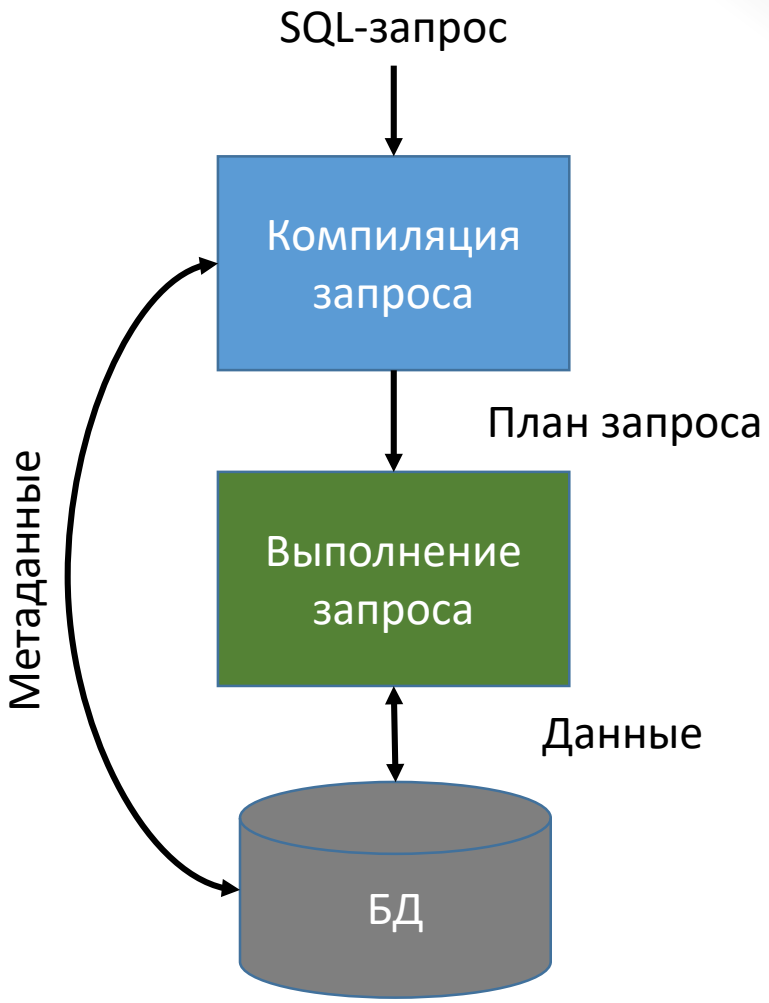
Оптимизация запросов

Технологии баз данных. Лекция 9

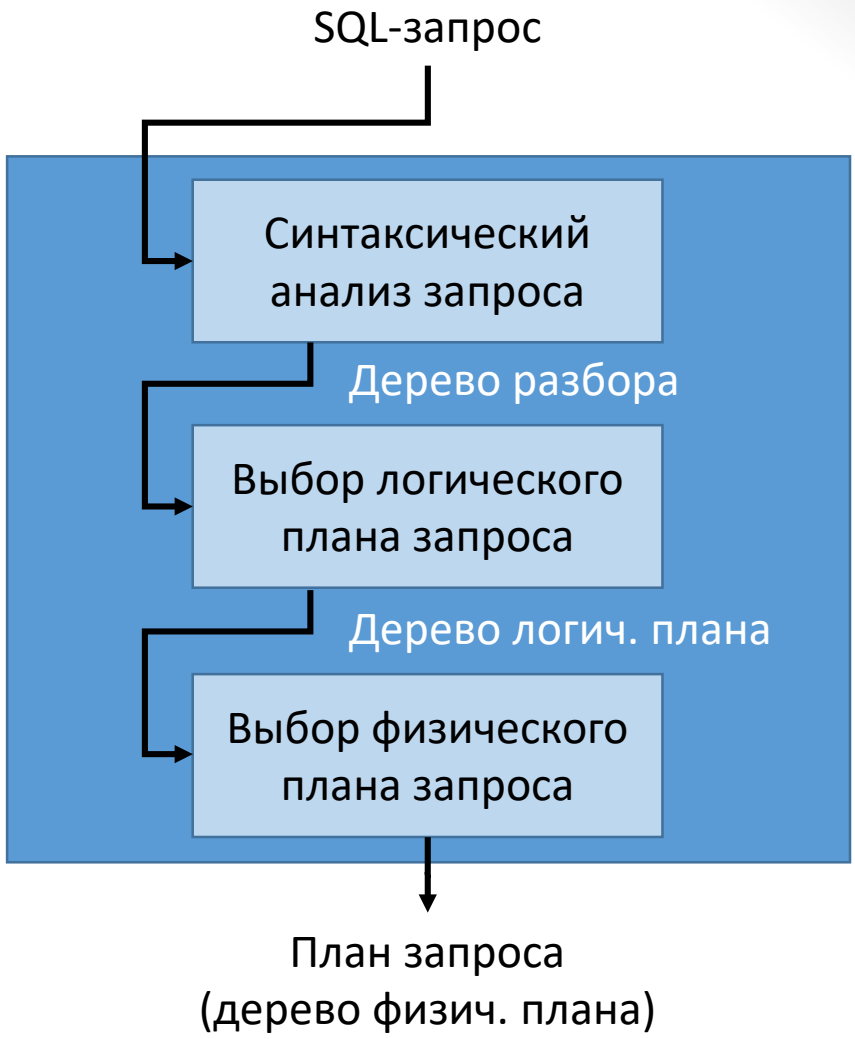
Процессор запросов

- ***Процессор запросов*** – группа компонентов СУБД, которые преобразуют вводимые пользователем запросы и команды в последовательности операций над базой данных и выполняют эти операции.

