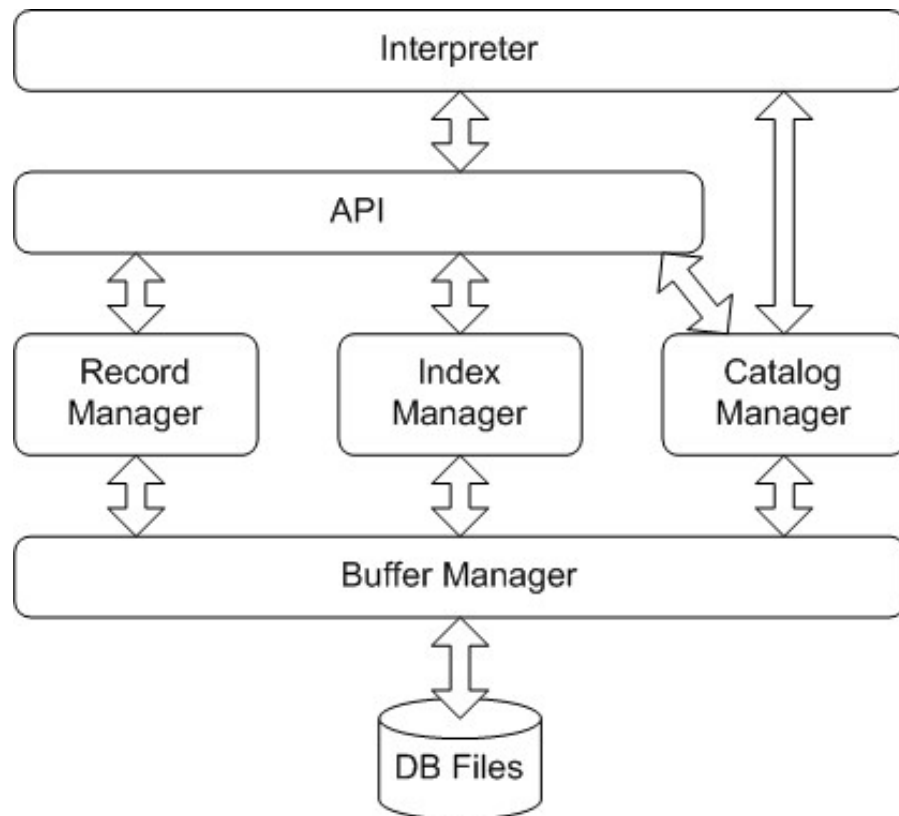
1. 数据类型：要求支持三种基本数据类型：integer, char(n), float。
2. 表定义：一个表可以定义多达 32 个属性，各属性可以指定是否为 unique；支持单属性的主键定义。
3. 索引定义：对于表的主属性自动建立 B+树索引，对于声明为 unique 的属性可以通过 SQL 语句由用户指定建立/删除 B+树索引(因此，所有的 B+树索引都是单属性单值的)。
4. 数据操作：可以通过指定用 and 连接的多个条件进行查询，支持等值查询和区间查询。支持每次一条记录的插入操作；支持每次一条或多条记录的删除操作。

### 1.3 设计背景

根据实验指导中对 MiniSQL 中 6 个模块的分析，我们将其划分为 3 部分，然后每人负责其中一部分。事前约定好可能使用到的接口，并且在编写代码过程中保持相互沟通、交流，最终再将代码拼接，并进行进一步的调试。

## 2. MiniSQL 结构设计

对于 MiniSQL 的结构，我们按照了实验指导中的要求，划分为 6 个模块，具体如下图。



各个模块的具体接口设计等内容将在各模块的详细报告中叙述。

最终的代码，我们将部分模块整合在了一起（例如 CatalogManager、Buffer Manager、Record Manager 都被放进了 RecordManager.h 文件内），并且添加了一些包含全局范围内会用到的接口的文件（例如 Attribute.h 等）。

## 3. 不足与问题

1. 在编写过程中，Index Manager 模块所使用的 B+Tree，在一定程度上参考了搜集到的资料。

2. 程序的 Interpreter 部分存在问题。即 SQL 语句中 where 子句内，涉及到运算符“=”、“<”、“>”等时，要求输入时在运算符前后都要有空格，否则无法正常读取。这个问题可以通过以下思路解决：在 Interpreter 模块接受到一条 SQL 指令的字符串时，

先遍历字符串，在读取到运算符时，在其前后插入空格，然后再对字符串进行解析。

## 4. 小组组成和设计分工

各模块的接口基本都是由共同讨论得到的，并且在编写过程中不断修改、添加，各成员负责了接口的具体实现工作。

彭子帆 3170105860

Buffer Manager

Record Manager

刘圣源 3170104757

Catalog Manager

Index Manager

陈宇威 3170105706

API

Interpreter