МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

## ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №4** по дисциплине

«ИНФОРМАЦИОННЫЕ СИСТЕМЫ И БАЗЫ ДАННЫХ»   
вариант 11209

***Выполнил:***

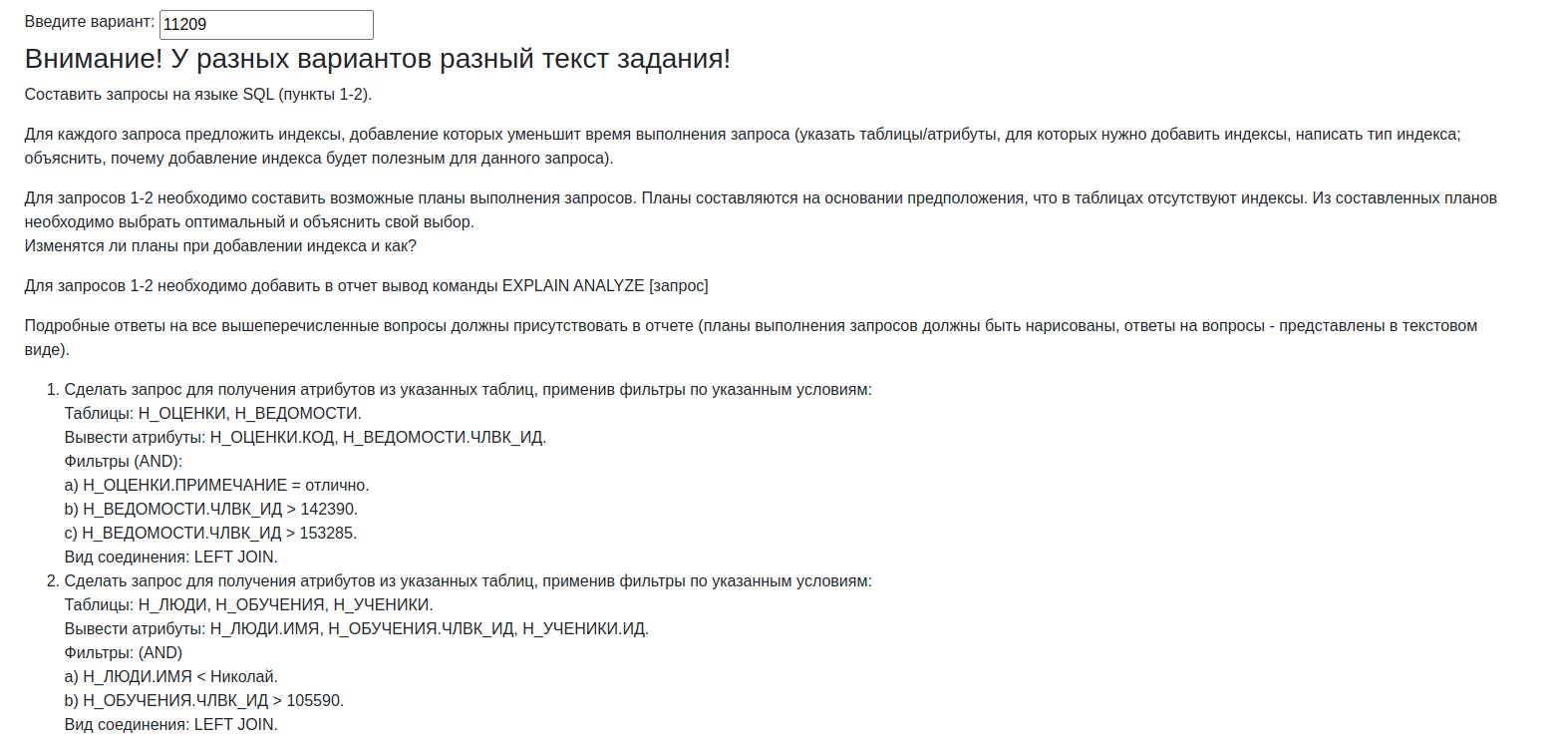
Студент группы P33312

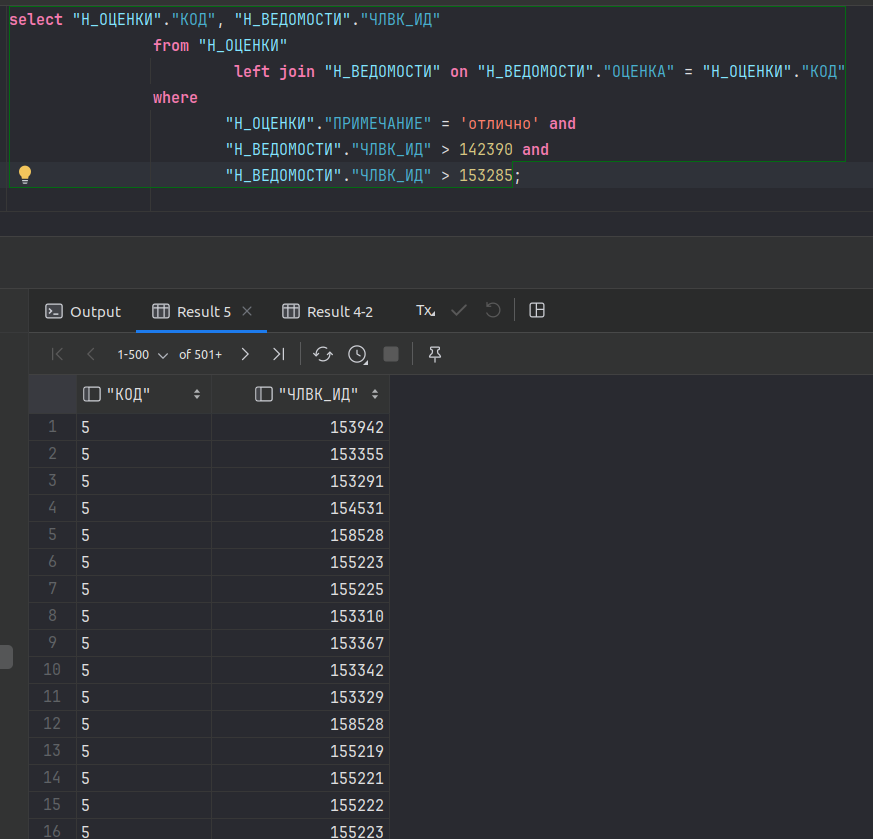
Лысенко А.К.

