**Week 2\_PL/SQL \_HandsOn**

**Exercise 1: Control Structures**

**Scenario 1:**

CREATE PROCEDURE ApplySeniorDiscount()

BEGIN

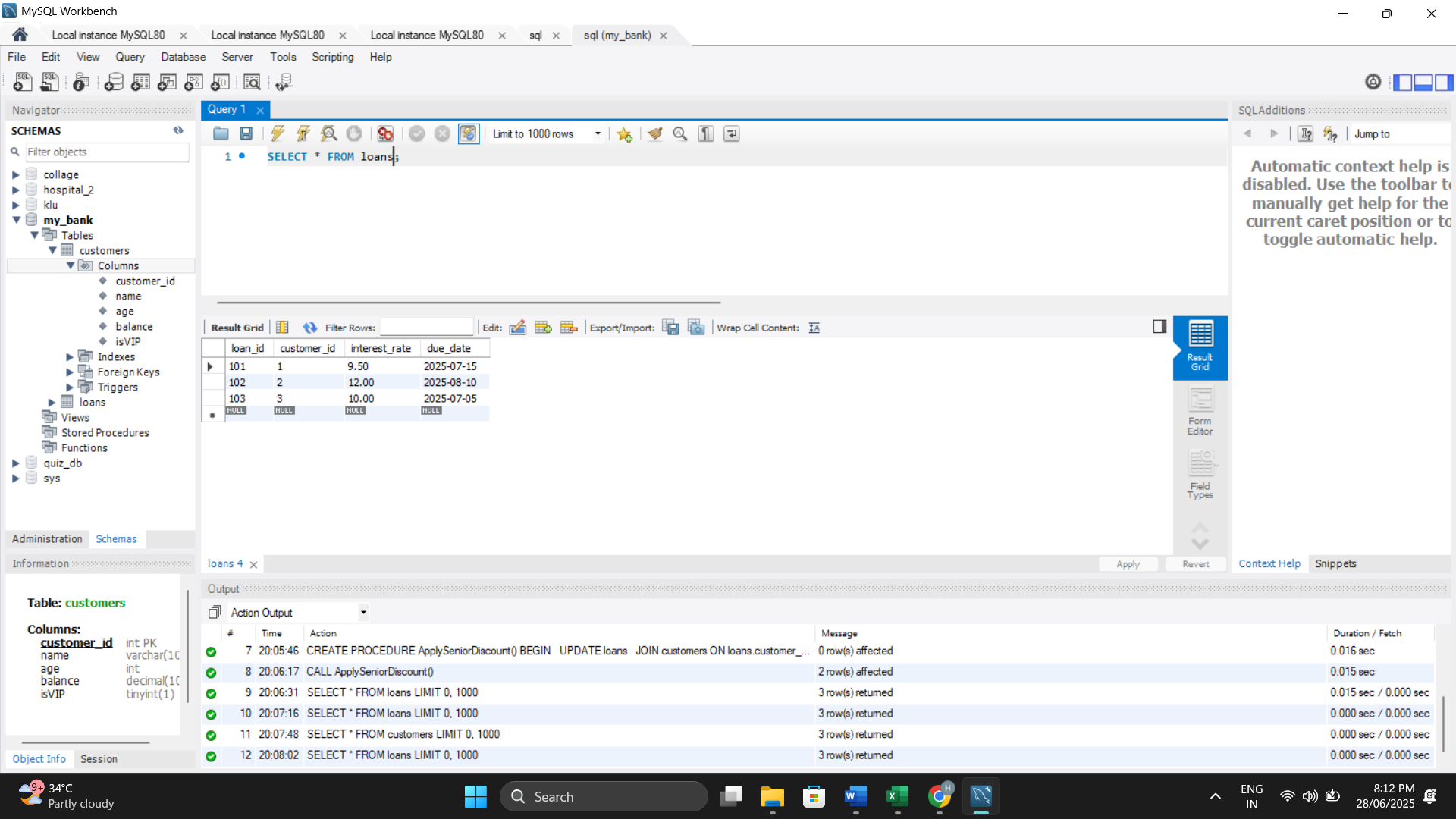
UPDATE loans

JOIN customers ON loans.customer\_id = customers.customer\_id

SET loans.interest\_rate = loans.interest\_rate - 1

WHERE customers.age > 60;

