МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное

автономное образовательное учреждение высшего образования

 «**Национальный исследовательский университет ИТМО**»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №4**

**«Исследование устройства баз данных»**

по дисциплине

«ИНФОРМАЦИОННЫЕ СИСТЕМЫ И БАЗЫ ДАННЫХ»

Вариант 76532

***Выполнила:***

Студентка группы P3118

Богданова Мария   
Михайловна

***Преподаватель:***

Инячина Диана

Александровна

Санкт-Петербург, 2023

**Содержание.**

Задание…………..……………………………………………….….3

Ход работы………………….…………………….............................4-5

Вывод…………...……………………………………………………6

**Задание.**

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:  
   Таблицы: Н\_ОЦЕНКИ, Н\_ВЕДОМОСТИ.  
   Вывести атрибуты: Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ, Н\_ВЕДОМОСТИ.ИД.  
   Фильтры (AND):  
   a) Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ < незачет.  
   b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 142390.  
   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.ИД, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ФАМИЛИЯ = Ёлкин.  
   b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 163484.  
   c) Н\_УЧЕНИКИ.ГРУППА < 4103.  
   Вид соединения: INNER JOIN.

**Составление запросов.**

**Запрос №1.**

Таблицы: Н\_ОЦЕНКИ, Н\_ВЕДОМОСТИ.

Вывести атрибуты: Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ, Н\_ВЕДОМОСТИ.ИД

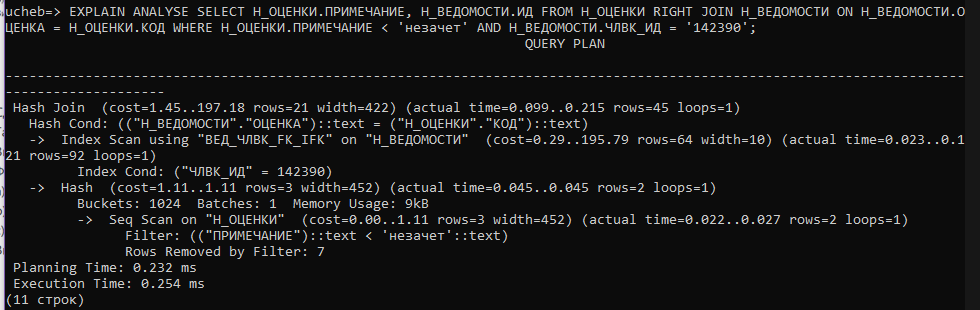
Фильтры (AND):

a) Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ < незачет.

b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 142390.

Вид соединения: RIGHT JOIN.

SELECT Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ, Н\_ВЕДОМОСТИ.ИД FROM Н\_ОЦЕНКИ RIGHT JOIN Н\_ВЕДОМОСТИ ON Н\_ВЕДОМОСТИ.ОЦЕНКА = Н\_ОЦЕНКИ.КОД WHERE Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ < 'незачет' AND Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = '142390';

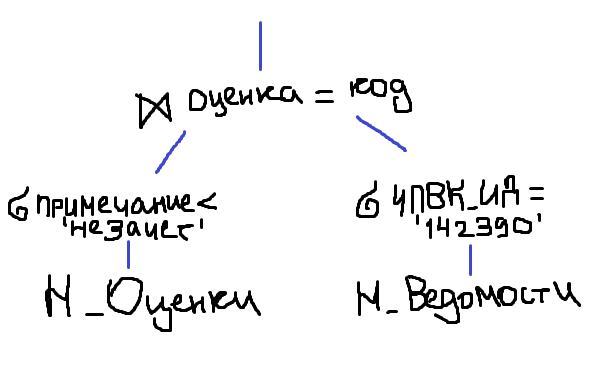


Оптимизировать запрос можно, применив индекс GIN к таблице "Н\_ОЦЕНКИ" к столбцу "ПРИМЕЧАНИЕ", т.к. выполняется полнотекстовый поиск.

Оптимизировать запрос можно, применив индекс Hash-index к таблице "Н\_ВЕДОМОСТИ" к столбцу "ЧЛВК\_ИД", т.к. используется оператор "=".

