Министерство науки и высшего образования Российской Федерации  
федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа по

Базам данных №4

Вариант 3

Работу выполнила:

Касьяненко В. М.

Группа:

P3120

Преподаватель:

Королева Ю. А.

Санкт-Петербург,

2023

**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.

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.

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

Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ.

Вывести атрибуты: Н\_ЛЮДИ.ИМЯ, Н\_ВЕДОМОСТИ.ИД.

Фильтры (AND):

a) Н\_ЛЮДИ.ФАМИЛИЯ < Иванов.

b) Н\_ВЕДОМОСТИ.ИД = 1250981.

c) Н\_ВЕДОМОСТИ.ИД = 1250972.

Вид соединения: RIGHT JOIN.

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

Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.

Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ДАТА, Н\_СЕССИЯ.ИД.

Фильтры (AND):

a) Н\_ЛЮДИ.ИД < 100012.

b) Н\_ВЕДОМОСТИ.ИД = 1457443.

Вид соединения: RIGHT JOIN.

**Запросы**

SELECT "Н\_ЛЮДИ"."ИМЯ","Н\_ЛЮДИ"."ФАМИЛИЯ", "Н\_ВЕДОМОСТИ"."ИД"

FROM "Н\_ЛЮДИ"

RIGHT JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"

WHERE "Н\_ЛЮДИ"."ФАМИЛИЯ" < 'Иванов'

AND "Н\_ВЕДОМОСТИ"."ИД" IN (1250981, 1250972);

SELECT "Н\_ЛЮДИ"."ОТЧЕСТВО", "Н\_ВЕДОМОСТИ"."ДАТА", "Н\_СЕССИЯ"."ИД"

FROM "Н\_ЛЮДИ"

RIGHT JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"

RIGHT JOIN "Н\_СЕССИЯ" ON "Н\_ВЕДОМОСТИ"."СЭС\_ИД" = "Н\_СЕССИЯ"."СЭС\_ИД"

WHERE "Н\_ЛЮДИ"."ИД" < 100012

AND "Н\_ВЕДОМОСТИ"."ИД" = 1457443

**Добавление индексов**

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

* Н\_ЛЮДИ.ФАМИЛИЯ - индекс типа B-Tree
* Н\_ВЕДОМОСТИ.ИД - индекс типа B-Tree
* Н\_ВЕДОМОСТИ.ЧЛВК\_ИД - индекс типа B-Tree

1. Индекс на атрибут "ФАМИЛИЯ" в таблице "Н\_ЛЮДИ" позволит ускорить выполнение операции WHERE. Без индекса, для выполнения данного условия WHERE придется просканировать все строки таблицы "Н\_ЛЮДИ". С индексом, системе удастся быстро найти нужные строки, что существенно ускорит выполнение запроса.
2. Индекс на атрибут "ИД" в таблице "Н\_ВЕДОМОСТИ" также позволит ускорить выполнение запроса. Без индекса придется просканировать всю таблицу, чтобы найти строки с нужными идентификаторами. С индексом система быстро найдет нужные строки, что ускорит выполнение запроса.
3. Индекс на атрибут "ЧЛВК\_ИД" в таблице "Н\_ВЕДОМОСТИ" поможет оптимизировать операцию объединения таблиц (JOIN). Без индекса системе придется выполнить вложенный цикл, просматривая каждую строку таблицы "Н\_ВЕДОМОСТИ" и для каждой строки искать соответствующую строку в таблице "Н\_ЛЮДИ". С индексом по атрибуту "ЧЛВК\_ИД" система сможет быстро найти нужные строки в таблице "Н\_ЛЮДИ", что существенно ускорит выполнение запроса.

Для второго запроса можно добавить B-Tree индексы на следующие атрибуты:

1. "Н\_ЛЮДИ"."ИД" - для таблицы "Н\_ЛЮДИ"
2. "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" - для таблицы "Н\_ВЕДОМОСТИ"
3. "Н\_ВЕДОМОСТИ"."СЭС\_ИД" - для таблицы "Н\_ВЕДОМОСТИ"

* Первый индекс на "Н\_ЛЮДИ"."ИД" позволит ускорить выборку строк из таблицы "Н\_ЛЮДИ", соответствующих условию "Н\_ЛЮДИ"."ИД" < 100012.
* Второй индекс на "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" поможет оптимизировать выполнение JOIN между таблицами "Н\_ЛЮДИ" и "Н\_ВЕДОМОСТИ", поскольку этот индекс будет использоваться для объединения строк в этих таблицах.
* Третий индекс на "Н\_ВЕДОМОСТИ"."СЭС\_ИД" ускорит выполнение JOIN между таблицами "Н\_ВЕДОМОСТИ" и "Н\_СЕССИЯ", так как этот индекс будет использоваться для объединения строк в этих таблицах.

**Планы выполнения запросов**

Планы выполнения запросов без индексов для первого запроса:

1. Производится полный скан таблицы "Н\_ЛЮДИ" с применением фильтра "ФАМИЛИЯ" < ‘Иванов’, затем производится полный скан таблицы "Н\_ВЕДОМОСТИ" с фильтром "ИД" IN (1250981, 1250972), затем производится объединение результатов с помощью операции RIGHT JOIN.
2. Производится полный скан таблицы "Н\_ВЕДОМОСТИ" с применением фильтра "ИД" IN (1250981, 1250972), затем производится полный скан таблицы "Н\_ЛЮДИ" с фильтром "ФАМИЛИЯ" < ‘Иванов’, затем производится объединение результатов с помощью операции RIGHT JOIN.

