

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

Факультет программной инженерии и компьютерной техники

Лабораторная работа по дисциплине «Базы данных» №4

Вариант 1403

Выполнил: Баукин Максим Александрович

Группа: Р3132

Принимающий: Афанасьев Дмитрий Борисович

Должность: Преподаватель факультета ПИиКТ

Г. Санкт-Петербург, 2024 г.

## Оглавление

Текст задания.....	3
Запрос №1 .....	3
Индексы, добавление которых уменьшит время выполнения запроса 1: .....	4
Таблица Н_ТИПЫ_ВЕДОМОСТЕЙ: .....	4
Таблица Н_ВЕДОМОСТИ:.....	4
Таблица Н_ВЕДОМОСТИ:.....	4
QUERY PLAN 1 .....	5
QUERY PLAN 2 .....	5
QUERY PLAN 3 .....	6
EXPLAIN ANALYZE_1.....	6
Запрос №2 .....	7
Индексы, добавление которых уменьшит время выполнения запроса 2: .....	8
Таблица Н_ОБУЧЕНИЯ:.....	8
Таблица Н_ЛЮДИ:.....	8
Таблица Н_УЧЕНИКИ: .....	8
QUERY PLAN 1 .....	9
QUERY PLAN 2 .....	10
EXPLAIN ANALYZE_2.....	11
Вывод.....	11

## 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. Изменяются 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).

## Запрос №1

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
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):  
а) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Экзаменационный лист.  
б) Н\_ВЕДОМОСТИ.ИД > 1250972;  
Вид соединения: INNER JOIN.

```
SELECT Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н_ВЕДОМОСТИ.ИД
FROM Н_ТИПЫ_ВЕДОМОСТЕЙ INNER JOIN Н_ВЕДОМОСТИ
ON Н_ВЕДОМОСТИ.ТВ_ИД = Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД
WHERE Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Экзаменационный лист'
AND Н_ВЕДОМОСТИ.ИД > 1250972
```

## **Индексы, добавление которых уменьшит время выполнения запроса 1:**

### **Таблица Н\_ТИПЫ\_ВЕДОМОСТЕЙ:**

Hash-index на атрибуте НАИМЕНОВАНИЕ оптимален по причине точечного запроса

Hash индекс на атрибуте ИД уменьшит время выполнения запроса потому что обращаться к hash-таблице быстрее и hash-join работает быстрее nested-loop join

Индексы GiST, SP\_GiST, GIN, BRIN не могут быть использованы из-за неподходящих условий выборки

### **Таблица Н\_ВЕДОМОСТИ:**

Индекс b-tree на атрибуте Н\_ВЕДОМОСТИ.ИД уменьшит время выполнения запроса, потому что

Индекс типа b-tree хорошо подходит для операций выборок определенных значений, а также поскольку таблица имеет сокращает количество строк после фильтрации

Индексы GiST, SP\_GiST, GIN, BRIN не могут быть использованы из-за неподходящих условий выборки

