Федеральное государственное автономное образовательное учреждение высшего образования Национальный исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

Базы данных

Лабораторная работа №4

Вариант № 726

Выполнил: студент группы P3108, Васильев Никита Алексеевич

Преподаватель: Афанасьев Дмитрий Борисович

Санкт-Петербург 2024

Содержание

[Текст задания 3](#_Toc167973507)

[Запросы на SQL 3](#_Toc167973508)

[Индексы для первого запроса 4](#_Toc167973509)

[Планы выполнения первого запроса 4](#_Toc167973510)

[Тест первого запроса 4](#_Toc167973511)

[Индексы для второго запроса 6](#_Toc167973512)

[Тест второго запроса 7](#_Toc167973513)

[Выводы по работе 8](#_Toc167973514)

# 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.  
Изменятся 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).

# Запросы на SQL

*-- 1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
-- Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.  
-- Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД, Н\_ВЕДОМОСТИ.ДАТА.  
-- Фильтры (AND):  
-- a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Перезачет.  
-- b) Н\_ВЕДОМОСТИ.ИД < 1426978.  
-- c) Н\_ВЕДОМОСТИ.ИД < 1250981.  
-- Вид соединения: RIGHT JOIN.*SELECT "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД", "Н\_ВЕДОМОСТИ"."ДАТА"  
FROM "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"  
RIGHT JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "Н\_ВЕДОМОСТИ"."ТВ\_ИД"  
WHERE "НАИМЕНОВАНИЕ" = 'Перезачет' AND  
 "Н\_ВЕДОМОСТИ"."ИД" < 1426978 AND  
 "Н\_ВЕДОМОСТИ"."ИД" < 1250981;  
  
*-- 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.ЧЛВК\_ИД, Н\_СЕССИЯ.ЧЛВК\_ИД.  
-- Фильтры (AND):  
-- a) Н\_ЛЮДИ.ИД < 163484.  
-- b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 163249.  
-- Вид соединения: LEFT JOIN.*SELECT "Н\_ЛЮДИ"."ОТЧЕСТВО"  
FROM "Н\_ЛЮДИ"  
LEFT JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"  
LEFT JOIN "Н\_СЕССИЯ" ON "Н\_ЛЮДИ"."ИД" = "Н\_СЕССИЯ"."ЧЛВК\_ИД"  
WHERE "Н\_ЛЮДИ"."ИД" < 163484 AND  
 "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" < 163249;

# Индексы для первого запроса

Для ускорения запроса можно было бы создать индекс на атрибуте "НАИМЕНОВАНИЕ" (Hash), но так как в таблице всего три элемента, это толбко удлинит фильтрацию строк с условием "Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ" = "Перезачет"

Также на атрибуте "ИД" (B-tree). Это ускорит фильтрацию строк с условием "Н\_ВЕДОМОСТИ"."ИД" < 1250981

CREATE INDEX IF NOT EXISTS ведомость\_ид ON "Н\_ВЕДОМОСТИ" USING btree("ИД");  
CREATE INDEX IF NOT EXISTS ведомость\_члвк\_ид ON "Н\_ВЕДОМОСТИ" USING btree("ЧЛВК\_ИД");

CREATE INDEX IF NOT EXISTS ведомость\_твид ON "Н\_ВЕДОМОСТИ" USING btree("ТВ\_ИД");

# Планы выполнения первого запроса

План №1:

1. Полный скан таблицы Н\_ТИПЫ\_ВЕДОМОСТЕЙ.
2. Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра Н\_ВЕДОМОСТИ.ИД < 1250981.
3. Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ\_ИД.
4. Фильтрация результата по условию Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Перезачет.

План №2:

1. Полный скан таблицы Н\_ТИПЫ\_ВЕДОМОСТЕЙ с применением фильтра Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Перезачет.
2. Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра Н\_ВЕДОМОСТИ.ИД < 1250981.
3. Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ\_ИД.

План №2 оптимальнее, так как фильтрация происходит до соединения таблиц, из-за чего необходимо обработать меньше строк.

При добавлении индексов вместо полного скана будет использоваться индексный скан, а Nested Loops Join станет быстрее благодаря индексу на атрибуте ТВ\_ИД.

