Graph Databases & Graph Theory

What is a Graph Database?

A Graph Database is a type of database designed to store and manage data using the graph data structure, which consists of:

- Nodes (Vertices): Represent entities (e.g., people, locations, products).
- Edges: Define relationships between nodes (e.g., "friend_of," "works_at").
- Properties: Attributes associated with nodes or edges (e.g., name, age, weight).

Graph databases allow for efficient graph-based queries such as traversals and shortest path calculations, making them suitable for applications involving complex relationships.

Where Do Graphs Appear?

Graphs are commonly used in many fields, including:

- Social Networks: Platforms like Instagram and Facebook rely on graphs to model relationships between users.
- The Web: The internet is essentially a massive graph of web pages (nodes) connected by hyperlinks (edges).
- Biological and Chemical Data: Graphs are used in genetics and molecular biology to model interactions between genes and proteins.

Basics of Graph Theory

Labeled Property Graphs

A Labeled Property Graph consists of:

- Nodes (Vertices): Represent individual entities.
- Edges (Relationships): Define connections between nodes.
- Labels: Categorize nodes into groups (e.g., "Person," "Car").
- Properties: Store key-value (KV) attributes on nodes and edges.

Example:

- Node Labels: "Person," "Car"
- Relationship Types: "Drives," "Owns," "Lives_with," "Married_to"

Nodes can exist without relationships, but edges must always connect two nodes.

Paths in a Graph

A Path is an ordered sequence of nodes connected by edges, where no nodes or edges repeat.

Example:

- Valid path: 1 → 2 → 6 → 5
- Invalid path: 1 → 2 → 6 → 2 → 3 (node 2 repeats)

Types of Graphs

Connected vs. Disconnected Graphs

- Connected Graph: There is a path between any two nodes.
- Disconnected Graph: Some nodes are not connected to the rest of the graph.

Weighted vs. Unweighted Graphs

- Weighted Graph: Each edge has a numerical weight (e.g., distance, cost).
- Unweighted Graph: All edges are treated equally.

Directed vs. Undirected Graphs

- Directed Graph: Edges have a direction (e.g., "follows" on social media).
- Undirected Graph: Edges do not have direction (e.g., "friends with").

Cyclic vs. Acyclic Graphs

- Cyclic Graph: Contains at least one cycle (a path that returns to the same node).
- Acyclic Graph: No cycles exist.

Sparse vs. Dense Graphs

- Sparse Graph: Few edges relative to the number of nodes.
- Dense Graph: Many edges relative to the number of nodes.

Trees (Special Type of Graphs)

A Tree is a type of acyclic and connected graph where there is exactly one path between any two nodes.

Types of Graph Algorithms

Pathfinding Algorithms

Pathfinding determines the shortest path between two nodes

- Shortest Path: Minimizing the number of edges or the total weight of a path.
- Average Shortest Path: Measures efficiency and resiliency of networks.
- Other pathfinding algorithms: Minimum spanning tree, cycle detection, max/min flow.

Breadth-First Search (BFS) vs. Depth-First Search (DFS)

- BFS: Explores neighbors first before moving deeper. Good for shortest paths.
- DFS: Explores deep into one branch before backtracking. Good for pathfinding in mazes.

Centrality & Community Detection Algorithms

- Centrality: Identifies important nodes in a network.
    - Example: Finding social media influencers.
- Community Detection: Identifies clusters or partitions in a graph.
    - Example: Detecting close-knit groups in social networks.

Famous Graph Algorithms

- Dijkstra's Algorithm: Finds the shortest path in graphs with positive edge weights.
- A Algorithm:* Similar to Dijkstra's but incorporates heuristics to speed up searches.
- PageRank: Measures the importance of nodes based on incoming links (used by Google Search).

Neo4j - A Graph Database

Neo4j is a leading graph database system that supports both transactional and analytical processing of graph-based dat

Features of Neo4j

- Schema-Optional: Does not require a predefined schema but allows one if needed.
- ACID Compliance: Supports atomicity, consistency, isolation, and durability.
- Indexing: Allows efficient retrieval of nodes and edges.
- Distributed Computing: Can scale horizontally across multiple machines.

Other Graph Databases

- Microsoft Cosmos DB
- Amazon Neptune

Graph databases are useful for applications where relationships between data points are as important as the data itself, such as social networks, recommendation systems, and fraud detection.