

Министерство науки и высшего образования Российской Федерации ФЕДЕРАЛЬНОЕ  
ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

“Национальный исследовательский университет ИТМО”

**Информационные системы и базы данных**

Лабораторная работа №4

Вариант №52342

***Работу выполнил:***

*Кузнецов Максим Александрович*

***Группа: Р33131***

***Преподаватель:***

*Николаев Владимир Вячеславович*

Санкт-Петербург, 2022 г.

# Лабораторная работа #4

## Задание.

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

Команда для подключения к базе данных ucheb:

```
psql -h pg -d ucheb
```

Отчёт по лабораторной работе должен содержать:

1. Текст задания.
2. Реализацию запросов на SQL.
3. Планы выполнения запросов.
4. Ответы на вопросы, представленные в задании.
5. Выводы по работе.

Темы для подготовки к защите лабораторной работы:

1. Индексы
2. Оптимизация запросов
3. Выбор плана выполнения запросов

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).

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

Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ.

Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД.

Фильтры (AND):

- a) Н\_ЛЮДИ.ИМЯ = Александр.
- b) Н\_ВЕДОМОСТИ.ИД = 1250981.
- c) Н\_ВЕДОМОСТИ.ИД < 1457443.

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.ИД, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.НАЧАЛО.

Фильтры: (AND)

- a) Н\_ЛЮДИ.ИД = 100865.
  - b) Н\_ОБУЧЕНИЯ.НЗК < 001000.
- Вид соединения: RIGHT JOIN.

## Выполнение

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

Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ.

Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД.

Фильтры (AND):

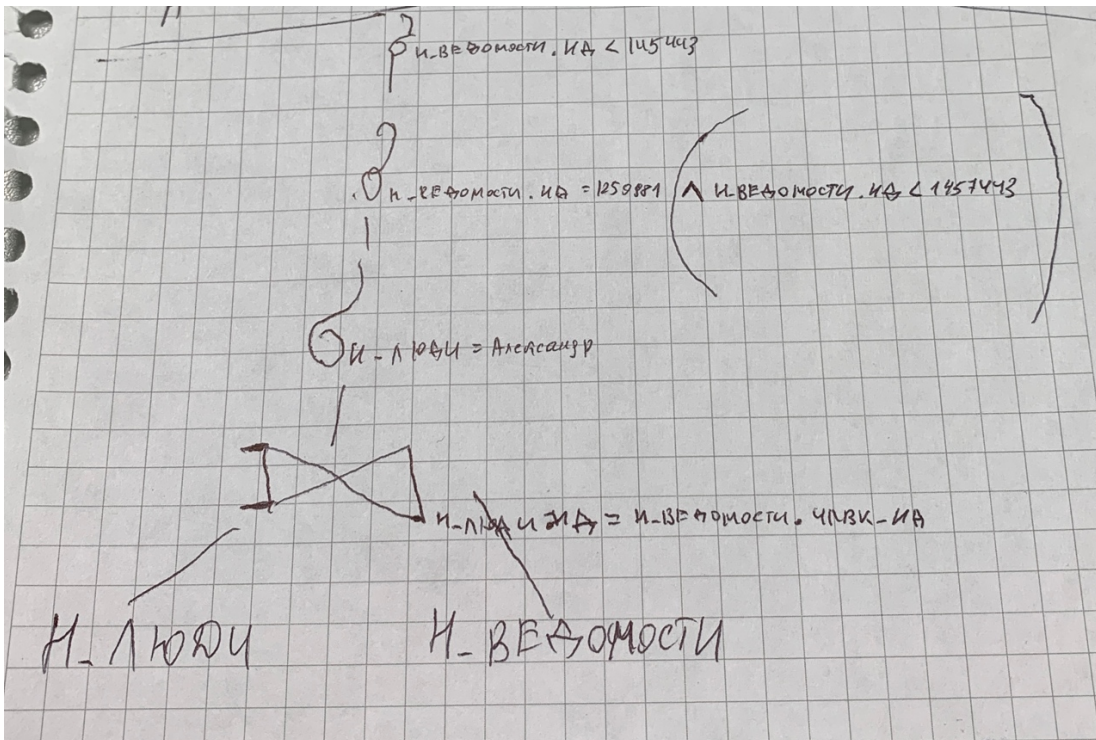
- a) Н\_ЛЮДИ.ИМЯ = Александр.
- b) Н\_ВЕДОМОСТИ.ИД = 1250981.
- c) Н\_ВЕДОМОСТИ.ИД < 1457443.

