**Exercise 1: Control Structures**

**Create table:**

SET SERVEROUTPUT ON;

BEGIN EXECUTE IMMEDIATE 'DROP TABLE Loans'; EXCEPTION WHEN OTHERS THEN NULL; END;

/

BEGIN EXECUTE IMMEDIATE 'DROP TABLE Customers'; EXCEPTION WHEN OTHERS THEN NULL; END;

/

CREATE TABLE Customers (

    CustomerID NUMBER PRIMARY KEY,

    Name VARCHAR2(100),

    Age NUMBER,

    Balance NUMBER,

    IsVIP VARCHAR2(5)

);

CREATE TABLE Loans (

    LoanID NUMBER PRIMARY KEY,

    CustomerID NUMBER,

    InterestRate NUMBER,

    DueDate DATE,

    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

INSERT INTO Customers VALUES (101, 'Krishna', 65, 15000, 'FALSE');

INSERT INTO Customers VALUES (102, 'Meera', 59, 8000, 'FALSE');

INSERT INTO Customers VALUES (103, 'Arun', 70, 12000, 'FALSE');

INSERT INTO Loans VALUES (1, 101, 10.5, SYSDATE + 15);

INSERT INTO Loans VALUES (2, 102, 12.0, SYSDATE + 40);

INSERT INTO Loans VALUES (3, 103, 11.0, SYSDATE + 10);

COMMIT;

**Scenario 1:**

DECLARE

    CURSOR cur\_customers IS

        SELECT CustomerID, InterestRate

        FROM Loans

        WHERE CustomerID IN (

            SELECT CustomerID FROM Customers WHERE Age > 60

        );

BEGIN

    FOR loan\_rec IN cur\_customers LOOP

        UPDATE Loans

        SET InterestRate = InterestRate - 1

        WHERE CustomerID = loan\_rec.CustomerID;

        DBMS\_OUTPUT.PUT\_LINE('Discount applied for Customer ID: ' || loan\_rec.CustomerID);

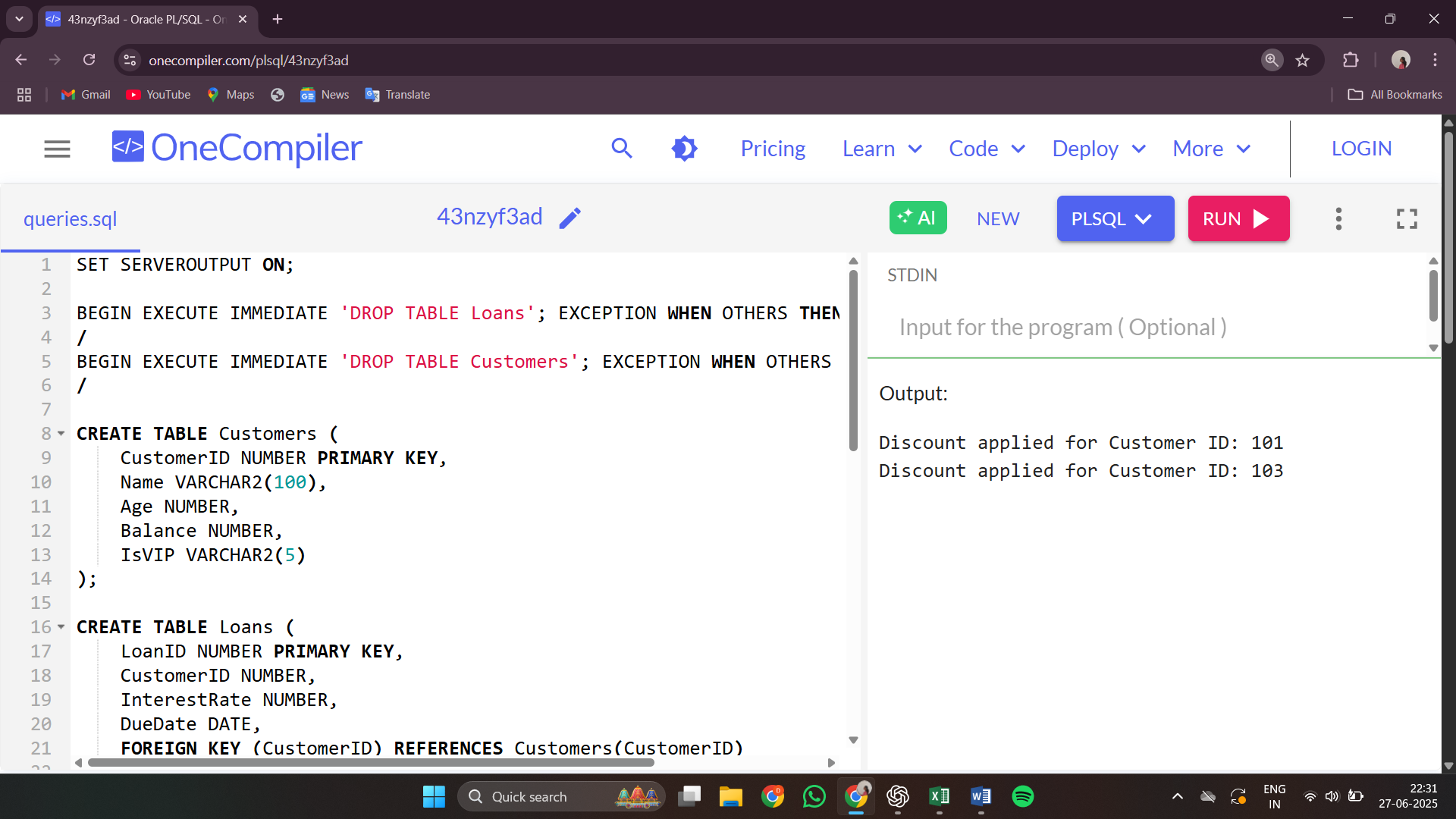
    END LOOP;

