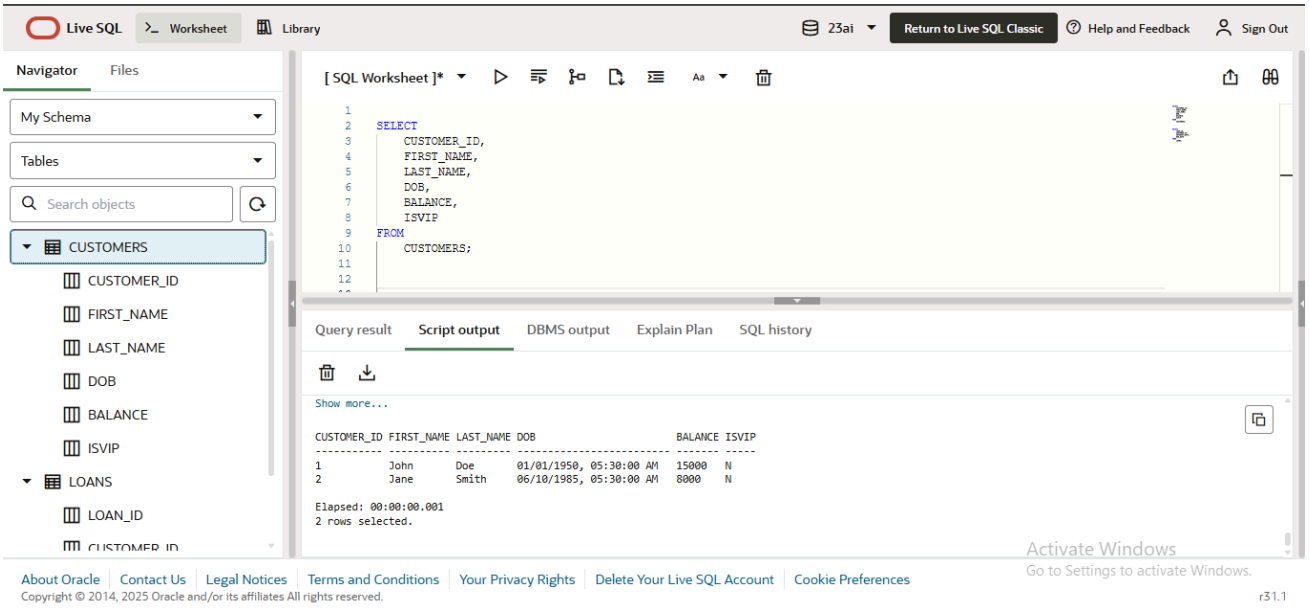
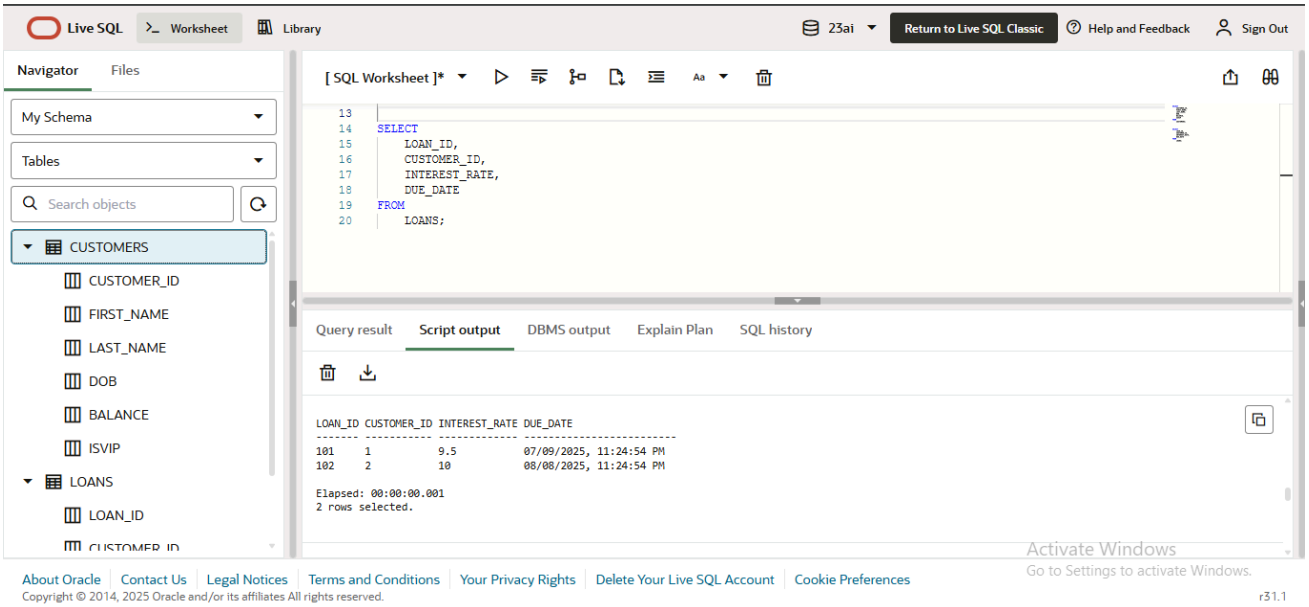
Creation of **CUSTOMER** Table

```
INSERT INTO customers VALUES (1, 'John', 'Doe', TO_DATE('1950-01-01','YYYY-MM-DD'), 15000, 'N');
INSERT INTO customers VALUES (2, 'Jane', 'Smith', TO_DATE('1985-06-10','YYYY-MM-DD'), 8000, 'N');
```



