

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 - Программная инженерия

Дисциплина - «Базы данных»

Лабораторная работа №4

Вариант – 32432

Студент – Мухсинов
Сардорбек Пулатович

Преподаватель –

Инячина Д.А

Группа - Р3117

Санкт-Петербург, 2023г

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.

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:
Таблицы: Н_ЛЮДИ, Н_СЕССИЯ.
Вывести атрибуты: Н_ЛЮДИ.ИМЯ, Н_СЕССИЯ.ИД.
Фильтры (AND):
а) Н_ЛЮДИ.ФАМИЛИЯ = Соколов.
б) Н_СЕССИЯ.ЧЛВК_ИД > 106059.
с) Н_СЕССИЯ.ЧЛВК_ИД = 100012.
Вид соединения: RIGHT 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.ЧЛВК_ИД, Н_УЧЕНИКИ.ГРУППА.
Фильтры: (AND)
а) Н_ЛЮДИ.ИД < 100865.
б) Н_ОБУЧЕНИЯ.НЗК = 933232.
Вид соединения: RIGHT JOIN.

Первый запрос:

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

Таблицы: Н_ЛЮДИ, Н_СЕССИЯ.

Вывести атрибуты: Н_ЛЮДИ.ИМЯ, Н_СЕССИЯ.ИД.

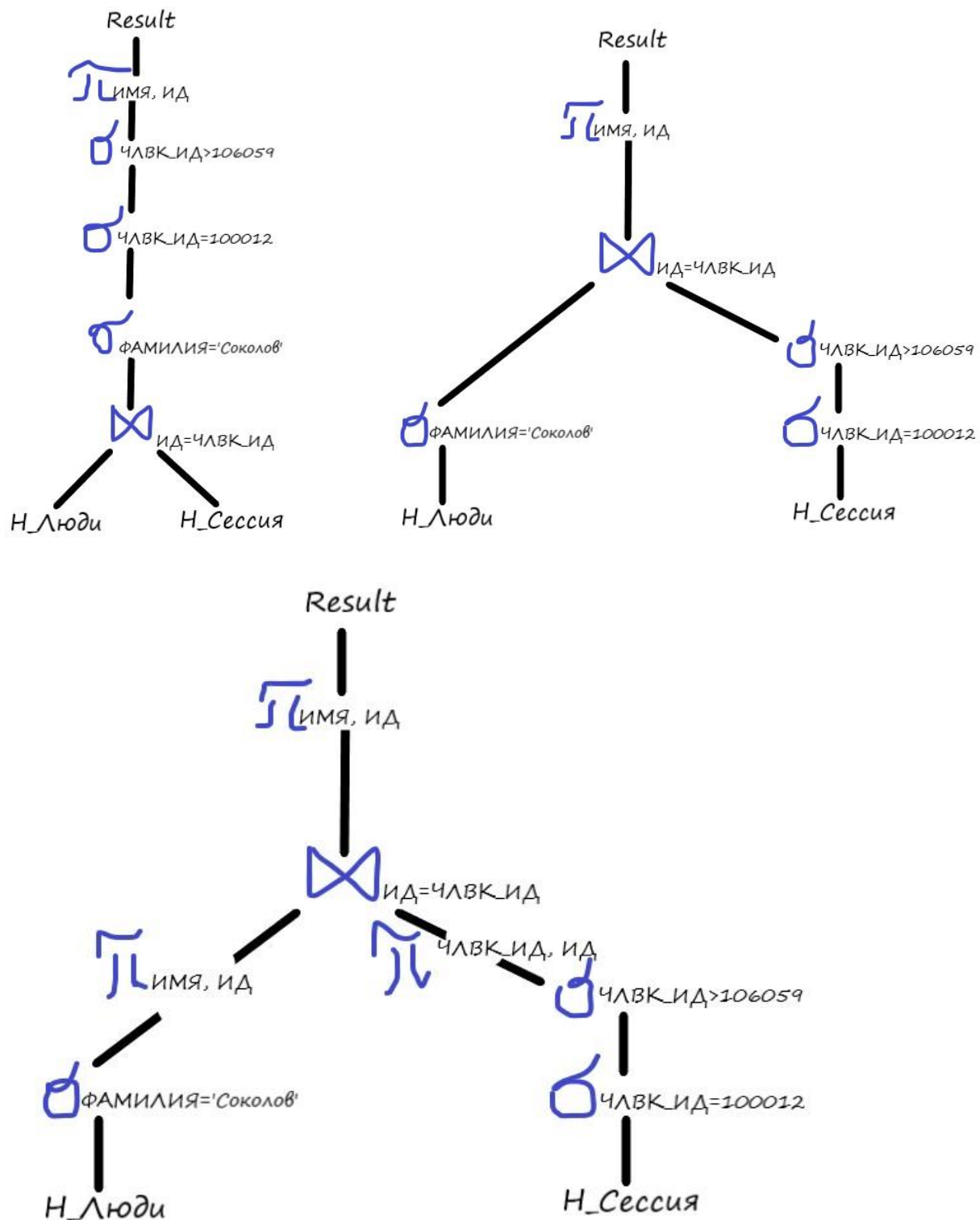
Фильтры (AND):

