

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

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

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ**

## **Лабораторная работа №4**

по дисциплине

**Информационные системы и базы данных**

Вариант 312493

Выполнил:

Студент группы Р33302

Ершова А. И.

Преподаватель:

Шешуков Д. М.

## Описание 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.

Изменяются ли планы при добавлении индекса и как?

Для запросов 1–2 необходимо добавить в отчет вывод команды EXPLAIN ANALYZE [запрос]

## Реализация первого запроса

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.УЧГОД.

Фильтры (AND):

а) Н\_ЛЮДИ.ИМЯ = Владимир.

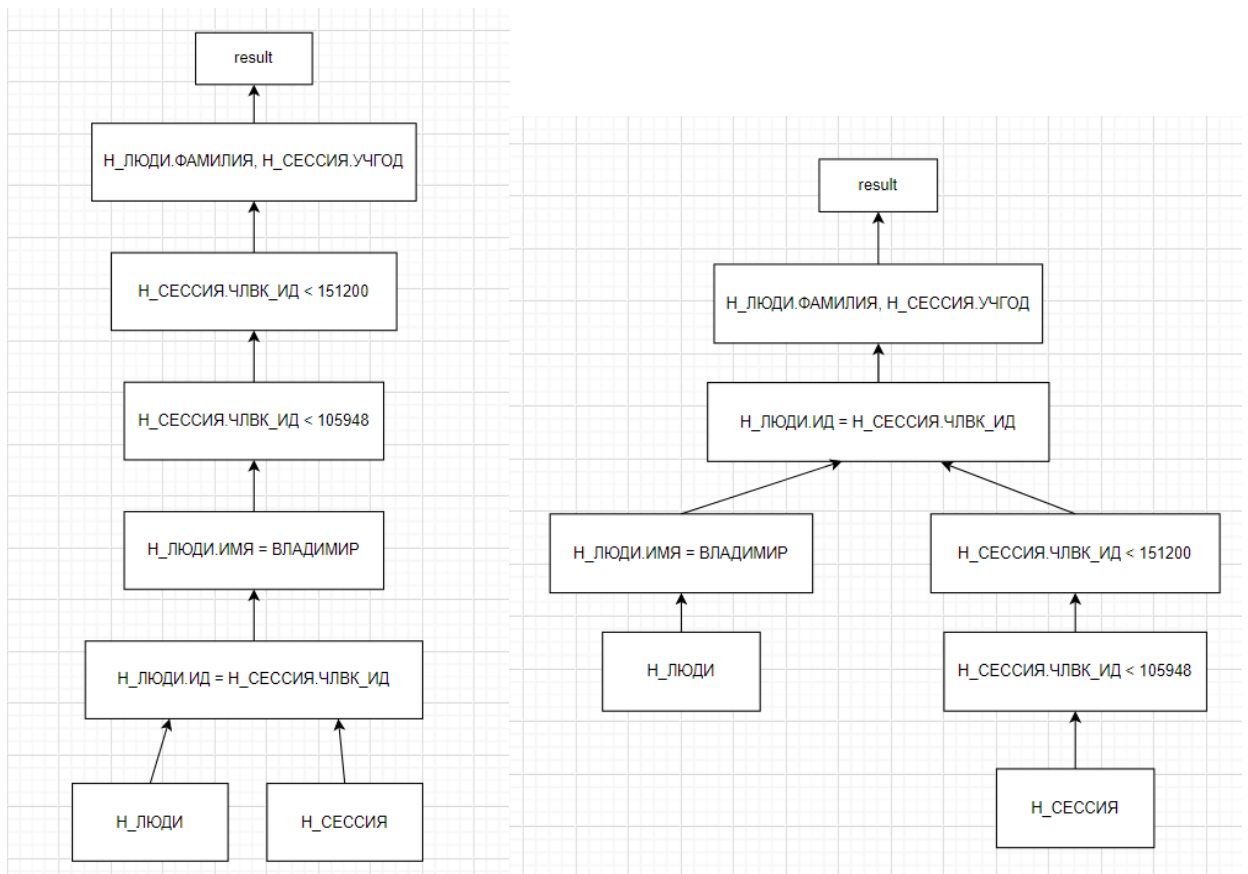
б) Н\_СЕССИЯ.ЧЛВК\_ИД < 105948.

с) Н\_СЕССИЯ.ЧЛВК\_ИД < 151200.

