

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа №4

по дисциплине

«Базы данных»

Вариант №1305

Выполнил:

Студент группы Р3113

Султанов А.Р.

Проверил:

Горбунов М.В.

г. Санкт-Петербург

2023г.

Оглавление

Оглавление	2
Задание	3
SQL-запросы	5
Планы запросов	7
Предложения по индексам	9
Вывод команды EXPLAIN ANALYZE	10
Заключение	11

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.

Изменяться 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:

Н_ТИПЫ_ВЕДОМОСТЕЙ, Н_ВЕДОМОСТИ.

Вывести атрибуты: Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ,
Н_ВЕДОМОСТИ.ДАТА.

Фильтры (AND):

а) Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД > 1.

б) Н_ВЕДОМОСТИ.ЧЛВК_ИД < 105590.

с) Н_ВЕДОМОСТИ.ЧЛВК_ИД < 153285.

Вид соединения: LEFT 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.ИМЯ, Н_ОБУЧЕНИЯ.НЗК,
Н_УЧЕНИКИ.ГРУППА.

Фильтры: (AND)

а) Н_ЛЮДИ.ОТЧЕСТВО > Александрович.

б) Н_ОБУЧЕНИЯ.НЗК < 999080.

Вид соединения: RIGHT JOIN.

SQL-запросы

Запрос №1:

```
SELECT
    "Н_ТИПЫ_ВЕДОМОСТЕЙ" . "НАИМЕНОВАНИЕ" ,
    "Н_ВЕДОМОСТИ" . "ДАТА"
FROM
    "Н_ТИПЫ_ВЕДОМОСТЕЙ"
    LEFT JOIN "Н_ВЕДОМОСТИ" ON "Н_ВЕДОМОСТИ" . "ТВ_ИД" =
    "Н_ТИПЫ_ВЕДОМОСТЕЙ" . "ИД"
WHERE
    "Н_ТИПЫ_ВЕДОМОСТЕЙ" . "ИД" > 1
    AND "Н_ВЕДОМОСТИ" . "ЧЛВК_ИД" < 105590
    AND "Н_ВЕДОМОСТИ" . "ЧЛВК_ИД" < 153285 ;
```

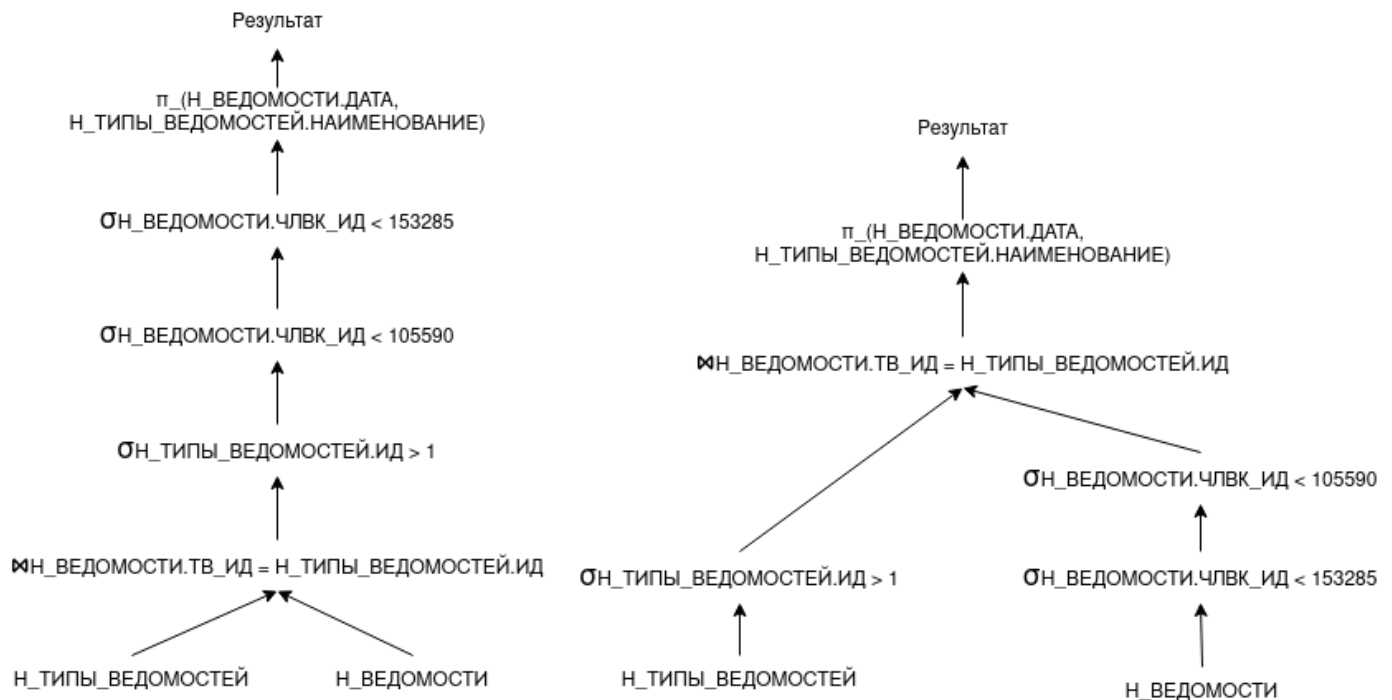
Запрос №2:

```
SELECT
    "Н_ЛЮДИ" . "ИМЯ" ,
    "Н_ОБУЧЕНИЯ" . "НЗК" ,
    "Н_УЧЕНИКИ" . "ГРУППА"
FROM
    "Н_ЛЮДИ"
    RIGHT JOIN "Н_ОБУЧЕНИЯ" ON "Н_ОБУЧЕНИЯ" . "ЧЛВК_ИД" =
    "Н_ЛЮДИ" . "ИД"
    RIGHT JOIN "Н_УЧЕНИКИ" ON "Н_УЧЕНИКИ" . "ЧЛВК_ИД" =
    "Н_ЛЮДИ" . "ИД"
WHERE
```

```
"Н_ЛЮДИ"."ОТЧЕСТВО" > 'Александрович'  
AND "Н_ОБУЧЕНИЯ"."НЗК" < '999080' ;
```

