Problem Statement 11 (DDL USING MYSQL)
Create following tables using a given schema and insert appropriate data into the
same:
Customer (CustID, Name, Cust_Address, Phone_no, Email_ID, Age)
Branch (Branch ID, Branch_Name, Address)
Account (Account_no, Branch ID, CustID, date_open, Account_type, Balance)
1. Create the tables with referential integrity.
2. Draw the ER diagram for the same.
3. Create an Index on primary key column of table Account
4. Create the view as Customer_Info displaying the customer details for age less
than 45.
5. Update the View with open date as 16/4/2017
6. Create a sequence on Branch able.
7. Create synonym 'Branch_info' for branch table.

```sql
-- Step A: Create the database
CREATE DATABASE IF NOT EXISTS BankDB;
USE BankDB;

-- Step B: Create the Customer table
CREATE TABLE Customer (
    CustID INT PRIMARY KEY,
    Name VARCHAR(50),
    Cust_Address VARCHAR(100),
    Phone_no VARCHAR(15),
    Email_ID VARCHAR(50),
    Age INT
);

-- Step C: Create the Branch table
CREATE TABLE Branch (
    BranchID INT PRIMARY KEY,
    Branch_Name VARCHAR(50),
    Address VARCHAR(100)
);

-- Step D: Create the Account table with foreign keys to Customer and Branch
CREATE TABLE Account (
    Account_no INT PRIMARY KEY,
    BranchID INT,
    CustID INT,
    date_open DATE,
    Account_type VARCHAR(20),
    Balance DECIMAL(15, 2),
    FOREIGN KEY (BranchID) REFERENCES Branch(BranchID),
    FOREIGN KEY (CustID) REFERENCES Customer(CustID)
);

-- Step E: Create an index on the primary key column of the Account table
CREATE INDEX idx_account_pk ON Account(Account_no);

-- Step F: Create a view as Customer_Info displaying the customer details for age
less than 45
CREATE VIEW Customer_Info AS
SELECT CustID, Name, Cust_Address, Phone_no, Email_ID, Age
FROM Customer
```

```sql
WHERE Age < 45;

-- Step G: Update date_open in Account for records with NULL values to '2017-04-16'
UPDATE Account
SET date_open = '2017-04-16'
WHERE date_open IS NULL;

-- Step H1: Drop the foreign key constraint on BranchID in Account
ALTER TABLE Account DROP FOREIGN KEY account_ibfk_1;

-- Step H2: Modify BranchID in the Branch table to be AUTO_INCREMENT
ALTER TABLE Branch MODIFY BranchID INT PRIMARY KEY AUTO_INCREMENT;

-- Step H3: Reinstate the foreign key constraint on BranchID in Account,
referencing Branch
ALTER TABLE Account
ADD CONSTRAINT account_ibfk_1 FOREIGN KEY (BranchID) REFERENCES Branch(BranchID);

-- Step I: Create a view 'Branch_info' as a synonym for the Branch table
CREATE OR REPLACE VIEW Branch_info AS
SELECT * FROM Branch;
```