

Методы и системы обработки больших данных

2. Разбор запроса

Схема компиляции запроса

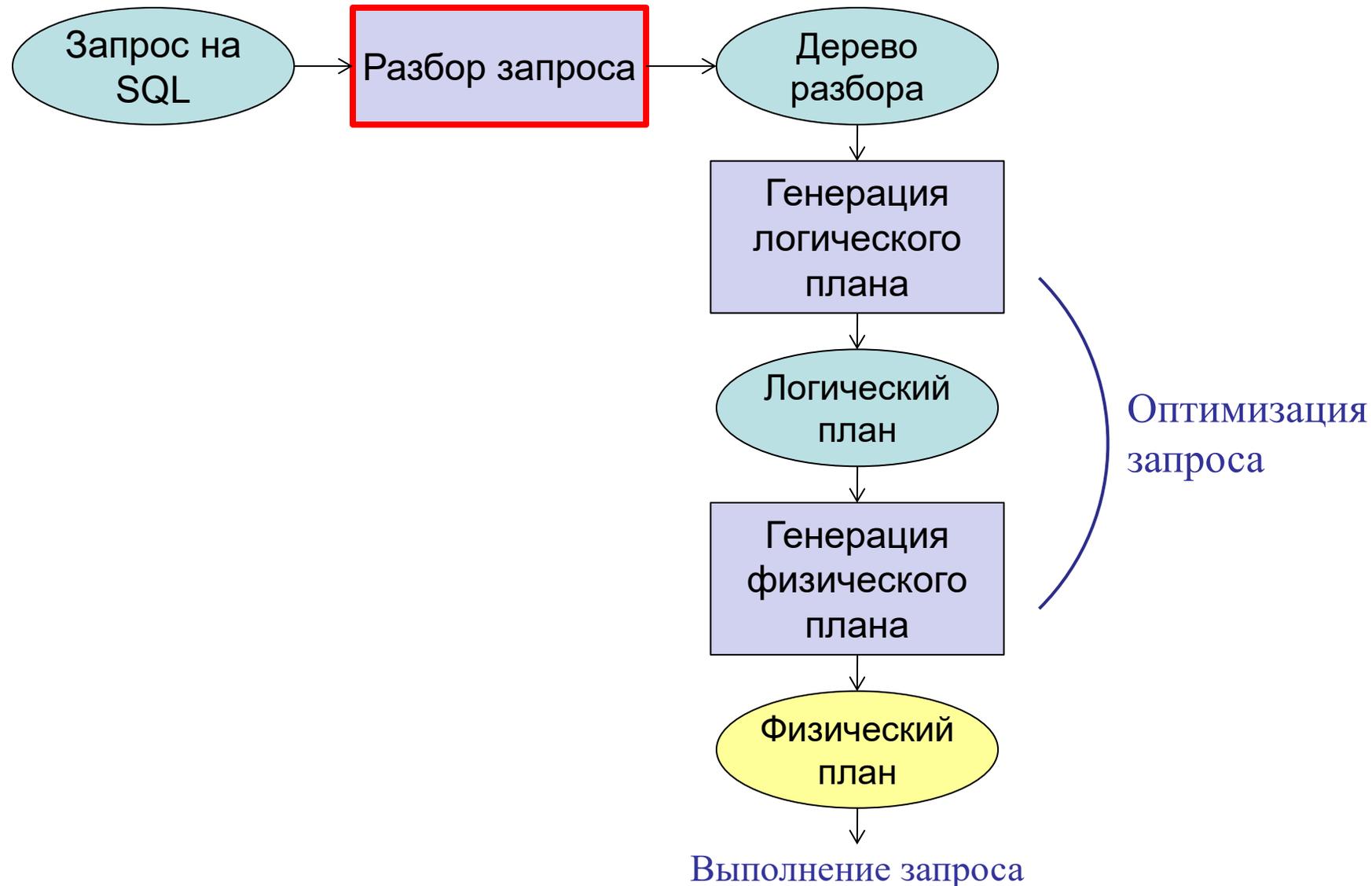
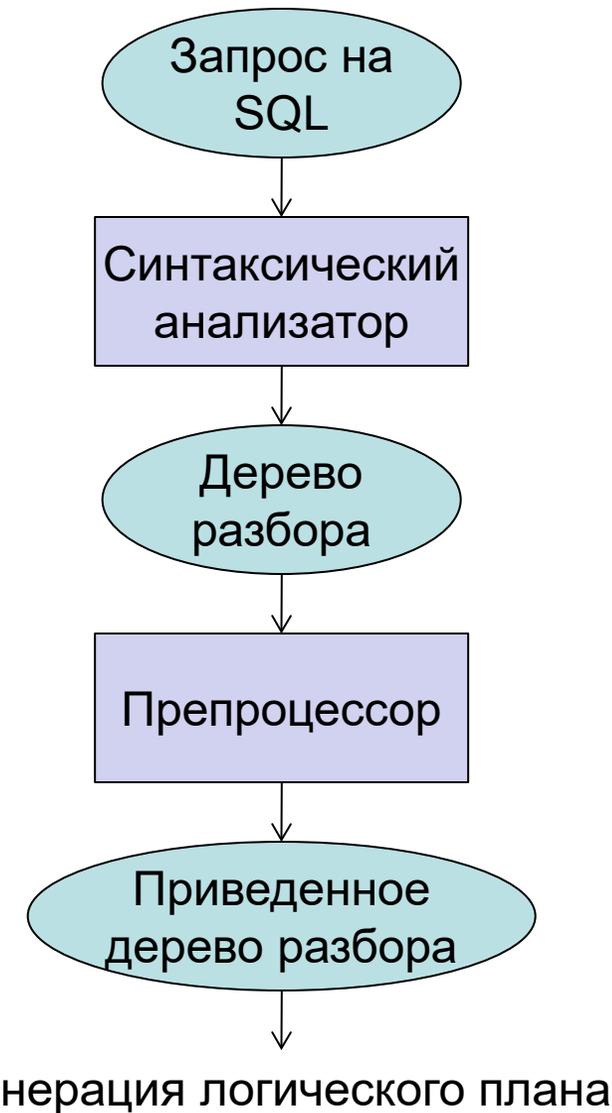


Схема разбора запроса

Фазы разбор запроса

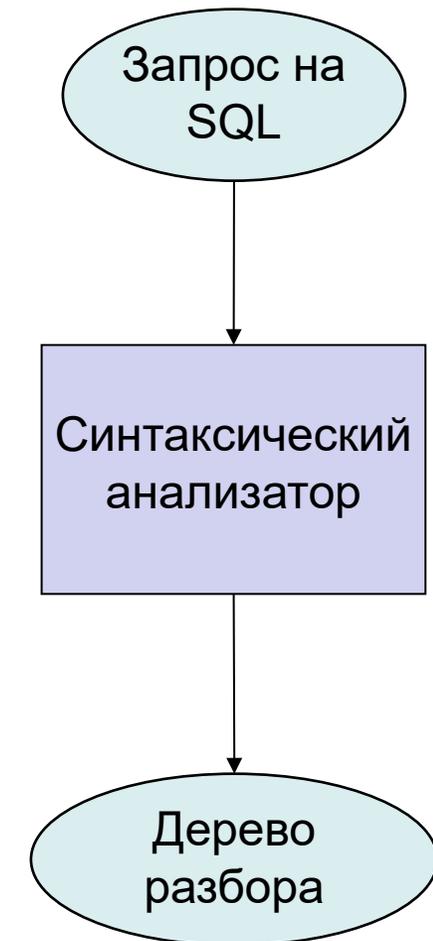
1. Синтаксический анализ запроса (выполняется *синтаксическим анализатором*)
2. Разрешение представлений и верификация запроса (выполняется *препроцессором*)



Синтаксический анализ

Дерево разбора состоит из узлов двух типов:

- 1. Атомы** - лексические элементы следующих типов: ключевые слова (например, SELECT); имена атрибутов или отношений; константы; скобки; операторы (например, + или >); и др.
- 2. Синтаксические категории** - имена семейств, представляющих часть запроса. Заключаются в угловые скобки. Например, <SFW>, <Condition>.



Грамматика SQL

<Query> ::= <SFW>

<Query> ::= (<Query>)

<SFW> ::= SELECT <SelList> FROM <FromList> WHERE <Condition>

<SelList> ::= <Attribute>, <SelList>

<SelList> ::= <Attribute>

<FromList> ::= <Relation>, <FromList>

<FromList> ::= <Relation>

<Condition> ::= <Condition> AND <Condition>

<Condition> ::= <Condition> OR <Condition>

<Condition> ::= NOT <Condition>

<Condition> ::= <Tuple> IN <Query>

<Condition> ::= <Attribute> = <Attribute>

<Condition> ::= <Attribute> < <Attribute>

<Condition> ::= <Attribute> > <Attribute>

• • • •

<Condition> ::= <Attribute> LIKE <Pattern>

<Condition> ::= <Attribute> EXIST <Pattern>

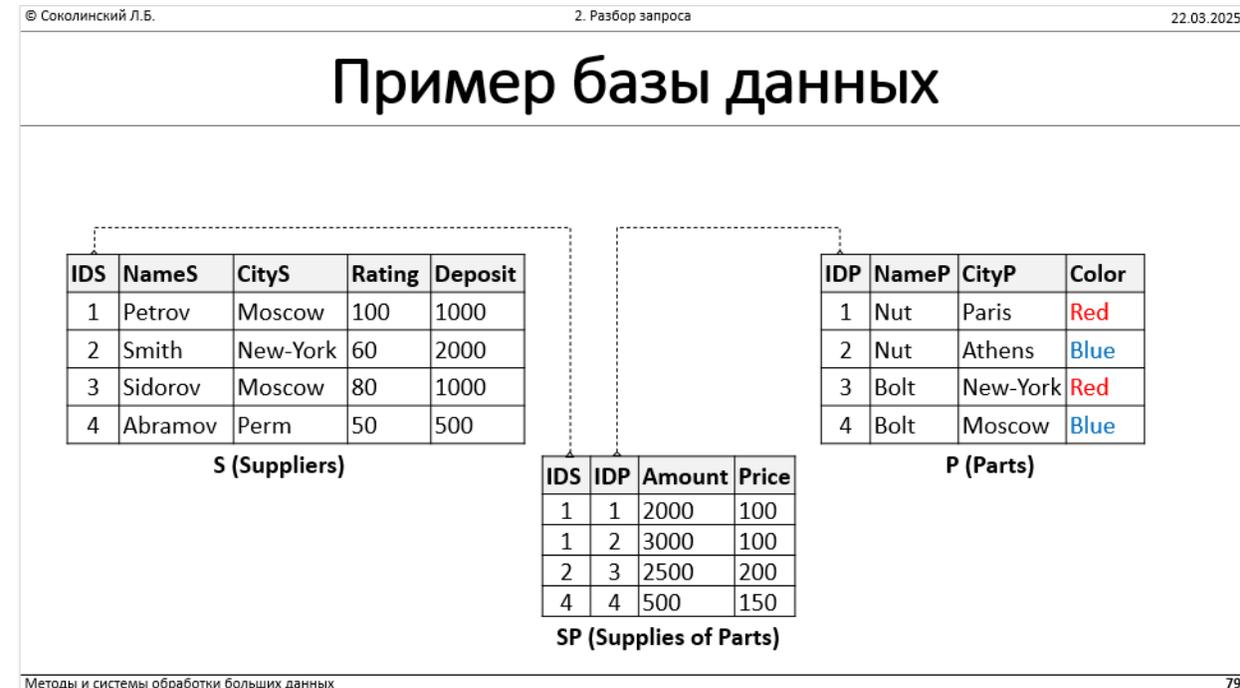
<Tuple> ::= <Attribute>

Базисные синтаксические категории

- $\langle \text{Attribute} \rangle = \text{идентификатор}$
- $\langle \text{Relation} \rangle = \text{идентификатор}$
- $\langle \text{Pattern} \rangle = \text{константа}$

Пример синтаксического анализа запроса

Имена поставщиков, поставляющих хотя бы одну деталь в количестве более 500 штук (в рамках одной поставки).



Построение дерева разбора

<Query>

Грамматика SQL

<Query> ::= <SFW>

<Query> ::= (<Query>)

<SFW> ::= SELECT <SelList> FROM <FromList> WHERE <Condition>

<SelList> ::= <Attribute>, <SelList>

<SelList> ::= <Attribute>

<FromList> ::= <Relation>, <FromList>

<FromList> ::= <Relation>

<Condition> ::= <Condition> AND <Condition>

<Condition> ::= <Condition> OR <Condition>

<Condition> ::= NOT <Condition>

<Condition> ::= <Tuple> IN <Query>

<Condition> ::= <Attribute> = <Attribute>

<Condition> ::= <Attribute> < <Attribute>

<Condition> ::= <Attribute> > <Attribute>

...

<Condition> ::= <Attribute> LIKE <Pattern>

<Condition> ::= <Attribute> EXIST <Pattern>

<Tuple> ::= <Attribute>

```
SELECT NameS
FROM S
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500)
```

Грамматика SQL

<Query> ::= <SFW>

<Query> ::= (<Query>)

<SFW> ::= SELECT <SelList> FROM <FromList> WHERE <Condition>

<SelList> ::= <Attribute>, <SelList>

<SelList> ::= <Attribute>

<FromList> ::= <Relation>, <FromList>

<FromList> ::= <Relation>

<Condition> ::= <Condition> AND <Condition>

<Condition> ::= <Condition> OR <Condition>

<Condition> ::= NOT <Condition>

<Condition> ::= <Tuple> IN <Query>

<Condition> ::= <Attribute> = <Attribute>

<Condition> ::= <Attribute> < <Attribute>

<Condition> ::= <Attribute> > <Attribute>

...

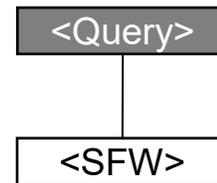
<Condition> ::= <Attribute> LIKE <Pattern>

<Condition> ::= <Attribute> EXIST <Pattern>

<Tuple> ::= <Attribute>

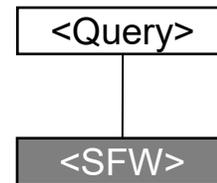
```
SELECT NameS
FROM S
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500)
```

Построение дерева разбора



$\langle \text{Query} \rangle ::= \langle \text{SFW} \rangle$

Построение дерева разбора



Грамматика SQL

<Query> ::= <SFW>

<Query> ::= (<Query>)

<SFW> ::= SELECT <SelList> FROM <FromList> WHERE <Condition>

<SelList> ::= <Attribute>, <SelList>

<SelList> ::= <Attribute>

<FromList> ::= <Relation>, <FromList>

<FromList> ::= <Relation>

<Condition> ::= <Condition> AND <Condition>

<Condition> ::= <Condition> OR <Condition>

<Condition> ::= NOT <Condition>

<Condition> ::= <Tuple> IN <Query>

<Condition> ::= <Attribute> = <Attribute>

<Condition> ::= <Attribute> < <Attribute>

<Condition> ::= <Attribute> > <Attribute>

...

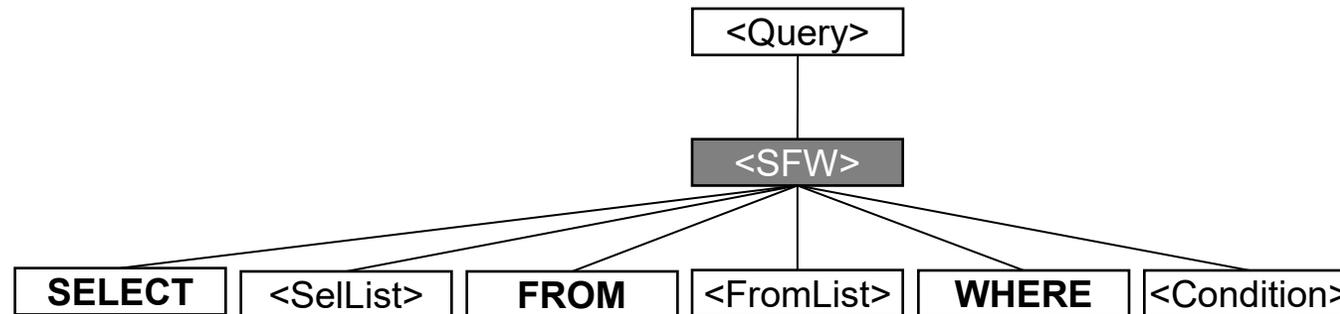
<Condition> ::= <Attribute> LIKE <Pattern>

<Condition> ::= <Attribute> EXIST <Pattern>

<Tuple> ::= <Attribute>

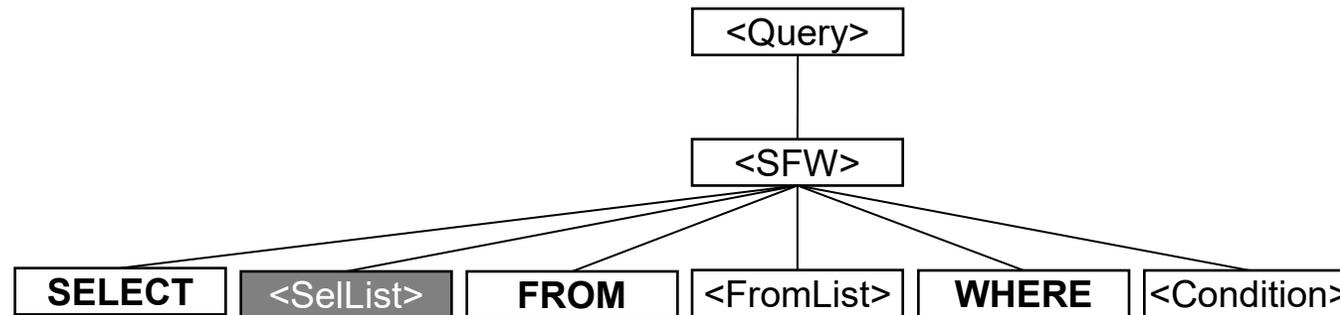
```
SELECT NameS
FROM S
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500)
```

Построение дерева разбора



$\langle \text{SFW} \rangle ::= \text{SELECT } \langle \text{SelList} \rangle \text{ FROM } \langle \text{FromList} \rangle \text{ WHERE } \langle \text{Condition} \rangle$

Построение дерева разбора



Грамматика SQL

<Query> ::= <SFW>

<Query> ::= (<Query>)

<SFW> ::= SELECT <SelList> FROM <FromList> WHERE <Condition>

<SelList> ::= <Attribute>, <SelList>

<SelList> ::= <Attribute>

<FromList> ::= <Relation>, <FromList>

<FromList> ::= <Relation>

<Condition> ::= <Condition> AND <Condition>

<Condition> ::= <Condition> OR <Condition>

<Condition> ::= NOT <Condition>

<Condition> ::= <Tuple> IN <Query>

<Condition> ::= <Attribute> = <Attribute>

<Condition> ::= <Attribute> < <Attribute>

<Condition> ::= <Attribute> > <Attribute>

...

<Condition> ::= <Attribute> LIKE <Pattern>

<Condition> ::= <Attribute> EXIST <Pattern>

<Tuple> ::= <Attribute>

```
SELECT NameS
FROM S
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500);
```

Грамматика SQL

<Query> ::= <SFW>

<Query> ::= (<Query>)

<SFW> ::= SELECT <SelList> FROM <FromList> WHERE <Condition>

<SelList> ::= <Attribute>, <SelList>

<SelList> ::= <Attribute>

<FromList> ::= <Relation>, <FromList>

<FromList> ::= <Relation>

<Condition> ::= <Condition> AND <Condition>

<Condition> ::= <Condition> OR <Condition>

<Condition> ::= NOT <Condition>

<Condition> ::= <Tuple> IN <Query>

<Condition> ::= <Attribute> = <Attribute>

<Condition> ::= <Attribute> < <Attribute>

<Condition> ::= <Attribute> > <Attribute>

...

