Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Инженерно-экономический факультет

Кафедра экономической информатики

Отчёт

по лабораторной работе №8

по системам и технологиям интеллектуальной обработки данных

Студент А.В. Свич

Минск 2020

**Задание**: используя СУБД MongoDB, разработайте базу данных, предназначенную  для хранения логов веб-сервера. Лог включает в себя следующие поля: адрес ресурса (URL), IP-адрес пользовательского компьютера, отметка времени начала просмотра ресурса, длительность просмотра ресурса.

**Указания к заданию**

1.     Разработайте консольную утилиту для преобразования лога веб-сервера в формате CSV (Comma Separated Values), в формат JSON. Лог должен содержать поля со следующими названиями: URL, IP, timeStamp, timeSpent.

2.     Разработайте запросы для загрузки полученных данных в формате JSON в СУБД MongoDB.

[**Разработка запросов в СУБД MongoDB**](file:///F:\%D0%A3%D1%80%D0%BE%D0%BA%D0%B8\sitiod\sitiod\lb8\lab8.htm#page6)

 1.     Разработайте следующие запросы, используя встроенные в СУБД MongoDB средствами выборки:

 1)    Выдать упорядоченный список URL ресурсов.

2)    Выдать упорядоченный список IP-адресов пользователей, посетивших ресурс с заданным URL.

 3)    Выдать упорядоченный список URL ресурсов, посещенных в заданный временной период.

 4)    Выдать упорядоченный список URL ресурсов, посещенных пользовате-лем с заданным IP-адресом.

 2.     Разработайте следующие запросы, используя встроенные в СУБД MongoDB средства программирования на основе парадигмы MapReduce:

 1)    Выдать список URL ресурсов с указанием суммарной длительности по-сещения каждого ресурса, упорядоченный по убыванию.

 2)    Выдать список URL ресурсов с указанием суммарного количества посе-щений каждого ресурса, упорядоченный по убыванию.

 3)    Выдать список URL ресурсов с указанием количества посещений каждого ресурса в день за заданный период, упорядоченный URL ресурса и убы-ванию количества посещений.

 4)    Выдать список IP-адресов c указанием суммарного количества и суммар-ной длительности посещений ресурсов, упорядоченный по адресу, убы-ванию количества и убыванию длительности.

Для данной лабораторной работы я использовала данный скрипт, написанный на javascript:

db = connect("localhost:27017/lab8");

function show(query) {

cursor = query;

if (Array.isArray(cursor)) {

printjson(cursor);

} else {

while (cursor.hasNext()) {

printjson(cursor.next());

}

}

}

print("1) Выдать упорядоченный список URL ресурсов.");

show(db.logs.distinct("URL"))

print("2) Выдать упорядоченный список IP-адресов пользователей, посетивших ресурс с заданным URL (https://ru.wikipedia.org/wiki/IP-адрес).");

show(db.logs.find({ "URL": "https://ru.wikipedia.org/wiki/IP-адрес" }, { IP: 1 }))

print("3) Выдать упорядоченный список URL ресурсов, посещенных в заданный временной период (2020-03-14T11:33:15.992Z - 2020-03-14T11:45:15.992Z).");

show(db.logs.distinct("URL", { timeStamp: { $gt: ISODate("2020-03-14T11:33:15.992Z"), $lt: ISODate("2020-03-14T11:45:15.992Z") } }))

print("4) Выдать упорядоченный список URL ресурсов, посещенных пользовате-лем с заданным IP-адресом (192.168.0.9).");

show(db.logs.find({ "IP": "192.168.0.9" }, { URL: 1 }))

print("5) Выдать список URL ресурсов с указанием суммарной длительности посещения каждого ресурса, упорядоченный по убыванию.");

var emit1 = function () { emit(this.URL, this.timeSpent); }

var red1 = function (key, values) { return Array.sum(values) }

db.logs.mapReduce(emit1, red1, { out: "map" });

show(db.map.find())

print("6) Выдать список URL ресурсов с указанием суммарного количества посещений каждого ресурса, упорядоченный по убыванию.");

var emit2 = function () { emit(this.URL, 1); }

var red2 = function (key, values) { return Array.sum(values) }

db.logs.mapReduce(emit2, red2, { out: "map" });

show(db.map.find())

print("7) Выдать список URL ресурсов с указанием количества посещений каждого ресурса в день за заданный период, упорядоченный URL ресурса и убыванию количества посещений.")