    COMMIT;

END;

/

**Output:**



**Scenario 2:**

DECLARE

    CURSOR cur\_vip\_customers IS

        SELECT CustomerID, Balance

        FROM Customers

        WHERE Balance > 10000;

BEGIN

    FOR cust\_rec IN cur\_vip\_customers LOOP

        UPDATE Customers

        SET IsVIP = 'TRUE'

        WHERE CustomerID = cust\_rec.CustomerID;

        DBMS\_OUTPUT.PUT\_LINE('Promoted to VIP: Customer ID ' || cust\_rec.CustomerID);

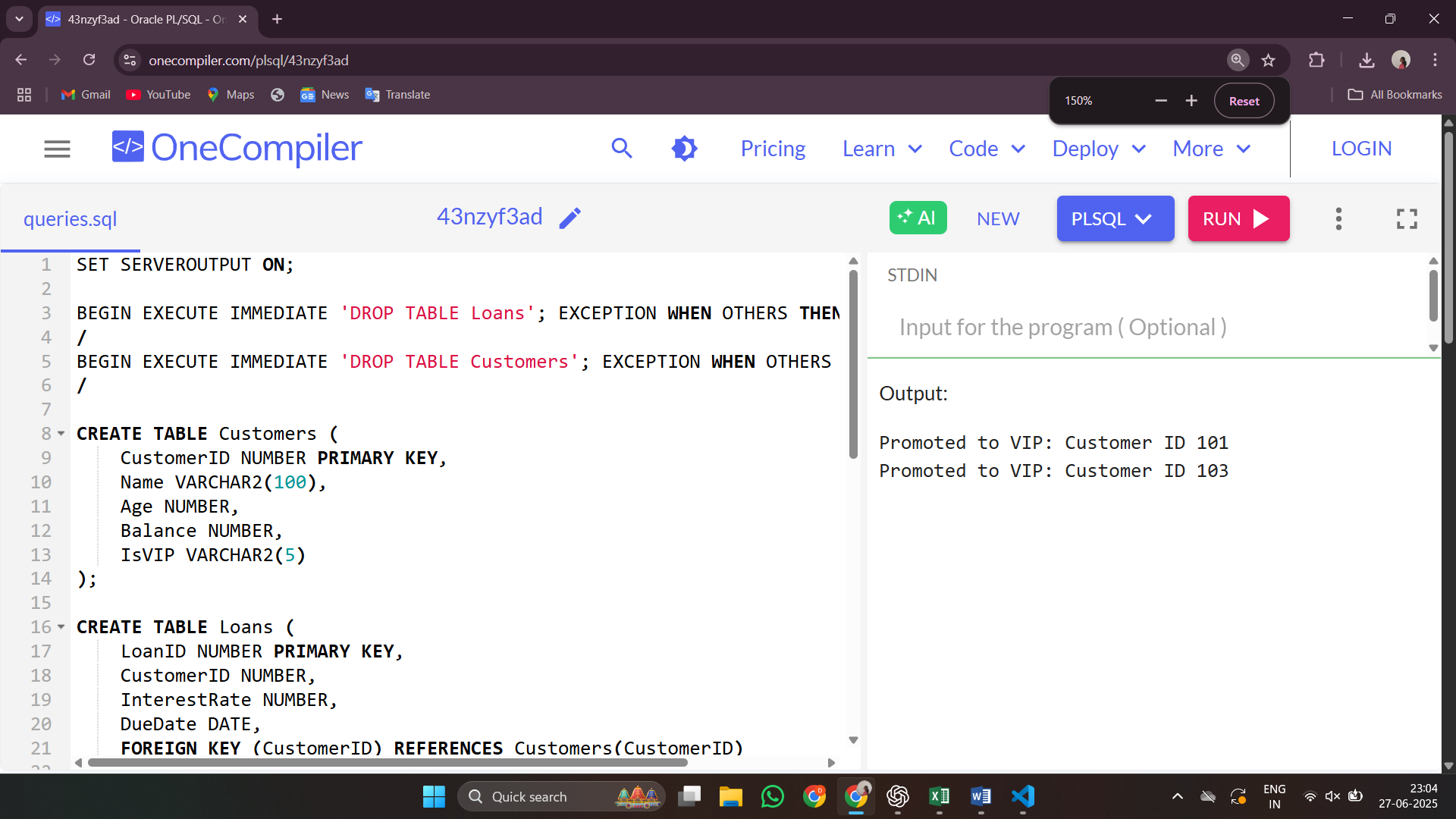
    END LOOP;

