**Final Submission Practical 2**

### **Banking Database**

A banking institution needs a relational database to manage information about its customers, accounts, transactions, employees, and branches. The database must support creating, updating, deleting, and retrieving data while ensuring data integrity through constraints. The schema design should incorporate various keys like primary key, foreign key, unique key, composite key, candidate key, alternate key, and super key. Below is the scenario.

### **Scenario**

* Customers: Each customer has a unique ID, name, contact details, and email address. Customers can hold multiple accounts.
* Accounts: Each account has a unique account number, account type (e.g., savings, checking), balance, and is associated with one customer.
* Transactions: Transactions record deposits, withdrawals, or transfers made by customers. Each transaction should include the account number, amount, date, and type of transaction.
* Employees: Bank employees manage customer accounts. Each employee has a unique ID, name, position, and email address.
* Branches: A branch has a unique ID, name, and address. Each branch has multiple employees and customers.

### **Tasks:**

### **1. Create the Tables**

* Create tables for Customers, Accounts, Transactions, Employees, and Branches.
* Use appropriate constraints:
  + Primary keys for unique identification.
  + Foreign keys for relationships between tables.
  + Unique keys for attributes like email.
  + Composite keys where multiple columns uniquely identify a record.

#### **2. Insert Data**

* Insert sample data into each table (e.g., 5-10 rows per table).
* Ensure data integrity (e.g., no duplicate IDs or invalid foreign key references).

#### 

#### **3. Update Data**

* Update account balances after a transaction.
* Update customer contact details or employee position.

#### **4. Delete Data**

* Remove a customer and all their associated accounts.
* Delete transactions older than a specific date.

#### **5. Alter Table**

* Add a column to the Customers table for the customer’s date of birth.
* Modify the Accounts table to add an interest rate column.

#### **6. Constraints and Keys**

* Implement the following:
  + Primary Key: Unique identification for all entities (e.g., CustomerID, AccountNumber).
  + Foreign Key: Relationships (e.g., Accounts to Customers, Employees to Branches).
  + Unique Key: Email for Customers and Employees.
  + Composite Key: AccountNumber and TransactionID in Transactions.
  + Candidate Key: Multiple unique attributes (e.g., Email and Phone for Customers).
  + Alternate Key: Use Phone as a key if Email is not the primary identifier.
  + Super Key: Any superset of primary key attributes.

#### **7. Queries**

* Write queries to perform the following:
  + Retrieve all transactions for a specific account.
  + List all customers for a specific branch.
  + Find employees managing the highest number of accounts.
  + Calculate the total balance across all accounts for a specific customer.

#### **Submission requirements**

* Scripts for creating tables with constraints.
* Insert, update, delete, and alter table queries.
* Query examples demonstrating the use of each key type and relationships.
* Test cases to verify integrity constraints (e.g., trying to insert a duplicate key).