Akash Bharti
B00837970

CSCI 5408
Data Management, Warehousing and Analytics
Assignment 2

Section A:

The dataset [1] provides information about the reservation of camping sites in the Nova Scotia Provincial Parks for 2016. The information has been collected through the reservation system for the general public so that they can reserve camping sites in the provincial parks. The dataset contains booking information for about 20 parks. The dataset has the following columns: ParkName, State, Country, Adult, Child, partySize, RateType, BookingType, Equipment, BookingStartDate, BookingEndDate, Night and Permits. The dataset comes under Nova Scotia Open Government License and it owned by Open Data Nova Scotia. The dataset contains around 35000 entries out of which around 22000 are for booking made in Nova Scotia. Each row in the dataset contains all the information about the booking. For example, partySize depicts the number of people for which the booking has been made. It also contains columns *Adult* and *Child* which informs about the partySize splits into number of adults and children. The RateType has four types: Full, Senior, Host and Veteran. There are three types of BookingType: Backcountry Permit, Campsite Permit and Yurt Permit.

Section B:

I have used the following python scripts to clean and transform the data:

1. ***Extract_and_transform.py***- this script performs the following 3 tasks:
    a. Extracts the data on parks of Canada from file *DNR_Camping_Parks_Reservation_Data_2016.csv* and creates a new csv file *file1.csv*
    b. Extracts data on ParkName, State, PartySize, BookingType, RateType, Equipment from *file1.csv* and write it into *file2.csv*.
    c. Reads data from *file2.csv*, replaces all “less than” with “LT” and “Single tent” with “ST” and writes it to *file3.csv*

2. ***Reduce.py -*** read the data from *file3.csv* and writes, for each park, the booking with the highest partySize into file *file5.csv.* Now, file5.csv will contain only 20 entries.

Section C:

I install Neo4j on my Ubuntu 18.04.4 LTS System. Figure 1 shows the installed neo4j console on the system. I have named the database “Parks”.

