Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

Национальный исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

**Лабораторная работа №4**

Вариант № 6023

Группа: P3112

Выполнил: Балин А. А.

Проверила: Лисицина В.В.

Оглавление

[Введение 3](#_Toc137134552)

[Текст задания 4](#_Toc137134553)

[Выполнение 5](#_Toc137134554)

[Заключение 11](#_Toc137134555)

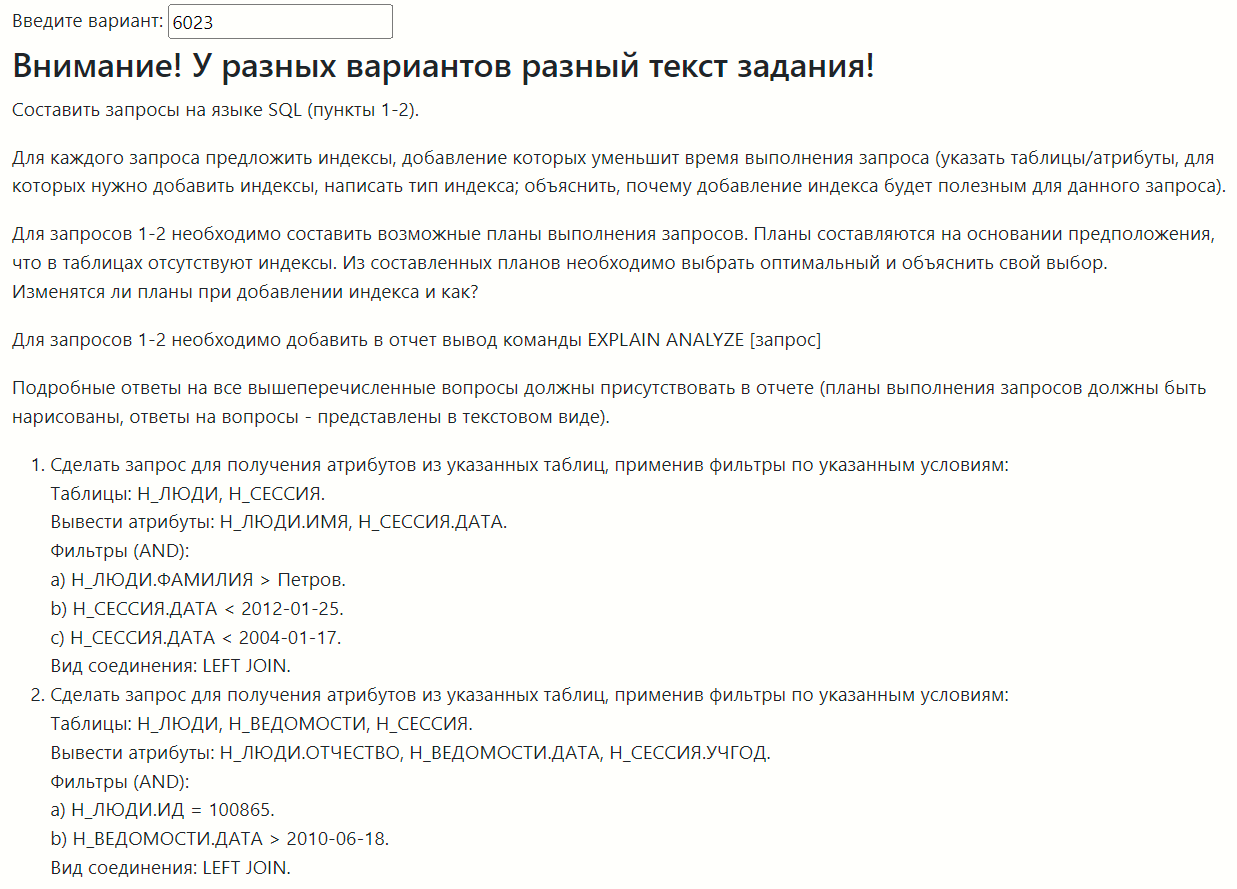
[Список литературы 12](#_Toc137134556)

# Введение

В данной лабораторной работе я изучу индексы, план выполнения SQL-запросов и транзакции.