Оптимальным планом выполнения данного запроса является первый вариант, так как фильтр по фамилии применяется к таблице "Н\_ЛЮДИ" до выполнения объединения, что позволяет уменьшить количество строк, обрабатываемых в запросе.

При добавлении индекса на столбец "ФАМИЛИЯ" таблицы "Н\_ЛЮДИ" план выполнения запроса изменится следующим образом:

1. Производится индексный поиск в таблице "Н\_ЛЮДИ" по фильтру "ФАМИЛИЯ" < 'Иванов', затем производится полный скан таблицы "Н\_ВЕДОМОСТИ" с фильтром "ИД" IN (1250981, 1250972), затем производится объединение результатов с помощью операции RIGHT JOIN.
2. Производится полный скан таблицы "Н\_ВЕДОМОСТИ" с применением фильтра "ИД" IN (1250981, 1250972), затем производится индексный поиск в таблице "Н\_ЛЮДИ" по фильтру "ФАМИЛИЯ" < 'Иванов', затем производится объединение результатов с помощью операции RIGHT JOIN.

При добавлении индекса на столбец "ФАМИЛИЯ" таблицы "Н\_ЛЮДИ" план выполнения запроса станет оптимальным, так как будет использован индексный поиск вместо полного скана таблицы "Н\_ЛЮДИ", что уменьшит время выполнения запроса.

Для второго запроса можно составить несколько возможных планов выполнения:

1. План с использованием полного сканирования таблиц. В этом случае для выполнения запроса будут просмотрены все строки таблиц "Н\_ЛЮДИ", "Н\_ВЕДОМОСТИ" и "Н\_СЕССИЯ", а затем произойдет сортировка и объединение результатов.
2. План с использованием индекса только на таблице "Н\_ВЕДОМОСТИ". В этом случае будут просмотрены только те строки таблицы "Н\_ВЕДОМОСТИ", которые удовлетворяют условию "Н\_ВЕДОМОСТИ"."ИД" = 1457443, затем произойдет объединение с таблицами "Н\_ЛЮДИ" и "Н\_СЕССИЯ" при помощи соединений по полям "Н\_ЛЮДИ"."ИД" и "Н\_ВЕДОМОСТИ"."СЭС\_ИД" соответственно.
3. План с использованием индексов на таблицах "Н\_ЛЮДИ" и "Н\_ВЕДОМОСТИ". В этом случае будут просмотрены только те строки таблиц "Н\_ЛЮДИ" и "Н\_ВЕДОМОСТИ", которые удовлетворяют условиям "Н\_ЛЮДИ"."ИД" < 100012 и "Н\_ВЕДОМОСТИ"."ИД" = 1457443 соответственно, затем произойдет объединение с таблицей "Н\_СЕССИЯ" при помощи соединения по полю "Н\_ВЕДОМОСТИ"."СЭС\_ИД". Также возможна сортировка результата.

Оптимальным планом выполнения запроса является план №2, который использует индекс на таблице "Н\_ВЕДОМОСТИ" и позволяет сократить количество просматриваемых строк.

При добавлении индекса на таблицу "Н\_ЛЮДИ" планы могут измениться, так как запрос может начать использовать индекс на этой таблице в сочетании с индексом на таблице "Н\_ВЕДОМОСТИ", что также может привести к сокращению количества просматриваемых строк. Однако, при добавлении индекса на таблицу "Н\_СЕССИЯ", планы измениться не должны, так как в данном запросе не используется условие для этой таблицы в качестве фильтра.

**EXPLAIN ANALYZE**

Начало формы

Конец формы

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Из результатов можно увидеть, что для выполнения запроса был выбран план выполнения с использованием двух вложенных циклов (Nested Loop). Первый вложенный цикл соединяет таблицы "Н\_ЛЮДИ" и "Н\_ВЕДОМОСТИ" с использованием условия соединения по полю "ЧЛВК\_ИД". Второй вложенный цикл не используется, так как в запросе нет условий соединения таблицы "Н\_ВЕДОМОСТИ" с таблицей "Н\_СЕССИЯ".

Также можно увидеть стоимость плана выполнения (cost), которая равна 0.70, и количество строк, которые были ожидаемо обработаны (rows), равное 1. В данном случае запрос был выполнен очень быстро (Execution Time: 0.065 ms), что свидетельствует о том, что он не нагружает базу данных.

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

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Из результатов можно увидеть, что для выполнения запроса был выбран план выполнения с использованием двух вложенных циклов (Nested Loop). Первый вложенный цикл соединяет таблицы "Н\_ЛЮДИ" и "Н\_ВЕДОМОСТИ" с использованием условия соединения по полю "ЧЛВК\_ИД". Второй вложенный цикл соединяет таблицы "Н\_ВЕДОМОСТИ" и "Н\_СЕССИЯ" с использованием условия соединения по полю "СЭС\_ИД".

Также можно увидеть стоимость плана выполнения (cost), количество строк, которые были ожидаемо обработаны (rows), и фактическое время выполнения (actual time).

Из результатов также видно, что были использованы индексные сканы для таблиц "Н\_ЛЮДИ", "Н\_ВЕДОМОСТИ" и "Н\_СЕССИЯ", что может ускорить выполнение запроса.

В данном случае запрос был выполнен очень быстро (Execution Time: 0.058 ms), что свидетельствует о том, что он не нагружает базу данных.

**Вывод**

При выполнении данной лабораторной работы я изучила различные виды индексов и узнала, как использовать их для оптимизации скорости выполнения запросов.