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

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Базы данных»

Тема: Нагрузочное тестирование БД

Студент гр. 1303 _____ Депрейс А.С.

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.

Текст задания

Вариант 7

1. Написать скрипт, заполняющий БД большим количеством тестовых данных.

2. Измерить время выполнения запросов, написанных в ЛР3.

Проверить для числа записей:

100 записей в каждой табличке

1.000 записей

10.000 записей

100.000 записей

1.000.000 записей

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.

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

Обратите внимание на:

Скорость сортировки больших табличек

Скорость JOIN

Выполнение работы

Для генерации данных использовалась библиотека faker-js.

После генерации всех данных они были отправлены базе данных, время, которое требовалось на заполнение таблиц измерялось.

Затем были сделаны запросы из лабораторной работы 3, там, где не подразумевалось сортировки, были созданы дублирующие запросы, но с сортировкой, они выполнялись после основных запросов.

Затем были добавлены индексы в таблицы: WatchingChickenInCage и Chicken, так как они используются чаще всего. Запросы были повторены.

Время каждого запроса было измерено:

	1	2	3	4	5	6	7	заполнение
Б. И. 100	4.538ms	5.726 ms	2.726ms	2.669ms	2.685ms	1.564ms	3.21ms	82.011ms
Б. И. 100 с Сортировкой	0.862ms		1.893ms	2.105ms	2.617ms			
Б. И. 1000	5.66ms	5.257 ms	4.004ms	2.014ms	7.885ms	2.498ms	3.867ms	215.123 ms
Б. И. 1000 с Сортировкой	0.905ms		2.382ms	1.889ms	7.662ms			

Б. И. 10000	4.751ms	6.174 ms	6.139ms	2.492ms	60.168 ms	8.501ms	15.711 ms	958.995 ms
Б. И. 10000 с Сортировкой	1.678ms		6.763ms	2.309ms	59.044 ms			
Б. И. 100000	12.8ms	15.194 ms	26.789 ms	2.581ms	595.168ms	101.107ms	105.49 ms	12.404s
Б. И. 100000 с Сортировкой	7.964ms		169.878ms	1.521ms	600.972ms			
Б. И. 1000000	68.009 ms	71.373 ms	119.821ms	45.81ms	5.549s	624.108ms	430.213ms	7:24.739 (m:ss.mmm)
Б. И. 1000000 с Сортировкой	141.012ms		130.517ms	8.416ms	6.376s			
С. И. 100	3.582ms	5.48ms	2.358ms	1.933ms	3.107ms	2.084ms	3.113ms	76.089ms
С. И. 100 с Сортировкой	0.988ms		2.569ms	1.663ms	2.208ms			

С. И. 1000	4.585ms	5.541 ms	4.208ms	2.107ms	8.812ms	1.847ms	9.103ms	215.471 ms
С. И. 1000 с Сортировкой	0.802ms		2.293ms	1.923ms	8.011ms			
С. И. 10000	4.789ms	5.804 ms	4.516ms	2.944ms	67.723 ms	2.1ms	11.587 ms	1.161s
С. И. 10000 с Сортировкой	1.905ms		4.709ms	2.393ms	57.689 ms			
С. И. 100000	11.136 ms	5.779 ms	4.364ms	1.901ms	456.535ms	2.077ms	11.475 ms	12.240s
С. И. 100000 с Сортировкой	29.919 ms		18.183 ms	1.663ms	555.173ms			
С. И. 1000000	75.87ms	51.795 ms	55.889 ms	48.484 ms	5.812s	110.627ms	130.014ms	8:43.913 (m:ss.mmm)
С. И. 1000000 с Сортировкой	98.977 ms		118.261ms	130.055ms	5.960s			

Таблица 1 – Время выполнения запросов

Из результатов видно, что при малых количествах данных, разница по времени запросов незначительна и является погрешностью. Когда же кол-во данных переходит за десять тысяч, можно наблюдать, как начинает изменяться время запросов, например 5 и 7. Запросы с сортировкой, выполненные после основных запросов показывали время меньшее, чем основные запросы, но после преодоления отметки в 1000000 запросы с сортировкой начинают занимать большее время, чем основные.

После добавление индексации, скорость значительно уменьшилась для некоторых запросов, в том числе для тех , в которых использовался join длдя таблиц, которые были проиндексированы. А скорость запросов с сортировкой почти никак не изменилась.

Выводы.

Таблицы заполнены большим количеством тестовых данных, измерено время выполнения запросов. Индексы значительно уменьшают время обработки обычных запросов.

ПРИЛОЖЕНИЕ А

Pull request: https://github.com/moevm/sql-2023-1303/pull/53