CS 460 Programming Assignment 3 Vivek Kotecha (vbk@bu.edu)

Question 1 NEO4j MBTA

1.

MATCH (a)-[r]->(b) RETURN b.name,COUNT(a) ORDER BY(COUNT(a)) DESC LIMIT 10;

```
neo4j-sh (?)$ MATCH (a)-[r]->(b) RETURN b.name,COUNT(a) ORDER BY(COUNT(a)) DESC LIMIT 10
+-----------------------------------------------------------+
| b.name                                          | COUNT(a) |
+-----------------------------------------------------------+
| "South Station - 700 Atlantic Ave."             | 5470     |
| "Boston Public Library - 700 Boylston St."      | 5069     |
| "Boylston St. at Arlington St."                 | 4376     |
| "TD Garden - Legends Way"                       | 3511     |
| "Newbury St / Hereford St"                      | 3327     |
| "Kenmore Sq / Comm Ave"                         | 2892     |
| "Seaport Square - Seaport Blvd. at Boston Wharf" | 2657     |
| "Back Bay / South End Station"                  | 2618     |
| "Beacon St / Mass Ave"                          | 2563     |
| "Prudential Center / Belvidere"                 | 2458     |
+-----------------------------------------------------------+
10 rows
```

2.

MATCH (a)<-[r]-(b) RETURN b.name,COUNT(a) ORDER BY(COUNT(a)) DESC LIMIT 10;

```
neo4j-sh (?)$ MATCH (a)<-[r]-(b) RETURN b.name,COUNT(a) ORDER BY(COUNT(a)) DESC LIMIT 10;
+-----------------------------------------------------------+
| b.name                                          | COUNT(a) |
+-----------------------------------------------------------+
| "South Station - 700 Atlantic Ave."             | 5431     |
| "Boston Public Library - 700 Boylston St."      | 5060     |
| "Boylston St. at Arlington St."                 | 4184     |
| "TD Garden - Legends Way"                       | 3805     |
| "Newbury St / Hereford St"                      | 3169     |
| "Back Bay / South End Station"                  | 2935     |
| "Kenmore Sq / Comm Ave"                         | 2749     |
| "Seaport Square - Seaport Blvd. at Boston Wharf" | 2615     |
| "Mayor Thomas M. Menino - Government Center"    | 2586     |
| "Beacon St / Mass Ave"                          | 2530     |
+-----------------------------------------------------------+
10 rows
```

