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

Факультет программной инженерии и компьютерной техники

Базы данных

Лабораторная работа №4

Вариант 71

Выполнил: Свечников Константин Денисович, P3130

Проверил: Перцев Т. С., преподаватель практики

**Оглавление**

[Задание 2](#_Toc199078449)

[Ход работы 3](#_Toc199078450)

[Запросы 3](#_Toc199078451)

[Добавление индексов 3](#_Toc199078452)

[Планы выполнения запросов 4](#_Toc199078453)

[EXPLAIN ANALYZE 7](#_Toc199078454)

[Заключение 8](#_Toc199078455)

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]

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.  
   Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД.  
   Фильтры (AND):  
   a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД < 2.  
   b) Н\_ВЕДОМОСТИ.ИД < 39921.  
   c) Н\_ВЕДОМОСТИ.ИД > 1250981.  
   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.ФАМИЛИЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.ГРУППА.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ИМЯ = Владимир.  
   b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД < 113409.  
   Вид соединения: LEFT JOIN.

Ход работы

## Запросы

select "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД", "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"  
from "Н\_ВЕДОМОСТИ"  
 right join "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" on "Н\_ВЕДОМОСТИ"."ВЕД\_ИД" = "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД"  
where "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД"<2 and "Н\_ВЕДОМОСТИ"."ИД"<39921 and "Н\_ВЕДОМОСТИ"."ИД"> 1250981

2.

select "Н\_ЛЮДИ"."ФАМИЛИЯ", "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД", "Н\_УЧЕНИКИ"."ГРУППА"  
from "Н\_ЛЮДИ"  
left join "Н\_ОБУЧЕНИЯ" on "Н\_ЛЮДИ"."ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД"  
left join "Н\_УЧЕНИКИ" on "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД"  
where "Н\_ЛЮДИ"."ИМЯ" = 'Владимир' and "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" < 113409

## Добавление индексов

Для первого запроса можно добавить индексы на следующие атрибуты и таблицы:

* Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД – индекс типа B-Tree
* Н\_ВЕДОМОСТИ.ИД – индекс типа B-Tree

1. Индекс на атрибут «ИД» в таблице «Н\_ТИПЫ\_ВЕДОМОСТЕЙ» позволит ускорить выполнение операций WHERE и JOIN. Без индекса, каждой из этих операций каждый раз придется сканировать таблицу «Н\_ТИПЫ\_ВЕДОМОСТЕЙ»
2. Индекс на атрибут «ИД» в таблице «Н\_ВЕДОМОСТИ» позволит ускорить выполнение операции WHERE, где при его отсутствии каждый раз придется заново сканировать всю таблицу для выполнения условий WHERE.

Для второго запроса можно добавить индексы на следующие атрибуты и таблицы:

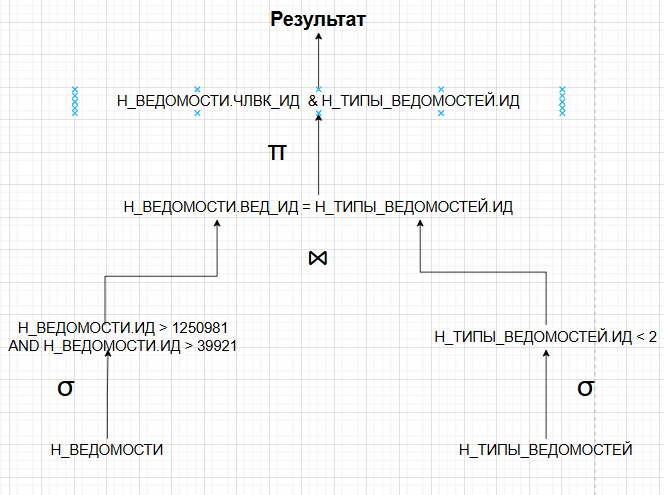
* Н\_ЛЮДИ.ИМЯ – индекс типа B-Tree
* Н\_ЛЮДИ.ИД – индекс типа B-Tree
* Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД – индекс типа B-Tree
* Н\_УЧЕНИКИ.ЧЛВК\_ИД – индекс типа B-Tree

