Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет

ИТМО»

*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.01 - Информатика и вычислительная техника, Компьютерные системы и технологии*

*Дисциплина «Базы данных»*

**Отчет**

**По лабораторной работе №4**

**Вариант №1313**

Студент:

Миронов Иван Николаевич

Группа: Р3132

Преподаватель:

Афанасьев Дмитрий Борисович

Санкт-Петербург, 2024 год

# **Оглавление**

[Оглавление 2](#_Toc166514335)

[Текст задания 3](#_Toc166514336)

[Запрос №1 4](#_Toc166514337)

[Индексы, добавление которых уменьшит время выполнения запроса: 4](#_Toc166514338)

[Возможные планы выполнения запроса без индексов: 5](#_Toc166514339)

[План выполнения запроса 6](#_Toc166514340)

[Запрос №2 7](#_Toc166514341)

[Индексы, добавление которых уменьшит время выполнения запроса: 7](#_Toc166514342)

[Возможные планы выполнения запроса без индексов: 8](#_Toc166514343)

[План выполнения запроса 9](#_Toc166514344)

[Вывод 10](#_Toc166514345)

# 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.НАИМЕНОВАНИЕ, Н\_ВЕДОМОСТИ.ИД.  
   Фильтры (AND):  
   a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.  
   b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 163249.  
   Вид соединения: RIGHT JOIN.
2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.  
   Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.ИД.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ФАМИЛИЯ < Петров.  
   b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 105590.  
   c) Н\_УЧЕНИКИ.ИД > 150308.  
   Вид соединения: RIGHT JOIN.

# Запрос №1

*-- 1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
-- Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.  
-- Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н\_ВЕДОМОСТИ.ИД.  
-- Фильтры (AND):  
-- a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.  
-- b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 163249.  
-- Вид соединения: RIGHT JOIN.*

SELECT Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н\_ВЕДОМОСТИ.ИД FROM Н\_ТИПЫ\_ВЕДОМОСТЕЙ  
RIGHT JOIN Н\_ВЕДОМОСТИ ON Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД = Н\_ВЕДОМОСТИ.ТВ\_ИД  
WHERE Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 163249  
AND Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Ведомость';

## Индексы, добавление которых уменьшит время выполнения запроса:

a) На таблице Н\_ТИПЫ\_ВЕДОМОСТЕЙ:

* Индекс на атрибуте НАИМЕНОВАНИЕ (Hash-index) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Ведомость'

Hash-index оптимален при точечных запросах

b) На таблице Н\_ВЕДОМОСТИ:

* Индекс на атрибуте ЧЛВК\_ИД (Hash-index)

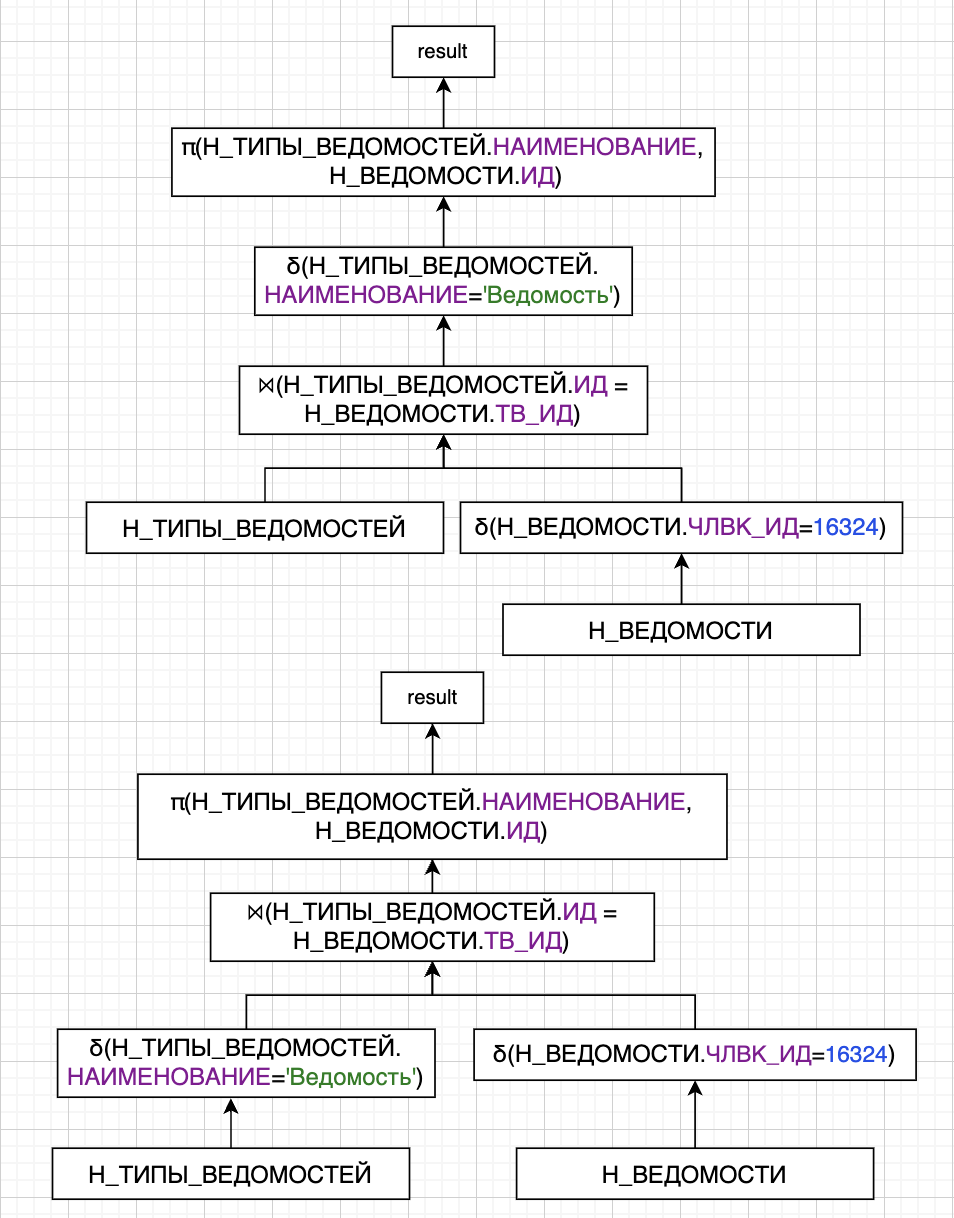
Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 163249

