# Exercise 3: Stored Procedures

## Scenario 1: Monthly Interest for 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.

Sample Table:

CREATE TABLE SavingsAccounts (  
 AccountID NUMBER PRIMARY KEY,  
 CustomerName VARCHAR2(50),  
 Balance NUMBER(10, 2)  
);

Sample Data:

INSERT INTO SavingsAccounts VALUES (1, 'Alice', 10000);  
INSERT INTO SavingsAccounts VALUES (2, 'Bob', 8000);  
INSERT INTO SavingsAccounts VALUES (3, 'Carol', 15000);  
COMMIT;

Stored Procedure:

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS  
BEGIN  
 FOR acc IN (SELECT AccountID FROM SavingsAccounts) LOOP  
 UPDATE SavingsAccounts  
 SET Balance = Balance \* 1.01  
 WHERE AccountID = acc.AccountID;  
 END LOOP;  
 COMMIT;  
END;

## Scenario 2: Employee Bonus Scheme

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.

Sample Table:

CREATE TABLE Employees (  
 EmpID NUMBER PRIMARY KEY,  
 EmpName VARCHAR2(50),  
 DepartmentID NUMBER,  
 Salary NUMBER(10, 2)  
);

Sample Data:

INSERT INTO Employees VALUES (101, 'John', 10, 40000);  
INSERT INTO Employees VALUES (102, 'Jane', 20, 45000);  
INSERT INTO Employees VALUES (103, 'Mike', 10, 42000);  
COMMIT;

Stored Procedure:

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(p\_dept\_id IN NUMBER, p\_bonus\_pct IN NUMBER) AS  
BEGIN  
 UPDATE Employees  
 SET Salary = Salary + (Salary \* p\_bonus\_pct / 100)  
 WHERE DepartmentID = p\_dept\_id;  
 COMMIT;  
END;

## Scenario 3: Fund Transfer Between 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.

Sample Table:

CREATE TABLE BankAccounts (  
 AccountNumber NUMBER PRIMARY KEY,  
 HolderName VARCHAR2(50),  
 Balance NUMBER(10, 2)  
);

Sample Data:

INSERT INTO BankAccounts VALUES (1001, 'Alice', 5000);  
INSERT INTO BankAccounts VALUES (1002, 'Bob', 3000);  
COMMIT;

Stored Procedure:

CREATE OR REPLACE PROCEDURE TransferFunds(p\_from NUMBER, p\_to NUMBER, p\_amount NUMBER) AS  
 v\_balance NUMBER;  
BEGIN  
 SELECT Balance INTO v\_balance FROM BankAccounts WHERE AccountNumber = p\_from;  
  
 IF v\_balance >= p\_amount THEN  
 UPDATE BankAccounts SET Balance = Balance - p\_amount WHERE AccountNumber = p\_from;  
 UPDATE BankAccounts SET Balance = Balance + p\_amount WHERE AccountNumber = p\_to;  
 COMMIT;  
 ELSE  
 RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient balance for transfer.');  
 END IF;  
END;