**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 |  | Бутыло Е.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 3

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.

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

**Выполнение работы**

Для выполнения работы был написан код на языке Python, для создания файлов типа json, в которых будут данные для заполнения БД.

Для генерации данных использовалась библиотека faker.

После получения данных, для их корректного использования были разработаны контроллеры, для каждого отношения, получающего информацию из файла.

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

1. 100 записей - 20ms
2. 1.000 записей - 20ms
3. 10.000 записей - 27ms
4. 100.000 записей - 160ms
5. 1.000.000 записей - 1s

Далее задали индексы для полей отношений Schedule и Teacher, так как они используются чаще всего.

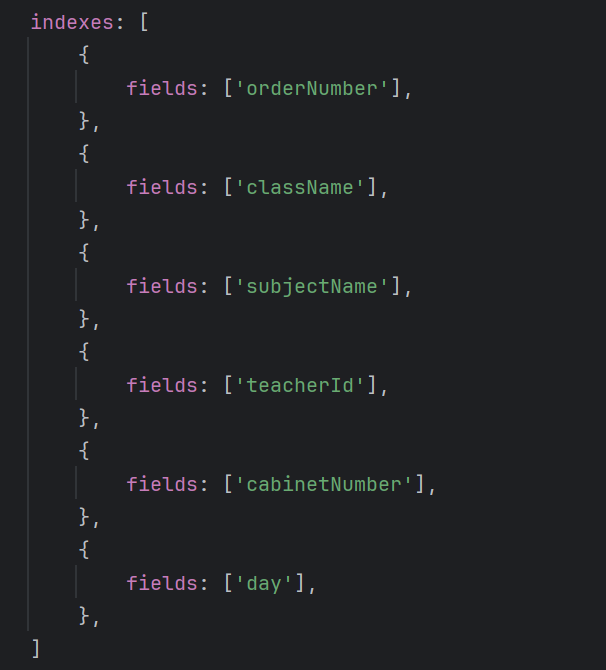


Рисунок 1. – Задали индексы для полей отношения Schedule.

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

Результаты:

1. 100 записей - 20ms

2. 1.000 записей - 20ms

3. 10.000 записей - 27ms

4. 100.000 записей - 35ms

5. 1.000.000 записей - 100ms

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

**Выводы.**

Заполнить большим количеством тестовых данных, измерили время выполнения запросов. Индексы значительно уменьшают время обработки запросов.

**ПРИЛОЖЕНИЕ А**

Pull request: <https://github.com/moevm/sql-2023-1303/pull/51>

**ПРИЛОЖЕНИЕ Б**

Файл lab2.sql:

CREATE OR REPLACE FUNCTION random\_between(low INT, high INT)

RETURNS INT AS

$$

BEGIN

RETURN floor(random() \* (high - low + 1) + low);

END;

$$ language 'plpgsql' STRICT;

CREATE TABLE class

(

class\_name TEXT PRIMARY KEY

);

CREATE TABLE subject

(

subject\_name TEXT PRIMARY KEY

);

CREATE TABLE cabinet

(

cabinet\_number SMALLINT PRIMARY KEY

);

CREATE TABLE teacher

(

teacher\_id SMALLSERIAL PRIMARY KEY,

second\_name TEXT NOT NULL,

first\_name TEXT NOT NULL,

patronymic TEXT NOT NULL

);

CREATE TABLE student

(

student\_id SMALLSERIAL PRIMARY KEY,

class\_name TEXT,

second\_name TEXT NOT NULL,

first\_name TEXT NOT NULL,

patronymic TEXT NOT NULL,

FOREIGN KEY (class\_name) REFERENCES class (class\_name) ON DELETE SET NULL

);

CREATE TABLE grade

(

student\_id SMALLINT NOT NULL,

subject\_name TEXT NOT NULL,

value SMALLINT,

PRIMARY KEY (student\_id, subject\_name),

FOREIGN KEY (student\_id) REFERENCES student (student\_id) ON DELETE CASCADE,

FOREIGN KEY (subject\_name) REFERENCES subject (subject\_name) ON DELETE CASCADE

);

CREATE TABLE teacher\_cabinet

(

teacher\_id SMALLINT PRIMARY KEY,

cabinet\_number SMALLINT NOT NULL,

FOREIGN KEY (teacher\_id) REFERENCES teacher (teacher\_id) ON DELETE CASCADE,

FOREIGN KEY (cabinet\_number) REFERENCES cabinet (cabinet\_number) ON DELETE CASCADE

);

CREATE TABLE school\_schedule

(

subject\_schedule\_id SERIAL PRIMARY KEY,

order\_number SMALLINT NOT NULL,

day TEXT NOT NULL,

class\_name TEXT NOT NULL,

subject\_name TEXT NOT NULL,

teacher\_id SMALLINT,

cabinet\_number SMALLINT,

FOREIGN KEY (class\_name) REFERENCES class (class\_name) ON DELETE CASCADE,

FOREIGN KEY (subject\_name) REFERENCES subject (subject\_name) ON DELETE CASCADE,