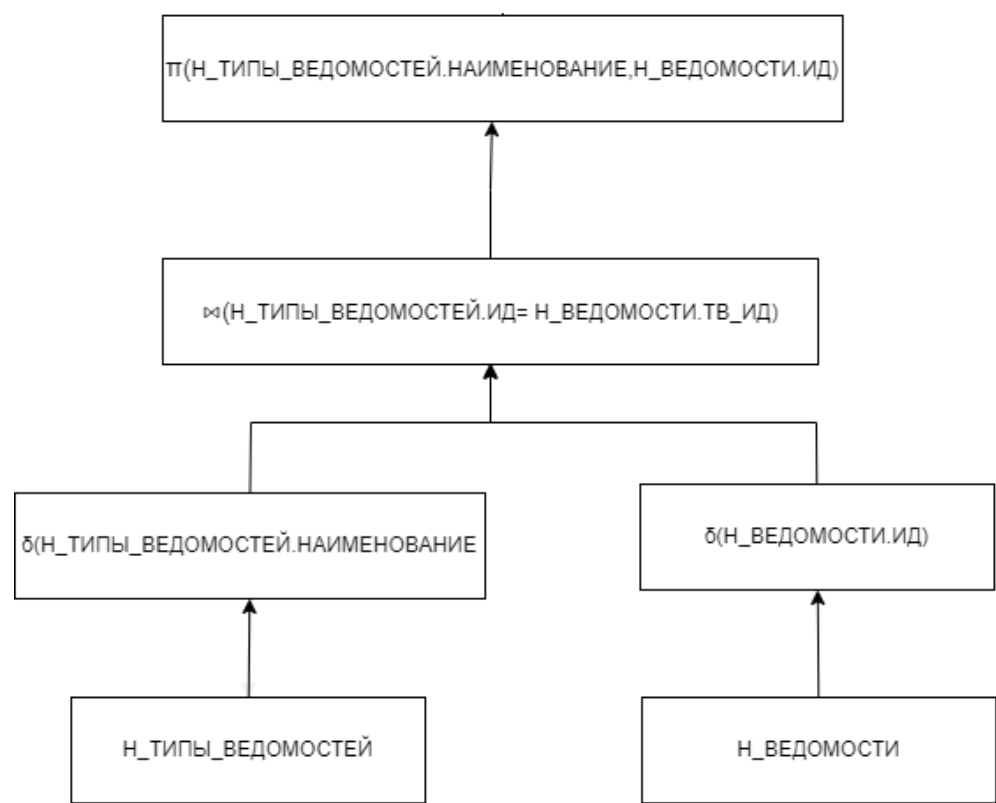
### **Таблица Н\_ВЕДОМОСТИ:**

```
CREATE INDEX "ИНД_ТВ_НАИМЕНОВАНИЕ" ON Н_ТИПЫ_ВЕДОМОСТЕЙ USING  
HASH(НАИМЕНОВАНИЕ);
```

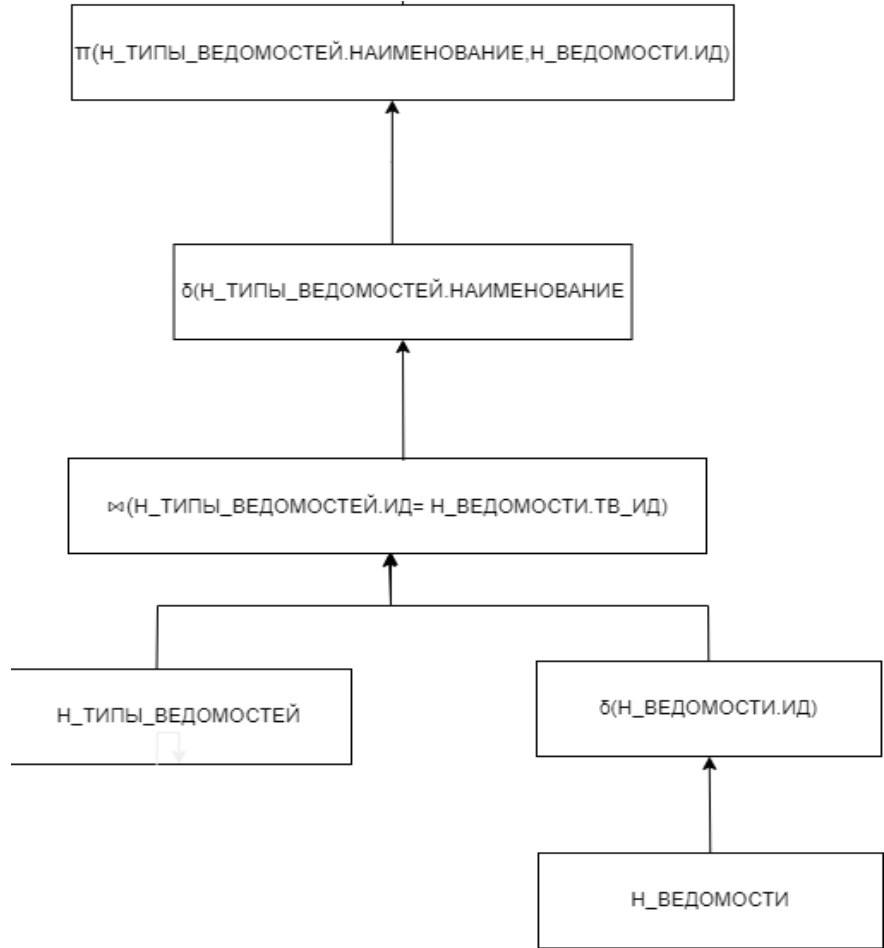
```
CREATE INDEX "ИНД_ВЕД_ИД" ON Н_ВЕДОМОСТИ USING BTREE(ИД);
```

```
CREATE INDEX "Н_ТВ_ИД" ON "Н_ТИПЫ_ВЕДОМОСТЕЙ" USING HASH(ИД);
```

