

Метод построения дерева синтаксического анализа языка SQL на основе графических примитивов

Студент: Сукочева Алис

Руководитель: Гаврилова Юлия Михайловна

Консультант: Ступников Андрей Алексеевич

Москва, 2022

Цель и задачи работы

Целью является разработка метода построения дерева синтаксического анализа языка SQL на основе графических примитивов.

Задачи:

- сравнить существующие решения;
- описать и разработать грамматику SQL-запроса для ограниченной предметной области;
- создать ПО, реализующее разработанный метод;
- исследовать зависимость времени построения дерева синтаксического анализа от сложности запроса.

Процесс создания отчета в хранилище данных

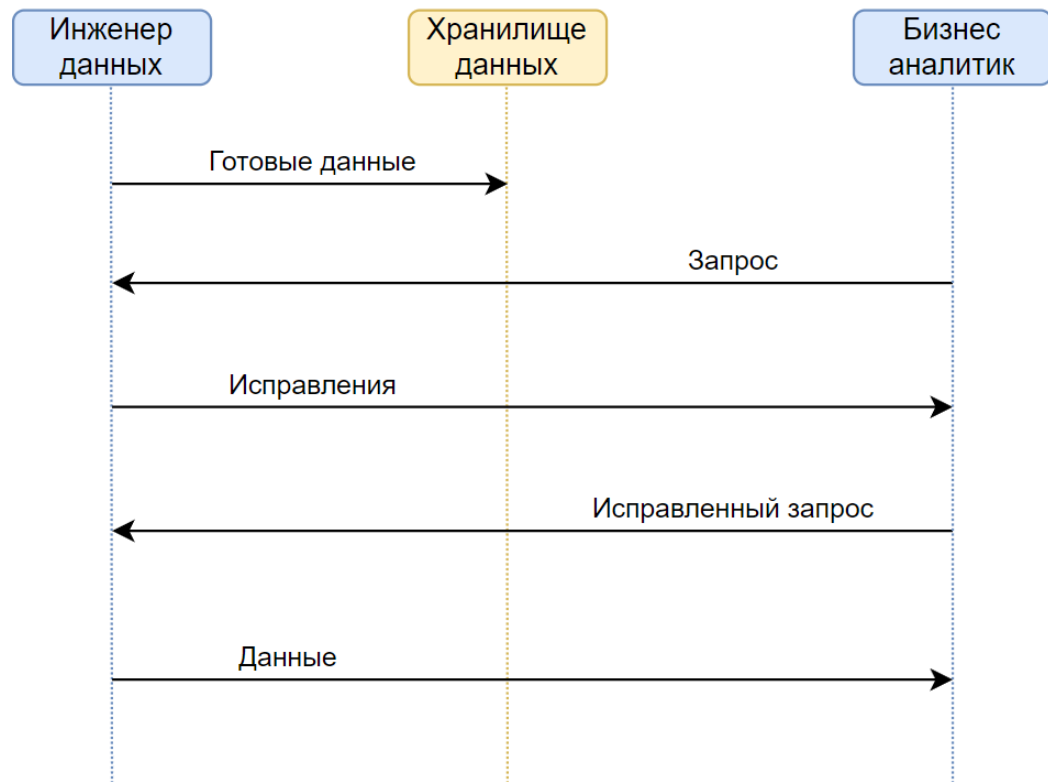


Диаграмма последовательности
без использования графического ETL
инструмента

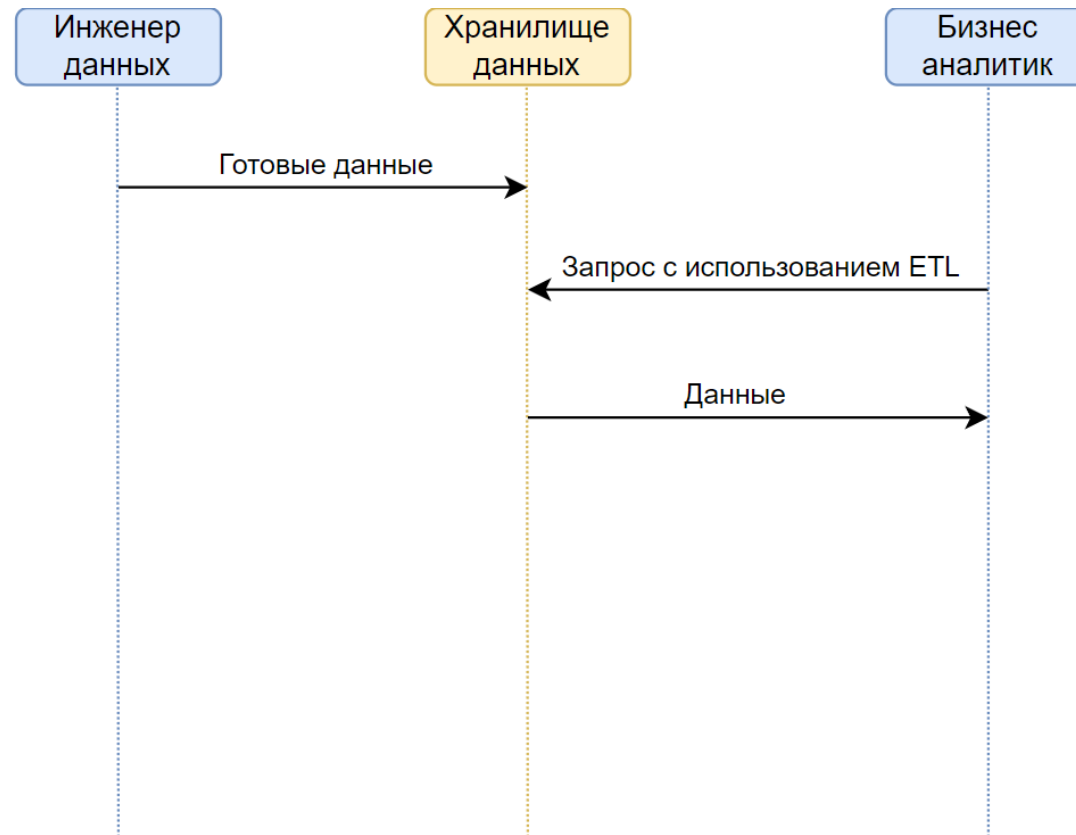
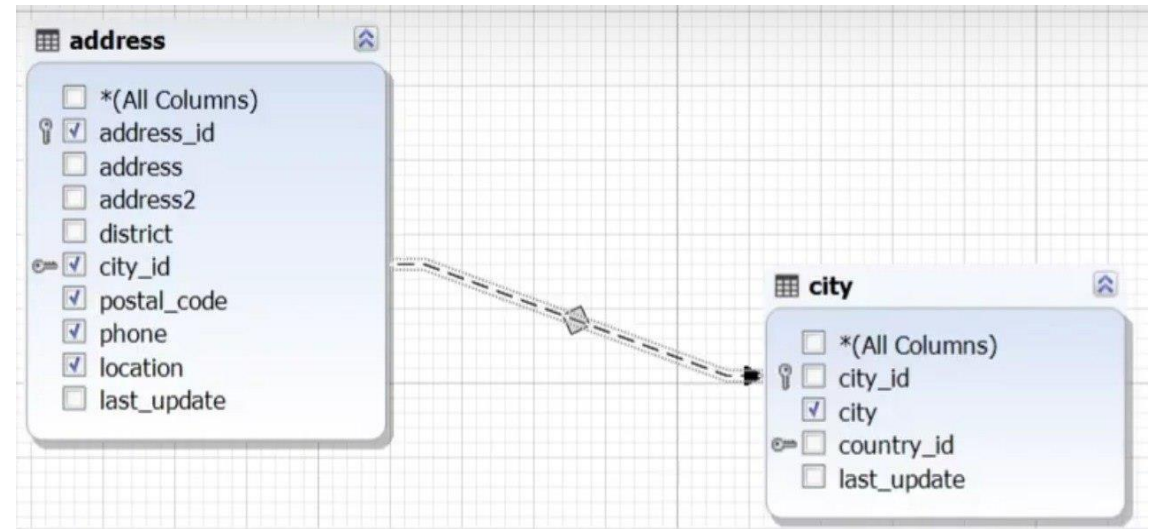


Диаграмма последовательности
с использованием графического ETL
инструмента

Использование графических ЕТL-инструментов

```
SELECT address.address_id,  
       address.city_id,  
       address.postal_code,  
       address.phone,  
       address.location,  
       City.city  
FROM address  
     INNER JOIN city  
           ON address.city_id = city.city_id
```



Существующие решения:

- SAS Data Integration Studio;
- Informatica Power Center;
- DBForge Studio;
- FlySpeed SQL Query.

