Hands on session

PLSQL Excerises

**Exercise 1: Control Structures**

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

* + 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.

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.

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.

**Objective** Perform conditional operations using loops and control structures based on business rules.

**Steps to implement**

* Step 1: Create Tables
* customers: Holds customer details like name, age, balance, VIP status
* loans: Holds loan details such as amount, interest rate, due date
* Step 2: Insert Sample Data
* Populate both table with realistic cutomer and loan information
* Step 3 : Scenario 1 – Age-Based Interest Discount
* For customers aged above 60, apply a 1% discount on their current loan interest rates.
* Step 4: Scenario 2 – VIP Based on Balance
* Set IsVIP = 1 for customers having balance > $10,000
* Step 5: Scenario 3 – Loan Due Reminders
* Send reminders for loans that are due within 30 days.
* Step 6: Commit Changes

**Code**

SET SERVEROUTPUT ON;

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

CREATE TABLE customers (

customer\_id NUMBER PRIMARY KEY,

customer\_name VARCHAR2(100),

age NUMBER,

balance NUMBER,

IsVIP NUMBER(1) DEFAULT 0

);

CREATE TABLE loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER REFERENCES customers(customer\_id),

loan\_amount NUMBER,

interest\_rate NUMBER,

due\_date DATE

);

INSERT INTO customers VALUES (1, 'John Smith', 65, 15000, 0);

INSERT INTO customers VALUES (2, 'Mary Johnson', 45, 8000, 0);

INSERT INTO customers VALUES (3, 'Robert Davis', 72, 25000, 0);

INSERT INTO customers VALUES (4, 'Sarah Wilson', 35, 12000, 0);

INSERT INTO customers VALUES (5, 'Michael Brown', 58, 5000, 0);

INSERT INTO customers VALUES (6, 'Lisa Anderson', 68, 18000, 0);

INSERT INTO loans VALUES (101, 1, 50000, 0.08, SYSDATE + 15);

INSERT INTO loans VALUES (102, 2, 30000, 0.09, SYSDATE + 45);

INSERT INTO loans VALUES (103, 3, 75000, 0.07, SYSDATE + 10);

INSERT INTO loans VALUES (104, 4, 40000, 0.085, SYSDATE + 60);

INSERT INTO loans VALUES (105, 5, 25000, 0.095, SYSDATE + 25);

INSERT INTO loans VALUES (106, 6, 60000, 0.075, SYSDATE + 5);

COMMIT;

-- Scenario 1: Apply 1% interest discount if age > 60

BEGIN

FOR rec IN (

SELECT c.customer\_id, c.customer\_name, c.age, l.loan\_id, l.interest\_rate

FROM customers c

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

) LOOP

IF rec.age > 60 THEN

UPDATE loans

SET interest\_rate = interest\_rate - 0.01

WHERE loan\_id = rec.loan\_id;

DBMS\_OUTPUT.PUT\_LINE('Customer: ' || rec.customer\_name ||

' | Age: ' || rec.age ||

' | Old Rate: ' || rec.interest\_rate ||

' | New Rate: ' || (rec.interest\_rate - 0.01));

END IF;

END LOOP;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('✅ Scenario 1 complete: Discount applied for customers above 60.');

END;

/

-- Scenario 2: Set VIP if balance > 10,000

BEGIN

FOR rec IN (SELECT customer\_id, customer\_name, balance FROM customers) LOOP

IF rec.balance > 10000 THEN

UPDATE customers

SET IsVIP = 1

WHERE customer\_id = rec.customer\_id;

DBMS\_OUTPUT.PUT\_LINE('Customer: ' || rec.customer\_name ||

' | Balance: ' || rec.balance ||

' | VIP: TRUE');

END IF;

END LOOP;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('✅ Scenario 2 complete: VIP status updated.');

END;

/

-- Scenario 3: Reminders for loans due in next 30 days

DECLARE

v\_reminder\_count NUMBER := 0;

