

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет ПИиКТ

Базы данных

Лабораторная работа № 4

Выполнил

студент

Набокова Алиса Владиславовна

Группа № 3120

Преподаватель: Николаев Владимир Вячеславович

г. Санкт-Петербург

2024

## Вариант: 9500

### Задание:

По варианту, выданному преподавателем, составить и выполнить запросы к базе данных "Учебный процесс".

Команда для подключения к базе данных `ucheб`:

```
psql -h pg -d ucheб
```

Отчёт по лабораторной работе должен содержать:

1. Текст задания.
2. Реализацию запросов на SQL.
3. Планы выполнения запросов.
4. Ответы на вопросы, представленные в задании.
5. Выводы по работе.

Темы для подготовки к защите лабораторной работы:

1. Индексы
2. Оптимизация запросов
3. Выбор плана выполнения запросов

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]

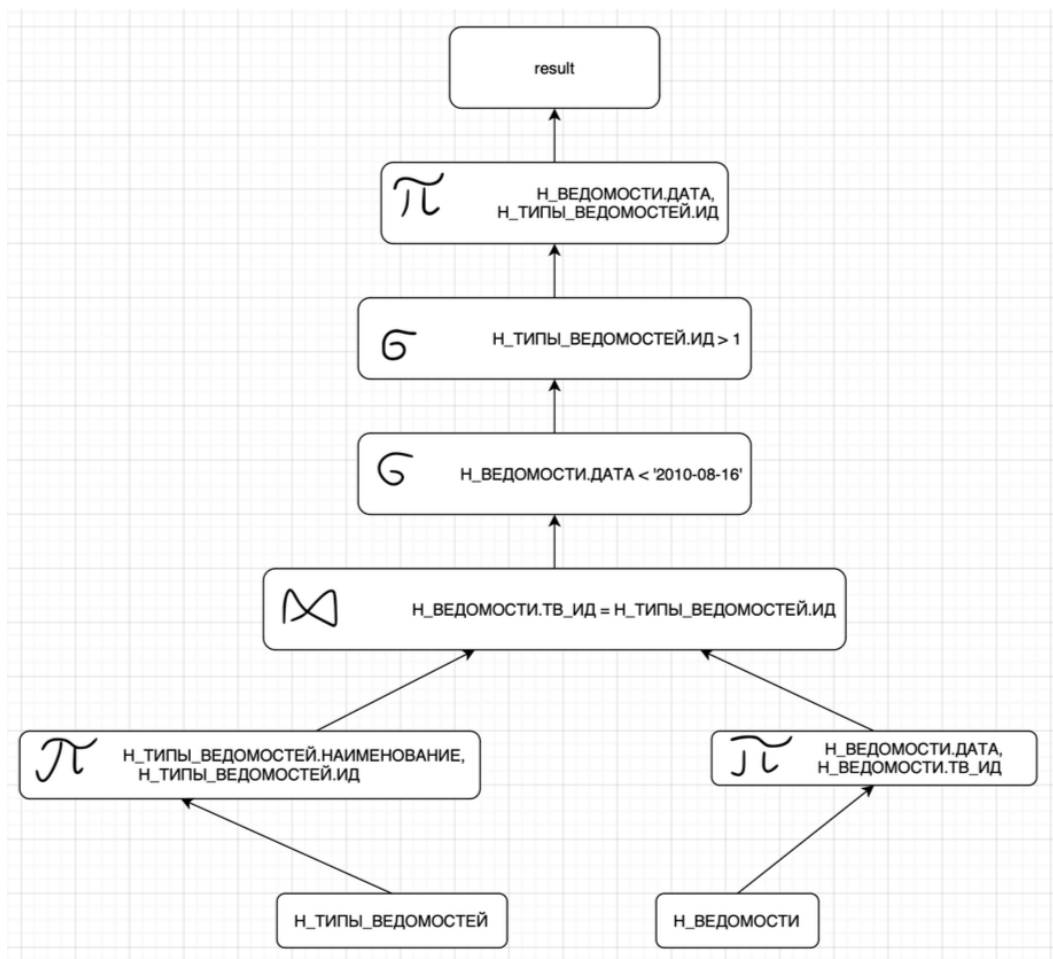
Подробные ответы на все вышеперечисленные вопросы должны присутствовать в отчете (планы выполнения запросов должны быть нарисованы, ответы на вопросы - представлены в текстовом виде).

