

НИУ ИТМО

Мегафакультет компьютерных технологий и управления

Факультет программной инженерии и компьютерной техники

Базы данных

Лабораторная работа №4

Вариант 6892

Выполнил: К. Кравцов, Р3211

Преподаватель: Николаев В. В.

Санкт-Петербург, 2023

## 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).

# Запросы на языке SQL

<https://github.com/killreal777/db-labs/blob/main/db-lab4/sql/queries.sql>

```
-- 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.ИД.
-- Фильтры (AND):
-- а) Н_ОЦЕНКИ.ПРИМЕЧАНИЕ < неявка.
-- б) Н_ВЕДОМОСТИ.ДАТА < 2010-06-18.
-- Вид соединения: INNER JOIN.

select оценки."КОД", ведомости."ИД"
from "Н_ОЦЕНКИ" оценки
join "Н_ВЕДОМОСТИ" ведомости on (оценки."КОД" = ведомости."ОЦЕНКА")
where
    (оценки."ПРИМЕЧАНИЕ" < 'неявка') and
    (ведомости."ДАТА" < to_date('2010-06-18', 'YYYY-MM-DD'));

-- 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.ИМЯ, Н_ОБУЧЕНИЯ.ЧЛВК_ИД, Н_УЧЕНИКИ.ИД.
-- Фильтры: (AND)
-- а) Н_ЛЮДИ.ФАМИЛИЯ < Иванов.
-- б) Н_ОБУЧЕНИЯ.НЗК < 933232.
-- с) Н_УЧЕНИКИ.ИД > 39.
-- Вид соединения: LEFT JOIN.

select люди."ИМЯ", обучения."ЧЛВК_ИД", ученики."ИД"
from "Н_ЛЮДИ" люди
left join "Н_ОБУЧЕНИЯ" обучения on (обучения."ЧЛВК_ИД" = люди."ИД")
left join "Н_УЧЕНИКИ" ученики on (обучения."ЧЛВК_ИД" = ученики."ЧЛВК_ИД")
where
    (люди."ФАМИЛИЯ" < 'Иванов') and
    (обучения."НЗК" < '933232') and
    (ученики."ИД" > 39);
```

# Индексы

## Query 1

- Hash-index -> Н\_ВЕДОМОСТИ.ОЦЕНКА  
для ускорения соединения таблиц.  
*Н\_ВЕДОМОСТИ.ОЦЕНКА – внешний ключ, используется прямое сравнение при соединении таблиц.*
- B-tree index -> Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ,  
B-tree index -> Н\_ВЕДОМОСТИ.ДАТА  
для ускорения выборки.  
*Используются сравнения, большое количество строк, предполагается большое количество уникальных значений.*

## Query 2

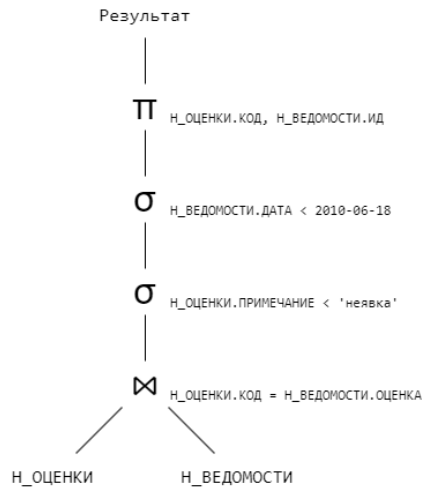
- Hash-index -> Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД,  
Hash-index -> Н\_УЧЕНИКИ.ЧЛВК\_ИД  
для ускорения соединения таблиц.  
*Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.ЧЛВК\_ИД – внешние ключи, используются прямые сравнения при соединении таблиц.*
- B-tree index -> Н\_ЛЮДИ.ФАМИЛИЯ,  
B-tree index -> Н\_ОБУЧЕНИЯ.НЗК  
для ускорения выборки.  
*Используются сравнения, большое количество строк, большое количество уникальных значений.*

Предполагается, что на первичных ключах индексы созданы автоматически.

