**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,   
МЕХАНИКИ И ОПТИКИ**

***Изображение выглядит как черный, темнота

Автоматически созданное описание***

**Факультет Программной Инженерии и Компьютерной Техники**

**Дисциплина:**

***Информационные системы и базы данных***

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ ЛАБОТЕ № 4**

**Выполнил:**

Студент гр. P33151 *Соловьев Артемий Александрович*

**Проверил:**

*Байрамова Хумай Бахруз Кызы*

Санкт-Петербург

2023г.

**Текст задания**

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\_ЛЮДИ, Н\_ВЕДОМОСТИ.  
   Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ДАТА.  
   Фильтры (AND):   
   a) Н\_ЛЮДИ.ИМЯ > Ярослав.  
   b) Н\_ВЕДОМОСТИ.ИД = 39921.  
   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.ОТЧЕСТВО, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.НАЧАЛО.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ФАМИЛИЯ > Ёлкин.  
   b) Н\_ОБУЧЕНИЯ.НЗК = 999080.  
   c) Н\_УЧЕНИКИ.ИД > 250098.  
   Вид соединения: LEFT JOIN.

**Обоснования необходимости индексов**

1. Запрос 1
   1. Необходимо создать hash-индекс для Н\_ВЕДОМОСТИ.ИД, так как это первичный ключ и к нему нужен быстрый доступ, кроме того, по нему выполняется фильтрация в.
   2. Необходимо создать b-tree-индекс для Н\_\_ЛЮДИ.ИМЯ, так как по нему выполняется фильтрация в.
   3. Необходимо создать hash-индексы для Н\_ЛЮДИ.ИД и Н\_ВЕДОМОСТИ.ЧЛВК\_ИД, так как по ним выполняется соединение.
2. Запрос 2
   1. Необходимо создать для b-tree-индекс для Н\_ЛЮДИ.ФАМИЛИЯ, hash-индекс для Н\_ОБУЧЕНИЯ.НЗК и b-tree-индекс для Н\_УЧЕНИКИ.ИД, так как по ним выполняется фильтрация.
   2. Необходимо создать hash-индексы для Н\_ЛЮДИ.ИД, Н\_УЧЕНИКИ.ЧЛВК\_ИД и Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД так как по ним выполняется соединение.

**Планирование и анализ запроса**

**Запрос 1:**

|  |  |
| --- | --- |
| *План\_1:* | *План 2:* |
| **Изображение выглядит как текст, диаграмма, линия, снимок экрана  Автоматически созданное описание** | **Изображение выглядит как текст, снимок экрана, чек, диаграмма  Автоматически созданное описание** |

Из этих планов мы выбираем первый план, так как при соединении таблиц убираются не так много строк. Поэтому есть смысл сначала применить фильтры, а потом соединить таблицы.

**Запрос 2**

|  |  |
| --- | --- |
| *План\_1:* | *План 2:* |
| **Изображение выглядит как текст, диаграмма, чек, снимок экрана  Автоматически созданное описание** | **Изображение выглядит как текст, чек, снимок экрана, диаграмма  Автоматически созданное описание** |

Из этих планов мы выбираем первый план, так как при соединении таблиц убираются не так много строк. Поэтому есть смысл сначала применить фильтры перед соединением таблиц, а выбрать только то, что нам нужно, через проекции, чтобы использовать меньше памяти в дальнейшем.

Необходимо создать hash-индекс для Н\_ОБУЧЕНИЯ.НЗК(5021 записей),

Необходимо создать hash-индекс для Н\_УЧЕНИКИ.ЧЛВК\_ИД (23311 записей), так как по ним выполняется фильтрация.

Необходимо создать hash-индексы для Н\_ЛЮДИ.ИД (5118 записей) так как по ним выполняется соединение.  
Добавлять индексы на НАЧАЛО и ОТЧЕСТВО не нужно, так как они не участвуют в фильтрации и по ним не выполняется соединение.

