CSCI 585- Database Systems

Homework Assignment 2

Description

Note: Look at discussion board for clarifications

NOTE: MOSS WILL BE USED FOR CHECKING CODE FROM ALL SEMESTERS.

The goal of this assignment is to design an application that queries a spatial database. This assignment will make you familiar with spatial data types using Oracle10*g*, Oracle Spatial features, and Java (JDBC).
You are required to write two Java programs to 1) store and 2) query your spatial database.

Scenario:
USC installed wireless internet access point, we need a system to keep tracking the location of wireless access-point and people who covered by wireless internet access. Each building can be regarded as polygons.

Input Files:
You will be given the following files:
1. Image file: MAP - an 820x580 JPEG file that is an image of some area of USC.
2. Following input files:
    a). building.xy. Each building is represented by a 2D polygon. Col 1: building ID. Col 2: building name. Col3: number of vertices on the polygon. The numbers after column 3 are the coordinates of the vertices. They are ordered as , , , , ..., , . For example, a row: b1, PHA, 4, 100, 120, 150, 130, 120, 200, 120, 220 represents a building with its building ID as "b1" and its name as "PHA". It has 4 vertices whose coordinates are (100,

120), (150, 130), (120, 200) and (120, 220) respectively.

b). people.xy. Col 1: personID Col2: x coordinate of the person location. Col3: y coordinate of the person location.

c). ap.xy. Col 1: apID. Col2: x coordinate of the access point location. Col3: y coordinate of the access point location. Col4: Radius of access point. People can use the internet if they are within the radius.

Required .sql files:

You are required to create two .sql files:

1. createdb.sql: This file should create all required tables. In addition, it should include constraints, indexes, and any other DDL statements you might need for your application.
2. dropdb.sql: This file should drop all tables and the other objects once created by your createdb.sql file.

Required Java Programs:

You are required to implement two Java programs:

1. populate.java: This program should get the names of the input files as command line parameters and populate them into your database. It should be executed as:

"> java populate building.xy people.xy ap.xy".

Note that every time you run this program, it should remove the previous data in your tables; otherwise the tables will have redundant data.

2. hw2.java: This program should provide a GUI, similar to figure 1, to query your database. The GUI should include:

a) An 820x580 panel that shows the map when the application is started up.

b) The tile of the main window should display your full name and your student ID.

c) Text field (or Label) that shows the coordinates (x, y) of the current mouse location as it moves over the image. *Please notice that the coordinates given in .xy files are based on the origin (0, 0) at the upper left corner of the image and (820, 580) at its lower right*

corner.

d) 3 Check boxes that specify the feature types that we are currently interested in. Multiple feature types can be checked at the same time. They are called active feature types.

e) 4 Radio buttons that specify the kind of query we are going to do. There are 4 kinds of queries: Whole Region, Range Query, Point Query, Find AP-covered people. Details are given later. Only one radio button can be checked at any moment.

f) One button to submit the required query.

g) One text field to display the SQL statements for the queries that has been submitted so far. Use incremental counter for the queries, and print the counter along with the SQL statement (e.g., "Query 1: select * from restaurants;", "Query 2: select * from people where …").

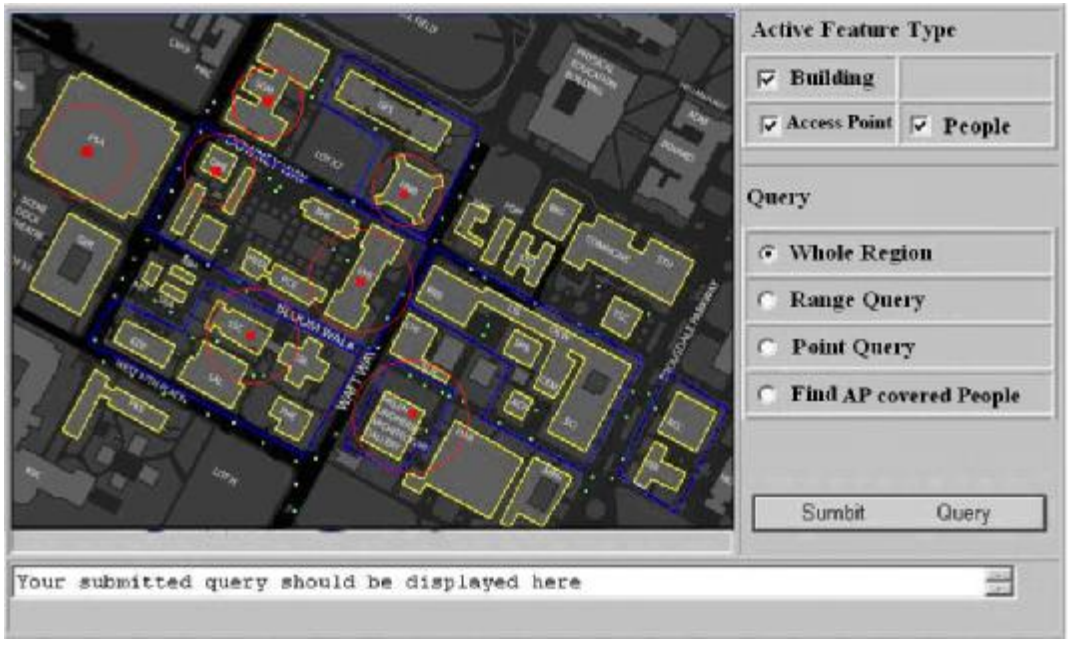

Figure 1

Queries:

1. Whole Region. This is to display all the features of the active feature types in the whole map. They should be displayed in the following way:

Feature	Color	Shape

Access Point	Red	Square (15x15 pixels) and covered area by each AP(access point).
Buildings	Yellow	Polygon (outline, not solid region)
People	Green	Rectangular (10x10 pixels)

Table 1

Graphical representation of People, Buildings and Access points should show up when we check corresponding check boxes and when we click the submit button. Access points also need to show the coverage of an access point by printing circle with radius of each access point from the location of certain access point.

2. Range Query. When this radio button is checked, the user can draw a polygon in the map. After pushing the Submit Query button, only the features of the active feature types that are inside (or intersect with) the polygons are displayed. These features should be displayed in the same way as specified in Table 1. The user draws the polygon by clicking the left mouse button to select its vertices sequentially and then clicking the right mouse button to close the polygon. The vertices should be connected by red line segments on the screen as they are being selected.
When the Range Query radio button is unchecked, the selected polygon should disappear.

3. Point Query. When this radio button is checked, the user can select a point in the map. This point is displayed as a red square (5x5 pixels). You should also display a red circle centered at this point whose radius is 70 pixels. After pushing the Submit Query button, only the active features that are inside (or intersect with) of the circle will be displayed. Their shapes are specified in Table 1. Their colors are as follows: for each active feature type, the feature that is nearest to the selected point among all the features of this type inside the red circle is displayed in yellow. All the other features are displayed in green color. When the Point Query radio button is unchecked, the selected point and the associated red circle should disappear.

4. Find wireless internet-covered people by an access point. When this

radio button is checked, all the features of all feature types in the map should be displayed (in the way specified in Table 1). The user can select an access point by clicking the left mouse button to select a point in the map. The access point that is nearest to this point is selected and is displayed as a blue square (15x15 pixels). Please disregard blue lines in Figure 1. After pushing the Submit Query button, multiple yellow lines between AP and the people who are covered (within radius) by this access point will be displayed. Also people who located "radius + 5" (a bit out of coverage) will be displayed in blue colored line. (Blue colored people means they have weak AP signal reception). Also people who located "radius + 10" will be displayed in Cyan colored line. Also Coverage radius of each access point is stored in ap.xy file.

You can disregard blue region in Figure 1. And your GUI work does not need to be exactly same as figure 1 however; it should have required components such as radio button, check box, text box and query button. And covered area (Red circle) in Fig. 1 is not exactly match with exact radius in ap.xy file.

Submission Guidelines

1. The links to the document for Oracle Spatial Reference are posted on the course website. The map image and the 3 input files are provided on the website.

2. Oracle JDBC Driver and Spatial Java APIs:
Oracle Spatial Java Library (sdoapi.zip), is posted with this description. It is required to manipulate spatial objects of Oracle10g (or higher) in Java program. Oracle Spatial Java API document is also posted.
You can compile your source as:

$ javac –classpath
.:$ORACLE_HOME/jdbc/lib/classes12.zip:$ORACLE_HOME/md/lib/sdoapi.zip hw2.java

You run your application as:

$ java –classpath
.:$ORACLE_HOME/jdbc/lib/classes12.zip:$ORACLE_HOME/md/lib/sdoapi.zip hw2

3. You need to have a readme.txt file that should include your name, student id, your user name on aludra.usc.edu, the list of the submitted files, , resolution of your homework and how to compile/run them. There is 20 points penalty if this file or some of the required information is missing from your submission.

4. For the second Java program (i.e., hw2.java), you may develop your assignment using more than one Java program. It is recommended (but not required) to separate the GUI codes and database related codes into different files.

5. You must make a xxx.tar (or zip) file to include all of your files in one file (e.g., hw2.tar or zip) and the compressed file includes **hw2.java createdb.sql dropdb.sql readme.txt or**
Do NOT include the .class files, input files, or sdoapi.zip in your .tar file. We will compile your .java files. Supplied sample codes do not have correct address for your Oracle. Please modify oracle addresses.

6. You need to submit the assignment electronically to den.usc.edu. Please make sure you pushed submit button.

7. You can write your Java programs on any machine you wish. You can use any Java Visual Programs (e.g., JBuilder, Visual Café, Visual J++) you wish to design your GUI, but make sure you can run your program from university computer which does not requires specific link or classpath before submitting it. Again please do not put absolute path for the linking and include files for grading. Otherwise it will be penalized.

8. Start working on your assignment early.

9. Grading guideline:

Points	
5	Creating/Dropping database tables.
10	Populating database

20	GUI containing all of the requirements mentioned for user interfaces. Hint: implement DB connectivity & queries first. If time left, move to GUI construction.
10	Whole Region
10	Range Query
10	Point Query
20	Find AP-covered people
10	A bit out of radius people
5	Out of radius people (very weak signal)

10. Again, please start early. Try to submit early. Any submission related issues should be tested/resolved before your submission. Do not try to submit the code right before the due. No late submissions will be accepted in any reason.