**WEEK 2 Exercise 1: Control Structures**

**SUPERSET ID : 6427473**

**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

CODE :

**Step 1: Create Required Tables**

CREATE TABLE Customers (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

DOB DATE,

Balance NUMBER,

IsVIP VARCHAR2(5)

);

CREATE TABLE Loans (

LoanID NUMBER PRIMARY KEY,

CustomerID NUMBER,

LoanAmount NUMBER,

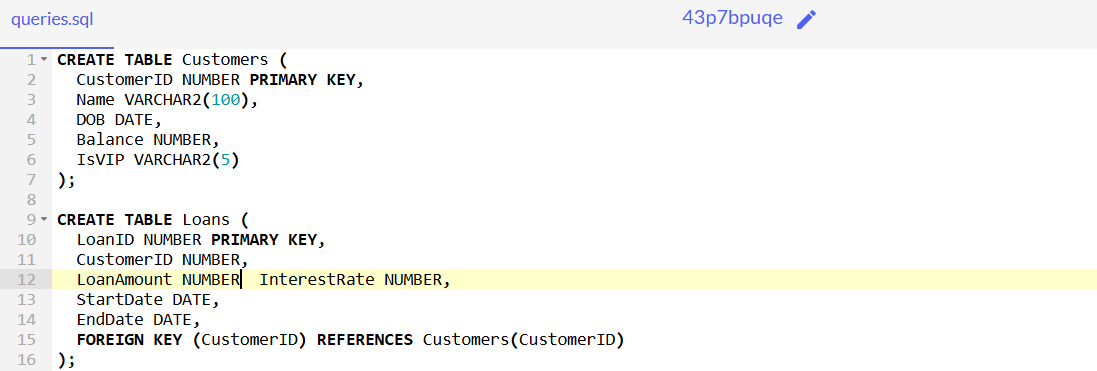
InterestRate NUMBER,

StartDate DATE,

EndDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);



**Step 2: Insert Sample Data**

**-- Insert customers**

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (1, 'John Doe', TO\_DATE('1950-01-01', 'YYYY-MM-DD'), 15000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (2, 'Jane Smith', TO\_DATE('1980-01-01', 'YYYY-MM-DD'), 8000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (3, 'Sundar Raj', TO\_DATE('1945-03-12', 'YYYY-MM-DD'), 25000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (4, 'Meena Kumari', TO\_DATE('1965-08-25', 'YYYY-MM-DD'), 15000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (5, 'Vignesh Kumar', TO\_DATE('1970-11-30', 'YYYY-MM-DD'), 11000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (6, 'Priya Mohan', TO\_DATE('1953-06-10', 'YYYY-MM-DD'), 18000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (7, 'Arun Vijay', TO\_DATE('1995-01-05', 'YYYY-MM-DD'), 7000, 'FALSE');

**-- Insert loans**

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

VALUES (1, 1, 50000, 5, SYSDATE, ADD\_MONTHS(SYSDATE, 24));

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

VALUES (2, 2, 30000, 6, SYSDATE, ADD\_MONTHS(SYSDATE, 36));

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

VALUES (3, 3, 40000, 7, SYSDATE, ADD\_MONTHS(SYSDATE, 36));

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

VALUES (4, 4, 30000, 6, SYSDATE, ADD\_MONTHS(SYSDATE, 24));

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

VALUES (5, 5, 35000, 6.5, SYSDATE, ADD\_MONTHS(SYSDATE, 36));

