

Университет ИТМО

Лабораторная работа №4

«Базы данных»

Студент: **Фарзекаев Артур Робертович**

Кафедра: **ВТ**

Дисциплина: **Базы Данных**

Факультет: **ПИиКТ**

Группа: **Р3100**

Вариант: **928**

Преподаватель: **Николаев Владимир**

Вячеславович.

г. Санкт-Петербург.

2018 г.

Задание.

По варианту, выданному преподавателем, составить и выполнить запросы к базе данных "Учебный процесс".

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).

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.ЧЛВК_ИД.

Фильтры (AND):

а) Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД > 2.

б) Н_ВЕДОМОСТИ.ДАТА = 2022-06-08.

с) Н_ВЕДОМОСТИ.ДАТА = 1998-01-05.

Вид соединения: LEFT JOIN.

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

Таблицы: Н_ЛЮДИ, Н_ОБУЧЕНИЯ, Н_УЧЕНИКИ.

Вывести атрибуты: Н_ЛЮДИ.ФАМИЛИЯ, Н_ОБУЧЕНИЯ.НЗК, Н_УЧЕНИКИ.ГРУППА.

Фильтры: (AND)

а) Н_ЛЮДИ.ФАМИЛИЯ = Иванов.

б) Н_ОБУЧЕНИЯ.НЗК = 001000.

Вид соединения: RIGHT JOIN.

Выполнение:

1.

```
ucheb=> SELECT "Н_ТИПЫ_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ", "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" FROM
ucheb-> "Н_ТИПЫ_ВЕДОМОСТЕЙ" LEFT JOIN "Н_ВЕДОМОСТИ" ON ("Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД" = "Н_ВЕДОМОСТИ"."ТВ_ИД")
ucheb-> WHERE ("Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД" > 2
ucheb(> AND ("Н_ВЕДОМОСТИ"."ДАТА" = '2022-06-08' OR "Н_ВЕДОМОСТИ"."ДАТА" = '1998-01-05'));
НАИМЕНОВАНИЕ | ЧЛВК_ИД
-----+-----
(0 rows)

ucheb=>
```

2.

```
ucheb=> SELECT "Н_ЛЮДИ"."ФАМИЛИЯ", "Н_ОБУЧЕНИЯ"."НЗК", "Н_УЧЕНИКИ"."ГРУППА" FROM
ucheb-> "Н_ЛЮДИ" RIGHT JOIN "Н_ОБУЧЕНИЯ" ON ("Н_ЛЮДИ"."ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД")
ucheb-> RIGHT JOIN "Н_УЧЕНИКИ" ON ("Н_ЛЮДИ"."ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД")
ucheb-> WHERE "Н_ЛЮДИ"."ФАМИЛИЯ" = 'Иванов' AND "Н_ОБУЧЕНИЯ"."НЗК" = '001000';
ФАМИЛИЯ | НЗК | ГРУППА
-----+-----+-----
(0 rows)

ucheb=>
```

Возможные индексы:

1. Вставка индексов в таблицу Н_ТИПЫ_ВЕДОМОСТЕЙ не имеет смысла, так как данная таблица содержит малое количество записей, а индексы неэффективны, если в таблице мало строк.

Что касается атрибута ДАТА таблицы Н_ВЕДОМОСТИ, то здесь ситуация неоднозначная, поскольку в фильтрах указан поиск по двум датам, а у каждого ИД своя уникальная дата, не может быть два разных времени для одного события, поэтому фильтр AND не применим к двум датам, ежели дата была бы только одна, то можно было бы использовать индекс B-tree (ускорение выполнения запроса произойдет, потому что искать ссылку на экземпляр по сбалансированному дереву быстрее чем искать по всей таблице)

```
CREATE INDEX index1 ON "Н_ВЕДОМОСТИ" USING btree("ДАТА");
```

2. Вставка B-tree индексов в таблицы Н_ЛЮДИ и Н_ОБУЧЕНИЯ для атрибутов ФАМИЛИЯ и НЗК соответственно является целесообразным, поскольку данные таблицы содержат довольно большое количество записей, и индексы существенно ускорят время поиска необходимых данных.

```
CREATE INDEX index1 ON "Н_ЛЮДИ" USING btree("ФАМИЛИЯ");
```

```
CREATE INDEX index2 ON "Н_ОБУЧЕНИЯ" USING btree("НЗК");
```