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.  
Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ,  
Н\_ВЕДОМОСТИ.ДАТА.  
Фильтры (AND):
  - а) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД > 1.
  - б) Н\_ВЕДОМОСТИ.ДАТА < 2010-06-18.
  - с) Н\_ВЕДОМОСТИ.ДАТА < 2010-06-18.Вид соединения: LEFT 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.ИД.  
Фильтры: (AND)
  - а) Н\_ЛЮДИ.ОТЧЕСТВО > Владимирович.
  - б) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД < 113409.Вид соединения: LEFT JOIN.

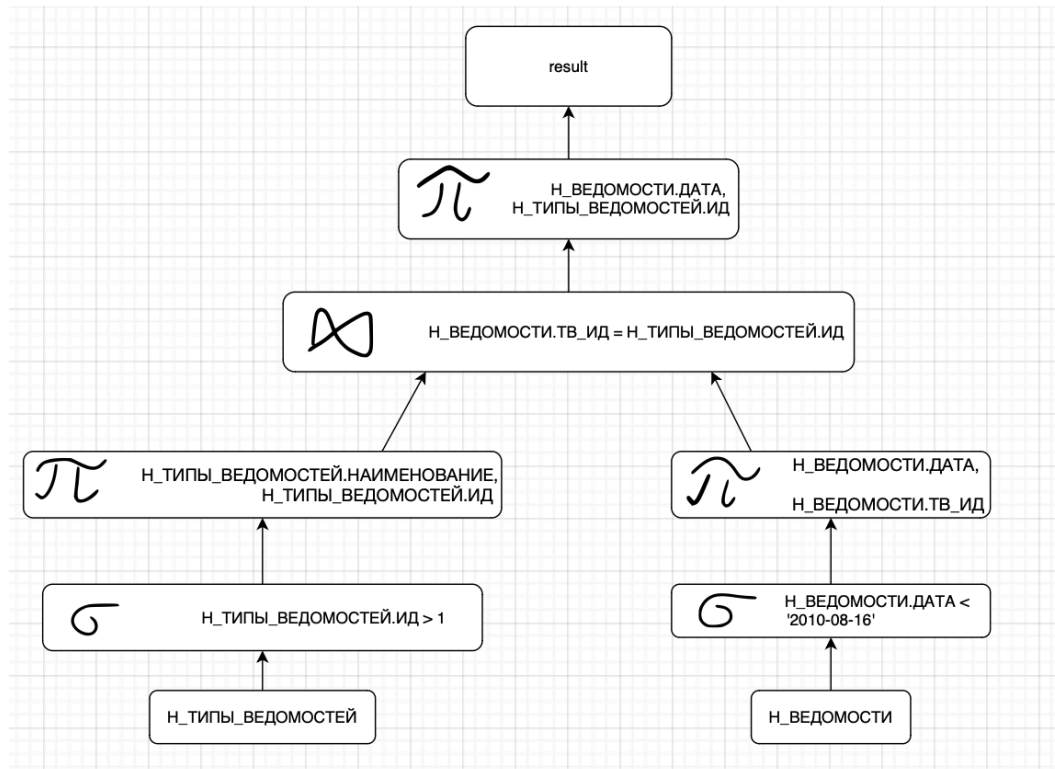
### **Запрос 1**

```
SELECT Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н_ВЕДОМОСТИ.ДАТА  
FROM Н_ТИПЫ_ВЕДОМОСТЕЙ  
LEFT JOIN Н_ВЕДОМОСТИ ON Н_ВЕДОМОСТИ.ТВ_ИД =  
Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД  
WHERE Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД > 1 AND Н_ВЕДОМОСТИ.ДАТА < '2010-06-18';
```

## Планы выполнения запроса №1



1 вариант плана



2 вариант плана

Наиболее оптимальным планом будет 2 , так как объединение происходит по отсортированным по условию атрибутам, что уменьшает количество данных при объединении и ускоряет время выполнения запроса

### Возможные индексы

Так как в запросе используется выборка по диапазону значений, то наиболее подходящий индекс в данном случае будет **B-tree**

