**Exercise 1: Control Structures**

**Scenario 1:**

CREATE TABLE customers (

    customer\_id NUMBER PRIMARY KEY,

    name VARCHAR2(50),

    age NUMBER,

    balance NUMBER,

    isVIP VARCHAR2(5),

    loan\_id NUMBER

);

CREATE TABLE loans (

    loan\_id NUMBER PRIMARY KEY,

    customer\_id NUMBER,

    interest\_rate NUMBER,

    due\_date DATE

);

INSERT INTO customers VALUES (1, 'John', 65, 15000, 'FALSE', 101);

INSERT INTO customers VALUES (2, 'Alice', 45, 8000, 'FALSE', 102);

INSERT INTO customers VALUES (3, 'Bob', 70, 12000, 'FALSE', 103);

INSERT INTO customers VALUES (4, 'Clara', 30, 20000, 'FALSE', 104);

INSERT INTO loans VALUES (101, 1, 8.5, SYSDATE + 10);

INSERT INTO loans VALUES (102, 2, 9.0, SYSDATE + 40);

INSERT INTO loans VALUES (103, 3, 7.5, SYSDATE + 5);

INSERT INTO loans VALUES (104, 4, 10.0, SYSDATE + 25);

COMMIT;

BEGIN

    FOR cust IN (

        SELECT customer\_id, age, loan\_id

        FROM customers

        WHERE age > 60

    ) LOOP

        UPDATE loans

        SET interest\_rate = interest\_rate - 1

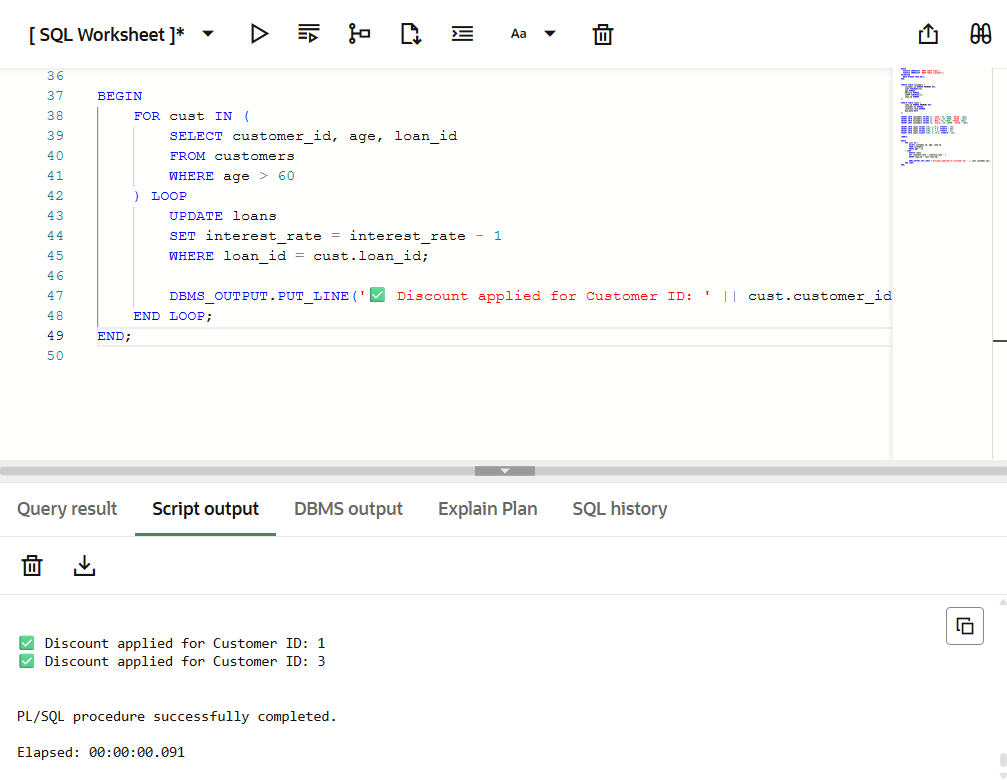
        WHERE loan\_id = cust.loan\_id;

        DBMS\_OUTPUT.PUT\_LINE('Discount applied for Customer ID: ' || cust.customer\_id);

    END LOOP;

END;

**OUTPUT:**



**Scenario 2:**

BEGIN

    FOR cust IN (

        SELECT customer\_id, balance

        FROM customers

        WHERE balance > 10000

    ) LOOP

        UPDATE customers

        SET isVIP = 'TRUE'