# Тест первого запроса

Тест без индексов:

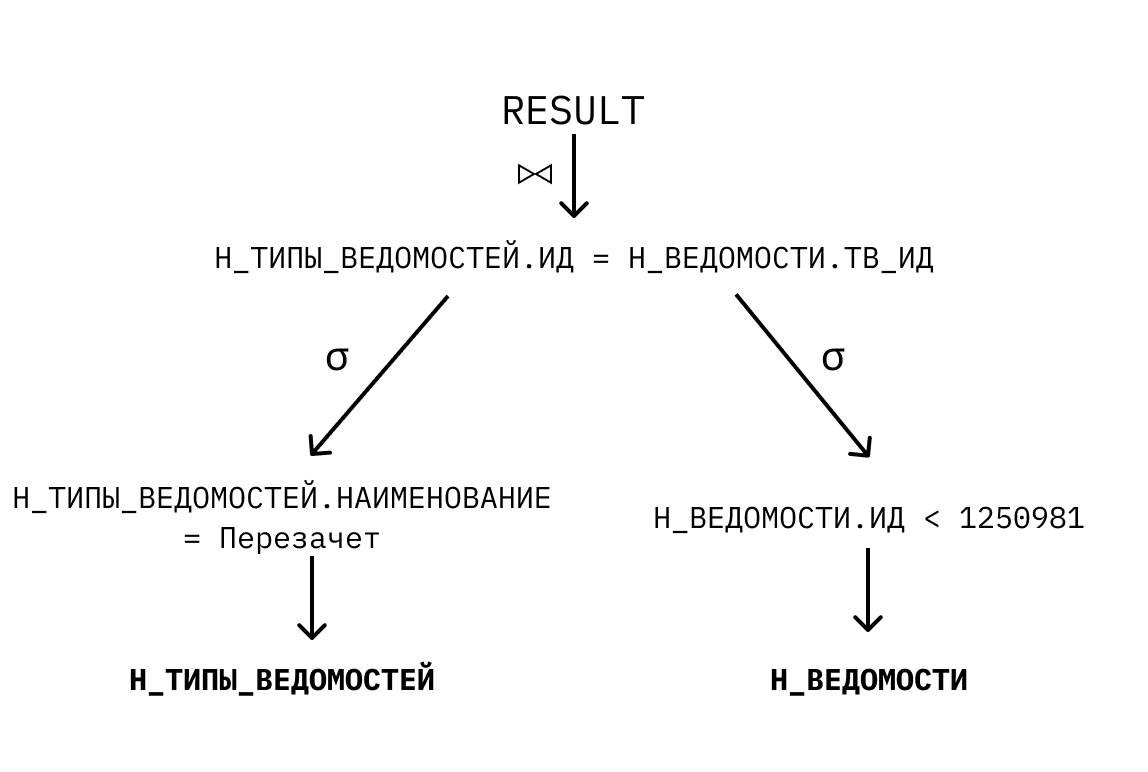
Gather (cost=1011.51..6946.01 rows=1640 width=12) (actual time=10.400..31.965 rows=6807 loops=1)  
 Workers Planned: 2  
 Workers Launched: 2  
 -> Hash Join (cost=11.51..5782.01 rows=683 width=12) (actual time=6.616..25.004 rows=2269 loops=3)  
" Hash Cond: (""Н\_ВЕДОМОСТИ"".""ТВ\_ИД"" = ""Н\_ТИПЫ\_ВЕДОМОСТЕЙ"".""ИД"")"  
" -> Parallel Seq Scan on ""Н\_ВЕДОМОСТИ"" (cost=0.00..5456.25 rows=81979 width=12) (actual time=0.019..15.941 rows=65438 loops=3)"  
" Filter: ((""ИД"" < 1426978) AND (""ИД"" < 1250981))"  
 Rows Removed by Filter: 8709  
 -> Hash (cost=11.50..11.50 rows=1 width=4) (actual time=0.035..0.036 rows=1 loops=3)  
 Buckets: 1024 Batches: 1 Memory Usage: 9kB  
" -> Seq Scan on ""Н\_ТИПЫ\_ВЕДОМОСТЕЙ"" (cost=0.00..11.50 rows=1 width=4) (actual time=0.026..0.026 rows=1 loops=3)"  
" Filter: ((""НАИМЕНОВАНИЕ"")::text = 'Перезачет'::text)"  
 Rows Removed by Filter: 2  
Planning Time: 0.214 ms  
Execution Time: 32.325 ms