EXPLAIN ANALYZE

1.

```

QUERY PLAN
-----
Hash Join (cost=109.83..4513.25 rows=1830 width=422) (actual time=4.103..4.103 rows=0 loops=1)
  Hash Cond: ("Н_ВЕДОМОСТИ"."ТВ ИД" = "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД")
  -> Bitmap Heap Scan on "Н_ВЕДОМОСТИ" (cost=108.78..4473.31 rows=5490 width=8) (actual time=0.407..2.309 rows=5196 loops=1)
        Recheck Cond: (("ДАТА" = '2022-06-08 00:00:00'::timestamp without time zone) OR ("ДАТА" = '1998-01-05 00:00:00'::timestamp without time zone))
        Heap Blocks: exact=253
  -> BitmapOr (cost=108.78..108.78 rows=5492 width=0) (actual time=0.381..0.381 rows=0 loops=1)
        -> Bitmap Index Scan on "ВЕД_ДАТА_I" (cost=0.00..4.96 rows=72 width=0) (actual time=0.010..0.010 rows=3 loops=1)
              Index Cond: ("ДАТА" = '2022-06-08 00:00:00'::timestamp without time zone)
        -> Bitmap Index Scan on "ВЕД_ДАТА_I" (cost=0.00..101.07 rows=5420 width=0) (actual time=0.371..0.371 rows=5193 loops=1)
              Index Cond: ("ДАТА" = '1998-01-05 00:00:00'::timestamp without time zone)
  -> Hash (cost=1.04..1.04 rows=1 width=422) (actual time=0.019..0.019 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=0.012..0.013 rows=1 loops=1)
              Filter: ("ИД" > 2)
              Rows Removed by Filter: 2
Planning time: 0.194 ms
Execution time: 4.156 ms
(17 rows)

ucheb=>
```

