数据库工程报告

小组成员：

计 33
2013011343
韩旭

分工：

所有工作均由一人完成。

源程序说明：

源程序在 src 中，gui.py 为图形化操作界面，main 为终端程序，运行脚本命令 bash make.sh 可以重新编译整个数据库工程。

操作方式一：./main 之后可以输入 SQL 语句进行处理。

操作方式二：可以写成 SQL 脚本，然后一次性交给终端处理，./main < SQL 脚本文件

操作方式三：python gui.py 之后将语句输入图形化操作界面处理。

github 地址：https://github.com/THUCSTHanxu13/DataBase_Project

操作注意：

数据库工程中 SQL 解析使用了 Lex 与 Yacc 来进行处理，所以编译需要操作系统中集成 flex 与 bison。

对于 Ubuntu 系统，

使用 sudo apt-get install flex，

以及 sudo apt-get install bison，

可以直接安装；对于 OSX，通常情况下已经安装了 flex 与 bison，所以可以直接运行编译文件。

虽然 Ubuntu 下基于 GUN 的 flex 与 bison 与 OS X 下 BSD 的 flex 与 bison 有区别，但本工程中没有跨平台问题。

对于 Windows 系统，工程可能编译存在问题，所以请避免在 Windows 系统上操作。

代码结构：

页式文件系统为

bufmanager/
Debug/
fileio/
utils/

数据库类型封装

DataType.h

该部分将 int，varchar，null 统一为同一种类型，避免了工程内部的复杂处理。

基本数据表封装

DataStruct.h

内部成员为 table 的 key 与 value，支持这些 key 的 value 读写访问等各种操作。

数据表管理模块

DataFileManager.h

内部实际管理了一个 DataStruct，将上层 Terminal 模块的 insert、update、delete 操作等等划归为底层 DataStruct 的操作。

控制端模块

Terminal.h

内部管理多个数据库以及每个数据库的表单文件，直接接受解析出的命令，并将命令传递到 DataFileManager 进行执行。

语法解析器

yacc.y

进行语法分析的模块，bison -d yacc.y 产生 yacc.tab.c 文件，这是实际工程使用的语法解析代码。

词法解析器

lex.l

进行词法解析的模块，flex lex.l 产生 lex.yy.c 文件，这是实际工程使用的词法解析代码。

SQL 表语句值的基类型

Entity.h

语法树的基类型

Expr.h

语法解析器的基类型

main.h

公用头文件

Header.h

键值查找加速辅助模块

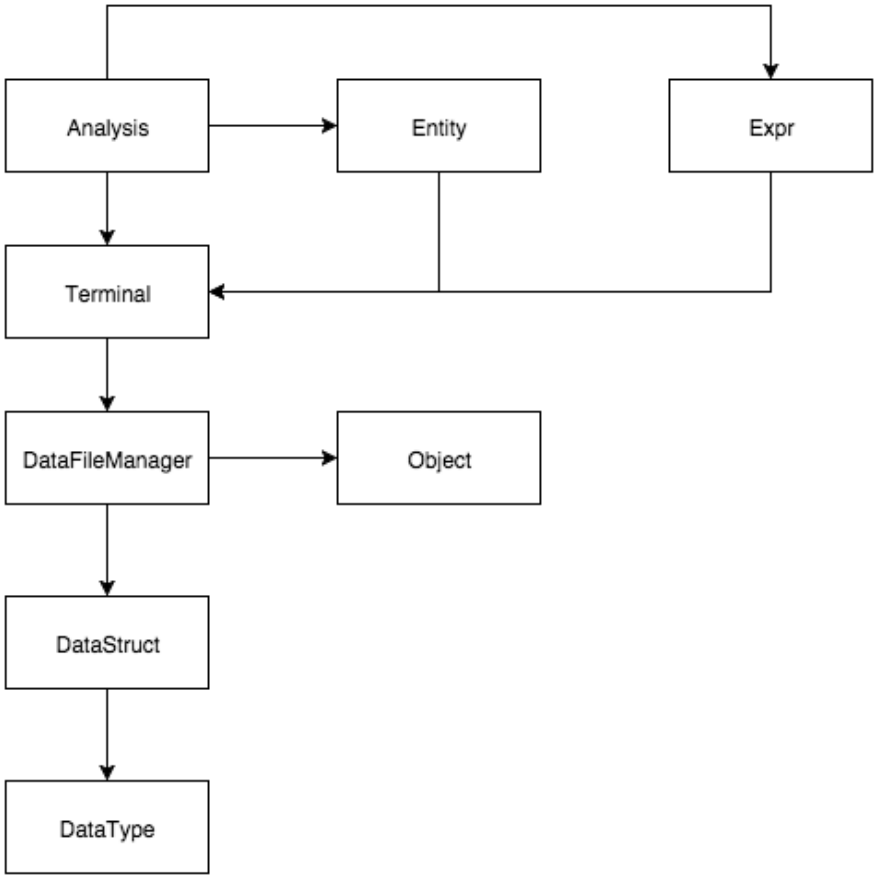
SearchTree.h

系统结构设计：

解析模块解析出结果传递到控制端模块，

控制端模块通过数据表管理模块将上层操作转化为底层操作，通入数据表管理模块，

数据表管理模块通过数据库类型封装将数据传递到页式文件系统，从而读写数据库。



系统功能：

构建数据库
构建数据表单
基本的插入、删除、选择、更新
域完整性约束，建表时的 check 关键字。
外键约束。
模糊匹配，例如" %, _" 等通配符。
支持三个或以上表的连接。
支持聚集查询 AVG, SUM, MIN, MAX。
支持分组聚集查询

支持更多的逻辑运算以及算术运算，字符串连接，包括 +、-、AND、XOR、OR、NOT。

图形化操作界面

设计原理：

基本功能实现：

顺序查找表单的每一项，取出后进行条件判断，满足则进行对应操作。

域完整性约束与外键约束：

每次与表值改变有关的操作均会提前进行检索，如果有冲突则直接拒绝操作。

模糊匹配，例如" %, _" 等通配符：

将"%"与"_"映射到正则表达式的".*"的"."来进行处理，c++11 的基础下可以使用 regex 来处理，如果不是 c++11 可以使用 boost 中的正则表达式库来实现。

支持三个或以上表的连接：

由于表的连接是通过栈而不是多层次的循环遍历来实现的，所以双表，三表以及多表连接没有任何区别。

支持聚集查询 AVG, SUM, MIN, MAX：

在内存中临时存下，操作对象的满足条件的个数，对应值的和，最大最小值，顺序查找一遍之后根据实际需要的聚集操作进行处理，最后得到结果。

分组聚集查询：

用分组的键值作为索引，在内存中临时存下，对应分组的个数，和，最大最小值，顺序查找一遍之后根据实际需要的聚集操作进行处理，最后得到结果。

运算实现：

逻辑运算以及算术运算，字符串连接，包括 +、-、AND、XOR、OR、NOT。封装的 varchar，int，null，所以类型是统一的，通过 flex 与 yacc 进行解析之后可以得到一棵语法树，根据语法树由下至上进行深度优先遍历即可以使得逻辑算术混合运算得到实现。

参考文献：

清华大学出版社 数据库系统设计与原理（第二版） 冯建华等

Lex Yacc 快速入门

W3School SQL 教程