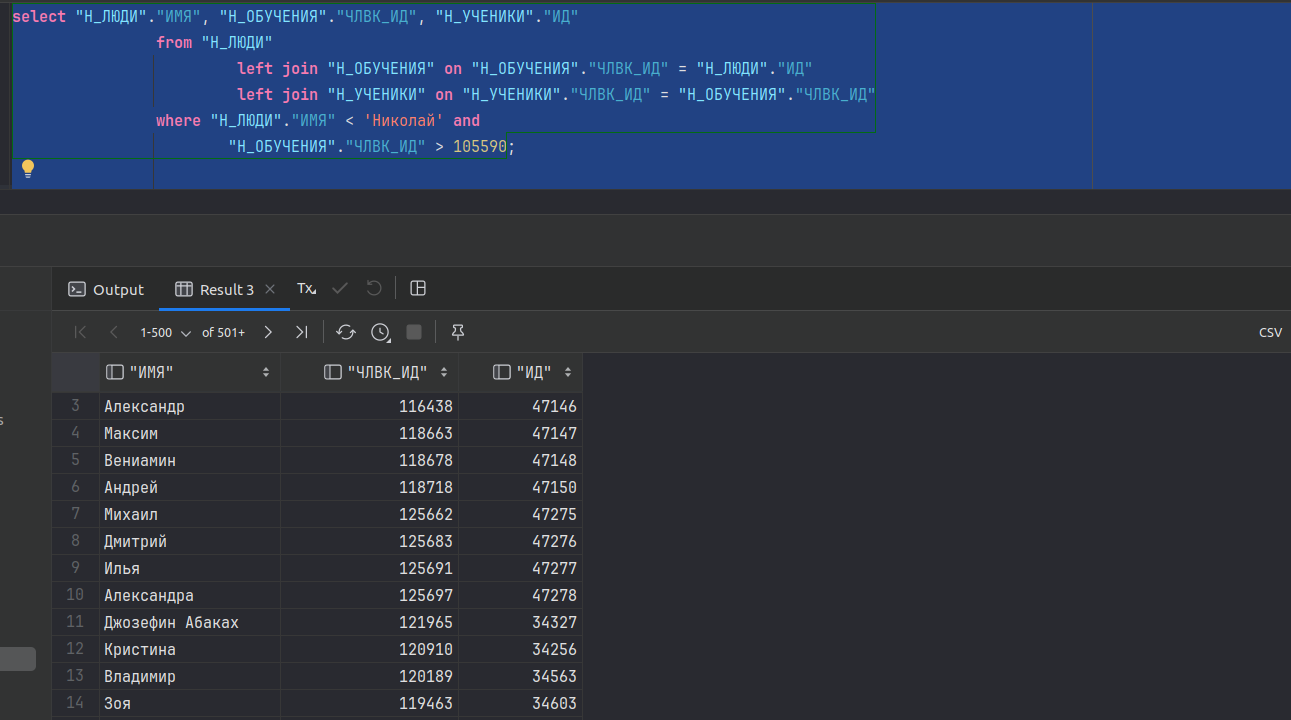
***Преподаватель:***  Наумова Н.А.

Санкт-Петербург, 2023

**Текст задания**

****

**1 запрос:**

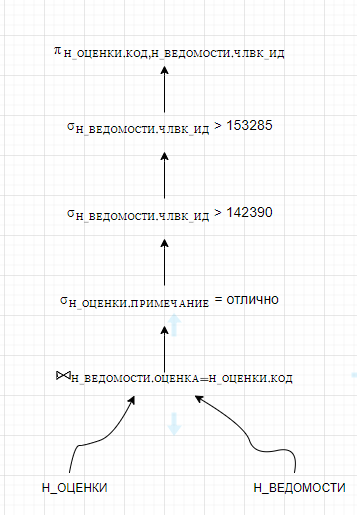
**2 запрос**

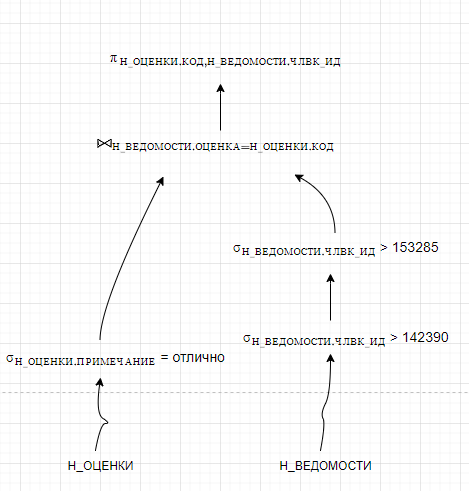
**Создание индексов**  
1 запрос:  
create index notes on "Н\_ОЦЕНКИ" using hash("ПРИМЕЧАНИЕ");  
create index ids on "Н\_ВЕДОМОСТИ" using btree("ЧЛВК\_ИД");

Для таблицы Н\_ОЦЕНКИ для атрибута ПРИМЕЧАНИЕ имеет смысл создать индекс, так как он используется в фильтре. Имеет смысл использовать хэш-индекс, так как в фильтре используется проверка на равенства. Хэши строк сравниваются за константу. Это поможет улучшить скорость операций WHERE и ON.  
  
Для таблицы Н\_ВЕДОМОСТИ для атрибута ЧЛВК\_ИД имеет смысл создать индекс, так как он используется в фильтре. Имеет смысл использовать индекс на основе b-tree дерева, так в фильтре используется оператор >. Используя такой индекс мы снизим время поиска с линейного до логарифмического (O(n) → O(logN) при выполнении выборки данных при использовании WHERE.

