

Университет ИТМО  
Факультет ФПИ и КТ

**Отчёт**  
**по лабораторной работе 4**  
**«Информационная система и база  
данных»**

Вариант 9273

Студент:

Ляо Ихун

Гр.Р33131

Преподаватель:

Байрамова Хумай Бахруз Кызы

## Текст задачи:

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.

Изменяться 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\_СЕССИЯ.  
Вывести атрибуты: Н\_ЛЮДИ.ИД, Н\_СЕССИЯ.ДАТА.  
Фильтры (AND):
  - а) Н\_ЛЮДИ.ОТЧЕСТВО = Георгиевич.
  - б) Н\_СЕССИЯ.ИД < 27640.
  - с) Н\_СЕССИЯ.ИД < 32199.Вид соединения: LEFT 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.ИД, Н\_СЕССИЯ.УЧГОД.  
Фильтры (AND):
  - а) Н\_ЛЮДИ.ФАМИЛИЯ = Иванов.
  - б) Н\_ВЕДОМОСТИ.ИД > 1250972.Вид соединения: LEFT JOIN.

## Выполнение

1.

(1)select Н\_ЛЮДИ.ИД,Н\_СЕССИЯ.ДАТА from  
Н\_ЛЮДИ LEFT JOIN Н\_СЕССИЯ on Н\_ЛЮДИ.ИД  
= Н\_СЕССИЯ.ЧЛВК\_ИД where Н\_СЕССИЯ.ИД <  
27640 AND Н\_ЛЮДИ.ОТЧЕСТВО = 'Георгиевич';

```
[s291007@helios ~/DB/lab/4]$ psql -h pg -d ucheb < 1.sql
```

ИД	ДАТА
105708	2002-01-09 00:00:00
104689	2009-01-13 00:00:00
104689	2008-01-10 00:00:00
100627	2011-06-15 00:00:00
100627	2011-01-17 00:00:00
100627	2011-01-17 00:00:00
104689	2010-01-09 00:00:00
100627	2011-01-15 00:00:00
100627	2011-01-17 00:00:00
100627	2010-01-13 00:00:00
100627	2010-01-09 00:00:00
100627	2010-01-09 00:00:00
100627	2010-06-14 00:00:00
100627	2010-06-15 00:00:00
100627	2010-06-18 00:00:00
100627	2010-06-18 00:00:00
100627	2010-01-15 00:00:00
100627	2010-06-18 00:00:00
100627	2011-06-13 00:00:00
100627	2011-06-13 00:00:00
100627	2011-06-14 00:00:00
100627	2011-06-14 00:00:00
100627	2011-06-15 00:00:00
100627	2011-06-15 00:00:00
100627	2011-01-15 00:00:00
100627	2011-01-16 00:00:00
100627	2011-01-16 00:00:00
100627	2010-06-15 00:00:00
100627	2010-01-13 00:00:00
100627	2010-01-15 00:00:00
100627	2010-06-14 00:00:00
100627	2010-01-15 00:00:00