END $$



**Scenario 2:**

CREATE PROCEDURE PromoteToVIP()

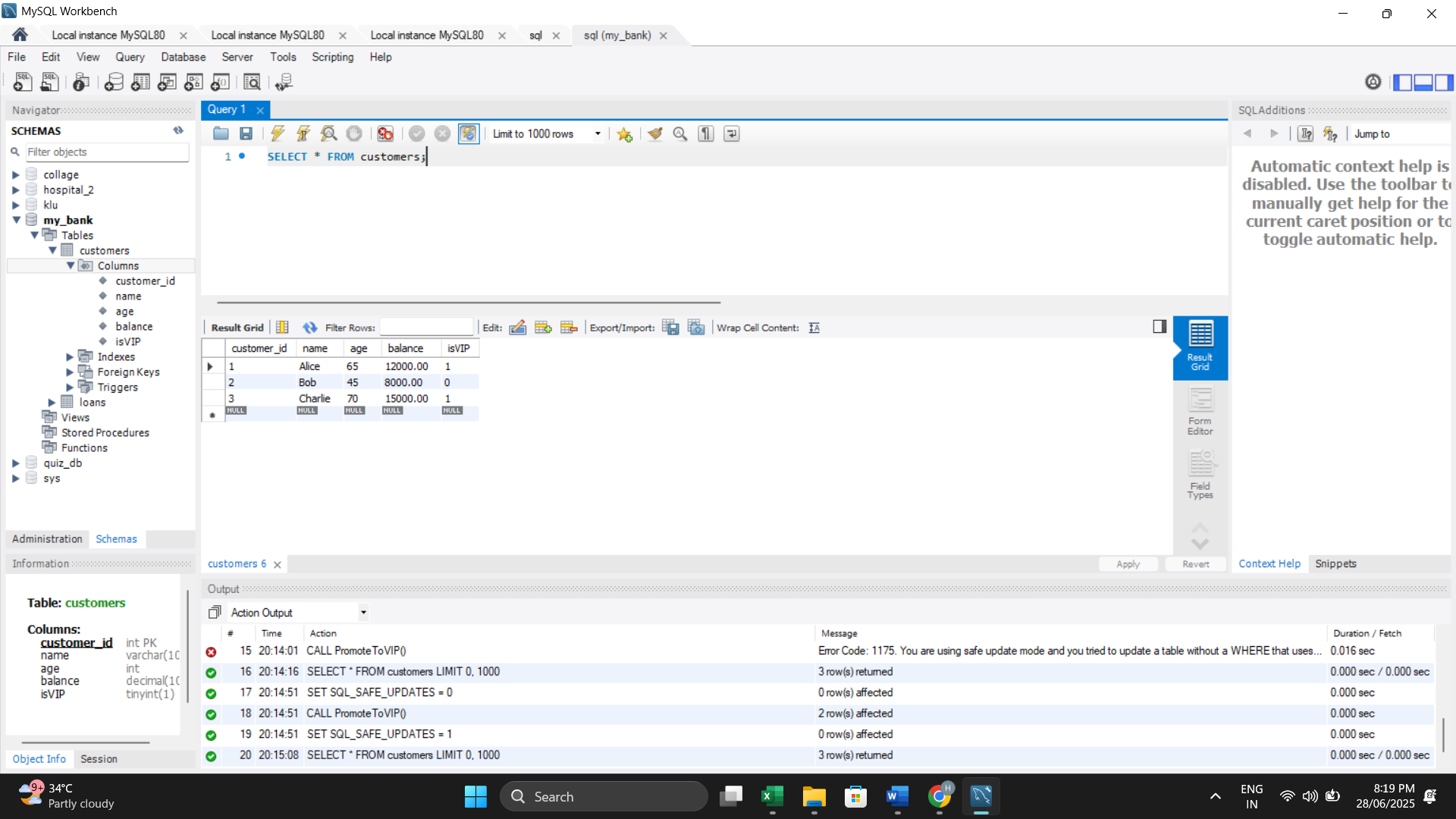
BEGIN

UPDATE customers

SET isVIP = TRUE

WHERE balance > 10000;

END $$



**Scenario 3:**

SELECT

CONCAT('Reminder: ', c.name, ', your loan is due on ', DATE\_FORMAT(l.due\_date, '%d-%b-%Y')) AS Reminder

