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

Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа по дисциплине Базы данных №4**

Вариант 863177

Выполнил: Снагин Станислав Максимович

Проверил: Райла Мартин

Группа: Р3115

Санкт Петербург 2025 г.

Содержание

[Задание 3](#__RefHeading___Toc152_2376174330)

[Ход работы 4](#__RefHeading___Toc2108_4178638284)

[Разбор терминов 4](#__RefHeading___Toc156_2376174330)

[Запросы SQL 4](#__RefHeading___Toc2122_205132324)

[1 Задание 4](#__RefHeading___Toc2124_205132324)

[Предлагаемые индексы 4](#__RefHeading___Toc2126_205132324)

[Планы выполнения запросов без индексов 5](#__RefHeading___Toc2128_205132324)

[Первый 5](#__RefHeading___Toc2130_205132324)

[Второй 5](#__RefHeading___Toc2132_205132324)

[Explain analyse 6](#__RefHeading___Toc2134_205132324)

[2 Задание 7](#__RefHeading___Toc2136_205132324)

[Предлагаемые индексы 7](#__RefHeading___Toc2138_205132324)

[Планы выполнения запросов без индексов 7](#__RefHeading___Toc2140_205132324)

[Первый 7](#__RefHeading___Toc2142_205132324)

[Второй 8](#__RefHeading___Toc2163_205132324)

[Explain analyse 9](#__RefHeading___Toc2144_205132324)

[Вывод 10](#__RefHeading___Toc2146_205132324)

# Задание

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

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:  
   Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.  
   Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД.  
   Фильтры (AND):   
   a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ > Экзаменационный лист.  
   b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 105590.  
   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)  
   a) Н\_ЛЮДИ.ФАМИЛИЯ > Ёлкин.  
   b) Н\_ОБУЧЕНИЯ.НЗК < 999080.  
   c) Н\_УЧЕНИКИ.НАЧАЛО < 2008-09-01.  
   Вид соединения: INNER JOIN.

# Ход работы

## Разбор терминов

**Индексы** — спискок всех значений в группе из одного или нескольких столбцов, упорядоченный в некотором приемлемом для данного типа данных смысле. Каждое значение имеет указатель на строку в таблице, где это значение встречается. (или по-простому это объекты базы данных, основная цель которого состоит в повышении производительности запросов).

План — пошаговое описание выполнения SQL запросов базой данных. Показывает, как будут извлекаться данные, в какой последовательности будут обрабатываться вложенные запросы и JOIN’ы, какие сортировки, фильтры, агрегации, операции будут применены.

## Запросы SQL

### 1 Задание

SELECT Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД  
FROM Н\_ТИПЫ\_ВЕДОМОСТЕЙ  
INNER JOIN Н\_ВЕДОМОСТИ  
 ON Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД = Н\_ВЕДОМОСТИ.ТВ\_ИД  
WHERE  
 Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Перезачет'  
 AND Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 105590;

#### 

#### Предлагаемые индексы

B-Tree Н\_ВЕДОМОСТИ.ЧЛВК\_ИД — ускорение фильтрации строк по условию Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 105590.

B-Tree Н\_ВЕДОМОСТИ.ТВ\_ИД — ускорение JOIN запросов

B-Tree Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ — если есть много повторяющихся строк, можно ускорить фильтрацию строк Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Перезачет'.

#### Планы выполнения запросов без индексов

##### Первый

Так как отсутствуют индексы, системы вынуждена делать полный скан таблицы Н\_ТИПЫ\_ВЕДОМОСТЕЙ.

В процессе построчного чтения проверяется условие НАИМЕНОВАНИЕ = «Перезачет». После полного сканирования отбираются только нужные строки (перезачет).

Далее происходит такое же полное построчное сканирование Н\_ВЕДОМОСТИ, условие ЧЛВК\_ИД > 105590 и проверяется равенство ТВ\_ИД с каждым ИД из Н\_ТИПЫ\_ВЕДОМОСТЕЙ. Отбираются только нужные строки.

Далее используется Nested Loops JOIN по аттрибуту ТВ\_ИД. Для каждой строки из Н\_ТИПЫ\_ВЕДОМОСТЕЙ перебираются все строки из Н\_ВЕДОМОСТИ, чтобы найти совпадение ТВ\_ИД = ИД.

##### Второй

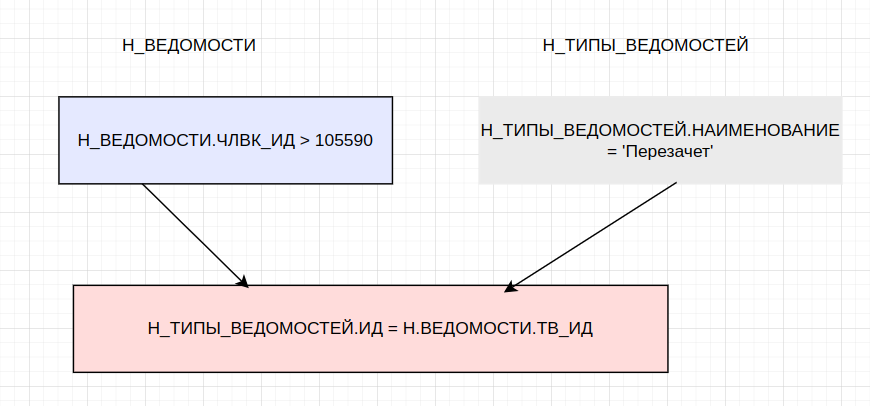
Полный скан таблицы Н\_ТИПЫ\_ВЕДОМОСТЕЙ с применением фильтра НАИМЕНОВАНИЕ = «Перезачет»

