```
-- Create Branch table
CREATE TABLE branch (
    branch_id INT NOT NULL AUTO_INCREMENT,
    branch_name VARCHAR(50) NOT NULL,
    address VARCHAR(100) NOT NULL,
    PRIMARY KEY (branch_id)
);

-- Create Banker table
CREATE TABLE banker (
    banker_id INT NOT NULL AUTO_INCREMENT,
    branch_id INT NOT NULL,
    banker_name VARCHAR(50) NOT NULL,
    PRIMARY KEY (banker_id),
    FOREIGN KEY (branch_id) REFERENCES branch(branch_id)
);

-- Create Account table
CREATE TABLE account (
    account_id INT NOT NULL AUTO_INCREMENT,
    balance DECIMAL(10, 2) NOT NULL,
    account_type ENUM('student', 'savings') NOT NULL,
    branch_id INT NOT NULL,
    PRIMARY KEY (account_id),
    FOREIGN KEY (branch_id) REFERENCES branch(branch_id)
);

-- Create Customer table
CREATE TABLE customer (
    customer_id INT NOT NULL AUTO_INCREMENT,
    customer_name VARCHAR(30) NOT NULL,
    mobileno VARCHAR(10) NOT NULL,
    dob DATE,
    account_id INT NOT NULL,
    PRIMARY KEY (customer_id),
    FOREIGN KEY (account_id) REFERENCES account(account_id)
);

-- Create Loan table
CREATE TABLE loan (
    loan_id INT NOT NULL AUTO_INCREMENT,
    issue_date DATE NOT NULL,
    remaining_balance DECIMAL(10, 2) NOT NULL,
    amount DECIMAL(10, 2) NOT NULL,
    loan_limit DECIMAL(10, 2) NOT NULL, -- Added loan_limit column
    branch_id INT NOT NULL,
    account_id INT NOT NULL,
    PRIMARY KEY (loan_id),
    FOREIGN KEY (branch_id) REFERENCES branch(branch_id),
    FOREIGN KEY (account_id) REFERENCES account(account_id)
);
```

```
-- Create Loan Pay table
CREATE TABLE loan_pay (
    payment_id INT NOT NULL AUTO_INCREMENT,
    loan_id INT NOT NULL,
    amount DECIMAL(10, 2) NOT NULL,
    PRIMARY KEY (payment_id),
    FOREIGN KEY (loan_id) REFERENCES loan(loan_id)
);

-- Create Transaction table
CREATE TABLE transaction (
    transaction_id INT NOT NULL AUTO_INCREMENT,
    account_id INT NOT NULL,
    customer_id INT NOT NULL,
    amount DECIMAL(10, 2) NOT NULL,
    PRIMARY KEY (transaction_id),
    FOREIGN KEY (account_id) REFERENCES account(account_id),
    FOREIGN KEY (customer_id) REFERENCES customer(customer_id)
);

-- Create Customer Credit Card table
CREATE TABLE customer_credit_card (
    credit_card_id INT NOT NULL AUTO_INCREMENT,
    expiry_date DATE NOT NULL,
    card_limit INT NOT NULL,
    customer_id INT NOT NULL,
    account_id INT NOT NULL,
    PRIMARY KEY (credit_card_id),
    FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
    FOREIGN KEY (account_id) REFERENCES account(account_id)
);

-- Insert data into Branch table
INSERT INTO branch (branch_name, address) VALUES
('Farmgate Branch', '123 Dhanmondi'),
('Green-road Branch', '456 Farmgate'),
('Dhanmondi Branch', '789 Dhanmondi'),
('Gulshan Branch', '101 Gulshan');

-- Insert data into Banker table
INSERT INTO banker (branch_id, banker_name) VALUES
(1, 'Aerifur Rahman'),
(2, 'Mamim'),
(3, 'Urmi'),
(4, 'Sajid Sahan');

-- Insert data into Account table
INSERT INTO account (balance, account_type, branch_id) VALUES
(1000.00, 'savings', 1),
(1500.00, 'student', 2),
```

```
(2000.00, 'savings', 3),
(2500.00, 'student', 4);

-- Insert data into Customer table
INSERT INTO customer (customer_name, mobileno, dob, account_id) VALUES
('Sakib', '015555555', '2000-03-15', 1),
('Riyad', '017777777', '2002-03-25', 2),
('rifat panda', '0133333333', '2002-10-01', 3),
('Sajid', '019999999', '2002-01-20', 4);

-- Insert data into Loan table
INSERT INTO loan (issue_date, remaining_balance, amount, loan_limit, branch_id, account_id) VALUES
('2023-01-01', 500.00, 1000.00, 1200.00, 1, 1),
('2023-02-01', 750.00, 1500.00, 1800.00, 2, 2),
('2023-03-01', 1000.00, 2000.00, 2500.00, 3, 3),
('2023-04-01', 1250.00, 2500.00, 3000.00, 4, 4);

-- Update the loan limit for all loans
UPDATE loan
SET loan_limit = amount * 1.2;

-- Insert data into Loan Pay table
INSERT INTO loan_pay (loan_id, amount) VALUES
(1, 100.00),
(2, 200.00),
(3, 300.00),
(4, 400.00);

-- Insert data into Customer Credit Card table
