

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

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №4

Вариант 686

Выполнил:

Шмунк Андрей Александрович

Группа Р3108

Преподаватели:

Афанасьев Дмитрий Борисович

Николаев Владимир Вячеславович

Санкт-Петербург 2024

Содержание

Задание	3
Запрос №1	3
Индексы.....	4
EXPLAIN ANALYZE.....	4
Запрос №2	5
Индексы.....	5
EXPLAIN ANALYZE.....	5
Вывод	6

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]

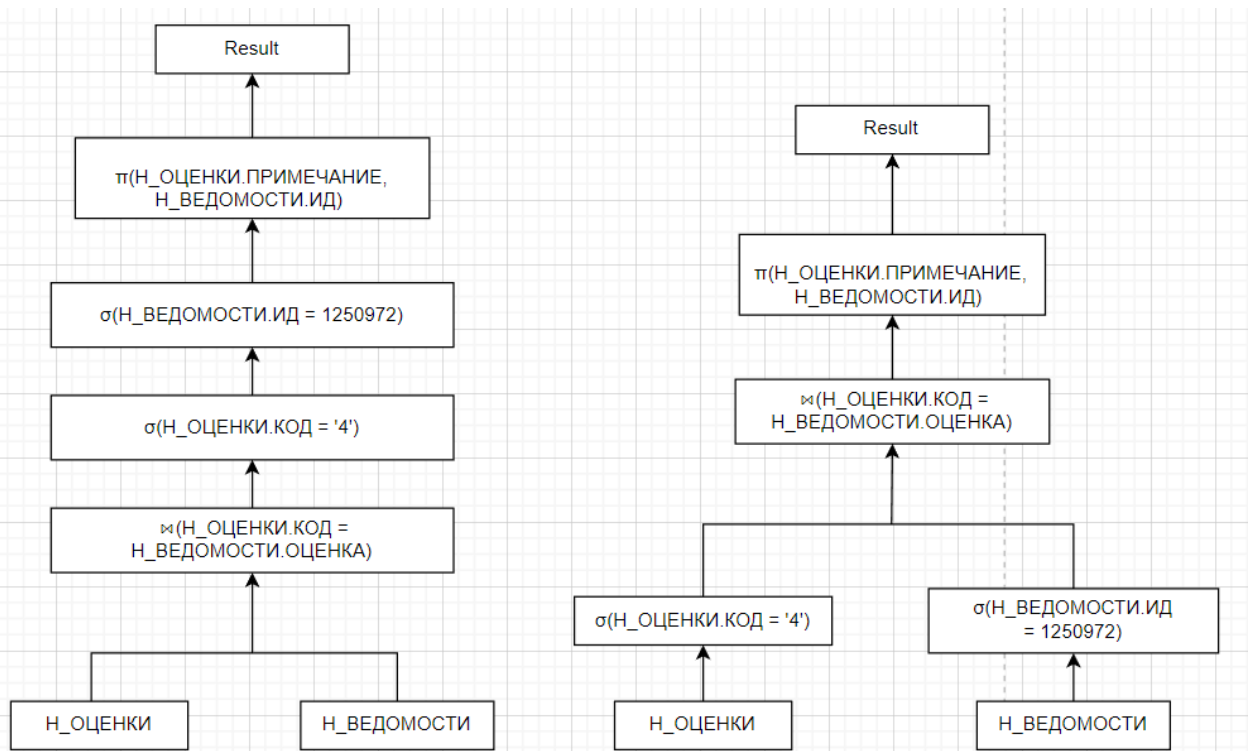
Подробные ответы на все вышеперечисленные вопросы должны присутствовать в отчете (планы выполнения запросов должны быть нарисованы, ответы на вопросы - представлены в текстовом виде).