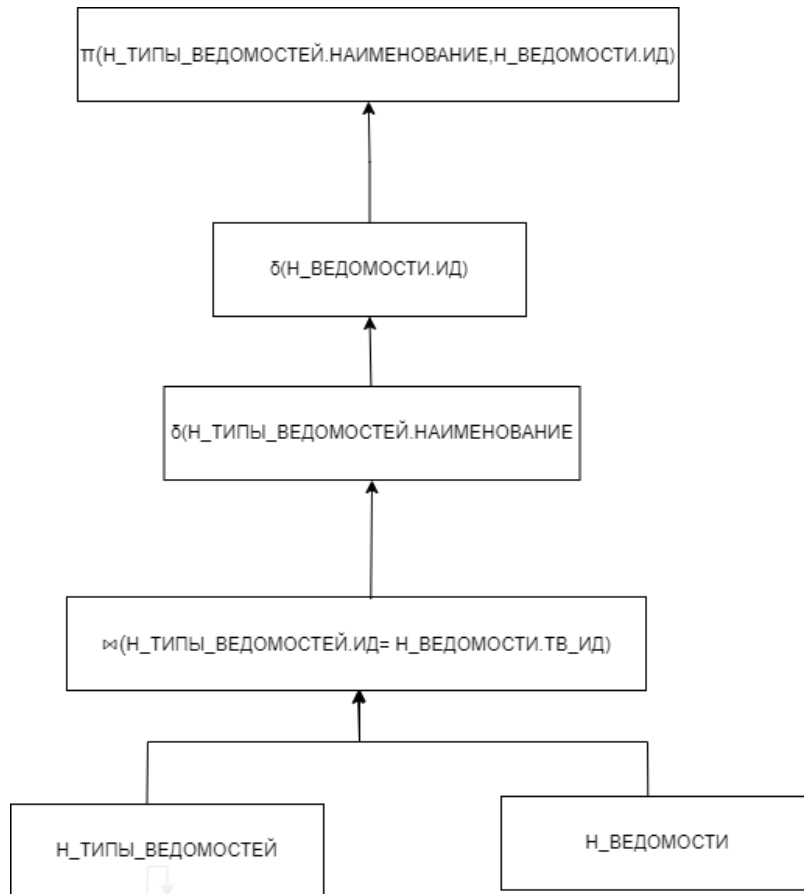
QUERY PLAN 1



QUERY PLAN 2



## QUERY PLAN 3



Оптимальный план:

- План 1, потому что фильтрация данных выполняется до соединения таблиц, что уменьшает количество строк для обработки.

При добавлении индексов планы выполнения запросов изменятся:

- Вместо полного скана таблиц будет использоваться индексный скан.
- Вместо Nested Loop Join использует Hash join .

## EXPLAIN ANALYZE\_1

Hash Join (cost=498.02..5099.11 rows=8798 width=422) (actual time=1.997..9.782 rows=3521 loops=1)

Hash Cond: ("H\_ВЕДОМОСТИ"."ТВ\_ИД" = "H\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД")

-> Bitmap Heap Scan on "H\_ВЕДОМОСТИ" (cost=496.97..4930.90 rows=26394 width=8) (actual time=1.931..5.717 rows=26134 loops=1)

Recheck Cond: ("ИД" > 1250972)

Heap Blocks: exact=642

-> Bitmap Index Scan on "ИНД\_ВЕД\_ИД" (cost=0.00..490.38 rows=26394 width=0) (actual time=1.839..1.840 rows=26134 loops=1)

Index Cond: ("ИД" > 1250972)

-> Hash (cost=1.04..1.04 rows=1 width=422) (actual time=0.019..0.019 rows=1 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 9kB

-> Seq Scan on "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=422) (actual time=0.013..0.014 rows=1 loops=1)

Filter: (("НАИМЕНОВАНИЕ")::text = 'Экзаменационный лист'::text)

Rows Removed by Filter: 2

Planning Time: 1.080 ms

Execution Time: 9.993 ms

## Запрос №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.НАЧАЛО.

Фильтры: (AND)

а) Н\_ЛЮДИ.ФАМИЛИЯ > Ёлкин

б) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 163484.