# Текст задания

По варианту, выданному преподавателем, составить и выполнить запросы к базе данных "Учебный процесс".



# Выполнение

Запросы

SELECT

    ИМЯ,

    ДАТА

FROM

    Н\_ЛЮДИ

    LEFT JOIN Н\_СЕССИЯ ON Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД

WHERE

    ФАМИЛИЯ > 'Петров'

    AND Н\_СЕССИЯ.ДАТА < '2012-01-25'

    AND Н\_СЕССИЯ.ДАТА < '2004-01-17';

SELECT

    Н\_ЛЮДИ.ОТЧЕСТВО,

    Н\_ВЕДОМОСТИ.ДАТА,

    Н\_СЕССИЯ.УЧГОД

FROM

    Н\_ЛЮДИ

    LEFT JOIN Н\_СЕССИЯ ON Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД

    LEFT JOIN Н\_ВЕДОМОСТИ ON Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

WHERE

    Н\_ЛЮДИ.ИД = 100865

    AND Н\_ВЕДОМОСТИ.ДАТА > '2010-06-18';

Полезные для данных запросов индексы

Для первого запроса можно сделать:

* Хэш-индексы на атрибутах Н\_ЛЮДИ.ИД и Н\_СЕССИЯ.ЧЛВК\_ИД, так как хэш-индексы оптимальны при проверке равенства
* B-tree индекс для атрибута Н\_СЕССИЯ.ДАТА, это ускорит выборку строк по условию WHERE

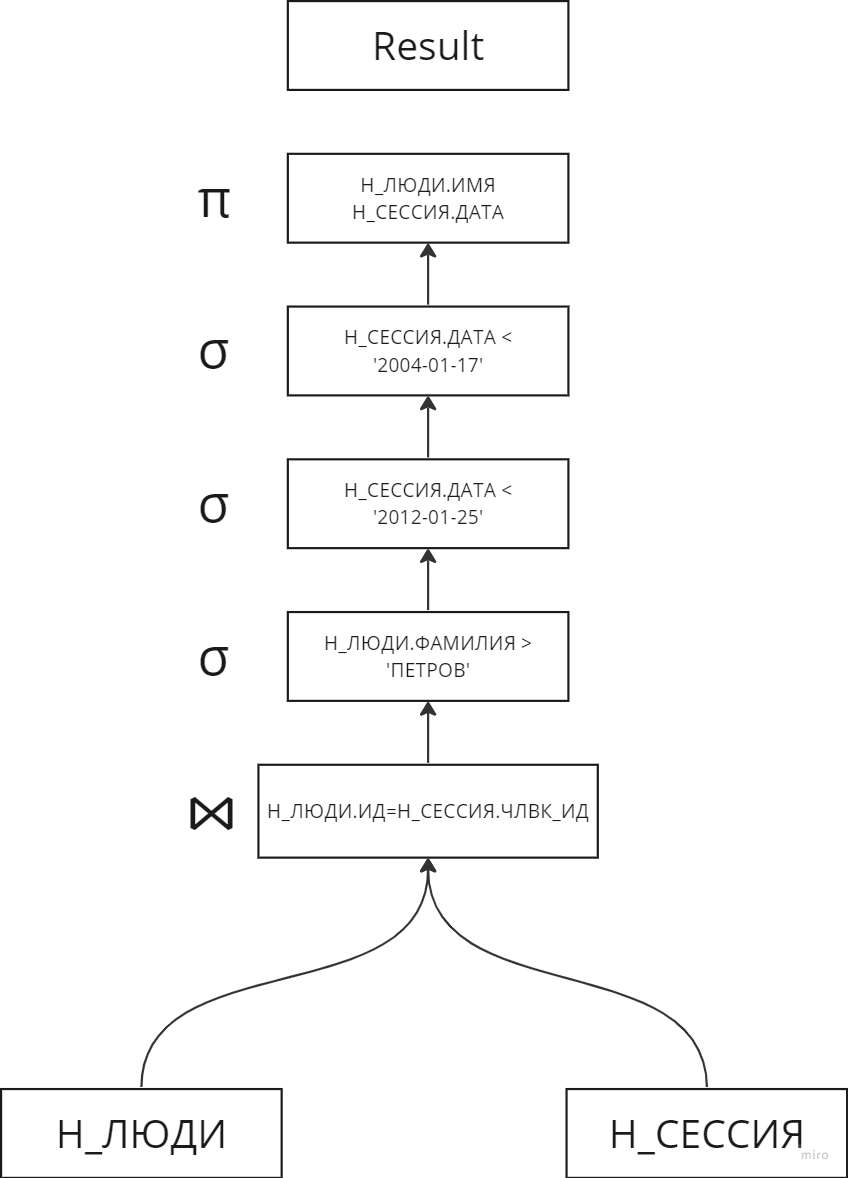
Для второго запроса можно сделать аналогичный хэш-индекс по первичным ключам используемых в JOIN таблиц, а также:

* Описанный выше хэш-индекс ускорит выборку строк по условию WHERE на атрибуте Н\_ЛЮДИ.ИД, так как хэш-индекс оптимален при проверке условия на равенство
* B-tree индекс на атрибут ДАТА в таблице Н\_ВЕДОМОСТИ, что опять же ускорит время выборки строк по условию в WHERE

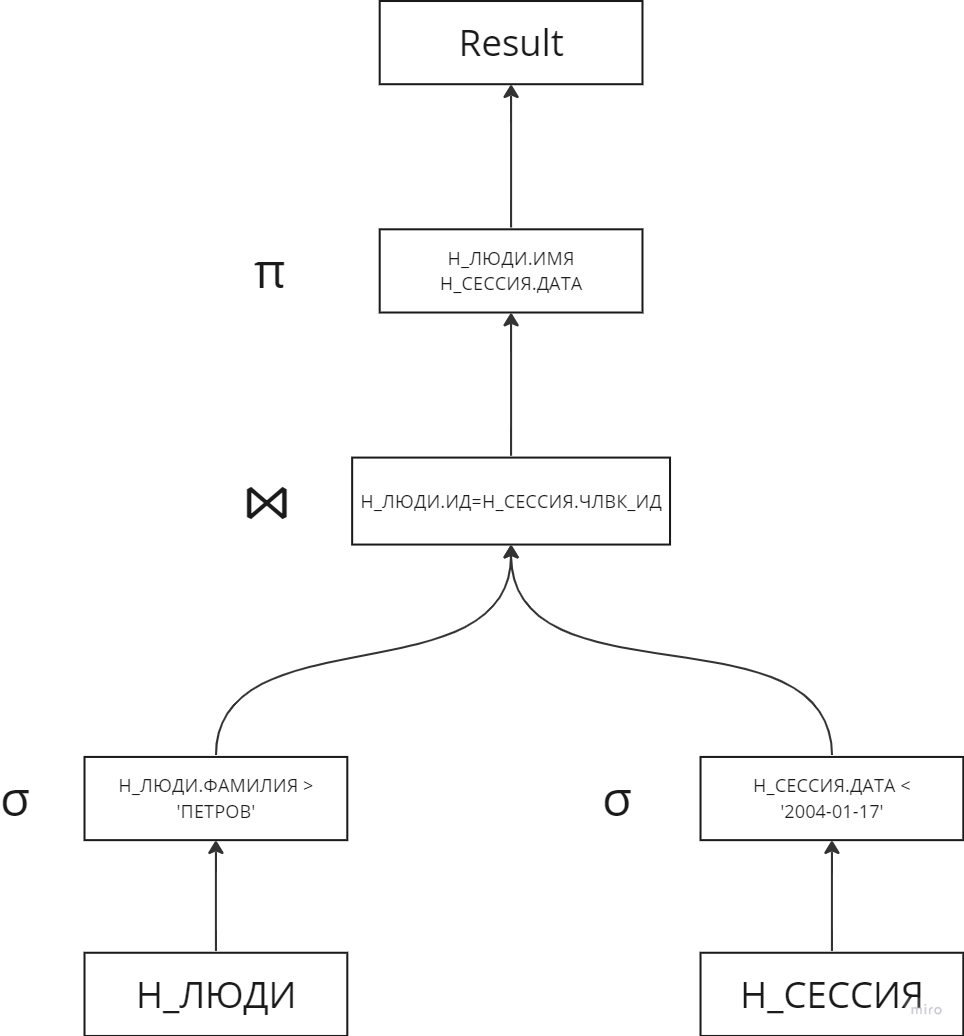
Планы выполнения запросов

1. Для первого запроса:

* Неоптимизированный план:

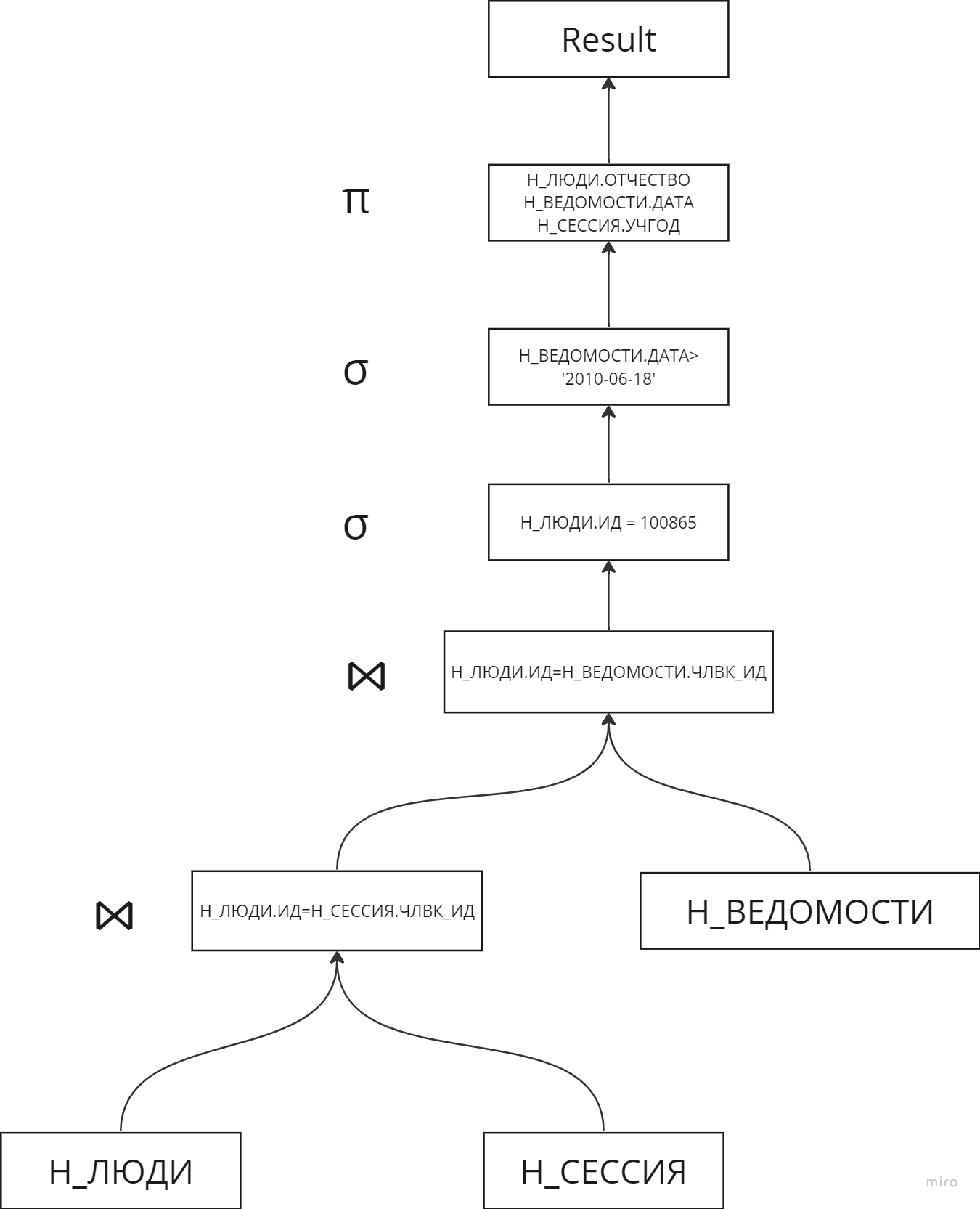


* Оптимизированный план: сделаем фильтрацию по нужным параметрам в самом начале, JOIN-соединение – как можно позже. Кроме того, можно включить голову и убрать лишнюю проверку условия с AND, так как одно исключает другое.