3.
MATCH (a)-[r]-(b) RETURN a.name,b.name,COUNT(a) ORDER BY(COUNT(a)) DESC LIMIT 20;

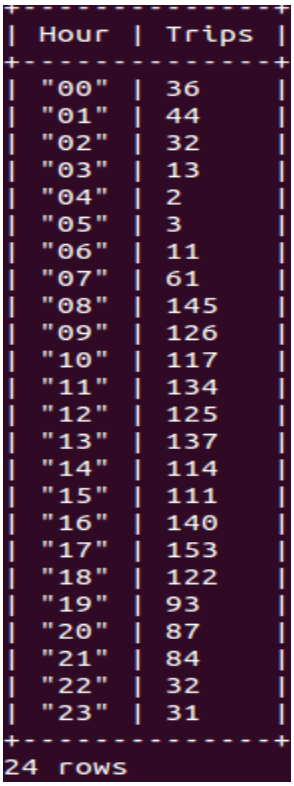
```
neo4j-sh (?)$ MATCH (a)-[r]-(b) RETURN a.name,b.name,COUNT(a) ORDER BY(COUNT(a)) DESC LIMIT 20;
+-------------------------------------------------------------------------------------------------------------+
| a.name                                          | b.name                                          | COUNT(a) |
+-------------------------------------------------------------------------------------------------------------+
| "Lewis Wharf - Atlantic Ave."                   | "South Station - 700 Atlantic Ave."             | 1127     |
| "South Station - 700 Atlantic Ave."             | "Lewis Wharf - Atlantic Ave."                   | 1127     |
| "South Station - 700 Atlantic Ave."             | "TD Garden - Legends Way"                       | 1053     |
| "TD Garden - Legends Way"                       | "South Station - 700 Atlantic Ave."             | 1053     |
| "Agganis Arena - 925 Comm Ave."                 | "B.U. Central - 725 Comm. Ave."                 | 681      |
| "B.U. Central - 725 Comm. Ave."                 | "Agganis Arena - 925 Comm Ave."                 | 681      |
| "South Station - 700 Atlantic Ave."             | "Rowes Wharf - Atlantic Ave"                    | 664      |
| "Rowes Wharf - Atlantic Ave"                    | "South Station - 700 Atlantic Ave."             | 664      |
| "TD Garden - Legends Way"                       | "Seaport Square - Seaport Blvd. at Boston Wharf" | 632      |
| "Seaport Square - Seaport Blvd. at Boston Wharf" | "TD Garden - Legends Way"                       | 632      |
| "South Station - 700 Atlantic Ave."             | "Boylston St / Berkeley St"                     | 587      |
| "Boylston St / Berkeley St"                     | "South Station - 700 Atlantic Ave."             | 587      |
| "South Station - 700 Atlantic Ave."             | "Boston Public Library - 700 Boylston St."      | 568      |
| "Boston Public Library - 700 Boylston St."      | "South Station - 700 Atlantic Ave."             | 568      |
| "Boylston St. at Arlington St."                 | "South Station - 700 Atlantic Ave."             | 561      |
| "South Station - 700 Atlantic Ave."             | "Boylston St. at Arlington St."                 | 561      |
| "Kenmore Sq / Comm Ave"                         | "Agganis Arena - 925 Comm Ave."                 | 548      |
| "Agganis Arena - 925 Comm Ave."                 | "Kenmore Sq / Comm Ave"                         | 548      |
| "Boston Public Library - 700 Boylston St."      | "Boylston St. at Arlington St."                 | 544      |
| "Boylston St. at Arlington St."                 | "Boston Public Library - 700 Boylston St."      | 544      |
+-------------------------------------------------------------------------------------------------------------+
20 rows
```

4.
MATCH (a)-[r]->(b) WHERE a.name="B.U. Central - 725 Comm. Ave." RETURN substring(split(r.startT," ")[1],0,2) AS Hour, count(substring(split(r.startT," ")[1],0,2)) AS Trips ORDER BY(substring(split(r.startT," ")[1],0,2)) ASC ;

Hour	Trips
"00"	31
"01"	15
"02"	29
"03"	6
"04"	3
"05"	4
"06"	11
"07"	22
"08"	34
"09"	32
"10"	61
"11"	88
"12"	150
"13"	153
"14"	157
"15"	164
"16"	240
"17"	208
"18"	152
"19"	104
"20"	86
"21"	79
"22"	43
"23"	62

24 rows

5.
MATCH (a)<-[r]-(b) WHERE a.name="B.U. Central - 725 Comm. Ave." RETURN substring(split(r.startT," ")[1],0,2) AS Hour, count(substring(split(r.startT," ")[1],0,2)) AS Trips ORDER BY(substring(split(r.startT," ")[1],0,2)) ASC;



Hour	Trips
"00"	36
"01"	44
"02"	32
"03"	13
"04"	2
"05"	3
"06"	11
"07"	61
"08"	145
"09"	126
"10"	117
"11"	134
"12"	125
"13"	137
"14"	114
"15"	111
"16"	140
"17"	153
"18"	122
"19"	93
"20"	87
"21"	84
"22"	32
"23"	31

24 rows

Question 2 NEO4j FLIGHTS and AIRPORTS

a.
Year of flights: 2008
Sample size: 10,000

b.
I downloaded the original dataset at extracted a sample size of 10,000 from it.

head -10000 2008.csv >> 2008_sample.csv

Then, in 2008_sample.csv I manually deleted the columns except DayOfWeek, DepTime, ArrTime,FlightNum,Origin, Destination,Cancelled and Diverted using LibreOffice.

In airports.csv I manually deleted all the columns except IATA and airport. And then I deleted the airports that had a matching 0 ID on it.

Then in terminal I ran the two commands to clean the double quotation marks (“”) from both the CSV files.

tr "\"" "'" <airports.csv > clean.csv

tr "\"" "'" <2008_sample.csv > clean2.csv

c.
HEADERS:
In 2008_sample.csv:
I added the **:START_ID** to Origin column and **:END_ID** to Destination column.