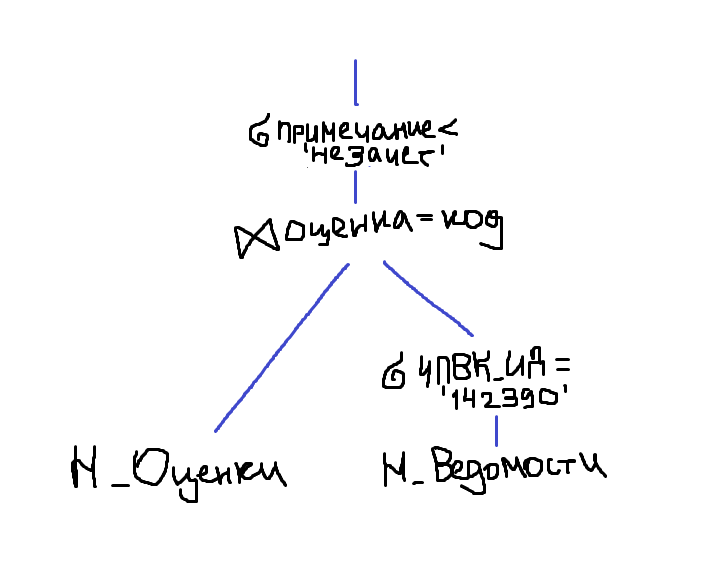
**План выполнения с использованием HASH JOIN:**

1. Полный скан таблицы Н\_ОЦЕНКИ С ПРИМЕНЕНИЕМ ФИЛЬТРА Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ < ‘незачет’
2. Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = ‘142390’
3. Соединение таблиц с использованием HASH JOIN по атрибутами Н\_ВЕДОМОСТИ.ОЦЕНКА = Н\_ОЦЕНКИ.КОД по условию RIGHT JOIN



**План выполнения с использованием Nested Loops:**

1. Полный скан таблицы Н\_ОЦЕНКИ
2. Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = ‘142390’
3. Соединение таблиц с использованием Nested Loops Join по атрибутам Н\_ВЕДОМОСТИ.ОЦЕНКА = Н\_ОЦЕНКИ.КОД по условию RIGHT JOIN
4. Фильтрация результата соединения с применением фильтра Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ < ‘незачет’



1. В данном случае, при отсутствии индексов, оптимальным планом выполнения запроса будет план с использованием Hash Join. Хэш-соединение обеспечивает более эффективное объединение результатов двух фильтров по условию RIGHT JOIN, т. е. оптимальным является первый план, так как фильтрация выполняется до соединения, что уменьшает количество обрабатываемых строк.
2. При добавлении индексов вместо полного скана будет использоваться индексный скан, а Nested Loops Join станет работать быстрее благодаря индексу на атрибутах Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

**Запрос №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.ИД, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО.**

**Фильтры: (AND)**

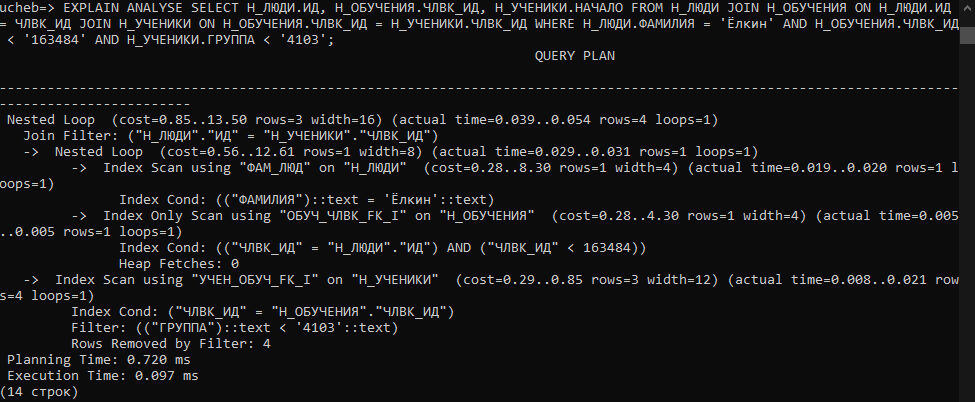
**a) Н\_ЛЮДИ.ФАМИЛИЯ = Ёлкин.**

**b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 163484.**

**c) Н\_УЧЕНИКИ.ГРУППА < 4103.**

**Вид соединения: INNER JOIN.**

SELECT Н\_ЛЮДИ.ИД, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО FROM Н\_ЛЮДИ JOIN Н\_ОБУЧЕНИЯ ON Н\_ЛЮДИ.ИД = ЧЛВК\_ИД JOIN Н\_УЧЕНИКИ ON Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД WHERE Н\_ЛЮДИ.ФАМИЛИЯ = 'Ёлкин' AND Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД < '163484' AND Н\_УЧЕНИКИ.ГРУППА < '4103';



Было изменено условие, т.к. первоначальный вариант запроса ничего не выводил. (члвк\_ид > '163483')