INSERT INTO customer_credit_card (expiry_date, card_limit, customer_id, account_id) VALUES
('2025-12-31', 5000, 1, 1),
('2026-11-30', 6000, 2, 2),
('2027-10-31', 7000, 3, 3),
('2028-09-30', 8000, 4, 4);

-- Insert data into Transaction table
INSERT INTO transaction (account_id, customer_id, amount) VALUES
(1, 1, 100.00),
(2, 2, 200.00),
(3, 3, 300.00),
(4, 4, 400.00);
-- 1. Print all data from Branch table
SELECT * FROM branch;

-- 2. Print all data from Banker table
SELECT * FROM banker;

-- 3. Print all data from Account table
SELECT * FROM account;

-- 4. Print all data from Customer table
```

```
SELECT * FROM customer;

-- 5. Print all data from Loan table
SELECT * FROM loan;

-- 6. Print all data from Loan Pay table
SELECT * FROM loan_pay;

-- 7. Print all data from Transaction table
SELECT * FROM transaction;

-- 8. Print all data from Customer Credit Card table
SELECT * FROM customer_credit_card;

-- 1. Find all customers with a savings account

SELECT customer_name, mobileno, dob
FROM customer
INNER JOIN account ON customer.account_id = account.account_id
WHERE account_type = 'savings';

-- 2. Get the total loan amount issued by each branch
SELECT branch.branch_name, SUM(loan.amount) AS total_loan_amount
FROM loan
INNER JOIN branch ON loan.branch_id = branch.branch_id
GROUP BY branch.branch_name;



-- 3. List all transactions made by a specific customer (sakib)

SELECT transaction.transaction_id, transaction.amount, transaction.account_id
FROM transaction
INNER JOIN customer ON transaction.customer_id = customer.customer_id
WHERE customer.customer_name = 'sakib';



-- 4. Find all accounts that have a balance greater than $2000

SELECT account_id, balance, account_type
FROM account
WHERE balance > 2000;

-- 5. Calculate the remaining loan balance per branch

SELECT branch.branch_name, SUM(loan.remaining_balance) AS total_remaining_balance
FROM loan
INNER JOIN branch ON loan.branch_id = branch.branch_id
GROUP BY branch.branch_name;
```

```
-- 6. Count the number of accounts per account type (student or savings)

SELECT account_type, COUNT(account_id) AS number_of_accounts
FROM account
GROUP BY account_type;

-- 7. Retrieve all loan payments for a specific loan

SELECT payment_id, amount
FROM loan_pay
WHERE loan_id = 1;

-- 8. Get all customers with their credit card limit
SELECT customer.customer_name, customer_credit_card.card_limit
FROM customer
INNER JOIN customer_credit_card ON customer.customer_id = customer_credit_card.customer_id;

-- 9. Show all bankers working at a specific branch
SELECT banker.banker_name
FROM banker
INNER JOIN branch ON banker.branch_id = branch.branch_id
WHERE branch.branch_name = 'Main Branch';

-- 10. Find customers who were born before 1980
SELECT customer_name, dob
FROM customer
WHERE dob < '2000-01-01';

-- 11. List all branches with the number of bankers working in each
SELECT branch.branch_name, COUNT(banker.banker_id) AS number_of_bankers
FROM branch
LEFT JOIN banker ON branch.branch_id = banker.branch_id
GROUP BY branch.branch_name;

-- 12. Calculate the average balance of accounts for each branch
SELECT branch.branch_name, AVG(account.balance) AS average_balance
FROM branch
INNER JOIN account ON branch.branch_id = account.branch_id
GROUP BY branch.branch_name;

-- 13. Find all customers who have both a loan and a credit card
SELECT DISTINCT customer.customer_name
FROM customer
INNER JOIN loan ON customer.account_id = loan.account_id
INNER JOIN customer_credit_card ON customer.customer_id = customer_credit_card.customer_id;

-- 14. Retrieve the highest loan amount issued at each branch
SELECT branch.branch_name, MAX(loan.amount) AS highest_loan_amount
FROM loan
INNER JOIN branch ON loan.branch_id = branch.branch_id
GROUP BY branch.branch_name;
```

```
-- 15. List customers with transactions above a certain amount (e.g., $250)
SELECT customer.customer_name, transaction.amount
FROM transaction
INNER JOIN customer ON transaction.customer_id = customer.customer_id
