

# Национальный исследовательский университет ИТМО

---

Факультет Программной Инженерии и Компьютерной техники

## **Лабораторная работа №4** **Вариант №81642751**

Выполнил:

Чусовлянов Максим Сергеевич

Группа Р3107

Преподаватели:

Байрамова Хумай

Николаев Владимир Вячеславович

Санкт-Петербург  
2024

## Оглавление

Задание:.....	3
Запрос №1:.....	4
Индексы:.....	4
Результат EXPLAIN ANALYSE:.....	5
Запрос №2:.....	6
Индексы:.....	6
Результат EXPLAIN ANALYSE:.....	7
Вывод:.....	8

## 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. Изменяются ли планы при добавлении индекса и как?

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

Таблицы: Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.

Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД, Н\_ВЕДОМОСТИ.ИД.

Фильтры (AND):

а) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Экзаменационный лист.

б) Н\_ВЕДОМОСТИ.ДАТА = 2022-06-08.

с) Н\_ВЕДОМОСТИ.ДАТА < 2022-06-08.

Вид соединения: LEFT JOIN.

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

Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.

Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО.

Фильтры: (AND)

а) Н\_ЛЮДИ.ОТЧЕСТВО = Сергеевич.

б) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 163276.

Вид соединения: RIGHT JOIN.

### Запрос №1:

Таблицы: Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.

Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД, Н\_ВЕДОМОСТИ.ИД.

Фильтры (AND):

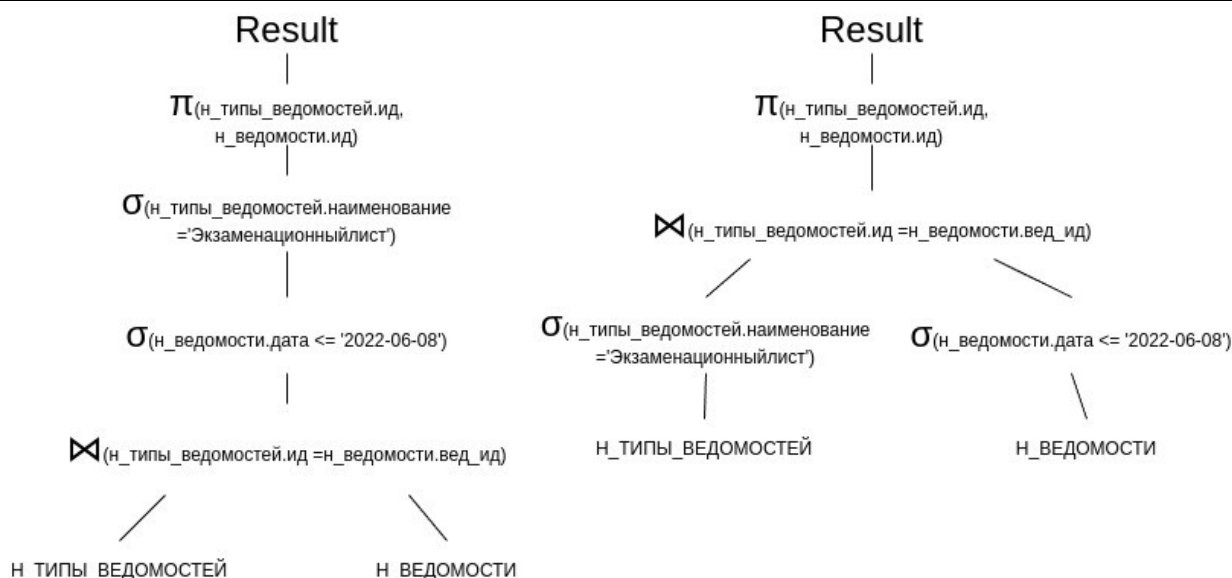
а) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Экзаменационный лист.

б) Н\_ВЕДОМОСТИ.ДАТА = 2022-06-08.

с) Н\_ВЕДОМОСТИ.ДАТА < 2022-06-08.

Вид соединения: LEFT JOIN.

```
SELECT Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД, Н_ВЕДОМОСТИ.ИД
FROM Н_ТИПЫ_ВЕДОМОСТЕЙ
LEFT JOIN Н_ВЕДОМОСТИ ON Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД = Н_ВЕДОМОСТИ.ВЕД_ИД
WHERE Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Экзаменационный лист'
AND Н_ВЕДОМОСТИ.ДАТА <= '2022-06-08';
```



Оптимальным является план №2, так как он производит объединение таблиц по ранее выбранным атрибутам, а не по таблицам целиком.