Вид соединения: LEFT JOIN.

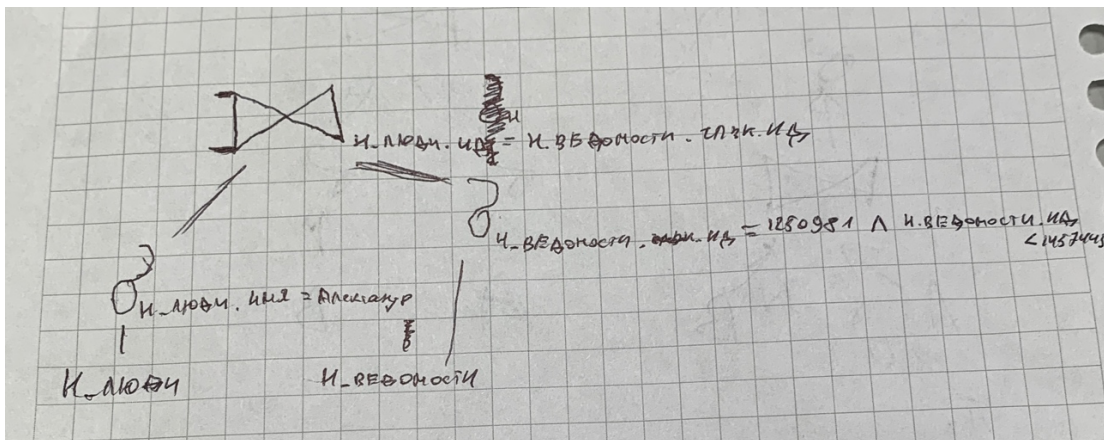
```
select "Н_ЛЮДИ"."ОТЧЕСТВО", "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" FROM "Н_ЛЮДИ"
left join "Н_ВЕДОМОСТИ" on "Н_ЛЮДИ"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
where "Н_ЛЮДИ"."ИМЯ" = 'Александр' and "Н_ВЕДОМОСТИ"."ИД" = 1250981 and
"Н_ВЕДОМОСТИ"."ИД" < 1457443
```

## Планы:

1 (а точнее 2)



3



На первом листке я привел не лучший план, так как там в первую очередь left join, а потом только лишь выборка нужных данных. В этом случае мы несем с собой множество «ненужных» данных, которые прямым образом влияют на скорость и результативность операции.

На втором листе изображен более рациональный подход, так как в таком случае мы изначально избавляемся от лишних данных (т. е. промежуточных данных меньше) и работаем впредь с теми, которые нас интересуют первостепенно.

## Индексы, которые можно предложить:

```
CREATE INDEX ON Н_ЛЮДИ USING btree(ИД);  
CREATE INDEX ON Н_ВЕДОМОСТИ USING btree(ЧЛВК_ИД);
```

```
CREATE INDEX ON Н_ЛЮДИ USING hash(ИМЯ);  
CREATE INDEX ON Н_ВЕДОМОСТИ USING btree(ИД);
```

- Имеет смысл (если в условии явно указано, что индексы по умолчанию отсутствуют) навесить индексы на id, по которым идет соединение таблиц

- В сущности Н\_ЛЮДИ выборка происходит по сопоставлению конкретного значения, поэтому оптимально использовать hash index
- В сущности Н\_ВЕДОМОСТИ выборка происходит по сопоставлению И конкретного значения И с использованием оператора '<', поэтому здесь имеет место использование btree index.

