МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Базы данных»

Тема: Нагрузочное тестирование БД.

Студент гр. 1303 _____ Кузнецов Н.А.

Преподаватель _____ Заславский М.М.

Санкт-Петербург
2023

Цель работы.

Заполнить большим количеством тестовых данных, измерить время выполнения запросов. Измерить влияние (или его отсутствие) индексов на скорость выполнения запросов.

Задание.

Вариант 13.

Написать скрипт, заполняющий БД большим количеством тестовых данных, рекомендуется использовать faker.js.

Измерить время выполнения запросов, написанных в ЛР3.

Проверить для числа записей:

100 записей в каждой табличке

1.000 записей

1.0000 записей

1.000.000 записей

можно больше.

Все запросы выполнять с фиксированным ограничением на вывод (LIMIT), т.к. запросы без LIMIT всегда будет выполняться O(n) от кол-ва записей.

Проверить влияние сортировки на скорость выполнения запросов.

Для измерения использовать фактическое (не процессорное и т.п.) время. Для node.js есть console.time и console.timeEnd.

Добавить в БД индексы (хотя бы 5 штук). Измерить влияние (или его отсутствие) индексов на скорость выполнения запросов.

Выполнение работы.

Для генерации данных использовалась библиотека faker.

Закончив заполнения базы данных было замерено время, необходимое для обработки запросов из лабораторной работы №3:

1. 100 записей - 11ms

2. 1.000 записей - 12ms

3. 10.000 записей - 26ms

4. 100.000 записей - 170ms

5. 1.000.000 записей - 1.6s

Данные представлены на графике 1

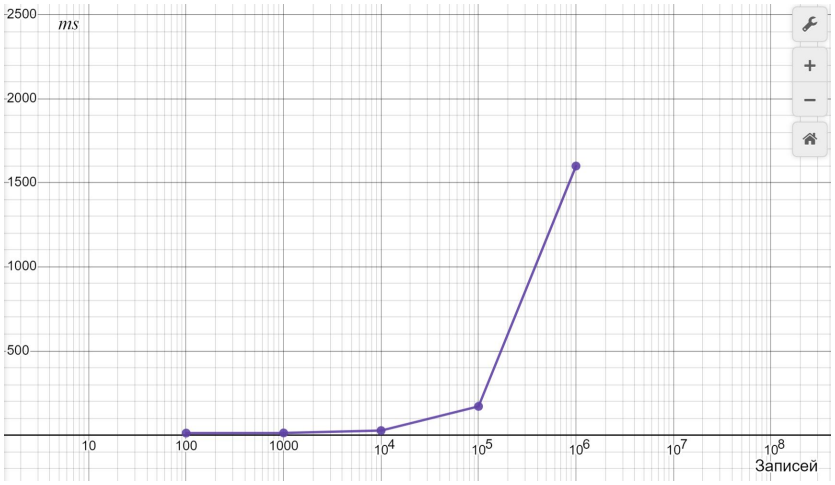

График 1 - запросы без сортировки и без индексов

Далее рассмотрим влияние сортировки при добавлении ее для каждого запроса.

1. 100 записей - 11ms

2. 1.000 записей - 14ms

3. 10.000 записей - 37ms

4. 100.000 записей - 461ms

5. 1.000.000 записей - 2.6s

Данные представлены на графике 2

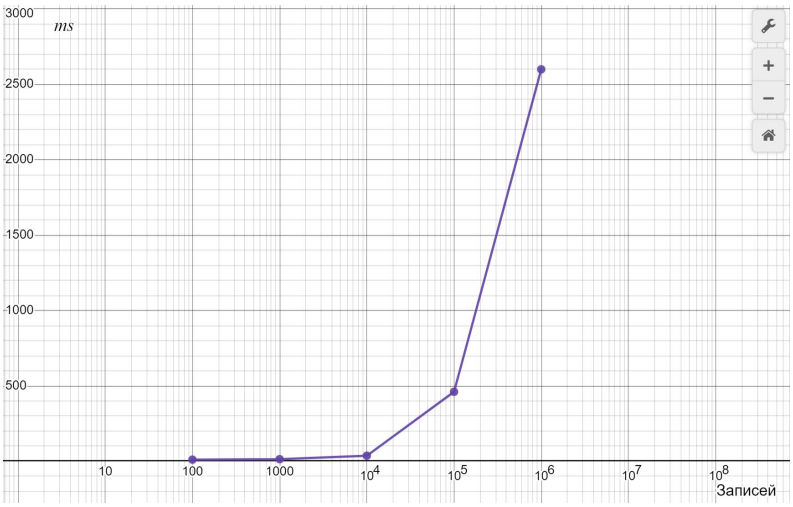

График 2 - запросы с сортировкой и без индексов

Можно сделать вывод, что при сортировке запросы выполнились медленнее.

Далее были заданы индексы для полей отношений Cinema, Film, Prize, Role, Session, так как поля данных таблиц чаще всего используются в запросах.

Замерено время после добавления индексов и получены результаты, которые говорят о том, что использование индексов уменьшает время необходимое на обработку запросов к БД.

Результаты:

1. 100 записей - 11ms
2. 1.000 записей - 12ms
3. 10.000 записей - 20ms
4. 100.000 записей - 86ms
5. 1.000.000 записей - 663ms

Данные представлены на графике 3

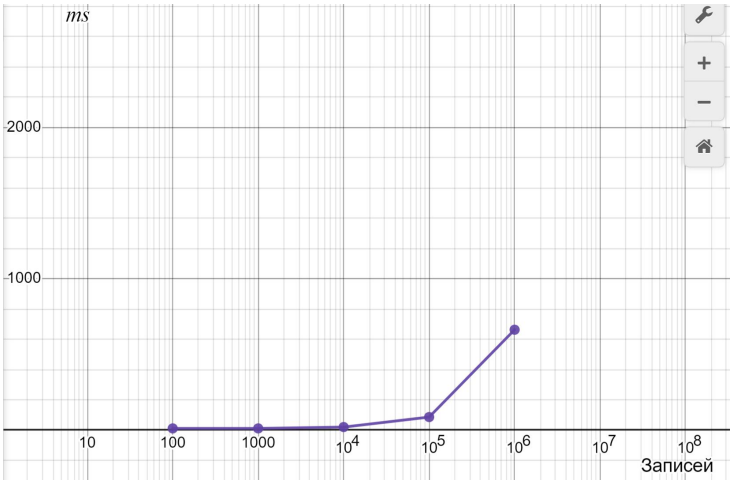

График 3 - запросы с индексами

Из результатов также видно, что при малых количествах данных, разница по времени в обработке запросов на уровне погрешности. Когда же кол-во данных больше или равно десять тысяч, разница является значительной, особенно это заметно при обработке 1.000.000 записей, время сократилось значительно.

Разработанный код см. в Приложении А.

Выводы.

В ходе лабораторной работы была заполнена база данных большим количеством тестовых данных: 100 записей в каждой таблице, 1000 записей, 100000 записей, 10000 записей, 1000000 записей, измерено время выполнения запросов. Также выявлено влияние индексов на скорость выполнения запросов и сортировки. Из результатов эксперимента можно сделать вывод, что при большом количестве данных в таблицах время обработки сильно сокращается при наличии индексов. При малых количествах данных разница по времени в обработке запросов незначительна.

ПРИЛОЖЕНИЕ А

Ссылка на PR https://github.com/moevm/sql-2023-1303/pull/59