

# Университет ИТМО

## Лабораторная работа №4 по базам данных Вариант #8696

Выполнил  
Студент группы Р3112  
Шаматульский Роман Константинович  
Преподаватель  
Максимов Андрей Николаевич

Санкт-Петербург, 2024

## Оглавление

<i>Текст задания .....</i>	<i>3</i>
<i>Реализация запросов.....</i>	<i>4</i>
<i>Возможные индексы для запросов.....</i>	<i>5</i>
<i>Explain analyze .....</i>	<i>9</i>
<i>Вывод.....</i>	<i>10</i>

## Текст задания

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. Изменятся 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\_ОЦЕНКИ, Н\_ВЕДОМОСТИ.  
Вывести атрибуты: Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ДАТА.  
Фильтры (AND):  
а) Н\_ОЦЕНКИ.КОД > 3.  
б) Н\_ВЕДОМОСТИ.ИД < 1490007.  
Вид соединения: LEFT JOIN.
2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.  
Вывести атрибуты: Н\_ЛЮДИ.ИД, Н\_ВЕДОМОСТИ.ДАТА, Н\_СЕССИЯ.ДАТА.  
Фильтры (AND):  
а) Н\_ЛЮДИ.ОТЧЕСТВО > Александрович.  
б) Н\_ВЕДОМОСТИ.ДАТА = 2010-06-18.  
с) Н\_СЕССИЯ.ИД < 1975.  
Вид соединения: LEFT JOIN.

## Реализация запросов

--- 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.ДАТА.

--- Фильтры (AND):

--- а) Н\_ОЦЕНКИ.КОД > 3.

--- б) Н\_ВЕДОМОСТИ.ИД < 1490007.

--- Вид соединения: LEFT JOIN.

```
SELECT Н_ОЦЕНКИ.КОД, Н_ВЕДОМОСТИ.ДАТА FROM Н_ОЦЕНКИ
LEFT JOIN Н_ВЕДОМОСТИ ON Н_ОЦЕНКИ.КОД = Н_ВЕДОМОСТИ.ОЦЕНКА
WHERE Н_ОЦЕНКИ.КОД > '3' AND Н_ВЕДОМОСТИ.ИД < 1490007;
```

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

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

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

--- Фильтры (AND):

--- а) Н\_ЛЮДИ.ОТЧЕСТВО > Александрович.

--- б) Н\_ВЕДОМОСТИ.ДАТА = 2010-06-18.

--- с) Н\_СЕССИЯ.ИД < 1975.

--- Вид соединения: LEFT JOIN.

```
SELECT Н_ЛЮДИ.ИД, Н_ВЕДОМОСТИ.ДАТА, Н_СЕССИЯ.ДАТА
FROM Н_ЛЮДИ
LEFT JOIN Н_ВЕДОМОСТИ ON Н_ВЕДОМОСТИ.ЧЛВК_ИД = Н_ЛЮДИ.ИД
LEFT JOIN Н_СЕССИЯ ON Н_ВЕДОМОСТИ.СЭС_ИД = Н_СЕССИЯ.СЭС_ИД
WHERE Н_ЛЮДИ.ОТЧЕСТВО > 'Александрович'
AND Н_ВЕДОМОСТИ.ДАТА = '2010-06-18'
AND Н_СЕССИЯ.ИД < 1975;
```

## Возможные индексы для запросов

Для первого запроса:

- Индекс (hash) на Н\_ВЕДОМОСТИ.ОЦЕНКА. Добавление индекса на этот атрибут позволит быстрее выполнять соединение.

