**Exercise 1: Control Structures**

**Scenario 1:** The bank wants to apply a discount to loan interest rates for customers above 60 years old.

***Note: I have done this in*** [***https://livesql.oracle.com/next/***](https://livesql.oracle.com/next/) ***which is an Online Compiler for PL/SQL by Oracle.***

**Question*:*** Write a PL/SQL block that loops through all customers, checks their age, and if they are above 60, apply a 1% discount to their current loan interest rates.

**CODE:**

*--Customers Table Creation*

CREATE TABLE customers (

    customer\_id NUMBER PRIMARY KEY,

    name VARCHAR2(100),

    age NUMBER,

    interest\_rate NUMBER,

    balance NUMBER,

    isVIP VARCHAR2(5)

);

*-- Sample customers*

INSERT INTO customers VALUES (1, 'John Doe', 65, 10.0, 15000, 'FALSE');

INSERT INTO customers VALUES (2, 'Jane Smith', 45, 11.0, 8000, 'FALSE');

INSERT INTO customers VALUES (3, 'Alice Brown', 70, 12.0, 12000, 'FALSE');

COMMIT;

*--PL/SQL block that loops through all customers, checks their age, and if they are above 60, apply a ----1% discount to their current loan interest rates.*

SET SERVEROUTPUT ON;

DECLARE

CURSOR senior\_customers\_cur IS

SELECT customer\_id, name, age, interest\_rate

FROM customers

WHERE age > 60;

BEGIN

FOR customer IN senior\_customers\_cur LOOP

UPDATE customers

SET interest\_rate = interest\_rate - 1

WHERE customer\_id = customer.customer\_id;

DBMS\_OUTPUT.PUT\_LINE('Customer ' || customer.name ||

' (ID: ' || customer.customer\_id ||

', Age: ' || customer.age ||

') received a 1% discount. New interest rate: ' ||

(customer.interest\_rate - 1));

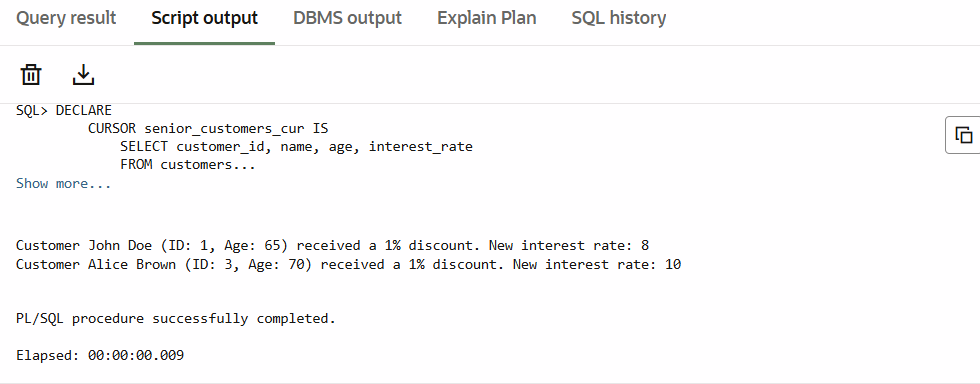
END LOOP;

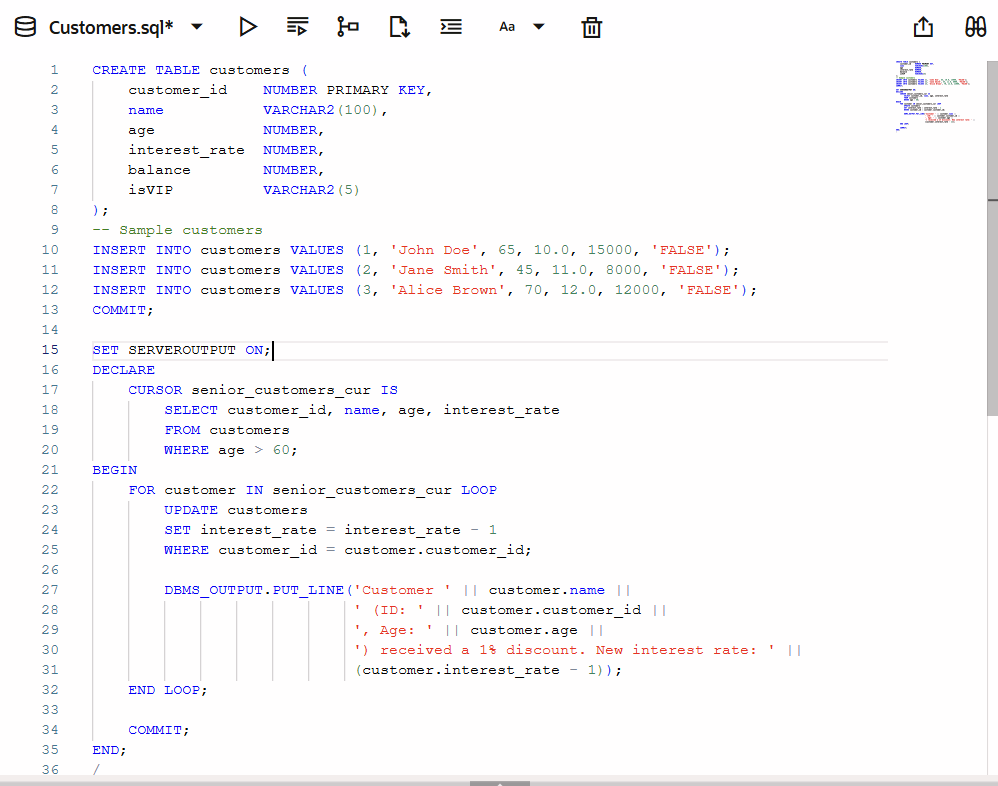
COMMIT;