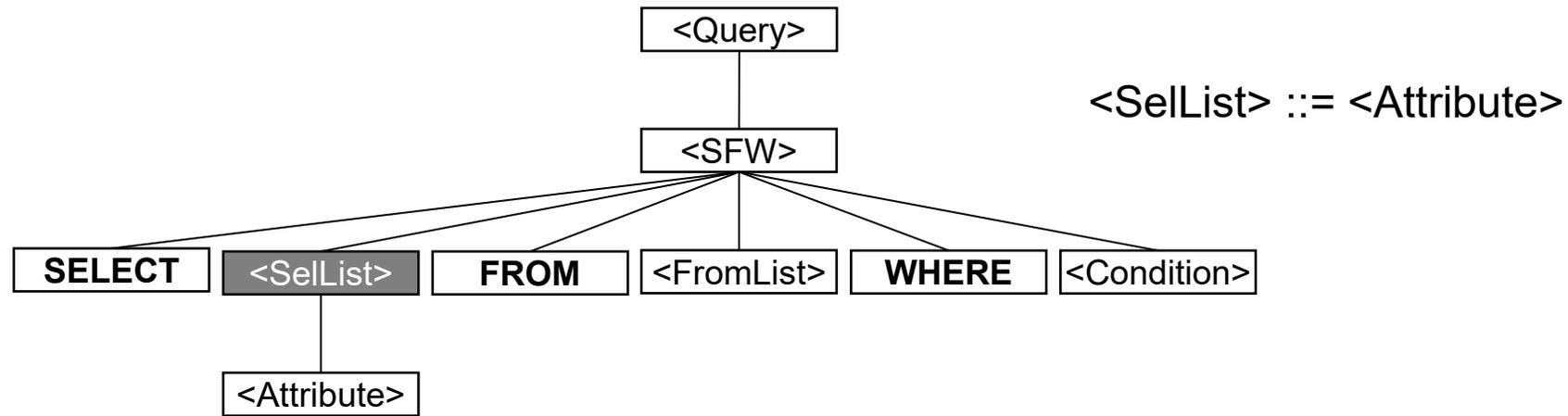
<Condition> ::= <Attribute> LIKE <Pattern>

<Condition> ::= <Attribute> EXIST <Pattern>

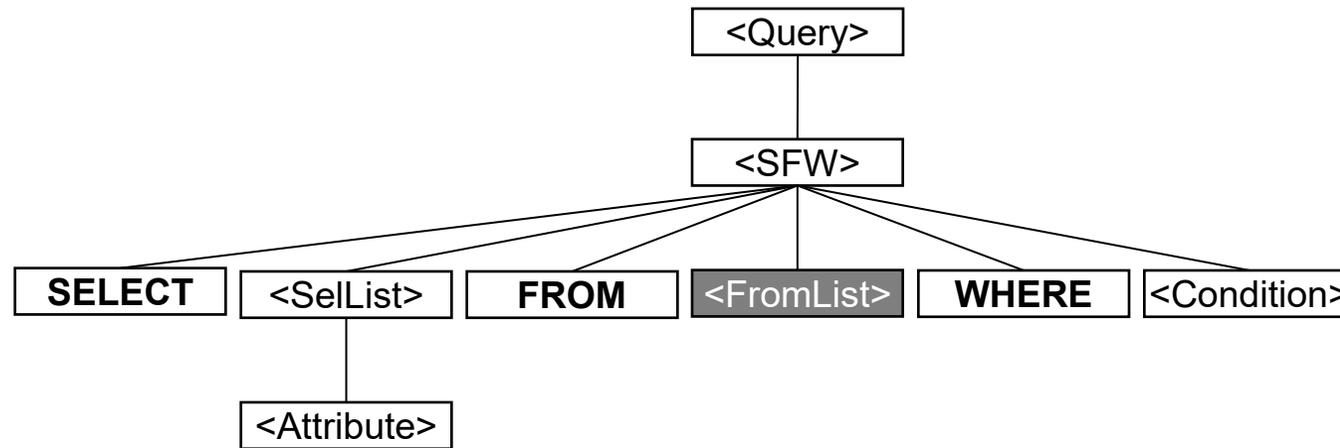
<Tuple> ::= <Attribute>

```
SELECT NameS
FROM S
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500);
```

Построение дерева разбора



Построение дерева разбора



Грамматика SQL

<Query> ::= <SFW>

<Query> ::= (<Query>)

<SFW> ::= SELECT <SelList> FROM <FromList> WHERE <Condition>

<SelList> ::= <Attribute>, <SelList>

<SelList> ::= <Attribute>

<FromList> ::= <Relation>, <FromList>

<FromList> ::= <Relation>

<Condition> ::= <Condition> AND <Condition>

<Condition> ::= <Condition> OR <Condition>

<Condition> ::= NOT <Condition>

<Condition> ::= <Tuple> IN <Query>

<Condition> ::= <Attribute> = <Attribute>

<Condition> ::= <Attribute> < <Attribute>

<Condition> ::= <Attribute> > <Attribute>

...

<Condition> ::= <Attribute> LIKE <Pattern>

<Condition> ::= <Attribute> EXIST <Pattern>

<Tuple> ::= <Attribute>

```
SELECT NameS
FROM S
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500);
```

Грамматика SQL

<Query> ::= <SFW>

<Query> ::= (<Query>)

<SFW> ::= SELECT <SelList> FROM <FromList> WHERE <Condition>

<SelList> ::= <Attribute>, <SelList>

<SelList> ::= <Attribute>

<FromList> ::= <Relation>, <FromList>

<FromList> ::= <Relation>

<Condition> ::= <Condition> AND <Condition>

<Condition> ::= <Condition> OR <Condition>

<Condition> ::= NOT <Condition>

<Condition> ::= <Tuple> IN <Query>

<Condition> ::= <Attribute> = <Attribute>

<Condition> ::= <Attribute> < <Attribute>

<Condition> ::= <Attribute> > <Attribute>

...

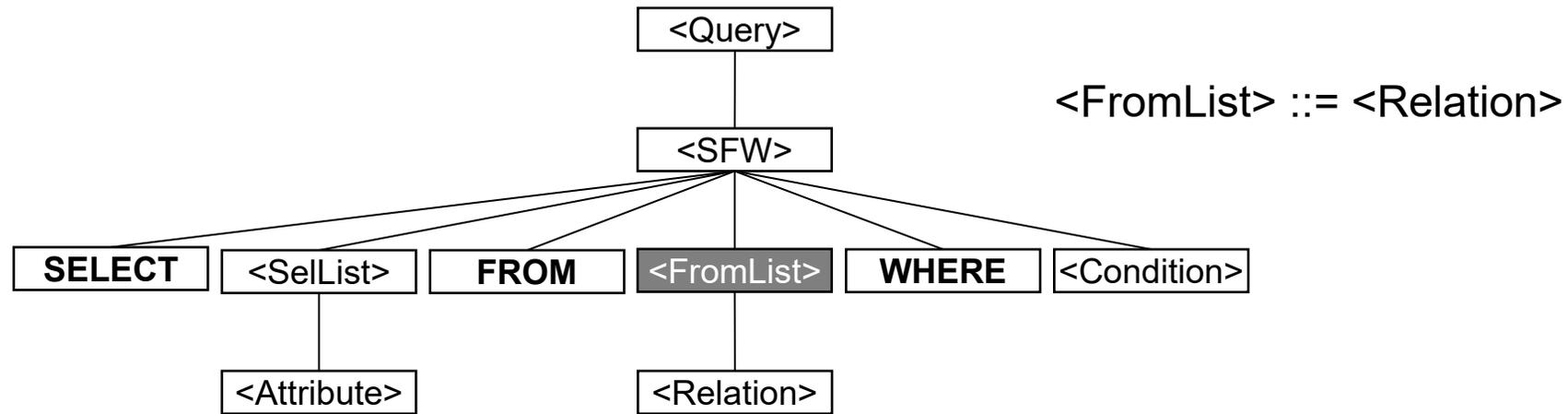
<Condition> ::= <Attribute> LIKE <Pattern>

<Condition> ::= <Attribute> EXIST <Pattern>

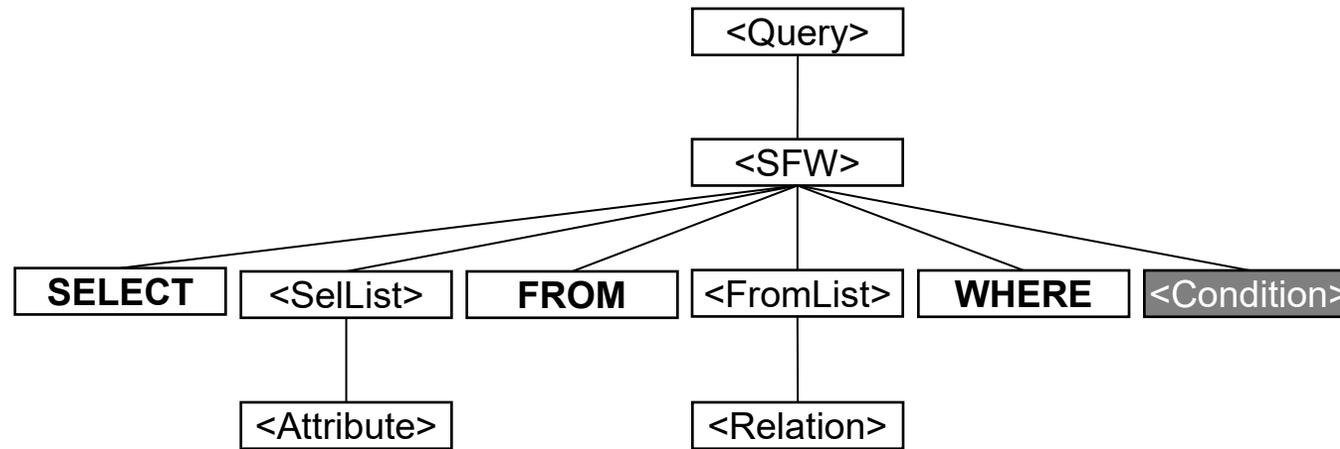
<Tuple> ::= <Attribute>

```
SELECT NameS
FROM S
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500);
```

Построение дерева разбора



Построение дерева разбора



Грамматика SQL

<Query> ::= <SFW>

<Query> ::= (<Query>)

<SFW> ::= SELECT <SelList> FROM <FromList> WHERE <Condition>

<SelList> ::= <Attribute>, <SelList>

<SelList> ::= <Attribute>

<FromList> ::= <Relation>, <FromList>

<FromList> ::= <Relation>

<Condition> ::= <Condition> AND <Condition>

<Condition> ::= <Condition> OR <Condition>

<Condition> ::= NOT <Condition>

<Condition> ::= <Tuple> IN <Query>

<Condition> ::= <Attribute> = <Attribute>

<Condition> ::= <Attribute> < <Attribute>

<Condition> ::= <Attribute> > <Attribute>

...

<Condition> ::= <Attribute> LIKE <Pattern>

<Condition> ::= <Attribute> EXIST <Pattern>

<Tuple> ::= <Attribute>

```
SELECT NameS
FROM S
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500);
```

Грамматика SQL

<Query> ::= <SFW>

<Query> ::= (<Query>)

<SFW> ::= SELECT <SelList> FROM <FromList> WHERE <Condition>

<SelList> ::= <Attribute>, <SelList>

<SelList> ::= <Attribute>

<FromList> ::= <Relation>, <FromList>

<FromList> ::= <Relation>

<Condition> ::= <Condition> AND <Condition>

<Condition> ::= <Condition> OR <Condition>

<Condition> ::= NOT <Condition>

<Condition> ::= <Tuple> IN <Query>

<Condition> ::= <Attribute> = <Attribute>

<Condition> ::= <Attribute> < <Attribute>

<Condition> ::= <Attribute> > <Attribute>

...

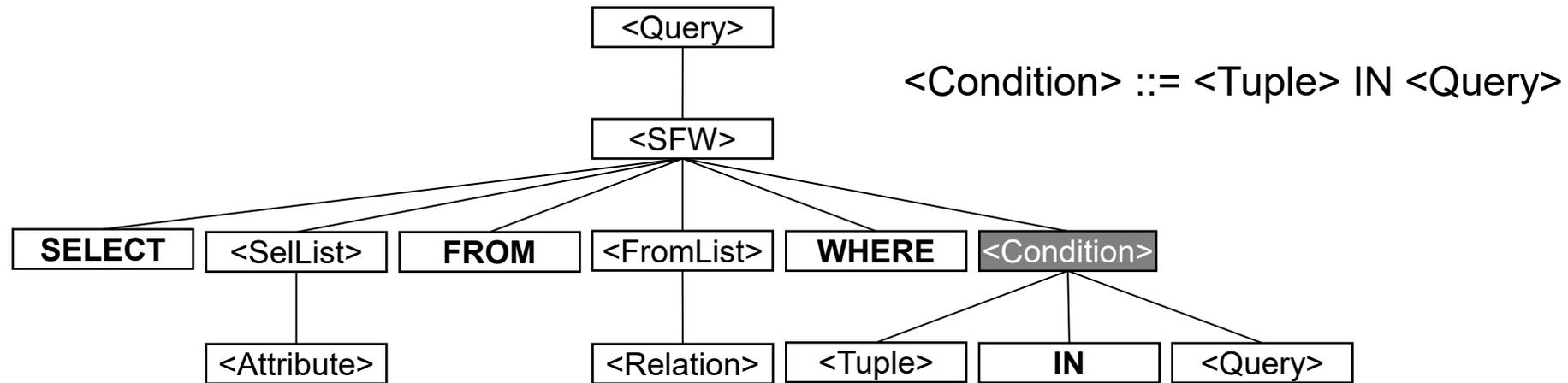
<Condition> ::= <Attribute> LIKE <Pattern>

<Condition> ::= <Attribute> EXIST <Pattern>

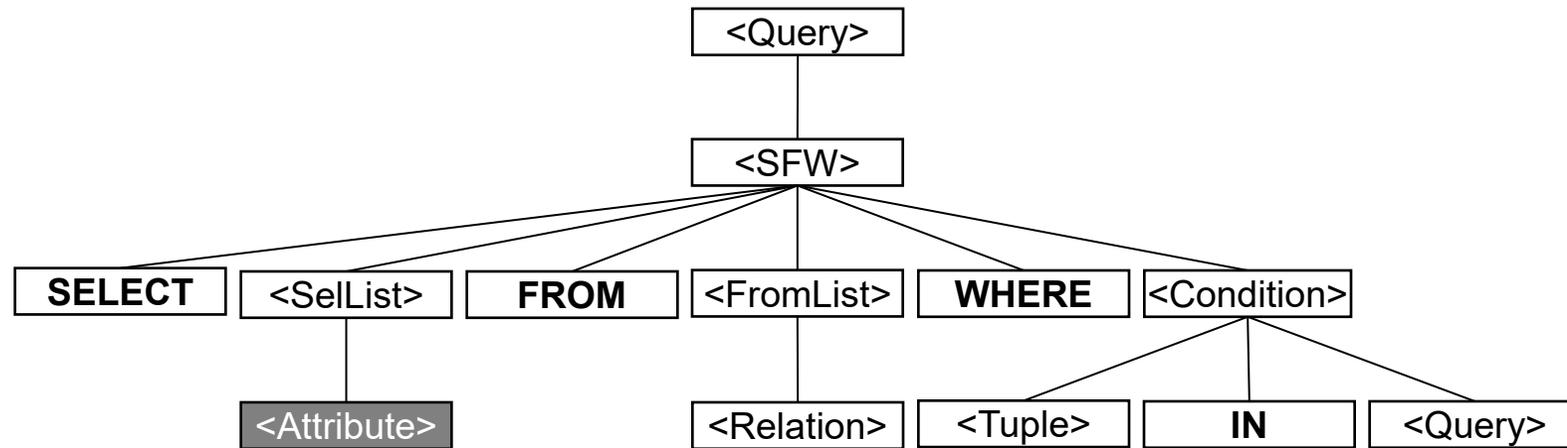
<Tuple> ::= <Attribute>

```
SELECT NameS
FROM S
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500);
```

Построение дерева разбора

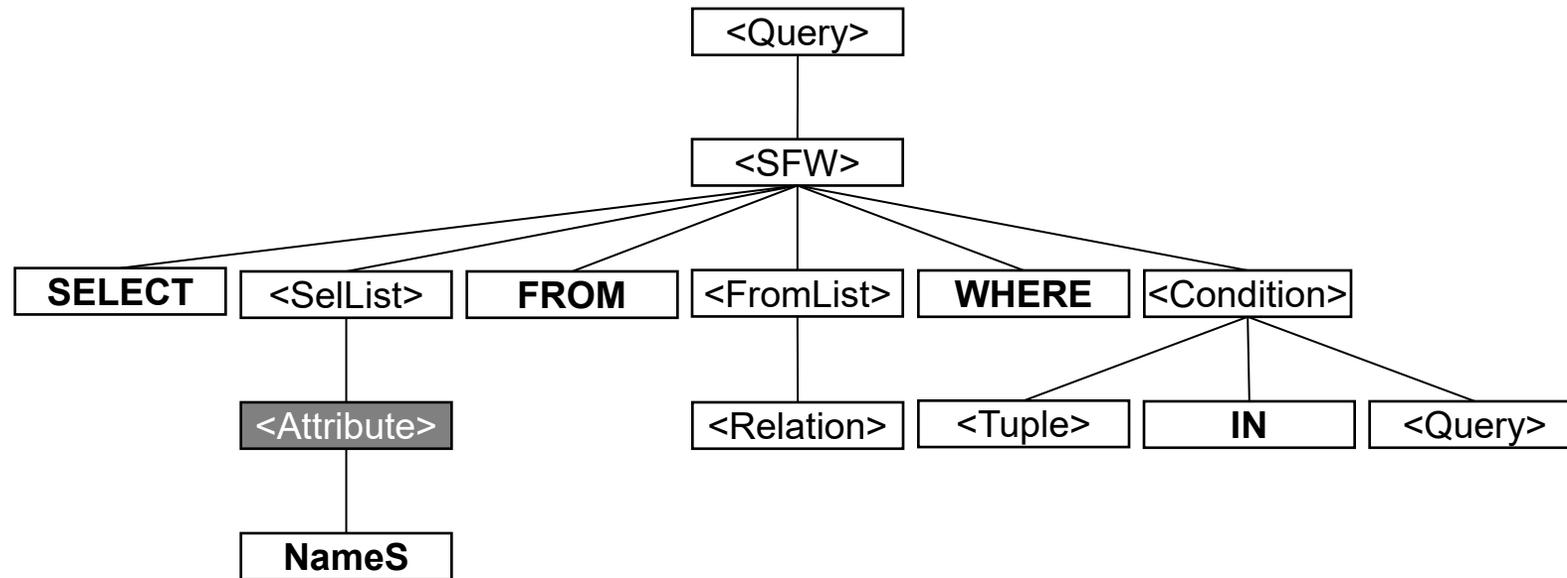


Построение дерева разбора



```
SELECT NameS
FROM S
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500);
```