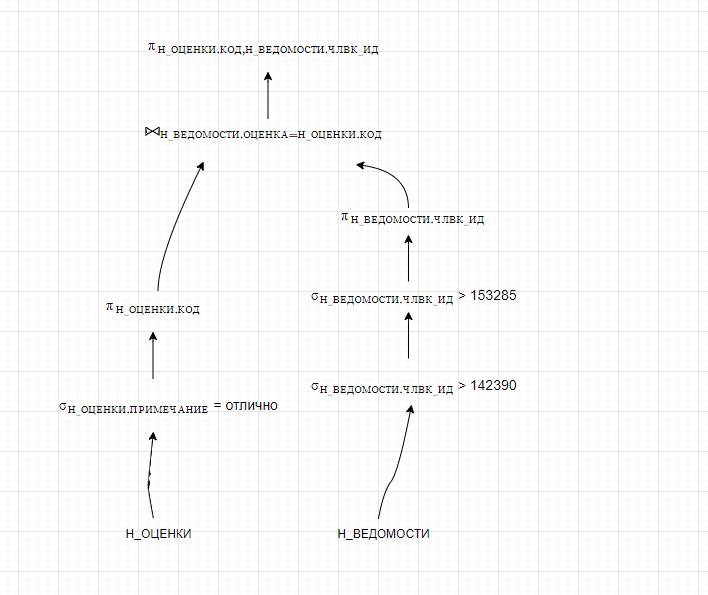
2 запрос:  
create index names on "Н\_ЛЮДИ" using btree("ИМЯ");  
create index ids on "Н\_ОБУЧЕНИЯ" using btree("ЧЛВК\_ИД");

Для таблицы Н\_ЛЮДИ для атрибута ИМЯ имеет смысл создать индекс, так как он используется в фильтре. Имеет смысл использовать индекс на основе b-tree дерева, так как мы используем оператор >. Создание данного индекса улучшить выборку данных при использовании WHERE.  
  
Для таблицы Н\_ОБУЧЕНИЯ для атрибута ЧЛВК\_ИД имеет смысл создать индекс, так как он используется в фильтре. Имеет смысл использовать индекс на основе b-tree дерева, так в фильтре используется оператор >. Используя такой индекс мы снизим время поиска с линейного до логарифмического (O(n) → O(logN) при выполнении выборки данных при использовании WHERE.

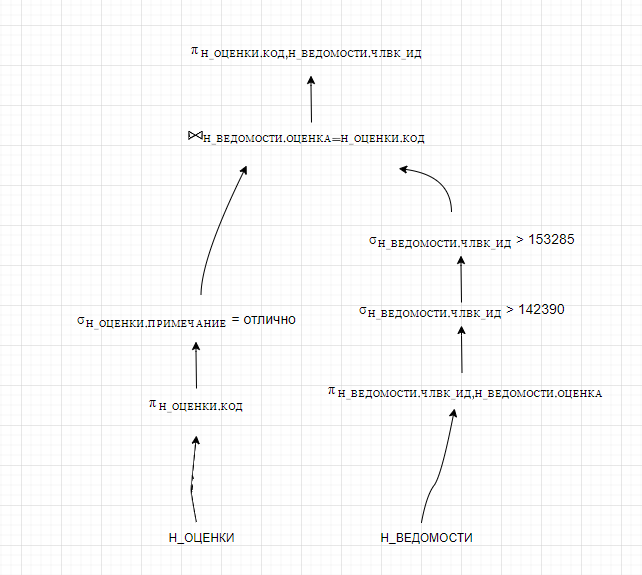
**1 запрос. План выполнения.**1) Сначала происходит соединение отношений, далее последовательная выборка, результатом будет проекция двух атрибутов  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
2) Сначала происходит операция выборки, после соединение двух отношений, результатом будет проекция двух атрибутов.