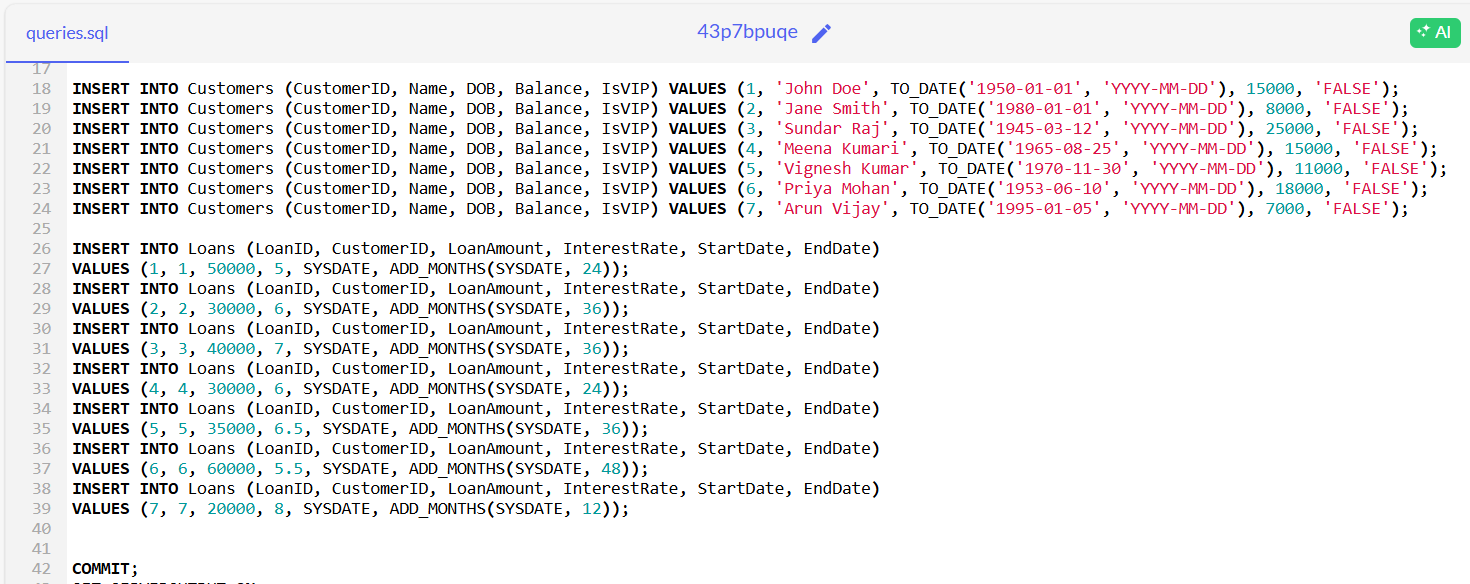
INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (6, 6, 60000, 5.5, SYSDATE, ADD\_MONTHS(SYSDATE, 48));

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

VALUES (7, 7, 20000, 8, SYSDATE, ADD\_MONTHS(SYSDATE, 12));

COMMIT;



**Step 3: PL/SQL Block to Apply Interest Discount**

**SET SERVEROUTPUT ON;**

BEGIN

FOR cust\_rec IN (

SELECT l.LoanID, c.Name, c.DOB, l.InterestRate

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

) LOOP

DECLARE

v\_age NUMBER;

BEGIN

v\_age := FLOOR(MONTHS\_BETWEEN(SYSDATE, cust\_rec.DOB) / 12);

IF v\_age > 60 THEN

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE LoanID = cust\_rec.LoanID;

DBMS\_OUTPUT.PUT\_LINE('Discount applied to ' || cust\_rec.Name ||

' (Age: ' || v\_age || ', LoanID: ' || cust\_rec.LoanID ||

', New Interest: ' || (cust\_rec.InterestRate - 1) || '%)');

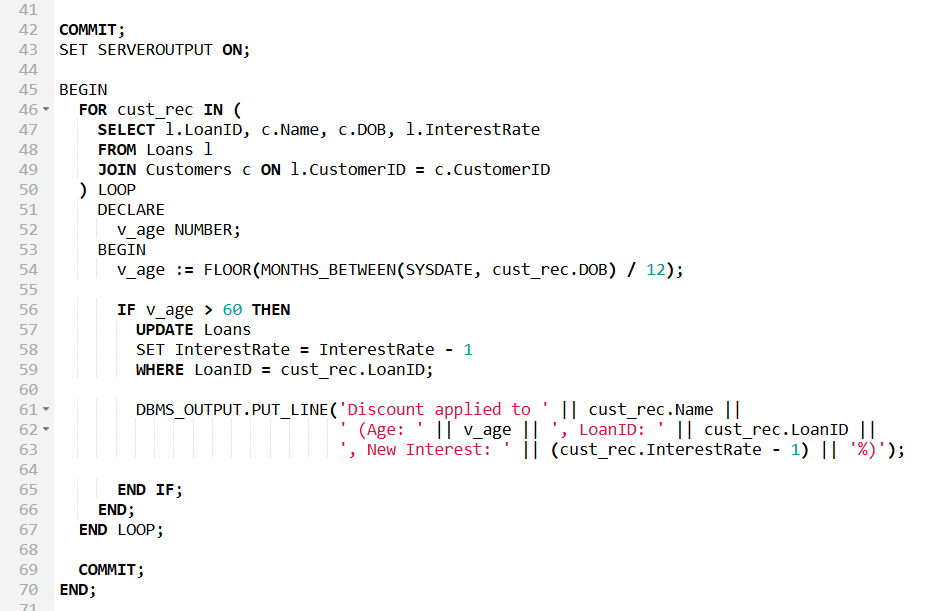
END IF;

