CSE3421 Fall 2007

Assignment #2: Due November 9, 2007 3 pm.

Weight 5%

This assignment can be completed in groups of up to 2 students per group.

In this assignment you are requested to translate some queries from English to SQL and then to build a Java application that employs JDBC and submits those queries to DB2.

Part A

(Task A.1.) Write in SQL the queries given in Appendix B and submit them to DB2.

(Task A.2.) Submit the queries to DB2, via a java program that uses JDBC. Name your java program `A21.java`.

Part B

(Task B.1.) Create a Java program, **A22.java**, which allows inputs for the following:

a. *Offered Price From* .. *Offered price To* – to be used in query 1 (e.g., 200000 .. 220000) – see task B.2
b. *Property type* – to be used in query 1 (e.g., townhouse). see task B.2
c. *Max willing to pay for real estate* – to be used in query 1 (e.g. 250000). see task B.2
d. *Listed price from* .. *Listed price to* – to be used for query 2. (see task B.2)

(Task B.2.) Use your java program (A22.java) from task B.1 to assemble queries like the ones of appendix B, by substituting the input values as set by the user using the drop-down lists of task B.1, into the queries of Appendix B, as applicable. Your A22.java should contain a GUI, featuring the drop-down lists, per task B.1, as well as four buttons named Q1, Q2, Q3a, Q3b, corresponding to the queries of appendix B. After setting the input values the user can click to any of the four buttons and the corresponding query should be executed and the results displayed in a text area, reasonably formatted.

Details:

1. For task A.1, write the queries into a file (single file, or many files) and then submit them to DB2 and receive the results. Save the results into a file named `A1.txt`. In order to run your queries you would need a database with tables filled in with data. Use the database that you created in your assignment 1. You may need to modify your programs of assignment 1 and rerun them to populate the database, so that the database is populated with data appropriate to illustrate the workings of your queries in this assignment. Note that for the queries Q4a and

Q4b you may even need to restructure your database so that is facilitates extraction of the month; you can do so by converting the address attribute into entity (and then associating attributes including an attribute month to that entity)[1], or by simply parsing the address attribute value and extract the moth value.

2. For task A.2, run your A21.java program and save the results of the queries into a file named `A2.txt`. Use the same database that you used for task A.1.
3. For task B.1, the inputs should be allowed via drop-down lists (that display choices and the user can select). In order to make this assignment testable, you should not let the user type any values since the given values may not make any sense to the values that you have entered into the DB that you created. Instead, the values displayed in the drop down lists should be the ones that are contained in your own DB. Name your java program `A22.java`. For input (a) -- *offered price from …offered price to* range, check for correctness that the "from" price is no greater than that "to" price. The "from" price and "to" price could be equal however, which would mean that the user is interested about a specific offered price rather than a range of prices. For input (b), apparently the only choices that should appear should be "apartment", "townhouse" and "house".
4. For task B.2, display the results of the executed query in a viewer-friendly format (the design is up to you). Make the text area scrollable, as applicable, and large enough. Note that not all input values ((a), (b), (c), (d) from task B.1) may be required for each query, nevertheless you should allow the user to enter all values, in task B.1. After clicking one of the query buttons and executing a query, the user should be able to click another query button and execute another query, using the same input values that were set in the drop-down lists. Note that for some of the selected values, a query may give empty output. In such a case, your program should display a related message on the GUI and also indicate which ones of the four inputs ((a), (b), (c), (d) from task B.1) might have caused this.
5. Use only the standard Java SDK libraries and not any third party packages. (your programs should compile and run without the need to have any add-on packages).

What to submit

Hand in the following items:

1) A **hardcopy** of your `A1.txt file.`
1) A **hardcopy** of your `A2.txt file.`
2) A **hardcopy** of your `file(s)` that contain the queries in SQL, as you did them for task A.1.
3) **Hardcopies** of screenshots that illustrate the workings of your assignment. These should include:
    a. Screenshots of running your A21.java.
    b. Screenshots of running your A22.java and setting the inputs. Use a variety

[1] This approach would obviously entail modifying your ER diagram and also modifying the programs you developed in assignment 1 so that they create appropriate tables and populate then according to the new diagram.

of runs to demonstrate all the workings of your assignment (for example, cases that *from* and *to* are different, i.e., a range, as well as when they are the same). Provide an outline of the test cases that you choose to show.

   c. Screenshots of running A22.java and executing each of the queries.
   d. Screenshots of the process of compiling your A21.java and A22.java programs; and screen shots of the directory showing the .class flies after the compilation.