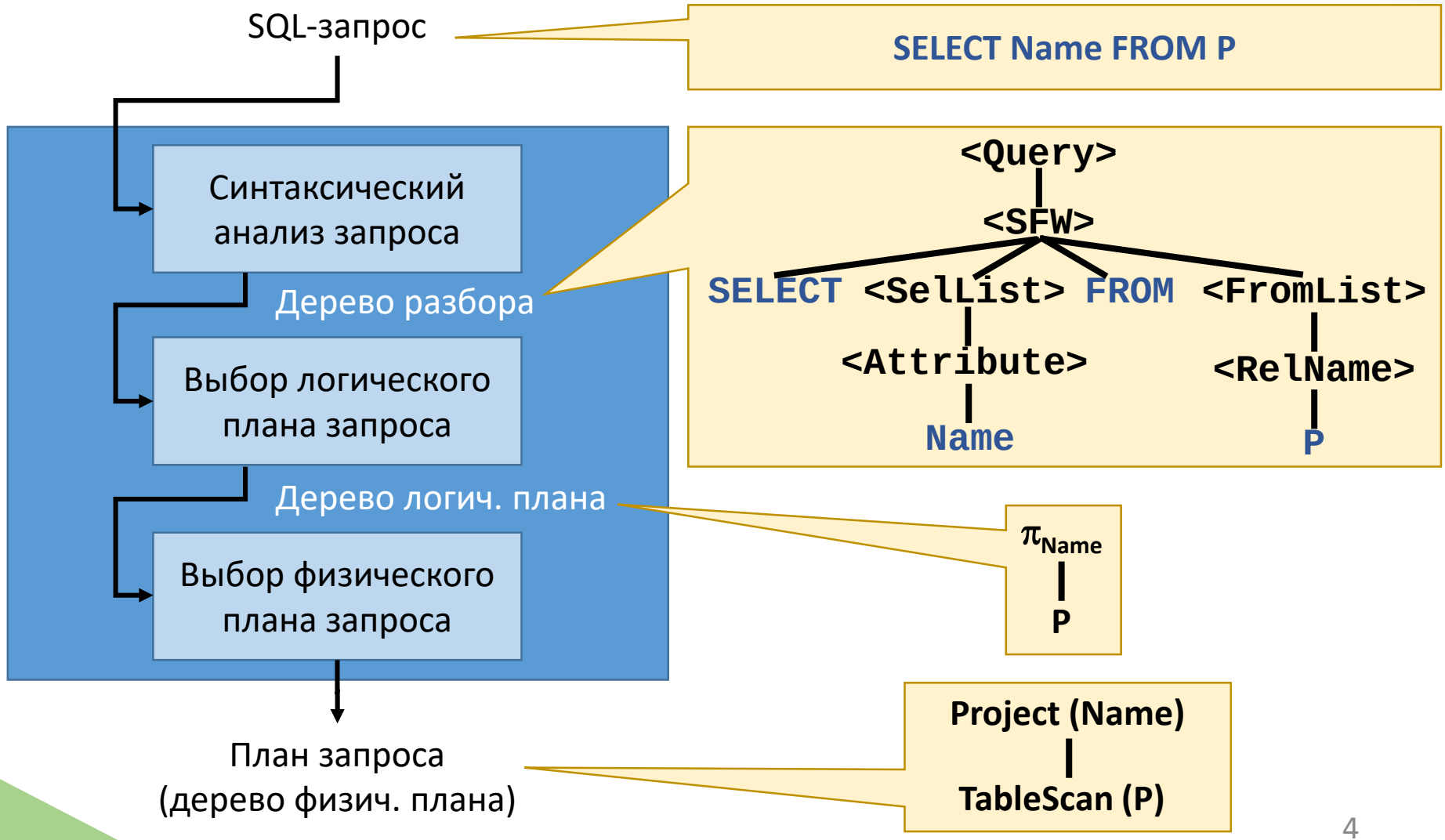


Компиляция запроса

1. Синтаксический анализ (parsing) – конструирование ***дерева запроса***, описывающего запрос и его структуру.
2. На следующем этапе происходит генерация из дерева разбора начального ***логического плана запроса*** (реляционное выражение). Затем план преобразуется к более эффективному.
3. Логический план превращается в ***физический план***, состоящий из алгоритмов, реализующих каждый логический (реляционный) оператор, и дополнительных алгоритмов.

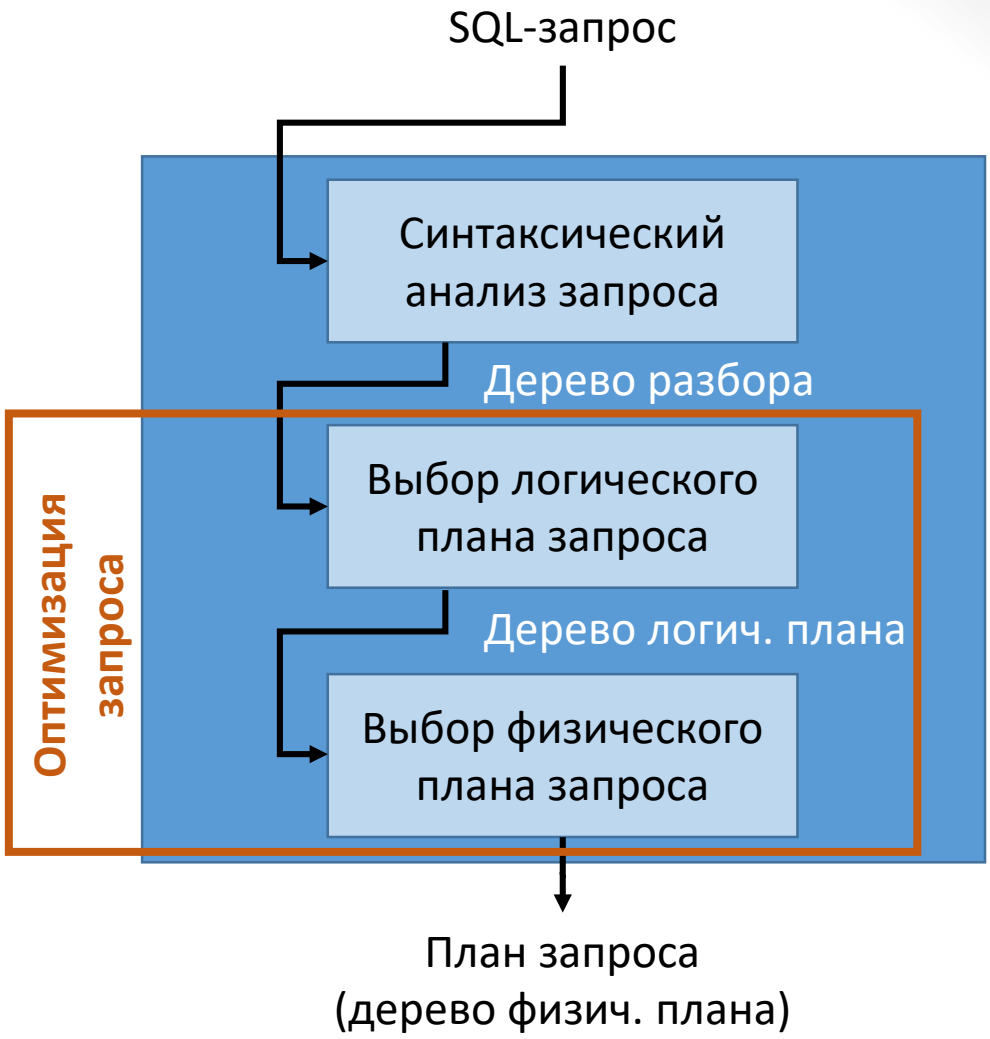


Компиляция запроса. Пример



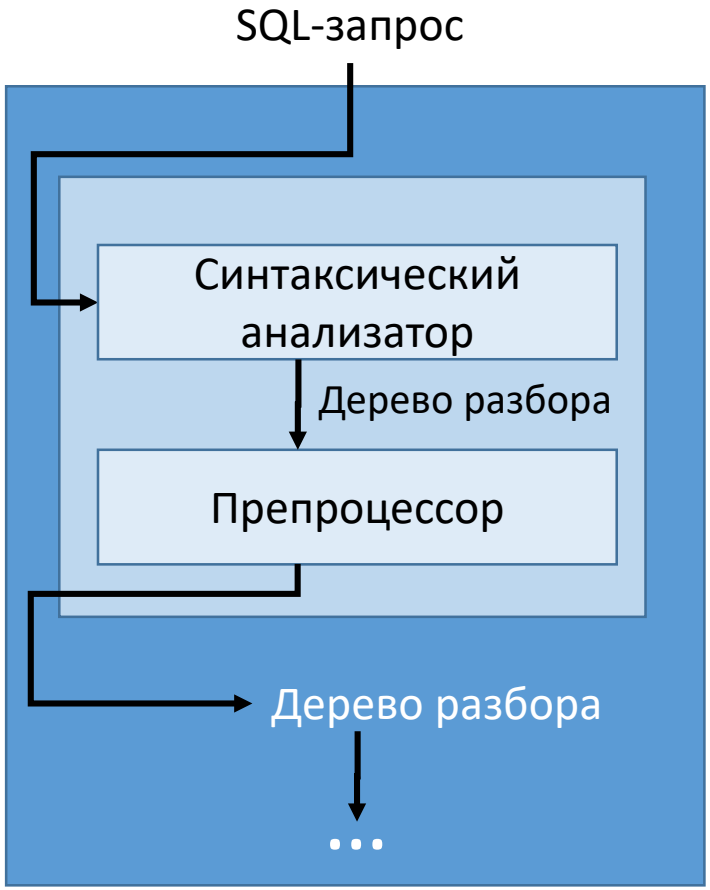
Оптимизация запроса

- Оптимизация запросов – наиболее трудоемкая фаза процесса компиляции запроса.
- Для оптимизации запроса нужно ответить на вопросы:
  - Какие **эквивалентные реляционные выражения** позволят наиболее эффективно получить результат?
  - Какие алгоритмы стоит выбрать для **реализации каждой реляционной операции**?
  - Каким образом следует организовать **передачу данных** от одной операции к другой (например, через буфер ОП)



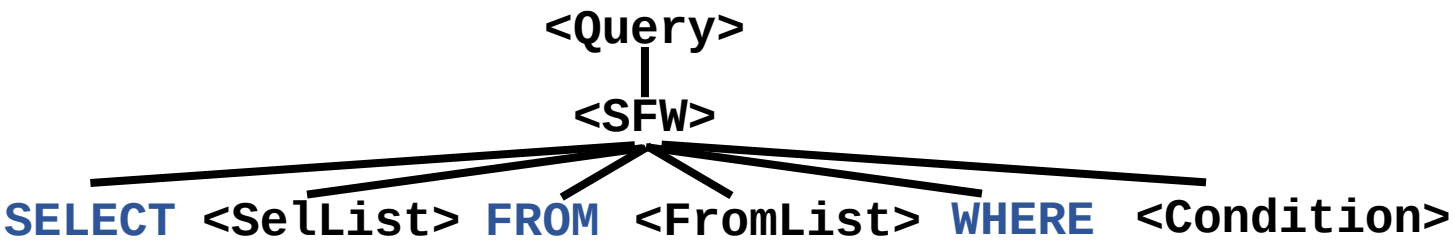
Синтаксический анализ запроса

- Задача ***синтаксического анализатора (parser)*** состоит в преобразовании текста на языке SQL в дерево разбора.
- Дерево разбора состоит из
  - ***атомов*** (например, служебные слова “SELECT”, “ORDER BY”, названия отношений и атрибутов “P”, “SP.Qty”, операторы “+”, “-”, др.)
  - ***синтаксических категорий*** (обозначения семейств частей запроса, например, <SFW>, <Condition>, <SelList>)