Вид соединения: INNER JOIN.

```
SELECT Н_ЛЮДИ.ФАМИЛИЯ, Н_СЕССИЯ.УЧГОД
FROM Н_ЛЮДИ
      INNER JOIN Н_СЕССИЯ ON Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД
WHERE ( Н_ЛЮДИ.ИМЯ = 'Владимир' AND Н_СЕССИЯ.ЧЛВК_ИД < 105948 AND
       Н_СЕССИЯ.ЧЛВК_ИД < 151200);
```

*Планы выполнения:*



Во втором плане происходит объединение только нужной выборки, а не всех сущностей. Размер промежуточных данных меньше, значит этот план является оптимальным.

### Индексы:

```
CREATE INDEX ON "Н_люди" USING HASH ("ИД");
CREATE INDEX ON "Н_люди" USING BTREE ("ОТЧЕСТВО");
CREATE INDEX ON "Н_сессия" USING BTREE ("УЧГОД");
CREATE INDEX ON "Н_сессия" USING BTREE ("ЧЛВК_ИД");
```

Выборка происходит с использованием операторов сравнения, поэтому оптимально использование BTREE. Соединение таблиц с использованием ИД – оптимально использование HASH.

### Explain Analyze:

```
Nested Loop (cost=0.29..279.69 rows=49 width=26) (actual
time=0.087..2.077 rows=173 loops=1)
```

```
-> Seq Scan on "Н_сессия" (cost=0.00..127.28 rows=2631 width=14)
(actual time=0.023..0.864 rows=2633 loops=1)
```

```
Filter: (("ЧЛВК_ИД" < 105948) AND ("ЧЛВК_ИД" < 151200))
```

```
Rows Removed by Filter: 1119
```

```
-> Memoize (cost=0.29..0.49 rows=1 width=20) (actual
time=0.000..0.000 rows=0 loops=2633)
```

Cache Key: "Н\_СЕССИЯ"."ЧЛВК\_ИД"

Cache Mode: logical

Hits: 2535 Misses: 98 Evictions: 0 Overflows: 0 Memory Usage: 7kB

-> Index Scan using "ЧЛВК\_ПК" on "Н\_ЛЮДИ" (cost=0.28..0.48 rows=1 width=20) (actual time=0.003..0.003 rows=0 loops=98)

Index Cond: ("ИД" = "Н\_СЕССИЯ"."ЧЛВК\_ИД")

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

Rows Removed by Filter: 1

Planning Time: 1.052 ms

Execution Time: 2.135 ms

	QUERY PLAN text	
1	Nested Loop (cost=0.29..279.69 rows=49 width=26) (actual time=0.046..1.925 rows=173 loops=1)	
2	[...] -> Seq Scan on "Н_СЕССИЯ" (cost=0.00..127.28 rows=2631 width=14) (actual time=0.011..0.823 rows=2633 loops=1)	
3	[...] Filter: (("ЧЛВК_ИД" < 105948) AND ("ЧЛВК_ИД" < 151200))	
4	[...] Rows Removed by Filter: 1119	
5	[...] -> Memoize (cost=0.29..0.49 rows=1 width=20) (actual time=0.000..0.000 rows=0 loops=2633)	
6	[...] Cache Key: "Н_СЕССИЯ"."ЧЛВК_ИД"	
7	[...] Cache Mode: logical	
8	[...] Hits: 2535 Misses: 98 Evictions: 0 Overflows: 0 Memory Usage: 7kB	
9	[...] -> Index Scan using "ЧЛВК_ПК" on "Н_ЛЮДИ" (cost=0.28..0.48 rows=1 width=20) (actual time=0.002..0.002 rows=0 loops=98)	
10	[...] Index Cond: ("ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД")	
11	[...] Filter: (("ИМЯ")::text = 'Владимир')::text)	
12	[...] Rows Removed by Filter: 1	
13	Planning Time: 0.339 ms	
14	Execution Time: 1.977 ms	

## Реализация второго запроса

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

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

Вывести атрибуты: Н\_ЛЮДИ.ИД, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.НАЧАЛО.

Фильтры: (AND)

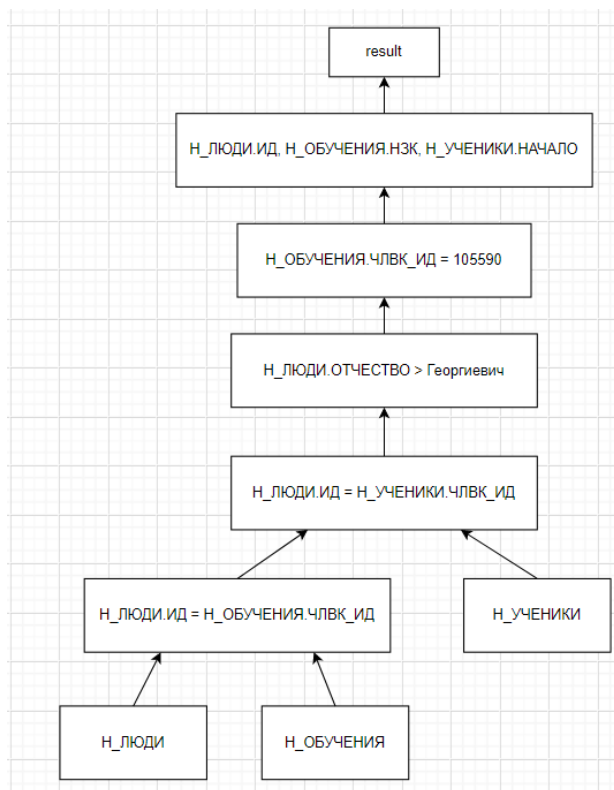
а) Н\_ЛЮДИ.ОТЧЕСТВО > Георгиевич.