4) **Electronic copies** of all the above, i.e.:
   a. **A1.txt** file – contains the results of the queries of Task A.1.
   b. **A2.txt** file – contains results of queries of Task A.2.
   c. A file named **queries.sql** – contains the queries, as done for Task A.1.
   d. **A21.java** file – program done for Task A.2.
   e. **A22.java** file – program done for Task B.1.
   f. The files **A11.java**, **A12.java**, **createTables.sql**, **insertData.sql**, from assignment 1. These files may be identical to the ones you submitted for your assignment 1, or you might have changed and tailored them for this assignment. Submit them with this assignment, either way.
   g. A file named **screenshots.doc** (MS Word format). This file should contain the screenshots that you accumulated, as is described for the hardcopies that you have to submit. Inside this file also clearly indicate what each set of screenshots represents.

How to submit your assignment.

- Please drop off your assignment (paper copy) in your section's CSE3421 drop-off box in the Computer Science Building (ground floor; across the Prism labs).
- Also, submit your electronic copies using the submit command. The following submit statement should do this:

```
submit 3421 a2 A1.txt A2.txt queries.sql A21.java A22.java A11.java
A12.java createTables.sql insertData.sql screenshots.doc
```

What happens if I miss the deadline:

If you miss the due date/time, but hand it in within 24 hours of the due date there will be a 20% deduction of your score for penalty (e.g. if your original score is 80%, you will end up getting 64%). After that, no late assignment will be accepted and a mark of 0 will be recorded.

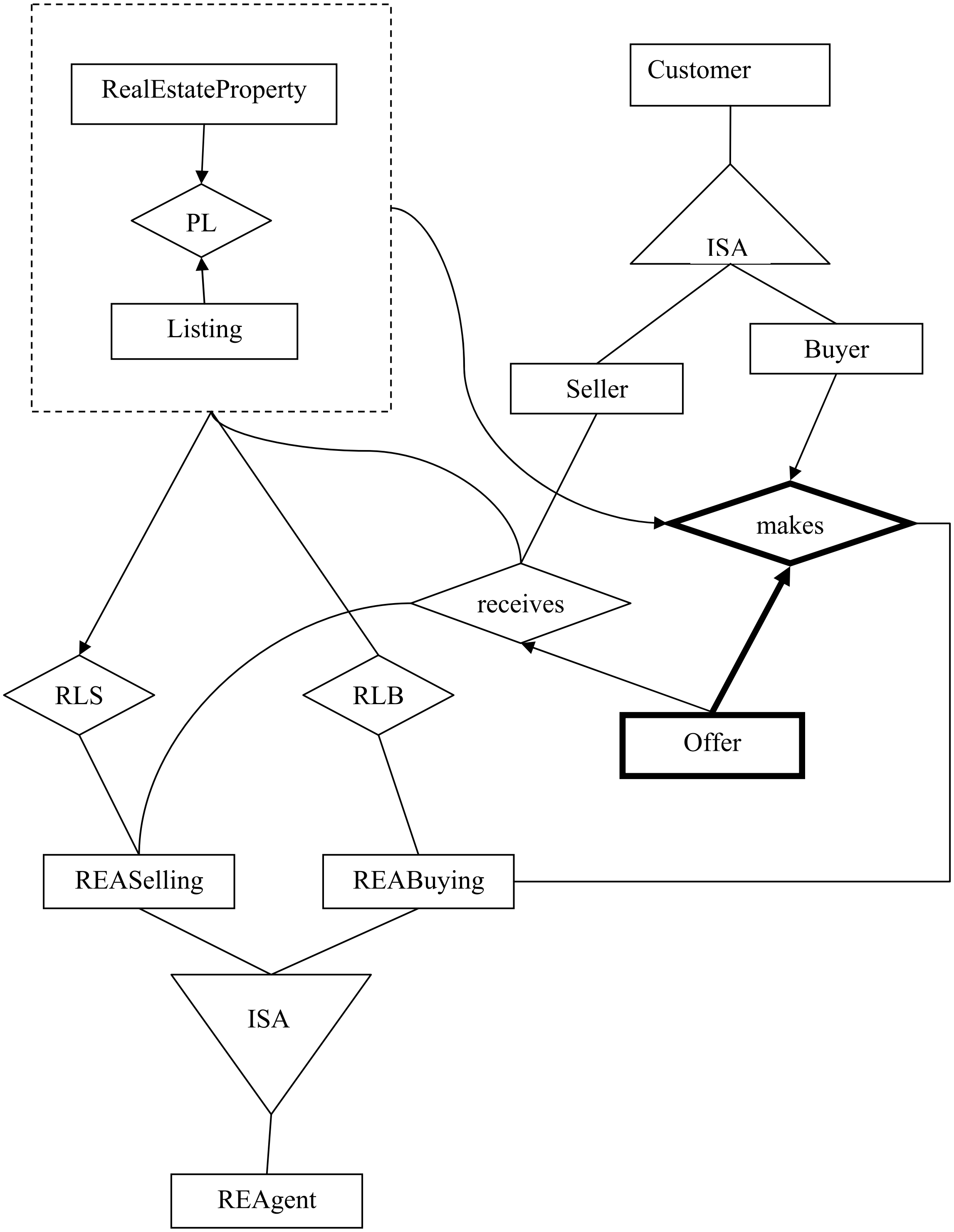
Use the following cover page for submitting your hardcopies, filling out your student number etc as follows:

CSE3421 Fall 2007 Assignment 2 (cover page)

	Student 1	**Student 2**
Student Number		
CS#		
Last Name		
First Name		
Email Address		

Appendix A – this is the same as on assignment 1 and it is given here for completeness regarding the semantics of the queries of Appendix B.

(ERD)



Attributes

The following attributes are associated with the entities and relationships in the above diagram:

Entity (or relationship)	*Attributes (**keys** are in **bold**)*
REAgent	**reaID**, name, address
Customer	**sin**, name, address
Seller	minPossibleSellingPrice[2], legalSellingFees
Buyer	maxPossibleOfferingPrice[3], closingCost
Offer	**oID**, offeredPrice, date, conditions
RealEstateProperty	**address,** type (apartment, townhouse, house)[4]
PL	price
Listing	**listingID**, date
REASelling (the selling real estate agent)	feeAmount, feePercentOnSoldPrice
REABuying (the buying real estate agent)	feeAmount, feePercentOnBoughtPrice
RLS	- (no attributes)
RLB	- (no attributes)
Receives	date
Makes	date

[2] It is assumed that the Seller has in mind a minimum possible selling price (which is not necessarily the same as the listing price). During negotiations of the offers, a seller may decrease the selling price and the value of the minPossibleSellingPrice is the lowest price that a seller would ever accept to negotiate.

[3] It is assumed the Buyer has in mind a maximum possible price (which is not necessarily the price that is indicated in an initial offer). During negotiations of the offer, a buyer may increase the offering price and the maxPossibleOfferingPrice is the highest price that the buyer would ever agree to offer.

[4] A property can only be either an apartment, or a townhouse, or a house, and nothing else.

Appendix B
(the queries)
The following queries are based on the ER diagram of Appendix A (which is the same diagram as the one of assignment 1).

(Query 1.) Find the names of all customers that have made offers of at least $200,000 but no more than $220,000 for a townhouse, and are willing to pay up to $250,000 (plus closing costs) for a real estate purchase (any type of real estate, not just townhouse).

(Query 2.) Find the names of all real estate agents who serve both as selling and as buying agent (for listings and ongoing offers) or real estate properties with a listed price between $400,000 and $500,000.

(Query 3.) Find what is the

a. busiest month in property listings
b. least-busy month in property listings,

across all years.