Relational Databases – DATA2201: Course Project

Rubric (Out of 100 total Marks (30%))

|  |  |
| --- | --- |
| Question number | Mark |
| Phase 1 | 50 |
| Phase 2 | 50 |

**Relational DB project case study**

You have received an email from your project liaison at SKS National Bank, and it contains the following details about the bank's operations:

**Branches at SKS:**

SKS National Bank is organized into branches, each situated in a specific city and distinguished by a unique name. Every branch maintains comprehensive records of both deposits and loan amounts. Additionally, branches are equipped with a team of dedicated staff, including personal bankers and loan officers, to assist customers with their financial needs.

**Customer Information:**

Bank customers are identified by their name, a customer ID, and a home address. In addition to their primary identification, customers may hold one or more accounts, which could be checking or savings. They can also apply for loans, and these loans may be held jointly by multiple customers. Some customers may have personal bankers or loan officers assigned to them for personalized financial advice and support.

**Employee Details:**

Bank employees, including bankers and loan officers, have unique employee IDs for identification purposes. Each employee is assigned a manager who oversees their work and progress. Employees have a start date, allowing the bank to calculate their length of employment, and they are associated with a name and home address. Employees can work at various locations, including branches and non-branch offices, depending on their roles and responsibilities within the bank.

**Account Information:**

Checking and savings accounts can be jointly held by multiple customers, promoting shared financial management. A customer may have multiple accounts, each with its own unique balance. The bank diligently tracks the most recent date each account was accessed by the customer to monitor account activity. Savings accounts are linked to an interest rate, which varies based on the account type and prevailing market conditions. Checking accounts are also designed to keep track of overdrafts, recording key information such as dates, amounts, and check numbers to ensure the smooth management of customer finances.

**Loan Management:**

Loans originate from specific branches and can be held by one or more customers, facilitating collaborative borrowing. The bank pays meticulous attention to loan management, including tracking loan amounts and payments. While a loan payment number may not uniquely identify a payment among all loans, it serves as a distinctive identifier for a specific payment within a particular loan account. Detailed information, such as the payment date and amount, is meticulously recorded for each payment, ensuring accurate and transparent financial records.

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

**Phase I – Conceptual Design and Logical Design (50 marks)**

**Step 1.** Create a set of requirements for the attached case study that will be used to create the ER Diagram in Step 2. (2 marks)

**Step 2.** Create an ER diagram or Relational schema that contains at least five entities and at least four binary relationships (1 to 1, or 1 to n). At least one of the relationships should be many to many. Show all attributes for all entities. Include the relations (with their attributes) and the referential integrity constraint lines. (10 marks)

**Step 3.** Resolve the identified many to many relation in Step 2 and show how your ER diagram pass 1NF, 2NF and 3NF. (3 marks)

**Step 4.** Based on that schema, create a set of tables. Name the database and its tables appropriately. Make sure to specify primary keys and create all the necessary relationships between the tables. (10 marks)

**Step 5**. Populate the tables with initial data (no less than five and no more than twenty records per table). (10 marks)

**Step 6.** Create at least 10 queries for this database in a stored procedure or function format. (Before you start writing a query you should think of a user story by studying the case study. (10 marks)

**Deliverables:** Database requirements, ER diagram passing till 2NF, creation of database and tables, and establishing relationship between them. Identification of 10 queries (including stored procedure and functions relevant to the case study. Final deliverable as .sql file or .bak file.

**Phase II – Implementation (using an RDBMS) (50 marks)**

**Step 1: Create Different User Levels and Assign Privileges (15 points)**

* Create user accounts: Create two user accounts, customer\_yourID with the password customer and accountant\_yourID with the password accountant.
* Define privileges for customer\_yourID: Ensure that when logged in with the customer\_yourID account, the user can only read selected tables related to customers, such as customers, accounts, and loans tables.
* Testing Query Script: Provide a testing query script to verify that the customer\_yourID account can only perform read operations on the specified tables.
* Define privileges for accountant\_yourID: Ensure that when logged in with the accountant\_yourID account, the user can read all tables but cannot update accounts, and loans tables.
* Testing Query Script: Provide a testing query script to confirm that the accountant\_yourID account has the correct privileges.

**Step 2: Create Different Triggers to Monitor Database Activities (15 points)**

* Develop a trigger that generates a message when a new customer registers or a new account is created.
* Develop a trigger that generates a message confirming that a loan payment has been made and updates accounts table).
* Implement a trigger that reports data updates during transaction performance on savings or checking accounts.
* Develop a trigger to report data reads that occur from a table of your choice (example for "Branches" Table). Hint: you can create a log table to log data read events.

**Step 3: Create Indexes Based on Frequently Used Attributes (15 points)**

* Replace the default cluster index with a non-key attribute for one table (e.g., customers table). Please specify the table and provide details if you used a GUI tool for this.
* Establish a composite clustered index for one table (e.g., Accounts table) by removing the default clustered index. Please specify the table and provide details if you used a GUI tool for this.
* Generate a non-clustered composite index for one of the tables (e.g., loans table). Please specify the table and provide details if you used a GUI tool for this.

**Step 4: Alter the Database to Include a JSON Storage Table (5 points)**

* Alter an existing table by adding a new column with a JSON data type to store JSON records.
* Populate the newly created JSON column with sample JSON data for testing and demonstration purposes.

***Deliverables:*** Database scripts/Stored procedures/function/trigger, .bak file of the DB, If you created the users and assigned the privilege from GUI. Step by step compiled screenshot of your database if you work any feature with GUI.