*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.04 – Программная инженерия,*

*Системное и прикладное программное обеспечение*

**Лабораторная работа по №4**

по дисциплине

«Базы данных»

**Вариант №312135**

Выполнил:

Ткачев Денис Владимирович

Группа P3111

Преподаватели:

Харитонова Анастасия Евгеньевна

Оглавление

[Задание 2](#_Toc196324644)

[**Реализация на PostgreSQL** 3](#_Toc196324645)

[**Уменьшение времени выполнения запроса 1.** 3](#_Toc196324646)

[**План выполнения запроса 1.** 4](#_Toc196324647)

[**Уменьшение времени выполнения запроса 2.** 5](#_Toc196324648)

[**План выполнения запроса 2.** 6](#_Toc196324649)

[**Вывод** 7](#_Toc196324650)

## 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.  
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.ФАМИЛИЯ, Н\_СЕССИЯ.УЧГОД.  
   Фильтры (AND):  
   a) Н\_ЛЮДИ.ИД < 152862.  
   b) Н\_СЕССИЯ.УЧГОД < 2011/2012.  
   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.ИМЯ, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.ГРУППА.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ОТЧЕСТВО > Александрович.  
   b) Н\_ОБУЧЕНИЯ.НЗК = 999080.  
   c) Н\_УЧЕНИКИ.ГРУППА = 3100.  
   Вид соединения: INNER JOIN.

## **Реализация на PostgreSQL**

-- 1. 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):  
-- a) Н\_ЛЮДИ.ИД < 152862.  
-- b) Н\_СЕССИЯ.УЧГОД < 2011/2012.  
-- 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"."ИД" < 152862 AND "Н\_СЕССИЯ"."УЧГОД" < '2011/2012';  
  
  
-- 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.ИМЯ, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.ГРУППА.  
-- Фильтры: (AND)  
-- a) Н\_ЛЮДИ.ОТЧЕСТВО > Александрович.  
-- b) Н\_ОБУЧЕНИЯ.НЗК = 999080.  
-- c) Н\_УЧЕНИКИ.ГРУППА = 3100.  
-- Вид соединения: INNER JOIN.  
  
SELECT "Н\_ЛЮДИ"."ИМЯ", "Н\_ОБУЧЕНИЯ"."НЗК", "Н\_УЧЕНИКИ"."ГРУППА"  
FROM "Н\_ЛЮДИ"  
INNER JOIN "Н\_ОБУЧЕНИЯ" ON "Н\_ЛЮДИ"."ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД"  
INNER JOIN "Н\_УЧЕНИКИ" ON "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД"  
WHERE "Н\_ЛЮДИ"."ОТЧЕСТВО" > 'Александрович' AND "Н\_ОБУЧЕНИЯ"."НЗК" = '999080' AND "Н\_УЧЕНИКИ"."ГРУППА" = '3100';

## **Уменьшение времени выполнения запроса 1.**

Индексы, добавление которых уменьшит время выполнения запроса:

a) На таблице Н\_ЛЮДИ:

* Индекс на атрибуте ИД (B-tree) Это ускорит фильтрацию строк с условием "Н\_ЛЮДИ.ИД < 152862.

b) На таблице Н\_СЕССИЯ:

* Индекс на атрибуте УЧГОД (B-tree) Это ускорит фильтрацию строк с условием "Н\_СЕССИЯ.УЧГОД < '2011/2012'.

Добавление индексов на указанные столбцы позволит ускорить поиск нужных записей по фильтрам в запросе.

Возможные планы выполнения запроса без индексов:

План 1:

* Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию Н\_ЛЮДИ.ИД < 152862.
* Полный скан таблицы Н\_СЕССИЯ с применением фильтра по условию

Н\_СЕССИЯ.УЧГОД < 2011/2012'.

* Соединение таблицы Н\_СЕССИЯ с таблицей Н\_ЛЮДИ с использованием Nested Loops Join по атрибуту ЧЛВК\_ИД.