Then I added a new column with header **:TYPE** to describe the relationship type and changed all values in the column to **Closed**.

Then I modified the Cancellation column and added the **:LABEL** header to it.

Then added the header **iata:ID** in airports.csv file.

d.

Then I copied **clean.csv** and **clean2.csv** to the **bin** folder inside the **neo4j** folder.
Than I executed the command:
./neo4j-import --into data/graph.db --nodes clean.csv --relationships clean2.csv

```
IMPORT DONE in 2s 660ms. Imported:
  3374 nodes
  9994 relationships
  56718 properties
vivek@VODAFONE:~/Desktop/neo4j-community-2.3.1/bin$
```

e.

1.
MATCH (a)-[r]->(b) RETURN b.airport,COUNT(a) ORDER BY(COUNT(a)) DESC LIMIT 10;

```
neo4j-sh (?)$ MATCH (a)-[r]->(b) RETURN b.airport,COUNT(a) ORDER BY(COUNT(a)) DESC LIMIT 10;
+-----------------------------------------------+
| b.airport                          | COUNT(a) |
+-----------------------------------------------+
| "McCarran International"           | 703      |
| "Chicago Midway"                   | 661      |
| "Phoenix Sky Harbor International" | 595      |
| "Baltimore-Washington International" | 498    |
| "Metropolitan Oakland International" | 435    |
| "William P Hobby"                  | 410      |
| "Dallas Love "                     | 361      |
| "Los Angeles International"        | 357      |
| "Orlando International"            | 353      |
| "San Diego International-Lindbergh " | 304    |
+-----------------------------------------------+
10 rows
```

2.

MATCH (a)<-[r]-(b) RETURN b.airport,COUNT(a) ORDER BY(COUNT(a)) DESC LIMIT 10;

```
neo4j-sh (?)$ MATCH (a)<-[r]-(b) RETURN b.airport,COUNT(a) ORDER BY(COUNT(a)) DESC LIMIT 10;
+-------------------------------------------------+
| b.airport                          | COUNT(a) |
+-------------------------------------------------+
| "McCarran International"           | 921      |
| "Chicago Midway"                   | 638      |
| "Phoenix Sky Harbor International" | 560      |
| "Baltimore-Washington International" | 474    |
| "Los Angeles International"        | 425      |
| "Metropolitan Oakland International" | 404    |
| "William P Hobby"                  | 393      |
| "Dallas Love "                     | 354      |
| "Orlando International"            | 345      |
| "San Diego International-Lindbergh " | 291    |
+-------------------------------------------------+
10 rows
```

3.

MATCH (a)-[r]-(b) WHERE r.DayOfWeek<="5" RETURN a.iata,b.iata,COUNT(a) ORDER BY(COUNT(a)) DESC LIMIT 10;

```
neo4j-sh (?)$ MATCH (a)-[r]-(b) WHERE r.DayOfWeek<="5" RETURN a.iata,b.iata,COUNT(a) ORDER BY(COUNT(a)) DESC LIMIT 10;
+---------------------------+
| a.iata | b.iata | COUNT(a) |
+---------------------------+
| "OAK"  | "LAX"  | 84       |
| "LAX"  | "OAK"  | 84       |
| "PHX"  | "LAS"  | 72       |
| "SAN"  | "OAK"  | 72       |
| "OAK"  | "SAN"  | 72       |
| "LAS"  | "PHX"  | 72       |
| "HOU"  | "DAL"  | 60       |
| "DAL"  | "HOU"  | 60       |
| "LAS"  | "LAX"  | 56       |
| "LAX"  | "SJC"  | 56       |
+---------------------------+
10 rows
```

4.

