Section A: Data Summary

The name of the dataset is "DNR Camping Parks Reservation Data 2016" which includes data related to the parks of Canada, USA and Other countries for the public to book camping sites. This dataset was last updated on July 5, 2017. This dataset is published on Jan 13, 2017, in the English language on the website data.novascotia.ca. This dataset is published under licence of "Nova Scotia Open Government Licence", which allows users to copy, modify, publish, translate, adapt, distribute or otherwise use the Information in any medium, mode or format for any lawful purpose[4].

This dataset includes 13 attributes[4]:

1. ParkName
    - Datatype: Plaintext
    - This attribute mentions the name of parks.
2. State
    - Datatype: Plaintext
    - This attribute contains only the initials of states.
3. Country
    - Datatype: Plaintext
    - This attribute contains the name of the country.
    - Values: Canada, USA, Other
4. Adult
    - Datatype: Number
    - This attribute contains the number of adults that it can accommodate in a party.
5. Child
    - Datatype: Number
    - This attribute contains number of children that it can accommodate in a party. It also contains null values if a child cannot be accommodated in a party.
6. partySize
    - Datatype: Number
    - This attribute contains total party capacity, which is the sum of adults and child attributes.
7. RateType
    - Datatype: Plaintext
    - This attribute contains the type of rate applicable for different age groups and occupations.
    - Values: Full, Senior, Veteran, Host
8. BookingType
    - Datatype: Plaintext
    - This attribute contains which permit is required for booking.
    - Values: CampsitePermit, BackcountryPermit, YurtPermit,
9. Equipment
    - Datatype: Plaintext
    - This attribute contains values that define what kind of equipment/housing are available at the park.
    - Values: Tents, Cabin, Single Tent, Less than 20ft, Less than 30ft, Less than 40ft, Greater than 40ft

10. BookingStartDate
    - Datatype: Date & Time
    - This attribute contains a date that is the starting date for booking for a park.
11. BookingEndDate
    - Datatype: Date & Time
    - This attribute contains a date that is the ending date for booking for a park.
12. Night
    - Datatype: Number
    - This attribute contains the count of nights for which the park is available. This count is count of nights between attribute BookingEndDate and BookingStartDate.
13. Permits
    - Datatype: Number
    - This attribute contains number of the permits acquired in attribute BookingType.

Section B: Python Script

Please find “cleaning_script.ipynb” or “cleaning_script.py” file in “implementation 1” folder.

I have used Kaggle online platform to perform cleaning. To run given notebook upload it on Kaggle with input file. All output files will be generated as per given instructions [6] [7] [8].