    COMMIT;

END;

/

**Output**:



**Scenario 3:**

DECLARE

    CURSOR cur\_due\_loans IS

        SELECT l.CustomerID, c.Name, l.DueDate

        FROM Loans l

        JOIN Customers c ON l.CustomerID = c.CustomerID

        WHERE l.DueDate BETWEEN SYSDATE AND SYSDATE + 30;

BEGIN

    FOR loan\_rec IN cur\_due\_loans LOOP

        DBMS\_OUTPUT.PUT\_LINE(

            'Reminder: Loan due for ' || loan\_rec.Name ||

            ' (Customer ID: ' || loan\_rec.CustomerID ||

            ') on ' || TO\_CHAR(loan\_rec.DueDate, 'DD-MON-YYYY')

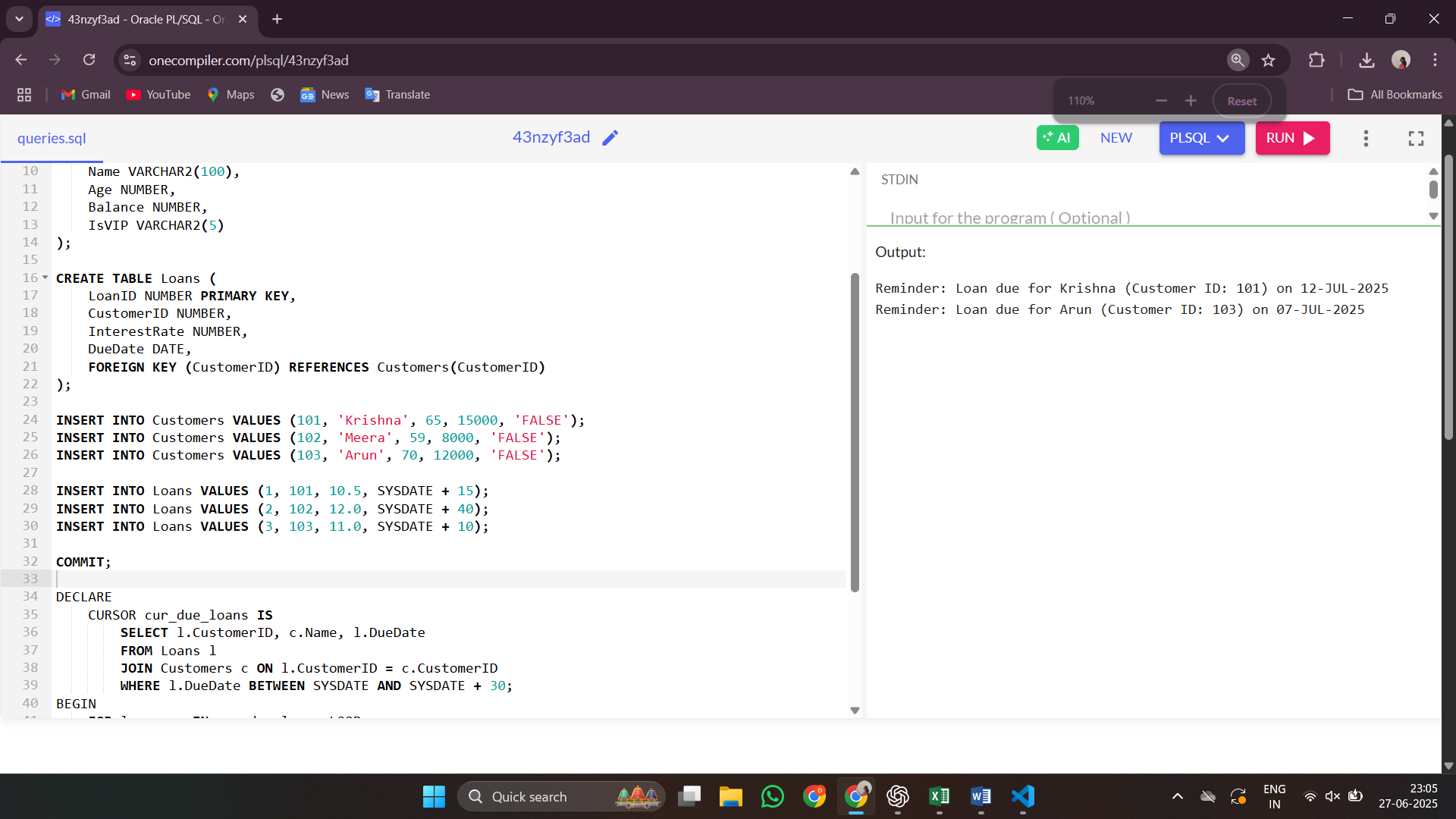
        );

