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

Факультет программной инженерии и компьютерной техники

Лабораторная работа по базам данных №4

Вариант № 31089

Выполнил:
Студент группы P3106
Хахулина Светлана Алексеевна
Проверил:
Вербовой Александр Александрович,
Преподаватель-практик ФПИиКТ

Санкт-Петербург, 2025

Оглавление

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).

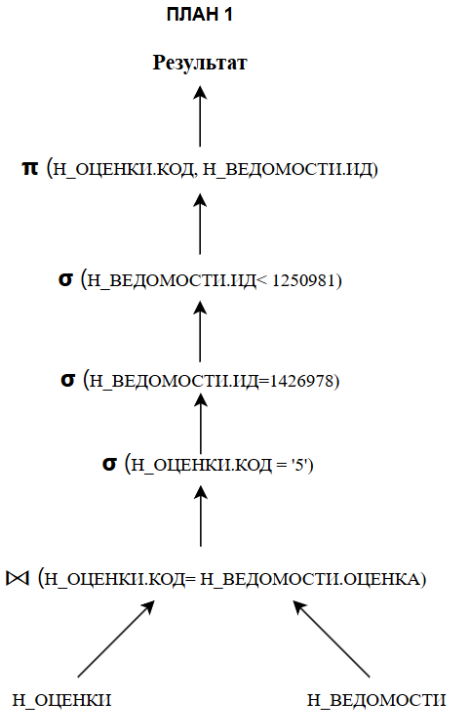
1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
   Таблицы: H_ОЦЕНКИ, H_ВЕДОМОСТИ.
   Вывести атрибуты: H_ОЦЕНКИ.КОД, H_ВЕДОМОСТИ.ИД.
   Фильтры (AND):
   a) H_ОЦЕНКИ.КОД = 5.
   b) H_ВЕДОМОСТИ.ИД = 1426978.
   c) H_ВЕДОМОСТИ.ИД < 1250981.
   Вид соединения: INNER JOIN.

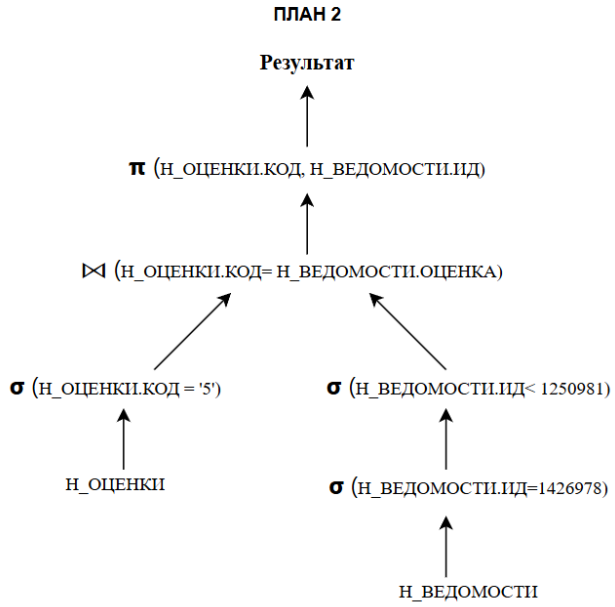
2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
   Таблицы: H_ЛЮДИ, H_ОБУЧЕНИЯ, H_УЧЕНИКИ.
   Вывести атрибуты: H_ЛЮДИ.ФАМИЛИЯ, H_ОБУЧЕНИЯ.НЗК, H_УЧЕНИКИ.ИД.
   Фильтры: (AND)
   a) H_ЛЮДИ.ОТЧЕСТВО = Александрович.
   b) H_ОБУЧЕНИЯ.ЧЛВК_ИД < 112514.
   Вид соединения: LEFT JOIN.