Типы запросов

Очень
простой

```
SELECT age  
FROM cats
```

Простой

```
SELECT age  
FROM cats  
WHERE  
    breed IN ('Sphinx', 'Persian', 'MaineCoon')  
    AND age >= 1 AND age < 5
```

Нормальный

```
SELECT owners.name, cats.name, breed, cats.age  
FROM cats  
INNER JOIN owners  
    ON owners.cat_name = cats.name  
WHERE breed IN ('sphinx', 'Persian', 'MaineCoon')  
    AND cats.age >= 1  
    AND cats.age < 5  
    AND cats.eyes = 'green'  
    AND cats.name LIKE '%Barsik%'  
    AND owners.name IN ('Alice', 'Katya')
```

Сложный

```
SELECT owners.name AS own, count(owners.name) AS con,  
    cats.name AS cn, breed AS b, cats.age AS ca,  
    dogs.eyes AS de, cats.eyes AS ce  
FROM  
(  
    SELECT *  
    FROM cats  
)  
INNER JOIN owners  
    ON owners.cat_name = cats.name  
WHERE breed in ('sphinx', 'Persian', 'Maine')  
    AND cats.age >= 1  
    AND cats.age < 5  
    AND cats.eyes = 'green'  
    AND cats.name LIKE '%Barsik%'  
    AND owners.name in ('Alice', 'Katya')  
GROUP BY owners.name  
HAVING owners.age >= 5  
    AND owners.age < 18  
ORDER BY count(owners.name) DESC
```

