

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.

```
CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS
BEGIN
    UPDATE Accounts
    SET Balance = Balance * 1.01
    WHERE AccountType = 'Savings';

    COMMIT;

    DBMS_OUTPUT.PUT_LINE('Monthly interest applied to all savings accounts.');
```

END ProcessMonthlyInterest;

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.

```
CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (  
    p_department IN VARCHAR2,  
    p_bonus_percentage IN NUMBER  
) AS  
BEGIN  
    UPDATE Employees  
    SET Salary = Salary * (1 + p_bonus_percentage / 100)  
    WHERE Department = p_department;  
  
    COMMIT;  
  
    DBMS_OUTPUT.PUT_LINE('Bonus applied to all employees in department: ' ||  
p_department);  
END UpdateEmployeeBonus;
```

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.

```
CREATE OR REPLACE PROCEDURE TransferFunds (  
    p_from_account IN NUMBER,  
    p_to_account IN NUMBER,  
    p_amount IN NUMBER  
) AS  
    v_balance NUMBER;  
BEGIN  
    SELECT Balance INTO v_balance  
    FROM Accounts  
    WHERE AccountID = p_from_account;  
  
    IF v_balance < p_amount THEN  
        RAISE_APPLICATION_ERROR(-20001, 'Insufficient funds in account ' ||  
p_from_account);  
    END IF;  
  
    BEGIN  
        UPDATE Accounts  
        SET Balance = Balance - p_amount  
        WHERE AccountID = p_from_account;  
  
        UPDATE Accounts  
        SET Balance = Balance + p_amount  
        WHERE AccountID = p_to_account;  
  
        COMMIT;  
  
        DBMS_OUTPUT.PUT_LINE('Transfer of ' || p_amount || ' from account  
' || p_from_account || ' to account ' || p_to_account || ' completed  
successfully.');
```

```
    EXCEPTION  
        WHEN OTHERS THEN  
            ROLLBACK;  
            DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);  
    END;  
END TransferFunds;
```