Федеральное государственное автономное образовательное учреждение высшего образования  
«Национальный исследовательский университет ИТМО»  
  
Факультет программной инженерии и компьютерной техники  
  
Направление подготовки 09.03.04 «Программная инженерия» – Системное и прикладное программное обеспечение

**Отчёт  
По лабораторной работе №4**

По дисциплине «Базы данных»  
**Вариант: 648644**

Выполнил:  
Студент 1 курса  
Васильев Артём Евгеньевич

Группа: 3119

Преподаватель:

Райла Мартин

2024 г.

# Текст задания:

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\_ЛЮДИ, Н\_ВЕДОМОСТИ.  
   Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД.  
   Фильтры (AND):  
   a) Н\_ЛЮДИ.ИМЯ = Николай.  
   b) Н\_ВЕДОМОСТИ.ИД < 1457443.  
   c) 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.ИМЯ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД, Н\_СЕССИЯ.ИД.  
   Фильтры (AND):  
   a) Н\_ЛЮДИ.ФАМИЛИЯ < Ёлкин.  
   b) Н\_ВЕДОМОСТИ.ИД > 1457443.  
   Вид соединения: INNER JOIN.

# Основные этапы работы

***Первый запрос***

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

Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ.

Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД.

Фильтры (AND):

a) Н\_ЛЮДИ.ИМЯ = Николай.

b) Н\_ВЕДОМОСТИ.ИД < 1457443.

c) Н\_ВЕДОМОСТИ.ИД < 1490007.

Вид соединения: LEFT JOIN.

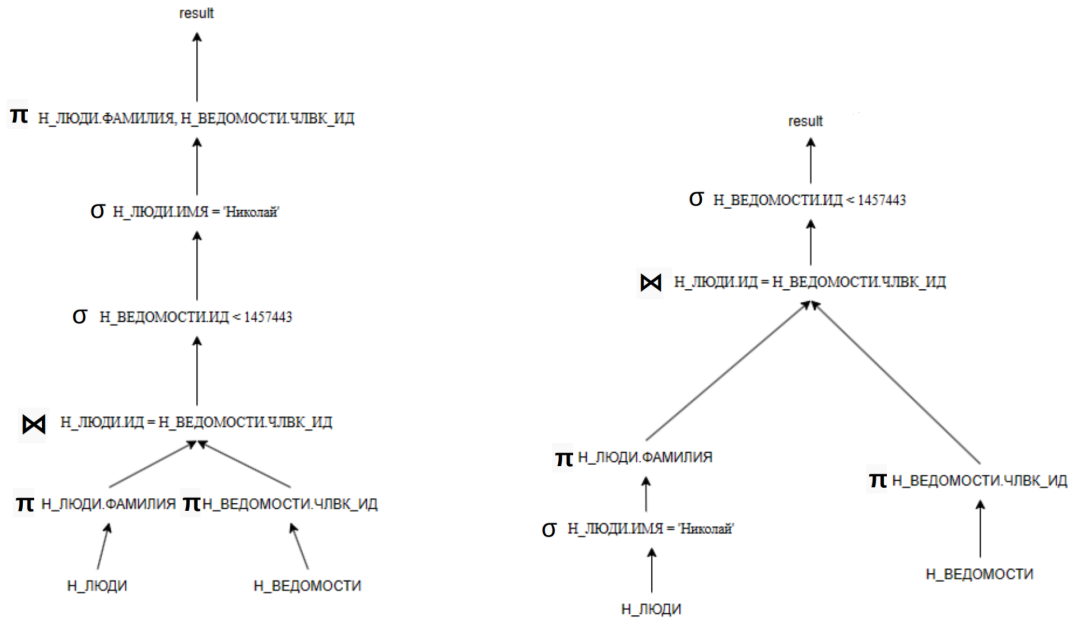
***Реализация***

SELECT Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД FROM Н\_ЛЮДИ

LEFT JOIN Н\_ВЕДОМОСТИ ON Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

WHERE Н\_ЛЮДИ.ИМЯ = 'Николай' AND Н\_ВЕДОМОСТИ.ИД < 1457443;

***Планы***



Оптимальным планом является второй, т. к. происходит объединение только необходимой выборки, вместо полного объединения таблиц.

