

Interpreter

蔡灿宇

主要功能

- 直接与用户交互，处理用户输入信息：
 - 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。
 - 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。

接口说明

API:

```
void Query();           //用于读取指令
void Choice();          //根据读入的指令对应相应接口
```

- 本模块提供接口给main函数，进行用户输入的处理，例如：

```
int main()
{
    bool execfile = false;
    while (i.flag == true)
    {
        i.Query();
        i.Choice();
    }
    system("pause");
    return 0;
}
```

- **Query()**
用于读取用户输入的指令，将其合成为整条字符串。
- **Choice()**
根据读入的指令判断对应的指令类型，跳转到相应的接口分析指令。

模块说明

- 类定义

```
class Interpreter
{
private:
```

```

std::string query;           //查询语句
public:
    bool flag = true;        //判断是否quit
    void Query();             //用于读取指令
    void Choice();            //根据读入的指令对应相应接口
    void Create_Table();
    void Create_Index();
    void Drop_Table();
    void Drop_Index();
    void Select();
    void Insert();
    void Delete();
    void Quit();
    void Execfile();

    int Next(int i);          //返回下一个单词的末尾位置
    int Next(int i, string str); //返回str字符串里面下一个单词的末尾位置
};

```

- 工作原理：
 - 通过Query()处理输入的指令，再通过Choice()初步分析指令类型并跳转到其他函数。
 - 通过Next()函数逐个解析指令的单词，准确保存任何有用信息。
 - 其他函数解析指令，将指令分别保存为attribute、condition等信息，传递到其他模块的接口完成操作。

- 函数说明：

- `Next(int i);`
`Next(int i, string str);`

通过指针判断当前位置，并识别空格，分号，\t等符号，进行对单词的切分。

- 样例说明：

create table

例子：

```

create table student (
    sno char(8),
    sname char(16) unique,
    sage int,
    sgender char(1),
    primary key (sno)
);

```

- 注意：
 - 括号内不能有空格
 - primary key结尾不能有逗号
 - 所有符号皆为英文格式
- 调用接口：

- 调用catalog.createTable(), 进行表格的创建
- 调用catlog.reload(), 进行文件的更新

create index

例子:

```
create index stunameidx on student (sname);
```

- 注意:
 - 括号内不能有空格
 - 每个字段都由空格隔开, 上述句子共有六个字段
 - 所有符号皆为英文格式
- 调用接口:
 - 调用catalog.createIndex(), 进行表格的创建
 - 调用catlog.reload(), 进行文件的更新

drop table

例子:

```
drop table student;
```

- 调用接口:
 - 调用catalog.dropTable(), 进行表格的删除
 - 调用catlog.reload(), 进行文件的更新

drop index

例子:

```
drop index stunameidx;
```

- 调用接口:
 - 调用catalog.dropTable(), 进行表格的删除
 - 调用catlog.reload(), 进行文件的更新

select * from

例子:

```
select * from student;  
select * from student where sno = '88888888';  
select * from student where sage > 20 and sgender = 'F';
```

- 注意:

- 每个字段都要以空格隔开，包括等于号=
- 单引号内不能有空格
- 所有符号皆为英文格式
- 调用接口：
 - 调用record.select()

insert into

例子：

```
insert into student values ('12345678','y',22,'M');
```

- 注意：
 - 每个字段都要以空格隔开
 - 括号、单引号内不允许有空格
 - 所有符号皆为英文格式
- 调用接口：
 - 调用record.insert()

delete from

例子：

```
delete from student;  
delete from student where sno = '88888888';
```

- 注意：
 - 每个字段都要以空格隔开
 - 单引号内不允许有空格
 - 所有符号皆为英文格式
- 调用接口：
 - 调用record.deleteR()

quit;与execfile filename;

例子：

```
quit;  
execfile filename;
```

- 注意：
 - 记得末尾的分号