

Федеральное государственное автономное образовательное учреждение  
высшего образования «Национальный исследовательский университет  
ИТМО»

Лабораторная работа №4

Вариант 8732

Выполнил:

Хабиров Тимур Рустемович

Группа Р3132

Преподаватели:

Гиря Максим Дмитриевич

Санкт-Петербург 2025

## **Содержание**

<b>Текст задания.....</b>	<b>3</b>
<b>Описание предметной области.....</b>	<b>3</b>
<b>Список сущностей и их описание.....</b>	<b>3</b>
<b>Инфологическая модель.....</b>	<b>4</b>
<b>Даталогическая модель.....</b>	<b>4</b>
<b>Реализация даталогической модели на SQL.....</b>	<b>5</b>
<b>Вывод.....</b>	<b>7</b>

### **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).

Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

1. Таблицы: Н\_ЛЮДИ, Н\_СЕССИЯ.

Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_СЕССИЯ.ИД.

Фильтры (AND):

а) Н\_ЛЮДИ.ИД > 163484.

б) Н\_СЕССИЯ.УЧГОД > 2001/2002.

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.ИМЯ, Н\_ВЕДОМОСТИ.ИД,  
Н\_СЕССИЯ.УЧГОД.

Фильтры (AND):

а) Н\_ЛЮДИ.ИМЯ = Александр.

б) Н\_ВЕДОМОСТИ.ДАТА < 2022-06-08.

с) Н\_СЕССИЯ.ДАТА = 2012-01-25.

Вид соединения: LEFT JOIN.

## ***Реализация первого запроса на SQL***

/\*

Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

Таблицы: Н\_ЛЮДИ, Н\_СЕССИЯ.

Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_СЕССИЯ.ИД.

Фильтры (AND):

а) Н\_ЛЮДИ.ИД > 163484.

б) Н\_СЕССИЯ.УЧГОД > 2001/2002.

Вид соединения: LEFT JOIN.

\*/

```
SELECT  Н_ЛЮДИ.ФАМИЛИЯ,  
        Н_СЕССИЯ.ИД  
FROM Н_ЛЮДИ  
      LEFT JOIN Н_СЕССИЯ  
            ON Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД AND  
               Н_СЕССИЯ.УЧГОД > '2001/2002'  
WHERE Н_ЛЮДИ.ИД > 163484  
ORDER BY Н_ЛЮДИ.ФАМИЛИЯ;  
-- состояние БД pg_dump
```

