**Exercise 1: Control Structures**

**Scenario 1:-**

BEGIN

FOR c IN (SELECT CustomerID, DOB FROM Customers) LOOP

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

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE CustomerID = c.CustomerID;

END IF;

END LOOP;

COMMIT;

END;

**Scenario 2:-**

ALTER TABLE Customers ADD IsVIP VARCHAR2(5);

BEGIN

FOR c IN (SELECT CustomerID, Balance FROM Customers) LOOP

IF c.Balance > 10000 THEN

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = c.CustomerID;

ELSE

UPDATE Customers

SET IsVIP = 'FALSE'

WHERE CustomerID = c.CustomerID;

END IF;

END LOOP;

COMMIT;

END;

**Scenario 3:-**

DECLARE

v\_name Customers.Name%TYPE;

BEGIN

FOR l IN (

SELECT LoanID, CustomerID, EndDate

FROM Loans

WHERE EndDate BETWEEN SYSDATE AND SYSDATE + 30

) LOOP

SELECT Name INTO v\_name FROM Customers WHERE CustomerID = l.CustomerID;

DBMS\_OUTPUT.PUT\_LINE('Reminder: Dear ' || v\_name ||

', your loan (Loan ID: ' || l.LoanID ||

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

END LOOP;

END;

Output:-

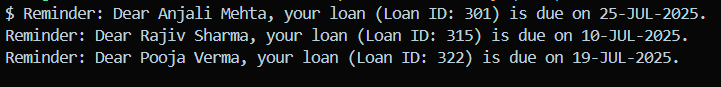
Scenario 1:-



Scenario 2:-



Scenario 3:-



**Exercise 3: Stored Procedures**

**Scenario 1:-**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

FOR acc IN (SELECT AccountID, Balance

FROM Accounts

WHERE AccountType = 'Savings') LOOP

UPDATE Accounts

SET Balance = Balance + (acc.Balance \* 0.01),

LastModified = SYSDATE

WHERE AccountID = acc.AccountID;

END LOOP;

COMMIT;

END;

**Scenario 2:-**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

p\_Department IN VARCHAR2,

p\_BonusPct IN NUMBER

) AS

BEGIN

UPDATE Employees

SET Salary = Salary + (Salary \* p\_BonusPct / 100)

WHERE Department = p\_Department;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error updating bonus: ' || SQLERRM);

END;

**Scenario 3:-**

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_FromAccountID IN NUMBER,

p\_ToAccountID IN NUMBER,

p\_Amount IN NUMBER

) AS

v\_balance NUMBER;

BEGIN

-- Fetch current balance of source account

SELECT Balance INTO v\_balance

FROM Accounts

WHERE AccountID = p\_FromAccountID;

IF v\_balance < p\_Amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds in source account.');

END IF;

-- Perform debit and credit

UPDATE Accounts

SET Balance = Balance - p\_Amount,

LastModified = SYSDATE

WHERE AccountID = p\_FromAccountID;

UPDATE Accounts

SET Balance = Balance + p\_Amount,

LastModified = SYSDATE

WHERE AccountID = p\_ToAccountID;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error transferring funds: ' || SQLERRM);

END;

Output:-

Scenario 1:-



Scenario 2:-



Scenario 3:-