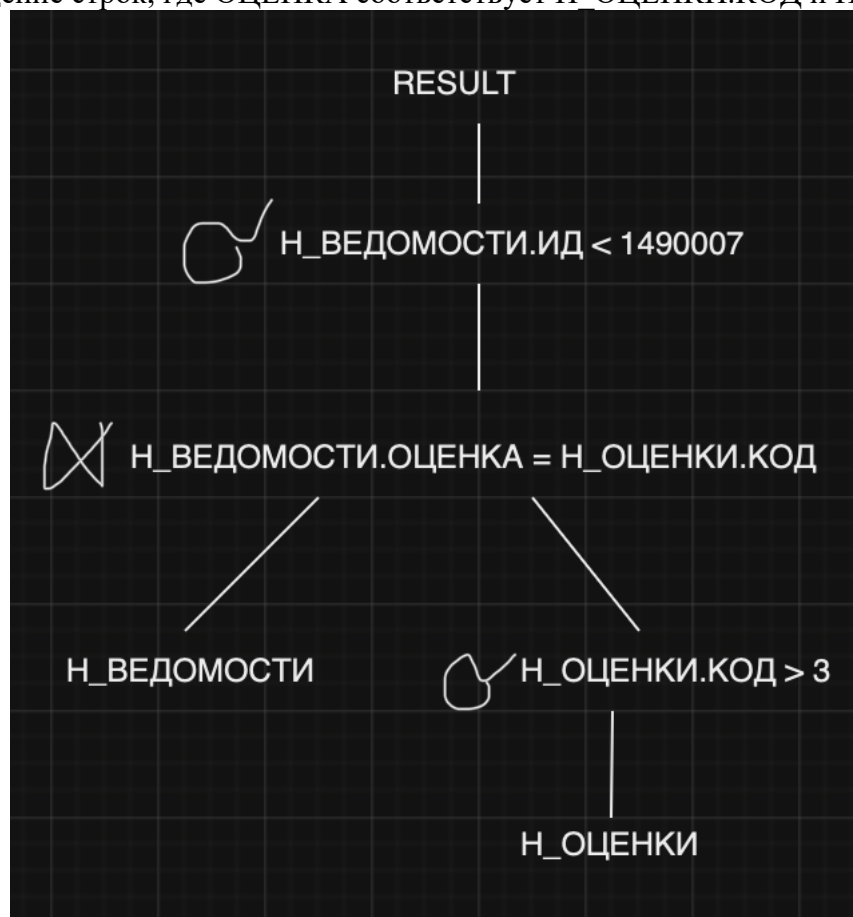
```
CREATE INDEX idx_ведомости_оценка ON Н_ВЕДОМОСТИ USING HASH (ОЦЕНКА);
```

- Другие индексы могут не понадобиться, так как Н\_ВЕДОМОСТИ.ИД содержит уникальные значения, а таблица Н\_ОЦЕНКИ небольшая, полный скан и так будет проходить быстро.

Возможные планы выполнения запроса без индексов:

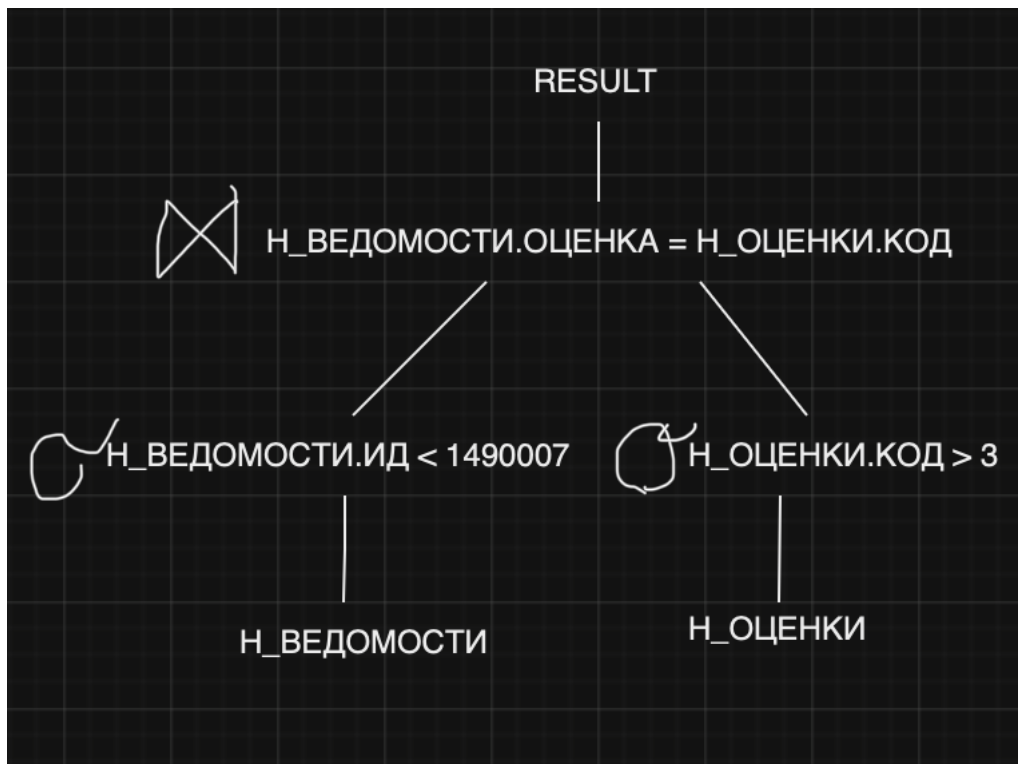
А)