MATCH (a)-[r]-(b) WHERE r.DayOfWeek>"5" RETURN a.iata,b.iata,COUNT(a) ORDER BY(COUNT(a)) DESC LIMIT 10;

```
neo4j-sh (?)$ MATCH (a)-[r]-(b) WHERE r.DayOfWeek>"5" RETURN a.iata,b.iata,COUNT(a) ORDER BY(COUNT(a)) DESC LIMIT 10;
+---------------------------+
| a.iata | b.iata | COUNT(a) |
+---------------------------+
| "HOU"  | "DAL"  | 87       |
| "DAL"  | "HOU"  | 87       |
| "LAS"  | "LAX"  | 51       |
| "LAX"  | "OAK"  | 51       |
| "OAK"  | "LAX"  | 51       |
| "LAX"  | "LAS"  | 51       |
| "LAS"  | "BUR"  | 48       |
| "BUR"  | "LAS"  | 48       |
| "PHX"  | "LAS"  | 47       |
| "LAS"  | "PHX"  | 47       |
+---------------------------+
10 rows
```

5.
I)

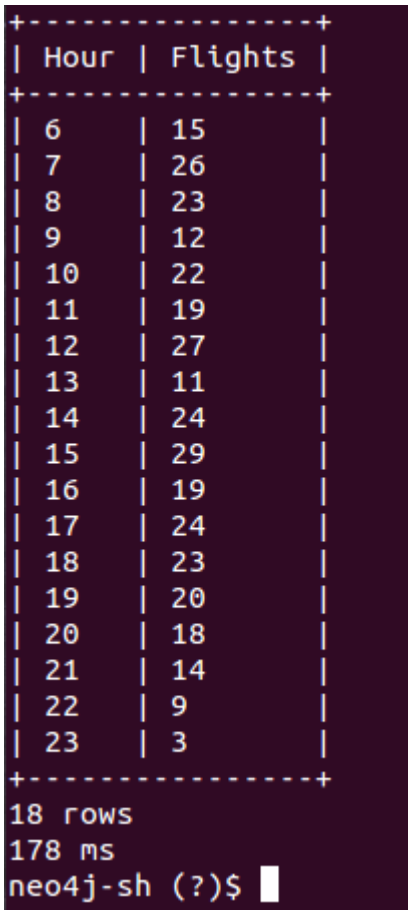
MATCH (a)-[r]->(b) WHERE a.iata='LAX' AND r.DepTime<>"NA" WITH r.DepTime as time, (CASE LENGTH(r.DepTime) WHEN 3 THEN 2 ELSE 3 END) AS len RETURN toInt(substring(time,0,len-1)) AS Hour ,count(substring(time,0,len-1)) As Flights ORDER BY(Hour) ASC;

```
neo4j-sh (?)$ MATCH (a)-[r
) AS len RETURN toInt(subs
+-----------------+
| Hour | Flights  |
+-----------------+
| 5    | 2        |
| 6    | 19       |
| 7    | 16       |
| 8    | 25       |
| 9    | 26       |
| 10   | 26       |
| 11   | 18       |
| 12   | 16       |
| 13   | 31       |
| 14   | 29       |
| 15   | 26       |
| 16   | 22       |
| 17   | 33       |
| 18   | 29       |
| 19   | 27       |
| 20   | 24       |
| 21   | 20       |
| 22   | 14       |
| 23   | 4        |
+-----------------+
19 rows
230 ms
neo4j-sh (?)$
```

Note: There are no flights departing Los Angeles airport(LAX) between midnight and 5 am.

II)

MATCH (a)<-[r]-(b) WHERE a.iata='LAX' AND r.DepTime<>"NA" WITH r.DepTime as time, (CASE LENGTH(r.DepTime) WHEN 3 THEN 2 ELSE 3 END) AS len RETURN toInt(substring(time,0,len-1)) AS Hour ,count(substring(time,0,len-1)) As Flights ORDER BY(Hour) ASC;



Hour	Flights
6	15
7	26
8	23
9	12
10	22
11	19
12	27
13	11
14	24
15	29
16	19
17	24
18	23
19	20
20	18
21	14
22	9
23	3

