

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”**

Факультет Программной Инженерии и Компьютерной Техники
Дисциплина: «Базы данных»

ОТЧЁТ

по лабораторной работе №4
Вариант 776

Выполнил:

Студент группы Р3108

Грищенко Андрей Викторович

Проверил:

Афанасьев Дмитрий Борисович

Санкт-Петербург

2024 г.

1. Оглавление

| | |
|---|---|
| 1. Оглавление..... | 2 |
| 2. Текст задания..... | 2 |
| 3. Запрос №1..... | 3 |
| Возможные планы выполнения запросов без индексов..... | 4 |
| EXPLAIN ANALYZE..... | 5 |
| Индексы..... | 5 |
| EXPLAIN ANALYZE..... | 6 |
| 4. Запрос №2..... | 6 |
| Возможные планы выполнения запросов без индексов..... | 7 |
| EXPLAIN ANALYZE..... | 8 |
| Индексы..... | 8 |
| EXPLAIN ANALYZE..... | 9 |
| 5. Вывод..... | 9 |

2. 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:
Н_ТИПЫ_ВЕДОМОСТЕЙ, Н_ВЕДОМОСТИ.
Вывести атрибуты: Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н_ВЕДОМОСТИ.ИД.
Фильтры (AND):
а) Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.
б) Н_ВЕДОМОСТИ.ДАТА = 2010-06-18.
с) Н_ВЕДОМОСТИ.ДАТА > 1998-01-05.
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.ОТЧЕСТВО, Н_ВЕДОМОСТИ.ДАТА, Н_СЕССИЯ.ЧЛВК_ИД.

Фильтры (AND):

а) Н_ЛЮДИ.ОТЧЕСТВО = Сергеевич.

б) Н_ВЕДОМОСТИ.ЧЛВК_ИД > 153285.

Вид соединения: INNER JOIN.

3. 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) :

-- а) Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.

-- б) Н_ВЕДОМОСТИ.ДАТА = 2010-06-18.

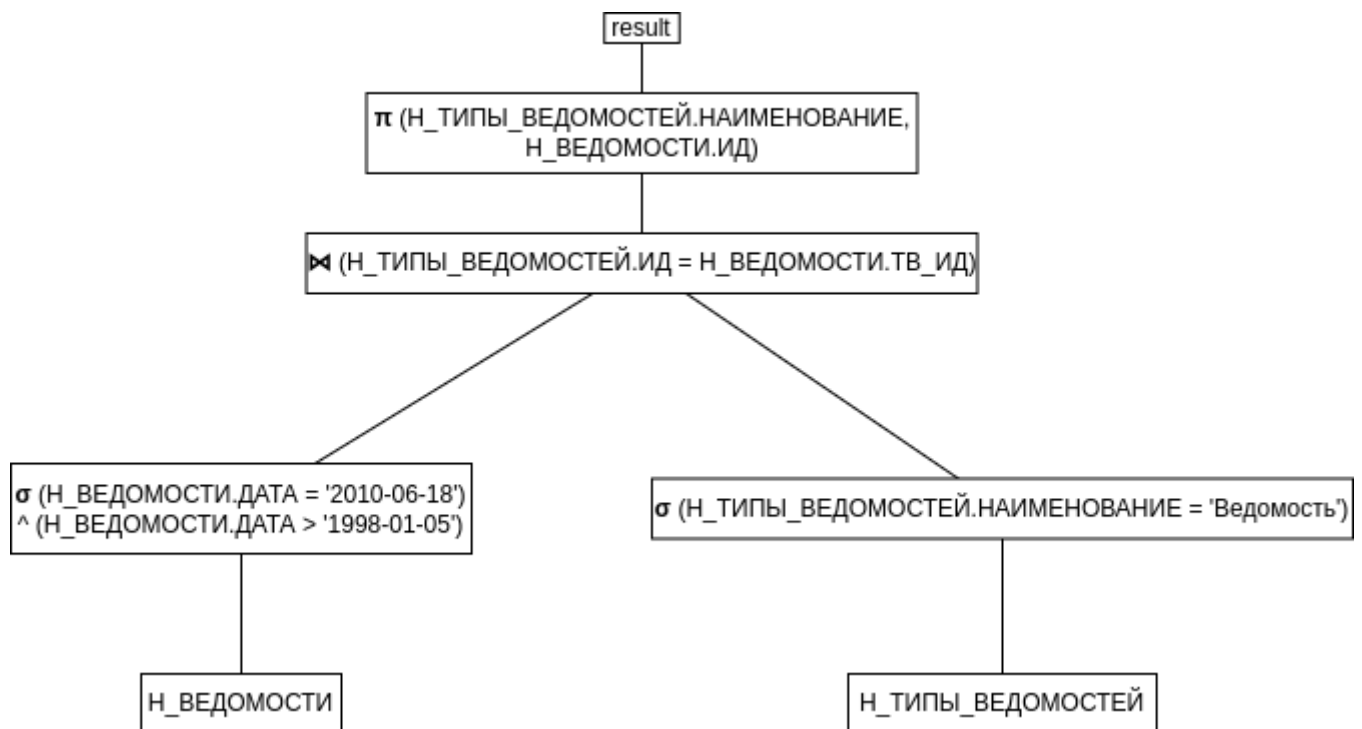
-- с) Н_ВЕДОМОСТИ.ДАТА > 1998-01-05.