1. Индекс на атрибут «ИМЯ» в таблице «Н\_ЛЮДИ» позволит ускорить выполнение операции WHERE, где при его отсутствии каждый раз придется заново сканировать всю таблицу для выполнения условий WHERE.
2. Индекс на атрибут «ИД» в таблице «Н\_ЛЮДИ» позволит ускорить выполнение операции JOIN. Без индекса, каждой из этих операций каждый раз придется сканировать таблицу «Н\_ЛЮДИ»
3. Индекс на атрибут «ЧЛВК\_ИД» в таблице «Н\_ОБУЧЕНИЯ» позволит ускорить выполнение операций WHERE и JOIN. Без индекса, каждой из этих операций каждый раз придется сканировать таблицу «Н\_ОБУЧЕНИЯ»
4. Индекс на атрибут «ЧЛВК\_ИД» в таблице «Н\_УЧЕНИКИ» позволит ускорить выполнение операции JOIN. Без индекса, каждой из этих операций каждый раз придется сканировать таблицу «Н\_УЧЕНИКИ»

## Планы выполнения запросов

Планы выполнения запросов без индексов для первого запроса:

1. Производится полный скан таблицы «Н\_ТИПЫ\_ВЕДОМОСТЕЙ» с применением фильтра «ИД» < 2, затем производится полный скан таблицы «Н\_ВЕДОМОСТИ» с фильтром «ИД» < 39921 AND «ИД» > 1250981, затем результаты объединяются через RIGHT JOIN



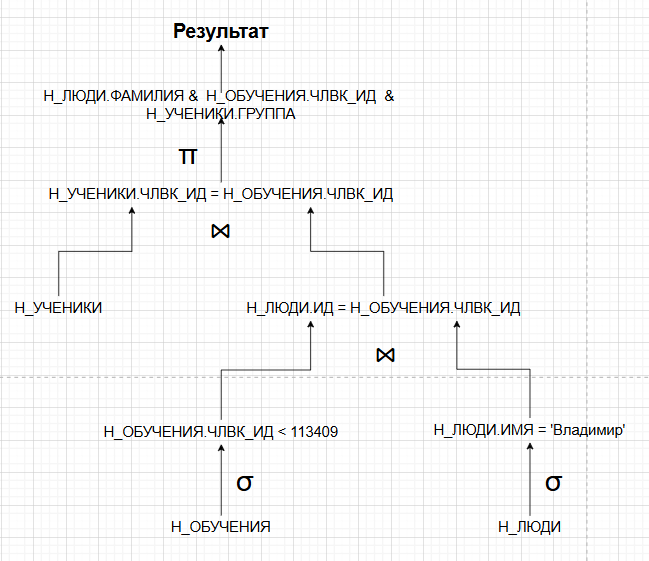
1. Производится полный скан таблицы «Н\_ВЕДОМОСТИ» с фильтром «ИД» < 39921 AND «ИД» > 1250981, затем производится полный скан таблицы «Н\_ТИПЫ\_ВЕДОМОСТЕЙ» с применением фильтра «ИД» < 2, потом результаты объединяются через RIGHT JOIN.

Оптимальным планом выполнения данного запроса является первый вариант, так как фильтр по ИД применяется к таблице «Н\_ТИПЫ\_ВЕДОМОСТЕЙ» до выполнения объединения, сокращая количество строк перед соединением

При добавлении индексов на столбцы «Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД» и «Н\_ВЕДОМОСТИ.ИД» первый план выполнения запроса станет оптимальнее, так как будет использован индексный поиск вместо полного сканирования таблицы.

Планы выполнения запросов без индексов для второго запроса:

1. Производится полный скан таблицы «Н\_ЛЮДИ» с фильтром «ИМЯ» = ‘Владимир’, затем производится полный скан таблицы «Н\_ОБУЧЕНИЯ» с фильтром “ЧЛВК\_ИД” < 113409, потом производится полный скан таблицы «Н\_УЧЕНИКИ», и наконец результаты объединяются через LEFT JOIN.