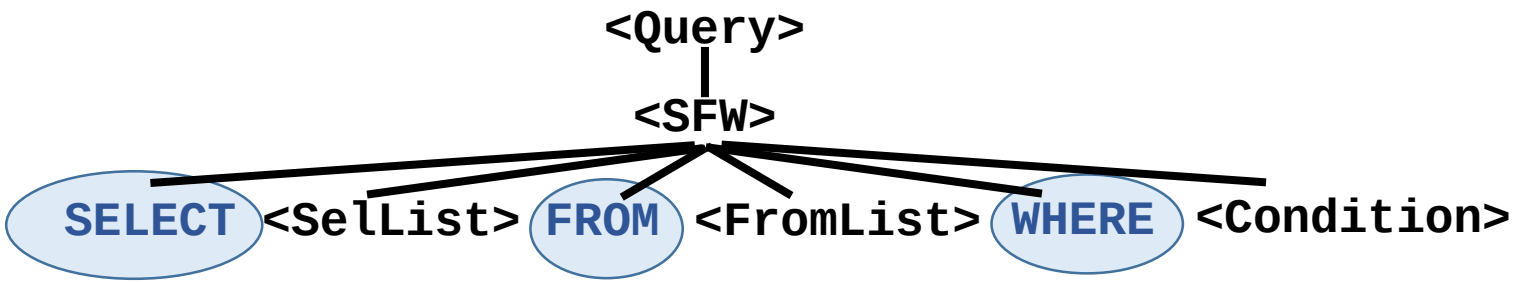
Пример дерева разбора

SELECT P.Name, SP.SID FROM P, SP WHERE P.PID = SP.PID



Пример дерева разбора

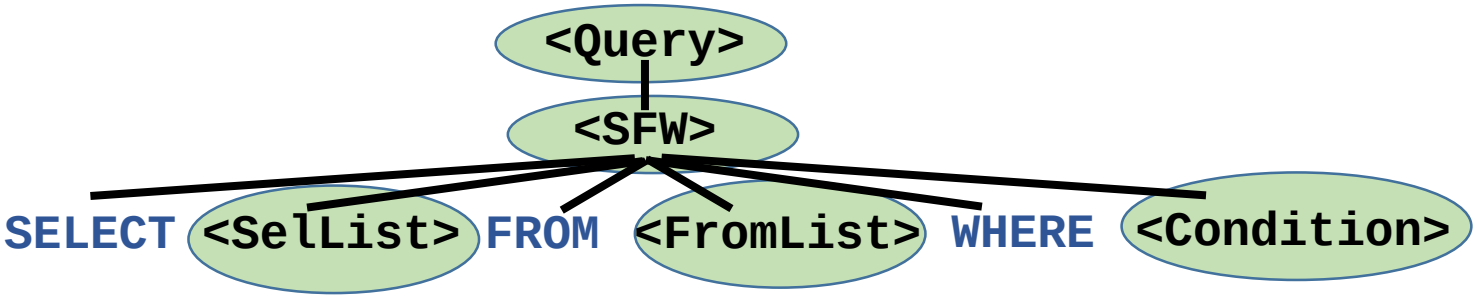
SELECT P.Name, SP.SID FROM P, SP WHERE P.PID = SP.PID



Атомы

Пример дерева разбора

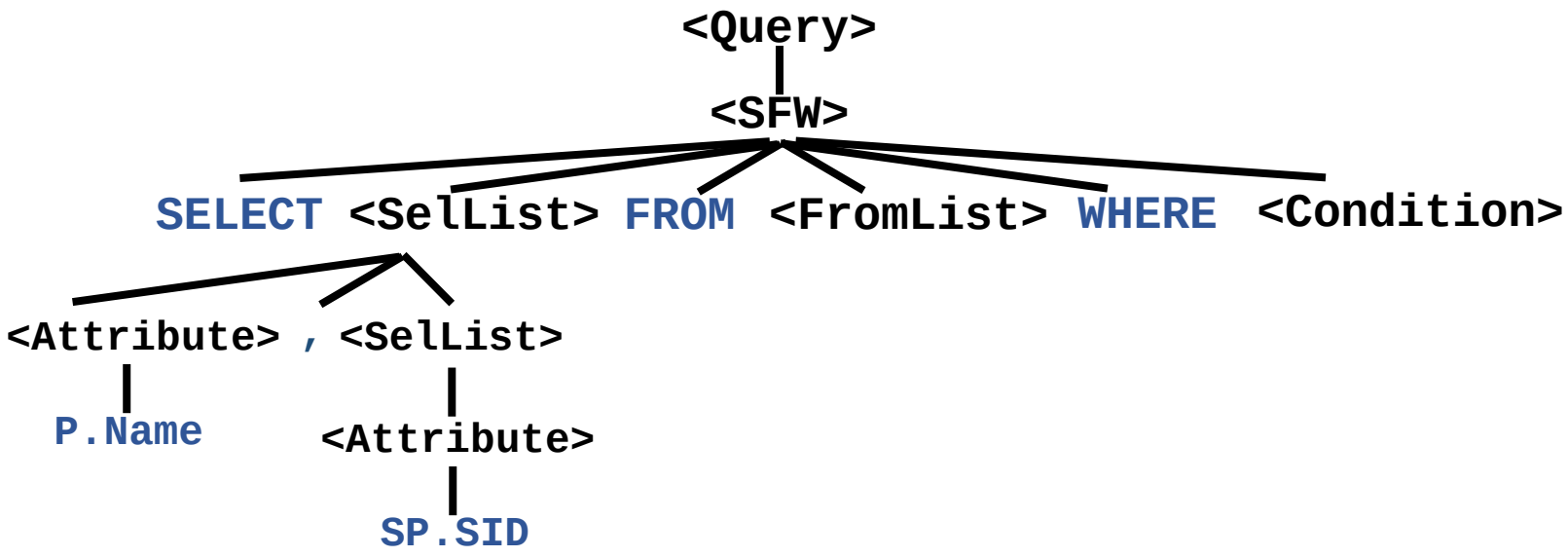
SELECT P.Name, SP.SID FROM P, SP WHERE P.PID = SP.PID