WHERE transaction.amount > 250;

-- 16. Get the total amount of loan payments made for each loan
SELECT loan_id, SUM(amount) AS total_payment
FROM loan_pay
GROUP BY loan_id;

-- 17. List all accounts with the number of transactions associated with each
SELECT account.account_id, COUNT(transaction.transaction_id) AS number_of_transactions
FROM account
LEFT JOIN transaction ON account.account_id = transaction.account_id
GROUP BY account.account_id;

-- 18. Retrieve customers who have a credit card with a limit greater than $6000
SELECT customer.customer_name, customer_credit_card.card_limit
FROM customer
INNER JOIN customer_credit_card ON customer.customer_id = customer_credit_card.customer_id
WHERE customer_credit_card.card_limit > 6000;

-- 19. Find all customers who have not made any transactions
SELECT customer.customer_name
FROM customer
LEFT JOIN transaction ON customer.customer_id = transaction.customer_id
WHERE transaction.transaction_id IS NULL;

-- 20. Show the total balance of all accounts in each branch
SELECT branch.branch_name, SUM(account.balance) AS total_branch_balance
FROM branch
INNER JOIN account ON branch.branch_id = account.branch_id
GROUP BY branch.branch_name;

-- 21. List all loans with their corresponding customers and branches
SELECT loan.loan_id, loan.amount, customer.customer_name, branch.branch_name
FROM loan
INNER JOIN customer ON loan.account_id = customer.account_id
INNER JOIN branch ON loan.branch_id = branch.branch_id;

-- 22. Find the customer with the highest account balance
SELECT customer.customer_name, account.balance
FROM customer
INNER JOIN account ON customer.account_id = account.account_id
ORDER BY account.balance DESC
LIMIT 1;

-- 23. Calculate the total number of loans and total loan amount per branch
```

```
SELECT branch.branch_name, COUNT(loan.loan_id) AS number_of_loans, SUM(loan.amount) AS total_loan_amount
FROM branch
LEFT JOIN loan ON branch.branch_id = loan.branch_id
GROUP BY branch.branch_name;

-- 24. List all accounts and their customers who were born in 2000 or later
SELECT account.account_id, customer.customer_name, customer.dob
FROM account
INNER JOIN customer ON account.account_id = customer.account_id
WHERE customer.dob >= '2000-01-01';

-- 25. Find all credit card holders with their account balances and credit limits
SELECT customer.customer_name, account.balance, customer_credit_card.card_limit
FROM customer
INNER JOIN account ON customer.account_id = account.account_id
INNER JOIN customer_credit_card ON customer.customer_id = customer_credit_card.customer_id;



-- Retrieve all loan and branch details using NATURAL JOIN
SELECT *
FROM loan
NATURAL JOIN branch;

-- Get all customer details and account balances using NATURAL JOIN
SELECT *
FROM customer
NATURAL JOIN account;

-- Find customers who have a loan and their loan details using NATURAL JOIN
SELECT *
FROM customer
NATURAL JOIN loan;

-- Example 1: Find all transactions for a specific customer
          SELECT
       transaction.transaction_id, transaction.amount, transaction.account_id
              FROM transaction

         INNER JOIN customer ON transaction.customer_id = customer.customer_id
            WHERE customer.customer_name = 'nonexistent_customer'
UNION
           SELECT NULL, 'No Related Data', NULL
                WHERE NOT EXISTS (
                    SELECT 1
            FROM transaction
                  INNER JOIN customer ON transaction.customer_id = customer.customer_id
                      WHERE customer.customer_name = 'nonexistent_customer'
);
```

```
        SELECT CAST(balance AS CHAR) AS balance
            FROM account
    WHERE balance > 5000
     UNION
    SELECT 'No Related Data'
        WHERE NOT EXISTS (
      SELECT 1
          FROM account
             WHERE balance > 5000
           );
```

```
SELECT l.loan_limit
FROM customer c
JOIN account a ON c.account_id = a.account_id
JOIN loan l ON a.account_id = l.account_id
WHERE c.customer_name = 'Sakib';
```