**Schema Creation:**

CREATE TABLE Customers (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

DOB DATE,

Balance NUMBER,

LastModified DATE

);

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (1, 'John Doe', TO\_DATE('1985-05-15', 'YYYY-MM-DD'), 1000, SYSDATE);

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (2, 'Jane Smith', TO\_DATE('1990-07-20', 'YYYY-MM-DD'), 1500, SYSDATE);

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (3, 'Raj Malhotra', TO\_DATE('1962-03-10', 'YYYY-MM-DD'), 12000, SYSDATE);

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (4, 'Emily Davis', TO\_DATE('1975-11-30', 'YYYY-MM-DD'), 8700, SYSDATE);

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (5, 'Carlos Rivera', TO\_DATE('1958-01-22', 'YYYY-MM-DD'), 25000, SYSDATE);

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (6, 'Ayesha Khan', TO\_DATE('1982-08-14', 'YYYY-MM-DD'), 950, SYSDATE);

CREATE TABLE Loans (

LoanID NUMBER PRIMARY KEY,

CustomerID NUMBER,

LoanAmount NUMBER,

InterestRate NUMBER,

StartDate DATE,

EndDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (1, 1, 5000, 5, SYSDATE, ADD\_MONTHS(SYSDATE, 60));

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (2, 2, 8000, 6, SYSDATE, ADD\_MONTHS(SYSDATE, 48));

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (3, 3, 15000, 4.5, SYSDATE, ADD\_MONTHS(SYSDATE, 36));

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (4, 4, 10000, 5.5, SYSDATE, ADD\_MONTHS(SYSDATE, 24));

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (5, 5, 20000, 6.2, SYSDATE, ADD\_MONTHS(SYSDATE, 72));

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (6, 6, 3000, 7, SYSDATE, ADD\_MONTHS(SYSDATE, 12));

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (7, 1, 2500, 5.5, SYSDATE, SYSDATE + 10);

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (8, 4, 4000, 6.0, SYSDATE, SYSDATE + 25);

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (9, 2, 6000, 5.8, SYSDATE, ADD\_MONTHS(SYSDATE, 18));

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (10, 5, 10000, 6.5, SYSDATE, ADD\_MONTHS(SYSDATE, 36));

**Scenario 1**: The bank wants to apply a discount to loan interest rates for customers above 60 years old.

**Question**: Write a PL/SQL block that loops through all customers, checks their age, and if they are above 60, apply a 1% discount to their current loan interest rates.

**Solution**:

BEGIN

FOR customer\_rec IN (

SELECT l.loanid, l.customerid, l.interestrate

FROM loans l

JOIN customers c ON l.customerid = c.customerid

WHERE MONTHS\_BETWEEN(SYSDATE, c.DOB) / 12 > 60

)

LOOP

UPDATE loans

SET interestrate = customer\_rec.interestrate - 1

WHERE loanid = customer\_rec.loanid;

DBMS\_OUTPUT.PUT\_LINE('1% discount applied to Loan ID ' || customer\_rec.loanid ||

' (Customer ID: ' || customer\_rec.customerid ||

', New Interest Rate: ' || (customer\_rec.interestrate - 1) || '%)');

END LOOP;

COMMIT;

END;

/

SELECT l.loan\_id, l.customer\_id, l.interest\_rate, c.name, c.dob

FROM loans l

JOIN customers c ON l.customer\_id = c.customer\_id

WHERE MONTHS\_BETWEEN(SYSDATE, c.DOB) / 12 > 60;

**Scenario** 2: A customer can be promoted to VIP status based on their balance.

**Question**: Write a PL/SQL block that iterates through all customers and sets a flag IsVIP to TRUE for those with a balance over $10,000.

**Solution**:

ALTER TABLE customers

ADD isVIP VARCHAR2(5);

UPDATE customers

SET isVIP = 'FALSE';

COMMIT;

BEGIN

FOR customer\_rec IN (SELECT customerid

FROM customers

WHERE balance > 10000)

LOOP

UPDATE customers

SET isVIP = 'TRUE'

WHERE customerid = customer\_rec.customerid;

END LOOP;

COMMIT;

END;

/

SELECT customerid, name, balance, isVIP

FROM customers

ORDER BY customerid;

**Scenario** 3: The bank wants to send reminders to customers whose loans are due within the next 30 days.

**Question**: Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.

**Solution**:

BEGIN

FOR loan\_rec IN (

SELECT customerid, enddate

FROM loans

WHERE enddate <= SYSDATE + 30

)

LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan for Customer ID ' || loan\_rec.customerid ||

' is due on ' || TO\_CHAR(loan\_rec.enddate, 'DD-MON-YYYY'));

END LOOP;

END;

/