APPENDIX-I

CREATING GRAPH DATABASES FROM RELATIONAL DATABASES

IMPLEMENTATION

The data are extracted from the relational database (here MySQL) using JDBC. The user gets a list of all the databases present and can choose which relational database to be converted into graph database (see fig. A-1).

```
list of Databases:
Database1: information_schema
Database2: mysql
Database3: neo4j
Database4: sup_part
Database5: test
Please enter name of the relational database to be converted into graph database:
```

When a database is chosen, the conversion is done table wise selecting data from the table field wise and then performing necessary operation based on the algorithm provided below. This is done for each field in all the tables in the selected database (see fig. A-2).

```
Please enter name of the relational database to be converted into graph database: sup_part
Connecting to database...
List of Tables in Database 'sup_part':
Table1: part
Table2: shipment
Table3: supplier
Selecting table [supplier] to convert into graph database.
No of col= 3
Column1 - sup_no
Column2 - sup_name
Column3 - city
Primary Key is : sup_no
Table [supplier] converted to graph database.
Selecting table [part] to convert into graph database.
No of col= 3
Column1 - part_no
Column2 - part_name
Column3 - color
Primary Key is: part_no
Table [part] converted to graph database.
Selecting table [shipment] to convert into graph database.
No of col= 4
Column1 - supplier
Column2 - parts
Column3 - date
Column4 - qty
Table [shipment] converted to graph database.
```

Figure A-1: Selecting the Relational Database to Convert

Figure A-2: Selecting the Relational Database Tables
After this, the graph gets created in Neo4j using the data. The graph database model is then viewed using a browser.

Software Requirements:

Operating System: Windows 7

Software Tool: MySQL, Neo4j, Eclipse IDE, Mozilla Firefox

Language used: Java, Structured Query Language, Cypher Query

**OUTPUTS**

1. Displaying Complete Database:

![SQL Query](image)

*Figure A-4: SQL Query for displaying complete database.*
2. Display The Details Of Supplier With Supplier No. ‘SI’:

```
mysql> select * from supplier where sup_no='si';
+-------+-----------+-------+
| sup_no | sup_name  | city  |
|--------+-----------+-------|
| s1     | Ram       | Kolkata|
+--------+-----------+-------+
1 row in set (0.00 sec)
```

**Figure A-7: SQL Query To Display The Details Of Supplier With Supplier No. ‘SI’**
3. Display the Details of Part with Part No. ‘P1’:

```
mysql> select * from part where part_no='p1';
+----+-------+-----+
<table>
<thead>
<tr>
<th>part_no</th>
<th>part_name</th>
<th>color</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Nut</td>
<td>Red</td>
</tr>
</tbody>
</table>
+--------+-----------+------+
1 row in set (0.00 sec)
```

4. Display The Details Of Supplier ‘S1’:

```
mysql> select * from shipment where supplier='s1';
+------+------+-----+-----+
<table>
<thead>
<tr>
<th>supplier</th>
<th>parts</th>
<th>date</th>
<th>qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>p1</td>
<td>1/1/13</td>
<td>100</td>
</tr>
<tr>
<td>S1</td>
<td>p2</td>
<td>1/1/13</td>
<td>200</td>
</tr>
</tbody>
</table>
+----------+-------+------|-----+
2 rows in set (0.00 sec)
```
Figure A-14: Cypher query to display the details of all the nodes corresponding to supplier 's1'.

```
match a-[e1]-(b)-[e2]-(c) where a.name = 's1' return a.name, b, c
```

Figure A-15: The Graph Resulting from the Cypher Query in Fig. A-14