Добавим индексы к атрибутам Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД и Н\_ВЕДОМОСТИ.ДАТА

```
CREATE INDEX index_ved_id ON Н_ВЕДОМОСТИ USING btree(ИД);
```

```
CREATE INDEX index_ved_date ON Н_ВЕДОМОСТИ USING btree(ДАТА);
```

**Выполним EXPLAIN (ANALYZE)**

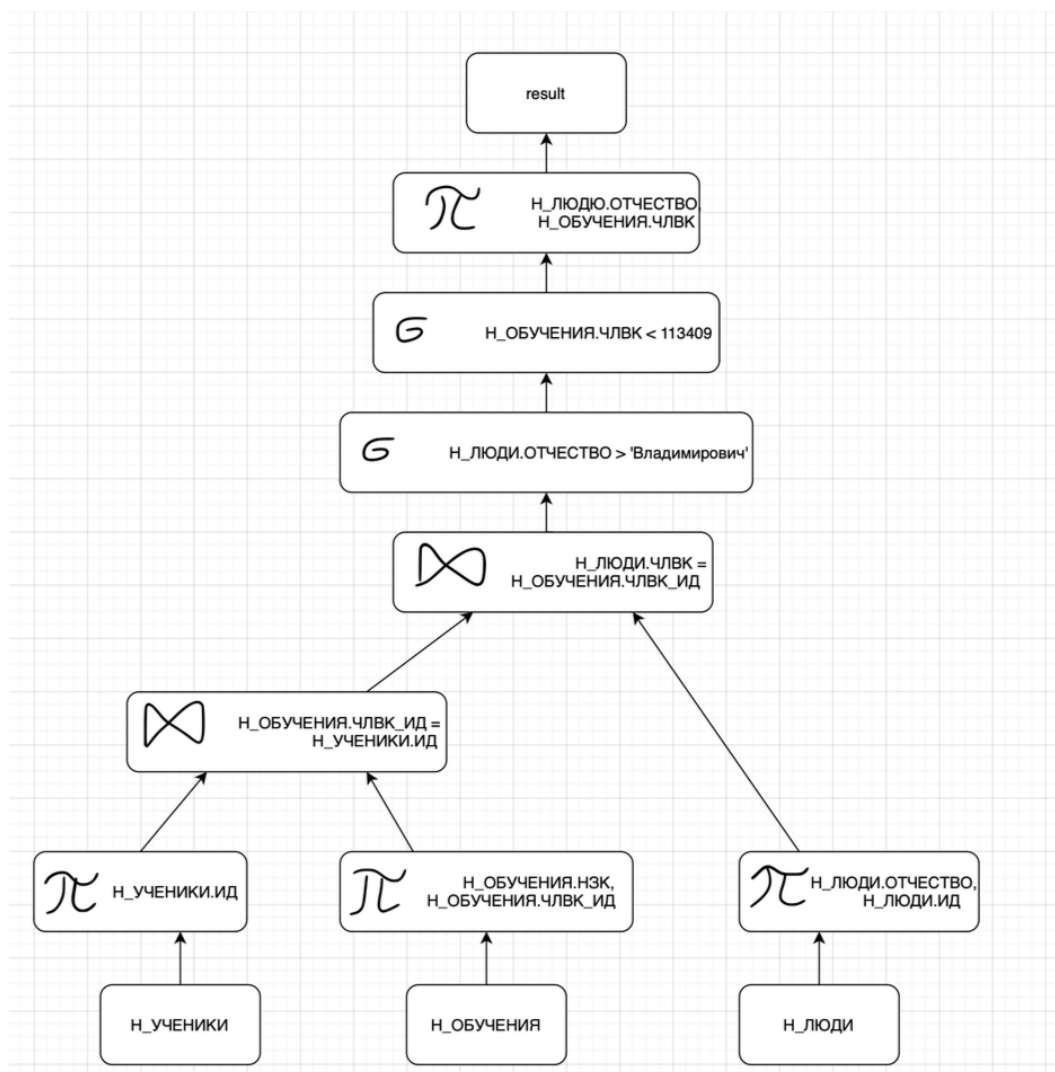
```
СТРОКА 3: FROM Н_ТИПЫ_ВЕДОМОСТЕЙ
      ^
ucheб=> EXPLAIN (ANALYZE) SELECT Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н_ВЕДОМОСТИ.ДАТА
FROM Н_ТИПЫ_ВЕДОМОСТЕЙ
LEFT JOIN Н_ВЕДОМОСТИ ON Н_ВЕДОМОСТИ.ТВ_ИД = Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД
WHERE Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД > 1 AND Н_ВЕДОМОСТИ.ДАТА < '2010-06-18';
      QUERY PLAN

-----
Hash Join  (cost=1.06..8089.97 rows=130779 width=34) (actual time=0.047..63.820 rows=24764 loops=1)
  Hash Cond: ("Н_ВЕДОМОСТИ"."ТВ_ИД" = "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД")
    -> Seq Scan on "Н_ВЕДОМОСТИ"  (cost=0.00..6846.50 rows=196169 width=12) (actual time=0.016..39.739 rows=196568 loops=1)
      Filter: ("ДАТА" < '2010-06-18 00:00:00'::timestamp without time zone)
      Rows Removed by Filter: 25872
    -> Hash  (cost=1.04..1.04 rows=2 width=30) (actual time=0.012..0.013 rows=2 loops=1)
      Buckets: 1024  Batches: 1  Memory Usage: 9kB
      -> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ"  (cost=0.00..1.04 rows=2 width=30) (actual time=0.004..0.004 rows=2 loops=1)
```

