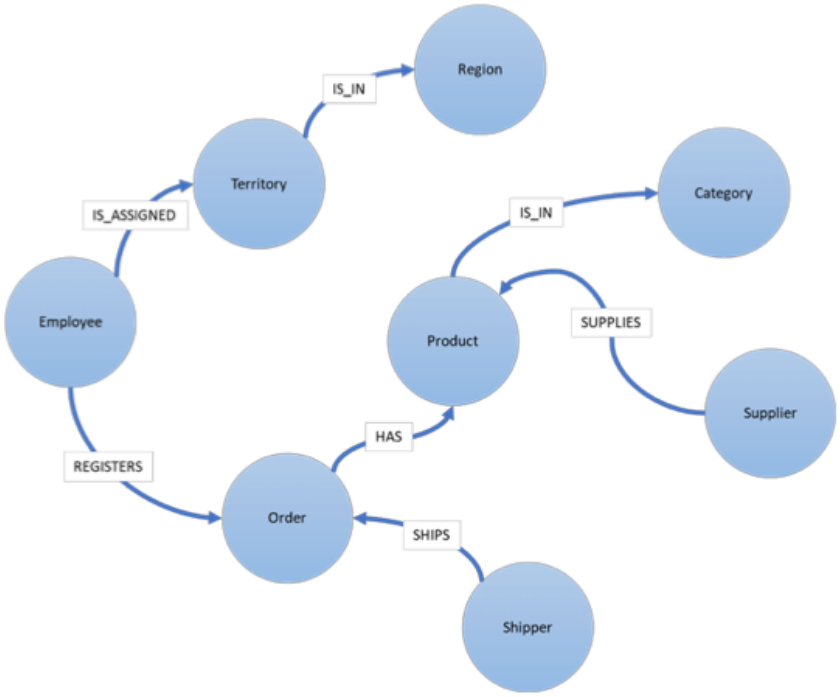
To technical audience:

Overview of Graph Database:

- Definition: Graph database (GDN) is a non-relational database which uses graph structures/relationships to represent and store data
- Some example of graph database are Neo4j, Amazon Neptune and Azure Cosmos
- No universal query language so far; popular vendor query languages include Gremlin, SPARQL, and Cypher
- No strict schema and data normalization required
- Can be accessed through APIs
- For example, a subset of the relationship in a graph database can be represented in the below diagram:



(Diagram cited from mssqltips.com)

Understanding Graph Structure and its components:

- For a Graph Database, its physical structure looks just like a graph showing above, with nodes, edges and directions showing the various relationship between entities.
- Components of a graph:
    - Node – represents entities
    - Edges — represents relationships between nodes
        - Relationship has first-class citizenship
        - Edges can connect nodes in any possible way — no rules

        - Each edge has at least 2 nodes associated with it
    - Properties such as directions, sub-graphs, weight of the edges that define relationship
- **Traversing** – the act of journeying through or visiting nodes in the graph
    - Comparable to the JOINs in relational databases

Graph Types:

- Social graph
- Intent graph
- Consumption graph
- Interest graph
- Mobile graph

Use Cases:

- Fraud detection
- Recommendation
- Personalization
- Asset administration

Benefits of Graph Databases over Relational Databases:

- Querying relationships is fast because they are perpetually stored in the database
- Object-oriented: this means very clear, explicit semantics for each query you write
- Useful for heavily inter-connected data and outperforms relational databases where many join operations on different attributes over multiple tables are required
- No strict schema, normalization, foreign key constraints to be considered, avoiding additional overhead
- Update data in real-time and support queries simultaneously: Graph databases can perform time to time updates on big data while supporting queries at that time
- Flexible online schema environment: you can constantly add and drop new vertex
- Combine and hierarchize multiple dimensions: Graph databases can combine multiple dimensions to manage big data, including time series, demographic, geo-dimensions, etc

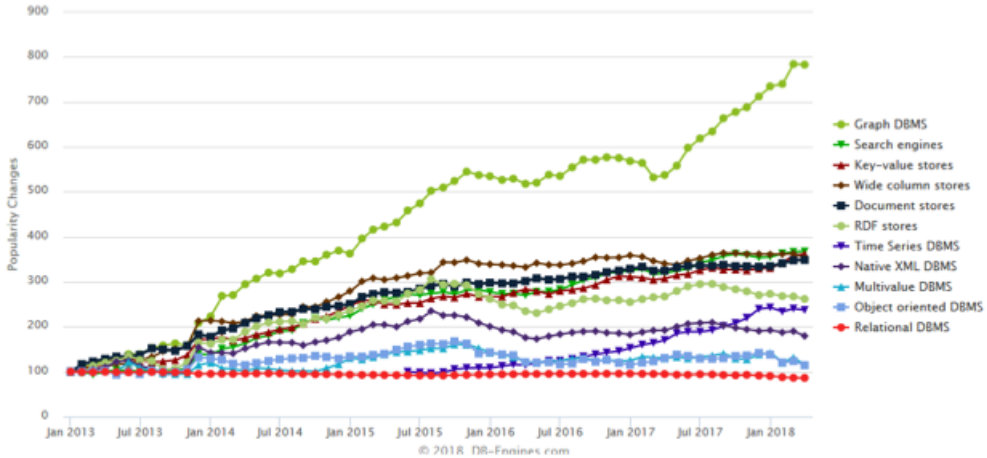
Limitations of Graph Databases over Relational Databases:

- The Graph Databases are typically slower at performing on large numbers of data elements and less capable of keeping an audit trail of changes
- Not a good fit for business intelligence and performance analytics, as they don't do aggregations as well as relational databases

To Non-technical audience:

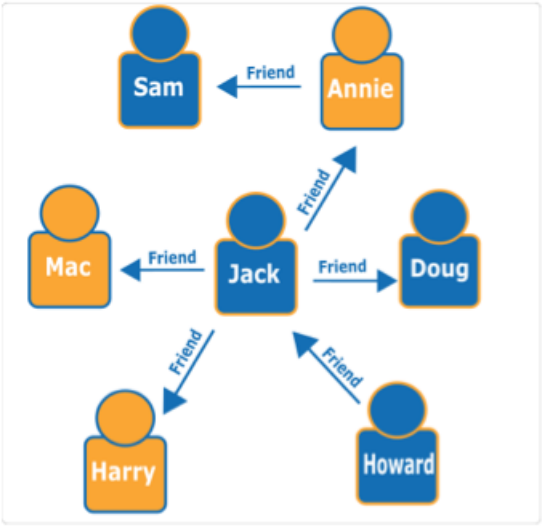
Data exists in various forms from very simple structure to very complex ones. At its simplest form, data can be expressed in the form of key - value pairs. Unlike Relational databases such as MySQL, PostgreSQL that present and store information in tables and columns, Graph databases use graph structure/relationship to do so.

Graph databases started to attract much attention starting from 2000s, due to its popular in major tech companies such as Google, Facebook and Linkedin. A graph database is essentially a database that is based on graph theory and it consists of a set of objects such as Nodes, Edges, and Properties.



(Cited from https://medium.com/the-andela-way/graph-databases-why-are-they-important-c438e1a224ae)

One of the most popular use cases of Graph Database is for network related data, where people and entities as well as their relationships are connected: in the following example, each node represents a person in this network, and their relationships are represented using edges with directions. The properties of the relationship are shown by the text 'Friend' associated with the edges, which means each person is friend to each other.



(Graph cited from aws.amazon.com)

So the essential question is — when should you adopt graph database over relational database? If you are keen to know what your customers are interested in to gear messages in their topic areas, or understand how a network map is laid out, a graph database is a perfect choice. Essentially, graph database can allow businesses to create well-rounded, diverse customer profiles.

Resources referred:

- **Benny Ogidan, Medium** - "Graph databases, Why are they important"
- **Jennifer Reif, Medium** – ''How Do You Know If a Graph Database Solves the Problem?"
- **MT Buzzer, Medium** — "Graph database vs. relational database"
- **Siddharth Mehta, MSSQLTips** – 'Relationship Databases vs Graph Databases'