Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА №4**

по дисциплине

‘Информационные системы и базы данных’

Вариант 11208

*Выполнил:*

Студент группы P33312

Соболев Иван Александрович

*Преподаватель:*

Наумова Надежда Александровна



Санкт-Петербург, 2023

**Задание:**

Изображение выглядит как текст, Шрифт, снимок экрана, алгебра

Автоматически созданное описание

**Реализация запросов на SQL:**

Изображение выглядит как текст, Шрифт, снимок экрана, алгебра

Автоматически созданное описание

SELECT Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ИД FROM Н\_ОЦЕНКИ  
 INNER JOIN Н\_ВЕДОМОСТИ ON Н\_ВЕДОМОСТИ.ОЦЕНКА = Н\_ОЦЕНКИ.КОД  
 WHERE Н\_ОЦЕНКИ.КОД = 'неявка' AND  
 Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 153285 AND  
 Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 142390;

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

SELECT Н\_ЛЮДИ.ИД, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.ИД  
FROM Н\_ЛЮДИ  
 LEFT JOIN Н\_ОБУЧЕНИЯ ON Н\_ЛЮДИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД  
 LEFT JOIN Н\_УЧЕНИКИ ON Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД  
WHERE Н\_ЛЮДИ.ИМЯ > 'Александр'  
 AND Н\_ОБУЧЕНИЯ.НЗК::integer > 001000;

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Индексы:**

**Для первого запроса целесообразно создать следующие индексы:**

create index points on Н\_ОЦЕНКИ using hash(КОД);  
create index statements on Н\_ВЕДОМОСТИ using btree(ЧЛВК\_ИД);

Для таблицы Н\_ОЦЕНКИ для атрибута КОД целесообразно создать индекс Hash, потому что в запросе используется только оператор «=», который Hash при этом алгоритмическая сложность у Hash – O(1), что гораздо быстрее, чем у B-tree. Добавление индекса ускорит выполнение операций WHERE и ON.

Для таблицы Н\_ВЕДОМОСТИ для атрибута ЧЛВК\_ИД целесообразно создать индекс B-tree, потому что помимо оператора «=» в запросе используется и оператор «<», который не поддерживается индексом. При этом алгоритмическая сложность будет – O(logN).

**Для второго запроса целесообразно создать следующие индексы:**

create index chlvk\_id\_index on Н\_ОБУЧЕНИЯ using hash(ЧЛВК\_ИД);  
create index name\_index on Н\_ЛЮДИ using btree(ИМЯ);  
create index nzk\_index on Н\_ОБУЧЕНИЯ using btree(НЗК);

Для таблицы Н\_ОБУЧЕНИЯ для атрибута ЧЛВК\_ИД целесообразно создать индекс Hash, потому что в запросе используется только оператор «=». Создание данного индекса ускорить операции соединения таблиц.

Для таблицы Н\_ЛЮДИ для атрибута ИМЯ целесообразно создать индекс B-tree, потому что помимо оператора «=» в запросе используется и оператор «>». Создание данного индекса ускорить операцию WHERE.

Для таблицы Н\_ОБУЧЕНИЯ для атрибута НЗК целесообразно создать индекс B-tree, потому что помимо оператора «=» в запросе используется и оператор «>». Создание данного индекса ускорить операцию WHERE.

**Возможные планы выполнения запросов:**

**Первый запрос:**

1. Сначала происходит соединение отношений, далее последовательная выборка, результатом является проекция двух атрибутов.

Изображение выглядит как текст, диаграмма, Шрифт, снимок экрана

Автоматически созданное описание

1. Изначально производятся операции выборки, далее соединение двух отношений. Результатом является проекция двух атрибутов. Изображение выглядит как текст, диаграмма, Шрифт, карта

   Автоматически созданное описание
2. Для начала производится выборка для двух отношений, далее получается проекция для двух отношений из атрибутов, необходимых для выборки, соединения и итоговой проекции, далее соединение двух отношений. Результатом является проекция двух атрибутов.