BEGIN

FOR rec IN (

SELECT c.customer\_name, l.loan\_id, l.due\_date, l.loan\_amount

FROM customers c

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

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

) LOOP

v\_reminder\_count := v\_reminder\_count + 1;

DBMS\_OUTPUT.PUT\_LINE('🔔 Reminder ' || v\_reminder\_count || ': ' ||

rec.customer\_name || ' has Loan ID ' || rec.loan\_id ||

' due on ' || TO\_CHAR(rec.due\_date, 'DD-MON-YYYY') ||

' | Amount: ' || rec.loan\_amount);

END LOOP;

IF v\_reminder\_count = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('✅ No loans due in next 30 days.');

ELSE

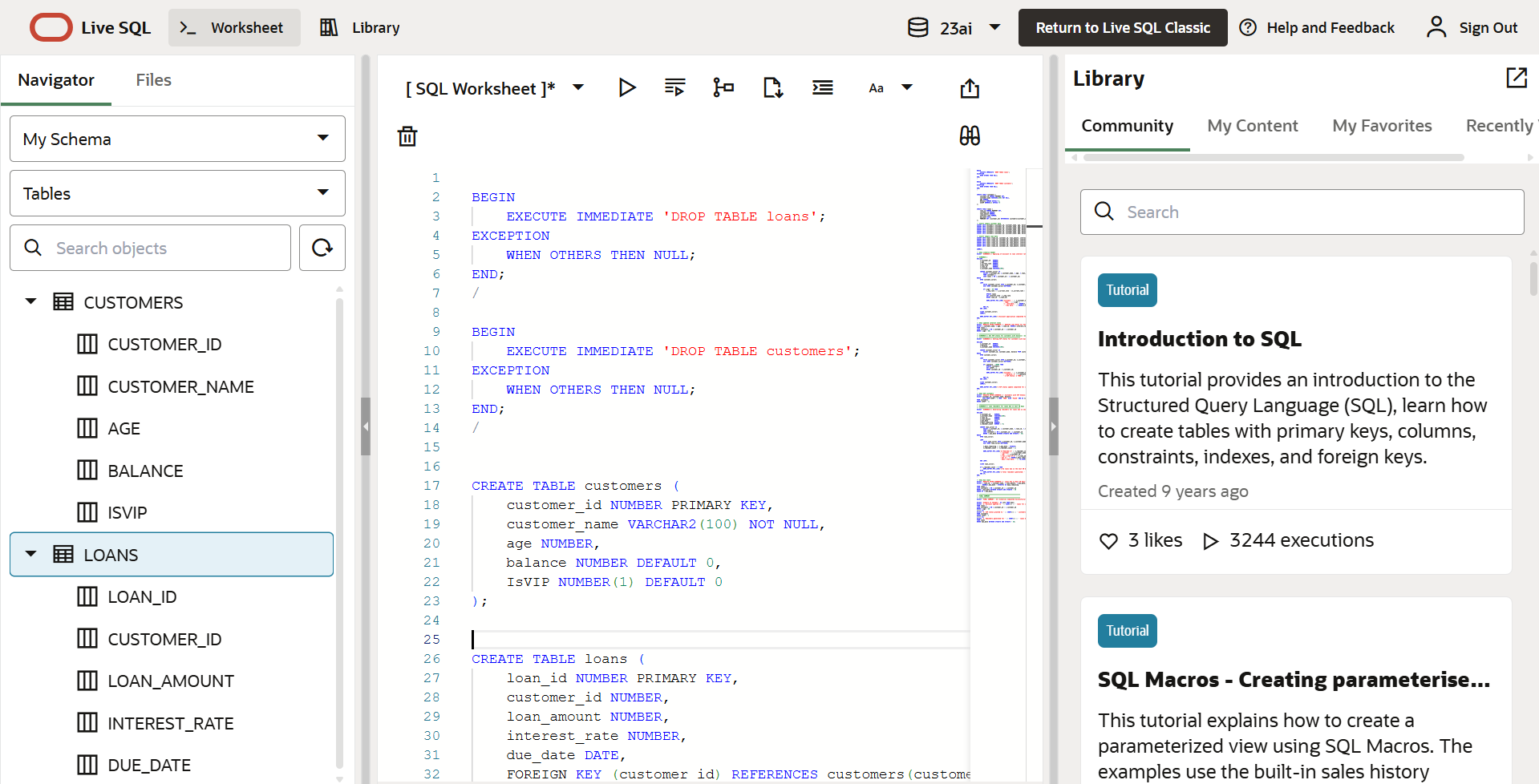
DBMS\_OUTPUT.PUT\_LINE('✅ Scenario 3 complete: Total reminders sent = ' || v\_reminder\_count);