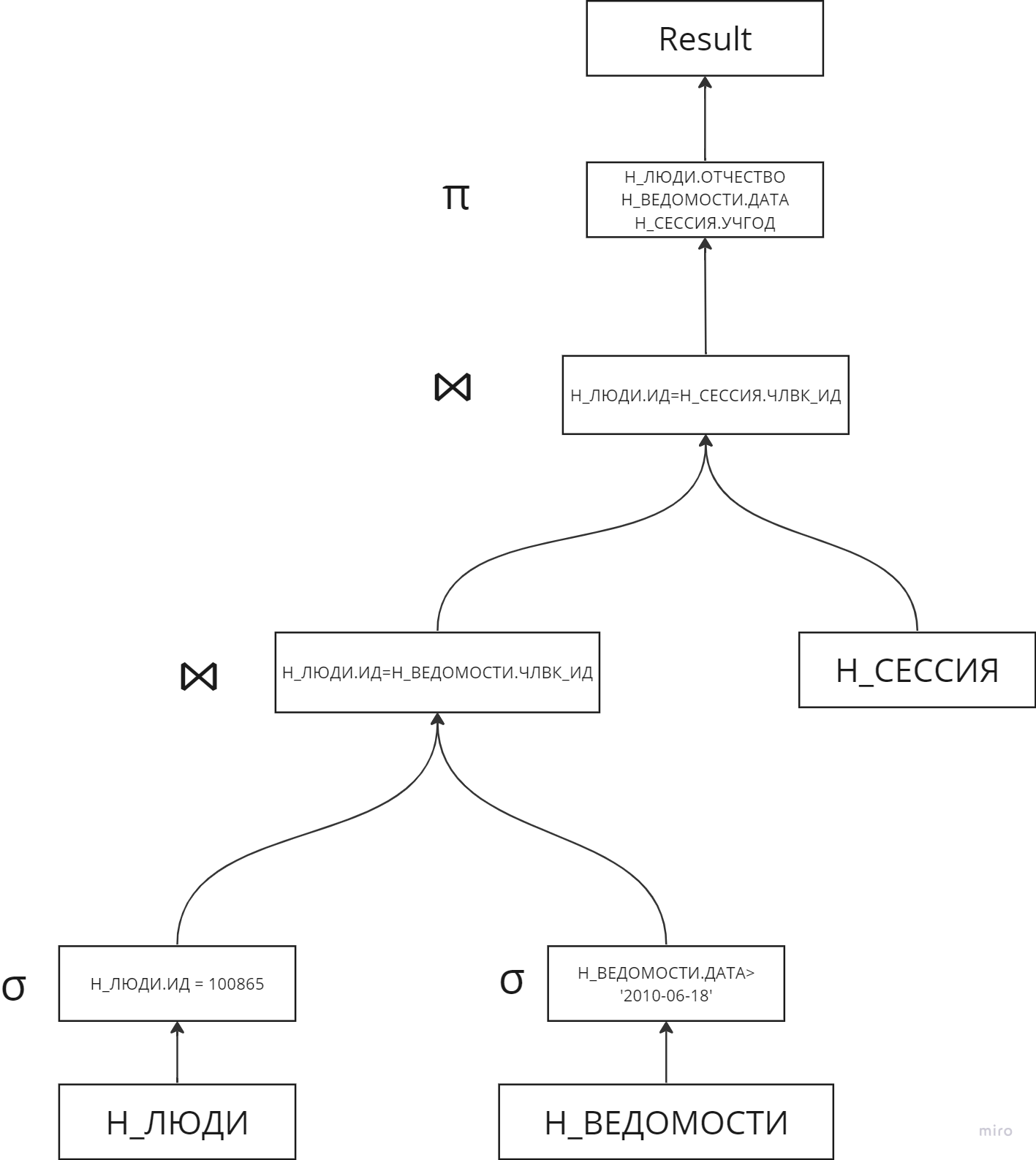


1. Для второго запроса:

* Неоптимизированный запрос:

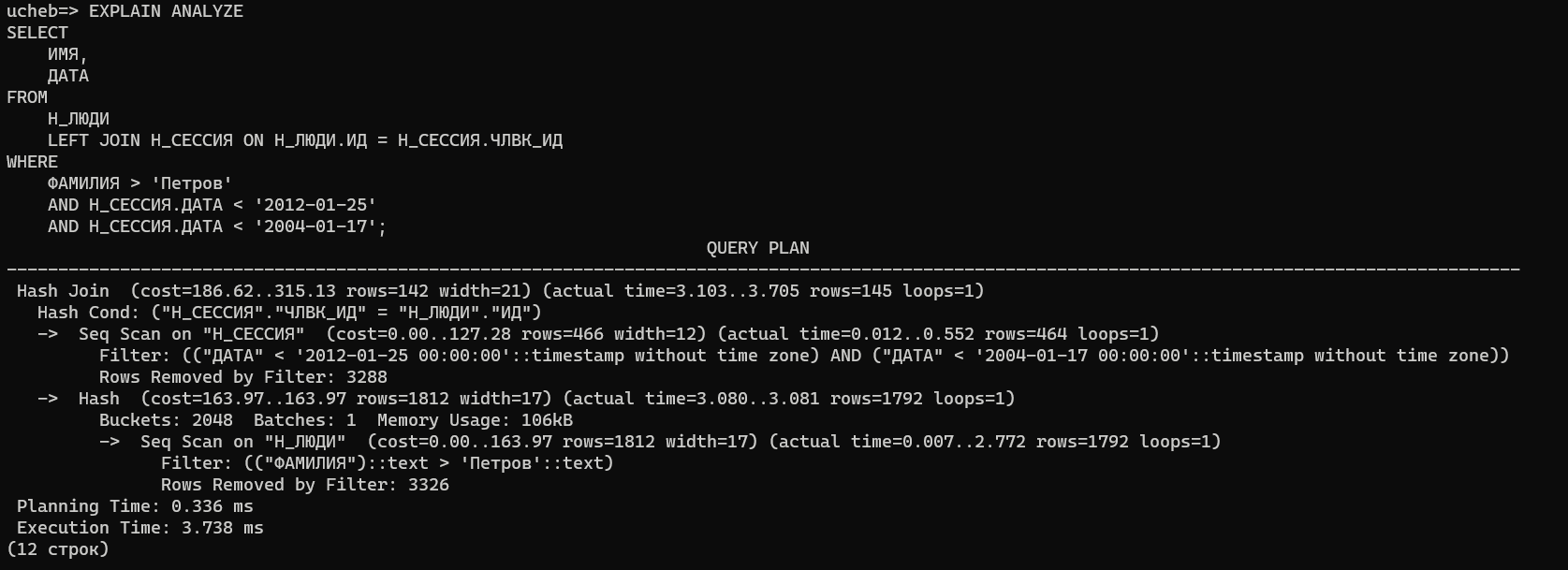


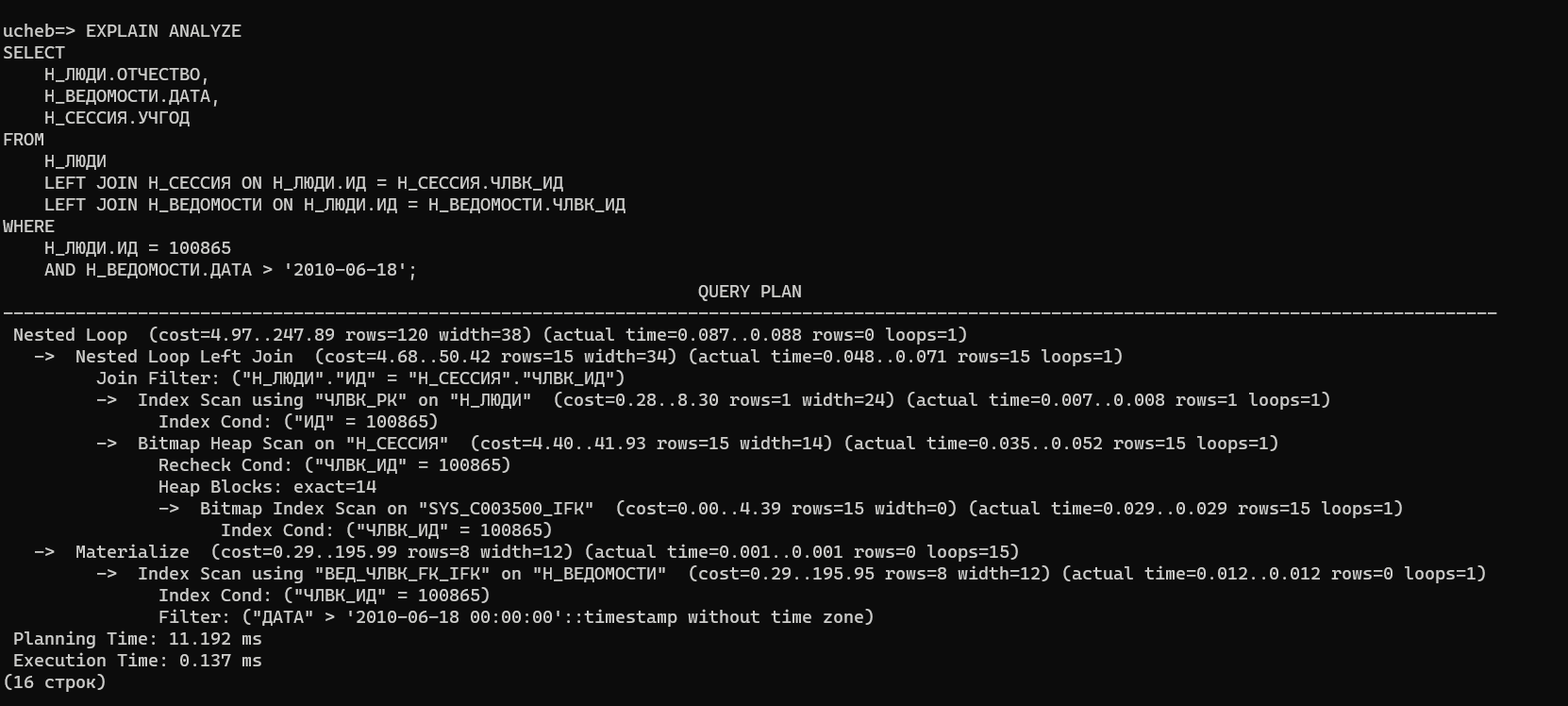
* Оптимизированный запрос: как можно раньше сортируем данные по условиям where, соединения таблиц как можно позже.



При добавлении индексов вместо просмотра всех строк из памяти в план добавится обращение к индексам. Хотя БД не во всех запросах использует индексы, в данном случае никаких ограничений на использование индексов нет. Ускорится выполнения выборки по условию в WHERE и MERGE-JOIN.

Запрос с EXPLAIN ANALYZE





# Заключение

Я разобрался в работе индексов, где они применимы, как SQL может оптимизировать запросы.

# Список литературы

**PostgreSQL 9.6.24 Documentation** [В Интернете]. - https://www.postgresql.org/docs/9.6/.