Запрос №1

Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
Таблицы: Н_ОЦЕНКИ, Н_ВЕДОМОСТИ.
Вывести атрибуты: Н_ОЦЕНКИ.КОД, Н_ВЕДОМОСТИ.ИД.
Фильтры (AND):
a) Н_ОЦЕНКИ.КОД = 5.
b) Н_ВЕДОМОСТИ.ИД = 1426978.
c) Н_ВЕДОМОСТИ.ИД < 1250981.
Вид соединения: INNER JOIN.

```
SELECT Н_ОЦЕНКИ.КОД, Н_ВЕДОМОСТИ.ИД

FROM Н_ОЦЕНКИ

JOIN Н_ВЕДОМОСТИ ON Н_ОЦЕНКИ.КОД = Н_ВЕДОМОСТИ.ОЦЕНКА

WHERE Н_ОЦЕНКИ.КОД = '5'

AND Н_ВЕДОМОСТИ.ИД = 1426978

AND Н_ВЕДОМОСТИ.ИД < 1250981;
```





Оптимальным является план 2, так как он фильтрует данные, а далее по выбранным атрибутам объединяет таблицы

Индексы

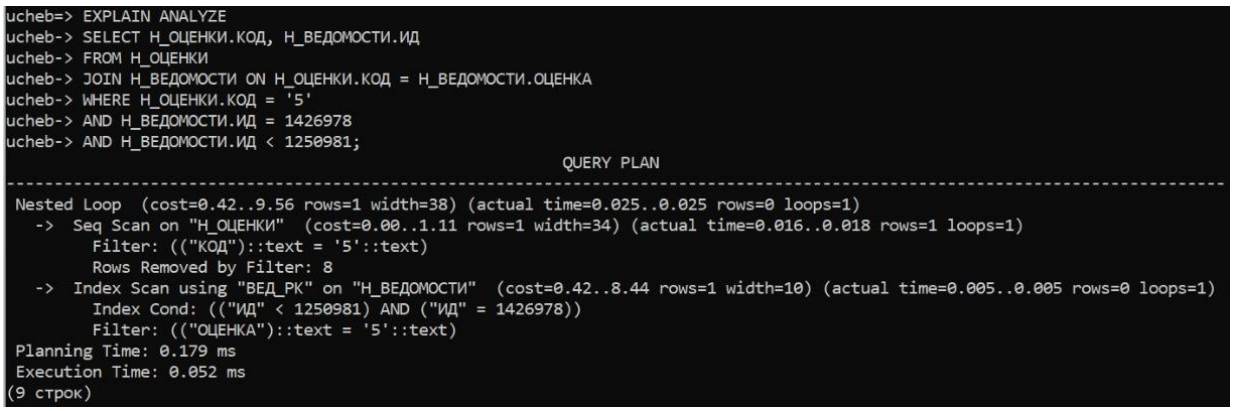
```
CREATE INDEX "ИНДЕКС_ОЦЕНКИ_КОД" ON "Н_ОЦЕНКИ" USING hash("КОД");

CREATE INDEX "ИНДЕКС_ВЕДОМОСТИ_ОЦЕНКА" ON "Н_ВЕДОМОСТИ" USING hash("ОЦЕНКА");

CREATE INDEX "ИНДЕКС_ВЕДОМОСТИ_ИД" ON "Н_ВЕДОМОСТИ" USING btree("ИД");
```

Добавление этих индексов должно ускорить выполнение запросов, так как по перечисленным полям происходит выборка с использованием оператора сравнения. Так же быстрее будет происходить соединение таблиц. В последнем случае используются операторы сравнения „>“ и „<“, так что эффективнее использовать btree. В первых **двух** случаях используется прямое сравнение, так что эффективнее использовать хэш-индекс.

При добавлении индексов планы выполнения запросов изменятся, так как будет происходить индексный скан и Hash Join станет быстрее благодаря индексам.

По итогу оба плана ускоряются, но план 2 остаётся более предпочтительным, так как минимизирует объем данных до соединения.

