CSE 132A - JDBC Assignment FAQ

1. **How do I use the input1.sql files to set up the database? (i.e. How do I set up pa2.db?)**
   Using the command line, you can do either of the following:
   sqlite3 pa2.db < input1.sql
   OR
   sqlite3 pa2.db
   .read input1.sql
   .exit

   Both of these options will load the “Flight” table into the pa2.db file. You should then use JDBC to open a connection to pa2.db in your PA2.java file using the following path (jdbc:sqlite:pa2.db). After opening the connection to the pa2.db file, you can then use the Flight table in your PA2.java file.

2. **Which JDBC driver do we download?**
   You can use either sqlite-jdbc-3.8.7.jar or sqlite-jdbc-3.8.6.jar; both are fine.
   You can download them here: https://bitbucket.org/xerial/sqlite-jdbc/downloads

3. **I’m getting a ClassNotFoundException when I run my program!**
   Make sure that the .jar file for your JDBC driver is in the same directory as your program files, and also, as stated in the program description, be sure to set the classpath when running your program. For example, if your file is called PA2.java, then you would do the following (also see the program description -- Use Case #1):
   javac PA2.java
   java -cp “.:sqlite-jdbc-3.8.7.jar” PA2 (or whatever driver you downloaded)

4. **I’m getting a UnsatisfiedLinkerError when I try to run my program on the ieng6 server!**
   ieng6 uses SQLite 3.6.20 and JDK 1.8. It may cause exception:
   [cs132a4@ieng6-201]:~$ java -cp .:sqlite-jdbc-3.8.7.jar PA2

   Exception in thread "main" java.lang.UnsatisfiedLinkError:
   org.sqlite.core.NativeDB._open(Ljava/lang/String;I)V
   at org.sqlite.core.NativeDB._open(Native Method)
   at org.sqlite.core.DB.open(DB.java:161)
   at org.sqlite.core.CoreConnection.open(CoreConnection.java:145)
   at org.sqlite.core.CoreConnection.<init>(CoreConnection.java:66)
   at org.sqlite.jdbc3.JDBC3Connection.<init>(JDBC3Connection.java:21)
   at org.sqlite.jdbc4.JDBC4Connection.<init>(JDBC4Connection.java:23)
   at org.sqlite.SQLiteConnection.<init>(SQLiteConnection.java:45)
   at org.sqlite.JDBC.createConnection(JDBC.java:114)
   at org.sqlite.JDBC.connect(JDBC.java:88)

at java.sql.DriverManager.getConnection(DriverManager.java:664)
at java.sql.DriverManager.getConnection(DriverManager.java:270)
at PA2.main(PA2.java:33)

The solution is to use an older JDBC jar. Download it by using the command

wget https://bitbucket.org/xerial/sqlite-jdbc/downloads/sqlite-jdbc-3.8.6.jar

And use this jar instead:
[cs132a4@ieng6-201]:~$ java -cp .:sqlite-jdbc-3.8.6.jar PA2
Opened database successfully.

Statement result:
A---B
B---C

Prepared statement result:
A---B
But this problem is only caused by environmental settings. As long as the program is correct, it should work when being tested on TA's machine (because SQLite3 is guaranteed to work with JDBC jar, the only possible errors are the PA2.java logic itself).

5. **Are we allowed to use a recursive query?**

NO, the assignment specifically says not to use recursive queries (i.e. you can't send a recursive SQL query to the database to compute the answer). As the assignment description says, the control (loop) needed for the recursion can be implemented on the Java side. Use semi-naive algorithm explained in the lecture.

6. **But it says the algorithm shouldn't be written in Java?**

What is meant by that is that you should be computing your answers by sending SQL queries to the database (i.e. you are not allowed to transfer data to the Java side from the database and calculating results there).

7. **Are we allowed to use ResultSet objects that stores the results of queries?**

No, you are not allowed to use ResultSet (please see the Algorithm section of the program description). The idea is that you should be doing calculations for your answer by sending SQL queries to the database. **However**, you may use ResultSet to perform a "SELECT COUNT(*) FROM …" query, as this may be useful in checking if a table is empty or not.

8. **Does the order of tuples in my solution matter?**

No.

9. **Do I need to output the contents of my Connected table at the end of my program?**

No, as long as you have the Connected table (with all the resulting paths put in from your queries) in the database at the end of your program, that will be fine. (Although, it may help to do so for debugging purposes, but make sure to take out this testing output before you turn your program in.)

10. **My program runs pretty slow; is there a time requirement for how long our program can run for?**

As long as you follow the instructions in the program description (you don't use a recursive query and you implement the algorithm on the database side (not the Java side)), it should be fine. The ideal solution takes less than 1 minute on the given sample inputs, so if your programs takes a lot of time on sample inputs, you might have implemented the semi-naive algorithm incorrectly

11. **Should we include tuples in our Connected table where the origin and the destination are the same?**

No, you should not include these tuples in your Connected table (e.g. Boston -> Boston should not be a tuple in your answer).

12. **Will JDBC be covered in lecture?**

No, but JDBC will be explained in detail in the discussion sections.