|  |
| --- |
| **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.  SET SERVEROUTPUT ON;  DECLARE  v\_customer\_id NUMBER;  v\_customer\_name VARCHAR2(100);  v\_dob DATE;  v\_age NUMBER;  v\_loan\_id NUMBER;  v\_current\_rate NUMBER;  v\_new\_rate NUMBER;  v\_discount\_count NUMBER := 0;  v\_total\_customers NUMBER := 0;  v\_counter NUMBER := 1;  v\_max\_customers NUMBER;  BEGIN  SELECT COUNT(\*) INTO v\_max\_customers FROM Customers;  WHILE v\_counter <= v\_max\_customers LOOP  SELECT CustomerID, Name, DOB  INTO v\_customer\_id, v\_customer\_name, v\_dob  FROM (SELECT CustomerID, Name, DOB, ROWNUM as rn FROM Customers)  WHERE rn = v\_counter;  v\_total\_customers := v\_total\_customers + 1;  v\_age := FLOOR(MONTHS\_BETWEEN(SYSDATE, v\_dob) / 12);  DBMS\_OUTPUT.PUT\_LINE('Customer: ' || v\_customer\_name || ' (Age: ' || v\_age || ')');  IF v\_age > 60 THEN  FOR loan\_rec IN (SELECT LoanID, InterestRate FROM Loans WHERE CustomerID = v\_customer\_id) LOOP  v\_loan\_id := loan\_rec.LoanID;  v\_current\_rate := loan\_rec.InterestRate;  DBMS\_OUTPUT.PUT\_LINE(' Loan ID: ' || v\_loan\_id || ', Current Rate: ' || v\_current\_rate || '%');  v\_new\_rate := v\_current\_rate - 1;  IF v\_new\_rate < 0 THEN  v\_new\_rate := 0;  END IF;  UPDATE Loans SET InterestRate = v\_new\_rate WHERE LoanID = v\_loan\_id;  v\_discount\_count := v\_discount\_count + 1;  DBMS\_OUTPUT.PUT\_LINE(' ✓ Discount Applied! New Rate: ' || v\_new\_rate || '%');  END LOOP;  ELSE  DBMS\_OUTPUT.PUT\_LINE(' - No discount (Age <= 60)');  END IF;    v\_counter := v\_counter + 1;  END LOOP;    DBMS\_OUTPUT.PUT\_LINE('Total customers processed: ' || v\_total\_customers);  DBMS\_OUTPUT.PUT\_LINE('Total discounts applied: ' || v\_discount\_count);    COMMIT;  END;  /  **OUTPUT:** |

**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.

SET SERVEROUTPUT ON;

DECLARE

    v\_customer\_id NUMBER;

    v\_customer\_name VARCHAR2(100);

    v\_balance NUMBER;

    v\_vip\_count NUMBER := 0;

    v\_total\_customers NUMBER := 0;

    v\_counter NUMBER := 1;

    v\_max\_customers NUMBER;

BEGIN

    DBMS\_OUTPUT.PUT\_LINE('VIP Threshold: $10,000');

    SELECT COUNT(\*) INTO v\_max\_customers FROM Customers;

    WHILE v\_counter <= v\_max\_customers LOOP

        SELECT CustomerID, Name, Balance

        INTO v\_customer\_id, v\_customer\_name, v\_balance

        FROM (SELECT CustomerID, Name, Balance, ROWNUM as rn FROM Customers)

        WHERE rn = v\_counter;

        v\_total\_customers := v\_total\_customers + 1;

        DBMS\_OUTPUT.PUT\_LINE('Customer: ' || v\_customer\_name ||

                           ' (Balance: $' || TO\_CHAR(v\_balance, '999,999.99') || ')');

        IF v\_balance > 10000 THEN

            UPDATE Customers SET IsVIP = 'Y' WHERE CustomerID = v\_customer\_id;

            v\_vip\_count := v\_vip\_count + 1;

            DBMS\_OUTPUT.PUT\_LINE('  ✓ VIP STATUS GRANTED');