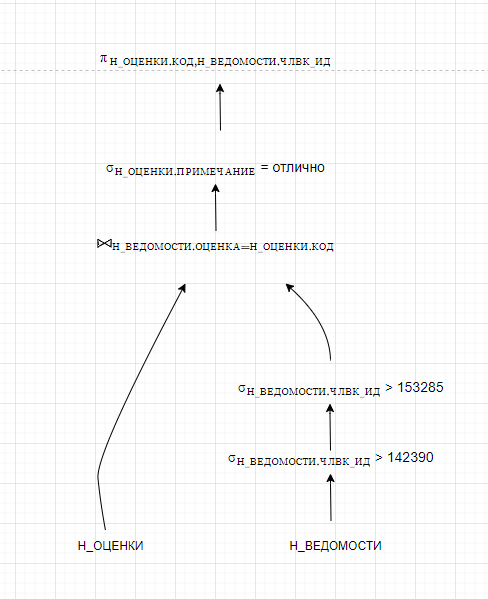
3) Сначала производится выборка для двух отношений, далее получается проекция для двух отношений из атрибутов, необходимых для выборки, соединения и итоговой проекции, далее соединение двух отношений, результатом является проекция двух атрибутов.

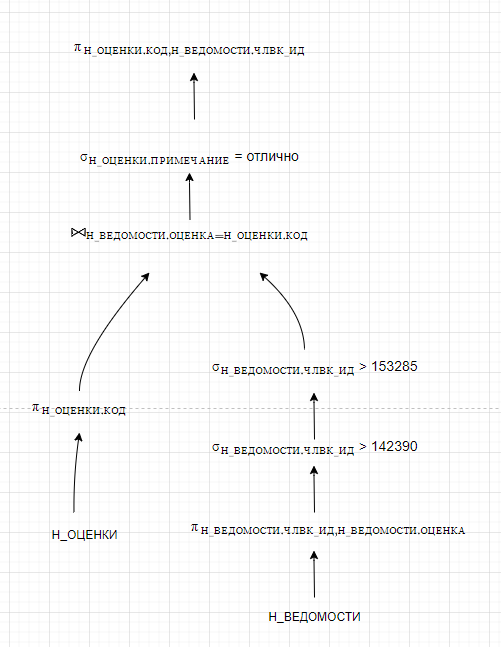


4) Для начала получается проекция для двух отношений из атрибутов, необходимых для выборки, соединения и итоговой проекции, далее производится выборка и соединение. Результатом является проекция двух атрибутов.