### Индексы:

```
CREATE INDEX "ИНДЕКС_ТИПЫ_ВЕДОМОСТЕЙ_ИД" ON "Н_ТИПЫ_ВЕДОМОСТЕЙ" USING
hash("ИД");
CREATE INDEX "ИНДЕКС_ВЕДОМОСТИ_ИД" ON "Н_ВЕДОМОСТИ" USING hash("ИД");
CREATE INDEX "ИНДЕКС_ТИПЫ_ВЕДОМОСТЕЙ_НАИМ" ON "Н_ТИПЫ_ВЕДОМОСТЕЙ"
USING hash("НАИМЕНОВАНИЕ");
CREATE INDEX "ИНДЕКС_ВЕДОМОСТИ_ДАТА" ON "Н_ВЕДОМОСТИ" USING btree("ДАТА");
```

Добавление этих индексов должно ускорить выполнение запросов, так как по перечисленным полям происходит выборка с использованием оператора сравнения. Так же быстрее будет происходить соединение таблиц. В последнем случае используются операторы сравнения „>“ и „<“, так что эффективнее использовать btree. В первых трех случаях используется прямое сравнение, так что эффективнее использовать хэш-индекс.

При добавлении индексов планы выполнения запросов изменятся, так как будет происходить индексный скан и Hash Join станет быстрее благодаря индексам.

### Результат EXPLAIN ANALYSE:

```
Hash Join (cost=1.05..5468.88 rows=1 width=8) (actual time=22.249..22.251
rows=0 loops=3)
    Hash Cond: ("Н_ВЕДОМОСТИ"."ВЕД_ИД" = "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД")
    -> Parallel Seq Scan on "Н_ВЕДОМОСТИ" (cost=0.00..5224.54 rows=92683
width=8) (actual time=0.016..14.163 rows=74147 loops=3)
        Filter: ("ДАТА" <= '2022-06-08 00:00:00'::timestamp without time zone)
    -> Hash (cost=1.04..1.04 rows=1 width=4) (actual time=0.028..0.029
rows=1 loops=3)
        Buckets: 1024 Batches: 1 Memory Usage: 9kB
    -> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1
width=4) (actual time=0.016..0.017 rows=1 loops=3)
        Filter: (("НАИМЕНОВАНИЕ")::text = 'Экзаменационный лист'::text)
    Rows Removed by Filter: 2
Planning Time: 0.311 ms
Execution Time: 28.339 ms
```

## Запрос №2:

Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.

Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД,  
Н\_УЧЕНИКИ.НАЧАЛО.

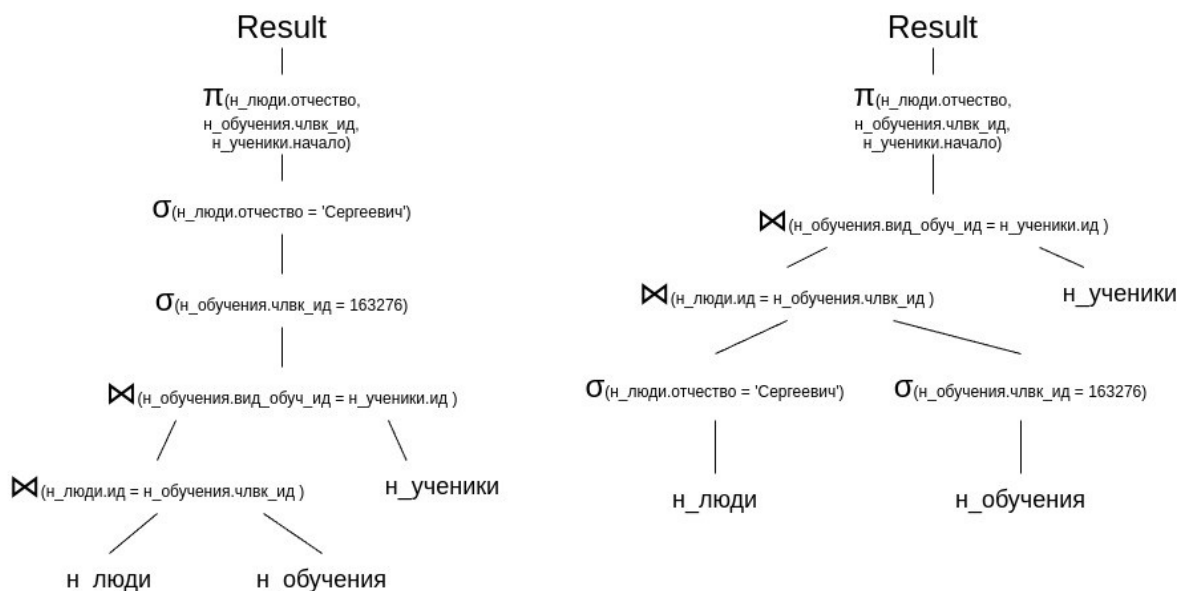
Фильтры: (AND)

а) Н\_ЛЮДИ.ОТЧЕСТВО = Сергеевич.

б) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 163276.

Вид соединения: RIGHT JOIN.

```
SELECT "Н_ЛЮДИ"."ОТЧЕСТВО", "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД", "Н_УЧЕНИКИ"."НАЧАЛО"  
FROM "Н_ЛЮДИ"  
RIGHT JOIN "Н_ОБУЧЕНИЯ" ON "Н_ЛЮДИ"."ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД"  
RIGHT JOIN "Н_УЧЕНИКИ" ON "Н_ОБУЧЕНИЯ"."ВИД_ОБУЧ_ИД" = "Н_УЧЕНИКИ"."ИД"  
WHERE "Н_ЛЮДИ"."ОТЧЕСТВО" = 'Сергеевич' AND "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = 163276;
```



Оптимальным является план №2, так как он производит объединение таблиц по ранее выбранным атрибутам, а не по таблицам целиком.

## Индексы:

```
CREATE INDEX "ИНДЕКС_ЛЮДИ_ОТЧЕСТВО" ON "Н_ЛЮДИ" USING hash("ОТЧЕСТВО");  
CREATE INDEX "ИНДЕКС_ОБУЧЕНИЯ_ЧЛВК_ИД" ON "Н_ОБУЧЕНИЯ" USING  
hash("ЧЛВК_ИД");  
CREATE INDEX "ИНДЕКС_УЧЕНИКИ_ИД" ON "Н_УЧЕНИКИ" USING hash("ИД");  
CREATE INDEX "ИНДЕКС_ОБУЧЕНИЯ_ВИД_ОБУЧЕНИЯ_ИД" ON "Н_ОБУЧЕНИЯ" USING  
hash("ВИД_ОБУЧ_ИД");  
CREATE INDEX "ИНДЕКС_ЛЮДИ_ИД" ON "Н_ЛЮДИ" USING hash("ИД");
```

Добавление этих индексов должно ускорить выполнение запросов, так как по перечисленным полям происходит выборка с использованием оператора сравнения. Так же быстрее будет происходить соединение таблиц. Во всех случаях происходит прямое сравнение, так что эффективнее использовать хэш-индексы.

При добавлении индексов планы выполнения запросов изменятся, так как будет происходить индексный скан и Nested Loop Join станет быстрее благодаря индексам.

## Результат EXPLAIN ANALYSE:

```
Nested Loop (cost=8.88..17.16 rows=1 width=32) (actual time=0.054..0.055 rows=0 loops=1)
  -> Merge Join (cost=8.60..8.85 rows=1 width=12) (actual time=0.042..0.044 rows=1 loops=1)
    Merge Cond: ("Н_УЧЕНИКИ"."ИД" = "Н_ОБУЧЕНИЯ"."ВИД_ОБУЧ_ИД")
    -> Index Scan using "УЧЕН_PK" on "Н_УЧЕНИКИ" (cost=0.29..2664.40 rows=23311 width=12) (actual time=0.007..0.008 rows=2 loops=1)
    -> Sort (cost=8.31..8.32 rows=1 width=8) (actual time=0.029..0.030 rows=1 loops=1)
      Sort Key: "Н_ОБУЧЕНИЯ"."ВИД_ОБУЧ_ИД"
      Sort Method: quicksort Memory: 25kB
      -> Index Scan using "ОБУЧ_ЧЛВК_FK_I" on "Н_ОБУЧЕНИЯ" (cost=0.28..8.30 rows=1 width=8) (actual time=0.013..0.014 rows=1 loops=1)
        Index Cond: ("ЧЛВК_ИД" = 163276)
      -> Index Scan using "ЧЛВК_PK" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1 width=24) (actual time=0.008..0.008 rows=0 loops=1)
        Index Cond: ("ИД" = 163276)
        Filter: (("ОТЧЕСТВО")::text = 'Сергеевич'::text)
        Rows Removed by Filter: 1
Planning Time: 0.412 ms
Execution Time: 0.102 ms
```

**Вывод:** Во время выполнения данной лабораторной работы я научился оптимизировать запросы, составлять наиболее выгодный план выполнения запросов, используя для этого подходящие виды индексов.