Interpreter 模块设计说明

作者：2016 级软件工程专业
傅诤哲 程浩然 叶慕祈 石磊 钱程

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。内部形式的语句将极为方便被 API 识别并执行相应的语句。

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;
```

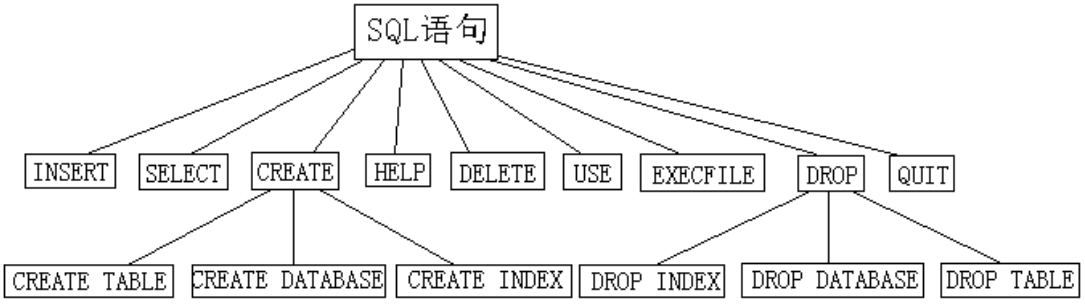
退出 MiniSQL 系统语句

```
QUIT;
```

执行 SQL 脚本文件语句

EXECFILE 脚本文件名；

其语句结构如下：



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：

第一步：根据语句中的第一个关键字进行分类，可分为：1.CREATE; 2. DROP; 3.SELECT; 4.DELETE; 5.INSERT; 6.USE; 7.EXECFILE; 8.HELP; 9.QUIT。

第二步：可将 CREATE 类型进行进一步的分类，可分为：1.CREATE DATABASE; 2.CREATE TABLE; 3.CREATE INDEX。相似地，可将 DROP 类型分类如下：

1.DROP DATABASE; 2.DROP TABLE; 3.DROP INDEX。

第三步：对各语句进行分别处理，并对各类型语句进行标号。其内部数据形式格式如下：语句编号＋语句信息。

创建数据库语句

CREATE DATABASE 数据库名；

只须验证数据库名是否合法，并将有效数据库名转化为内部数据形式。

创建表语句

```
CREATE TABLE 表名 （
    列名 类型
    列名 类型
    ……..
    PRIMARY KEY(列名)
    ）;
```

须验证表名是否合法，且各列名及列的数据类型都为有效，即可将表名，各列名及数据类型和列属性（如唯一或主键定义等）转化为内部数据形式。

创建索引语句

CREATE INDEX 索引名 ON 表名（列名）；

须验证索引名，表名及表的列名是否合法，即可将索引名，表名及列名转化为内部数据形式。

选择语句

SELECT * FROM 表名； SELECT 列 1，列 2，….列 n FROM 表名；

须验证表名及各列名是否有效，即可将表名及各列名转化为内部数据形式。

SELECT * FROM 表名 WHERE 条件； SELECT 列 1，列 2，…列 n FROM 表名 WHERE 条件

须验证表名及各列名是否有效，同时条件符合要求（如列名合法，比较符有效等）即可将表名及各列名和条件转化为内部数据形式。

插入记录语句

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

须验证表名及各属性值有效，即可将表名及各属性值转化为内部数据形式。

删除记录语句

DELETE FORM 表名；

须验证表名是否有效，即可将表名转化为内部数据形式。

DELETE FORM 表名 WHERE 条件；

须验证表名及 WHERE 条件符合要求，即可将表名及 WHERE 条件转化为内部数据形式。

退出 MiniSQL 系统语句

QUIT;

只需指定语句类型号即可。

执行 SQL 脚本文件语句

EXECFILE 脚本文件名；

须验证脚本文件名为合法文件名，即可将脚本文件名转化为内部数据形式。

删除索引语句

DROP INDEX 索引名；

须验证索引名存在于指定的数据库内，即可将索引名转化为内部数据形式。

删除表语句

DROP TALBE 表名；

须验证表名存在于指定的数据库内，即可将表名转化为内部数据形式。

删除数据库语句

DROP DATABASE 数据库名；

须验证数据库名为合法数据库名，即可将数据库名转化为内部数据形式。

使用数据库语句

USE 数据库名；

须验证数据库名的合法性，即可将数据库名转化为内部数据形式。

显示帮助信息语句

HELP;

只需指定语句类型即可。

其主要函数的功能描述如下：

获取用户输入，并对输入作有效性检查，若正确，返回语句的内部表示形式

1． CString Interpreter(CString statement);

读取用户输入

2． CString read_input();

验证 create 语句是否有效

3． CString create_clause(CString SQL,int start);

验证 create database 语句是否有效

4． CString create_database(CString SQL,int start);

验证 create table 语句是否有效

5． CString create_table(CString SQL,int start);

验证 create index 语句是否有效

6． CString create_index(CString SQL,int start);

验证 drop 语句是否有效

7． CString drop_clause(CString SQL,int start);

验证 drop database 语句是否有效

8．CString drop_database(CString SQL,int start);

验证 drop table 语句是否有效

9．CString drop_table(CString SQl,int start);

验证 drop index 语句是否有效

10．CString drop_index(CString SQL,int start);

验证 select 语句是否有效

11．CString select_clause(CString SQL,int start);

验证 insert 语句是否有效

12．CString insert_clause(CString SQL,int start);

验证 insert into values 语句是否有效

13．CString insert_into_values(CString SQL,int start,CString sql);

验证 delete 语句是否有效

14．CString delete_clause(CString SQL,int start);

验证 delete from where 语句是否有效

15．CString delete_from_where(CString SQL,int start,CString sql);

将表达式转化为内部形式

16．CString get_expression(CString temp,CString sql);

获取表达式组的每个表达式

17．CString get_each(CString T,CString sql,CString condition);

验证 use 语句是否有效

18．CString use_clause(CString SQL,int start);

验证 execfile 语句是否有效

19．CString execfile_clause(CString SQL,int start);

验证 quit 语句是否有效

20．CString quit_clause(CString SQL,int start);

通过 Interpreter 模块的处理，可将由用户输入的 SQL 语句及命令语句转化为初步的内部数据形式，保证了 SQL 语句及命令语句的格式，语法及部分语义的正确性。

附 interpreter 翻译语法（基本）

create database name;	00name
create table name(a int, b float, c char(1), primary key(a));	01name,a + 0,b - 0,c 1 0,a #, 这里最后的逗号没多打（下面也一样
create index name on table(attr);	02name table attr
drop database name;	10name
drop table name;	11name
drop index name;	12name
select * from table;	20*,table
select title, name from table;	20title-name,table
select * from table where a = 10 and b >= 20;	21*,table,a = 10, b >= 20,
insert into student values (‘12345678’,’wy’,22,’M’);	30name,12345678,wy,22,m,
delete from name;	40name
delete from name where a = 10	41name,a = 10
execfile filename;	50filename
help;	60
quit;	70