FROM

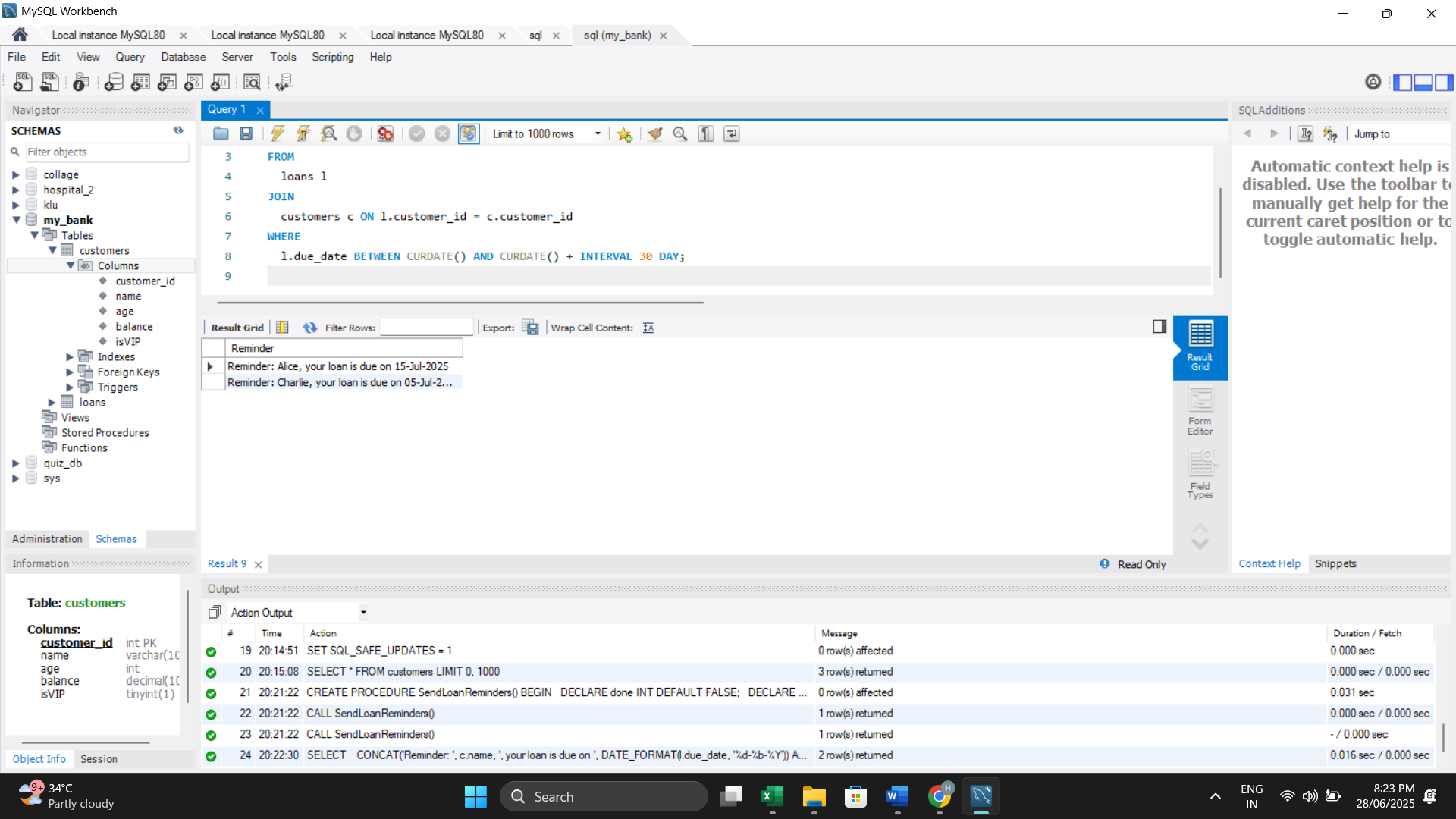
loans l

JOIN

customers c ON l.customer\_id = c.customer\_id

WHERE

l.due\_date BETWEEN CURDATE() AND CURDATE() + INTERVAL 30 DAY;