Тест с индексами:

Nested Loop (cost=828.79..6859.70 rows=1640 width=12) (actual time=0.990..4.724 rows=6807 loops=1)  
" -> Seq Scan on ""Н\_ТИПЫ\_ВЕДОМОСТЕЙ"" (cost=0.00..11.50 rows=1 width=4) (actual time=0.010..0.011 rows=1 loops=1)"  
" Filter: ((""НАИМЕНОВАНИЕ"")::text = 'Перезачет'::text)"  
 Rows Removed by Filter: 2  
" -> Bitmap Heap Scan on ""Н\_ВЕДОМОСТИ"" (cost=828.79..6192.37 rows=65583 width=12) (actual time=0.976..3.643 rows=6807 loops=1)"  
" Recheck Cond: (""ТВ\_ИД"" = ""Н\_ТИПЫ\_ВЕДОМОСТЕЙ"".""ИД"")"  
" Filter: ((""ИД"" < 1426978) AND (""ИД"" < 1250981))"  
 Rows Removed by Filter: 3268  
 Heap Blocks: exact=758  
" -> Bitmap Index Scan on ""ведомость\_твид"" (cost=0.00..812.40 rows=74147 width=0) (actual time=0.526..0.526 rows=10075 loops=1)"  
" Index Cond: (""ТВ\_ИД"" = ""Н\_ТИПЫ\_ВЕДОМОСТЕЙ"".""ИД"")"  
Planning Time: 0.305 ms  
Execution Time: 5.084 ms

Сравнение:

|  |  |  |
| --- | --- | --- |
|  | **Без индексов** | **С индексами** |
| Время выполнения всего запроса | 32.325 ms | 5.084 ms |
| Время, затраченное для получения всех строк | 31.965 ms | 4.724 ms |
| Стоимость | 1011.51 .. 6946.01 | 828.79 .. 6859.70 |
| Количество обработанных строк | 65438 | 10075 |



# Индексы для второго запроса

Для ускорения запроса можно создать индекс на атрибуте " ЧЛВК\_ИД" (B-tree) Это ускорит фильтрацию строк с условием "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" < 163249 и соединение таблиц по атрибуту ЧЛВК\_ИД.

Также на атрибуте "ИД" (B-tree). Это ускорит фильтрацию строк с условием "Н\_ЛЮДИ"."ИД" < 163484.

Также на атрибуте " ЧЛВК\_ИД " (B-tree). Это ускорит присоединение строк JOIN "Н\_СЕССИЯ" ON "Н\_ЛЮДИ"."ИД" = "Н\_СЕССИЯ"."ЧЛВК\_ИД".

CREATE INDEX IF NOT EXISTS сессия\_члвк\_ид ON "Н\_СЕССИЯ" USING btree("ЧЛВК\_ИД");  
CREATE INDEX IF NOT EXISTS люди\_ид ON "Н\_ЛЮДИ" USING btree("ИД");

План №1:

1. Полный скан таблицы Н\_ЛЮДИ с применением фильтра Н\_ЛЮДИ.ИД < 163484.
2. Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 163249.
3. Соединение таблиц с использованием Nested Loops Join по атрибуту ЧЛВК\_ИД.
4. Соединение таблиц с таблицей Н\_СЕССИЯ с использованием Nested Loops Join по атрибуту ЧЛВК\_ИД.

План №2:

1. Полный скан таблицы Н\_ЛЮДИ с применением фильтра Н\_ЛЮДИ.ИД < 163484.
2. Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 163249.
3. Соединение таблиц с использованием Hash Join по атрибуту ЧЛВК\_ИД.
4. Соединение таблиц с таблицей Н\_СЕССИЯ с использованием Hash Join по атрибуту ЧЛВК\_ИД.

План №1 оптимальнее, так как использование Nested Loops Join предпочтительнее при обработке небольшого количества данных.

При добавлении индексов вместо полного скана будет использоваться индексный скан, а Nested Loops Join станет быстрее благодаря индексу на атрибуте ЧЛВК\_ИД.

# Тест второго запроса

Тест без индексов:

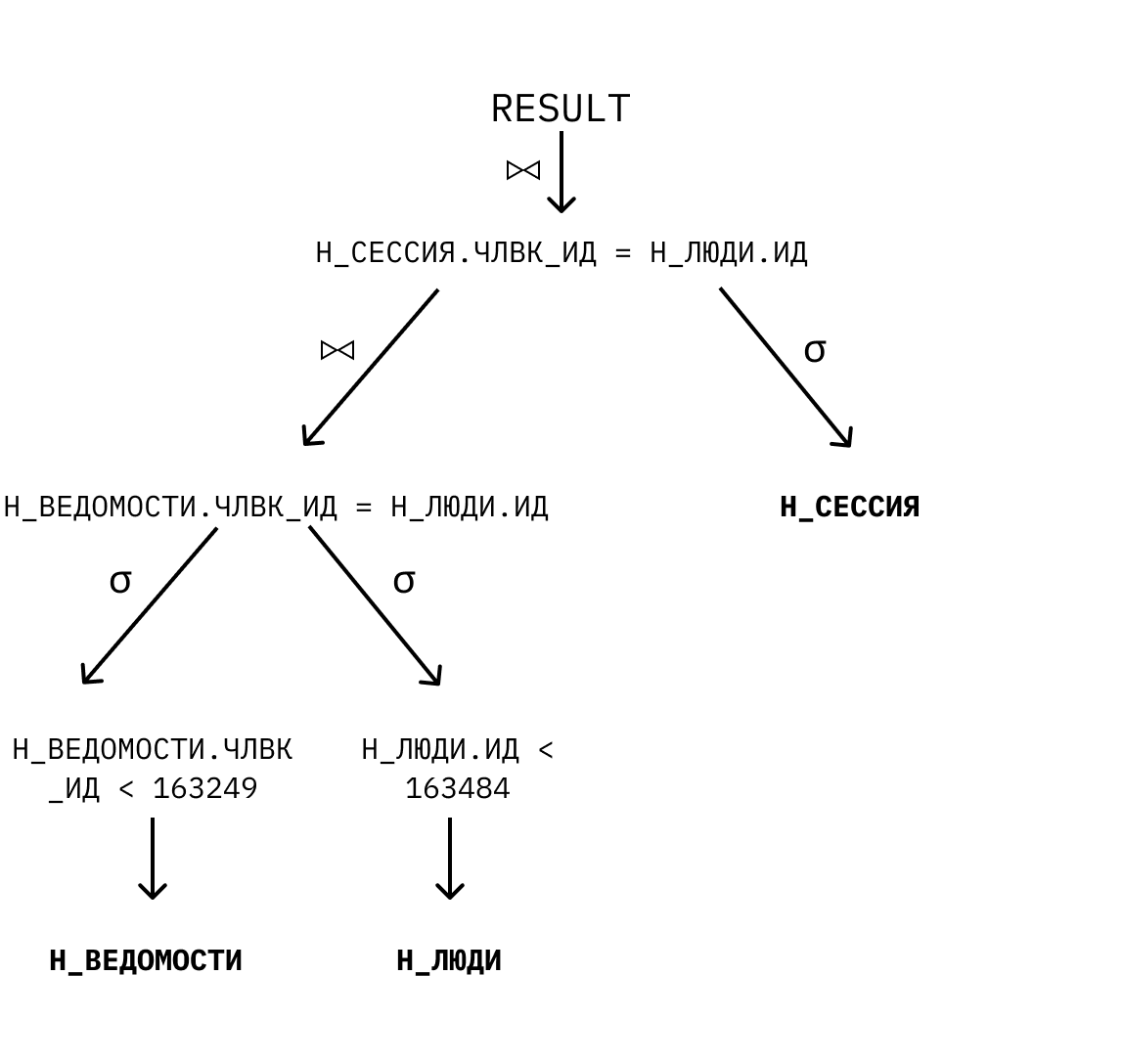
Hash Join (cost=800.28..10704.01 rows=222332 width=20) (actual time=8.779..102.524 rows=228287 loops=1)  
" Hash Cond: (""Н\_ВЕДОМОСТИ"".""ЧЛВК\_ИД"" = ""Н\_ЛЮДИ"".""ИД"")"  
" -> Seq Scan on ""Н\_ВЕДОМОСТИ"" (cost=0.00..6846.50 rows=222375 width=4) (actual time=0.015..42.156 rows=222421 loops=1)"  
" Filter: (""ЧЛВК\_ИД"" < 163249)"  
 Rows Removed by Filter: 19  
 -> Hash (cost=736.32..736.32 rows=5117 width=24) (actual time=8.713..8.718 rows=8176 loops=1)  
 Buckets: 8192 Batches: 1 Memory Usage: 529kB  
 -> Hash Right Join (cost=617.94..736.32 rows=5117 width=24) (actual time=2.707..4.888 rows=8176 loops=1)  
" Hash Cond: (""Н\_СЕССИЯ"".""ЧЛВК\_ИД"" = ""Н\_ЛЮДИ"".""ИД"")"  
" -> Seq Scan on ""Н\_СЕССИЯ"" (cost=0.00..108.52 rows=3752 width=4) (actual time=0.007..0.689 rows=3752 loops=1)"  
 -> Hash (cost=553.98..553.98 rows=5117 width=24) (actual time=2.653..2.655 rows=5117 loops=1)  
 Buckets: 8192 Batches: 1 Memory Usage: 357kB  
" -> Seq Scan on ""Н\_ЛЮДИ"" (cost=0.00..553.98 rows=5117 width=24) (actual time=0.026..1.676 rows=5117 loops=1)"  
" Filter: (""ИД"" < 163484)"  
 Rows Removed by Filter: 1  
Planning Time: 0.973 ms  
Execution Time: 113.276 ms

