Neo4j Presentation

1. Graph Database Landscape

Graph databases store data as nodes (entities) and relationships (connections) with properties. They are ideal for highly connected data and are optimized for queries that involve traversing these relationships.

```
# Example Query:
MATCH (p:Person)-[:ACTED_IN]->(m:Movie)
RETURN p.name, m.title;
```

2. Difference Between Relational and Graph Databases

Relational databases store data in tables, relying on foreign keys and JOIN operations to connect entities. Graph databases use nodes, relationships, and properties, enabling faster and more natural queries for connected data.

```
# SQL Example:
SELECT p.name, m.title FROM Person p
JOIN Movie m ON p.movie_id = m.id;

# Cypher Example:
MATCH (p:Person)-[:ACTED_IN]->(m:Movie)
RETURN p.name, m.title;
```

3. Migrating from Relational to Graph Database

To migrate, identify tables as nodes, foreign keys as relationships, and attributes as properties. Use import tools like Neo4j Data Importer to load data.

```
# Example Migration:
LOAD CSV WITH HEADERS FROM 'file:///users.csv' AS row
CREATE (:Person {name: row.name, age: row.age});
```

4. Preparing or Creating a Graph Dataset

To create a graph dataset, start by defining entities (nodes), their connections (relationships), and properties. Use Cypher or import tools to populate your graph.

```
# Example:
CREATE (:Person {name: 'Alice'})-[:FRIENDS_WITH]->(:Person {name: 'Bob'});
```

5. SQL vs. Cypher QL

```
# SQL Query:
```

```
SELECT p.name, m.title FROM Person p
JOIN Movie m ON p.movie_id = m.id;

# Cypher Query:
MATCH (p:Person)-[:ACTED_IN]->(m:Movie)
RETURN p.name, m.title;
```

6. Landscape of Projections

Projections in Neo4j allow you to create in-memory representations of specific parts of your graph for algorithms and analysis.

```
# Example:
CALL gds.graph.project(
  'movieGraph',
  ['Person', 'Movie'],
  ['ACTED_IN']
);
```

7. Landscape of Perspectives

Perspectives in Neo4j are customizable views of the graph, allowing you to focus on specific types of nodes, relationships, and properties.

```
# Example:
MATCH (p:Person)-[:ACTED_IN]->(m:Movie)
RETURN p.name, m.title;
```