PLSQL

**Exercise 1: Control Structures**

# SET SERVEROUTPUT ON;

-- Create temporary tables for demonstration BEGIN

EXECUTE IMMEDIATE 'DROP TABLE demo\_customers PURGE'; EXECUTE IMMEDIATE 'DROP TABLE demo\_loans PURGE';

# EXCEPTION

WHEN OTHERS THEN NULL; END;

/

-- Create demo customers table CREATE TABLE demo\_customers (

customer\_id NUMBER PRIMARY KEY, first\_name VARCHAR2(50),

last\_name VARCHAR2(50), birth\_date DATE, account\_balance NUMBER,

is\_vip VARCHAR2(5) DEFAULT 'FALSE', email VARCHAR2(100)

);

-- Create demo loans table CREATE TABLE demo\_loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER, interest\_rate NUMBER, due\_date DATE,

status VARCHAR2(20),

CONSTRAINT fk\_customer FOREIGN KEY (customer\_id) REFERENCES demo\_customers(customer\_id)

);

-- Insert sample data

INSERT INTO demo\_customers VALUES (1, 'John', 'Doe', TO\_DATE('1950- 05-15', 'YYYY-MM-DD'), 8500, 'FALSE', ['john.doe@email.com');](mailto:%27john.doe@email.com)

INSERT INTO demo\_customers VALUES (2, 'Jane', 'Smith', TO\_DATE('1975- 08-20', 'YYYY-MM-DD'), 12000, 'FALSE', ['jane.smith@email.com');](mailto:%27jane.smith@email.com)

INSERT INTO demo\_customers VALUES (3, 'Robert', 'Johnson', TO\_DATE('1948-11-30', 'YYYY-MM-DD'), 5000, 'FALSE',

['robert.j@email.com');](mailto:%27robert.j@email.com)

INSERT INTO demo\_customers VALUES (4, 'Emily', 'Davis', TO\_DATE('1985-03-10', 'YYYY-MM-DD'), 15000, 'FALSE',

['emily.d@email.com');](mailto:%27emily.d@email.com)

INSERT INTO demo\_loans VALUES (101, 1, 5.5, SYSDATE + 15, 'ACTIVE');

INSERT INTO demo\_loans VALUES (102, 2, 4.8, SYSDATE + 45, 'ACTIVE');

INSERT INTO demo\_loans VALUES (103, 3, 6.2, SYSDATE + 10, 'ACTIVE');

INSERT INTO demo\_loans VALUES (104, 4, 5.0, SYSDATE + 60, 'ACTIVE');

# COMMIT;

-- Scenario 1: Apply discount for senior customers DECLARE

v\_discount\_rate NUMBER := 1.0; -- 1% discount v\_count\_seniors NUMBER := 0;

# BEGIN

DBMS\_OUTPUT.PUT\_LINE('=== SCENARIO 1: Applying Senior

Discounts ===');

FOR c IN (SELECT c.customer\_id, c.birth\_date, l.loan\_id, l.interest\_rate FROM demo\_customers c

JOIN demo\_loans l ON c.customer\_id = l.customer\_id

WHERE MONTHS\_BETWEEN(SYSDATE, c.birth\_date)/12 > 60)

# LOOP

UPDATE demo\_loans

SET interest\_rate = interest\_rate - v\_discount\_rate WHERE loan\_id = c.loan\_id;

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

DBMS\_OUTPUT.PUT\_LINE('Applied 1% discount to customer ' || c.customer\_id ||

'. Old rate: ' || c.interest\_rate || '%, New rate: ' || (c.interest\_rate

- v\_discount\_rate) || '%'); END LOOP;

IF v\_count\_seniors = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('No senior customers found for discount.'); ELSE

DBMS\_OUTPUT.PUT\_LINE('Applied discounts to ' || v\_count\_seniors || ' senior customers.');

# END IF;

END;

