The steps I followed here are:

1. Create the indexes on the nodes (Before loading the data)

*CREATE CONSTRAINT ON (c:Cid\_id) ASSERT c.id IS UNIQUE;*

*CREATE CONSTRAINT ON (r:Rid\_id) ASSERT r.id IS UNIQUE;*

*CREATE CONSTRAINT ON (o:Oid\_id) ASSERT o.id IS UNIQUE;*

2.       Load the data using the Python Neo4j driver

3.       Run the queries to get the desired results

**While loading the data using http protocol, following steps I followed:**

**Instructions to load the data:**

Create http server for the corresponding folder using the below commands

<http://www.pythonforbeginners.com/modules-in-python/how-to-use-simplehttpserver/>

**python -m SimpleHTTPServer 8080**

After that start ngrok to make the port accessible using the ngrok tunnel, now you have the url for the data files which are accessible

Reason for taking this approach was due to unsuccessful attempts to load file from local server.

**Questions:**

1. **Please** write an application that will load the data into the graph database.

* Please follow the GitHub link for queries

1. How many rid nodes are there with no edges?

* None

MATCH (r:Rid\_id) OPTIONAL MATCH (r)--(x) WITH r, x WHERE x IS NULL RETURN COUNT(r);

1. How many cid nodes have more than one edge to an oid node?

* 810 cid

MATCH (:Oid\_id)-[r:HAS]->(c:Cid\_id)

WITH c, count(r) as rel\_cnt

WHERE rel\_cnt > 1

RETURN COUNT(c);

1. Are there any rid nodes with multiple cid nodes? If so, what percent?

* MATCH (r:Rid\_id)-[m:HAS]->(:Cid\_id)

WITH r, count(m) as rel\_cnt

WHERE rel\_cnt > 1

RETURN COUNT(r); // giving out count of **52**

Match (r:Rid\_id)-[m:HAS]->(:Cid\_id)

with r, count(m) as cnt

MATCH (n:Rid\_id)

with r,  cnt , count(n) as total

where cnt >1

return (count(\*)\*100.0)/total; //to get the percentage of **3.74**

5) **Please leave the Neo4j database with all data that you loaded.**