

This file contains Cypher Commands used to create and work with the graph in Neo4J.

```
CALL apoc.periodic.iterate("
CALL
apoc.load.json('file:///Users/shaily/Desktop/Fall2018/Data_Mining/Project/Sample_data/Yelp_
Updated_Data/users.json')
YIELD value RETURN value
",
"
MERGE (u:User{id:value.user_id})
SET u += apoc.map.clean(value, ['friends','user_id'],[0])
WITH u,value.friends as friends
UNWIND friends as friend
MERGE (u1:User{id:friend})
MERGE (u)-[:FRIEND]-(u1)
",{batchSize: 100, iterateList: true});
```

```
CALL apoc.periodic.iterate("
CALL
apoc.load.json('file:///Users/shaily/Desktop/Fall2018/Data_Mining/Project/Sample_data/Yelp_U
pdated_Data/business.json') YIELD value RETURN value
",
"
MERGE (b:Business{id:value.business_id})
SET b += apoc.map.clean(value,
['attributes','hours','business_id','categories','address','postal_code'],[])
",{batchSize: 10000, iterateList: true});
```

```
CALL apoc.periodic.iterate("
CALL
apoc.load.json('file:///Users/shaily/Desktop/Fall2018/Data_Mining/Project/Sample_data/Yelp_U
pdated_Data/review.json')
YIELD value RETURN value
",
"
MERGE (b:Business{id:value.business_id})
MERGE (u:User{id:value.user_id})
MERGE (r:Review{id:value.review_id})
MERGE (u)-[:WROTE]->(r)
MERGE (r)-[:REVIEWS]->(b)
SET r += apoc.map.clean(value, ['business_id','user_id','review_id','text'],[0])
```

```
",{batchSize: 10000, iterateList: true});
```

```
CALL apoc.periodic.iterate("
```

```
CALL
```

```
apoc.load.json('file:///Users/shaily/Desktop/Fall2018/Data_Mining/Project/yelp_dataset/yelp_academic_dataset_tip.json') YIELD value RETURN value
```

```
", "
```

```
MATCH (b:Business{id:value.business_id})
```

```
MERGE (u:User{id:value.user_id})
```

```
MERGE (u)-[:TIP{date:value.date,likes:value.likes}]->(b)
```

```
",{batchSize: 20000, iterateList: true});
```

```
CALL algo.pageRank('User', 'FRIEND',
```

```
{iterations:20, dampingFactor:0.85, write: true,writeProperty:"pagerank"})
```

```
YIELD nodes, iterations, loadMillis, computeMillis, writeMillis, dampingFactor, write, writeProperty
```

```
#
```

```
MATCH
```

```
(u2:USER)-[:REVIEWS]->(c:BUSINESS)-[:REVIEWS]-(a:USER)-[:REVIEWS]->(b:BUSINESS)
```

```
<-[:REVIEWS]-(u1:users) WITH COUNT(a) as common_users WHERE a!=b and
```

```
not((u2:USER)-[:REVIEWS]->(b:BUSINESS)) and not
```

```
((c:BUSINESS)-[:REVIEWS]-(u1:users)) and common_users>1000 RETURN u1,u2
```

```
MATCH
```

```
(u2:USER)-[:REVIEWS]->(c:BUSINESS)-[:REVIEWS]-(a:USER)-[:REVIEWS]->(b:BUSINESS)
```

```
<-[:REVIEWS]-(u1:users) WITH COUNT(a) as common_users WHERE a!=b and
```

```
not((u2:USER)-[:REVIEWS]->(b:BUSINESS)) and not
```

```
((c:BUSINESS)-[:REVIEWS]-(u1:users)) and common_users>500 RETURN u1,u2
```

```
MATCH
```

```
(u2:USER)-[:REVIEWS]->(c:BUSINESS)-[:REVIEWS]-(a:USER)-[:REVIEWS]->(b:BUSINESS)
```

```
<-[:REVIEWS]-(u1:users) WITH COUNT(a) as common_users WHERE a!=b and
```

```
not((u2:USER)-[:REVIEWS]->(b:BUSINESS)) and not
```

```
((c:BUSINESS)-[:REVIEWS]-(u1:users)) and common_users>100 RETURN u1,u2
```

```
MATCH
```

```
(u2:USER)-[:REVIEWS]->(c:BUSINESS)-[:REVIEWS]-(a:USER)-[:REVIEWS]->(b:BUSINESS)
```

```
<-[:REVIEWS]-(u1:users) WITH COUNT(a) as common_users WHERE a!=b and
```

```
not((u2:USER)-[:REVIEWS]->(b:BUSINESS)) and not  
((c:BUSINESS)-[:REVIEWS]-(u1:users)) and common_users>50 RETURN u1,u2
```