CSCD 327 Project (30 points)
Due: 11:59pm, August 6th, 2014

In this project, you will focus on writing SQL queries. In addition, you will embed your SQL queries into Java (using JDBC) to implement a standalone program that answers several queries about the crime database (*You need to import the database using database_crime.sql*). Please refer to *schema.pdf* to find the partial description of the database.

1. Java Files

You are provided two java files: 'TestMyQuery.java' and 'MyQuery.java'.

TestMyQuery.java

This file provides the main function for running the program. You should only modify three variables (mydatabase, username, and password), replacing them with your own information.

```
String serverName = "localhost";
String mydatabase = "YourDBUserName_crime";
String url = "jdbc:mysql://" + serverName + "/" + mydatabase; // a JDBC url
String username = "YourDBUsername";
String password = "YourDBPassword";
```

MyQuery.java

This is the file in which you need to implement the query functions. Feel free to make any modifications to the file.

2. Queries (30 points)

There are 6 queries in this assignment. The points are evenly distributed (5 points per query). However, the queries may vary in terms of difficulty. If you get stuck on a harder query, try an easier one first, and then come back to the tough one.

Query 1: List the name of each officer who has reported more than the average number of crimes officers have reported.

Here is the correct query result for your reference:

First	Last	cnt
Leigh	Hart	9

Query 2: List the information on crime charges for each charge that has had a fine (*fine_amount*) above average and a paid amount (*amount_paid*) below average.

Here is the correct query result for your reference:

charge_id
5000

Query 3: List all the names of all criminals who have had any of the crime code charges involved in crime ID 10089.

Here is the correct query result for your reference:

first	last
Sam	Phelps
Dave	Caulk
Tommy	Cat
Tim	Simon
Reed	Pints
Nancy	Mansville
Cart	Perry
Penny	Statin
Lee	Panner

Query 4: List criminals (ID and name) who have multiple sentences assigned.

Here is the correct query result for your reference:

criminal_id	last	first	cnt_sentence
1030	Panner	Lee	2

Query 5: List the total number of crime charges successfully defended (i.e, *charge_status = 'GL'*) by precinct. Include only precincts with at least seven guilty charges.

Here is the correct query result for your reference:

precinct	charge_cnt
WAVE	8

Query 6: Write a stored procedure to get the number of crimes reported by an officer.

First you define a stored procedure in *database_crime* named *getNumber*. This procedure takes an *officer_id* as input, and returns the number of crimes reported by the officer as output. In other words, *getNumber*() has one input parameter and one output parameter. Test this procedure in MySQL to make sure it functions properly. Next, your application program asks the user to enter an officer_id (e.g., "111115"), and the program should return the number of crimes reported by the officer accordingly. Here is a snapshot of the output.

```
******** Query 6 ********
Please enter the officer_id for the query:
111115
Officer 111115 has reported 9 crimes.
```

3. Compiling and Running Your Code

For security consideration, the database server is only opened up for access via **localhost**. This means to access your databases, your application program must reside on the same machine as the database server (i.e., ***cscd-327-dev.eastern.ewu.edu***, the port number is **9090**). Use your **Eastern credential** to access the server. I have already installed JDK and JDBC on the server for your application development, but you need to complete one extra task before you start working on your code, i.e. set up the environmental variable *CLASSPATH* properly. If you are off campus, the first thing you need to do is to connect to EWU student VPN network before you make connection to our database server (for more information, please refer to http://access.ewu.edu/it/services/accounts-passwords/vpn).

Use *vim* or *nano* to edit your profile file named *.profile* under your **home directory**. Copy and paste the following line to the bottom of your *.profile* file.

export CLASSPATH=$CLASSPATH:/usr/share/java/mysql-connector-java.jar

Save the file. **Logout the server, and log back in**. Type *javac* to make sure the system recognizes this command. Now you are ready to edit, compile, and run your application code on the server.

- **How to edit**: Your Eastern NetStorage folder has been mapped onto our server. I would suggest you to put all your project files on NetStorage so that you can easily access and edit your source files on a Windows machine (I assume Windows is your preferred operating system).
- **How to compile**: ssh to the server and enter "*javac *.java*".
- **How to run**: ssh to the server and enter "*java TestMyQuery*".

4. Submission

You need to submit your work through Blackboard online submission system. Include the following files into a single .zip file, name it as YourFirstName_YourLastName.zip, and submit this file:

- TestMyQuery.java
- MyQuery.java
- result.txt, result.doc, result.pdf, or result.jpg

You don't need to provide the results in any fancy format, but I hope the results are organized clearly and neatly.