Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

Университет ИТМО

Факультет программной инженерии и компьютерной техники

**Отчет по лабораторной работе №4**

**по дисциплине**

**«Информационные системы и базы данных»**

**Вариант 9446**

Выполнил: студент группы P33131

Бусыгин Дмитрий Алексеевич

Преподаватель:

Харитонова Анастасия Евгеньевна

Санкт-Петербург

2023

1. 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 (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\_СЕССИЯ.  
     Вывести атрибуты: Н\_ЛЮДИ.ИМЯ, Н\_СЕССИЯ.ЧЛВК\_ИД.  
     Фильтры (AND):  
     a) Н\_ЛЮДИ.ФАМИЛИЯ = Иванов.  
     b) Н\_СЕССИЯ.ИД < 14.  
     Вид соединения: LEFT 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.ИД, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.ГРУППА.  
     Фильтры: (AND)  
     a) Н\_ЛЮДИ.ИМЯ > Владимир.  
     b) Н\_ОБУЧЕНИЯ.НЗК > 001000.  
     c) Н\_УЧЕНИКИ.ИД < 100410.  
     Вид соединения: RIGHT JOIN.

1. Реализация запросов на SQL:

*select* НЛ."ИМЯ", НС."ЧЛВК\_ИД" *from* "Н\_ЛЮДИ" НЛ  
*left join* "Н\_СЕССИЯ" НС *on* НЛ."ИД" = НС."ЧЛВК\_ИД"  
*where* НЛ."ФАМИЛИЯ" = 'Иванов' *and* НС."ИД" < 20000

*select* НЛ."ИД", НО."НЗК", НУ."ГРУППА" *from* "Н\_ЛЮДИ" НЛ  
*right join* "Н\_ОБУЧЕНИЯ" НО *on* НЛ."ИД" = НО."ЧЛВК\_ИД"  
*right join* "Н\_УЧЕНИКИ" НУ *on* НО."ЧЛВК\_ИД" = НУ."ЧЛВК\_ИД"  
*where* НЛ."ИМЯ" > 'Владимир' *and* НО."НЗК" > '001000' *and* НУ."ИД" < 100410

1. Возможные планы выполнения запросов:

Изображение выглядит как текст, рукописный текст, Шрифт, каллиграфия

Автоматически созданное описание Изображение выглядит как текст, рукописный текст, Шрифт, каллиграфия

Автоматически созданное описание Изображение выглядит как текст, рукописный текст, Шрифт, линия

Автоматически созданное описание

Заметим, что при отсутствии индексов самой долгой операцией будет являться join, т.к. данных в таблицах много и будет полный квадратичный перебор обоих таблиц (абстрагируемся от встроенных индексов)

Фильтры же уменьшают выборки из таблиц в сотни раз (в конкретном примере), поэтому лучшим вариантом считаю вариант 2, где по обоим таблицам проходит фильтрация, а только потом происходит их объединение.

Изображение выглядит как текст, рукописный текст, Шрифт, каллиграфия

Автоматически созданное описание Изображение выглядит как текст, рукописный текст, Шрифт, линия

Автоматически созданное описание Изображение выглядит как текст, рукописный текст, Шрифт, каллиграфия

Автоматически созданное описание

Известно, что делать выборку раньше соединений – выгодно. Это связано с тем, что мы уменьшаем количество данных, которые будут проходить больше итераций (отсекаем ненужное сразу). В данном случае это актуально, т.к. каждый из фильтров разительно уменьшает выборку.

Поэтому остается выбрать между вариантами 2 и 3: какое объединение сделать раньше выгоднее. Мой выбор – вариант 2. Связываю это с тем, что выборка после объединения НЛ и НО весьма мала, а при объединении НО и НУ почти не изменяется.

1. Результаты EXPLAIN ANALYZE:

Запрос 1.

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

Автоматически созданное описание

Запрос 2.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Интересно, что, судя по плану, отрабатывают индексы, и правда, например я нашел индекс на фамилию в таблице с людьми

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

Автоматически созданное описание

1. Рекомендованные индексы:

Если посмотреть на то, сколько времени занимают этапы выполнения запросов, то можно сделать предположение о необходимости следующих индексов:

Запрос 1.

1. НЛ.ФАМИЛИЯ (btree, хотя можно было бы использовать hash, т.к. проверка на равенство)
2. НС.ЧЛВК\_ИД (btree, т.к. число) – ИЗЛИШНЕ

Запрос 2.

1. НЛ.ИМЯ (btree)
2. НО.НЗК (btree)
3. НУ.ИД (есть по умолчанию)

Каждый из этих индексов может ускорить работу КОНКРЕТНЫХ запросов, но при принятии решения о их добавлении необходимо в целом анализировать систему.

Вывод по работе:

В процессе выполнения работы я ознакомился с принципами работы разных типов индексов БД, покопался в изучении планов исполнения запросов, процесса их планирования. Помимо этого, я в очередной раз закрепил умение писать SQL-запросы.