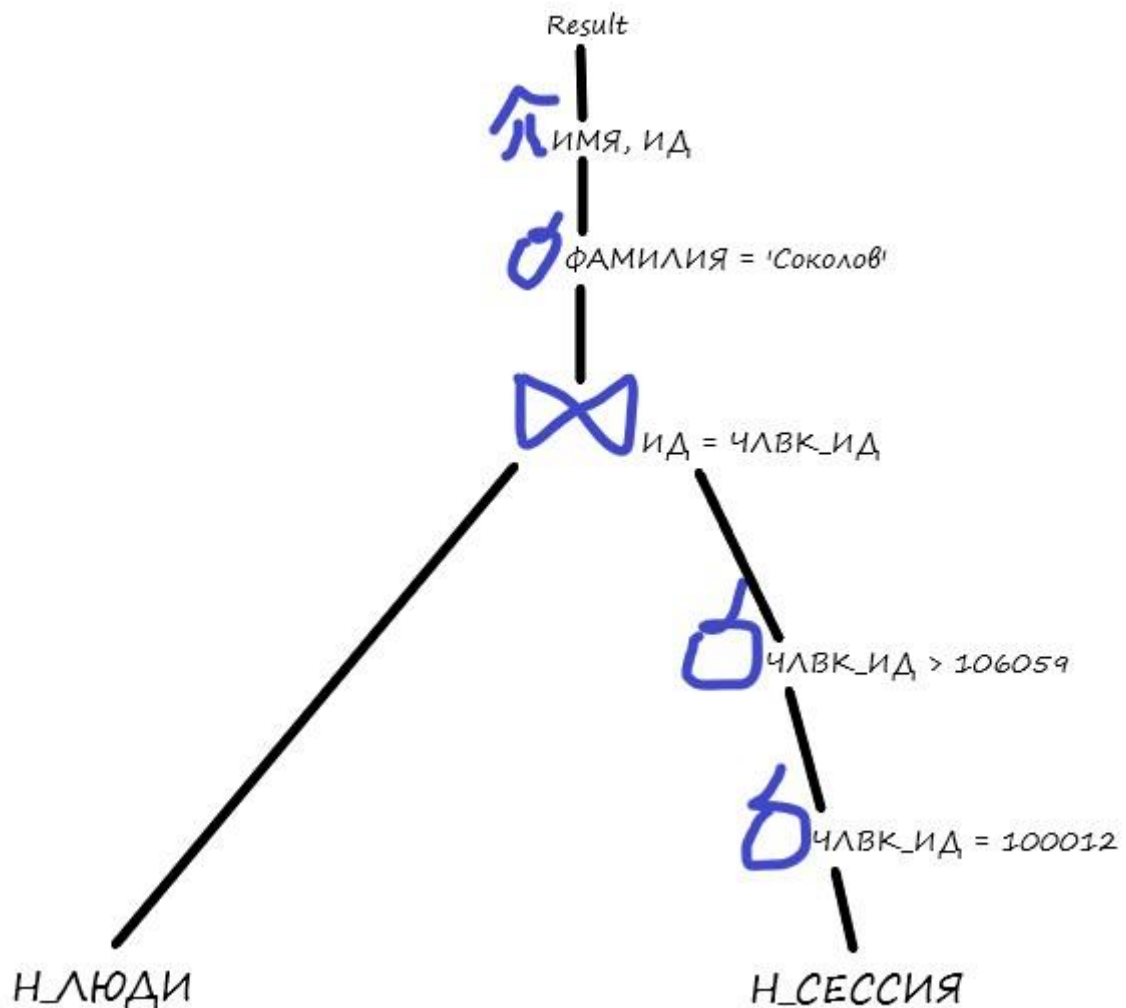
- а) Н_ЛЮДИ.ФАМИЛИЯ = Соколов.
- б) Н_СЕССИЯ.ЧЛВК_ИД > 106059.
- с) Н_СЕССИЯ.ЧЛВК_ИД = 100012.

