**PL/SQL -Exercise 1, Exercise 3**

**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:

**STEP1: Inserting Sample Data into Customers and Loans Tables**

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (3, 'Old Customer', TO\_DATE('1955-01-01', 'YYYY-MM-DD'), 5000, SYSDATE)

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (2, 3, 5000, 5, SYSDATE, ADD\_MONTHS(SYSDATE, 60))

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (4, 'Manager', TO\_DATE('1960-06-01', 'YYYY-MM-DD'), 8000, SYSDATE)

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (3, 4, 10000, 6, SYSDATE, ADD\_MONTHS(SYSDATE, 60))

**Explanation:**

* This step adds sample customers and their corresponding loan details into the system.
* Each customer has:
  + A CustomerID, Name, DOB, Balance, and LastModified timestamp.
* Each loan includes:
  + A LoanID, linked CustomerID, LoanAmount, InterestRate, StartDate, and EndDate.

**STEP 2: PL/SQL Block to Apply 1% Interest Discount**

DECLARE

v\_age NUMBER;

BEGIN

FOR c IN (

SELECT c.CustomerID, c.Name, c.DOB,

l.LoanID, l.InterestRate

FROM Customers c

JOIN Loans l ON c.CustomerID = l.CustomerID

)

LOOP

-- Calculate age

v\_age := TRUNC(MONTHS\_BETWEEN(SYSDATE, c.DOB) / 12);

-- Show age for verification

DBMS\_OUTPUT.PUT\_LINE('Customer ' || c.Name || ' is ' || v\_age || ' years old.');

-- Apply discount if age > 60

IF v\_age > 60 THEN

UPDATE Loans

SET InterestRate = c.InterestRate - 1

WHERE LoanID = c.LoanID;

DBMS\_OUTPUT.PUT\_LINE(' Discount applied to ' || c.Name ||

' → New Interest Rate: ' || TO\_CHAR(c.InterestRate - 1));

END IF;

END LOOP;

COMMIT;

END;

**Explanation:**

* We declare a variable v\_age to hold the customer's age.
* A FOR LOOP iterates through all customers who have loans.
* Inside the loop:
  1. Age is calculated using Oracle's MONTHS\_BETWEEN() function:

v\_age := TRUNC(MONTHS\_BETWEEN(SYSDATE, c.DOB) / 12);

* 1. Condition checked: If v\_age > 60, then
  2. Loan interest is updated:

UPDATE Loans SET InterestRate = c.InterestRate - 1 WHERE LoanID = c.LoanID

**STEP 3:** **Verifying the Results**

SELECT c.CustomerID, c.Name, c.DOB,

       TRUNC(MONTHS\_BETWEEN(SYSDATE, c.DOB)/12) AS Age,

       l.LoanID, l.InterestRate

FROM Customers c

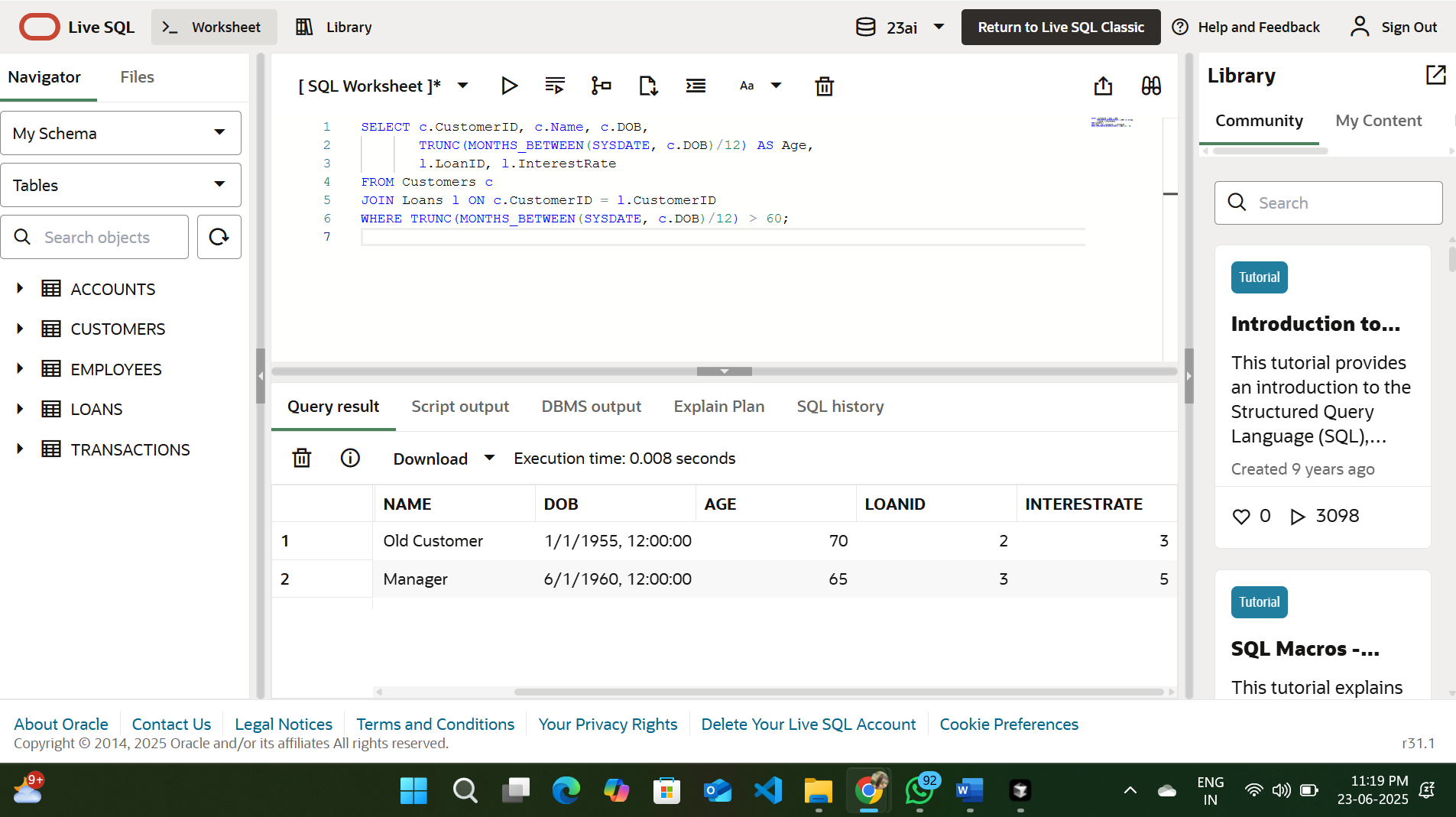
JOIN Loans l ON c.CustomerID = l.CustomerID

WHERE TRUNC(MONTHS\_BETWEEN(SYSDATE, c.DOB)/12) > 60;

**Explanation:**

* A SELECT query joins the Customers and Loans tables to:
  + Recalculate and **display customer age**.
  + **Display updated interest rates** for those above 60

**OUTPUT:**



**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:

**STEP1**: This table does not yet have a column called IsVIP.

So First Step (Very Important):

You need to add a new column to store VIP status using alter command

Steps to check:

1)Go through each row in the Customers table.

2)Check the Balance value.

3)If Balance > 10000, set IsVIP = 'TRUE'.

“If a customer has more than $10,000 in their account, mark them as VIP.”

**ALTER TABLE Customers**

**ADD IsVIP VARCHAR2(5);**

**STEP2:Inserting two customers into customer table**

-- A VIP customer

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (5, 'Ganesh', TO\_DATE('1975-02-10', 'YYYY-MM-DD'), 15000, SYSDATE);

-- A normal customer

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (6, 'Joe', TO\_DATE('1990-11-25', 'YYYY-MM-DD'), 17000, SYSDATE

**Step 3:PL/SQL Block to Set VIP Status**

BEGIN

FOR c IN (SELECT CustomerID, Balance FROM Customers) LOOP

IF c.Balance > 10000 THEN

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = c.CustomerID;

END IF;

END LOOP;

COMMIT;

END;