Далее происходит такое же полное построчное сканирование Н\_ВЕДОМОСТИ, условие ЧЛВК\_ИД > 105590 и проверяется равенство ТВ\_ИД с каждым ИД из Н\_ТИПЫ\_ВЕДОМОСТЕЙ. Отбираются только нужные строки.

Далее используется Nested Loops JOIN по аттрибуту ТВ\_ИД, так как нет индекса на нем. Для каждой строки из Н\_ТИПЫ\_ВЕДОМОСТЕЙ перебираются все строки из Н\_ВЕДОМОСТИ, чтобы найти совпадение ТВ\_ИД = ИД.

Во 2 плане фильтр применяется сразу при сканировании, в отличие от первого. Получается более эффективно, так как уменьшается кол-во строк, с которыми надо работать дальше.

#### Explain analyse

                                                             QUERY PLAN                                              
                    
--------------------------------------------------------------------------------------------------------------------  
------------------   
Nested Loop  (cost=830.93..6789.42 rows=74125 width=8) (actual time=0.517..5.136 rows=10075 loops=1)   
  ->  Seq Scan on "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"  (cost=0.00..1.04 rows=1 width=4) (actual time=0.014..0.015 rows=1 loops=1)   
        Filter: (("НАИМЕНОВАНИЕ")::text = 'Перезачет'::text)   
        Rows Removed by Filter: 2   
  ->  Bitmap Heap Scan on "Н\_ВЕДОМОСТИ"  (cost=830.93..6047.13 rows=74125 width=8) (actual time=0.497..3.957 rows=1  
0075 loops=1)   
        Recheck Cond: ("Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "ТВ\_ИД")   
        Filter: ("ЧЛВК\_ИД" > 105590)   
        Heap Blocks: exact=758   
        ->  Bitmap Index Scan on "ВЕД\_ТВ\_FK\_I"  (cost=0.00..812.40 rows=74147 width=0) (actual time=0.408..0.408 ro  
ws=10075 loops=1)   
              Index Cond: ("ТВ\_ИД" = "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД")   
Planning Time: 0.683 ms   
Execution Time: 5.629 ms   
(12 строк)  


### 2 Задание

SELECT Н\_ЛЮДИ.ИМЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО  
FROM Н\_ОБУЧЕНИЯ  
INNER JOIN Н\_УЧЕНИКИ ON Н\_УЧЕНИКИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД  
INNER JOIN Н\_ЛЮДИ ON Н\_ЛЮДИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД  
WHERE Н\_ЛЮДИ.ФАМИЛИЯ > 'Ёлкин'  
AND (Н\_ОБУЧЕНИЯ.НЗК)::integer < 999080  
AND Н\_УЧЕНИКИ.НАЧАЛО < '2008-09-01'::date;

#### 

#### Предлагаемые индексы

B-Tree Н\_ЛЮДИ.ФАМИЛИЯ — ускорение фильтрации по Н\_ЛЮДИ.ФАМИЛИЯ > «Ёлкин»

B-Tree Н\_УЧЕНИКИ.НАЧАЛО — ускорение фильтрации по НАЧАЛО < «2008-09-01»

B-Tree Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД — ускорение соединения таблиц по этому атрибуту

#### Планы выполнения запросов без индексов

##### Первый

Система делает полный скан таблицы Н\_ЛЮДИ с применением фильтра ФАМИЛИЯ > «Ёлкин»

Система делает полный скан таблицы Н\_УЧЕНИКИ с применением фильтра НАЧАЛО < «2008-09-01»

Соедниение Н\_ЛЮДИ и Н\_УЧЕНИКИ через Hash Join (быстрее Nested’а, так как использует готовую хэш-таблицу)

Система делает полный скан таблицы Н\_ОБУЧЕНИЯ с применением фильтра НЗК < 990800

Соединение результата с Н\_ОБУЧЕНИЯ через Hash Join Н\_УЧЕНИКИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД

##### Второй

Система делает полный скан таблицы Н\_ЛЮДИ с применением фильтра ФАМИЛИЯ > «Ёлкин»

Система делает полный скан таблицы Н\_УЧЕНИКИ с применением фильтра НАЧАЛО < «2008-09-01»

