Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет

ИТМО»

*Факультет программной инженерии и компьютерной техники*

**Лабораторная работа №4**

По дисциплине:

*“Базы данных”*

*Вариант:*

*1003*

Выполнил:

Щукин Егор Вячеславович

Группа: Р3114

Преподаватель:

Кривоносов Егор Дмитриевич

Санкт-Петербург,

2024г

**Текст задания:**

Задание.

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).

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):  
   a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.  
   b) Н\_ВЕДОМОСТИ.ДАТА < 2010-06-18.  
   c) Н\_ВЕДОМОСТИ.ДАТА < 2010-06-18.  
   Вид соединения: LEFT JOIN.
2. 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.НЗК, Н\_УЧЕНИКИ.ИД.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ОТЧЕСТВО > Георгиевич.  
   b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД < 105590.  
   Вид соединения: RIGHT JOIN.

Команда для подключения к базе данных ucheb:

*psql -h pg -d ucheb*

Отчёт по лабораторной работе должен содержать:

1. Текст задания.
2. Реализацию запросов на SQL.
3. Планы выполнения запросов.
4. Ответы на вопросы, представленные в задании.
5. Выводы по работе.

Темы для подготовки к защите лабораторной работы:

1. Индексы
2. Оптимизация запросов
3. Выбор плана выполнения запросов
4. **Реализация запросов на языке 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.НАИМЕНОВАНИЕ, Н\_ВЕДОМОСТИ.ИД.  
      Фильтры (AND):  
      a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.  
      b) Н\_ВЕДОМОСТИ.ДАТА < 2010-06-18.  
      c) Н\_ВЕДОМОСТИ.ДАТА < 2010-06-18.  
      Вид соединения: LEFT JOIN.

SELECT "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ", "Н\_ВЕДОМОСТИ"."ИД"  
FROM "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"  
 LEFT JOIN "Н\_ВЕДОМОСТИ" on "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "Н\_ВЕДОМОСТИ"."ТВ\_ИД"  
WHERE "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ" = 'Ведомость'  
 AND "Н\_ВЕДОМОСТИ"."ДАТА"< '2010-06-18';

* 1. 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.НЗК, Н\_УЧЕНИКИ.ИД.  
     Фильтры: (AND)  
     a) Н\_ЛЮДИ.ОТЧЕСТВО > Георгиевич.  
     b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД < 105590.  
     Вид соединения: RIGHT JOIN.

SELECT "Н\_ЛЮДИ"."ОТЧЕСТВО", "Н\_ОБУЧЕНИЯ"."НЗК", "Н\_УЧЕНИКИ"."ИД"  
FROM "Н\_ЛЮДИ"  
 RIGHT JOIN "Н\_ОБУЧЕНИЯ" ON "Н\_ЛЮДИ"."ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД"  
 RIGHT JOIN "Н\_УЧЕНИКИ" ON "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД"  
WHERE "Н\_ЛЮДИ"."ОТЧЕСТВО">'Георгиевич' AND "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД"< 105590;

1. **Оптимизация запроса №1**

Создание индекса на столбце "НАИМЕНОВАНИЕ" ускорит обор записей соответствующих условию "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ" = 'Ведомость'.

CREATE INDEX ВЕДОМОСТИ\_НАИМЕНОВАНИЕ\_ИНДЕКС ON "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"("НАИМЕНОВАНИЕ");

Теперь нужно создать индекс для "Н\_ВЕДОМОСТИ"."ДАТА" получим CREATE INDEX ВЕДОМОСТИ\_ДАТА\_ИНДЕКС ON "Н\_ВЕДОМОСТИ"("ДАТА") потому что у нас есть условие "Н\_ВЕДОМОСТИ"."ДАТА"< '2010-06-18' это тоже ускорит отбор и работу запроса.

Также создание индекса для атрибута "Н\_ВЕДОМОСТИ"."ТВ\_ИД" это ускорит соединение таблиц по ТВ\_ИД.