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
Таблицы: Н_ОЦЕНКИ, Н_ВЕДОМОСТИ.
Вывести атрибуты: Н_ОЦЕНКИ.ПРИМЕЧАНИЕ, Н_ВЕДОМОСТИ.ИД.
Фильтры (AND):
а) Н_ОЦЕНКИ.КОД = 4.
б) 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)
а) Н_ЛЮДИ.ОТЧЕСТВО > Георгиевич.
б) Н_ОБУЧЕНИЯ.ЧЛВК_ИД = 163484.
с) 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;
```

```
ПРИМЕЧАНИЕ | ИД  
-----+-----  
(0 строк)
```



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
BTREE("ИД");
CREATE INDEX "ИНД_ВЕДОМОСТИ_ОЦЕНКА" ON "Н_ВЕДОМОСТИ" USING
HASH("ОЦЕНКА")
```

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

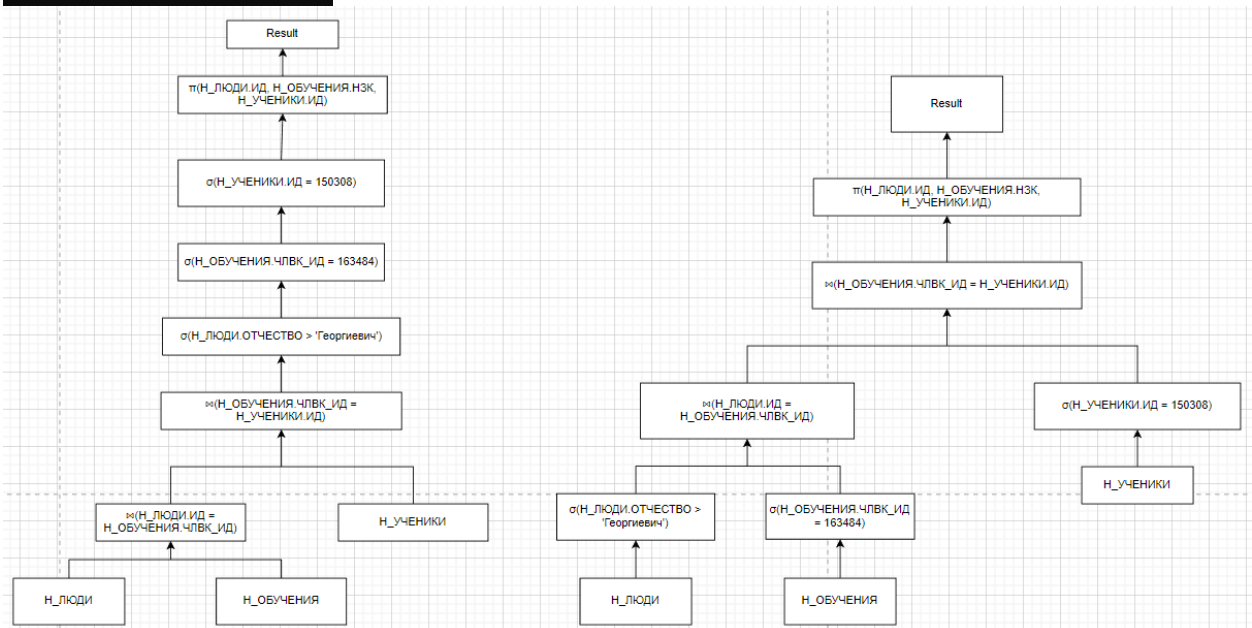
EXPLAIN ANALYZE

```
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 "БЕД_ПК" 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;
```

```
ИД | НЗК | ИД
----+-----+----
(0 строк)
```



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
HASH("ОТЧЕСТВО");
CREATE INDEX "ИНД_ОБУЧЕНИЯ_ЧЛВК_ИД" ON "Н_ОБУЧЕНИЯ" USING
BTREE("ЧЛВК_ИД");
CREATE INDEX "ИНД_УЧЕНИКИ_ИД" ON "Н_УЧЕНИКИ" USING BTREE("ИД");
```

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.