ФАМИЛИЯ	ИД
---------	----

(0 строк)

```
SELECT ФАМИЛИЯ
FROM Н_ЛЮДИ
WHERE ИД > 163484; -- 0 строк
```

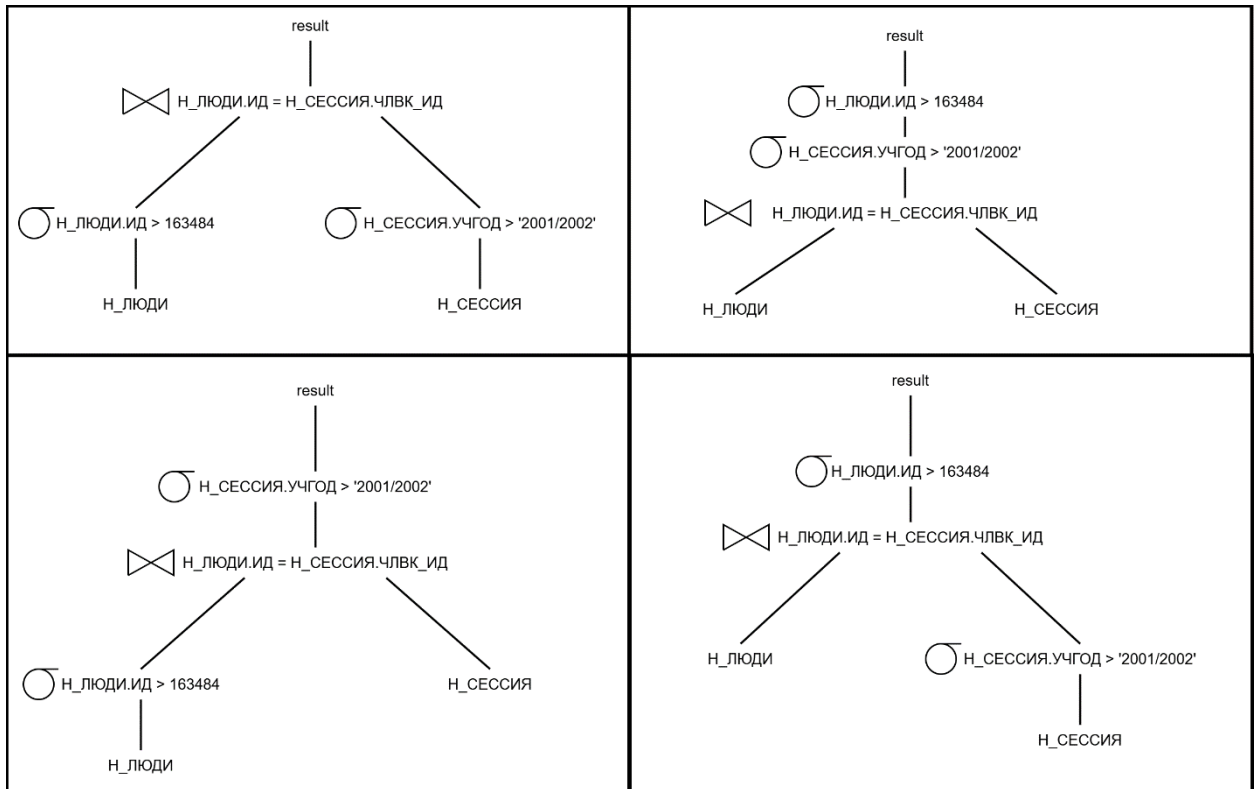
## Планы выполнения и индексы для первого запроса

Индексы:

```
CREATE INDEX year_session_index ON H_СЕССИЯ USING  
BTREE(УЧГОД);  
CREATE INDEX id_people_index ON H_ЛЮДИ USING HASH(ИД);
```

Выборка данных идет по оператору сравнения > (для полей H\_ЛЮДИ.ИД и H\_СЕССИЯ.УЧГОД), для которого оптимально BTREE. А соединение таблиц по оператору сравнения =, для которого оптимален HASH

## Планы выполнения:



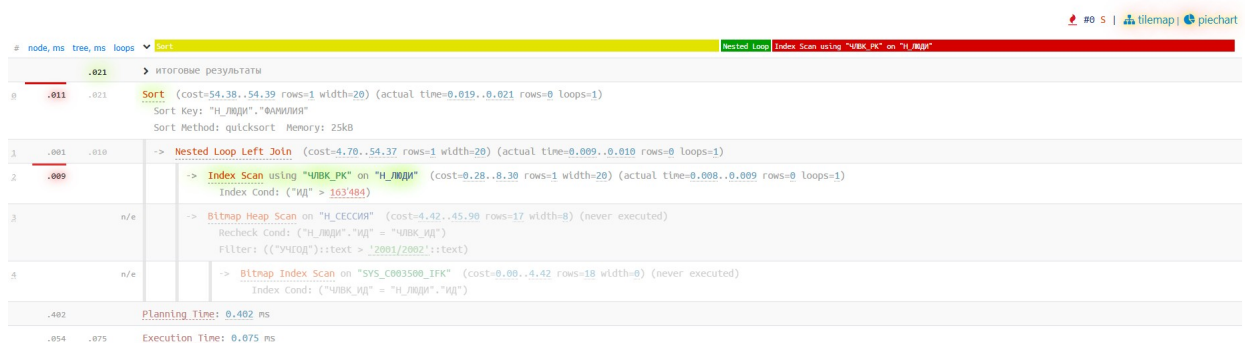
Наиболее эффективным является первый план, поскольку сразу, по заданным условиям, отсеивается большое количество строк



```
EXPLAIN ANALYZE
SELECT  Н_ЛЮДИ.ФАМИЛИЯ,
        Н_СЕССИЯ.ИД
FROM Н_ЛЮДИ
      LEFT JOIN Н_СЕССИЯ
            ON Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД AND
              Н_СЕССИЯ.УЧГОД > '2001/2002'
WHERE Н_ЛЮДИ.ИД > 163484
ORDER BY Н_ЛЮДИ.ФАМИЛИЯ;
```