**Запрос 1.**

***До добавления индекса:***

|  |
| --- |
| Nested Loop (cost=1000.00..6387.63 rows=1 width=28) (actual time=21.373..23.151 rows=0 loops=1)  " Join Filter: (""Н\_ЛЮДИ"".""ИД"" = ""Н\_ВЕДОМОСТИ"".""ЧЛВК\_ИД"")"  Rows Removed by Join Filter: 1  " -> Seq Scan on ""Н\_ЛЮДИ"" (cost=0.00..163.97 rows=1 width=24) (actual time=2.240..2.619 rows=1 loops=1)"  " Filter: ((""ИМЯ"")::text > 'Ярослав'::text)"  Rows Removed by Filter: 5117  -> Gather (cost=1000.00..6223.64 rows=1 width=12) (actual time=4.656..20.525 rows=1 loops=1)  Workers Planned: 2  Workers Launched: 2  " -> Parallel Seq Scan on ""Н\_ВЕДОМОСТИ"" (cost=0.00..5223.54 rows=1 width=12) (actual time=3.532..8.205 rows=0 loops=3)"  " Filter: (""ИД"" = 39921)"  Rows Removed by Filter: 74146  Planning Time: 0.396 ms  Execution Time: 23.180 ms |

***После добавления индекса*:**

|  |
| --- |
| Nested Loop (cost=0.00..16.05 rows=1 width=28) (actual time=0.032..0.033 rows=0 loops=1)  " -> Index Scan using task11 on ""Н\_ВЕДОМОСТИ"" (cost=0.00..8.02 rows=1 width=12) (actual time=0.017..0.018 rows=1 loops=1)"  " Index Cond: (""ИД"" = 39921)"  " -> Index Scan using task13 on ""Н\_ЛЮДИ"" (cost=0.00..8.02 rows=1 width=24) (actual time=0.011..0.011 rows=0 loops=1)"  " Index Cond: (""ИД"" = ""Н\_ВЕДОМОСТИ"".""ЧЛВК\_ИД"")"  " Filter: ((""ИМЯ"")::text > 'Ярослав'::text)"  Rows Removed by Filter: 1  Planning Time: 0.533 ms  Execution Time: 0.067 ms |

**Запрос 2.**

***До добавления индекса*:**

|  |
| --- |
| Nested Loop (cost=119.78..1130.30 rows=1 width=34) (actual time=6.678..6.680 rows=0 loops=1)  Join Filter: (""Н\_ОБУЧЕНИЯ"".""ЧЛВК\_ИД"" = ""Н\_УЧЕНИКИ"".""ЧЛВК\_ИД"")"  -> Hash Join (cost=119.78..297.89 rows=1 width=34) (actual time=0.846..3.720 rows=1 loops=1)  " Hash Cond: (""Н\_ЛЮДИ"".""ИД"" = ""Н\_ОБУЧЕНИЯ"".""ЧЛВК\_ИД"")"  " -> Seq Scan on ""Н\_ЛЮДИ"" (cost=0.00..163.97 rows=3767 width=24) (actual time=0.017..2.771 rows=3778 loops=1)"  " Filter: ((""ФАМИЛИЯ"")::text > 'Ёлкин'::text)"  Rows Removed by Filter: 1340  -> Hash (cost=119.76..119.76 rows=1 width=10) (actual time=0.610..0.611 rows=1 loops=1)  Buckets: 1024 Batches: 1 Memory Usage: 9kB  " -> Seq Scan on ""Н\_ОБУЧЕНИЯ"" (cost=0.00..119.76 rows=1 width=10) (actual time=0.130..0.600 rows=1 loops=1)"  " Filter: ((""НЗК"")::text = '999080'::text)"  Rows Removed by Filter: 5020  " -> Seq Scan on ""Н\_УЧЕНИКИ"" (cost=0.00..832.39 rows=2 width=12) (actual time=2.956..2.956 rows=0 loops=1)"  " Filter: (""ИД"" > 250098)"  Rows Removed by Filter: 23311  Planning Time: 0.377 ms  Execution Time: 6.735 ms |

***После добавления индекса*:**

|  |
| --- |
| Nested Loop (cost=0.00..16.87 rows=1 width=34) (actual time=0.044..0.045 rows=0 loops=1)  " Join Filter: (""Н\_ОБУЧЕНИЯ"".""ЧЛВК\_ИД"" = ""Н\_УЧЕНИКИ"".""ЧЛВК\_ИД"")"  -> Nested Loop (cost=0.00..16.05 rows=1 width=34) (actual time=0.026..0.028 rows=1 loops=1)  " -> Index Scan using task26 on ""Н\_ОБУЧЕНИЯ"" (cost=0.00..8.02 rows=1 width=10) (actual time=0.011..0.012 rows=1 loops=1)"  " Index Cond: ((""НЗК"")::text = '999080'::text)"  " -> Index Scan using task25 on ""Н\_ЛЮДИ"" (cost=0.00..8.02 rows=1 width=24) (actual time=0.011..0.011 rows=1 loops=1)"  " Index Cond: (""ИД"" = ""Н\_ОБУЧЕНИЯ"".""ЧЛВК\_ИД"")"  " Filter: ((""ФАМИЛИЯ"")::text > 'Ёлкин'::text)"  " -> Index Scan using task23 on ""Н\_УЧЕНИКИ"" (cost=0.00..0.81 rows=1 width=12) (actual time=0.016..0.016 rows=0 loops=1)"  " Index Cond: (""ЧЛВК\_ИД"" = ""Н\_ЛЮДИ"".""ИД"")"  " Filter: (""ИД"" > 250098)"  Rows Removed by Filter: 6  Planning Time: 0.708 ms  Execution Time: 0.091 ms |

