**Neo4j**

Neo4j is an ACID-compliant transactional database with native graph storage and processing.

**Significance of Neo4j:**

* Performs way better than relational (SQL) and non-relational (NOSQL) databases for huge data sets with relationships
* Treats relationships between data as equally important to the data itself
* Stores connections (relationships) alongside the data in the model unlike other databases which compute relationships at query time through JOIN operations (Time Consuming).
* ACID compliance, cluster support, runtime failover
* **Constant time traversals** in big graphs for both depth and breadth due to efficient representation of nodes and relationships
* Has drivers popular programming languages, including Java, JavaScript, .NET, Python, and many more
* Performance doesn’t drop even after increasing the data queries exponentially
* **possibility of adding more nodes and relationships to an existing graph is huge**

**Limitations of Neo4j:**

* Neo4j is not as useful for operational use cases because they are not efficient at processing high volumes of transactions and they are not good at handling queries that span the entire database, you would need to combine a graph database with a relational or NoSQL database
* Hard to visualize complex data sets

[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]

<https://en.wikipedia.org/wiki/Neo4j>

<https://neo4j.com/blog/why-graph-data-relationships-matter/>

<https://neo4j.com/developer/graph-database/#property-graph>

<https://tdwi.org/articles/2017/03/14/good-bad-and-hype-about-graph-databases-for-mdm.aspx>

<https://www.tutorialspoint.com/neo4j/neo4j_features_advantages.htm>