Метод выполнения запроса к реляционной БД

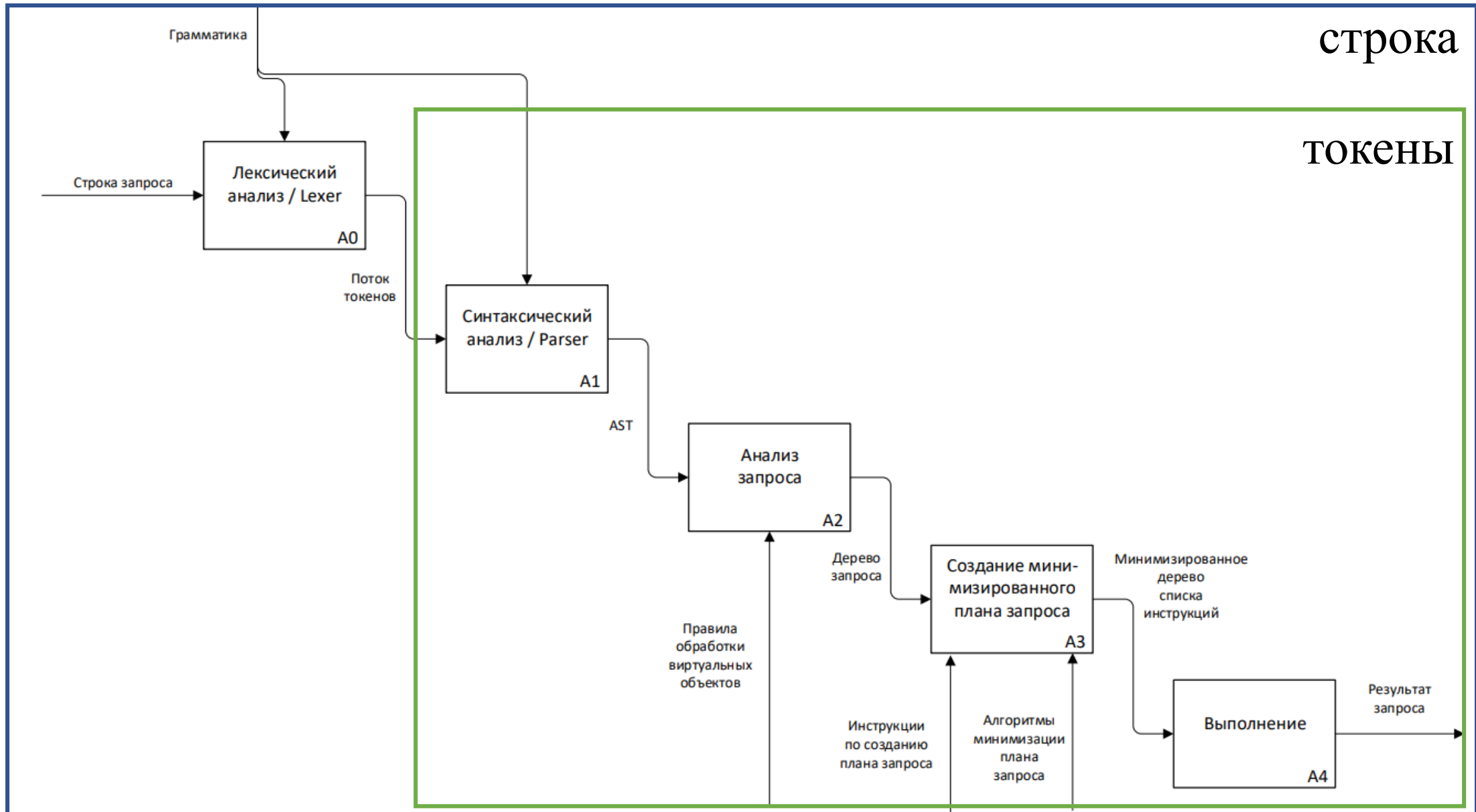


