**PLSQL\_Exercises**

**Control Structures**

SET SERVEROUTPUT ON;

-- Create Customers Table

CREATE TABLE Customers (

  CustomerID NUMBER PRIMARY KEY,

  Name VARCHAR2(100),

  Age NUMBER,

  Balance NUMBER(10, 2),

  LoanInterestRate NUMBER(5, 2),

  IsVIP VARCHAR2(5)

);

-- Create Loans Table

CREATE TABLE Loans (

  LoanID NUMBER PRIMARY KEY,

  CustomerID NUMBER REFERENCES Customers(CustomerID),

  DueDate DATE

);

INSERT INTO Customers VALUES (1, 'Alice', 65, 12000, 9.5, 'FALSE');

INSERT INTO Customers VALUES (2, 'Bob', 45, 8000, 10.0, 'FALSE');

INSERT INTO Customers VALUES (3, 'Charlie', 70, 11000, 8.5, 'FALSE');

INSERT INTO Loans VALUES (101, 1, SYSDATE + 15);

INSERT INTO Loans VALUES (102, 2, SYSDATE + 40);

INSERT INTO Loans VALUES (103, 3, SYSDATE + 10);

COMMIT;

-- 1. Discount for Customers > 60

DECLARE

  CURSOR cust\_cursor IS

    SELECT CustomerID, Age, LoanInterestRate

    FROM Customers

    WHERE Age > 60;

BEGIN

  FOR cust\_rec IN cust\_cursor LOOP

    UPDATE Customers

    SET LoanInterestRate = LoanInterestRate - 1

    WHERE CustomerID = cust\_rec.CustomerID;

  END LOOP;

  COMMIT;

  DBMS\_OUTPUT.PUT\_LINE('Discount applied to customers above 60.');

END;

/

--2. Promote to VIP for Balance > 10000

DECLARE

  CURSOR cust\_cursor IS

    SELECT CustomerID, Balance

    FROM Customers

    WHERE Balance > 10000;

BEGIN

  FOR cust\_rec IN cust\_cursor LOOP

    UPDATE Customers

    SET IsVIP = 'TRUE'

    WHERE CustomerID = cust\_rec.CustomerID;

  END LOOP;

  COMMIT;

  DBMS\_OUTPUT.PUT\_LINE('VIP status updated for eligible customers.');

END;

/

--3. Send Loan Due Reminders

DECLARE

  CURSOR loan\_cursor IS

    SELECT CustomerID, LoanID, DueDate

    FROM Loans

    WHERE DueDate BETWEEN SYSDATE AND SYSDATE + 30;

  customer\_name VARCHAR2(100);

BEGIN

  FOR loan\_rec IN loan\_cursor LOOP

    SELECT Name INTO customer\_name

    FROM Customers

    WHERE CustomerID = loan\_rec.CustomerID;

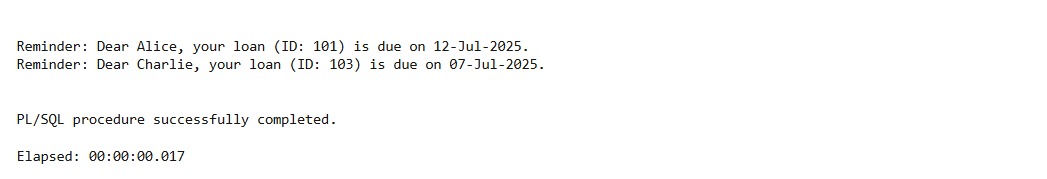
    DBMS\_OUTPUT.PUT\_LINE('Reminder: Dear ' || customer\_name || ', your loan (ID: ' || loan\_rec.LoanID || ') is due on ' || TO\_CHAR(loan\_rec.DueDate, 'DD-Mon-YYYY') || '.');

  END LOOP;

END;

/

**OUTPUT:**



**Stored Procedures**

BEGIN

  EXECUTE IMMEDIATE 'DROP TABLE employees CASCADE CONSTRAINTS';

EXCEPTION

  WHEN OTHERS THEN NULL;

END;

/

BEGIN

  EXECUTE IMMEDIATE 'CREATE TABLE employees (

    emp\_id NUMBER PRIMARY KEY,

    emp\_name VARCHAR2(100),

    department\_id NUMBER,

    salary NUMBER

  )';

EXCEPTION

  WHEN OTHERS THEN NULL;

END;

/

INSERT INTO employees VALUES (1, 'Anu', 2, 30000);

INSERT INTO employees VALUES (2, 'Ravi', 2, 35000);

INSERT INTO employees VALUES (3, 'Meena', 1, 28000);

COMMIT;

/

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

  p\_department\_id IN NUMBER,

  p\_bonus\_percent IN NUMBER

)

IS

BEGIN

  UPDATE employees

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

  WHERE department\_id = p\_department\_id;

  COMMIT;

  DBMS\_OUTPUT.PUT\_LINE('Employee bonuses updated successfully.');

EXCEPTION

  WHEN OTHERS THEN

    ROLLBACK;

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

END;

/

BEGIN

  UpdateEmployeeBonus(2, 10);

END;

/

BEGIN

  ProcessMonthlyInterest;

END;

/

BEGIN

  TransferFunds(101, 102, 500);

END;

/

**OUTPUT:**