Akash Bharti
B00837970

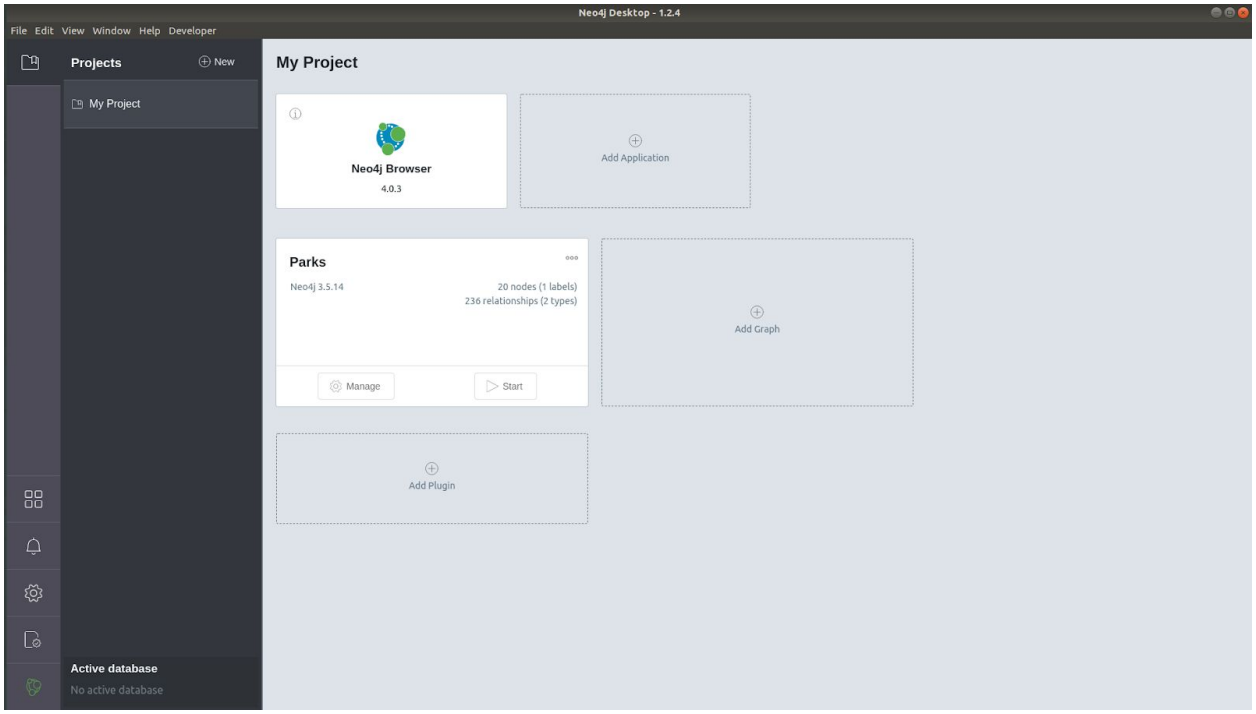

Figure 1: Neo4j console

Figure 2 shows all the nodes in the graph database, with each node representing a park.

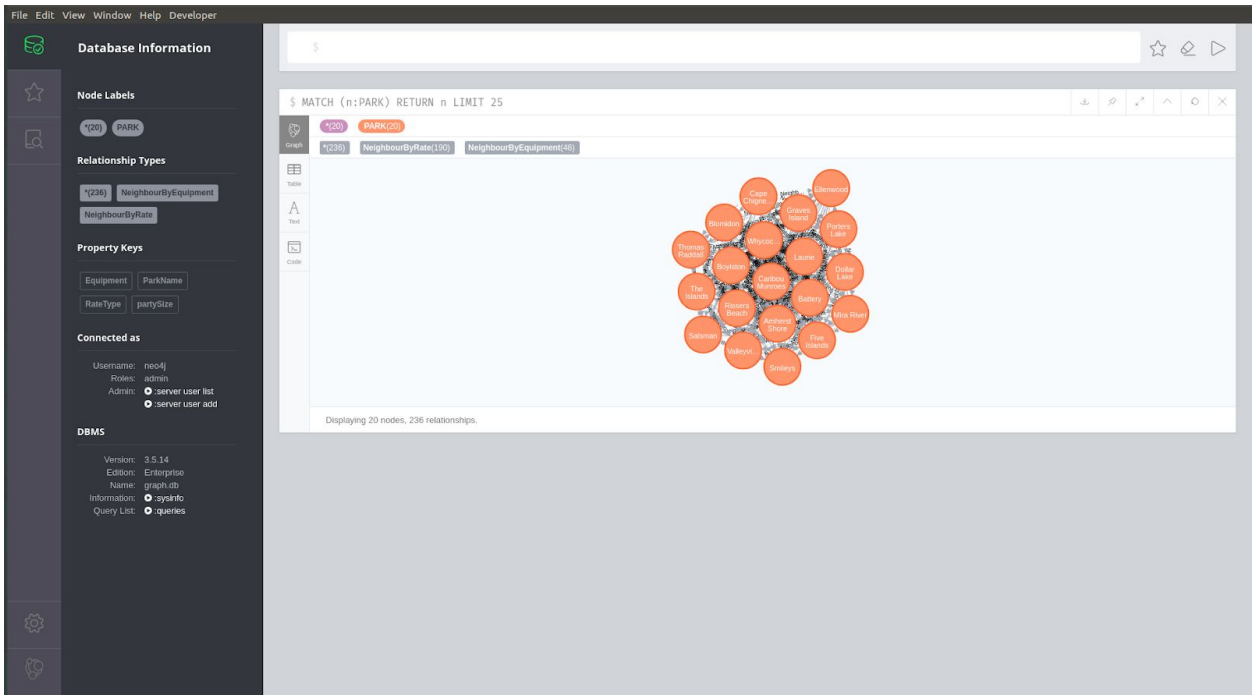

Figure 2: Graph showing all the parks in NS

Akash Bharti
B00837970

Section D:

I have used *file5.csv* created by the script *Reduce.py* to create the graph database. Figure 3 shows 10 nodes in the database. Each node represents a park. I have used Cypher to limit the nodes shown in figure 3 to 10 for illustrating purpose. I used the following Cypher query [2][3] to import data from the file and create nodes in the database.

```
LOAD CSV FROM 'file:///file5.csv' AS line
CREATE (:PARK { ParkName: line[0], PartySize:  toInteger(line[2]), RateType: line[3], Equipment: line[4]})
```

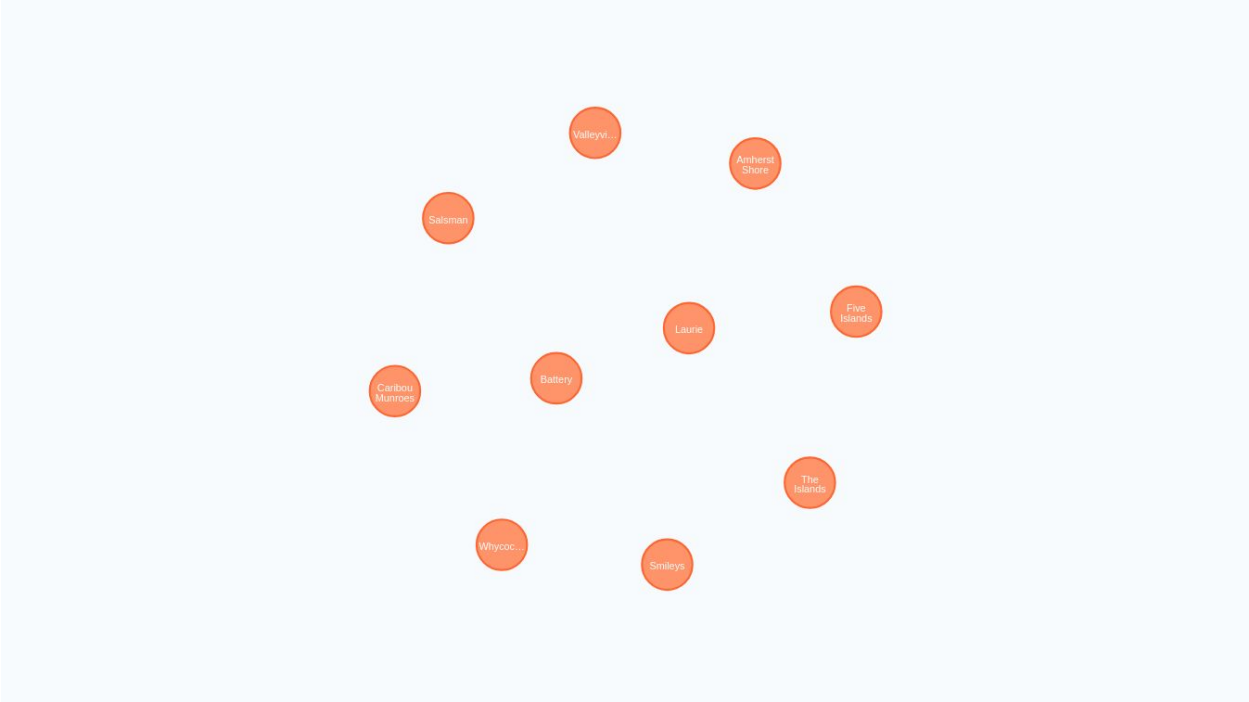

Figure 3: Nodes in the database (limit 10)

Figure 4 shows nodes connected by relationship NeighbourByEquipment. I used the following query [4] to create the relationship

```
MATCH (a:PARK),(b:PARK)
WHERE a.Equipment=b.Equipment AND ID(a)<>ID(b) AND NOT
(b:PARK)-[:NeighbourByEquipment]->(a:PARK)
CREATE (a)-[r:NeighbourByEquipment]->(b)
RETURN type(r)
```

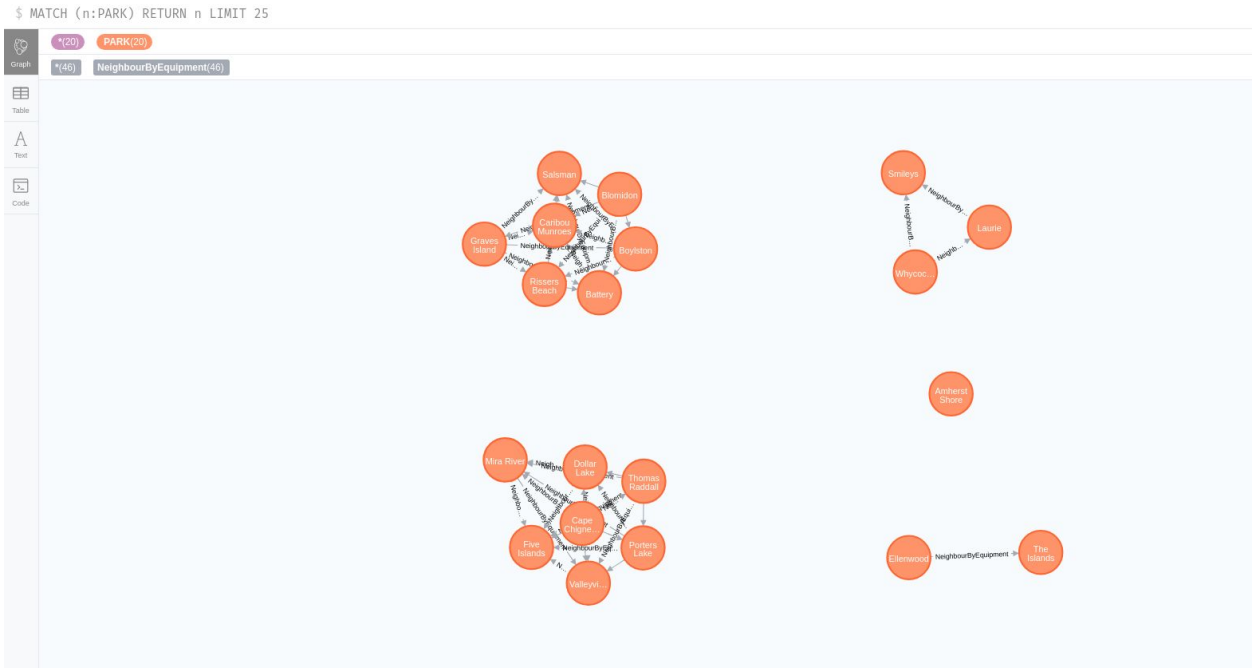

Figure 4: Nodes connected by NeighbourByEquipment relationship

I used the following query to create the relationship "NeighbourByRate"

```
MATCH (a:PARK),(b:PARK)
WHERE a.RateType=b.RateType AND ID(a)<>ID(b) AND NOT
(b:PARK)-[:NeighbourByRate]->(a:PARK)
CREATE (a)-[r:NeighbourByRate]->(b)
RETURN type(r)
```

Figure 5 shows the nodes connected by both the relationships(NeighbourbyEquipment and NeighbourByRate). The graph has 20 nodes and 236 relationships.

