
中国人民大学信息学院

数据库系统原理与实现

实验四 《查询分析》实验报告

Github: <https://github.com/taoyouxian/tinydb> (可查看 Wiki 相关介绍与 FAQ)

小组成员: 张恒、陶友贤、赵旺、修晔良

2017. 12. 06

目录

1 实验内容	3
2 实验要求	3
3 实验分工安排	4
4 实验环境	4
5 实验原理与程序设计	4
5.1 实验原理	4
5.2 具体实现	5
5.2.1.支持的 SQL 语句:	5
5.2.2.语法树:	6
5.2.3.数据结构	7
5.2.4.函数调用	8
6 使用手册	8
6.1 使用手册简单说明.....	8
6.2 相关执行命令行代码.....	9
7 实验结果	9
8 实验总结	9

1 实验内容

实现查询分析功能，具体包括

- 句法分析
- 语法分析
- 语义分析

2 实验要求

实现SQL语句的增删改查操作

实现语句：

- CREATE TABLE
- DROP TABLE
- INSERT
- DELETE
- SELECT

3 实验分工安排

任务名称	人员
系统设计	修晔良、赵旺、张恒、陶友贤
程序实现	修晔良、赵旺
系统测试	赵旺、修晔良
文档编写	赵旺
构图设计	陶友贤、张恒
文档修改	陶友贤、修晔良

4 实验环境

- 操作系统平台:图形界面程序 windows 10/win 8/win 7
- 编程语言:C++、C
- 程序开发环境: flex、bison ,gcc

5 实验原理与程序设计

5.1 实验原理

本实验使用 lex 进行词法分析, lex 由定义段、规则段、用户子例程段构成。

.....定义段.....

%%

.....规则段.....

%%

用户子例程段

-
- 定义段中包括文字块、定义、内部表声明、起始条件和转换。
 - 规则段中包括模式行和 C 代码，当 lex 扫描程序时，它把输入与规则段的模式进行匹配，每次发现一种匹配时就执行与模式相关的 C 代码。
 - 用户子例程段的内容被 lex 逐字拷贝到 C 文件。这一部分通常包括从规则中调用的例程。

同时本实验使用 yacc 来识别完整的语法。yacc 语法具有和 lex 一样的三部

分结构。

- 第一部分定义段处理 yacc 生成的语法分析程序的控制信息
- 第二部分规则段包含语法分析程序的规则
- 第三部分用户子例程将被逐字拷贝到生成的 C 代码中

5.2 具体实现

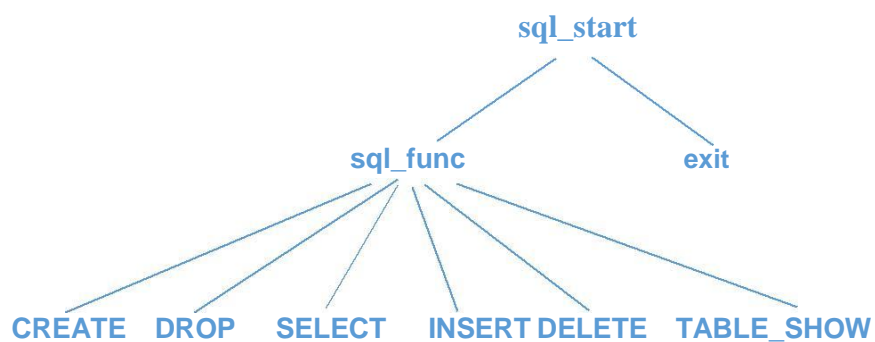
5.2.1.支持的 SQL 语句:

- 新建表: CREATE
TABLE(table)

