ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки: Информатика и вычислительная техника

Дисциплина «Информационные системы и базы данных»

Лабораторная работа №4

Вариант 2810

Выполнил:

Богатов А. С.

P33302

Преподаватель

Гаврилов А. В.

г. Санкт-Петербург, 2022 г.

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?

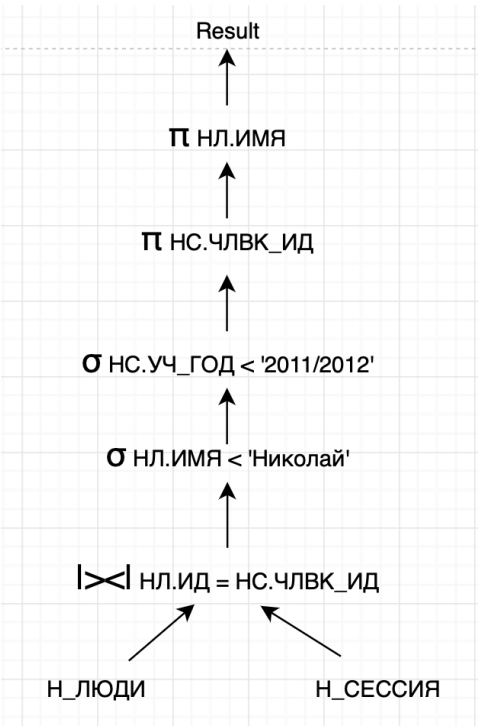
Для запросов 1-2 необходимо добавить в отчет вывод команды EXPLAIN ANALYZE [запрос]

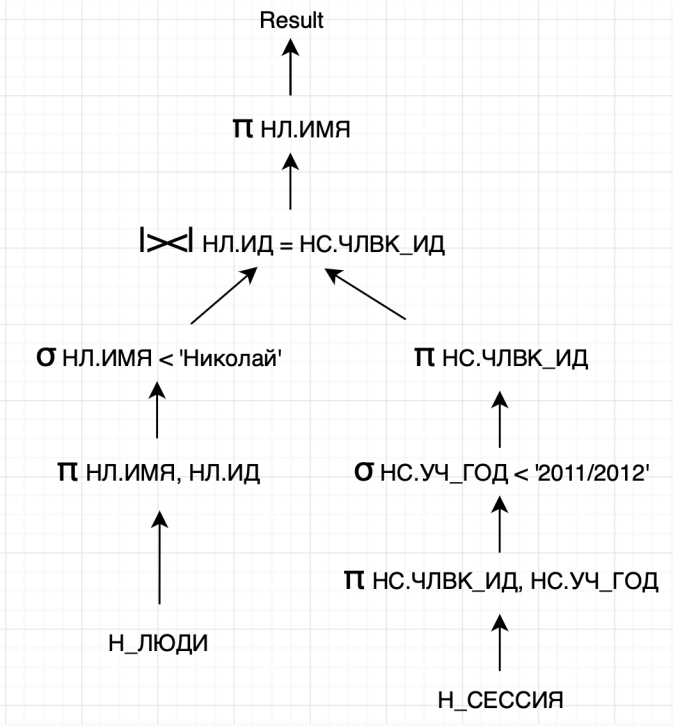
Подробные ответы на все вышеперечисленные вопросы должны присутствовать в отчете (планы выполнения запросов должны быть нарисованы, ответы на вопросы - представлены в текстовом виде).

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
   Таблицы: H_ЛЮДИ, H_СЕССИЯ.
   Вывести атрибуты: H_ЛЮДИ.ИМЯ, H_СЕССИЯ.ЧЛВК_ИД.
   Фильтры (AND):
   a) H_ЛЮДИ.ИМЯ < Николай.
   b) H_СЕССИЯ.УЧГОД < 2011/2012.
   Вид соединения: LEFT 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.ИД, H_ВЕДОМОСТИ.ИД, H_СЕССИЯ.ДАТА.
   Фильтры (AND):
   a) H_ЛЮДИ.ИМЯ = Ярослав.
   b) H_ВЕДОМОСТИ.ДАТА > 2022-06-08.
   c) H_СЕССИЯ.ДАТА = 2002-01-04.
   Вид соединения: INNER JOIN.

SQL запросы:

1. select НЛ.ИМЯ, НС.ЧЛВК_ИД from Н_ЛЮДИ НЛ left join Н_СЕССИЯ НС on НЛ.ИД = НС.ЧЛВК_ИД where НЛ.ИМЯ < 'Николай' and НС.УЧГОД < '2011/2012';





Во втором плане мы выбираем только необходимые на каждом этапе атрибуты => минимизируем количество временно хранимых данных => снижаем количество операций по чтению/записи из/в памяти.

