Федеральное государственное автономное образовательное высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники  
Направление подготовки: 09.03.01 – Информатика и вычислительная техника

**Отчет**

**по лабораторной работе №4**

**по дисциплине «Базы данных»**

Вариант 313100

Выполнил:

Колмаков Дмитрий Владимирович, P3131

Преподаватель:

Наумова Надежда Александровна

г. Санкт-Петербург, 2023 г.

Оглавление

[Задание 3](#_Toc134656054)

[Текст задания 3](#_Toc134656055)

[Ход работы 4](#_Toc134656056)

[Функциональные зависимости 4](#_Toc134656057)

[Нормальные формы 4](#_Toc134656058)

[Денормализация 5](#_Toc134656059)

[Триггеры 6](#_Toc134656060)

[Вывод 7](#_Toc134656061)

# Задание

## Текст задания

Составить запросы на языке SQL (пункты 1-2).

Для каждого запроса предложить индексы, добавление которых уменьшит время выполнения запроса (указать таблицы/атрибуты, для которых нужно добавить индексы, написать тип индекса; объяснить, почему добавление индекса будет полезным для данного запроса).

Для запросов 1-2 необходимо составить возможные планы выполнения запросов. Планы составляются на основании предположения, что в таблицах отсутствуют индексы. Из составленных планов необходимо выбрать оптимальный и объяснить свой выбор.  
Изменятся ли планы при добавлении индекса и как?

Для запросов 1-2 необходимо добавить в отчет вывод команды EXPLAIN ANALYZE [запрос]

Подробные ответы на все вышеперечисленные вопросы должны присутствовать в отчете (планы выполнения запросов должны быть нарисованы, ответы на вопросы - представлены в текстовом виде).

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.  
   Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД, Н\_ВЕДОМОСТИ.ИД.  
   Фильтры (AND):  
   a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД < 3.  
   b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 105590.  
   Вид соединения: LEFT JOIN.
2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.  
   Вывести атрибуты: Н\_ЛЮДИ.ИД, Н\_ВЕДОМОСТИ.ДАТА, Н\_СЕССИЯ.УЧГОД.  
   Фильтры (AND):  
   a) Н\_ЛЮДИ.ИМЯ < Роман.  
   b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 117219.  
   c) Н\_СЕССИЯ.ЧЛВК\_ИД < 106059.  
   Вид соединения: LEFT JOIN.

# Ход работы

## Реализация запросов на SQL

-- 1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
-- - Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.  
-- - Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД, Н\_ВЕДОМОСТИ.ИД.  
-- - Фильтры (AND):  
-- a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД < 3.  
-- b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 105590.  
-- - Вид соединения: LEFT JOIN.  
  
SELECT "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД", "Н\_ВЕДОМОСТИ"."ИД" FROM "Н\_ВЕДОМОСТИ"  
LEFT JOIN "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" ON "Н\_ВЕДОМОСТИ"."ТВ\_ИД" = "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД"  
WHERE "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" < 3  
 AND "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" = 105590;  
  
-- 2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
-- - Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.  
-- - Вывести атрибуты: Н\_ЛЮДИ.ИД, Н\_ВЕДОМОСТИ.ДАТА, Н\_СЕССИЯ.УЧГОД.  
-- - Фильтры (AND):  
-- a) Н\_ЛЮДИ.ИМЯ < Роман.  
-- b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 117219.  
-- c) Н\_СЕССИЯ.ЧЛВК\_ИД < 106059.  
-- - Вид соединения: LEFT JOIN.  
  
SELECT "Н\_ЛЮДИ"."ИД", "Н\_ВЕДОМОСТИ"."ДАТА", "Н\_СЕССИЯ"."УЧГОД" FROM "Н\_ВЕДОМОСТИ"  
LEFT JOIN "Н\_СЕССИЯ" on "Н\_ВЕДОМОСТИ"."СЭС\_ИД" = "Н\_СЕССИЯ"."СЭС\_ИД"  
LEFT JOIN "Н\_ЛЮДИ" on "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  
WHERE "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" > 117219  
 AND "Н\_СЕССИЯ"."ЧЛВК\_ИД" < 106059

AND "Н\_ЛЮДИ"."ИМЯ" < 'Роман';

## Первый запрос

### Уменьшение времени выполнения

**Индексы, добавление которых уменьшит время выполнения запроса:**

Н\_ТИПЫ\_ВЕДОМОСТЕЙ:

* Индекс на атрибуте ИД (B-tree): так как ИД является PK, ограничения первичного ключа автоматически создают индекс для столбцов первичного ключа (B-tree). Это ускоряет фильтрацию с условием "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" < 3.

