Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант № 32932

**Лабораторная работа №4**

По дисциплине

Базы Данных

Выполнил студент группы P3121:

Фам Данг Чунг Нгиа

Преподаватель:

Карапетян Эрик Акопович

Санкт-Петербург 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).

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

Таблицы: Н\_ЛЮДИ, Н\_СЕССИЯ.

Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_СЕССИЯ.ИД.

Фильтры (AND):

a) Н\_ЛЮДИ.ИМЯ = Александр.

b) Н\_СЕССИЯ.УЧГОД < 2008/2009.

c) Н\_СЕССИЯ.УЧГОД > 2008/2009.

Вид соединения: INNER JOIN.

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

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

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

Фильтры (AND):

a) Н\_ЛЮДИ.ОТЧЕСТВО > Сергеевич.

b) Н\_ВЕДОМОСТИ.ИД < 39921.

Вид соединения: INNER JOIN.

1. **Реализация запросов на SQL**

|  |
| --- |
| -- Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  -- Таблицы: Н\_ЛЮДИ, Н\_СЕССИЯ.  -- Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_СЕССИЯ.ИД.  -- Фильтры (AND):  -- a) Н\_ЛЮДИ.ИМЯ = Александр.  -- b) Н\_СЕССИЯ.УЧГОД < 2008/2009.  -- c) Н\_СЕССИЯ.УЧГОД > 2008/2009.  -- Вид соединения: INNER JOIN.  SELECT Н\_ЛЮДИ.ФАМИЛИЯ, Н\_СЕССИЯ.ИД FROM Н\_ЛЮДИ      INNER JOIN Н\_СЕССИЯ ON Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД      WHERE Н\_ЛЮДИ.ИМЯ = 'Александр'          AND Н\_СЕССИЯ.УЧГОД < '2008/2009'          AND Н\_СЕССИЯ.УЧГОД > '2008/2009';  -- Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  -- Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.  -- Вывести атрибуты: Н\_ЛЮДИ.ИД, Н\_ВЕДОМОСТИ.ИД, Н\_СЕССИЯ.ИД.  -- Фильтры (AND):  -- a) Н\_ЛЮДИ.ОТЧЕСТВО > Сергеевич.  -- b) Н\_ВЕДОМОСТИ.ИД < 39921.  -- Вид соединения: INNER JOIN.  SELECT Н\_ЛЮДИ.ИД, Н\_ВЕДОМОСТИ.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.ОТЧЕСТВО > 'Сергеевич'          AND Н\_ВЕДОМОСТИ.ИД < 39921; |

1. **Уменьшение времени выполнения 1 запроса**

|  |
| --- |
| Индексы, добавление которых уменьшит время выполнения запроса:  a) На таблице **Н\_ЛЮДИ**:   * Индекс на атрибуте **ИМЯ** (B-tree) Это ускорит фильтрацию строк с условием **Н\_ЛЮДИ.ИМЯ = 'Александр'**.   b) На таблице **Н\_СЕССИЯ**:   * Индекс на атрибуте **УЧГОД** (B-tree) Это ускорит фильтрацию строк с условием **Н\_СЕССИЯ.УЧГОД < '2008/2009'** и **Н\_СЕССИЯ.УЧГОД > '2008/2009'**. * Индекс на атрибуте **ЧЛВК\_ИД** (B-tree) Это ускорит соединение таблиц по атрибуту **ЧЛВК\_ИД**.   Добавление индексов на указанные столбцы позволит ускорить поиск нужных записей по фильтрам в запросе.  Возможные планы выполнения запроса без индексов:  План 1:   * Полный скан таблицы **Н\_ЛЮДИ**. * Полный скан таблицы **Н\_СЕССИЯ** с применением фильтра по условиям   **Н\_СЕССИЯ.УЧГОД < '2008/2009'** и **Н\_СЕССИЯ.УЧГОД > '2008/2009'**.   * Соединение таблиц с использованием Nested Loops Join по атрибуту **ЧЛВК\_ИД**. * Фильтрация результата соединения по условию **Н\_ЛЮДИ.ИМЯ = 'Александр'**.   План 2:   * Полный скан таблицы **Н\_ЛЮДИ** с применением фильтра по условию **Н\_ЛЮДИ.ИМЯ = 'Александр'**. * Полный скан таблицы **Н\_СЕССИЯ** с применением фильтра по условиям **Н\_СЕССИЯ.УЧГОД < '2008/2009'** и **Н\_СЕССИЯ.УЧГОД > '2008/2009'**. * Соединение таблиц с использованием Nested Loops Join по атрибуту **ЧЛВК\_ИД**.   Оптимальный план:   * План 2, потому что фильтрация данных выполняется до соединения таблиц, что уменьшает количество строк для обработки.   При добавлении индексов планы выполнения запросов изменятся:   * Вместо полного скана таблиц будет использоваться индексный скан. * Nested Loops Join станет быстрее благодаря индексу на атрибуте **ЧЛВК\_ИД**. |

1. **План выполнения 1 запроса**

|  |
| --- |
| 1st QUERY PLAN |

1. **Уменьшение времени выполнения 2 запроса**

|  |
| --- |
| Индексы, добавление которых уменьшит время выполнения запроса:  a) На таблице **Н\_ЛЮДИ**:   * Индекс на атрибуте **ОТЧЕСТВО** (B-tree). Это ускорит фильтрацию строк с условием **Н\_ЛЮДИ.ОТЧЕСТВО > 'Сергеевич'**.   b) На таблице **Н\_СЕССИЯ**:   * Индекс на атрибуте **ЧЛВК\_ИД** (B-tree). Это ускорит соединение таблиц по атрибуту **ЧЛВК\_ИД**.   c) На таблице **Н\_ВЕДОМОСТИ**:   * Индекс на атрибуте **ИД** (B-tree). Это ускорит фильтрацию строк с условием **Н\_ВЕДОМОСТИ.ИД < 39921**.   Добавление индексов на указанные столбцы позволит ускорить поиск нужных записей по фильтрам в запросе.  Возможные планы выполнения запросов без индексов:  План 1:   * Полный скан таблицы **Н\_ЛЮДИ** с применением фильтра по условию **Н\_ЛЮДИ.ОТЧЕСТВО > 'Сергеевич'**. * Полный скан таблицы **Н\_ВЕДОМОСТИ** с применением фильтра по условию **Н\_ВЕДОМОСТИ.ИД < 39921**. * Соединение таблиц **Н\_ЛЮДИ** и **Н\_ВЕДОМОСТИ** с использованием Nested Loops Join по атрибуту **ЧЛВК\_ИД**. * Соединение результатов предыдущего этапа с таблицей **Н\_СЕССИЯ** с использованием Nested Loops Join по атрибуту **ЧЛВК\_ИД**.   План 2:   * Полный скан таблицы **Н\_ЛЮДИ** с применением фильтра по условию **Н\_ЛЮДИ.ОТЧЕСТВО > 'Сергеевич'**. * Соединение таблиц **Н\_ЛЮДИ** и **Н\_СЕССИЯ** с использованием Nested Loops Join по атрибуту **ЧЛВК\_ИД**. * Полный скан таблицы **Н\_ВЕДОМОСТИ** с применением фильтра по условию **Н\_ВЕДОМОСТИ.ИД < 39921**. * Соединение результатов предыдущих 2-х этапов вместе с использованием Hash Join по атрибуту **ЧЛВК\_ИД**.   Оптимальный план:   * План 1, так как Nested Loops Join предпочтительнее Hash Join в случаях, когда обрабатываемые наборы данных небольшие, и заранее известно, что результаты фильтрации будут содержать малое количество строк.   При добавлении индексов планы выполнения запросов изменятся:   * Вместо полного скана таблиц будет использоваться индексный скан. * Nested Loops Join станет быстрее благодаря индексам на атрибутах **ЧЛВК\_ИД**. |

1. **План выполнения 2 запроса**

|  |
| --- |
| 2nd QUERY PLAN |

1. **Вывод**

В ходе выполнения лабораторной работы я освоил работу с реляционной алгеброй и научился строить планы выполнения запросов, а также их диаграммы. Я изучил различные виды индексов и узнал, как использовать их для оптимизации запросов. Теперь я могу применять полученные знания для эффективной работы с базами данных и повышения производительности SQL-запросов.