Hash-index оптимален при точечных запроса

* Индекс на атрибуте ТВ\_ИД (Hash-index)

CREATE INDEX "ИНД\_НАИМЕНОВАНИЕ" ON "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" USING HASH("НАИМЕНОВАНИЕ");  
CREATE INDEX "ИНД\_ЧЛВК\_ИД" ON "Н\_ВЕДОМОСТИ" USING HASH("ЧЛВК\_ИД");  
CREATE INDEX "ИНД\_ТВ\_ИД" ON "Н\_ВЕДОМОСТИ" USING HASH("ТВ\_ИД");

## Возможные планы выполнения запроса без индексов:



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.
* Nested Loop Join станет быстрее благодаря индексу на атрибуте ТВ\_ИД.

## План выполнения запроса

Н\_ВЕДОМОСТИ.ЧЛВК\_ИД=163249 изменен на Н\_ВЕДОМОСТИ.ЧЛВК\_ИД=142390

QUERY PLAN  
*-----------------------------------------------------------------------------------------------------------------* Nested Loop (cost=0.29..201.22 rows=22 width=30) (actual time=0.023..0.156 rows=89 loops=1)  
 Join Filter: ("Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "Н\_ВЕДОМОСТИ"."ТВ\_ИД")  
 Rows Removed by Join Filter: 3  
 -> Seq Scan on "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=30) (actual time=0.010..0.011 rows=1 loops=1)  
 Filter: (("НАИМЕНОВАНИЕ")::text = 'Ведомость'::text)  
 Rows Removed by Filter: 2  
 -> Index Scan using "ВЕД\_ЧЛВК\_FK\_IFK" on "Н\_ВЕДОМОСТИ" (cost=0.29..199.37 rows=65 width=8) (actual time=0.009..0.121 rows=92 loops=1)  
 Index Cond: ("ЧЛВК\_ИД" = 142390)  
 Planning Time: 0.254 ms  
 Execution Time: 0.191 ms

QUERY PLAN  
*-----------------------------------------------------------------------------------------------------------------* Nested Loop (cost=4.14..76.13 rows=1 width=422) (actual time=0.014..0.015 rows=0 loops=1)  
 Join Filter: ("Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "Н\_ВЕДОМОСТИ"."ТВ\_ИД")  
 -> Index Scan using "ИНД\_НАИМЕНОВАНИЕ" on "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" (cost=0.00..8.02 rows=1 width=422) (actual time=0.013..0.013 rows=0 loops=1)  
 Index Cond: (("НАИМЕНОВАНИЕ")::text = 'Ведомость'::text)  
 -> Bitmap Heap Scan on "Н\_ВЕДОМОСТИ" (cost=4.14..67.89 rows=18 width=8) (never executed)  
 Recheck Cond: ("ЧЛВК\_ИД" = 163249)  
 -> Bitmap Index Scan on "ИНД\_ЧЛВК\_ИД" (cost=0.00..4.13 rows=18 width=0) (never executed)  
 Index Cond: ("ЧЛВК\_ИД" = 163249)  
 Planning Time: 0.713 ms  
 Execution Time: 0.084 ms