Н\_ВЕДОМОСТИ:

* Индекс на атрибуте ЧЛВК\_ИД (B-tree): Это ускорит фильтрацию строк с условием "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" = 105590.
* Индекс на атрибуте ТВ\_ИД (B-tree): Это ускорит соединение таблиц по атрибуту ТВ\_ИД.

Добавление индексов на указанные столбцы позволит ускорить поиск нужных записей по фильтрам в запросе.

**Возможные планы выполнения запросов без индексов:**

План 1:

* Полный скан таблицы Н\_ТИПЫ\_ВЕДОМОСТЕЙ.
* Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" = 105590.
* Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ\_ИД.
* Фильтрация результата соединения по условию "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" < 3.

План 2:

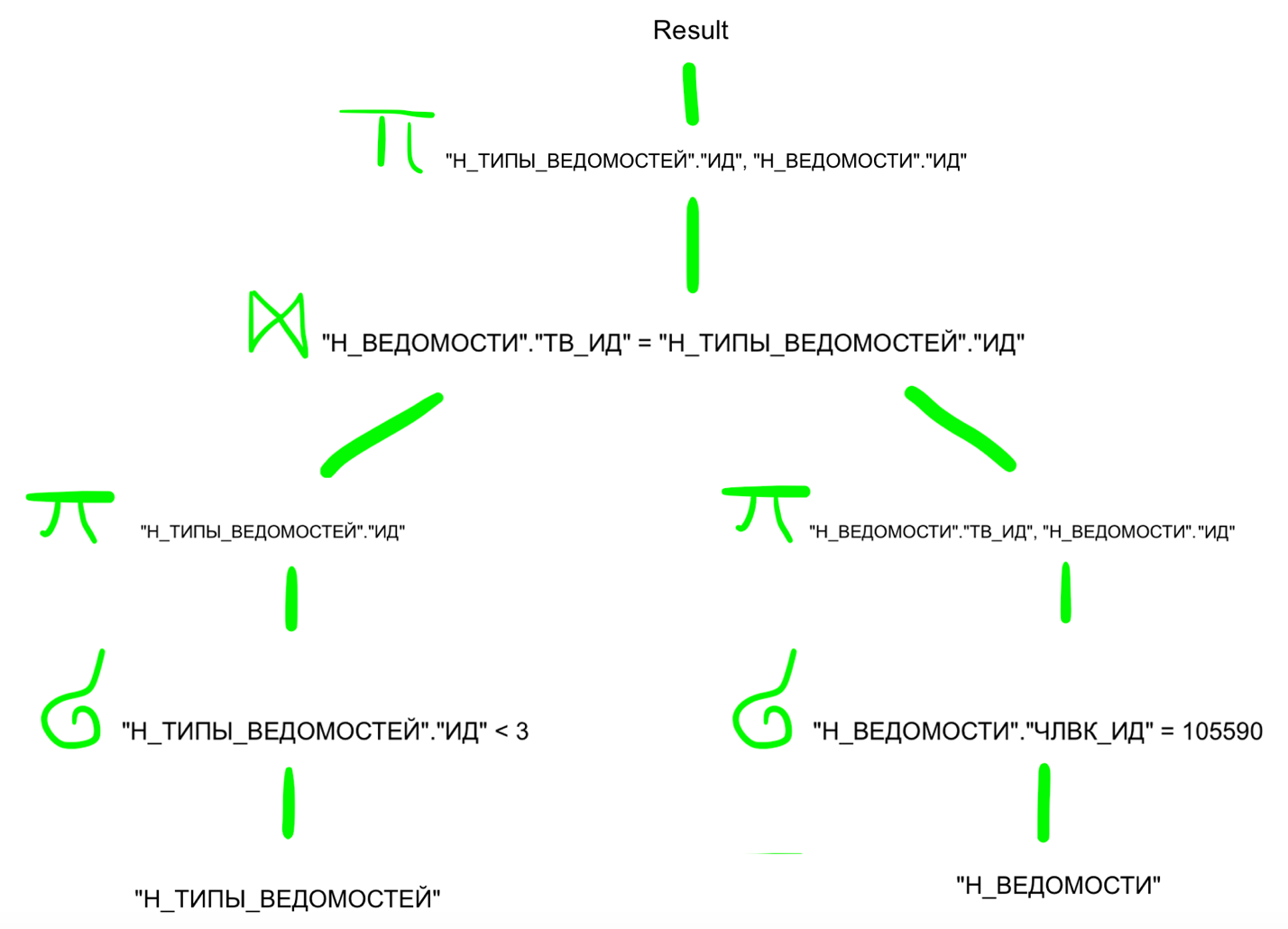
* Полный скан таблицы Н\_ТИПЫ\_ВЕДОМОСТЕЙ с применением фильтра по условию "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" < 3.
* Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" = 105590.
* Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ\_ИД.

