Национальная научно-образовательная корпорация ИТМО

Факультет программной инженерии и компьютерной техники

ЛАБОРАТОРНАЯ РАБОТА №4

по дисциплине

«БАЗЫ ДАННЫХ»

Вариант № 12052025

Выполнил:

Студент группы P3111

Наземцев Сергей Дмитриевич

Преподаватель:

Николаев Владимир Вячеславович

Практик:

Чупанов Аликылыч Алибекович

Санкт-Петербург, 2023

# 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\_ОЦЕНКИ, Н\_ВЕДОМОСТИ.  
   Вывести атрибуты: Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ, Н\_ВЕДОМОСТИ.ИД.  
   Фильтры (AND):  
   a) Н\_ОЦЕНКИ.КОД = 4.  
   b) Н\_ВЕДОМОСТИ.ИД = 1250972.  
   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.ИД, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.ИД.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ОТЧЕСТВО > Георгиевич.  
   b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 163484.  
   c) Н\_УЧЕНИКИ.ИД = 150308.  
   Вид соединения: LEFT JOIN.

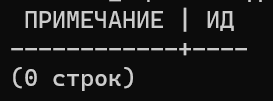
# Запрос №1

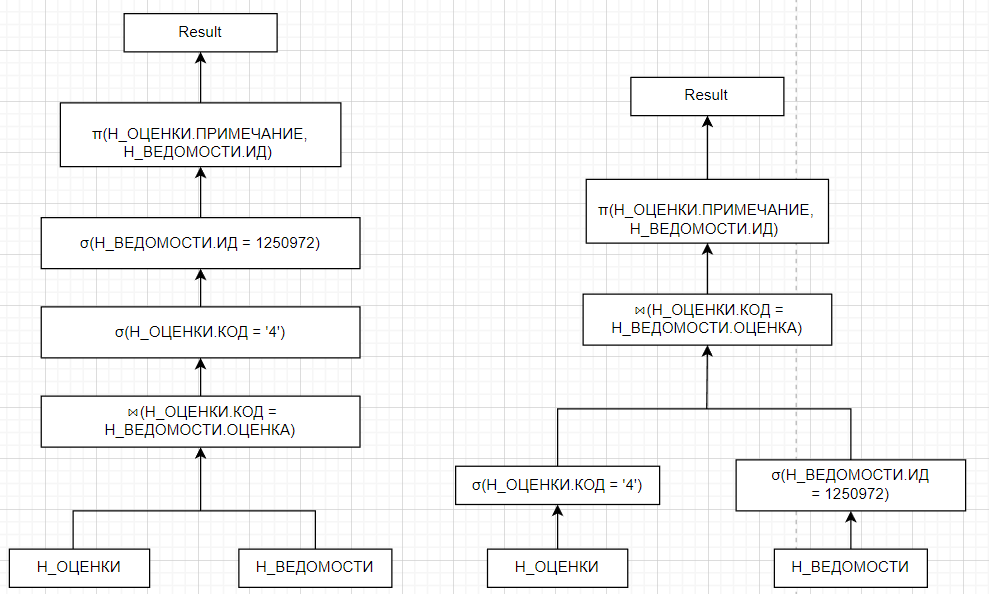
SELECT Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ, Н\_ВЕДОМОСТИ.ИД FROM Н\_ОЦЕНКИ

LEFT JOIN Н\_ВЕДОМОСТИ

ON Н\_ОЦЕНКИ.КОД = Н\_ВЕДОМОСТИ.ОЦЕНКА

WHERE Н\_ОЦЕНКИ.КОД = '4' AND Н\_ВЕДОМОСТИ.ИД = 1250972;





Оптимальным является план №2, так как он производит объединение таблиц по уже выбранным ранее атрибутам, а не по таблицам целиком.

# Индексы

CREATE INDEX "ИНД\_ОЦЕНКИ\_КОД" ON "Н\_ОЦЕНКИ" USING HASH("КОД");

CREATE INDEX "ИНД\_ВЕДОМОСТИ\_ИД" ON "Н\_ВЕДОМОСТИ" USING HASH("ИД");

CREATE INDEX “ИНД\_ВЕДОМОСТИ\_ОЦЕНКА” ON “Н\_ВЕДОМОСТИ” USING HASH(“ОЦЕНКА”)

Добавление этих индексов должно ускорить выполнение запросов, так как по перечисленным полям происходит выборка с использованием оператора сравнения, а также теперь соединение таблиц будет происходить быстрее. Используется только прямое сравнение, поэтому будут эффективны хэш-индексы.

При добавлении индексов планы выполнения запросов изменятся, так как вместо полного сканирования таблиц будет производиться индексный скан и Nested Loops Join станет быстрее.

