**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

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1303 |  | Новак П.И. |
| 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 16

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.

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 установки необходимых библиотек использовались следующие команды:

npm install sequelize

npm i @faker-js/faker

Для генерации необходимых данных был создан файл generator.js, где с помощью faker.js были сгенерированы данные клиентов, служащих, даты заезда и выезда. Все данные были сгенерированы в размере 1000000. Сгенерированные данные были записаны в json файл с соответствующим названием. При заполнении базы данных импортировались и использовались данные из необходимых json файлов.

Затем было замерено время выполнения всех запросов для разного количества записей в таблице. Ниже представлены средние значения времени для разных количеств записей:

1. 100 записей – 5.508 мс
2. 1000 записей – 4.873 мс
3. 10000 записей – 5.204 мс
4. 100000 записей – 5.455 мс
5. 1000000 записей – 11.759 мс

Далее для самых часто используемых отношений были добавлены индексы. Ниже представлены средние значения времени для разных количеств записей с использованием индексов:

1. 100 – 4.99 мс
2. 1000 – 5.36 мс
3. 10000 – 5.183 мс
4. 100000 – 5.227 мс
5. 1000000 – 10.947 мс

Можно заметить, что время выполнения запросов уменьшилось благодаря использованию индексов.

**Выводы.**

В ходе лабораторной работы база данных была заполнена большим количеством тестовых данных, было измерено время выполнения запросов. Выяснили, что индексы значительно уменьшают время обработки запросов.

**ПРИЛОЖЕНИЕ А**

Pull request: <https://github.com/moevm/sql-2023-1303/pull/52>