2.

```

QUERY PLAN
-----
Nested Loop (cost=4.61..144.43 rows=5 width=26) (actual time=1.402..1.402 rows=0 loops=1)
  Join Filter: ("Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
  -> Nested Loop (cost=4.33..142.76 rows=5 width=18) (actual time=0.241..1.352 rows=2 loops=1)
        -> Seq Scan on "Н_ОБУЧЕНИЯ" (cost=0.00..119.76 rows=1 width=10) (actual time=0.174..1.271 rows=1 loops=1)
              Filter: (("НЭК")::text = '001000'::text)
              Rows Removed by Filter: 5020
        -> Bitmap Heap Scan on "Н_УЧЕНИКИ" (cost=4.33..22.95 rows=5 width=8) (actual time=0.063..0.075 rows=2 loops=1)
              Recheck Cond: ("ЧЛВК_ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД")
              Heap Blocks: exact=2
              -> Bitmap Index Scan on "УЧЕН_ОБУЧ_ФК_I" (cost=0.00..4.32 rows=5 width=0) (actual time=0.038..0.038 rows=2 loops=1)
                    Index Cond: ("ЧЛВК_ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД")
  -> Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..0.32 rows=1 width=20) (actual time=0.023..0.023 rows=0 loops=2)
        Index Cond: ("ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД")
        Filter: (("ФАМИЛИЯ")::text = 'Иванов'::text)
        Rows Removed by Filter: 1
Planning time: 3.203 ms
Execution time: 1.457 ms
(17 rows)

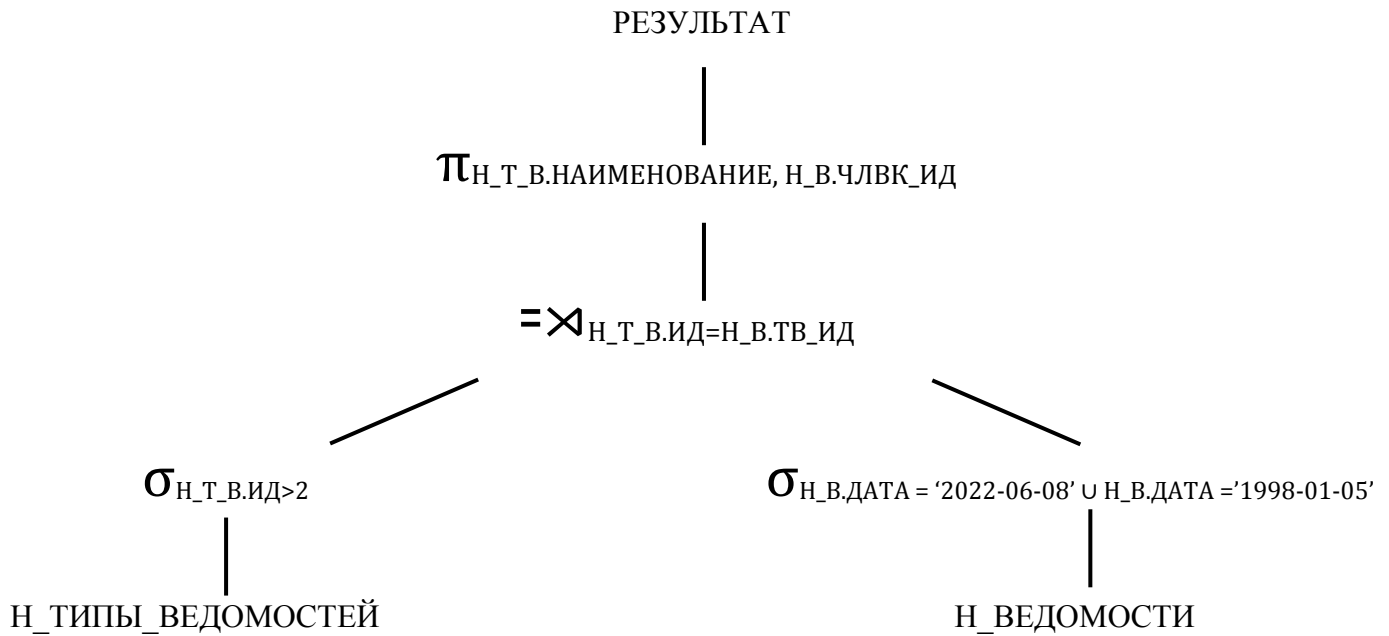
ucheb=>
```

Вывод:

В данной лабораторной работе, я предложил использование индексов для своих запросов с целью ускорения их выполнения, а также составил планы выполнения запросов и выбрал оптимальный из них, выполнил команду EXPLAIN ANALYZE для каждого запроса.

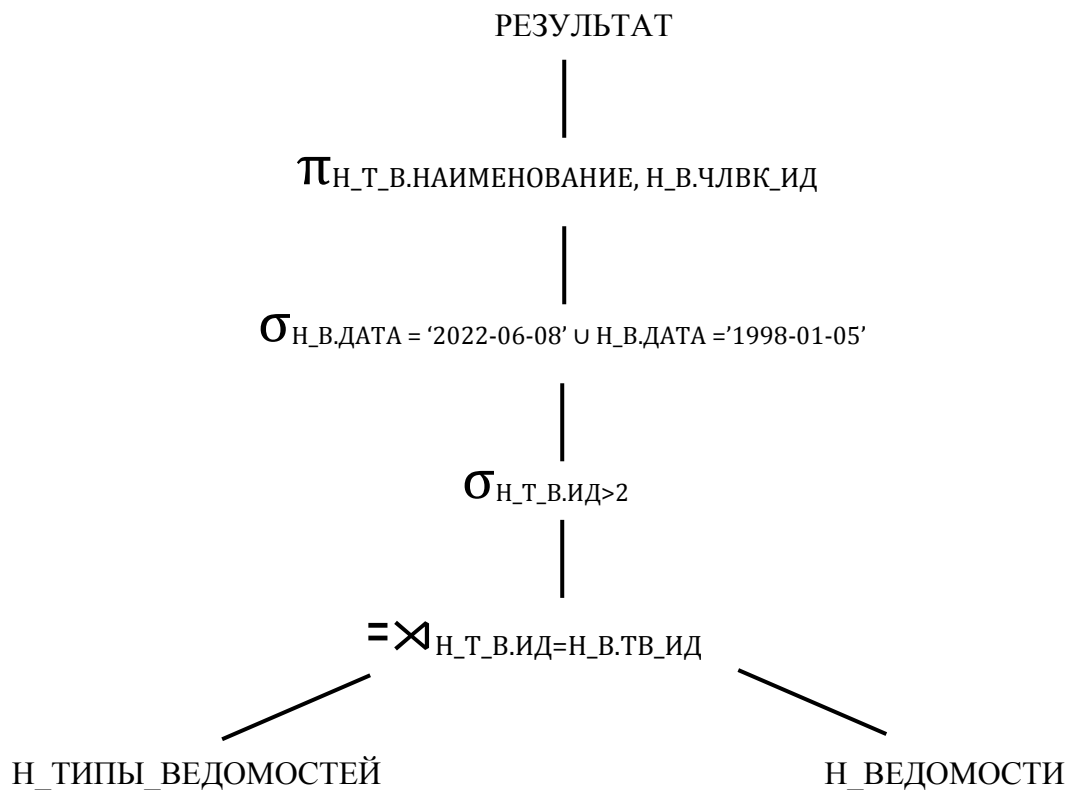
План для 1 запроса:

1 вариант.



При построении этого плана я использую конвейерную обработку, исключая блокирующие операции, делаю выборку как можно раньше, проекция реализую только после соединения двух таблиц и получаю окончательный результат.

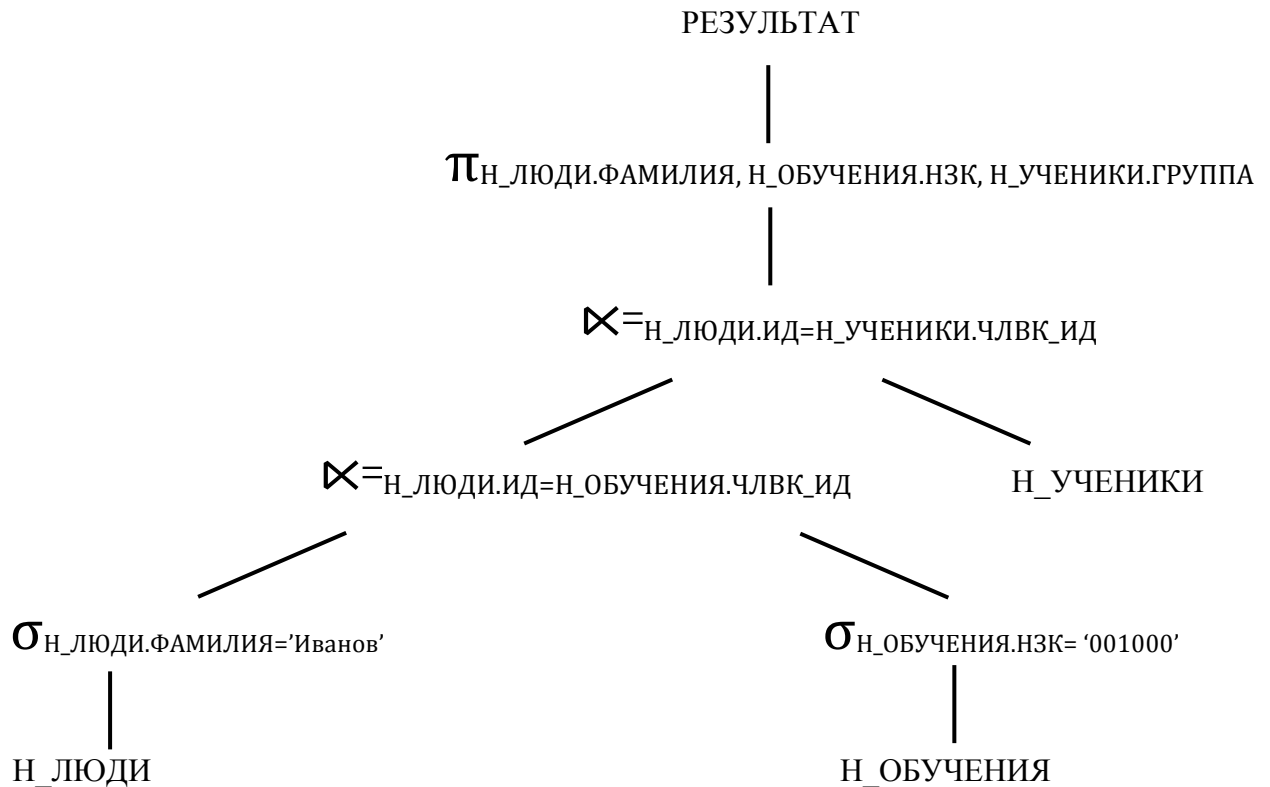
2 вариант.