Синтаксические категории

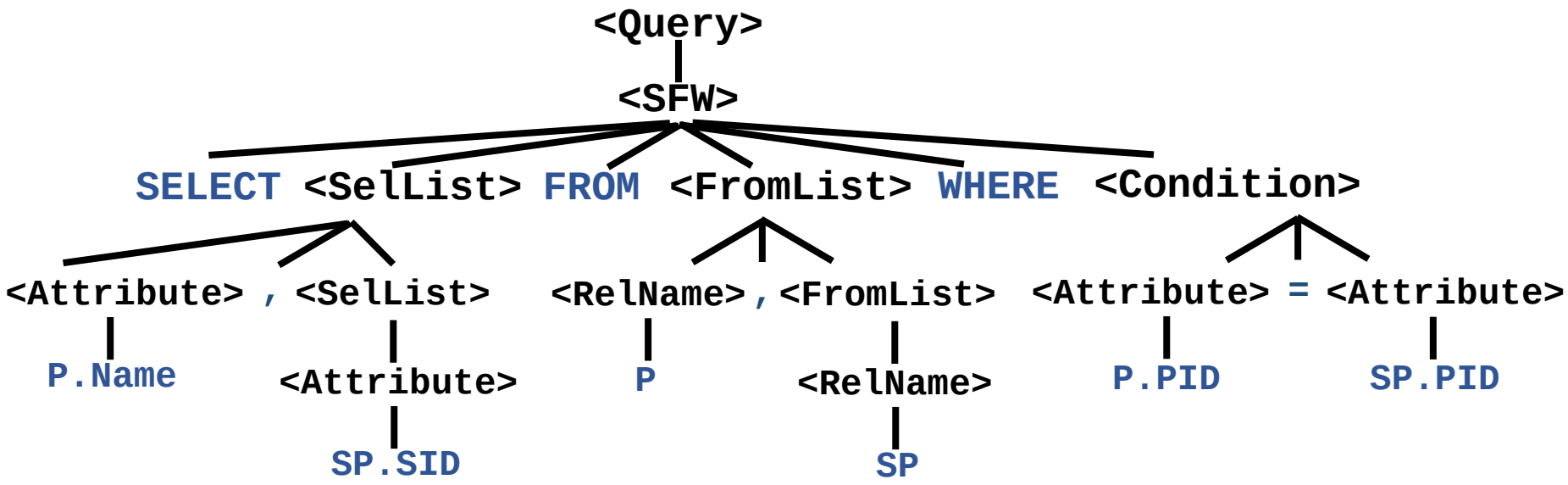
Пример дерева разбора

SELECT P.Name, SP.SID FROM P, SP WHERE P.PID = SP.PID



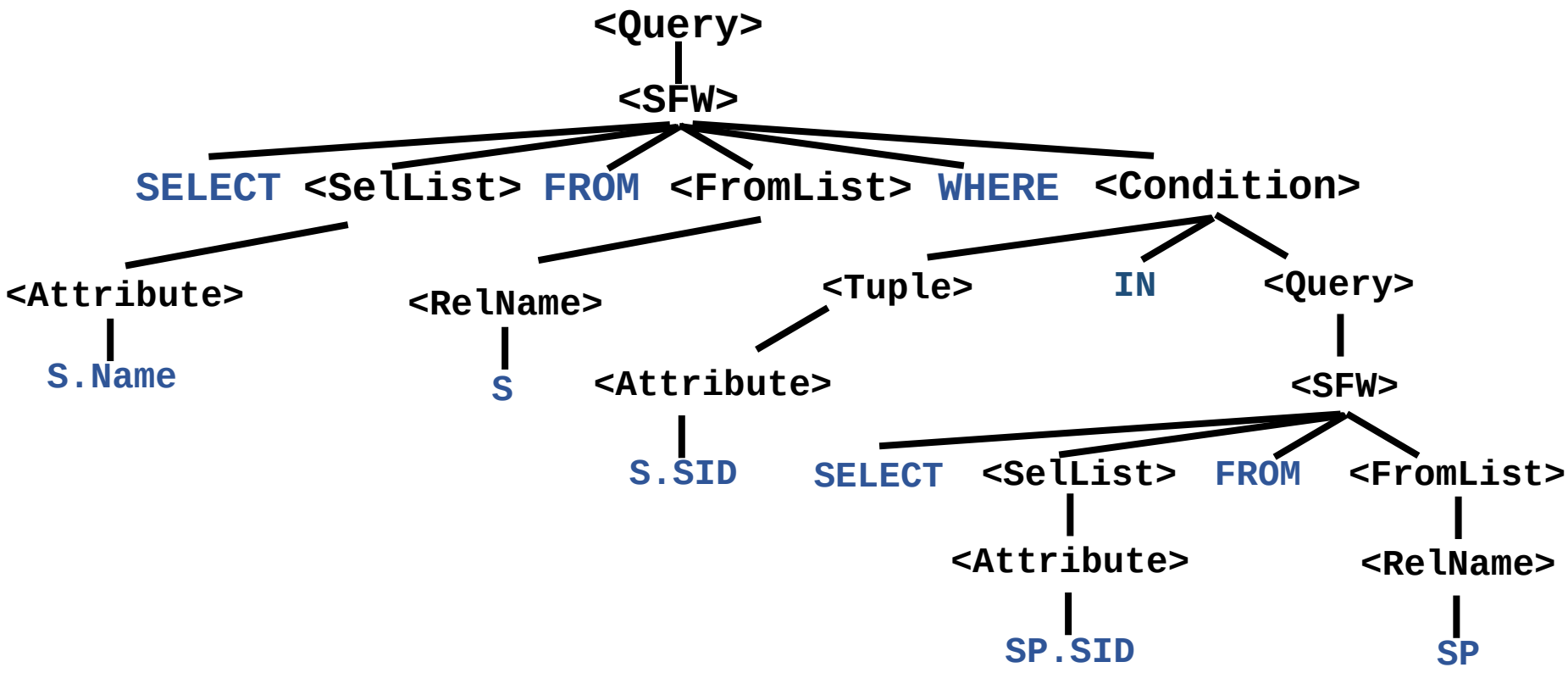
Пример дерева разбора

SELECT P.Name, SP.SID FROM P, SP WHERE P.PID = SP.PID



Пример дерева разбора

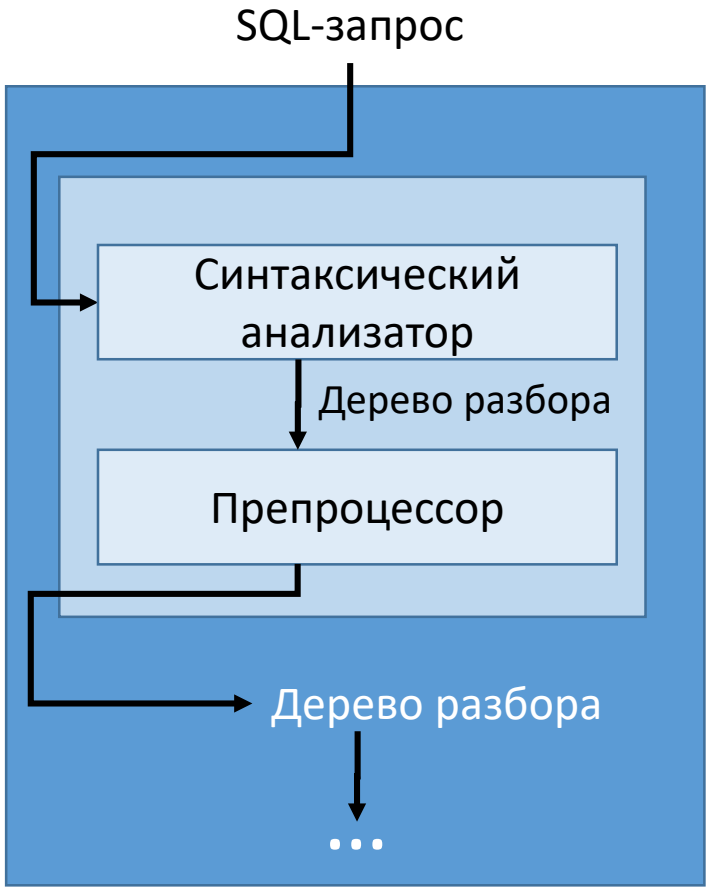
SELECT S.Name
FROM S WHERE S.SID IN (SELECT SP.SID FROM SP);



Синтаксический анализ запроса

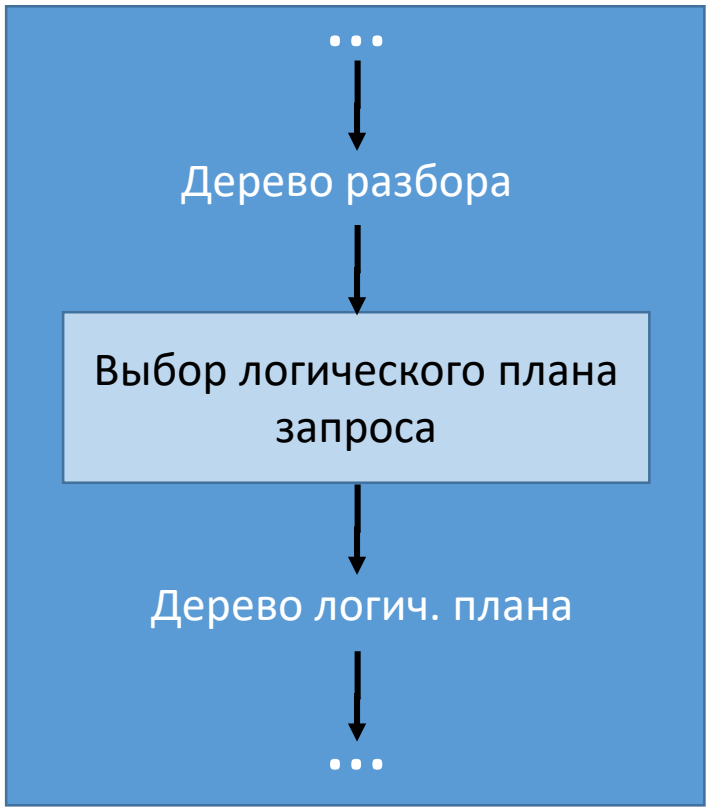
- Задачи ***препроцессора (preprocessor)***
  - *распознавание виртуальных таблиц и представлений*, замена из на соответствующий план запроса;
  - *семантический контроль*:
    - контроль употребления имен отношений
    - контроль типов

- Дерево разбора, успешно прошедшее все стадии контроля, называется ***действительным (valid)***.