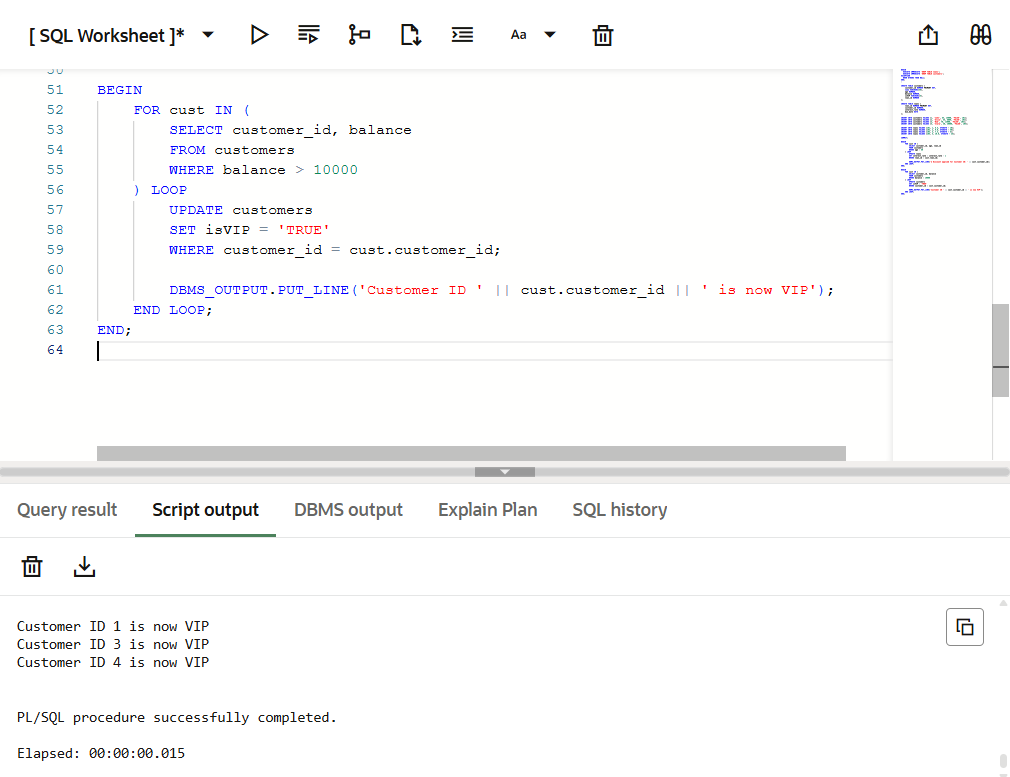
        WHERE customer\_id = cust.customer\_id;

        DBMS\_OUTPUT.PUT\_LINE('Customer ID ' || cust.customer\_id || ' is now VIP');

    END LOOP;

END;

**OUTPUT:**



**Scenario 3:**

BEGIN

    FOR loan\_rec IN (

        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

    ) LOOP

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

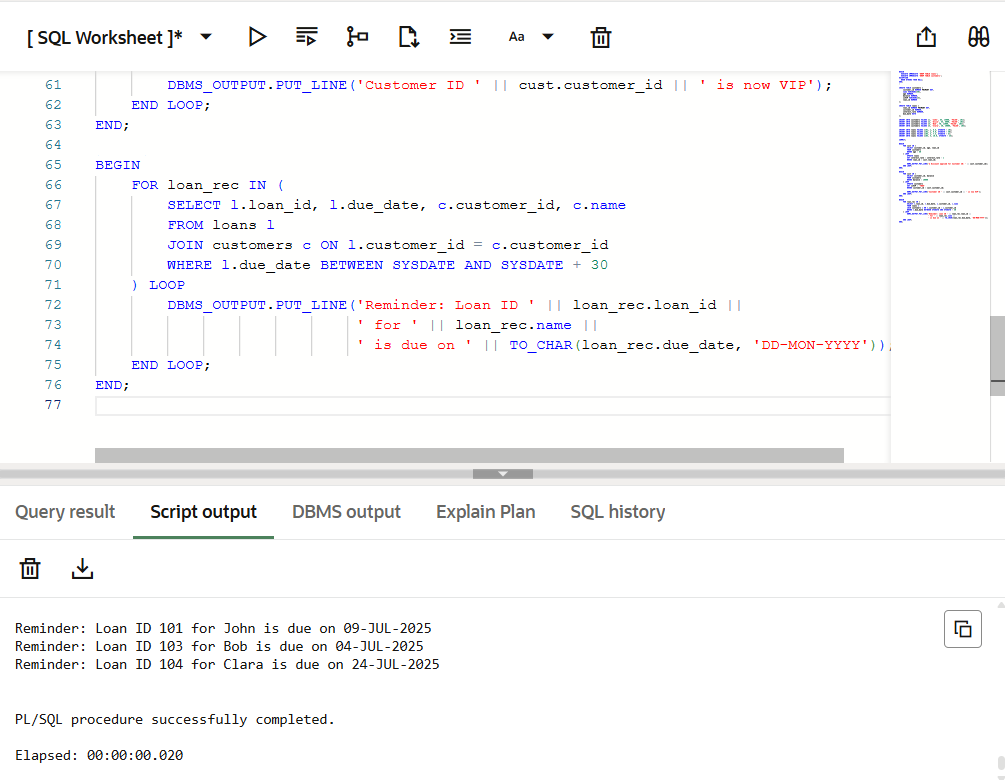
                             ' for ' || loan\_rec.name ||

