**MiniSQL总体设计报告**

# MiniSQL系统概述

## 背景

### 编写目的

本学期数据库系统原理的综合实验要求设计并实现一个单用户 SQL 数据库：MiniSQL，要求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
* 表记录的插入/删除/查找

本项目的编写即是为了完成该项任务。

### 项目背景

为全面提高学生创新和实践能力，浙江大学数据库系统原理课程分为课堂教学和综合性实验两部分。综合性实验采取分组形式完成，每 3~4 个学生为 1 组，分别设有组长、主程序员、程序员、测试员、文档员等角色，全面锻炼学生的系统设计与实现能力、系统编程能力、测试能力、组织文档能力、以及团队合作能力，并加深学生对数据库管理系统事先技术的理解。

综合性实验要求设计并实现一个单用户 SQL 数据库：MiniSQL，实验内容贯穿全部重要知识点，通过 2个步骤的阶段性验收，锻炼学生综合运用每个环节所学知识解决实际问题的能力。

## 功能描述

### 表

* 一个表最多可以定义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

### 索引

* 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
* 所有 B+ 树索引都是单属性单值

### 数据类型

支持以下数据类型：

* integer
* char(n) (1 <= n <= 256)
* float

### 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

### SQL语句支持：

* 本项目预计将支持多数常见的标准SQL语句，包括但不仅限于：
* 创建表和删除表
* 创建索引和删除索引
* 选择，插入，删除
* 退出系统
* 执行SQL脚本
* 支持的关键字包括但不仅限于：
* select
* from
* where
* natural join
* join using
* distinct
* as
* order by
* union
* intersect
* except
* avg
* min
* max
* sum
* count
* group by
* having
* some
* all
* exists
* unique
* < = >
* char
* integer
* float
* create table
* primary key
* foreign key
* references
* not null
* insert into values
* delete from
* drop table
* update set
* 每一条SQL可以一行或多行，以分号结尾，要求其中关键字均为小写。
* 每一条SQL语句最多支持一层嵌套。
* 不支持定义视图，事务，函数，触发器等SQL高级特性。

## 运行环境和配置

项目在Linux平台下采用C++编写，通过g++编译，故优先支持Linux平台。但预计同样可在Windows/Mac OS等支持C++编译运行的平台运行。

项目编写过程中采用校内自建Git服务器进行版本管理和团队协作。

项目预计仅使用C++标准库，故无需安装配置第三方库。

## 参考资料

* 数据库系统概念 / （美）Abraham Silberschatz等著；杨冬清等译. —北京：机械工程出版社， 2012.11
* C++ Primer中文版 / （美）Lippman,S.B等著；王刚等译. —北京：电子工程出版社，2013.9

# MiniSQL系统结构设计

## 总体设计

（该部分是本文档的重点，需要将所有可能出现的模块都列举出来，并说明如何实现，实现的说明不需要到代码层面，必须要有软件体系结构图）

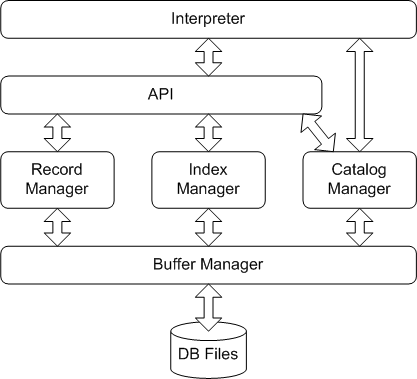


图1 总体设计

## Interpreter 模块

Interpreter模块直接与用户交互，主要实现以下功能：

1. 程序流程控制，即“启动并初始化——‘接收命令、处理命令、显示命令结果’循环——退出”流程。

2. 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。

Interpreter模块负责接受命令，接收到命令后解释器会将命令格式化构建成相应的对象，对象中会保存再进一步处理时需要的数据。之后根据对象的类型将该对象传入API模块中的函数进行相应的处理。如果这个过程中出现SQL语言的语法错误，程序将抛出一个异常，以便提示用户出现了异常。

该模块中将使用以下类来处理SQL语言。

* UseDatabase
* ShowDatabase
* CreateDatabase
* DropDatabase
* ShowTables
* CreateTable
* DropTable
* CreateIndex
* DropIndex
* Selcet
* Insert
* Delete
* Update

## API模块

API模块是整个系统的核心，其主要功能为提供执行SQL语句的接口，供Interpreter层调用。该接口以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。

在这个模块中将为Create,Drop,Insert等在Interpreter中的不同类提供处理接口。根据不同类型的语句，API模块将调用Record Manager、Index Manager和Catalog Manager提供的相应接口进行处理。例如要创建一个表，在API模块的CreateTable函数中，要调用Record Manager中的函数来创建文件，并调用Catalog Manager中的对应函数把表的属性写入文件中。

## CatalogManager模块

Catalog Manager负责管理数据库的所有模式信息，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、是否唯一等。

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。

Catalog Manager还必需提供访问及操作上述信息的接口，供Interpreter和API模块使用。

这一模块中应包含以下类：

* Database
* Table
* Index
* Attribute

在这些类中将提供Create, Drop, Use, Show等函数来处理相应的命令，以便Interpreter和API模块调用；同时提供接口来访问各种模式信息。

## RecordManager模块

（文本）

## Indexmanager模块

（文本）

## BufferManager模块

（文本）

# 测试方案和测试样例

（这部分主要针对上面程序功能来设计测试案例）

# 分组与设计分工

本组成员（姓名 学号）：

董玮豪 3150104577

王大鑫 3150103559

本系统的分工如下：

**（尽可能的详细）**

**对于本模板的说明如下：**

**Word 2010版本；**

**模板内容（主要包含标题的增加和删除，内容的自行定义，对于一级标题表示的是本实验报告必须具有的部分，当然内容等效也可）可以自行修改，文本样式（标题，字体，字号）已经设定好，标题不要超过四级；**

**文本格式，宋体，小四，段落格式可以自行设定。**