

Федеральное государственное автономное образовательное учреждение высшего  
образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

# Отчёт

по лабораторной работе №4

вариант 1335

Выполнил: Тимошкин Р. В., группа Р3131

Преподаватель: Харитонов А. Е.

# 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.

Изменяться 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:

Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.

Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД.

Фильтры (AND):

а) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ > Презачет.

б) Н\_ВЕДОМОСТИ.ДАТА > 2010-06-18.

с) Н\_ВЕДОМОСТИ.ДАТА = 1998-01-05.

Вид соединения: INNER 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.ИМЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО.

Фильтры: (AND)

а) Н\_ЛЮДИ.ИМЯ = Роман.

б) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 163276.

Вид соединения: LEFT JOIN.

# Реализация запросов на SQL

```
-- 1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
-- Н_ТИПЫ_ВЕДОМОСТЕЙ, Н_ВЕДОМОСТИ.
-- Вывести атрибуты: Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД, Н_ВЕДОМОСТИ.ЧЛВК_ИД.
-- Фильтры (AND):
-- а) Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ > Перезачет.
-- б) Н_ВЕДОМОСТИ.ДАТА > 2010-06-18.
-- с) Н_ВЕДОМОСТИ.ДАТА = 1998-01-05.
-- Вид соединения: INNER JOIN.
```

```
SELECT t."ИД", v."ЧЛВК_ИД"
FROM "Н_ТИПЫ_ВЕДОМОСТЕЙ" AS t
JOIN "Н_ВЕДОМОСТИ" AS v ON t."ИД" = v."ТВ_ИД"
WHERE t."НАИМЕНОВАНИЕ" > 'Перезачет' AND
      v."ДАТА" > '2010-06-18' AND
      v."ДАТА" = '1998-01-05';
```

```
-- 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.НАЧАЛО.
-- Фильтры: (AND)
-- а) Н_ЛЮДИ.ИМЯ = Роман.
-- б) Н_ОБУЧЕНИЯ.ЧЛВК_ИД = 163276.
-- Вид соединения: LEFT JOIN.
```

```
SELECT p."ИМЯ", edu."ЧЛВК_ИД", st."НАЧАЛО"
FROM "Н_ЛЮДИ" AS p
LEFT JOIN "Н_ОБУЧЕНИЯ" AS edu ON p."ИД" = edu."ЧЛВК_ИД"
LEFT JOIN "Н_УЧЕНИКИ" AS st ON edu."ЧЛВК_ИД" = st."ЧЛВК_ИД"
WHERE p."ИМЯ" = 'Роман' AND
      edu.ЧЛВК_ИД = '163276';
```

## Уменьшение времени выполнения запросов

### Запрос №1

#### Индексы, добавление которых уменьшит время выполнения запроса

На таблице Н\_ТИПЫ\_ВЕДОМОСТЕЙ:

- Индекс на атрибуте НАИМЕНОВАНИЕ ускорит фильтрацию строк с условием  
Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ > Перезачет

На таблице Н\_ВЕДОМОСТИ

- Индекс на атрибуте ДАТА ускорит фильтрацию строк с условиями Н\_ВЕДОМОСТИ.ДАТА > 2010-06-18  
и Н\_ВЕДОМОСТИ.ДАТА = 1998-01-05
- Индекс на атрибуте ТВ\_ИД ускорит соединение таблиц по атрибуту ТВ\_ИД

#### Возможные планы выполнения запроса без индексов

План 1:

- Полный скан таблицы Н\_ТИПЫ\_ВЕДОМОСТЕЙ

- Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию Н\_ВЕДОМОСТИ.ДАТА > 2010-06-18
- Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ\_ИД
- Фильтрация результата соединения по условию Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ > Презачет
- Фильтрация результата соединения по условию Н\_ВЕДОМОСТИ.ДАТА = 1998-01-05

План 2:

- Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию Н\_ВЕДОМОСТИ.ДАТА > 2010-06-18 AND Н\_ВЕДОМОСТИ.ДАТА = 1998-01-05
- Полный скан таблицы Н\_ТИПЫ\_ВЕДОМОСТЕЙ с применением фильтра по условию Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ > Презачет
- Соединение таблиц с использованием Hash Join по атрибуту ТВ\_ИД

Оптимальным является план 2, потому что фильтрация данных выполняется до соединения таблиц, что уменьшает количество строк для обработки

При добавлении индексов вместо полного скана таблиц будет использоваться индексный скан

## План выполнения

Уберем из фильтров условие Н\_ВЕДОМОСТИ.ДАТА = 1998-01-05, иначе нет подходящих записей

```
Hash Join (cost=300.41..4'859.84 rows=17'467 width=8) (actual time=1.061..9.652 rows=3'493 loops=1)
  Hash Cond: (v."ТВ_ИД" = t."ИД")
    -> Bitmap Heap Scan on "Н_ВЕДОМОСТИ" v (cost=299.35..4'692.85 rows=26'200 width=8) (actual time=0.978..5.473 rows=25'731 loops=1)
      Recheck Cond: ("ДАТА" > '2010-06-18 00:00:00'::timestamp without time zone)
      Heap Blocks: exact=651
      -> Bitmap Index Scan on "ВЕД_ДАТА_I" (cost=0.00..292.80 rows=26'200 width=0) (actual time=0.871..0.871 rows=25'731 loops=1)
        Index Cond: ("ДАТА" > '2010-06-18 00:00:00'::timestamp without time zone)
    -> Hash (cost=1.04..1.04 rows=2 width=4) (actual time=0.018..0.019 rows=1 loops=1)
      Buckets: 1'024 Batches: 1 Memory Usage: 9kB
      -> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ" t (cost=0.00..1.04 rows=2 width=4) (actual time=0.010..0.011 rows=1 loops=1)
        Filter: (("НАИМЕНОВАНИЕ")::text > 'Презачет'::text)
        Rows Removed by Filter: 2
```

Planning Time: 1.181 ms

Execution Time: 9.898 ms

### #1 Hash Join

on v."TB\_ИД" = t."ИД"

**Hash Join Node** joins two record sets by hashing one of them (using a **Hash Scan**).

General IO & Buffers Output Workers Misc

Timing: 4.16ms | 42%

Rows: 3,493 (Planned: 17,467) | ↑ over estimated by 5 ×

Cost: 166 (Total: 4,860)

### #2 Bitmap Heap Scan

on "Н\_ВЕДОМОСТИ" as v

**Bitmap Heap Scan Node** searches through the pages returned by the **Bitmap Index Scan** for relevant rows.

General IO & Buffers Output Workers Misc

Timing: 4.6ms | 46%

Rows: 25,731 (Planned: 26,200) | ↑ over estimated by 1 ×

Cost: 4,400 (Total: 4,690)

### #3 Bitmap Index Scan

using "ВЕД\_ДАТА\_I"

**Bitmap Index Scan Node** uses a **Bitmap Index** (index which uses 1 bit per page) to find all relevant pages. Results of this node are fed to the **Bitmap Heap Scan**.

General IO & Buffers Output Workers Misc

Actual Startup Time 0.871ms

Actual Total Time 0.871ms

Index Cond ("ДАТА" > '2010-06-18 00:00:00'::timestamp without time zone)

\* Calculated value

### #4 Hash

**Hash Node** generates a hash table from the records in the input recordset. Hash is used by **Hash Join**.

General IO & Buffers Output Workers Misc

Timing: 0.008ms | 0%

Rows: 1 (Planned: 2) | ↑ over estimated by 2 ×

### #5 Seq Scan

on "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" as t

**Seq Scan Node** finds relevant records by sequentially scanning the input record set. When reading from a table, Seq Scans (unlike Index Scans) perform a single read operation (only the table is read).