/

-- Scenario 2: Promote customers to VIP based on balance DECLARE

v\_vip\_threshold NUMBER := 10000; v\_count\_vips NUMBER := 0;

# BEGIN

DBMS\_OUTPUT.PUT\_LINE(CHR(10) || '=== SCENARIO 2: VIP

Promotions ===');

FOR c IN (SELECT customer\_id, first\_name, last\_name, account\_balance FROM demo\_customers

WHERE account\_balance > v\_vip\_threshold AND is\_vip = 'FALSE')

# LOOP

UPDATE demo\_customers SET is\_vip = 'TRUE'

WHERE customer\_id = c.customer\_id;

v\_count\_vips := v\_count\_vips + 1; DBMS\_OUTPUT.PUT\_LINE('Promoted ' || c.first\_name || ' ' || c.last\_name

||

' (ID: ' || c.customer\_id || ') to VIP. Balance: $' ||

c.account\_balance); END LOOP;

IF v\_count\_vips = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('No customers qualified for VIP promotion.');

# ELSE

DBMS\_OUTPUT.PUT\_LINE('Promoted ' || v\_count\_vips || ' customers to VIP status.');

# END IF; END;

/

-- Scenario 3: Loan due reminders DECLARE

v\_count\_reminders NUMBER := 0; BEGIN

DBMS\_OUTPUT.PUT\_LINE(CHR(10) || '=== SCENARIO 3: Loan Due

Reminders ===');

FOR l IN (SELECT l.loan\_id, l.due\_date, c.customer\_id, c.first\_name, c.last\_name, c.email

FROM demo\_loans l

JOIN demo\_customers c ON l.customer\_id = c.customer\_id WHERE l.due\_date BETWEEN SYSDATE AND SYSDATE + 30 AND l.status = 'ACTIVE') LOOP

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

DBMS\_OUTPUT.PUT\_LINE('REMINDER: ' || l.first\_name || ' ' || l.last\_name ||

' (Loan ID: ' || l.loan\_id || ') has payment due on ' || TO\_CHAR(l.due\_date, 'YYYY-MM-DD') || ' (' || ROUND(l.due\_date - SYSDATE) || ' days). Email: ' || l.email);

# END LOOP;

IF v\_count\_reminders = 0 THEN

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

DBMS\_OUTPUT.PUT\_LINE('Generated ' || v\_count\_reminders || ' payment reminders.');

# END IF; END;

/

-- Display final results BEGIN

# DBMS\_OUTPUT.PUT\_LINE(CHR(10) || '=== FINAL RESULTS ===');

DBMS\_OUTPUT.PUT\_LINE(CHR(10) || 'Customer Status:'); FOR c IN (SELECT customer\_id, first\_name, last\_name, is\_vip,

account\_balance

FROM demo\_customers ORDER BY customer\_id) LOOP DBMS\_OUTPUT.PUT\_LINE(c.customer\_id || ': ' || c.first\_name || ' ' ||

c.last\_name ||

' | VIP: ' || c.is\_vip || ' | Balance: $' || c.account\_balance);

# END LOOP;

DBMS\_OUTPUT.PUT\_LINE(CHR(10) || 'Loan Status:');

FOR l IN (SELECT l.loan\_id, l.customer\_id, c.first\_name, c.last\_name, l.interest\_rate, l.due\_date

FROM demo\_loans l

JOIN demo\_customers c ON l.customer\_id = c.customer\_id

ORDER BY l.loan\_id) LOOP

DBMS\_OUTPUT.PUT\_LINE('Loan ' || l.loan\_id || ': ' || l.first\_name || ' ' || l.last\_name ||

# END LOOP; END;

/

**OUTPUT:**

' | Rate: ' || l.interest\_rate || '% | Due: ' || TO\_CHAR(l.due\_date, 'YYYY-MM-DD') || ' (' || ROUND(l.due\_date - SYSDATE) || ' days)');