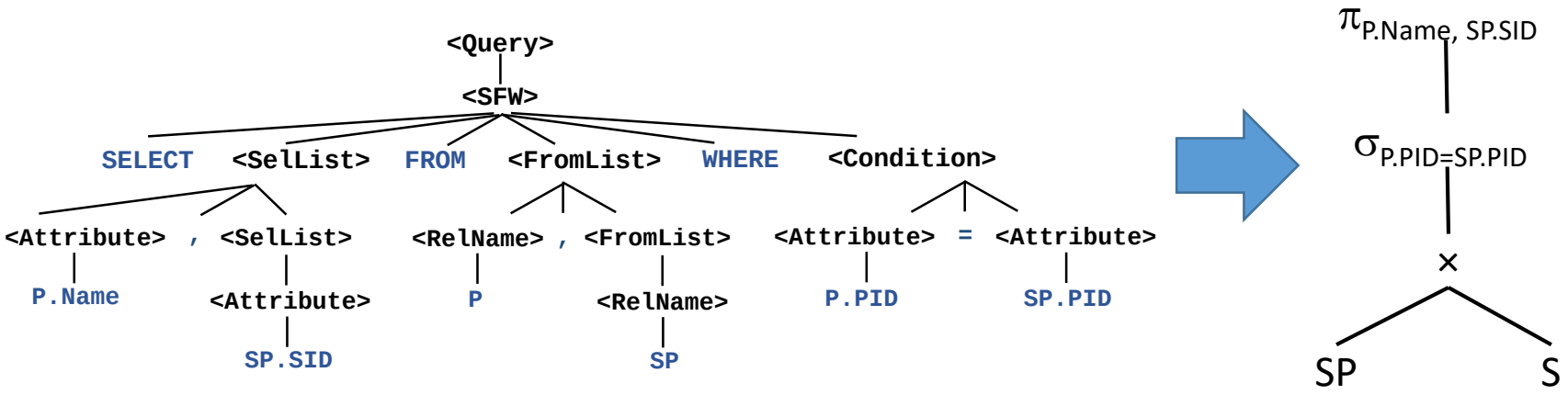
Выбор логического плана запроса

- После составления дерева разбора строится первоначальный логический план.
- Полученное реляционное выражение преобразуется в такое эквивалентное выражение, от которого можно ожидать наиболее эффективный физический план запроса.



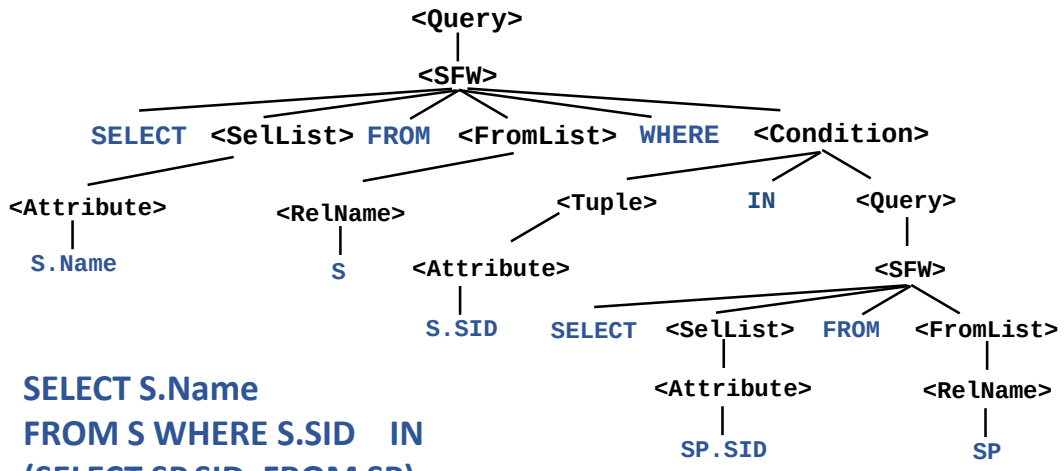
Первоначальный логический план

- В первоначальном логическом плане происходит замена вершин и структуры дерева разбора соответствующими группами операторов реляционной алгебры, с использованием определенных правил.
- Правила преобразования деревьев разбора (пример):
  - конструкция <SFW> **без подзапроса** целиком заменяется на $\pi_{A1, \ldots, AN}(\sigma_C(R_1 \times \ldots \times R_k))$

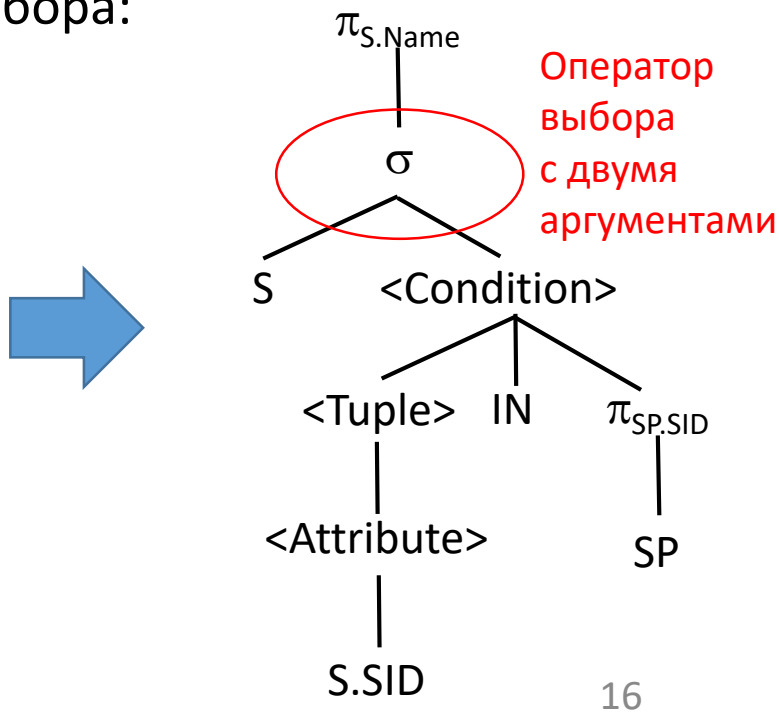


Первоначальный логический план

- В первоначальном логическом плане происходит замена вершин и структуры дерева разбора соответствующими группами операторов реляционной алгебры, с использованием определенных правил.
- Правила преобразования деревьев разбора:
  - конструкция <SFW> **с подзапросом**