# EXPLAIN ANALYZE

Nested Loop (cost=0.42..9.56 rows=1 width=422) (actual time=0.033..0.034 rows=0 loops=1)

-> Seq Scan on "Н\_ОЦЕНКИ" (cost=0.00..1.11 rows=1 width=452) (actual time=0.014..0.015 rows=1 loops=1)

Filter: (("КОД")::text = '4'::text)

Rows Removed by Filter: 8

-> Index Scan using "ВЕД\_PK" on "Н\_ВЕДОМОСТИ" (cost=0.42..8.44 rows=1 width=10) (actual time=0.017..0.017 rows=0 loops=1)

Index Cond: ("ИД" = 1250972)

Filter: (("ОЦЕНКА")::text = '4'::text)

Rows Removed by Filter: 1

Planning Time: 0.209 ms

Execution Time: 0.075 ms

# Запрос №2

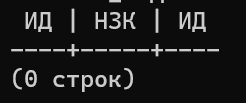
SELECT Н\_ЛЮДИ.ИД, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.ИД

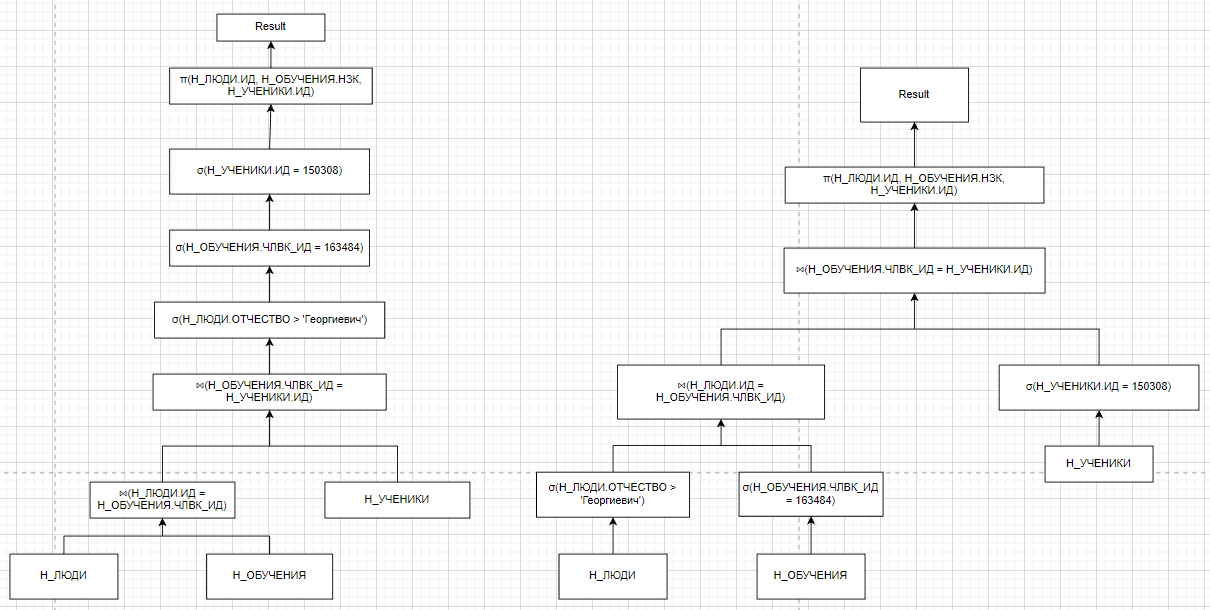
FROM Н\_ЛЮДИ

LEFT JOIN Н\_ОБУЧЕНИЯ ON Н\_ЛЮДИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД

LEFT JOIN Н\_УЧЕНИКИ ON Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = Н\_УЧЕНИКИ.ИД

WHERE Н\_ЛЮДИ.ОТЧЕСТВО > 'Георгиевич' AND Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 163484 AND Н\_УЧЕНИКИ.ИД = 150308;





Оптимальным является план №2, так как он производит объединение таблиц по уже выбранным ранее атрибутам, а не по таблицам целиком.

# Индексы

CREATE INDEX "ИНД\_ЛЮДИ\_ОТЧЕСТВО" ON "Н\_ЛЮДИ" USING BTREE("ОТЧЕСТВО”);

CREATE INDEX "ИНД\_ОБУЧЕНИЯ\_ЧЛВК\_ИД" ON "Н\_ОБУЧЕНИЯ" USING HASH("ЧЛВК\_ИД”);

CREATE INDEX "ИНД\_УЧЕНИКИ\_ИД" ON "Н\_УЧЕНИКИ" USING HASH("ИД");

Добавление этих индексов должно ускорить выполнение запросов, так как по перечисленным полям происходит выборка с использованием операторов сравнения. В двух случаях используется только прямое сравнение, поэтому будут эффективны хэш-индексы, в одном случае используется оператор «>», поэтому нужно использовать индекс BTREE.

При добавлении индексов планы выполнения запросов изменятся, так как вместо полного сканирования таблиц будет производиться индексный скан и Nested Loops Join станет быстрее.

# EXPLAIN ANALYZE

Result (cost=0.00..0.00 rows=0 width=14) (actual time=0.001..0.001 rows=0 loops=1)

One-Time Filter: false

Planning Time: 0.342 ms

Execution Time: 0.014 ms

# Вывод

При выполнении лабораторной работы я познакомился с использованием индексов для ускорения обработки запросов в SQL, а также планами выполнения запросов, их построением и анализом.