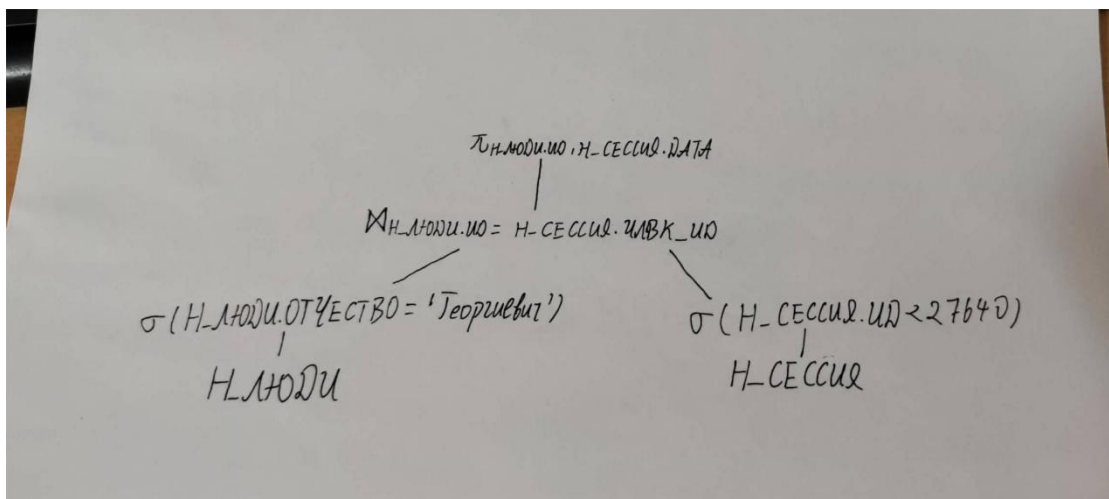
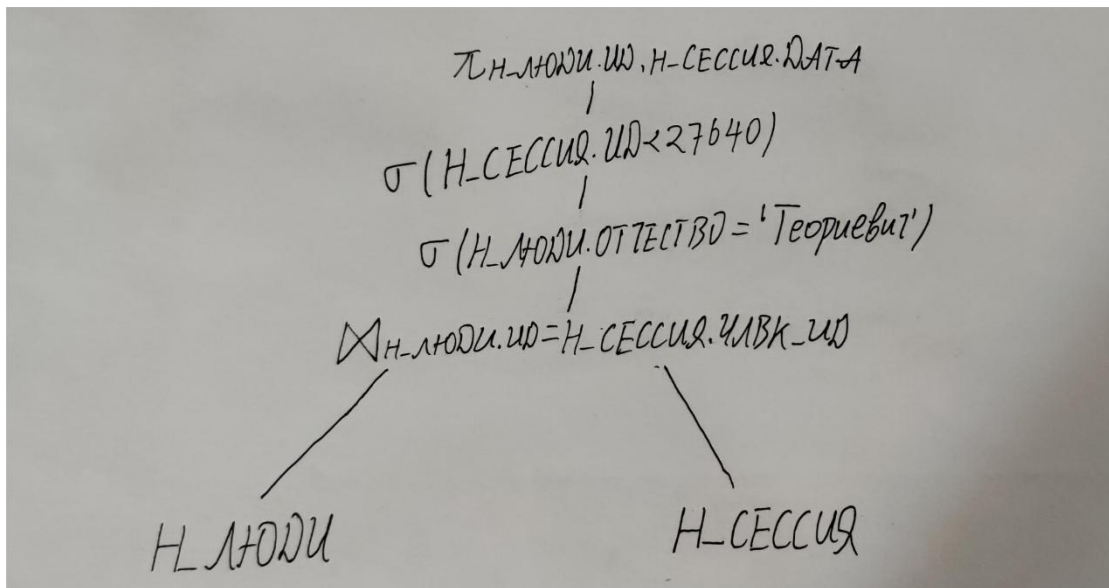
(32 строки)

(2) Explain analyze select

Н\_ЛЮДИ.ИД,Н\_СЕССИЯ.ДАТА from Н\_ЛЮДИ  
LEFT JOIN Н\_СЕССИЯ on Н\_ЛЮДИ.ИД =  
Н\_СЕССИЯ.ЧЛВК\_ИД where Н\_СЕССИЯ.ИД <  
27640 AND Н\_ЛЮДИ.ОТЧЕСТВО = 'Георгиевич';

```
1:34 19, 2019 SunMedano
[s291007@helios ~/DB/lab/4]$ psql -h pg -d ucheb < 1.sql
QUERY PLAN
-----
Nested Loop (cost=0.29..283.39 rows=15 width=12) (actual time=0.541..2.904 rows=32 loops=1)
-> Seq Scan on "Н_СЕССИЯ" (cost=0.00..117.90 rows=3461 width=12) (actual time=0.012..1.059 rows=3460 loops=1)
    Filter: ("ИД" < 27640)
    Rows Removed by Filter: 292
-> Memoize (cost=0.29..0.45 rows=1 width=4) (actual time=0.000..0.000 rows=0 loops=3460)
    Cache Key: "Н_СЕССИЯ"."ЧЛВК_ИД"
    Cache Mode: logical
    Hits: 3285 Misses: 175 Evictions: 0 Overflows: 0 Memory Usage: 12kB
    -> Index Scan using "ЧЛВК_PK" on "Н_ЛЮДИ" (cost=0.28..0.44 rows=1 width=4) (actual time=0.003..0.003 rows=0 loops=175)
        Index Cond: ("ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД")
        Filter: (("ОТЧЕСТВО")::text = 'Георгиевич'::text)
        Rows Removed by Filter: 1
Planning Time: 1.416 ms
Execution Time: 3.007 ms
(14 строк)
```

(3) План:



Второй вариант лучше. Тут делать выборку раньше, благодаря которого уменьшает количество проверок в join. И он тоже является вариантом системы.

4) Предложенные индексы:

В таблице  $H\_ЛЮДИ$  идет выборка столбцы  $ОТЧЕСТВО$ , поэтому можно создать hash index на  $ОТЧЕСТВО$ .

