Introduction to DBMS

ASSESSMENT

Q1. Create a Bank table, Account Holder table, and Loan table.

=>

```
-- Bank Table
CREATE TABLE Bank (
   branch_id INT PRIMARY KEY,
   branch_name VARCHAR(50),
   branch_city VARCHAR(50)
);

-- Account Holder Table
CREATE TABLE Account_Holder (
   account_holder_id INT PRIMARY KEY,
   account_no VARCHAR(20) UNIQUE,
   account_holder_name VARCHAR(100),
   city VARCHAR(50),
   contact VARCHAR(15),
   date_of_account_created DATE,
   account_status VARCHAR(20) CHECK (account_status IN ('Active', 'Terminated')),
   account_type VARCHAR(30),
   balance DECIMAL(12,2)
```

```sql
);

-- Loan Table
CREATE TABLE Loan (
   loan_no INT PRIMARY KEY,
   branch_id INT,
   account_holder_id INT,
   loan_amount DECIMAL(12,2),
   loan_type VARCHAR(50),
   FOREIGN KEY (branch_id) REFERENCES Bank(branch_id),
   FOREIGN KEY (account_holder_id) REFERENCES Account_Holder(account_holder_id)
);
```

```sql
-- Insert into Bank
INSERT INTO Bank (branch_id, branch_name, branch_city) VALUES
(101, 'Central Bank', 'Mumbai'),
(102, 'West End Bank', 'Delhi'),
(103, 'City Bank', 'Mumbai');

-- Insert into Account Holder
INSERT INTO Account_Holder
(account_holder_id, account_no, account_holder_name, city, contact, date_of_account_created, account_status, account_type, balance)
VALUES
```

```
(1, 'A123', 'Ravi Sharma', 'Mumbai', '9876543210', '2025-03-12', 'Active',
'Savings', 5000.00),

(2, 'B456', 'Neha Verma', 'Mumbai', '9876500011', '2025-03-20', 'Active',
'Current', 3000.00),

(3, 'C789', 'Amit Singh', 'Delhi', '9123456789', '2025-04-18', 'Active', 'Savings',
7000.00),

(4, 'D234', 'Priya Gupta', 'Pune', '9988776655', '2025-01-10', 'Terminated',
'Current', 2000.00);


-- Insert into Loan

INSERT INTO Loan (loan_no, branch_id, account_holder_id, loan_amount,
loan_type) VALUES

(201, 101, 1, 200000.00, 'Home Loan'),

(202, 103, 2, 50000.00, 'Car Loan'),

(203, 102, 3, 120000.00, 'Education Loan');
```

Q2. Perform an intra-bank transfer (Account A → B, $100). Ensure both accounts are updated.

=>

```
START TRANSACTION;


-- Debit from Account A

UPDATE Account_Holder

SET balance = balance - 100

WHERE account_no = 'A123';


-- Credit to Account B
```

```
UPDATE Account_Holder

SET balance = balance + 100

WHERE account_no = 'B456';


COMMIT;
```

Q3. Also fetch the details of the account holder who are related from the same city.

=>

```
SELECT account_holder_name, city

FROM Account_Holder a1

WHERE EXISTS (

   SELECT 1

   FROM Account_Holder a2

   WHERE a1.city = a2.city AND a1.account_holder_id <> a2.account_holder_id

);
```

Q4. Write a query to fetch account number and account holder name, whose accounts were created after 15th of any month.

=>

```
SELECT account_no, account_holder_name

FROM Account_Holder

WHERE DAY(date_of_account_created) > 15;
```

Q5. Write a query to display the city name and count the branches in that city. Give the count of branches an alias name of Count_Branch.

=>

```
SELECT branch_city, COUNT(branch_id) AS Count_Branch

FROM Bank

GROUP BY branch_city;
```

Q6. Write a query to display the account holder’s id, account holder’s name, branch id, and loan amount for people who have taken loans. (NOTE : use sql join concept to solve the query)

=>

```
SELECT ah.account_holder_id, ah.account_holder_name, l.branch_id, l.loan_amount

FROM Account_Holder ah

INNER JOIN Loan l ON ah.account_holder_id = l.account_holder_id;
```
