

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

НИУ ИТМО

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И
КОМПЬЮТЕРНОЙ ТЕХНИКИ**

ЛАБОРАТОРНАЯ РАБОТА № 4

по дисциплине

Информационные системы и базы данных

Вариант 83476519

Выполнил:

Студент группы Р33092

Скрябин Иван Александрович,

Студент группы Р33102

Тюрин Иван Николаевич

Преподаватель:

Сагайдак Алина Алексеевна

Санкт-Петербург

2023

Задание

Введите вариант:

Внимание! У разных вариантов разный 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. Изменяются 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_ЛЮДИ, Н_ВЕДОМОСТИ.

Вывести атрибуты: Н_ЛЮДИ.ИМЯ, Н_ВЕДОМОСТИ.ИД.

Фильтры (AND):

a) Н_ЛЮДИ.ФАМИЛИЯ < Афанасьев.

b) Н_ВЕДОМОСТИ.ИД > 39921.

c) Н_ВЕДОМОСТИ.ИД < 1250981.

Вид соединения: INNER 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.ФАМИЛИЯ, Н_ВЕДОМОСТИ.ЧЛВК_ИД, Н_СЕССИЯ.ДАТА.

Фильтры (AND):

a) Н_ЛЮДИ.ФАМИЛИЯ = Иванов.

b) Н_ВЕДОМОСТИ.ДАТА < 2022-06-08.

Вид соединения: RIGHT JOIN.

Реализация

1. **SELECT** Н_ЛЮДИ.ИМЯ, Н_ВЕДОМОСТИ.ИД
FROM Н_ЛЮДИ
INNER JOIN Н_ВЕДОМОСТИ
ON Н_ЛЮДИ.ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД
WHERE Н_ЛЮДИ.ФАМИЛИЯ < 'Афанасьев'
AND Н_ВЕДОМОСТИ.ИД > 39921
AND Н_ВЕДОМОСТИ.ИД < 1250981;
2. **SELECT** Н_ЛЮДИ.ФАМИЛИЯ, Н_ВЕДОМОСТИ.ЧЛВК_ИД, Н_СЕССИЯ.ДАТА
FROM Н_ЛЮДИ
RIGHT JOIN Н_ВЕДОМОСТИ
ON Н_ЛЮДИ.ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД
RIGHT JOIN Н_СЕССИЯ
ON Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД
WHERE Н_ЛЮДИ.ФАМИЛИЯ = 'Иванов'
AND Н_ВЕДОМОСТИ.ДАТА < '2022-06-08'; -- all before 2012

Анализ

1. EXPLAIN ANALYZE SELECT 1 ;

	QUERY PLAN text	🔒
1	Gather (cost=1111.39..7614.95 rows=8324 width=17) (actual time=0.650..25.405 rows=7295 loops=1)	
2	Workers Planned: 2	
3	Workers Launched: 2	
4	-> Hash Join (cost=111.39..5782.55 rows=3468 width=17) (actual time=0.423..19.373 rows=2432 loops=3)	
5	Hash Cond: ("Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")	
6	-> Parallel Seq Scan on "Н_ВЕДОМОСТИ" (cost=0.00..5456.25 rows=81802 width=8) (actual time=0.016..11.892 rows=6543...	
7	Filter: (("ИД" > 39921) AND ("ИД" < 1250981))	
8	Rows Removed by Filter: 8709	
9	-> Hash (cost=108.68..108.68 rows=217 width=17) (actual time=0.338..0.339 rows=214 loops=3)	
10	Buckets: 1024 Batches: 1 Memory Usage: 19kB	
11	-> Bitmap Heap Scan on "Н_ЛЮДИ" (cost=5.96..108.68 rows=217 width=17) (actual time=0.166..0.292 rows=214 loops=...	
12	Recheck Cond: (("ФАМИЛИЯ")::text < 'Афанасьев')::text)	
13	Heap Blocks: exact=78	
14	-> Bitmap Index Scan on "ФАМ_ЛЮД" (cost=0.00..5.91 rows=217 width=0) (actual time=0.146..0.146 rows=214 loop...	
15	Index Cond: (("ФАМИЛИЯ")::text < 'Афанасьев')::text)	
16	Planning Time: 0.491 ms	
17	Execution Time: 25.809 ms	