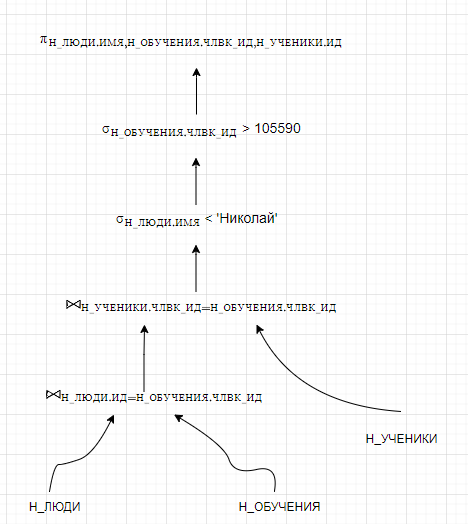
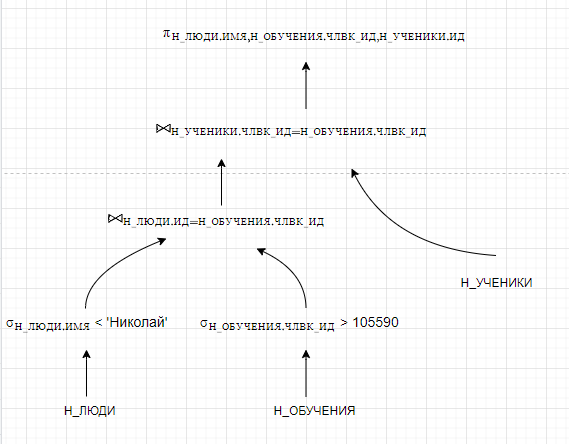
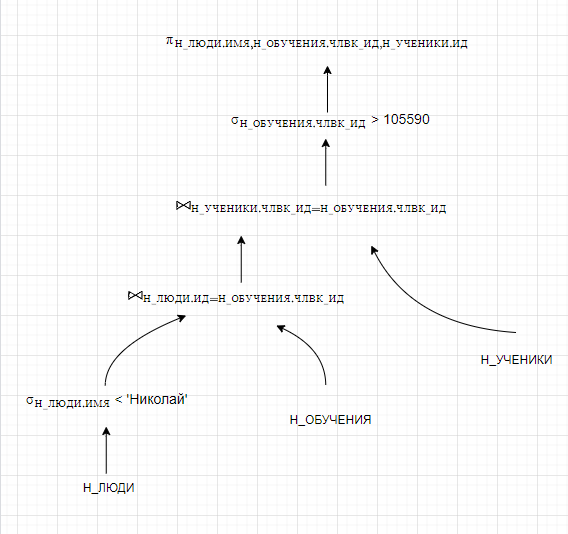
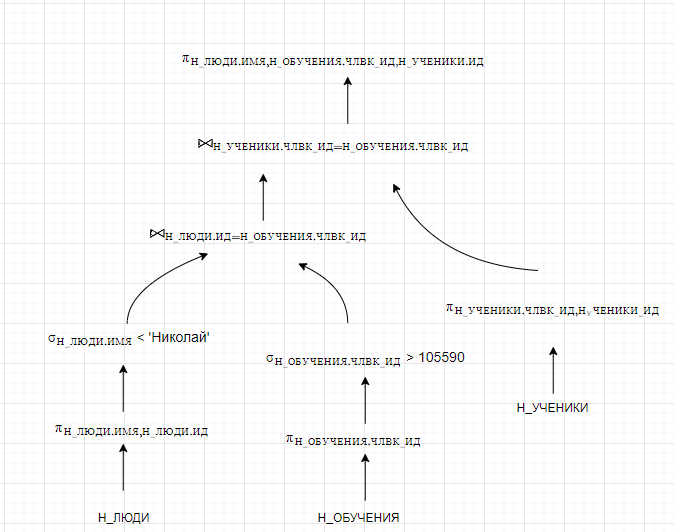
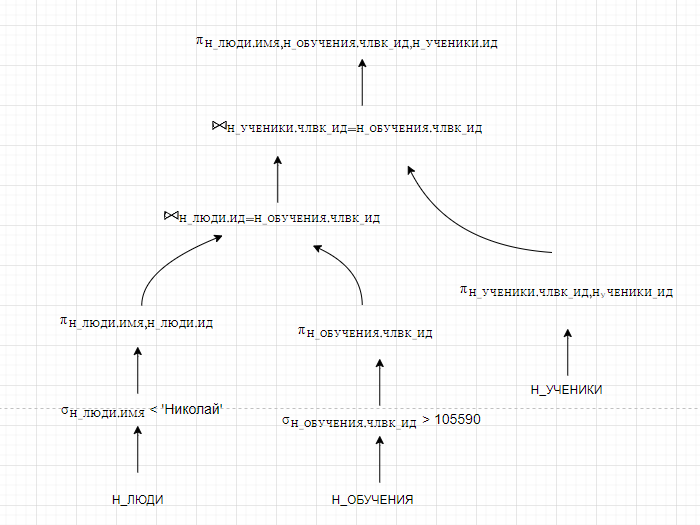
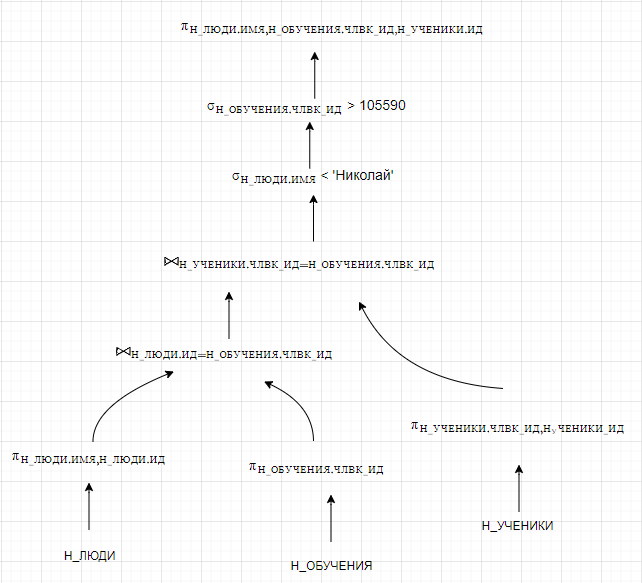
5) Для начала получается выборка для одного из отношений из атрибутов, далее происходит соединение таблиц, после которого производится выборка для результата соединения. Результатом является проекция двух атрибутов. В данном плане атрибуты выборки после соединения могут меняться – могут быть выборки из ветки Н\_ВЕДОМОСТИ.



