

Федеральное государственное автономное образовательное учреждение
высшего образования “Национальный исследовательский университет
ИТМО”
Факультет программной инженерии и компьютерной техники

Отчет по лабораторной работе №4
по дисциплине «Базы данных»
Вариант № 473781

Выполнила: Шевченко Дарья Павловна,
группа Р3130
Проверил: Барсуков Илья Александрович

Санкт-Петербург - 2023

1. 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. Изменяются ли планы при добавлении индекса и как?

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

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
Н_ТИПЫ_ВЕДОМОСТЕЙ, Н_ВЕДОМОСТИ.
Вывести атрибуты: Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ,
Н_ВЕДОМОСТИ.ИД.
Фильтры (AND):
а) Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД > 2.
б) Н_ВЕДОМОСТИ.ИД > 1250981.
Вид соединения: INNER JOIN.

```
SELECT "Н_ТИПЫ_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ", "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД"  
FROM "Н_ТИПЫ_ВЕДОМОСТЕЙ"  
      INNER JOIN "Н_ВЕДОМОСТИ" ON "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД" =  
      "Н_ВЕДОМОСТИ"."ТВ_ИД"  
WHERE "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД" > 2  
      AND "Н_ВЕДОМОСТИ"."ИД" > 1250981;
```

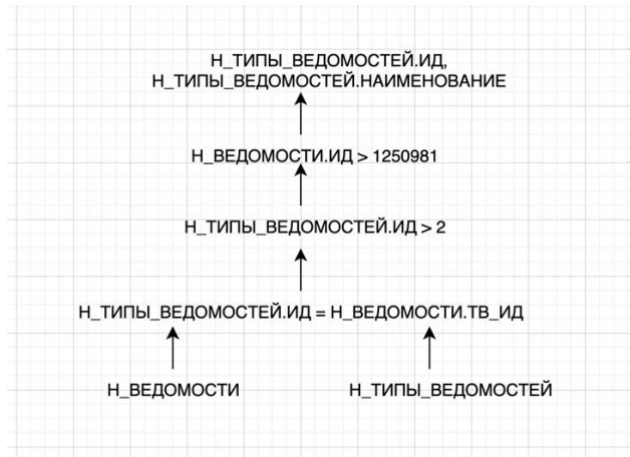
Индексы, добавление которых уменьшит время выполнения запроса

При использовании операторов сравнения оптимально использование BTREE, а при соединении таблиц с ТВ_ИД оптимально использование HASH

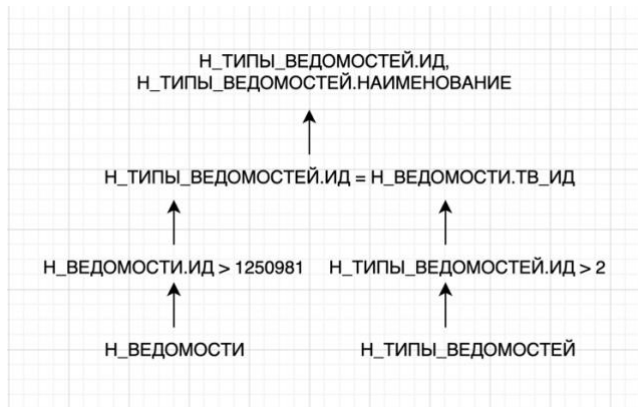
Добавление будет полезным, так как ускорит поиск нужных записей для фильтров в запросе и ускорит соединение таблиц по атрибуту ТВ_ИД

```
CREATE INDEX ON "Н_ТИПЫ_ВЕДОМОСТЕЙ" USING BTREE ("ИД");  
CREATE INDEX ON "Н_ВЕДОМОСТИ" USING BTREE ("ИД");  
CREATE INDEX ON "Н_ВЕДОМОСТИ" USING HASH ("ТВ_ИД");
```

План 1



План 2



План 2 является оптимальным, так как фильтрация данных происходит до соединения таблиц, соответственно уменьшается количество строк для обработки

EXPLAIN ANALYZE

```

QUERY PLAN
Hash Join (cost=491.19..5047.40 rows=8676 width=422) (actual time=1.368..9.044 rows=3268 loops=1)
  Hash Cond: ("Н_ВЕДОМОСТИ"."ТВ_ИД" = "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД")
    -> Bitmap Heap Scan on "Н_ВЕДОМОСТИ" (cost=490.14..4881.51 rows=26029 width=4) (actual time=1.297..4.982 rows=26125 loops=1)
      Recheck Cond: ("ИД" > 1250981)
      Heap Blocks: exact=642
      -> Bitmap Index Scan on "ВЕД_ПК" (cost=0.00..483.64 rows=26029 width=0) (actual time=1.214..1.214 rows=26125 loops=1)
        Index Cond: ("ИД" > 1250981)
    -> Hash (cost=1.04..1.04 rows=1 width=422) (actual time=0.027..0.029 rows=1 loops=1)
      Buckets: 1024 Batches: 1 Memory Usage: 9kB
      -> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=422) (actual time=0.014..0.015 rows=1 loops=1)
        Filter: ("ИД" > 2)
        Rows Removed by Filter: 2
Planning Time: 0.240 ms
Execution Time: 9.264 ms
  
```

2 запрос

Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ, Н_СЕССИЯ.

Вывести атрибуты: Н_ЛЮДИ.ФАМИЛИЯ, Н_ВЕДОМОСТИ.ЧЛВК_ИД, Н_СЕССИЯ.УЧГОД.