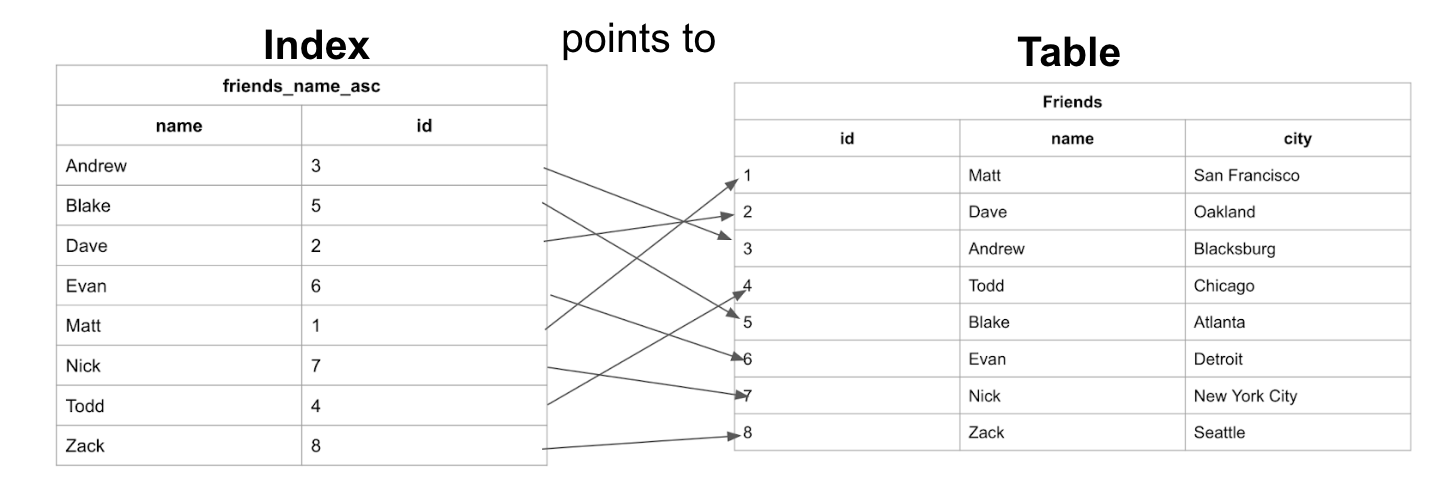
Соедниение Н\_ЛЮДИ и Н\_УЧЕНИКИ через Nested Loops Join

Система делает полный скан таблицы Н\_ОБУЧЕНИЯ с применением фильтра НЗК < 990800

Соединение результата с Н\_ОБУЧЕНИЯ через Nested Loops Join Н\_УЧЕНИКИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД

Способ с Hash JOIN эффективен при работе с большими данными и заранее проставленными индексами (тогда мы добавляем написанные выше индексы); Nested Loops JOIN предпочтительнее в ситуациях, когда в результате будет мало строк.

Планы изменятся, так как вместо полного сканирования будет сканироваться только заранее подготовленные индексы, что позволит сделать работу Nested Loop JOIN более эффективной.



#### Explain analyse

#### QUERY PLAN                                                            -------------------------------------------------------------------------------------------------------------------- --------- Hash Join  (cost=376.87..1303.40 rows=193 width=25) (actual time=6.714..11.471 rows=214 loops=1)   Hash Cond: ("Н\_УЧЕНИКИ"."ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД")   ->  Hash Join  (cost=211.07..1122.21 rows=2693 width=29) (actual time=4.963..9.658 rows=249 loops=1)         Hash Cond: ("Н\_УЧЕНИКИ"."ИД" = "Н\_ЛЮДИ"."ИД")         ->  Seq Scan on "Н\_УЧЕНИКИ"  (cost=0.00..867.39 rows=16661 width=12) (actual time=0.007..3.914 rows=16666 l oops=1)               Filter: ("НАЧАЛО" < '2008-09-01'::date)               Rows Removed by Filter: 6645         ->  Hash  (cost=163.97..163.97 rows=3768 width=17) (actual time=3.972..3.973 rows=3777 loops=1)               Buckets: 4096  Batches: 1  Memory Usage: 221kB               ->  Seq Scan on "Н\_ЛЮДИ"  (cost=0.00..163.97 rows=3768 width=17) (actual time=0.011..3.184 rows=3777 loops=1)                     Filter: (("ФАМИЛИЯ")::text > 'Ёлкин'::text)                     Rows Removed by Filter: 1341   ->  Hash  (cost=144.87..144.87 rows=1674 width=4) (actual time=1.732..1.732 rows=4382 loops=1)         Buckets: 8192 (originally 2048)  Batches: 1 (originally 1)  Memory Usage: 219kB         ->  Seq Scan on "Н\_ОБУЧЕНИЯ"  (cost=0.00..144.87 rows=1674 width=4) (actual time=0.013..1.023 rows=4382 loo ps=1)               Filter: (("НЗК")::integer < 999080)               Rows Removed by Filter: 639 Planning Time: 1.991 ms Execution Time: 11.578 ms (19 строк)

# Вывод

В ходе лабораторной работы я научился базовым приемам оптимизации SQL запросов с использованием индексов, узнал про Nested Loops JOIN и Hash JOIN, Explain Analyse. На 2 примерах показал, как можно оптимизировать эти запросы.