При добавлении индексов планы выполнения запросов изменятся:

Вместо полного скана таблиц будет использоваться индексный план.

Nested Loops Join станет быстрее благодаря индексу на атрибуте ЧЛВК\_ИД.

***Индексы***

CREATE INDEX ИН\_ЛЮДИ\_ИМЯ ON Н\_ЛЮДИ USING HASH(ИМЯ);

Индекс ускорит фильтрацию с условием Н\_ЛЮДИ.ИМЯ = Николай

CREATE INDEX ИН\_ВЕДОМОСТИ\_ЧЛВК ON Н\_ВЕДОМОСТИ USING HASH(ЧЛВК\_ИД);

Индекс ускорит объединение таблиц

CREATE INDEX ИН\_ВЕДОМОСТИ\_ИД ON Н\_ВЕДОМОСТИ USING BTREE(ИД); Индекс ускорит фильтрацию с условием Н\_ВЕДОМОСТИ.ИД < 1457443

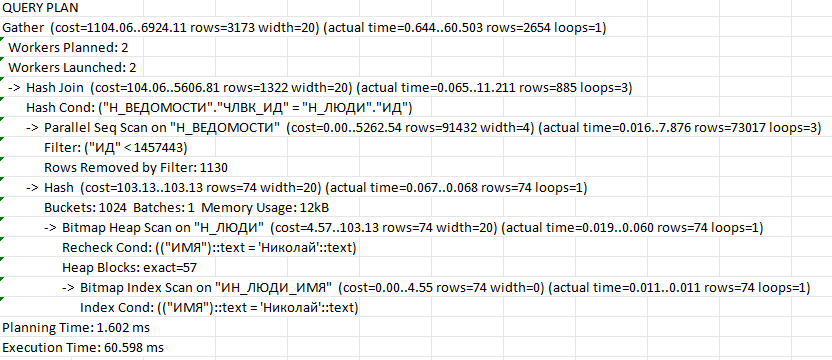
***Explain analyze***

EXPLAIN ANALYZE

SELECT Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД FROM Н\_ЛЮДИ

LEFT JOIN Н\_ВЕДОМОСТИ ON Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

WHERE Н\_ЛЮДИ.ИМЯ = 'Николай' AND Н\_ВЕДОМОСТИ.ИД < 1457443;



***Второй запрос***

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.ИМЯ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД, Н\_СЕССИЯ.ИД.

Фильтры (AND):

a) Н\_ЛЮДИ.ФАМИЛИЯ < Ёлкин.

b) Н\_ВЕДОМОСТИ.ИД > 1457443.

Вид соединения: INNER JOIN.

***Реализация***

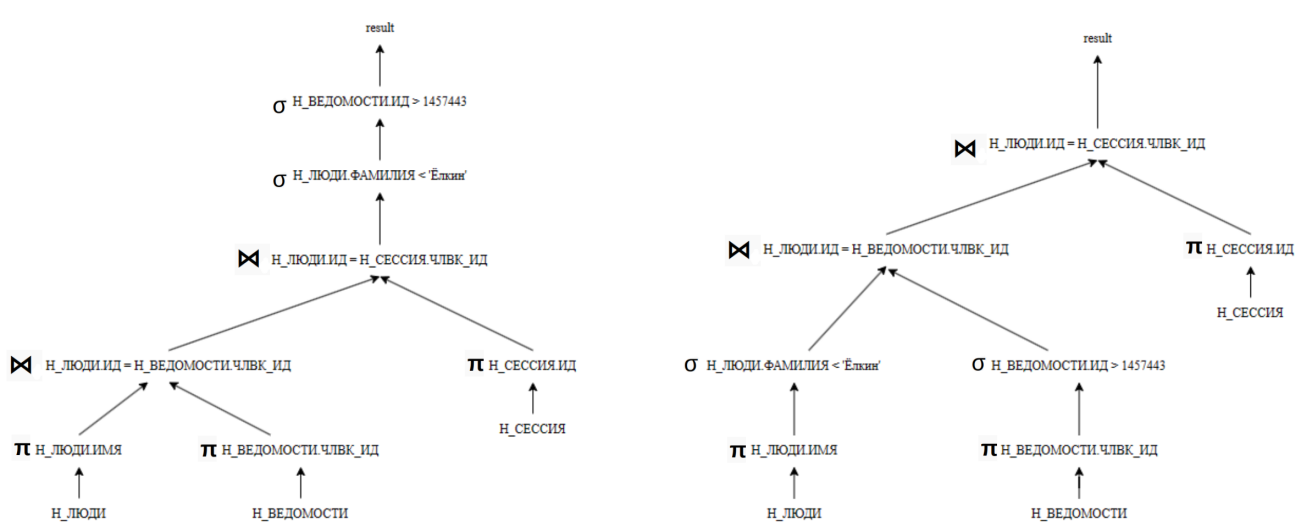
SELECT Н\_ЛЮДИ.ИМЯ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД, Н\_СЕССИЯ.ИД FROM Н\_ЛЮДИ