План 2:

* Полный скан таблицы Н\_ЛЮДИ.
* Полный скан таблицы Н\_СЕССИЯ.
* Соединение таблицы Н\_СЕССИЯ с таблицей Н\_ЛЮДИ с использованием Nested Loops Join по атрибуту ЧЛВК\_ИД с применением фильтра Н\_ЛЮДИ.ИД < 152862 AND Н\_СЕССИЯ.УЧГОД < 2011/2012'.

Оптимальный план:

* План 1, потому что фильтрация данных из двух таблиц 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.
* Nested Loops Join станет быстрее благодаря (неявному индексу первичного ключа PRIMARY KEY) Н\_ЛЮДИ.ИД и индекса не первичного атрибута Н\_СЕССИЯ.УЧГОД.

## **План выполнения запроса 1.**

Nested Loop (cost=0.29..284.88 rows=2569 width=26) (actual time=0.028..4.181 rows=3141 loops=1)

" -> Seq Scan on ""Н\_СЕССИЯ"" (cost=0.00..117.90 rows=3543 width=14) (actual time=0.012..2.006 rows=3543 loops=1)"

" Filter: ((""УЧГОД"")::text < '2011/2012'::text)"

Rows Removed by Filter: 209

-> Memoize (cost=0.29..0.44 rows=1 width=20) (actual time=0.000..0.000 rows=1 loops=3543)

" Cache Key: ""Н\_СЕССИЯ"".""ЧЛВК\_ИД"""

Cache Mode: logical

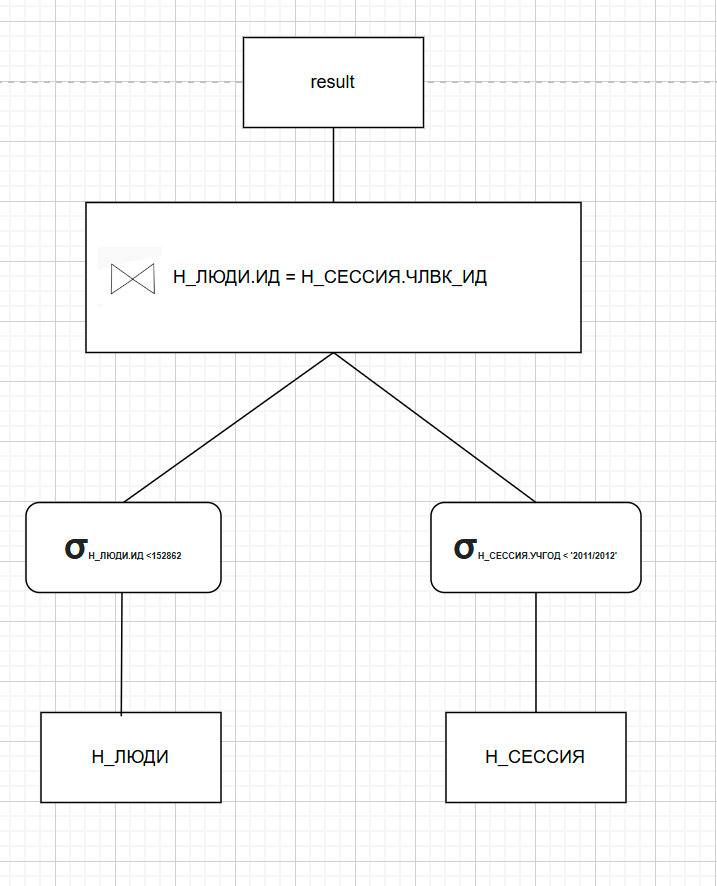
Hits: 3367 Misses: 176 Evictions: 0 Overflows: 0 Memory Usage: 21kB

" -> Index Scan using ""ЧЛВК\_PK"" on ""Н\_ЛЮДИ"" (cost=0.28..0.43 rows=1 width=20) (actual time=0.002..0.002 rows=1 loops=176)"

" Index Cond: ((""ИД"" = ""Н\_СЕССИЯ"".""ЧЛВК\_ИД"") AND (""ИД"" < 152862))"

Planning Time: 0.309 ms

Execution Time: 4.369 ms



## **Уменьшение времени выполнения запроса 2.**

Индексы, добавление которых уменьшит время выполнения запроса:

a) На таблице Н\_ЛЮДИ:

* Индекс на атрибуте ОТЧЕСТВО (B-tree). Это ускорит фильтрацию строк с условием "Н\_ЛЮДИ.ОТЧЕСТВО > 'Александрович'.

b) На таблице Н\_УЧЕНИКИ:

* Индекс на атрибуте ГРУППА (B-tree). Это ускорит фильтрацию строк с условием

"Н\_УЧЕНИКИ.ГРУППА = '3100'.

c) На таблице Н\_ОБУЧЕНИЯ:

* Индекс на атрибуте НЗК (B-tree). Это ускорит фильтрацию строк с условием "Н\_ОБУЧЕНИЯ.НЗК = '999080'.

Добавление индексов на указанные столбцы позволит ускорить поиск нужных записей по фильтрам в запросе.

Возможные планы выполнения запросов без индексов:

План 1:

* Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию "Н\_ЛЮДИ.ОТЧЕСТВО > 'Александрович'.
* Полный скан таблицы Н\_ОБУЧЕНИЯ с применением фильтра по условию Н\_ОБУЧЕНИЯ.НЗК = '999080'.
* Соединение таблицы Н\_ОБУЧЕНИЯ с таблицей Н\_ЛЮДИ с использованием Nested Loops Join по атрибуту ЧЛВК\_ИД.
* Полный скан таблицы Н\_УЧЕНИКИ с применением фильтра по условию "Н\_УЧЕНИКИ.ГРУППА = '3100'.
* Соединение результатов предыдущего этапа с таблицей Н\_УЧЕНИКИ с использованием Nested Loops Join по атрибуту ЧЛВК\_ИД.

План 2:

* Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию Н\_ЛЮДИ.ОТЧЕСТВО > 'Александрович'.
* Полный скан таблицы Н\_ОБУЧЕНИЯ с применением фильтра по условию Н\_ОБУЧЕНИЯ.НЗК = '999080'.
* Соединение таблицы Н\_ОБУЧЕНИЯ с таблицей Н\_ЛЮДИ с использованием Hash Join по атрибуту ЧЛВК\_ИД.
* Полный скан таблицы Н\_УЧЕНИКИ с применением фильтра по условию "Н\_УЧЕНИКИ.ГРУППА = '3100'.
* Соединение результатов предыдущего этапа с таблицей Н\_УЧЕНИКИ с использованием Hash Join по атрибуту ЧЛВК\_ИД.

Оптимальный план:

* План 1, так как Nested Loops Join предпочтительнее Hash Join в случаях, когда обрабатываемые наборы данных небольшие, и заранее известно, что результаты фильтрации будут содержать малое количество строк.

При добавлении индексов планы выполнения запросов изменятся:

* Вместо полного скана таблиц будет использоваться индексный скан.

## **План выполнения запроса 2.**

Nested Loop (cost=0.57..129.03 rows=1 width=23) (actual time=0.610..0.611 rows=0 loops=1)

-> Nested Loop (cost=0.28..128.07 rows=1 width=27) (actual time=0.149..0.551 rows=1 loops=1)

" -> Seq Scan on ""Н\_ОБУЧЕНИЯ"" (cost=0.00..119.76 rows=1 width=10) (actual time=0.118..0.520 rows=1 loops=1)"

" Filter: ((""НЗК"")::text = '999080'::text)"

Rows Removed by Filter: 5020

" -> Index Scan using ""ЧЛВК\_PK"" on ""Н\_ЛЮДИ"" (cost=0.28..8.30 rows=1 width=17) (actual time=0.028..0.028 rows=1 loops=1)"

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

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

" -> Index Scan using ""УЧЕН\_ОБУЧ\_FK\_I"" on ""Н\_УЧЕНИКИ"" (cost=0.29..0.95 rows=1 width=8) (actual time=0.058..0.058 rows=0 loops=1)"