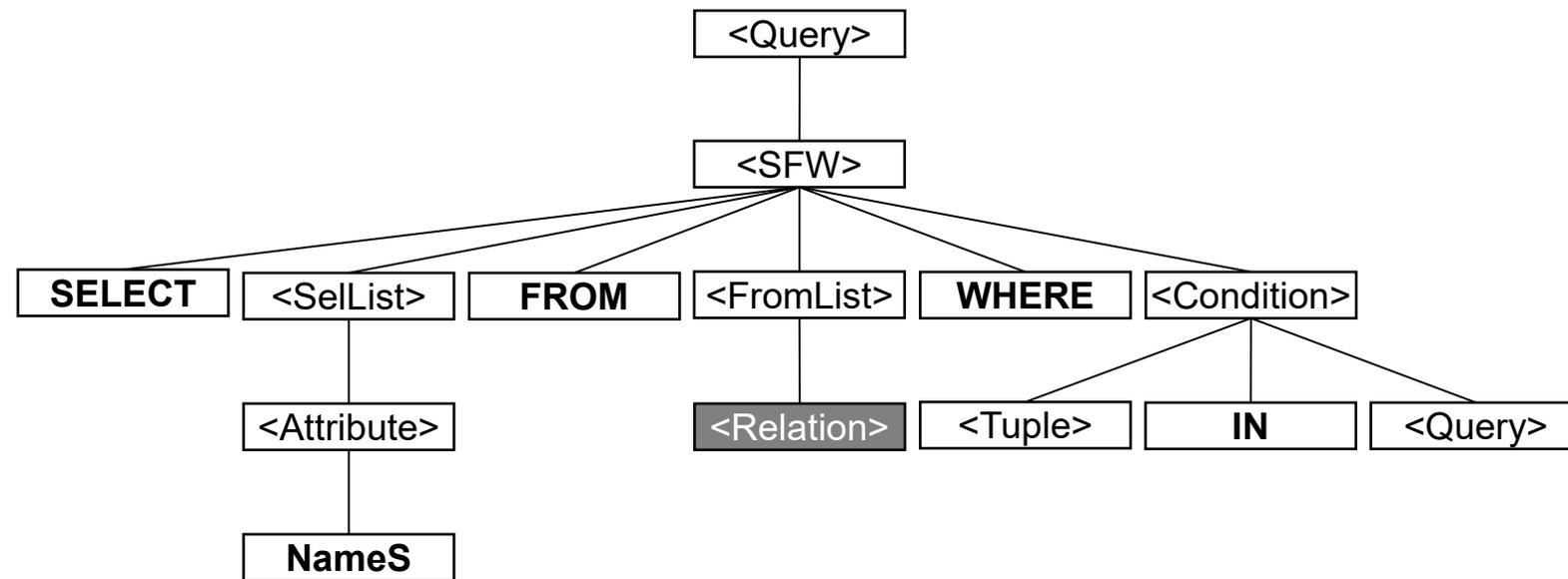
Построение дерева разбора



<Attribute> = *идентификатор*

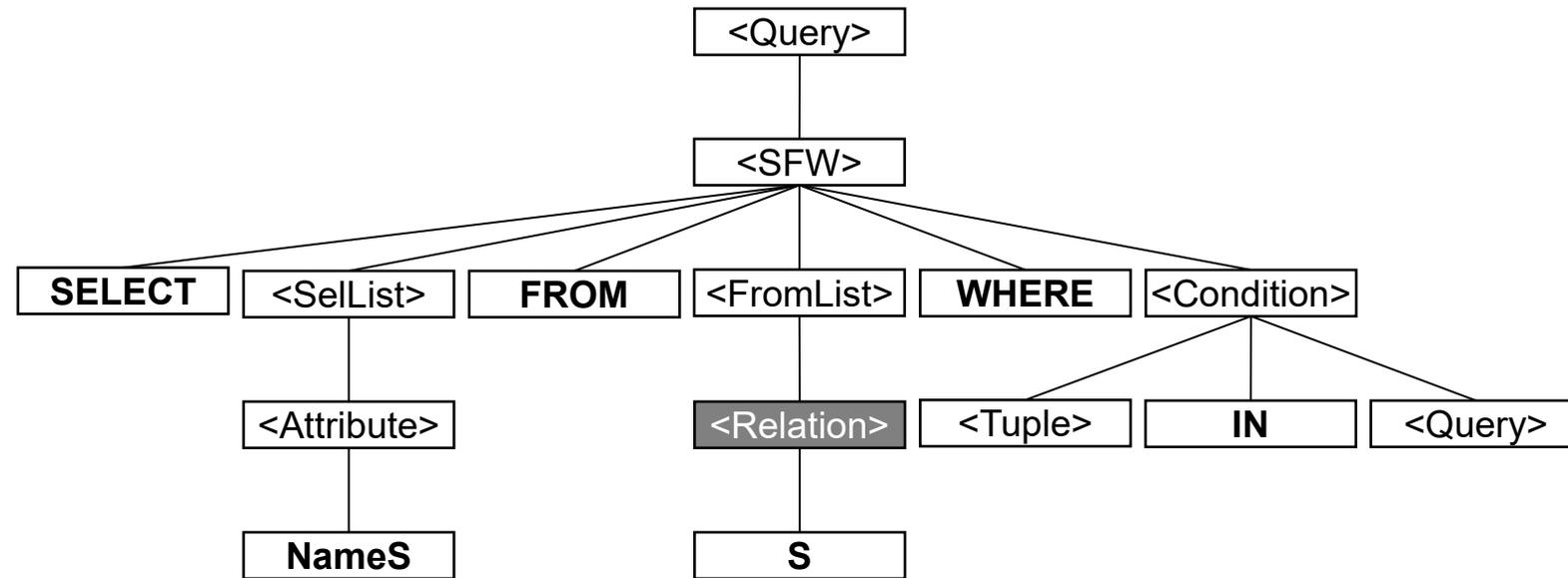
```
SELECT NameS  
FROM S  
WHERE IDS IN (  
  SELECT IDS  
  FROM SP  
  WHERE Amount > 500);
```

Построение дерева разбора



```
SELECT NameS
FROM S
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500);
```

Построение дерева разбора

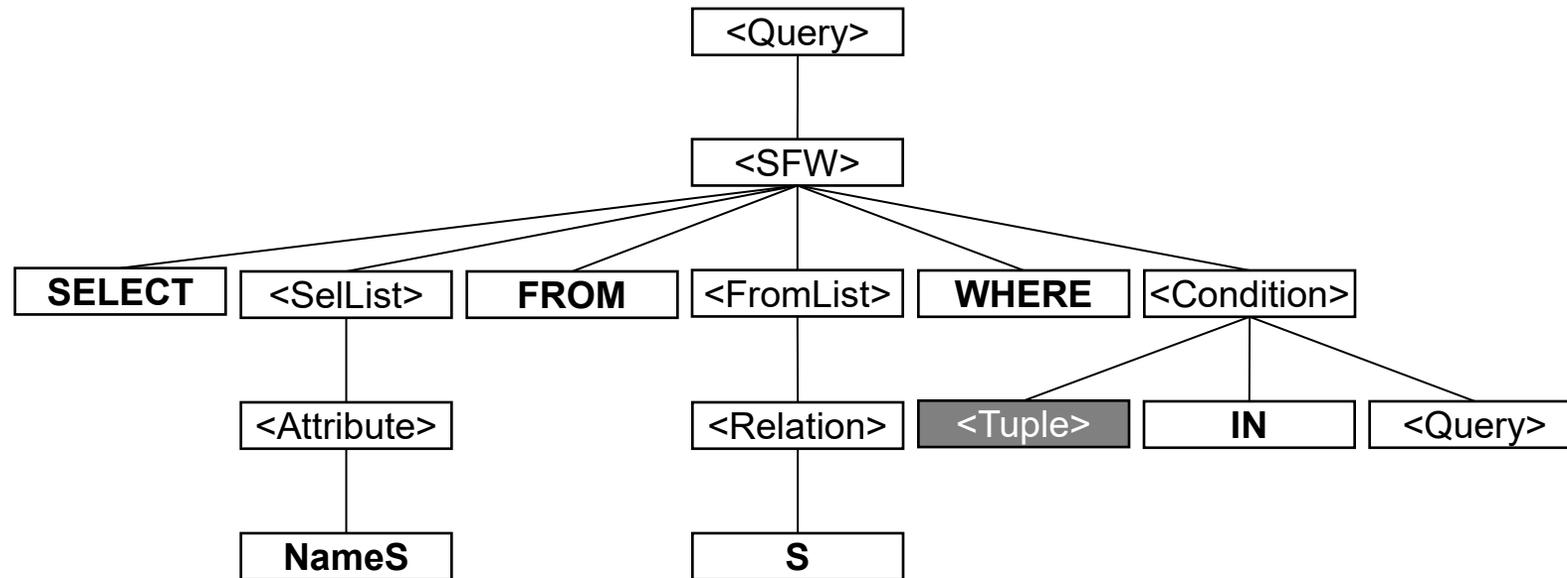


<Relation> = *идентификатор*

```

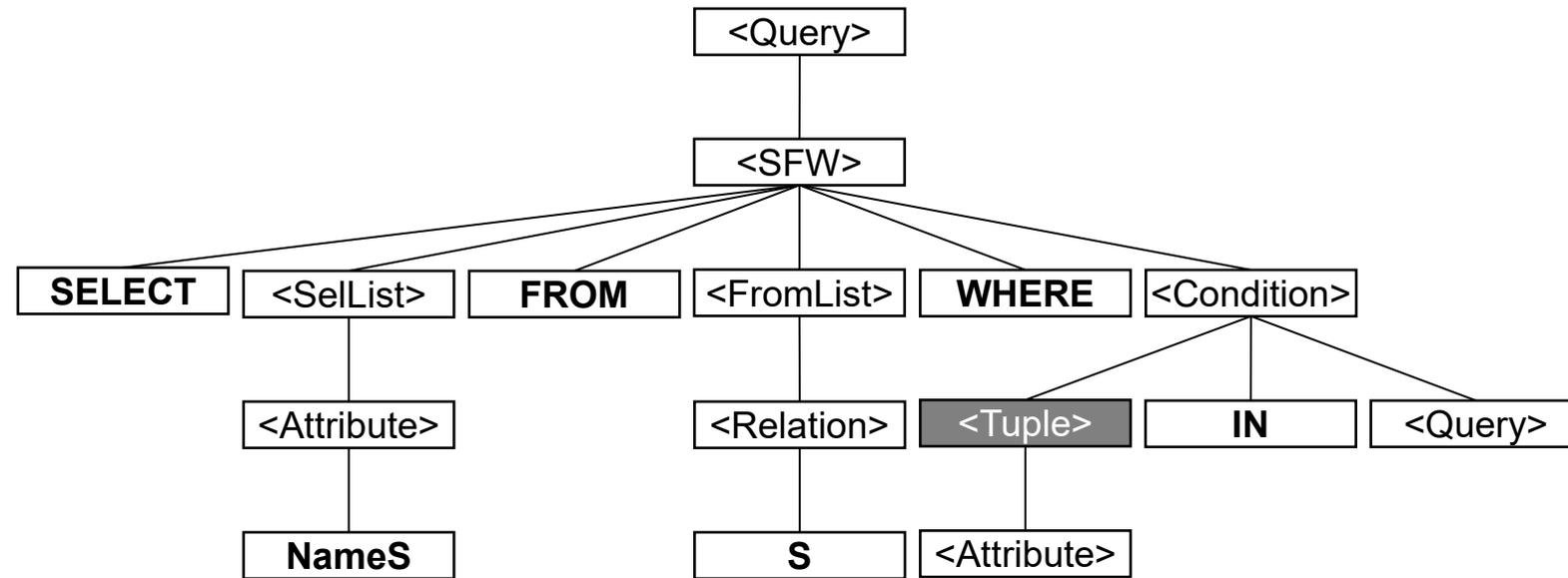
SELECT NameS
FROM S
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500);
  
```

Построение дерева разбора



```
SELECT NameS
FROM
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500);
```

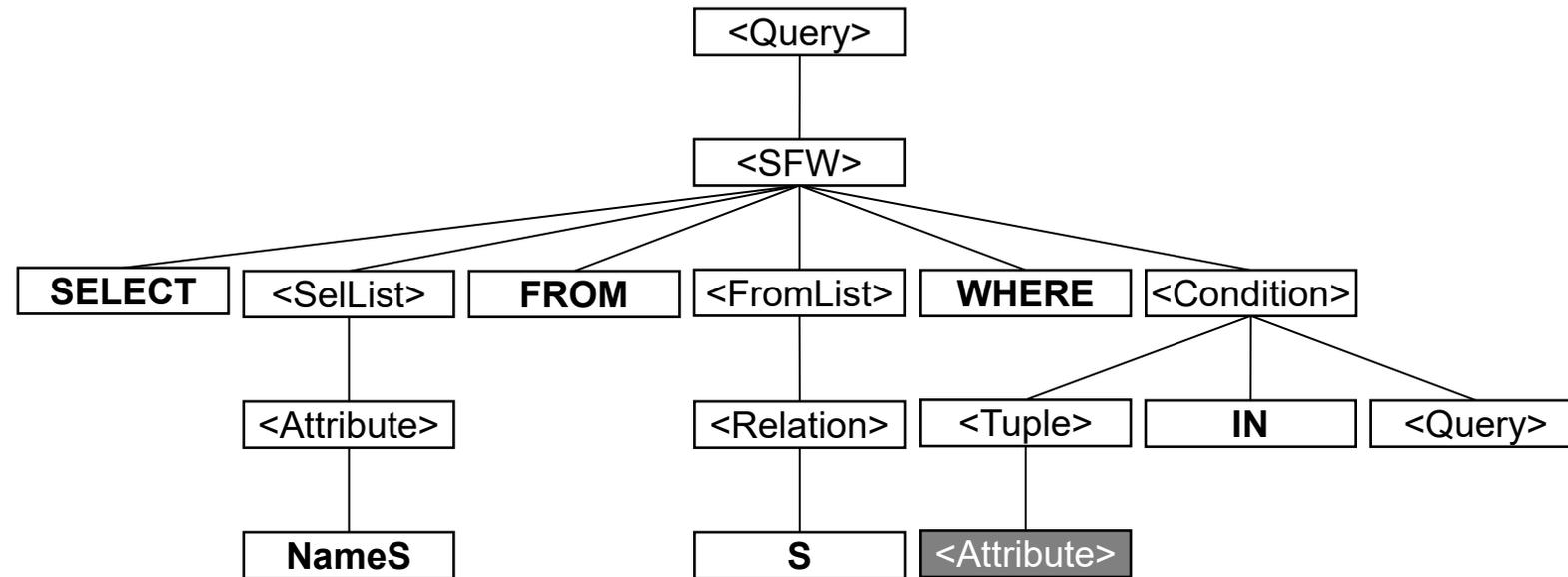
Построение дерева разбора



$\langle \text{Tuple} \rangle ::= \langle \text{Attribute} \rangle$

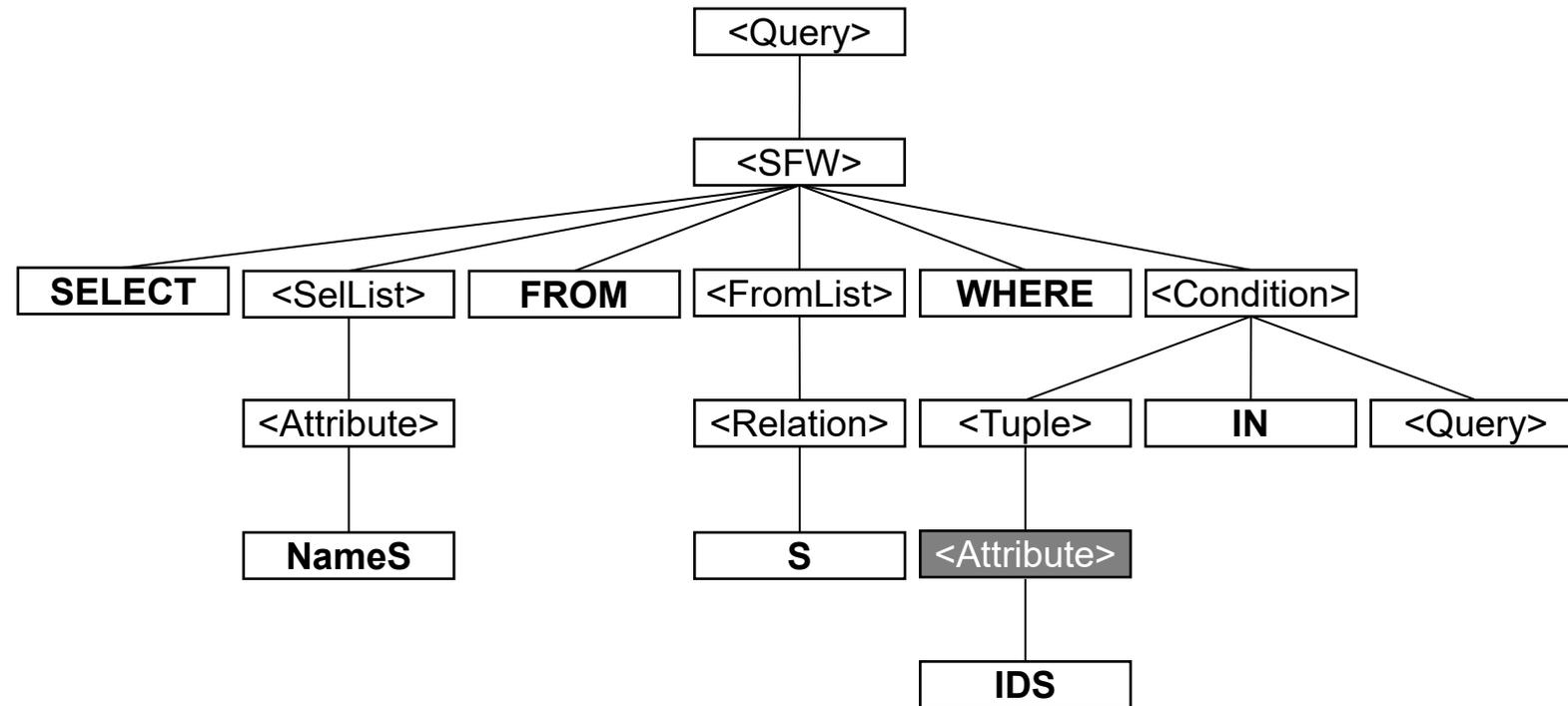
```
SELECT NameS
FROM
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500);
```

Построение дерева разбора



```
SELECT NameS
FROM
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500);
```

Построение дерева разбора

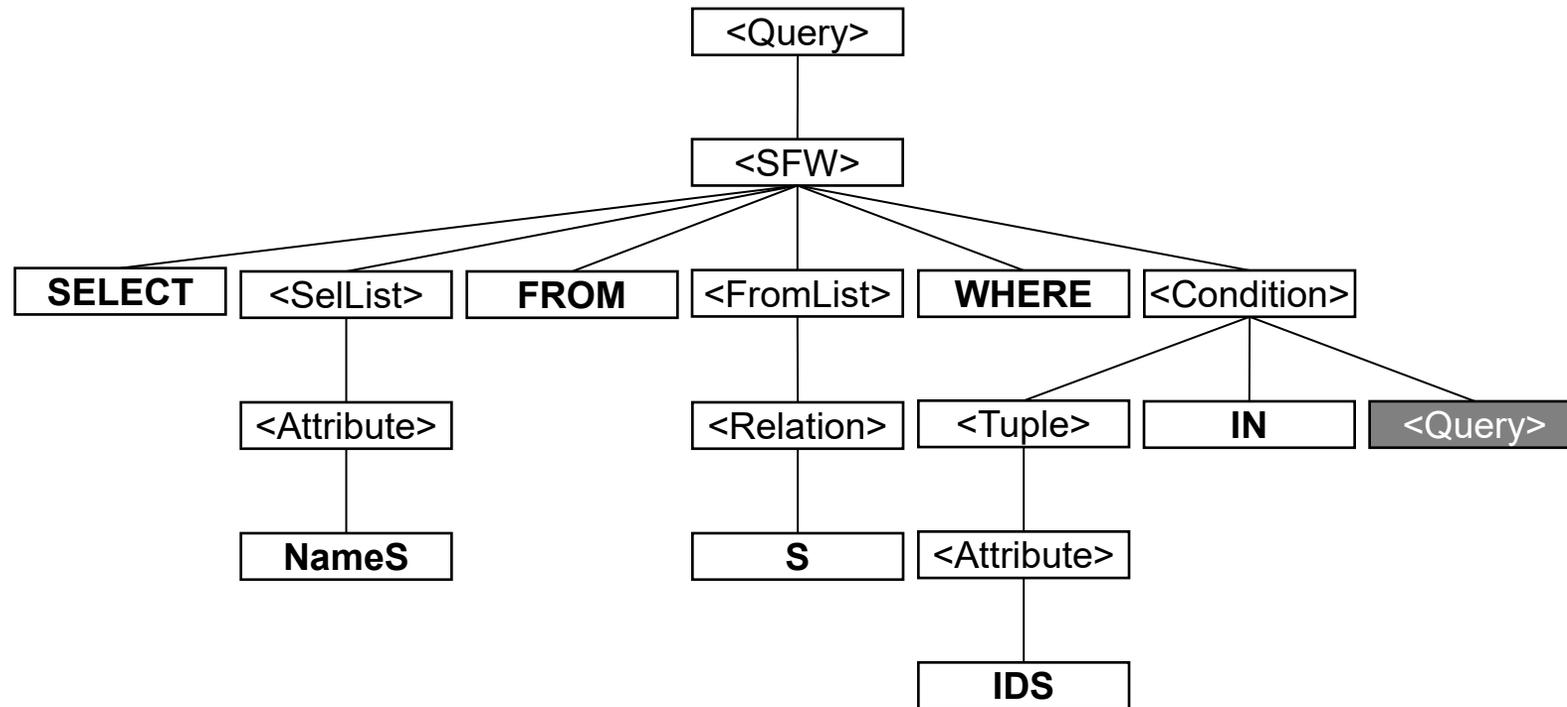


<Attribute> = *идентификатор*

```

SELECT NameS
FROM
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500);
  
```

Построение дерева разбора



Грамматика SQL

<Query> ::= <SFW>

<Query> ::= (<Query>)

<SFW> ::= SELECT <SelList> FROM <FromList> WHERE <Condition>

<SelList> ::= <Attribute>, <SelList>

<SelList> ::= <Attribute>

<FromList> ::= <Relation>, <FromList>

<FromList> ::= <Relation>

<Condition> ::= <Condition> AND <Condition>

<Condition> ::= <Condition> OR <Condition>

<Condition> ::= NOT <Condition>

<Condition> ::= <Tuple> IN <Query>

<Condition> ::= <Attribute> = <Attribute>

<Condition> ::= <Attribute> < <Attribute>

<Condition> ::= <Attribute> > <Attribute>

...