General IO & Buffers Output Workers Misc

Relation Name "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"

Alias t

Plan Width 4 Bytes

Actual Startup Time 0.01ms

Actual Total Time 0.011ms

Filter (("НАИМЕНОВАНИЕ")::text > 'Перезачет'::text)

\* Calculated value

## Запрос №2

### Индексы, добавление которых уменьшит время выполнения запроса

На таблице Н\_ЛЮДИ

- Индекс на атрибуте ИМЯ ускорит фильтрацию строк с условием Н\_ЛЮДИ.ИМЯ = Роман

На таблице Н\_ОБУЧЕНИЯ

- Индекс на атрибуте ЧЛВК\_ИД ускорит фильтрацию строк с условием Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 163276 и соединение таблиц по атрибуту ЧЛВК\_ИД

### Возможные планы выполнения запроса без индексов

План 1:

- Полный скан таблицы Н\_ЛЮДИ с применением фильтра Н\_ЛЮДИ.ИМЯ = Роман
- Полный скан таблицы Н\_ОБУЧЕНИЯ с применением фильтра Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 163276
- Соединение таблиц Н\_ЛЮДИ и Н\_ОБУЧЕНИЯ с использованием Hash Join по атрибуту ЧЛВК\_ИД
- Полный скан таблицы Н\_УЧЕНИКИ
- Соединение результатов предыдущего этапа с таблицей Н\_УЧЕНИКИ с использованием Nested Loop Join по атрибуту ЧЛВК\_ИД

План 2:

- Полный скан таблицы Н\_ЛЮДИ с применением фильтра Н\_ЛЮДИ.ИМЯ = Роман
- Полный скан таблицы Н\_ОБУЧЕНИЯ с применением фильтра Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 163276
- Соединение таблиц Н\_ЛЮДИ и Н\_ОБУЧЕНИЯ с использованием Nested Loop Join по атрибуту ЧЛВК\_ИД
- Полный скан таблицы Н\_УЧЕНИКИ
- Соединение результатов предыдущего этапа с таблицей Н\_УЧЕНИКИ с использованием Nested Loop Join по атрибуту ЧЛВК\_ИД

Оптимальным является план 1, потому что использование Hash Join значительно ускоряет выполнение соединений в сравнении с Nested Loop Join.

При добавлении индексов вместо полного скана таблиц будет использоваться индексный скан и Nested Loop Join станет быстрее благодаря индексам на атрибутах ЧЛВК\_ИД.

