**Exercise 1: Control Structures**

**Scenario 1**

CREATE PROCEDURE ApplyDiscountForSeniorCustomers()

BEGIN

UPDATE customers

SET loan\_interest\_rate = loan\_interest\_rate - 1

WHERE age > 60;

END $$

**Scenario 2**

CREATE PROCEDURE PromoteVIPCustomers()

BEGIN

UPDATE customers

SET isVIP = TRUE

WHERE balance > 10000;

END $$

**Scenario 3**

CREATE PROCEDURE RemindDueLoans()

BEGIN

DECLARE done INT DEFAULT FALSE;

DECLARE c\_id INT;

DECLARE l\_id INT;

DECLARE due DATE;

DECLARE cust\_name VARCHAR(100);

DECLARE cur CURSOR FOR

SELECT loans.loan\_id, loans.customer\_id, loans.due\_date, customers.name

FROM loans

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

WHERE loans.due\_date <= CURDATE() + INTERVAL 30 DAY;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN cur;

read\_loop: LOOP

FETCH cur INTO l\_id, c\_id, due, cust\_name;

IF done THEN

LEAVE read\_loop;

END IF;

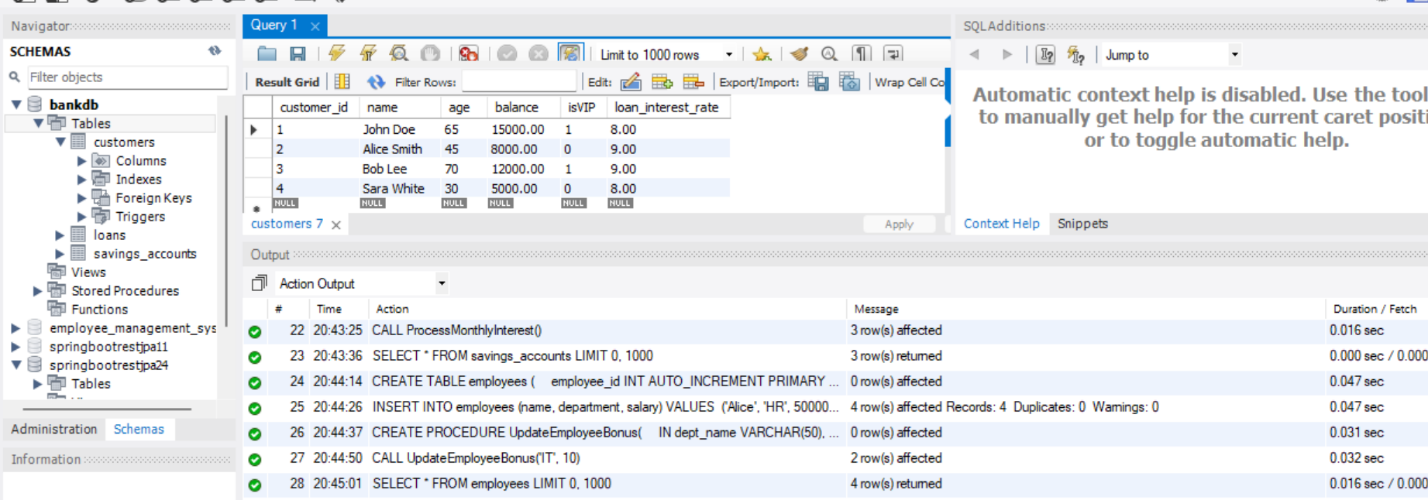
SELECT CONCAT('Reminder: Customer ', cust\_name, ' (ID: ', c\_id, ') has a loan due on ', due) AS ReminderMessage;

