**CSE 5330-002 SPRING 2022**

**Project #2 Description**

In this project, you will design and implement a database for a Bank Enterprise to reflect the following scenario. You will first design an ER/EER schema diagram for this database application. Then, you will map the EER schema into a relational database schema and implement it on ORACLE/MySQL in Omega. Finally, you will load some data into your database, and create some queries and update transactions.

The bank is organized into branches. Each branch is located in a particular city and is identified by a unique name. The bank monitors the assets of each branch.

Bank customers are identified by their Social Security numbers. The bank stores each customer’s name, and the street and city where the customer lives. Customers may have accounts, and can take out loans. A customer may be associated with a particular banker, who may act as a loan officer or personal banker for that customer.

Bank employees are also identified by their Social Security numbers. The bank administration stores the name and telephone number of each employee, the names of the employee’s dependents, and the Social Security number of the employee’s manager. The bank also keeps track of employee’s start date and thus, length of employment.

The bank offers two types of accounts - savings and checking accounts. Accounts can be held by more than one customer, and a customer can have more than one account. Each account is assigned a unique account number. The bank maintains a record of each account’s balance, on the most recent date on which the account was accessed by each customer holding the account. In addition, each savings account has an interest rate, and overdrafts are recorded for each checking account.

A loan originates at a particular branch, and can be held by one or more customers. A loan is identified by a unique loan number. For each loan, the bank keeps track of the loan amount and the loan payments. Although a loan payment number does not uniquely identify a particular payment among those for the all bank’s loans, a payment number does identify a particular payment for a specific loan. The date and amount are recorded for each payment.

In this banking enterprise, the bank would keep track of deposits and withdrawals from savings and checking accounts just as it keeps track of payments to loan accounts.

You will first design ER/EER schema diagram based upon the above Bank database requirements and create the schema diagram and documentation report describing your design choices. As part of this assignment, you should identify any missing or incomplete requirements, and explicitly state them in your documentation. You should also explicitly state any assumptions you made that were not part of the requirements listed above.

The second part of the assignment will be to map the EER schema design to a relational database schema, and create the tables corresponding to the relational schema using the ORACLE/MySQL DBMS. You will add to your report a listing of the CREATE TABLE statements. Specify as many constraints (key, referential integrity) as you can in the relational schema. You should state the choices you made during the ER/EER-to-relational mapping, and the reasons for your choices.

The third part of the project is to load some data into the database and apply certain update transactions and retrieval queries. You will create your own data that needs to be sizable.

For transactions, create a user friendly interface to enter the information needed by the transaction. This can either be a Web-based interface, a command line interface, or a forms interface

The following are the tasks for the third part of the project: (TBD)

1. Load some initial data (as discussed above) into the database tables that you created in Part 2 of the assignment. Write a suitable loading program, or use SQL/PLUS (insert command), or use SQL/FORMS. Your data should be kept in files so that it can easily be reloaded during debugging. The data format should be designed by you. (Note: You can also use the transactions created by you in item 3 below to load some of the data).
2. Write queries to retrieve and print all the data you entered. Try to print the data so that it is easy to understand (print appropriate headings first before printing data).
3. Write a query that will prepare a report for weekly Borrowing activity by Subject area, by Author, number of copies and number of days loaned out.
4. Write the following database update transactions using JAVA/JDBC or PHP or some other programming/scripting language.
   1. Add a new customer who is a borrower as well as a depositor.
   2. Are there new loan for the same customer with all loan details.
   3. Add a new employee who is a manager.
   4. Executor loan payment transaction.
   5. add a new savings account for a customer.
   6. Open a new branch for the bank.
5. Generate a report for a branch containing loans, the total outstanding balance against these loans, accounts, savings and checking and the balance they carry on each of these accounts, by customer.
6. Write following two triggers –
   1. Notify a customer that Hey recent transaction has exceeded his overdraft amount.
   2. Notify an employee that he has a 10 year work anniversary.

Due Dates:

1. Parts 1 and 2 Due Date: Sunday Apr 3. This should include for Part 1 the EER diagrams for your designs as well as documentation describing any assumptions you made, and the reasons for your design choices. Draw the EER diagrams using the notation in the textbook. You can also use UML class diagrams notation. You can use any drawing tool for drawing your diagrams. For part 2, what you turn in should include your relational schema diagram design and your CREATE table statements, and documentation describing your EER-to-relational mapping choices.
2. Part 3 Due Date: Sunday Apr 17. This will include an online demo demonstrating that your implementation works, as well as demonstrating your transactions to the grader. Source code of all your transactions should be submitted, as well as the data files. A demo schedule will be determined before the due date.

The files that you turn in should be submitted via Canvas to the GTA by midnight of each due date.

Files to be submitted –

1. Soft/scanned copies
   1. of your EER diagrams and documentation (Part 1)
   2. relational schema diagram with EER-to-relational mapping documentation (Part 2)
2. data files as loaded in the database as discussed in item 2 of Part 3 of the assignment.

Late penalty: -5% per day late.

Project teams: Each project can be done in a team of 2 persons, or individually.