DATA2201 – Relational Databases
Group Project: SKS National Bank

Case Study

You are an employee of BVC Technology Group and have been hired by SKS National Bank to build a new database system for their entire banking operations. Here are the requirements given to you from the project liaison at SKS National Bank:

SKS National Bank is organized into branches. Each branch is in a particular city and is identified by a unique name. Each branch keeps a total of all the deposits and loan amounts.

Bank customers have a name, a customer ID, and a home address. A customer may have a chequing account, a savings account, and may take out loans. Customers may have personal bankers or loan officers that they always work with.

Bank employees (including personal bankers and loan officers) have unique employee IDs. Each employee has a manager, a start date, a name, a home address, and a set of locations where they work. A location may be a bank branch or an office that is not in a branch.

Chequing and savings accounts can be held by more than one customer (for example, spouses can have joint accounts) and a customer can have more than one account. Each account has a balance and the most recent date that the account was accessed by the customer. Savings accounts have an associated interest rate. Chequing accounts keep track of dates, amounts, and check numbers for overdrafts.

A loan originates at a particular branch and can be held by one or more customers. The bank tracks the loan amount and payments. 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. The date and amount are recorded for each payment.

Project Overview

Your task in this assessment is to work as a group to create a database with advanced features for SKS National Bank. This project has two phases:

1. Database Design and Implementation (50%)
2. Advanced Database Operations (50%)

The requirements and rubric for each phase are detailed below.

Phase 1: Database Design and Implementation

Phase 1 is about designing your database and implementing it with initial data and basic features.

Entity Relationship Diagram

Create a well-designed and detailed ERD or relational schema that meets all the requirements of the case study above. The ERD or relational schema should include the following information:

- Tables with names and attributes.
- Indications of which columns are primary or foreign keys.
- Relationships between tables, including ordinality.
- Any special constraints that should be noted.
- Check each of the entities for normalization to avoid and eliminate anomalies.

Database Creation

Using the ERD or relational schema, create a database and set of tables. Create an SQL file named *"create_database.sql"*. This file should perform the following actions:

- Delete the database if it already exists.
- Create the database.
- Create all tables based on your design.
- Create the appropriate primary and foreign keys.
- Create the appropriate constraints and relationships.

Database Population

With your newly created and empty database, populate it with sample data. Create an SQL file named *"populate_database.sql".* This file should perform the following actions:

- Populate each table in the database with sample data.
- Include at least 5 rows of data per table (if possible).
- Use primary keys and foreign keys appropriately.
- The sample data should act as test cases for your prepared queries.

Prepared Queries

Prepare relevant queries for your populated database. Create an SQL file named *"prepared_queries.sql"*. This file should contain 10 queries that meet the following requirements:

- Each query is in a stored procedure or user-defined function format.
- Each query performs a meaningful action based on the case study.
- Each query includes a comment that describes the purpose of the query.
- Each query has a separate SQL statement that tests the query.

You may assume that the database and tables exist, but you should connect to your database with the USE command at the start of the file.

Submission Instructions

- Submit 4 files:
  - ERD as an image file, PDF, or Word document.
  - *create_database.sql*
  - *populate_database.sql*
  - *prepared_queries.sql*
- Include each group member's name in each file.
- Properly reference any outside resources that you use.
- Place all source files into a zipped (compressed) folder, then upload it to D2L.

Rubric

- 50 points available.
- Partial credit will be given.
- Worth 15% of final grade.

Grading Items	**Points**
Entity Relationship Diagram: • Tables are identified. • Attributes are identified. • Relationships are identified. • Primary keys and foreign keys are identified. • Data is normalized to 3NF.	10
Database creation: • Script runs without errors. • Database structure matches the ERD. • Data types are reasonable.	10

• Primary keys, foreign keys, and other constraints are reasonable.	
Database population: • Script runs without errors. • At least 5 rows of data per table. • Data represents reasonable test values for queries.	10
Prepared queries: • 10 queries (2 points each). • Queries are in a stored procedure or user-defined function format. • Each query includes a comment and a separate SQL statement to test it. • Script runs without errors. • Queries return correct results. • Queries are formatted according to best practices.	20
TOTAL	**50**

Phase 2: Advanced Database Operations

Phase 2 is about implementing advanced database operations onto the database you created in Phase 1.