END;

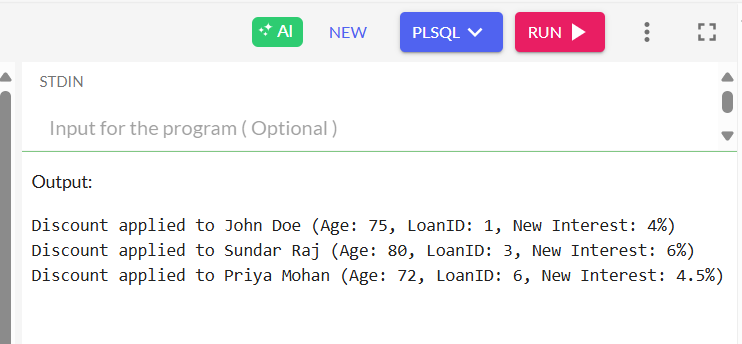
END LOOP;

COMMIT;

END;

****

**OUTPUT – SCREENSHOTS**

****

**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.

**Code :**

**Step 1: Create the Tables**

CREATE TABLE Customers (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

DOB DATE,

Balance NUMBER,

IsVIP VARCHAR2(5)

);

CREATE TABLE Loans (

LoanID NUMBER PRIMARY KEY,

CustomerID NUMBER,

LoanAmount NUMBER,

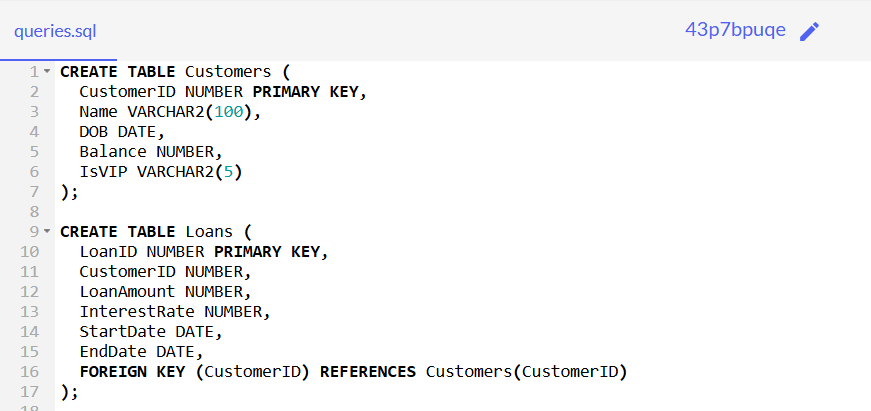
InterestRate NUMBER,

StartDate DATE,

EndDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);



**Step 2: Insert Data**

**-- Customers**

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (1, 'John Doe', TO\_DATE('1950-01-01', 'YYYY-MM-DD'), 15000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (2, 'Jane Smith', TO\_DATE('1980-01-01', 'YYYY-MM-DD'), 8000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (3, 'Sundar Raj', TO\_DATE('1945-03-12', 'YYYY-MM-DD'), 25000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (4, 'Meena Kumari', TO\_DATE('1965-08-25', 'YYYY-MM-DD'), 15000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (5, 'Vignesh Kumar', TO\_DATE('1970-11-30', 'YYYY-MM-DD'), 11000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (6, 'Priya Mohan', TO\_DATE('1953-06-10', 'YYYY-MM-DD'), 18000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (7, 'Arun Vijay', TO\_DATE('1995-01-05', 'YYYY-MM-DD'), 7000, 'FALSE');

**-- Loans**

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

VALUES (1, 1, 50000, 5, SYSDATE, ADD\_MONTHS(SYSDATE, 24));

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

VALUES (2, 2, 30000, 6, SYSDATE, ADD\_MONTHS(SYSDATE, 36));

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

VALUES (3, 3, 40000, 7, SYSDATE, ADD\_MONTHS(SYSDATE, 36));

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

VALUES (4, 4, 30000, 6, SYSDATE, ADD\_MONTHS(SYSDATE, 24));

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

VALUES (5, 5, 35000, 6.5, SYSDATE, ADD\_MONTHS(SYSDATE, 36));

