**Exercise 3: Stored Procedures**

**Scenario 1:**

**CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS**

**BEGIN**

**UPDATE savings\_accounts**

**SET balance = balance \* 1.01;**

**COMMIT;**

**END ProcessMonthlyInterest;**

**/**

**Scenario 2:**

**CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (**

**p\_department\_id IN NUMBER,**

**p\_bonus\_percentage IN NUMBER**

**) IS**

**BEGIN**

**UPDATE employees**

**SET salary = salary \* (1 + p\_bonus\_percentage / 100)**

**WHERE department\_id = p\_department\_id;**

**COMMIT;**

**END UpdateEmployeeBonus;**

**/**

**Scenario 3:**

**CREATE OR REPLACE PROCEDURE TransferFunds (**

**p\_from\_account\_id IN NUMBER,**

**p\_to\_account\_id IN NUMBER,**

**p\_amount IN NUMBER**

**) IS**

**insufficient\_funds EXCEPTION;**

**insufficient\_funds\_message VARCHAR2(100);**

**BEGIN**

**DECLARE**

**from\_balance NUMBER;**

**BEGIN**

**SELECT balance INTO from\_balance FROM accounts WHERE account\_id = p\_from\_account\_id;**

**IF from\_balance < p\_amount THEN**

**insufficient\_funds\_message := 'Insufficient funds in account ' || p\_from\_account\_id;**

**RAISE insufficient\_funds;**

**END IF;**

**END;**

**UPDATE accounts SET balance = balance - p\_amount WHERE account\_id = p\_from\_account\_id;**

**UPDATE accounts SET balance = balance + p\_amount WHERE account\_id = p\_to\_account\_id;**

**COMMIT;**

**EXCEPTION**

**WHEN insufficient\_funds THEN**

**ROLLBACK;**

**DBMS\_OUTPUT.PUT\_LINE(insufficient\_funds\_message);**

**WHEN OTHERS THEN**

**ROLLBACK;**

**DBMS\_OUTPUT.PUT\_LINE('An error occurred during fund transfer: ' || SQLERRM);**

**END TransferFunds;**

**/**