-- Вид соединения: INNER JOIN.

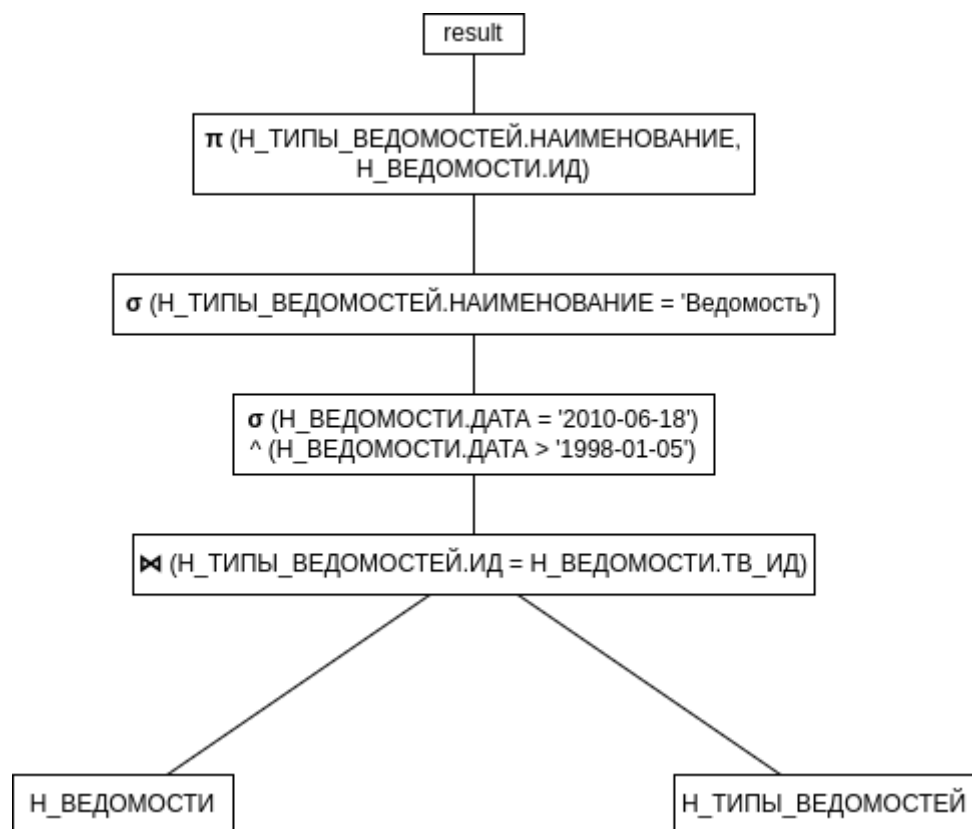
```
SELECT Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н_ВЕДОМОСТИ.ИД
FROM Н_ТИПЫ_ВЕДОМОСТЕЙ
INNER JOIN Н_ВЕДОМОСТИ ON Н_ВЕДОМОСТИ.ТВ_ИД = Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД
WHERE Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Ведомость'
AND Н_ВЕДОМОСТИ.ДАТА = '2010-06-18'
AND Н_ВЕДОМОСТИ.ДАТА > '1998-01-05';
```

