**Exercise 1: Control Structures**

-- Scenario 1: Apply 1% discount to loan interest rates for customers above 60 years old

BEGIN

    FOR cust\_rec IN (SELECT CustomerID, Age FROM Customers) LOOP

        IF cust\_rec.Age > 60 THEN

            UPDATE Loans

            SET InterestRate = InterestRate - 1

            WHERE CustomerID = cust\_rec.CustomerID;

        END IF;

    END LOOP;

    COMMIT;

END;

/

-- Scenario 2: Set IsVIP flag to TRUE for customers with balance over $10,000

BEGIN

    FOR cust\_rec IN (SELECT CustomerID, Balance FROM Customers) LOOP

        IF cust\_rec.Balance > 10000 THEN

            UPDATE Customers

            SET IsVIP = 'TRUE'

            WHERE CustomerID = cust\_rec.CustomerID;

        END IF;

    END LOOP;

    COMMIT;

END;

/

-- Scenario 3: Send reminders for loans due within the next 30 days

BEGIN

    FOR loan\_rec IN (

        SELECT CustomerID, DueDate FROM Loans

        WHERE DueDate BETWEEN SYSDATE AND SYSDATE + 30

    ) LOOP

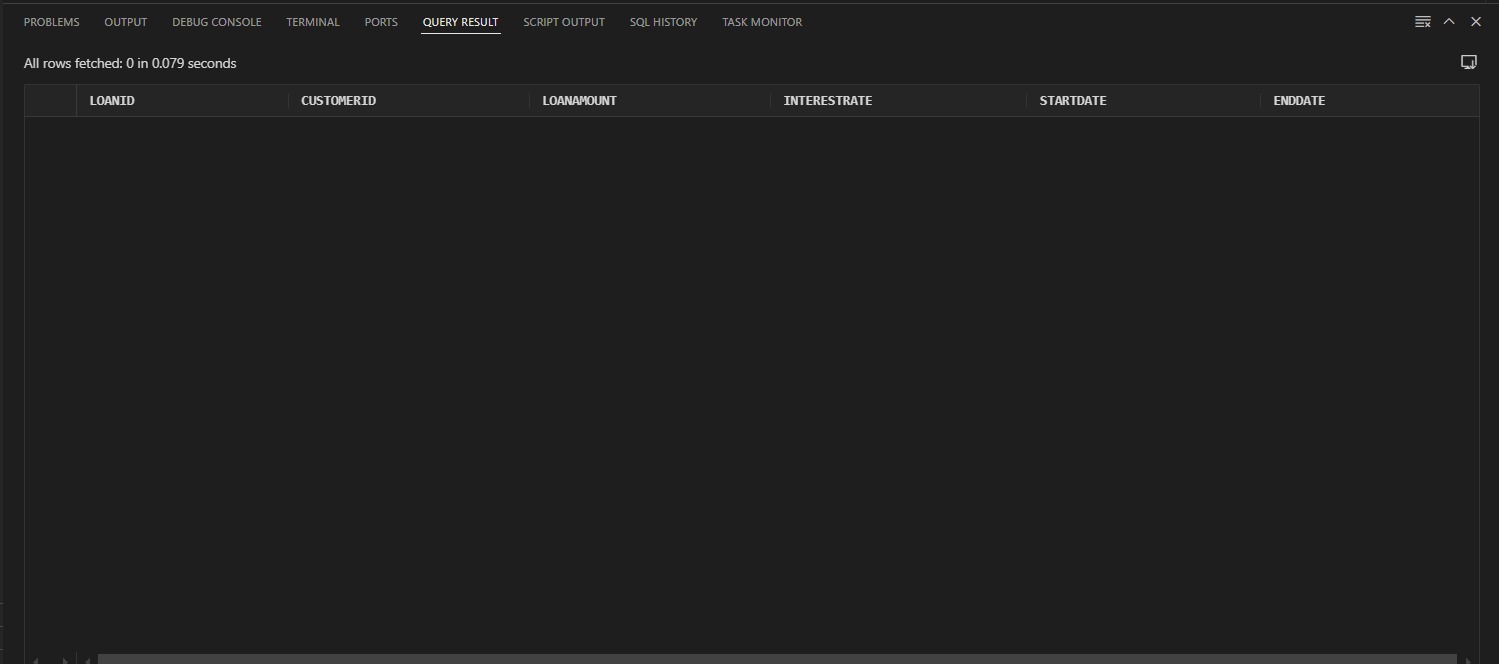
        DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan due for Customer ID ' || loan\_rec.CustomerID || ' on ' || TO\_CHAR(loan\_rec.DueDate, 'DD-MON-YYYY'));

