МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРОНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №4**

по дисциплине

«Базы данных»

***Выполнил работу:***

Студент группы P3118

Рамеев Тимур Ильгизович

***Преподаватель:***

Инячина Диана

Содержание

[Задание 3](#_Toc147147720)

[Запрос №1 4](#_Toc147147721)

[Планы 5](#_Toc147147722)

[Анализ 5](#_Toc147147723)

[Запрос № 2 5](#_Toc147147724)

[Планы 5](#_Toc147147725)

[Анализ 5](#_Toc147147726)

[Влияние индексов на планы 5](#_Toc147147727)

[Вывод 5](#_Toc147147728)

# 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).

Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ.  
Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ДАТА.  
Фильтры (AND):  
a) Н\_ЛЮДИ.ФАМИЛИЯ > Петров.  
b) Н\_ВЕДОМОСТИ.ИД = 1457443.  
c) Н\_ВЕДОМОСТИ.ИД > 1250972.  
Вид соединения: RIGHT JOIN.

Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.  
Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД, Н\_СЕССИЯ.ИД.  
Фильтры (AND):  
a) Н\_ЛЮДИ.ФАМИЛИЯ = Афанасьев.  
b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 117219.  
Вид соединения: RIGHT JOIN.

# Запрос №1

Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ.  
Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ДАТА.  
Фильтры (AND):  
a) Н\_ЛЮДИ.ФАМИЛИЯ > Петров.  
b) Н\_ВЕДОМОСТИ.ИД = 1457443. -- null (убрал в SQL запросе)  
c) Н\_ВЕДОМОСТИ.ИД > 1250972.  
Вид соединения: RIGHT JOIN.

SQL:

select Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ДАТА from Н\_ЛЮДИ

right join Н\_ВЕДОМОСТИ on Н\_ЛЮДИ.ИД=Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

where Н\_ЛЮДИ.ФАМИЛИЯ > 'Петров' and

Н\_ВЕДОМОСТИ.ИД > 1250972;

Для оптимизации запроса необходимо использовать индекс по атрибуту Н\_ВЕДОМОСТИ.ИД с типом BTREE, так как он фигурирует в выборке WHERE. Также, стоит создать индекс для атрибута Н\_ЛЮДИ.ФАМИЛИЯ, который тоже присутствует в выборке.

К тому же, для оптимизации запроса целесообразно использовать многоатрибутный индекс для таблиц Н\_ВЕДОМОСТИ и Н\_ЛЮДИ, включив в индексы атрибуты Н\_ЛЮДИ.ИД и Н\_ВЕДОМОСТИ.ЧЛВК\_ИД участвующие в объединении JOIN.

SQL:

Create index Н\_ВЕДОМОСТИ\_ИНДЕКС on Н\_ВЕДОМОСТИ(btree) (ЧЛВК\_ИД, Н\_ВЕДОМОСТИ);

Create index Н\_ЛЮДИ\_ИНДЕКС on Н\_ЛЮДИ(btree) (ИД, ФАМИЛИЯ);

## Изображение выглядит как текст, число, линия, Шрифт Автоматически созданное описаниеПланы

Представленный план неэффективен, так как операции выборки и проекции производятся в конце, после операции соединения, что негативно сказывается на время выполнения запроса.

## Изображение выглядит как текст, линия, диаграмма, число Автоматически созданное описание

В данном случае представленный план будет наиболее эффективным, так как представить план со структурой левостороннего дерева нельзя. Выборка и проекции отношений производятся до операции соединения.

## Анализ

QUERY PLAN

-----------------------------------------------------------------------------------------------------------------------------------

Hash Join (cost=677.44..5138.48 rows=9246 width=28) (actual time=4.566..13.873 rows=8788 loops=1)

Hash Cond: ("Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД")

-> Bitmap Heap Scan on "Н\_ВЕДОМОСТИ" (cost=490.81..4883.25 rows=26115 width=12) (actual time=1.326..5.789 rows=26134 loops=1)

Recheck Cond: ("ИД" > 1250972)

Heap Blocks: exact=642

-> Bitmap Index Scan on "ВЕД\_PK" (cost=0.00..484.28 rows=26115 width=0) (actual time=1.237..1.237 rows=26134 loops=1)

Index Cond: ("ИД" > 1250972)

-> Hash (cost=163.97..163.97 rows=1812 width=24) (actual time=3.227..3.228 rows=1792 loops=1)

Buckets: 2048 Batches: 1 Memory Usage: 118kB

-> Seq Scan on "Н\_ЛЮДИ" (cost=0.00..163.97 rows=1812 width=24) (actual time=0.018..2.886 rows=1792 loops=1)

Filter: (("ФАМИЛИЯ")::text > 'Петров'::text)

Rows Removed by Filter: 3326

Planning Time: 0.496 ms

Execution Time: 14.337 ms

# Запрос № 2

Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.  
Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД, Н\_СЕССИЯ.ИД.  
Фильтры (AND):  
a) Н\_ЛЮДИ.ФАМИЛИЯ = Афанасьев.  
b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 117219.  
Вид соединения: RIGHT JOIN.

