

浙江大学



数据库系统课程Project: miniSQL

总体设计报告

小组编号: B

张溢弛 3180103772

张琦 3180103162

聂俊哲 3180103501

一、miniSQL项目简介

1.1 项目简介

- 本项目是2020年春夏学期数据库系统(Database System)课程的课程project, 我们将基于C++设计并实现一个简化的单用户SQL引擎(DBMS)miniSQL。允许用户在控制台界面中通过输入SQL语句实现数据表的建立, 删除, 索引的建立和删除以及表中数据记录的插入, 删除, 查找。
- 通过本项目的设计与实现, 我们将加深对数据库系统的理解, 并提高自身的系统编程能力

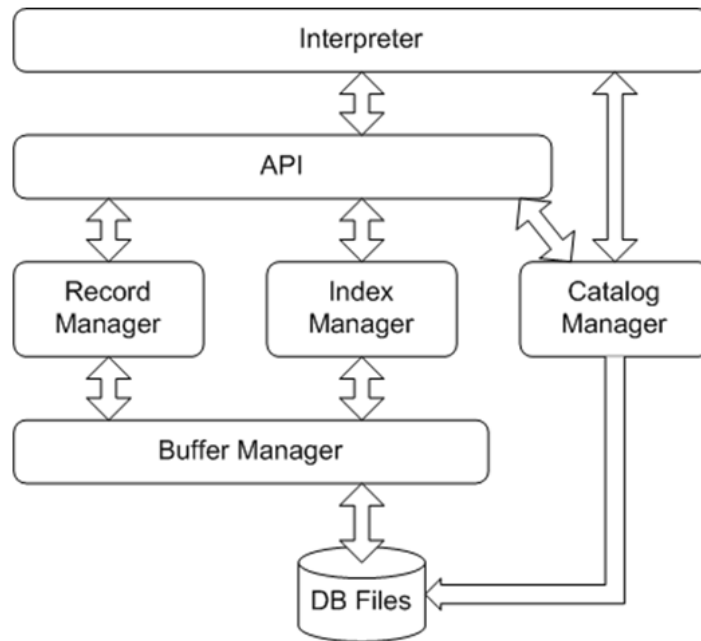
1.2 开发环境

- 本项目的开发环境如下所示
 - 编程语言: C++ 14
 - 集成开发环境(IDE): JetBrains Clion 2020.01版
 - 操作系统: Windows 10家庭版 和 macOS
 - 代码版本托管工具: git, 使用GitHub中的私有代码仓库进行项目管理

二、miniSQL总体设计与功能实现

2.1 系统模块设计

- 本系统的总体结构设计如下图所示



- 每个模块的功能如下所示

- Interpreter 层

- 直接为用户进行交互，实现程序流程的控制，包括启动miniSQL,接受和处理SQL语句并显示命令的执行结果，退出等过程
- 接收并解释用户输入的命令，生成命令的内部数据结构表示，同时检查命令的语法正确性和语义正确性，对正确的命令调用API层提供的函数执行并显示执行结果，对不正确的命令显示错误信息

- API 层

- 为需要执行的SQL语句提供接口，使得Interpreter层可以调用，以Interpreter层解释生成的命令内部表示为输入，根据Catalog Manager提供的信息确定执行规则，并调用Record Manager、Index Manager和Catalog Manager提供的相应接口进行执行，最后返回执行结果给Interpreter模块

- Catalog Manger层

- 管理数据库的所有信息模式，包括
 - 数据库中所有表的定义信息，包括表的名称，字段，主键和索引等信息
 - 表中每个字段的定义信息，包括字段类型，是否唯一
 - 数据库中所有索引的定义
- 提供访问及操作上述信息的接口，供Interpreter和API模块使用
- Catalog模块采用直接访问磁盘文件的形式，不通过Buffer Manager，Catalog中的数据也不要求分块存储，这是为了减小模块之间的耦合性