с) Н\_УЧЕНИКИ.ИД < 154782.

Вид соединения: LEFT JOIN.

```
SELECT Н_ЛЮДИ.ФАМИЛИЯ, Н_ОБУЧЕНИЯ.ЧЛВК_ИД, Н_УЧЕНИКИ.НАЧАЛО
FROM Н_ОБУЧЕНИЯ
LEFT JOIN Н_ЛЮДИ ON Н_ЛЮДИ.ИД = Н_ОБУЧЕНИЯ.ЧЛВК_ИД
LEFT JOIN Н_УЧЕНИКИ ON Н_УЧЕНИКИ.ЧЛВК_ИД = Н_ОБУЧЕНИЯ.ЧЛВК_ИД
WHERE Н_ЛЮДИ.ФАМИЛИЯ > 'Ёлкин'
AND Н_ОБУЧЕНИЯ.ЧЛВК_ИД > 163484
AND Н_УЧЕНИКИ.ИД < 154782;
```

## **Индексы, добавление которых уменьшит время выполнения запроса 2:**

### **Таблица Н\_ОБУЧЕНИЯ:**

Хеш индексы этой таблице не требуются из-за использования индекса btree. Для hash join будет быстрее вызвать операцию hash, чем воспользоваться существующим индексом

Индекс b-tree на атрибуте Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД уменьшит время выполнения запроса, потому что

Индекс типа b-tree хорошо подходит для операций выборки определенных значений, а также поскольку таблица имеет сокращает количество строк после фильтрации

Индексы GiST,SP\_GiST, GIN,BRIN не могут быть использованы из-за неподходящих условий выборки

### **Таблица Н\_ЛЮДИ:**

Индекс b-tree на атрибуте Н\_ЛЮДИ.ИД уменьшит время выполнения запроса, потому что

Индекс типа b-tree хорошо подходит для операций выборки определенных значений, а также поскольку таблица имеет сокращает количество строк после фильтрации

Индексы GiST,SP\_GiST, GIN,BRIN не могут быть использованы из-за неподходящих условий выборки

### **Таблица Н\_УЧЕНИКИ:**

Индекс b-tree на атрибуте Н\_УЧЕНИКИ.ЧЛВК.ИД уменьшит время выполнения запроса, потому что

Индекс типа b-tree хорошо подходит для операций выборки определенных значений, а также поскольку таблица имеет сокращает количество строк после фильтрации

Индексы GiST,SP\_GiST, GIN,BRIN не могут быть использованы из-за неподходящих условий выборки