END;

/

**OUTPUT: (SCREENSHOTS)**

****

****

**Scenario 2:** A customer can be promoted to VIP status based on their balance.

**Question:** Write a PL/SQL block that iterates through all customers and sets a flag IsVIP to TRUE for those with a balance over $10,000.

**CODE:**

SET SERVEROUTPUT ON;

DECLARE

    CURSOR vip\_cursor IS

        SELECT customer\_id, name, balance

        FROM customers;

BEGIN

    FOR customer IN vip\_cursor LOOP

        IF customer.balance > 10000 THEN

            UPDATE customers

            SET isVIP = 'TRUE'

            WHERE customer\_id = customer.customer\_id;

            DBMS\_OUTPUT.PUT\_LINE('Customer ' || customer.name ||

                                 ' (ID: ' || customer.customer\_id ||

                                 ') promoted to VIP. Balance: $' || customer.balance);

        END IF;

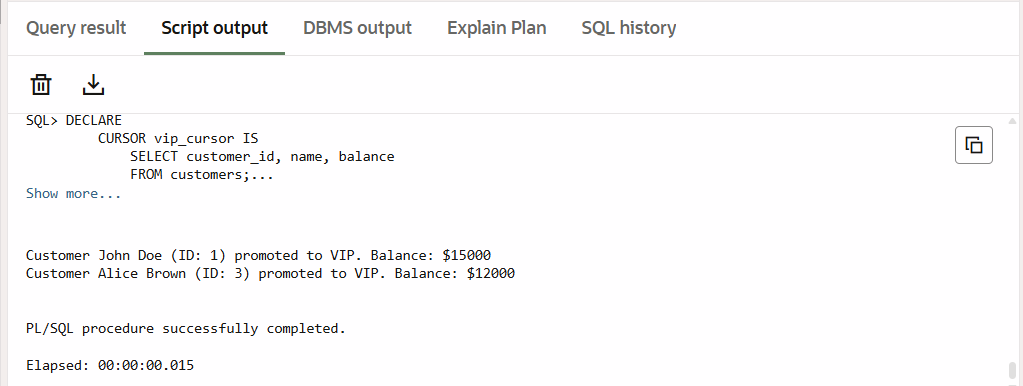
    END LOOP;

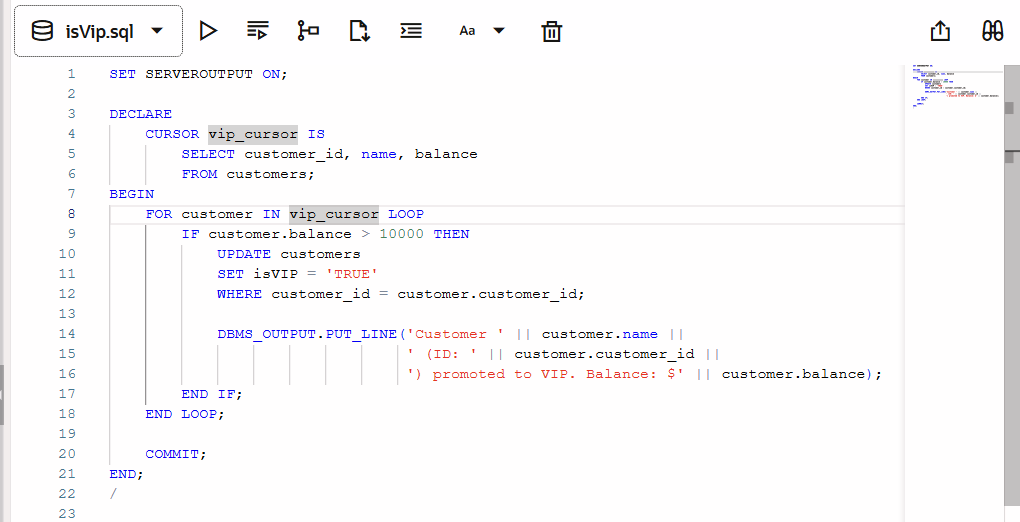
    COMMIT;

END;