Section C: Install and Explore Neo4j

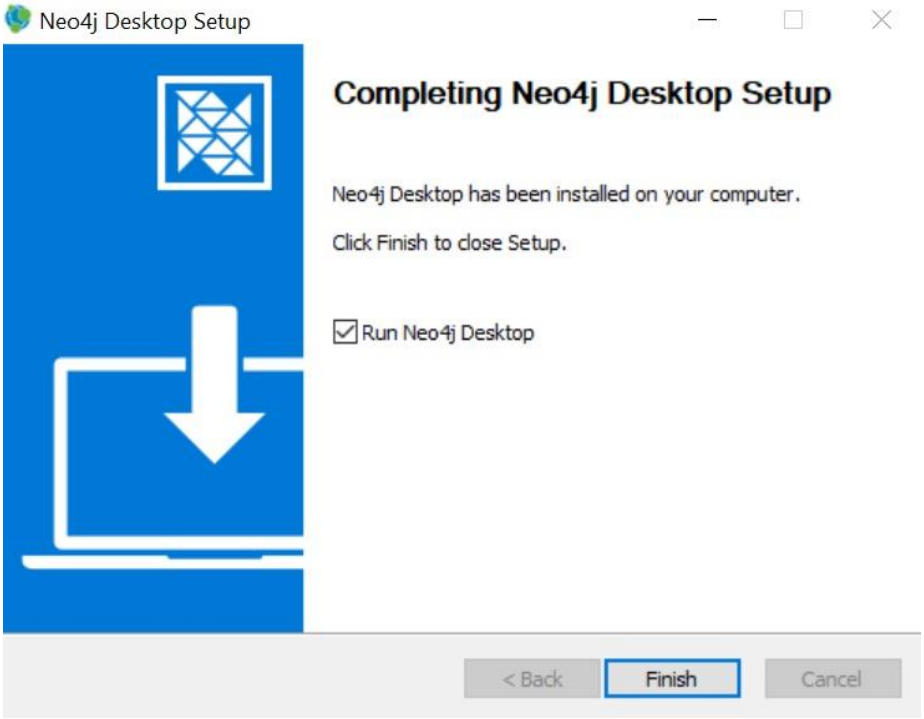

Figure 1: Install Neo4j

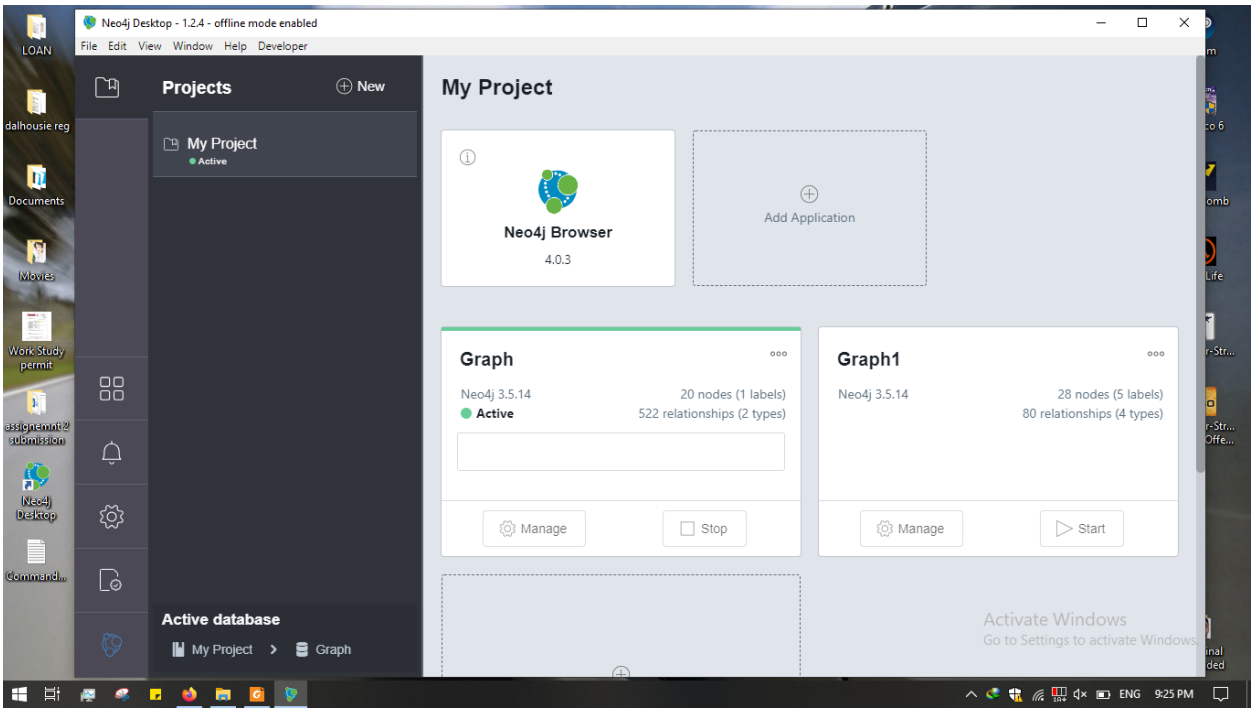

Figure 2: Explore Neo4j

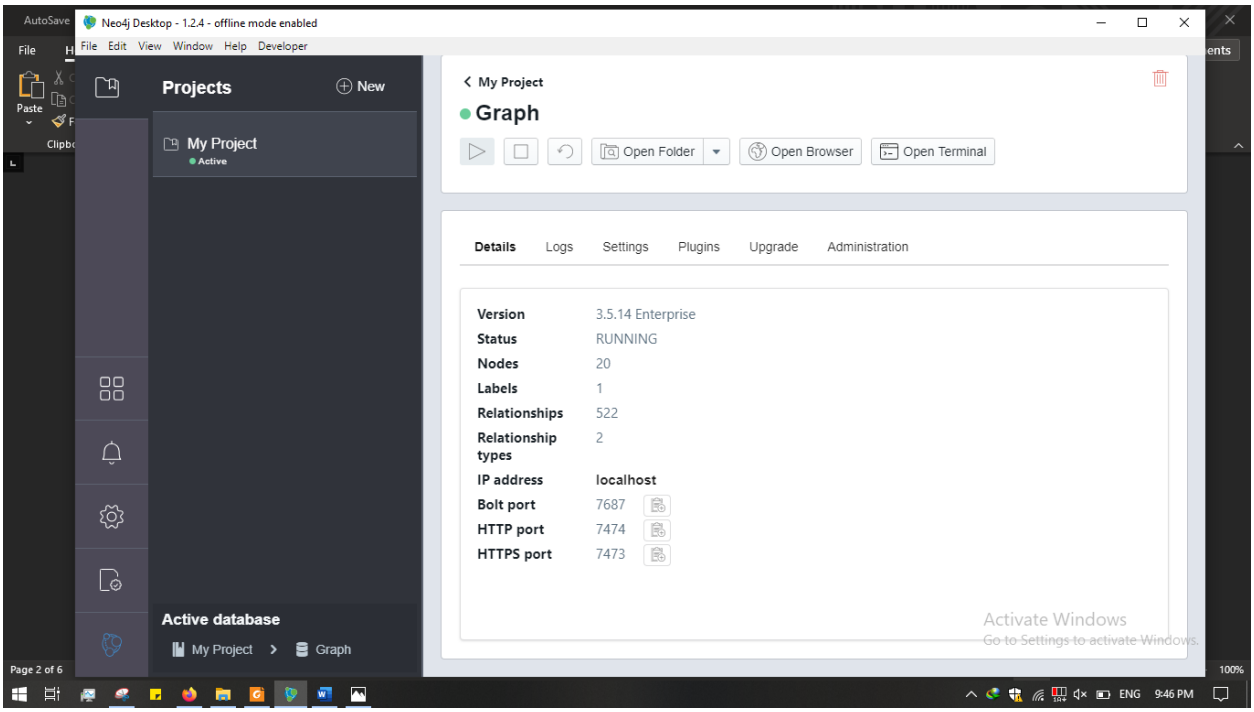

Figure 3: Explore Neo4j

Section D:

- Please find the images of all the relationships as per given instructions in "**Implementation 1 > Images**" folder.
- Please find the images of better approach mentioned at the end of **Section E** in "**Implementation 2 > Images**".

Queries [1]:

1. Load data from csv and create graph

   ```
   load csv with headers from "file:///file4.csv" as r
   create (p: parks)
   set p=r
   ```

2. Create relation between nodes based on same equipment

   ```
   match (a: Parks),(b: Parks)
   where a.Equipment = b.Equipment and a.ParkName <> b.ParkName
   create (a) - [r: NeighbuorByEquipment] -> (b)
   return r
   ```

3. Create relation between nodes based on same ratetype

   ```
   match (a: Parks), (b: Parks)
   where a.RateType = b.RateType and a.ParkName <> b.ParkName
   create (a) - [r: NeighbourByRatetype] -> (b)
   return r
   ```

4. Convert column to integer datatype from string

   ```
   match (a: Parks)
   set a.partySize = toInteger(a.partySize)
   ```

5. Find max PartySize

   ```
   match (p: Parks)
   return p
   order by p.partySize desc limit 1
   ```

Section E: Neo4j Technical Summary

