**Exercise 1: Control Structures**

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

**o 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.

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

**o 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.

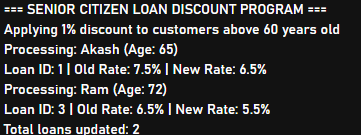
**Scenario 3:** The bank wants to send reminders to customers whose loans are due within the next 30 days.

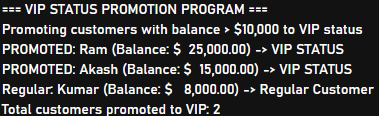
**o Question:** Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.

**Source Code :**

|  |
| --- |
| CREATE TABLE customers (  customer\_id NUMBER PRIMARY KEY,  name VARCHAR2(100),  age NUMBER,  balance NUMBER(15,2),  is\_vip VARCHAR2(5) DEFAULT 'FALSE' );  -- Create Loans table CREATE TABLE loans (  loan\_id NUMBER PRIMARY KEY,  customer\_id NUMBER,  loan\_amount NUMBER(15,2),  interest\_rate NUMBER(5,2),  due\_date DATE,  FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id) );  -- Insert sample data INSERT INTO customers VALUES (1, 'Akash', 65, 15000.00, 'FALSE'); INSERT INTO customers VALUES (2, 'Kumar', 45, 8000.00, 'FALSE'); INSERT INTO customers VALUES (3, 'Ram', 72, 25000.00, 'FALSE');   INSERT INTO loans VALUES (1, 1, 50000.00, 7.5, DATE '2025-07-15'); INSERT INTO loans VALUES (2, 2, 30000.00, 8.0, DATE '2025-07-20'); INSERT INTO loans VALUES (3, 3, 75000.00, 6.5, DATE '2025-08-10');  COMMIT;  -- SCENARIO 1: Senior Citizen Loan Discount  DECLARE  v\_customer\_id customers.customer\_id%TYPE;  v\_name customers.name%TYPE;  v\_age customers.age%TYPE;  v\_old\_rate loans.interest\_rate%TYPE;  v\_new\_rate loans.interest\_rate%TYPE;  v\_count NUMBER := 0;    CURSOR c\_senior\_customers IS  SELECT DISTINCT c.customer\_id, c.name, c.age  FROM customers c  INNER JOIN loans l ON c.customer\_id = l.customer\_id  WHERE c.age > 60;   BEGIN  DBMS\_OUTPUT.PUT\_LINE('=== SENIOR CITIZEN LOAN DISCOUNT PROGRAM ===');  DBMS\_OUTPUT.PUT\_LINE('Applying 1% discount to customers above 60 years old');  DBMS\_OUTPUT.PUT\_LINE('');    FOR rec IN c\_senior\_customers LOOP  v\_customer\_id := rec.customer\_id;  v\_name := rec.name;  v\_age := rec.age;    DBMS\_OUTPUT.PUT\_LINE('Processing: ' || v\_name || ' (Age: ' || v\_age || ')');    FOR loan\_rec IN (SELECT loan\_id, interest\_rate   FROM loans   WHERE customer\_id = v\_customer\_id) LOOP    v\_old\_rate := loan\_rec.interest\_rate;  v\_new\_rate := v\_old\_rate - 1.0;     IF v\_new\_rate < 1.0 THEN  v\_new\_rate := 1.0;  END IF;    UPDATE loans   SET interest\_rate = v\_new\_rate   WHERE loan\_id = loan\_rec.loan\_id;    DBMS\_OUTPUT.PUT\_LINE(' Loan ID: ' || loan\_rec.loan\_id ||   ' | Old Rate: ' || v\_old\_rate || '% | New Rate: ' || v\_new\_rate || '%');  v\_count := v\_count + 1;  END LOOP;    DBMS\_OUTPUT.PUT\_LINE('');  END LOOP;    DBMS\_OUTPUT.PUT\_LINE('Total loans updated: ' || v\_count);  DBMS\_OUTPUT.PUT\_LINE('');   COMMIT;   EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('Error occurred: ' || SQLERRM);  ROLLBACK; END; /  -- SCENARIO 2: VIP Status Promotion   DECLARE  v\_customer\_id customers.customer\_id%TYPE;  v\_name customers.name%TYPE;  v\_balance customers.balance%TYPE;  v\_count NUMBER := 0;    CURSOR c\_all\_customers IS  SELECT customer\_id, name, balance  FROM customers  ORDER BY balance DESC;   BEGIN  DBMS\_OUTPUT.PUT\_LINE('=== VIP