var emit3 = function () { emit(this.URL, 1); }

var red3 = function (key, values) { return Array.sum(values) }

db.logs.mapReduce(emit3, red3, { query: { timeStamp: { $gt: ISODate("2020-03-14T11:33:15.992Z"), $lt: ISODate("2020-03-14T11:45:15.992Z") } }, out: "map" });

show(db.map.find().sort({ "pos": -1 }))

print("8) Выдать список IP-адресов c указанием суммарного количества и суммар-ной длительности посещений ресурсов, упорядоченный по адресу, убы-ванию количества и убыванию длительности.");

var emit4 = function () { emit(this.URL, { connections: 1, timeSpent: this.timeSpent }); }

var red4 = function (key, values) {

connections\_val = 0;

time\_val = 0;

for (i = 0; i < values.length; i++) {

connections\_val += values[i].connections;

time\_val += values[i].timeSpent;

}

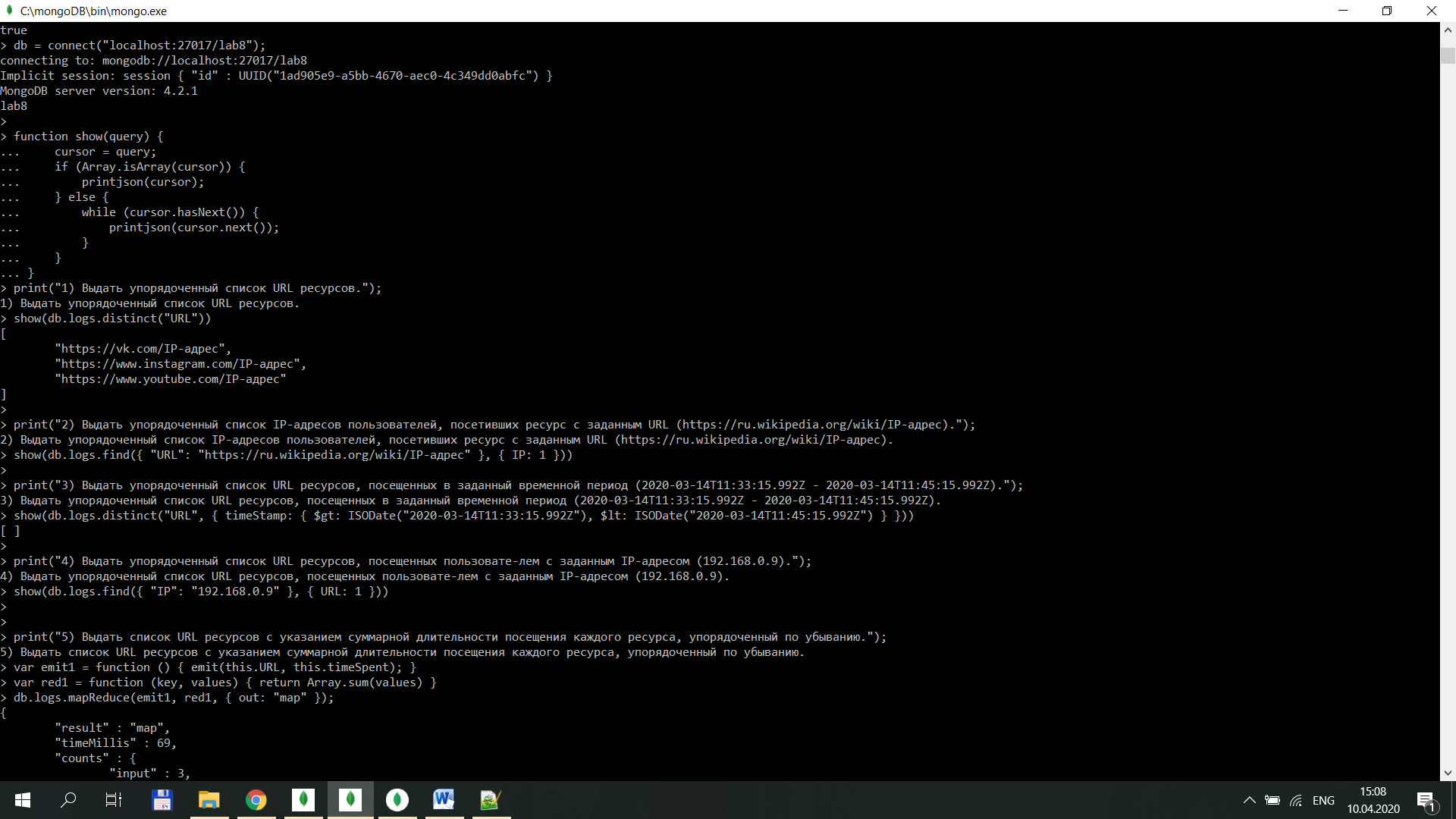
return { connections: connections\_val, time: time\_val };

