**QUESTION:**

**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.

**SCENARIO 1:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

    FOR rec IN (SELECT AccountID, Balance

                FROM Accounts

                WHERE AccountType = 'Savings')

    LOOP

        UPDATE Accounts

        SET Balance = Balance + (rec.Balance \* 0.01)

        WHERE AccountID = rec.AccountID;

    END LOOP;

    COMMIT;

END;

/

A screenshot of a computer screen

AI-generated content may be incorrect.

**SCENARIO 2:**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

    p\_department IN Employees.Department%TYPE,

    p\_bonus\_percent IN NUMBER

) AS

BEGIN

    UPDATE Employees

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

    WHERE Department = p\_department;

    COMMIT;

END;

/

A screen shot of a computer

AI-generated content may be incorrect.

**SCENARIO 3:**

CREATE OR REPLACE PROCEDURE TransferFunds (

    p\_from\_account IN Accounts.AccountID%TYPE,

    p\_to\_account IN Accounts.AccountID%TYPE,

    p\_amount IN NUMBER

) AS

    v\_balance NUMBER;

BEGIN

    -- Check balance in source account

    SELECT Balance INTO v\_balance FROM Accounts WHERE AccountID = p\_from\_account;

    IF v\_balance < p\_amount THEN

        RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient balance in source account.');

    END IF;

    -- Deduct from source

    UPDATE Accounts

    SET Balance = Balance - p\_amount

    WHERE AccountID = p\_from\_account;

    -- Add to destination

    UPDATE Accounts

    SET Balance = Balance + p\_amount

    WHERE AccountID = p\_to\_account;

    COMMIT;

END;

/

A screenshot of a computer screen

AI-generated content may be incorrect.