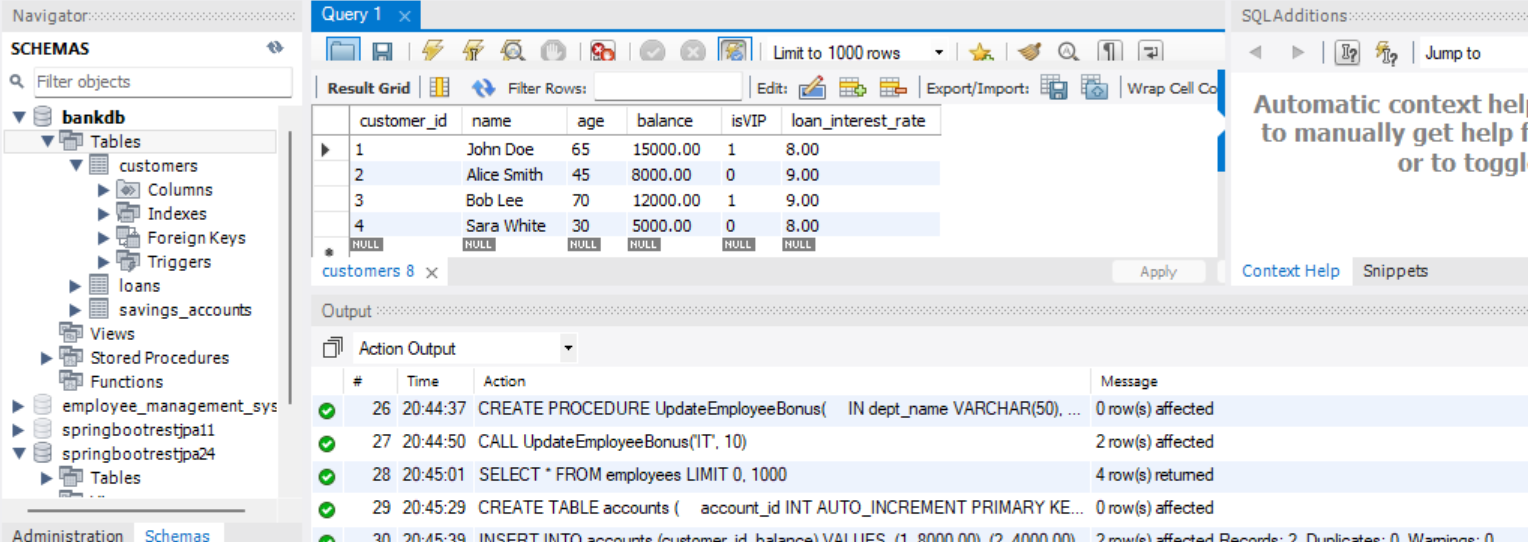
END LOOP;

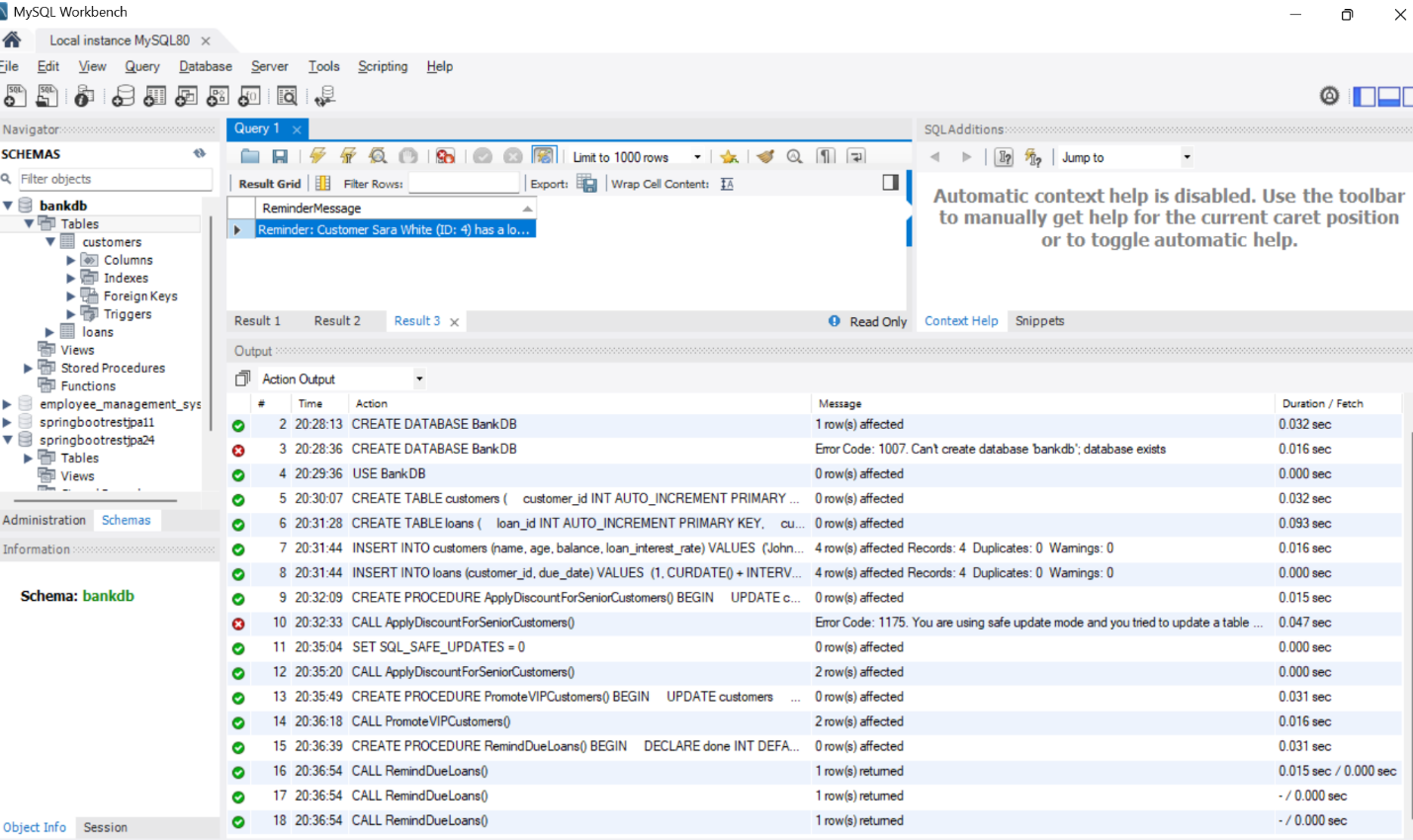
CLOSE cur;

