

## **Построение AST-дерева для SQL**

### **Использование ANTLR4. Язык программирования - python.**

ANTLR4 используется для формализации грамматики и генерации кода для лексического и синтаксического анализа.

На выходе сгенерированного ANTLR парсера получаем дерево разбора, узлы которого соответствуют правилам описанной грамматики.

Следующий этап - преобразование дерева разбора в абстрактное синтаксическое дерево (AST). Каждый узел такого дерева представлен в виде конструкции программы без привязки к синтаксису описанного языка.

Для преобразования дерева разбора ANTLR в AST используется библиотека antlr-ast. Принцип её использования заключается в следующем: создаются классы для представления узлов AST, затем определяются функции для преобразования основных узлов дерева разбора в узлы AST, а затем запускается процесс преобразования дерева. Библиотека позволяет описывать правила преобразования узлов дерева разбора в узлы AST с помощью настройки массива соответствий между их атрибутами.

В результате применения преобразований получено AST.

#### **В структуру проекта входит:**

- грамматика, описанная в файле `sql.g4`
- классы, описанные в файле `sqlast.py`
- файл с sql-запросами `sql_stmt.sql`
- класс, включающий функцию чтения файлов `utils.py`
- класс `main.py` - точка входа

#### **Основные классы дерева:**

- `class RootQueryNode: ['queries']`
- `class QueryNode: ['query_statements']`
- `class FactoredStatementNode: ['select_parts', 'operators']`

- class SelectNode: ['columns', 'tables', 'where', 'group\_by', 'having']
- class ExprNode: ['database\_name', 'table\_name', 'column\_name', 'select\_stmt', 'expr', 'literal\_value']

**Реализованы следующие sql-запросы:**

- insert;
- delete;
- update;
- select;

Также включает синтаксис sql-выражений: имена переменных, операторы сравнения, логические операторы, арифметические операции.

! Примеры точек входа-выхода можно найти в файле Examples.pdf