**SQL Server Assignment: Bank Management System**

**Mark: 20**

**Submission date: 30-October-2025**

**📘 Scenario:**

You’ve been hired as a database developer by a local bank. Your task is to design and implement a **Bank Management System** to manage **customers**, **accounts**, **transactions**, and **staff**. The database should ensure data integrity, prevent duplicate entries, and support key banking operations.

**🔹 Assignment Tasks**

**🔸 1. Database Design & Creation**

Create a database named BankDB. Then create the following tables:

**Customers**

* CustomerID (INT, Primary Key)
* FirstName (VARCHAR)
* LastName (VARCHAR)
* Email (VARCHAR, Unique)
* Phone (VARCHAR)
* Address (VARCHAR)
* DateJoined (DATE)

**Accounts**

* AccountID (INT, Primary Key)
* CustomerID (Foreign Key)
* AccountType (VARCHAR) — e.g., Savings, Current
* Balance (DECIMAL(12,2))
* DateOpened (DATE)
* Status (VARCHAR) — e.g., Active, Closed

**Transactions**

* TransactionID (INT, Primary Key)
* AccountID (Foreign Key)
* TransactionType (VARCHAR) — e.g., Deposit, Withdrawal, Transfer
* Amount (DECIMAL(12,2))
* TransactionDate (DATETIME)
* Remarks (VARCHAR)

**Staff**

* StaffID (INT, Primary Key)
* FullName (VARCHAR)
* Email (VARCHAR)
* Role (VARCHAR) — e.g., Manager, Teller
* Branch (VARCHAR)

**A. SELECT Queries (Data Retrieval)**

1. **List all customer details** who joined in the last 6 months.
2. **Display all accounts** along with the full name of the customer who owns each account.
3. Show all **transactions made on a specific date** (e.g., '2025-10-01').
4. Find the **total balance of all active accounts**.
5. Display the **top 3 customers** who have the highest account balances.
6. Show all **transactions of type 'Withdrawal'** where the amount is greater than 10,000.
7. List **all staff members working in a specific branch** (e.g., 'Main Branch').
8. Find the **number of accounts per account type** (group by AccountType).
9. Show **all customers who don’t have any accounts.**
10. List **all accounts that were opened before 2022** and are still marked as "Active".

**B. UPDATE Queries**

1. Update the **phone number** of a customer with CustomerID = 101.
2. Increase the **balance** of a specific account (AccountID = 1005) by 5000 due to interest credit.
3. Change the **status** of all accounts with a balance of 0 to 'Closed'.
4. Assign a **new role** 'Senior Teller' to the staff member with StaffID = 2002.
5. Update all transactions of type 'Transfer' to include the remark 'Processed via internal system' where remarks are NULL.

**C. DELETE Queries**

1. Delete a **transaction record** with TransactionID = 3005.
2. Remove **all accounts** that are marked 'Closed' and have Balance = 0.
3. Delete **all customers** who haven’t opened any account.
4. Remove **staff records** of those who are assigned to a non-existent branch (you can assume a fake branch name for this).
5. Delete all **transactions older than 5 years** from the current date.

**D. Bonus SELECTs with JOINs or Subqueries**

1. List the **customer full names** along with their **total number of transactions**.
2. Show each staff member and the **number of customers served** in their branch (you can assume branch-wise assignment).
3. List **all accounts along with their most recent transaction date and amount**.

**E. Add Er diagram**

**Add trigger in transaction table after update insert or delete**