УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Информационный системы и базы данных»

**Лабораторная работа №4**

Вариант 40273

Студент

Левченко Ярослав

P3118

Преподаватель

Инячина Диана

Санкт-Петербург, 2023 г.

**Описание 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 (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.ИМЯ, Н\_ВЕДОМОСТИ.ИД.  
   Фильтры (AND):   
   a) Н\_ЛЮДИ.ИД > 152862.  
   b) Н\_ВЕДОМОСТИ.ДАТА < 1998-01-05.  
   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.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД, Н\_СЕССИЯ.ДАТА.  
   Фильтры (AND):   
   a) Н\_ЛЮДИ.ИД > 100012.  
   b) Н\_ВЕДОМОСТИ.ИД < 1250981.  
   c) Н\_СЕССИЯ.ДАТА < 2002-01-04.  
   Вид соединения: INNER JOIN.

**Реализация первого запроса**

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.ИД > 152862.  
   b) Н\_ВЕДОМОСТИ.ДАТА < 1998-01-05.  
   Вид соединения: LEFT JOIN.

**Запрос:**

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

Автоматически созданное описание**

**Планы выполнения:**

а)

Изображение выглядит как текст, снимок экрана, диаграмма, дизайн

Автоматически созданное описание

б)

Изображение выглядит как текст, снимок экрана, диаграмма

Автоматически созданное описание

в)

Изображение выглядит как текст, снимок экрана, диаграмма

Автоматически созданное описание

Оптимальный план – б, т.к. сначала производится выбор строк по указанным условиям, а только потом отфильтрованные строки соединяются с помощью left join.

**Индексы:**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Вывод Explain Analyze:Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Реализация второго запроса**

**Запрос:**

**Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание**

а)

Изображение выглядит как текст, снимок экрана, диаграмма

Автоматически созданное описание

б)

Изображение выглядит как снимок экрана, текст, диаграмма, дизайн

Автоматически созданное описание

Оптимальным является план б, потому что все три таблицы имеют большое количество строк, а после выполнения последнего соединения их останется небольшое количество. За счет использования проекции и фильтрации на ранних этапах происходит соединение только нужных нам атрибутов, следовательно промежуточные данные меньше.

**Индексы:**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеВывод Explain Analyze:**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

Выводы

При выполнении лабораторной работы я познакомилась с особенностями составления и обработки планов СУБД при использовании и без использования индексов. Были изучены основные виды индексов и стратегии соединения таблиц, применяемых в данной СУБД.