-This goes through every customer and sets IsVIP = 'TRUE' if balance > 10000

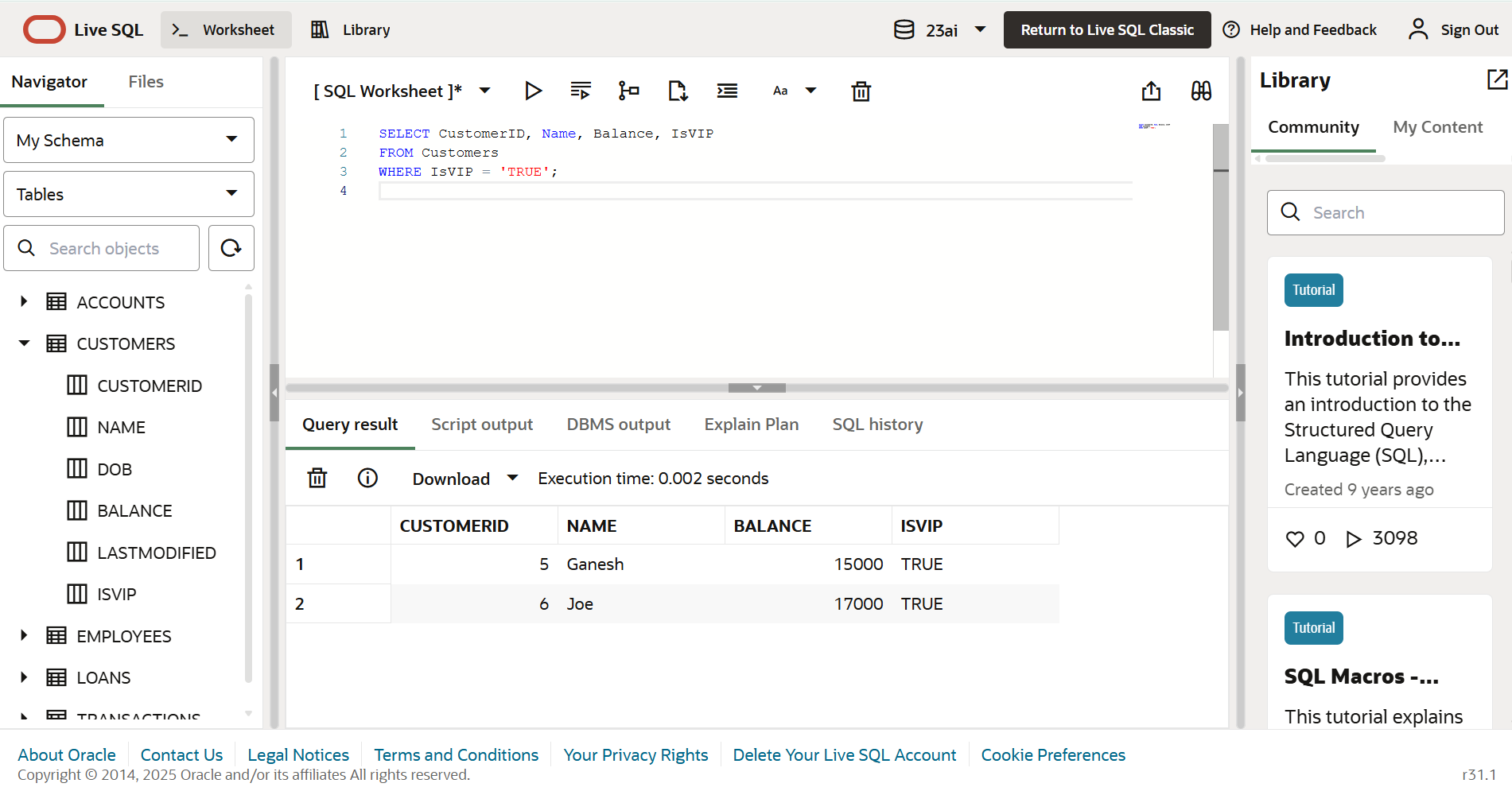
**Step4:Query Only VIP Customers**

SELECT CustomerID, Name, Balance, IsVIP

FROM Customers

WHERE IsVIP = 'TRUE';

**OUTPUT:**



**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.

**STEP 1:Inserting values into the table**

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (8, 'Goutham', TO\_DATE('1980-01-01', 'YYYY-MM-DD'), 9000, SYSDATE)

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)

VALUES (5, 8, 6000, 6, SYSDATE, SYSDATE + 10)

COMMIT;

**STEP 2:PL/SQl block**

BEGIN

FOR r IN (

SELECT l.LoanID, l.EndDate, c.Name

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE(

'Reminder: Customer ' || r.Name ||

'''s loan (LoanID: ' || r.LoanID || ') is due on ' || TO\_CHAR(r.EndDate, 'DD-Mon-YYYY')

);

END LOOP;

END;

**OUTPUT:**