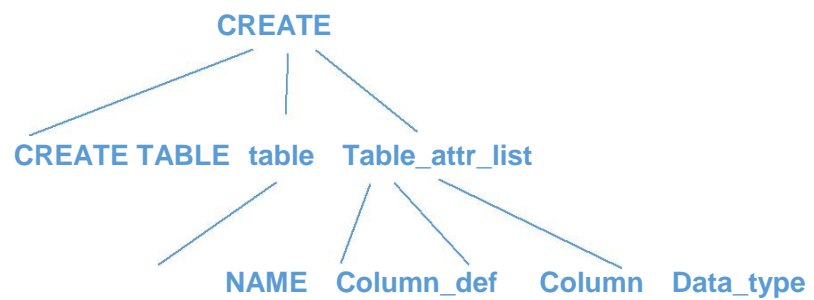
-
- 删除表: DROP TABLE(table)
 - 新建索引: CREATE INDEX table
 - 删除索引: DROP INDEX table
 - 查询: SELECT
 - 插入行: INSERT INTO
 - 删除行: DELETE FROM

5.2.2.语法树:

1) 顶层语法树

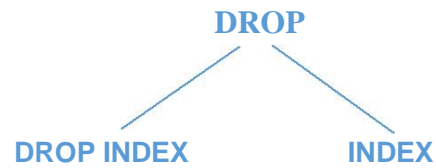


2) CREATE 语句



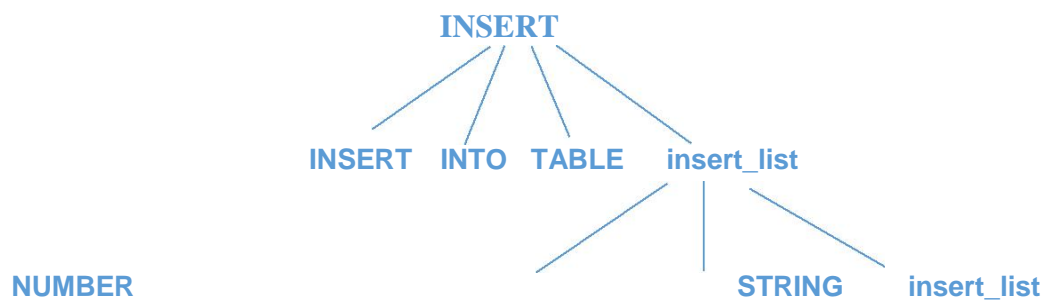
CREATE INDEX 与上类似

3) DROP 语句

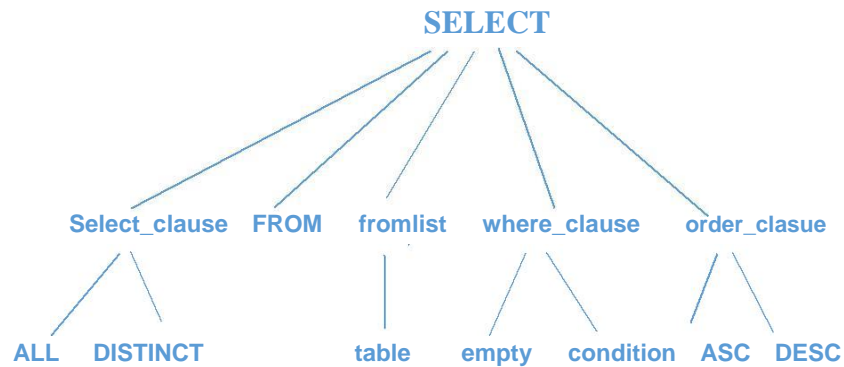


DROP TABLE 与上类似

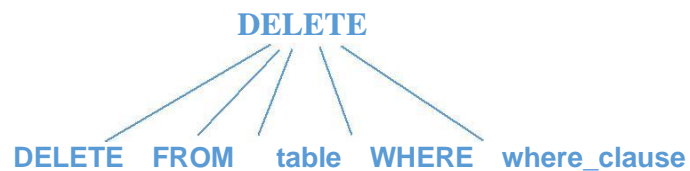
4) INSERT 语句



5) SELECT 语句



6) DELETE 语句



5.2.3.数据结构

用到的数据结构为 `struct condition`,其成员包括:

```
char* data;           // 匹配条件变量
int data_type;        // 匹配条件变量类型(1 整型, 2 字符串型)
int op;               // 匹配算子: =,>,<,>=,<=,<>
char* attrname;       // 待匹配的记录属性
```

