Note: We will use network "sample" for demonstration.

Files needed:

File	Description
node_file_gen.py	Generates uel, del and nodes file from input file
gen_n_file.qsub	Executes node_file_gen.py on sfx
preparefiles.py	Generates uel and node files with attributes for database
preparefiles.qsub	Executes preparefiles.py on sfx
dataLoader.py	Loads network data from file into database
Loaddata.qsub	Executes dataLoader.py on sfx through qsub

Processing and generating data on shadowfax:

1. Go to you scripts directory under v1/doc/manu01
2. Edit gen_n_file.qsub and replace graph name with sample2.
3. Execute qsub gen_n_file.qsub on sfx1, this will generate: sample.uel, sample.del and sample.nodes. These files are needed by InterSim.
4. Edit preparefiles.qsub and replace graph name with sample2. Note: make sure the path for the preparefiles.py is correct.
5. Edit preparefiles.py and replace all directories for .uel, .nodes, .uels, and .info files with the correct ones.
6. Execute preparefiles.qsub on sfx1, this will generate: sample2.info and sample2.uel2.

Creating database entries

1. Go to the database directory.
2. Type sqlite3 Edison2.db, this will connect you to Edison database.
3. You can rename database file to any other name if needed.
4. Type .schema to get info about current tables and indexes in the database.
5. type .schema <table name> to get information about specific table
6. Now we will create two tables for the new network: sample_node and sample_edge.
7. The number and data type of the new tables' attributes should be consistent with data columns in the two newly generated files sample2.info and sample2.uel2.
8. After creating the two tables. type .schema sample_node to make sure table is created successfully. Do the same step again .schema sample_edge.
9. Now to add the metadata for the network, we use table network. To get familiar with existing networks type select * from network. Available networks to Edison have the attribute "available" set to true.

10. Before changing/ adding data in network table, dump existing data to a save place as backup. To do these steps:

```
sqlite> .mode csv
sqlite> .output network_backup_4_7_2016.csv
sqlite> select * from network;
sqlite> .output stdout
```

11. Using sql insert command enter the new network data, don't include id as it is automatically generated in the dbms.
12. all networks should have available attribute set to false at beginning until they pass all testing.

Importing data into database

1. Go to scripts directory. Edit loaddata.qsub file, put the name of the table first, in this case sample. Followed by name of node and edge attributes files sample2.uel and sample2.info. This will load the data from each file to the corresponding table. Data includes node/edge ids and attributes.
2. To verify data is loaded correctly:
    a. login back to database
    b. verify number of nodes correct "select count(*) from sample_node" and compare with network number of nodes.
    c. verify number of edges correct "select count(*) from sample_edge" and compare with network number of edges.
    d. verify if isolated nodes exist "select count(*) from sample_node where id not in (select start from sample_edge) and id not in (select end from sample_edge)" if count == 0 then no isolated nodes exist.
    e. verify if network is undirected or has no duplicate edges "select count(*)/2 as duplicates from sample_edge a, sample_edge b where a.start = b.end and a.end = b.start" if duplicates == 0 then no duplicate edges exist.
    f. verify if self loops exists "select count(*) from sample_edge a where start == end " if count == 0 that means no self loops.
    g. verify if node ids are not negative "select count(*) from temp_node where id < 0 " and "select count(*) from temp_edge where start < 0 or end < 0" count should be 0 in both cases.
    h. verify if nodes ids are numeric select count(*) from sample_edge where typeof(start) = "integer" and typeof(end) = "integer" and select(*) from sample_node where typeof(id) = "integer"