**Exercise 1: Control Structures**

**Scenario 1:Discount interest rates for customers above 60**

BEGIN

FOR cust IN (SELECT CustomerID FROM Customers WHERE MONTHS\_BETWEEN(SYSDATE, DOB)/12 > 60) LOOP

UPDATE Loans

SET InterestRate = InterestRate - 1

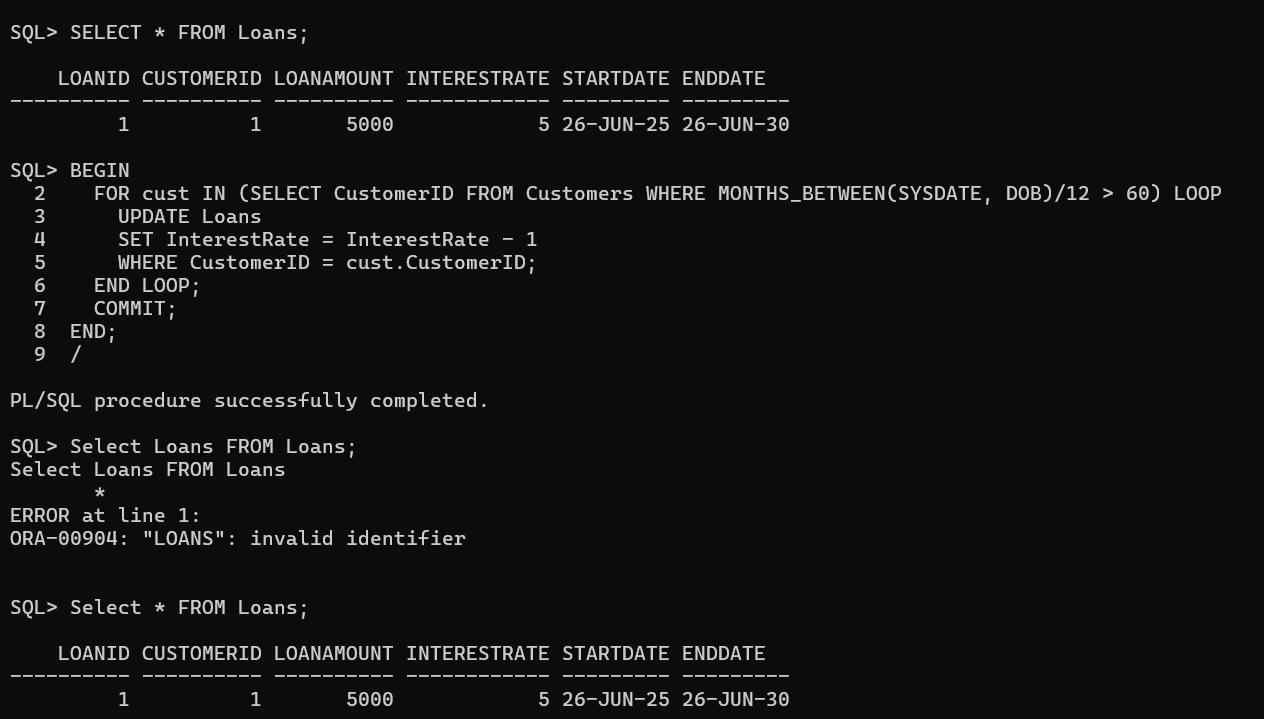
WHERE CustomerID = cust.CustomerID;

END LOOP;

COMMIT;

END;

/



**Scenario 2: Set IsVIP flag**

BEGIN

FOR cust IN (SELECT CustomerID FROM Customers WHERE Balance > 10000) LOOP

UPDATE Customers

SET Balance = Balance, -- just to trigger update

LastModified = SYSDATE

WHERE CustomerID = cust.CustomerID;