Вид соединения: RIGHT JOIN.

```
SELECT Н_ЛЮДИ.ИМЯ, Н_СЕССИЯ.ИД
FROM Н_ЛЮДИ
      RIGHT JOIN Н_СЕССИЯ ON Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД
WHERE (Н_ЛЮДИ.ФАМИЛИЯ = 'Соколов' AND Н_СЕССИЯ.ЧЛВК_ИД = 100012 AND
Н_СЕССИЯ.ЧЛВК_ИД > 106059 );
```

Планы выполнения:





Последний план является оптимальным, так как в нём сначала выполняется выборка, следовательно, происходит объединение только нужной выборки, а не всех сущностей. Таким образом в последующих действиях будет участвовать меньшее количество кортежей, то есть размер промежуточных данных будет меньше. К тому же проекция выполняется только один раз, что уменьшает количество записей временных отношений в память. К тому же благодаря условию **ЧЛВК_ИД = 100012** дальше пройдёт только несколько записей, что ускорит выборку по условию **Н_ЛЮДИ.ФАМИЛИЯ = 'Соколов'**.

Индексы:

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

Для атрибута **Н_СЕССИЯ.ЧЛВК_ИД** целесообразно использовать Btree-index, так как выборка происходит с помощью знаков сравнения (<). Для атрибута **Н_ЛЮДИ.ИД** уже существует Btree индекс по умолчанию (как для первичного ключа).

EXPLAIN ANALYZE:

```
Nested Loop (cost=0.56..16.61 rows=1 width=17) (actual time=0.012..0.013 rows=0 loops=1)
-> Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1 width=17) (actual time=0.012..0.012 rows=0 loops=1)
    Index Cond: ("ИД" = 100012)
    Filter: (("ФАМИЛИЯ")::text = 'Соколов'::text)
    Rows Removed by Filter: 1
-> Index Scan using "SYS_C003500_IFK" on "Н_СЕССИЯ" (cost=0.28..8.30 rows=1 width=8) (never executed)
    Index Cond: (("ЧЛВК_ИД" > 106059) AND ("ЧЛВК_ИД" = 100012))
Planning Time: 0.129 ms
Execution Time: 0.036 ms
(9 строк)
```

Второй запрос:

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.ЧЛВК_ИД, Н_УЧЕНИКИ.ГРУППА.

Фильтры: (AND)