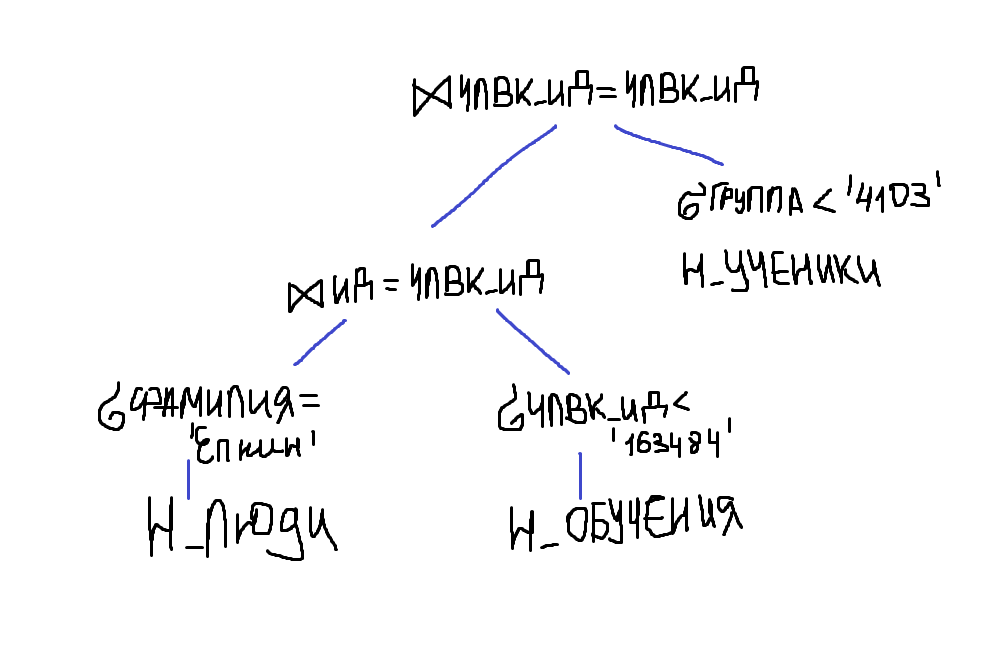
Оптимизировать запрос можно, применив индекс B-tree к таблице "Н\_ОБУЧЕНИЯ" столбцу "ЧЛВК\_ИД" и к таблице "Н\_УЧЕНИКИ" к столбцу "группа", т.к. используется оператор сравнения "<" и нам не помешала бы сортировка этого столбца.

Оптимизировать запрос также можно, применив индекс GIN к таблице "Н\_ЛЮДИ" к столбцу "ФАМИЛИЯ", т.к. выполняется полнотекстовый поиск.

Оптимизировать запрос можно также, применив hash-индекс к таблице "Н\_УЧЕНИКИ" к столбцу "ЧЛВК\_ИД", т.к используется оператор "=".

**План выполнения с использованием NESTED LOOP:**

1. Полный скан таблицы Н\_ЛЮДИ с применением фильтра Н\_ЛЮДИ.ФАМИЛИЯ = Ёлкин
2. Полный скан таблицы Н\_ОБУЧЕНИЯ с применением ФИЛЬТРА Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД < ‘163484’
3. Соединение таблиц Н\_ОБУЧЕНИЯ И Н\_ЛЮДИ по атрибутам Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД с использованием NESTED LOOP JOIN
4. Полный скан таблицы Н\_УЧЕНИКИ с фильтром Н\_УЧЕНИКИ.ГРУППА < ‘4103’
5. Объединение результата п.3 и таблицы Н\_УЧЕНИКИ по атрибутам Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД NESTED LOOP JOIN



**План выполнения с использованием HASH JOIN:**

1. Полный скан таблицы Н\_ЛЮДИ с применением фильтра Н\_ЛЮДИ.ФАМИЛИЯ = Ёлкин
2. Полный скан таблицы Н\_ОБУЧЕНИЯ с применением ФИЛЬТРА Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД < ‘163484’
3. Соединение таблиц Н\_ОБУЧЕНИЯ И Н\_ЛЮДИ по атрибутам Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД с использованием HASH JOIN
4. Полный скан таблицы Н\_УЧЕНИКИ с фильтром Н\_УЧЕНИКИ.ГРУППА < ‘4103’
5. Объединение результата п.3 и таблицы Н\_УЧЕНИКИ по атрибутам Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД HASH JOIN
6. Оптимальным является первый план, так как в ситуациях, когда обрабатываемые наборы небольшие и при этом известно, что результаты фильтрации будут содержать малое количество строк, использование Nested Loops Join оптимальнее, чем Hash Join.
7. При добавлении индексов вместо полного скана будет использоваться индексный скан, а Nested Loops Join станет быстрее благодаря индексу на атрибутах Н\_ЛЮДИ.ИД и Н\_УЧЕНИКИ.ЧЛВК\_ИД.

**Вывод.**

В ходе выполнения данной лабораторной работы я научилась составлять планы запросов и узнала, что такое индексы в SQL.