Creation of **LOAN** Table

```
INSERT INTO loans VALUES (101, 1, 9.5, SYSDATE + 10);
INSERT INTO loans VALUES (102, 2, 10.0, SYSDATE + 40);
```



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.

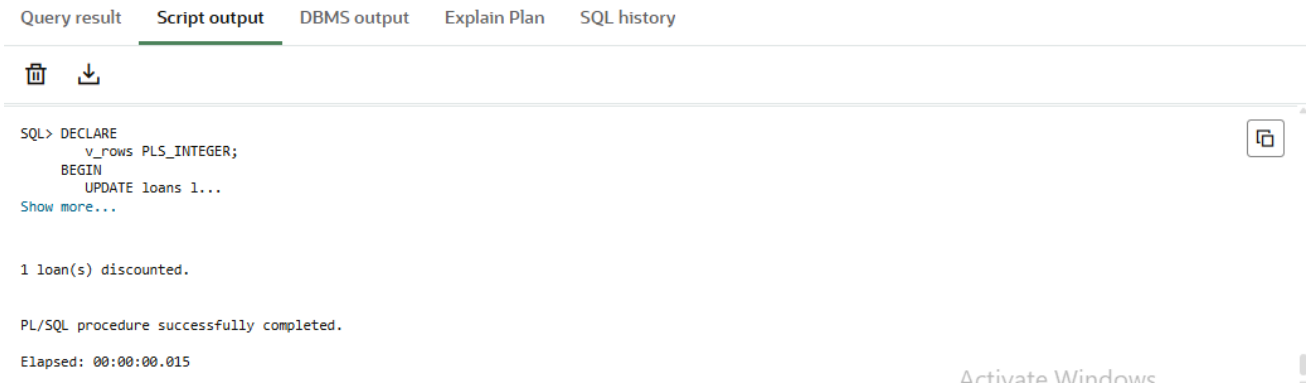
Solution:-

```
SET SERVEROUTPUT ON;

DECLARE
   v_rows PLS_INTEGER;
BEGIN
   UPDATE loans l
   SET    interest_rate = interest_rate * 0.99
   WHERE  EXISTS (
            SELECT 1
            FROM   customers c
            WHERE  c.customer_id = l.customer_id
            AND    TRUNC(MONTHS_BETWEEN(SYSDATE, c.dob)/12) > 60
          );

   v_rows := SQL%ROWCOUNT;
   DBMS_OUTPUT.PUT_LINE(v_rows || ' loan(s) discounted.');
   COMMIT;
END;
/
```

OUTPUT:-



After Senorio 1 the updated table are as follows

```
SELECT * FROM loans;

SELECT * FROM customers;
```

LOAN_ID CUSTOMER_ID INTEREST_RATE DUE_DATE

------- ----------- ------------- -------------------------

101 1 9.41 07/09/2025, 11:24:54 PM

102 2 10 08/08/2025, 11:24:54 PM

Elapsed: 00:00:00.003

2 rows selected.