INNER JOIN Н\_ВЕДОМОСТИ ON Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

INNER JOIN Н\_СЕССИЯ ON Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД

WHERE Н\_ЛЮДИ.ФАМИЛИЯ < 'Ёлкин' AND Н\_ВЕДОМОСТИ.ИД > 1457443;

***Планы***



Оптимальным планом является второй, т. к. происходит объединение только необходимой выборки, вместо полного объединения таблиц.

При добавлении индексов планы выполнения запросов изменятся:

Вместо полного скана таблиц будет использоваться индексный план.

Nested Loops Join станет быстрее благодаря индексу на атрибуте ЧЛВК\_ИД.

***Индексы***

CREATE INDEX ИН\_Н\_ЛЮДИ\_ФАМИЛИЯ ON Н\_ЛЮДИ USING BTREE(ФАМИЛИЯ);

Индекс ускорит фильтрацию с условием Н\_ЛЮДИ.ФАМИЛИЯ < Ёлкин

CREATE INDEX ИН\_Н\_ВЕДОМОСТИ\_ИД ON Н\_ВЕДОМОСТИ USING BTREE(ИД);

Индекс ускорит фильтрацию с условием Н\_ВЕДОМОСТИ.ИД > 1457443

CREATE INDEX ИН\_Н\_ЛЮДИ\_ИД ON Н\_ЛЮДИ USING HASH(ИД);

Индекс ускорит объединение таблиц

***Explain analyze***

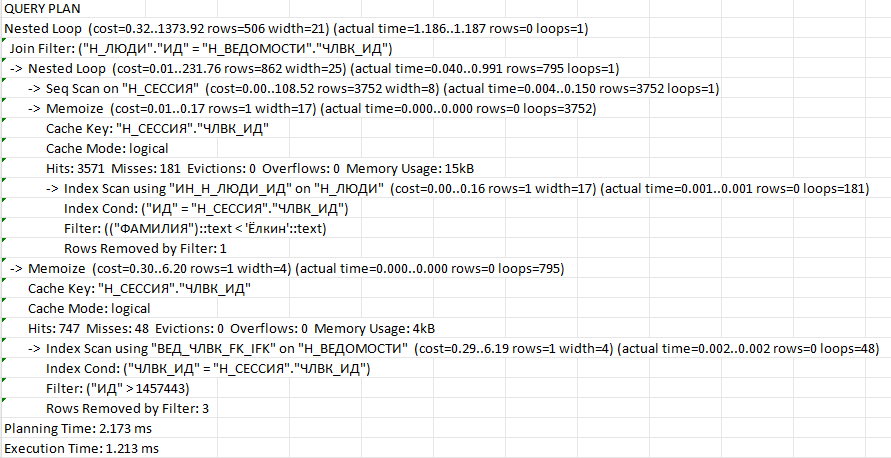
EXPLAIN ANALYZE

SELECT Н\_ЛЮДИ.ИМЯ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД, Н\_СЕССИЯ.ИД FROM Н\_ЛЮДИ

INNER JOIN Н\_ВЕДОМОСТИ ON Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

INNER JOIN Н\_СЕССИЯ ON Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД

WHERE Н\_ЛЮДИ.ФАМИЛИЯ < 'Ёлкин' AND Н\_ВЕДОМОСТИ.ИД > 1457443;



# Заключение

В ходе выполнения данной лабораторной работы я научился оптимизировать SQL запросы, составлять планы запросов, а также закрепил работу с синтаксиcом SQL.