

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

Факультет программной инженерии и компьютерной техники

Базы данных

## Лабораторная работа №4

Вариант 71

Выполнил: Свечников Константин Денисович, Р3130

Проверил: Перцев Т. С., преподаватель практики

## Оглавление

Задание.....	2
Ход работы .....	3
Запросы .....	3
Добавление индексов .....	3
Планы выполнения запросов.....	4
EXPLAIN ANALYZE .....	7
Заключение.....	8

## 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.

Изменяться ли планы при добавлении индекса и как?

Для запросов 1-2 необходимо добавить в отчет вывод команды EXPLAIN ANALYZE [запрос]

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.ИД < 2.  
б) Н\_ВЕДОМОСТИ.ИД < 39921.  
с) Н\_ВЕДОМОСТИ.ИД > 1250981.  
Вид соединения: RIGHT 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.ФАМИЛИЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.ГРУППА.

Фильтры: (AND)

а) Н\_ЛЮДИ.ИМЯ = Владимир.

б) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД < 113409.

Вид соединения: LEFT JOIN.

## Ход работы

### Запросы

1.

```
select "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД", "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
from "Н_ВЕДОМОСТИ"
      right join "Н_ТИПЫ_ВЕДОМОСТЕЙ" on "Н_ВЕДОМОСТИ"."ВЕД_ИД" =
"Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД"
where "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД"<2 and "Н_ВЕДОМОСТИ"."ИД"<39921 and
"Н_ВЕДОМОСТИ"."ИД"> 1250981
```

2.

```
select "Н_ЛЮДИ"."ФАМИЛИЯ", "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД", "Н_УЧЕНИКИ"."ГРУППА"
from "Н_ЛЮДИ"
left join "Н_ОБУЧЕНИЯ" on "Н_ЛЮДИ"."ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД"
left join "Н_УЧЕНИКИ" on "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД"
where "Н_ЛЮДИ"."ИМЯ" = 'Владимир' and "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" < 113409
```

### Добавление индексов

Для первого запроса можно добавить индексы на следующие атрибуты и таблицы:

- Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД – индекс типа B-Tree
- Н\_ВЕДОМОСТИ.ИД – индекс типа B-Tree

1. Индекс на атрибут «ИД» в таблице «Н\_ТИПЫ\_ВЕДОМОСТЕЙ» позволит ускорить выполнение операций WHERE и JOIN. Без индекса, каждой из этих операций каждый раз придется сканировать таблицу «Н\_ТИПЫ\_ВЕДОМОСТЕЙ»
2. Индекс на атрибут «ИД» в таблице «Н\_ВЕДОМОСТИ» позволит ускорить выполнение операции WHERE, где при его отсутствии каждый раз придется заново сканировать всю таблицу для выполнения условий WHERE.

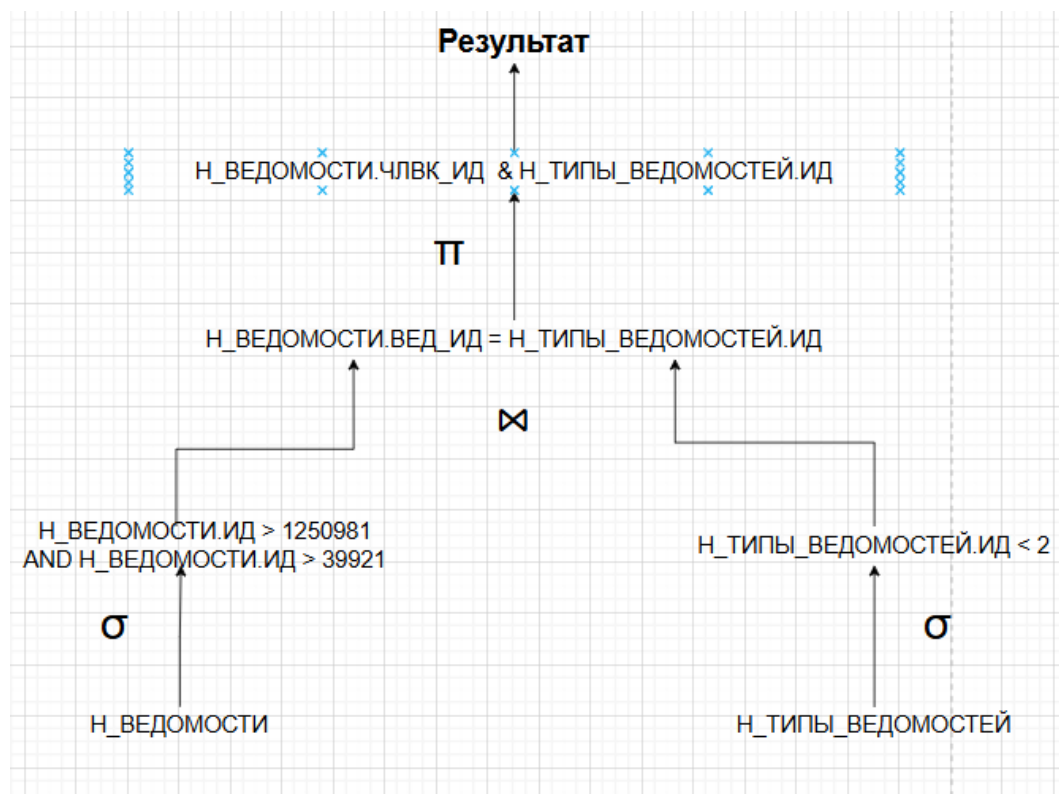
Для второго запроса можно добавить индексы на следующие атрибуты и таблицы:

- Н\_ЛЮДИ.ИМЯ – индекс типа B-Tree
  - Н\_ЛЮДИ.ИД – индекс типа B-Tree
  - Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД – индекс типа B-Tree
  - Н\_УЧЕНИКИ.ЧЛВК\_ИД – индекс типа B-Tree
1. Индекс на атрибут «ИМЯ» в таблице «Н\_ЛЮДИ» позволит ускорить выполнение операции WHERE, где при его отсутствии каждый раз придется заново сканировать всю таблицу для выполнения условий WHERE.
  2. Индекс на атрибут «ИД» в таблице «Н\_ЛЮДИ» позволит ускорить выполнение операции JOIN. Без индекса, каждой из этих операций каждый раз придется сканировать таблицу «Н\_ЛЮДИ»
  3. Индекс на атрибут «ЧЛВК\_ИД» в таблице «Н\_ОБУЧЕНИЯ» позволит ускорить выполнение операций WHERE и JOIN. Без индекса, каждой из этих операций каждый раз придется сканировать таблицу «Н\_ОБУЧЕНИЯ»
  4. Индекс на атрибут «ЧЛВК\_ИД» в таблице «Н\_УЧЕНИКИ» позволит ускорить выполнение операции JOIN. Без индекса, каждой из этих операций каждый раз придется сканировать таблицу «Н\_УЧЕНИКИ»

## Планы выполнения запросов

Планы выполнения запросов без индексов для первого запроса:

1. Производится полный скан таблицы «Н\_ТИПЫ\_ВЕДОМОСТЕЙ» с применением фильтра «ИД» < 2, затем производится полный скан таблицы «Н\_ВЕДОМОСТИ» с фильтром «ИД» < 39921 AND «ИД» > 1250981, затем результаты объединяются через RIGHT JOIN



