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

Факультет программной инженерии

и компьютерной техники

**Отчет по лабораторной работе №3**

**Вариант № 171310**

Выполнил:

Мартышов Данила Викторович,

Группа Р3107

Преподаватель:

Байрамова Хумай Бахруз Кызы

Санкт-Петербург, 2024

**Содержание:**

**Задание3**

**Основные этапы вычисления4**

**Заключение8**

**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.ИД, Н\_ВЕДОМОСТИ.ИД.  
   Фильтры (AND):  
   a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД > 2.  
   b) Н\_ВЕДОМОСТИ.ДАТА < 1998-01-05.  
   Вид соединения: RIGHT 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.ФАМИЛИЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.ИД.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ФАМИЛИЯ > Петров.  
   b) Н\_ОБУЧЕНИЯ.НЗК < 933232.  
   Вид соединения: INNER JOIN.

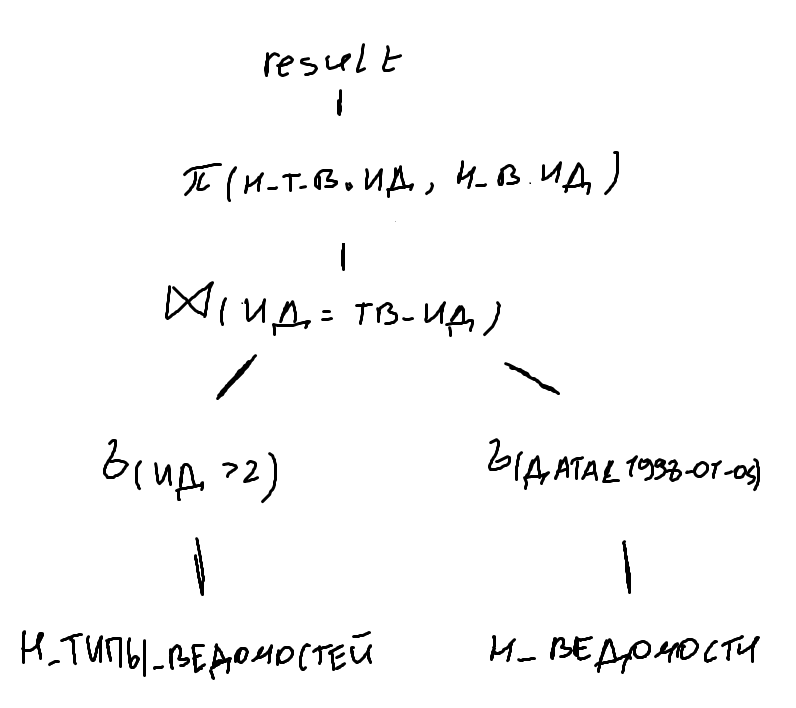
**SQL-запросы:**

*-- QUERY 1:*SELECT  
 "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД", "Н\_ВЕДОМОСТИ"."ИД"  
FROM  
 "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"  
RIGHT JOIN  
 "Н\_ВЕДОМОСТИ"  
ON  
 "Н\_ВЕДОМОСТИ"."ТВ\_ИД"="Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД"  
WHERE  
 "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД">2 AND "Н\_ВЕДОМОСТИ"."ДАТА"<'1998-01-05';  
  
*-- QUERY 2:*SELECT  
 "Н\_ЛЮДИ"."ФАМИЛИЯ", "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД", "Н\_УЧЕНИКИ"."ИД" FROM "Н\_ЛЮДИ"  
INNER JOIN  
 "Н\_ОБУЧЕНИЯ"  
ON "Н\_ЛЮДИ"."ИД"="Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД"  
INNER JOIN  
 "Н\_УЧЕНИКИ"  
ON "Н\_УЧЕНИКИ"."ЧЛВК\_ИД"="Н\_ЛЮДИ"."ИД"  
WHERE  
 "Н\_ЛЮДИ"."ФАМИЛИЯ">'Петров' AND "Н\_ОБУЧЕНИЯ"."НЗК"<'933232';