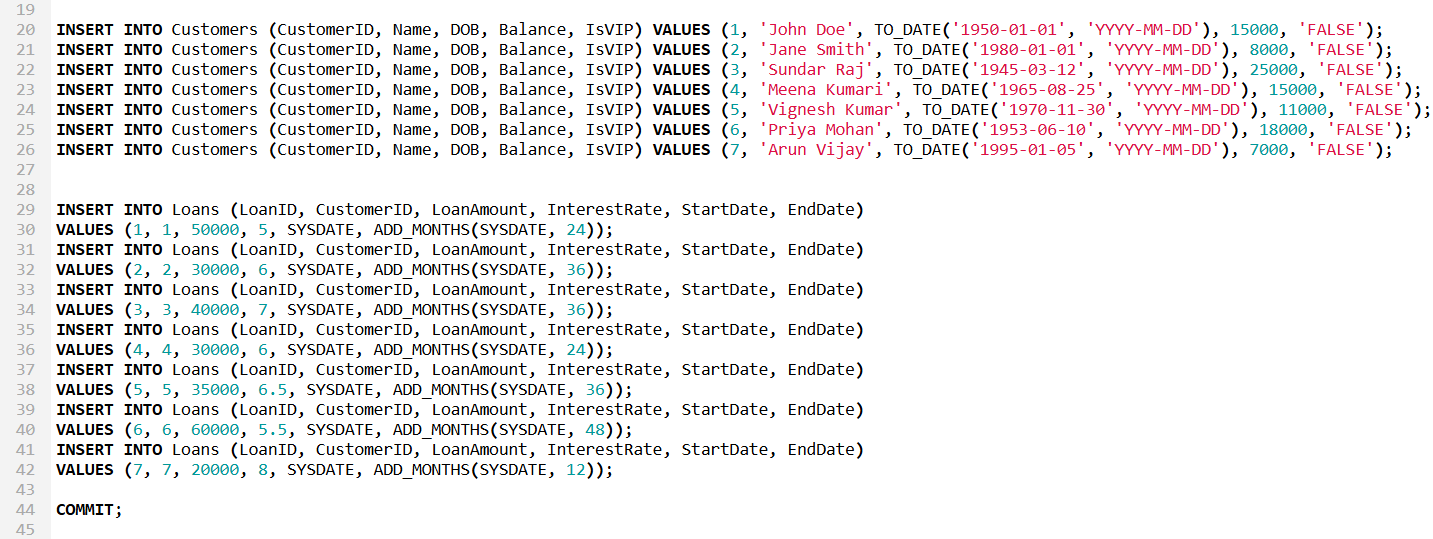
INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (6, 6, 60000, 5.5, SYSDATE, ADD\_MONTHS(SYSDATE, 48));

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

VALUES (7, 7, 20000, 8, SYSDATE, ADD\_MONTHS(SYSDATE, 12));

COMMIT;



**Step 3: PL/SQL Block for VIP Promotion**

SET SERVEROUTPUT ON;

BEGIN

FOR cust\_rec IN (

SELECT CustomerID, Name, Balance

FROM Customers

) LOOP

IF cust\_rec.Balance > 10000 THEN

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = cust\_rec.CustomerID;

DBMS\_OUTPUT.PUT\_LINE('VIP status applied to ' || cust\_rec.Name ||

' (CustomerID: ' || cust\_rec.CustomerID ||

', Balance: $' || cust\_rec.Balance || ')');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Not eligible for VIP: ' || cust\_rec.Name ||

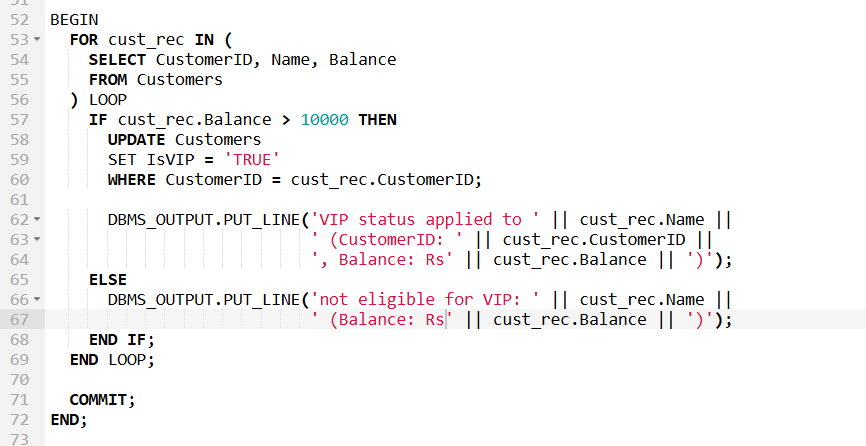
' (Balance: $' || cust\_rec.Balance || ')');