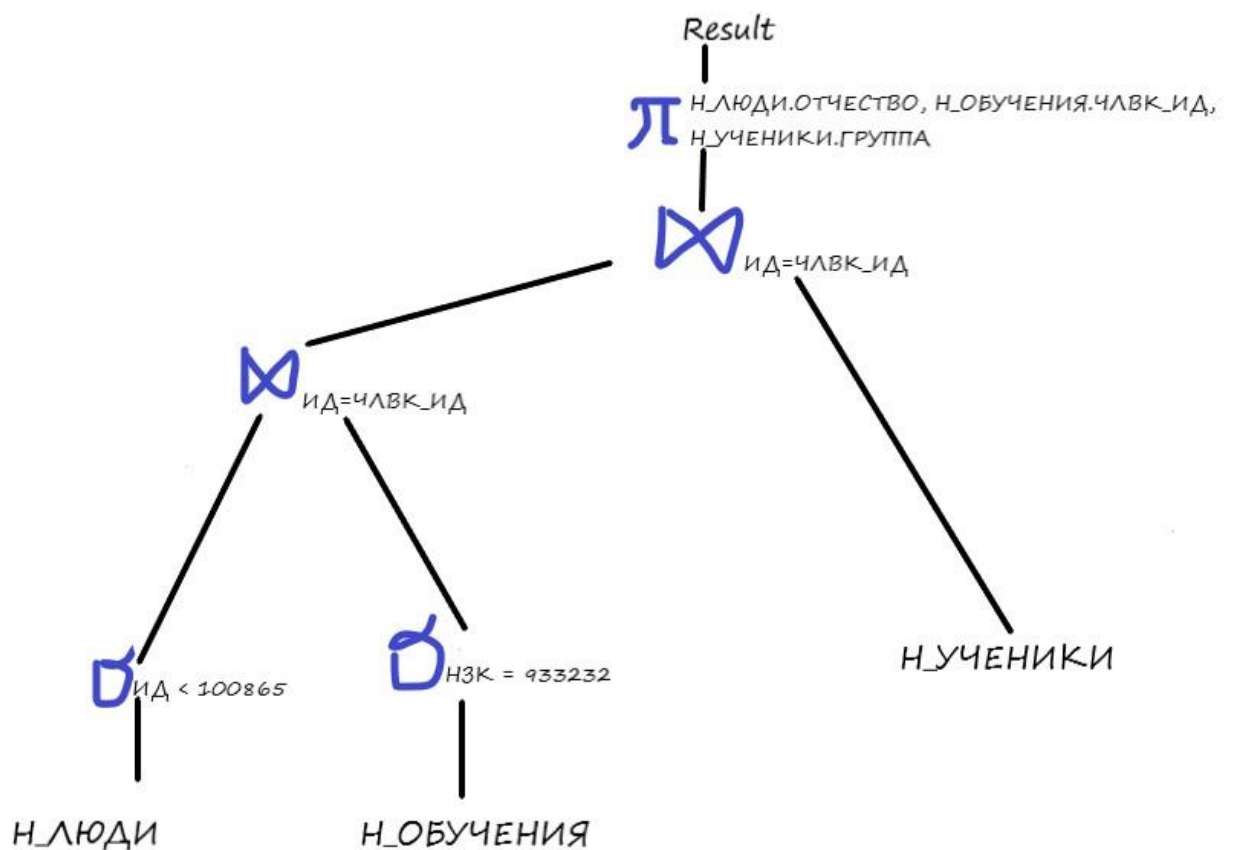
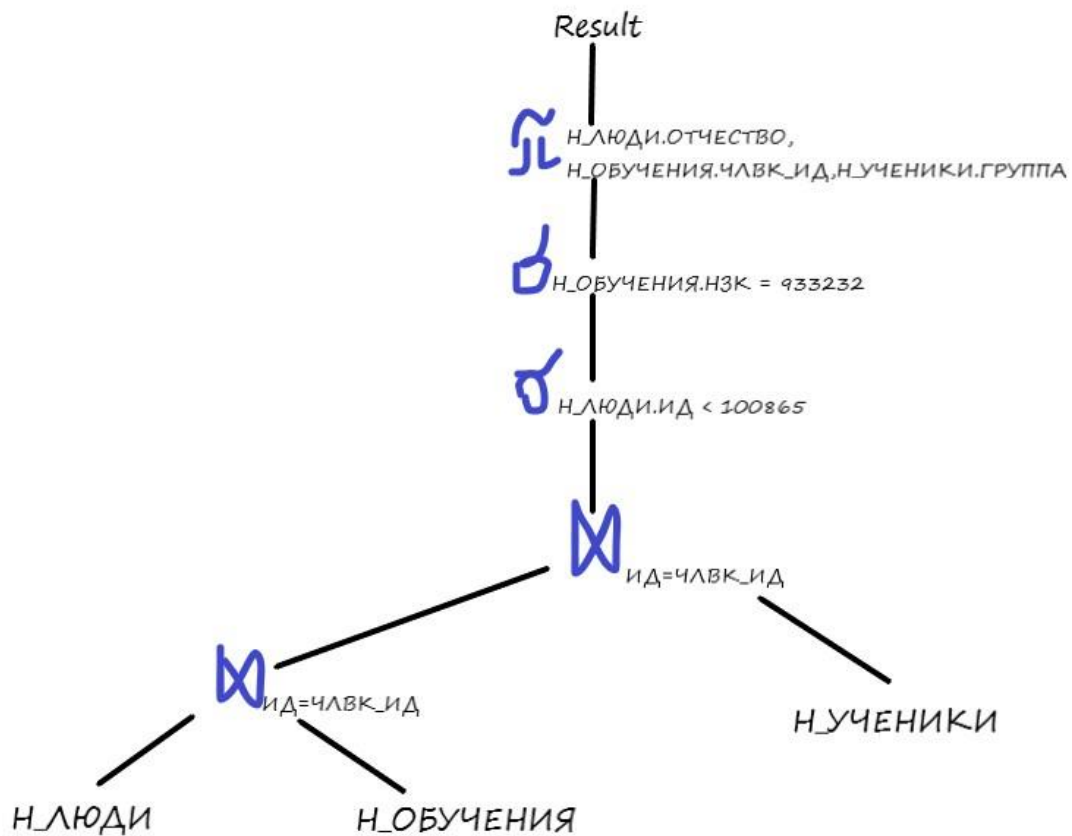
а) Н_ЛЮДИ.ИД < 100865.

