**Interpreter 模块设计说明**

作者：2007级计算机科学与技术专业

郭舟东 学号：3071102324

Interpreter 模块的主要功能是接受用户输入的SQL语句及其他命令语句，并检验用户输入的SQL语句及其他命令语句的格式，语法和语义的正确性。同时将符合要求的语句转化为内部形式，供API及Catalog模块使用；而对不符合要求的语句，显示其出错信息，供用户参考。

## 由实验要求，程序应须能实现以下各SQL语句及命令语句：

创建表语句

CREATE TABLE 表名 （

列名 类型

列名 类型

…………..

PRIMARY KEY(列名)

）；

**删除表语句**

DROP TABLE 表名；

创建索引语句

CRATE INDEX 索引名 ON 表名（列名）；

删除索引语句

DROP INDEX 索引名；

选择语句

SELECT \* FROM 表名；

或

SELECT \* FROM 表名 WHERE 条件；

插入记录语句

INSERT INTO 表名 VALUES ( 值1，值2，…….,值n)；

删除记录语句

DELETE FORM 表名；

或

DELETE FORM 表名 WHERE 条件；

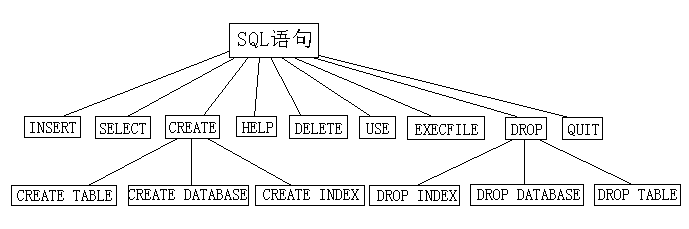
退出MiniSQL系统语句

QUIT;

执行SQL脚本文件语句

EXECFILE 脚本文件名；

## 其语句结构如下：



# 具体实现：

## 首先介绍一下本模块的实现思路：

在实现本模块时，考虑的对语句解析后的各成分输出，将整个模块作为一个类来定义，即Interpreter类。各成分则作为该类的成员。于是就可以实现通过一条SQL语句对一个Interpreter实例进行初始化，然后返回整个实例以进行下面对应的操作。

而对语句的具体解析过程则放在了此类的构造函数中。

## 其具体实现方式是将各语句逐次进行分类，再根据各语句进行分别处理和实现：

### 函数介绍：

整个Interpreter类就包含两个成员函数——int Interpreter::interpreter(string s)和string Interpreter::getWord(string s, int \*tmp)。一个即是核心的构造函数，另一个是构造函数实现中所要调用的子函数。

### 函数具体实现

#### Interpreter类中成员介绍

int firstKey; 第一关键字，即1.2中图中所示的insert、select等SQL语句中的第一个关键字。

int secondKey; 第二关键字，即1.2中图中所示的在第一关键字为create和drop时SQL语句中用来确定操作对象类型的第二个关键字。

string fileName; 操作对象，即SQL语句进行操作的对象名，如建表时的表名，查询中的表名，删除索引时的索引名等。

int condNum; SQL语句中where语句后的条件数，当where后出现语法错误时，该值为-1。

int logic; 当where关键字后的条件数为两个时，连接两个条件的逻辑符，其值为and或者or。

string operater1; where关键字后第一个条件中的逻辑运算符(>, < ,>=, <=, =)

string operater2; where关键字后第二个条件中的逻辑运算符(>, <, >=, <=, =)

string col1; where关键字后第一个条件中涉及的列名。

string col2; where关键字后第二个条件中涉及的列名。

string condition1; where关键字后第一个条件中逻辑运算符后的常量。

string condition2; where关键字后第二个条件中逻辑运算符后的常量。

vector<string> insertValue; insert语句中values后括号中的各个值。（存在一个vector变量中）

vector<string> type; create语句建表时的列属性，按顺序存在一个vector变量中

vector<string> col; create语句建表时的列名。

vector<int> uniq; create语句建表时每个列的unique属性。

string primKey; create语句建表时定义的该表的primary key。

#### 首先是子函数string Interpreter::getWord(string s, int \*tmp)

这个函数有两个参数，首先是字符串变量s，再则是整形变量tmp(以传地址方式传递)。S即为构造一个Interpreter类所需要的一个SQL语句，而tmp则是当前操作的一个指针。

这个函数的作用即为得到tmp指针后的一个关键字，通过判读空格、左右括号、单引号、结束符、换行符等特征字符来决定一个关键字的开始于结束，每次调用此函数后，返回一个关键字，并将tmp指针相应的向后移动。

#### 再则是关键的构造函数 int Interpreter::interpreter(string s)

首先分析SQL语句第一层：

通过子函数getWord（）得到一个关键字后于预设值进行匹配，确定firstKey的值，即确定SQL的功能。

如果firstKey是CREATE或DROP则需要通过第二关键字的对应匹配来确定secondKey的值。

在确定了SQL语句的具体功能后，通过逐字解析，截取判断，再截取再判断的过程，逐一比较相应语句的语法，储存必要的信息，最后得以完成SQL语句的解析。