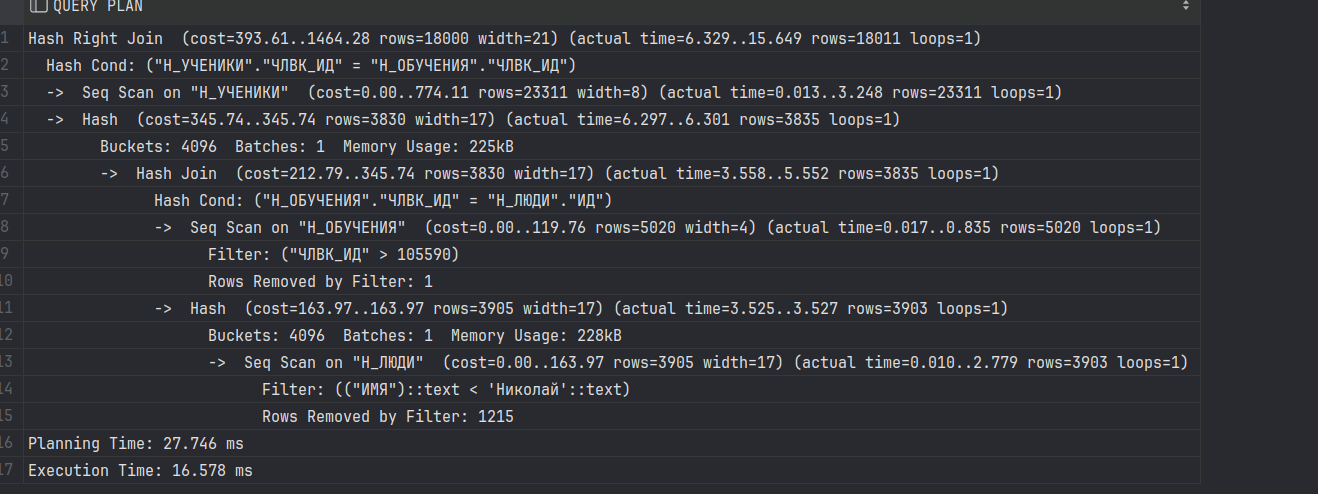
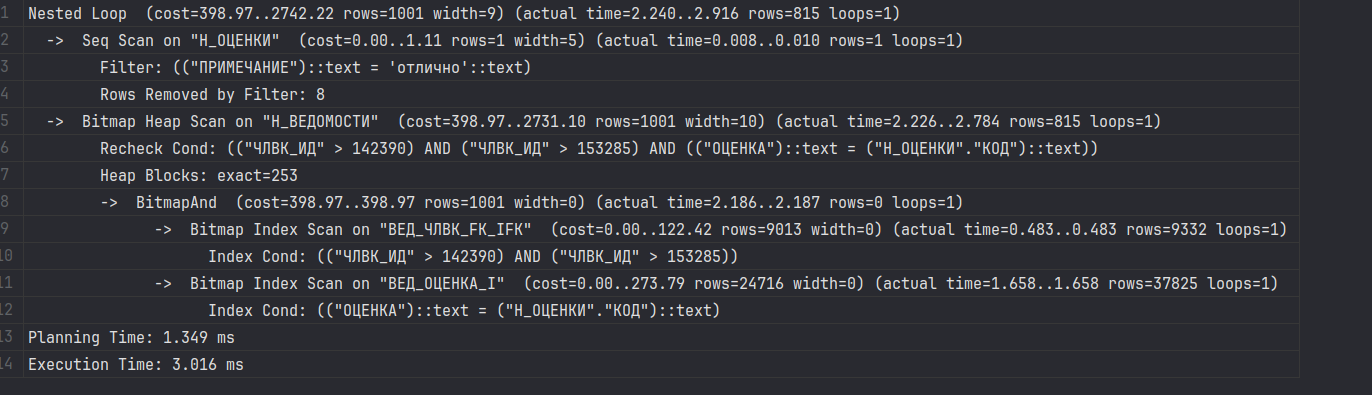
6) Для начала получается проекция для двух отношений из атрибутов, необходимых для выборки, соединения и итоговой проекции, далее производится выборка для одного из отношений, а после соединения двух отношений, после которого производится выборка для результата отношения. Результатом является проекция двух атрибутов. В данном плане атрибуты выборки после соединения могут меняться – могут быть выборки из ветки Н\_ВЕДОМОСТИ.  


