CSE 111 – DATABASE SYSTEMS

Lab 7, October 25, 2012

In this Lab session you will get familiar with the DataBase Connectivity API to access a SQLite DB from a high-level program languages, such as Java, C++, PHP etc. Specifically, below we are going to describe the connection between Java and SQLite using JDBC API.

To connect Java and SQLite database you need:

1. Java JDK version 6. (http://www.oracle.com/technetwork/java/javase/downloads/index.html)
2. SQLite version 3.6 or later (http://www.sqlite.org/)
3. JDBC SQLite driver (http://files.zentus.com/sqlitejdbc/sqlitejdbc-v056.jar)

Then, create a simple Java class and use following commands to connect and control the database from within Java:

- Initialize a class responsible for the connection with SQLite.

```
Class.forName("org.sqlite.JDBC");
```

- Create an instance of the class `Connection` and connect to the SQLite database. Notice, that the connection parameter has 3 fields separated by ":" first is always jdbc, second is ODBC name and third is a path to your database (in our case is the same folder as the main java class):

```
Connection conn = DriverManager.getConnection("jdbc:sqlite:test.db");
```

- The main object that you need to create to control the database is a statement:

```
Statement stat = conn.createStatement();
```

- With a statement object you can drop and create tables or insert values in the existing table:

```
stat.executeUpdate("drop table if exists student;");
stat.executeUpdate("create table student (name, university);");
stat.executeUpdate("insert into student values ('John Smith', 'UC Merced');");
```

- You can get values from the table using the class `ResultSet`:

```
ResultSet rs = stat.executeQuery("select * from people;");
while (rs.next()) {
  System.out.println("name = " + rs.getString("name"));
  System.out.println("university = " + rs.getString("occupation"));
}
rs.close();
```

- At the end of the program always disconnect from the DB:

```
conn.close();
```

When you compile you code, make sure that `sqlitejdbc-v056.jar` is included in your library path.

As a lab assignment you are required to create Java methods that will be able to perform following actions:

1. Connect Java to your TPCH DB from the previous labs.
2. Create a Java method that will add a new table with the name `warehouse` and the following attributes:

```
w_warehousekey decimal(3,0) not null,
w_name char(25) not null,
w_supplierkey decimal(2,0) not null,
w_capacity decimal(6,2) not null,
w_address varchar(40) not null,
w_nationkey decimal(2,0) not null
```

3. Create a method that will ask the user for a new entry in the `warehouse` table. Additionally, `w_warehousekey` should be generated automatically by the system and the user should be able to provide information for the `w_supplierkey` and `w_nationkey` attributes either as an actual key (then the input will be integer) or as a supplier and/or nation name (in this case the input will be string). In the later case the system should automatically find the appropriate `s_supplierkey` and/or `n_nationkey` and set it in the new tuple. The example of a user entry :

```
Name:  Warehouse#000000001
Supplier:  Supplier#000000005
Capacity:  100
Address:  10, Blumenstrasse, Berlin
Nation:  7
```

4. Create a method that will execute the following queries and output the results to the screen:
   - For every supplier find the total capacity of its warehouses.
   - Find the supplier with the biggest number of warehouses.
   - List all the warehouses in Asia with capacity greater than `X`, where `X` is given by the user.
   - For a supplier name given by the user find out whether all its warehouses are able to fit all its products (see `ps_partsupp`).
5. Disconnect from the DB.

All the methods should be controlled from the context menu that should start automatically at the start of the program. The menu should have 5 items, one for each method described above.

Note, that while you are expected to do this lab using Java, you can use any object-oriented language (e.g., because you are using other language for your term project). Here you can find information on how to connect SQLite to other languages: http://www.sqlite.org/cvstrac/wiki?p=SqliteWrappers.