

Федеральное государственное автономное образовательное

учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**Отчёт по лабораторной работе 4**

Предмет: БД

Варинат: 877642

Выполнил**:** студент группы Р3115 Храбров Артём Алексеевич

Проверил**:** Райла Мартин

Дата сдачи: xx.xx.xx

2025

## 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.  
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:  
   Таблицы: Н\_ЛЮДИ, Н\_СЕССИЯ.  
   Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_СЕССИЯ.ДАТА.  
   Фильтры (AND):  
   a) Н\_ЛЮДИ.ФАМИЛИЯ < Афанасьев.  
   b) Н\_СЕССИЯ.УЧГОД < 2006/2007.  
   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.ЧЛВК\_ИД, Н\_СЕССИЯ.УЧГОД.  
   Фильтры (AND):  
   a) Н\_ЛЮДИ.ИМЯ = Александр.  
   b) Н\_ВЕДОМОСТИ.ИД > 39921.  
   c) Н\_СЕССИЯ.УЧГОД > 2004/2005.  
   Вид соединения: RIGHT JOIN.

## Запрос 1

## 1

SELECT hum.ФАМИЛИЯ, ses.ДАТА FROM Н\_ЛЮДИ AS hum

LEFT JOIN Н\_СЕССИЯ AS ses ON hum.ИД = ses.ЧЛВК\_ИД

WHERE hum.ФАМИЛИЯ < 'Афанасьев' AND ses.УЧГОД < '2006/2007';

## План выполнения запроса



План 2 будет оптимальнее, потому что не приходится объединять таблицы, в которых много не нужных впоследствие атрибутов. Количество и размер передаваемых данных меньше чем в первом случае.

## Индексы

CREATE INDEX idx\_Н\_ЛЮДИ\_ФАМИЛИЯ ON Н\_ЛЮДИ(ФАМИЛИЯ);

CREATE INDEX idx\_Н\_СЕССИЯ\_УЧГОД ON Н\_СЕССИЯ(УЧГОД);

Эти индексы ускорят выбор строк из таблиц, подходящих под условия фильтрации.

## EXPLAIN ANALYZE

### БЕЗ ИНДЕКСОВ

Hash Join (cost=111.42..232.99 rows=51 width=24) (actual time=0.432..1.832 rows=28 loops=1)

Hash Cond: (ses."ЧЛВК\_ИД" = hum."ИД")

-> Seq Scan on "Н\_СЕССИЯ" ses (cost=0.00..117.90 rows=1396 width=12) (actual time=0.020..1.340 rows=1396 loops=1)

Filter: (("УЧГОД")::text < '2006/2007'::text)

Rows Removed by Filter: 2356

-> Hash (cost=108.70..108.70 rows=218 width=20) (actual time=0.344..0.346 rows=216 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 20kB

-> Bitmap Heap Scan on "Н\_ЛЮДИ" hum (cost=5.97..108.70 rows=218 width=20) (actual time=0.083..0.296 rows=216 loops=1)

Recheck Cond: (("ФАМИЛИЯ")::text < 'Афанасьв'::text)

Heap Blocks: exact=91

-> Bitmap Index Scan on "ФАМ\_ЛЮД" (cost=0.00..5.92 rows=218 width=0) (actual time=0.061..0.061 rows=216 loops=1)

Index Cond: (("ФАМИЛИЯ")::text < 'Афанасьв'::text)

Planning Time: 0.660 ms

Execution Time: 1.925 ms

### С ИНДЕКСАМИ

Hash Join (cost=134.52..226.64 rows=51 width=24) (actual time=0.412..0.726 rows=28 loops=1)

Hash Cond: (ses."ЧЛВК\_ИД" = hum."ИД")

-> Bitmap Heap Scan on "Н\_СЕССИЯ" ses (cost=23.10..111.55 rows=1396 width=12) (actual time=0.073..0.250 rows=1396 loops=1)

Recheck Cond: (("УЧГОД")::text < '2006/2007'::text)

Heap Blocks: exact=32

-> Bitmap Index Scan on "idx\_Н\_СЕССИЯ\_УЧГОД" (cost=0.00..22.75 rows=1396 width=0) (actual time=0.061..0.061 rows=1396 loops=1)

Index Cond: (("УЧГОД")::text < '2006/2007'::text)

-> Hash (cost=108.70..108.70 rows=218 width=20) (actual time=0.250..0.250 rows=216 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 20kB

-> Bitmap Heap Scan on "Н\_ЛЮДИ" hum (cost=5.97..108.70 rows=218 width=20) (actual time=0.066..0.211 rows=216 loops=1)

Recheck Cond: (("ФАМИЛИЯ")::text < 'Афанасьв'::text)