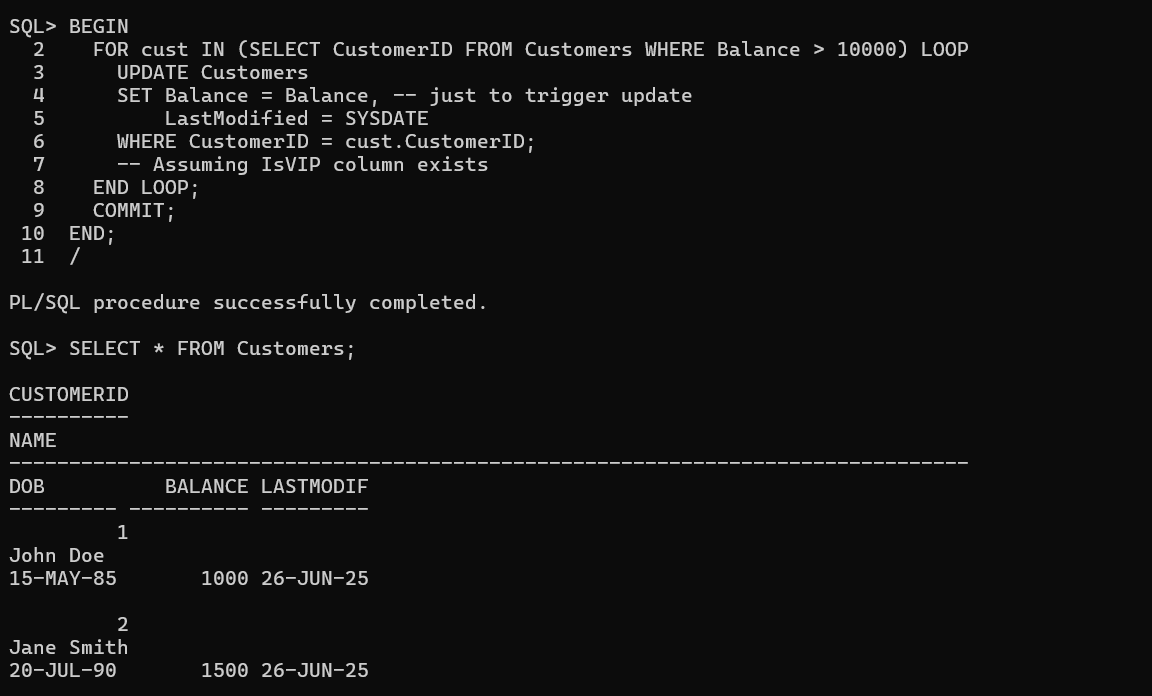
-- Assuming IsVIP column exists

END LOOP;

COMMIT;

END;

/



**Scenario 3: Loan reminders**

BEGIN

FOR loan\_rec IN (

SELECT c.Name, l.DueDate

FROM Loans l JOIN Customers c ON l.CustomerID = c.CustomerID

WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan for ' || loan\_rec.Name || ' is due on ' || TO\_CHAR(loan\_rec.DueDate, 'DD-Mon-YYYY'));

END LOOP;

END;

/

**Exercise 3: Stored Procedures**

**Scenario 1: Process monthly interest**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE Accounts

SET Balance = Balance + (Balance \* 0.01)

WHERE AccountType = 'Savings';

COMMIT;

END;

/

**Scenario 2: Employee bonus**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(p\_dept VARCHAR2, p\_bonus\_percent NUMBER) IS

BEGIN

UPDATE Employees

SET Salary = Salary + (Salary \* p\_bonus\_percent / 100)

WHERE Department = p\_dept;

COMMIT;

END;

/

**Scenario 3: Transfer funds**

CREATE OR REPLACE PROCEDURE TransferFunds(p\_from NUMBER, p\_to NUMBER, p\_amt NUMBER) IS

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance FROM Accounts WHERE AccountID = p\_from;

IF v\_balance >= p\_amt THEN

UPDATE Accounts SET Balance = Balance - p\_amt WHERE AccountID = p\_from;

UPDATE Accounts SET Balance = Balance + p\_amt WHERE AccountID = p\_to;

COMMIT;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Error: Insufficient balance.');

ROLLBACK;

END IF;

END;

/