Если данный запрос является популярным, оптимально ввести следующие индексы:

CREATE INDEX НС_ГОД ON "Н_СЕССИЯ" USING btree("УЧГОД");

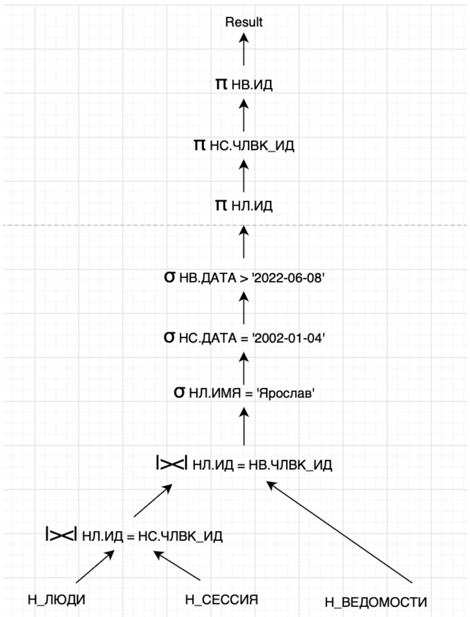
CREATE INDEX НЛ_ИМЯ ON "Н_ЛЮДИ" USING btree("ИМЯ");

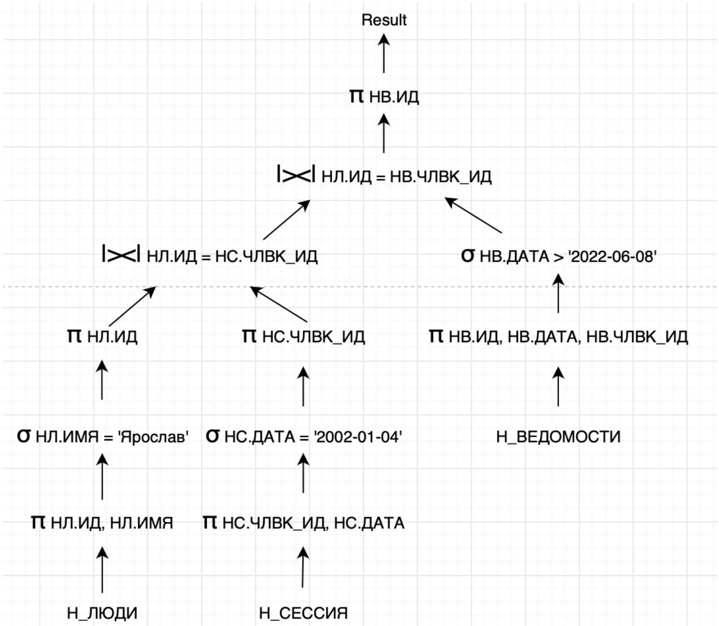
CREATE INDEX НС_ЧЛВК ON "Н_СЕССИЯ" USING hash("ЧЛВК_ИД");

Для атрибутов участвующих в выборке по сравнению оптимальнее использовать индексирование методом сбалансированного дерева. При соединении проверяем точное соответствие идентификаторов, для главных ключей происходит автоматическое индексирование, для внешних ключей оптимально вводить индексирование методом хеширования.

```
ucheb=> EXPLAIN ANALYZE select НЛ.ИМЯ, НС.ЧЛВК_ИД from Н_ЛЮДИ НЛ left join Н_СЕССИЯ НС on НЛ.ИД = НС.ЧЛВК_ИД where НЛ.ИМЯ < 'Николай' and НС.УЧГОД < '2011/2012';
                                                              QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=0.29..284.88 rows=2334 width=17) (actual time=0.039..3.912 rows=2238 loops=1)
   ->  Seq Scan on "Н_СЕССИЯ" "НС"  (cost=0.00..117.90 rows=3543 width=4) (actual time=0.013..1.868 rows=3543 loops=1)
         Filter: (("УЧГОД")::text < '2011/2012'::text)
         Rows Removed by Filter: 209
   ->  Memoize  (cost=0.29..0.44 rows=1 width=17) (actual time=0.000..0.000 rows=1 loops=3543)
         Cache Key: "НС"."ЧЛВК_ИД"
         Cache Mode: logical
         Hits: 3367  Misses: 176  Evictions: 0  Overflows: 0  Memory Usage: 19kB
         ->  Index Scan using "ЧЛВК_PK" on "Н_ЛЮДИ" "НЛ"  (cost=0.28..0.43 rows=1 width=17) (actual time=0.003..0.003 rows=1 loops=176)
               Index Cond: ("ИД" = "НС"."ЧЛВК_ИД")
               Filter: (("ИМЯ")::text < 'Николай'::text)
               Rows Removed by Filter: 0
 Planning Time: 0.434 ms
 Execution Time: 4.084 ms
(14 строк)
```