Результат EXPLAIN ANALYZE

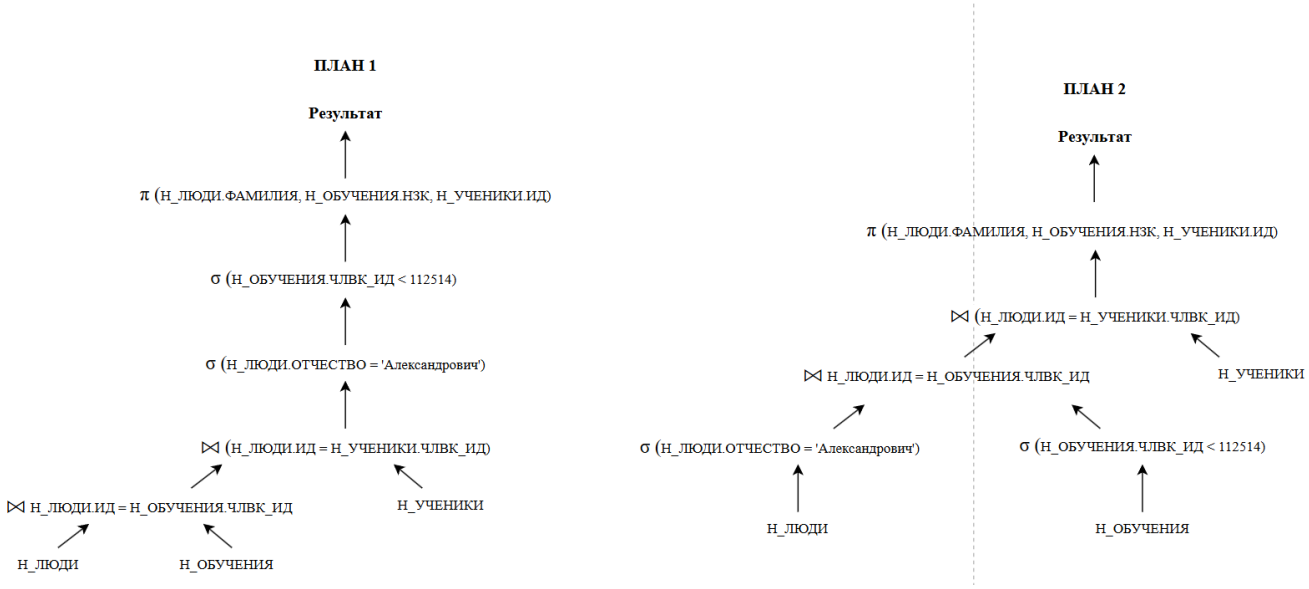


```
ucheb=> EXPLAIN ANALYZE
ucheb-> SELECT Н_ОЦЕНКИ.КОД, Н_ВЕДОМОСТИ.ИД
ucheb-> FROM Н_ОЦЕНКИ
ucheb-> JOIN Н_ВЕДОМОСТИ ON Н_ОЦЕНКИ.КОД = Н_ВЕДОМОСТИ.ОЦЕНКА
ucheb-> WHERE Н_ОЦЕНКИ.КОД = '5'
ucheb-> AND Н_ВЕДОМОСТИ.ИД = 1426978
ucheb-> AND Н_ВЕДОМОСТИ.ИД < 1250981;
                                                                  QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=0.42..9.56 rows=1 width=38) (actual time=0.025..0.025 rows=0 loops=1)
   ->  Seq Scan on "Н_ОЦЕНКИ"  (cost=0.00..1.11 rows=1 width=34) (actual time=0.016..0.018 rows=1 loops=1)
         Filter: (("КОД")::text = '5'::text)
         Rows Removed by Filter: 8
   ->  Index Scan using "ВЕД_РК" on "Н_ВЕДОМОСТИ"  (cost=0.42..8.44 rows=1 width=10) (actual time=0.005..0.005 rows=0 loops=1)
         Index Cond: (("ИД" < 1250981) AND ("ИД" = 1426978))
         Filter: (("ОЦЕНКА")::text = '5'::text)
 Planning Time: 0.179 ms
 Execution Time: 0.052 ms
(9 строк)
```