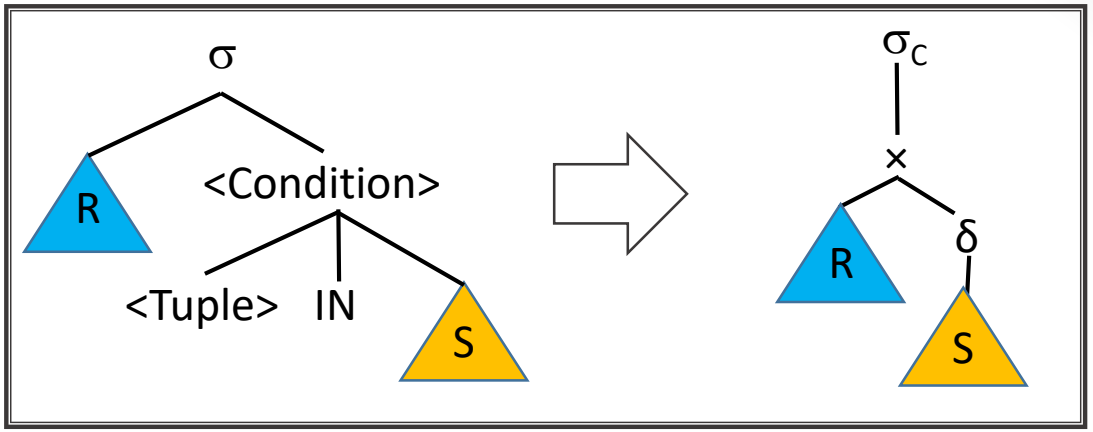


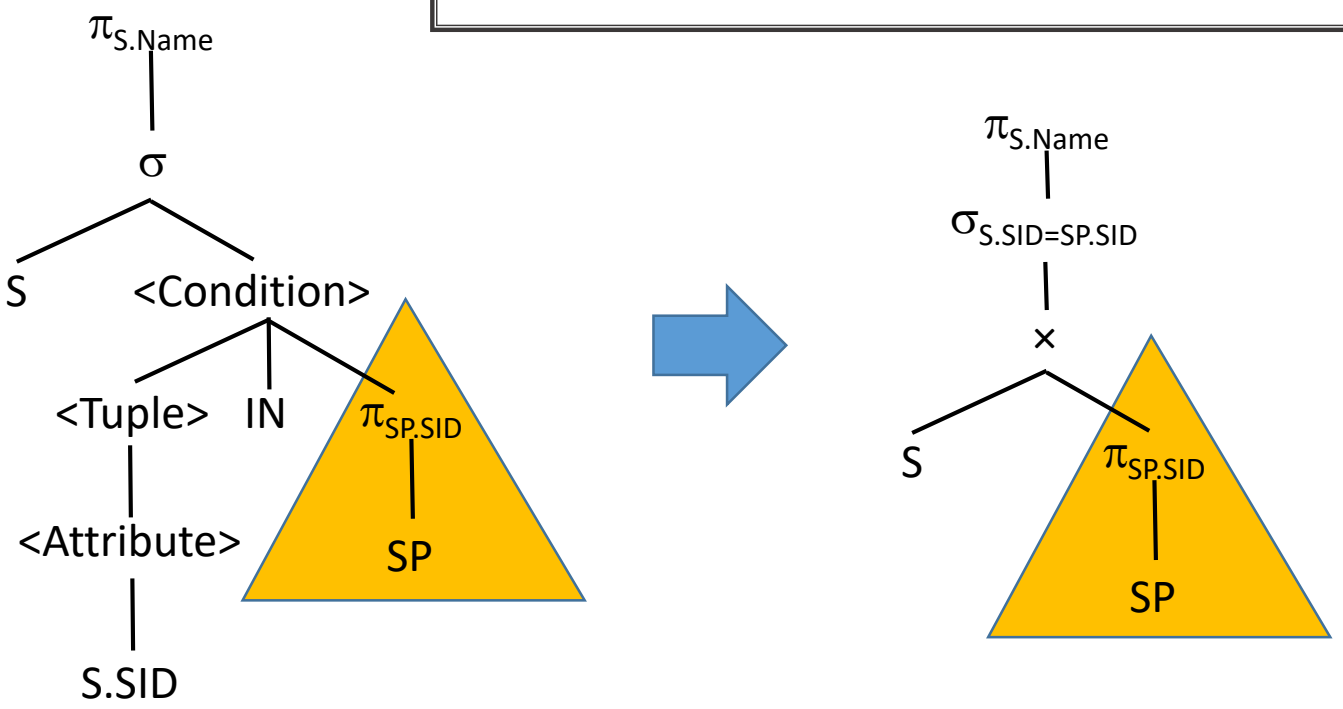
SELECT S.Name
FROM S WHERE S.SID IN
(SELECT SP.SID FROM SP);



Первоначальный логический план

- Преобразование оператора выбора с двумя аргументами для **НЕзависимого подзапроса**:

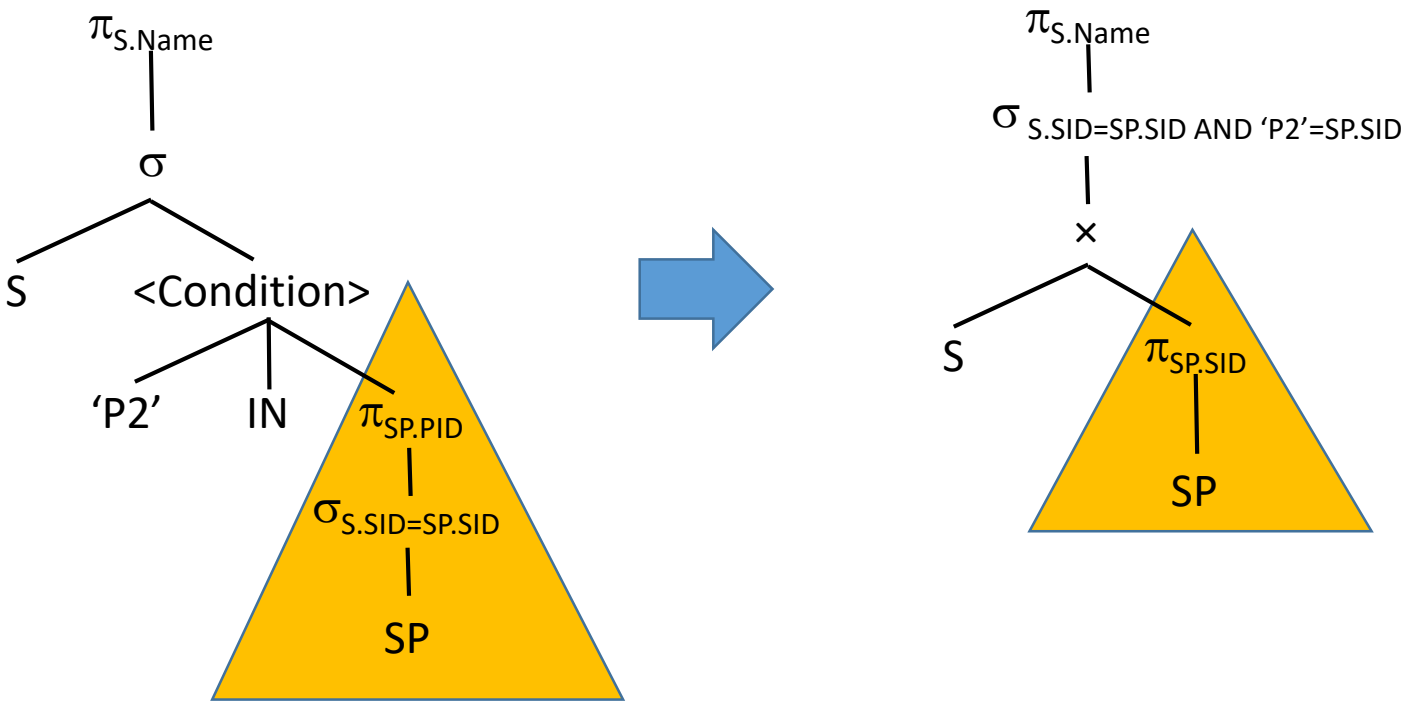




Первоначальный логический план

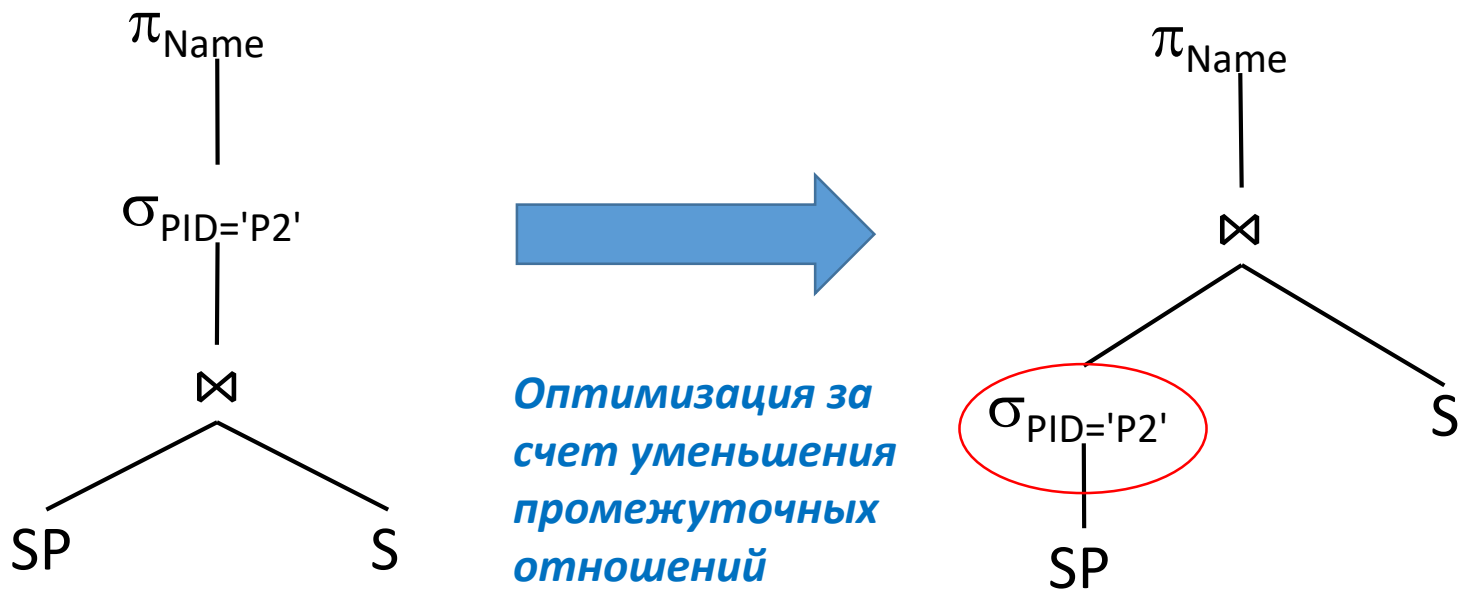
- Преобразование оператора выбора с двумя аргументами для **зависимого подзапроса**:

SELECT S.Name FROM S WHERE 'P2' = IN
(SELECT SP.PID FROM SP WHERE S.SID=SP.SID);