18 rows
178 ms
neo4j-sh (?)$

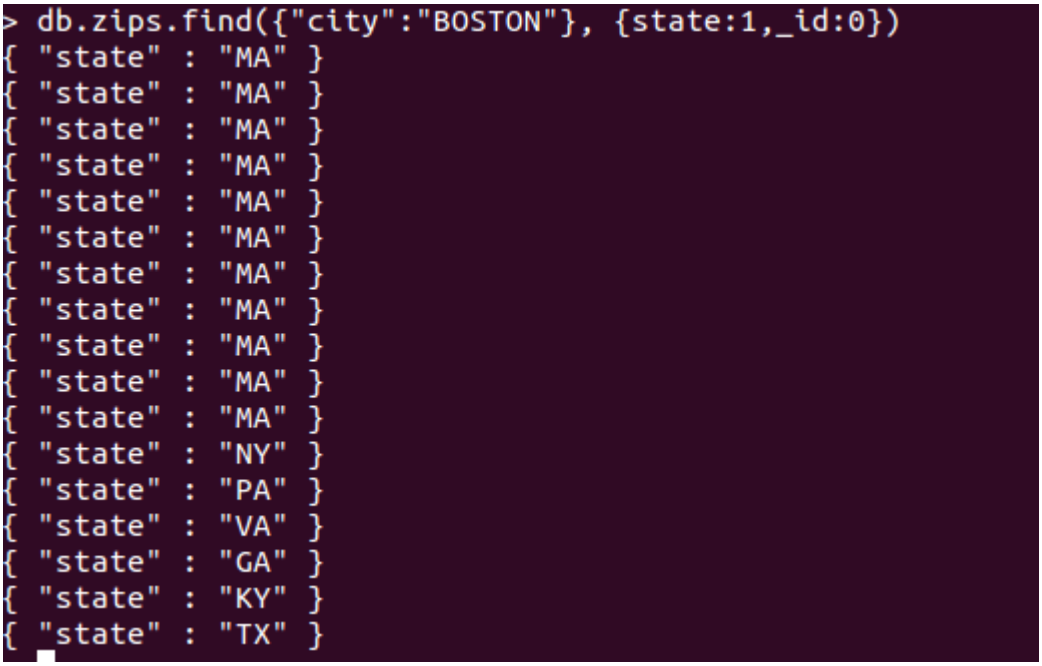
Note: There are no flights arriving at Los Angeles airport(LAX) between midnight and 6 am.

Question 3 MONGO DB.

Created database using the command:
mongoimport --host localhost --db test -c zips --type json --file zips.json –headerline

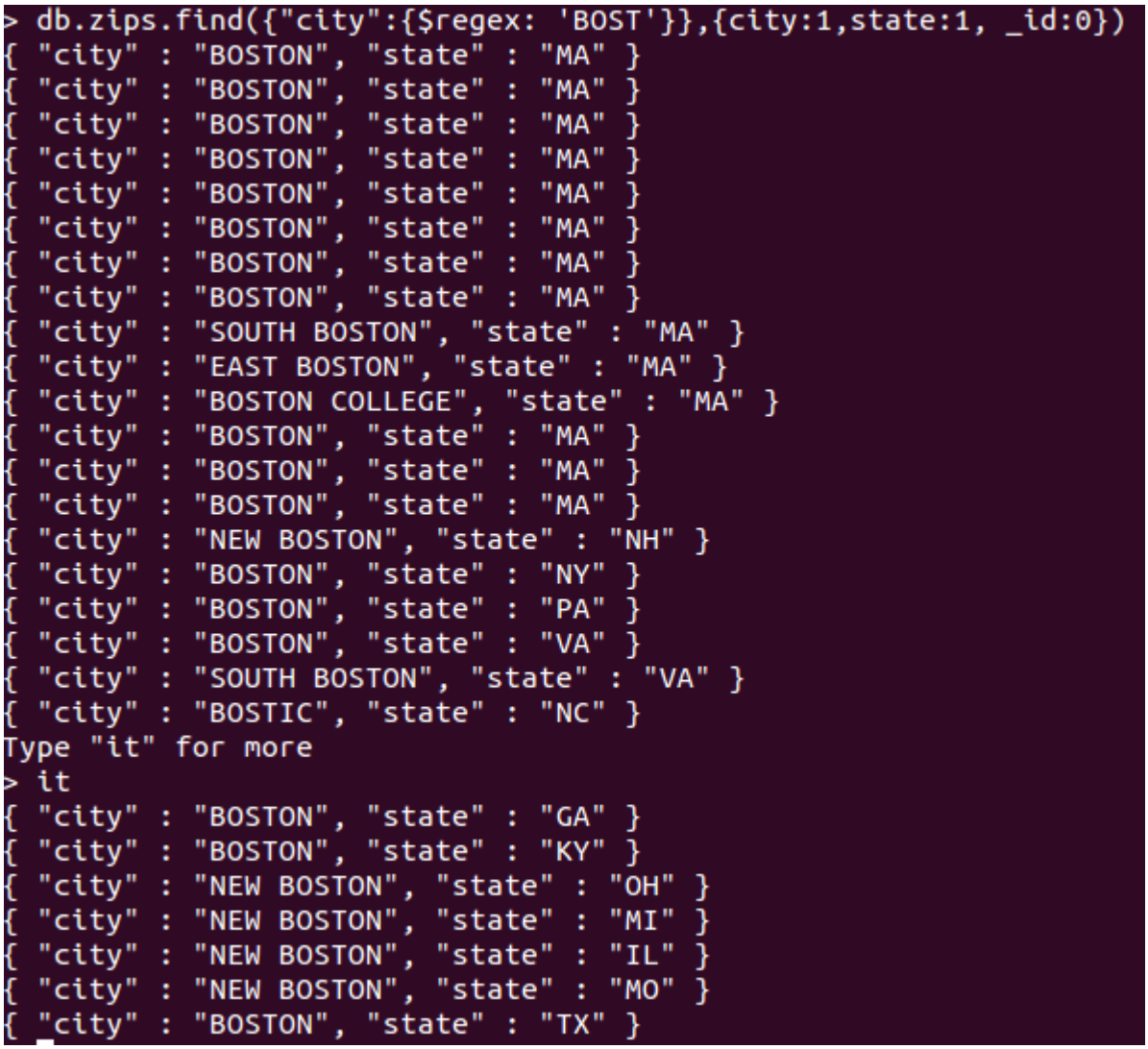
1.