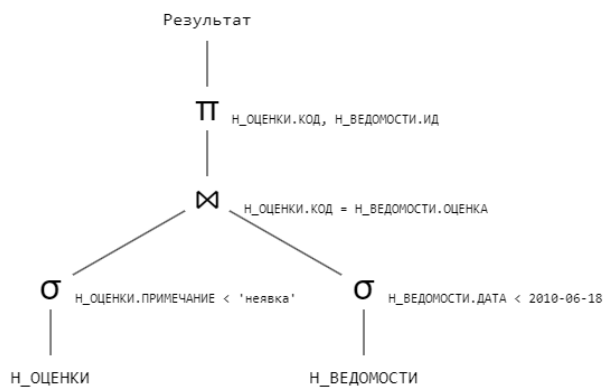
# Планы выполнения запросов

## Query 1

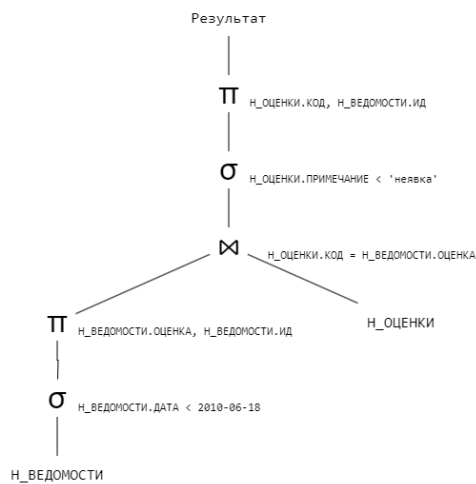
### 1) Соединение таблиц в первую очередь



### 2) Выборка до соединения таблиц



### 3) Выборки и проекции до соединения таблиц в левых ветвях

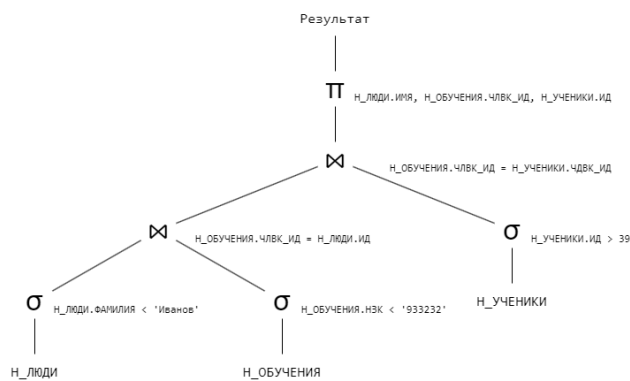


## Query 2

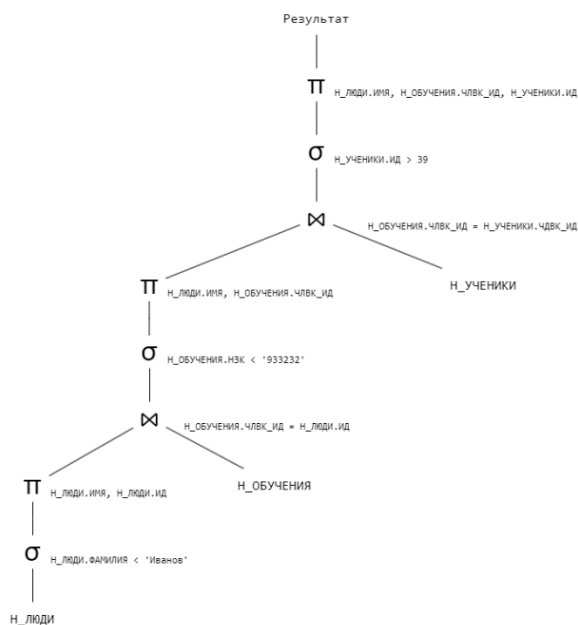
### 1) Соединение таблиц в первую очередь



### 2) Выборка до соединения таблиц



### 3) Выборки и проекции до соединения таблиц в левых ветвях



Оптимальными планами являются третьи из-за конвейерной обработки, ранних выборок и проекций.

При добавлении индексов планы выполнения запросов не должны поменяться.

## EXPLAIN ANALYZE

```
uscheb=> explain analyze select оценки."КОД", ведомости."ИД"
from "Н_ОЦЕНКИ" оценки
join "Н_ВЕДОМОСТИ" ведомости on (оценки."КОД" = ведомости."ОЦЕНКА")
where (оценки."ПРИМЕЧАНИЕ" < 'неявка') and (ведомости."ДАТА" < to_date('2010-06-18', 'YYYY-MM-DD'));
                                QUERY PLAN

-----
Hash Join  (cost=1.16..8161.20 rows=87186 width=9) (actual time=0.081..164.949 rows=100528 loops=1)
  Hash Cond: (("ведомости"."ОЦЕНКА")::text = (оценки."КОД")::text)
  -> Seq Scan on "Н_ВЕДОМОСТИ" "ведомости"  (cost=0.00..7402.60 rows=196169 width=10) (actual time=0.018..128.336 rows=196568 loops=1)
      Filter: ("ДАТА" < to_date('2010-06-18'::text, 'YYYY-MM-DD'::text))
      Rows Removed by Filter: 25872
  -> Hash  (cost=1.11..1.11 rows=4 width=5) (actual time=0.027..0.028 rows=4 loops=1)
      Buckets: 1024 Batches: 1 Memory Usage: 9kB
      -> Seq Scan on "Н_ОЦЕНКИ" "оценки"  (cost=0.00..1.11 rows=4 width=5) (actual time=0.006..0.011 rows=4 loops=1)
          Filter: (("ПРИМЕЧАНИЕ")::text < 'неявка'::text)
          Rows Removed by Filter: 5
Planning Time: 1.405 ms
Execution Time: 169.879 ms
(12 строк)
```

```
uscheb=> explain analyze select люди."ИМЯ", обучения."ЧЛВК_ИД", ученики."ИД"
from "Н_ЛЮДИ" люди
left join "Н_ОБУЧЕНИЯ" обучения on (обучения."ЧЛВК_ИД" = люди."ИД")
left join "Н_УЧЕНИКИ" ученики on (обучения."ЧЛВК_ИД" = ученики."ЧЛВК_ИД")
where (люди."ФАМИЛИЯ" < 'Иванов') and (обучения."НЗК" < '933232') and (ученики."ИД" > 39);
                                QUERY PLAN

-----
Hash Join  (cost=322.27..1290.96 rows=4903 width=21) (actual time=3.732..11.036 rows=4409 loops=1)
  Hash Cond: ("ученики"."ЧЛВК_ИД" = "обучения"."ЧЛВК_ИД")
  -> Seq Scan on "Н_УЧЕНИКИ" "ученики"  (cost=0.00..832.39 rows=23272 width=8) (actual time=0.010..4.013 rows=23272 loops=1)
      Filter: ("ИД" > 39)
      Rows Removed by Filter: 39
  -> Hash  (cost=309.21..309.21 rows=1045 width=21) (actual time=3.701..3.703 rows=1011 loops=1)
      Buckets: 2048 Batches: 1 Memory Usage: 71kB
      -> Hash Join  (cost=180.65..309.21 rows=1045 width=21) (actual time=1.309..3.488 rows=1011 loops=1)
          Hash Cond: ("обучения"."ЧЛВК_ИД" = "люди"."ИД")
          -> Seq Scan on "Н_ОБУЧЕНИЯ" "обучения"  (cost=0.00..119.76 rows=3346 width=4) (actual time=0.008..1.616 rows=3347 loops=1)
              Filter: (("НЗК")::text < '933232'::text)
              Rows Removed by Filter: 1674
          -> Hash  (cost=160.66..160.66 rows=1599 width=17) (actual time=1.279..1.280 rows=1573 loops=1)
              Buckets: 2048 Batches: 1 Memory Usage: 95kB
              -> Bitmap Heap Scan on "Н_ЛЮДИ" "люди"  (cost=40.67..160.66 rows=1599 width=17) (actual time=0.545..0.984 rows=1573 loops=1)
                  Recheck Cond: (("ФАМИЛИЯ")::text < 'Иванов'::text)
                  Heap Blocks: exact=99
              -> Bitmap Index Scan on "ФАМ_ЛЮД"  (cost=0.00..40.27 rows=1599 width=0) (actual time=0.529..0.529 rows=1573 loops=1)
                  Index Cond: (("ФАМИЛИЯ")::text < 'Иванов'::text)
Planning Time: 16.415 ms
Execution Time: 11.349 ms
(21 строка)
```