**Исходный код.**

-- task 1  
explain analyse SELECT "Н\_ЛЮДИ"."ОТЧЕСТВО", "Н\_ВЕДОМОСТИ"."ДАТА" FROM "Н\_ЛЮДИ"  
RIGHT JOIN "Н\_ВЕДОМОСТИ" on "Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"  
WHERE "Н\_ЛЮДИ"."ИМЯ" > 'Ярослав' and "Н\_ВЕДОМОСТИ"."ИД"= 39921;  
  
create index task11 on "Н\_ВЕДОМОСТИ" USING hash("ИД") ;  
create index task12 on "Н\_ВЕДОМОСТИ" USING hash("ЧЛВК\_ИД");  
create index task13 on "Н\_ЛЮДИ" using hash("ИД");  
create index task14 on "Н\_ЛЮДИ" using btree("ИМЯ");  
  
-- task 1 index  
explain analyse SELECT "Н\_ЛЮДИ"."ОТЧЕСТВО", "Н\_ВЕДОМОСТИ"."ДАТА" FROM "Н\_ЛЮДИ"  
RIGHT JOIN "Н\_ВЕДОМОСТИ" on "Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"  
WHERE "Н\_ЛЮДИ"."ИМЯ" > 'Ярослав' and "Н\_ВЕДОМОСТИ"."ИД"= 39921;  
  
drop index task11;  
drop index task12;  
drop index task13;  
drop index task14;  
  
-- task 2  
explain analyse SELECT "Н\_ЛЮДИ"."ОТЧЕСТВО", "Н\_ОБУЧЕНИЯ"."НЗК", "Н\_УЧЕНИКИ"."НАЧАЛО" FROM "Н\_УЧЕНИКИ"  
LEFT JOIN "Н\_ОБУЧЕНИЯ" on "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД"  
LEFT JOIN "Н\_ЛЮДИ" on "Н\_ЛЮДИ"."ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД"  
WHERE "Н\_ЛЮДИ"."ФАМИЛИЯ" > 'Ёлкин' and "Н\_ОБУЧЕНИЯ"."НЗК" = '999080' and "Н\_УЧЕНИКИ"."ИД" > 250098;  
  
create index task21 on "Н\_УЧЕНИКИ" using hash("ИД");  
create index task22 on "Н\_ОБУЧЕНИЯ" using hash("ЧЛВК\_ИД");  
create index task23 on "Н\_УЧЕНИКИ" using hash("ЧЛВК\_ИД");  
create index task24 on "Н\_ЛЮДИ" using btree("ФАМИЛИЯ");  
create index task25 on "Н\_ЛЮДИ" using hash("ИД");  
create index task26 on "Н\_ОБУЧЕНИЯ" using hash("НЗК");  
  
--task 2 index2  
explain analyse SELECT "Н\_ЛЮДИ"."ОТЧЕСТВО", "Н\_ОБУЧЕНИЯ"."НЗК", "Н\_УЧЕНИКИ"."НАЧАЛО" FROM "Н\_УЧЕНИКИ"  
LEFT JOIN "Н\_ОБУЧЕНИЯ" on "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД"  
LEFT JOIN "Н\_ЛЮДИ" on "Н\_ЛЮДИ"."ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД"  
WHERE "Н\_ЛЮДИ"."ФАМИЛИЯ" > 'Ёлкин' and "Н\_ОБУЧЕНИЯ"."НЗК" = '999080' and "Н\_УЧЕНИКИ"."ИД" > 250098;  
  
drop index task21;  
drop index task22;  
drop index task23;  
drop index task24;  
drop index task25;  
drop index task26;

**Вывод.**

В ходе выполнения лабораторной работы было изучены и применены навыки составления плана запроса и добавления индексов, что сильно уменьшает время обработки запроса.