Andreas Landgrebe
October 16, 2014
Laboratory Assignment # 4

## 2. **An overview of how the sqlt-graph command line options influence the resulting diagrams**

The sqlt-graph command line are able to output different parts of the datasets. When using the sqlt-graph it is important to be able to follow specific syntax when writing this commands through the terminal. The first parameter to remember is to be the command ”-f”. This will tell the sql visualizer which of SQL was used to write the file. For this laboratory assignment, in the shared repository, all of the SQL files were written in PostgreSQL. If I were to format this file to MySQL or sqlite, then one would receive an error message. The next parameter is the command ”-o”. This parameter will have one name the output file of a specified extension. The next parameter is ”-t”. This will specify the user to decide on the type for format that the database schema will have visualized. By default, the type of file that is used is png files. After all of this information is specified in the command line, then the last part is the specify the sql file that is going to be visualized.

The command-line options listed for the laboratory assignment each performed an important task to be able to understand the provided schemas in the shared repository. The ”–show-datatypes” command line options when executed will allow you to see all of the data types of the attributes in table. This helps one to be able to see what data and values will be put in each attribute. This is important to be able to know if integers or other characters are going to be put in different row and columns. The next command line options mentioned in the laboratory assignment is ”–show-sizes”. This command-line option allows to see how many characters an attribute is allowed to have. This is important when importing data to be able to know how many characters one is allowed to put into a specific attribute so one does not know a risk of errors if one inputs too many characters to a particular value. The last command line options that was mentioned is ”–show-constraints”. This options will allow one to be able to see the constraints of each attribute in a schema. This constraints may three of the following: PK, FK, N, and U. These constraints stand for the primary key, foreign key, and not null. The primary key is the record that uniquely identifies in the table. The foreign key is the key that will be used to be able to use the join command for structured query languages to be able to establish a link between two tables. The U stands for unique. This constraint does not necessarily make a particular attribute a key, it is just considered to be unique.

## 3.**A brief commentary on the visualizations from the textbook and the sqlt-graph tool.**

The visualizations from the textbook allow one to be able to see how to use the join command for SQL. In the textbook, there is already data added into the tables. According to the visualization of the textbook, the join command is able to see a common attribute which in this case is the id. After one can see that the common attribute between the two tables is the id, the join command will be able to join the two tables which is shown in Figure 4.3.

The sqlt-graph tool is particular helpful when one has or is in the process of creating the schema. This tool allows one to be able to see if the join command can be used between the different tables in the database. After the tool is run, one can see from the PNG file or the file that one decides to output, there is lines between different tables to be able to show that these tables can be used to join together since they both have an attribute that can be used to compare to be able to join together.

4.**A brief commentary on the visualizations produced for each of the provided schemas.**

In the schemas, iTrust.sql, there seems to be a large number of tables. Despite that there are a large number of tables, there is only only connection between the different tables. In this connection, the OVReactionOverride tables inside has a foreign key as a constraint called OverMedicationID. This attribute is also a table that is part of the iTrust schema. DUe to the attribute inside the OVReactionOverride table, this causes the foreign key inside this table to be able to be connected to the table of OVMedicationID.

In the schema, Inventory.sql, there also seems to be one table. Since there is only one table inside of this schema, there cannot be any connections on different tables.

In the schema, NistDML181, there are only two tables inside of this sql file. Inside of this sql file, the LONG_NAMED_PEOPLE table is pointing the table ORDERS. This is the case because the ORDERS is the table where it has two attributes that consists of foreign keys. Since the ORDERS table consists of two attributes that have the constraint of foreign keys, the visualization tool of sqlt-graph will have the LONG_NAMED_PEOPLE table point towards the ORDERS table.

In the schema, StackOverflow, there are four tables. Despite there are four tables, there are no arrows or connections pointing to any tables. This is the case because none of the attributes in any table consist of a constraint of a foreign key. Since there are no foreign keys in the attributes as a constraint, none of the tables can be pointed to each other.

In the schema, University, there 10 tables inside. When looking into this schema, there are many different connection going on in this schema. The department tables has arrows has a white arrow pointing toward the course, instructor and student tables. These tables have a foreign key that have different constraints compared to the attributes in the department table which causes the connection between the three table to have a connection with a white arrow. The course table has black arrow that has pointing towards the prereq table. This is the case because the attribute that is being compared, also holds a foreign key. Once a foreign key in an attribute has being compared to another attribute with a foreign key, then the connection will have a black arrow. If a certain table does not have a foreign key constraint, then it is not possible for the table to have a black arrow out-coming of the table. The instructor has two arrows coming out. The two arrows point towards the teaches table and the advisor table. The arrow to the teaches is a black arrow and the arrow to the advisor is white. The student table also has two connections coming out. There are both black arrows

and there are pointing toward to the foreign keys in the advisor table and the takes table. The classroom and time_slot do not have any arrow in-coming or out-coming in this schema.
In the schema, UnixUsage, there are 8 tables inside. IN this schema, there are also many different arrows coming in and out of different tables. The UNIX_COMMAND does not have any arrows in-coming or out-coming. The USER_INFO has three incoming connection for different attributes. The three arrows are white since the three tables of DEPT_INFO, OFFICE_INFO, and RACE_INFO do not hold a foreign key in any attribute. The DEPT_INFO also have another out-coming arrow. This white arrow is going towards the COURSE_INFO table. The COURSE_INFO has one black arrow out-coming. This arrow is pointed toward the TRANSCRIPT table. The USER_INFO has two black out going connections. These arrows go towards the TRANSCRIPT and USAGE_HISTORY tables.