

**Федеральное государственное автономное образовательное
учреждение
высшего образования
«Национальный исследовательский университет ИТМО»
Факультет Программной Инженерии и Компьютерной Техники**



**Вариант №6353
Лабораторная работа №4
по дисциплине
Базы данных**

Выполнил Студент группы Р3112
Кобелев Роман Павлович
Преподаватель:
Лисицина Василиса Васильева.

г. Санкт-Петербург
2023г.

1 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.

Изменяются 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).

2 Запросы и индексы

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

Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ.

Вывести атрибуты: Н_ЛЮДИ.ИД, Н_ВЕДОМОСТИ.ИД.

Фильтры (AND):

- Н_ЛЮДИ.ИМЯ = Николай.
- Н_ВЕДОМОСТИ.ЧЛВК_ИД < 142390.
- Н_ВЕДОМОСТИ.ЧЛВК_ИД > 153285.

Вид соединения: INNER JOIN.

Для ускорения запроса можно предложить следующие варианты индекса:

- (а) Так как у нас в WHERE используется неравенство, то запрос можно ускорить, добавив B-tree индексацию по атрибуту ЧЛВК_ИД в таблице Н_ВЕДОМОСТ.
- (б) Для таблицы Н_ЛЮДИ можно проиндексировать атрибут ИМЯ

1_task_index.sql

```
1 SELECT "нлюди_".ИД, "нведомости_".ИД
2 FROM "нлюди_"
3     INNER JOIN "нведомости_" ON ("нведомости_".ЧЛВКИД_ = "нлюди_".ИД)
4 WHERE ИМЯ = 'Николай'
5     AND "нведомости_".ЧЛВКИД_ < 42390
6     AND "нведомости_".ЧЛВКИД_ > 53285;
7
8 -----1-----
9 CREATE INDEX ЧЛВКИДindex_ ON "нведомости_" USING btree ("ЧЛВКИД_");
10
11 -----2-----
12 CREATE INDEX ИМЯindex_ ON "нлюди_" USING btree ("ИМЯ")
```

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.ИД, Н_ОБУЧЕНИЯ.НЗК, Н_УЧЕНИКИ.ИД.

Фильтры: (AND)

- Н_ЛЮДИ.ОТЧЕСТВО > Георгиевич.
- Н_ОБУЧЕНИЯ.ЧЛВК_ИД = 163484.

Вид соединения: INNER JOIN. Для ускорения запроса можно предложить следующие варианта индекса:

- (a) Так как у нас идёт сравнение ОТЧЕСТВО, то для ускорения запроса можно создать индекс в таблице Н_ЛЮДИ, чтобы у нас хранились значения в листьях и они были отсортированными.
- (b) Для таблицы Н_ОБУЧЕНИЯ можно проиндексировать атрибут ЧЛВК_ИД

