DATA2201 – Relational Databases: Phase 1 Final Report – SKS National Bank

Submitted by:

Dylan Retana (ID: 467710)

Freddy Munini (ID: 473383)

Ime Iquoho (ID: 460765)

Bow Valley College

---

1. Introduction

The SKS National Bank database system was designed to manage and organize essential banking operations, including customer accounts, loans, branches, employees, and transactions. The primary objective of this project was to create a normalized, relational database structure that ensures data integrity, minimizes redundancy, and supports reliable analytical queries for reporting and management purposes.

2. Entity-Relationship Diagram (ERD)

The ERD models the relationships between customers, accounts, loans, employees, branches, and other related entities. It enforces normalization up to the Third Normal Form (3NF), ensuring that all functional dependencies are properly managed. The ERD includes the following key components:

The Entity-Relationship Diagram (ERD) for the SKS National Bank represents how all key entities within the database interact to model real banking operations. The system consists of twelve main entities: Branch, Location, Employee, Customer, Account, Loan, Overdraft, LoanPayment, and four junction tables (CustomerAccount, CustomerLoan, EmployeeLocation, and CustomerAssignedStaff). The Branch entity maintains details about each banking location and has a one-to-many relationship with Account, Loan, and Employee, meaning each branch can manage multiple accounts, loans, and employees. The Location entity records both branch and office addresses and connects to EmployeeLocation through a many-to-many relationship, allowing employees to work at several sites. The Employee entity stores staff information and includes a self-referencing relationship through the ManagerID field, showing that an employee may report to another employee. The Customer entity represents clients and has many-to-many relationships with both Account and Loan through CustomerAccount and CustomerLoan, respectively, which supports joint accounts and shared loans. The CustomerAssignedStaff table

establishes another many-to-many relationship linking Customer and Employee, used to identify which personal banker or loan officer serves each client. The Account table holds details for chequing and savings accounts and has a one-to-many relationship with Overdraft, since one account can have multiple overdraft events, but each overdraft belongs to one account. The Loan entity tracks borrowing information and is linked to LoanPayment in a one-to-many relationship, as each loan can have many payments, each identified by a composite primary key (LoanID, PaymentNo). Together, these relationships enforce referential integrity, ensure accurate representation of banking rules, and maintain normalization to Third Normal Form (3 NF). The ERD demonstrates a robust, scalable design that accurately supports the operational and analytical needs of the SKS National Bank.

3. SQL Scripts Overview

Three SQL scripts were developed and tested to meet the project requirements:

a) create_database.sql – Defines the database schema.

This script creates all necessary tables, relationships, constraints, and indexes. It enforces referential integrity using primary and foreign keys, includes CHECK and UNIQUE constraints, and establishes normalization rules. The database includes validation queries to confirm the schema structure.

b) populate_database.sql – Inserts sample data.

This script populates the database with at least five rows per table. It includes realistic data for branches, employees, customers, accounts, loans, and overdrafts. Foreign keys are populated dynamically to maintain integrity, and validation queries confirm row counts.

c) prepared_queries.sql – Defines and tests stored procedures.

Ten stored procedures were created to provide meaningful insights based on the SKS National Bank case study. Each procedure includes a comment describing its purpose and a test EXEC statement. The procedures handle calculations such as total balances per customer, branch performance, overdraft tracking, and more.

4. Testing and Validation

All scripts were executed successfully in Microsoft SQL Server Management Studio (SSMS). Testing confirmed that every script runs without errors, that each table contains valid data, and that stored procedures return accurate and expected results. The ERD_Diagram_Testing_Report.pdf provides screenshots and query outputs demonstrating functional validation.

5. References

The following resources were consulted to support the SQL syntax, concepts, and structure used in this project:

• W3Schools – SQL CREATE TABLE Statement: https://www.w3schools.com/sql/sql_create_table.asp
• W3Schools – SQL Constraints (PRIMARY KEY, FOREIGN KEY, CHECK): https://www.w3schools.com/sql/sql_constraints.asp
• W3Schools – SQL INSERT INTO Statement: https://www.w3schools.com/sql/sql_insert.asp
• W3Schools – SQL JOINs Overview: https://www.w3schools.com/sql/sql_join.asp
• Microsoft Learn – CREATE PROCEDURE (Transact-SQL): https://learn.microsoft.com/en-us/sql/t-sql/statements/create-procedure-transact-sql
• Microsoft Learn – Aggregate Functions (SUM, COUNT): https://learn.microsoft.com/en-us/sql/t-sql/functions/aggregate-functions-transact-sql
• Microsoft Learn – STRING_AGG Function: https://learn.microsoft.com/en-us/sql/t-sql/functions/string-agg-transact-sql
• Microsoft Learn – OVER Clause for Window Functions: https://learn.microsoft.com/en-us/sql/t-sql/queries/select-over-clause-transact-sql
• Bow Valley College – DATA2201 Course Notes and Labs (Instructor: Prof. Dorsey)