**WEEK – 2 : PL/SQL PROGRAMMING**

**EXERCISE 1: CONTROL STRUCTURES**

**STEP 1: CREATE REQUIRED TABLES IN ORACLE LIVE SQL**

CREATE TABLE Customers (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

Age NUMBER,

Balance NUMBER,

LastModified DATE

);

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

CustomerID NUMBER,

AccountType VARCHAR2(20),

Balance NUMBER,

LastModified DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

CREATE TABLE Loans (

LoanID NUMBER PRIMARY KEY,

CustomerID NUMBER,

LoanAmount NUMBER,

InterestRate NUMBER,

LoanType VARCHAR2(20)

);

**STEP 2: INSERT SAMPLE DATA**

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

VALUES (1, 'Ravi Kumar', 68, 100000, SYSDATE);

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

VALUES (2, 'Anjali Sharma', 42, 120000, SYSDATE);

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

VALUES (3, 'Sunil Mehta', 75, 90000, SYSDATE);

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

VALUES (4, 'Priya Nair', 33, 85000, SYSDATE);

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

VALUES (5, 'Mohammed Irfan', 66, 110000, SYSDATE);

COMMIT;

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

VALUES (101, 1, 'Savings', 8000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

VALUES (102, 1, 'Checking', 3000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

VALUES (103, 2, 'Savings', 5000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

VALUES (104, 3, 'Savings', 12000, SYSDATE);

-- Insert home loan

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, LoanType)

VALUES (1, 101, 500000, 8.5, 'Home');

-- Insert personal loan

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, LoanType)

VALUES (2, 102, 200000, 10.0, 'Personal');

-- Insert business loan

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, LoanType)

VALUES (3, 103, 1000000, 12.0, 'Business');

-- Insert home loan

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, LoanType)

VALUES (4, 104, 750000, 9.0, 'Home');

-- Insert personal loan

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, LoanType)

VALUES (5, 105, 150000, 11.0, 'Personal');

COMMIT;

**SCENARIO 1: DISCOUNT FOR CUSTOMERS OVER 60**

BEGIN

FOR rec IN (SELECT CustomerID, Age, Balance FROM Customers) LOOP

IF rec.Age > 60 THEN

UPDATE Customers

SET Balance = Balance \* 0.99, -- Apply 1% discount

LastModified = SYSDATE

WHERE CustomerID = rec.CustomerID;

END IF;

END LOOP;

COMMIT;

END;

**SCENARIO 2: PROMOTE CUSTOMERS TO VIP BASED ON BALANCE**

BEGIN

FOR cust\_rec IN (

SELECT c.CustomerID,

NVL(SUM(a.Balance), 0) AS TotalBalance

FROM Customers c

LEFT JOIN Accounts a ON c.CustomerID = a.CustomerID

GROUP BY c.CustomerID

) LOOP

IF cust\_rec.TotalBalance > 10000 THEN

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = cust\_rec.CustomerID;

ELSE

UPDATE Customers

SET IsVIP = 'FALSE'

WHERE CustomerID = cust\_rec.CustomerID;

END IF;

END LOOP;

COMMIT;

END;

**SCENARIO 3: LOAN DUE DATE REMINDER (NEXT 30 DAYS)**

DECLARE

CURSOR UpdateLoanInterestRates IS

SELECT LoanID, InterestRate, LoanType

FROM Loans;

BEGIN

FOR loan\_rec IN UpdateLoanInterestRates LOOP

IF loan\_rec.LoanType = 'Home' THEN

UPDATE Loans

SET InterestRate = loan\_rec.InterestRate - 1

WHERE LoanID = loan\_rec.LoanID;

ELSIF loan\_rec.LoanType = 'Personal' THEN

UPDATE Loans

SET InterestRate = loan\_rec.InterestRate + 0.5

WHERE LoanID = loan\_rec.LoanID;

END IF;

END LOOP;

COMMIT;

END;

**OUTPUT**

**QUERY RESULT**