б) Н_ОБУЧЕНИЯ.НЗК = 933232.

Вид соединения: RIGHT JOIN.

```
SELECT Н_ЛЮДИ.ОТЧЕСТВО, Н_ОБУЧЕНИЯ.ЧЛВК_ИД, Н_УЧЕНИКИ.ГРУППА
FROM Н_ЛЮДИ
      RIGHT JOIN Н_ОБУЧЕНИЯ ON Н_ЛЮДИ.ИД = Н_ОБУЧЕНИЯ.ЧЛВК_ИД
      RIGHT JOIN Н_УЧЕНИКИ ON Н_ЛЮДИ.ИД = Н_УЧЕНИКИ.ЧЛВК_ИД
WHERE (Н_ЛЮДИ.ИД < 100865 AND Н_ОБУЧЕНИЯ.НЗК = 933232);
```

Планы выполнения:



Последний план является оптимальным, так как в нём сначала выполняется выборка `Н_ОБУЧЕНИЯ.НЗК = '933232'` следовательно, происходит объединение только нужной выборки, а не всех сущностей. Таким образом в последующих действиях будет участвовать меньшее количество кортежей, то есть размер промежуточных данных будет меньше. К тому же проекция выполняется только один раз, что уменьшает количество записей временных отношений в память.

Также данный план является левосторонним, что позволит использовать конвейерную обработку и избежать материализации.

Если мы добавим индексы в отношения, которые участвуют в данных запросах, то вместо последовательного чтения (Seq scan) выборка будет происходить через индекс и время выполнения запроса может значительно уменьшиться. С другой стороны может увеличиться время выбора рационального плана выполнения,

Индексы:

```
CREATE INDEX ON "Н_ЛЮДИ" USING BTREE ("ИД");  
CREATE INDEX ON "Н_ОБУЧЕНИЯ" USING HASH ("НЗК");
```

Для атрибута `Н_ОБУЧЕНИЯ.НЗК` целесообразно использовать Hash-index, т.к. выборка происходит с помощью прямого сравнения. Для атрибута `Н_ЛЮДИ.ИД` уже существует Btree индекс по умолчанию (как для первичного ключа).

EXPLAIN ANALYZE:

```
Nested Loop (cost=4.61..144.43 rows=5 width=28) (actual time=0.629..0.630 rows=0 loops=1)  
  Join Filter: ("Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")  
  -> Nested Loop (cost=4.33..142.76 rows=5 width=12) (actual time=0.040..0.624 rows=3 loops=1)  
    -> Seq Scan on "Н_ОБУЧЕНИЯ" (cost=0.00..119.76 rows=1 width=4) (actual time=0.014..0.592 rows=1 loops=1)  
        Filter: (("НЗК")::text = '933232'::text)  
        Rows Removed by Filter: 5020  
    -> Bitmap Heap Scan on "Н_УЧЕНИКИ" (cost=4.33..22.95 rows=5 width=8) (actual time=0.020..0.025 rows=3 loops=1)  
        Recheck Cond: ("ЧЛВК_ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД")  
        Heap Blocks: exact=3  
        -> Bitmap Index Scan on "УЧЕН_ОБУЧ_FK_I" (cost=0.00..4.32 rows=5 width=0) (actual time=0.012..0.012 rows=3 loops=1)  
            Index Cond: ("ЧЛВК_ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД")  
    -> Index Scan using "ЧЛВК_ПК" on "Н_ЛЮДИ" (cost=0.28..0.32 rows=1 width=24) (actual time=0.001..0.001 rows=0 loops=3)  
        Index Cond: (("ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД") AND ("ИД" < 100865))  
Planning Time: 0.689 ms  
Execution Time: 0.679 ms
```

Вывод:

В ходе выполнения лабораторной работы я познакомился с индексами, которые позволяют ускорить обработку запросов в SQL, а также с планами выполнения запросов и их оптимизацией.