**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 |  | Коренев Д.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.

**Текст задания**

Вариант 11

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.

1. 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. Таблицы были заполнены сгенерированными данными. Затем были сделаны запросы из лабораторной работы 3. Запросы выполнялись асинхронно, а общее время выполнения всех запросов засекалось от момента вызова первого запроса и заканчивалось, когда все запросы были выполнены.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Объем данных | task1 | task2 | task3 | task4 | task5 | task6 | Общее время |
| 100 | 38.87ms | 47.528ms | 55.548ms | 25.454ms | 71.745ms | 67.662ms | 77.596ms |
| 1000 | 32.201ms | 39.983ms | 41.572ms | 18.964ms | 63.352ms | 55.87ms | 67.75ms |
| 10000 | 28.117ms | 33.604ms | 35.199ms | 18.947ms | 96.289ms | 71.413ms | 99.935ms |
| 100000 | 38.191ms | 51.253ms | 61.28ms | 26.744ms | 558.682ms | 107.082ms | 562.58ms |
| 1000000 | 294.477ms | 658.571ms | 563.511ms | 380.544ms | 10.239s | 569.824ms | 10.247s |

В таблице 1 представлено время выполнения задач и общее время их выполнения в зависимости от количества записей в базе данных.

Таблица 1 время выполнения запросов без индексов

Затем для каждого отношения были добавлены индексы. В таблице 2 представлено время выполнения задач и общее время их выполнения в зависимости от количества записей в базе данных с индексами. На их основе можно сделать вывод: добавление инедексов ускоряет процесс выполнения запросов (по крайней мере в представленной работе).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Объем данных | task1 | task2 | task3 | task4 | task5 | task6 | Общее время |
| 100 | 19.15ms | 21.911ms | 22.893ms | 12.161ms | 30.56ms | 29.005ms | 33.751ms |
| 1000 | 17.371ms | 19.77ms | 20.578ms | 11.299ms | 32.364ms | 29.212ms | 34.812ms |
| 10000 | 20.015ms | 21.835ms | 23.768ms | 13.254ms | 49.132ms | 47.34ms | 52.276ms |
| 100000 | 26.469ms | 28.546ms | 29.623ms | 19.529ms | 249.019ms | 78.24ms | 251.559ms |
| 1000000 | 113.599ms | 117.206ms | 117.117ms | 113.299ms | 3.167s | 266.285ms | 3.170s |

Таблица 2 время выполнения запросов с индексами

В каждый запрос, в котором нет сортировки, добавлю сортировку по первому столбцу. В таблице 3 представлены данные выполнения запросов без индексов, а в таблице 4 с индексами.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Объем данных | task1 | task2 | task3 | task4 | task5 | task6 | Общее время |
| 100 | 19.856ms | 21.924ms | 29.75ms | 12.271ms | 31.982ms | 29.75ms | 34.723ms |
| 1000 | 18.929ms | 21.677ms | 22.574ms | 12.328ms | 33.956ms | 29.717ms | 36.687ms |
| 10000 | 22.174ms | 25.761ms | 29.246ms | 12.676ms | 72.182ms | 50.45ms | 74.788ms |
| 100000 | 63.894ms | 71.636ms | 84.185ms | 20.44ms | 502.584ms | 89.78ms | 505.19ms |
| 1000000 | 137.813ms | 227.871ms | 254.448ms | 264.491ms | 348.843ms | 6.436s | 6.440s |

Таблица 3 время выполнения запросов с сортировкой и без индексов

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Объем данных | task1 | task2 | task3 | task4 | task5 | task6 | Общее время |
| 100 | 24.303ms | 26.412ms | 27.269ms | 15.226ms | 36.231ms | 34.134ms | 39.094ms |
| 1000 | 17.54ms | 19.564ms | 20.693ms | 11.551ms | 36.172ms | 30.519ms | 38.761ms |
| 10000 | 30.731ms | 37.101ms | 38.157ms | 17.299ms | 83.632ms | 59.411ms | 86.421ms |
| 100000 | 31.269ms | 89.691ms | 97.041ms | 22.432ms | 559.47ms | 101.593ms | 562.747ms |
| 1000000 | 116.955ms | 7.068s | 117.634ms | 127.025ms | 7.678s | 250.458ms | 7.681s |

Таблица 4 время выполнения запросов с сортировкой и индексами

Сортировка занимает больше ресурсов и, соответственно, с времени выполнения. Однако с время выполнения с индексами оказалось немного больше.

**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/55