Фильтры (AND):

- Н_ЛЮДИ.ОТЧЕСТВО = Сергеевич.
- Н_ВЕДОМОСТИ.ИД = 39921.

с) Н_СЕССИЯ.ДАТА = 2002-01-04.

Вид соединения: LEFT JOIN.

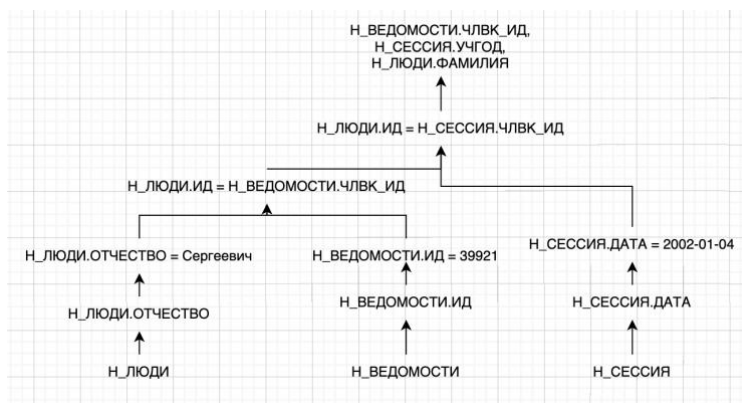
```
SELECT "Н_ЛЮДИ"."ФАМИЛИЯ", "Н_ВЕДОМОСТИ"."ЧЛВК_ИД", "Н_СЕССИЯ"."УЧГОД"  
FROM "Н_ЛЮДИ"  
      LEFT JOIN "Н_ВЕДОМОСТИ" ON "Н_ЛЮДИ"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"  
      LEFT JOIN "Н_СЕССИЯ" ON "Н_ЛЮДИ"."ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД"  
WHERE "Н_ЛЮДИ"."ОТЧЕСТВО" = 'Сергеевич'  
      AND "Н_ВЕДОМОСТИ"."ИД" = 39921  
      AND "Н_СЕССИЯ"."ДАТА" = '2002-01-04';
```

Индексы, добавление которых уменьшит время выполнения запроса

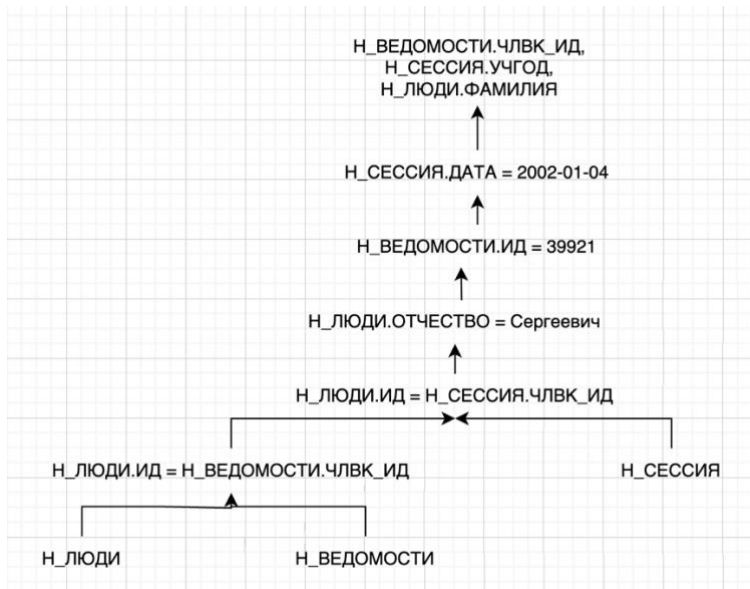
При использовании операторов сравнения оптимально использование BTREE

```
CREATE INDEX ON "Н_ЛЮДИ" USING BTREE ("ОТЧЕСТВО");  
CREATE INDEX ON "Н_СЕССИЯ" USING BTREE ("ДАТА");  
CREATE INDEX ON "Н_ВЕДОМОСТИ" USING BTREE ("ИД");
```

План 1



План 2



Оптимальным является план 1, так как фильтрация происходит на более ранних этапах, объединяются только нужные нам атрибуты, то есть размер промежуточных данных меньше.

EXPLAIN ANALYZE

	🔍 QUERY PLAN
1	Nested Loop (cost=0.98..18.21 rows=1 width=30) (actual time=0.045..0.046 rows=0 loops=1)
2	-> Nested Loop (cost=0.70..16.78 rows=1 width=24) (actual time=0.045..0.046 rows=0 loops=1)
3	-> Index Scan using "ВЕД_ПК" on "Н_ВЕДОМОСТИ" (cost=0.42..8.44 rows=1 width=4) (actual time=0.035..0.035 rows=1 loops=1)
4	Index Cond: ("ИД" = 39921)
5	-> Index Scan using "ЧЛВК_ПК" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1 width=20) (actual time=0.007..0.007 rows=0 loops=1)
6	Index Cond: ("ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД")
7	Filter: (("ОТЧЕСТВО")::text = 'Сергеевич'::text)
8	Rows Removed by Filter: 1
9	-> Index Scan using "SYS_C003500_IFK" on "Н_СЕССИЯ" (cost=0.28..1.42 rows=1 width=14) (never executed)
10	Index Cond: ("ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
11	Filter: ("ДАТА" = '2002-01-04 00:00:00'::timestamp without time zone)
12	Planning Time: 1.686 ms
13	Execution Time: 0.116 ms

Вывод

В ходе выполнения лабораторной работы я узнала, что такое индексы в sql и зачем они нужны, узнала подробнее как выполняются запросы.