



**Вариант №8871**  
**Лабораторная работа №4**  
**По дисциплине**  
**Базы Данных**

*Выполнил:*  
Студент группы Р3116  
Воронов Григорий  
Алексеевич

*Преподаватель:*  
Гаврилов Антон  
Валерьевич  
Николаев Владимир  
Вячеславович

## 1. Текст задания

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.ИД, Н\_СЕССИЯ.УЧГОД.  
Фильтры (AND):  
а) Н\_ЛЮДИ.ФАМИЛИЯ < Иванов.  
б) Н\_СЕССИЯ.ИД = 1975.  
Вид соединения: INNER 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.ГРУППА.  
Фильтры: (AND)  
а) Н\_ЛЮДИ.ИМЯ = Ярослав.  
б) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 112514.  
с) Н\_УЧЕНИКИ.ИД = 150308.  
Вид соединения: LEFT JOIN

## 2. Реализация запросов на SQL

```
-- Сделать запрос для получения атрибутов из указанных таблиц, применив
-- фильтры по указанным условиям:
-- Таблицы: Н_ЛЮДИ, Н_СЕССИЯ.
-- Вывести атрибуты: Н_ЛЮДИ.ИД, Н_СЕССИЯ.УЧГОД.
-- Фильтры (AND):
-- а) Н_ЛЮДИ.ФАМИЛИЯ < Иванов.
-- б) Н_СЕССИЯ.ИД = 1975.
-- Вид соединения: INNER JOIN.
SELECT Н_ЛЮДИ.ИД, Н_СЕССИЯ.УЧГОД FROM Н_ЛЮДИ
JOIN Н_СЕССИЯ ON Н_СЕССИЯ.ЧЛВК_ИД=Н_ЛЮДИ.ИД
WHERE Н_ЛЮДИ.ФАМИЛИЯ < 'Иванов' AND Н_СЕССИЯ.ИД = 1975;

-- 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.ГРУППА.
```

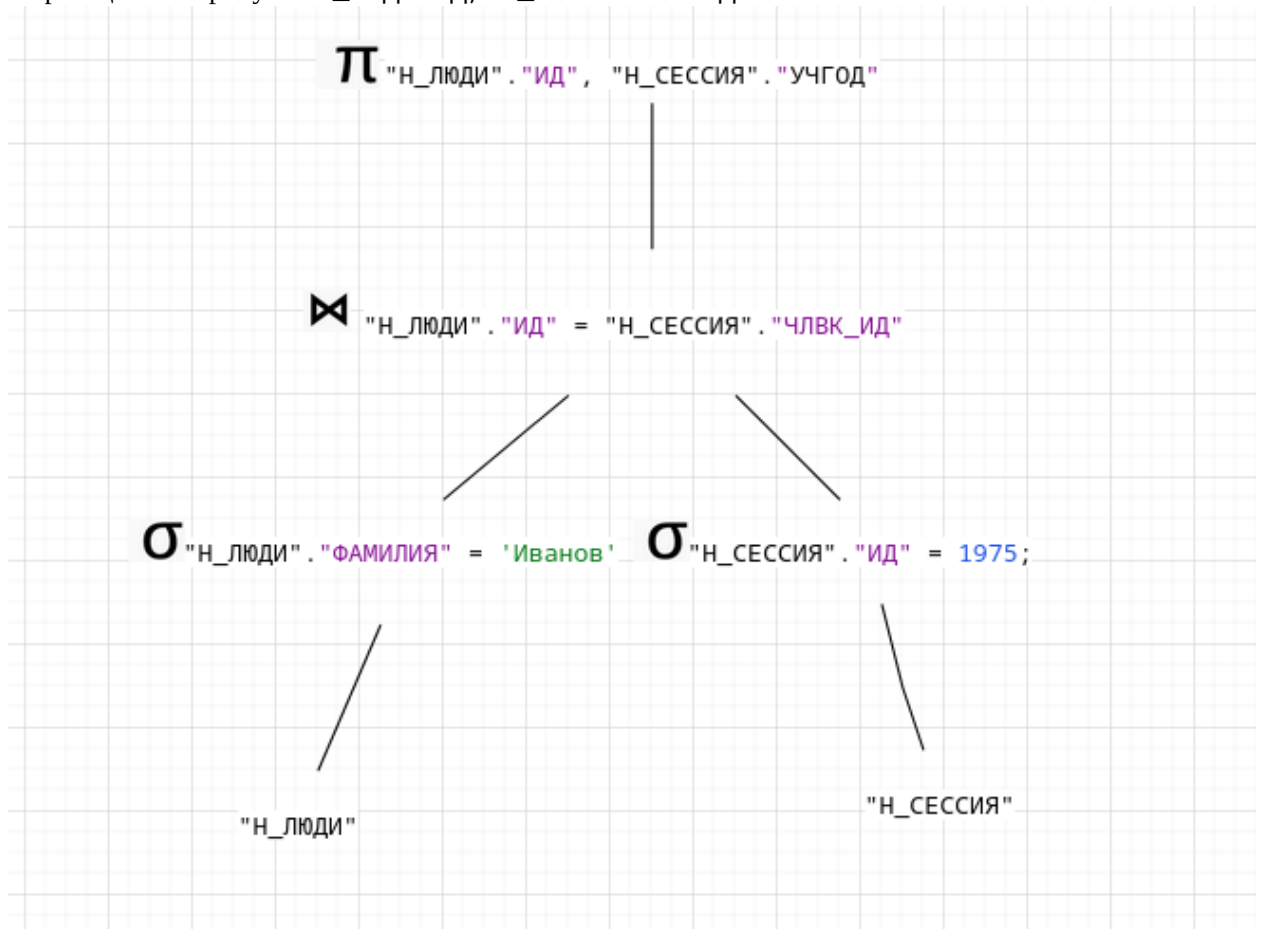
```
-- Фильтры: (AND)
-- а) Н_ЛЮДИ.ИМЯ = Ярослав.
-- б) Н_ОБУЧЕНИЯ.ЧЛВК_ИД = 112514.
-- с) Н_УЧЕНИКИ.ИД = 150308.
-- Вид соединения: LEFT JOIN.
SELECT Н_ЛЮДИ.ОТЧЕСТВО, Н_ОБУЧЕНИЯ.НЗК, Н_УЧЕНИКИ.ГРУППА FROM Н_ЛЮДИ
LEFT JOIN Н_ОБУЧЕНИЯ ON Н_ОБУЧЕНИЯ.ЧЛВК_ИД = Н_ЛЮДИ.ИД
LEFT JOIN Н_УЧЕНИКИ ON Н_УЧЕНИКИ.ЧЛВК_ИД = Н_ЛЮДИ.ИД
WHERE Н_ЛЮДИ.ИМЯ = 'Ярослав' AND Н_ОБУЧЕНИЯ.ЧЛВК_ИД = 112514 AND
Н_УЧЕНИКИ.ИД = 150308;
```

### 3. Оптимизация 1 запроса

- 1) Индекс на атрибуте "Н\_ЛЮДИ"."ФАМИЛИЯ" ускорит фильтрацию строк с условием "Н\_ЛЮДИ"."ФАМИЛИЯ" = 'Иванов'  
CREATE INDEX people\_surname\_index ON "Н\_ЛЮДИ" ("ФАМИЛИЯ");
- 2) Индекс на атрибуте "Н\_СЕССИЯ"."ЧЛВК\_ИД" ускорит соединение таблиц "Н\_СЕССИЯ" И "Н\_ЛЮДИ"  
CREATE INDEX session\_people\_id\_index ON "Н\_СЕССИЯ"("ЧЛВК\_ИД");