- Record Manager层

- 管理记录表中数据的数据文件。主要功能为实现数据文件的创建与删除（由表的定义与删除引起）、记录的插入、删除与查找操作，并对外提供相应的接口。其中记录的查找操作要求能够支持不带条件的查找和带一个条件的查找（包括等值查找、不等值查找和区间查找）
- 数据文件由一个或多个数据块组成，块大小应与缓冲区块大小相同。一个块中包含一条至多条记录

- Index Manager层

- 实现B+树索引，包括B+树的创建和删除，等值查找，插入和删除键值等操作
- B+树中节点大小应与缓冲区的块大小相同，B+树的叉数由节点大小与索引键大小计算得到

- Buffer Manager层

- 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
- 实现缓冲区的替换算法，当缓冲区满的时候选择合适的页进行替换
- 记录缓冲区中各页的状态，比如是否被修改过
- 提供缓冲区页的pin功能，锁定缓冲区的页不允许被替换
- DB Files
 - 主要是各类数据库系统中使用的文件

2.2 系统实现的功能

2.2.1 简介

- 本系统主要通过输入一系列SQL语句执行来完成相应的操作和功能，SQL语句支持单行和多行的输入，最后必须用分号结尾作为SQL语句结束的标志
- 所有的关键字都采用小写的形式，如果输入的是大写字母会被自动转换为小写

2.2.2

- 启动系统：需要将源代码进行编译或者打开可执行文件进行启动
- 退出miniSQL系统的语句：`quit;`
- 创建表的语句：关键字为 `create table` 具体的语法格式如下

```
1 create table table_name(  
2     attribution1 date_type1,  
3     attribution2 data_type2 (unique),  
4     .....  
5     primary key(attribution_name)  
6 );
```

- 创建索引的语句：关键字为 `create index`，具体的语法格式如下

```
1 create index index_name on table_name (attribution_name);
```

- 删除表的语句

```
1 drop table table_name;
```

- 删除索引的语句

```
1 drop index index_name;
```

- 选择语句，关键字为 `select`，只支持 `select *` 即显示全部属性

```
1 select * from table_name;  
2 select * from table_name where conditions;
```

- 插入记录语句，关键字为 `insert into`，支持where的条件表达式，语法格式如下

```
1 insert into table_name values (value1, value2, value3.....);
```

- 删除记录语句，关键字为 `delete from`，语法格式如下

```
1 delete from table_name;
2 delete from table_name where conditions;
```

- 执行SQL脚本文件

```
1 exefile file_name
```

2.3 系统的设计细节

- 支持的数据类型是 `int`, `float`, `char(n)` 等三种类型，并且 `char` 类型的长度在1到255之间
- 一张表最多定义32个属性，属性可以设置为 `unique` 和 `primary key`
- 对于一张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语句来建立B+树的索引，所有的索引都是单属性单值的
- 查找记录的过程中可以通过用 `and` 进行多个条件的连接
- 插入只支持单条记录的插入，删除操作支持一条和多条记录的删除
- 为了减小各模块之间的耦合性，Catalog模块采用直接访问磁盘文件的形式，不通过Buffer Manager，Catalog中的数据也不要求分块存储
- 数据文件由一个或多个数据块组成，块大小应与缓冲区块大小相同。一个块中包含一条至多条记录，为简单起见，只要求支持定长记录的存储，且不要求支持记录的跨块存储
- 为提高磁盘I/O操作的效率，缓冲区与文件系统交互的单位是块，块的大小应为文件系统与磁盘交互单位的整数倍，一般可定为4KB或8KB

三、时间安排与规划

- 小组内的初步时间安排如下

时间段	目标任务
5.20-5.31	开会研讨miniSQL的具体要求，进行分工安排和总体设计
6.1-6.15	小组成员完成基本的代码模块编写
6.15-6.20	校内成员之间完成代码的集成与debug
6.20-6.25	优化系统的性能与代码debug，撰写详细设计报告
六月底或七月初	通过视频会议进行系统的验收

- 小组内的分工安排暂定如下
 - 张溢弛：interpreter层，API层，Catalog Manager层
 - 张琦：Buffer Manager层，Record Manager层
 - 聂俊哲：Index Manager层