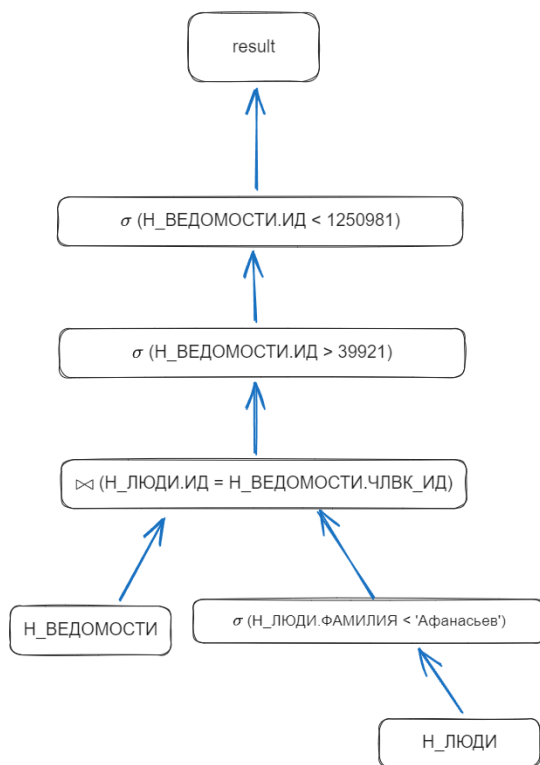
2. EXPLAIN ANALYZE SELECT 2 ;

	QUERY PLAN text	🔒
1	Nested Loop (cost=86.53..400.69 rows=1237 width=28) (actual time=0.969..0.971 rows=0 loops=1)	
2	Join Filter: ("Н_ЛЮДИ"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД")	
3	-> Hash Join (cost=86.24..204.62 rows=28 width=32) (actual time=0.463..0.954 rows=5 loops=1)	
4	Hash Cond: ("Н_СЕССИЯ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")	
5	-> Seq Scan on "Н_СЕССИЯ" (cost=0.00..108.52 rows=3752 width=12) (actual time=0.007..0.450 rows=3752 loops=1)	
6	-> Hash (cost=85.68..85.68 rows=45 width=20) (actual time=0.097..0.097 rows=45 loops=1)	
7	Buckets: 1024 Batches: 1 Memory Usage: 11kB	
8	-> Bitmap Heap Scan on "Н_ЛЮДИ" (cost=4.63..85.68 rows=45 width=20) (actual time=0.036..0.081 rows=45 loops=1)	
9	Recheck Cond: (("ФАМИЛИЯ")::text = 'Иванов')::text)	
10	Heap Blocks: exact=30	
11	-> Bitmap Index Scan on "ФАМ_ЛЮД" (cost=0.00..4.62 rows=45 width=0) (actual time=0.025..0.025 rows=45 loops=1)	
12	Index Cond: (("ФАМИЛИЯ")::text = 'Иванов')::text)	
13	-> Index Scan using "ВЕД_ЧЛВК_ФК_ИФК" on "Н_ВЕДОМОСТИ" (cost=0.29..6.15 rows=68 width=4) (actual time=0.003..0.003 rows=0 loop...	
14	Index Cond: ("ЧЛВК_ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД")	
15	Filter: ("ДАТА" < '2022-06-08 00:00:00')::timestamp without time zone)	
16	Planning Time: 0.653 ms	
17	Execution Time: 1.024 ms	

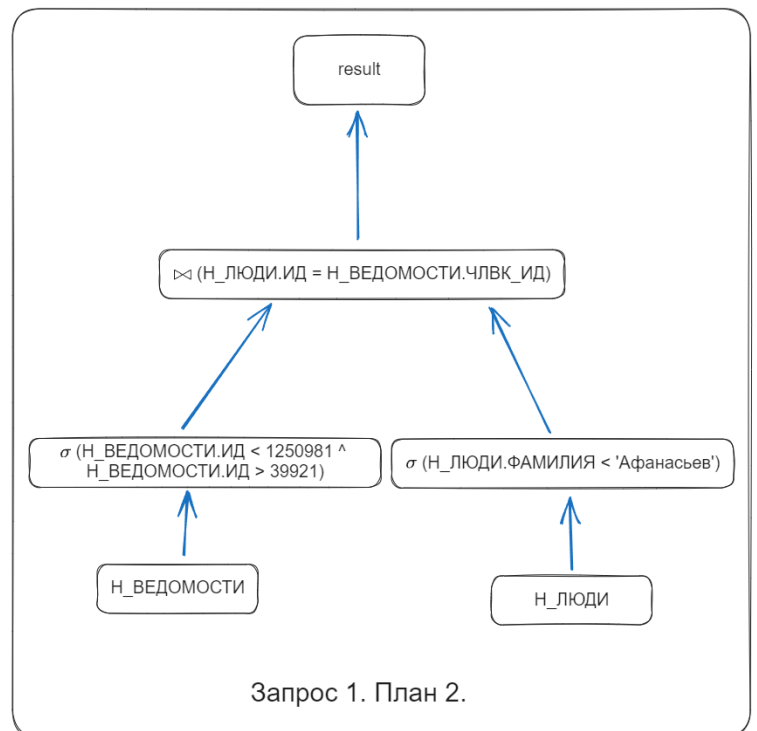
Возможные планы выполнения

Наиболее эффективным планом для запроса 1 является план 2, поскольку он наиболее сильно сокращает количество кортежей, участвующих в соединении. Исходя из предметной области, можно предположить, что условие на фамилию наиболее сильно сократит количество подходящих кортежей. Соединение же, скорее всего не уменьшит количество кортежей. Условие на индекс скорее всего незначительно уменьшит количество кортежей.