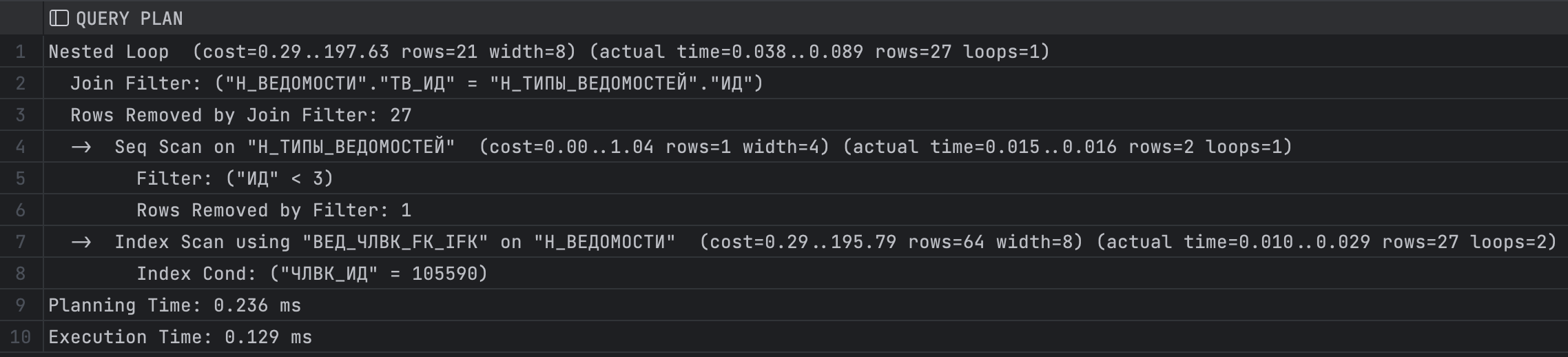
Оптимальный план - план 2, потому что фильтрация данных выполняется до соединения таблиц, что уменьшает количество строк для обработки.

При добавлении индексов планы выполнения запросов изменятся:

* Вместо полного скана таблиц будет использоваться индексный скан, ускорится фильтрация.
* Nested Loop Join станет быстрее благодаря индексу на атрибуте ТВ\_ИД, ускорится соединение таблиц.

### План выполнения запроса





## Второй запрос

### Уменьшение времени выполнения

**Индексы, добавление которых уменьшит время выполнения запроса:**

Н\_ВЕДОМОСТИ:

* Индекс на атрибуте СЭС\_ИД ИД (B-tree): Это ускорит соединение таблиц по атрибуту ТВ\_ИД.
* Индекс на атрибуте ЧЛВК\_ИД (B-tree): Это ускорит фильтрацию строк с условием "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" > 117219.

Н\_СЕССИЯ:

* Индекс на атрибуте СЭС\_ИД ИД (B-tree): Это ускорит соединение таблиц по атрибуту ТВ\_ИД.
* Индекс на атрибуте ЧЛВК\_ИД (B-tree): Это ускорит фильтрацию строк с условием "Н\_СЕССИЯ"."ЧЛВК\_ИД" < 106059.

Н\_ЛЮДИ:

* Индекс на атрибуте ИМЯ (B-tree): Это ускорит фильтрацию строк с условием "Н\_ЛЮДИ"."ИМЯ" < 'Роман'.

Добавление индексов на указанные столбцы позволит ускорить поиск нужных записей по фильтрам в запросе.

**Возможные планы выполнения запросов без индексов:**

План 1:

* Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" > 117219.
* Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию "Н\_ЛЮДИ"."ИМЯ" < 'Роман'.
* Соединение таблиц Н\_ВЕДОМОСТИ и Н\_ЛЮДИ с использованием Nested Loop Join по атрибуту ЧЛВК\_ИД.
* Полный скан таблицы Н\_СЕССИЯ с применением фильтра по условию "Н\_СЕССИЯ"."ЧЛВК\_ИД" < 106059.
* Соединение результатов предыдущего этапа с таблицей Н\_СЕССИЯ с использованием Nested Loop Join по атрибуту СЭС\_ИД.