    END LOOP;

END;

/

**Output:**



**Exercise 3: Stored Procedures**

**Create table:**

SET SERVEROUTPUT ON;

BEGIN EXECUTE IMMEDIATE 'DROP TABLE Accounts'; EXCEPTION WHEN OTHERS THEN NULL; END;

/

BEGIN EXECUTE IMMEDIATE 'DROP TABLE Employees'; EXCEPTION WHEN OTHERS THEN NULL; END;

/

CREATE TABLE Accounts (

    AccountID NUMBER PRIMARY KEY,

    CustomerName VARCHAR2(100),

    AccountType VARCHAR2(20),

    Balance NUMBER

);

CREATE TABLE Employees (

    EmployeeID NUMBER PRIMARY KEY,

    Name VARCHAR2(100),

    Department VARCHAR2(50),

    Salary NUMBER

);

INSERT INTO Accounts VALUES (1, 'Krishna', 'Savings', 10000);

INSERT INTO Accounts VALUES (2, 'Meera', 'Savings', 8000);

INSERT INTO Accounts VALUES (3, 'Arun', 'Current', 12000);

INSERT INTO Accounts VALUES (4, 'Kavya', 'Savings', 15000);

INSERT INTO Employees VALUES (101, 'Ravi', 'Sales', 50000);

INSERT INTO Employees VALUES (102, 'Divya', 'IT', 60000);

INSERT INTO Employees VALUES (103, 'Manoj', 'Sales', 55000);

INSERT INTO Employees VALUES (104, 'Sneha', 'HR', 48000);

COMMIT;

**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 + (Balance \* 0.01)

        WHERE AccountID = acc.AccountID;

        DBMS\_OUTPUT.PUT\_LINE('Interest added for Account ID: ' || acc.AccountID);

    END LOOP;

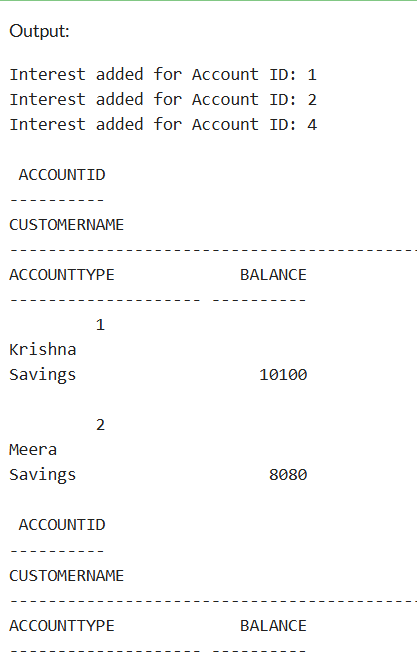
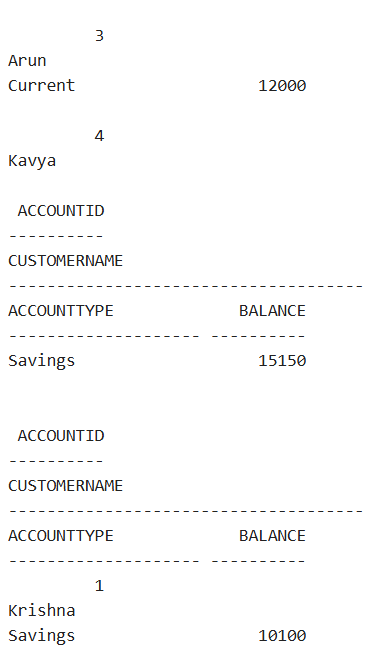
    COMMIT;