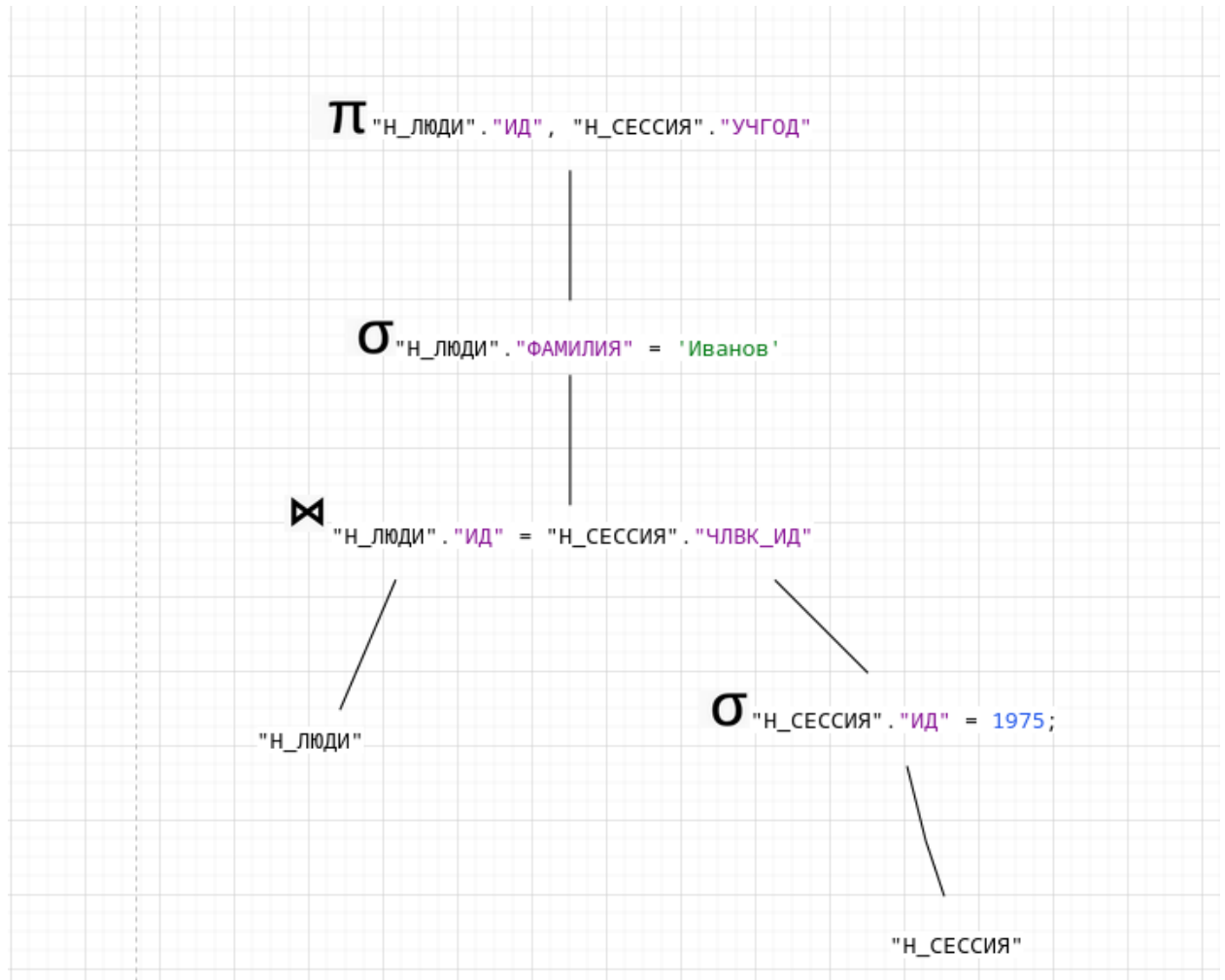
**Возможный план выполнения запроса, при отсутствии индексов:**

- 1)
  - Полный скан таблицы "Н\_ЛЮДИ" с фильтрацией по условию "Н\_ЛЮДИ"."ФАМИЛИЯ" = 'Иванов'
  - Полный скан таблицы "Н\_СЕССИЯ" с фильтрацией "Н\_СЕССИЯ"."ИД" = 1975
  - Соединение таблиц по атрибутам Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД
  - Проекция по атрибутам Н\_ЛЮДИ.ИД, Н\_СЕССИЯ.УЧГОД



2)

- Полный скан таблицы "Н\_ЛЮДИ".
- Полный скан таблицы "Н\_СЕССИЯ" с использованием фильтрации по условию "Н\_СЕССИЯ"."ИД" = 1975.
- Соединение таблиц по атрибутам "Н\_ЛЮДИ.ИД" и "Н\_СЕССИЯ"."ЧЛВК\_ИД".
- Фильтрация результирующего набора по условию "Н\_ЛЮДИ"."ФАМИЛИЯ" = 'Иванов'
- Проекция по атрибутам Н\_ЛЮДИ.ИД, Н\_СЕССИЯ.УЧГОД



### Оптимальный план:

План 1, так как фильтрация данных по "Н\_ЛЮДИ"."ФАМИЛИЯ" выполняется до соединения таблиц, что уменьшает объем данных для обработки.

### План после добавления индексов:

- Вместо полного сканирования таблиц будет использоваться индексный доступ к данным.
- Соединение будет работать быстрее благодаря индексу на атрибутах "Н\_СЕССИЯ.ЧЛВК\_ИД" и "Н\_ЛЮДИ.ФАМИЛИЯ".
- Оптимизация производительности запроса за счет использования индексов.

