CS 550 - Database Systems

Fall 2016

Instructor: Dr. Jessica Lin

Project Assignment

General. Your project is to build an online auction system called **SimpleAuction**. At the back-end, SimpleAuction manages all of its data in the Oracle database management system. At the front-end, you are required to implement a command-line interface to its users using Java and JDBC (although if you love challenge and want to build a web interface, you can do so for extra credit). SimpleAuction is modeled roughly after the eBay online auction system.

I will expect certain minimal functionality in each SimpleAuction system - beyond that, the sky's the limit. Minimal functionality includes a variety of queries and browsing capabilities over the current listings (the users should be able to search by product name, seller ID, etc.), the bids on those items, and the sellers and bidders. SimpleAuction also must provide a means for entering bids and ratings/reviews, maintaining average ratings, and concluding auctions (by this we mean change the status from "active" to "completed", and determine the winning bid). Various integrity constraints must be monitored.

The followings are the "basic" information that the database should store:

1. Customers: Users of the auction site can be sellers and/or buyers. A user can list one or more products for auction, and he/she can bid on one or more products. Obviously, a user cannot bid on his/her own listing. Each user has a unique ID, a name, age, gender, date of joining, and average rating

2. Product: A product has a unique product ID, product name, and category. Each product is classified into one or more categories such as Electronic, DVD, Book, Music, Apparel, etc. Each category is given a label (like those just given). Note that a product is not a unique *item*. For example, for the DVD "Star Wars: Episode I – the Phantom Menace," it has a product ID, but there may be multiple listings on the same product, by the same or different users.

3. Listing: Each listing has a unique listing ID, ID of the product that the listing sells, ID of the user that listed the product, item condition ('new', 'used'), minimum starting bid, status ('active', 'completed'), auction start time and end time.

4. Bid History: Each listing has a bid history, which contains listing ID, bidder ID, bid amount, and timestamp. The first bid should equal the minimum starting bid. A bidder can bid on a listing more than once. Each bid must be higher than the current highest bid.

For each completed listing, the winning bidder (the buyer) can rate and comment on the seller, and the seller can rate/comment on the buyer. These ratings will be used to calculate the average rating for each user.

When users search for listings by product name, the search should retrieve all listings for which the product names contain the search terms, e.g. if the search term is "Harry Potter", then all listings with product names containing "Harry Potter" should be retrieved.

Note: The descriptions above summarize the minimal information your database should contain. It does not necessarily imply the organization of the relations. You may re-arrange the attributes or add more attributes. For example, you might decide that it's better to separate the active listings and completed listings into different tables, or not. The design decision is yours, but your design should be reasonably efficient, and well justified. You will be graded on both correctness and quality of the design.

You'll be graded for each phase. The project accounts for 15% of your total grade.

Phase 1 (2%): Conceptual Design. You need to submit a corresponding ER diagram according to the project description and the assumption you have made.

Phase 2 (4%): Schema Design & Database Implementation. After your conceptual design is finished, you need to translate the ER diagram into relation schema. For this phase, submit the following:

(1) your relation schema based on your (revised?) ER diagram
(2) SQL script that you use to implement your database, including all necessary triggers (e.g. to check the validity of bids, to update average user rating, etc.). Insert enough tuples into each table. You should have at least 10 users and 10 listings, and some bids for the listings
(3) In the same script, write the following SQL queries:
    a. Display completed listings with no bid
    b. Find the highest rated user
    c. Find user with the highest number of active listings
    d. Find the most expensive item sold
    e. Find the most popular item (with the most bids)

Phase 3 (7%): JDBC Implementation. Write a program in JDBC that will allow users to access and query your database. Command-line menu is acceptable. More information on this phase will be given later in the semester. In general, your program should allow users to do the following:

1. View table content: Give user a list of existing tables so that he/she can select one to view the tuples.
2. Add records: Enter information for new customers (buyers/sellers), listings and bids. Update/delete information on customers, listings and bids.
3. Search database: search listings based on product names or seller ID; search for specific user and show comments as well as ratings; search for specific user and show average rating; show all bids for a given listing

Extra credit (2%): Instead of command-line interface, you can create a graphical user interface (GUI) for extra credit.

Putting it together. (2%) Prepare a final report. Your report should contain the following:

Put the following in one PDF file:

- README describing how to run your program
- Screenshots on program/query execution
- Write-up: Discuss and justify your design decisions and challenges you encountered. List any part(s), if any, that do not work properly or you did not implement.
- Contribution of each member in the team.

Additional files to submit:

- Java source code
- Script file used to create/populate your database

Timeline:

10/4: project released

10/27: Phase 1 (ER diagram) due

11/10: Phase 2 due

12/8: Phase 3 and final report due