# Запрос №2

*-- 2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
-- Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.  
-- Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.ИД.  
-- Фильтры: (AND)  
-- a) Н\_ЛЮДИ.ФАМИЛИЯ < Петров.  
-- b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 105590.  
-- c) Н\_УЧЕНИКИ.ИД > 150308.  
-- Вид соединения: RIGHT JOIN.*

SELECT Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.ИД FROM Н\_ОБУЧЕНИЯ  
RIGHT JOIN Н\_ЛЮДИ ON Н\_ЛЮДИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД  
RIGHT JOIN Н\_УЧЕНИКИ ON Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД  
WHERE Н\_ЛЮДИ.ФАМИЛИЯ < 'Петров'

AND Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 105590

AND Н\_УЧЕНИКИ.ИД > 150308;

## Индексы, добавление которых уменьшит время выполнения запроса:

a) На таблице Н\_ЛЮДИ:

* Индекс на атрибуте ФАМИЛИЯ (B-tree)

Н\_ЛЮДИ.ФАМИЛИЯ < 'Петров'

Оператор сравнения, поэтому B-tree

b) На таблице Н\_ОБУЧЕНИЯ:

* Индекс на атрибуте ЧЛВК\_ИД (Hash-index)

Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 105590

JOIN Н\_УЧЕНИКИ ON Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД

Hash-index оптимален при точечных запросах

с) На таблице Н\_УЧЕНИКИ:

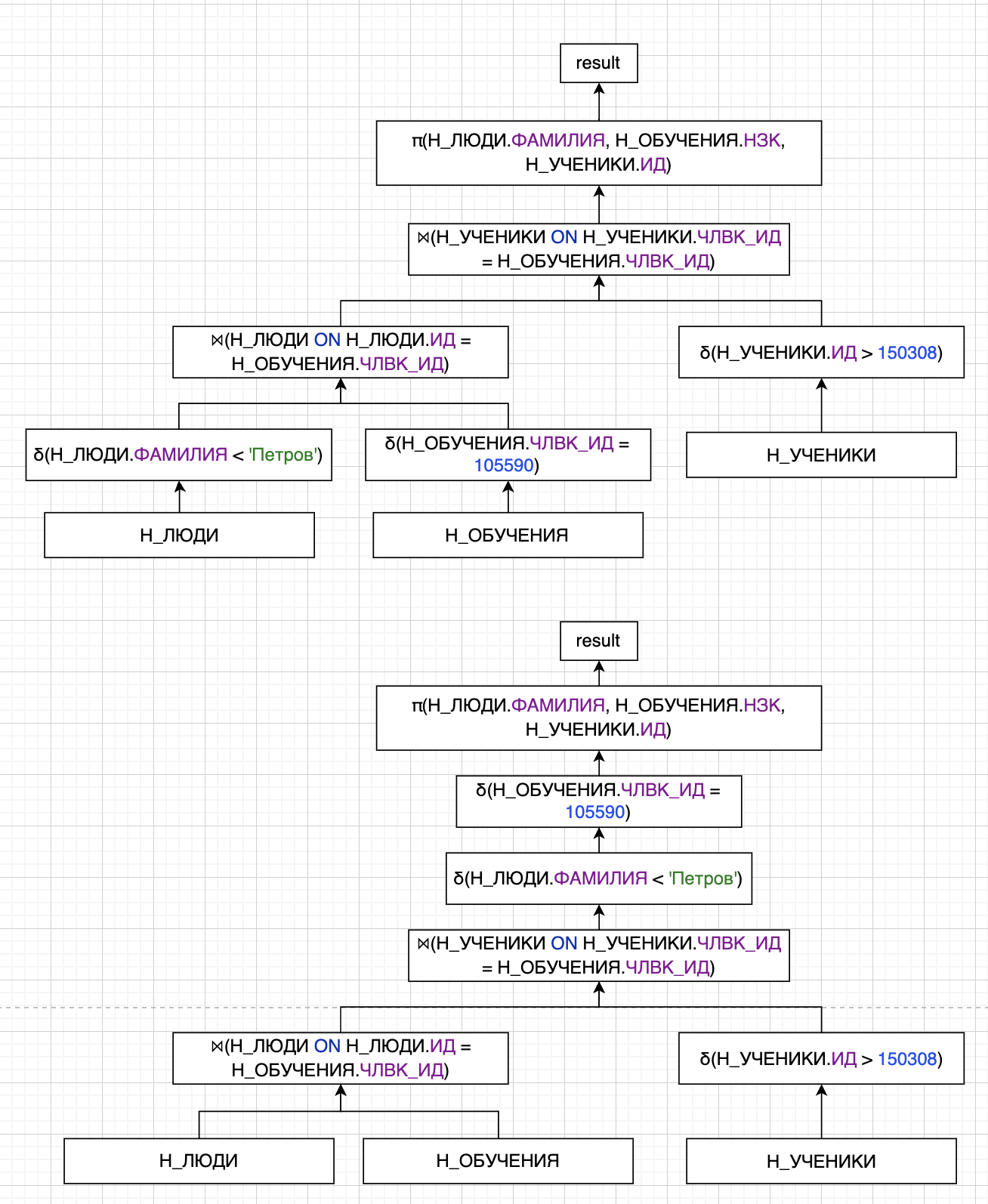
* Индекс на атрибуте ЧЛВК\_ИД (B-tree)

