

Модельный интерпретатор SQL

Архитектура проекта

Проект состоит из 5 папок с файлами: `sql`, `lib`, `executor`, `models`, `parser`.

Связующим звеном всего проекта является папка sql, в которой содержится файл SQL.cpp, в котором есть класс SQL с private данными SQLExecutor и SQLParser. А также метод run, который запускает весь процесс.

В папке `lib` хранится файл `Table.cpp` для работы с таблицами через C++.

В папке `executor` содержится файл `SQLExecutor.cpp`, который нужен для выполнения парсера и папка `poliz`, с помощью которой мы вычисляем все арифметические выражения.

В папке `models` хранятся все необходимые структуры данных для парсинга запросов.

В папке parser содержится файл *SQLParser.cpp* , который выполняем парсер всех запросов SQL , заполняет для каждого запроса нужную структуру данных из models и передаёт эту структуру в *SOLExecutore.cpp*.

Программа запускается из main.cpp. Создаётся объект класса SQL и считывается строка из терминала. Входная строка передается в метод SQL::run , происходит парсинг выражения и само выполнение запроса.

Синтаксис запросов

```
SQL-предложение> ::= <SELECT-предложение> | <INSERT-предложение> |  
                    <UPDATE-предложение> | <DELETE-предложение> |  
                    <CREATE-предложение> | <DROP-предложение>
```

**<SELECT-предложение> ::= SELECT <список полей> FROM <имя таблицы> <WHERE-
кlausуза>**

$$\langle \text{список полей} \rangle ::= \langle \text{имя поля} \rangle \{ , \langle \text{имя поля} \rangle \}^*$$

<имя таблицы> ::= <имя>

<имя поля> ::= <имя>

<имя>::= <идентификатор языка Си>

<INSERT-предложение> ::= INSERT INTO <имя таблицы> (<значение поля> { , <значение

поля> })

<значение поля> ::= <строка> | <длинное целое>

<строка> ::= '<символ> {<символ>}'

<символ> ::= <любой, представляемый в компьютере символ, кроме апострофа '>

<UPDATE-предложение> ::= UPDATE <имя таблицы> SET <имя поля> = <выражение>
<WHERE-клауза>

<DELETE-предложение> ::= DELETE FROM <имя таблицы> <WHERE-клауза>

<CREATE-предложение> ::= CREATE TABLE <имя таблицы> (<список описаний полей>)

<список описаний полей> ::= <описание поля> { , <описание поля> }

<описание поля> ::= <имя поля> <тип поля>

<тип поля> ::= TEXT (<целое без знака>) | LONG

<DROP-предложение> ::= DROP TABLE <имя таблицы>

<WHERE-клауза> ::= WHERE <имя поля типа TEXT> [NOT] LIKE <строка-образец> |
<выражение> [NOT] IN (<список констант>) | WHERE <логическое
выражение> | WHERE ALL

<строка-образец> ::= <строка>

<выражение> ::= <Long-выражение> | <Text-выражение>

<список констант> ::= <строка> { , <строка> } | <длинное целое> { , <длинное целое> }

<Long-выражение> ::= <Long-слагаемое> { <+|-> <Long-слагаемое> }

<+|-> ::= + | -

<Long-слагаемое> ::= <Long-множитель> { <*/|/%> <Long-множитель> }

<*/|/%> ::= * | / | %

<Long-множитель> ::= <Long-величина> | (<Long-выражение>)

<Long-величина> ::= <имя поля типа LONG> | <длинное целое>

<Text-выражение> ::= <имя поля типа TEXT> | <строка>

<логическое выражение> ::= <логическое слагаемое> { OR <логическое слагаемое> }

<логическое слагаемое> ::= <логический множитель> { AND <логический множитель> }

<логический множитель> ::= NOT <логический множитель> | (<логическое выражение>)
| (<отношение>)

<отношение> ::= <Text-отношение> | <Long-отношение>

<Text-отношение> ::= <Text-выражение> <операция сравнения> <Text-выражение>

$\langle \text{Long-отношение} \rangle ::= \langle \text{Long-выражение} \rangle \langle \text{операция сравнения} \rangle \langle \text{Long-выражение} \rangle$

$\langle \text{операция сравнения} \rangle ::= = \mid > \mid < \mid >= \mid <= \mid !=$