CREATE INDEX ВЕДОМОСТИ\_ТВ\_ИД\_ИНДЕКС ON "Н\_ВЕДОМОСТИ"("ТВ\_ИД");

1. **План для запроса №1**

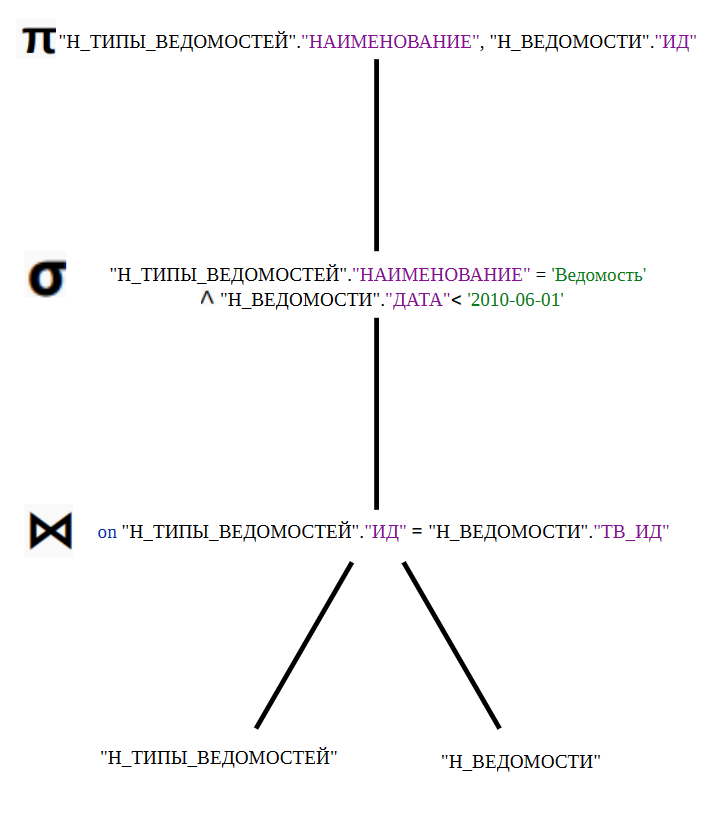
**План №1**

- Полное сканирование таблицы Н\_ТИПЫ\_ВЕДОМОСТЕЙ.

- Полное сканирование таблицы Н\_ВЕДОМОСТИ.

-Соединение таблиц с иcпользованием Nested Loop Join по атрибуту ТВ\_ИД.

- Фильтрация результата по Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ =’Ведомость’ и Н\_ВЕДОМОСТИ.ДАТА < 2010-06-18.