1. Полный скан Н\_ОЦЕНКИ, проверка каждой строки на соответствие КОД > '3'
2. Для каждой подходящей строчки из Н\_ОЦЕНКИ: полный скан Н\_ВЕДОМОСТИ, нахождение строк, где ОЦЕНКА соответствует Н\_ОЦЕНКИ.КОД и ИД < 1490007



В)

1. Полный скан Н\_ВЕДОМОСТИ и фильтрация по ИД < 1490007, временное сохранение.
2. Полный скан Н\_ОЦЕНКИ, поиск строк с КОД > '3'.
3. Проход по сохраненным строчкам из Н\_ВЕДОМОСТИ и сопоставление с Н\_ОЦЕНКИ.



Более оптимальный план – В, так как частое полное сканирование H\_ВЕДОМОСТИ неэффективно из-за того, что она содержит большой объем данных.

Благодаря добавлению предложенного индекса соединение двух таблиц будет выполняться быстрее, так как не будет необходимости в полном сканировании сохраненных строчек из H\_ВЕДОМОСТИ.

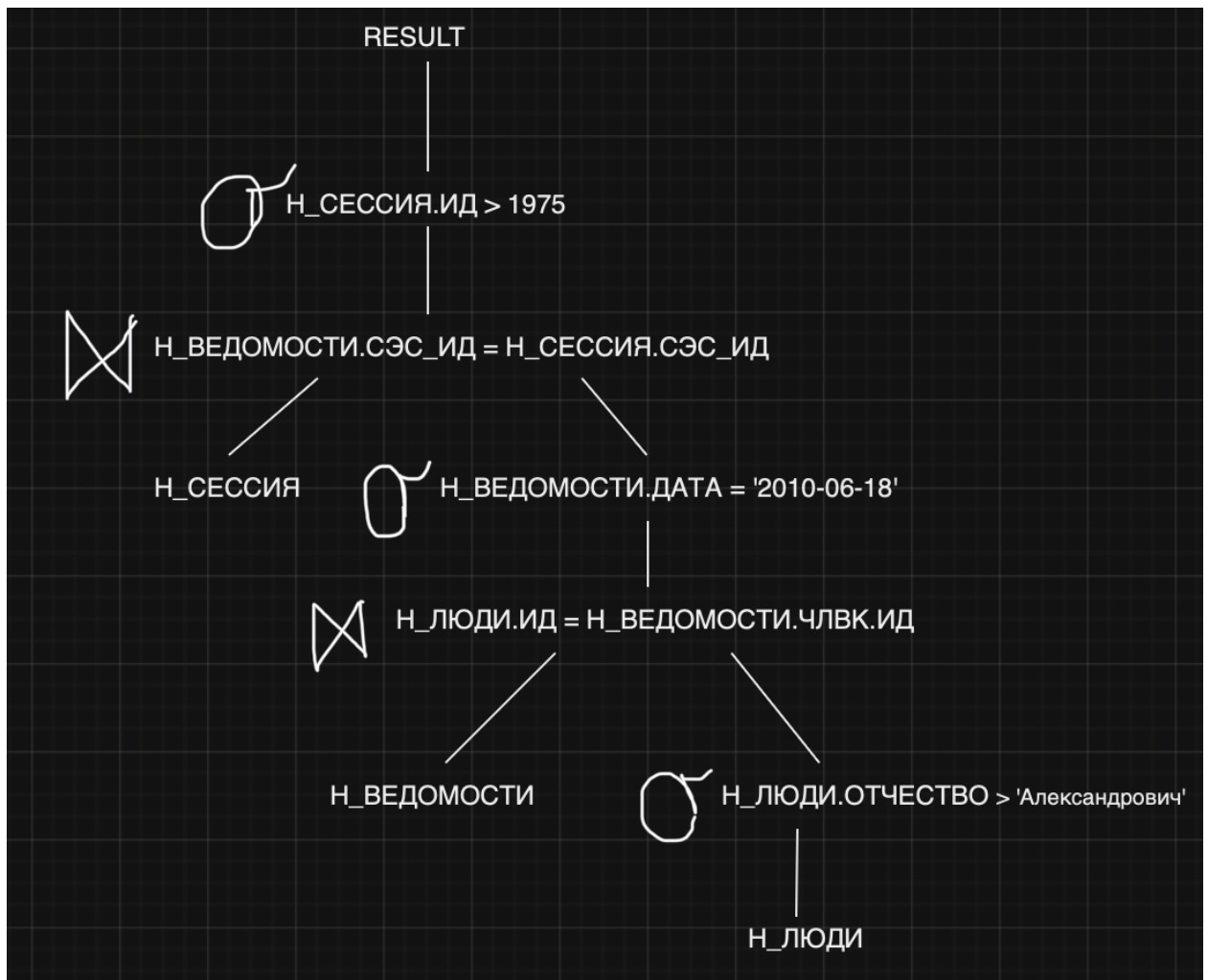