Neo4j is a graph database management system in which Cypher query language (CQL) is used to the query database. Neo4j is built in Java language. Cypher query used in Neo4j is inspired by Structured Query Language (SQL) since SQL is industry standard and has a long time in practice. Cypher query uses similar query structure and keywords, such as WHERE, CREATE and ORDER BY, used in SQL, which makes it easy for developers as well as professionals. Whereas the pattern matching approach is inspired by SPARQL. List semantics used is Cypher query are identical to Python and Haskell [1]. Other popular graph databases are GraphBase, Oracle NoSQL Database, HypherGraphDB, InfiniteGraph, OrientDB, and AllegroGraph [1].

Relational database management systems store data in structured format only and not the relationship between data. Nowadays data is useless if it can not be connected, thus graph database came into the picture. Graph database store connections and relations as an entity. The simple form of the database used in Neo4j makes it easy to insert, update and delete a small amount of data. So, it can be used as the Online Transactional Processing (OLTP) database as well. In OLTP, a large number of users or transactions can be dealt with easily [2].

In Graph database compare o RDBMs, tables are considered as Graphs, rows as nodes, attributes as properties and values, constraints as relationships, and join as traversals. In Neo4j the Graphs contains nodes, nodes contain attributes and values in key-value pair. Nodes are connected to each other through edges which are called relationships [2].