        ELSE

            UPDATE Customers SET IsVIP = 'N' WHERE CustomerID = v\_customer\_id;

            DBMS\_OUTPUT.PUT\_LINE('  - Regular Customer');

        END IF;

        v\_counter := v\_counter + 1;

    END LOOP;

    DBMS\_OUTPUT.PUT\_LINE('Total customers processed: ' || v\_total\_customers);

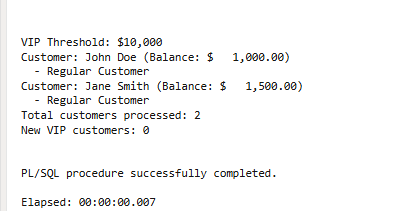
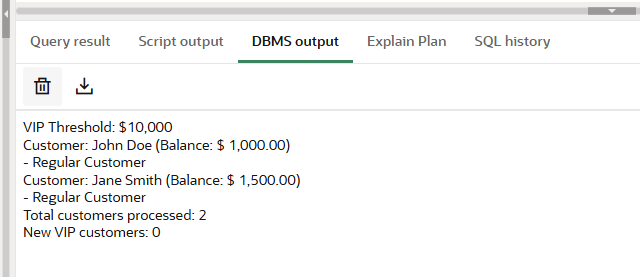
    DBMS\_OUTPUT.PUT\_LINE('New VIP customers: ' || v\_vip\_count);

    COMMIT;

END;

/

**OUTPUT:**



**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.

SET SERVEROUTPUT ON;

DECLARE

v\_loan\_id NUMBER;

v\_customer\_id NUMBER;

v\_customer\_name VARCHAR2(100);

v\_loan\_amount NUMBER;

v\_end\_date DATE;

v\_days\_until\_due NUMBER;

v\_reminder\_count NUMBER := 0;

v\_counter NUMBER := 1;

v\_max\_loans NUMBER;

v\_urgency VARCHAR2(10);

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Checking loans due within 30 days...');

SELECT COUNT(\*) INTO v\_max\_loans

FROM Loans

WHERE EndDate BETWEEN SYSDATE AND (SYSDATE + 30);

IF v\_max\_loans = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('No loans are due within the next 30 days.');

ELSE

FOR loan\_rec IN (SELECT l.LoanID, l.CustomerID, c.Name, l.LoanAmount, l.EndDate

FROM Loans l, Customers c

WHERE l.CustomerID = c.CustomerID

AND l.EndDate BETWEEN SYSDATE AND (SYSDATE + 30)

ORDER BY l.EndDate) LOOP

v\_loan\_id := loan\_rec.LoanID;

v\_customer\_id := loan\_rec.CustomerID;

v\_customer\_name := loan\_rec.Name;

v\_loan\_amount := loan\_rec.LoanAmount;

v\_end\_date := loan\_rec.EndDate;

v\_days\_until\_due := ROUND(v\_end\_date - SYSDATE);

v\_reminder\_count := v\_reminder\_count + 1;

IF v\_days\_until\_due <= 7 THEN

v\_urgency := 'URGENT';

ELSIF v\_days\_until\_due <= 15 THEN

v\_urgency := 'HIGH';

ELSE

v\_urgency := 'MEDIUM';

END IF;

DBMS\_OUTPUT.PUT\_LINE('>>> REMINDER [' || v\_urgency || '] <<<');

DBMS\_OUTPUT.PUT\_LINE('Dear ' || v\_customer\_name || ',');

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\_end\_date, 'DD-MON-YYYY'));

DBMS\_OUTPUT.PUT\_LINE('Days Remaining: ' || v\_days\_until\_due);

IF v\_urgency = 'URGENT' THEN

DBMS\_OUTPUT.PUT\_LINE(' URGENT: Contact us immediately!');

ELSIF v\_urgency = 'HIGH' THEN

DBMS\_OUTPUT.PUT\_LINE(' HIGH PRIORITY: Arrange payment soon.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('REMINDER: Prepare for payment.');

END IF;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Total reminders sent: ' || v\_reminder\_count);

END IF;

END;

/

**OUTPUT:**