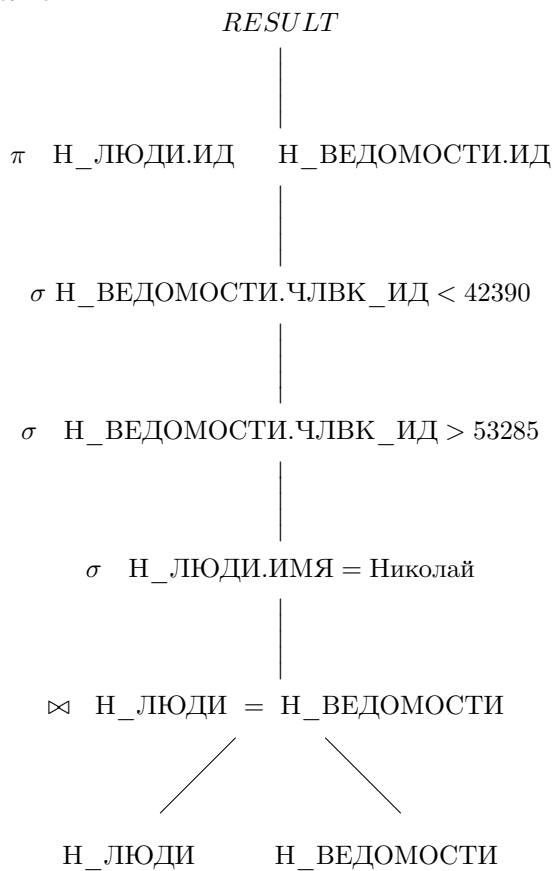
2_task_index.sql

```
1 EXPLAIN ANALYZE
2 SELECT  "нлюди_". "ид", "нобучения_". "нзк", "нученики_". "ид"
3 FROM  "нлюди_"
4 INNER JOIN "нобучения_" ON ("нобучения_". "члвк_ид" = "нлюди_". "ид")
5 INNER JOIN "нученики_" ON ("нобучения_". "члвк_ид" = "нлюди_". "ид")
6 WHERE "нлюди_". "отчество" > 'Георгиевич' AND "нобучения_". "члвк_ид" = 163484;
7
8
9 -----1-----
10 CREATE INDEX отчествоindex_ ON "нлюди_" USING btree ("отчество");
11 CLUSTER отчествоindex_ ON "нлюди_";
12 -----2-----
13 CREATE INDEX члвк_идindex__ ON "нобучения_" USING btree ("члвк_ид");
```