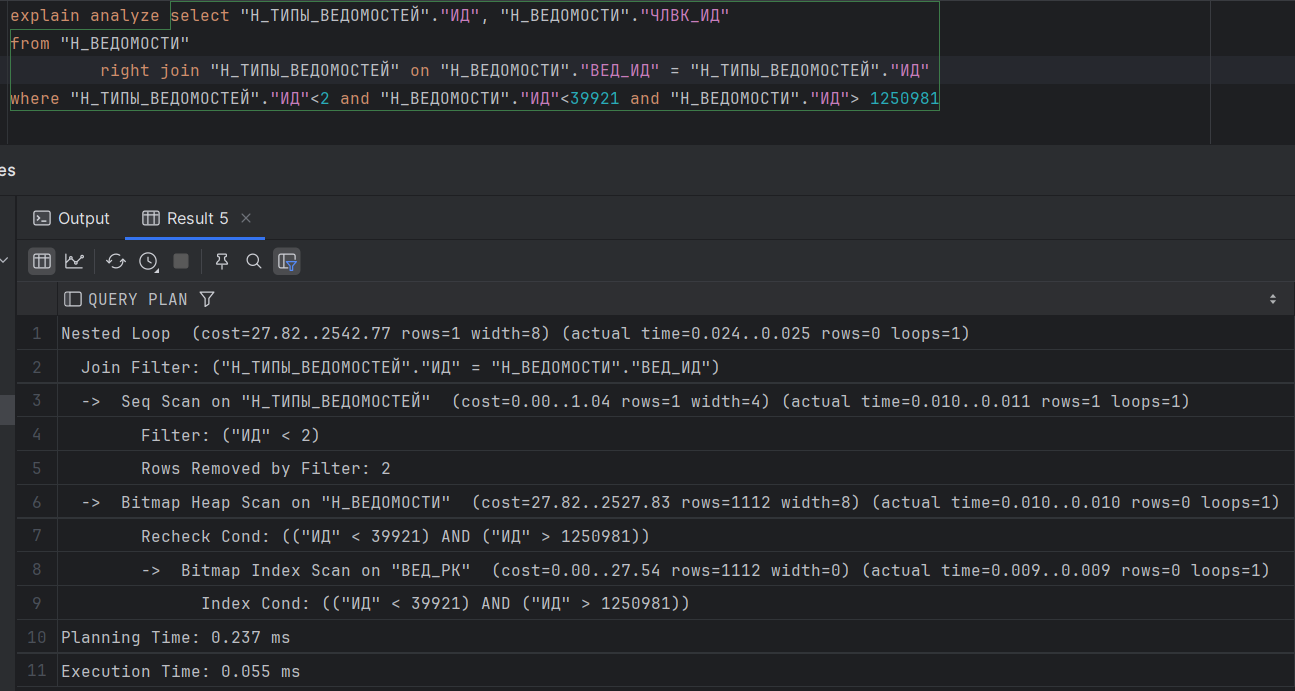


1. Производится полный скан таблицы «Н\_ОБУЧЕНИЯ» с фильтром «ЧЛВК\_ИД» < 113409, затем производится полный скан таблицы «Н\_ЛЮДИ» с фильтром “ИМЯ” = ‘Владимир’, потом производится полный скан таблицы «Н\_УЧЕНИКИ», и наконец результаты объединяются через LEFT JOIN

Оптимальным планом выполнения данного запроса является первый план, так как фильтр “ИМЯ” = ‘Владимир’ сокращает число строк в таблице «Н\_ЛЮДИ» на раннем этапе.

При добавлении индексов на столбцы «Н\_ЛЮДИ.ИМЯ» и «Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД» план выполнения становится оптимальнее за счет индексного поиска вместе полного сканирования таблиц.