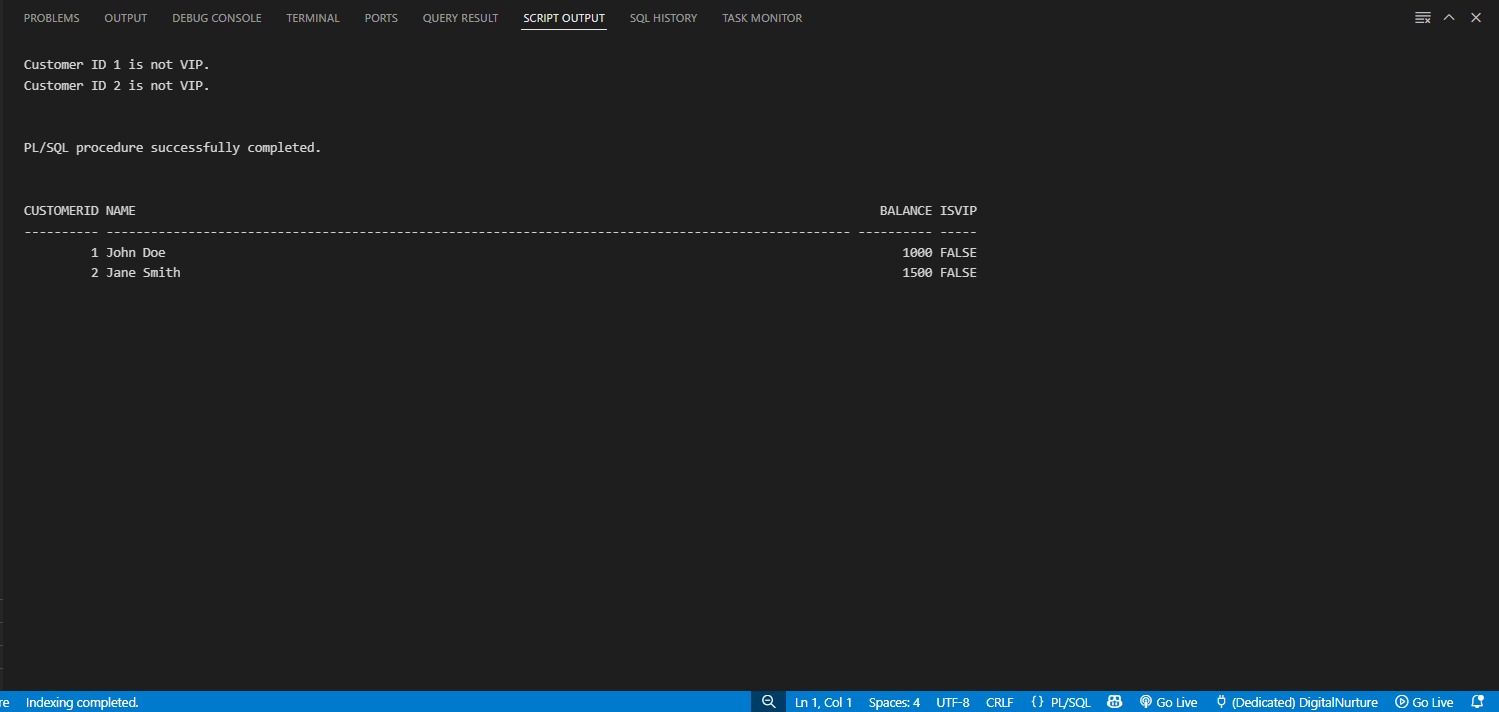
    END LOOP;

