Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 «Программная инженерия» –

Системное и прикладное программное обеспечение

**Отчёт**

**По лабораторной работе №4**

по дисциплине «Базы данных»

**Вариант: 5557**

Выполнил:

Молчанов Фёдор Денисович

Группа: Р3113

Принял:

Николаев Владимир Вячеславович

г. Санкт-Петербург, 2024

Оглавление

[Задание 3](#_Toc167922260)

[Выполнение задания 3](#_Toc167922261)

[Запросы и их explain analyze 3](#_Toc167922262)

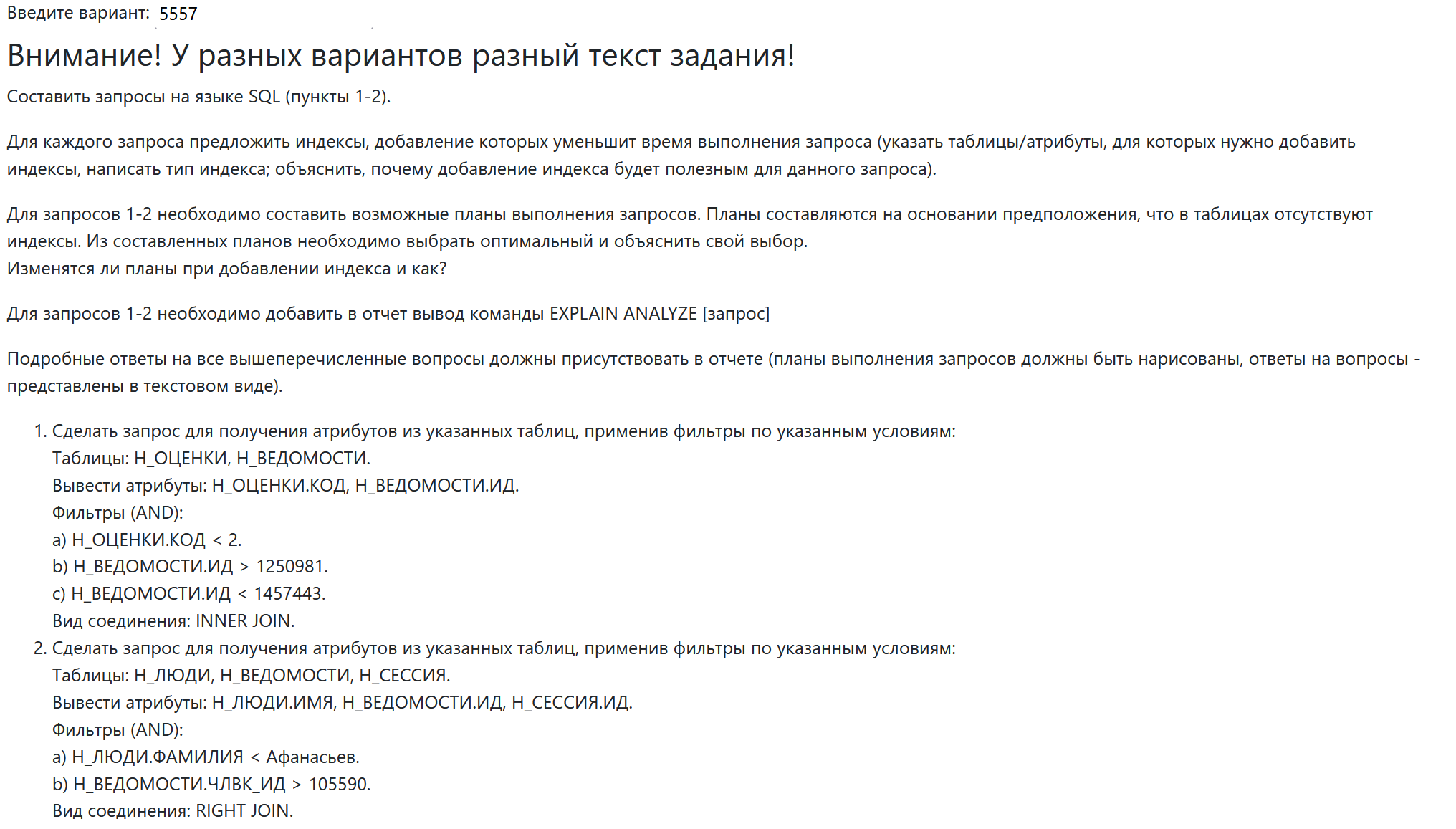
[Возможные планы выполнения запросов без индексов: 3](#_Toc167922263)