План 2:

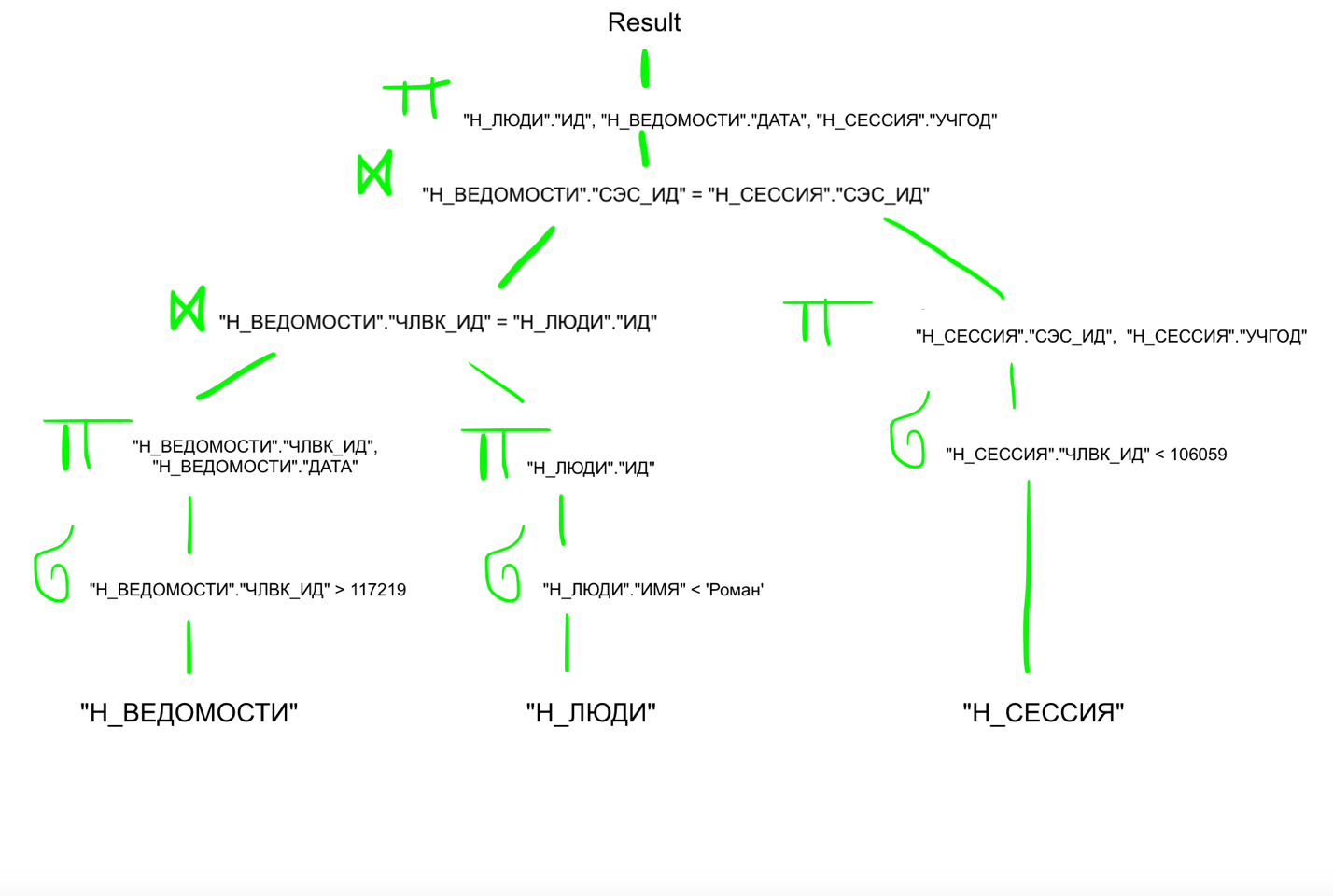
* Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" > 117219.
* Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию "Н\_ЛЮДИ"."ИМЯ" < 'Роман'.
* Соединение таблиц Н\_ВЕДОМОСТИ и Н\_ЛЮДИ с использованием Hash Join по атрибуту ЧЛВК\_ИД.
* Полный скан таблицы Н\_СЕССИЯ с применением фильтра по условию "Н\_СЕССИЯ"."ЧЛВК\_ИД" < 106059.
* Соединение результатов предыдущего этапа с таблицей Н\_СЕССИЯ с использованием Hash Join по атрибуту СЭС\_ИД.