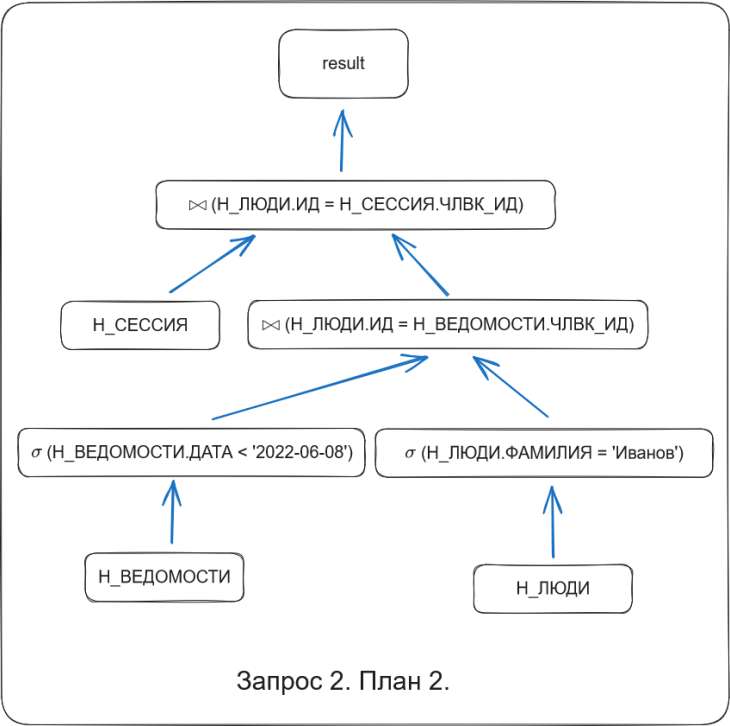
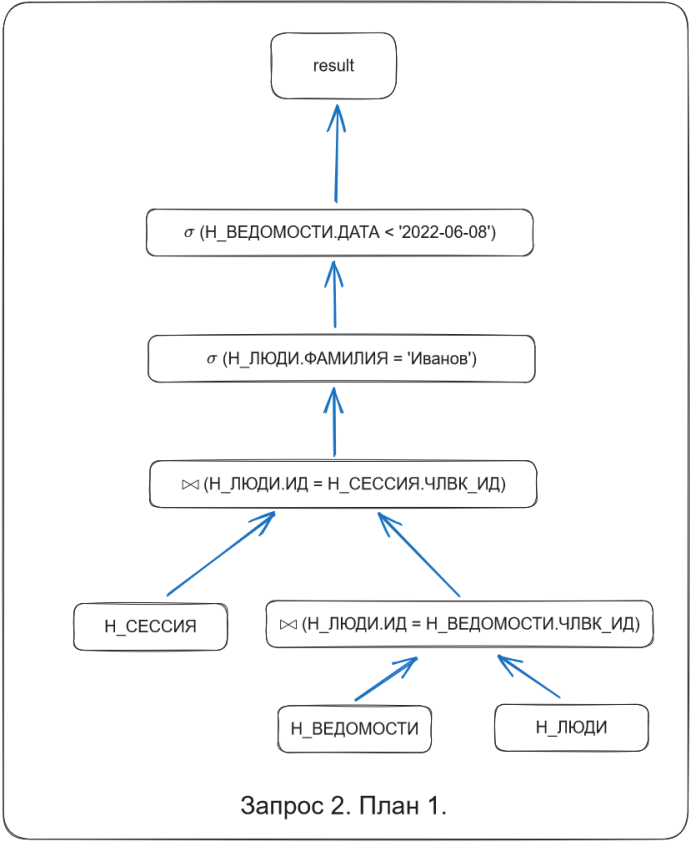
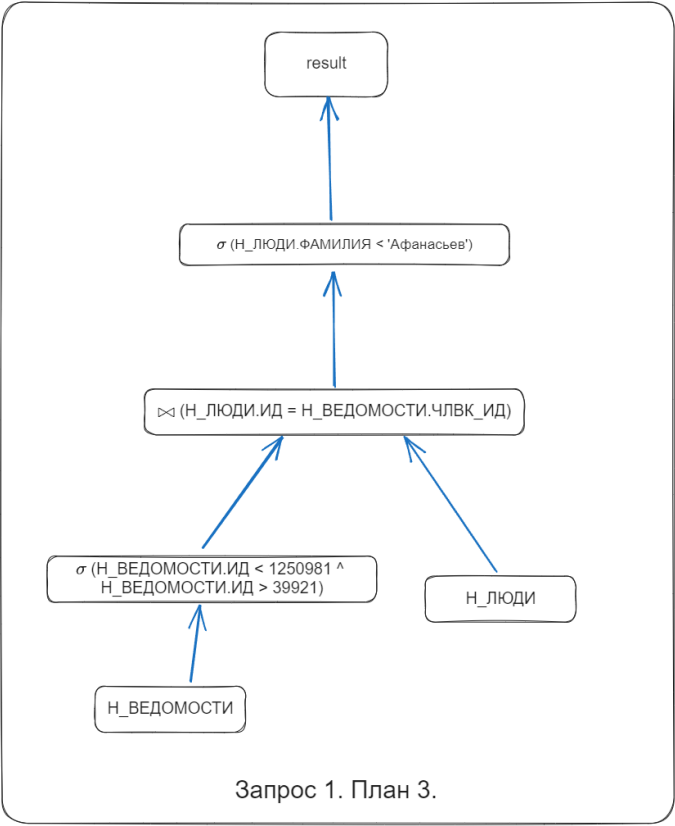
Для запроса 2 план 2 также является оптимальным, потому что сокращается количество кортежей при соединении. Правое соединение зависит от того, что с чем будем соединять, количество людей при правом соединении сильнее сокращает количество обрабатываемых кортежей, поэтому фильтруя его заранее, мы получаем лучший результат. Фильтрация по Н_ВЕДОМОСТИ.ДАТА ничего не дает, т.к. все записи датируются до 2022 года.