Heap Blocks: exact=91

-> Bitmap Index Scan on "idx\_Н\_ЛЮДИ\_ФАМИЛИЯ" (cost=0.00..5.92 rows=218 width=0) (actual time=0.048..0.048 rows=216 loops=1)

Index Cond: (("ФАМИЛИЯ")::text < 'Афанасьв'::text)

Planning Time: 1.222 ms

Execution Time: 0.853 ms

## Запрос 2

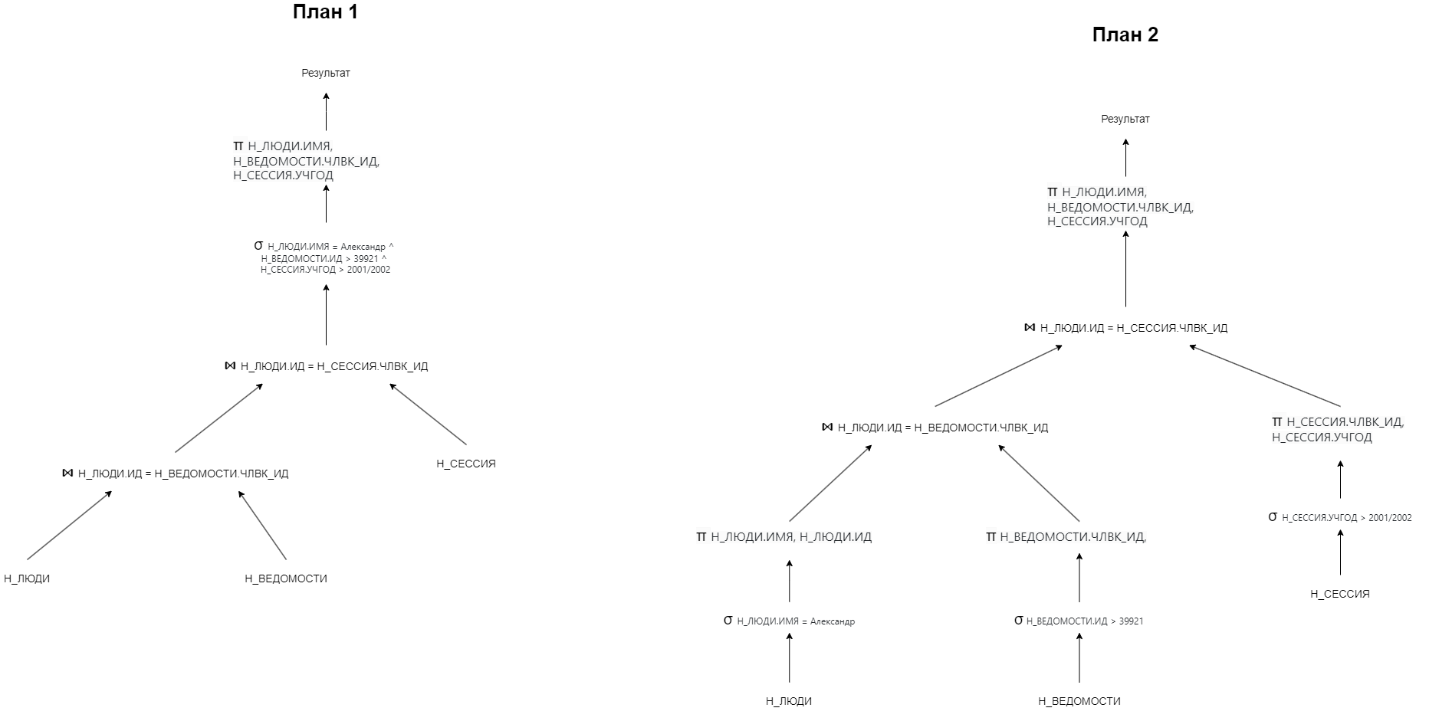
SELECT hum.ИМЯ, ved.ЧЛВК\_ИД, ses.УЧГОД FROM Н\_ЛЮДИ AS hum

RIGHT JOIN Н\_ВЕДОМОСТИ AS ved ON hum.ИД = ved.ЧЛВК\_ИД

RIGHT JOIN Н\_СЕССИЯ AS ses ON hum.ИД = ses.ЧЛВК\_ИД

WHERE hum.ИМЯ = 'Александр' AND ved.ИД > 39921 AND ses.УЧГОД > '2004/2005';

## План выполнения запроса



## Индексы

CREATE INDEX idx\_Н\_ЛЮДИ\_ИМЯ ON Н\_ЛЮДИ(ИМЯ);

CREATE INDEX idx\_Н\_ВЕДОМОСТИ\_ИД ON Н\_ВЕДОМОСТИ(ИД);

CREATE INDEX idx\_Н\_СЕССИЯ\_УЧГОД ON Н\_СЕССИЯ(УЧГОД);

Индексы ускорят отбор нужных строк во время фильтрации данных из таблиц. Первый индекс использует hash для создания индекса, это ускоряет поиск, но не позволяет использовать этот индекс для операций сравнения.