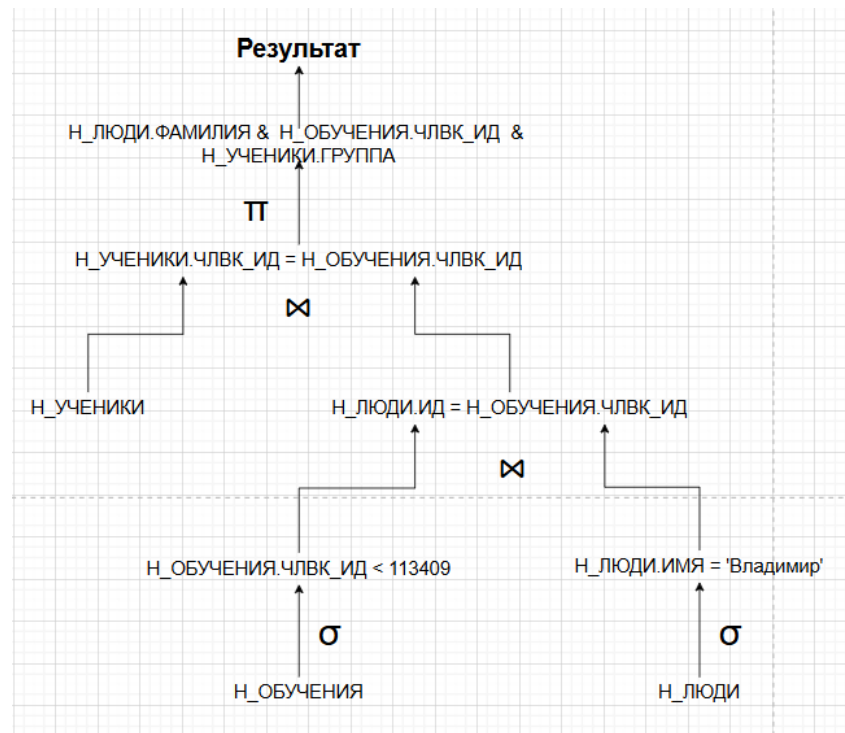
2. Производится полный скан таблицы «Н\_ВЕДОМОСТИ» с фильтром «ИД» < 39921 AND «ИД» > 1250981, затем производится полный скан таблицы «Н\_ТИПЫ\_ВЕДОМОСТЕЙ» с применением фильтра «ИД» < 2, потом результаты объединяются через RIGHT JOIN.

Оптимальным планом выполнения данного запроса является первый вариант, так как фильтр по ИД применяется к таблице «Н\_ТИПЫ\_ВЕДОМОСТЕЙ» до выполнения объединения, сокращая количество строк перед соединением

При добавлении индексов на столбцы «Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД» и «Н\_ВЕДОМОСТИ.ИД» первый план выполнения запроса станет оптимальнее, так как будет использован индексный поиск вместо полного сканирования таблицы.

Планы выполнения запросов без индексов для второго запроса:

1. Производится полный скан таблицы «Н\_ЛЮДИ» с фильтром «ИМЯ» = 'Владимир', затем производится полный скан таблицы «Н\_ОБУЧЕНИЯ» с фильтром «ЧЛВК\_ИД» < 113409, потом производится полный скан таблицы «Н\_УЧЕНИКИ», и наконец результаты объединяются через LEFT JOIN.



2. Производится полный скан таблицы «Н\_ОБУЧЕНИЯ» с фильтром «ЧЛВК\_ИД» < 113409, затем производится полный скан таблицы «Н\_ЛЮДИ» с фильтром «ИМЯ» = 'Владимир', потом производится полный скан таблицы «Н\_УЧЕНИКИ», и наконец результаты объединяются через LEFT JOIN

Оптимальным планом выполнения данного запроса является первый план, так как фильтр «ИМЯ» = 'Владимир' сокращает число строк в таблице «Н\_ЛЮДИ» на раннем этапе.

При добавлении индексов на столбцы «Н\_ЛЮДИ.ИМЯ» и «Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД» план выполнения становится оптимальнее за счет индексного поиска вместо полного сканирования таблиц.

# EXPLAIN ANALYZE

```
explain analyze select "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД", "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
from "Н_ВЕДОМОСТИ"
      right join "Н_ТИПЫ_ВЕДОМОСТЕЙ" on "Н_ВЕДОМОСТИ"."ВЕД_ИД" = "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД"
where "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД" < 2 and "Н_ВЕДОМОСТИ"."ИД" < 39921 and "Н_ВЕДОМОСТИ"."ИД" > 1250981
```

Output Result 5

QUERY PLAN

- 1 Nested Loop (cost=27.82..2542.77 rows=1 width=8) (actual time=0.024..0.025 rows=0 loops=1)
- 2 Join Filter: ("Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "Н\_ВЕДОМОСТИ"."ВЕД\_ИД")
- 3 -> Seq Scan on "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=4) (actual time=0.010..0.011 rows=1 loops=1)
- 4 Filter: ("ИД" < 2)
- 5 Rows Removed by Filter: 2
- 6 -> Bitmap Heap Scan on "Н\_ВЕДОМОСТИ" (cost=27.82..2527.83 rows=1112 width=8) (actual time=0.010..0.010 rows=0 loops=1)
- 7 Recheck Cond: (("ИД" < 39921) AND ("ИД" > 1250981))
- 8 -> Bitmap Index Scan on "ВЕД\_РК" (cost=0.00..27.54 rows=1112 width=0) (actual time=0.009..0.009 rows=0 loops=1)
- 9 Index Cond: (("ИД" < 39921) AND ("ИД" > 1250981))
- 10 Planning Time: 0.237 ms
- 11 Execution Time: 0.055 ms