Возможные планы выполнения запросов без индексов

План 1:



План 2:



Оптимальным является план 1, так как он производит объединение таблиц по уже отфильтрованным строкам.

EXPLAIN ANALYZE

Nested Loop (cost=20.98..29.94 rows=2 width=422) (actual time=11.752..11.906 rows=123 loops=1)

-> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=422) (actual time=0.020..0.026 rows=1 loops=1)

Filter: (("НАИМЕНОВАНИЕ")::text = 'Ведомость')::text)

Rows Removed by Filter: 2

-> Bitmap Heap Scan on "Н_ВЕДОМОСТИ" (cost=20.98..28.89 rows=2 width=8) (actual time=11.724..11.831 rows=123 loops=1)

Recheck Cond: (("ДАТА" > '1998-01-05 00:00:00')::timestamp without time zone) AND ("ДАТА" = '2010-06-18 00:00:00')::timestamp without time zone) AND ("Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД" = "ТВ_ИД"))

Heap Blocks: exact=22

-> BitmapAnd (cost=20.98..20.98 rows=2 width=0) (actual time=11.701..11.702 rows=0 loops=1)

-> Bitmap Index Scan on "БЕД_ДАТА_I" (cost=0.00..8.00 rows=371 width=0) (actual time=0.044..0.044 rows=141 loops=1)

Index Cond: (("ДАТА" > '1998-01-05 00:00:00')::timestamp without time zone) AND ("ДАТА" = '2010-06-18 00:00:00')::timestamp without time zone))

-> Bitmap Index Scan on "БЕД_ТВ_FK_I" (cost=0.00..12.63 rows=1112 width=0) (actual time=11.535..11.535 rows=190897 loops=1)

Index Cond: ("ТВ_ИД" = "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД")

Planning Time: 0.396 ms

Execution Time: 11.981 ms

Индексы

```
CREATE INDEX "ИНД_ТИПЫ_ВЕДОМОСТЕЙ_НАИМЕНОВАНИЕ" ON "Н_ТИПЫ_ВЕДОМОСТЕЙ"
USING BTREE ("НАИМЕНОВАНИЕ") ;
```

```
CREATE INDEX "ИНД_ВЕДОМОСТИ_ДАТА" ON "Н_ВЕДОМОСТИ" USING BTREE ("ДАТА") ;
```

```
CREATE INDEX "ИНД_ВЕДОМОСТИ_ТВ_ИД" ON "Н_ВЕДОМОСТИ" USING
BTREE ("ТВ_ИД") ;
```

При добавлении индексов планы выполнения запросов изменятся, так как вместо полного сканирования таблиц будет производиться индексный скан и Nested Loops Join станет быстрее

EXPLAIN ANALYZE

Nested Loop (cost=0.29..208.18 rows=23 width=422) (actual time=0.024..0.173 rows=123 loops=1)

Join Filter: ("Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД" = "Н_ВЕДОМОСТИ"."ТВ_ИД")

Rows Removed by Join Filter: 18

-> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=422) (actual time=0.010..0.012 rows=1 loops=1)

Filter: (("НАИМЕНОВАНИЕ")::text = 'Ведомость'::text)

Rows Removed by Filter: 2

-> Index Scan using "ВЕД_ДАТА_I" on "Н_ВЕДОМОСТИ" (cost=0.29..206.28 rows=69 width=8) (actual time=0.010..0.115 rows=141 loops=1)

Index Cond: (("ДАТА" > '1998-01-05 00:00:00'::timestamp without time zone) AND ("ДАТА" = '2010-06-18 00:00:00'::timestamp without time zone))

Planning Time: 1.471 ms

Execution Time: 0.233 ms

4. Запрос №2

-- 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.ОТЧЕСТВО, Н_ВЕДОМОСТИ.ДАТА, Н_СЕССИЯ.ЧЛВК_ИД.