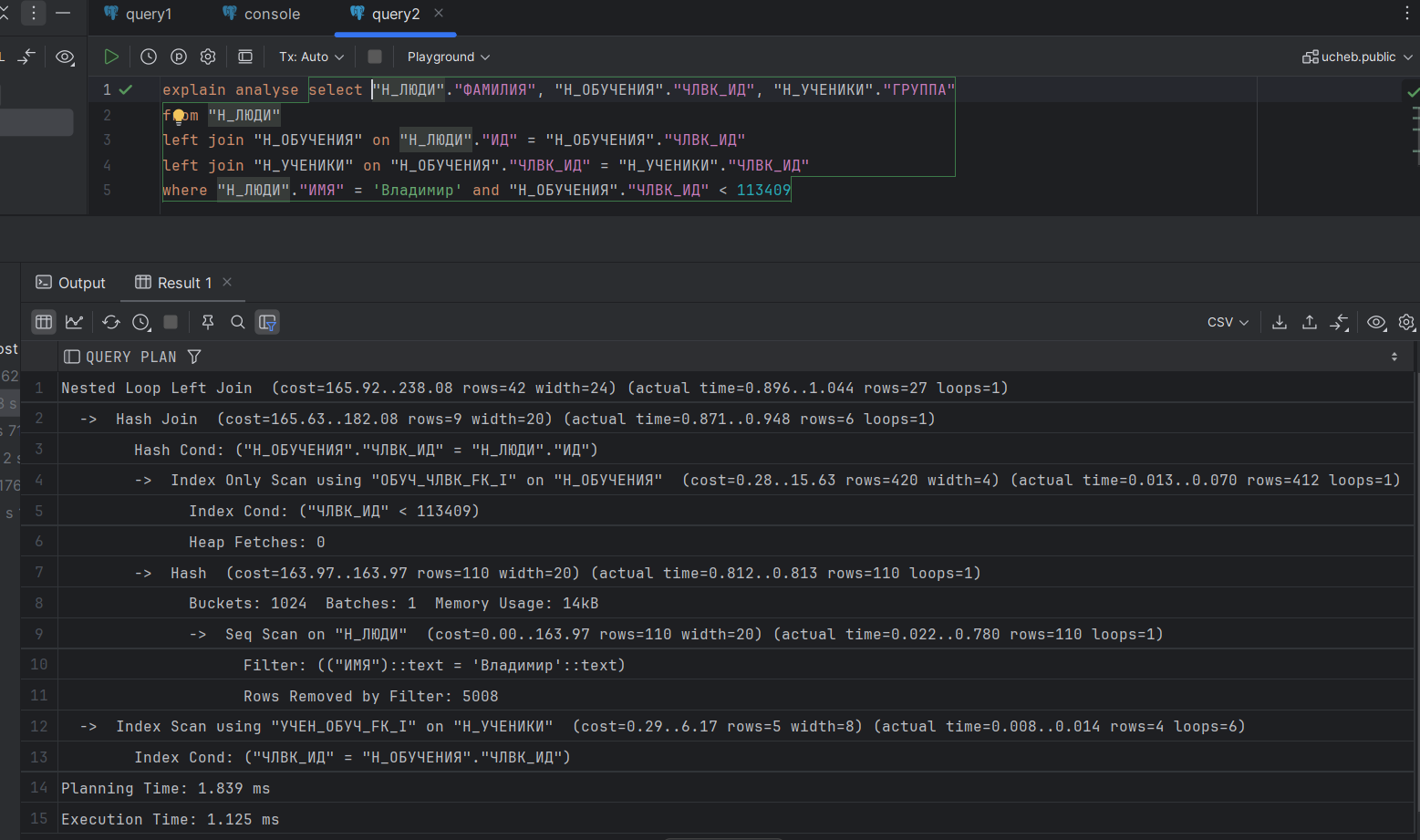
## EXPLAIN ANALYZE



Из результата можно увидеть, что для выполнения запроса был выбран план выполнения с использованием вложенного цикла (Nested Loop). Он соединяет таблицы «Н\_ТИПЫ\_ВЕДОМОСТЕЙ» и «Н\_ВЕДОМОСТИ», учитывая условие соединения по полю «ВЕД\_ИД». Затем производится полное сканирование таблицы «Н\_ТИПЫ\_ВЕДОМОСТЕЙ» (Seq Scan) с фильтром «ИД» < 2. После этого проводятся последовательные битовые индексные сканирования таблицы «Н\_ВЕДОМОСТИ» и поля «ВЕД\_РК» с учетом фильтра «ИД» < 39921 AND «ИД» > 1250981.

Также видно, что стоимость выполнения плана составила 27.8, а затраченное время составило 0.055 ms, что довольно быстро.

Из описания индексных сканов можно сделать вывод, что для оптимизации запроса были использованы индексы, что позволило ускорить выполнение запроса.



Из результата можно видеть, что для выполнения запроса был использован вложенный цикл, который сперва соединил по хэшу таблицы «Н\_ОБУЧЕНИЯ» и «Н\_ЛЮДИ», затем провел сканирование только по индексу с фильтром «ЧЛВК\_ИД» < 113409. Затем было сделано хэширование результатов сканирования, после чего провел полное сканирование таблицы «Н\_ЛЮДИ» с фильтром «ИМЯ» = ‘Владимир’. В конце он провел сканирование по индексу для «Н\_УЧЕНИКИ» выполняя условие «ЧЛВК\_ИД» = «Н\_ОБУЧЕНИЯ».«ЧЛВК\_ИД».

Стоимость выполнения составила 165.9, а время выполнения – 1.125 ms.

Заключение

В ходе выполнения данной лабораторной работы я изучил различные виды индексов и узнал, как использовать их для оптимизации скорости выполнения запросов.