Bank database Management System
1. Customer (customer_id, name, address, phone, email)
2. Account (account_id, customer_id, account_type, balance, branch_id)
3. Branch (branch_id, branch_name, location, manager_id)
4. Transaction (transaction_id, account_id, transaction_type, amount, transaction_date)
5. Loan (loan_id, customer_id, amount, loan_type, status)
6. Employee (employee_id, name, position, branch_id, salary)

Write queries for the following questions:
1. Find employees working in a specific branch (e.g., Branch ID: 3)
2. Get the details of the highest transaction made
3. Find accounts with a balance less than Rs. 5000
4. Update account balance after a deposit of Rs. 2000 in account ID 105
5. Delete inactive loan applications (status = 'Rejected')
6. Calculate the total loan amount per loan type

```
CREATE TABLE Customer (
    customer_id INT PRIMARY KEY,
    name VARCHAR(100),
    address VARCHAR(200),
    phone VARCHAR(15),
    email VARCHAR(100)
);

CREATE TABLE Branch (
    branch_id INT PRIMARY KEY,
    branch_name VARCHAR(100),
    location VARCHAR(100),
    manager_id INT
);

CREATE TABLE Account (
    account_id INT PRIMARY KEY,
    customer_id INT,
    account_type VARCHAR(20),
    balance DECIMAL(10,2),
    branch_id INT,
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id),
    FOREIGN KEY (branch_id) REFERENCES Branch(branch_id)
);

CREATE TABLE Transaction (
    transaction_id INT PRIMARY KEY,
    account_id INT,
    transaction_type VARCHAR(20),
    amount DECIMAL(10,2),
```

```
    transaction_date DATE,
    FOREIGN KEY (account_id) REFERENCES Account(account_id)
);

CREATE TABLE Loan (
    loan_id INT PRIMARY KEY,
    customer_id INT,
    amount DECIMAL(10,2),
    loan_type VARCHAR(50),
    status VARCHAR(20),
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
);

CREATE TABLE Employee (
    employee_id INT PRIMARY KEY,
    name VARCHAR(100),
    position VARCHAR(50),
    branch_id INT,
    salary DECIMAL(10,2),
    FOREIGN KEY (branch_id) REFERENCES Branch(branch_id)
);

-- Customers
INSERT INTO Customer VALUES
(1, 'Ravi', 'Mumbai', '9876543210', 'ravi@gmail.com'),
(2, 'Anita', 'Delhi', '9123456789', 'anita@gmail.com');

-- Branches
INSERT INTO Branch VALUES
(1, 'Main', 'Mumbai', 101),
(2, 'South', 'Delhi', 102),
(3, 'North', 'Chennai', 103);

-- Accounts
INSERT INTO Account VALUES
(101, 1, 'Savings', 8000, 1),
(102, 2, 'Current', 4000, 3),
(105, 1, 'Savings', 10000, 3);

-- Transactions
INSERT INTO Transaction VALUES
(1, 101, 'Deposit', 3000, '2024-12-01'),
(2, 102, 'Withdrawal', 500, '2024-12-03'),
(3, 101, 'Deposit', 20000, '2024-12-05');

-- Loans
INSERT INTO Loan VALUES
(1, 1, 150000, 'Home Loan', 'Approved'),
(2, 2, 30000, 'Personal Loan', 'Rejected');
```

```
-- Employees
INSERT INTO Employee VALUES
(101, 'Manoj', 'Manager', 1, 50000),
(102, 'Sita', 'Clerk', 3, 20000),
(103, 'Arjun', 'Assistant', 3, 18000);

-- Find employees working in a specific branch (e.g., Branch ID: 3)
SELECT * FROM Employee WHERE branch_id = 3;

-- Get the details of the highest transaction made
SELECT * FROM Transaction
ORDER BY amount DESC
LIMIT 1;

-- Find accounts with a balance less than Rs. 5000
SELECT * FROM Account
WHERE balance < 5000;

-- Update account balance after a deposit of Rs. 2000 in account ID 105
UPDATE Account
SET balance = balance + 2000
WHERE account_id = 105;

-- Delete inactive loan applications (status = 'Rejected')
SET SQL_SAFE_UPDATES = 0;

DELETE FROM Loan
WHERE status = 'Rejected';

SET SQL_SAFE_UPDATES = 1;

-- Calculate the total loan amount per loan type
SELECT loan_type, SUM(amount) AS total_amount
FROM Loan
GROUP BY loan_type;
```