## QUERY PLAN

Sort (cost=54.38..54.39 rows=1 width=20) (actual time=0.019..0.021 rows=0 loops=1)  
Sort Key: "Н\_ЛЮДИ"."ФАМИЛИЯ"  
Sort Method: quicksort Memory: 25kB  
-> Nested Loop Left Join (cost=4.70..54.37 rows=1 width=20) (actual time=0.009..0.010 rows=0 loops=1)  
-> Index Scan using "ЧЛВК\_ПК" on "Н\_ЛЮДИ" (cost=0.28..8.30 rows=1 width=20) (actual time=0.008..0.009 rows=0 loops=1)  
Index Cond: ("ИД" > 163484)  
-> Bitmap Heap Scan on "Н\_СЕССИЯ" (cost=4.42..45.90 rows=17 width=8) (never executed)  
Recheck Cond: ("Н\_ЛЮДИ"."ИД" = "ЧЛВК\_ИД")  
Filter: (("УЧГОД")::text > '2001/2002')::text)  
-> Bitmap Index Scan on "SYS\_C003500\_IFK" (cost=0.00..4.42 rows=18 width=0) (never executed)  
Index Cond: ("ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД")  
Planning Time: 0.402 ms  
Execution Time: 0.075 ms  
(13 строк)



При добавлении индекса на Н\_СЕССИЯ.УЧГОД добавиться еще один bitmap index scan по заданному условию

### **Реализация второго запроса на SQL**

/\*

Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.

Вывести атрибуты: Н\_ЛЮДИ.ИМЯ, Н\_ВЕДОМОСТИ.ИД, Н\_СЕССИЯ.УЧГОД.

Фильтры (AND):

а) Н\_ЛЮДИ.ИМЯ = Александр.

б) Н\_ВЕДОМОСТИ.ДАТА < 2022-06-08.

с) Н\_СЕССИЯ.ДАТА = 2012-01-25.

Вид соединения: LEFT JOIN.

\*/

```
SELECT  Н_ЛЮДИ.ИМЯ,  
        Н_ВЕДОМОСТИ.ИД,  
        Н_СЕССИЯ.УЧГОД  
FROM Н_ЛЮДИ  
      LEFT JOIN Н_ВЕДОМОСТИ  
            ON Н_ЛЮДИ.ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД AND  
              Н_ВЕДОМОСТИ.ДАТА < '2022-06-08'  
      LEFT JOIN Н_СЕССИЯ  
            ON Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД AND  
              Н_СЕССИЯ.ДАТА = '2012-01-25'  
WHERE Н_ЛЮДИ.ИМЯ = 'Александр';
```

ИМЯ	ИД	УЧГОД
Александр	1250978	
Александр	1210780	
Александр	1210782	
Александр	1210783	
Александр	1211019	
Александр	1216975	
Александр	1216609	
Александр	1216672	
Александр	1254262	
Александр	1216890	
Александр	1216894	
Александр	1255658	
Александр	1221476	
Александр	1255769	
...		
Александр	1038061	

(100 строк)

```
SELECT ДАТА
FROM Н_СЕССИЯ
WHERE ДАТА = '2012-01-25'; -- Одна строка
```

```
SELECT Н_ЛЮДИ.ИМЯ
FROM Н_ЛЮДИ
JOIN Н_СЕССИЯ
ON Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД AND
Н_СЕССИЯ.ДАТА = '2012-01-25'; -- Антон
```