Запрос 1. План 1.



Запрос 1. План 2.



Предложения по улучшению

Список доступных индексов можно узнать с помощью запроса:

```
SELECT amname FROM pg_am;
```

1. В запросе используется фильтрация по целочисленному идентификатору ведомости, соединение по целочисленному идентификатору человека и фильтрация по строковому атрибуту фамилия. Поэтому можно предложить создать дополнительные индексы по используемым колонкам, при условии отсутствия оных. Также дополнительно, при достаточно большом количестве данных вызванном продолжительным сбором статистики имеет смысл применить индекс BRIN для колонки Н_ВЕДОМОСТИ.ЧЛВК_ИД, поскольку эти данные в целом имеют привязку к времени добавления запроса.

```
CREATE INDEX Н_ЛЮДИ_ИД_ИНД -- created by default for PK  
ON Н_ЛЮДИ (ИД);
```

```
CREATE INDEX Н_ЛЮДИ_ФАМИЛИЯ_ИНД -- varchar compare  
ON Н_ЛЮДИ (ФАМИЛИЯ);
```

```
CREATE INDEX Н_ВЕДОМОСТИ_ЧЛВК_ИД_ИНД -- integer equality  
ON Н_ВЕДОМОСТИ (ЧЛВК_ИД);
```

```
CREATE INDEX Н_ВЕДОМОСТИ_ЧЛВК_ИД_ИНД -- integer equality  
ON Н_ВЕДОМОСТИ USING BRIN(ЧЛВК_ИД);
```

```
CREATE INDEX Н_ВЕДОМОСТИ_ИД_ИНД -- integer compare  
ON Н_ВЕДОМОСТИ (ИД);
```

2. В запросе используются соединения по целочисленным идентификаторам и фильтрации по timestamp'у и строковому атрибуту. Поэтому можно предложить создать дополнительные индексы по используемым колонкам, при условии отсутствия оных. И аналогично можно добавить BRIN индекс для Н_ВЕДОМОСТИ, т.к. данные там хорошо коррелируют со временем добавления

```
CREATE INDEX Н_ЛЮДИ_ИД_ИНД -- created by default for PK
```

```
ON Н_ЛЮДИ (ИД);          -- сам разберется какой нужен

CREATE INDEX Н_ЛЮДИ_ФАМИЛИЯ_ИНД -- varchar equality
ON Н_ЛЮДИ
USING HASH (ФАМИЛИЯ);

CREATE INDEX Н_ВЕДОМОСТИ_ЧЛВК_ИД_ИНД -- integer equality
ON Н_ВЕДОМОСТИ
USING BRIN (ЧЛВК_ИД);

CREATE INDEX Н_ВЕДОМОСТИ_ДАТА_ИНД -- timestamp compare
ON Н_ВЕДОМОСТИ
USING BRING (ДАТА);

CREATE INDEX Н_СЕССИЯ_ЧЛВК_ИД_ИНД -- integer equality
ON Н_СЕССИЯ
USING BTREE (ЧЛВК_ИД);
```

Вывод

В результате лабораторной работы нами были изучены планы выполнения запросов в СУБД Postgresql, способы ускорения выполнения этих запросов с использованием индексов. Также изучили правила создания индексов, их преимущества и недостатки.