Запрос №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.НЗК, Н_УЧЕНИКИ.ИД.
Фильтры: (AND)
a) Н_ЛЮДИ.ОТЧЕСТВО = Александрович.
b) Н_ОБУЧЕНИЯ.ЧЛВК_ИД < 112514.
Вид соединения: LEFT JOIN.

```
SELECT Н_ЛЮДИ.ФАМИЛИЯ, 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_ЛЮДИ.ИД = Н_УЧЕНИКИ.ЧЛВК_ИД
WHERE Н_ЛЮДИ.ОТЧЕСТВО = 'Александрович'
AND Н_ОБУЧЕНИЯ.ЧЛВК_ИД < 112514;
```



Оптимальным является план 2, так как он фильтрует данные, а далее по выбранным атрибутам объединяет таблицы

Индексы

```
CREATE INDEX "ИНДЕКС_ЛЮДИ_ОТЧЕСТВО" ON "Н_ЛЮДИ" USING hash("ОТЧЕСТВО");
CREATE INDEX "ИНДЕКС_ОБУЧЕНИЯ_ЧЛВК_ИД" ON "Н_ОБУЧЕНИЯ" USING btree("ЧЛВК_ИД");
CREATE INDEX "ИНДЕКС_ЛЮДИ_ИД" ON "Н_ЛЮДИ" USING hash("ИД");
```

Добавление этих индексов ускоряет выполнение запросов, поскольку по перечисленным полям происходят фильтрации и соединения. Для условий с операторами сравнения (<, >) оптимален B-tree индекс.

Планы выполнения изменятся, так как вместо последовательного сканирования будут использоваться индексные сканы, а соединения станут быстрее. Особенно это заметно в плане 2, где фильтрация происходит до соединений, что минимизирует объем данных. Поэтому план 2 остаётся более оптимальным даже после добавления индексов.

Результат EXPLAIN ANALYZE

```
                                                            QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------------
Nested Loop Left Join  (cost=70.71..379.45 rows=123 width=26) (actual time=0.233..1.239 rows=115 loops=1)
  ->  Hash Join  (cost=70.42..237.18 rows=27 width=26) (actual time=0.205..1.028 rows=28 loops=1)
        Hash Cond: ("Н_ЛЮДИ"."ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД")
        ->  Seq Scan on "Н_ЛЮДИ"  (cost=0.00..163.97 rows=503 width=20) (actual time=0.016..0.784 rows=503 loops=1)
              Filter: (("ОТЧЕСТВО")::text = 'Александрович'::text)
              Rows Removed by Filter: 4615
        ->  Hash  (cost=66.93..66.93 rows=279 width=10) (actual time=0.159..0.160 rows=264 loops=1)
              Buckets: 1024  Batches: 1  Memory Usage: 19kB
              ->  Bitmap Heap Scan on "Н_ОБУЧЕНИЯ"  (cost=6.44..66.93 rows=279 width=10) (actual time=0.058..0.108 rows=264 loops=1)
                    Recheck Cond: ("ЧЛВК_ИД" < 112514)
                    Heap Blocks: exact=7
                    ->  Bitmap Index Scan on "ОБУЧ_ЧЛВК_FK_I"  (cost=0.00..6.37 rows=279 width=0) (actual time=0.036..0.036 rows=264 loops=1)
                          Index Cond: ("ЧЛВК_ИД" < 112514)
  ->  Index Scan using "УЧЕН_ОБУЧ_FK_I" on "Н_УЧЕНИКИ"  (cost=0.29..5.22 rows=5 width=8) (actual time=0.003..0.006 rows=4 loops=28)
        Index Cond: ("ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
Planning Time: 1.934 ms
Execution Time: 1.357 ms
(17 строк)
```

Вывод

Во время выполнения данной лабораторной работы я научилсась оптимизировать запросы, составлять наиболее выгодный план выполнения запросов, используя для этого подходящие виды индексов.