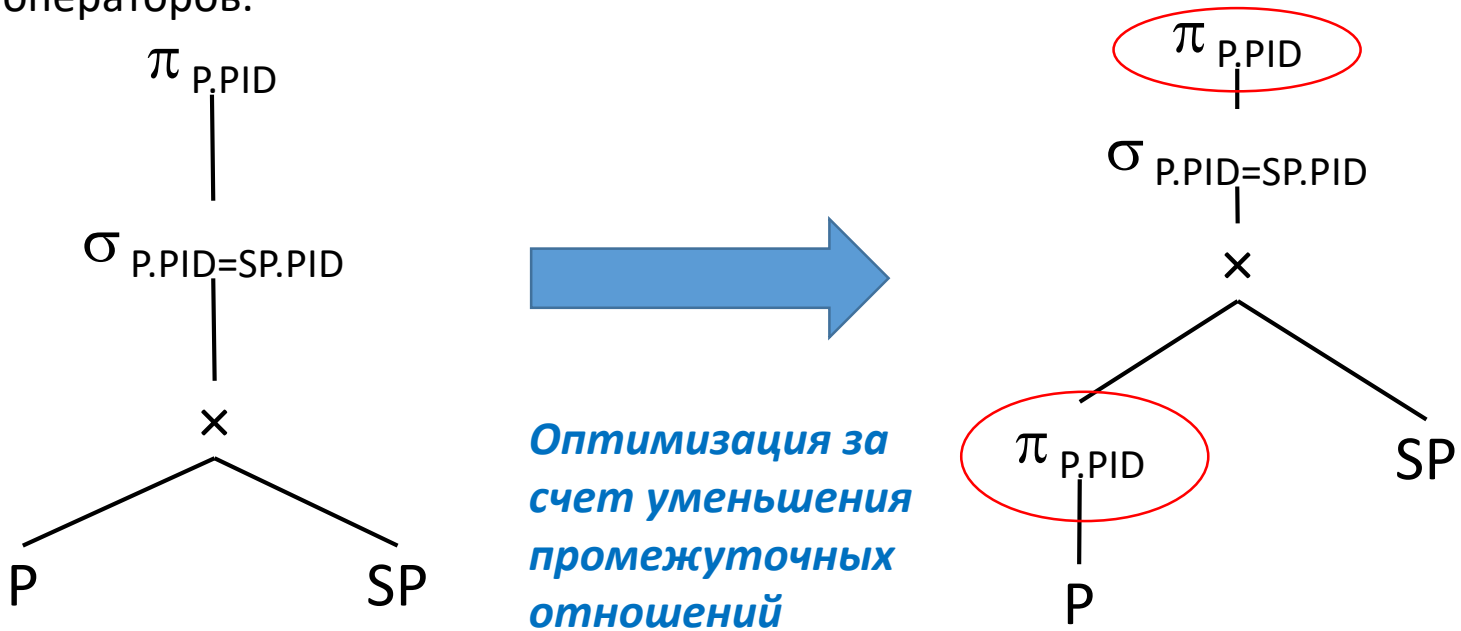
Улучшение логического плана запроса

- Выполняется на основе правил преобразования реляционного выражения в эквивалентную форму (операции выборки и проекции, распределительный закон, коммутативность и ассоциативность, идемпотентность и поглощение).
- Наиболее широко применимы следующие подходы:
  - ***Продвижение оператора выбора «вниз»*** по дереву логического плана до максимально «глубокого» уровня.



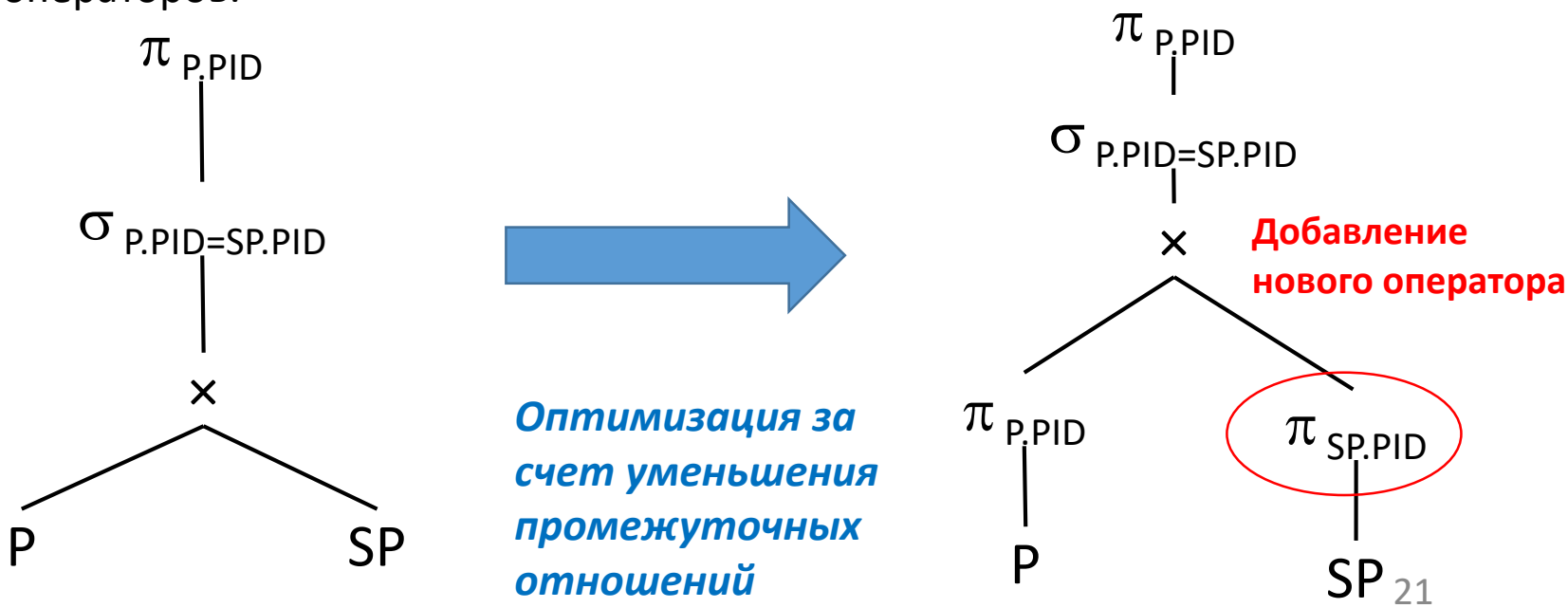
Улучшение логического плана запроса

- Выполняется на основе правил преобразования реляционного выражения в эквивалентную форму (операции выборки и проекции, распределительный закон, коммутативность и ассоциативность, идемпотентность и поглощение).
- Наиболее широко применимы следующие подходы:
  - ***Продвижение оператора проекции «вниз»*** по дереву логического плана до максимально «глубокого» уровня или добавление новых операторов.



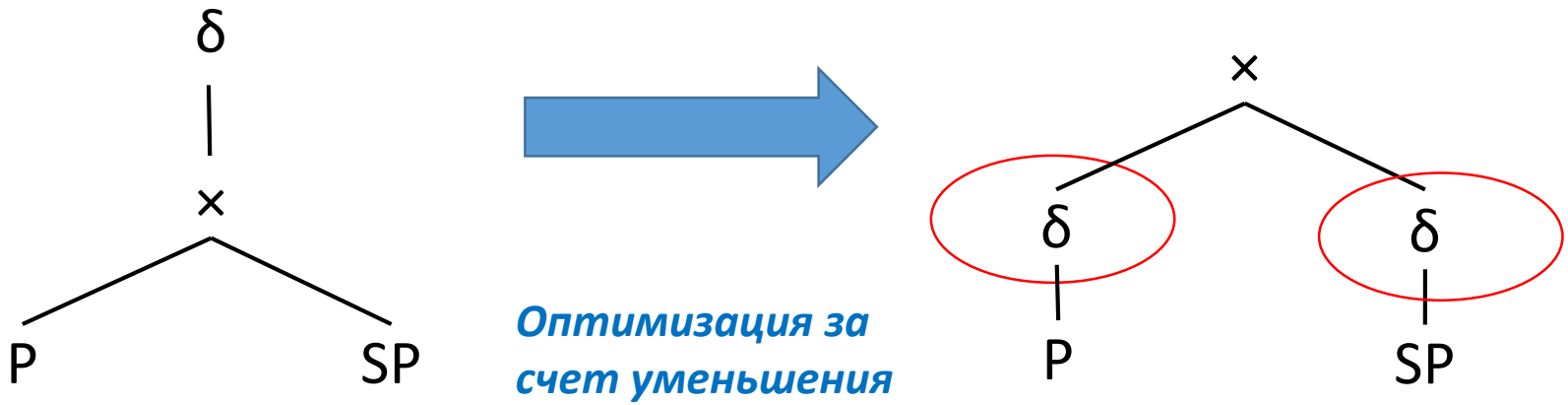
Улучшение логического плана запроса

- Выполняется на основе правил преобразования реляционного выражения в эквивалентную форму (операции выборки и проекции, распределительный закон, коммутативность и ассоциативность, идемпотентность и поглощение).
- Наиболее широко применимы следующие подходы:
  - ***Продвижение оператора проекции «вниз»*** по дереву логического плана до максимально «глубокого» уровня или добавление новых операторов.



