Mysql-3Consider following Relation Account (Acc_no, branch_name,balance) Branch(branch_name,branch_city,assets) Customer(cust_name,cust_street,cust_city) Depositor(cust_name,acc_no) Loan(loan_no,branch_name,amount) Borrower(cust_name,loan_no) Create above tables with appropriate constraints like primary key, foreign key, not null etc.

1. Find the branches where average account balance > 15000.
2. Find number of tuples in customer relation.
3. Calculate total loan amount given by bank.
4. Delete all loans with loan amount between 1300 and 1500.
5. Find the average account balance at each branch
6. Find name of Customer and city where customer name starts with Letter P.

```
-- Branch Table
CREATE TABLE Branch (
    branch_name VARCHAR(50) PRIMARY KEY,
    branch_city VARCHAR(50) NOT NULL,
    assets DECIMAL(15,2) CHECK (assets >= 0)
);

-- Account Table
CREATE TABLE Account (
    acc_no INT PRIMARY KEY,
    branch_name VARCHAR(50) NOT NULL,
    balance DECIMAL(15,2) CHECK (balance >= 0),
    FOREIGN KEY (branch_name) REFERENCES Branch(branch_name)
        ON DELETE CASCADE ON UPDATE CASCADE
);
```

-- Customer Table

```
CREATE TABLE Customer (
    cust_name VARCHAR(50) PRIMARY KEY,
    cust_street VARCHAR(100),
    cust_city VARCHAR(50)
);

-- Depositor Table (Many-to-Many between Customer and Account)
CREATE TABLE Depositor (
    cust_name VARCHAR(50),
    acc_no INT,
    PRIMARY KEY (cust_name, acc_no),
    FOREIGN KEY (cust_name) REFERENCES Customer(cust_name)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (acc_no) REFERENCES Account(acc_no)
        ON DELETE CASCADE ON UPDATE CASCADE
);

-- Loan Table
CREATE TABLE Loan (
    loan_no INT PRIMARY KEY,
    branch_name VARCHAR(50) NOT NULL,
    amount DECIMAL(15,2) CHECK (amount >= 0),
    FOREIGN KEY (branch_name) REFERENCES Branch(branch_name)
        ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
-- Borrower Table (Many-to-Many between Customer and Loan)
CREATE TABLE Borrower (
    cust_name VARCHAR(50),
    loan_no INT,
    PRIMARY KEY (cust_name, loan_no),
    FOREIGN KEY (cust_name) REFERENCES Customer(cust_name)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (loan_no) REFERENCES Loan(loan_no)
        ON DELETE CASCADE ON UPDATE CASCADE
);
```

Step 1: Insert Data into Branch Table

```
INSERT INTO Branch (branch_name, branch_city, assets) VALUES
('Wadia College', 'Pune', 1500000),
('Camp Branch', 'Pune', 1000000),
('FC Road', 'Pune', 1200000),
('Deccan', 'Mumbai', 2000000),
('MG Road', 'Nashik', 1800000);
```

step 2: Insert Data into Account Table

```
INSERT INTO Account (acc_no, branch_name, balance) VALUES
(101, 'Wadia College', 50000),
(102, 'Camp Branch', 35000),
(103, 'FC Road', 25000),
(104, 'Deccan', 75000),
(105, 'Wadia College', 15000);
```

Step 3: Insert Data into Customer Table

```
INSERT INTO Customer (cust_name, cust_street, cust_city) VALUES
('Amit', 'Laxmi Road', 'Pune'),
('Sneha', 'MG Road', 'Nashik'),
('Rahul', 'JM Road', 'Pune'),
('Priya', 'Deccan Gym', 'Mumbai'),
('Kiran', 'FC Road', 'Pune');
```

Step 4: Insert Data into Depositor Table

```
INSERT INTO Depositor (cust_name, acc_no) VALUES
('Amit', 101),
('Sneha', 102),
('Rahul', 103),
('Priya', 104),
('Kiran', 105);
```

Step 5: Insert Data into Loan Table

```
INSERT INTO Loan (loan_no, branch_name, amount) VALUES
(201, 'Wadia College', 10000),
(202, 'Camp Branch', 15000),
(203, 'Wadia College', 25000),
(204, 'Deccan', 18000),
(205, 'MG Road', 12000);
```

Step 6: Insert Data into Borrower Table

```
INSERT INTO Borrower (cust_name, loan_no) VALUES
('Amit', 201),
('Sneha', 202),
```

```
('Rahul', 203),
('Priya', 204),
('Kiran', 205);
```

Step 7: Test the Queries

✅ **Query 1 – Find the branches where average account balance > 15000**

```
SELECT branch_name, AVG(balance) AS avg_balance
FROM Account
GROUP BY branch_name
HAVING AVG(balance) > 15000;
```

Query 2 – Find number of tuples (rows) in Customer relation

```
SELECT COUNT(*) AS total_customers
FROM Customer;
```

Query 3 – Calculate total loan amount given by the bank

```
SELECT SUM(amount) AS total_loan_amount
FROM Loan;
```

Query 4 – Delete all loans with amount between 1300 and 1500

```
DELETE FROM Loan
WHERE amount BETWEEN 1300 AND 1500;
```

Query 5 – Find the average account balance at each branch

```
SELECT branch_name, AVG(balance) AS avg_balance
FROM Account
GROUP BY branch_name;
```

Query 6-Find name and city of customers where name starts with letter 'P'

```
SELECT cust_name, cust_city
FROM Customer
WHERE cust_name LIKE 'P%';
```