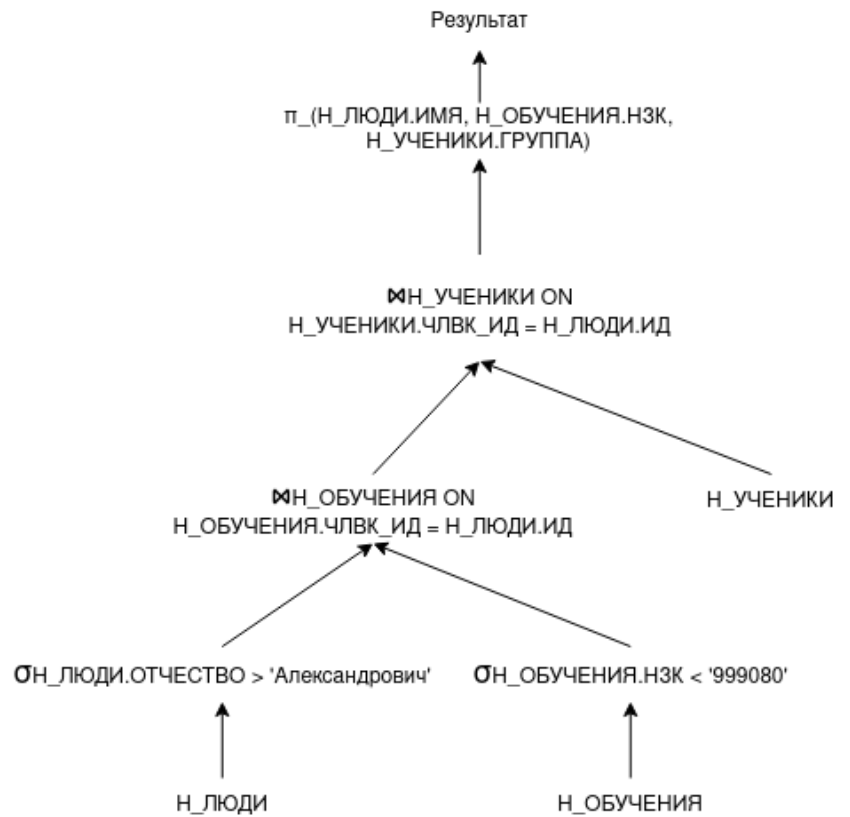
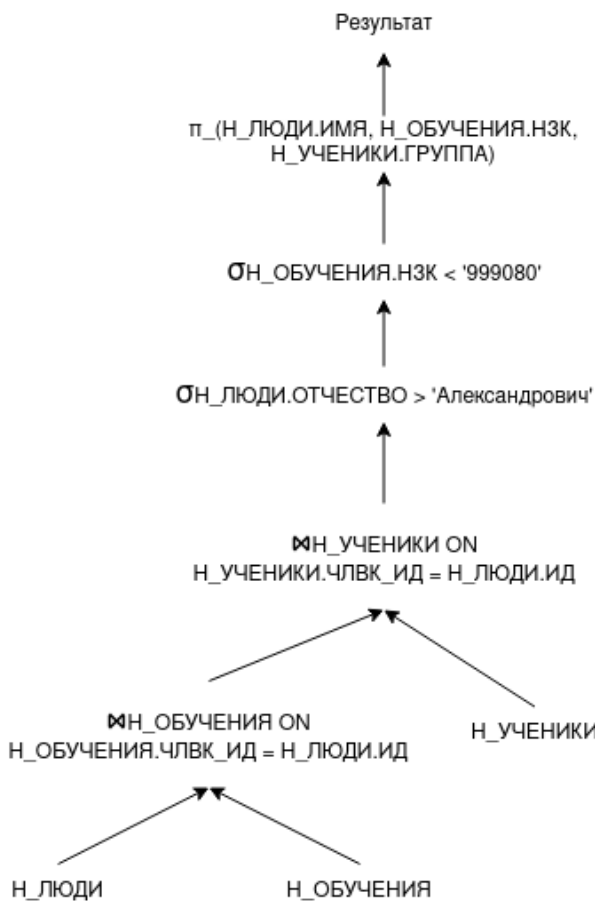
Планы запросов

Запрос №1



Первый план менее оптимален, так как, в отличие от второго, на момент соединения таблиц в них еще есть “неотфильтрованные” строки, что повлечет за собой излишние операции соединения, а после - фильтрации соединенных строк.