Advantages of Neo4j [2]:

1. Flexible data model:
   The data model provided in Neo4j can be easily changed based on the requirement of application and domain yet remains powerful.
2. Real-time insights:
   Results are based on real-time data available in the database.
3. High availability:
   Neo4j provides a transactional guarantee, which can be used in real-time applications, providing high availability.
4. Connected and semi-structured data:
   Neo4j allows the storage of semi-structured data. Moreover, allows connections between them, not like RDBMS.
5. Easy retrieval:
   Neo4j allows creation paths and relationships between nodes, which allows faster data retravel through traversal compare to rival Graph databases.
6. Cypher Query Language:
   Cypher Query Language is a human-readable query language similar to Structured Query Language, which makes Neo4j more accessible for developers and professional analytics. CQL is a declarative language that uses English like statements to query graph databases.
7. No Joins:
   Compare to RDBMS, Neo4j does not requires joins to fetch related data from multiple tables. Neo4j allows the creation of relationships between nodes, which makes it easy to retrieve connected nodes by simply traversing graphs without indexing or joining.

Features [1][2]:

1. ACID Properties:
   Neo4j supports ACID properties because of this, Neo4j can be used as the Online Transactional Processing database.
2. Scalability:
   Neo4j allows users to modify the number of reads/writes on the database without effecting speed and integrity.
3. Reliability:
   Neo4j provides the replication feature, which increases the reliability of the database.
4. Built-in web application:
   Neo4j browser is a built-in web application that allows users to browse data by querying it without installing additional software.
5. Drivers:
   Neo4j supports multiples drivers using which Neo4j data accessed by different programming languages and frameworks.
   Neo4j supports REST API, which can be used by Java, Spring, python etc. It also supports Cypher API and Native Java API, through which Java application can use Neo4j graphs.
6. Indexing:
   Neo4j supports indexing of data like the Relational Database Management System. It uses Apache Lucene to provide an indexing feature.

Limitations [3]:

1. No range index:
   Neo4j doesn't support range indexing thus sorting operation may take a lot of time.
2. Lack of built-in distribution of data:
   Neo4j uses master-slave topology while using replication. So, each machine contains whole data rather than a chunk of data. Every write goes through a master node only, which can be the bottleneck in case of heavy write operations. It also acts as a single point of failure, if the master node goes down, the whole database goes down.
3. Slow read for large database:
   Once the database exceeds the limit of available RAM in the machine, whole data is stored in hard drive. Which results in more costly read operations.

Alternative of Neo4j:

There are plenty of graph databases available in the market but Neo4j is leading and pure graph database. ArangoDB is the closest competitor having almost all features like Neo4j. Comparatively, ArangoDB is easier to admin and business [5].

An alternative approach to present given dataset:

The given dataset is not normalized and does not have the primary key. To make this data more presentable, data should be normalized first, which will make graphs with relationships, easier to understand. The same is implemented and screenshots are attached in the document (**See 2nd Implementation folder**).

References:

[1] "Chapter 1. Introduction - The Neo4j Cypher Manual v4.0", *Neo4j.com*, 2020. [Online]. Available: https://neo4j.com/docs/cypher-manual/current/introduction/#cypher-introduction. [Accessed: 27- Feb- 2020]

[2] "Neo4j Tutorial - Tutorialspoint", *Tutorialspoint.com*, 2020. [Online]. Available: https://www.tutorialspoint.com/neo4j/index.htm. [Accessed: 27- Feb- 2020]

[3] Available: https://www.quora.com/What-are-the-weaknesses-of-Neo4j. [Accessed: 27- Feb- 2020]

[4] *Data.novascotia.ca*, 2020. [Online]. Available: https://data.novascotia.ca/Lands-Forests-and-Wildlife/DNR-Camping-Parks-Reservation-Data-2016/4zt7-x443. [Accessed: 26- Feb- 2020]

[5] Available: https://www.quora.com/What-are-some-open-source-alternatives-to-Neo4j. [Accessed: 27- Feb- 2020]

[6] "Kaggle: Your Machine Learning and Data Science Community", *Kaggle.com*, 2020. [Online]. Available: https://www.kaggle.com. [Accessed: 27- Feb- 2020]

[7]"Python | Pandas dataframe.drop_duplicates() - GeeksforGeeks", *GeeksforGeeks*, 2020. [Online]. Available: https://www.geeksforgeeks.org/python-pandas-dataframe-drop_duplicates/. [Accessed: 27- Feb- 2020]

[8]"User Guide — pandas 1.0.1 documentation", *Pandas.pydata.org*, 2020. [Online]. Available: https://pandas.pydata.org/docs/user_guide/index.html. [Accessed: 27- Feb- 2020]