```
-- Create Bank table
CREATE TABLE Bank (
    branch_id INT PRIMARY KEY,
    branch_name VARCHAR(100),
    branch_city VARCHAR(100)
);

-- Create Account Holder table
CREATE TABLE Account_Holder (
    acc_holder_id INT PRIMARY KEY,
    acc_no VARCHAR(20) UNIQUE,
    acc_holder_name VARCHAR(100),
    city VARCHAR(100),
    contact VARCHAR(15),
    acc_created_date DATE,
    acc_status VARCHAR(20),
    acc_type VARCHAR(50),
    balance DECIMAL(10, 2)
);

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

);

2. Fund Transfer Transaction (Intra Bank Transfer)

```
START TRANSACTION;

-- Deduct from Account A
UPDATE Account_Holder
SET balance = balance - 100
WHERE acc_no = 'A001';

-- Add to Account B
UPDATE Account_Holder
SET balance = balance + 100
WHERE acc_no = 'B001';

-- Check if both operations were successful before committing
-- You can simulate conditions in procedures or apps, but here's the basic transaction:
COMMIT;
```

3. Fetch Account Holders from the Same City

```
SELECT *
FROM Account_Holder AH1
WHERE EXISTS (
   SELECT 1
   FROM Account_Holder AH2
   WHERE AH1.city = AH2.city
     AND AH1.acc_holder_id <> AH2.acc_holder_id
);
```

4. Accounts Created After 15th of Any Month

```
SELECT acc_no, acc_holder_name
FROM Account_Holder
```

```
WHERE DAY(acc_created_date) > 15;
```

5. Count Branches in Each City

```
SELECT branch_city, COUNT(*) AS Count_Branch
FROM Bank
GROUP BY branch_city;
```

6. Account Holders with Loans (Using JOIN)

```
SELECT
    AH.acc_holder_id,
    AH.acc_holder_name,
    L.branch_id,
    L.loan_amount
FROM
    Account_Holder AH
JOIN
    Loan L ON AH.acc_holder_id = L.acc_holder_id;
```