**Exercise 3: Stored Procedures**

**Scenario 1:** The bank needs to process monthly interest for all savings accounts.

* + **Question:** Write a stored procedure **ProcessMonthlyInterest** that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.

**Scenario 2:** The bank wants to implement a bonus scheme for employees based on their performance.

* + **Question:** Write a stored procedure **UpdateEmployeeBonus** that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.

**Scenario 3:** Customers should be able to transfer funds between their accounts.

* + **Question:** Write a stored procedure **TransferFunds** that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

**Code:** **Scenario 1**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest

IS

BEGIN

UPDATE accounts set balance=balance\*1.01

Where accounttype=’SAVINGS’;

END;

/

**Code:** **Scenario 2**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(DEPART VARCHAR2, BONUS\_PERCENTAGE NUMBER)

IS

BEGIN

UPDATE EMPLOYEE

SET SALARY = SALARY \* (1 + BONUS\_PERCENTAGE / 100)

WHERE DEPARTMENT = DEPART;

COMMIT;

END;

/

SQL> BEGIN

2 UpdateEmployeeBonus('HR', 10);

3 END;

4 /

PL/SQL procedure successfully completed.

**Code:** **Scenario 3**

SQL> select \* from accounts;

ACCOUNTID CUSTOMERID ACCOUNTTYPE BALANCE LASTMODIF

---------- ---------- -------------------- ---------- ---------

1 1 Savings 1000 04-AUG-24

2 2 Checking 1500 04-AUG-24

SQL> CREATE OR REPLACE PROCEDURE TransferFunds (src\_accid NUMBER, dest\_accid NUMBER, amt NUMBER)

IS

total\_balance NUMBER;

BEGIN

SELECT balance INTO total\_balance FROM accounts WHERE accountid = src\_accid;

IF total\_balance < amt THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds');

END IF;

UPDATE accounts SET balance = balance - amt WHERE accountid = src\_accid;

UPDATE accounts SET balance = balance + amt WHERE accountid = dest\_accid;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('ERROR DURING FUNDS TRANSFER: ' || SQLERRM);

END;

/

Procedure created.

SQL> set serverout on;

SQL> BEGIN

TransferFunds(1, 2, 100);

END;

/

PL/SQL procedure successfully completed.

SQL> SELECT \* FROM accounts;

ACCOUNTID CUSTOMERID ACCOUNTTYPE BALANCE LASTMODIF

---------- --------- -------------------- ---------- ---------

1 1 Savings 900 04-AUG-24

2 2 Checking 1600 04-AUG-24