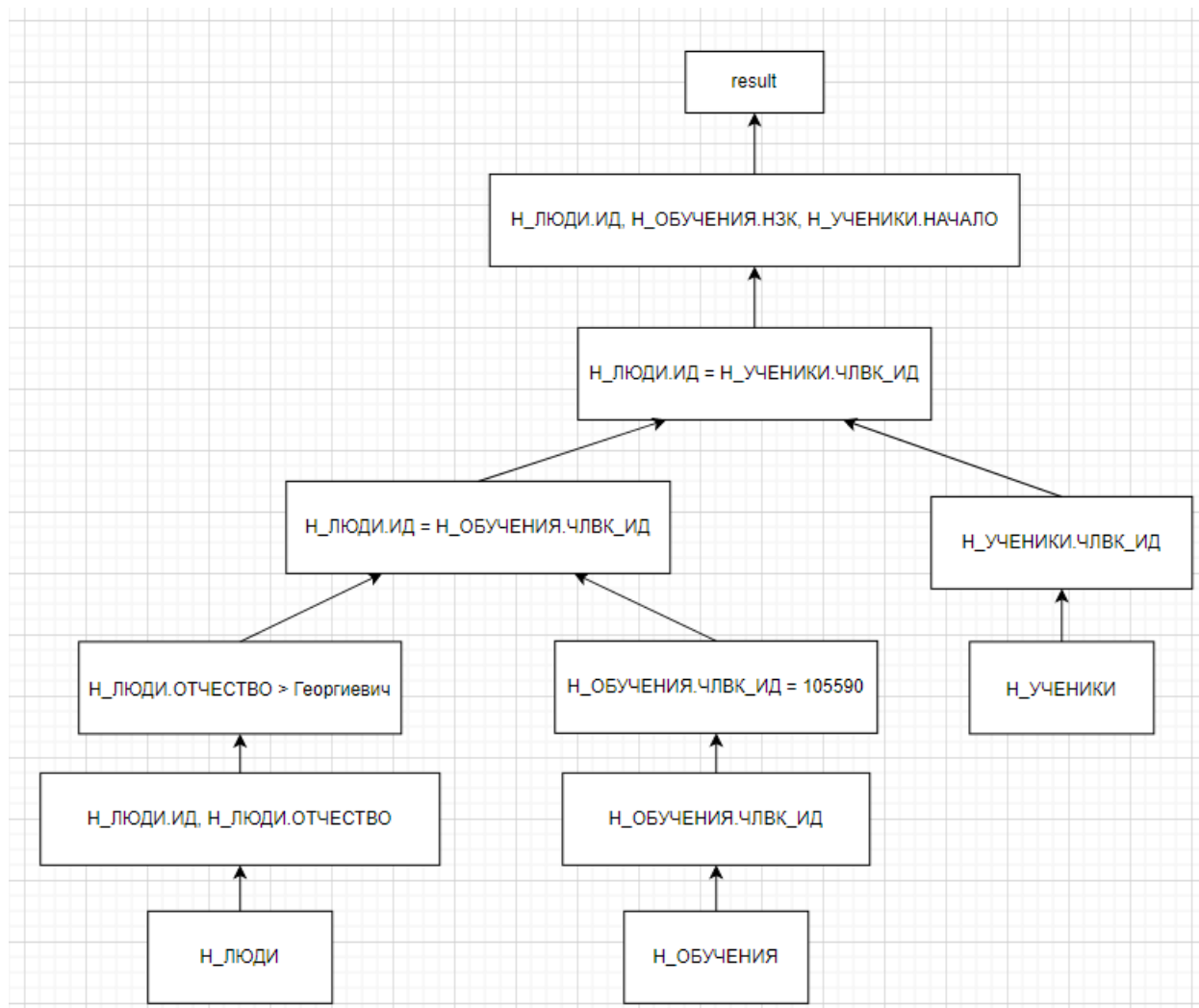
б) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 105590.

Вид соединения: INNER JOIN.

```
SELECT Н_ЛЮДИ.ИД, Н_ОБУЧЕНИЯ.НЗК, Н_УЧЕНИКИ.НАЧАЛО
FROM Н_ЛЮДИ
      INNER JOIN Н_ОБУЧЕНИЯ ON Н_ЛЮДИ.ИД = Н_ОБУЧЕНИЯ.ЧЛВК_ИД
      INNER JOIN Н_УЧЕНИКИ ON Н_ЛЮДИ.ИД = Н_УЧЕНИКИ.ЧЛВК_ИД
WHERE (Н_ЛЮДИ.ОТЧЕСТВО > 'Георгиевич' AND Н_ОБУЧЕНИЯ.ЧЛВК_ИД =
105590);
```

*Планы выполнения:*





Второй план является оптимальным. Из-за того, что выборка происходит на более ранних этапах, идет соединение только нужных атрибутов, и размер промежуточных данных меньше.

*Индексы:*

```
CREATE INDEX "Н_люди" USING HASH ("ИД");
```

```
CREATE INDEX "Н_люди" USING BTREE ("ОТЧЕСТВО");
```

```
CREATE INDEX "Н_обучения" USING BTREE ("ЧЛВК_ИД");
```

Выборка происходит с использованием операторов сравнения, поэтому оптимально использование BTREE. Соединение таблиц с использованием ИД – оптимально использование HASH.

```
Nested Loop (cost=4.89..39.61 rows=5 width=18) (actual
time=0.016..0.017 rows=0 loops=1)
```

```
-> Nested Loop (cost=0.56..16.61 rows=1 width=10) (actual
time=0.016..0.016 rows=0 loops=1)
```

```
-> Index Scan using "ЧЛВК_ПК" on "Н_люди" (cost=0.28..8.30
rows=1 width=4) (actual time=0.015..0.015 rows=0 loops=1)
```

Index Cond: ("ИД" = 105590)

Filter: (("ОТЧЕСТВО")::text > 'Георгиевич'::text)

Rows Removed by Filter: 1

-> Index Scan using "ОБУЧ\_ЧЛВК\_ФК\_I" on "Н\_ОБУЧЕНИЯ"  
(cost=0.28..8.30 rows=1 width=10) (never executed)

Index Cond: ("ЧЛВК\_ИД" = 105590)

-> Bitmap Heap Scan on "Н\_УЧЕНИКИ" (cost=4.33..22.95 rows=5  
width=12) (never executed)

Recheck Cond: ("ЧЛВК\_ИД" = 105590)

-> Bitmap Index Scan on "УЧЕН\_ОБУЧ\_ФК\_I" (cost=0.00..4.32  
rows=5 width=0) (never executed)

Index Cond: ("ЧЛВК\_ИД" = 105590)

Planning Time: 1.122 ms

Execution Time: 0.066 ms

QUERY PLAN		
	text	
1	Nested Loop (cost=4.89..39.61 rows=5 width=18) (actual time=0.025..0.026 rows=0 loops=1)	
2	[...] -> Nested Loop (cost=0.56..16.61 rows=1 width=10) (actual time=0.025..0.025 rows=0 loops=1)	
3	[...] -> Index Scan using "ЧЛВК_ПК" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1 width=4) (actual time=0.024..0.025 rows=0 loops=1)	
4	[...] Index Cond: ("ИД" = 105590)	
5	[...] Filter: (("ОТЧЕСТВО")::text > 'Георгиевич'::text)	
6	[...] Rows Removed by Filter: 1	
7	[...] -> Index Scan using "ОБУЧ_ЧЛВК_ФК_I" on "Н_ОБУЧЕНИЯ" (cost=0.28..8.30 rows=1 width=10) (never executed)	
8	[...] Index Cond: ("ЧЛВК_ИД" = 105590)	
9	[...] -> Bitmap Heap Scan on "Н_УЧЕНИКИ" (cost=4.33..22.95 rows=5 width=12) (never executed)	
10	[...] Recheck Cond: ("ЧЛВК_ИД" = 105590)	
11	[...] -> Bitmap Index Scan on "УЧЕН_ОБУЧ_ФК_I" (cost=0.00..4.32 rows=5 width=0) (never executed)	
12	[...] Index Cond: ("ЧЛВК_ИД" = 105590)	
13	Planning Time: 0.336 ms	
14	Execution Time: 0.069 ms	

## Выводы

При выполнении лабораторной работы я познакомилась с использованием индексов для ускорения обработки запросов в SQL. Изучила, как поэтапно происходит выполнение запроса, какие планы могут использоваться и как индексы влияют на процесс.