

DATABASE SYSTEMS TERM PROJECT

CS/SE 6360 – Fall 2009

I. OVERVIEW

Motivated by the 50% increase in oil prices in the last six months, a good friend of yours would like to set up a local shop that can sell oil to investors in Dallas. She asked your help to develop software to assist traders working in her company in managing oil transactions issued by customers. In particular, she wants you to create convenient and easy-to-use software for oil traders who are trying to buy and sell oil for their clients. To help your friend, you offer to develop a software system called OTS (Oil Transaction System) that is based on a relational DBMS and modern technologies such as JDBC.

II. PROJECT DESCRIPTION

Based on your interactions with your friend, you gathered the following pieces of information:

- OTS will be used by the traders to buy and sell oil for their clients.
- Each client has a unique client id generated by the system, a name (first and last), a phone number, a cell-phone number, an e-mail address, and an address (including street address, city, state, and zip code).
- Since sometimes purchased oil could be shipped to clients, it is important to retrieve the city and zip code information for each client easily.
- Each client is assigned to one of two different levels based on his or her past transaction volume. Once a client trades more than 30 barrel in any single month, the client is classified as a “Gold” customer and is charged a different commission rate for all subsequent transactions. Otherwise, the client is classified as “Silver”.
- When a client wants to make a transaction, the client calls the trader and specifies the amount of oil he/she wants to buy or sell. (Assume that the automated call system verifies the client’s identity by asking the client to enter a password.) If the client wants to sell oil, the system should automatically check if the client has enough oil stored by the company to satisfy the client’s request. The client also needs to specify whether he or she wants to pay the commission for the transaction in oil or cash. Based on the client’s choices, the trader places the order. The system calculates the transaction commission based on the client’s

classification. If the transaction fee is paid in oil, the system automatically adjusts the amount of oil left in the customer account. On the other hand, if the customer chooses to pay the commission in cash, the system must automatically compute the fee based on current oil prices.

- The value of the transaction (e.g., the value of the oil bought or sold), the date of the transaction, and the commission paid should be stored separately for each transaction.
- From time to time, clients will pay money to settle their transaction costs. For each payment transaction, you need to store the amount paid, the date, and the information related to the trader who accepted the payment
- In some cases, traders may want to cancel certain payment and oil transactions. Although the system should allow such cancellations, logs should be stored for such cancelations for auditing purposes.
- In the final phase of the software, you need to have a GUI where a trader can issue transactions for a client, can search the client history for specific client based on name, address and etc. In addition, you should provide a GUI for the manager that can give aggregate information for daily, weekly and monthly total transactions based on the dates entered by the manager.

Please note that this document does not claim to be complete. Many design issues need careful analysis. Some of these include: how user history will be stored, what attributes entities should have, how users are upgraded to “gold” category. Yet, given this description, filling in the blanks should be easy.

III. PROJECT STEPS

Based on the descriptions given above, you decided to decompose the project into the following steps:

PRELIMINARY STEP. Form a group of size 3 and notify the TA Mustafa Canim (canim@utdallas.edu) before Sept 30.

STEP 1. (30%) Draw the ER/EER diagram, and then convert it to a relational schema. Indicate primary keys, foreign keys and any other constraints. Follow the notation used throughout the textbook. Clearly specify any assumptions you make and your rationale. Submit all your work via WebCT in **JPEG, DOC or PDF** format by **Oct 14 Midnight (CST)**.

You will not work on your ER/EER diagram at Step 2. We will provide you with the solution for Step 1 through WebCT on Oct 15.

STEP 2. (30%) Create database tables according to the relational schema in Step 1 and populate them with a reasonable number of tuples. Also include a series of ‘drop table’ queries that will drop all tables created.

Submit the SQL commands via WebCT as a **TXT** file by **4 Nov Midnight (CST)**. Any line that is not part of an SQL command should be commented out. In other words, you need to submit an SQL script. Please notice that although SQL is a standard, query syntax differs by DBMS vendors. Your script should run on the campus SQL Server without any errors.

You will not work on your relational schema at Step 3. We will provide you with the solution for Step 2 through WebCT on 5 Nov.

STEP 3. (40%) Design and implement a program for end-users. This program should connect to the database and provide the functionalities discussed in the project description. The GUI for this program should be simple and easy to use.

You may use any programming language you wish, but we will give support for only Java. The basics of JDBC (Java Database Connectivity) will be discussed using a simple application in class.

Submit your source code and executable as a **ZIP** file via WebCT by **25 Nov Midnight (CST)**. A demonstration schedule will be created for you to present your work.

IV. SUBMISSION GUIDELINES

Please read the information below carefully. Compliance with these guidelines is a vital part of the grading process.

- Please pay attention to submit your files in proper format, specified for each step. Only one submission per group is sufficient. Include names and NetIDs of all members in every file.
- Do not submit at the last minute. Your clock most possibly is not synchronous with the WebCT system clock. Unless there is a good reason (i.e. WebCT system crash), only online submissions will be accepted. In order to encourage submitting partial work, multiple submissions are allowed. You can always retrieve your files, make necessary changes and re-submit until the due date.

- ❑ The TA is *not* a system administrator. Please direct all questions related to connection problems to cs-tech@utdallas.edu .
- ❑ Source code will be tested using software plagiarism tools. Plagiarized work is very easy to detect, and all necessary measures will be taken to identify and penalize such behavior.

GOOD LUCK!