JOIN Н\_УЧЕНИКИ ON Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД Н\_УЧЕНИКИ.ИД > 150308

Оператор сравнения, поэтому B-tree

CREATE INDEX "ИНД\_ФАМИЛИЯ" ON "Н\_ЛЮДИ" USING BTREE("ФАМИЛИЯ");  
CREATE INDEX "ИНД\_ОБУЧЕНИЯ\_ЧЛВК\_ИД" ON "Н\_ОБУЧЕНИЯ" USING HASH("ЧЛВК\_ИД");  
CREATE INDEX "ИНД\_УЧЕНИКИ\_ЧЛВК\_ИД" ON "Н\_УЧЕНИКИ" USING BTREE("ЧЛВК\_ИД");

## Возможные планы выполнения запроса без индексов:



Оптимальный план:

* План 1, 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.
* Nested Loop Join станет быстрее благодаря индексу на атрибуте Н\_УЧЕНИКИ.ЧЛВК\_ИД и Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД.

## План выполнения запроса

Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 105590 изменен на Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 119457

QUERY PLAN  
*---------------------------------------------------------------------------------------------------------------------------* Nested Loop (cost=4.98..80.29 rows=5 width=26) (actual time=0.059..0.083 rows=4 loops=1)  
 -> Nested Loop (cost=0.56..16.61 rows=1 width=26) (actual time=0.038..0.039 rows=1 loops=1)  
 -> Index Scan using "ОБУЧ\_ЧЛВК\_FK\_I" on "Н\_ОБУЧЕНИЯ" (cost=0.28..8.30 rows=1 width=10) (actual time=0.021..0.021 rows=1 loops=1)  
 Index Cond: ("ЧЛВК\_ИД" = 119457)  
 -> Index Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" (cost=0.28..8.30 rows=1 width=20) (actual time=0.013..0.014 rows=1 loops=1)  
 Index Cond: ("ИД" = 119457)  
 Filter: (("ФАМИЛИЯ")::text < 'Петров'::text)  
 -> Bitmap Heap Scan on "Н\_УЧЕНИКИ" (cost=4.42..63.63 rows=5 width=8) (actual time=0.018..0.040 rows=4 loops=1)  
 Recheck Cond: ("ЧЛВК\_ИД" = 119457)  
 Filter: ("ИД" > 150308)  
 Rows Removed by Filter: 13  
 Heap Blocks: exact=17  
 -> Bitmap Index Scan on "УЧЕН\_ОБУЧ\_FK\_I" (cost=0.00..4.42 rows=17 width=0) (actual time=0.010..0.011 rows=17 loops=1)  
 Index Cond: ("ЧЛВК\_ИД" = 119457)  
 Planning Time: 0.416 ms  
 Execution Time: 0.134 ms

QUERY PLAN  
*-----------------------------------------------------------------------------------------------------------------* Nested Loop (cost=0.14..27.10 rows=1 width=106) (actual time=0.042..0.043 rows=0 loops=1)  
 -> Nested Loop (cost=0.00..18.93 rows=1 width=106) (actual time=0.041..0.042 rows=0 loops=1)  
 -> Index Scan using "ИНД\_ОБУЧЕНИЯ\_ЧЛВК\_ИД" on "Н\_ОБУЧЕНИЯ" (cost=0.00..8.02 rows=1 width=38) (actual time=0.041..0.041 rows=0 loops=1)  
 Index Cond: ("ЧЛВК\_ИД" = 105590)  
 -> Seq Scan on "Н\_ЛЮДИ" (cost=0.00..10.90 rows=1 width=72) (never executed)  
 Filter: ((("ФАМИЛИЯ")::text < 'Петров'::text) AND ("ИД" = 105590))  
 -> Index Scan using "ИНД\_УЧЕНИКИ\_ЧЛВК\_ИД" on "Н\_УЧЕНИКИ" (cost=0.14..8.16 rows=1 width=8) (never executed)  
 Index Cond: ("ЧЛВК\_ИД" = 105590)  
 Filter: ("ИД" > 150308)  
 Planning Time: 5.291 ms  
 Execution Time: 0.119 ms

# Вывод

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