Из составленных возможных планов выполнения запроса лучшим является четвертый, поскольку в нем изначально получаются проекции и производятся выборки, а уже после этого выполняются соединение. Это позволяет уменьшить размер хранимых данных.  
  
При создании индексов четвертый план останется оптимальным, при этом даже ускорится за счёт ускорения поиска. Также довольно эффективным будет третий план, потому что скорость поиска в нем увеличится. В нем также соединение происходит после выборки, что позволяет ускорить выполнение запроса, но, в отличие от четвертого плана, в нем выборка будет происходить по все таблице целиком, а не по отдельным проекциям.

**2 запрос. План выполнения.**

1. Соединение двух отношений, далее результат соединения соединяется с третьим отношением, а для результата двух соединений последовательно производится выборка. Результатом является проекция трех атрибутов.  
   
2. Для начала производится выборка для первых двух отношений, далее производится соединение этих двух отношений, а после результат соединения соединяется третьим отношением. Результатом является проекция трёх атрибутов.  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   3) Сначала производится выборка для одного отношения, далее производится соединение первых двух отношений, а после результат соединения соединяется с третьим отношением. Происходит выборка по результату соединения. Результатом является проекция трёх атрибутов. На первом шаге выборка может быть у ветки Н\_ОБУЧЕНИЯ.  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   4) Сначала получается проекция для двух отношений, после по ним производится выборка и соединение. Далее результат соединения соединяется с третьим отношением, для которого перед соединением была получена проекция. Результатом является проекция трёх атрибутов.  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   5) Сначала формируются выборки для каждого отношения, далее строятся проекции. После чего данные проекции соединяются. Результатом является проекция трёх атрибутов.  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   6) Получаются проекции для двух отношений, после чего эти отношения соединяются, а результат соединения соединяется с третьим отношением, для которого перед соединением тоже была получена проекция. Далее производится последовательная выборка. Результатом является проекция трёх атрибутов.   
   

Из составленных возможных планов выполнения запросы лучшим является четвертый, поскольку в нем изначально получаются проекции и производятся выборки, а уже после этого выполняется соединение. Это позволяет уменьшить размер хранимых данных.  
  
При создании индексов четвертый план останется оптимальным, при этом даже ускорится за счёт ускорения поиска. Также довольно эффективным будет пятый план, потому что скорость поиска в нем увеличится. В нем также соединение происходит после выборки, что позволяет ускорить выполнение запроса, но, в отличие от четвертого плана, в нем выборка будет происходить по всей таблице целиком, а не по отдельным проекциям.



Вывод: узнал, что такое индексы, какие индексы бывают и как они помогают уменьшить время выполнения запроса.