

НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ КОРПОРАЦИЯ ИТМО



Факультет программной инженерии и компьютерной техники

Информационные системы и базы данных.

Лабораторная работа №4

Вариант 5162

Выполнил:

студент группы Р33302

Нестеров Иван Алексеевич

Преподаватель:

Гаврилов Антон Валерьевич

г. Санкт-Петербург

2022 г.

## 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).

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

Таблицы: Н\_ОЦЕНКИ, Н\_ВЕДОМОСТИ.

Вывести атрибуты: Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД.

Фильтры (AND):

а) Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ = зачет.

б) Н\_ВЕДОМОСТИ.ИД > 1457443.

с) Н\_ВЕДОМОСТИ.ИД = 39921.

Вид соединения: LEFT JOIN.

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

Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.

Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ДАТА, Н\_СЕССИЯ.ИД.

Фильтры (AND):

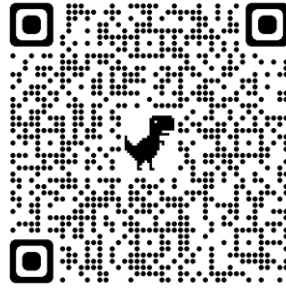
а) Н\_ЛЮДИ.ИМЯ = Владимир.

б) Н\_ВЕДОМОСТИ.ИД < 1250972.

Вид соединения: INNER JOIN.

## Ход работы:

[Код запросов](#) (для удобства чтения размещен на GitHub):



## Планы выполнения запроса:

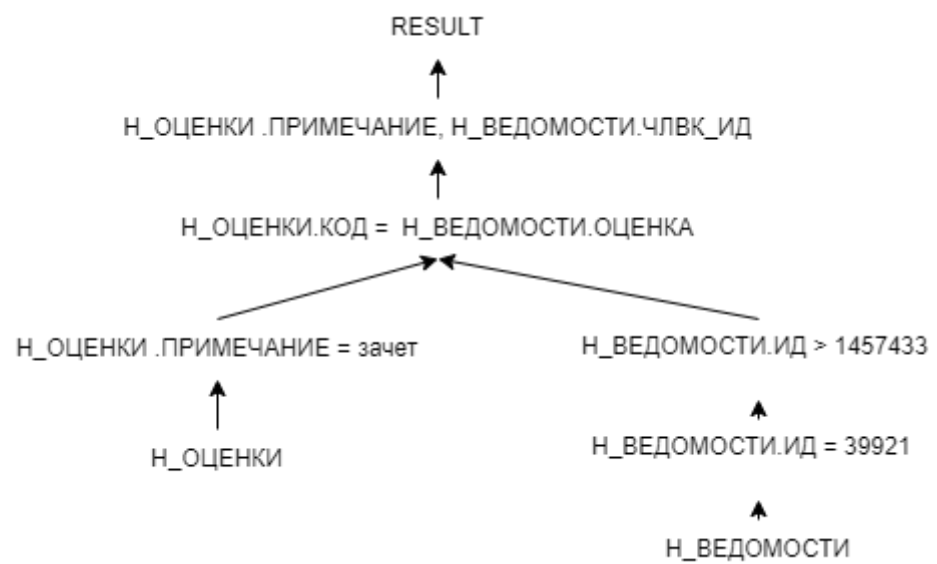
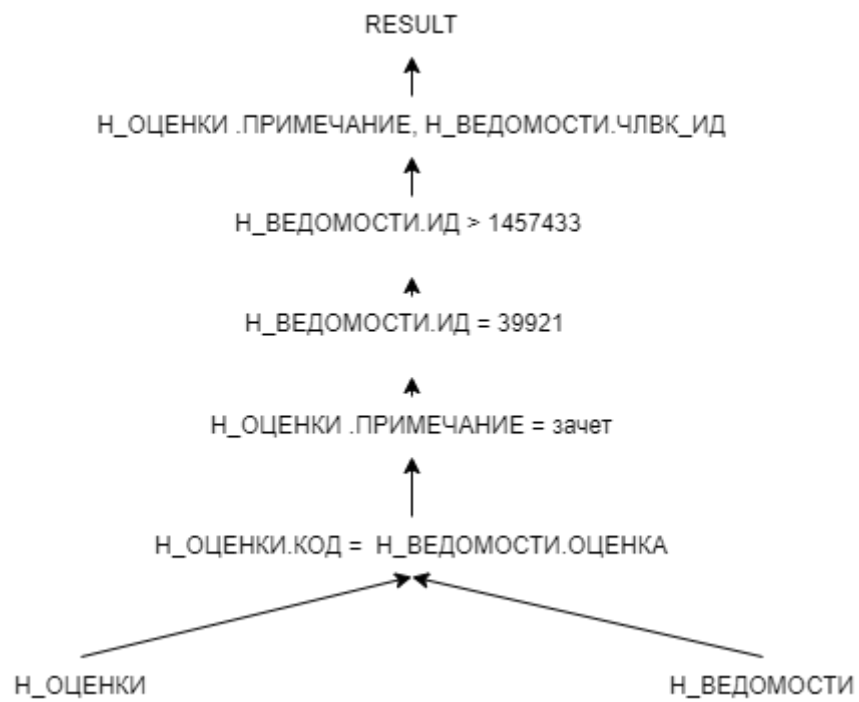
Первого:

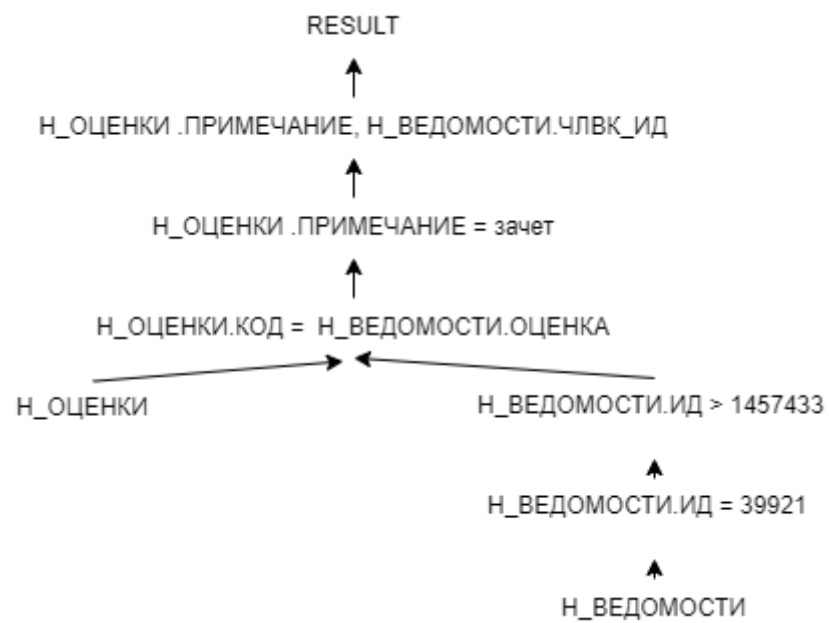
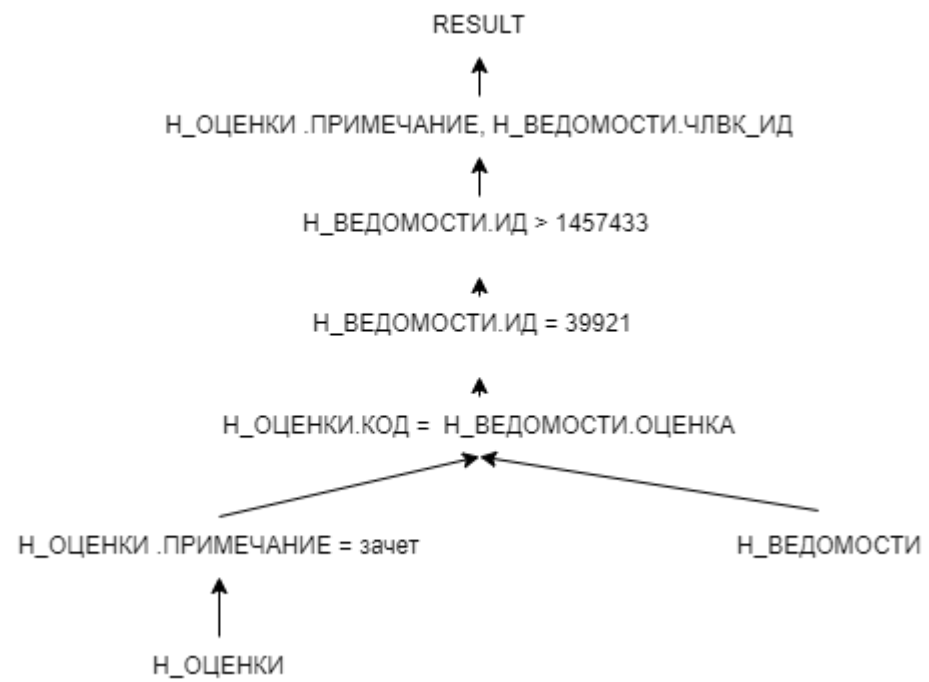
Изобразим четыре возможных плана выполнения запроса:

1. Сначала процесс объединения таблиц, а потом выборка по их столбцам;
2. Сначала выборка каждой таблицы отдельно, а затем объединение получившихся результатов;
3. Сначала объединение результата выборки первой таблицы со второй, а затем выборка по столбцам второй таблицы;
4. Сначала объединение первой таблицы с результатами выборки второй, а затем выборка по столбцам первой таблицы.

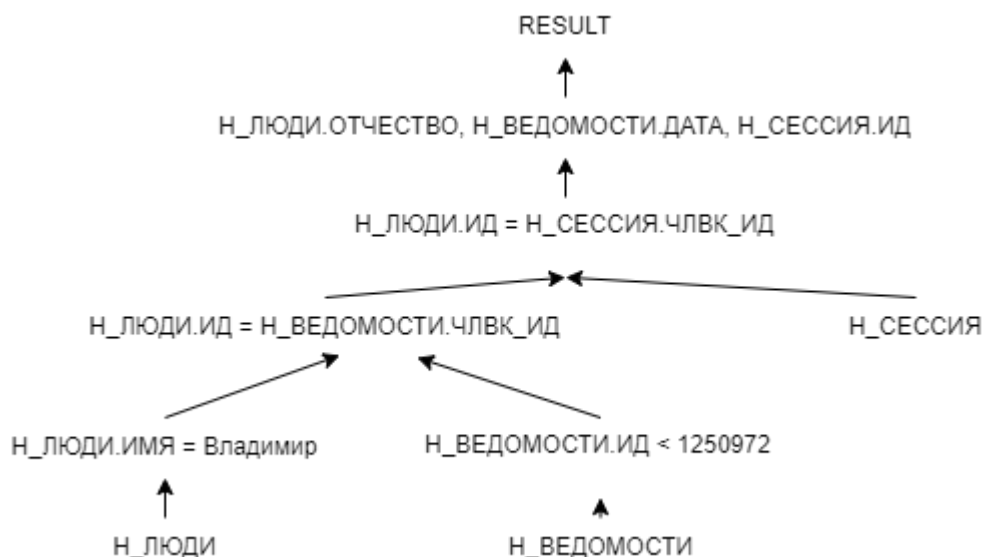
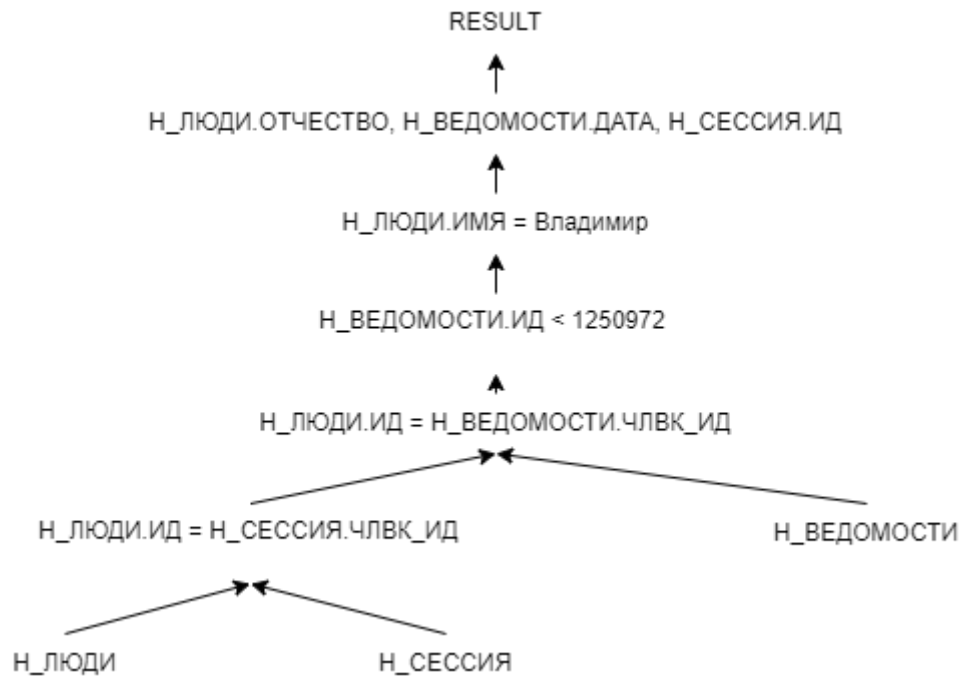
На мой взгляд, оптимальнее является второй, так как мы объединяем не все строки, а только нужные нам выборки из них, следовательно, размер промежуточного отношения получится лучше, что положительно повлияет на производительность.