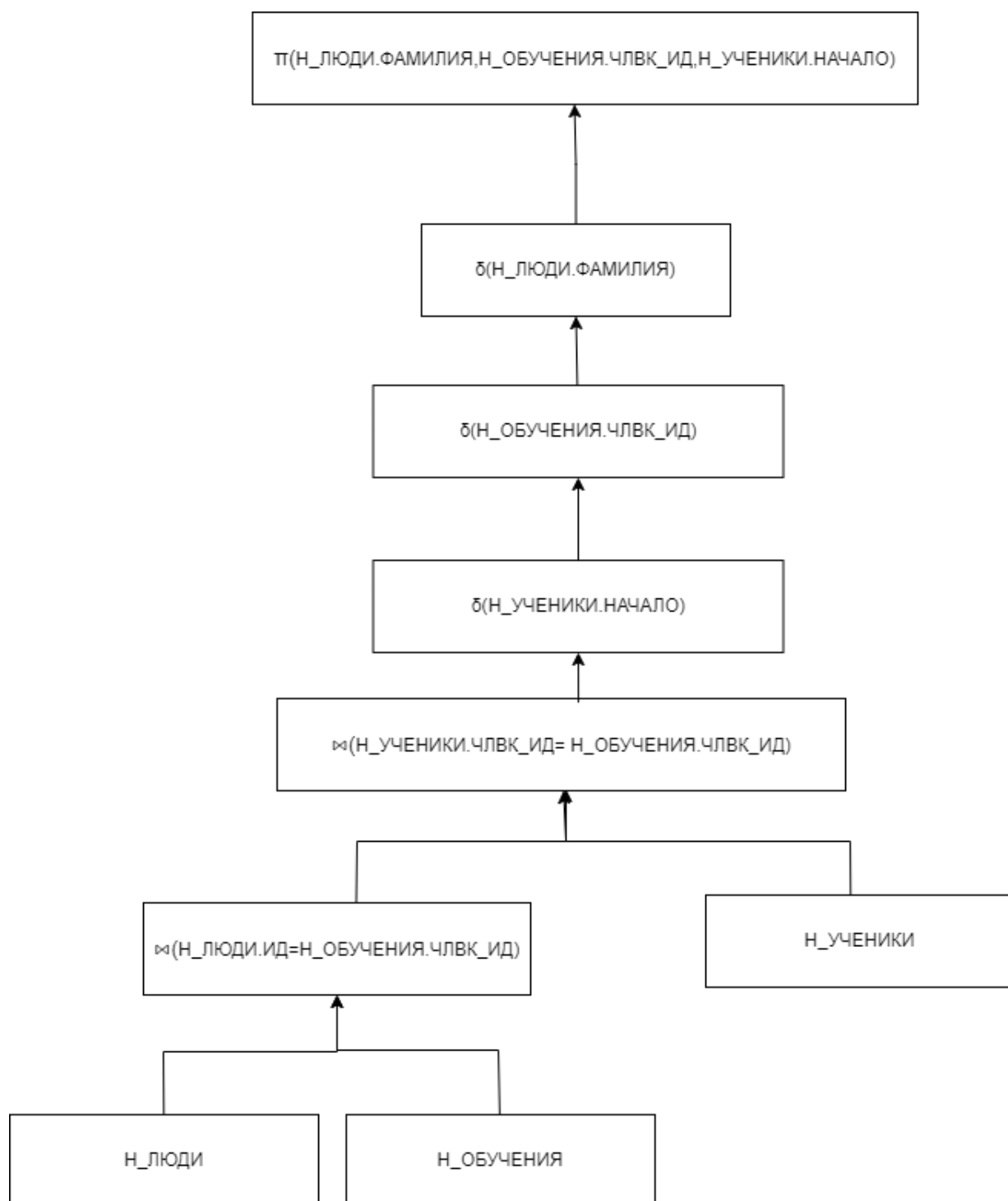
```
CREATE INDEX "ИНД_УЧ_ЧЛ_ИД" ON Н_УЧЕНИКИ USING BTREE(ЧЛВК_ИД);
```

```
CREATE INDEX "ИНД_ЛЮДИ_ИД" ON Н_ЛЮДИ USING BTREE(ИД);
```