### QUERY PLAN

-----  
Nested Loop (cost=0.28..126.21 rows=1 width=14) (actual time=0.424..0.425 rows=0 loops=1)

```

-> Seq Scan on "Н_СЕССИЯ" (cost=0.00..117.90 rows=1 width=14) (actual
time=0.009..0.417 rows=1 loops=1)
  Filter: ("ИД" = 1975)
  Rows Removed by Filter: 3751
-> Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1
width=4) (actual time=0.004..0.004 rows=0 loops=1)
  Index Cond: ("ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД")
  Filter: (("ФАМИЛИЯ")::text < 'Иванов')::text)
Planning Time: 1.202 ms
Execution Time: 0.484 ms

```

#### 4. Оптимизация 2 запроса

```

SELECT Н_ЛЮДИ.ОТЧЕСТВО, Н_ОБУЧЕНИЯ.НЗК, Н_УЧЕНИКИ.ГРУППА FROM Н_ЛЮДИ
LEFT JOIN Н_ОБУЧЕНИЯ ON Н_ОБУЧЕНИЯ.ЧЛВК_ИД = Н_ЛЮДИ.ИД
LEFT JOIN Н_УЧЕНИКИ ON Н_УЧЕНИКИ.ЧЛВК_ИД = Н_ЛЮДИ.ИД
WHERE Н_ЛЮДИ.ИМЯ = 'Ярослав' AND Н_ОБУЧЕНИЯ.ЧЛВК_ИД = 112514 AND
Н_УЧЕНИКИ.ИД = 150308;

```

1) Индекс на атрибуте Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД ускорит соединение таблиц Н\_ЛЮДИ и Н\_ОБУЧЕНИЯ

```
CREATE INDEX obuch_people_id_index ON Н_ОБУЧЕНИЯ(ЧЛВК_ИД);
```

2) Индекс на атрибуте Н\_УЧЕНИКИ.ЧЛВК\_ИД ускорит соединение таблиц Н\_ЛЮДИ и Н\_УЧЕНИКИ

```
CREATE INDEX students_people_id_index ON Н_УЧЕНИКИ(ЧЛВК_ИД);
```

3) Индекс на атрибуте Н\_ЛЮДИ.ИМЯ ускорит фильтрацию Н\_ЛЮДИ.ИМЯ = 'Ярослав'

```
CREATE INDEX people_name_index ON Н_ЛЮДИ USING hash(ИМЯ);
```

Возможный план выполнения запроса при отсутствии индексов:

1)

- Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию Н\_ЛЮДИ.ИМЯ = 'Ярослав'