### Explain Analyze:

```
Nested Loop (cost=0.70..16.78 rows=1 width=24) (actual time=0.004..0.005 rows=0 loops=1)
-> Index Scan using "ВЕД_РК" on "Н_ВЕДОМОСТИ" (cost=0.42..8.44 rows=1 width=4) (actual time=0.003..0.003 rows=0 loops=1)
    Index Cond: (("ИД" < 145443) AND ("ИД" = 1250981))
-> Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1 width=24) (never executed)
    Index Cond: ("ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД")
    Filter: (("ИМЯ")::text = 'Александр'::text)
Planning Time: 0.352 ms
Execution Time: 0.037 ms
```

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.НАЧАЛО.

Фильтры: (AND)

а) Н\_ЛЮДИ.ИД = 100865.

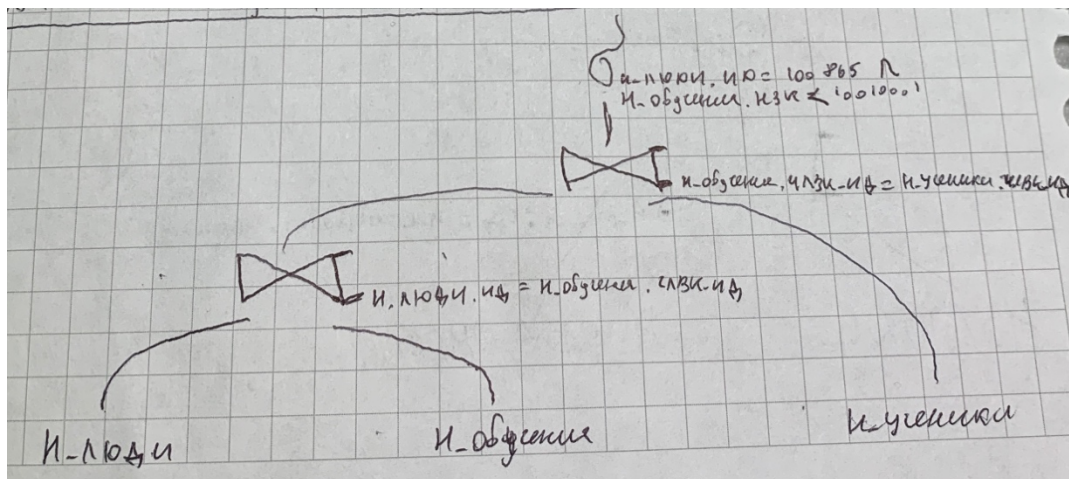
б) Н\_ОБУЧЕНИЯ.НЗК < 001000.

Вид соединения: RIGHT JOIN.

```
select "Н_ЛЮДИ"."ИД", "Н_ОБУЧЕНИЯ"."НЗК", "Н_УЧЕНИКИ"."НАЧАЛО" from "Н_ЛЮДИ"
right join "Н_ОБУЧЕНИЯ" on "Н_ЛЮДИ"."ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД"
right join "Н_УЧЕНИКИ" on "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД"
where "Н_ЛЮДИ"."ИД" = 100865 and "Н_ОБУЧЕНИЯ"."НЗК" < '001000'
```