В таблице  $H\_СЕССИЯ$  идет сравнение столбцы  $ИД$ . Поэтому можно создать b-tree index на  $ИД$ .

2.

(1)select

```
Н_ЛЮДИ.ИД,Н_ВЕДОМОСТИ.ИД,Н_СЕССИЯ.УЧ
ГОД from Н_ЛЮДИ LEFT JOIN Н_ВЕДОМОСТИ
ON Н_ВЕДОМОСТИ.ЧЛВК_ИД = Н_ЛЮДИ.ИД
LEFT JOIN Н_СЕССИЯ on Н_ЛЮДИ.ИД =
Н_СЕССИЯ.ЧЛВК_ИД where
Н_ВЕДОМОСТИ.ИД > 1250927 AND
Н_ЛЮДИ.ФАМИЛИЯ = 'Иванов';
```

```
[s291007@helios ~/DB/lab/4]$ psql -h pg -d ucheb < 2.sql
  ИД   |   ИД   |  УЧГОД
-----+-----+-----
142011 | 1302027 |
142159 | 1355745 |
137808 | 1329032 |
142011 | 1343125 |
153328 | 1254114 |
142158 | 1344718 |
142158 | 1383138 |
142158 | 1383139 |
148829 | 1321718 |
153328 | 1362894 |
142158 | 1343140 |
142158 | 1343149 |
153328 | 1329154 |
153328 | 1254074 |
152923 | 1315855 |
153328 | 1310610 |
153328 | 1310630 |
137808 | 1426982 |
152923 | 1457444 |
153328 | 1340239 |
137808 | 1318376 |
152923 | 1308955 |
152923 | 1308978 |
152923 | 1309001 |
157095 | 1309752 |
157095 | 1381274 |
152923 | 1367668 |
142159 | 1316919 |
157095 | 1318643 |
```

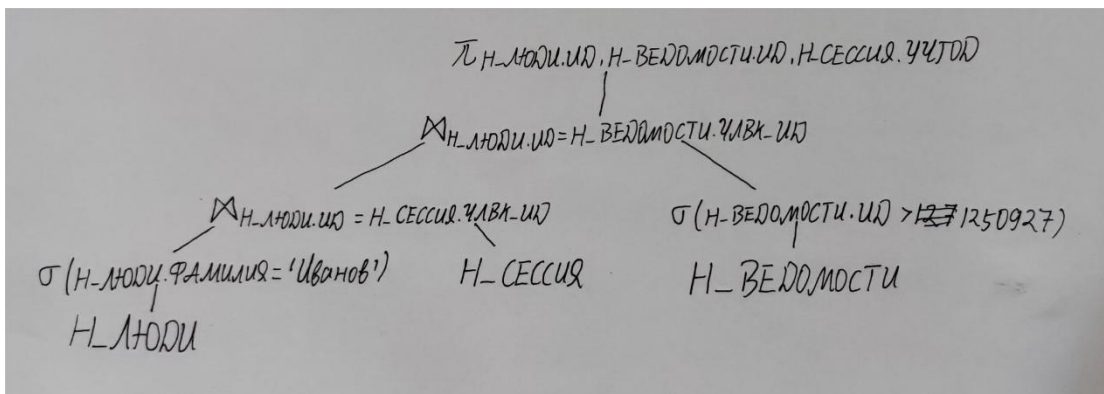
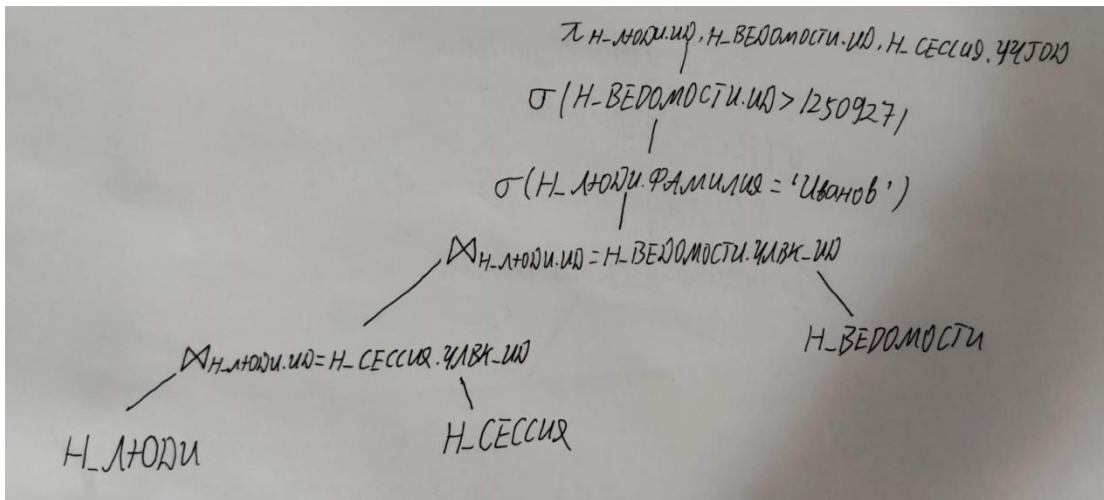
(2) Explain analyze select Н\_ЛЮДИ.ИД,  
Н\_ВЕДОМОСТИ.ИД,Н\_СЕССИЯ.УЧГОД from

