PL/SQL programming

**Exercise 1: Control Structures**

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

* + **Question:** 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.

CODE:

DELIMITER //  
  
CREATE PROCEDURE *ApplyDiscountForSeniors*()  
BEGIN  
 DECLARE done INT DEFAULT FALSE;  
 DECLARE cust\_id INT;  
 DECLARE interest\_rate DECIMAL(5,2);  
 DECLARE cur CURSOR FOR  
 SELECT c.CustomerID  
 FROM CUSTOMERS c  
 JOIN LOANS l ON c.CustomerID = l.CustomerID  
 WHERE *TIMESTAMPDIFF*(YEAR, c.DateOfBirth, *CURDATE*()) > 60;  
 DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;  
  
 OPEN cur;  
  
 read\_loop: LOOP  
 FETCH cur INTO cust\_id;  
 IF done THEN  
 LEAVE read\_loop;  
 END IF;  
  
 UPDATE LOANS  
 SET InterestRate = InterestRate - 1  
 WHERE CustomerID = cust\_id;  
  
 END LOOP;  
  
 CLOSE cur;  
END //  
  
DELIMITER ;

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

* + **Question:** 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.

CODE:

DELIMITER //  
  
CREATE PROCEDURE *PromoteVIPCustomers*()  
BEGIN  
 UPDATE CUSTOMERS  
 SET IsVIP = TRUE  
 WHERE Balance > 10000;  
  
  
END //  
  
  
DELIMITER ;

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

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

CODE:  
DELIMITER //  
  
CREATE PROCEDURE *SendLoanReminders*()  
BEGIN  
 DECLARE done INT DEFAULT FALSE;  
 DECLARE loan\_id INT;  
 DECLARE cust\_name VARCHAR(100);  
 DECLARE due\_date DATE;  
 DECLARE cur CURSOR FOR  
 SELECT l.LoanID, c.CustomerName, l.DueDate  
 FROM LOANS l  
 JOIN CUSTOMERS c ON l.CustomerID = c.CustomerID  
 WHERE l.DueDate BETWEEN *CURDATE*() AND *DATE\_ADD*(*CURDATE*(), INTERVAL 30 DAY);  
 DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;  
  
 OPEN cur;  
  
 read\_loop: LOOP  
 FETCH cur INTO loan\_id, cust\_name, due\_date;  
 IF done THEN  
 LEAVE read\_loop;  
 END IF;  
  
 SELECT *CONCAT*('Reminder: LoanID ', loan\_id, ' for Customer ', cust\_name, ' is due on ', *DATE\_FORMAT*(due\_date, '%d-%b-%Y')) AS Reminder;  
  
 END LOOP;  
  
 CLOSE cur;  
END //  
  
DELIMITER ;