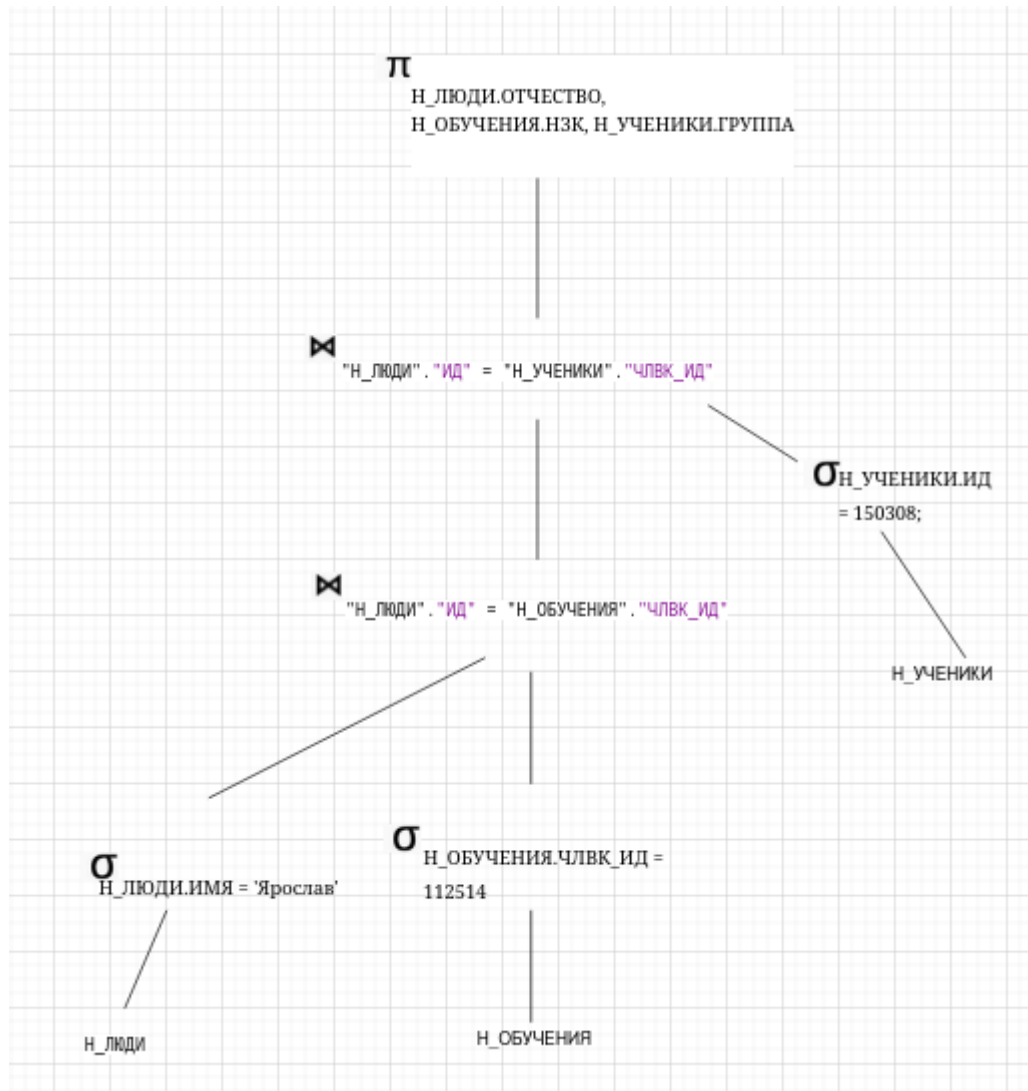
- Полный скан таблицы Н\_ОБУЧЕНИЯ с применением фильтра по условию Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 112514

- Соединение таблиц по атрибуту ЧЛВК\_ИД

- Полный скан таблицы Н\_УЧЕНИКИ с применением фильтра по условию Н\_УЧЕНИКИ.ИД = 150308;

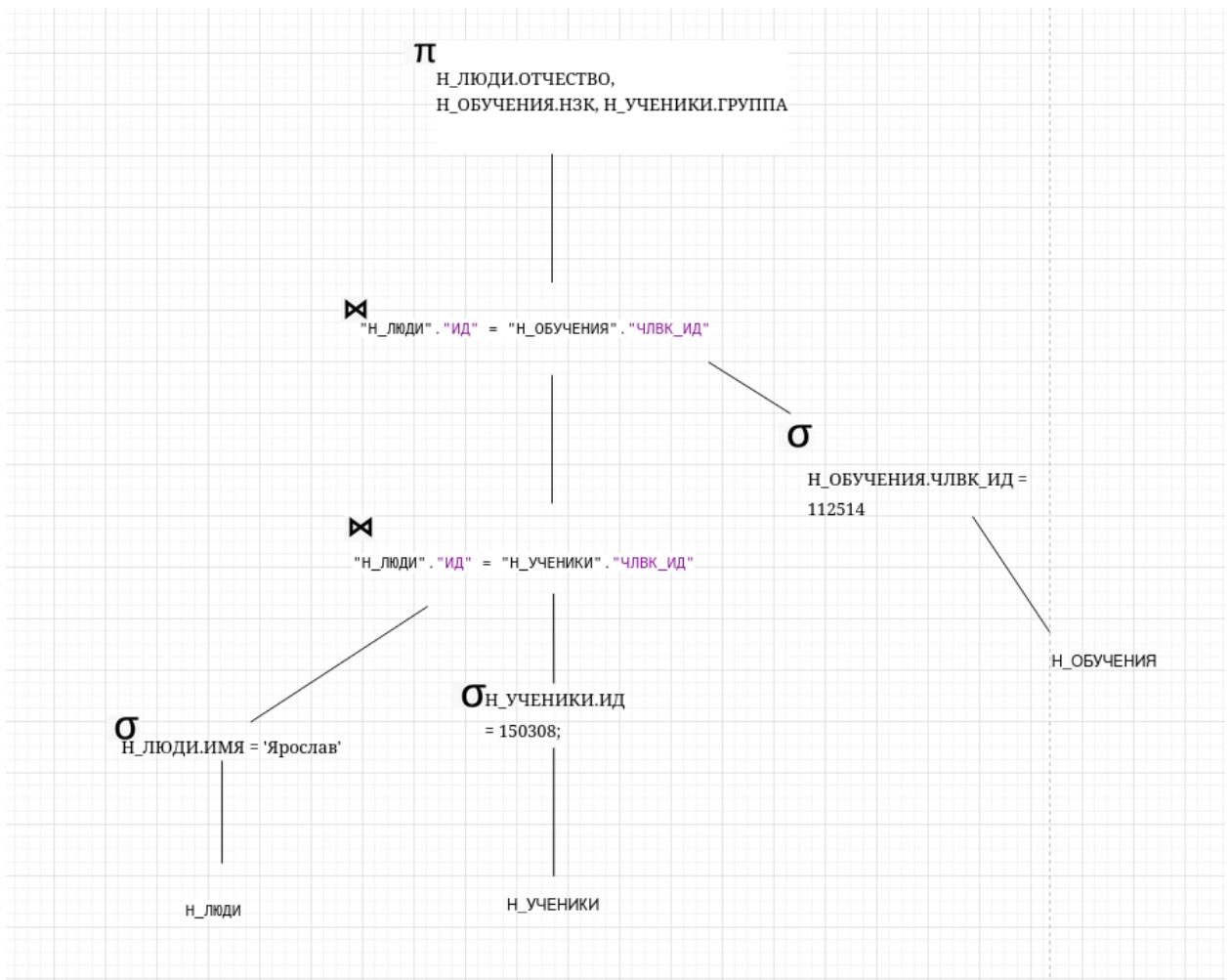
- Соединение таблиц по атрибуту ЧЛВК\_ИД