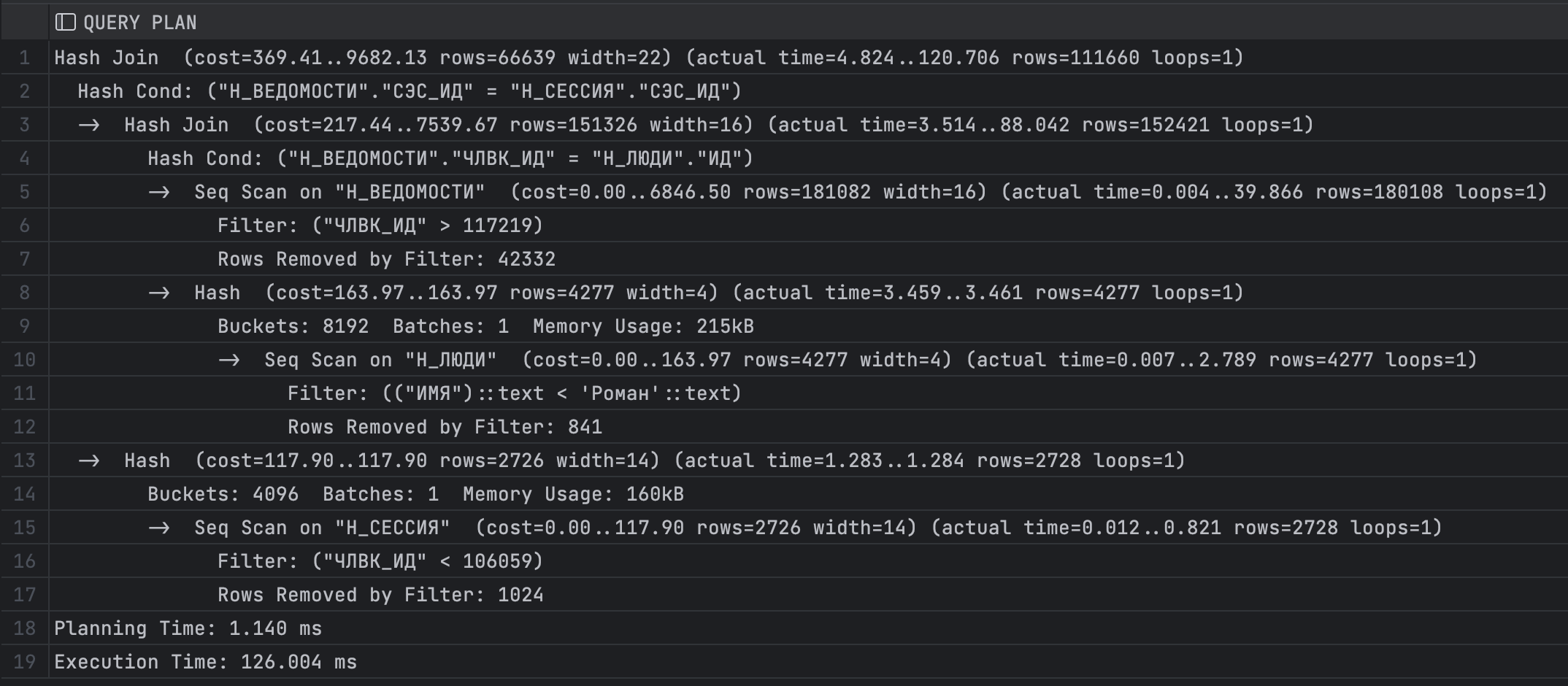
Оптимальный план - план 2, так как Hash Join предпочтительнее Nested Loop Join в случаях, когда обрабатываемые наборы данных большие, и исполняется JOIN с использованием условия эквивалентности.

При добавлении индексов планы выполнения запросов изменятся:

* Вместо полного скана таблиц будет использоваться индексный скан, ускорится фильтрация.
* Hash Join станет быстрее благодаря индексам на атрибутах ЧЛВК\_ИД и СЭС\_ИД, ускорится соединение таблиц.

### План выполнения запроса





# Вывод

В ходе выполнения лабораторной работы я освоил работу с реляционной алгеброй и научился строить планы выполнения запросов, а также их диаграммы. Я изучил различные виды индексов и узнал, как использовать их для оптимизации скорости выполнения запросов. Теперь я могу применять полученные знания для эффективной работы с базами данных и повышения производительности SQL-запросов.