**Exercise 1: Control Structures**

create database ControlStructures;

use ControlStructures;

CREATE TABLE customers (

customer\_id INTEGER PRIMARY KEY,

name VARCHAR(50),

age INTEGER,

balance NUMERIC(10, 2),

isvip BOOLEAN DEFAULT FALSE

);

CREATE TABLE loans (

loan\_id INTEGER PRIMARY KEY,

customer\_id INTEGER REFERENCES customers(customer\_id),

interest\_rate NUMERIC(5, 2),

due\_date DATE

);

INSERT INTO customers (customer\_id, name, age, balance, isvip) VALUES

(1, 'Alice', 65, 15000.00, FALSE),

(2, 'Bob', 45, 8000.00, FALSE),

(3, 'Charlie', 70, 5000.00, FALSE),

(4, 'Diana', 30, 12000.00, FALSE);

INSERT INTO loans (loan\_id, customer\_id, interest\_rate, due\_date) VALUES

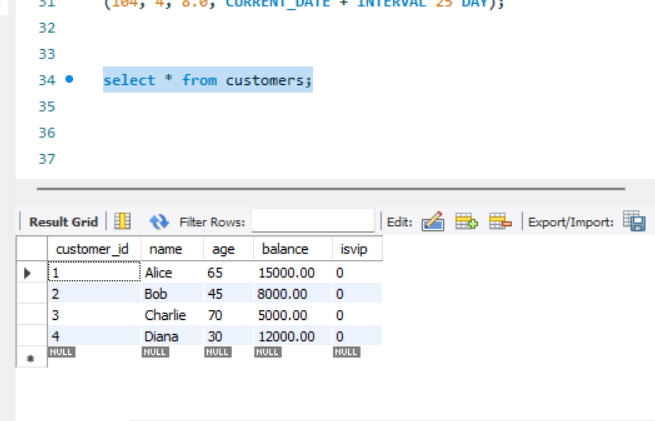
(101, 1, 8.5, CURRENT\_DATE + INTERVAL 10 DAY),

(102, 2, 9.0, CURRENT\_DATE + INTERVAL 40 DAY),

(103, 3, 7.5, CURRENT\_DATE + INTERVAL 5 DAY),

(104, 4, 8.0, CURRENT\_DATE + INTERVAL 25 DAY);

select \* from customers;



**Scenario 1:**

CREATE PROCEDURE ApplyInterestDiscount()

BEGIN

DECLARE done INT DEFAULT FALSE;

DECLARE loanId INT;

DECLARE cur CURSOR FOR

SELECT l.loan\_id

FROM loans l

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

WHERE c.age > 60;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN cur;

read\_loop: LOOP

FETCH cur INTO loanId;

IF done THEN

LEAVE read\_loop;

END IF;

UPDATE loans

SET interest\_rate = interest\_rate \* 0.99

WHERE loan\_id = loanId;

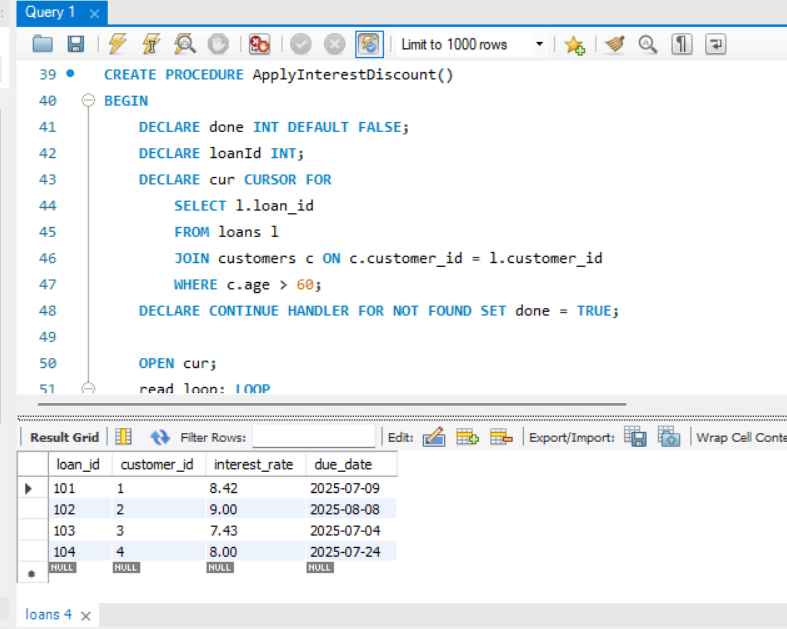
END LOOP;

CLOSE cur;

END$$

Call ApplyInterestDiscount();

select \* from loans;



**Scenario 2:**

CREATE PROCEDURE PromoteToVIP()

BEGIN

DECLARE done INT DEFAULT FALSE;

DECLARE custId INT;

DECLARE cur CURSOR FOR

SELECT customer\_id FROM customers WHERE balance > 10000;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN cur;

read\_loop: LOOP

FETCH cur INTO custId;

IF done THEN

LEAVE read\_loop;

END IF;

UPDATE customers

SET isvip = TRUE

WHERE customer\_id = custId;

END LOOP;

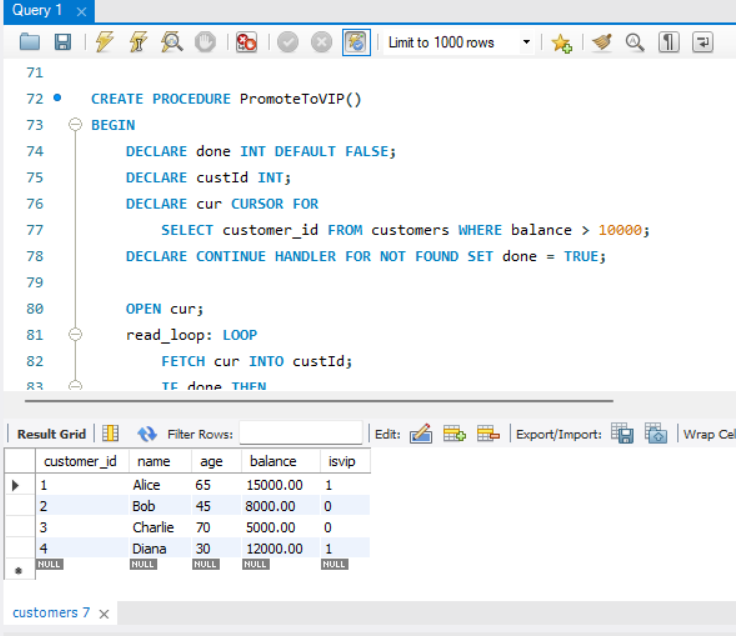
CLOSE cur;

END$$

DELIMITER ;

CALL PromoteToVIP();

select \* from customers;



**Scenario 3:**

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

BEGIN

DECLARE done INT DEFAULT FALSE;

DECLARE loanId INT;

DECLARE custName VARCHAR(50);

DECLARE dueDate DATE;

DECLARE cur CURSOR FOR

SELECT l.loan\_id, c.name, l.due\_date

FROM loans l

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

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

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN cur;

read\_loop: LOOP

FETCH cur INTO loanId, custName, dueDate;

IF done THEN

LEAVE read\_loop;

END IF;

SELECT CONCAT('Reminder: ', custName, '''s loan (ID ', loanId, ') is due on ', DATE\_FORMAT(dueDate, '%d-%b-%Y')) AS message;

END LOOP;

CLOSE cur;

END

