**WEEK-1**

**Ex- 1 : CONTROL SEQUENCE**

**Scenario 1:**

BEGIN

FOR rec IN (SELECT CustomerID, Age FROM CUSTOMERS) LOOP

IF rec.Age > 60 THEN

UPDATE LOANS

SET InterestRate = InterestRate - 1

WHERE CustomerID = rec.CustomerID;

END IF;

END LOOP;

COMMIT;

END;

/

**Scenario 2:**

BEGIN

FOR rec IN (SELECT CustomerID, Balance FROM CUSTOMERS) LOOP

IF rec.Balance > 10000 THEN

UPDATE CUSTOMERS

SET IsVIP = 'Y'

WHERE CustomerID = rec.CustomerID;

END IF;

END LOOP;

COMMIT;

END;

/

**Scenario 3:**

BEGIN

FOR rec IN (

SELECT c.Name, l.DueDate

FROM CUSTOMERS c

JOIN LOANS l ON c.CustomerID = l.CustomerID

WHERE l.DueDate BETWEEN SYSDATE AND SYSDATE + 30

) LOOP

INSERT INTO TEMP\_MESSAGES (msg)

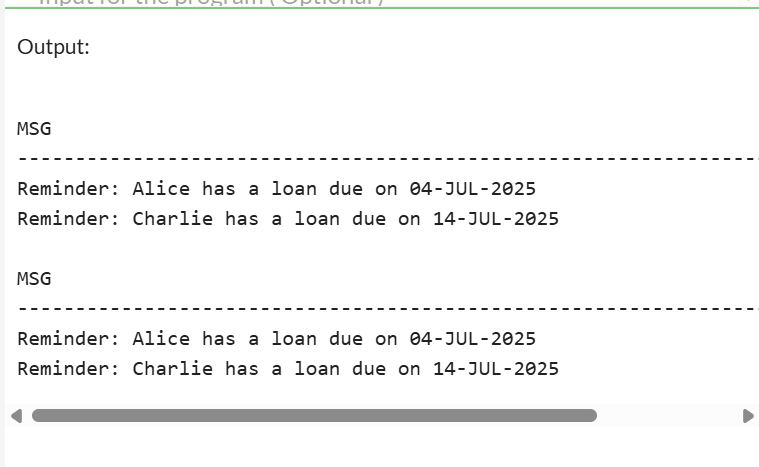
VALUES ('Reminder: ' || rec.Name || ' has a loan due on ' || TO\_CHAR(rec.DueDate, 'DD-MON-YYYY'));

END LOOP;

COMMIT;

END;

/



**Exercise 3:**

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE ACCOUNTS';

EXCEPTION

WHEN OTHERS THEN

IF SQLCODE != -942 THEN -- ORA-00942: table does not exist

RAISE;

END IF;

END;

/

CREATE TABLE ACCOUNTS (

AccountID NUMBER PRIMARY KEY,

CustomerID NUMBER,

AccountType VARCHAR2(20),

Balance NUMBER

);

INSERT INTO ACCOUNTS VALUES (101, 1, 'SAVINGS', 1000);

INSERT INTO ACCOUNTS VALUES (102, 1, 'CHECKING', 500);

INSERT INTO ACCOUNTS VALUES (103, 2, 'SAVINGS', 3000);

INSERT INTO ACCOUNTS VALUES (104, 3, 'SAVINGS', 700);

COMMIT;

**Scenario 1:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE ACCOUNTS

SET Balance = Balance + (Balance \* 0.01)

WHERE UPPER(AccountType) = 'SAVINGS';

DBMS\_OUTPUT.PUT\_LINE(SQL%ROWCOUNT || ' savings accounts updated with interest.');

COMMIT;

END;

/

BEGIN

DBMS\_OUTPUT.ENABLE;

END;

/

BEGIN

ProcessMonthlyInterest;

END;

**Scenario 2**:

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_DepartmentID IN NUMBER,

p\_BonusPercent IN NUMBER

) IS

BEGIN

UPDATE EMPLOYEES

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

WHERE DepartmentID = p\_DepartmentID;

DBMS\_OUTPUT.PUT\_LINE(SQL%ROWCOUNT || ' employee(s) updated with a ' || p\_BonusPercent || '% bonus.');

COMMIT;

END;

/

**Scenario 3**

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_FromAccountID IN NUMBER,

p\_ToAccountID IN NUMBER,

p\_Amount IN NUMBER

) IS

v\_FromBalance NUMBER;

BEGIN

SELECT Balance INTO v\_FromBalance

FROM ACCOUNTS

WHERE AccountID = p\_FromAccountID

FOR UPDATE;

IF v\_FromBalance < p\_Amount THEN

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

END IF;

UPDATE ACCOUNTS

SET Balance = Balance - p\_Amount

WHERE AccountID = p\_FromAccountID

UPDATE ACCOUNTS

SET Balance = Balance + p\_Amount

WHERE AccountID = p\_ToAccountID;

DBMS\_OUTPUT.PUT\_LINE('Transferred ' || p\_Amount || ' from account ' || p\_FromAccountID || ' to account ' || p\_ToAccountID);

COMMIT;

END;

/