STATUS PROMOTION PROGRAM ===');  DBMS\_OUTPUT.PUT\_LINE('Promoting customers with balance > $10,000 to VIP status');  DBMS\_OUTPUT.PUT\_LINE('');      FOR rec IN c\_all\_customers LOOP  v\_customer\_id := rec.customer\_id;  v\_name := rec.name;  v\_balance := rec.balance;    IF v\_balance > 10000 THEN  UPDATE customers   SET is\_vip = 'TRUE'   WHERE customer\_id = v\_customer\_id;    DBMS\_OUTPUT.PUT\_LINE('PROMOTED: ' || v\_name || ' (Balance: $' ||   TO\_CHAR(v\_balance, '999,999.99') || ') -> VIP STATUS');  v\_count := v\_count + 1;  ELSE  DBMS\_OUTPUT.PUT\_LINE('Regular: ' || v\_name || ' (Balance: $' ||   TO\_CHAR(v\_balance, '999,999.99') || ') -> Regular Customer');  END IF;  END LOOP;    DBMS\_OUTPUT.PUT\_LINE('');  DBMS\_OUTPUT.PUT\_LINE('Total customers promoted to VIP: ' || v\_count);  COMMIT;   EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('Error occurred: ' || SQLERRM);  ROLLBACK; END; /  -- SCENARIO 3: Loan Due Reminders  DECLARE  v\_loan\_id loans.loan\_id%TYPE;  v\_customer\_id loans.customer\_id%TYPE;  v\_customer\_name customers.name%TYPE;  v\_loan\_amount loans.loan\_amount%TYPE;  v\_due\_date loans.due\_date%TYPE;  v\_days\_until\_due NUMBER;  v\_count NUMBER := 0;    CURSOR c\_due\_loans IS  SELECT l.loan\_id, l.customer\_id, c.name, l.loan\_amount, l.due\_date,  (l.due\_date - SYSDATE) AS days\_until\_due  FROM loans l  INNER JOIN customers c ON l.customer\_id = c.customer\_id  WHERE l.due\_date BETWEEN SYSDATE AND SYSDATE + 30  ORDER BY l.due\_date;   BEGIN  DBMS\_OUTPUT.PUT\_LINE('=== LOAN DUE REMINDER SYSTEM ===');  DBMS\_OUTPUT.PUT\_LINE('Checking for loans due within the next 30 days...');  DBMS\_OUTPUT.PUT\_LINE('Current Date: ' || TO\_CHAR(SYSDATE, 'DD-MON-YYYY'));  DBMS\_OUTPUT.PUT\_LINE('');    FOR rec IN c\_due\_loans LOOP  v\_loan\_id := rec.loan\_id;  v\_customer\_id := rec.customer\_id;  v\_customer\_name := rec.name;  v\_loan\_amount := rec.loan\_amount;  v\_due\_date := rec.due\_date;  v\_days\_until\_due := ROUND(rec.days\_until\_due);    DBMS\_OUTPUT.PUT\_LINE(' REMINDER NOTICE #' || (v\_count + 1));  DBMS\_OUTPUT.PUT\_LINE(' Customer: ' || v\_customer\_name || ' (ID: ' || v\_customer\_id || ')');  DBMS\_OUTPUT.PUT\_LINE(' Loan ID: ' || v\_loan\_id);  DBMS\_OUTPUT.PUT\_LINE(' Amount: $' || TO\_CHAR(v\_loan\_amount, '999,999.99'));  DBMS\_OUTPUT.PUT\_LINE(' Due Date: ' || TO\_CHAR(v\_due\_date, 'DD-MON-YYYY'));    IF v\_days\_until\_due <= 7 THEN  DBMS\_OUTPUT.PUT\_LINE(' URGENT: Payment due in ' || v\_days\_until\_due || ' days!');  ELSIF v\_days\_until\_due <= 15 THEN  DBMS\_OUTPUT.PUT\_LINE(' IMPORTANT: Payment due in ' || v\_days\_until\_due || ' days');  ELSE  DBMS\_OUTPUT.PUT\_LINE(' NOTICE: Payment due in ' || v\_days\_until\_due || ' days'); END IF;    DBMS\_OUTPUT.PUT\_LINE(' Message: Dear ' || v\_customer\_name || ', your loan payment is due soon. Please ensure timely payment.');  DBMS\_OUTPUT.PUT\_LINE('');    v\_count := v\_count + 1;  END LOOP;    IF v\_count = 0 THEN  DBMS\_OUTPUT.PUT\_LINE('loans due within the next 30 days.');  ELSE  DBMS\_OUTPUT.PUT\_LINE('SUMMARY: ' || v\_count || ' reminder(s) generated for loans due within 30 days.');  END IF;   EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('Error occurred: ' || SQLERRM); END; / |

**Output :**

****

****

