Strengths and weaknesses of databases cont'd

5. Graph databases

The modern graph database is a data storage and processing engine that makes the persistence and exploration of data and relationships more efficient. Graph Database is essentially a collection of relationships. Graph databases are designed for data whose relations are well represented as a graph consisting of elements connected by a finite number of relations. A graph data store manages two types of information, nodes and edges. Nodes represent entities, and edges specify the relationships between these entities. Each memory (a node) symbolizes an entity (a business, person, or object) which is connected to another. The connection is called an "edge" and represents a relationship between two nodes. Both nodes and edges can have properties that provide information about that node or edge, similar to columns in a table. Edges can also have a direction indicating the nature of the relationship. Examples of data include social relations, public transport links, road maps, network topologies, etc.

Each node within a Graph Database includes a unique identifier, a set of incoming edges and/or outgoing edges, and characteristics that are represented as "key-value pairs." Each edge also comes with a unique identifier, an ending and/or a starting place node, and a collection of properties.

The purpose of a graph data store is to allow an application to efficiently perform queries that traverse the network of nodes and edges, and to analyze the relationships between entities.

Strengths

- Performance : Graph databases provides much better performance when it comes to querying deeply connected data that has many relationships expressed with complex

joins. The performance stays constant with very large datasets which is not possible in RDBMS because in the graph data model, the query will check only the part of the graph that will be traversed by the query and not the whole graph.

- Flexibility: The data captured can be easily changed and extended for additional attributes and objects.
- Search: It can run fast relationship-based searches
- Agility: Developing with graph databases aligns perfectly with today's agile, test-driven development practices, allowing your graph database to evolve in step with the rest of the application and any changing business requirements. Modern graph databases are equipped for frictionless development and graceful systems maintenance.
- Indexing: Graph databases are naturally indexed by, providing faster access compared to relational data

Weaknesses

- Not as useful for operational use cases because they are not efficient at processing high volumes of transactions
- Not good at handling queries that span the entire database because they are not optimized to store and retrieve business entities such as customers or suppliers, you would need to combine a graph database with a relational or NoSQL database.
- Graph databases do not create better relationships. They simply provide speedy data retrieval for connected data. Improved search is great but not if the relationship wasn't captured effectively in the first place.
- You have to store all the data on one server in many cases, i.e. limited to a single node and can't scale beyond a certain point.
- Graph databases are not optimized for large-volume analytics queries typical of data warehousing.

Final Verdict

Based on the various parameters, strengths and weaknesses, a Graph database may be used for

- Visualizing, analyzing, or helping in finding connections between different pieces of data.

- Fraud detection
- Network Analysis
- Real-time recommendation engines
- Master data management (MDM)
- Network and IT operations
- Identity and access management (IAM)

Some of the examples of Graph databases are Infinite Graph, Neo4J, InfoGrid, HyperGraphDB, GraphBase, Titan etc.

Assignment

1. What is Graph database? Where they are used?
2. In what situations, Graph databases are not suggested?