-- Фильтры (AND) :

-- а) Н_ЛЮДИ.ОТЧЕСТВО = Сергеевич.

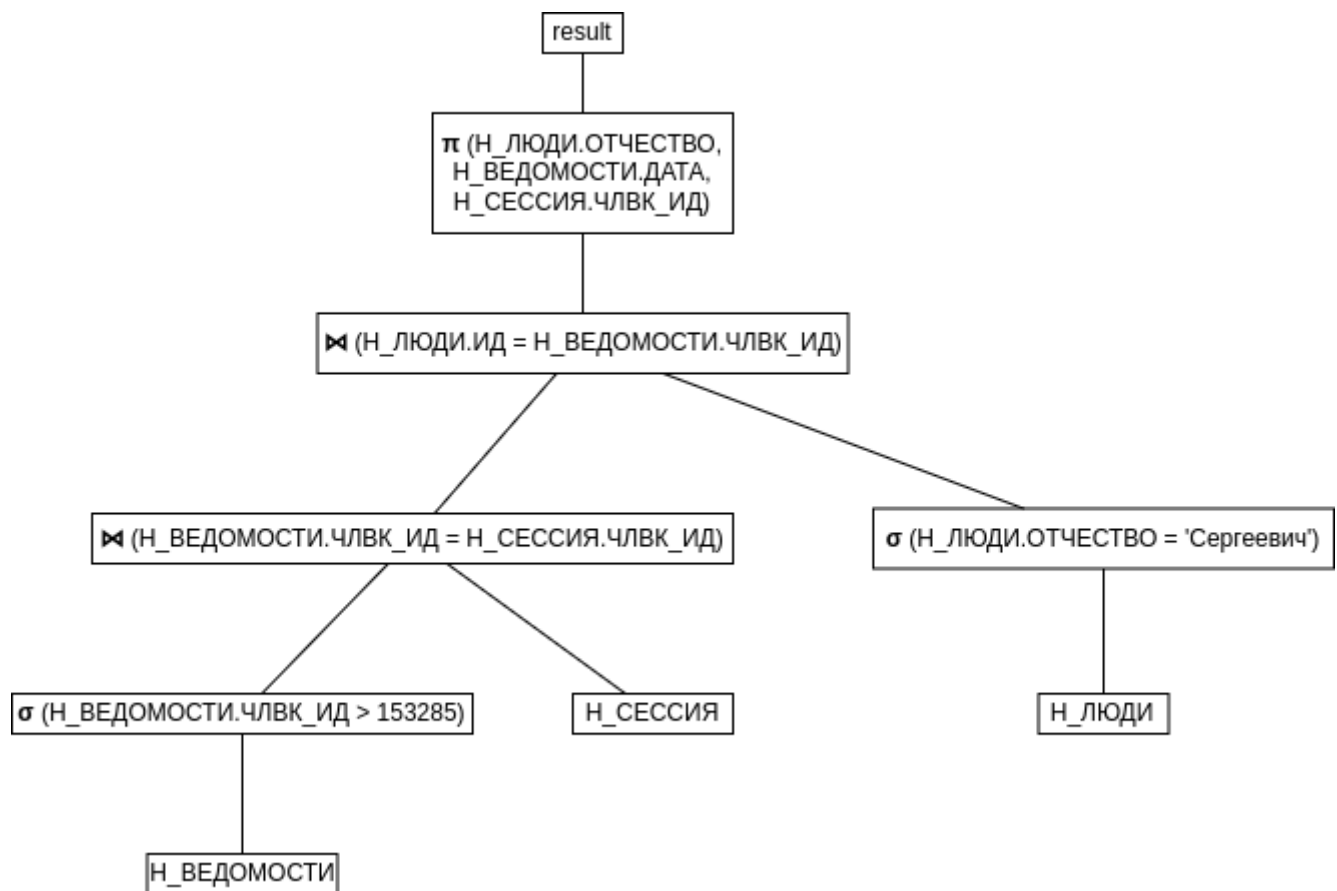
-- б) Н_ВЕДОМОСТИ.ЧЛВК_ИД > 153285.