END $$

**Output:**

****

****

****

**Exercise 3: Stored Procedures**

**Scenario 1:**

CREATE PROCEDURE ProcessMonthlyInterest()

BEGIN

UPDATE savings\_accounts

SET balance = balance \* 1.01;

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\_account INT,

IN to\_account INT,

IN amount DECIMAL(12,2)

)

BEGIN

DECLARE from\_balance DECIMAL(12,2);

-- Get balance of source account

SELECT balance INTO from\_balance

FROM accounts

WHERE account\_id = from\_account;

-- Check if sufficient

IF from\_balance >= amount THEN

-- Deduct from source

UPDATE accounts

SET balance = balance - amount

WHERE account\_id = from\_account;

-- Add to target

UPDATE accounts

SET balance = balance + amount

WHERE account\_id = to\_account;

ELSE

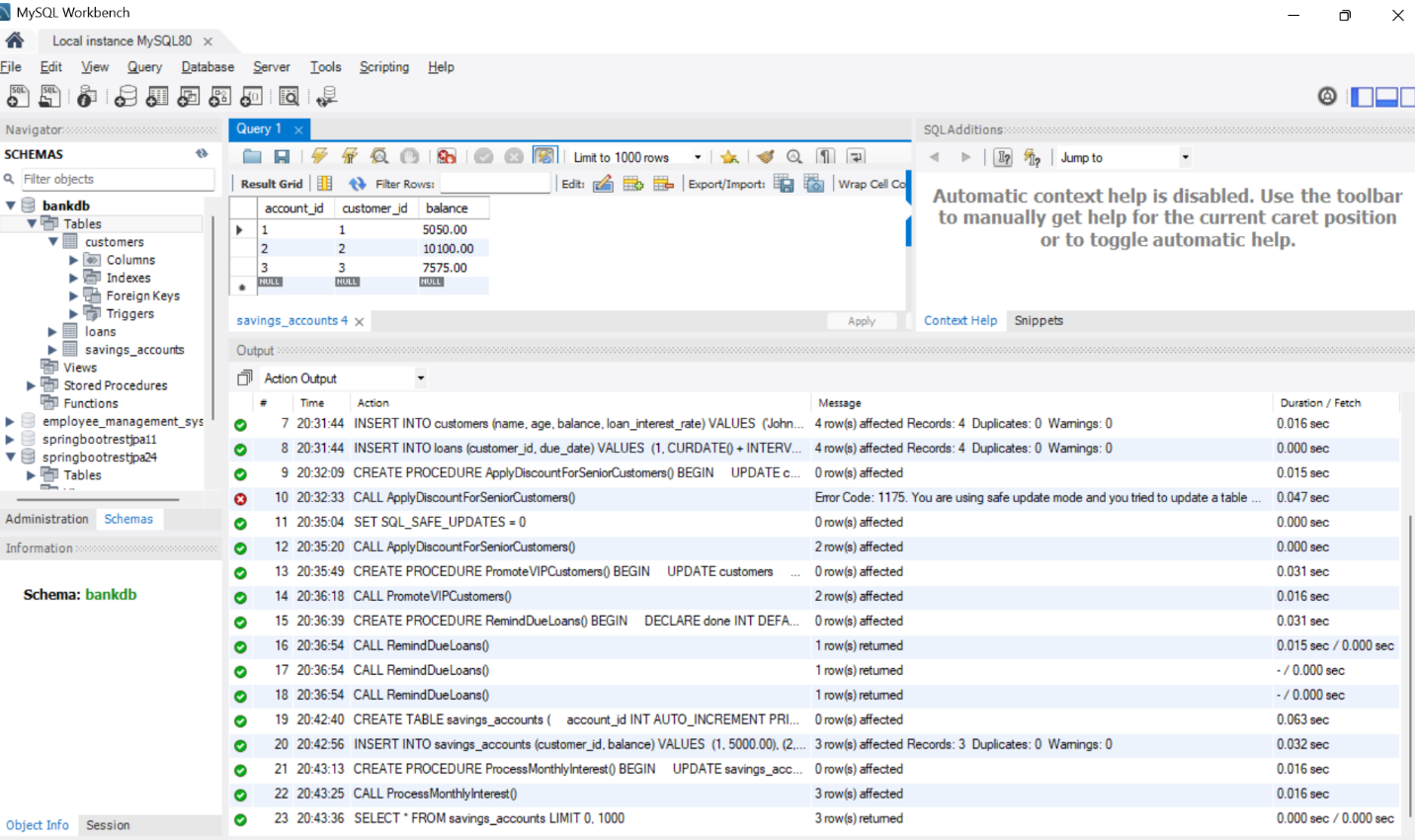
SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Insufficient balance in source account';