/

**OUTPUT: (SCREENSHOTS)**

****

****

**Scenario 3:** The bank wants to send reminders to customers whose loans are due within the next 30 days.

**Question:** Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.

**CODE:**

*-- Loans table:*

CREATE TABLE loans (

    loan\_id      NUMBER PRIMARY KEY,

    customer\_id  NUMBER,

    due\_date     DATE,

    amount       NUMBER,

    CONSTRAINT fk\_customer FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

*-- Insert sample loans*

INSERT INTO loans VALUES (101, 1, SYSDATE + 15, 5000);  -- Due soon

INSERT INTO loans VALUES (102, 2, SYSDATE + 40, 7000);  -- Due later

INSERT INTO loans VALUES (103, 3, SYSDATE + 5, 9000);   -- Due soon

COMMIT;

*--Code to fetch all dues in next 30 days*

SET SERVEROUTPUT ON;

DECLARE

    CURSOR due\_loans\_cur IS

        SELECT l.loan\_id, l.due\_date, c.customer\_id, c.name

        FROM loans l

        JOIN customers c ON l.customer\_id = c.customer\_id

        WHERE l.due\_date BETWEEN SYSDATE AND SYSDATE + 30;

BEGIN

    FOR loan IN due\_loans\_cur LOOP

        DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || loan.loan\_id ||

                             ' for customer ' || loan.name ||

                             ' (ID: ' || loan.customer\_id || ') is due on ' ||

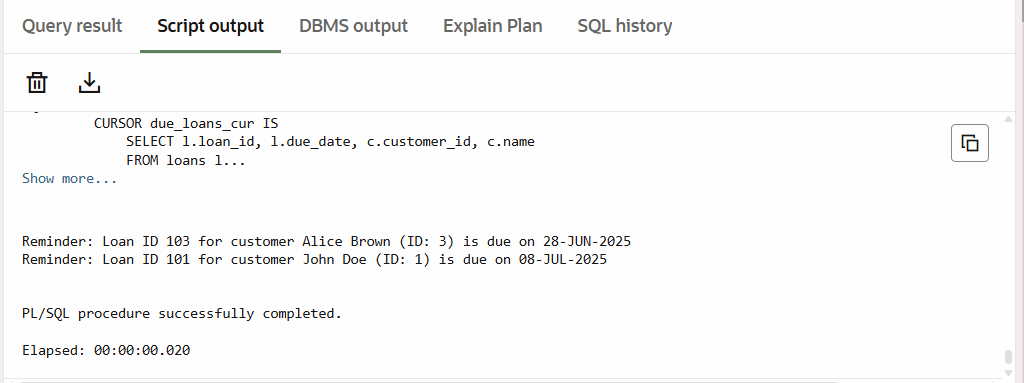
                             TO\_CHAR(loan.due\_date, 'DD-MON-YYYY'));

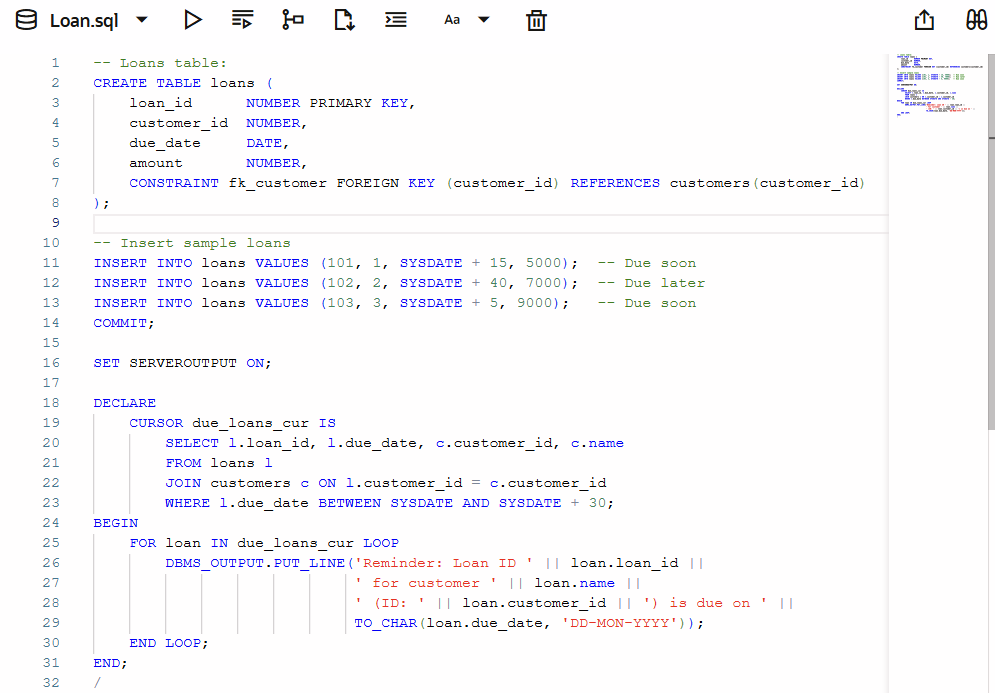
    END LOOP;

END;

/

**OUTPUT: (SCREENSHOTS)**

****

****