Запрос №2



В данном случае второй план более оптимален, так как перед соединением таблиц происходит предварительная их фильтрация, благодаря чему соединение производится только среди отфильтрованных строк, а значит, произойдет меньше операций.

Предложения по индексам

Для таблиц, задействованных в запросе №1:

```
CREATE INDEX "н_ведомости_члвк_ид_индекс" ON  
"Н_ВЕДОМОСТИ" ("ЧЛВК_ИД") ;
```

На "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" есть смысл повесить индекс btree, так как в запросе происходит фильтрация (<) по этому полю.

Для таблиц, задействованных в запросе №2:

```
CREATE INDEX "н_обучения_нзк_индекс" ON "Н_ОБУЧЕНИЯ" ("НЗК") ;
```

На "Н_ОБУЧЕНИЯ"."НЗК" есть смысл повесить индекс btree, так как в запросе происходит фильтрация (<) по этому полю.

```
CREATE INDEX "н_люди_отчество_индекс" ON "Н_ЛЮДИ" ("ОТЧЕСТВО") ;
```

На "Н_ЛЮДИ"."ОТЧЕСТВО" есть смысл повесить индекс btree, так как в запросе происходит фильтрация (<) по этому полю.

```
CREATE INDEX "н_обучения_члвк_ид_индекс" ON "Н_ОБУЧЕНИЯ" USING  
HASH("ЧЛВК_ИД") ;
```

```
CREATE INDEX "н_ученики_члвк_ид_индекс" ON "Н_УЧЕНИКИ" USING  
HASH("ЧЛВК_ИД") ;
```

На "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" и "Н_УЧЕНИКИ"."ЧЛВК_ИД" есть смысл повесить хэш-индекс, так как в запросе происходит соединение (=) по этим полям (и нет других операций, например, сравнения). Btree здесь подходит чуть меньше, так как хорошая производительность при других операциях здесь не нужна, но при этом есть проигрыш по памяти в сравнении с хэш-индексом.

Добавление индексов в таблицы не повлияет на планы запросов. Оно лишь может сказаться на их производительности.

Вывод команды EXPLAIN ANALYZE

Запрос №1

```
Nested Loop (cost=0.29..8.32 rows=1 width=430) (actual time=0.022..0.022 rows=0 loops=1)
  Join Filter: ("Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД" = "Н_ВЕДОМОСТИ"."ТВ_ИД")
    -> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=422) (actual
time=0.014..0.015 rows=2 loops=1)
      Filter: ("ИД" > 1)
      Rows Removed by Filter: 1
    -> Index Scan using "ВЕД_ЧЛВК_FK_IFK" on "Н_ВЕДОМОСТИ" (cost=0.29..7.27 rows=1 width=16)
(actual time=0.003..0.003 rows=0 loops=2)
      Index Cond: (("ЧЛВК_ИД" < 105590) AND ("ЧЛВК_ИД" < 153285))
Planning Time: 0.317 ms
Execution Time: 0.064 ms
```

Запрос №2

```
Hash Join (cost=395.99..1461.44 rows=17479 width=23) (actual time=7.888..18.230 rows=17230 loops=1)
  Hash Cond: ("Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД")
    -> Seq Scan on "Н_УЧЕНИКИ" (cost=0.00..774.11 rows=23311 width=8) (actual time=0.009..2.503 rows=23311 loops=1)
    -> Hash (cost=349.50..349.50 rows=3719 width=27) (actual time=7.863..7.865 rows=3716 loops=1)
      Buckets: 4096 Batches: 1 Memory Usage: 263kB
      -> Hash Join (cost=218.21..349.50 rows=3719 width=27) (actual time=4.156..6.993 rows=3716 loops=1)
        Hash Cond: ("Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
        -> Seq Scan on "Н_ОБУЧЕНИЯ" (cost=0.00..119.76 rows=4387 width=10) (actual time=0.010..1.728 rows=4387
loops=1)
          Filter: (("НЗК")::text < '999080'::text)
          Rows Removed by Filter: 634
        -> Hash (cost=163.97..163.97 rows=4339 width=17) (actual time=4.128..4.129 rows=4339 loops=1)
          Buckets: 8192 Batches: 1 Memory Usage: 283kB
          -> Seq Scan on "Н_ЛЮДИ" (cost=0.00..163.97 rows=4339 width=17) (actual time=0.006..3.350
rows=4339 loops=1)
            Filter: (("ОТЧЕСТВО")::text > 'Александрович'::text)
            Rows Removed by Filter: 779
Planning Time: 0.793 ms
Execution Time: 19.093 ms
```

Заключение

В рамках данной лабораторной работы я изучил индексы в БД (PostgreSQL в частности), научился составлять примерные планы выполнения запросов, а также познакомился с командой EXPLAIN.