Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования  
**"Национальный Исследовательский Университет ИТМО"**

Факультет Программной Инженерии и Компьютерных Технологий

**Лабораторная работа №4**

по дисциплине

**«Информационные системы и базы данных»**

Вариант 87236

Выполнил:

Студент группы P33302

Тюрин Святослав Вячеславович

Преподаватель

Николаев Владимир Вячеславович

Санкт Петербург

2023

**Изображение выглядит как текст, План, диаграмма, черно-белый

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание**

**Ход работы:**

Первый запрос

select Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД  
from Н\_ЛЮДИ  
 join Н\_ВЕДОМОСТИ on Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД  
where Н\_ЛЮДИ.ИД > 163484  
 and Н\_ВЕДОМОСТИ.ИД < 1250981  
 and Н\_ВЕДОМОСТИ.ИД < 1457443;

**Индексы**

Для таблицы Н\_ЛЮДИ для атрибута ИД целесообразно создать индекс b-tree, потому что в запросе используется оператор = и <. Добавление индекса ускорит выполнение операций WHERE и ON.

Для таблицы Н\_ВЕДОМОСТИ для атрибута ЧЛВК\_ИД 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 ON.

create index Н\_ВЕДОМОСТИ\_ИД on Н\_ВЕДОМОСТИ hash(ЧЛВК\_ИД);

create index Н\_ЛЮДИ\_ИД on Н\_ЛЮДИ using btree(ИД);

**План выполнения**

**Изображение выглядит как текст, диаграмма, Шрифт, число

Автоматически созданное описание**

Оптимальным планом является второй.

В первом плане мы соединяем 2 полные таблицы без предварительной проекции и отбрасывания ненужных строк, а значит тащим ненужные данные наверх. И только после этого проверяемся на условия и делаем проекцию. Цель – уменьшение размера промежуточных данных, не выполнена.

И напротив, во 2-ом плане мы в самом начале делаем выборку строк и тут же делаем проекцию, таким образом мы избавляемся от большого кол-ва ненужных нам данных в начале, и уже на соединении таблиц, на тяжелой операции, данных намного меньше.

Изображение выглядит как снимок экрана

Автоматически созданное описание**Eplain analyze**

Nested loop всегда имеет два дочерних узла. Первый это внешний набор данных, а второй – внутренний. И условие по склейке таблиц лежит во втором узле. Сначала выбирается требуемые строки из первой таблицы, а потом для каждой строки результата идёт запрос ко второй таблице для выборки нужных данных.

У нас происходит индексное сканирование по таблице Н\_ЛЮДИ с условием(index cond).

Потом для каждой строки, происходит индексное сканирование по второй таблице с условием объединения, и потом данные фильтруются

Второй запрос

select Н\_ЛЮДИ.ИМЯ, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.ГРУППА

from Н\_ОБУЧЕНИЯ

right join Н\_ЛЮДИ on Н\_ЛЮДИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД

right join Н\_УЧЕНИКИ on Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД

where Н\_ЛЮДИ.ОТЧЕСТВО < 'Сергеевич'

and Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 112514;

Индексы

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

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

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

create index Н\_ЛЮДИ\_ИД on Н\_ЛЮДИ using hash (ИД);

create index Н\_УЧЕНИКИ\_ЧЛВК\_ИД on Н\_УЧЕНИКИ using hash(ЧЛВК\_ИД);

create index Н\_ОБУЧЕНИЕ\_ЧЛВК\_ИД on H\_ОБУЧЕНИЕ using btree(ЧЛВК\_ИД);

**ПЛАН ВЫПОЛНЕНИЯ**

**Изображение выглядит как текст, диаграмма, Шрифт, число

Автоматически созданное описание**

В первом плане мы соединяем 3 полные таблицы без предварительной проекции и отбрасывания ненужных строк, а значит тащим ненужные данные наверх. И только после этого проверяемся на условия и делаем проекцию. Цель – уменьшение размера промежуточных данных, не выполнена.

И напротив, во 2-ом плане мы в самом начале делаем выборку строк и тут же делаем проекцию, таким образом мы избавляемся от большого кол-ва ненужных нам данных в начале, и уже на соединении таблиц, на тяжелой операции, данных намного меньше.

**Eplain analyze**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

Объединение хешированием подразумевает загрузку строк-кандидатов из одной таблицы в хеш-таблицу (Hash в плане выполнения запроса), после чего они сравниваются с каждой строкой другой таблицы. Таким образом, устраняется недостаток объединения с использованием вложенных циклов (Nested loops), потому что внутренний цикл (тестирование выборки из первой таблицы с каждой строкой второй таблицы) теперь состоит из быстрых операций, благодаря хеш-таблице.

SEQ SCAN — это последовательный перебор всех строк базы в поисках интересующего значения. Обычно операция выполняется в десятки раз медленнее, чем индексное сканирование, но то, какая из операций будет применяться в конкретных обстоятельствах, зависит от ограничивающих условий запроса. Если индекса для столбца, в котором нужно найти определенное значение, вообще не создано, то будет выполняться последовательное сканирование. В некоторых случаях, например, при большом объеме данных для одного значения индексного поля, последовательное сканирование может быть эффективнее индексного, так как обычно оно работает с большими блоками данных, поэтому за одну операцию доступа потенциально может выбрать большее количество данных, чем индексное сканирование, соответственно, нужно меньше операций доступа и выше скорость.

Можем заметить, что у нас происходит сканирование и фильтрация Н\_ЛЮДИ, потом результат хешируется, потом сканирование и фильтрация Н\_ОБУЧЕНИЕ, результат обьединяется и хешируется, и сканирование Н\_УЧЕНИКИ и обьединение.

**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.