Concordia University
Dept. of Computer Science & Software Engineering
COMP 353 – Databases
Fall 2018
Main Project

Project Description:

Title: A Simple database for a Financial Institution.
Due: November 26, 2018
Mark: 18%

Your team is required to design a relational database application system for a "realistic" situation, followed by an implementation of the system running on the DB server MySQL managed by AITS. The application is a two-tier system, a client side as a browser and http server with PHP parser at the server side. The system is expected to support all “representative” queries and transactions, and produce various desired reports.

The system should support the information required for the Customers of a Bank. As a database designer, you must ensure the integrity, security and reliability of the system by optimizing the design. Any user should not be able to access the application without successful authentication/login.

As we know, the Bank is locally operated (i.e. in Canada only) but has several branches located in different areas within a city as well as in different cities across Canada. It has a head office in downtown of Montreal. It provides banking, investment, and insurance services for different levels of needs, i.e. personal, business, corporate. All the banking services can be accessible through online and telephone as well. Anyone can be its potential client as soon as he/she opens any kind of account through any outlet. Each client may have different facilities based on different plans/options they choose and he/she will be charged accordingly. There is no accumulated interest rate for checking accounts, but it has only 2.0% interest rate for savings accounts.

Each client may have different credit cards, line of credit of different credit limits and rates based on certain rules and policies. For example, no one will get a credit card unless he/she maintains a satisfactorily level of banking transactions for at least six months, and no one will get a line of credit before one year. Mostly, the initial credit limit for a personal level would be CND 500, but it could vary from client to client. You are welcome to come up with other banking rules and policies.

Your database will be responsible for supporting

- The branch (branch_id, location, phone, fax, opening_date, manager_name, etc.)
- The employee (employee_id, title, name, address, start_date, salary, email, address, Phone Number, etc.)
- The client (client_id, name, date of birth, joining_date, address, email address, Phone Number, category, etc.)
- The account (account_number, type, option, balance, etc.)
- The services (banking, investment, insurance, etc.)
- The interest rate (kind of service, type of account, percentage, etc.)
- The charge plans (option, limit, charge, etc.)

Other tables might be needed to capture all the requirements.

Each branch must have one manager to manage the branch. The bank has one president and one general manager for each service provided by the bank. President of the bank and general managers are located in the head office of the bank. The client can view all of his/her account balances after a successful login and also carry out all his/her banking transactions via online.

The Project

Here is the possible scenario how your implemented Bank will serve

- Basic Insertion, deletion, and modification operations on any set of information; especially Clients, Employees, Accounts.
- Open new accounts for different level (personal, business, corporate) and as well as different types (checking, savings, foreign currency account, etc.) by its potential clients.
- Sign up by the Clients (using client card number and password) and carry on any banking transactions. After each successful transaction, the balance will be updated accordingly.

  In addition, the client should at least be able to:

  - ➢ Check his/her account balances (The client can view the account details and transaction history for the last ten years, also the client can have the choice of receiving alerts about his/her banking activities.);
  - ➢ Transfer money between accounts (The client can move money to the savings account, pay down the credit card, loan or credit line, or make a payment on the mortgage.);
  - ➢ Pay his/her bills (At any time, the client can pay one or multiple bills at once (it will only count as one transaction which will be only one charge), and set up future payments for regular bills.);
  - ➢ Send money fast (The client can send money to anyone with an email address or cell phone number using the Interac e-transfer service.).
  - ➢ Easily toggle between personal and business accounts whenever applicable.

- The monthly service charges will be automatically deducted from each client's balance at the end of each month. In addition, if a client exceeds the number of transactions allowed by the plan, he or she will be charged accordingly.
- Keep track of all of its employees' information including their schedule, payroll, holidays, sick days, etc. (Note: all employees are monthly paid).
- Calculation of the annual profits and losses of the Bank (including branch wise, city wise, and overall)

Note: If you need to change (add/remove) some attributes within tables/relationships from your Warm-up Project, you are allowed to do so, but needs to justify!

Guidelines

You will do this project as a team: the team must have a designated DBA or team manager. Each team member will be responsible for the entire project and at least a well-defined portion of the project, to be agreed upon by the team members. Divide up the work into components, with each component to be done by a subset of two members. However, the team is responsible for finishing the project in time and in a presentable form. The implementation of the RDBMS must be demonstrated by each team at the end of the semester. You will be provided a 30 minutes time slot to demonstrate that your system performs as expected. A schedule for the demos will be prepared and posted. The project report must be submitted at demo time. **All members must be present for the demo**.

The team is required to prepare a final report documenting their project and must include details on:

- The design of the DB using a E/R data model.
- Its conversion into a relational model satisfying at least 3NF.
- The user-interface for each supported application and reports. Use the MySQL DBMS to develop a miniature database application system for the bank. The system should provide its users with a good graphical user interface that is simple and dedicated for novice users.
- A sample session for each application (user guide).
- A list of members' contribution.
- An installation guide.
- All codes.

Note: It is desirable to proceed as follows:

Think about the application you are going to implement. Do a search on Web or visit your local Bank to gather information about its operations. Figure out the processing steps for the applications and the data required for such steps.

Design the E/R data model of your system and note the entities and relationships and their attributes. The design should be as compact as possible without sacrificing the required objectives. Make sure you state clearly any reasonable assumption made in your design, which is not specified in the requirements specified above.

Convert the E/R model into relational model. Make necessary refinements to the schema, if possible. Identify various integrity constraints such as primary keys, foreign keys, functional dependencies, and referential constraints. Make sure that your database schema is at least in 3NF. All the steps must be documented.

Provide implementation details of your database system in MySQL with a suitable user interface using HTML and PHP. Populate your tables with enough data to show various functionalities of your system (10 to 20 tuples per table on average).

A working version of the project should be presented before the lab instructors during the presentation. Every member of the group MUST be present during their demo.

At the demo, you also need to submit a hard copy of your project report documenting your project and must include details on:

- The design of the DB using an E/R data model.
- Its conversion into a relational model satisfying at least 3NF.
- The user-interface for each supported application and reports.
- A sample session for each application (user guide).
- All DDL codes.
- List of members' contributions as its last part. The title of this section should be "Contributions", indicating who did what in the project. It is wise to be realistic since the lab instructors will also evaluate each team member's contributions and ask relevant questions.

Note 1: The document report should be printed on a laser printer. The source of the code you demonstrate at demo time should be provided on a CD and submitted at demo time as part of your project report.

Note 2: Your project report must be properly bound in a folder (or binder) with official names of the team members, student ID's clearly appearing on the cover. And make sure your submission includes a signed originality form. Inappropriate submission will be penalized. If you find any resources that can further help enrich your project, it is fine to use it/them, however it is important that your report includes proper citation and acknowledgements.

Note 3: The source code of the system that you will present at the demo time should be submitted through Moodle as a SINGLE zip file by **NOVEMBER 26** at midnight. Your project report (of about 12 pages) is also due on **NOVEMBER 26** at midnight and must be submitted through Moodle as well.

The demonstration of the project will start in the last week of your class and finish before your Final Exam. Your report must be ready and hand-in during your demo.

Any ambiguities in this problem statement will have to be resolved. Some of these could be done via discussions which hopefully lead to design decisions. Your report should give rational explanations for all assumptions and decisions.

Good Luck!