Н\_ЛЮДИ LEFT JOIN Н\_ВЕДОМОСТИ ON  
 Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД LEFT  
 JOIN Н\_СЕССИЯ on Н\_ЛЮДИ.ИД =  
 Н\_СЕССИЯ.ЧЛВК\_ИД where  
 Н\_ВЕДОМОСТИ.ИД > 1250927 AND  
 Н\_ЛЮДИ.ФАМИЛИЯ = 'Иванов';

```

[s291007@helios ~/DB/lab/4]$ psql -h pg -d ucheb < 2.sql
QUERY PLAN
-----
Hash Join (cost=695.37..5186.73 rows=229 width=18) (actual time=3.591..12.317 rows=177 loops=1)
  Hash Cond: ("Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
    -> Bitmap Heap Scan on "Н_ВЕДОМОСТИ" (cost=490.19..4881.63 rows=26035 width=8) (actual time=1.788..7.863 rows=26139 loops=1)
        Recheck Cond: ("ИД" > 1250927)
        Heap Blocks: exact=642
        -> Bitmap Index Scan on "ВЕД_РК" (cost=0.00..483.68 rows=26035 width=0) (actual time=1.686..1.686 rows=26139 loops=1)
            Index Cond: ("ИД" > 1250927)
    -> Hash (cost=204.62..204.62 rows=45 width=14) (actual time=1.613..1.616 rows=48 loops=1)
        Buckets: 1024 Batches: 1 Memory Usage: 10kB
        -> Hash Right Join (cost=86.24..204.62 rows=45 width=14) (actual time=0.833..1.602 rows=48 loops=1)
            Hash Cond: ("Н_СЕССИЯ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
            -> Seq Scan on "Н_СЕССИЯ" (cost=0.00..108.52 rows=3752 width=14) (actual time=0.018..0.619 rows=3752 loops=1)
            -> Hash (cost=83.68..85.68 rows=45 width=4) (actual time=0.219..0.220 rows=45 loops=1)
                Buckets: 1024 Batches: 1 Memory Usage: 10kB
                -> Bitmap Heap Scan on "Н_ЛЮДИ" (cost=4.63..85.68 rows=45 width=4) (actual time=0.097..0.202 rows=45 loops=1)
                    Recheck Cond: (("ФАМИЛИЯ")::text = 'Иванов')::text)
                    Heap Blocks: exact=30
                    -> Bitmap Index Scan on "ФАМ_ЛЮД" (cost=0.00..4.62 rows=45 width=0) (actual time=0.086..0.086 rows=45 loops=1)
                        Index Cond: (("ФАМИЛИЯ")::text = 'Иванов')::text)
Planning Time: 2.003 ms
Execution Time: 12.462 ms
(21 строка)
  
```

(3) План:



Второй вариант лучше. Как первый запрос, тут тоже делать выборку раньше, благодаря которого уменьшает количество проверок в join. И он тоже является вариантом системы.

(4)Предложенные индексы:

В таблице Н\_ЛЮДИ идет выборка столбцы ФАМИЛИЯ, поэтому можно создать hash index на ФАМИЛИЯ.

В таблице Н\_ВЕДОМОСТИ идет сравнение столбцы ИД. Поэтому можно создать b-tree index на ИД.

## **Ответ на вопрос:**

При добавлении индекса структура плана не изменяется, но его реализации изменяется. Например, если добавим hash индекс в первую таблицу, тут структура план будет как текущая, но Nested Loop будет меняться Hash Join.

## **Вывод**

В ходе выполнения лабораторной работы я познакомился со средствами анализа производительности запросов, изучил различные типы индексов и условия их применения.