DATA2201: Course Project

The aim of the project is to for you to use the knowledge you gained regarding designing relational databases and utilize it to design and implement a proof-of-concept database based on a case study. The course project contributes 30% of your final grade. This is an individual course project; we encourage collaboration to share knowledge but all students must contribute original work. Essentially identical submissions will be examined carefully and may result in technical challenges to confirm originality!

Case Study
You are a database developer at the **SKS National Bank**, a major bank with branches in all major Canadian cities.

The bank is thinking about replacing their aging account management system with a new one designed internally, and wants you to carry out some database development work as a pilot project. Your work is important, because it will be used as a proof-of-concept model to evaluate the feasibility and risks associated with a full-scale development (this is a common practice in industry -- build a proof-of-concept model to help understand the problem).

To get the pilot project started, you receive an email message from the project analyst at the SKS National Bank. Because she is extremely busy you know you will not be able to get much more information about this pilot project. You'll need to work with what you have. Here is the email message --

Hello again, and thanks for helping with this important pilot project. I've sketched out the basic information to help you build a data model.

SKS is organized into branches. Each branch is located in a commercial building in a particular city and is identified by a unique branch name. Each branch keeps a record of accounts, transactions, and loans that originate at that branch.

Bank customers have a name, a customer ID, and a home address. A customer may have an account (checking or savings) and may take out loans. Customers may have personal banking representatives or loan officers that they have formed a business relationship with; customers usually want to continue to be served by their representatives.

Bank employees (including bankers and loan officers) have unique employee IDs. Each employee has a manager, an employment start date, a name, a home postal address, and a set of locations where they work. A location may be a branch that serves customers or an administrative office.

Any number of customers can hold checking and/or savings accounts, and a customer can have more than one account. Each account has a current balance and a dated list of transactions, plus the monthly service fee, the balance below which the monthly service fee is charged, and an associated interest rate. Each account transaction is characterized by the transaction amount in CAD$, the type (deposit or withdrawal), the check number (for checking account withdrawals only).

A loan is associated with a particular SKS bank branch and is held by a customer who holds any account type (savings or checking). The bank tracks the loan amount and monthly payments. Each loan payment is associated with a monthly payment date and a loan payment number. A loan payment number does not uniquely identify a particular payment among all loans, but it does identify a particular payment for a specific loan.

(adapted from Database System Concepts, A. Silberschatz, H. Korth, and S. Sudarshan, 4th edition, McGraw-Hill, 2002, pp. 59-60.)

Phase I - Analysis & Design (20%)

By analyzing the information you received in the email, create a **set of requirements** that will be used to create the ER Diagram in the next step. Make sure your requirements are clear, consistent, and traceable. Your requirements should be in a well-organized document suitable for presentation to the project analyst and bank executives.

Create an **ER diagram** that represents the data model for this pilot project. Show all primary key attributes for all entities. You can use free online tools for ER drawing if you wish but this is not required. You can also use Visio, OmniGraffle, or any drawing tool that yields professional results. Please make sure that --

- your work can be opened without requiring any special tool or account (i.e. save to PDF)
- your work is presentation quality (you will be presenting it to your class)

Phase II - Implementation (30%)

All your implementation work will require query scripts (i.e. don't use the visual interface of SSMS to build tables and queries). Create your query scripts using any editor you wish. Make sure query script files end in a *.sql* extension, and remember to document your code! Make sure all query scripts are *deterministic* unless noted otherwise ... they can be run with the same outcome no matter what state your database is in. For example, if your script has a CREATE TABLE statement, you will want to ensure the table does not already exist.

Please name your query script files exactly as shown.

Based on your knowledge of the requirements and your ER diagram, create a query script "CREATE BANK TABLES" to build a **set of tables** in your SQL Server Sandbox. Name the tables appropriately and consistently. Make sure to specify primary keys and create all the necessary relationships between the tables. Add at least three constraints to your tables outside of primary key constraints.

Create a query script "POPULATE BANK TABLES" to **populate your tables** with initial data (no less than five and no more than twenty records per table). You can be creative here and make up data, but aim for realism. Remember your proof of concept model will be reviewed by your project analyst (and other stakeholders) so the more realistic and *representative* you can make your data, the better it will be (this is stressed in assessment). For the minimum balance to avoid service charges use $500. For the interest rates, use 4% for savings accounts and 1% for checking accounts.

Phase III - Queries (40%)

Create **query scripts** for your database to carry out the interactions below. For all query scripts, ensure result columns are named clearly and results are tabular unless specified otherwise.

1. "LIST BRANCH SUMMARY" Display a summary row for each bank branch showing the name, postal address (as one field), number of accounts, number of loans, total account balance, total loan balance, and average transaction amount against savings or checking accounts.

2. "ADD SERVICE CHARGE" Add a one-time service charge of $25 to all checking accounts (this does not need to be deterministic).

3. "LIST ACCOUNTS BELOW MINIMUM BALANCE" Display a summary row for each customer with a balance less than the minimum balance that triggers a service charge, for all checking and savings accounts.

4. "APPLY INTEREST CREDITS" Update all accounts to add a transaction that credits the interest owing for the current month, based on the current balance and applicable interest rate (this does not need to be

deterministic).

5. "APPLY SERVICE CHARGE" Update all accounts to add a transaction that applies the service charge, based on the current balance and minimum balance amount associated with each account (this does not need to be deterministic).

6. "DELETE CUSTOMER TEST" Attempt to delete a customer from your tables and demonstrate that your referential integrity rules prevent this from happening.

7. "WITHDRAW LOAN PAYMENT" Update all accounts to add a transaction that withdraws the monthly loan payment for all loans from a customer's checking account (if it exists) or else a customer's savings account, but only if the date matches the loan payment date (in other words, don't take out a payment if it is not the payment date for the loan). Make sure your sample data lets you test this scenario.

8. "LIST ACCOUNTS" Display a row for each account across all customers and branches showing the branch name, customer name, account type, current balance, and date of last transaction.

Phase IV - Presentation (10%)

Present your project to the class as if you are presenting to your project analyst and some senior members of the bank's executive suite. Try to summarize the main requirements, your approach, what constraints you identified to safeguard referential integrity, and any challenges you identified. Your presentation should take 10 minutes. Use any visual aids you wish; to use the classroom projector your work will need to be on a Google Docs or Drive somewhere. Check with your instructor if you have other presentation ideas. You will be assessed up to ten points on clarity (3 points), completeness (5 points), and style/professionalism (2 points).

How To Submit This Project

1. **ER Diagrams & Query Scripts** Submit all project files in a zip file to a D2L dropbox (instructor to set up) and please use this naming convention --

   DATA2201 Course Project By <your first and last name> .zip

2. **SQL Server Sandbox** Work directly in your sandbox. When you submit your project, the database tables and sample data in the sandbox will be inspected, and three of your query scripts will be executed (the same three for all students).

Resources Available To You

- your in-class notes and learning from participation
- your BVC SQL server sandbox (feel free to experiment)
- stackoverflow.com (lots of info but make sure it relates to TSQL)
- blog.sqlauthority.com (one of my favorites)
- microsoft.com/en-us/sql-server/sql-server-2016

If You Are Stuck

Your instructor team is here to help you. We are available throughout the week either by appointment or during regularly scheduled office hours. Please feel free to reach out if you find something difficult, or if you are not clear what is being requested. Kindly check D2L for the best method to reach your instructor (usually in the **Contact Your Instructor** section for your course). If you're not comfortable coming to an open classroom during office hours, please reach out to set up a private and confidential appointment for tutoring or guidance. We're here to help you succeed!