-- Вид соединения: INNER JOIN.

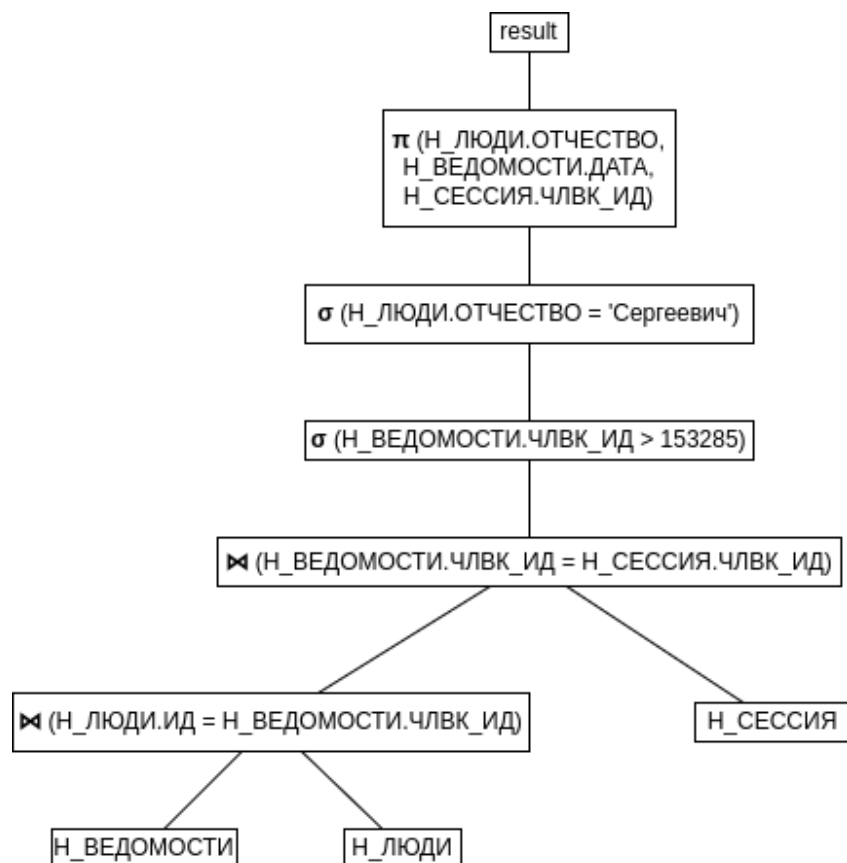
```
SELECT Н_ЛЮДИ.ОТЧЕСТВО, Н_ВЕДОМОСТИ.ДАТА, Н_СЕССИЯ.ЧЛВК_ИД
FROM Н_ЛЮДИ
INNER JOIN Н_ВЕДОМОСТИ ON Н_ВЕДОМОСТИ.ЧЛВК_ИД = Н_ЛЮДИ.ИД
INNER JOIN Н_СЕССИЯ ON Н_СЕССИЯ.ЧЛВК_ИД = Н_ЛЮДИ.ИД
WHERE Н_ЛЮДИ.ОТЧЕСТВО = 'Сергеевич'
AND Н_ВЕДОМОСТИ.ЧЛВК_ИД > 153285;
```

Возможные планы выполнения запросов без индексов

План 1:



План 2:



Оптимальным является план 1, так как он производит объединение таблиц по уже отфильтрованным строкам.