[1 запрос 4](#_Toc167922264)

[2 запрос 4](#_Toc167922265)

[Вывод 4](#_Toc167922266)

# Задание



# Выполнение задания

## Запросы и их explain analyze

1 запрос

SELECT Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ИД FROM Н\_ОЦЕНКИ

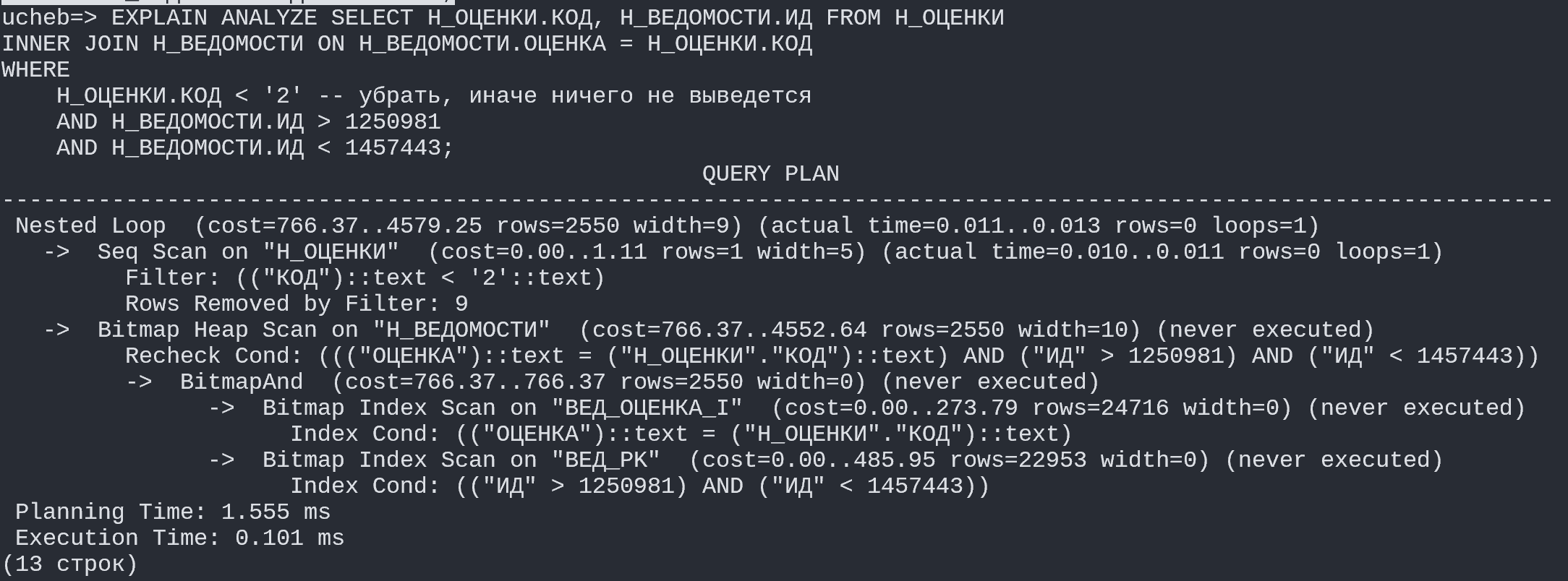
INNER JOIN Н\_ВЕДОМОСТИ ON Н\_ВЕДОМОСТИ.ОЦЕНКА = Н\_ОЦЕНКИ.КОД

WHERE

Н\_ОЦЕНКИ.КОД <'2' *-- убрать, иначе ничего не выведется*

    AND Н\_ВЕДОМОСТИ.ИД > 1250981

    AND Н\_ВЕДОМОСТИ.ИД < 1457443;



