**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.

**Answer:**

DELIMITER //

CREATE PROCEDURE ApplyDiscountToSeniorCustomers()

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE customer\_id INT;

DECLARE current\_interest\_rate DECIMAL(5, 2);

DECLARE age INT;

DECLARE customer\_cursor CURSOR FOR

SELECT id, loan\_interest\_rate, TIMESTAMPDIFF(YEAR, date\_of\_birth, CURDATE()) AS age

FROM customers;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN customer\_cursor;

read\_loop: LOOP

FETCH customer\_cursor INTO customer\_id, current\_interest\_rate, age;

IF done THEN

LEAVE read\_loop;

END IF;

IF age > 60 THEN

SET current\_interest\_rate = current\_interest\_rate \* 0.99;

UPDATE customers

SET loan\_interest\_rate = current\_interest\_rate

WHERE id = customer\_id;

END IF;

END LOOP;

CLOSE customer\_cursor;

END //

DELIMITER ;

**Call the procedure**

CALL ApplyDiscountToSeniorCustomers();

**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.

**Answer:**

DELIMITER //

CREATE PROCEDURE PromoteToVIP()

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE customer\_id INT;

DECLARE balance DECIMAL(10, 2);

DECLARE customer\_cursor CURSOR FOR

SELECT CustomerID, Balance FROM Customers;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN customer\_cursor;

read\_loop: LOOP

FETCH customer\_cursor INTO customer\_id, balance;

IF done THEN

LEAVE read\_loop;

END IF;

IF balance > 10000 THEN

UPDATE Customers

SET IsVIP = TRUE

WHERE CustomerID = customer\_id;

END IF;

END LOOP;

CLOSE customer\_cursor;

END //

DELIMITER ;

**Call the procedure**

CALL PromoteToVIP();

**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.

**Answer:**

DELIMITER //

CREATE PROCEDURE SendLoanReminders()

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE loan\_id INT;

DECLARE customer\_id INT;

DECLARE due\_date DATE;

DECLARE loan\_cursor CURSOR FOR

SELECT LoanID, CustomerID, EndDate

FROM Loans

WHERE EndDate BETWEEN CURDATE() AND CURDATE() + INTERVAL 30 DAY;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN loan\_cursor;

read\_loop: LOOP

FETCH loan\_cursor INTO loan\_id, customer\_id, due\_date;

IF done THEN

LEAVE read\_loop;

END IF;

CALL DBMS\_OUTPUT.PUT\_LINE(CONCAT('Reminder: Customer ', customer\_id, ', your loan ', loan\_id, ' is due on ', due\_date));

END LOOP;

CLOSE loan\_cursor;

END //

DELIMITER ;

**Call the procedure**

CALL SendLoanReminders();