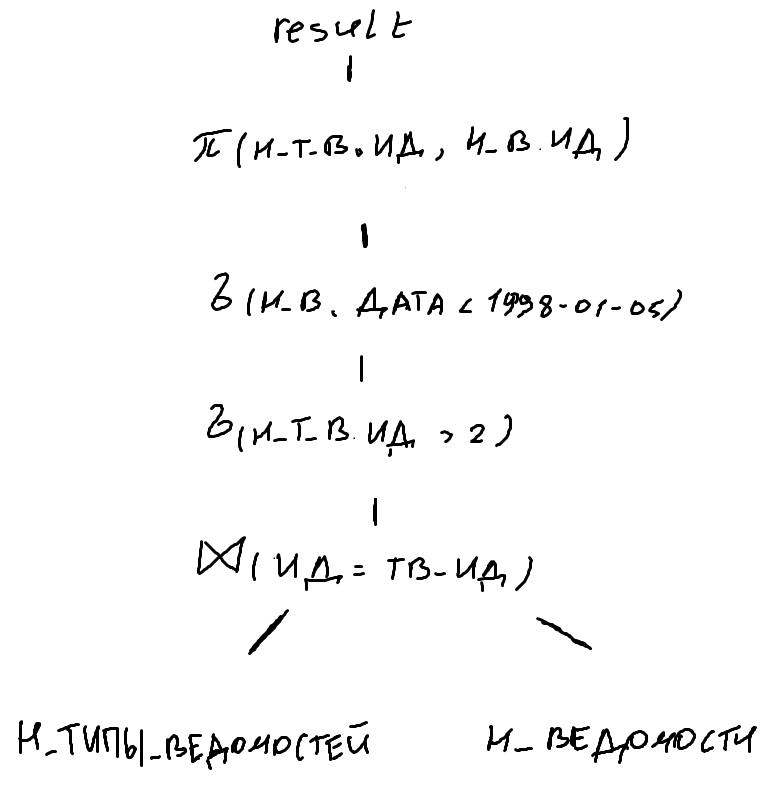
**Выполнение первого запроса:**

**Возможные планы:**

1)



**2)**



Первый план наиболее оптимальный, так как соединение происходит по меньшему числу записей, чем во втором плане.

**Explain analyze 1:**

Nested Loop (cost=0.29..8.26 rows=1 width=8) (actual time=0.004..0.005 rows=0 loops=1)

" Join Filter: (""Н\_ТИПЫ\_ВЕДОМОСТЕЙ"".""ИД"" = ""Н\_ВЕДОМОСТИ"".""ТВ\_ИД"")"

" -> Index Scan using ""ВЕД\_ДАТА\_I"" on ""Н\_ВЕДОМОСТИ"" (cost=0.29..7.20 rows=1 width=8) (actual time=0.004..0.004 rows=0 loops=1)"

" Index Cond: (""ДАТА"" < '1998-01-05 00:00:00'::timestamp without time zone)"

" -> Seq Scan on ""Н\_ТИПЫ\_ВЕДОМОСТЕЙ"" (cost=0.00..1.04 rows=2 width=4) (never executed)"

" Filter: (""ИД"" > 2)"