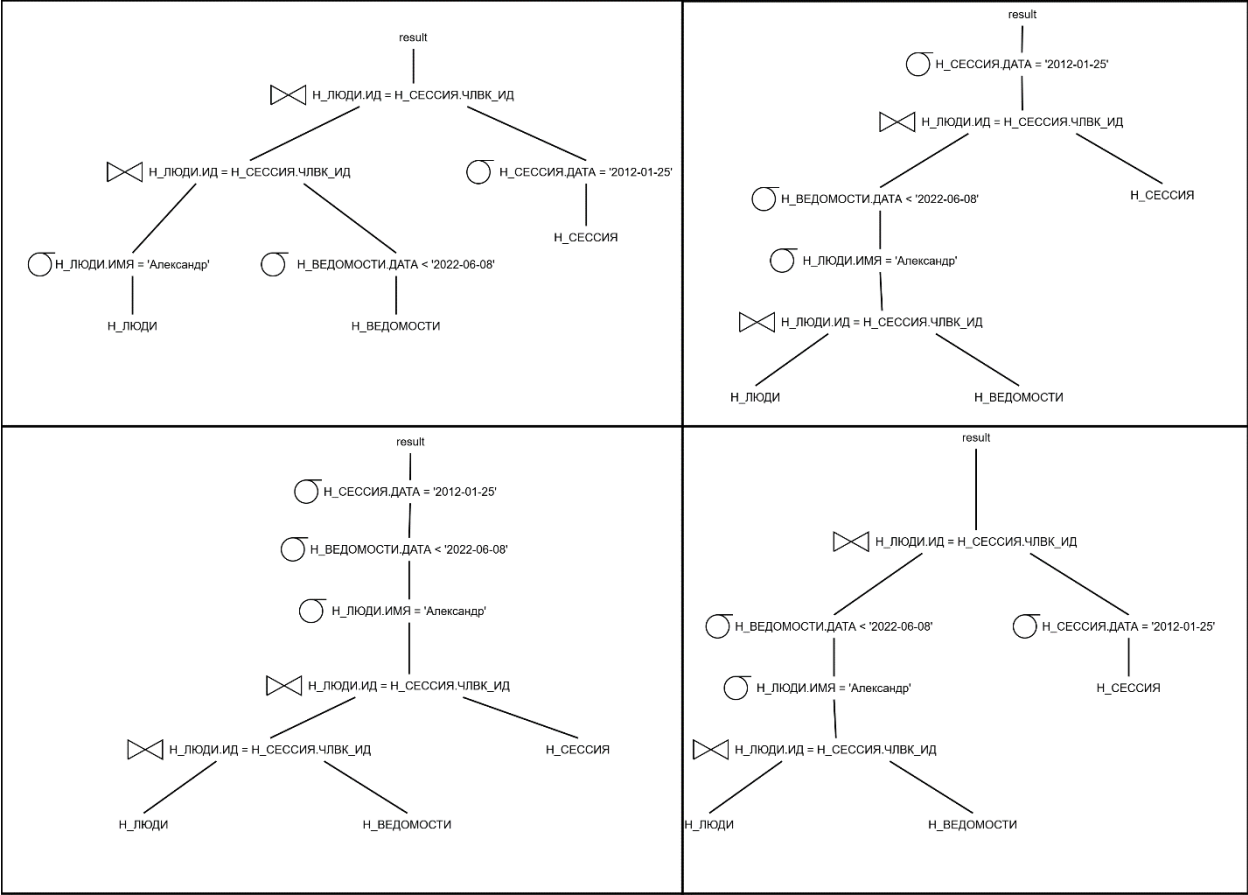
## **Планы выполнения и индексы для второго запроса**

Индексы:

```
CREATE INDEX name_people_index ON Н_ЛЮДИ USING HASH(ИМЯ);  
CREATE INDEX date_vedomosti_index ON Н_ВЕДОМОСТИ USING  
BTREE(ДАТА);  
CREATE INDEX id_people_index ON Н_ЛЮДИ USING HASH(ИД);
```

Соединение таблиц происходит с помощью оператора сравнения =, для которого оптимален HASH. Выборка данных происходит по нескольким атрибутам по операторам сравнения = и >, для второго оптимально BTREE

Планы выполнения:



Наиболее эффективным является первый план, поскольку он сразу отсеивает большое количество строк по заданным условиям

```
EXPLAIN ANALYZE
SELECT  Н_ЛЮДИ.ИМЯ,
        Н_ВЕДОМОСТИ.ИД,
        Н_СЕССИЯ.УЧГОД
FROM Н_ЛЮДИ
      LEFT JOIN Н_ВЕДОМОСТИ
            ON Н_ЛЮДИ.ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД AND
              Н_ВЕДОМОСТИ.ДАТА < '2022-06-08'
      LEFT JOIN Н_СЕССИЯ
            ON Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД AND
              Н_СЕССИЯ.ДАТА = '2012-01-25'
WHERE Н_ЛЮДИ.ИМЯ = 'Александр'
LIMIT 100;
```

