Национальный исследовательский университет ИТМО

Факультет Программной Инженерии и Компьютерной техники

Лабораторная работа №4
Вариант №3156

Выполнил:
Родионов Максим Артемович
Группа P3131
Преподаватели:
Вербовой Александр Александрович

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_ЛЮДИ, H_СЕССИЯ.
   Вывести атрибуты: H_ЛЮДИ.ФАМИЛИЯ, H_СЕССИЯ.УЧГОД.
   Фильтры (AND):
   a) H_ЛЮДИ.ФАМИЛИЯ > Соколов.
   b) H_СЕССИЯ.ЧЛВК_ИД > 100622.
   Вид соединения: RIGHT JOIN.
2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
   Таблицы: H_ЛЮДИ, H_ВЕДОМОСТИ, H_СЕССИЯ.
   Вывести атрибуты: H_ЛЮДИ.ФАМИЛИЯ, H_ВЕДОМОСТИ.ИД, H_СЕССИЯ.ДАТА.
   Фильтры (AND):
   a) H_ЛЮДИ.ОТЧЕСТВО < Александрович.
   b) H_ВЕДОМОСТИ.ДАТА < 2010-06-18.
   c) H_СЕССИЯ.УЧГОД = 2008/2009.
   Вид соединения: RIGHT JOIN.

Запрос №1:

Таблицы: Н_ЛЮДИ, Н_СЕССИЯ.

Вывести атрибуты: Н_ЛЮДИ.ФАМИЛИЯ, Н_СЕССИЯ.УЧГОД.

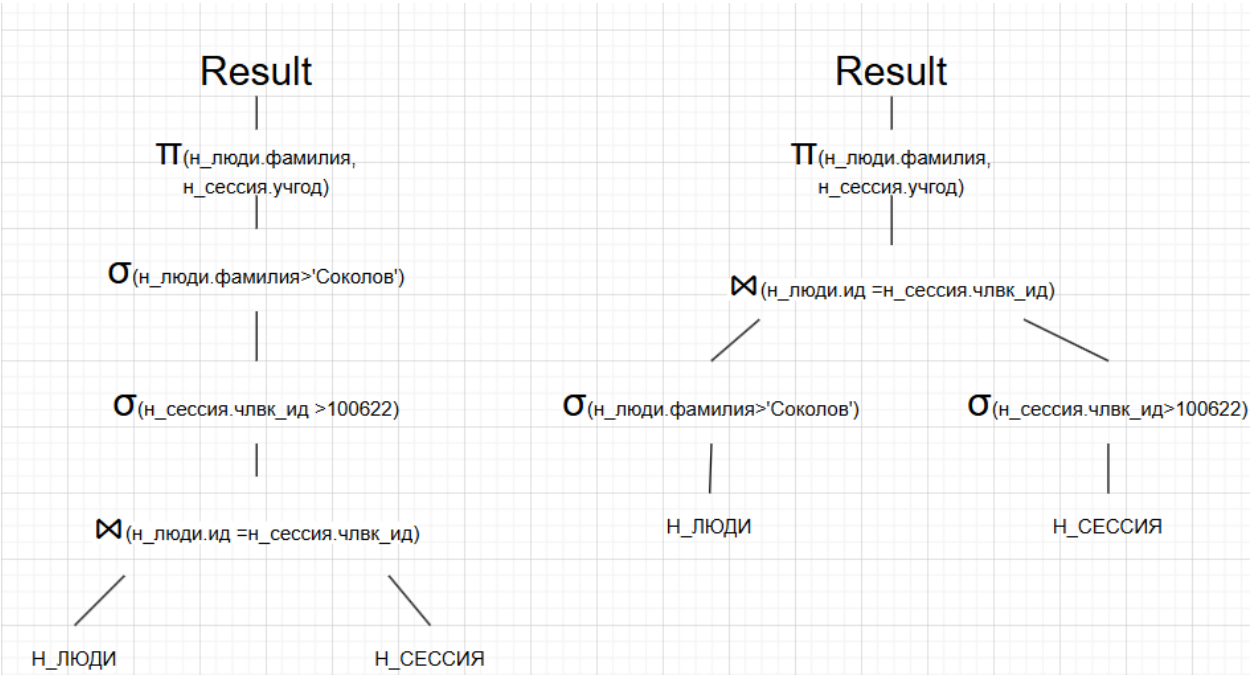
Фильтры (AND):

a) Н_ЛЮДИ.ФАМИЛИЯ > Соколов.

b) Н_СЕССИЯ.ЧЛВК_ИД > 100622.