END;

/

EXEC ProcessMonthlyInterest;

SELECT \* FROM Accounts;

**Scenario 2:**

SET SERVEROUTPUT ON;

BEGIN EXECUTE IMMEDIATE 'DROP TABLE Accounts'; EXCEPTION WHEN OTHERS THEN NULL; END;

/

BEGIN EXECUTE IMMEDIATE 'DROP TABLE Employees'; EXCEPTION WHEN OTHERS THEN NULL; END;

/

CREATE TABLE Accounts (

    AccountID NUMBER PRIMARY KEY,

    CustomerName VARCHAR2(100),

    AccountType VARCHAR2(20),

    Balance NUMBER

);

CREATE TABLE Employees (

    EmployeeID NUMBER PRIMARY KEY,

    Name VARCHAR2(100),

    Department VARCHAR2(50),

    Salary NUMBER

);

INSERT INTO Accounts VALUES (1, 'Krishna', 'Savings', 10000);

INSERT INTO Accounts VALUES (2, 'Meera', 'Savings', 8000);

INSERT INTO Accounts VALUES (3, 'Arun', 'Current', 12000);

INSERT INTO Accounts VALUES (4, 'Kavya', 'Savings', 15000);

INSERT INTO Employees VALUES (101, 'Ravi', 'Sales', 50000);

INSERT INTO Employees VALUES (102, 'Divya', 'IT', 60000);

INSERT INTO Employees VALUES (103, 'Manoj', 'Sales', 55000);

INSERT INTO Employees VALUES (104, 'Sneha', 'HR', 48000);

COMMIT;

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

    FOR acc IN (SELECT AccountID, Balance FROM Accounts WHERE AccountType = 'Savings') LOOP

        UPDATE Accounts

        SET Balance = Balance + (Balance \* 0.01)

        WHERE AccountID = acc.AccountID;

        DBMS\_OUTPUT.PUT\_LINE('Interest added for Account ID: ' || acc.AccountID);

    END LOOP;

    COMMIT;

END;

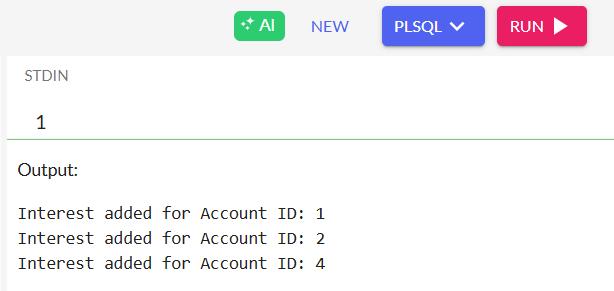
/

BEGIN

    ProcessMonthlyInterest;

END;

/

**Output:**  
  


**Scenario 3:**

CREATE OR REPLACE PROCEDURE TransferFunds(

    fromAccID IN NUMBER,

    toAccID IN NUMBER,

    amount IN NUMBER

) IS

    fromBalance NUMBER;

BEGIN

    SELECT Balance INTO fromBalance

    FROM Accounts

    WHERE AccountID = fromAccID

    FOR UPDATE;

    IF fromBalance < amount THEN

        DBMS\_OUTPUT.PUT\_LINE('Insufficient balance in Account ID: ' || fromAccID);

    ELSE

        UPDATE Accounts SET Balance = Balance - amount WHERE AccountID = fromAccID;

        UPDATE Accounts SET Balance = Balance + amount WHERE AccountID = toAccID;

        DBMS\_OUTPUT.PUT\_LINE('Transferred rupees ' || amount ||

                             ' from Account ID ' || fromAccID ||

                             ' to Account ID ' || toAccID);

    END IF;

    COMMIT;

END;

/

BEGIN

    TransferFunds(1, 2, 2000);

END;

/

SELECT \* FROM Accounts;