2 запрос

SELECT Н\_ЛЮДИ.ИМЯ, Н\_ВЕДОМОСТИ.ИД, Н\_СЕССИЯ.ИД FROM Н\_ВЕДОМОСТИ

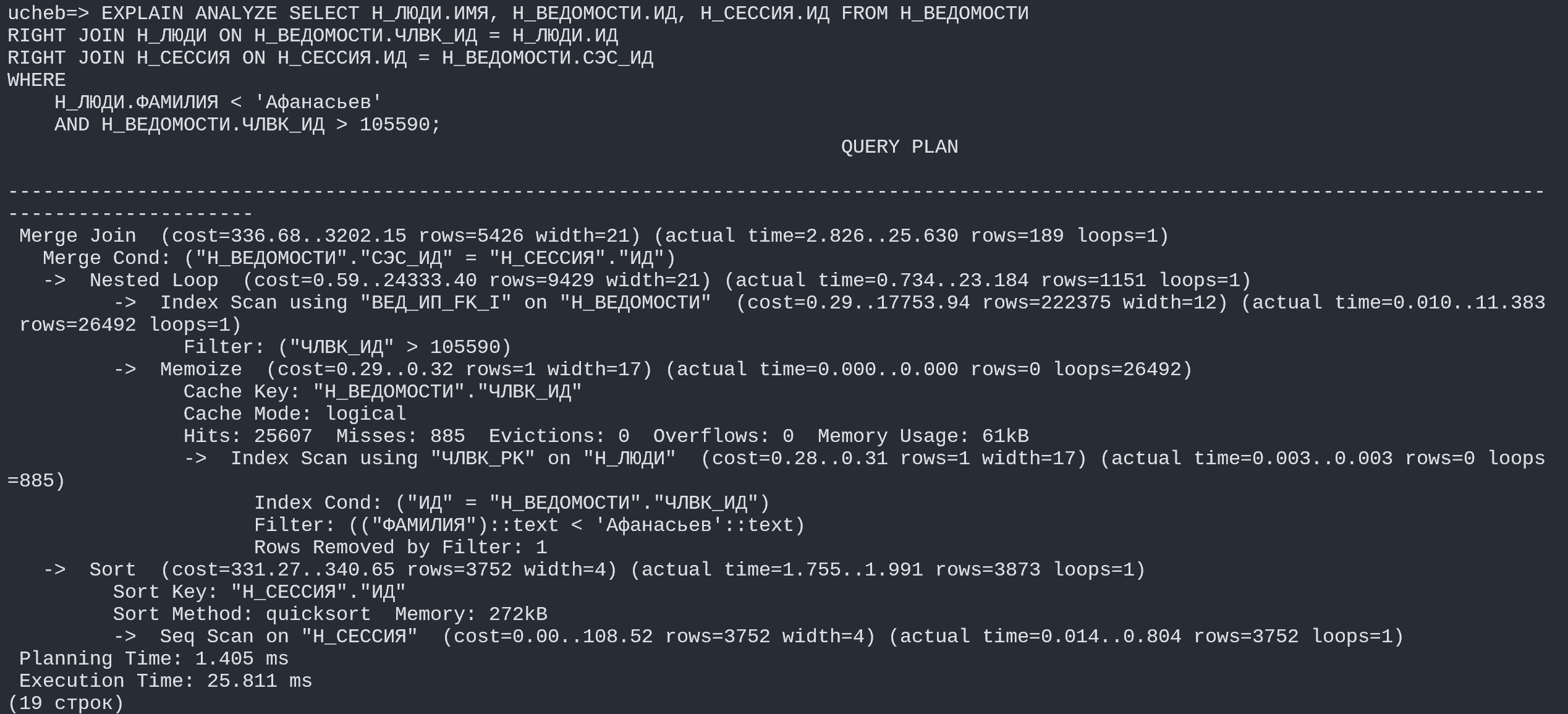
RIGHT JOIN Н\_ЛЮДИ ON Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД

RIGHT JOIN Н\_СЕССИЯ ON Н\_СЕССИЯ.ИД = Н\_ВЕДОМОСТИ.СЭС\_ИД

WHERE

    Н\_ЛЮДИ.ФАМИЛИЯ <'Афанасьев'

    AND Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 105590;



## Возможные планы выполнения запросов без индексов:

### 1 запрос



\*ошибка в плане 2. Соединять надо для того, чтобы корректно работало соединение, нужно добавить проекцию по Н\_ВЕДОМОСТИ.ОЦЕНКА

1 план:

1. Соединение Н\_ВЕДОМОСТИ.ИД = Н\_ОЦЕНКИ.КОД
2. Выборка Н\_ОЦЕНКИ.КОД <'2' и Н\_ВЕДОМОСТИ.ИД > 1250981, Н\_ВЕДОМОСТИ.ИД < 1457443

2 план:

1. Выборка Н\_ОЦЕНКИ.КОД <'2' и Н\_ВЕДОМОСТИ.ИД > 1250981, Н\_ВЕДОМОСТИ.ИД < 1457443
2. Соединение Н\_ВЕДОМОСТИ.ИД = Н\_ОЦЕНКИ.КОД

Оптимальным является 2 план, так как он проводит объединение таблиц по уже выбранным ранее атрибутам, а не по таблицам целиком.

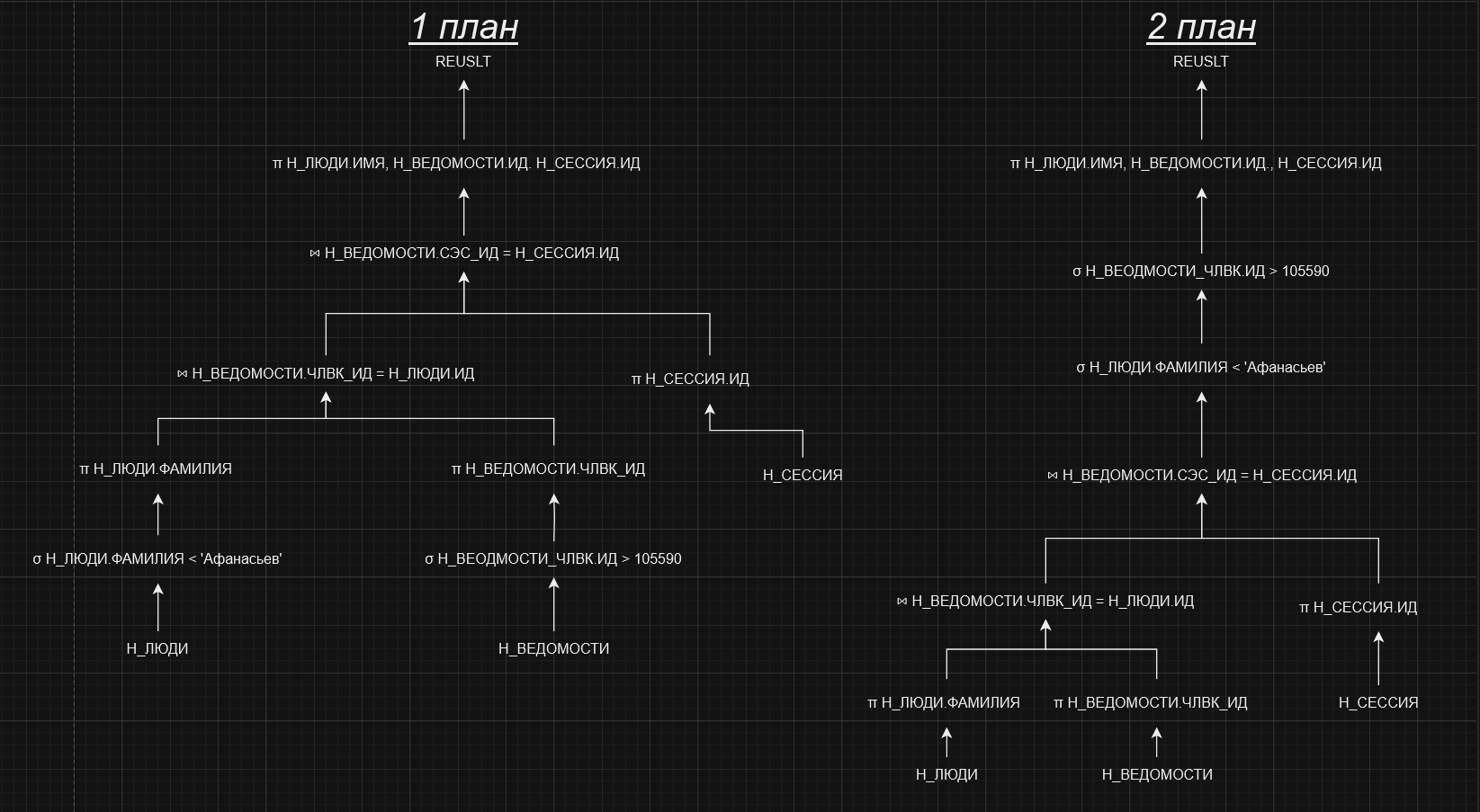
Индексы:

а) Для таблицы Н\_ВЕДОМОСТИ подойдёт B-tree индекс, так как он ускорит фильтрацию строк, использующих where [оператор сравнения].

* CREATE INDEX Ведомости\_Ид\_Индекс ON TABLE Н\_ВЕДОМОСТИ(ИД)

б) На таблице Н\_ОЦЕНКИ всего 9 возможных строк, так что индексы там не требуются.

### 2 запрос



\*ошибка в обоих планах. В левом: добавить проекцию Н\_ЛЮДИ.ИД. В правом: добавить проекцию Н\_ЛЮДИ.ИД

1 план:

1. Выборка Н\_ЛЮДИ.ФАМИЛИЯ <'Афанасьев' и Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 105590;
2. Соединение Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД
3. Соединение Н\_ВЕДОМОСТИ.СЭС\_ИД = Н\_СЕССИЯ.ИД

2 план:

1. Соединение Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД
2. Соединение Н\_ВЕДОМОСТИ.СЭС\_ИД = Н\_СЕССИЯ.ИД
3. Выборка Н\_ЛЮДИ.ФАМИЛИЯ <'Афанасьев' и Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 105590;

Оптимальным является 1 план, так как он проводит объединение таблиц по уже выбранным ранее атрибутам, а не по таблицам целиком.

Индексы:

а) Для таблицы Н\_ВЕДОМОСТИ подойдёт B-tree индекс, так как он ускорит фильтрацию строк, использующих where [оператор сравнения].

* CREATE INDEX Ведомости\_Ид\_Индекс ON Н\_ВЕДОМОСТИ(ИД)

б) Для таблицы Н\_ЛЮДИ **возможно** подойдёт GIST индекс, но только в случае, если там будет нетривиальный запрос, использующий вайлдкарды, который B-tree не сможет эффективно решить.

* CREATE INDEX Н\_ЛЮДИ\_ФАМИЛ\_ИНДЕКС ON Н\_ЛЮДИ USING GIST(LOWER(“Пуш%”))

в) Для таблицы Н\_СЕССИЯ в данном сценарии индекс не понадобится, так как её ИД мы получаем через join с Н\_ВЕДОМОСТИ

# Вывод

При выполнении лабораторной работы я познакомилась с использованием индексов для ускорения обработки запросов в SQL, а также планами выполнения запросов, их построением и анализом.