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

    END LOOP;

END;

**OUTPUT:**



**Exercise 3: Stored Procedures**

**Scenario 1:**

CREATE TABLE accounts (

    acc\_id NUMBER PRIMARY KEY,

    cust\_name VARCHAR2(50),

    acc\_type VARCHAR2(20),

    balance NUMBER

);

CREATE TABLE employees (

    emp\_id NUMBER PRIMARY KEY,

    emp\_name VARCHAR2(50),

    department VARCHAR2(30),

    salary NUMBER

);

INSERT INTO accounts VALUES (1, 'John', 'savings', 10000);

INSERT INTO accounts VALUES (2, 'Alice', 'current', 8000);

INSERT INTO accounts VALUES (3, 'Bob', 'savings', 15000);

INSERT INTO accounts VALUES (4, 'Clara', 'savings', 12000);

INSERT INTO employees VALUES (101, 'Emma', 'HR', 50000);

INSERT INTO employees VALUES (102, 'Liam', 'Sales', 60000);

INSERT INTO employees VALUES (103, 'Olivia', 'Sales', 58000);

COMMIT;

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

    FOR acc IN (

        SELECT acc\_id, balance

        FROM accounts

        WHERE acc\_type = 'savings'

    ) LOOP

        UPDATE accounts

        SET balance = balance + (balance \* 0.01)

        WHERE acc\_id = acc.acc\_id;

        DBMS\_OUTPUT.PUT\_LINE('✅ Interest added to Account ID: ' || acc.acc\_id);

    END LOOP;

END;

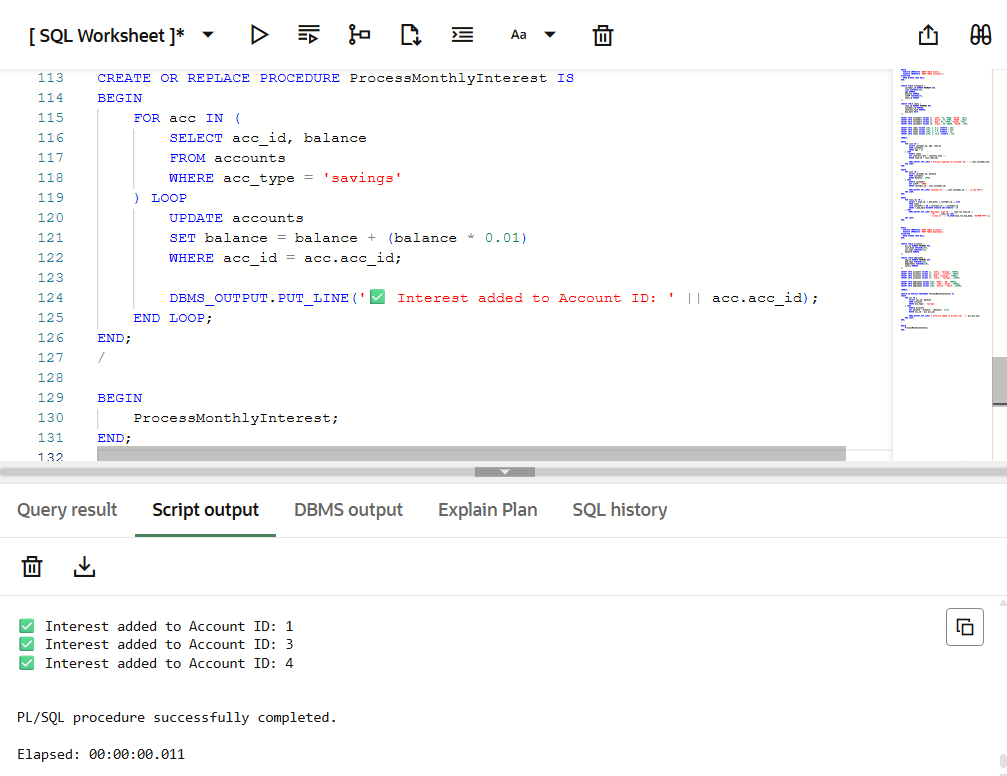
/

BEGIN

    ProcessMonthlyInterest;