CUSTOMER_ID FIRST_NAME LAST_NAME DOB BALANCE ISVIP

----------- ---------- --------- ------------------------- ------- -----

1 John Doe 01/01/1950, 05:30:00 AM 15000 N

2 Jane Smith 06/10/1985, 05:30:00 AM 8000 N

Elapsed: 00:00:00.002

2 rows selected.

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.

Solution:-

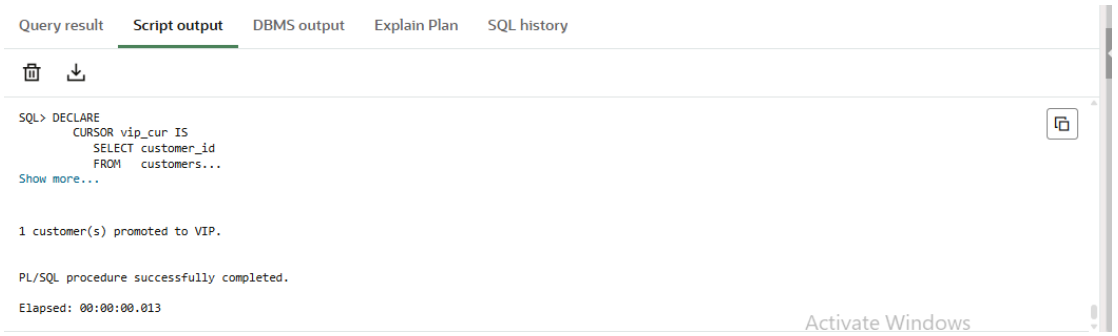
```
SET SERVEROUTPUT ON;

DECLARE
   CURSOR vip_cur IS
      SELECT customer_id
      FROM   customers
      WHERE  balance > 10000;

   v_count PLS_INTEGER := 0;
BEGIN
   FOR rec IN vip_cur LOOP
      UPDATE customers
      SET    isvip = 'Y'
      WHERE  customer_id = rec.customer_id;
      v_count := v_count + 1;
   END LOOP;

   DBMS_OUTPUT.PUT_LINE(v_count || ' customer(s) promoted to VIP.');
   COMMIT;
END;
/
```

OUTPUT:-



After Senorio 2 the updated table is as follows

```
SELECT customer_id, first_name, balance, isvip FROM customers;
```

```
SQL> SELECT customer_id, first_name, balance, isvip FROM customers

CUSTOMER_ID FIRST_NAME BALANCE ISVIP
----------- ---------- ------- -----
1           John       15000   Y
2           Jane       8000    N

Elapsed: 00:00:00.004
2 rows selected.
```

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.

Solution:-

```
SET SERVEROUTPUT ON;

BEGIN
   FOR rec IN (
      SELECT
         l.loan_id,
         l.due_date,
         c.customer_id,
         c.first_name || ' ' || c.last_name AS full_name
      FROM
         loans l
      JOIN
         customers c ON c.customer_id = l.customer_id
      WHERE
         l.due_date BETWEEN SYSDATE AND SYSDATE + 30
      ORDER BY
         l.due_date
   ) LOOP
      DBMS_OUTPUT.PUT_LINE(
         '📅 Reminder: Loan ID ' || rec.loan_id ||
         ' for customer ' || rec.full_name ||
         ' (Customer ID: ' || rec.customer_id ||
         ') is due on ' || TO_CHAR(rec.due_date, 'DD-MON-YYYY')
      );
   END LOOP;
END;
/
```

OUTPUT:-