Users and Privileges

Create different levels of users and assign appropriate privileges. Create an SQL file named *"create_users.sql".* This file should perform the following actions:

- Create a login and user named "customer_group_[?]" where [?] is your group letter. (For example, "customer_group_A".)
    - Their password should be "customer".
    - Their user account should only be able to read tables that are related to customers, based on your ERD. (For example, tables related to customer information, accounts, loans, and payments).
- Create a login and user named "accountant_group_[?]" where [?] is your group letter. (For example, "accountant_group_B".)
    - Their password should be "accountant".
    - Their user account should be able to read all tables.

    - Their user account should not be able to update, insert or delete in tables that are related to accounts, payments and loans, based on your ERD. Those permissions should be revoked.
- Provide SQL statements that test the enforcement of the privileges on the two users created above.

Triggers

Create triggers to monitor the different activities on your database. Create an SQL file named *"create_triggers.sql".* This file should perform the following actions:

- Create a new table called "Audit" to track changes made in the database. At minimum, this table should contain the following 3 pieces of information:
    - Primary key.
    - An explanation of exactly what happened in the database.
    - Timestamp.
- Create 3 triggers of your choice that meet the following minimum requirements:
    - Each trigger should log an entry into the Audit table.
    - Each trigger should provide meaningful information to the database administrator at SKS National Bank.
- Provide SQL statements that test each of the 3 triggers.

JSON and Spatial Data

Support JSON and spatial data in your database. Create an SQL file named *"create_json_spatial.sql".* This file should perform the following actions:

- Add one column with the JSON data type to any table of your choice. Populate it with sample data.
- Add one column to a table to store the spatial information of the bank's branches. Populate it with sample data.

Backup

Create a backup (.bak file) of your final database after all the above features have been implemented. Include this backup file as part of your submission.

Final ERD

Update your ERD to create a final version that includes all new tables and columns from Phase 2 and any feedback you received from the instructor from Phase 1.

Presentation

At a date and time specified at the end of the course each group will present their project to the rest of the class. In approximately 5 minutes: explain your final ERD, present and explain the Audit table and three triggers working correctly, and discuss the most important lesson that was learned in this project (for example, if your group had to start the project over, what would they do differently or the same?). All group members must participate.

You do not need to prepare any form of slideshow. If there is a reason that a member or the entire group cannot make the presentation, please speak with the instructor as soon as possible to discuss alternative arrangements.

Submission Instructions

- Submit 5 files:
  - *create_users.sql*
  - *create_triggers.sql*
  - *create_json_spatial.sql*
  - A backup (.bak) of your final database.
  - Final ERD as an image file, PDF, or Word document.
- Include each group member's name in each file.
- Properly reference any outside resources that you use.
- Place all source files into a zipped (compressed) folder, then upload it to D2L.

Rubric

- 50 points available.
- Partial credit will be given.
- Worth 15% of final grade.

Grading Items	**Points**
Users and privileges: • Script runs without errors. • Both users are created. • Privileges are enforced correctly.	10
Triggers: • Script runs without errors. • Audit table is created. • Triggers are created and produce expected results.	15
JSON and spatial data: • Script runs without errors.	5

• JSON column created and populated. • Spatial data column created and populated.	
Backup: • The backup file (.bak) restores without errors.	5
Final ERD: • Updated ERD that includes all new tables and columns from Phase 2 and any feedback from Phase 1.	5
Presentation (approximately 5 minutes): • All group members must participate. • Present and explain your final ERD. • Present the Audit table and three triggers working correctly. Explain your design decisions. • Discuss the most important lesson that was learned from this project. If you had to start the project over, what would you make sure to do differently or the same?	10
TOTAL	**50**