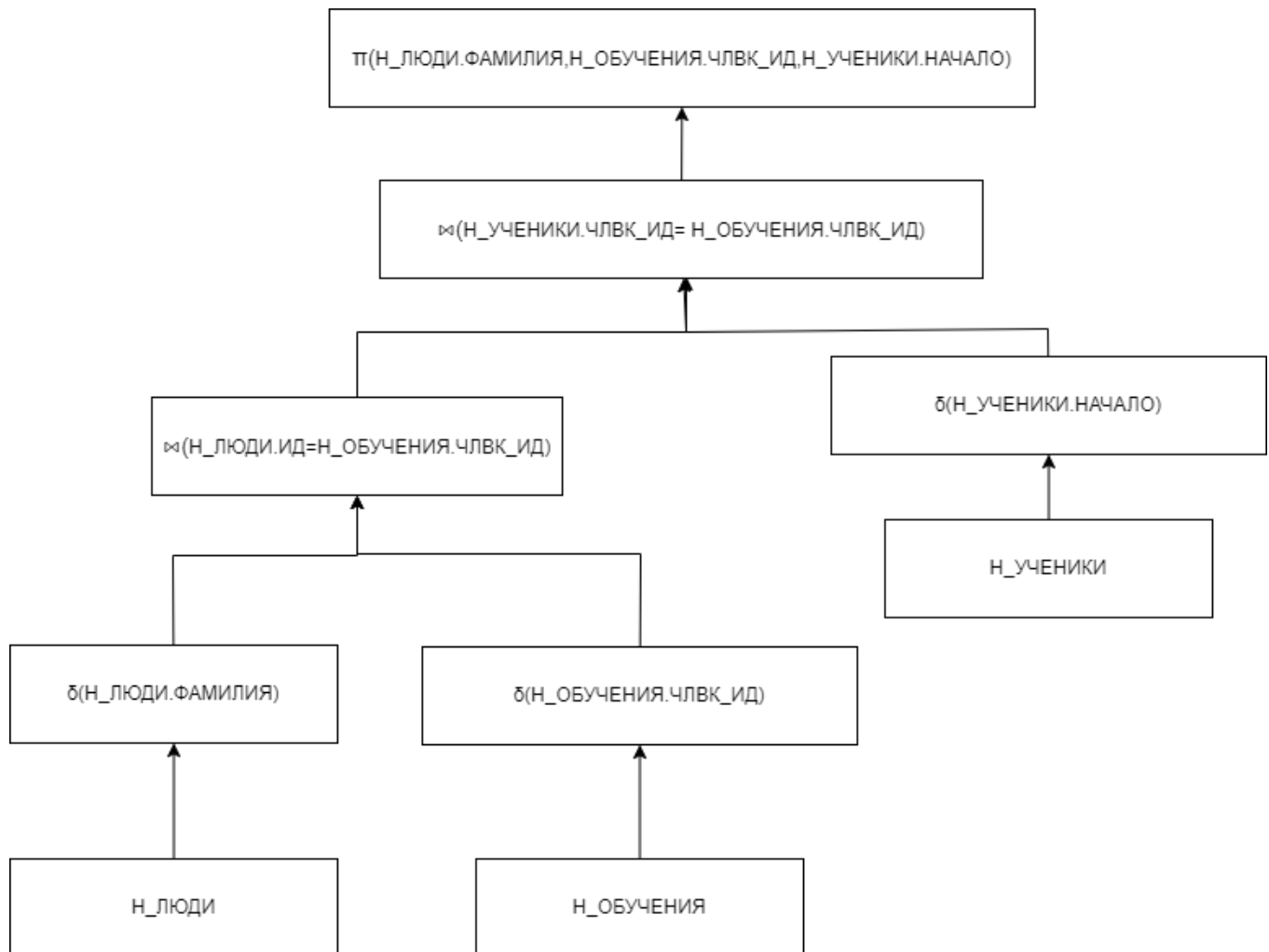
```
CREATE INDEX "ИНД_ОБУЧ_ЧЛ_ИД" ON Н_ОБУЧЕНИЯ USING BTREE(ЧЛВК_ИД);
```



## QUERY PLAN 1



## QUERY PLAN 2



Оптимальный план:

- План 2, потому что фильтрация данных выполняется до соединения таблиц, что уменьшает количество строк для обработки.

## EXPLAIN ANALYZE\_2

--Для выборки был ЧЛВК.ИД > 163484 на ЧЛВК.ИД > 143484--

Hash Join (cost=262.15..1221.64 rows=3015 width=28) (actual time=4.657..9.138 rows=289 loops=1)

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

-> Seq Scan on "Н\_УЧЕНИКИ" (cost=0.00..867.39 rows=16520 width=12) (actual time=0.013..3.458 rows=16506 loops=1)

Filter: ("ИД" < 154782)

Rows Removed by Filter: 6805

-> Hash (cost=250.48..250.48 rows=934 width=24) (actual time=3.785..3.786 rows=915 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 60kB

-> Hash Join (cost=58.32..250.48 rows=934 width=24) (actual time=0.366..3.584 rows=915 loops=1)

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

-> Seq Scan on "Н\_ЛЮДИ" (cost=0.00..163.97 rows=3768 width=20) (actual time=0.008..2.678 rows=3777 loops=1)

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

Rows Removed by Filter: 1341

-> Hash (cost=42.47..42.47 rows=1268 width=4) (actual time=0.347..0.347 rows=1269 loops=1)

Buckets: 2048 Batches: 1 Memory Usage: 61kB

-> Index Only Scan using ""ИНД\_ОБУЧ\_ЧЛ\_ИД"" on "Н\_ОБУЧЕНИЯ" (cost=0.28..42.47 rows=1268 width=4) (actual time=0.018..0.168 rows=1269 loops=1)

Index Cond: ("ЧЛВК\_ИД" > 143484)

Heap Fetches: 0

Planning Time: 0.857 ms

Execution Time: 9.207 ms

## Вывод

При выполнении лабораторной работы я познакомился с использованием индексов для ускорения обработки запросов в SQL, а также планами выполнения запросов, их построением и анализом.