END IF;

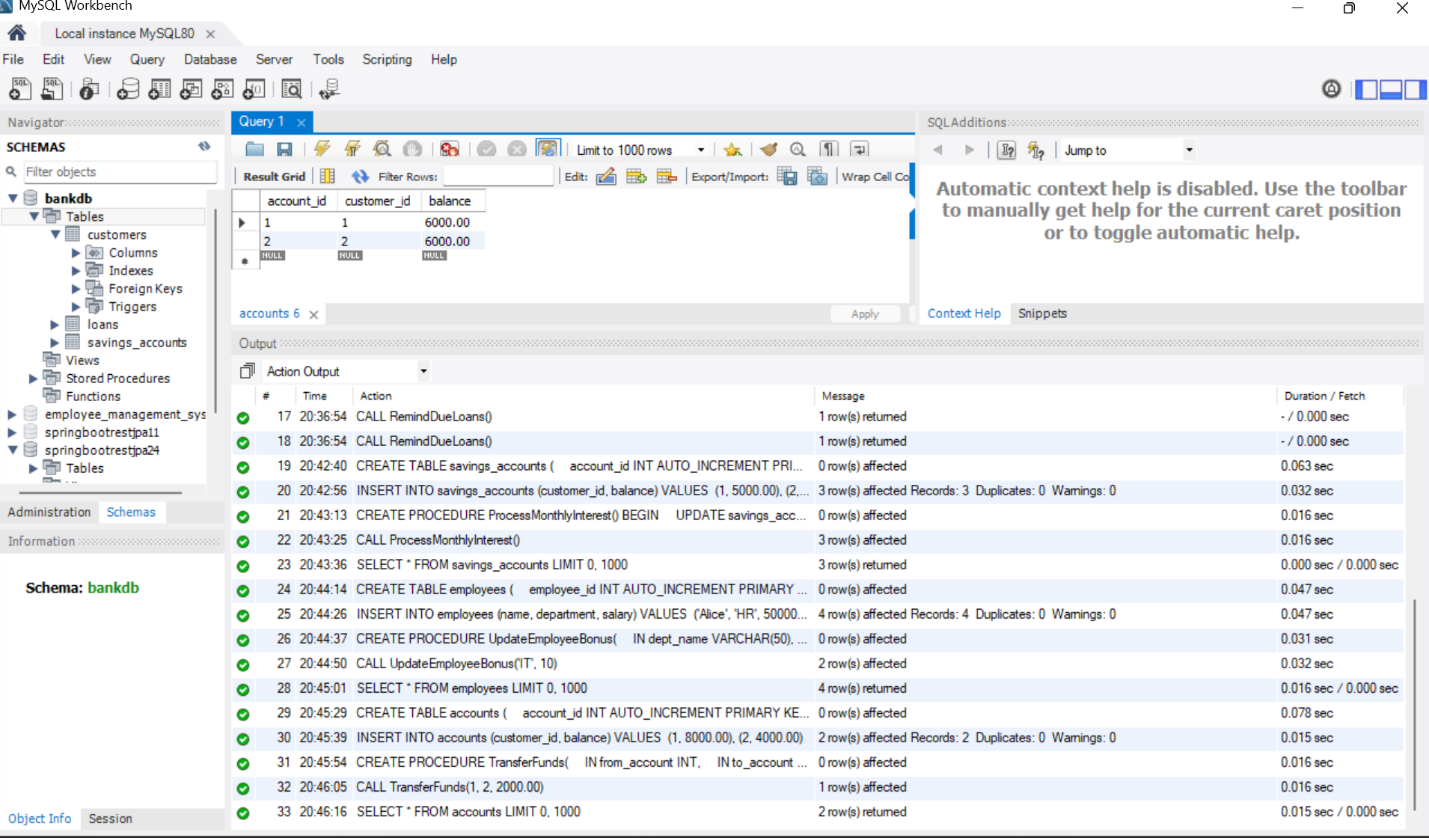
END $$

**Output:**

****

**A screenshot of a computer

AI-generated content may be incorrect.**

****