2. select НЛ.ИД, НВ.ИД, НС.ДАТА from Н_ЛЮДИ НЛ join Н_СЕССИЯ НС on НЛ.ИД = НС.ЧЛВК_ИД join Н_ВЕДОМОСТИ НВ on НЛ.ИД = НВ.ЧЛВК_ИД where НЛ.ИМЯ = 'Ярослав' and НВ.ДАТА > '2022-06-08' and НС.ДАТА = '2002-01-04';





Во втором плане мы выбираем только необходимые на каждом этапе атрибуты => минимизируем количество временно хранимых данных => снижаем количество операций по чтению/записи из/в памяти.

Если данный запрос является популярным, оптимально ввести следующие индексы:

CREATE INDEX НВ_ДАТА ON "Н_ВЕДОМОСТИ" USING btree("ДАТА");

CREATE INDEX НС_ДАТА ON "Н_СЕССИЯ" USING hash("ДАТА");

CREATE INDEX НЛ_ИМЯ ON "Н_ЛЮДИ" USING hash("ИМЯ");

CREATE INDEX НС_ЧЛВК ON "Н_СЕССИЯ" USING hash("ЧЛВК_ИД");

CREATE INDEX НВ_ЧЛВК ON "Н_ВЕДОМОСТИ" USING hash("ЧЛВК_ИД");

Для атрибутов участвующих в выборке по сравнению оптимальнее использовать индексирование методом сбалансированного дерева. При соединении проверяем точное соответствие идентификаторов, для главных ключей происходит автоматическое индексирование, для внешних ключей оптимально вводить индексирование методом хеширования.

```
ucheb=> EXPLAIN ANALYZE select НЛ.ИД, НВ.ИД, НС.ДАТА from Н_ЛЮДИ НЛ join Н_СЕССИЯ НС on НЛ.ИД = НС.ЧЛВК_ИД join Н_ВЕДОМОСТИ НВ on НЛ.ИД = НВ.ЧЛВК_ИД where НЛ.ИМЯ = 'Ярослав' and НВ.ДАТА > '2022-06-08' and
 НС.ДАТА = '2002-01-04';
                                                       QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=2.40..34.52 rows=1 width=16) (actual time=0.006..0.007 rows=0 loops=1)
   ->  Nested Loop  (cost=0.58..16.39 rows=1 width=12) (actual time=0.006..0.007 rows=0 loops=1)
         ->  Index Scan using "ВЕД_ДАТА_I" on "Н_ВЕДОМОСТИ" "НВ"  (cost=0.29..7.20 rows=1 width=8) (actual time=0.005..0.006 rows=0 loops=1)
               Index Cond: ("ДАТА" > '2022-06-08 00:00:00'::timestamp without time zone)
         ->  Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" "НЛ"  (cost=0.28..8.30 rows=1 width=4) (never executed)
               Index Cond: ("ИД" = "НВ"."ЧЛВК_ИД")
               Filter: (("ИМЯ")::text = 'Ярослав'::text)
   ->  Bitmap Heap Scan on "Н_СЕССИЯ" "НС"  (cost=1.83..18.12 rows=1 width=12) (never executed)
         Recheck Cond: ("ЧЛВК_ИД" = "НЛ"."ИД")
         Filter: ("ДАТА" = '2002-01-04 00:00:00'::timestamp without time zone)
         ->  Bitmap Index Scan on "SYS_C003500_IFK"  (cost=0.00..1.83 rows=18 width=0) (never executed)
               Index Cond: ("ЧЛВК_ИД" = "НЛ"."ИД")
 Planning Time: 0.645 ms
 Execution Time: 0.075 ms
(14 строк)
```

NOTE: Планы запросов выведенные EXPLAIN ANALYZE учитывают индексы в общей базе данных, наличие индексов на как таковую структуру плана не влияет, однако стандартное сканирование в плане заменяется на сканирование по индексу. Сканирование по индексу существенно быстрее при выборке по одному столбцу.

Вывод:

При выполнении данной лабораторной работы был изучен принцип исполнения запросов к базе данных, принципы построения оптимального плана и влияние индексирования на план исполнения.