Диаграмма предлагаемого метода

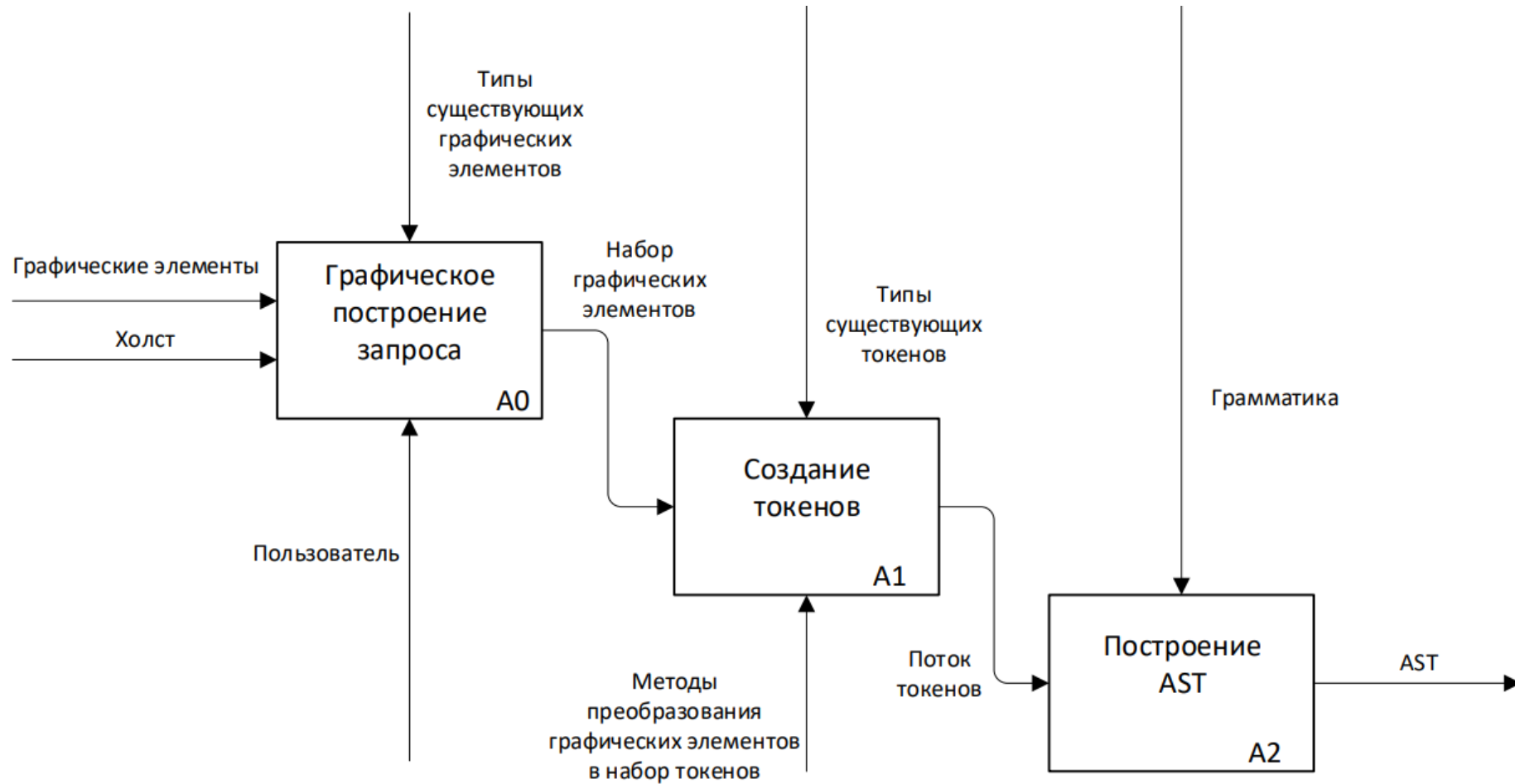
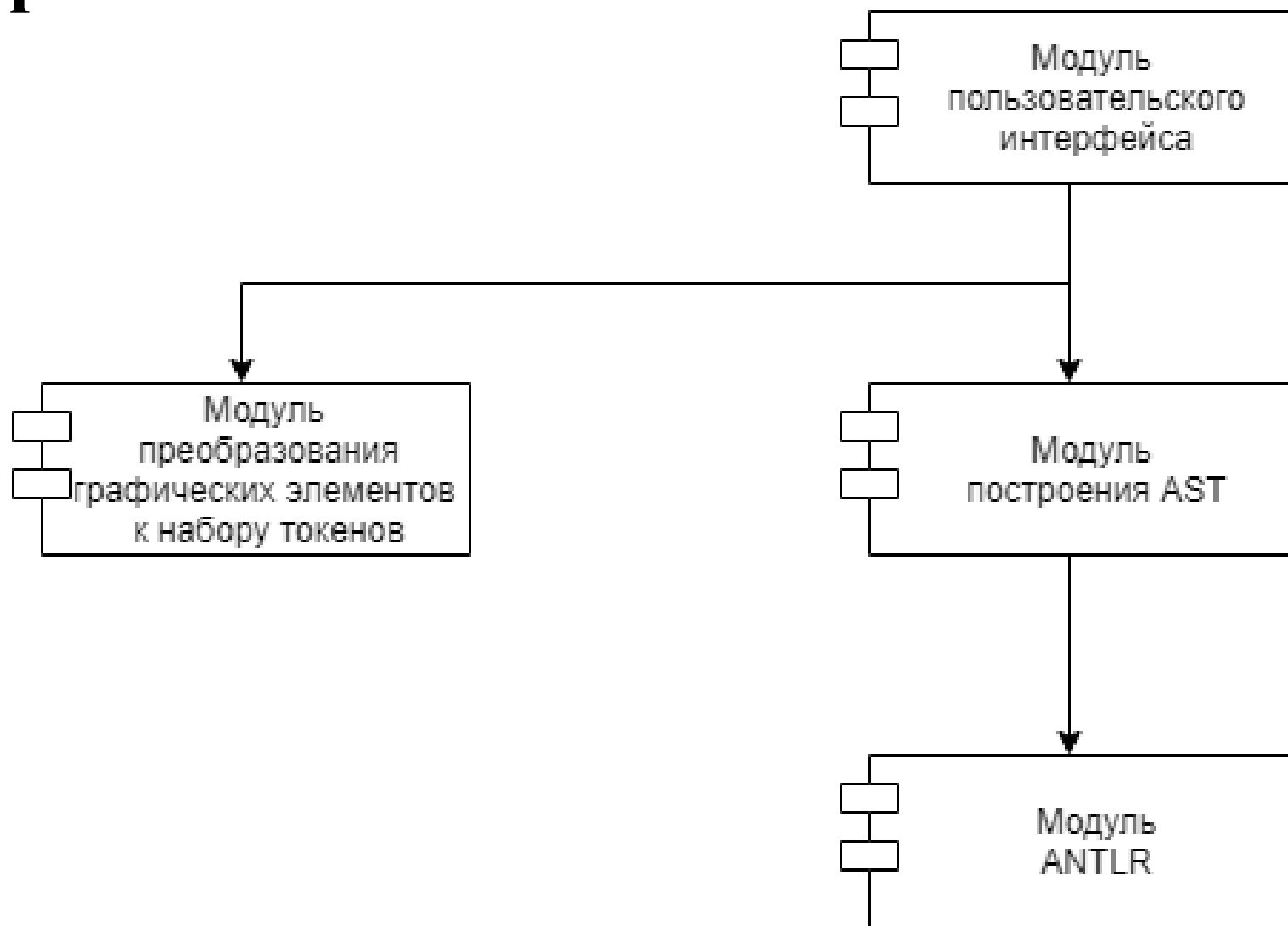