Изобразим каждый план в виде диаграммы:





Аналогично для второго запроса:



Здесь мне оптимальнее кажется второй вариант, так как вместо полного объединения таблиц происходит объединение только после необходимой выборки по таблице, что, как и в случае для первого запроса, позволяет сделать вывод о повышении производительности.

### Индексы:

Для первого запроса:

1. CREATE INDEX "ИД\_ВЕДОМОСТИ\_ЧЛВК\_ИД" ON "H\_ВЕДОМОСТИ" USING btree ("ЧЛВК\_ИД");

Добавление такого индекса позволяет существенно ускорить запросы, так как по столбцу ЧЛВК\_ИД идет выборка с использованием операторов равенства и сравнения, а данный индекс даст возможность выполнять операции поиска за  $O(\log n)$ .

Для второго запроса:

1. CREATE INDEX “ИД\_ЛЮДИ\_ИД” ON “Н\_ЛЮДИ” USING hash(“ИД”);
2. CREATE INDEX “ИД\_ЛЮДИ\_ФАМИЛИЯ” ON “Н\_ЛЮДИ” USING btree(“ФАМИЛИЯ”);
3. CREATE INDEX “ИД\_ВЕДОМОСТИ\_ИД” ON “Н\_ВЕДОМОСТИ” USING btree(“ИД”);

Данные индексы позволят ускорить скорость выполнения запросов, так как по данным атрибутам идет выборка с использованием все тех же операторов сравнения, а так же операции соединения таблиц.

### Результат работы запросов:

#### Первого:

```
[s312621@helios ~]$ psql -h pg -d ucheb -a -f inf-systems-and-databases/lab4/lab4-q1.sql
```

Пароль пользователя s312621:

/\*

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

Таблицы: Н\_ОЦЕНКИ, Н\_ВЕДОМОСТИ.

Вывести атрибуты: Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД.

Фильтры (AND):

а) Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ = зачет.

б) Н\_ВЕДОМОСТИ.ИД > 1457443.

с) Н\_ВЕДОМОСТИ.ИД = 39921.

Вид соединения: LEFT JOIN

\*/

EXPLAIN ANALYZE

SELECT

```

ОЦЕНКИ.КОД,
ВЕДОМОСТИ.ЧЛВК_ИД
FROM
(
SELECT
*
FROM
H_ОЦЕНКИ
WHERE
ПРИМЕЧАНИЕ = 'зачет'
) AS ОЦЕНКИ
LEFT JOIN (
SELECT
*
FROM
H_ВЕДОМОСТИ
WHERE
ИД = 39921
AND ИД > 1457443
) AS ВЕДОМОСТИ ON ВЕДОМОСТИ.ОЦЕНКА = ОЦЕНКИ.КОД;

```

#### QUERY PLAN

```

-----
-----
Nested Loop Left Join (cost=0.42..9.56 rows=1 width=38) (actual time=0.027..0.029 rows=1
loops=1)

```



Join Filter: (("H\_ВЕДОМОСТИ"."ОЦЕНКА")::text = ("H\_ОЦЕНКИ"."КОД")::text)

-> Seq Scan on "H\_ОЦЕНКИ" (cost=0.00..1.11 rows=1 width=34) (actual time=0.022..0.023 rows=1 loops=1)

Filter: (("ПРИМЕЧАНИЕ")::text = 'зачет')::text)

Rows Removed by Filter: 8

-> Index Scan using "БЕД\_ПК" on "H\_ВЕДОМОСТИ" (cost=0.42..8.44 rows=1 width=10) (actual time=0.002..0.002 rows=0 loops=1)