<Condition> ::= <Attribute> LIKE <Pattern>

<Condition> ::= <Attribute> EXIST <Pattern>

<Tuple> ::= <Attribute>

```
SELECT NameS
FROM S
WHERE IDS IN (
SELECT IDS
FROM SP
WHERE Amount > 500)
```

Грамматика SQL

<Query> ::= <SFW>

<Query> ::= (<Query>)

<SFW> ::= SELECT <SelList> FROM <FromList> WHERE <Condition>

<SelList> ::= <Attribute>, <SelList>

<SelList> ::= <Attribute>

<FromList> ::= <Relation>, <FromList>

<FromList> ::= <Relation>

<Condition> ::= <Condition> AND <Condition>

<Condition> ::= <Condition> OR <Condition>

<Condition> ::= NOT <Condition>

<Condition> ::= <Tuple> IN <Query>

<Condition> ::= <Attribute> = <Attribute>

<Condition> ::= <Attribute> < <Attribute>

<Condition> ::= <Attribute> > <Attribute>

...

<Condition> ::= <Attribute> LIKE <Pattern>

<Condition> ::= <Attribute> EXIST <Pattern>

<Tuple> ::= <Attribute>

```
SELECT NameS
FROM S
WHERE IDS IN (
  SELECT IDS
  FROM SP
  WHERE Amount > 500)
```

Дерево разбора

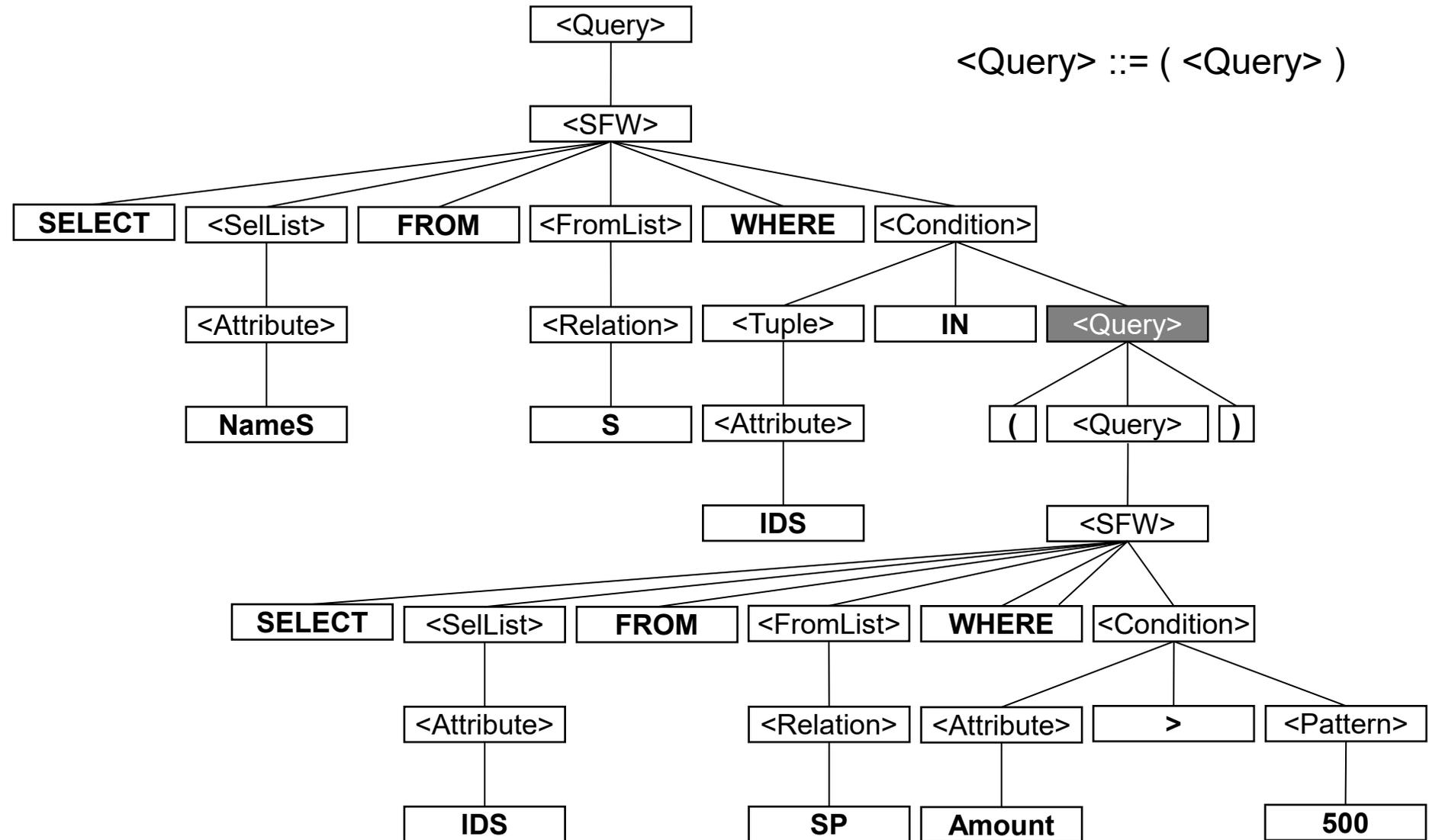
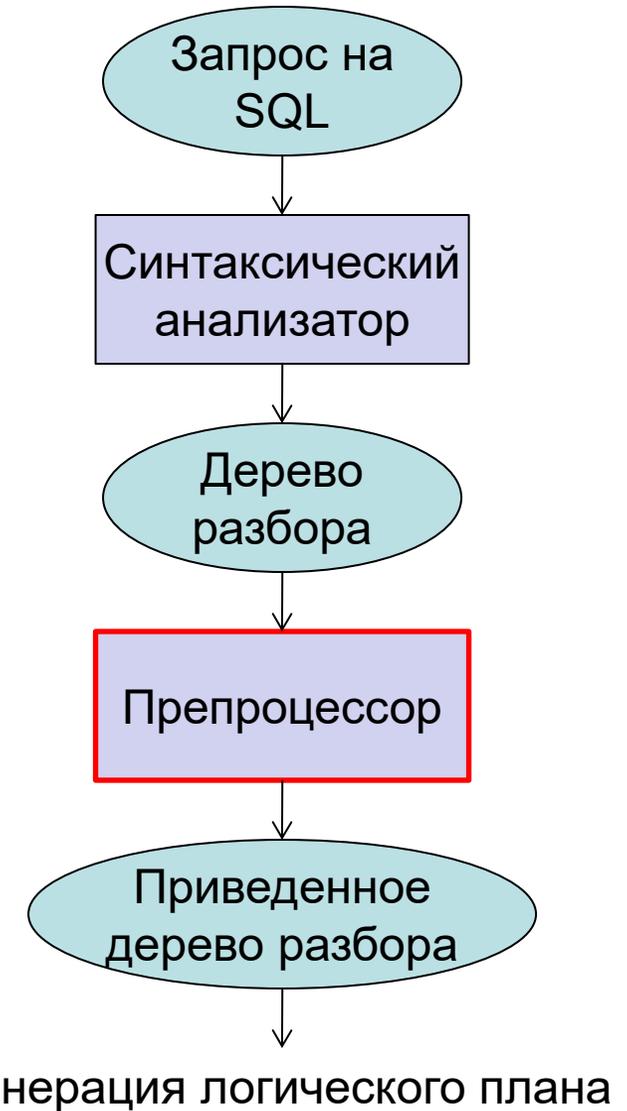


Схема разбора запроса

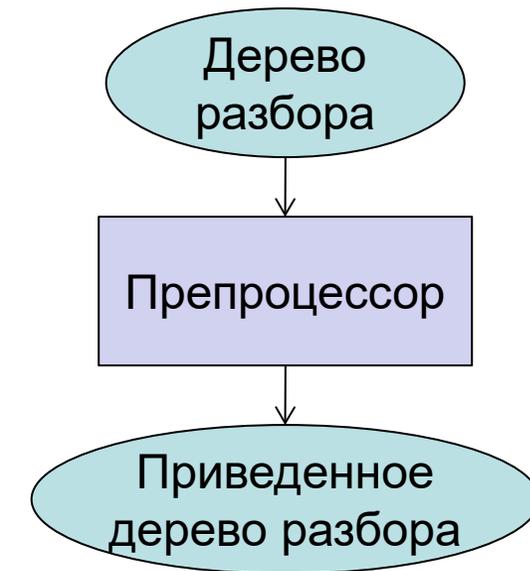
Фазы разбор запроса

1. Синтаксический анализ запроса
(выполняется *синтаксическим анализатором*)
2. Разрешение представлений и верификация запроса
(выполняется *препроцессором*)



Преппроцессор

- Разрешение представлений
- Квалификация и верификация запроса



Разрешение представлений

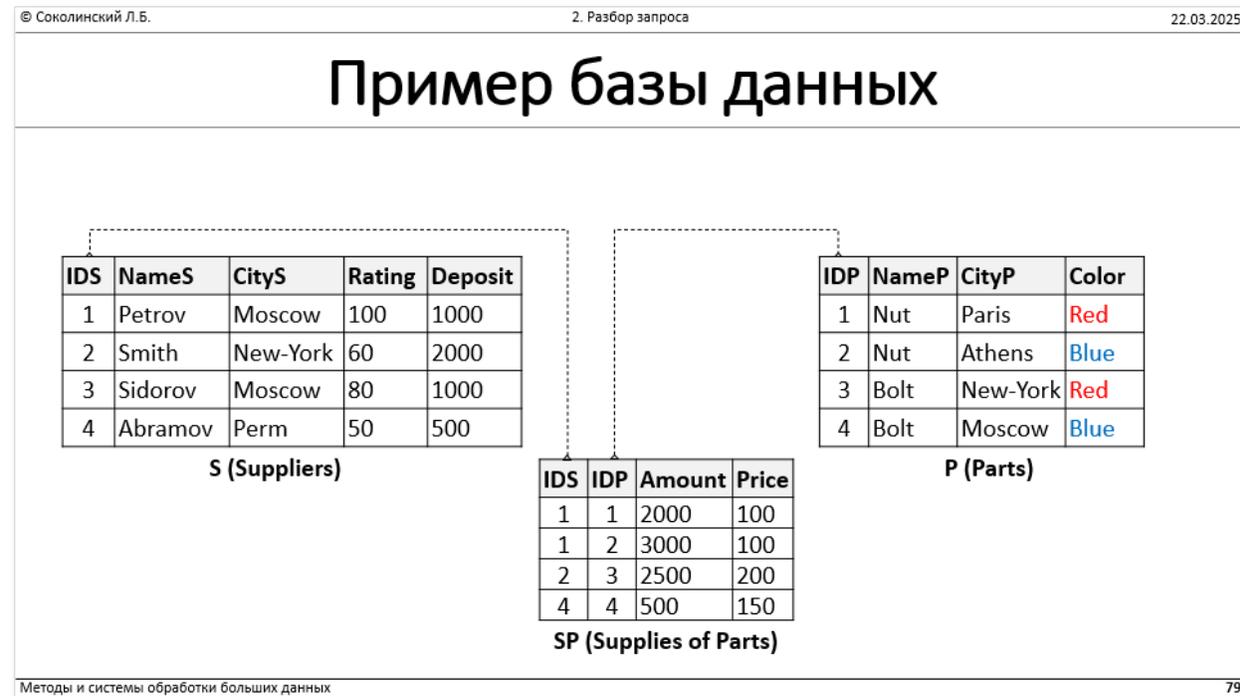
/ Коды Красных Деталей */*

```
CREATE VIEW RedParts(IDRP)  
AS SELECT IDP  
FROM P  
WHERE Color='Red';
```

...

/ Коды поставщиков, поставляющих хотя бы одну красную деталь */*

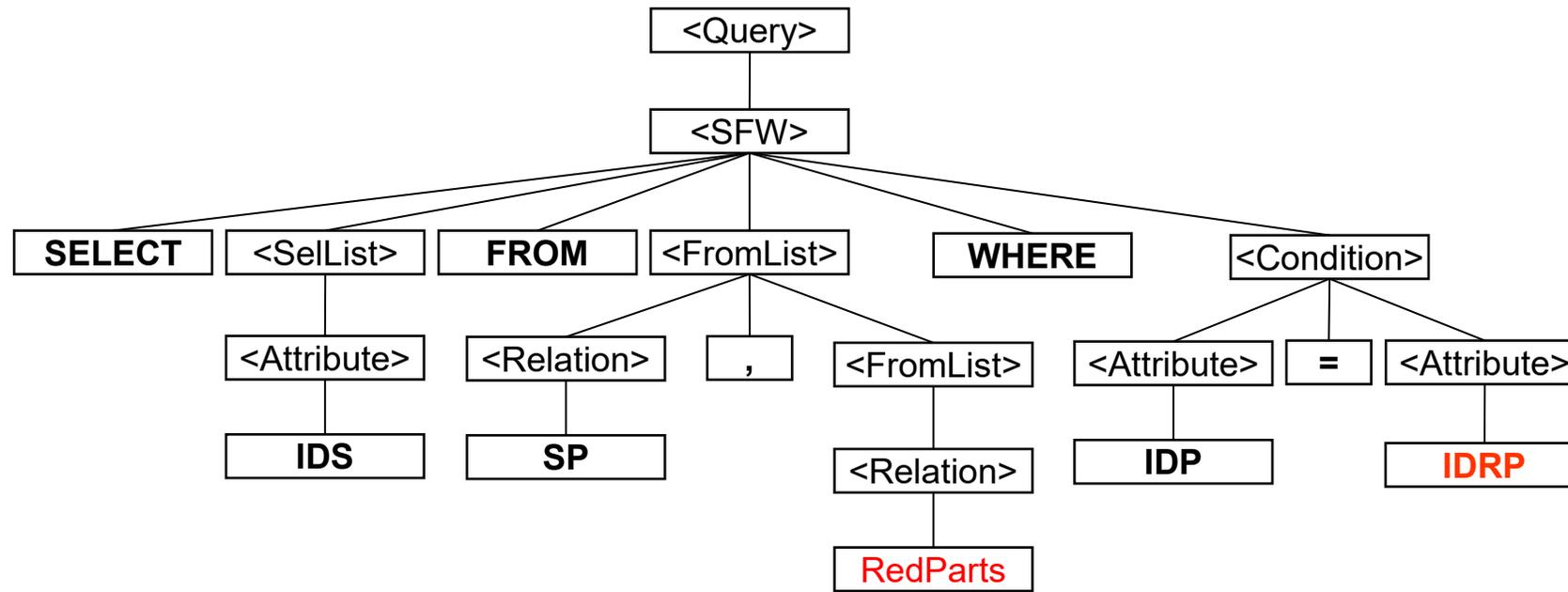
```
SELECT IDS  
FROM SP, RedParts  
WHERE IDP= IDRP;
```



Дерево разбора основного запроса

/ Коды поставщиков, поставляющих хотя бы одну красную деталь */*

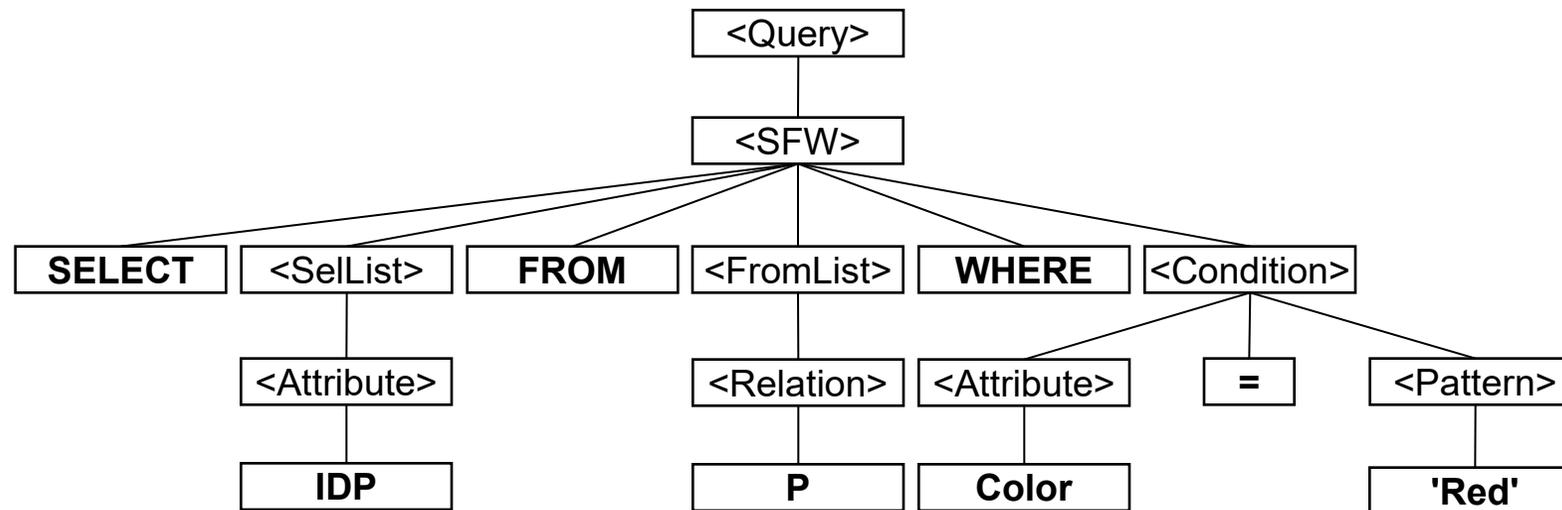
SELECT **IDS**
FROM **SP, RedParts**
WHERE **IDP= IDR**;



Дерево разбора для VIEW

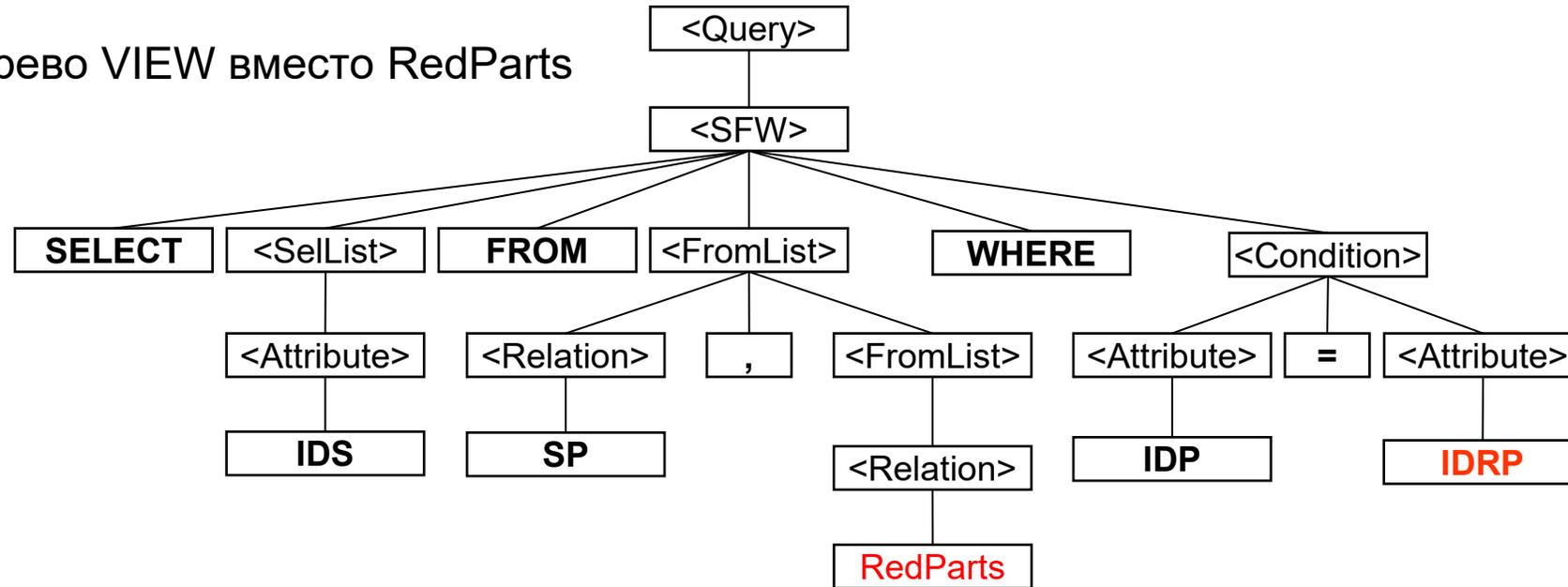
/ Коды Красных Деталей */*

```
CREATE VIEW IDRП (IDRП)  
AS SELECT IDRП  
FROM P  
WHERE Color='Red';
```



