**WEEK-2**

**Exercise 1: Control Structures**

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

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.

**Query**

UPDATE loans

SET InterestRate = InterestRate - 1

WHERE CustomerID IN (

SELECT CustomerID

FROM Customers

WHERE EXTRACT(YEAR FROM curdate()) - EXTRACT(YEAR FROM DOB) > 60

);

SELECT \* FROM loans

WHERE CustomerID IN (

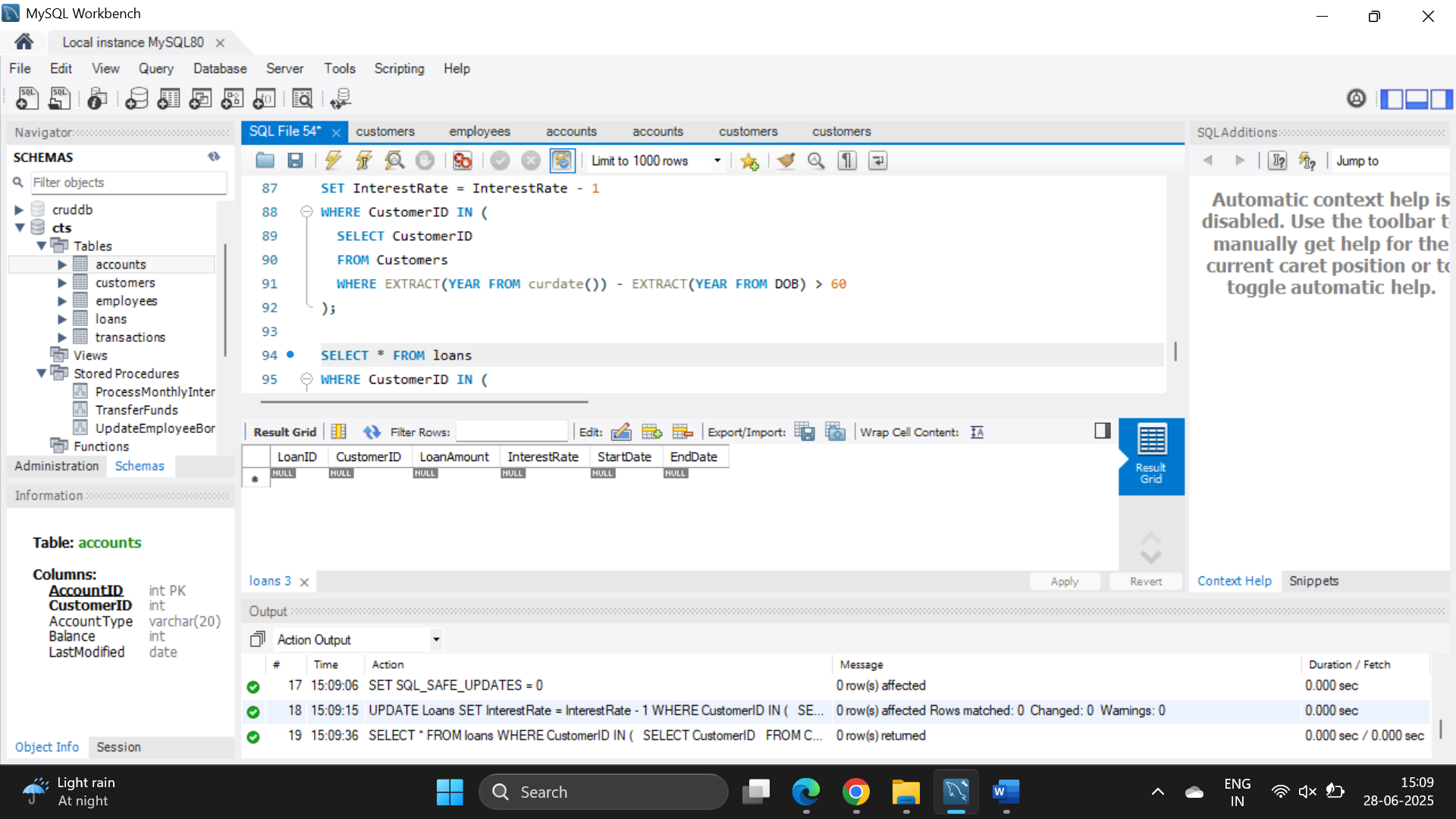
SELECT CustomerID

FROM Customers

WHERE EXTRACT(YEAR FROM curdate()) - EXTRACT(YEAR FROM DOB) > 60

);

OUTPUT



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

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.

