**WEEK-2**

**Scenario 1:** The bank needs to process monthly interest for all savings accounts.

Write a stored procedure **ProcessMonthlyInterest** that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.

Procedure:

CREATE DEFINER=`root`@`localhost` PROCEDURE `ProcessMonthlyInterest`()

BEGIN

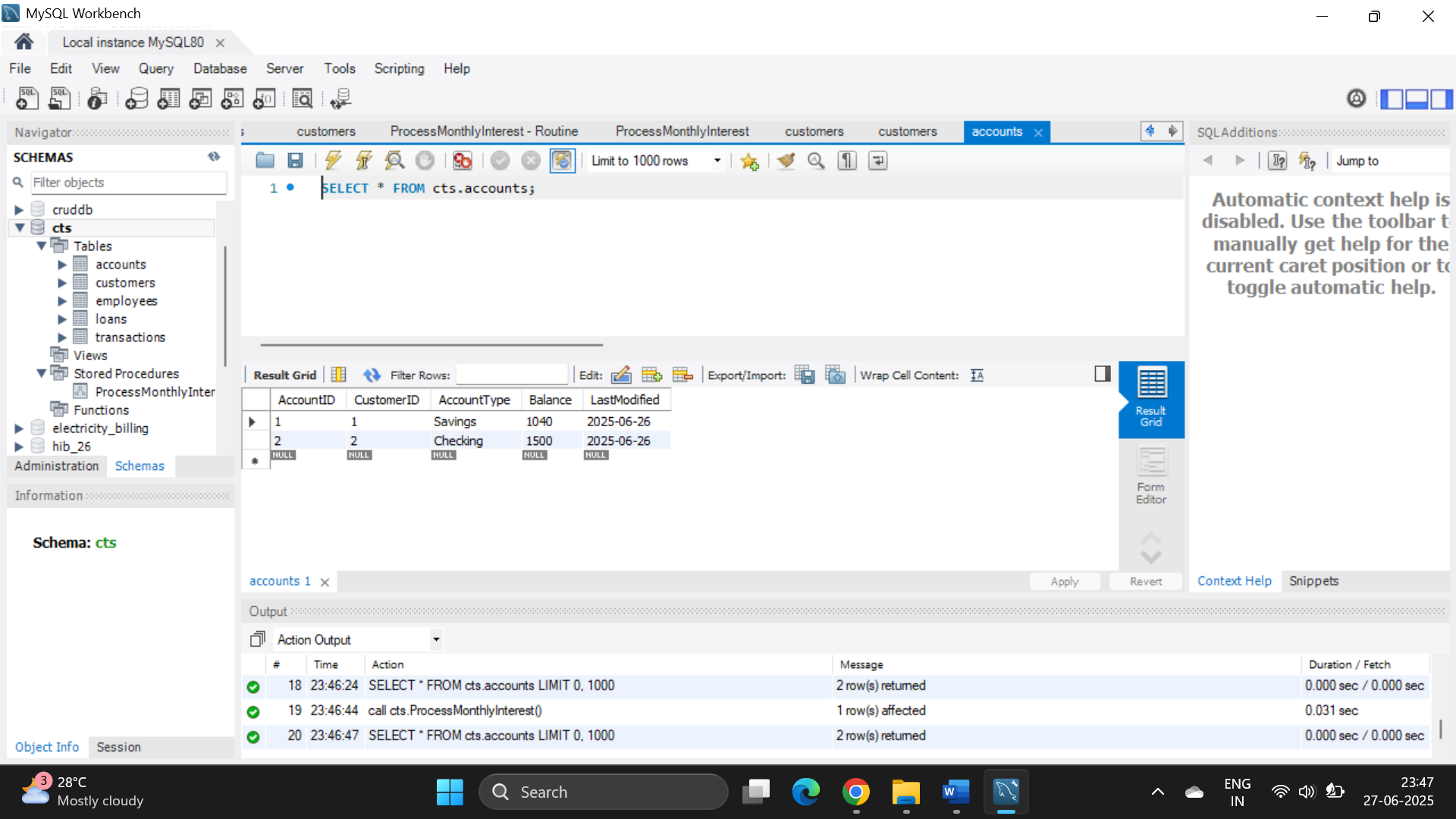
UPDATE Accounts

SET Balance = Balance + (Balance \* 0.01)

WHERE AccountType = 'Savings';

END

OUTPUT

****

Scenario 2: The bank wants to implement a bonus scheme for employees based on their performance.

Write a stored procedure UpdateEmployeeBonus that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.