FOREIGN KEY (teacher\_id) REFERENCES teacher (teacher\_id) ON DELETE SET NULL,

FOREIGN KEY (cabinet\_number) REFERENCES cabinet (cabinet\_number) ON DELETE SET NULL

);

INSERT INTO cabinet

(cabinet\_number)

VALUES (11),

(12),

(13), -- free

(14),

(15),

(16),

(17);

INSERT INTO teacher

(second\_name, first\_name, patronymic)

VALUES ('Иванов', 'Иван', 'Иванович'),

('Забудь', 'Светлана', 'Ивановна'),

('Макаревич', 'Андрей', 'Игоревич'),

('Семашко', 'Зинаида', 'Климовна'), -- free

('Попова', 'Ульяна', 'Владиславовна'),

('Вечёрко', 'Анастасия', 'Дмитриевна'),

('Ожог', 'Николай', 'Витальевич');

INSERT INTO teacher\_cabinet

(teacher\_id, cabinet\_number)

VALUES (1, 11),

(2, 12),

(3, 14),

(5, 15),

(6, 16),

(7, 17);

INSERT INTO class

(class\_name)

VALUES ('4В'),

('7А'),

('7Б'),

('9Б'),

('11А');

INSERT INTO student

(class\_name, second\_name, first\_name, patronymic)

VALUES ('4В', 'Степанов', 'Василий', 'Анатольевич'),

('4В', 'Авдеев', 'Георгий', 'Фёдорович'),

('7А', 'Виноградов', 'Анатолий', 'Алексеевич'),

('7А', 'Кузнецов', 'Илья', 'Михайлович'),

('7Б', 'Смирнов', 'Николай', 'Денисовчи'),

('7Б', 'Назарова', 'Мария', 'Михайловна'),

('9Б', 'Михеева', 'Елизавета', 'Алексеевна'),

('9Б', 'Ковалева', 'Анастасия', 'Сегреевна'),

('11А', 'Анисимов', 'Ярослав', 'Юрьевич'),

('11А', 'Нечаева', 'Светлана', 'Генадьевна');

INSERT INTO subject

(subject\_name)

VALUES ('Математика'),

('Физика'),

('Русский язык'),

('География'),

('Химия'),

('Биология'),

('Физкультура');

INSERT INTO grade

(student\_id, subject\_name)

SELECT grade\_temp.student\_id, grade\_temp.subject\_name

FROM (student

CROSS JOIN subject) AS grade\_temp;

UPDATE grade

SET value = random\_between(2, 10)

WHERE value IS NULL;

INSERT INTO school\_schedule

(order\_number, day, class\_name, subject\_name, teacher\_id, cabinet\_number)

VALUES (1, 'Понедельник', '4В', 'Математика', 1, 17),

(1, 'Вторник', '4В', 'Физика', 2, 12),

(1, 'Среда', '4В', 'Математика', 1, 17),

(2, 'Среда', '4В', 'Физика', 1, 12),

(3, 'Среда', '4В', 'Физкультура', 5, 16),

(4, 'Среда', '4В', 'География', 3, 14),

(5, 'Среда', '4В', 'Химия', 4, 12),

(1, 'Четверг', '4В', 'Русский язык', 7, 11),

(1, 'Пятница', '4В', 'Математика', 1, 17),

(1, 'Среда', '9Б', 'Математика', 1, 13);

-- Какой предмет будет в заданном классе, в заданный день недели на заданном уроке?

SELECT subject\_name AS subject

FROM school\_schedule

WHERE class\_name = '4В'

AND day = 'Среда'

AND order\_number = 2;

-- Кто из учителей преподает в заданном классе?

SELECT DISTINCT concat\_ws(' ', second\_name, first\_name, patronymic) AS teacher

FROM teacher

INNER JOIN school\_schedule ON teacher.teacher\_id = school\_schedule.teacher\_id

WHERE class\_name = '4В';

-- В каком кабинете будет 5-й урок в среду у некоторого класса?

SELECT cabinet\_number AS cabinet

FROM school\_schedule

WHERE order\_number = 5

AND day = 'Среда'

AND class\_name = '4В';

-- В каких классах преподает заданный предмет заданный учитель?

SELECT DISTINCT class\_name

FROM school\_schedule

INNER JOIN teacher ON teacher.teacher\_id = school\_schedule.teacher\_id

WHERE subject\_name = 'Математика'

AND (second\_name = 'Иванов'

AND first\_name = 'Иван'

AND patronymic = 'Иванович');

-- Расписание на заданный день недели для указанного класса?

SELECT order\_number, subject\_name AS subject, concat\_ws(' ', second\_name, first\_name, patronymic) AS teacher

FROM school\_schedule

INNER JOIN teacher ON teacher.teacher\_id = school\_schedule.teacher\_id

WHERE day = 'Среда'

AND class\_name = '4В'

ORDER BY order\_number;

-- Сколько учеников в указанном классе?

SELECT COUNT(\*) AS count\_of\_students

FROM student

WHERE class\_name = '4В';