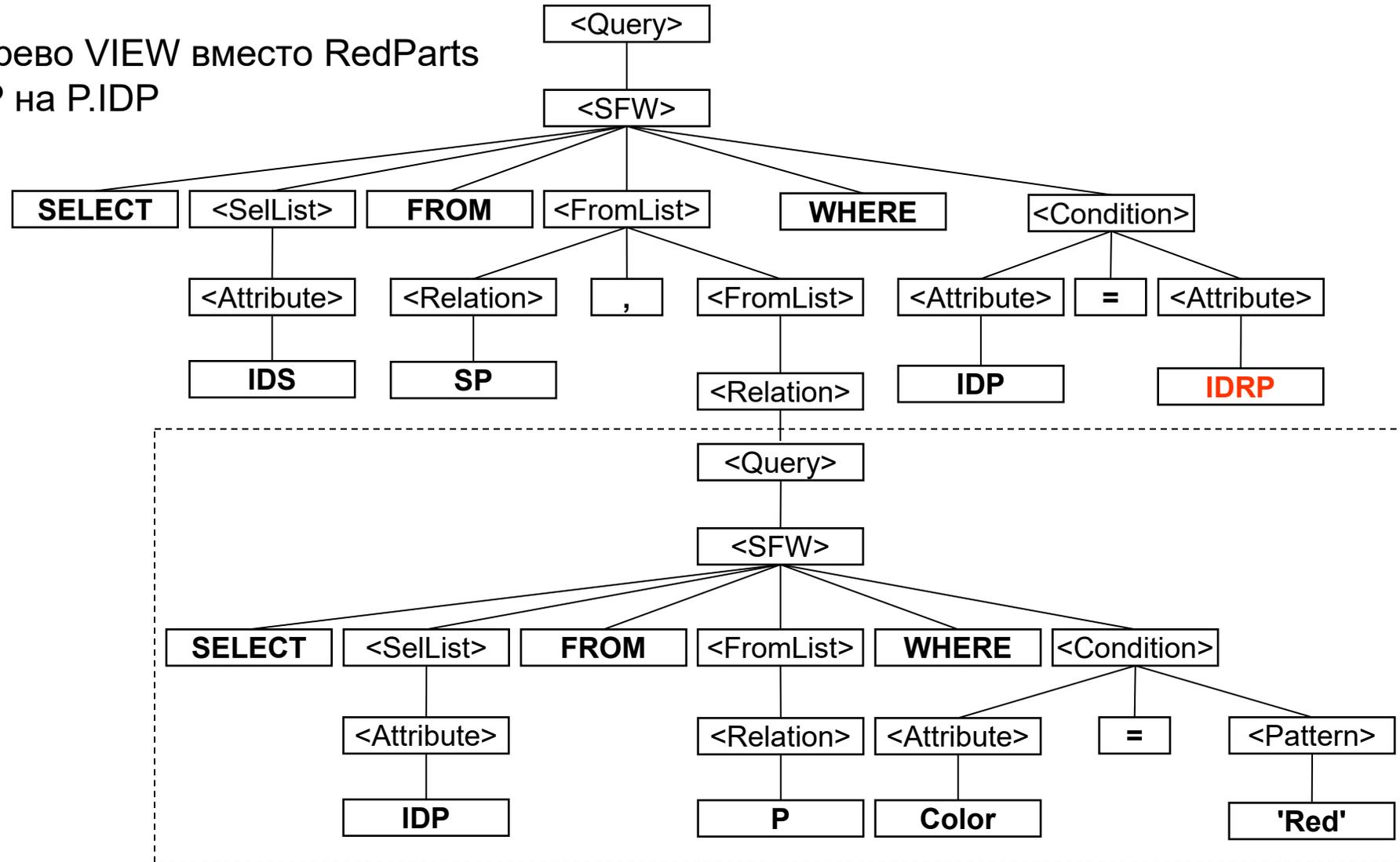
Преобразование

1. Подставить дерево VIEW вместо RedParts



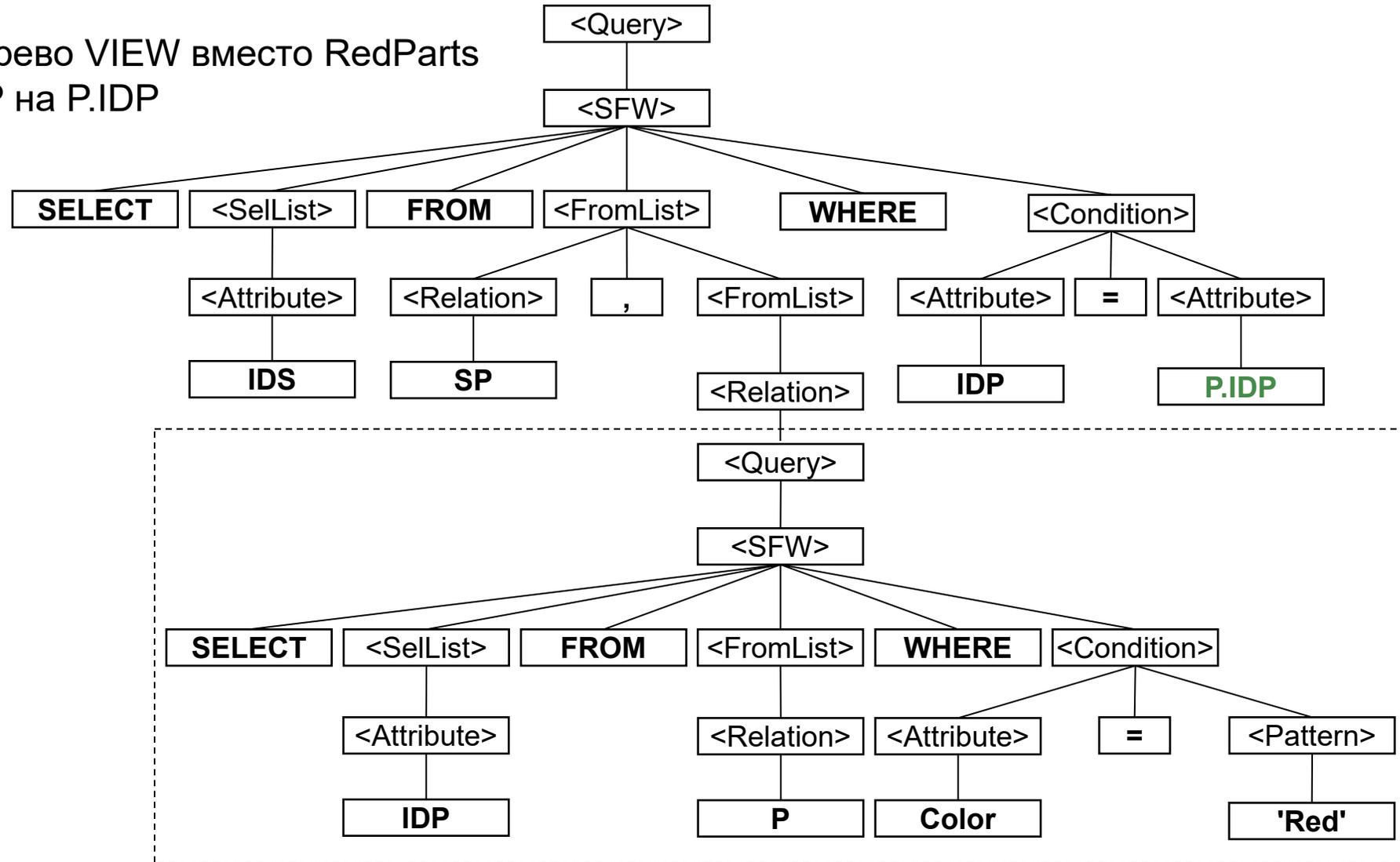
Преобразование

1. Подставить дерево VIEW вместо RedParts
2. Заменить IDRП на P.IDP



Преобразование

1. Подставить дерево VIEW вместо RedParts
2. Заменить IDRП на P.IDP



Верификация запроса

1. Проверка используемых отношений

Каждое отношение, упоминаемое в предложении FROM, должно быть отношением в схеме базы данных, над которой выполняется запрос

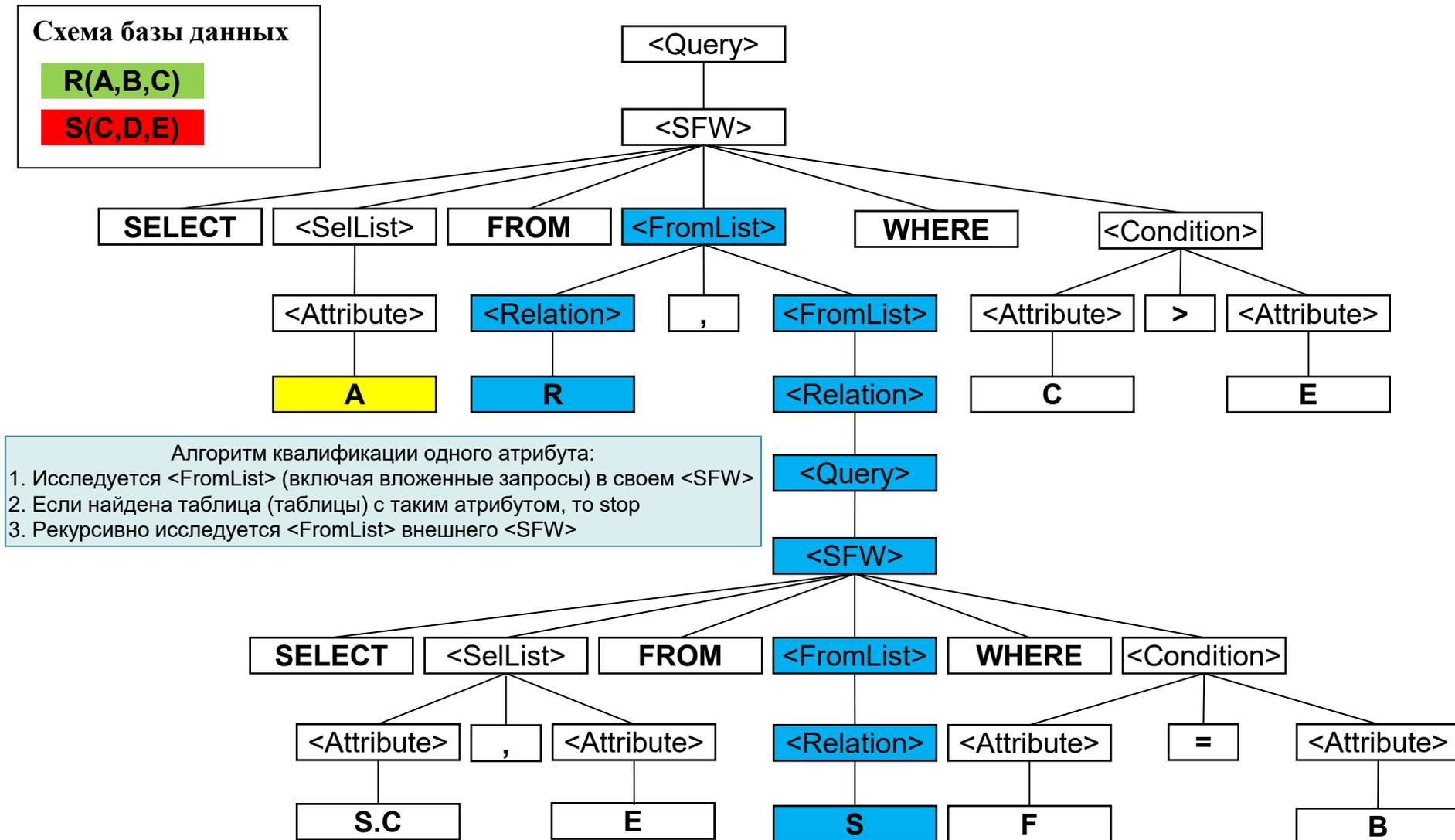
2. Проверка используемых атрибутов и их квалификация

Атрибут, входящий в дерево разбора, должен быть атрибутом некоторого отношения

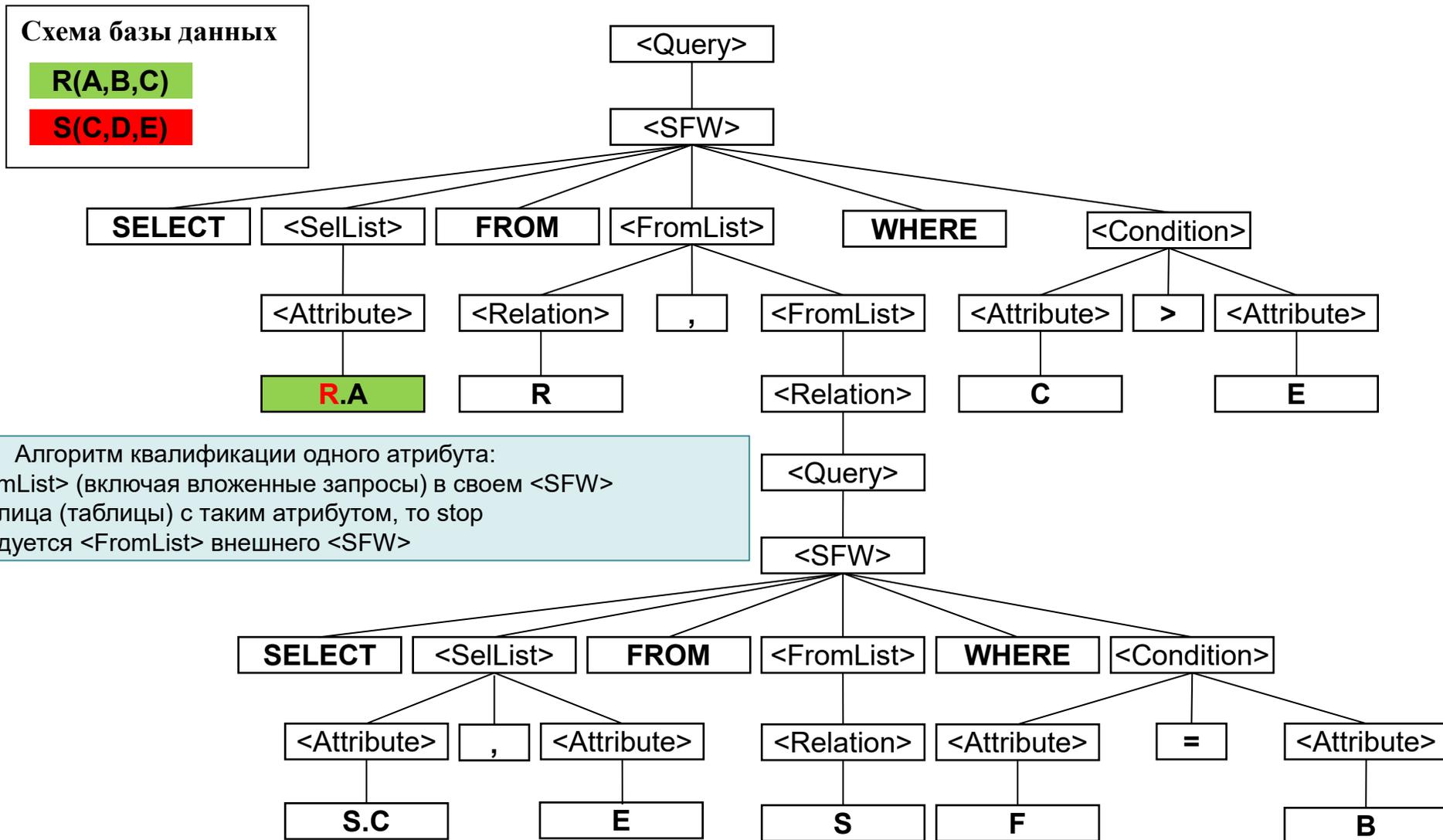
3. Проверка соответствия типов

Каждый атрибут должен иметь тип, соответствующий его использованию

Проверка и квалификация атрибутов



Проверка и квалификация атрибутов

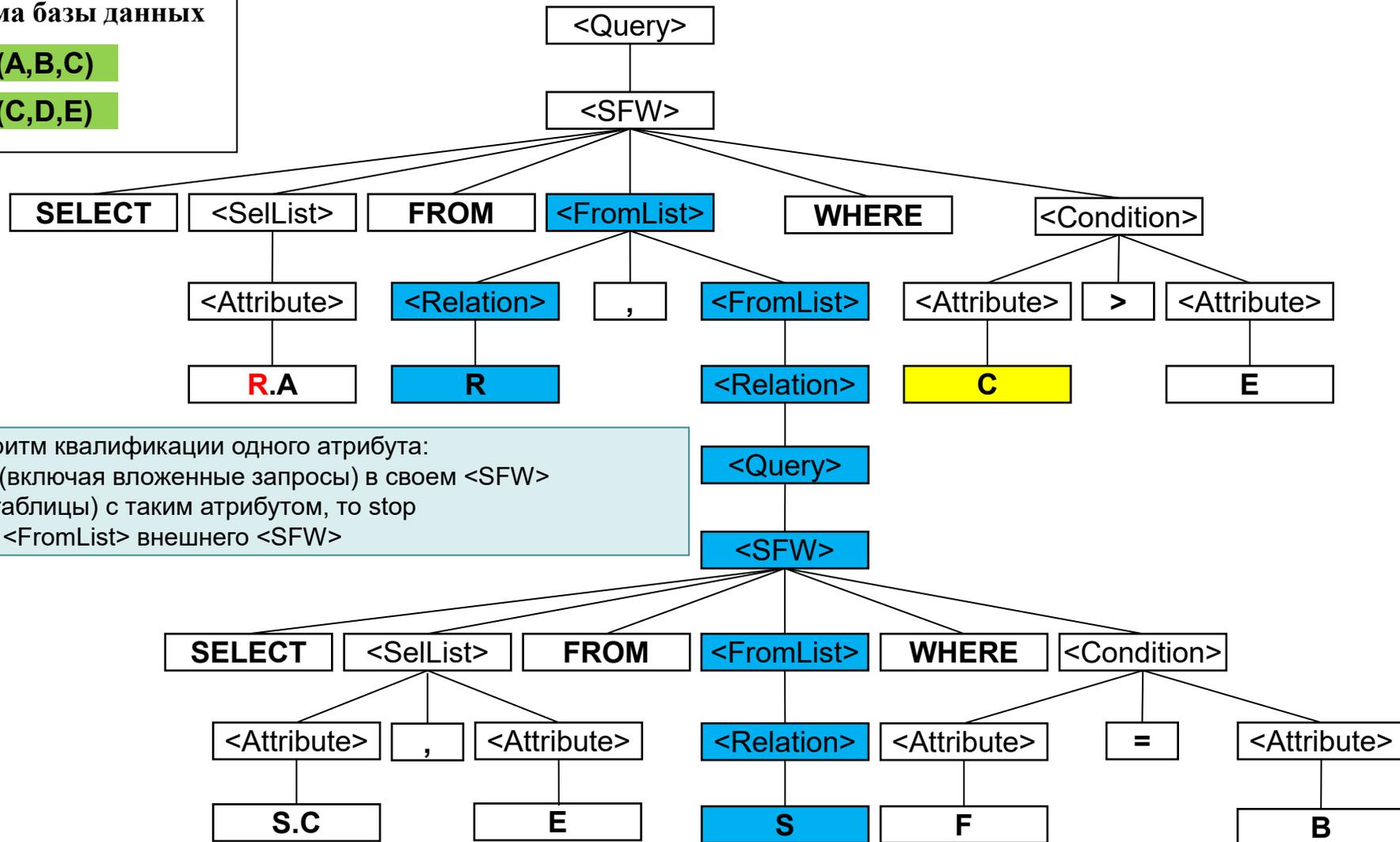


Проверка и квалификация атрибутов

Схема базы данных

R(A,B,C)