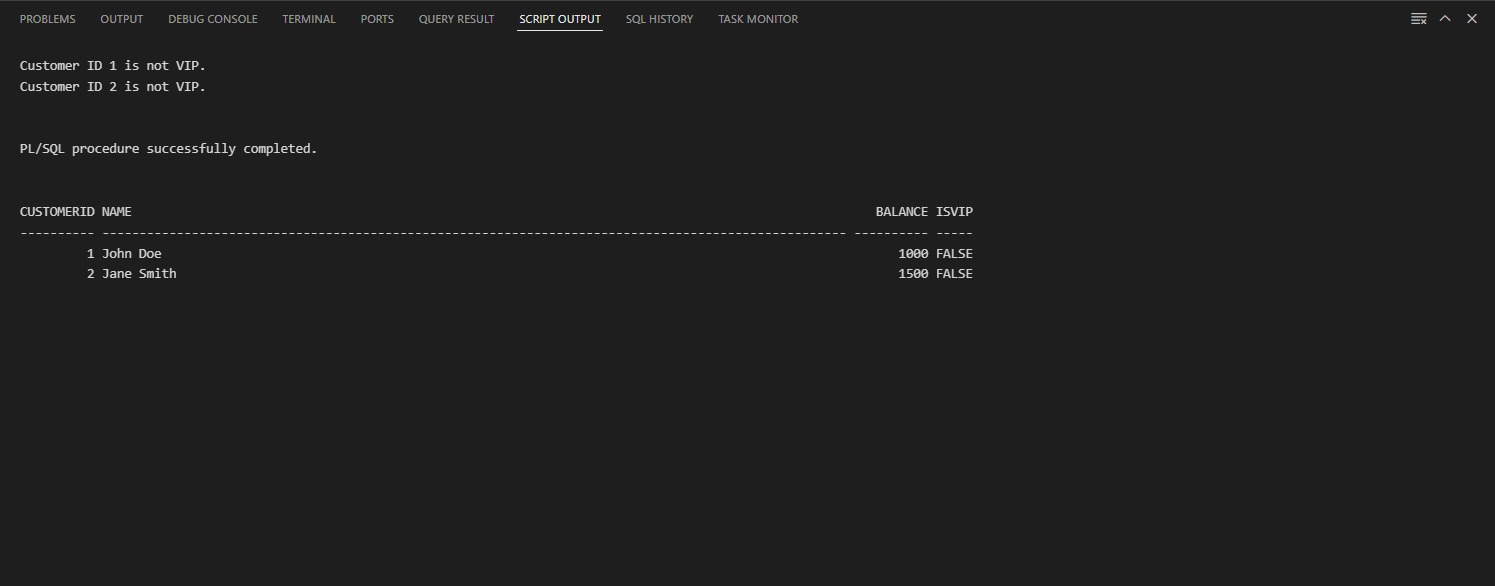
END;

/

**OUTPUT**







**Exercise 3: Stored Procedures**

-- Scenario 1: ProcessMonthlyInterest stored procedure

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

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

        UPDATE Accounts

        SET Balance = Balance + (Balance \* 0.01)

        WHERE AccountID = acc\_rec.AccountID;

    END LOOP;

    COMMIT;

END;

/

-- Scenario 2: UpdateEmployeeBonus stored procedure

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

    dept\_id IN NUMBER,

    bonus\_percent IN NUMBER

) IS

BEGIN

    UPDATE Employees

    SET Salary = Salary + (Salary \* bonus\_percent / 100)

    WHERE DepartmentID = dept\_id;

    COMMIT;

END;

/

-- Scenario 3: TransferFunds stored procedure

CREATE OR REPLACE PROCEDURE TransferFunds(

    from\_account\_id IN NUMBER,

    to\_account\_id IN NUMBER,

    amount IN NUMBER

) IS

    from\_balance NUMBER;

BEGIN

    SELECT Balance INTO from\_balance FROM Accounts WHERE AccountID = from\_account\_id FOR UPDATE;

    IF from\_balance >= amount THEN

        UPDATE Accounts

        SET Balance = Balance - amount

        WHERE AccountID = from\_account\_id;

        UPDATE Accounts

        SET Balance = Balance + amount

        WHERE AccountID = to\_account\_id;

        COMMIT;

    ELSE

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

    END IF;

END;

/

**OUTPUT**