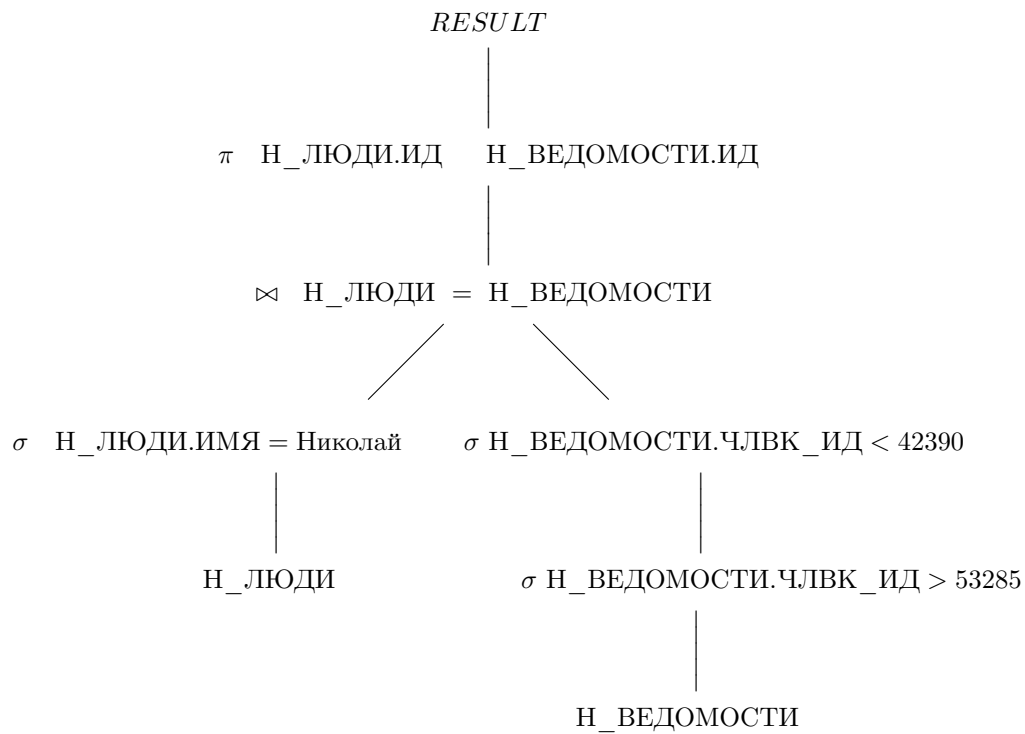
3 План выполнения

1. Первый запрос

(a) Неоптимальный план выполнения

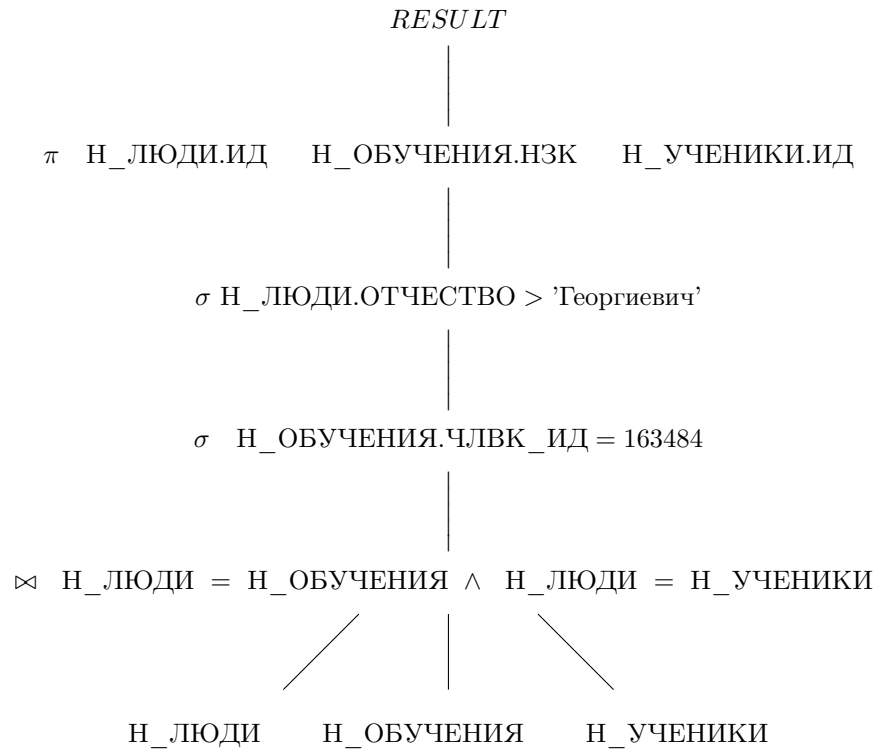


(b) Оптимальный план выполнения



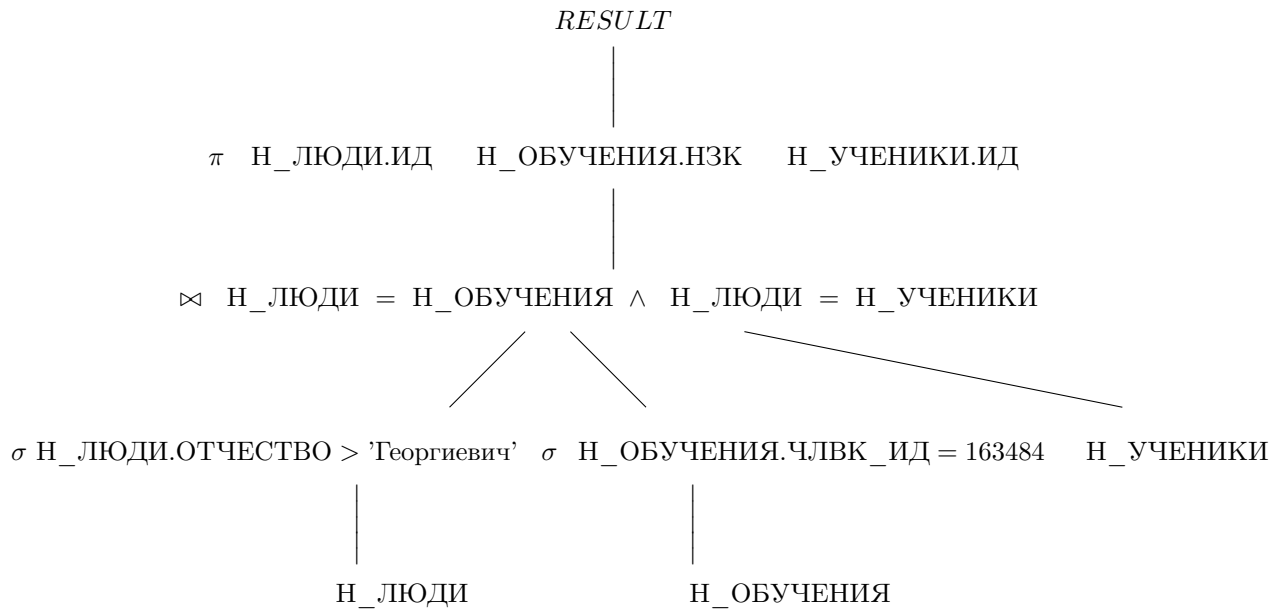
2. Второй запрос

(а) Неоптимальный план выполнения



(b) оптимальный план выполнения

Целью оптимизации является уменьшение размеров промежуточных данных -> уменьшение числа операций чтения записи во внешнюю память



Эта модель является оптимизированной, так как мы достигли цели. Мы достигли момента, когда количество обращений к данным - минимально.

4 Влияние индекса на выбор плана

1. Если запрос содержит условие, которое поддерживается каким-либо индексом, то при добавлении индекса у нас увеличится скорость выполнения запроса. Индекс увеличит скорость выборки элементов из БД.
2. При выполнении объединения таблиц наличие индексов может увеличить производительность объединения, так как индексы могут быстрее найти данные, которые будем объединять
3. Планировщик задач может выбрать новый план выполнения, учитывая наличие индекса и его стоимость. Это приводит к изменению порядка выполнения операций.

5 EXPLAIN ANALYZE

1. 1 запрос

```
Nested Loop (cost=0.58..16.83 rows=1 width=8) (actual time=0.006..0.007 rows=0 loops=1)
-> Index Scan using "ВЕД_ЧЛВК_FK_IFK" on "Н_ВЕДОМОСТИ" (cost=0.29..8.31 rows=1 width=8)