END IF;

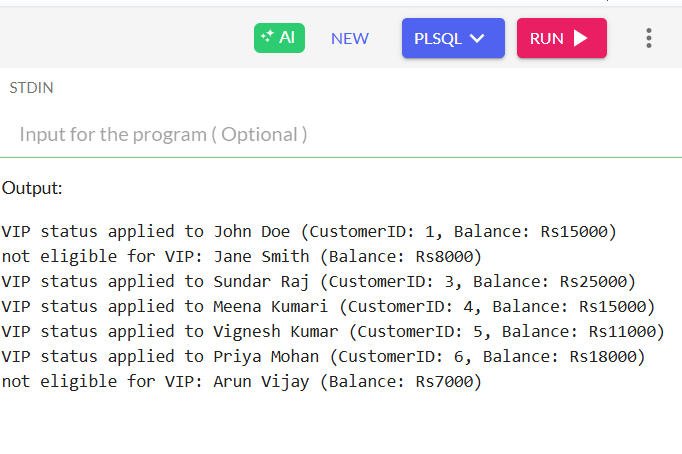
END LOOP;

COMMIT;

END;



**Output Screenshot**

****

**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.

**Step 1: Create the Tables**

CREATE TABLE Customers (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

DOB DATE,

Balance NUMBER,

IsVIP VARCHAR2(5)

);

CREATE TABLE Loans (

LoanID NUMBER PRIMARY KEY,

CustomerID NUMBER,

LoanAmount NUMBER,

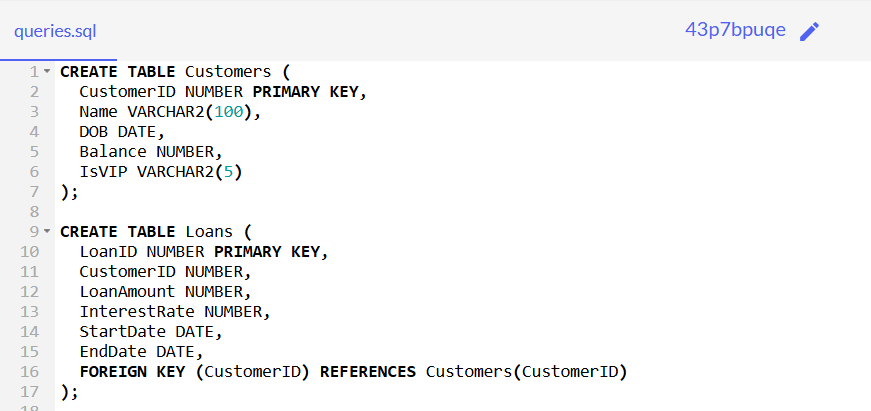
InterestRate NUMBER,

StartDate DATE,

EndDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);



**Step 2: Insert the Sample inputs**

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (1, 'John Doe', TO\_DATE('1950-01-01', 'YYYY-MM-DD'), 15000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (2, 'Jane Smith', TO\_DATE('1980-01-01', 'YYYY-MM-DD'), 8000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (3, 'Sundar Raj', TO\_DATE('1945-03-12', 'YYYY-MM-DD'), 25000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (4, 'Meena Kumari', TO\_DATE('1965-08-25', 'YYYY-MM-DD'), 15000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (5, 'Vignesh Kumar', TO\_DATE('1970-11-30', 'YYYY-MM-DD'), 11000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (6, 'Priya Mohan', TO\_DATE('1953-06-10', 'YYYY-MM-DD'), 18000, 'FALSE');

INSERT INTO Customers (CustomerID, Name, DOB, Balance, IsVIP) VALUES (7, 'Arun Vijay', TO\_DATE('1995-01-05', 'YYYY-MM-DD'), 7000, 'FALSE');

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

VALUES (1, 1, 50000, 5, SYSDATE, ADD\_MONTHS(SYSDATE, 24));

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

VALUES (2, 2, 30000, 6, SYSDATE, ADD\_MONTHS(SYSDATE, 36));

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

VALUES (3, 3, 40000, 7, SYSDATE, ADD\_MONTHS(SYSDATE, 36));

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

VALUES (4, 4, 30000, 6, SYSDATE, ADD\_MONTHS(SYSDATE, 24));