**QUERY**

ALTER TABLE customers ADD IsVIP VARCHAR(5);

UPDATE customers

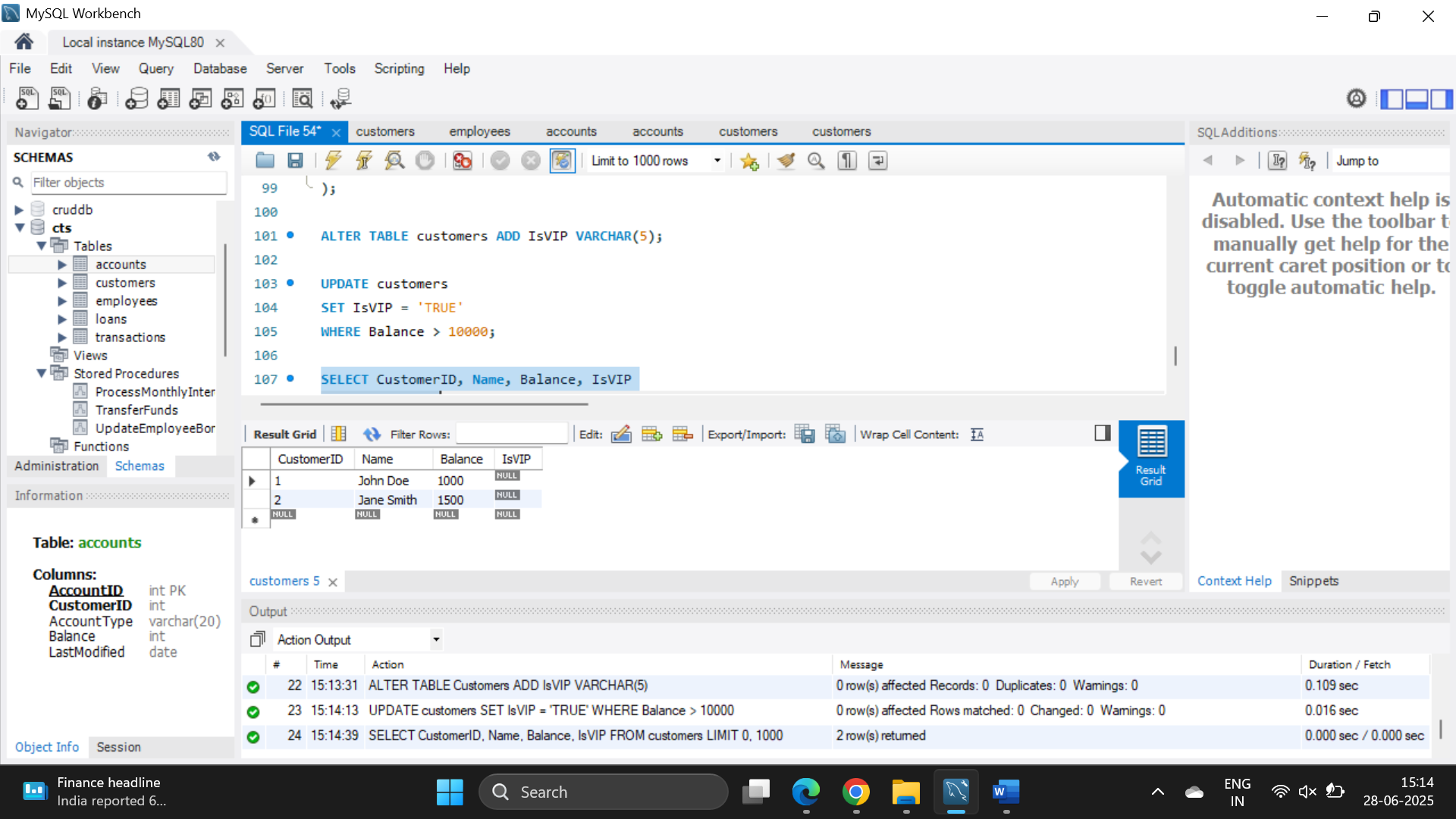
SET IsVIP = 'TRUE'

WHERE Balance > 10000;

SELECT CustomerID, Name, Balance, IsVIP

FROM customers;

OUTPUT



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

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

**QUERY**

SELECT

l.LoanID,

c.Name AS CustomerName,

l.EndDate,

CONCAT('Reminder: Loan ID ', l.LoanID, ' for ', c.Name,

' is due on ', DATE\_FORMAT(l.EndDate, '%d-%b-%Y')) AS ReminderMessage

FROM loans l

JOIN customers c ON l.CustomerID = c.CustomerID

WHERE l.EndDate BETWEEN CURDATE() AND CURDATE() + INTERVAL 30 DAY;

OUTPUT