Тест с индексами:

Hash Join (cost=419.21..8147.99 rows=222332 width=20) (actual time=5.465..76.950 rows=228287 loops=1)  
" Hash Cond: (""Н\_ВЕДОМОСТИ"".""ЧЛВК\_ИД"" = ""Н\_ЛЮДИ"".""ИД"")"  
" -> Index Only Scan using ""ведомость\_члвк\_ид"" on ""Н\_ВЕДОМОСТИ"" (cost=0.29..4671.86 rows=222375 width=4) (actual time=0.008..24.363 rows=222421 loops=1)"  
" Index Cond: (""ЧЛВК\_ИД"" < 163249)"  
 Heap Fetches: 0  
 -> Hash (cost=354.95..354.95 rows=5117 width=24) (actual time=5.439..5.442 rows=8176 loops=1)  
 Buckets: 8192 Batches: 1 Memory Usage: 529kB  
 -> Hash Right Join (cost=228.22..354.95 rows=5117 width=24) (actual time=2.330..4.113 rows=8176 loops=1)  
" Hash Cond: (""Н\_СЕССИЯ"".""ЧЛВК\_ИД"" = ""Н\_ЛЮДИ"".""ИД"")"  
" -> Index Only Scan using ""сессия\_члвк\_ид"" on ""Н\_СЕССИЯ"" (cost=0.28..80.56 rows=3752 width=4) (actual time=0.009..0.409 rows=3752 loops=1)"  
 Heap Fetches: 0  
 -> Hash (cost=163.97..163.97 rows=5117 width=24) (actual time=2.307..2.309 rows=5117 loops=1)  
 Buckets: 8192 Batches: 1 Memory Usage: 357kB  
" -> Seq Scan on ""Н\_ЛЮДИ"" (cost=0.00..163.97 rows=5117 width=24) (actual time=0.011..1.383 rows=5117 loops=1)"  
" Filter: (""ИД"" < 163484)"  
 Rows Removed by Filter: 1  
Planning Time: 0.601 ms  
Execution Time: 87.749 ms

Сравнение:

|  |  |  |
| --- | --- | --- |
|  | **Без индексов** | **С индексами** |
| Время выполнения всего запроса | 113.276 ms | 87.749 ms |
| Время, затраченное для получения всех строк | 102.524 ms | 76.950 ms |
| Стоимость | 800.28..10704.01 | 419.21..8147.99 |
| Количество обработанных строк | 5117 | 5117 |



# Выводы по работе

В ходе выполнения данной лабораторной работы я познакомился с использованием индексов в Postgres, узнал, как с их помощью можно ускорить время запросов.