## QUERY PLAN

Limit (cost=0.29..89.66 rows=100 width=27) (actual time=0.512..0.660 rows=100 loops=1)  
-> Nested Loop Left Join (cost=0.29..14832.11 rows=16597 width=27) (actual time=0.511..0.648 rows=100 loops=1)  
-> Nested Loop Left Join (cost=0.00..310.54 rows=382 width=27) (actual time=0.498..0.501 rows=2 loops=1)  
Join Filter: ("H\_ЛЮДИ"."ИД" = "H\_СЕССИЯ"."ЧЛВК\_ИД")  
Rows Removed by Join Filter: 2  
-> Seq Scan on "H\_ЛЮДИ" (cost=0.00..163.97 rows=382 width=17) (actual time=0.014..0.015 rows=2 loops=1)  
Filter: (("ИМЯ")::text = 'Александр'::text)  
Rows Removed by Filter: 25  
-> Materialize (cost=0.00..117.93 rows=5 width=14) (actual time=0.216..0.240 rows=1 loops=2)  
-> Seq Scan on "H\_СЕССИЯ" (cost=0.00..117.90 rows=5 width=14) (actual time=0.428..0.475 rows=1 loops=1)  
Filter: ("ДАТА" = '2012-01-25 00:00:00'::timestamp without time zone)  
Rows Removed by Filter: 3751  
-> Index Scan using "ВЕД\_ЧЛВК\_FK\_IFK" on "H\_ВЕДОМОСТИ" (cost=0.29..37.33 rows=68 width=8) (actual time=0.007..0.066 rows=50 loops=2)  
Index Cond: ("ЧЛВК\_ИД" = "H\_ЛЮДИ"."ИД")  
Filter: ("ДАТА" < '2022-06-08 00:00:00'::timestamp without time zone)  
Planning Time: 0.559 ms  
Execution Time: 0.714 ms  
(17 строк)

The screenshot shows a query plan visualization tool. The top bar includes a search icon, a refresh icon, and a pie chart icon. The main area is divided into two panes. The left pane shows a tree diagram of the query plan with nodes representing different operations and their costs, rows, and widths. The right pane shows the detailed text of the query plan, including the same operations and their parameters. The text view is color-coded to match the tree diagram. The bottom of the screen shows the planning and execution times.

#	node	ms	tree	ms	rows	RRBF	Q	loops	cost	rows	width	actual time	actual rows	actual loops
0	Limit	0.512	0.660	100	3.778			1	(cost=0.29..89.66 rows=100 width=27)	100	27	0.512..0.660	100	1
1	Nested Loop Left Join	0.511	0.648	100				1	(cost=0.29..14832.11 rows=16597 width=27)	16597	27	0.511..0.648	100	1
2	Nested Loop Left Join	0.498	0.501	2	50.05			1	(cost=0.00..310.54 rows=382 width=27)	382	27	0.498..0.501	2	1
3	Seq Scan	0.014	0.015	2	92.6%			1	(cost=0.00..163.97 rows=382 width=17)	382	17	0.014..0.015	2	1
4	Materialize	0.216	0.240	1				2	(cost=0.00..117.93 rows=5 width=14)	5	14	0.216..0.240	1	2
5	Seq Scan	0.428	0.475	1	100.0%			1	(cost=0.00..117.90 rows=5 width=14)	5	14	0.428..0.475	1	1
6	Index Scan	0.007	0.066	50				2	(cost=0.29..37.33 rows=68 width=8)	68	8	0.007..0.066	50	2

Planning Time: 0.559 ms  
Execution Time: 0.714 ms

При добавлении индексов seq scans заменятся на index scans, filters заменятся на index scans



**Вывод**

В ходе выполнения лабораторной работы я научился строить планы выполнения запросов, освоил работу с реляционной алгеброй, изучил различные индексы, узнал, как их использовать для оптимизации запросов