Вид соединения: RIGHT JOIN.

```
SELECT Н_ЛЮДИ.ФАМИЛИЯ, Н_СЕССИЯ.УЧГОД
FROM Н_ЛЮДИ
RIGHT JOIN Н_СЕССИЯ ON Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД
WHERE Н_ЛЮДИ.ФАМИЛИЯ > 'Соколов'
  AND Н_СЕССИЯ.ЧЛВК_ИД > 100622;
```



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

Индексы:

```
CREATE INDEX idx_сессия_члвк_ид ON Н_СЕССИЯ USING btree(ЧЛВК_ИД);
CREATE INDEX idx_люди_фамилия ON Н_ЛЮДИ USING btree(ФАМИЛИЯ);
CREATE INDEX idx_люди_ид ON Н_ЛЮДИ USING btree(ИД);
```

Добавление предложенных индексов должно ускорить выполнение запроса, так как выборка происходит по полям **ЧЛВК_ИД** и **ФАМИЛИЯ** с использованием опера-

тора сравнения >. Кроме того, соединение таблиц **Н_ЛЮДИ** и **Н_СЕССИЯ** осуществляется по полям **ИД** и **ЧЛВК_ИД**, для которых также создаются индексы.

Для всех этих случаев оптимальным типом индекса является **btree**, поскольку он поддерживает упорядоченные сравнения (>, <, BETWEEN), в отличие от hash, который работает только с точным равенством. Использование btree-индексов позволяет СУБД применять индексный скан, а при соединении таблиц может быть выбран более эффективный план, например Hash Join или Merge Join, в зависимости от статистики.

Таким образом, добавление индексов уменьшает количество строк, обрабатываемых при соединении и фильтрации, что существенно снижает общее время выполнения запроса.

.

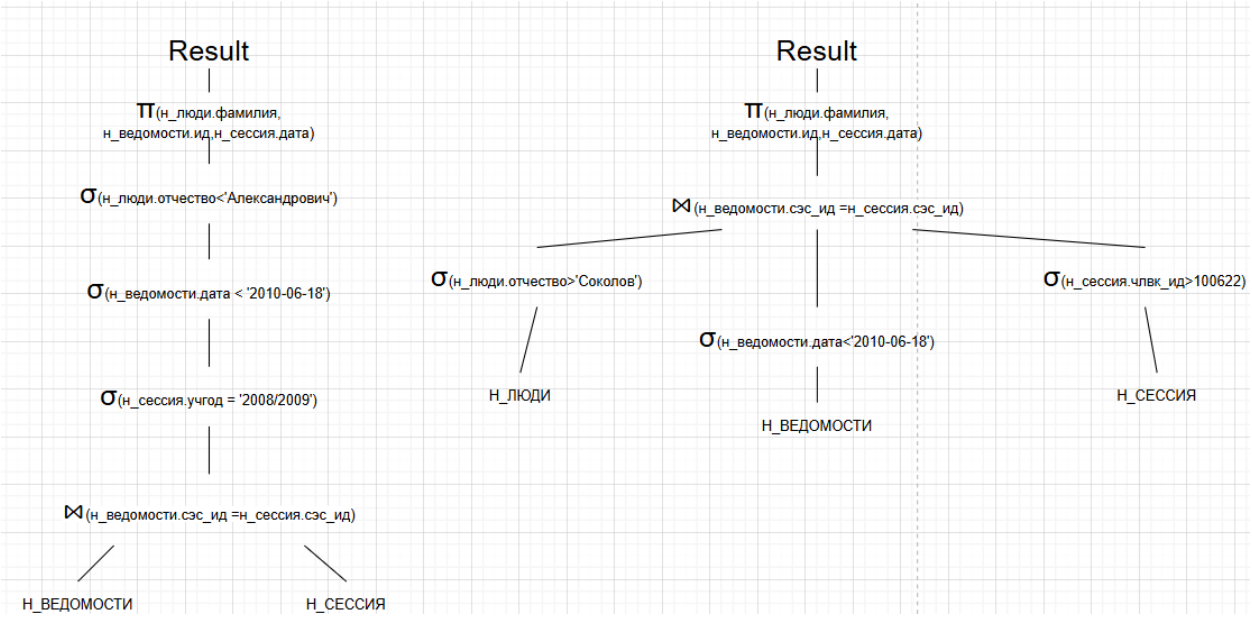
Результат EXPLAIN ANALYSE:

```
Nested Loop  (cost=0.29..264.97 rows=382 width=26) (actual time=0.053..4.001 rows=542 loops=1)
  ->  Seq Scan on "Н_СЕССИЯ"  (cost=0.00..117.90 rows=2078 width=14) (actual time=0.014..1.143 rows=2079 loops=1)
        Filter: ("ЧЛВК_ИД" > 100622)
        Rows Removed by Filter: 1673
  ->  Memoize  (cost=0.29..0.54 rows=1 width=20) (actual time=0.001..0.001 rows=0 loops=2079)
        Cache Key: "Н_СЕССИЯ"."ЧЛВК_ИД"
        Cache Mode: logical
        Hits: 1924  Misses: 155  Evictions: 0  Overflows: 0  Memory Usage: 12kB
        ->  Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ"  (cost=0.28..0.53 rows=1 width=20) (actual time=0.007..0.007 rows=0 loops=155)
              Index Cond: ("ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД")
              Filter: (("ФАМИЛИЯ")::text > 'Соколов'::text)
              Rows Removed by Filter: 1
Planning Time: 1.896 ms
Execution Time: 4.164 ms
```