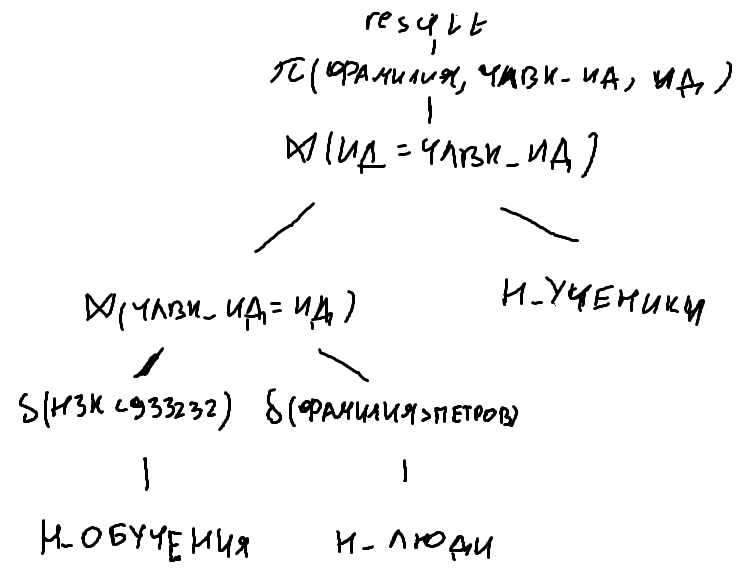
Planning Time: 1.331 ms

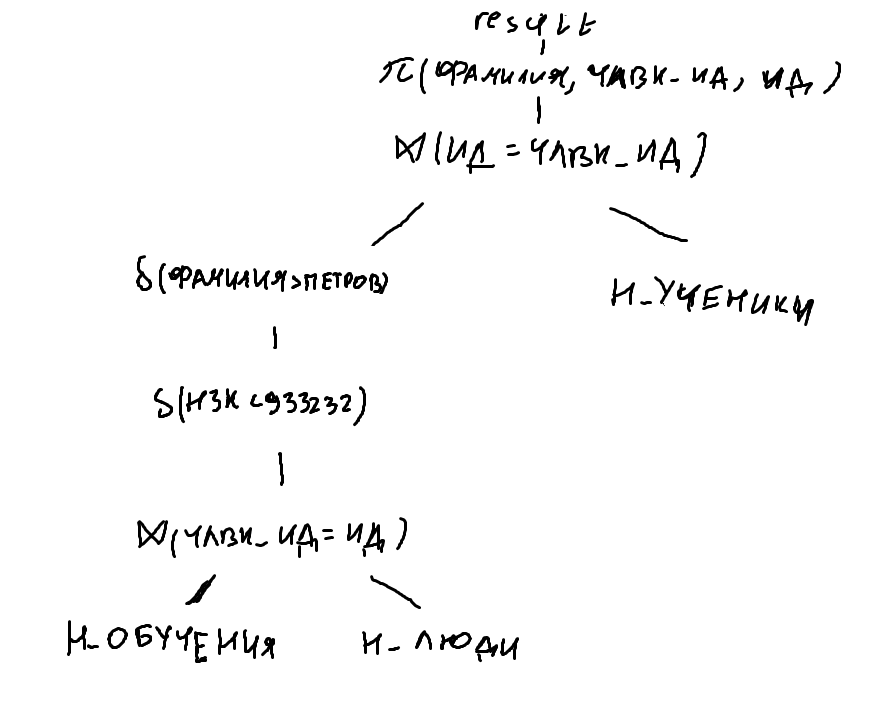
Execution Time: 0.064 ms

**Индексы:**

CREATE INDEX index\_1 ON "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" using btree("ИД");  
CREATE INDEX index\_2 ON "Н\_ВЕДОМОСТИ" using btree("ДАТА");  
CREATE INDEX index\_3 ON "Н\_ВЕДОМОСТИ" using hash("ТВ\_ИД");

**Выполнение второго запроса:**

1) 

2) 

Первый план наиболее оптимальный, так как соединение происходит по меньшему числу записей, чем во втором плане.

**Explain analyze 2:**

Hash Join (cost=719.99..1635.49 rows=5397 width=24) (actual time=6.170..13.707 rows=5083 loops=1)

" Hash Cond: (""Н\_УЧЕНИКИ"".""ЧЛВК\_ИД"" = ""Н\_ЛЮДИ"".""ИД"")"

" -> Seq Scan on ""Н\_УЧЕНИКИ"" (cost=0.00..774.11 rows=23311 width=8) (actual time=0.008..3.701 rows=23311 loops=1)"

-> Hash (cost=705.18..705.18 rows=1185 width=24) (actual time=6.138..6.140 rows=1166 loops=1)

Buckets: 2048 Batches: 1 Memory Usage: 82kB

-> Hash Join (cost=576.62..705.18 rows=1185 width=24) (actual time=3.787..5.876 rows=1166 loops=1)

" Hash Cond: (""Н\_ОБУЧЕНИЯ"".""ЧЛВК\_ИД"" = ""Н\_ЛЮДИ"".""ИД"")"

" -> Seq Scan on ""Н\_ОБУЧЕНИЯ"" (cost=0.00..119.76 rows=3346 width=4) (actual time=0.018..1.519 rows=3347 loops=1)"

" Filter: ((""НЗК"")::text < '933232'::text)"

Rows Removed by Filter: 1674

-> Hash (cost=553.98..553.98 rows=1812 width=20) (actual time=3.713..3.714 rows=1792 loops=1)

Buckets: 2048 Batches: 1 Memory Usage: 111kB

" -> Seq Scan on ""Н\_ЛЮДИ"" (cost=0.00..553.98 rows=1812 width=20) (actual time=0.102..3.345 rows=1792 loops=1)"

" Filter: ((""ФАМИЛИЯ"")::text > 'Петров'::text)"

Rows Removed by Filter: 3326

Planning Time: 1.802 ms

Execution Time: 14.022 ms

**Индексы:**

CREATE INDEX index\_1 ON "Н\_ЛЮДИ" using hash("ИД");  
CREATE INDEX index\_2 ON "Н\_ЛЮДИ" using btree("ФАМИЛИЯ");  
CREATE INDEX index\_3 ON "Н\_ОБУЧЕНИЯ" using hash("ЧЛВК\_ИД");  
CREATE INDEX index\_3 ON "Н\_ОБУЧЕНИЯ" using btree("НЗК");  
CREATE INDEX index\_5 ON "Н\_УЧЕНИКИ" using hash("ЧЛВК\_ИД");

**Заключение:**

В процессе выполнения лабораторной работы я ознакомился с индексами в SQL, а также с формированием плана выполнения запроса и его оптимизацией.