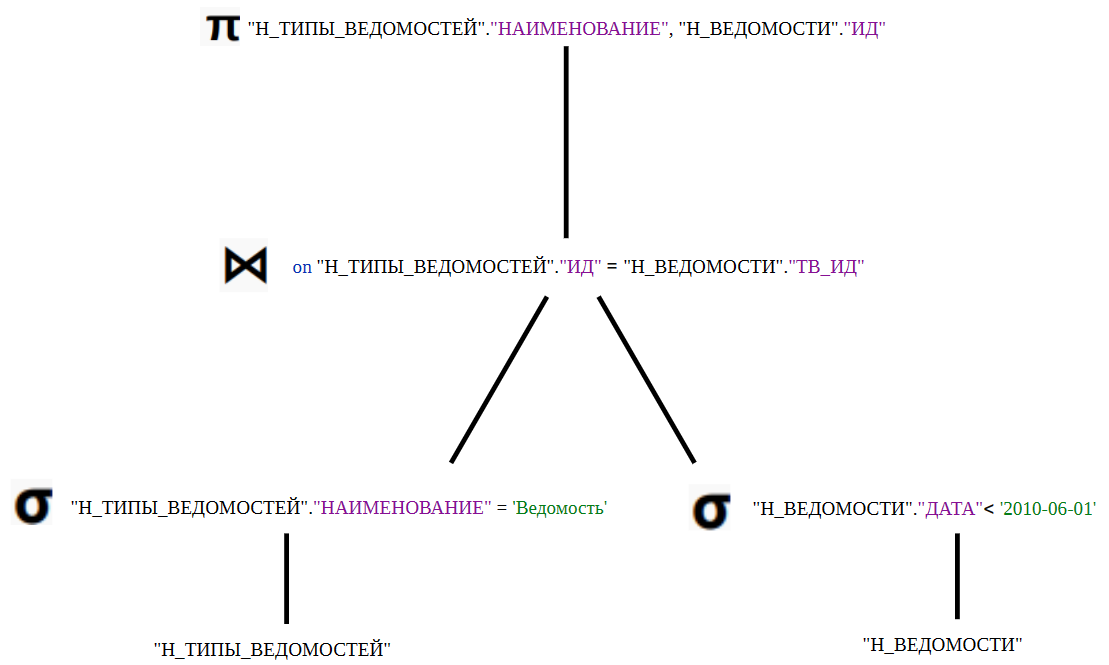


**План №2**

- Полное сканирование таблицы Н\_ТИПЫ\_ВЕДОМОСТЕЙ с использованием фильтра Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ =’Ведомость’

- Полное сканирование таблицы Н\_ВЕДОМОСТИ с применением фильтра Н\_ВЕДОМОСТИ.ДАТА < 2010-06-18.

-Соединение таблиц с иcпользованием Nested Loop Join по атрибуту ТВ\_ИД.



**Оптимальный план:**

Если сравнивать по оптимальности мои планы, то победит второй, так как фильтрация данных происходит до соединения таблиц.

**План №3 с индексами**

-Индексное сканирование будет происходить вместо полного сканирования

-Соединение таблиц с иcпользованием Nested Loop Join по атрибуту ТВ\_ИД.

**EXPLAINE ANALYZE**

Nested Loop (cost=828.49..6651.35 rows=64362 width=30) (actual time=6.919..69.461 rows=168494 loops=1)  
" -> Seq Scan on ""Н\_ТИПЫ\_ВЕДОМОСТЕЙ"" (cost=0.00..1.04 rows=1 width=30) (actual time=0.009..0.014 rows=1 loops=1)"  
" Filter: ((""НАИМЕНОВАНИЕ"")::text = 'Ведомость'::text)"  
 Rows Removed by Filter: 2  
" -> Bitmap Heap Scan on ""Н\_ВЕДОМОСТИ"" (cost=828.49..6006.69 rows=64362 width=8) (actual time=6.906..49.416 rows=168494 loops=1)"  
" Recheck Cond: (""ТВ\_ИД"" = ""Н\_ТИПЫ\_ВЕДОМОСТЕЙ"".""ИД"")"  
" Filter: (""ДАТА"" < '2010-06-01 00:00:00'::timestamp without time zone)"  
 Rows Removed by Filter: 22403  
 Heap Blocks: exact=4040  
" -> Bitmap Index Scan on ""ВЕД\_ТВ\_FK\_I"" (cost=0.00..812.40 rows=74147 width=0) (actual time=6.260..6.260 rows=190897 loops=1)"  
" Index Cond: (""ТВ\_ИД"" = ""Н\_ТИПЫ\_ВЕДОМОСТЕЙ"".""ИД"")"  
Planning Time: 1.176 ms  
Execution Time: 77.286 ms

1. **Оптимизация запроса №2**

Создание индекса на таблице Н\_ЛЮДИ, которое ускорит фильтрацию строк, где "Н\_ЛЮДИ"."ОТЧЕСТВО">'Георгиевич'.

CREATE INDEX ЛЮДИ\_ОТЧЕСТВО\_ИНДЕКС ON "Н\_ЛЮДИ"("ОТЧЕСТВО");

Создание индекса на таблицу "Н\_ОБУЧЕНИЯ", которое ускорит фильтрацию строк "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД"< 105590.