(actual time=0.005..0.006 rows=0 loops=1)
Index Cond: (("ЧЛВК_ИД" < 42390) AND ("ЧЛВК_ИД" > 53285))
-> Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1 width=4) (never executed)
Index Cond: ("ИД" < "Н_ВЕДОМОСТИ"."ЧЛВК_ИД")
Filter: (("ИМЯ")::text = 'Николай'::text)
Planning Time: 1.367 ms
Execution Time: 0.065 ms
```

2. 2 запрос

```
Nested Loop (cost=0.85..863.67 rows=23311 width=14) (actual time=0.026..0.027 rows=0 loops=1)
-> Nested Loop (cost=0.56..16.61 rows=1 width=10) (actual time=0.026..0.026 rows=0 loops=1)
-> Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1 width=4)
(actual time=0.026..0.026 rows=0 loops=1)
Index Cond: ("ИД" < 163484)
Filter: (("ОТЧЕСТВО")::text > 'Георгиевич'::text)
Rows Removed by Filter: 1
-> Index Scan using "ОБУЧ_ЧЛВК_FK_Г" on "Н_ОБУЧЕНИЯ" (cost=0.28..8.30 rows=1 width=10) (never
executed)
Index Cond: ("ЧЛВК_ИД" < 163484)
-> Index Only Scan using "УЧЕН_РК" on "Н_УЧЕНИКИ" (cost=0.29..613.95 rows=23311 width=4) (never
executed)
Heap Fetches: 0
Planning Time: 0.878 ms
Execution Time: 0.062 ms
```

6 Выводы

В данной лабораторной работе я познакомился с индексами в языке PostgreSQL. Узнал, что это такое, и какую роль они играют в оптимизации запросов. Увидел, как индексы инициализировать в СУБД и каких видов бывают.