END;

**OUTPUT:**



**Scenario 2:**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

    dept\_name IN VARCHAR2,

    bonus\_pct IN NUMBER

) IS

BEGIN

    FOR emp IN (

        SELECT emp\_id, salary FROM employees WHERE department = dept\_name

    ) LOOP

        UPDATE employees

        SET salary = salary + (salary \* bonus\_pct / 100)

        WHERE emp\_id = emp.emp\_id;

        DBMS\_OUTPUT.PUT\_LINE('Bonus applied to Employee ID: ' || emp.emp\_id);

    END LOOP;

END;

/

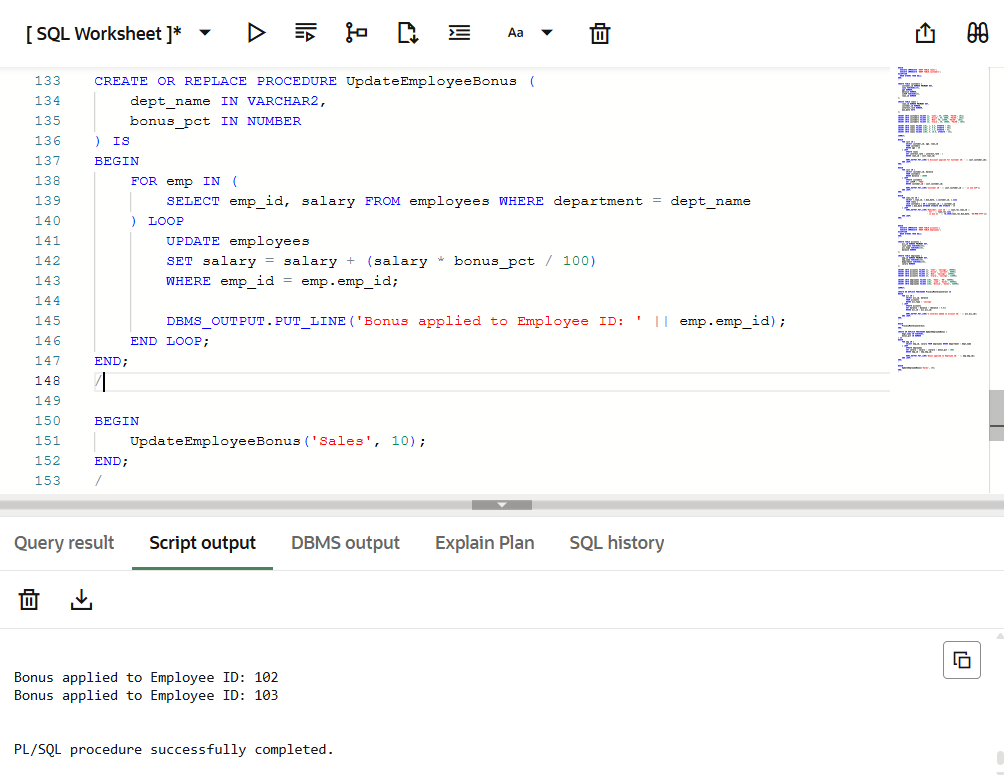
BEGIN

    UpdateEmployeeBonus('Sales', 10);

END;

/

**OUTPUT:**



**Scenario 3:**

CREATE OR REPLACE PROCEDURE TransferFunds (

    from\_acc IN NUMBER,

    to\_acc IN NUMBER,

    amount IN NUMBER

) IS

    from\_balance NUMBER;

BEGIN

    SELECT balance INTO from\_balance FROM accounts WHERE acc\_id = from\_acc;

    IF from\_balance >= amount THEN

        UPDATE accounts

        SET balance = balance - amount

        WHERE acc\_id = from\_acc;

        UPDATE accounts

        SET balance = balance + amount

        WHERE acc\_id = to\_acc;

        DBMS\_OUTPUT.PUT\_LINE(' Transfer of Rs.' || amount ||

                             ' from Account ' || from\_acc ||

                             ' to Account ' || to\_acc || ' completed.');

    ELSE

        DBMS\_OUTPUT.PUT\_LINE('Transfer failed: Insufficient balance in Account ' || from\_acc);

    END IF;

END;

/

BEGIN

    TransferFunds(1, 2, 3000);

END;

/

**OUTPUT:**