Схема программного обеспечения



Соотношение разработанной грамматики и спецификации

Спецификация СУБД 602SQL Open Server

```
query_specification ::=  
    SELECT [ ALL | DISTINCT ] expression  
        [ rename ] { , expression [ rename ] }  
    FROM generalized table { , generalized table }  
    [ WHERE where_condition ]  
    [ GROUP BY expression_group { , expression_group } ]  
    [ HAVING having_condition ]
```

Разработанная грамматика

```
query_specification :  
    SELECT (distinct)? qexpr (COMMA qexpr)*  
    FROM generalized_table_specification  
    ( WHERE where_condition )?  
    ( GROUP BY expression_group )?  
    ( HAVING having_condition )?  
    ;
```

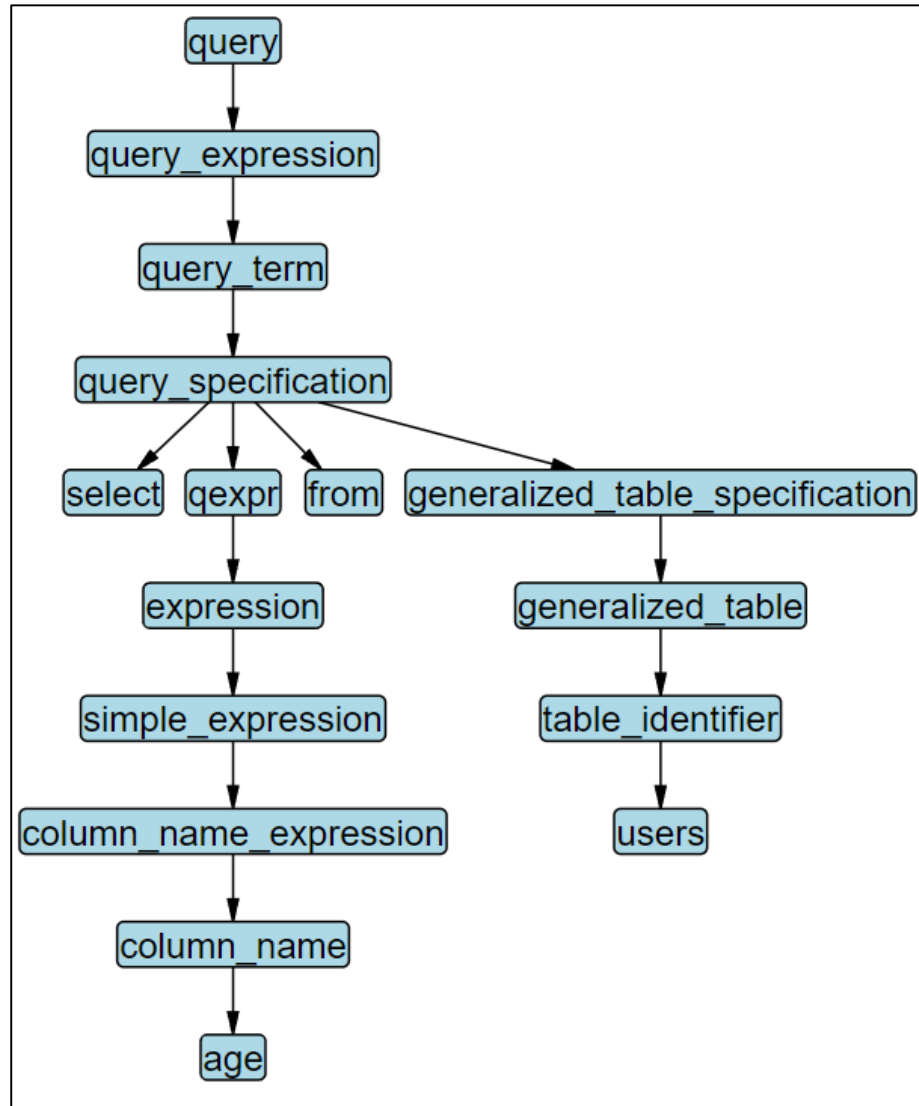
Правила разработанной грамматики

+ select	+ aggregate_function	- update
+ from	+ character_string	- delete
+ where	+ binary_string	- insert
+ group by	+ rename	- create
+ having	+ int_number	
+ limit	+ float_number	
+ offset	+ join_type	
+ order by	+ ...	

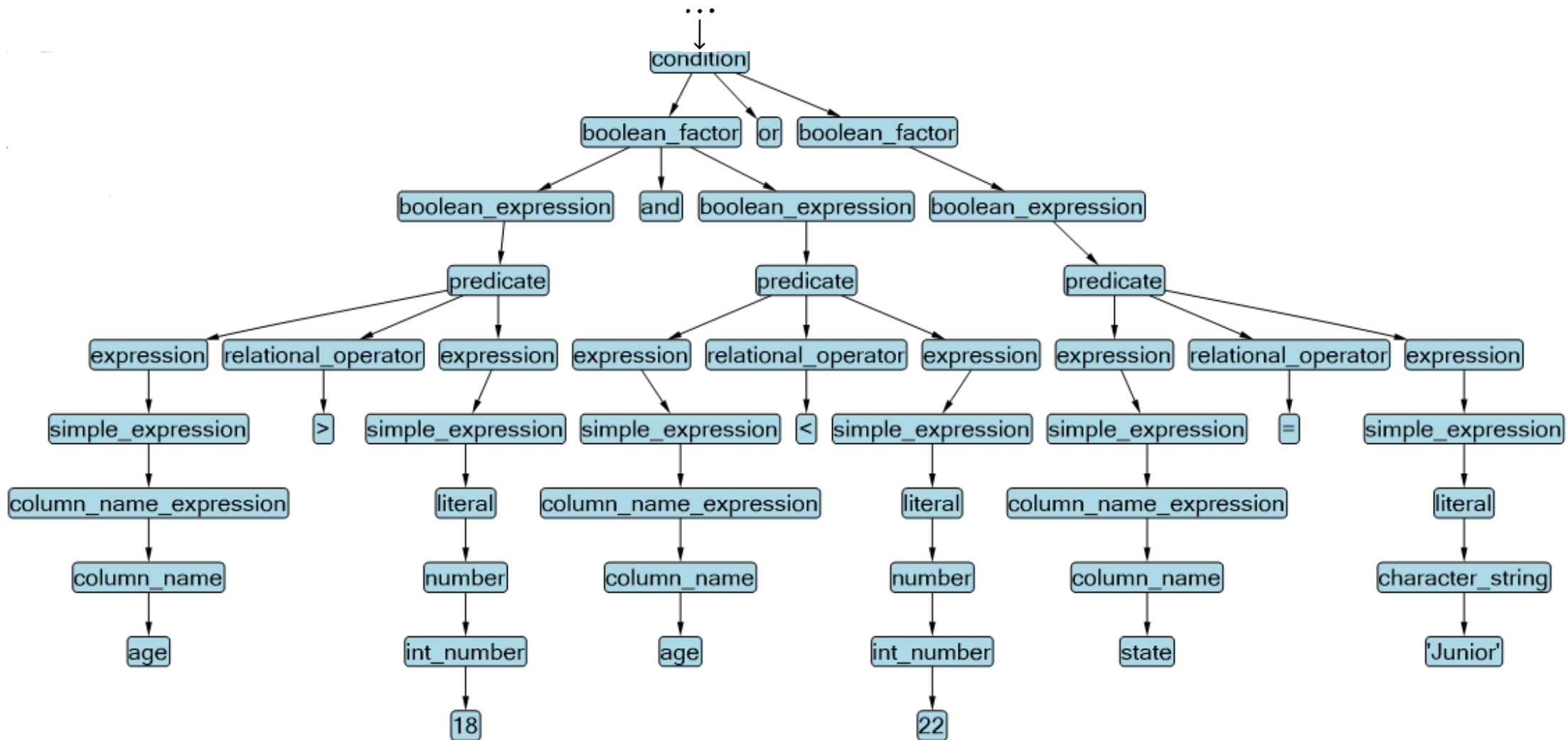
Правил: **60**; типов токенов: **81**

Построение AST для SQL-запроса

SELECT age
FROM users



Построение запроса с WHERE-условием

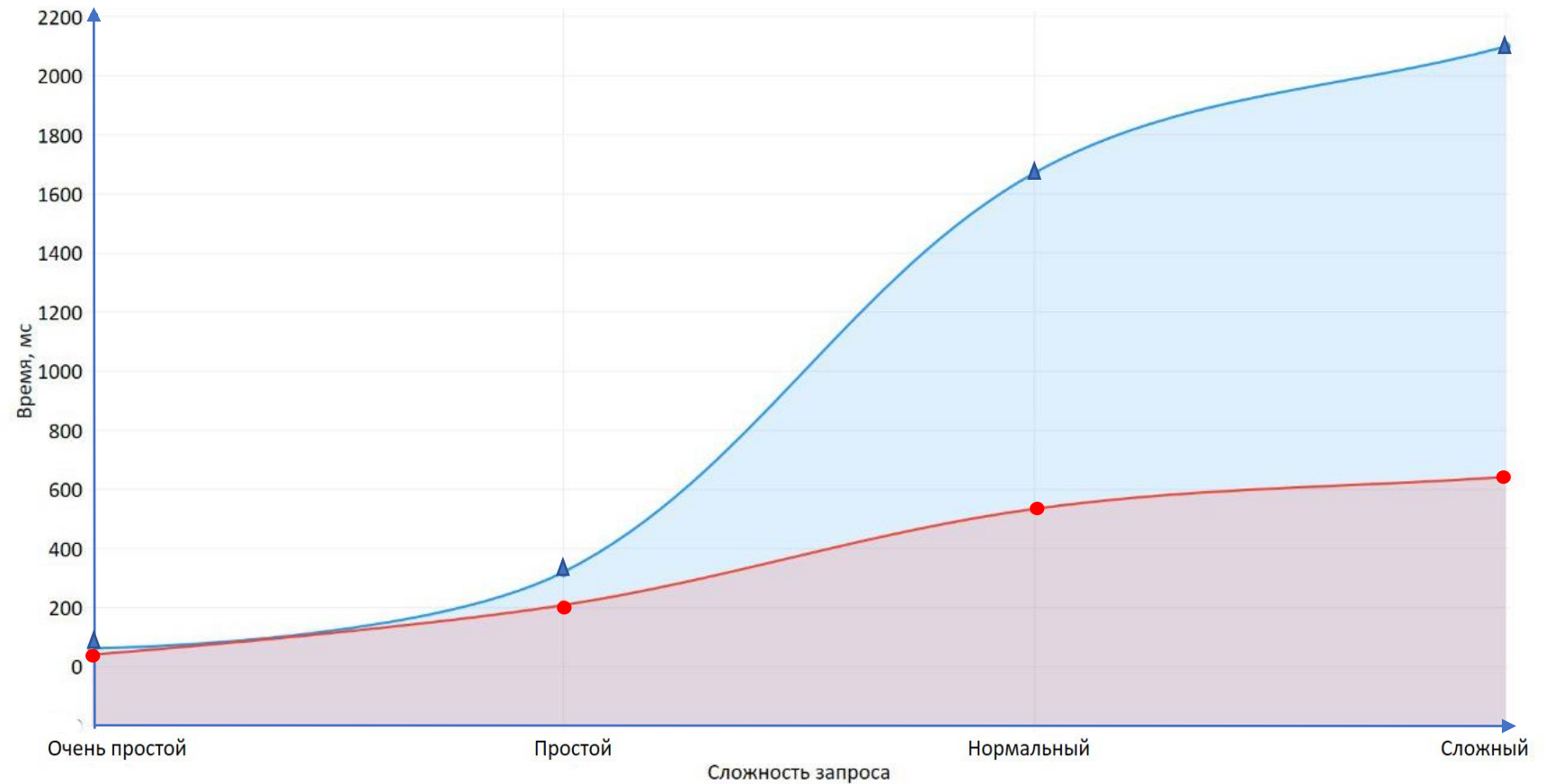


SELECT age
FROM users
WHERE age > 18 **AND** age < 22
 OR state = 'Junior'

Исследование зависимости времени построения AST от сложности запроса

Построение AST на основе:

▲ - строки;
● - токенов.



Заключение

Поставленная цель была достигнута

Решены следующие задачи:

- произведено сравнение существующих решений;
- описана и разработана грамматика SQL-запроса для ограниченной предметной области;
- создано ПО, реализующее разработанный метод;
- произведено исследование зависимости времени построения дерева синтаксического анализа от сложности запроса.

Направление дальнейшего развития

- расширение грамматики;
- анализ запроса и построение минимизированного плана запроса;
- выполнение построенного плана запроса.