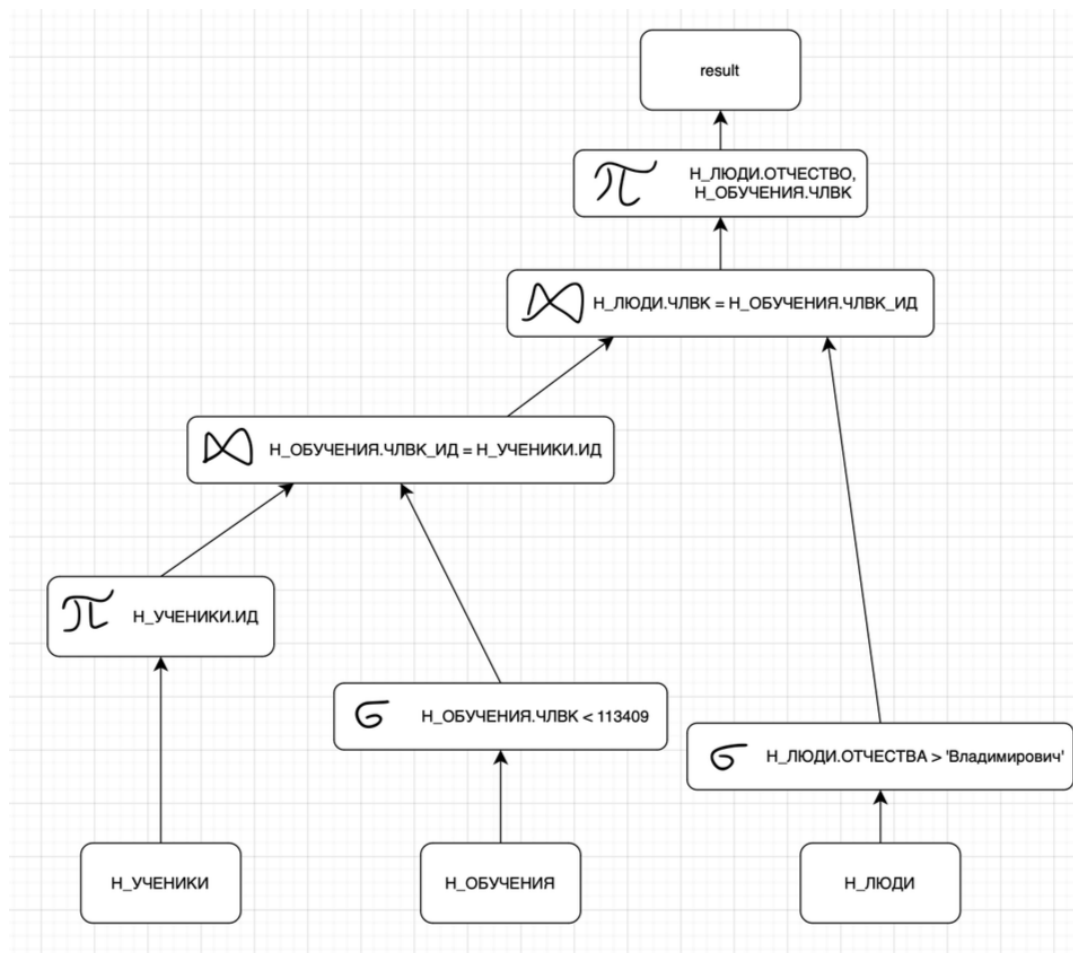
## 2 запрос

```
SELECT Н_ЛЮДИ.ОТЧЕСТВО, Н_ОБУЧЕНИЯ.НЗК, Н_УЧЕНИКИ.ИД
FROM Н_УЧЕНИКИ
LEFT JOIN Н_ОБУЧЕНИЯ ON Н_ОБУЧЕНИЯ.ЧЛВК_ИД = Н_УЧЕНИКИ.ИД
LEFT JOIN Н_ЛЮДИ ON Н_ЛЮДИ.ИД = Н_ОБУЧЕНИЯ.ЧЛВК_ИД
WHERE Н_ЛЮДИ.ОТЧЕСТВО > Владимирovich AND Н_ОБУЧЕНИЯ.ЧЛВК_ИД <
113409;
```