Запрос №2:

Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ, Н_СЕССИЯ.
Вывести атрибуты: Н_ЛЮДИ.ФАМИЛИЯ, Н_ВЕДОМОСТИ.ИД, Н_СЕССИЯ.ДАТА.
Фильтры (AND):
a) Н_ЛЮДИ.ОТЧЕСТВО < Александрович.
b) Н_ВЕДОМОСТИ.ДАТА < 2010-06-18.
c) Н_СЕССИЯ.УЧГОД = 2008/2009.
Вид соединения: RIGHT JOIN.

```
SELECT Н_ЛЮДИ.ФАМИЛИЯ, Н_ВЕДОМОСТИ.ИД, Н_СЕССИЯ.ДАТА
FROM Н_ЛЮДИ
RIGHT JOIN Н_СЕССИЯ ON Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД
JOIN Н_ВЕДОМОСТИ ON Н_СЕССИЯ.СЭС_ИД = Н_ВЕДОМОСТИ.СЭС_ИД
WHERE Н_ЛЮДИ.ОТЧЕСТВО < 'Александрович'
  AND Н_ВЕДОМОСТИ.ДАТА < DATE '2010-06-18'
  AND Н_СЕССИЯ.УЧГОД = '2008/2009';
```



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

Индексы:

```
CREATE INDEX idx_сессия_члвк_ид ON Н_СЕССИЯ USING btree(ЧЛВК_ИД);
CREATE INDEX idx_люди_ид ON Н_ЛЮДИ USING btree(ИД);
CREATE INDEX idx_сессия_сэс_ид ON Н_СЕССИЯ USING btree(СЭС_ИД);
CREATE INDEX idx_ведомости_сэс_ид ON Н_ВЕДОМОСТИ USING btree(СЭС_ИД);
CREATE INDEX idx_люди_отчество ON Н_ЛЮДИ USING btree(ОТЧЕСТВО);
CREATE INDEX idx_ведомости_дата ON Н_ВЕДОМОСТИ USING btree(ДАТА);
CREATE INDEX idx_сессия_учгод ON Н_СЕССИЯ USING btree(УЧГОД);
```

Добавление этих индексов позволяет оптимизировать как соединения между таблицами, так и выполнение фильтрации. В частности:

Индексы по полям **ИД**, **ЧЛВК_ИД**, **СЭС_ИД** ускоряют операции **JOIN** между таблицами **Н_ЛЮДИ**, **Н_СЕССИЯ** и **Н_ВЕДОМОСТИ**.

Поля **ОТЧЕСТВО**, **ДАТА** и **УЧГОД** участвуют в условиях фильтрации с операторами сравнения (<, =), поэтому наиболее эффективно использовать индексы типа **btree**, так как они поддерживают упорядоченные операции и диапазонные запросы.

Использование этих индексов позволяет выполнять **индексный скан** вместо последовательного, что уменьшает число обрабатываемых строк и снижает общее время выполнения запроса.

При наличии индексов план запроса может измениться: например, вместо **Nested Loop** может быть выбран **Hash Join** или **Merge Join**, что также способствует ускорению выполнения.

Таким образом, предложенные индексы соответствуют фильтрам и условиям соединения в запросе и позволяют значительно сократить ресурсы, затрачиваемые на его выполнение.

Результат EXPLAIN ANALYSE:

```
Nested Loop  (cost=167.73..938.23 rows=628 width=28) (actual time=3.399..4.841 rows=1261 loops=1)
  ->  Hash Join  (cost=167.44..286.45 rows=20 width=28) (actual time=3.353..3.979 rows=32 loops=1)
        Hash Cond: ("Н_СЕССИЯ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
        ->  Seq Scan on "Н_СЕССИЯ"  (cost=0.00..117.90 rows=424 width=16) (actual time=0.046..0.757
rows=424 loops=1)
              Filter: (("УЧГОД")::text = '2008/2009'::text)
              Rows Removed by Filter: 3328
        ->  Hash  (cost=163.97..163.97 rows=277 width=20) (actual time=3.146..3.146 rows=278 loops=1)
              Buckets: 1024  Batches: 1  Memory Usage: 23kB
              ->  Seq Scan on "Н_ЛЮДИ"  (cost=0.00..163.97 rows=277 width=20) (actual time=0.006..3.072
rows=278 loops=1)
                    Filter: (("ОТЧЕСТВО")::text < 'Александрович'::text)
                    Rows Removed by Filter: 4840
  ->  Index Scan using "ВЕД_ИП_FK_I" on "Н_ВЕДОМОСТИ"  (cost=0.29..32.29 rows=30 width=8) (actual
time=0.006..0.021 rows=39 loops=32)
```

```
      Index Cond: ("СЭС_ИД" = "Н_СЕССИЯ"."СЭС_ИД")
      Filter: ("ДАТА" < '2010-06-18'::date)
      Rows Removed by Filter: 1
Planning Time: 1.918 ms
Execution Time: 4.985 ms
(17 строк) Planning Time: 0.412 ms
Execution Time: 0.102 ms
```

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