

## DB Commands

### Graph DB

- Good for social networks, the web, chemical and biological data.
- Properties are attributes (key value pairs) that can exist on nodes and relationships
- Centrality: determining which nodes are more important in a network compared to other nodes.
  - Degree: number of connections
  - Betweenness: which node has control over flow between nodes and groups
  - Closeness: which node can easily reach all others in a graph
  - PageRank: ranking connections
- Community detection: evaluate clustering or partitioning of nodes of a graph and tendency to strengthen or break apart.
- Neo4j is ACID compliant, and schema optional.

### Neo4J Syntax

Create a Node: `CREATE (:User {name: "Jake", birthplace: "Paris"})`

- Before the ":" is the variable name for the specific node, then the node label, which in this case is "User", followed by any properties of the node

Label a node: `MATCH (alice:User {name: "Alice"})`

- Sets the existing node to have the variable name "alice"

Create an edge between nodes: `CREATE (alice)-[:KNOWS {since: "2022"}]->(bob)`

- Alice and bob are the variable names created. The relationship must be directed.

Query: `MATCH (usr:User {birthPlace: "London"}), RETURN usr.name, usr.birthplace`

- Find all users who were born in London

### Redis: Key-Value DB

### MongoDB: Document DB