

Description

**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 Oracle11g, Oracle Spatial features, and Java (JDBC).

You are required to write two Java programs to 1) store and 2) query your spatial database.

Prerequisite:

It is highly recommended that students be familiar with JDBC/ODBC concepts, Spatial database and writing spatial queries.

Scenario:

A new Public Announcement System is getting installed at the campus. We need a system to keep track of all the area that this new announcement System can cover. Each Announcement System is represented as a point with a radius for coverage and each building is represented as a polygon.

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). buildings.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 comma separated. 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).students.xy

Col 1: personID Col2: x coordinate of the student location.
Col3: y coordinate of the student location.

c). announcementSystems.xy

Col 1: asID. Col2: x coordinate of the announcement system location. Col3: y coordinate of the announcement system location. Col4: Radius of announcement system. People can listen to the announcements 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 buildings.xy students.xy announcementSystems.x”. 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 title 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) 5 Radio buttons that specify the kind of query we are going to do. Only one radio button can be checked at any moment. When a different radio button is selected, the screen should clear the previous results.

f) One button to submit the required query.

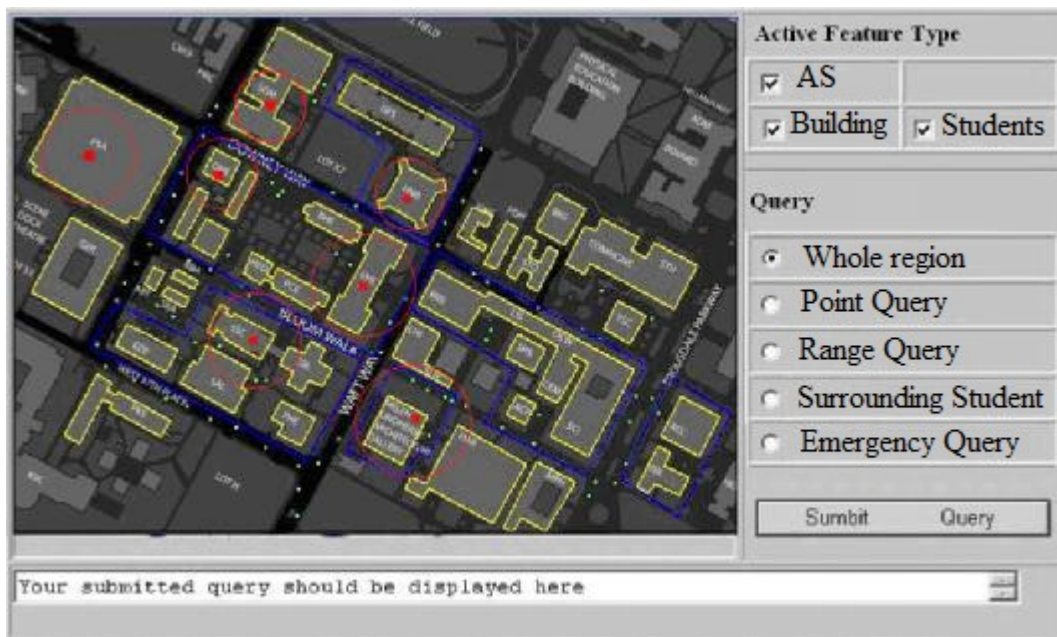
g) One text field to display the SQL statements for the queries that has been submitted so far. Use incremental counter for the queries, and print the counter along with the SQL statement (e.g., “Query 1: select * from restaurants;” , “Query 2: select *

from people where ...”).

Table 1:

Feature	Color	Shape
Buildings	Yellow	Polygon(outline, not solid region)
Students	Green	Square(10X10 pixels)
Announcement Systems	Red	Square (15x15 pixels) and circle indicating covered area by each AS(announcement system).

Figure 1: GUI demonstration



Queries:

1. 10 Points

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:

Graphical representation of Students, Buildings and Transmission points if checked should show up when we click the submit button.

2. 10 Points

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 50 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.

3. 15 Points

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. Red line segments on the screen should connect the vertices as they are being selected. When the Range Query radio button is unchecked, the selected

polygon should disappear.

4. 15 Points

Surrounding Students Query: When this radio button is clicked, the user can select a point in the map. The nearest Announcement System should be highlighted at this time. When the submit query button is pushed, the students in the region of the highlighted AS should be displayed.

5. 20 Points

Emergency Query: Imagine a AS breaks down. The students near this AS cannot hear the announcements and hence need to go to the next nearest AS. You need to help them.

When this radio button is clicked, the user can select a point in the map. The nearest Announcement System should be highlighted at this time, indicating that this is the AS that is broken. When the submit query button is pushed, the students that were covered by the broken AS system should be color coordinated with the remaining announcement systems, i.e. a student should have the same color as its second nearest announcement system. You can use any color for the announcement system here, but all AS should have different colors.

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 (ojdbc6.jar), is posted with this description. It is required to manipulate spatial objects of Oracle11g (or higher) in Java program. You can compile your source as:

```
$ javac -classpath /path/to/your/ojdbc6.jar hw2.java
```

You run your application as:

```
$ java -classpath /path/to/your/ojdbc6.jar hw2
```

3. You need to have a readme.txt file that should include your name, student id, your user name, the list of the submitted files, resolution of your homework and how to compile/run them. There are 25 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** Do NOT include the .class files, input files, or ojdbc6.jar in your .tar file. We will compile your .java files.

6. You need to submit the assignment electronically to blackboard.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. 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	Tasks
5	Creating/Dropping database tables
10	Populating database

15	GUI containing all of requirements mentioned for user interfaces. Hint: implement DB connectivity and queries first. If time left, move to GUI construction
10	Whole region
10	Point Query
15	Range Query
15	Surrounding Students Query
20	Emergency Query

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.