**Exercise 3: Stored Procedures**

**Scenario 1:**

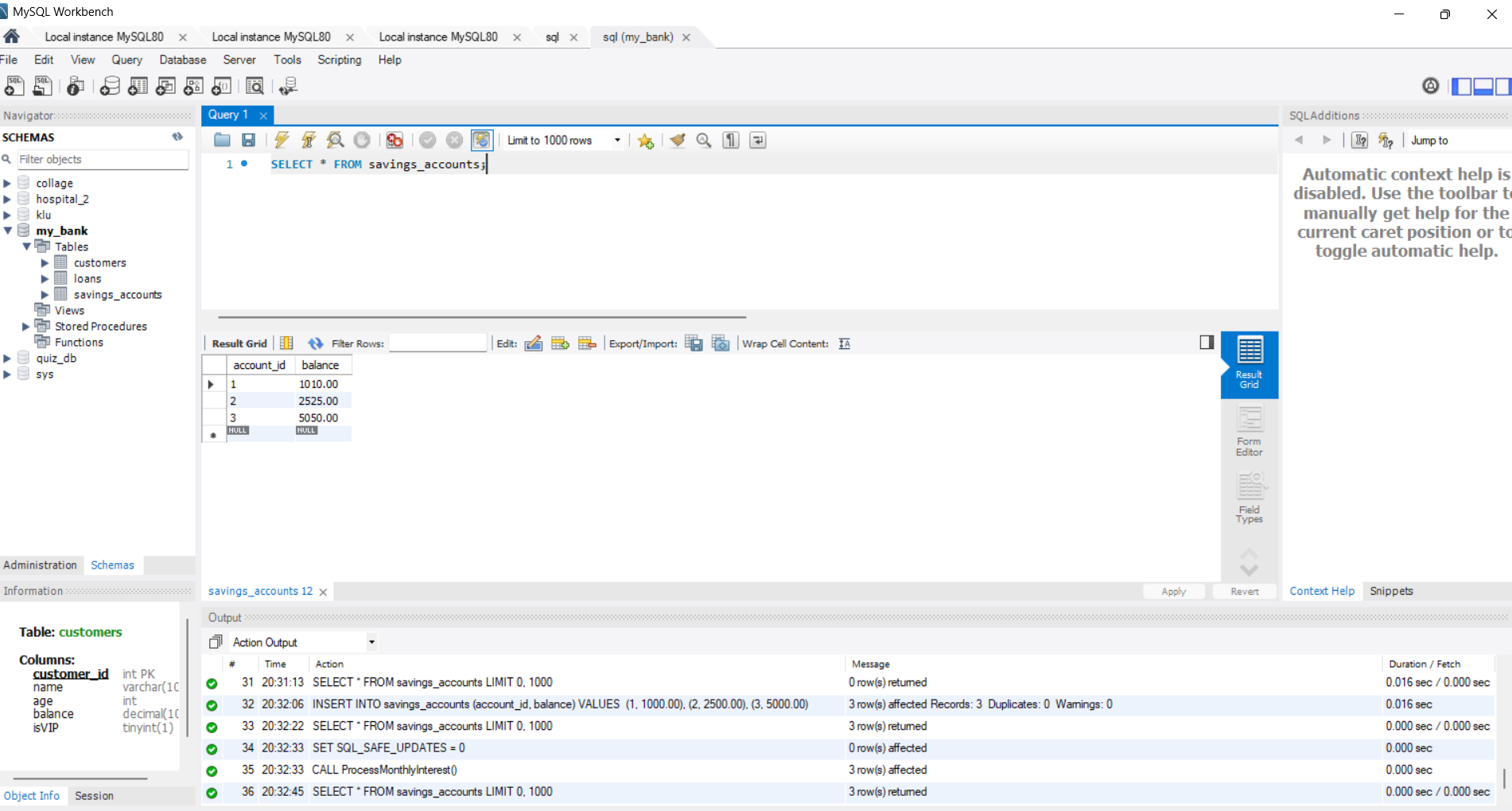
CREATE PROCEDURE ProcessMonthlyInterest()

BEGIN

UPDATE savings\_accounts

SET balance = balance + (balance \* 0.01);

END $$



**Scenario 2:**

CREATE PROCEDURE UpdateEmployeeBonus(IN dept\_name VARCHAR(50), IN bonus\_percent DECIMAL(5,2))