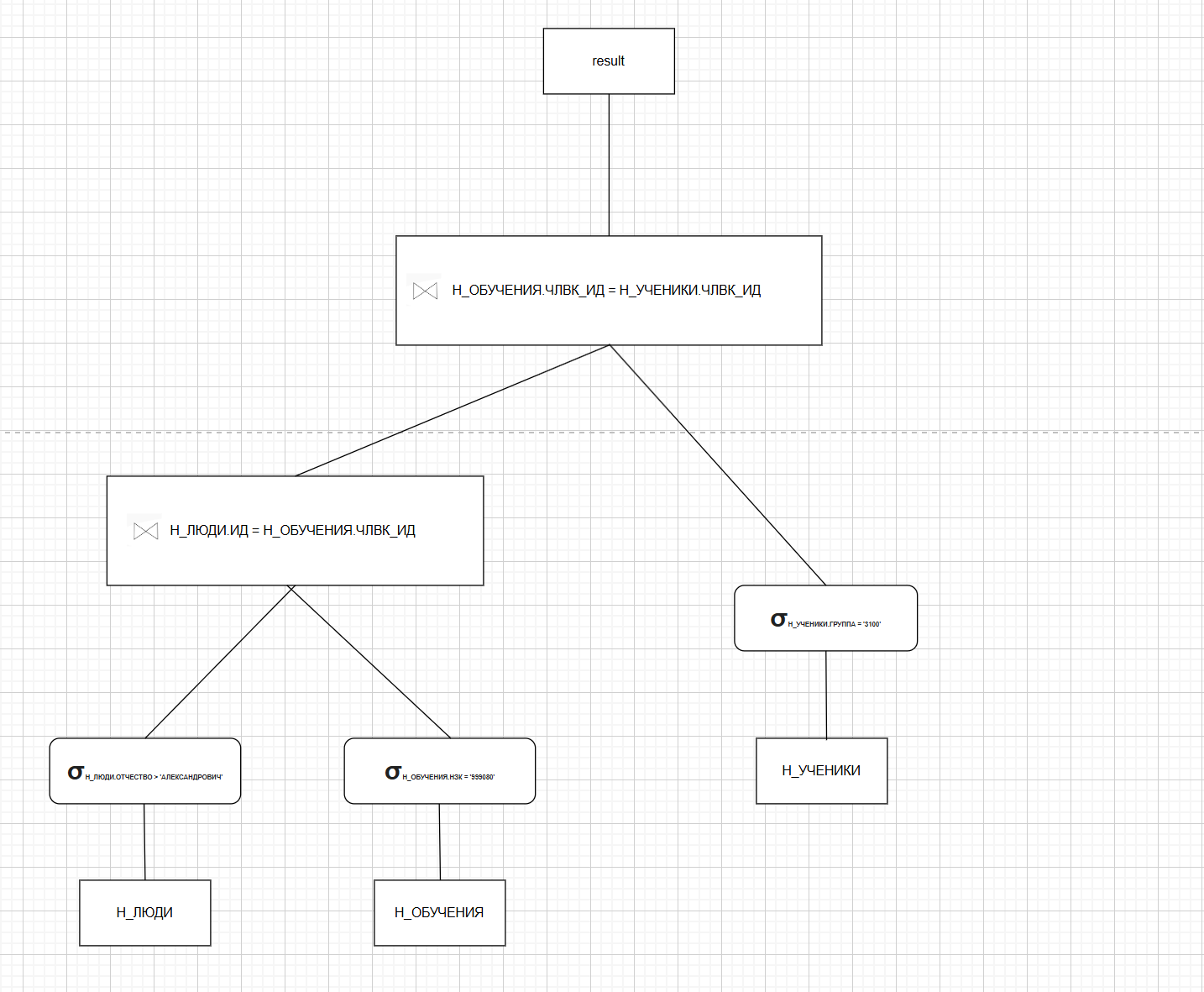
" Index Cond: (""ЧЛВК\_ИД"" = ""Н\_ЛЮДИ"".""ИД"")"

" Filter: ((""ГРУППА"")::text = '3100'::text)"

Rows Removed by Filter: 6

Planning Time: 0.675 ms

Execution Time: 0.663 ms



## **Вывод**

В ходе выполнения лабораторной работы я освоил работу с реляционной алгеброй и научился строить планы выполнения запросов, а также их диаграммы. Я изучил различные виды индексов и узнал, как использовать их для оптимизации запросов. Теперь я могу применять полученные знания для эффективной работы с базами данных и повышения производительности SQL-запросов.