Из результата можно увидеть, что для выполнения запроса был выбран план выполнения с использованием вложенного цикла (Nested Loop). Он соединяет таблицы «Н\_ТИПЫ\_ВЕДОМОСТЕЙ» и «Н\_ВЕДОМОСТИ», учитывая условие соединения по полю «ВЕД\_ИД». Затем производится полное сканирование таблицы «Н\_ТИПЫ\_ВЕДОМОСТЕЙ» (Seq Scan) с фильтром «ИД» < 2. После этого проводятся последовательные битовые индексные сканирования таблицы «Н\_ВЕДОМОСТИ» и поля «ВЕД\_РК» с учетом фильтра «ИД» < 39921 AND «ИД» > 1250981.

Также видно, что стоимость выполнения плана составила 27.8, а затраченное время составило 0.055 ms, что довольно быстро.

Из описания индексных сканов можно сделать вывод, что для оптимизации запроса были использованы индексы, что позволило ускорить выполнение запроса.

```
query1 console query2 x
1 explain analyse select "Н_ЛЮДИ"."ФАМИЛИЯ", "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД", "Н_УЧЕНИКИ"."ГРУППА"
2 from "Н_ЛЮДИ"
3 left join "Н_ОБУЧЕНИЯ" on "Н_ЛЮДИ"."ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД"
4 left join "Н_УЧЕНИКИ" on "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД"
5 where "Н_ЛЮДИ"."ИМЯ" = 'Владимир' and "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" < 113409
```

Output Result 1

QUERY PLAN

- 1 Nested Loop Left Join (cost=165.92..238.08 rows=42 width=24) (actual time=0.896..1.044 rows=27 loops=1)
- 2 -> Hash Join (cost=165.63..182.08 rows=9 width=20) (actual time=0.871..0.948 rows=6 loops=1)
- 3 Hash Cond: ("Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД")
- 4 -> Index Only Scan using "ОБУЧ\_ЧЛВК\_ФК\_I" on "Н\_ОБУЧЕНИЯ" (cost=0.28..15.63 rows=420 width=4) (actual time=0.013..0.070 rows=412 loops=1)
- 5 Index Cond: ("ЧЛВК\_ИД" < 113409)
- 6 Heap Fetches: 0
- 7 -> Hash (cost=163.97..163.97 rows=110 width=20) (actual time=0.812..0.813 rows=110 loops=1)
- 8 Buckets: 1024 Batches: 1 Memory Usage: 14kB
- 9 -> Seq Scan on "Н\_ЛЮДИ" (cost=0.00..163.97 rows=110 width=20) (actual time=0.022..0.788 rows=110 loops=1)
- 10 Filter: (("ИМЯ")::text = 'Владимир')::text)
- 11 Rows Removed by Filter: 5008
- 12 -> Index Scan using "УЧЕН\_ОБУЧ\_ФК\_I" on "Н\_УЧЕНИКИ" (cost=0.29..6.17 rows=5 width=8) (actual time=0.008..0.014 rows=4 loops=6)
- 13 Index Cond: ("ЧЛВК\_ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД")
- 14 Planning Time: 1.839 ms
- 15 Execution Time: 1.125 ms

Из результата можно видеть, что для выполнения запроса был использован вложенный цикл, который сперва соединил по хэшу таблицы «Н\_ОБУЧЕНИЯ» и «Н\_ЛЮДИ», затем провел сканирование только по индексу с фильтром «ЧЛВК\_ИД» < 113409. Затем было сделано хэширование результатов сканирования, после чего провел полное сканирование таблицы «Н\_ЛЮДИ» с фильтром «ИМЯ» = 'Владимир'. В конце он провел сканирование по индексу для «Н\_УЧЕНИКИ» выполняя условие «ЧЛВК\_ИД» = «Н\_ОБУЧЕНИЯ».«ЧЛВК\_ИД».

Стоимость выполнения составила 165.9, а время выполнения – 1.125 ms.

## **Заключение**

В ходе выполнения данной лабораторной работы я изучил различные виды индексов и узнал, как использовать их для оптимизации скорости выполнения запросов.