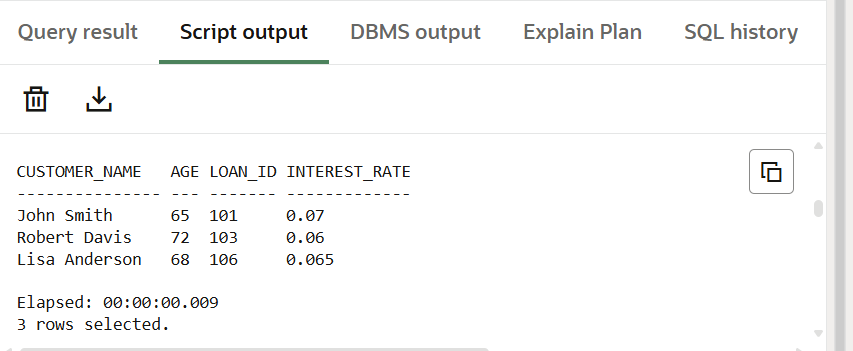
END IF;

END;

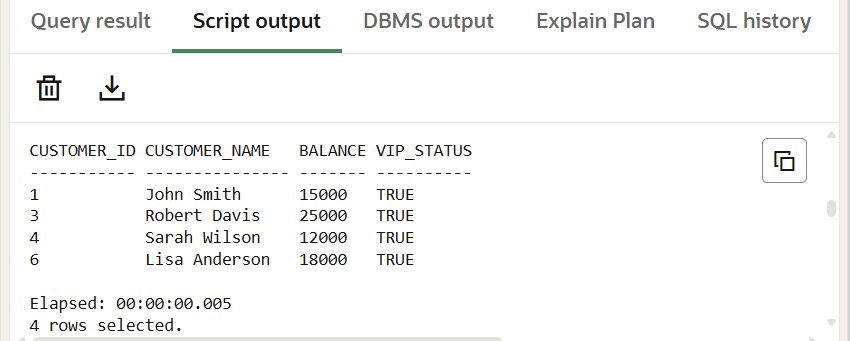
/

**Output**

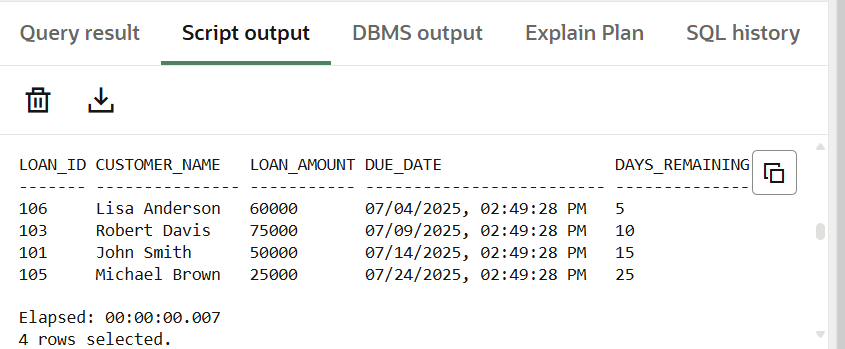
****

****

Scenario 1

****

Scenario 2

****

Scenario 3