What are the main parts of your project, algorithms, not easy concepts, anything important for people outside of INFSCI 2711 class to be able to continue your project

Step1:

First, we need to visit a relational database provided by the user and know the structure of the database through SQL queries. The main method can be found in *mySQL.java*. After that, we can know the database, table, column, record information. Meanwhile, we also need to know the Foreign Key (FK) in the database, so we the *FK.java* are used to find the Foreign Key.

Step2:

After we have the access to the relational database, we need to import it into Neo4j, a graphic database system. (<http://neo4j.com/>) We rebuild the data structure into Neo4j. Our basic assumption is that there is no circle in the nodes of database.The main methods can be found in *SQL2neo4j.java*( main class to implement the methods in *Neo4j.java*)

There are 4 levels of the node label: record, column, table, database. The details are listed as follows:

|  |  |
| --- | --- |
| node | Property |
| database | value(database name), type(database) |
| record | value(record value), type(record) |
| column | value(column name), type(record), parent (table name) |
| table | value(table name), type(table), parent (database) |

There are 3 relational labels: belong to, link, path. The details are listed as follows:

|  |  |  |
| --- | --- | --- |
| Relational Label | Description | Property |
| Belong to | Record-(belong to)-Column  Column-(belong to)-Table  Table-(belong to)-Database | RelationType:  column-table  record-column  table-database |
| Link | In the same database, if 2 columns have FK constraint.  In 2 different database, if 2 columns have the same name. | RelationType  PK-FK  (same database column)  SameNameCol  (different database) |
| Path | If 2 columns can be joined together, there is a path between 2 tables. | RelationType: table.columnname  cost:  same database=1  different database=2 |

Step3:

we need to search the data according to the query. This function is realized in *KeywordSearchDao.java.*

Step4:

For the joined the columns, we need to find the shortest path with minimum cost value among tables. To implement this method, we use the steiner Tree algorithm. (<http://en.wikipedia.org/wiki/Steiner_tree_problem>) This function is implement in *Join.java* (main function), *Kruskai.java*(find the shortest path between 2 nodes）/ *SteineerTree.java.*