Index Cond: (("ИД" > 1457443) AND ("ИД" = 39921))

Planning Time: 1.287 ms

Execution Time: 0.124 ms

(9 строк)

```
[s312621@helios ~]$ psql -h pg -d ucheb -a -f inf-systems-and-databases/lab4/lab4-q2.sql
```

**Второго:**

```
[s312621@helios ~]$ psql -h pg -d ucheb -a -f inf-systems-and-databases/lab4/lab4-q2.sql
```

Пароль пользователя s312621:

/\*

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

Таблицы: H\_ЛЮДИ, H\_ВЕДОМОСТИ, H\_СЕССИЯ.

Вывести атрибуты: H\_ЛЮДИ.ОТЧЕСТВО, H\_ВЕДОМОСТИ.ДАТА, H\_СЕССИЯ.ИД.

Фильтры (AND):

а) H\_ЛЮДИ.ФАМИЛИЯ = Владимир.

б) H\_ВЕДОМОСТИ.ИД < 1250972

Вид соединения: INNER JOIN.

\*/

EXPLAIN ANALYZE

```
SELECT
    ЛЮДИ.ОТЧЕСТВО,
    ВЕДОМОСТИ.ДАТА,
    СЕССИЯ.ИД
FROM
    (
        SELECT
            *
        FROM
            Н_ЛЮДИ
        WHERE
            ИМЯ = 'Владимир'
    ) AS ЛЮДИ
INNER JOIN (
    SELECT
        *
    FROM
        Н_ВЕДОМОСТИ
    WHERE
        ИД < 1250972
    ) AS ВЕДОМОСТИ ON ЛЮДИ.ИД = ВЕДОМОСТИ.ЧЛВК_ИД
INNER JOIN Н_СЕССИЯ AS СЕССИЯ ON ЛЮДИ.ИД = СЕССИЯ.ЧЛВК_ИД;
```

QUERY PLAN

-----  
-----

Nested Loop (cost=0.59..762.56 rows=2671 width=32) (actual time=2.962..3.997 rows=412 loops=1)

Join Filter: ("H\_ЛЮДИ"."ИД" = "H\_ВЕДОМОСТИ"."ЧЛВК\_ИД")

-> Nested Loop (cost=0.29..279.41 rows=70 width=32) (actual time=0.098..2.950 rows=242 loops=1)

-> Seq Scan on "H\_СЕССИЯ" "СЕССИЯ" (cost=0.00..108.52 rows=3752 width=8) (actual time=0.019..0.462 rows=3752 loops=1)

-> Memoize (cost=0.29..0.44 rows=1 width=24) (actual time=0.000..0.000 rows=0 loops=3752)

Cache Key: "СЕССИЯ"."ЧЛВК\_ИД"

Cache Mode: logical

Hits: 3571 Misses: 181 Evictions: 0 Overflows: 0 Memory Usage: 13kB

-> Index Scan using "ЧЛВК\_ПК" on "H\_ЛЮДИ" (cost=0.28..0.43 rows=1 width=24) (actual time=0.003..0.003 rows=0 loops=181)

Index Cond: ("ИД" = "СЕССИЯ"."ЧЛВК\_ИД")

Filter: (("ИМЯ")::text = 'Владимир'::text)

Rows Removed by Filter: 1

-> Index Scan using "ВЕД\_ЧЛВК\_FK\_IFK" on "H\_ВЕДОМОСТИ" (cost=0.29..6.15 rows=60 width=12) (actual time=0.001..0.004 rows=2 loops=242)

Index Cond: ("ЧЛВК\_ИД" = "СЕССИЯ"."ЧЛВК\_ИД")

Filter: ("ИД" < 1250972)

Planning Time: 2.096 ms

Execution Time: 4.150 ms

(17 строк)

**Вывод:** в ходе данной работы я ознакомился с новыми для себя возможностями PostgreSQL, сравнил некоторые запросы между собой, исследовал возможные варианты их исполнения, рассмотрел принципы работы индексов и написал несколько. Заметил, что использование индексов эффективно не всегда, так как в некоторых случаях накладные

расходы на работу с ними перебивают пользу, ими приносимую. Так же в лабораторной работе проанализировал запросы с помощью EXPLAIN ANALYZE.