CREATE INDEX ЛЮДИ\_ИД\_ИНДЕКС ON "Н\_ОБУЧЕНИЯ"("ЧЛВК\_ИД");

1. **План для запроса №1**

**План №1**

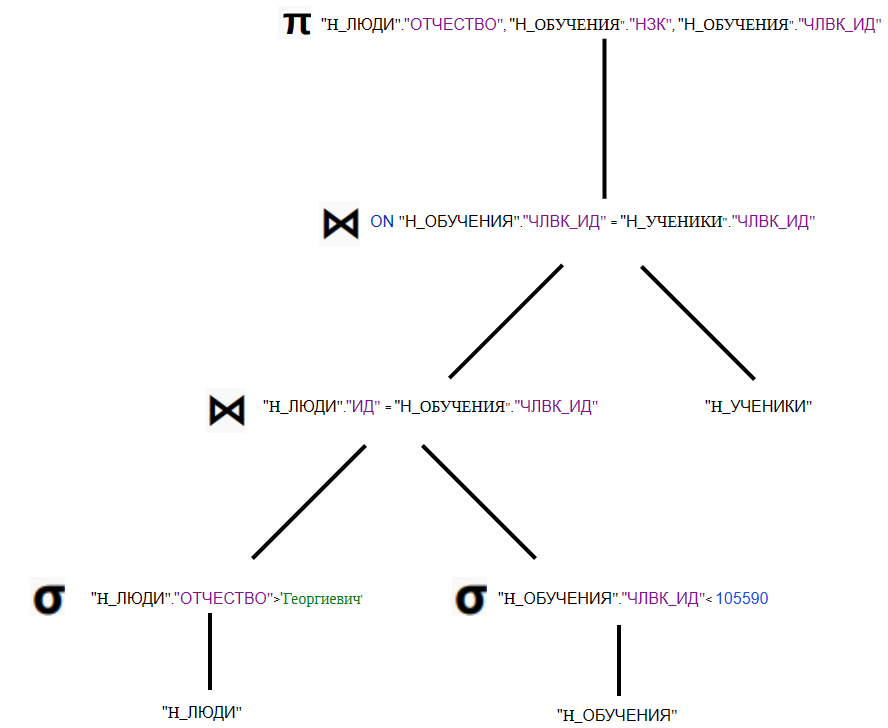
- Полный скан таблицы Н\_ЛЮДИ с применением фильтра "Н\_ЛЮДИ"."ОТЧЕСТВО" > 'Георгиевич'.

- Полный скан таблицы Н\_ОБУЧЕНИЯ с применением фильтра "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД"< 105590.

- Соединение таблиц Н\_ЛЮДИ и Н\_ОБУЧЕНИЯ с использованием Nested Loop Join по атрибутам "Н\_ЛЮДИ"."ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД".

- Полный скан таблицы Н\_УЧЕНИКИ.

- Соединение результатов предыдущего этапа с таблицей Н\_УЧЕНИКИ с использованием Nested Loop Join.



**План №2**

- Полный скан таблицы Н\_ЛЮДИ.

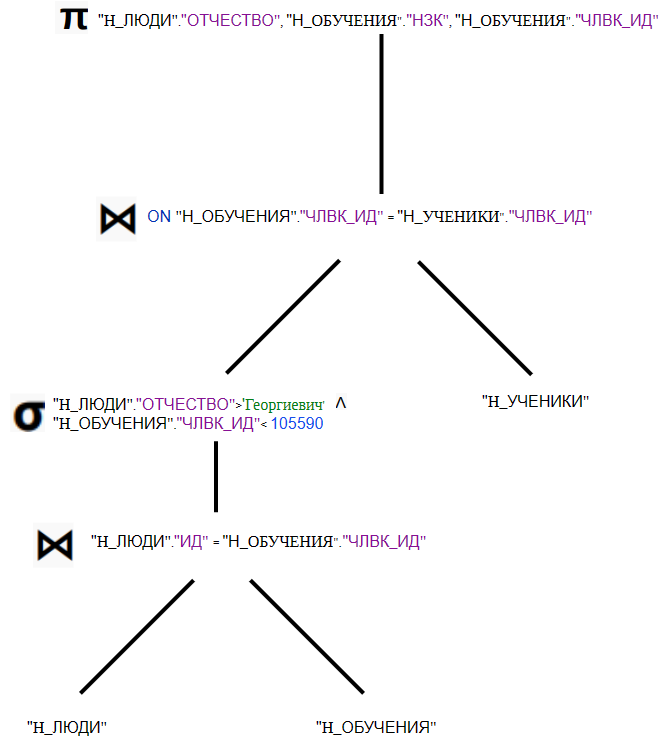
- Полный скан таблицы Н\_ОБУЧЕНИЯ.

