Neo4j

and graph databases

Presented By: Stephanie McIntyre
Graph Databases: The Database Model

The Database Model determines the logical structure.

Graph databases (GDB) have the following characteristics:

- Data & Schema are represented by a graph.
- Manipulation of data is expressed by graph transformations.

The driving force of this model is the focus on the relationships between entities.
Graph Databases: Data & Schema

A GDB is a directed, labelled, attributed multigraph.

- **Nodes**: connected entities
  - defined by unique id
- **Edges**: binary relations
  - direction, start node, end node
- **Properties**: descriptive information
  - on nodes, sometimes on edges

Nodes & Edges are tagged with **labels**.
Graph Databases: Data & Schema

- No global adjacency index.
  (exception: Neo4j for starting points).

- Each node and edge stores a mini-index of all of the connected objects.

- No NULLs

- GDBs have an ad-hoc schema that is easily mutable.
  - Neo4j supports schema-less
Graph Databases: Going for a Walk

Query processing makes use of graph-based algorithms and retrieval of information requires graph traversal, aka “walking”.

When retrieving information, it only looks at the objects directly connected to the node.

Graph databases are typically better than relational databases at handling recursion.
Graph Databases: Example
# Graph vs Relational Databases

Some differences...

<table>
<thead>
<tr>
<th></th>
<th>Graph Databases (Neo4j)</th>
<th>Relational Databases</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mathematical Model</strong></td>
<td>Property Graphs</td>
<td>N-ary Predicates</td>
</tr>
<tr>
<td><strong>Relationships</strong></td>
<td>Record Level</td>
<td>Table Level</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Schema-less or on labels</td>
<td>Schema Based</td>
</tr>
<tr>
<td><strong>Query Processing</strong></td>
<td>Localized</td>
<td>Indexes, Keys</td>
</tr>
<tr>
<td><strong>Costly Queries</strong></td>
<td>Find all</td>
<td>Many-to-Many Table Joins</td>
</tr>
<tr>
<td><strong>Data Manipulation</strong></td>
<td>Easily Mutable</td>
<td>Table Alterations</td>
</tr>
<tr>
<td><strong>Integrity Constraints</strong></td>
<td>....</td>
<td>^_^</td>
</tr>
</tbody>
</table>
Neo4j’s Integrity Constraints

**Inherent Constraints:** Node IDs are unique and edges are composed of labels and the nodes where the edge occurs.

**Node Property Uniqueness:** given property is unique for nodes.

**Relationship Property Existence:** relationships with a given label have a given property.

**Node Keys:** existence of given properties and uniqueness of the combination thereof for a given label.