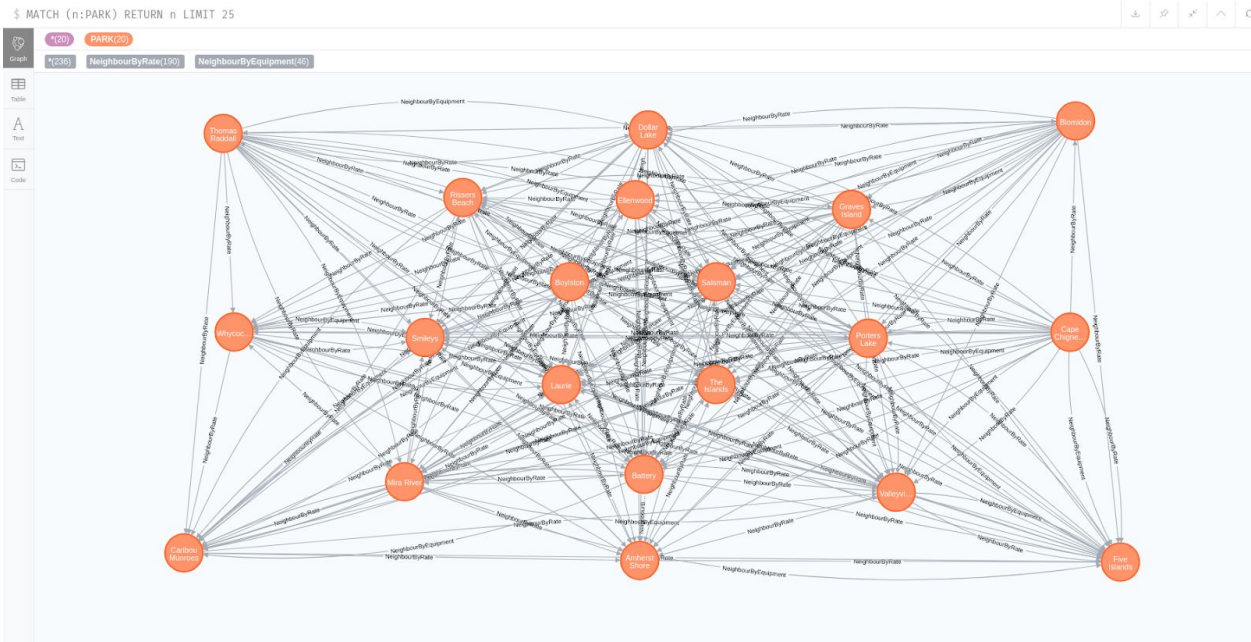

Figure 5: Nodes and all the relationships between them.

Parks with the highest number of PartySize.

I used the following query[5] to get the parks with the highest number of PartySize.

```
MATCH (n)
RETURN labels(n),n.ParkName, n.partySize
ORDER BY n.partySize DESC
```

Figure 6 shows the list of parks in the decreasing order of their partySize. Clearly, the park “Graves Island” has the highest partySize 35.

labels(n)	n.ParkName	n.partySize
["PARK"]	"Graves Island"	35
["PARK"]	"Cape Chignecto"	14
["PARK"]	"Ellenwood Lake"	12
["PARK"]	"Valleyview"	11
["PARK"]	"Dollar Lake"	11
["PARK"]	"Blomidon"	10
["PARK"]	"Caribou Munroes"	9
["PARK"]	"Battery"	9
["PARK"]	"Mira River"	9

Figure 6: Parks in the decreasing order of their partySize

References:

[1] “DNR Camping Parks Reservation Data 2016,” *Novascotia.ca*, 23-Jan-2017. [Online]. Available: https://data.novascotia.ca/Lands-Forests-and-Wildlife/DNR-Camping-Parks-Reservation-Data-2016/4zt7-x443. [Accessed: 25-Feb-2020].

[2] “The Neo4j Cypher Manual v4.0,” *Neo4j.com*, 2015. [Online]. Available: https://neo4j.com/docs/cypher-manual/current/. [Accessed: 25-Feb-2020].

[3] “CSV Import Guide: How to Use LOAD CSV Command for Neo4j,” *Neo4j Graph Database Platform*, 26-Feb-2020. [Online]. Available: https://neo4j.com/developer/guide-import-csv/. [Accessed: 25-Feb-2020].

[4] “3.10. CREATE - Chapter 3. Clauses,” *Neo4j.com*, 2015. [Online]. Available: https://neo4j.com/docs/cypher-manual/current/clauses/create/. [Accessed: 25-Feb-2020].

[5] “3.7. ORDER BY - Chapter 3. Clauses,” *Neo4j.com*, 2015. [Online]. Available: https://neo4j.com/docs/cypher-manual/current/clauses/order-by/. [Accessed: 25-Feb-2020].