S(C,D,E)



Алгоритм квалификации одного атрибута:

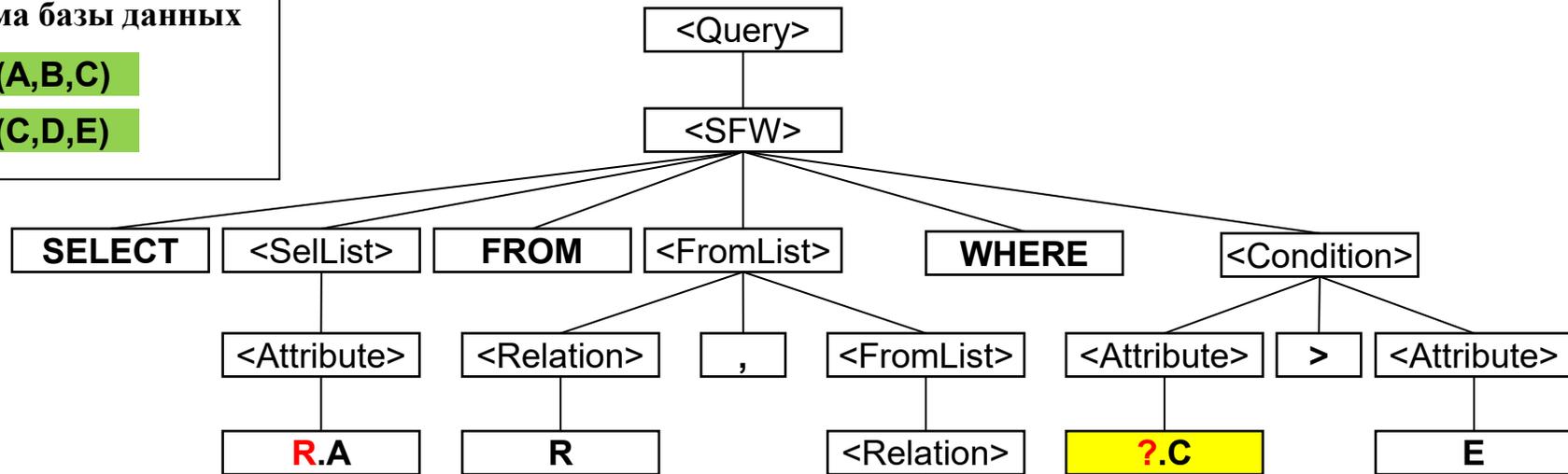
1. Исследуется `<FromList>` (включая вложенные запросы) в своем `<SFW>`
2. Если найдена таблица (таблицы) с таким атрибутом, то stop
3. Рекурсивно исследуется `<FromList>` внешнего `<SFW>`

Проверка и квалификация атрибутов

Схема базы данных

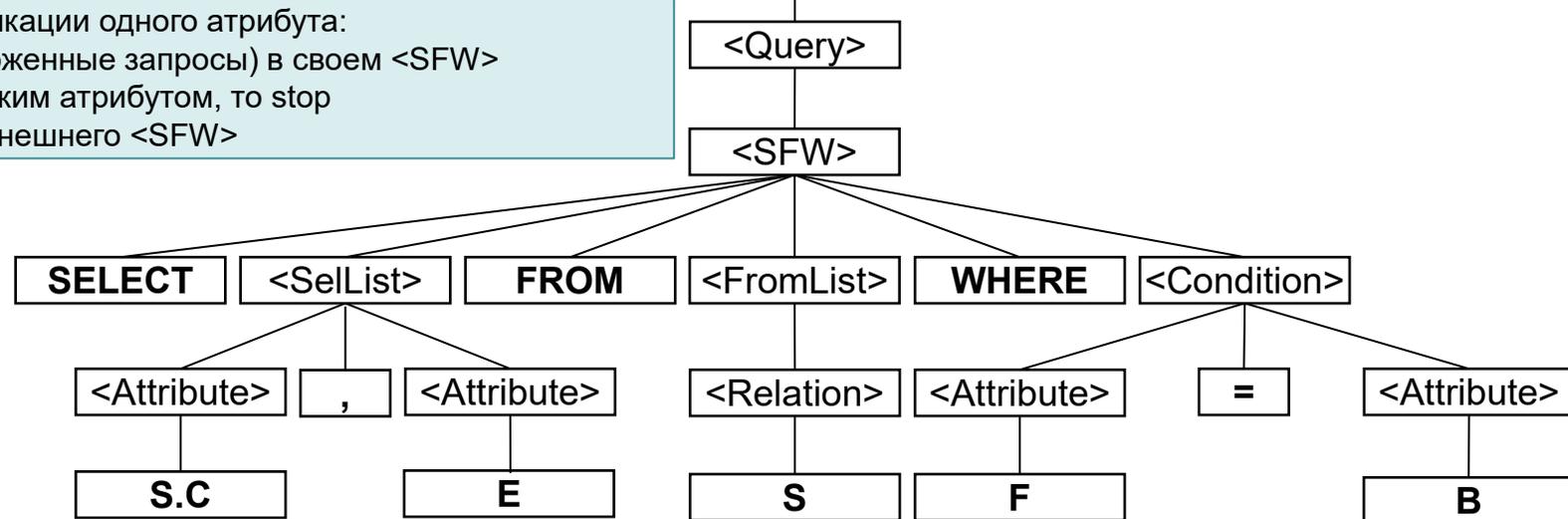
R(A,B,C)

S(C,D,E)



Алгоритм квалификации одного атрибута:

1. Исследуется <FromList> (включая вложенные запросы) в своем <SFW>
2. Если найдена таблица (таблицы) с таким атрибутом, то stop
3. Рекурсивно исследуется <FromList> внешнего <SFW>

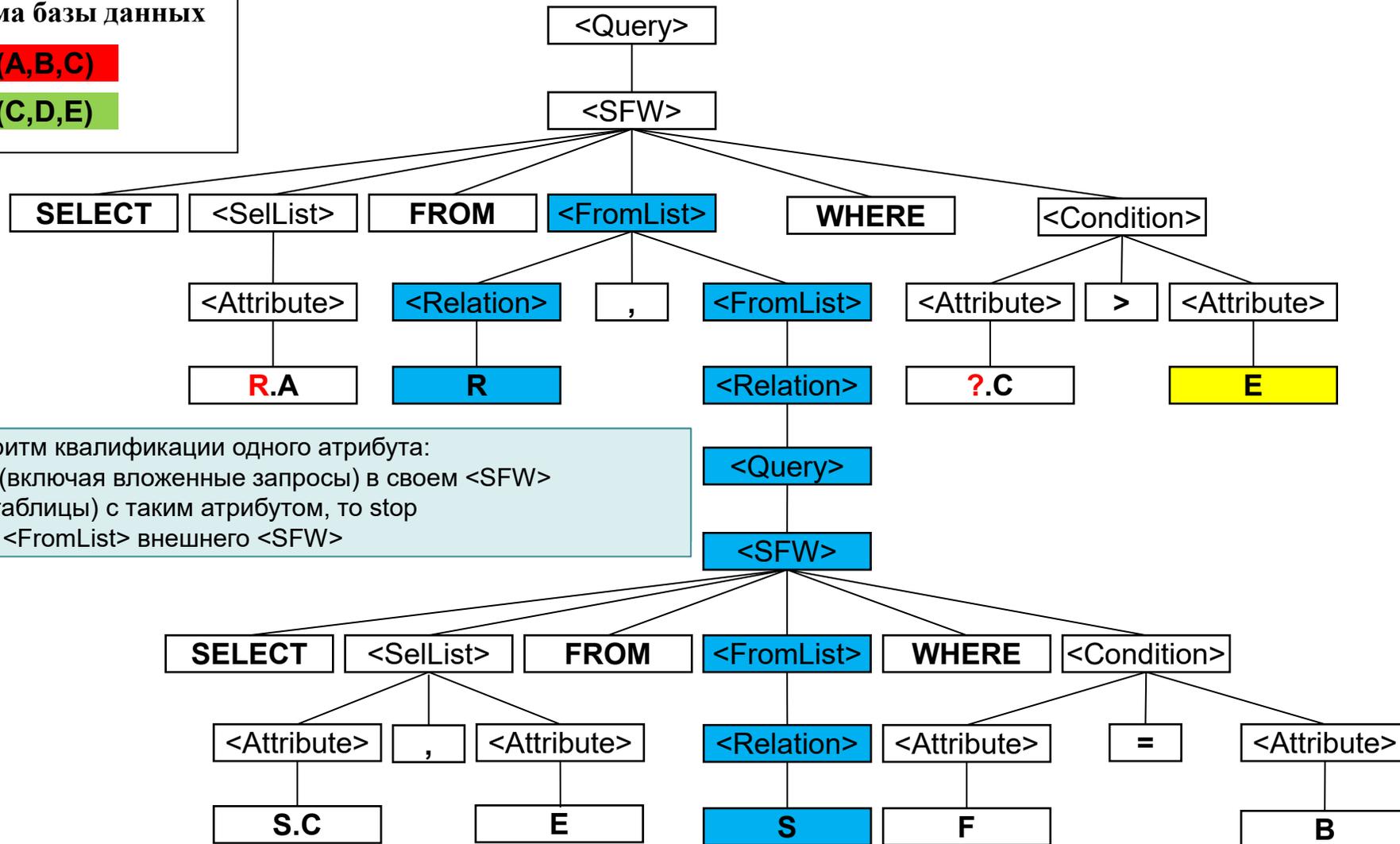


Проверка и квалификация атрибутов

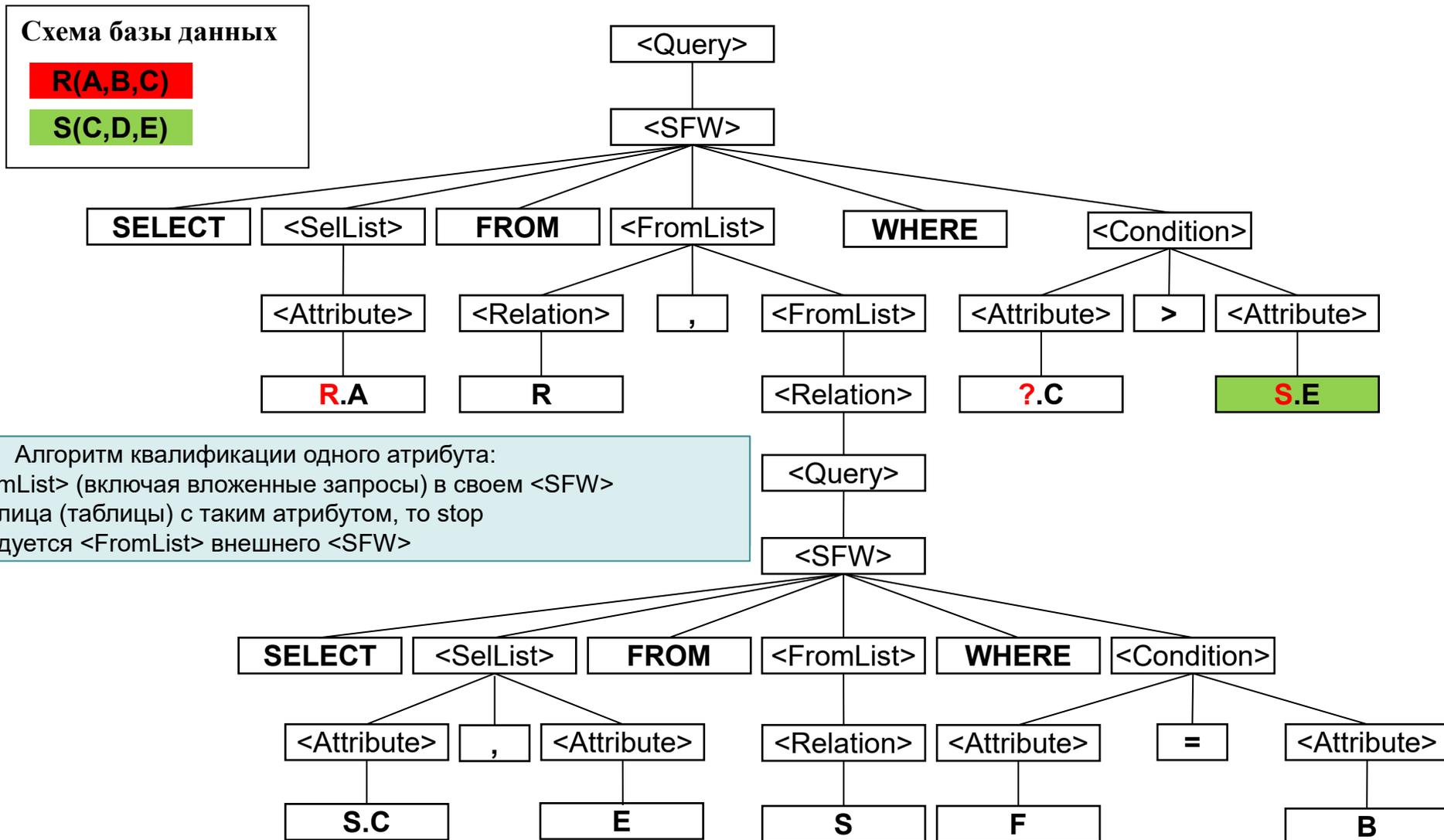
Схема базы данных

R(A,B,C)

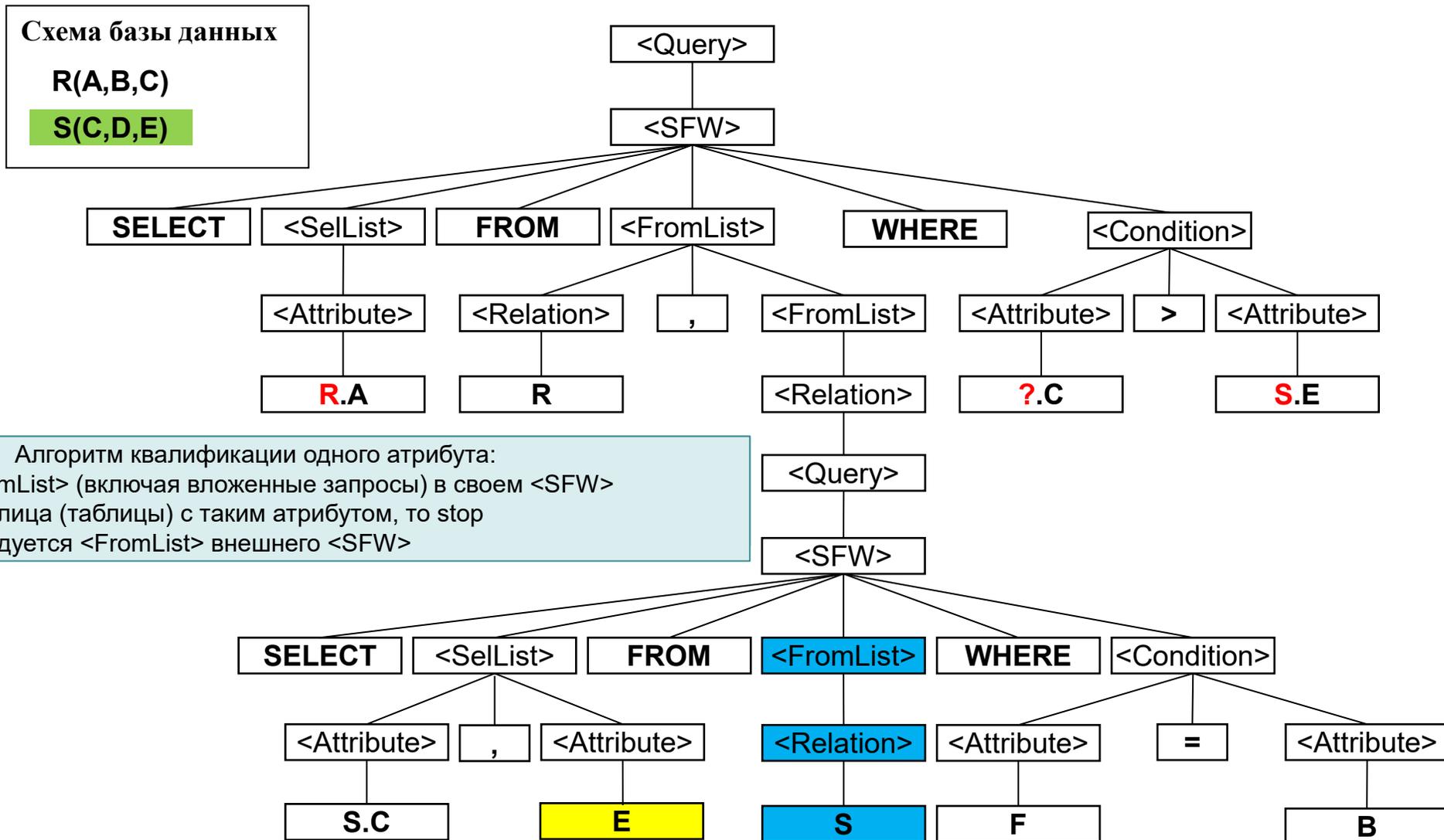
S(C,D,E)



Проверка и квалификация атрибутов



Проверка и квалификация атрибутов

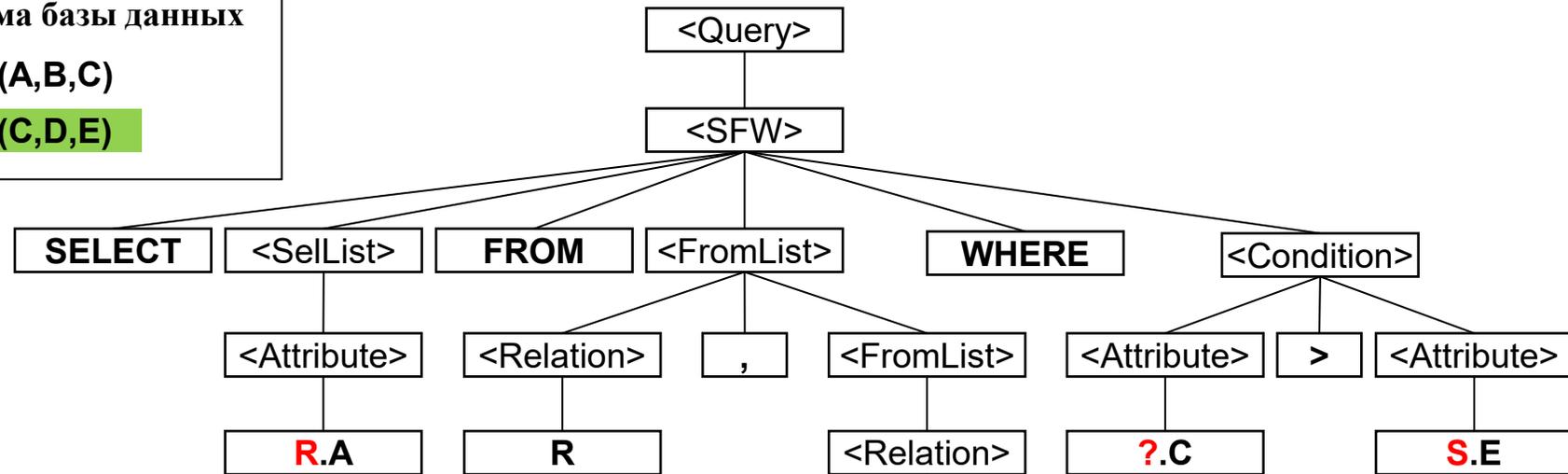


Проверка и квалификация атрибутов

Схема базы данных

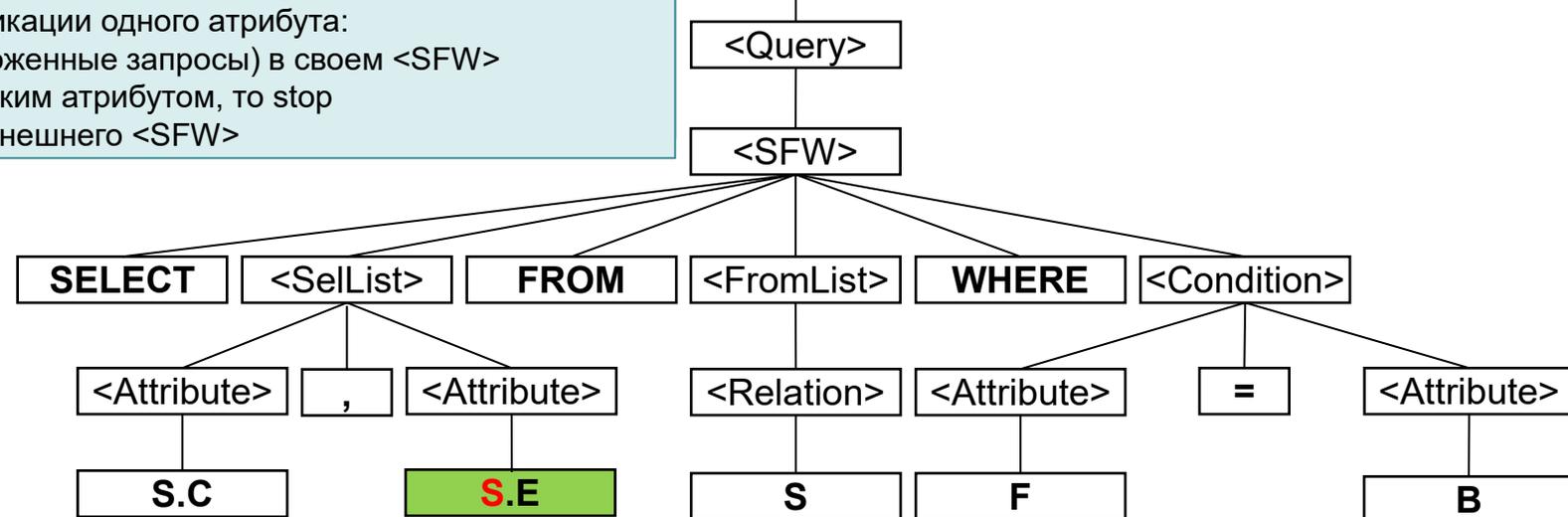
R(A,B,C)