Procedure:

CREATE DEFINER=`root`@`localhost` PROCEDURE `UpdateEmployeeBonus`(

IN deptName VARCHAR(50),

IN bonusPercent DECIMAL(5,2)

)

BEGIN

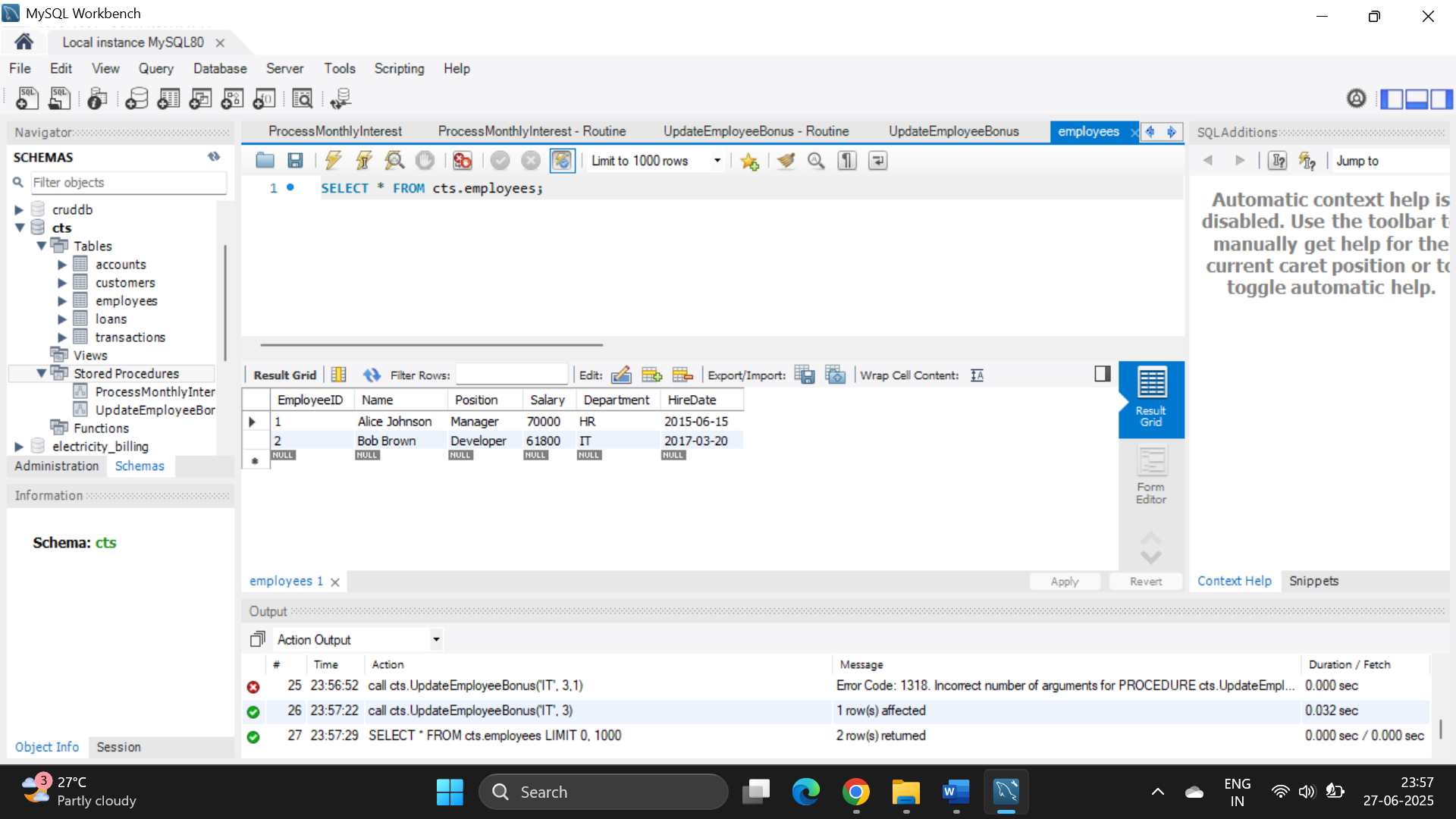
UPDATE Employees

SET Salary = Salary + (Salary \* (bonusPercent / 100))

WHERE Department = deptName;

END

OUTPUT

****

Scenario 3: Customers should be able to transfer funds between their accounts.

Write a stored procedure TransferFunds that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

Procedure

CREATE DEFINER=`root`@`localhost` PROCEDURE `TransferFunds`(

IN from\_acc INT,

IN to\_acc INT,

IN amount DECIMAL(10,2)

)

BEGIN

DECLARE bal DECIMAL(10,2);

SELECT balance INTO bal FROM Accounts WHERE AccountID = from\_acc;

IF bal >= amount THEN

UPDATE Accounts SET balance = balance - amount WHERE AccountID = from\_acc;

UPDATE Accounts SET balance = balance + amount WHERE AccountID = to\_acc;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Not enough balance!';

END IF;

END

OUTPUT

