**PL/SQL Control Structures for Banking Scenarios**

**Scenario 1: Senior Citizen Interest Rate Discount**

DECLARE

CURSOR customer\_loans IS

SELECT c.customer\_id, l.loan\_id, l.interest\_rate,

FLOOR(MONTHS\_BETWEEN(SYSDATE, c.date\_of\_birth)/12) AS age

FROM customers c

JOIN loans l ON c.customer\_id = l.customer\_id;

v\_discount\_rate NUMBER := 0.01; -- 1% discount

BEGIN

FOR cust\_loan IN customer\_loans LOOP

IF cust\_loan.age > 60 THEN

-- Apply discount and update

UPDATE loans

SET interest\_rate = interest\_rate - v\_discount\_rate

WHERE loan\_id = cust\_loan.loan\_id;

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

cust\_loan.customer\_id ||

' (Age: ' || cust\_loan.age || ')');

END IF;

END LOOP;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Senior discount processing complete');

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

/

**Scenario 2: VIP Customer Promotion**

DECLARE

CURSOR customer\_balances IS

SELECT customer\_id, account\_balance

FROM customers;

v\_vip\_threshold NUMBER := 10000; -- $10,000 threshold

v\_count NUMBER := 0;

BEGIN

FOR cust IN customer\_balances LOOP

IF cust.account\_balance > v\_vip\_threshold THEN

-- Update VIP status

UPDATE customers

SET is\_vip = 'Y' -- Assuming Y/N flag

WHERE customer\_id = cust.customer\_id;

v\_count := v\_count + 1;

DBMS\_OUTPUT.PUT\_LINE('Promoted customer ' ||

cust.customer\_id ||

' to VIP status');

END IF;

END LOOP;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('VIP promotion complete. ' ||

v\_count || ' customers promoted');

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

/

**Scenario 3: Loan Payment Reminders**

DECLARE

CURSOR due\_loans IS

SELECT l.loan\_id, c.customer\_id, c.customer\_name, c.email,

l.due\_date, l.amount\_due

FROM loans l

JOIN customers c ON l.customer\_id = c.customer\_id

WHERE l.due\_date BETWEEN SYSDATE AND SYSDATE + 30

AND l.status = 'ACTIVE';

v\_reminder\_text VARCHAR2(500);

v\_count NUMBER := 0;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Sending loan payment reminders...');

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

FOR loan IN due\_loans LOOP

v\_reminder\_text := 'Dear ' || loan.customer\_name ||

', your loan payment of $' || loan.amount\_due ||

' is due on ' || TO\_CHAR(loan.due\_date, 'DD-MON-YYYY') ||

'. Please make payment to avoid late fees.';

-- In real implementation, you would call email API here

DBMS\_OUTPUT.PUT\_LINE('To: ' || loan.email);

DBMS\_OUTPUT.PUT\_LINE('Subject: Loan Payment Reminder');

DBMS\_OUTPUT.PUT\_LINE(v\_reminder\_text);

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

v\_count := v\_count + 1;

-- Optional: Record reminder in audit table

INSERT INTO loan\_reminders (loan\_id, reminder\_date, reminder\_text)

VALUES (loan.loan\_id, SYSDATE, v\_reminder\_text);

END LOOP;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Reminders processed: ' || v\_count);

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

/