S(C,D,E)



Алгоритм квалификации одного атрибута:

1. Исследуется <FromList> (включая вложенные запросы) в своем <SFW>
2. Если найдена таблица (таблицы) с таким атрибутом, то stop
3. Рекурсивно исследуется <FromList> внешнего <SFW>

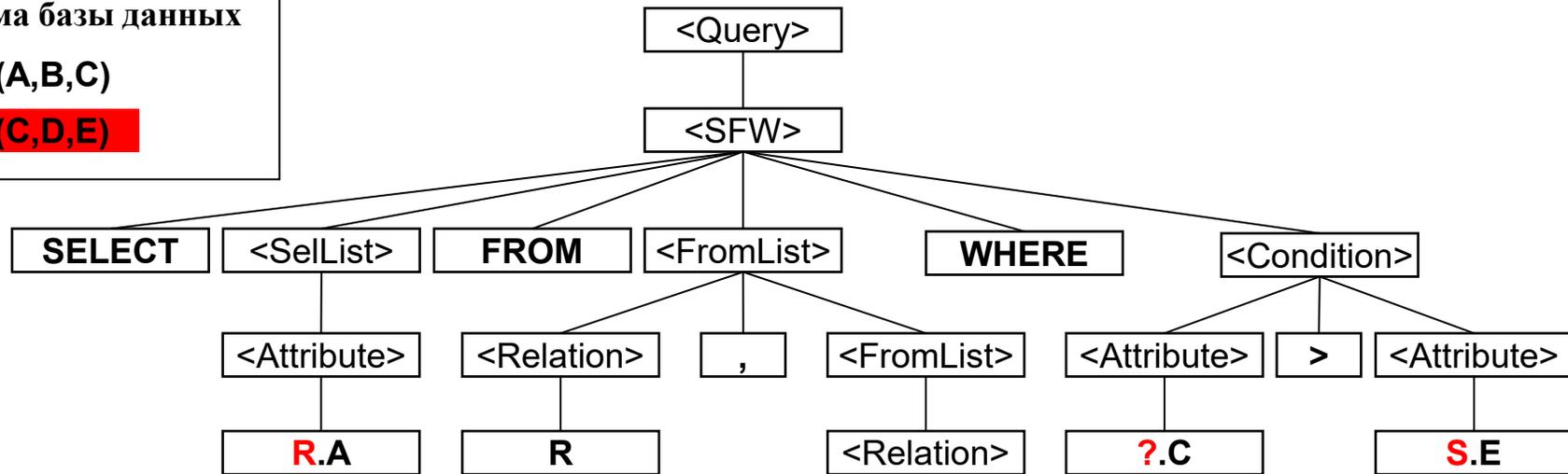


Проверка и квалификация атрибутов

Схема базы данных

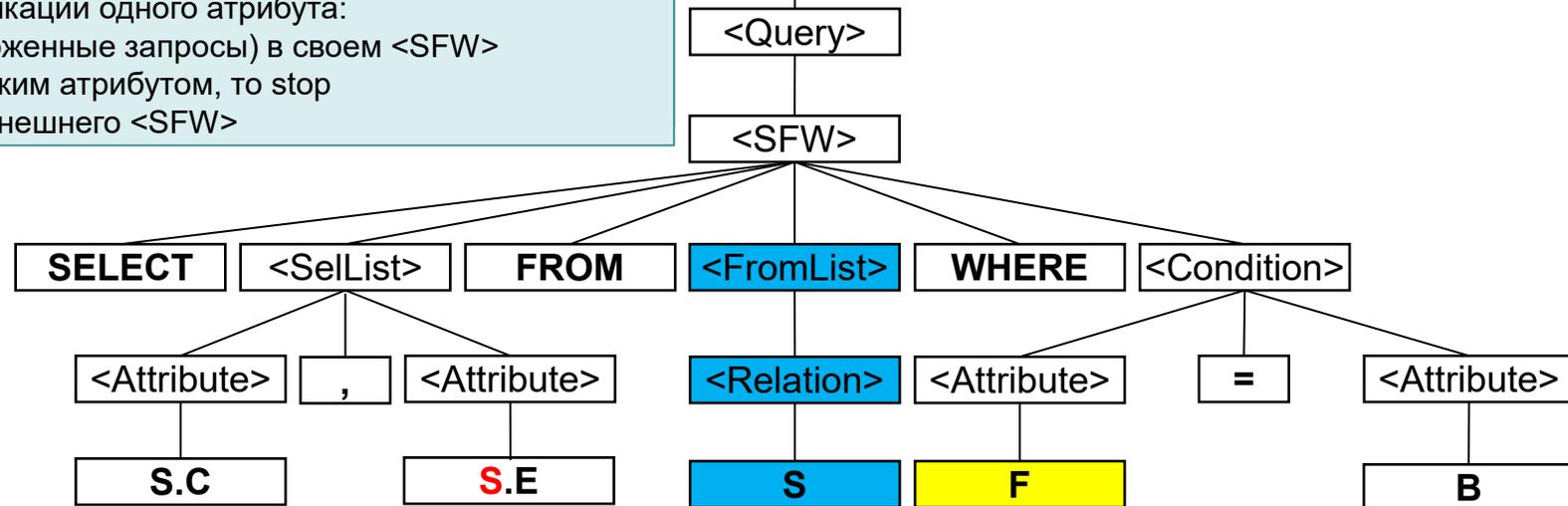
R(A,B,C)

S(C,D,E)

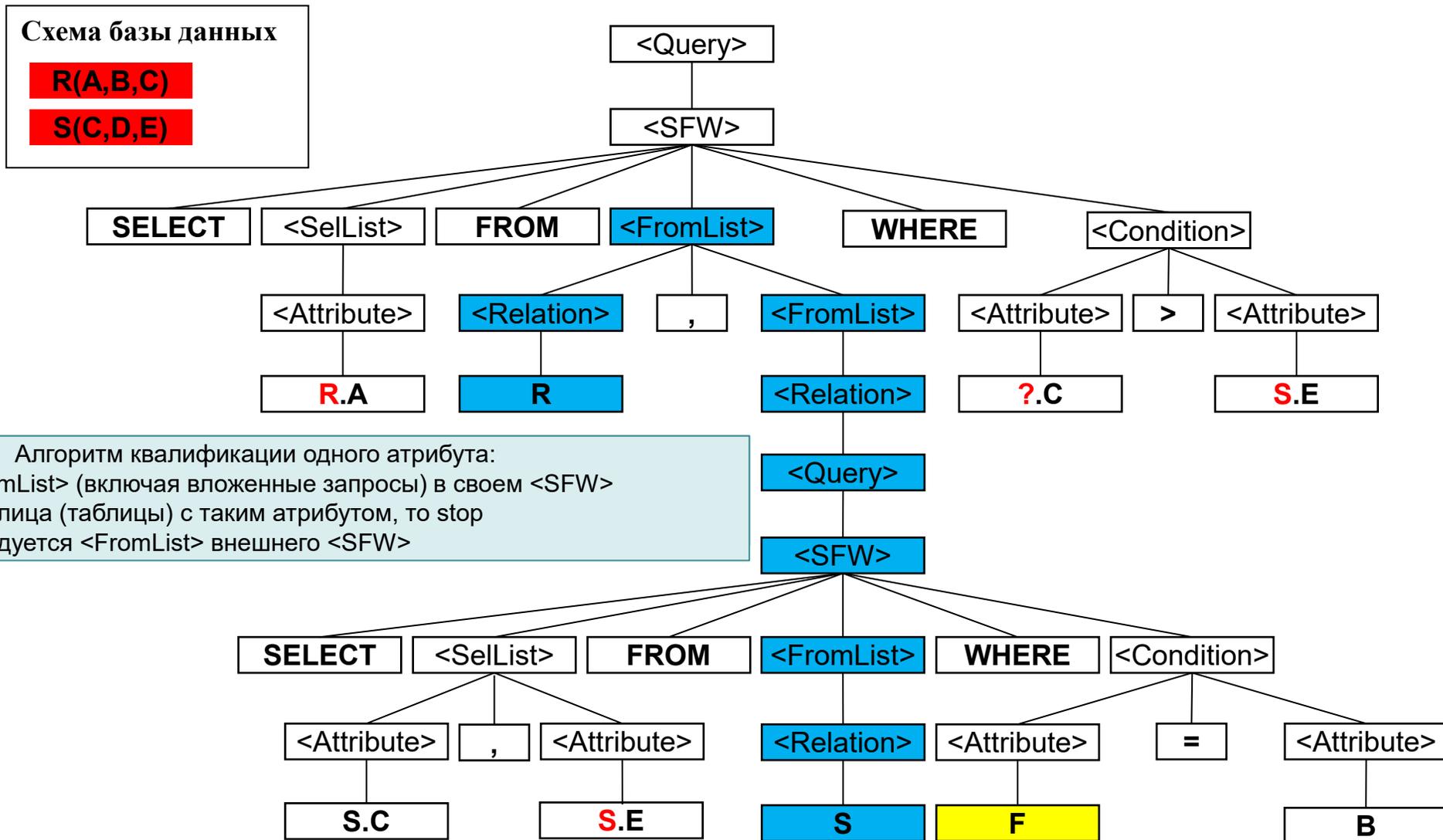


Алгоритм квалификации одного атрибута:

1. Исследуется <FromList> (включая вложенные запросы) в своем <SFW>
2. Если найдена таблица (таблицы) с таким атрибутом, то stop
3. Рекурсивно исследуется <FromList> внешнего <SFW>



Проверка и квалификация атрибутов

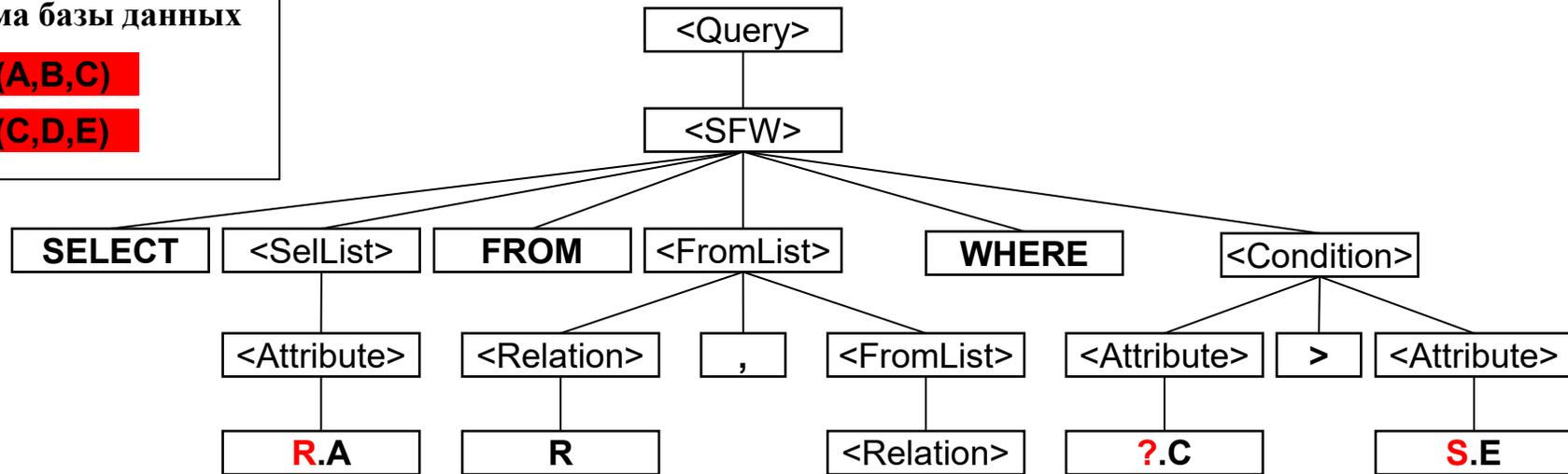


Проверка и квалификация атрибутов

Схема базы данных

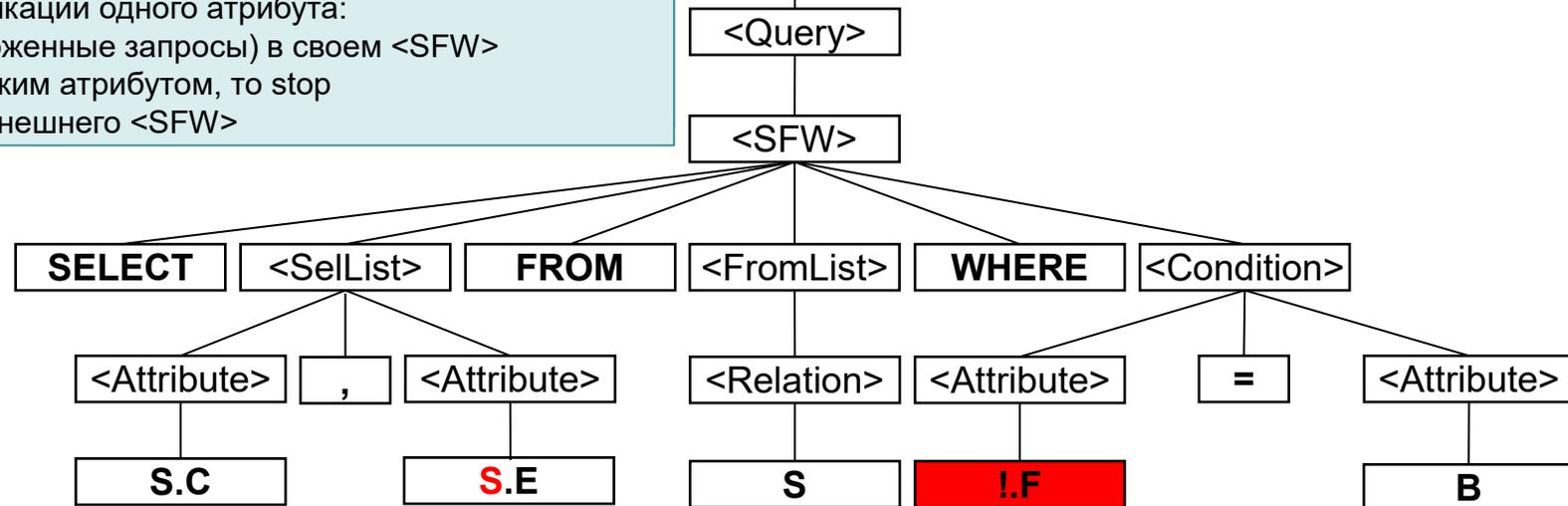
R(A,B,C)

S(C,D,E)



Алгоритм квалификации одного атрибута:

1. Исследуется `<FromList>` (включая вложенные запросы) в своем `<SFW>`
2. Если найдена таблица (таблицы) с таким атрибутом, то stop
3. Рекурсивно исследуется `<FromList>` внешнего `<SFW>`

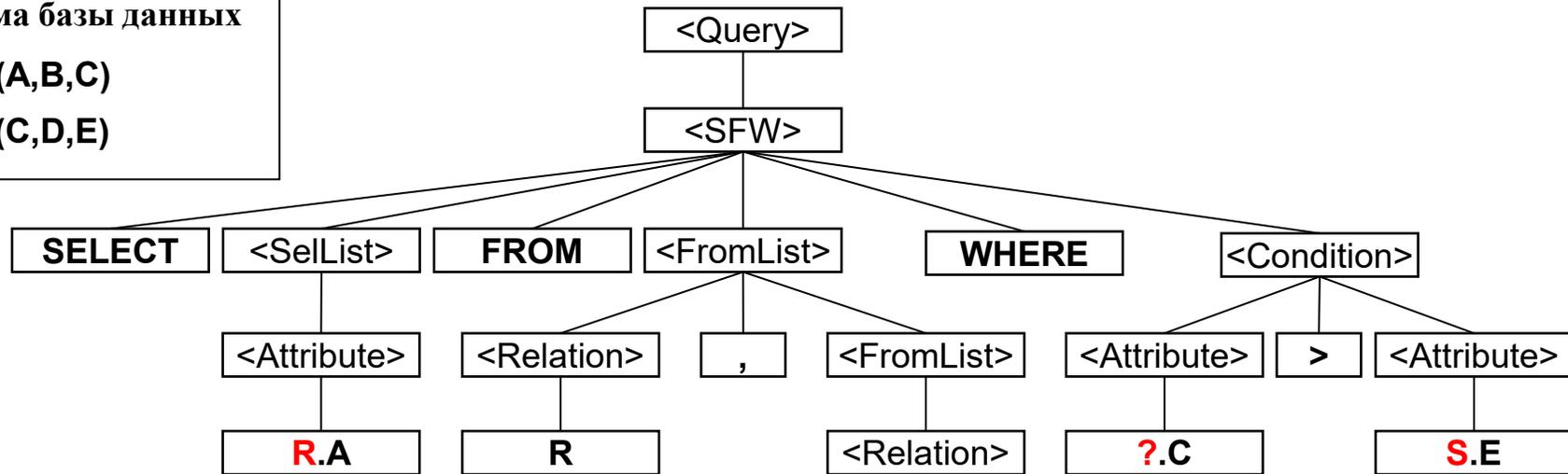


Проверка и квалификация атрибутов

Схема базы данных

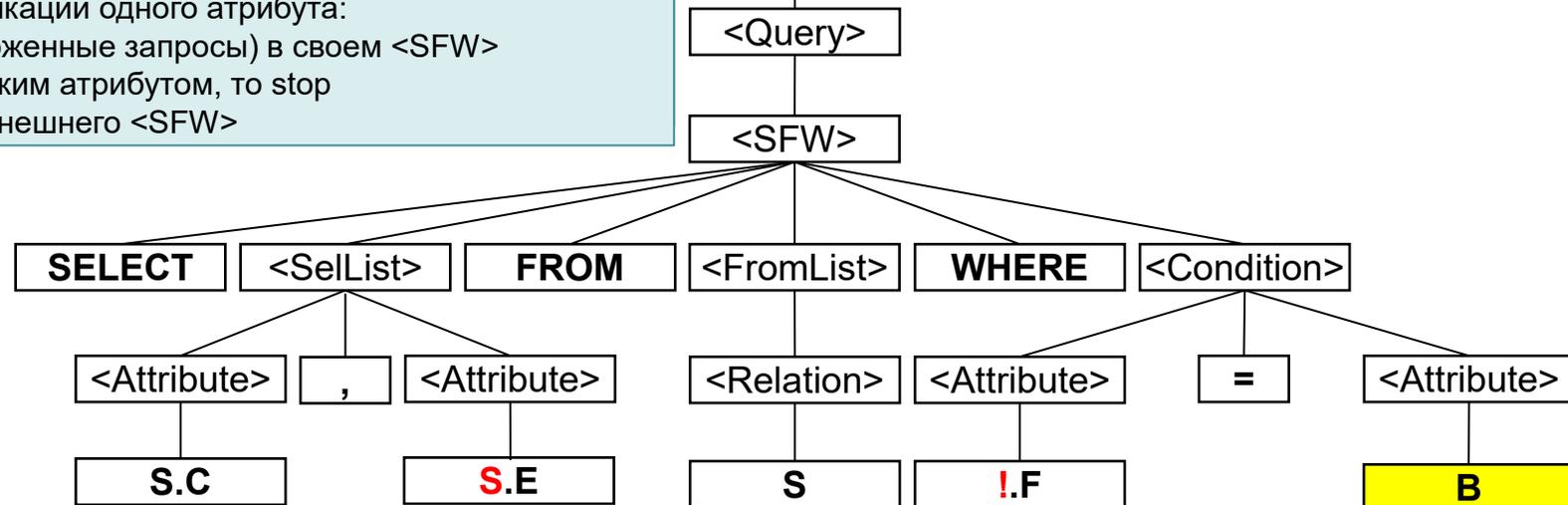
R(A,B,C)