**Для второго запроса:**

- Индекс (b-tree) на атрибуте ОТЧЕСТВО. Это ускорит фильтрацию строк с условием H\_ЛЮДИ.ОТЧЕСТВО > 'Александрович'.  
**CREATE INDEX idx\_люди\_отчество ON H\_ЛЮДИ (ОТЧЕСТВО);**
- Индекс (hash) на атрибуте ДАТА. Это ускорит фильтрацию строк с условием H\_ВЕДОМОСТИ.ДАТА = '2010-06-18'.  
**CREATE INDEX idx\_ведомости\_дата ON H\_ВЕДОМОСТИ USING hash (ДАТА);**

**Возможные планы выполнения запроса без индексов:**

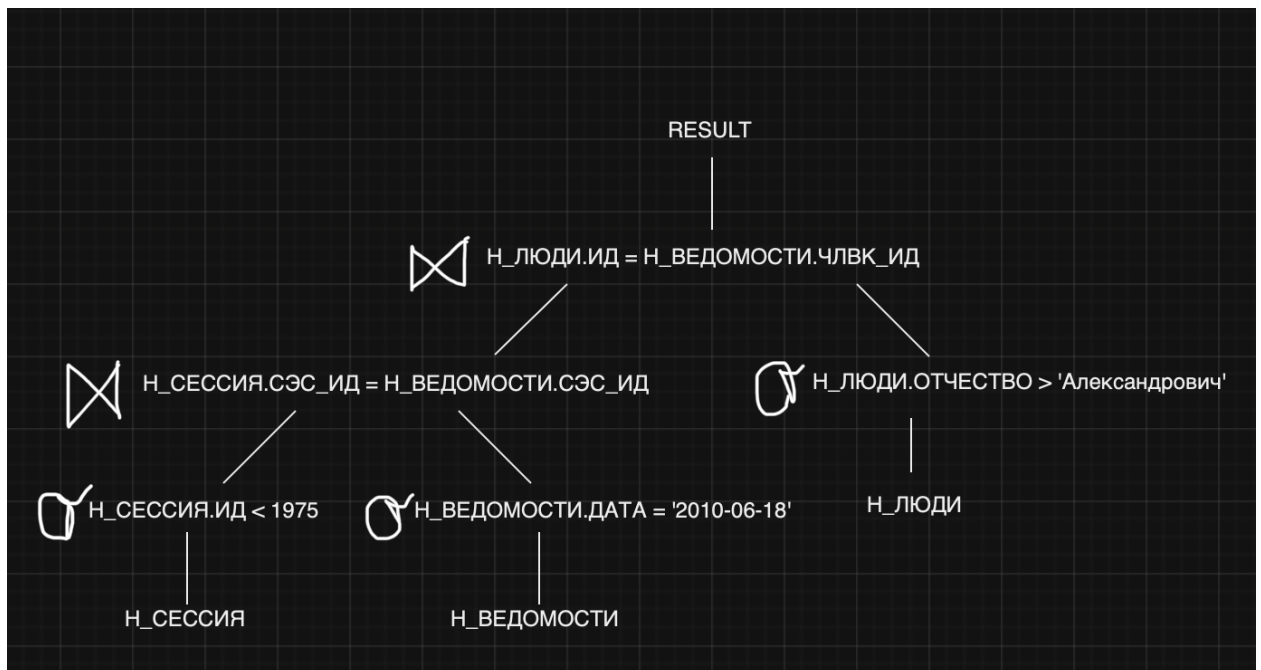
А)

1. Полное сканирование H\_ЛЮДИ: проверка каждой строки на условие ОТЧЕСТВО > 'Александрович'.
2. Полное сканирование H\_ВЕДОМОСТИ для каждой подходящей строки из H\_ЛЮДИ: поиск строк, где ЧЛВК\_ИД = ИД и ДАТА = '2010-06-18'.
3. Полное сканирование H\_СЕССИЯ для каждой подходящей строки из H\_ВЕДОМОСТИ: поиск строк, где H\_СЕССИЯ.СЭС\_ИД = H\_ВЕДОМОСТИ.СЭС\_ИД и ИД < 1975.



В)

1. Полное сканирование H\_СЕССИЯ с фильтрацией по ИД < 1975: временное хранение результатов для быстрого доступа.
2. Полное сканирование H\_ВЕДОМОСТИ с фильтрацией по ДАТА = '2010-06-18': соединение с предварительно отфильтрованными результатами из H\_СЕССИЯ.
3. Полное сканирование H\_ЛЮДИ с фильтрацией по ОТЧЕСТВО > 'Александрович': соединение с предварительно отфильтрованными результатами из H\_ВЕДОМОСТИ и H\_СЕССИЯ.



Более оптимальный план – В, так как полное сканирование для таблиц выполняется один раз, вместо полного скана целой таблицы для сопоставления ее с одной строчкой другой таблицы.

Благодаря добавления предложенных индексов, индексный скан будет быстро находить нужные строчки в H\_ВЕДОМОСТИ и H\_ЛЮДИ.



## Explain analyze

Для первого запроса:

```
Hash Join (cost=1.21..7706.57 rows=197724 width=13) (actual time=0.035..95.543
rows=187009 loops=1)
  Hash Cond: (("H_ВЕДОМОСТИ"."ОЦЕНКА")::text = ("H_ОЦЕНКИ"."КОД")::text)
    -> Seq Scan on "H_ВЕДОМОСТИ" (cost=0.00..6846.50 rows=222439 width=14)
    (actual time=0.010..35.200 rows=222439 loops=1)
      Filter: ("ИД" < 1490007)
      Rows Removed by Filter: 1
    -> Hash (cost=1.11..1.11 rows=8 width=5) (actual time=0.017..0.019 rows=7 loops=1)
      Buckets: 1024 Batches: 1 Memory Usage: 9kB
      -> Seq Scan on "H_ОЦЕНКИ" (cost=0.00..1.11 rows=8 width=5) (actual
time=0.004..0.008 rows=7 loops=1)
        Filter: (("КОД")::text > '3')::text)
        Rows Removed by Filter: 2
  Planning Time: 0.281 ms
  Execution Time: 104.696 ms
```

Для второго запроса:

```
Nested Loop (cost=126.40..382.09 rows=6 width=20) (actual time=0.680..0.682 rows=0
loops=1)
  -> Hash Join (cost=126.12..341.47 rows=7 width=20) (actual time=0.680..0.681
rows=0 loops=1)
    Hash Cond: ("H_ВЕДОМОСТИ"."СЭС_ИД" = "H_СЕССИЯ"."СЭС_ИД")
    -> Index Scan using "ВЕД_ДАТА_I" on "H_ВЕДОМОСТИ" (cost=0.29..214.94
rows=72 width=16) (actual time=0.019..0.075 rows=141 loops=1)
      Index Cond: ("ДАТА" = '2010-06-18 00:00:00'::timestamp without time zone)
    -> Hash (cost=117.90..117.90 rows=634 width=12) (actual time=0.571..0.571
rows=632 loops=1)
      Buckets: 1024 Batches: 1 Memory Usage: 36kB
      -> Seq Scan on "H_СЕССИЯ" (cost=0.00..117.90 rows=634 width=12) (actual
time=0.011..0.458 rows=632 loops=1)
        Filter: ("ИД" < 1975)
        Rows Removed by Filter: 3120
    -> Index Scan using "ЧЛВК_РК" on "H_ЛЮДИ" (cost=0.28..5.80 rows=1 width=4)
(never executed)
      Index Cond: ("ИД" = "H_ВЕДОМОСТИ"."ЧЛВК_ИД")
      Filter: (("ОТЧЕСТВО")::text > 'Александрович')::text)
  Planning Time: 1.218 ms
  Execution Time: 0.738 ms
```

## Вывод

В ходе выполнения данной лабораторной работы я научился работать с реляционной алгеброй, строить план выполнения запроса. Полученные мной знания лягут в основу дальнейшего изучения баз данных.