При построении этого плана я соединяю обе таблицы, делаю выборку и проекцию, и получаю окончательный результат. Данный план является неэффективным, поскольку соединяются две таблицы и после этого совершается выборка и проекция, из-за этого все ненужные данные тянутся с двух таблиц. Таким образом 1 вариант плана является эффективным.

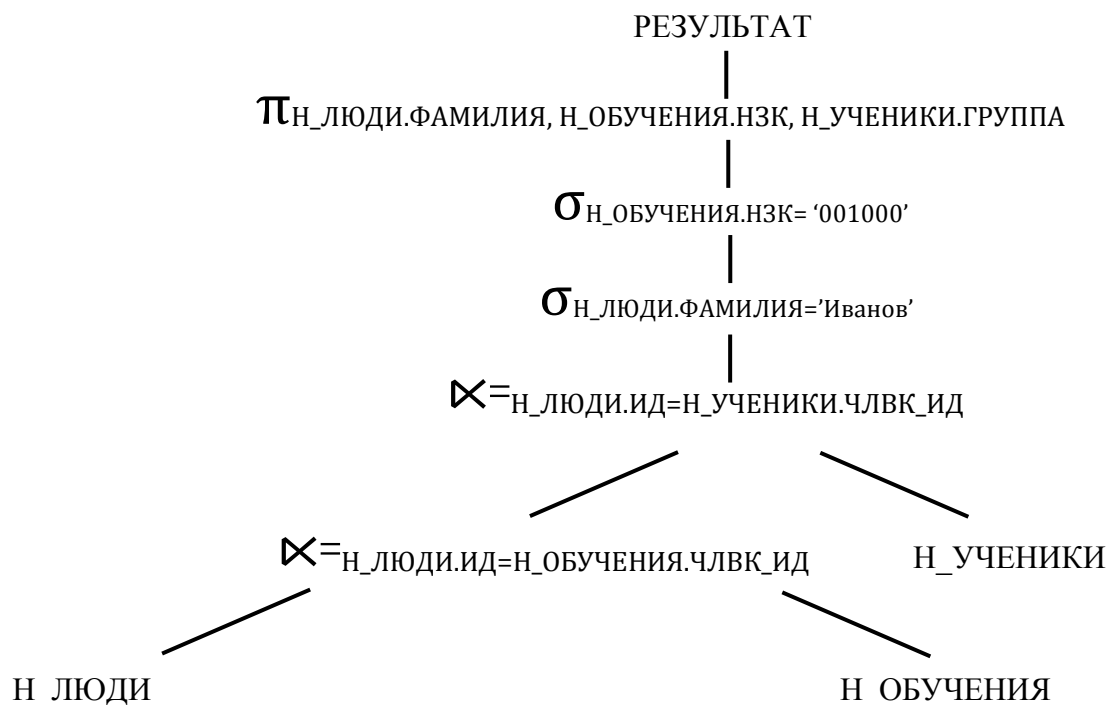
План для 2 запроса:

1 вариант.



При построении этого плана я делаю выборку, соединяю две таблицы, затем соединяю с третьей таблицей и выбираю нужные проекции, после чего получаю результат.

2 вариант.



При построении этого плана я соединяю все таблицы и только после этого делаю выборку и проекцию, после чего получаю окончательный результат. Данный план является неэффективным, поскольку соединяются три таблицы и после этого совершается выборка и проекция, из-за этого все ненужные данные тянутся с этих трех таблиц. Таким образом 1 вариант плана является эффективным.