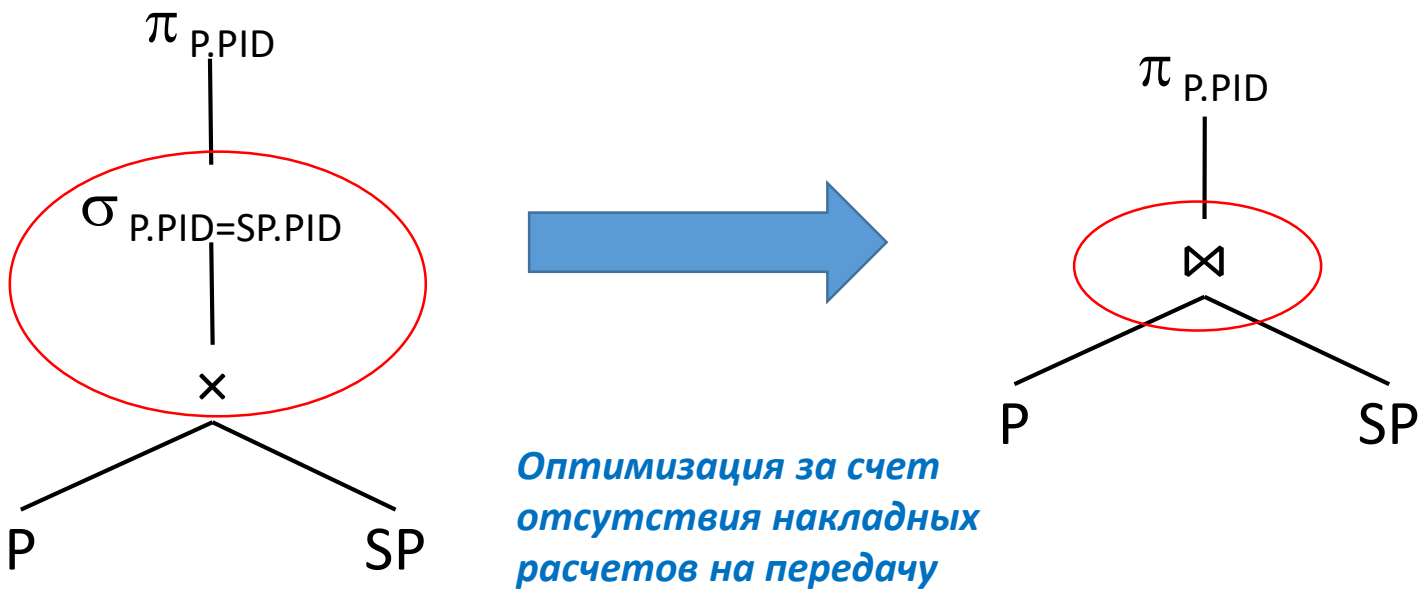
Улучшение логического плана запроса

- Выполняется на основе правил преобразования реляционного выражения в эквивалентную форму (операции выборки и проекции, распределительный закон, коммутативность и ассоциативность, идемпотентность и поглощение).
- Наиболее широко применимы следующие подходы:
  - Изъятие ***оператора удаления дубликатов*** или перемещение в требуемые позиции дерева.



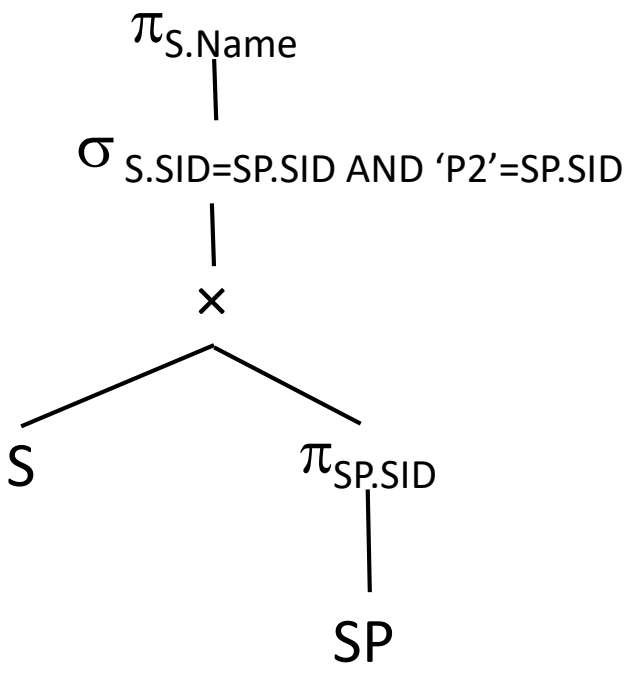
Улучшение логического плана запроса

- Выполняется на основе правил преобразования реляционного выражения в эквивалентную форму (операции выборки и проекции, распределительный закон, коммутативность и ассоциативность, идемпотентность и поглощение).
- Наиболее широко применимы следующие подходы:
  - ***Замена нескольких операций одной***.



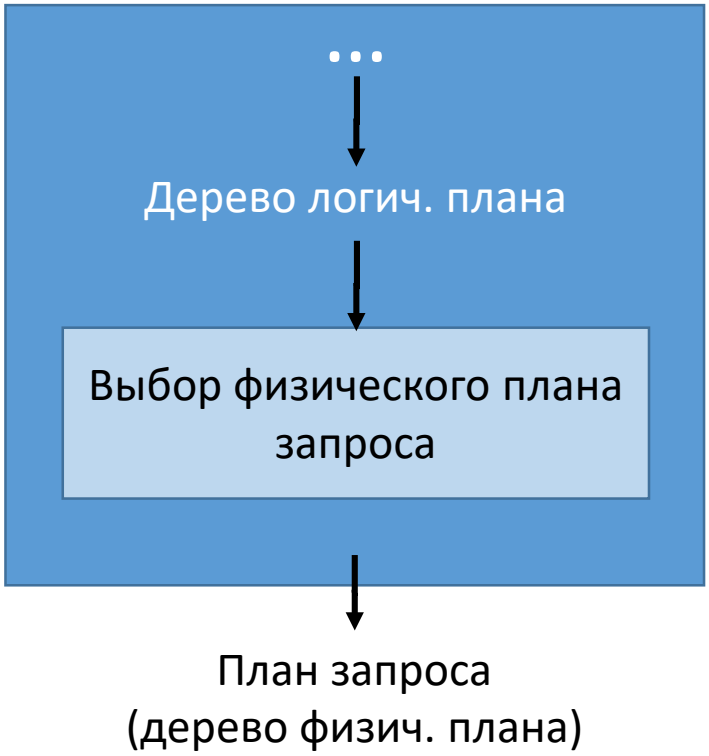
Оптимизация за счет отсутствия накладных расчетов на передачу промежуточного отношения

Задача: улучшите логический план

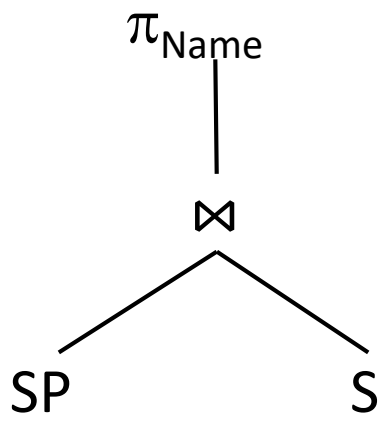


Выбор физического плана запроса

- На данном этапе выбираются:
  - алгоритм реализации каждого оператора логического плана
  - дополнительные операторы (сканирования, сортировки), необходимые для реализации физического плана, но явно отсутствующие в логическом
  - способ передачи значений аргументов от одного оператора к другому (например, через хранение промежуточных отношений на диске, либо использование итераторов)



Какие алгоритмы стоит выбрать для реализации каждой реляционной операции?



Варианты реализации операции естественного соединения:

- Алгоритм вложенных циклов
- Алгоритм соединения с сортировкой и слиянием
- Алгоритм соединения с хэшированием
- …