}

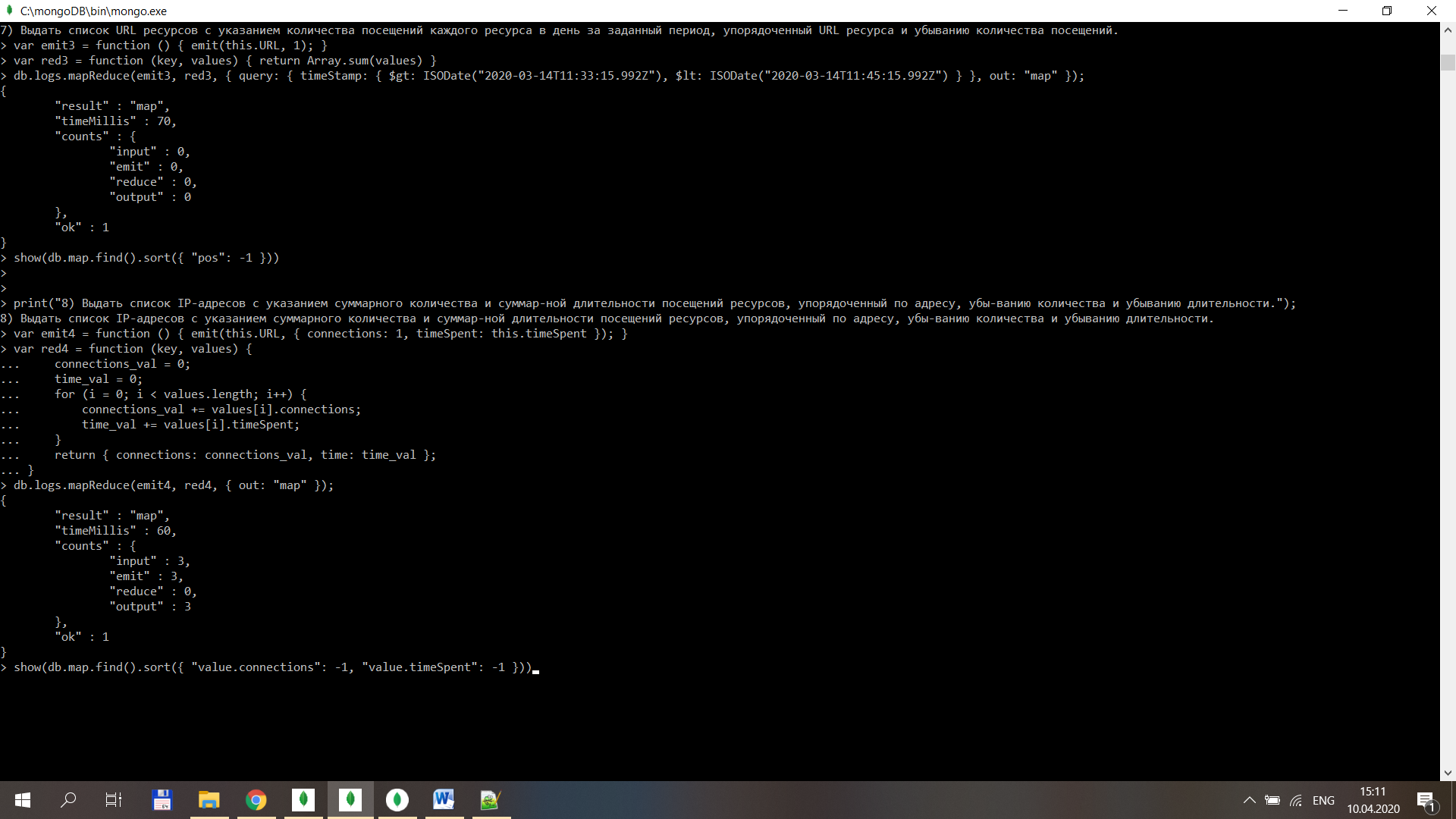
db.logs.mapReduce(emit4, red4, { out: "map" });

show(db.map.find().sort({ "value.connections": -1, "value.timeSpent": -1 }))

Результатом работы программы будет:







Если открыть MongoDB Compass:

