

# **CSCI 5408 – Data Management, Warehousing and Analytics**

## **Group Project Report (Group-29)**

**Vamsi Gamidi (B00834696), Meet Patel (B00840009)**

In this group project, we have used *Neo4j* graph database.

### **Significance of Neo4j:**

- It does not require complex joins to retrieve connected/related data as it is very easy to retrieve its adjacent node or relationship details without joins or indexes[3].
- Performs way better than relational (SQL) and non-relational (NOSQL) databases for huge data sets with complex relationships as it treats relationships between data as equally important to the data itself[1].
- Highly scalable and available[2].
- Constant time traversals in big graphs for both depth and breadth due to efficient representation of nodes and relationships[2].
- Performance doesn't drop even after increasing the data queries exponentially[1].

### **Limitations of Neo4j:**

- Hard to visualize complex data sets.
- It does not support sharding[3].
- Limitation of supporting number of nodes, relationships and properties[3].

### **Individual Contribution:**

Both of us worked on data scraping and Neo4j individually after initial discussion, compared our works and came up with the final outcome together.

### **References:**

[1] Bryce Merkl Sasaki's 'Graph Databases for beginners: Why connected data matters', July 17, 2018 [Online], Available: <https://neo4j.com/blog/why-graph-data-relationships-matter/> [Accessed on Oct. 26, 2019]

[2] Neo4j developer guide, 'The property graph model', Available: <https://neo4j.com/developer/graph-database/#property-graph> [Accessed on Oct. 26, 2019]

[3] Tutorials point, 'Neo4J features and advantages', Available: [https://www.tutorialspoint.com/neo4j/neo4j\\_features\\_advantages.htm](https://www.tutorialspoint.com/neo4j/neo4j_features_advantages.htm) [Accessed on Oct. 26, 2019]