Term Project - Online Auction Database

SWE3003 Database Systems - Fall 2020

Due date: December 4 (Fri) 11:59pm

1 Purpose of the Project

You are requested to analyze the requirements for the design, implementation, documentation, and testing of a database system for an online auction site. To build a web site, you need a client-side development team including web designers (often called UI designers) focusing on the design aspect of the site and front-end developers who code user interfaces with HTML, CSS, JSP, JavaScript, etc. While the front-end development is working on how your web site looks like, the back-end developers create core services that deal with business logic, data modeling, databases, file servers, cloud services and many more.

In this project, your role is a back-end developer. Unfortunately, your company has not hired any front-end developer yet, so you need to implement the auction application without GUI-based menus but with a text-based menu-driven application. For example, you need to write a main function that shows menus as follows.

```
$ ./a.out
----(1) Login
----(2) Sign Up
----(3) Login as Administrator (Manager of Auction System)
----(4) Quit
    Your choice: 1
    enter your ID: bnam
    enter your password: mypassword
```

This is what your main function should print out in a while loop. If a user selects one of the choices, its corresponding function should be called. As a walking example, if a user selects (1), you should call `login()` function, which asks you to enter ID and password. After the ID and password are verified, the login function should print out the next choices as follows.

```
----(1) Sell Item
----(2) Check Status of Your Item Listed on Auction // (Seller menu)
----(3) Buy Item
----(4) Check Status of Your Bid   // (Buyer menu)
----(5) Check Your Account
----(6) Quit
    Your choice: 3
```

If a user selects a menu you should call its corresponding function, which prints out the next level menu as follows. If a user selects Go Back menu, you return to the caller so that the previous level menu can be shown.

```
----How do you want to search items?
----(1) By category
----(2) By keyword
----(3) By seller
----(4) By date
----(5) Go Back
----(6) Quit
    Your choice: 2
    enter keyword: monitor
```

Please note that UI/UX is not your main concern in this project since you are a back-end developer. Later, when a front-end developer joins your company, he/she will replace your text-based interface with a front-end web site. You are going to learn how to design web interfaces using HTML5, CSS, JavaScript, etc in another course, not this database course.

For the back-end web application development, Java is known to be the most popular programming language. So, for this project, your company asked you to implement the back-end modules using Java and JDBC.

2 Requirement

The following descriptions are just a basic outline of requirements. Detailed and additional features should be designed and implemented by you. Make sure that these additions are appropriately documented in your report.

2.1 Data and Queries

A seller lists an item in your auction system, and buyers browse and purchase it. This is the basic functionality your system should support. For this basic functionality, your database needs to manage at least but not limited to the following data. Please note that the followings are a quick sketch of a data model that your application needs to revise and refine. Please note that you would need to decompose the following data model to make it satisfy one of the normal forms (3NF, BCNF, etc).

- **User Data**

  name, e-mail address, seller rating, number of items sold, number of items purchased, watch list, etc.

- **Item Data**

  item category (used for category search), item description (used for keyword search), condition (new, like-new, very good, good, acceptable), most recent bid price, buy-it-now price, status, number of bids, bid history, date posted, bid ending date, etc.

- **Transactions and Billing Data**

  sold item, purchase date, sellers, buyers, amount due buyers need to pay, amount of money sellers need to get paid, invoice per sold item, profit of auction company, etc.

The business model of your auction company is as follows. The people who sells an item, pays a commission fee - 2% or 1% of the item price to the company. If a seller sold two or more items this month, the seller gets a 1% discount for the commission fee from the third item. I.e., the seller pays for 1% of the item price as a commission for 3rd, 4th, and more items sold.

When the bidding is over, your DBMS should trigger a temporal trigger, also known as a MySQL event, which determines who is the winner and charges the winner the price of sold item. In this project, you are not going to implement a payment system. Instead, your application will show a monthly bill for each user, i.e., there should be a menu that shows how much money a user (buyer) has to pay to the auction company and how much money the company owes to the user (seller). The monthly bill must show the summary of transactions, i.e., the list of sold or purchased items, the item prices, the commission fees, and so on.

- A buyer can browse auction items by category, by keyword, by seller, and by date. You should show the search results in a well-formatted table using `System.out.printf()` method. The table must include when the auction ends, the current highest bidding price, the current higher bidder, and the buy-it-now price.

- A seller should be allowed to view the list and the status of his auction items.

- Administrators should be able to view the list and the status of all auction items. In addition, administrators should be allowed to view general statics, such as profits of the company, how many items were traded per category, and many other interesting queries that you should come up with. You must implement at least 5 analytical queries for administrators.

3 Report

A well-formatted report should be handed in for grading at the end of semester. The report should consist of the followings.

- Schema diagram using E-R model
- List of attributes of each entity and relationship
- DDL statements
- DML statements for queries
- User manual for each function

4 How to Submit

Submit your source code files using db_submit command in 'swji' node as follows.

```
$ db_submit term project.tar.gz
```

You should compress your source codes using 'tar czvf project.tar.gz {source_code_directory}' command. Note that you can submit multiple times. But only the last submission will be graded. Using the following command, you can check whether your file has been correctly submitted.

```
$ db_check_submission term
```

For any questions, please post them in Piazza so that we can share your questions and answers with other students and TAs. Please feel free to raise any issues and post any questions. Also, if you can answer other students' questions, you are welcome to do so.