EXPLAIN ANALYZE

Будем выбирать Н_ЛЮДИ.ОТЧЕСТВО = 'Петрович' и Н_ВЕДОМОСТИ.ЧЛВК_ИД > 1, иначе нет подходящих записей

```
Nested Loop (cost=0.57..353.51 rows=1308 width=32) (actual time=0.351..1.562 rows=540 loops=1)
  Join Filter: ("Н_ЛЮДИ"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД")
    -> Nested Loop (cost=0.28..212.78 rows=20 width=28) (actual time=0.339..1.033 rows=22 loops=1)
      -> Seq Scan on "Н_ЛЮДИ" (cost=0.00..163.97 rows=32 width=24) (actual time=0.064..0.941 rows=32 loops=1)
        Filter: (("ОТЧЕСТВО")::text = 'Петрович'::text)
        Rows Removed by Filter: 5086
      -> Index Only Scan using "SYS_C003500_IFK" on "Н_СЕССИЯ" (cost=0.28..1.34 rows=18 width=4) (actual time=0.002..0.002 rows=1 loops=32)
        Index Cond: ("ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
        Heap Fetches: 0
    -> Index Scan using "ВЕД_ЧЛВК_FK_IFK" on "Н_ВЕДОМОСТИ" (cost=0.29..6.19 rows=68 width=12) (actual time=0.003..0.016 rows=25 loops=22)
      Index Cond: (("ЧЛВК_ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД") AND ("ЧЛВК_ИД" > 1))
Planning Time: 0.833 ms
Execution Time: 2.040 ms
```

Индексы

```
CREATE INDEX "инд_люди_отчество" ON "Н_люди" USING BTREE ("отчество");
CREATE INDEX "инд_люди_ид" ON "Н_люди" USING BTREE ("ид");
CREATE INDEX "инд_ведомости_члвк_ид" ON "Н_ведомости" USING BTREE ("члвк_ид");
```

При добавлении индексов планы выполнения запросов изменятся, так как вместо полного сканирования таблиц будет производиться индексный скан и Nested Loops Join станет быстрее

EXPLAIN ANALYZE

Будем выбирать Н_ЛЮДИ.ОТЧЕСТВО = 'Петрович' и Н_ВЕДОМОСТИ.ЧЛВК_ИД > 1, иначе нет подходящих записей

Nested Loop (cost=4.82..261.74 rows=1308 width=32) (actual time=0.107..0.852 rows=540 loops=1)

Join Filter: ("Н_ЛЮДИ"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД")

-> Nested Loop (cost=4.53..121.00 rows=20 width=28) (actual time=0.097..0.190 rows=22 loops=1)

-> Bitmap Heap Scan on "Н_ЛЮДИ" (cost=4.25..72.20 rows=32 width=24) (actual time=0.046..0.098 rows=32 loops=1)

Recheck Cond: (("ОТЧЕСТВО")::text = 'Петрович'::text)

Heap Blocks: exact=28

-> Bitmap Index Scan on "ИНД_ЛЮДИ_ОТЧЕСТВО" (cost=0.00..4.24 rows=32 width=0) (actual time=0.027..0.027 rows=32 loops=1)

Index Cond: (("ОТЧЕСТВО")::text = 'Петрович'::text)

-> Index Only Scan using "SYS_C003500_IFK" on "Н_СЕССИЯ" (cost=0.28..1.34 rows=18 width=4) (actual time=0.002..0.002 rows=1 loops=32)

Index Cond: ("ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")

Heap Fetches: 0

-> Index Scan using "ИНД_ВЕДОМОСТИ_ЧЛВК_ИД" on "Н_ВЕДОМОСТИ" (cost=0.29..6.19 rows=68 width=12) (actual time=0.003..0.020 rows=25 loops=22)

Index Cond: (("ЧЛВК_ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД") AND ("ЧЛВК_ИД" > 1))

Planning Time: 1.331 ms

Execution Time: 0.949 ms

5. Вывод

В результате проделанной лабораторной работы я изучил индексы и планы выполнения запросов, а также проанализировал выданные по варианту запросы.