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

VALUES (5, 5, 35000, 6.5, SYSDATE, ADD\_MONTHS(SYSDATE, 36));

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

VALUES (6, 6, 60000, 5.5, SYSDATE, ADD\_MONTHS(SYSDATE, 48));

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

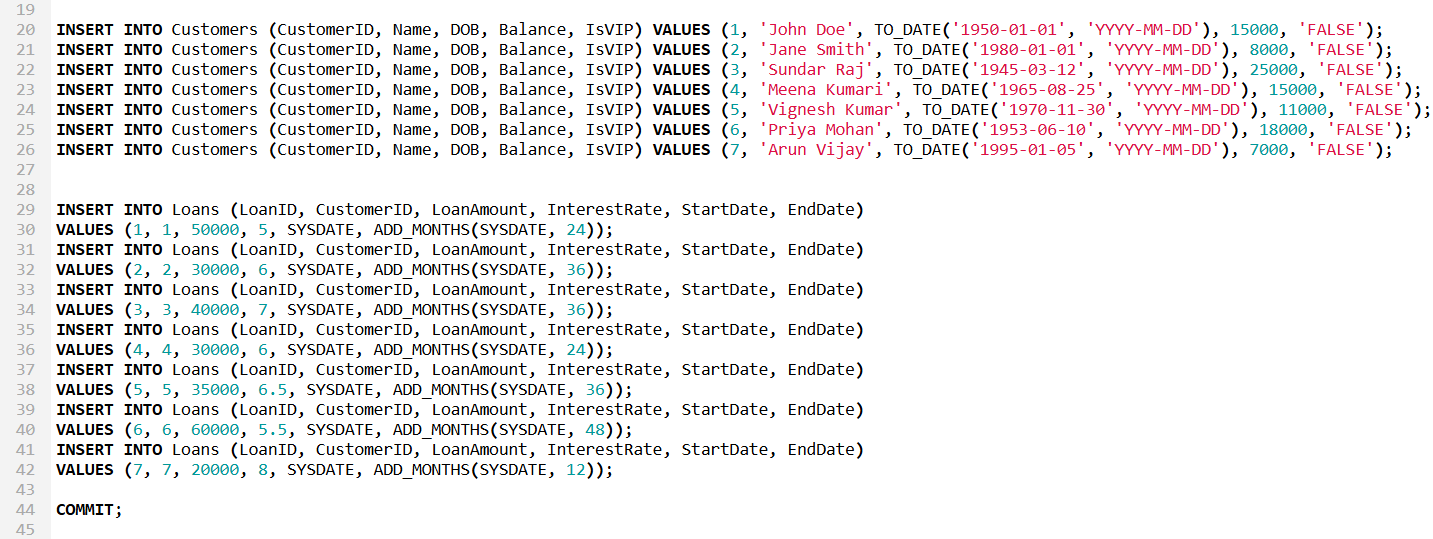
VALUES (7, 7, 20000, 8, SYSDATE, ADD\_MONTHS(SYSDATE, 12));

UPDATE Loans

SET EndDate = SYSDATE + 10

WHERE LoanID = 7;

COMMIT;



**Step 3: PL/SQL Block to Display Reminders**

BEGIN

FOR rec IN (

SELECT c.Name, l.LoanID, l.EndDate

FROM Customers c

JOIN Loans l ON c.CustomerID = l.CustomerID

WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30

) LOOP

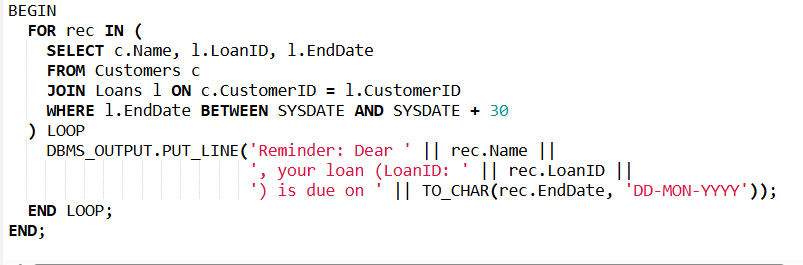
DBMS\_OUTPUT.PUT\_LINE('Reminder: Dear ' || rec.Name ||

', your loan (LoanID: ' || rec.LoanID ||

') is due on ' || TO\_CHAR(rec.EndDate, 'DD-MON-YYYY'));

END LOOP;

END;



Expected Output:

