# WEEK 2:

## PL/SQL PROGRAMMING

### EXERCISE 1: CONTROL STRUCTURES

**Scenario 1:**BEGIN

FOR cust IN (

SELECT customer\_id, age, loan\_id, interest\_rate

FROM customers c

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

)

LOOP

IF cust.age > 60 THEN

UPDATE loans

SET interest\_rate = interest\_rate - 1

WHERE loan\_id = cust.loan\_id;

END IF;

END LOOP;

COMMIT;

END;

**Scenario 2:**BEGIN

FOR cust IN (

SELECT customer\_id, balance

FROM customers

WHERE balance > 10000

)

LOOP

UPDATE customers

SET isvip = 'TRUE'

WHERE customer\_id = cust.customer\_id;

END LOOP;

COMMIT;

END;

**Scenario 3:**BEGIN

FOR due\_loan IN (

SELECT c.customer\_id, c.name, l.due\_date

FROM loans l

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

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

)

LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Dear ' || due\_loan.name ||

', your loan is due on ' || TO\_CHAR(due\_loan.due\_date, 'DD-Mon-YYYY'));

END LOOP;

END;

### EXERCISE 3: STORED PROCEDURES

**Scenario 1:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

FOR acc IN (

SELECT account\_id, balance

FROM accounts

WHERE account\_type = 'SAVINGS'

)

LOOP

UPDATE accounts

SET balance = balance + (balance \* 0.01)

WHERE account\_id = acc.account\_id;

END LOOP;

COMMIT;

END;

**Scenario 2:**CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_department\_id IN NUMBER,

p\_bonus\_percent IN NUMBER

) AS

BEGIN

UPDATE employees

SET salary = salary + (salary \* p\_bonus\_percent / 100)

WHERE department\_id = p\_department\_id;

COMMIT;

END;  
  
**Scenario 3:**CREATE OR REPLACE PROCEDURE TransferFunds (

p\_from\_account\_id IN NUMBER,

p\_to\_account\_id IN NUMBER,

p\_amount IN NUMBER

) AS

v\_balance NUMBER;

BEGIN

-- Check balance of source account

SELECT balance INTO v\_balance

FROM accounts

WHERE account\_id = p\_from\_account\_id

FOR UPDATE;

IF v\_balance < p\_amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds in the source account');

END IF;

-- Deduct from source account

UPDATE accounts

SET balance = balance - p\_amount

WHERE account\_id = p\_from\_account\_id;

-- Add to destination account

UPDATE accounts

SET balance = balance + p\_amount

WHERE account\_id = p\_to\_account\_id;

COMMIT;

END;