5.2.4. 函数调用

- CREATE TABLE:
int CreateTable(char* tbname,int attrnum,attrinfo* attrs,char* errmsg)
- DROP TABLE:
int DropTable(char* tbname,char* errmsg)
- INSERT:
int InsertRecord(char* dbname, char* tbname,char* recordstr,char* errmsg)
- DELETE:
int DeleteRecord(char* tbname,int condcnt,Condition* cond,char* errmsg)
- SELECT:
int SelectRecord(char* tbname,int condcnt,Condition* cond,char* errmsg)

6 使用手册

6.1 使用手册简单说明

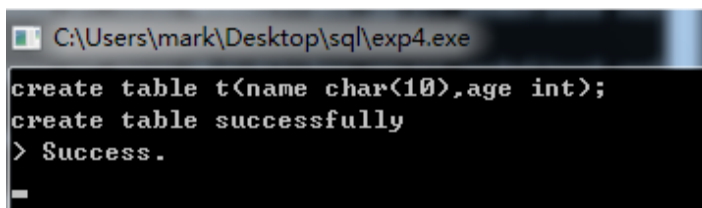
这部分实验在运行的时候主要分为二阶段，第一阶段是通过在 windows 命令行条件下，用 flex 工具编译.l 文件生成词法分析器，在该过程之后会生成 lex.yy.c 文件。然后通过 bison 工具编译.y 文件生成语法解析器，在该语句执行后会生成 y.tab.c 和 y.tab.h 二个文件。然后通过 gcc 编译 lex.yy.c 文件生成一个.exe 的可执行文件。之后我们通过在这个可执行文件中可以解析 SQL 语句。

6.2 相关执行命令行代码

```
Flex sql2.lex           //生成词法解析器
Bison -dy tinySQL.y     //生成语法解析器
Gcc lex.yy.c y.tab.c -o exp4.exe //生成可执行文件
```

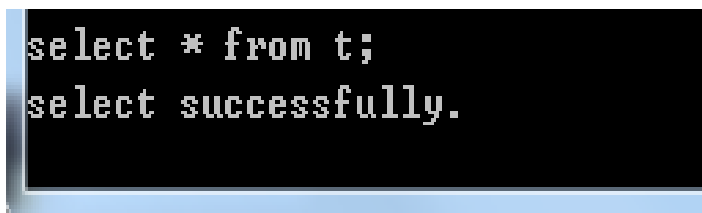
7 实验结果

1. 向数据库中创建一条 sql 语句,



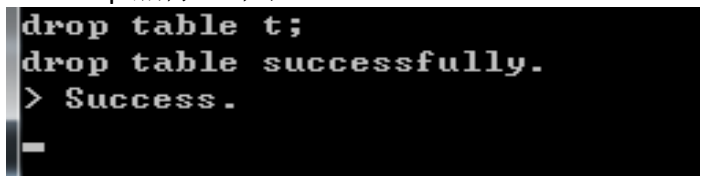
```
C:\Users\mark\Desktop\sql\exp4.exe
create table t(name char(10),age int);
create table successfully
> Success.
_
```

- 3.通过 select 查找一条语句



```
select * from t;
select successfully.
```

2. 通过 drop 删除一个表。



```
drop table t;
drop table successfully.
> Success.
_
```

8 实验总结

由于实验三和实验四完成人员并不相同, 目前尚未做到将实验三的功能加入到本 DBMS 系统中。目前本实验还有很多不足之处, 功能不健全, 代码风格略混乱, 有待完善。

由于底层的语法分析和词法分析并不熟悉, 在我们实现上遇到很大难度, 因此我们也在这部分讨论了很长时间, 后面会进一步同实验一、二、三 想结合。