db.zips.find({"city":"BOSTON"}, {state:1,_id:0})



2.

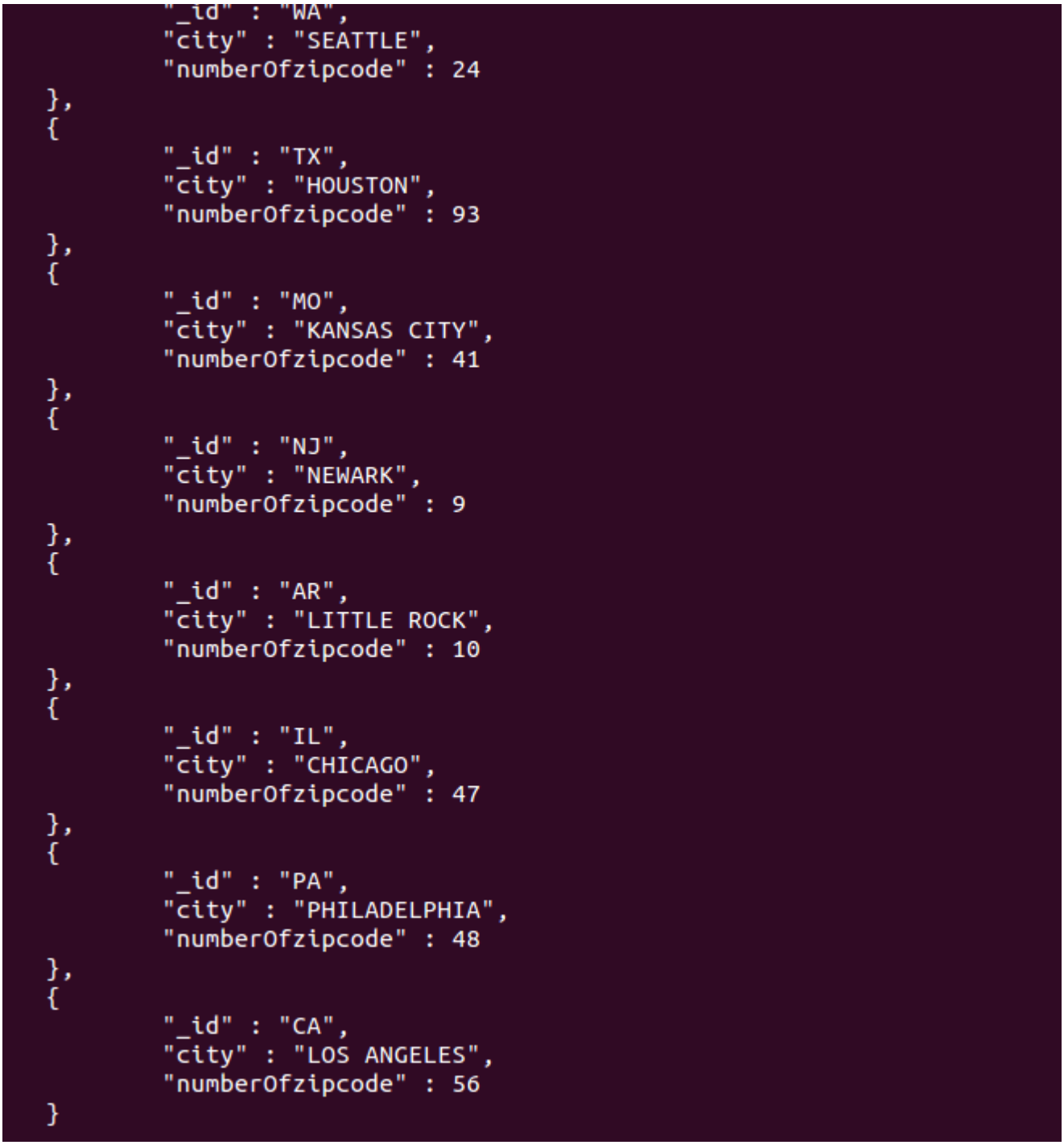
db.zips.find({"city":{$regex: 'BOST'}},{city:1,state:1, _id:0})