Изображение выглядит как текст, диаграмма, Шрифт, снимок экрана

Автоматически созданное описание

1. Для начала получается проекция для двух отношений из атрибутов, необходимых для выборки, соединения и итоговой проекции, далее производится выборка и соединение. Результатом является проекция двух атрибутов. Изображение выглядит как текст, диаграмма, Шрифт, карта

   Автоматически созданное описание
2. Для начала получается выборка для одного из отношений из атрибутов, далее происходит соединение таблиц, после которого производится выборка для результата соединения. Результатом является проекция двух атрибутов. В данном плане атрибуты выборки после соединения могут меняться – могут быть выборки из ветки Н\_ВЕДОМОСТИ.

Изображение выглядит как текст, диаграмма, Шрифт

Автоматически созданное описание

1. Для начала получается проекция для двух отношений из атрибутов, необходимых для выборки, соединения и итоговой проекции, далее производится выборка для одного из отношений, а после соединение двух отношений, после которого производится выборка для результата соединения. Результатом является проекция двух атрибутов. В данном плане атрибуты выборки после соединения могут меняться – могут быть выборки из ветки Н\_ВЕДОМОСТИ.

Изображение выглядит как текст, диаграмма, Шрифт, зарисовка

Автоматически созданное описание

Из составленных возможных планов выполнения запроса лучшим является четвертый, поскольку в нем изначально получаются проекции и производятся выборки, а уже после этого выполняется соединение. Это позволяет уменьшить размер хранимых данных.

При создании индексов четвертый план останется оптимальным, при этом даже ускорится за счёт ускорения поиска. Также довольно эффективным будет третий план, потому что скорость поиска в нем увеличится. В нем также соединение происходит после выборки, что позволяет ускорить выполнение запроса, но, в отличие от четвертого плана, в нем выборка будет происходить по всей таблице целиком, а не по отдельным проекциям.

**Второй запрос:**

1. Соединение двух отношений, далее результат соединения соединяется с третьим отношением, а для результата двух соединений последовательно производится выборка. Результатом является проекция трех атрибутов.

Изображение выглядит как текст, диаграмма, Шрифт, чек

Автоматически созданное описание

1. Для начала производится выборка для первых двух отношений, далее производится соединение этих двух отношений, а после результат соединения соединяется с третьим отношением. Результатом является проекция трех атрибутов.

Изображение выглядит как текст, диаграмма, Шрифт, карта

Автоматически созданное описание

1. Сначала производится выборка для одного отношения, далее производится соединение первых двух отношений, а после результат соединения соединяется с третьим отношением. Происходит выборка по результату соединения. Результатом является проекция трех атрибутов. На первом шаге выборка может быть у ветки Н\_ОБУЧЕНИЯ.

Изображение выглядит как текст, диаграмма, Шрифт, карта

Автоматически созданное описание

1. Сначала получается проекция для двух отношений, после по ним производится выборка и соединение. Далее результат соединения соединяется с третьим отношением, для которого перед соединением была получена проекция. Результатом является проекция трех атрибутов.

Изображение выглядит как текст, диаграмма, Шрифт

Автоматически созданное описание

1. Сначала формируются выборки для каждого отношения, далее строятся проекции. После чего данные проекции соединяются. Результатом является проекция трех атрибутов.

Изображение выглядит как текст, диаграмма, Шрифт, карта

Автоматически созданное описание

1. Получаются проекции для двух отношений, после чего эти отношения соединяются, а результат соединения соединяется с третьим отношением, для которого перед соединением тоже была получена проекция. Далее производится последовательная выборка. Результатом является проекция трех атрибутов.

Изображение выглядит как текст, диаграмма, Шрифт, чек

Автоматически созданное описание

Из составленных возможных планов выполнения запроса лучшим является четвертый, поскольку в нем изначально получаются проекции и производятся выборки, а уже после этого выполняется соединение. Это позволяет уменьшить размер хранимых данных.

При создании индексов четвертый план останется оптимальным, при этом даже ускорится за счёт ускорения поиска. Также довольно эффективным будет пятый план, потому что скорость поиска в нем увеличится. В нем также соединение происходит после выборки, что позволяет ускорить выполнение запроса, но, в отличие от четвертого плана, в нем выборка будет происходить по всей таблице целиком, а не по отдельным проекциям.

**Команда EXPLAIN ANALYZE:**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Выводы по работе:**

В ходе выполнения данной лабораторной работы я познакомился с индексами, тем как они влияют на нагрузку на систему. Также я познакомился с планом выполнения запроса, узнал каким образом СУБД выбирает оптимальный. Узнал, что выполняет команда EXPLAIN ANALYZE.