Basic Queries in Neo4j With Cypher - Supplementary Resources | Coursera

Basic Graph Operations with CYPHER//Counting the number of nodes

```
match (n:MyNode)

return count(n)
```

//Counting the number of edges

```
match (n:MyNode)-[r]->()

return count(r)
```

//Finding leaf nodes:

```
match (n:MyNode)-[r:TO]->(m)

where not ((m)-->())

return m
```

//Finding root nodes:

```
match (m)-[r:TO]->(n:MyNode)

where not (()-->(m))

return m
```

//Finding triangles:

```
match (a)-[:TO]->(b)-[:TO]->(c)-[:TO]->(a)

return distinct a, b, c
```

//Finding 2nd neighbors of D:

```
match (a)-[:TO*..2]-(b)

where a.Name='D'

return distinct a, b
```

//Finding the types of a node:

```
match (n)
```

```
where n.Name = 'Afghanistan'

return labels(n)
```

//Finding the label of an edge:

```
match (n {Name: 'Afghanistan'})<-[r]-()

return distinct type(r)
```

//Finding all properties of a node:

```
match (n:Actor)

return * limit 20
```

//Finding loops:

```
match (n)-[r]->(n)

return n, r limit 10
```

//Finding multigraphs:

```
match (n)-[r1]->(m), (n)-[r2]-(m)

where r1 <> r2

return n, r1, r2, m limit 10
```

//Finding the induced subgraph given a set of nodes:

```
match (n)-[r:TO]-(m)

where n.Name in ['A', 'B', 'C', 'D', 'E'] and m.Name in ['A', 'B', 'C', 'D', 'E']

return n, r, m
```