- Проекция по атрибутам Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.ГРУППА



2)

- Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию  $\text{Н\_ЛЮДИ.ИМЯ} = \text{'Ярослав'}$
- Полный скан таблицы Н\_УЧЕНИКИ с применением фильтра по условию  $\text{Н\_УЧЕНИКИ.ИД} = 150308$ ;
- Соединение таблиц по атрибуту члвк\_ид
- Полный скан таблицы Н\_ОБУЧЕНИЯ с применением фильтра по условию  $\text{Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД} = 112514$
- Соединение таблиц по атрибуту члвк\_ид
- Проекция по атрибутам  $\text{Н\_ЛЮДИ.ОТЧЕСТВО}$ ,  $\text{Н\_ОБУЧЕНИЯ.НЗК}$ ,  $\text{Н\_УЧЕНИКИ.ГРУППА}$



Оптимальный план:

- План 2, так как фильтрация N\_УЧЕНИКИ.ИД выполняется до первого соединения таблиц, что уменьшает время выполнения запроса (поскольку первичные ключи неявно индексируются, из-за чего эта фильтрация выполняется быстрее)

При добавлении индексов на "N\_ОБУЧЕНИЯ"("ЧЛВК\_ИД"), "N\_ЛЮДИ"("ИМЯ") и "N\_УЧЕНИКИ"("ЧЛВК\_ИД") планы выполнения запросов могут измениться значительно:

- Используя эти индексы, можно сделать индексный скан вместо полного сканирования таблиц.
- Использование Nested Loops Join будет более эффективным благодаря наличию индекса на атрибуте "ЧЛВК\_ИД".

```
Nested Loop (cost=0.85..24.93 rows=1 width=30) (actual
time=0.031..0.032 rows=0 loops=1)
-> Nested Loop (cost=0.56..16.61 rows=1 width=30) (actual
time=0.031..0.031 rows=0 loops=1)
-> Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..8.30
rows=1 width=24) (actual time=0.030..0.030 rows=0 loops=1)
Index Cond: ("ИД" = 112514)
Filter: (("ИМЯ")::text = 'Ярослав'::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" = 112514)
-> Index Scan using "УЧЕН_РК" on "Н_УЧЕНИКИ" (cost=0.29..8.31
rows=1 width=8) (never executed)
Index Cond: ("ИД" = 150308)
Filter: ("ЧЛВК_ИД" = 112514)
Planning Time: 0.959 ms
Execution Time: 0.086 ms
```

## Вывод

Выполняя данную лабораторную работу, я познакомился с индексами, оптимизацией запросов, обозначениями реляционной алгебры, научился строить планы выполнения запросов.