## EXPLAIN ANALYZE

### БЕЗ ИНДЕКСОВ

Nested Loop (cost=0.60..1666.79 rows=10161 width=27) (actual time=0.949..10.168 rows=3244 loops=1)

Join Filter: (hum."ИД" = ved."ЧЛВК\_ИД")

-> Nested Loop (cost=0.29..286.49 rows=234 width=31) (actual time=0.346..6.887 rows=254 loops=1)

-> Seq Scan on "Н\_СЕССИЯ" ses (cost=0.00..117.90 rows=3630 width=14) (actual time=0.053..1.966 rows=3630 loops=1)

Filter: (("УЧГОД")::text > '2001/2002'::text)

Rows Removed by Filter: 122

-> Memoize (cost=0.29..0.44 rows=1 width=17) (actual time=0.001..0.001 rows=0 loops=3630)

Cache Key: ses."ЧЛВК\_ИД"

Cache Mode: logical

Hits: 3452 Misses: 178 Evictions: 0 Overflows: 0 Memory Usage: 13kB

-> Index Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" hum (cost=0.28..0.43 rows=1 width=17) (actual time=0.015..0.015 rows=0 loops=178)

Index Cond: ("ИД" = ses."ЧЛВК\_ИД")

Filter: (("ИМЯ")::text = 'Александр'::text)

Rows Removed by Filter: 1

-> Memoize (cost=0.30..6.33 rows=68 width=4) (actual time=0.002..0.009 rows=13 loops=254)

Cache Key: ses."ЧЛВК\_ИД"

Cache Mode: logical

Hits: 236 Misses: 18 Evictions: 0 Overflows: 0 Memory Usage: 19kB

-> Index Scan using "ВЕД\_ЧЛВК\_FK\_IFK" on "Н\_ВЕДОМОСТИ" ved (cost=0.29..6.32 rows=68 width=4) (actual time=0.022..0.107 rows=27 loops=18)

Index Cond: ("ЧЛВК\_ИД" = ses."ЧЛВК\_ИД")

Filter: ("ИД" > 39921)

Planning Time: 6.061 ms

Execution Time: 10.803 ms

### С ИНДЕКСАМИ

Nested Loop (cost=117.10..1624.54 rows=10161 width=27) (actual time=0.895..3.427 rows=3244 loops=1)

Join Filter: (hum."ИД" = ved."ЧЛВК\_ИД")

-> Hash Join (cost=116.79..244.23 rows=234 width=31) (actual time=0.584..1.748 rows=254 loops=1)

Hash Cond: (ses."ЧЛВК\_ИД" = hum."ИД")

-> Seq Scan on "Н\_СЕССИЯ" ses (cost=0.00..117.90 rows=3630 width=14) (actual time=0.013..0.857 rows=3630 loops=1)

Filter: (("УЧГОД")::text > '2001/2002'::text)

Rows Removed by Filter: 122

-> Hash (cost=112.02..112.02 rows=382 width=17) (actual time=0.499..0.500 rows=382 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 29kB

-> Bitmap Heap Scan on "Н\_ЛЮДИ" hum (cost=7.24..112.02 rows=382 width=17) (actual time=0.149..0.428 rows=382 loops=1)

Recheck Cond: (("ИМЯ")::text = 'Александр'::text)

Heap Blocks: exact=98

-> Bitmap Index Scan on "idx\_Н\_ЛЮДИ\_ИМЯ" (cost=0.00..7.15 rows=382 width=0) (actual time=0.111..0.111 rows=382 loops=1)

Index Cond: (("ИМЯ")::text = 'Александр'::text)

-> Memoize (cost=0.30..6.33 rows=68 width=4) (actual time=0.003..0.005 rows=13 loops=254)

Cache Key: ses."ЧЛВК\_ИД"

Cache Mode: logical

Hits: 236 Misses: 18 Evictions: 0 Overflows: 0 Memory Usage: 19kB

-> Index Scan using "ВЕД\_ЧЛВК\_FK\_IFK" on "Н\_ВЕДОМОСТИ" ved (cost=0.29..6.32 rows=68 width=4) (actual time=0.029..0.049 rows=27 loops=18)

Index Cond: ("ЧЛВК\_ИД" = ses."ЧЛВК\_ИД")

Filter: ("ИД" > 39921)

Planning Time: 10.973 ms

Execution Time: 3.692 ms

## Вывод

Я узнал как создавать индексы, для чего они используются и как помогают оптимизировать поиск данных в базе.