SQL:

select Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД, Н\_СЕССИЯ.ИД from Н\_ЛЮДИ

right join Н\_ВЕДОМОСТИ on Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД

right join Н\_СЕССИЯ on Н\_СЕССИЯ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД

where Н\_ЛЮДИ.ФАМИЛИЯ = 'Афанасьев' and

Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 117219;

Для оптимизации запроса необходимо использовать индекс по атрибуту Н\_ВЕДОМОСТИ.ЧЛВК\_ИД с типом BTREE, так как он фигурирует в выборке WHERE. Также, стоит создать индекс для атрибута Н\_ЛЮДИ.ФАМИЛИЯ, который тоже присутствует в выборке.

К тому же, для оптимизации запроса целесообразно использовать многоатрибутный индекс для таблицы Н\_ЛЮДИ, включив в индекс атрибут Н\_ЛЮДИ.ИД участвующий в объединении JOIN.

SQL:

Create index Н\_ВЕДОМОСТИ\_ИНДЕКС on Н\_ВЕДОМОСТИ(btree) (ЧЛВК\_ИД);

Create index Н\_ЛЮДИ\_ИНДЕКС on Н\_ЛЮДИ(hash) (ИД, ФАМИЛИЯ);

## Изображение выглядит как текст, линия, число, снимок экрана Автоматически созданное описаниеПланы

Представленный план, несмотря на структуру левостороннего дерева, является неэффективным ввиду того, что операции выборки и проекции производятся в конце, после операций соединения, что сильно замедляет выполнение запроса, так как необходимо производить больше обменов с памятью.

Изображение выглядит как текст, диаграмма, линия, число

Автоматически созданное описание

Данный план является наиболее эффективным, так как в нем при его строении левостороннего дерева соблюден порядок выполнения операций – сначала – выборка и проекции, потом – соединение.

## Анализ

QUERY PLAN

----------------------------------------------------------------------------------------------------------------------------------------------

Nested Loop (cost=32.77..179.94 rows=69 width=24) (actual time=0.477..1.061 rows=38 loops=1)

Join Filter: ("Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД")

-> Hash Join (cost=32.48..150.86 rows=6 width=28) (actual time=0.466..1.022 rows=2 loops=1)

Hash Cond: ("Н\_СЕССИЯ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД")

-> Seq Scan on "Н\_СЕССИЯ" (cost=0.00..108.52 rows=3752 width=8) (actual time=0.009..0.525 rows=3752 loops=1)

-> Hash (cost=32.36..32.36 rows=9 width=20) (actual time=0.060..0.061 rows=9 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 9kB

-> Bitmap Heap Scan on "Н\_ЛЮДИ" (cost=4.35..32.36 rows=9 width=20) (actual time=0.039..0.053 rows=9 loops=1)

Recheck Cond: (("ФАМИЛИЯ")::text = 'Афанасьев'::text)

Heap Blocks: exact=8

-> Bitmap Index Scan on "ФАМ\_ЛЮД" (cost=0.00..4.35 rows=9 width=0) (actual time=0.028..0.029 rows=9 loops=1)

Index Cond: (("ФАМИЛИЯ")::text = 'Афанасьев'::text)

-> Index Only Scan using "ВЕД\_ЧЛВК\_FK\_IFK" on "Н\_ВЕДОМОСТИ" (cost=0.29..4.68 rows=13 width=4) (actual time=0.005..0.015 rows=19 loops=2)

Index Cond: (("ЧЛВК\_ИД" = "Н\_СЕССИЯ"."ЧЛВК\_ИД") AND ("ЧЛВК\_ИД" < 117219))

Heap Fetches: 38

Planning Time: 0.691 ms

Execution Time: 1.130 ms

# Влияние индексов на планы

Наличие индексов для отношений, использующихся в запросе, не меняют самой структуры плана, но позволяют более эффективно исполнять операции выборки и тета-соединения. Иначе говоря, оптимизируется фактическая реализация операций.

# Вывод

При выполнении лабораторной работы я познакомился с индексами в СУБД PSQL, планированием запросов и основами реляционной алгебры. Я составил планы для 2-х запросов и выбрал самый эффективный для каждого