In [36]:

```
#import necessary libraries and files
import pandas as pd
from neo4j import GraphDatabase
import credentials as c
parent = pd.read_csv("DX_Chapters.csv")
children = pd.read_csv("CCSR_Categories.csv")
```

In [37]:

```
# establish connection to neo4j
uri = "bolt:http//127.0.0.1:7687"
driver = GraphDatabase.driver(uri, auth = (c.username, c.password))
session = driver.session()
```

In [38]:

```
#convert csv files to dictionaries

#{INF: Certain Infectious and Parasitic Diseases, NEO: Neoplasms, etc}
parent_dictionary = dict(zip(parent.Abbreviation, parent.Chapter))

#{BLD001:Nutritional anemia, BLD002:Hemolytic anemia, etc}
children_dictinary = dict(zip(children.Code, children.Description))
```

In [39]:

```
# creating parent nodes (DX Chapter)
parent_query = "CREATE "
i = 0
for key in parent_dictionary:
    value = parent_dictionary.get(key)
    statement = "(" + key + ":Chapter" + " { name: " + "'"+ key + "'" + ", FullName: " +
"'"+value + "'" + "}),"
    parent_query = parent_query + statement
```

In [40]:

```
# creating child nodes
children_query = ""
for key in children_dictinary:
    value = children_dictinary.get(key)
    statement = "(" + key + ":Chapter" + " { name: " + "'"+ key + "'" + ", FullName: " +
"'"+value + "'" + "}),"
    children_query = children_query + statement
```

In [41]:

```
parent_list = parent['Abbreviation'].tolist()
children_list = children['Code'].tolist()
```

In [42]:

```
# creating relationship between parent and child nodes

#This dictionary contains key of the three letter code.
#The value associated to each key is a list of all the CSSR categories associated with th
ree letter code
parent_child = {}
for i in range(len(parent_list)):
    listArray = []
    for j in range(len(children_list)):
        if parent_list[i] == children_list[j][0:3]:
            listArray.append(children_list[j])

    parent_child[parent_list[i]] = listArray
```

In [45]:

```
#creating query that maps relatioinships between parent and child node from dictionary cr
eated in previous cell
relationship_query = ""
for parent in parent_child:
    for child in parent_child.get(parent):
        statement = "(" + child + ")-[:BELONGS_TO]->" + "(" + parent +"),"
        relationship_query = relationship_query + statement
relationship_query = relationship_query[:-1]



#combining the queries for child nodes, parent nodes, and relationship between parent and
child
full_query = parent_query + children_query + relationship_query
```

In [46]:

```
session.run(full_query)
```

Out[46]:

```
<neo4j.work.result.Result at 0x1210d7e20>
```