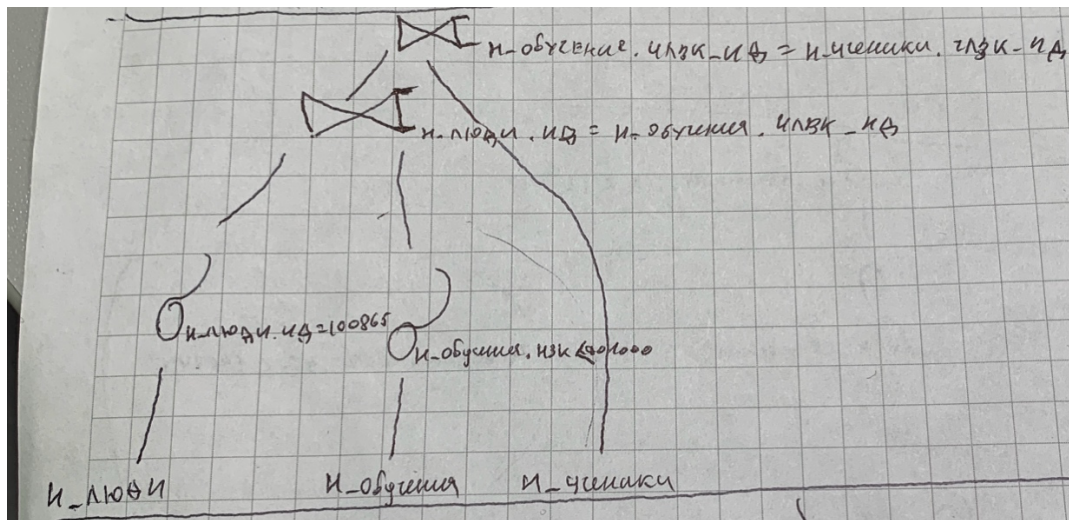
### Планы:

1



2





Опять-таки, в первом случае (плане) происходит изначальное объединение сущностей, которое «протаскивает» через весь запрос с собой много лишних данных. Во втором же плане мы сначала делаем выборку по нужным данным, а затем лишь делаем right join.

**Индексы, которые можно предложить:**

```
CREATE INDEX ON Н_ЛЮДИ USING btree(ИД);
CREATE INDEX ON Н_ОБУЧЕНИЯ USING btree(ЧЛВК_ИД);
CREATE INDEX ON Н_УЧЕНИКИ USING btree(ЧЛВК_ИД);

CREATE INDEX ON Н_ОБУЧЕНИЯ USING btree(ИЗК);
```

- Имеет смысл (если в условии явно указано, что индексы по умолчанию отсутствуют) навесить индексы на id, по которым идет соединение таблиц, а также так как выборка в сущности Н\_ЛЮДИ идет в дальнейшем по сопоставлению значения, то такой индекс вдвойне необходим.
- В сущности Н\_ОБУЧЕНИЯ выборка происходит с использованием оператора '>', поэтому здесь имеет место использование btree index.

**Explain Analyze:**

```
Nested Loop (cost=4.89..35.61 rows=5 width=18) (actual time=0.036..0.037 rows=0 loops=1)
-> Nested Loop (cost=0.56..12.61 rows=1 width=14) (actual time=0.035..0.036 rows=0 loops=1)
    -> Index Only Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..4.30 rows=1 width=4) (actual time=0.024..0.025 rows=1 loops=1)
        Index Cond: ("ИД" = 100865)
        Heap Fetches: 0
    -> Index Scan using "ОБУЧ_ЧЛВК_ФК_I" on "Н_ОБУЧЕНИЯ" (cost=0.28..8.30 rows=1 width=10) (actual time=0.008..0.008 rows=0 loops=1)
        Index Cond: ("ЧЛВК_ИД" = 100865)
        Filter: (("ИЗК")::text < '001000'::text)
-> Bitmap Heap Scan on "Н_УЧЕНИКИ" (cost=4.33..22.95 rows=5 width=12) (never executed)
    Recheck Cond: ("ЧЛВК_ИД" = 100865)
-> Bitmap Index Scan on "УЧЕН_ОБУЧ_ФК_I" (cost=0.00..4.32 rows=5 width=0) (never executed)
    Index Cond: ("ЧЛВК_ИД" = 100865)
Planning Time: 0.292 ms
Execution Time: 0.086 ms
```

## Вывод

Я ходе лабораторной работы я познакомился с процессом планирования и выполнения запроса, его стоимостью, а также с различными типами индексов и их уместным применением в различных ситуациях.