S(C,D,E)

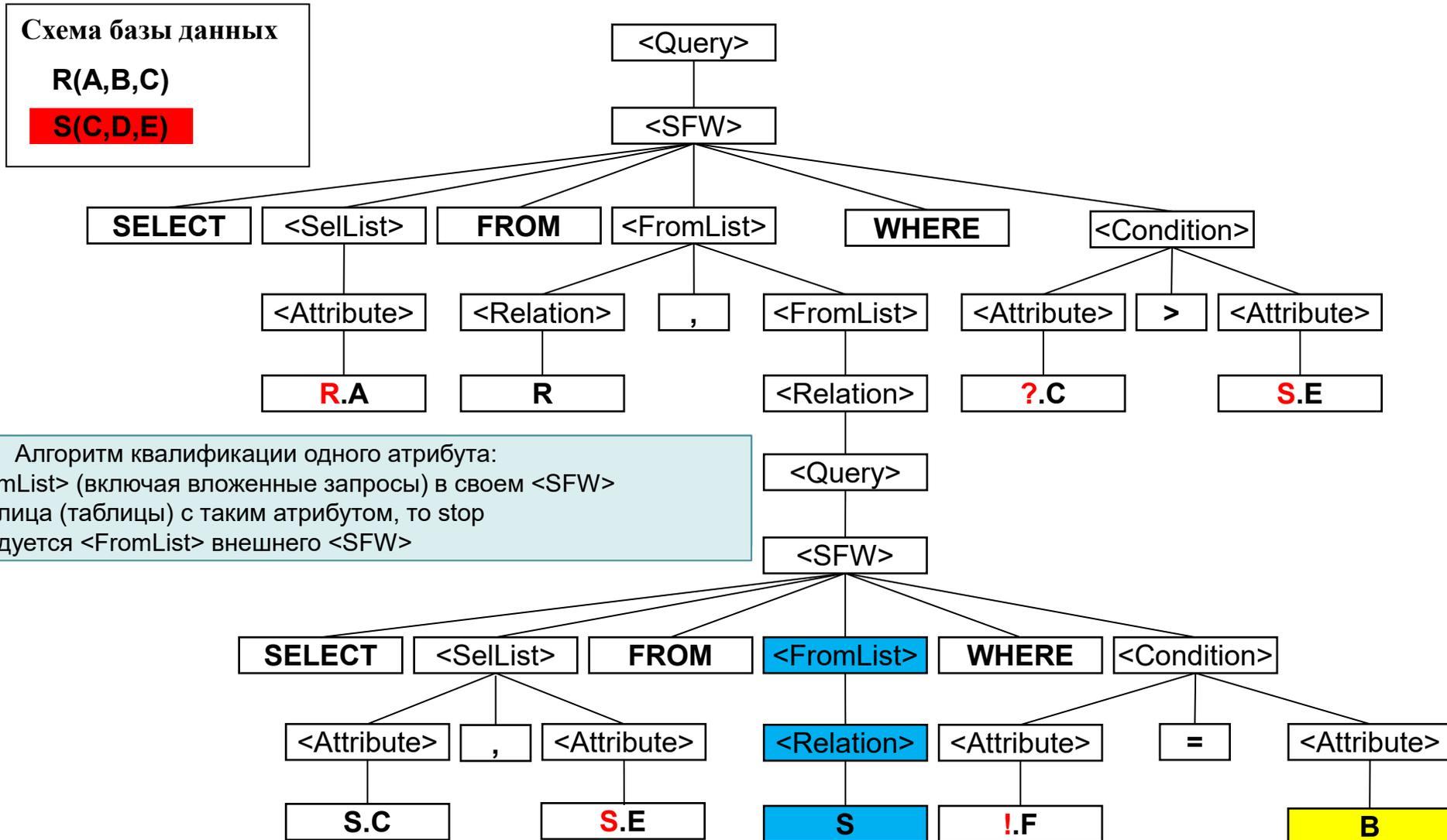


Алгоритм квалификации одного атрибута:

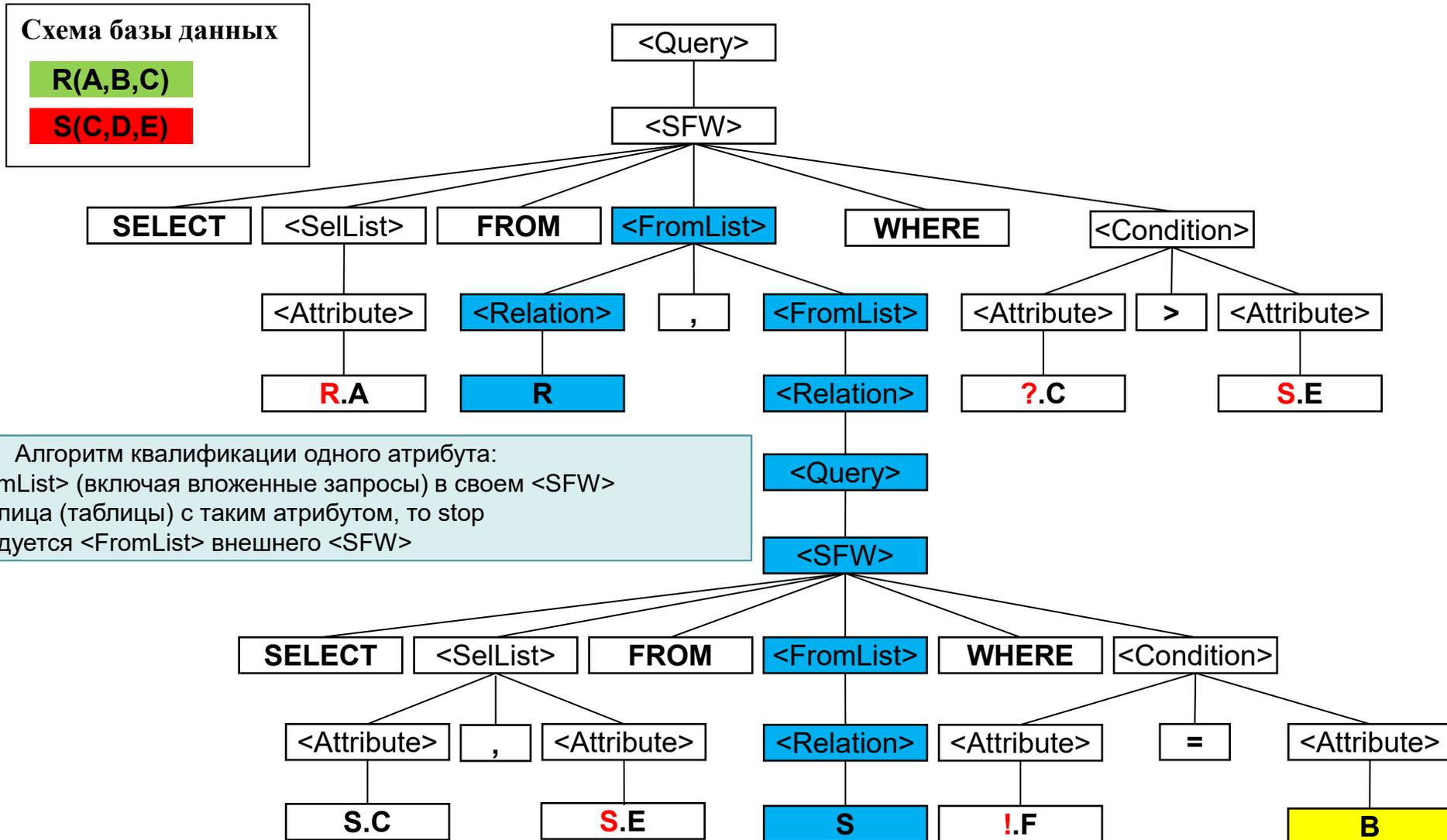
1. Исследуется <FromList> (включая вложенные запросы) в своем <SFW>
2. Если найдена таблица (таблицы) с таким атрибутом, то stop
3. Рекурсивно исследуется <FromList> внешнего <SFW>



Проверка и квалификация атрибутов



Проверка и квалификация атрибутов

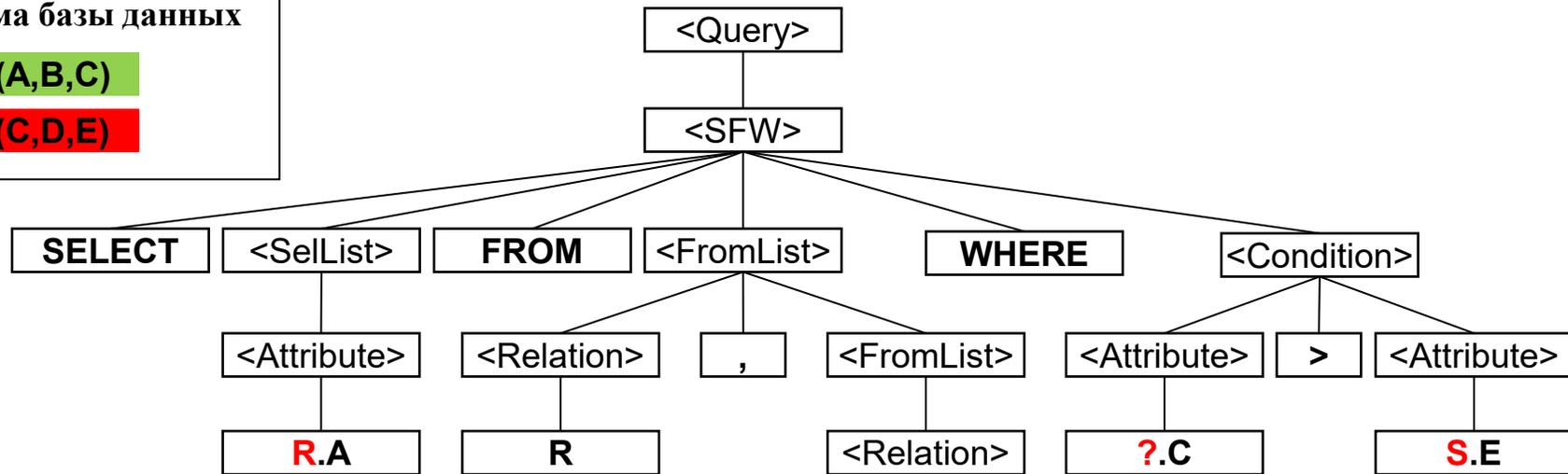


Проверка и квалификация атрибутов

Схема базы данных

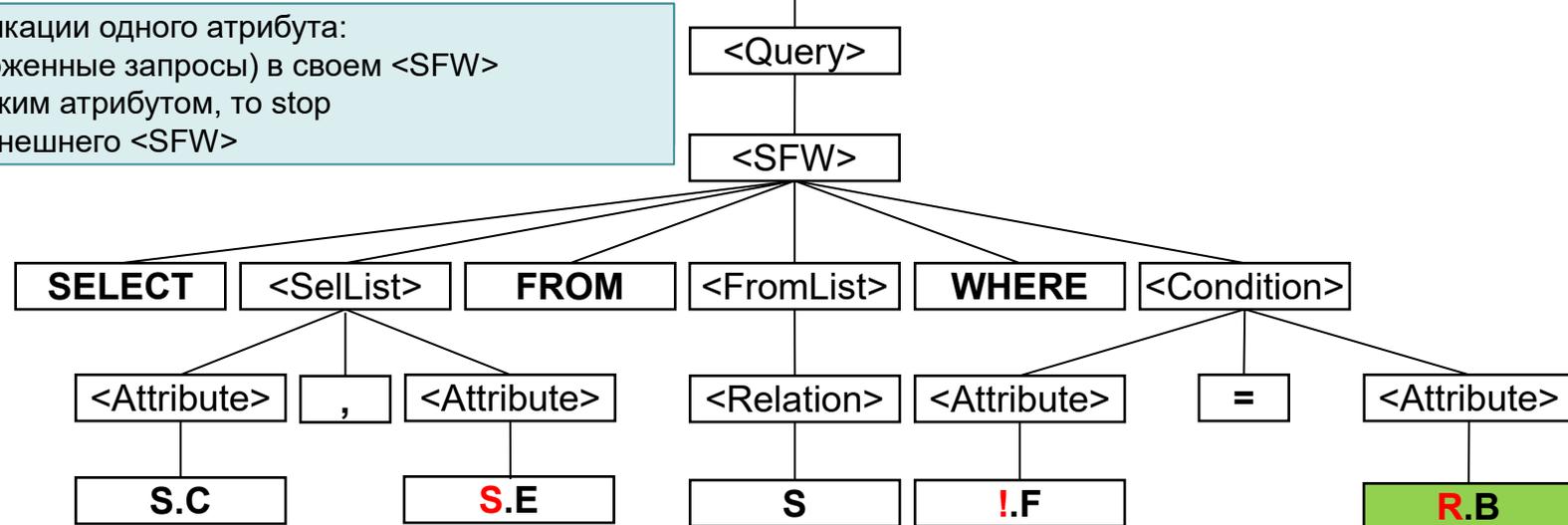
R(A,B,C)

S(C,D,E)



Алгоритм квалификации одного атрибута:

1. Исследуется <FromList> (включая вложенные запросы) в своем <SFW>
2. Если найдена таблица (таблицы) с таким атрибутом, то stop
3. Рекурсивно исследуется <FromList> внешнего <SFW>

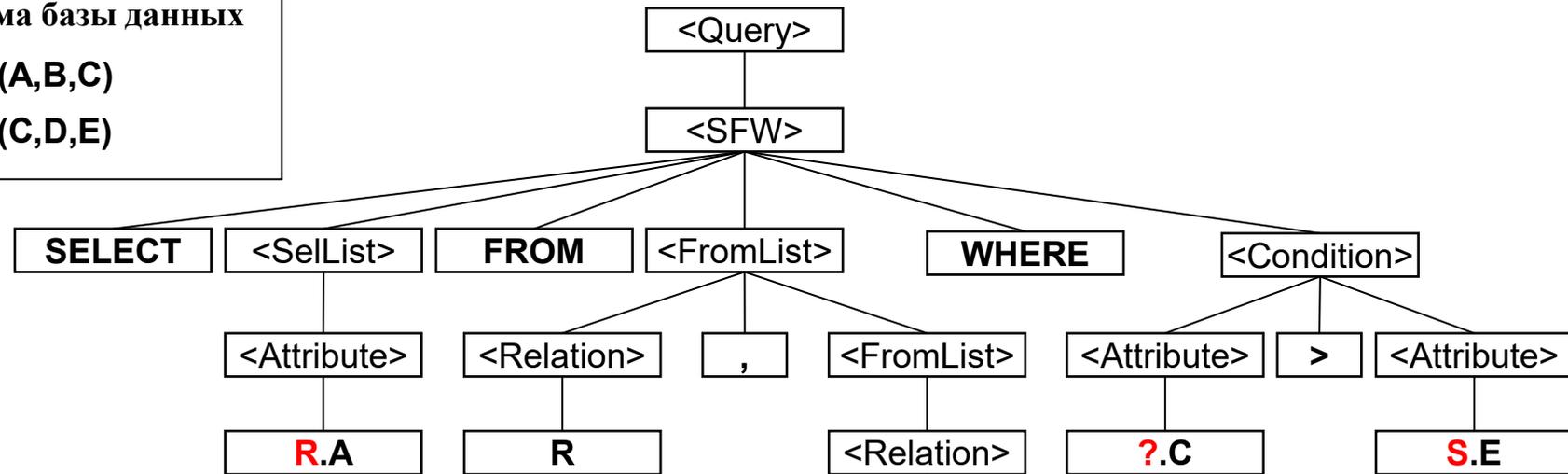


Проверка и квалификация атрибутов

Схема базы данных

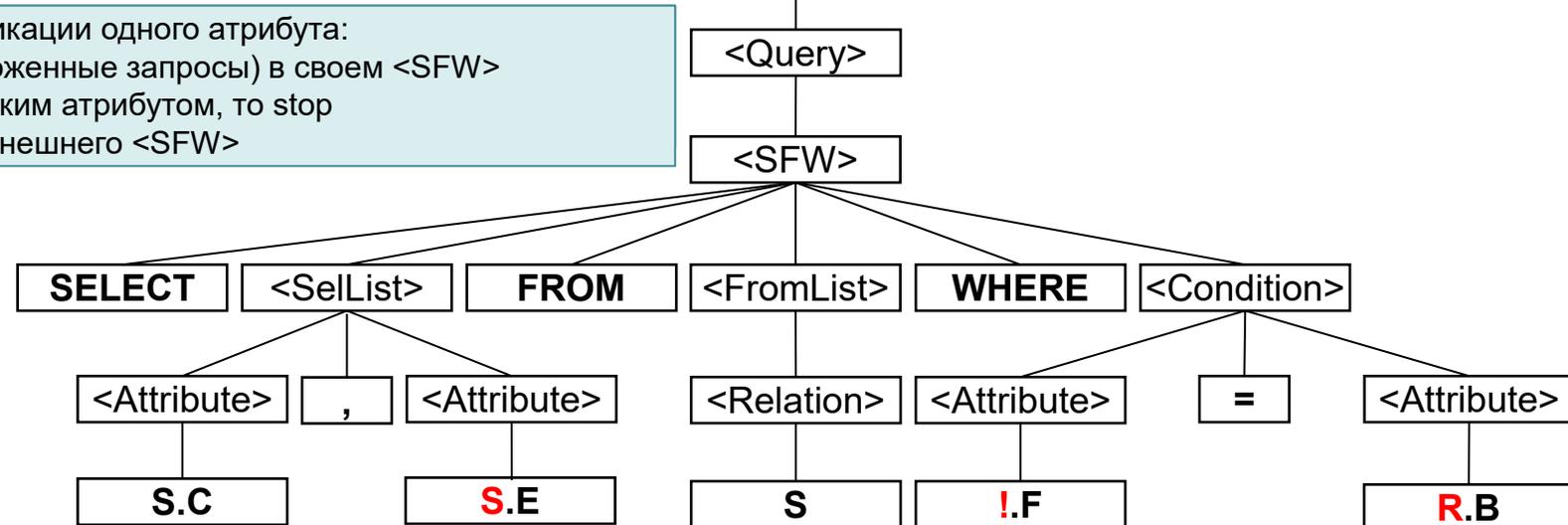
R(A,B,C)

S(C,D,E)



Алгоритм квалификации одного атрибута:

1. Исследуется `<FromList>` (включая вложенные запросы) в своем `<SFW>`
2. Если найдена таблица (таблицы) с таким атрибутом, то stop
3. Рекурсивно исследуется `<FromList>` внешнего `<SFW>`



Конец лекции 2

Вспомогательные слайды

Пример базы данных

IDS	NameS	CityS	Rating	Deposit
1	Petrov	Moscow	100	1000
2	Smith	New-York	60	2000
3	Sidorov	Moscow	80	1000
4	Abramov	Perm	50	500

S (Suppliers)

IDP	NameP	CityP	Color
1	Nut	Paris	Red
2	Nut	Athens	Blue
3	Bolt	New-York	Red
4	Bolt	Moscow	Blue

P (Parts)

IDS	IDP	Amount	Price
1	1	2000	100
1	2	3000	100
2	3	2500	200
4	4	500	150

SP (Supplies of Parts)