## Планы выполнения запроса №2

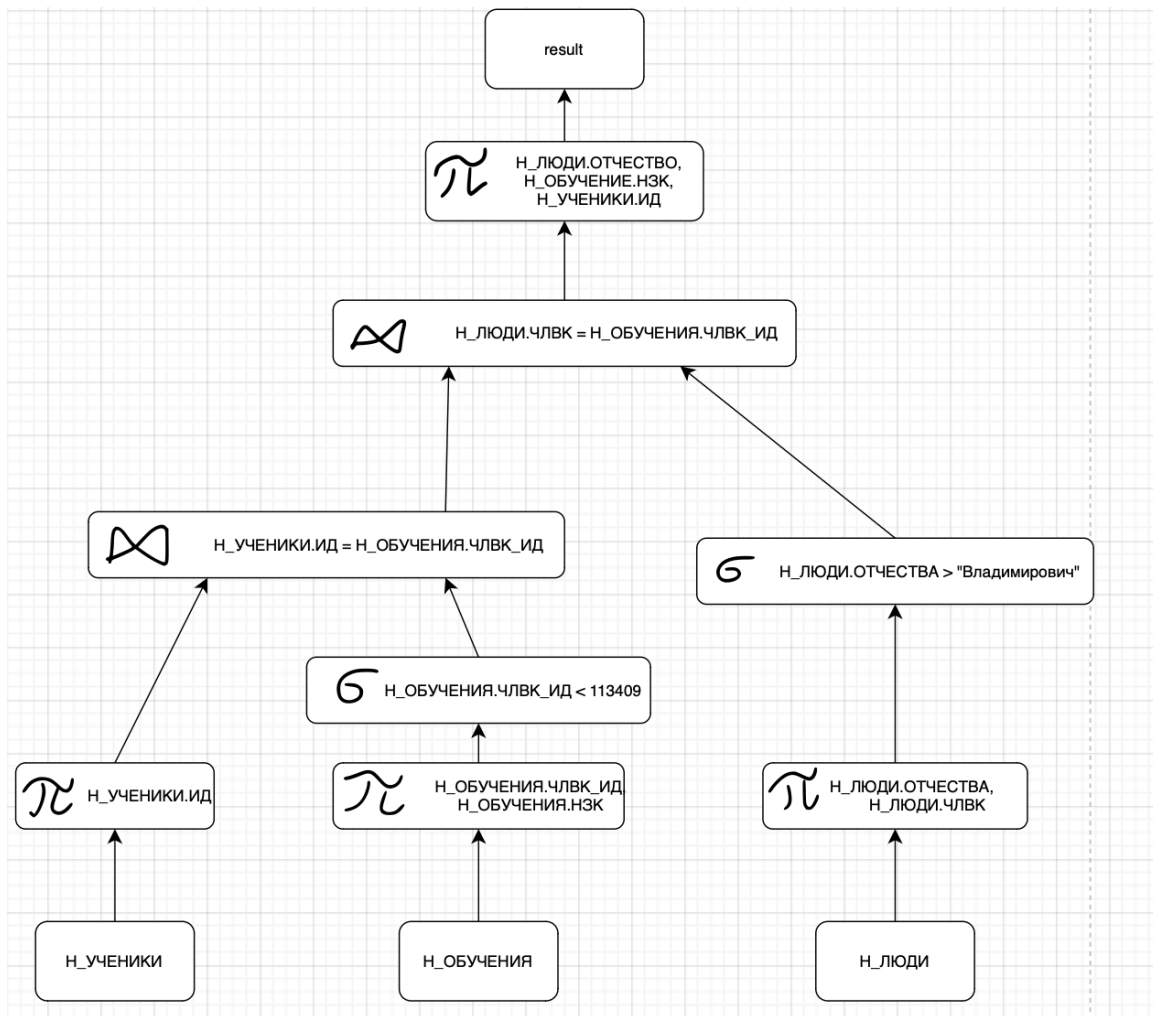


1 вариант плана



2 вариант плана





3 вариант плана

Наиболее оптимальным планом будет 3, так как объединение происходит по отсортированным по условию атрибутам, что уменьшает количество данных при объединении и ускоряет время выполнения запроса

## Возможные индексы

По тому же принципу добавим индексы btree к атрибутам Н\_ЛЮДИ.ОТЧЕСТВО и Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД

```
CREATE INDEX index_otchestvo ON Н_ЛЮДИ USING btree(ОТЧЕСТВО);
```

```
CREATE INDEX index_human_id ON Н_ОБУЧЕНИЯ USING(ЧЛВК_ИД);
```

## Выполним EXPLAIN (ANALYZE)

```
ucheb=> EXPLAIN (ANALYZE) SELECT Н_ЛЮДИ.ОТЧЕСТВО, Н_ОБУЧЕНИЯ.НЗК, Н_УЧЕНИКИ.ИД
FROM Н_УЧЕНИКИ
LEFT JOIN Н_ОБУЧЕНИЯ ON Н_ОБУЧЕНИЯ.ЧЛВК_ИД = Н_УЧЕНИКИ.ИД
LEFT JOIN Н_ЛЮДИ ON Н_ЛЮДИ.ИД = Н_ОБУЧЕНИЯ.ЧЛВК_ИД
WHERE Н_ЛЮДИ.ОТЧЕСТВО > 'Владимирович' AND Н_ОБУЧЕНИЯ.ЧЛВК_ИД < 113409;
```

QUERY PLAN

```
[ Nested Loop (cost=79.32..913.72 rows=209 width=30) (actual time=0.924..4.166 rows=4 loops=1)
  Join Filter: ("Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = "Н_УЧЕНИКИ"."ИД")
  -> Hash Join (cost=79.04..647.82 rows=209 width=34) (actual time=0.292..3.816 rows=231 loops=1)
    Hash Cond: ("Н_ЛЮДИ"."ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД")
    -> Seq Scan on "Н_ЛЮДИ" (cost=0.00..553.98 rows=2543 width=24) (actual time=0.020..3.320 rows=2544 loops=1)
      Filter: (("ОТЧЕСТВО")::text > 'Владимирович'::text)
      Rows Removed by Filter: 2574
    -> Hash (cost=73.79..73.79 rows=420 width=10) (actual time=0.182..0.183 rows=412 loops=1)
      Buckets: 1024 Batches: 1 Memory Usage: 25kB
      -> Bitmap Heap Scan on "Н_ОБУЧЕНИЯ" (cost=11.54..73.79 rows=420 width=10) (actual time=0.035..0.109 rows=412 loops=1)
        Recheck Cond: ("ЧЛВК_ИД" < 113409)
        Heap Blocks: exact=10
        -> Bitmap Index Scan on "ОБУЧ_ЧЛВК_ФК_I" (cost=0.00..11.43 rows=420 width=0) (actual time=0.024..0.025 rows=412 loops=1)
          Index Cond: ("ЧЛВК_ИД" < 113409)
      -> Index Only Scan using "УЧЕН_РК" on "Н_УЧЕНИКИ" (cost=0.29..1.26 rows=1 width=4) (actual time=0.001..0.001 rows=0 loops=231)
        Index Cond: ("ИД" = "Н_ЛЮДИ"."ИД")
        Heap Fetches: 4
    Planning Time: 0.779 ms
    Execution Time: 4.220 ms
(19 строк)
```