- Соединение таблиц Н\_ЛЮДИ и Н\_ОБУЧЕНИЯ с использованием Nested Loop Join по атрибутам "Н\_ЛЮДИ"."ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД".

- Теперь применим фильтр для результата, где "Н\_ЛЮДИ"."ОТЧЕСТВО" > 'Георгиевич' и "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД"< 105590.

- Полный скан таблицы Н\_УЧЕНИКИ.

- Соединение результатов предыдущего этапа с таблицей Н\_УЧЕНИКИ с использованием Nested Loop Join.



**Оптимальный план:**

План №1 является более оптимальным, так как фильтрация происходит до соединения таблиц это уменьшает количество строк до объединения.

**План №3 с индексами**

- Будет использоваться скан по индексам

- Nested Loop Join станет быстрее благодаря индексу на атрибуте ЧЛВК\_ИД.

**EXPLAINE ANALYZE**

Нужно изменить условие, так как при условие "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД"< 105590 запрос не выведет ничего. Поменяем на < 115590.

Nested Loop (cost=95.90..1177.49 rows=1612 width=30) (actual time=0.899..10.557 rows=1801 loops=1)  
 -> Hash Join (cost=95.61..664.39 rows=354 width=34) (actual time=0.857..7.107 rows=380 loops=1)  
" Hash Cond: (""Н\_ЛЮДИ"".""ИД"" = ""Н\_ОБУЧЕНИЯ"".""ЧЛВК\_ИД"")"  
" -> Seq Scan on ""Н\_ЛЮДИ"" (cost=0.00..553.98 rows=2253 width=24) (actual time=0.024..5.677 rows=2251 loops=1)"  
" Filter: ((""ОТЧЕСТВО"")::text > 'Георгиевич'::text)"  
 Rows Removed by Filter: 2867  
 -> Hash (cost=85.56..85.56 rows=804 width=10) (actual time=0.680..0.682 rows=828 loops=1)  
 Buckets: 1024 Batches: 1 Memory Usage: 43kB  
" -> Bitmap Heap Scan on ""Н\_ОБУЧЕНИЯ"" (cost=18.51..85.56 rows=804 width=10) (actual time=0.126..0.409 rows=828 loops=1)"  
" Recheck Cond: (""ЧЛВК\_ИД"" < 115590)"  
 Heap Blocks: exact=18  
" -> Bitmap Index Scan on ""ОБУЧ\_ЧЛВК\_FK\_I"" (cost=0.00..18.31 rows=804 width=0) (actual time=0.110..0.110 rows=828 loops=1)"  
" Index Cond: (""ЧЛВК\_ИД"" < 115590)"  
" -> Index Only Scan using ""УЧЕН\_ОБУЧ\_FK\_I"" on ""Н\_УЧЕНИКИ"" (cost=0.29..1.40 rows=5 width=4) (actual time=0.003..0.007 rows=5 loops=380)"  
" Index Cond: (""ЧЛВК\_ИД"" = ""Н\_ЛЮДИ"".""ИД"")"  
 Heap Fetches: 1801  
Planning Time: 1.038 ms  
Execution Time: 10.818 ms

**Вывод:** я изучил отличия индексов, также научился оптимизировать запросы используя индексы.