Source: https://stackoverflow.com/questions/10242501/how-to-find-a-substring-in-a-field-in-mongodb

3.

db.zips.aggregate([{$sort:{pop:1}}, {$group:{_id:{city:'$city', state:'$state'}, numberOfzipcodes: {$sum:1}}}, {$sort:{numberOfzipcodes:-1}}, {$group:{_id:'$_id.state', city: {$first:'$_id.city'}, numberOfzipcode:{$max:'$numberOfzipcodes'}}},]).result



Source: https://stackoverflow.com/questions/16368638/mongodb-distinct-aggregation

4.

db.zips.aggregate({$match:{loc: { $geoWithin: { $centerSphere: [[-72, 42], 2] } }}}, { $group: {_id:'$state',cities:{$sum:1}, population:{$sum:'$pop'}}},{$sort:{cities:1}}).result


```
{
        "_id" : "MO",
        "cities" : 994,
        "population" : 5110648
},
{
        "_id" : "OH",
        "cities" : 1007,
        "population" : 10846517
},
{
        "_id" : "IL",
        "cities" : 1237,
        "population" : 11427576
},
{
        "_id" : "PA",
        "cities" : 1458,
        "population" : 11881643
},
{
        "_id" : "CA",
        "cities" : 1516,
        "population" : 29754890
},
{
        "_id" : "NY",
        "cities" : 1595,
        "population" : 17990402
},
{
        "_id" : "TX",
        "cities" : 1671,
        "population" : 16984601
}
```


Source: https://docs.mongodb.org/manual/reference/operator/query/centerSphere/

5.

db.zips.aggregate({$sort:{pop:1}}, {$match:{$and:[{"loc.1":{$gt:30,$lt:40},"loc.0":{$gt:-90,$lt:-80}}] }}, {$group:{_id:'$city',population:{$sum:'$pop'}}},{$limit:10}).result


```
[
        {
                "_id" : "CHILLICOTHE",
                "population" : 54615
        },
        {
                "_id" : "WESTERN HILLS",
                "population" : 48302
        },
        {
                "_id" : "HATTON",
                "population" : 46563
        },
        {
                "_id" : "BEECH ISLAND",
                "population" : 45886
        },
        {
                "_id" : "BURTON",
                "population" : 43849
        },
        {
                "_id" : "HINESVILLE",
                "population" : 42962
        },
        {
                "_id" : "MARTINEZ",
                "population" : 42573
        },
```