## План выполнения

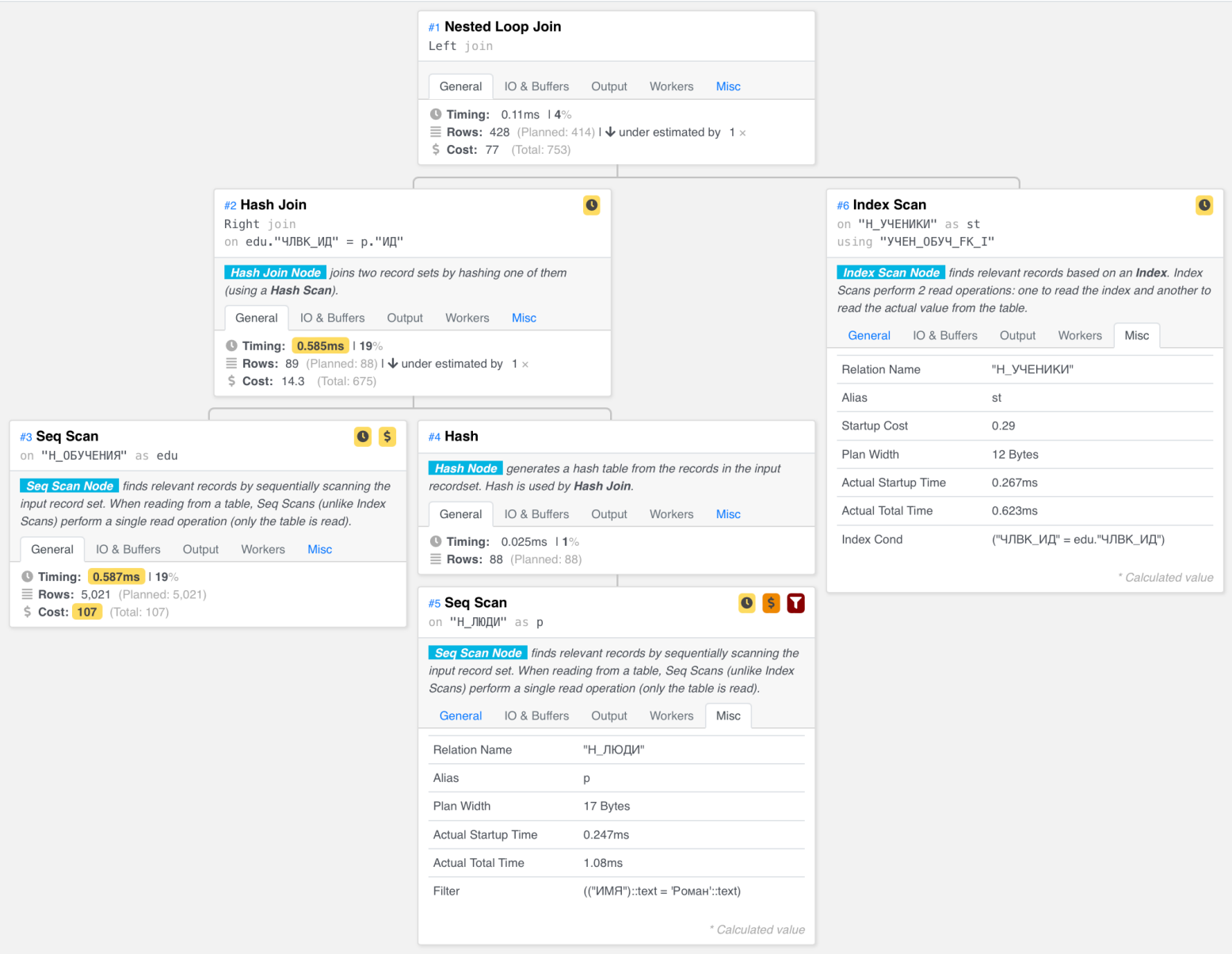
Уберем из фильтров условие Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 163276, иначе нет подходящих записей

```
Nested Loop Left Join (cost=555.36..753.33 rows=414 width=25) (actual time=1.168..3.010 rows=428 loops=1)
-> Hash Right Join (cost=555.08..675.48 rows=88 width=17) (actual time=1.133..2.277 rows=89 loops=1)
    Hash Cond: (edu."ЧЛВК_ИД" = p."ИД")
    -> Seq Scan on "Н_ОБУЧЕНИЯ" edu (cost=0.00..107.21 rows=5'021 width=4) (actual time=0.007..0.587 rows=5'021 loops=1)
    -> Hash (cost=553.98..553.98 rows=88 width=17) (actual time=1.105..1.105 rows=88 loops=1)
        Buckets: 1'024 Batches: 1 Memory Usage: 13kB
        -> Seq Scan on "Н_ЛЮДИ" p (cost=0.00..553.98 rows=88 width=17) (actual time=0.247..1.080 rows=88 loops=1)
            Filter: (("ИМЯ")::text = 'Роман')::text)
            Rows Removed by Filter: 5'030
    -> Index Scan using "УЧЕН_ОБУЧ_ФК_I" on "Н_УЧЕНИКИ" st (cost=0.29..0.83 rows=5 width=12) (actual time=0.003..0.007 rows=5 loops=89)
```

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

Planning Time: 1.353 ms

Execution Time: 3.084 ms



# Вывод

В ходе выполнения лабораторной работы я освоил работу с реляционной алгеброй и научился строить планы выполнения запросов, а также их диаграммы. Я изучил различные виды индексов и узнал, как использовать их для оптимизации запросов.