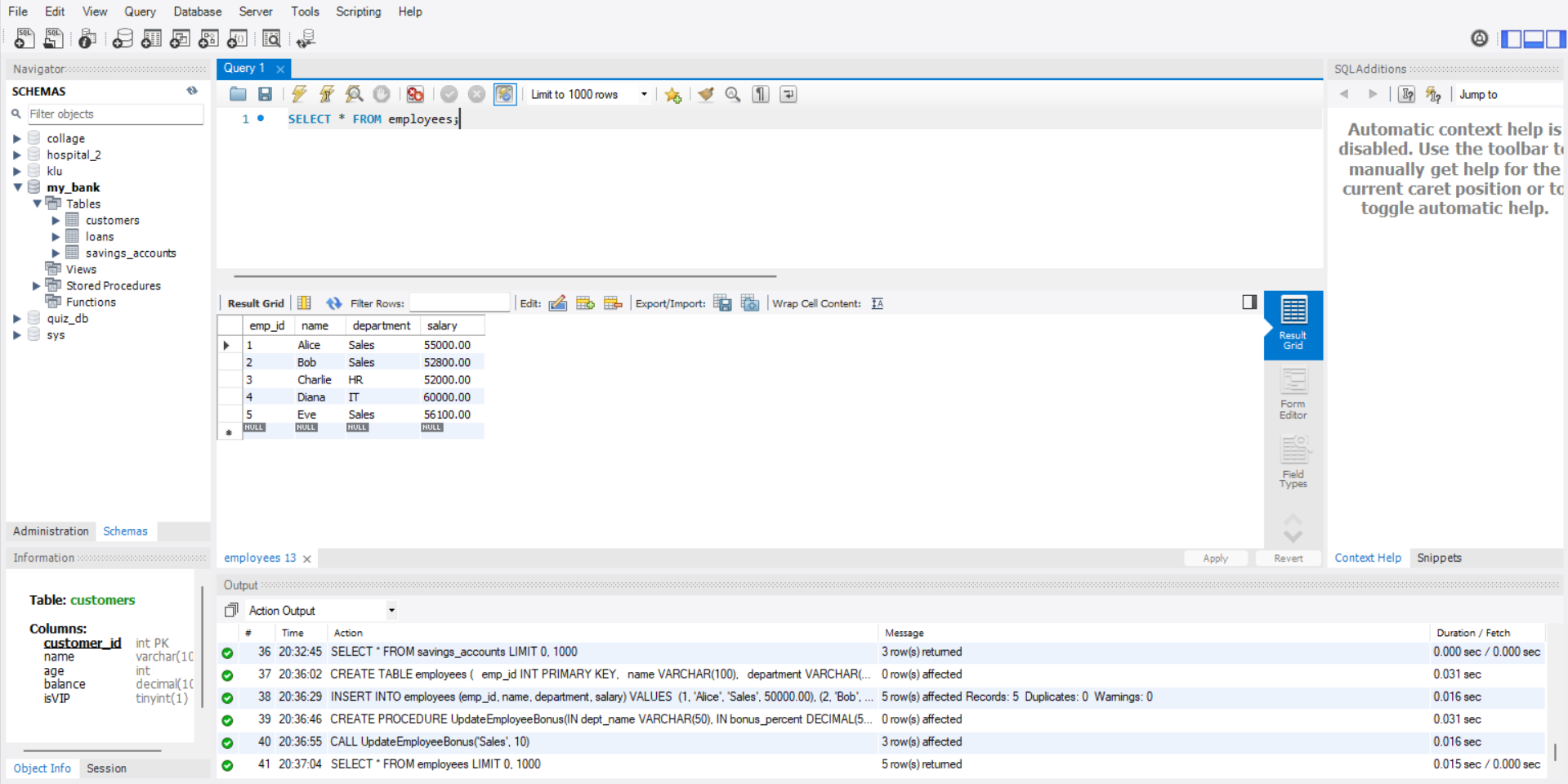
BEGIN

UPDATE employees

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

WHERE department = dept\_name;

END $$



**Scenario 3:**

CREATE PROCEDURE TransferFunds(

IN from\_acc INT,

IN to\_acc INT,

IN amount DECIMAL(10,2)

)

BEGIN

DECLARE from\_balance DECIMAL(10,2);

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

IF from\_balance >= amount THEN

UPDATE accounts SET balance = balance - amount WHERE account\_id = from\_acc;

UPDATE accounts SET balance = balance + amount WHERE account\_id = to\_acc;

ELSE

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Insufficient funds';

END IF;

END $$

