

## Exercise 3: Stored Procedures

---

### Scenario 1: Process Monthly Interest

**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.

---

**\*Assumed Table:**

Accounts(account\_id NUMBER, balance NUMBER, account\_type VARCHAR2(20))

---

**Procedure Code:**

```
CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS
BEGIN
    UPDATE Accounts
    SET balance = balance + (balance * 0.01)
    WHERE account_type = 'Savings';

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

END;

/

**To Run:**

```
BEGIN
    ProcessMonthlyInterest;
END;
```

/

---

**Expected Output:**

Monthly interest applied to all savings accounts.

---

**Scenario 2: Update Employee Bonus**

**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.

---

**\*Assumed Table:**

Employees(emp\_id NUMBER, department\_id NUMBER, salary NUMBER)

---

**Procedure Code:**

```
CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(
    dept_id IN NUMBER,
    bonus_percent IN NUMBER
) AS
BEGIN
    UPDATE Employees
    SET salary = salary + (salary * bonus_percent / 100)
    WHERE department_id = dept_id;

    DBMS_OUTPUT.PUT_LINE('Bonus applied to department ID: ' || dept_id);
END;
/
```

**To Run:**

```
BEGIN
    UpdateEmployeeBonus(101, 10); -- 10% bonus for dept 101
```

END;  
/

---

### **Expected Output:**

Bonus applied to department ID: 101

---

### **Scenario 3: Transfer Funds Between Accounts**

**Question:** Write a stored procedure `TransferFunds` that transfers a specified amount from one account to another, checking that the source account has enough balance.

---

### **\*Assumed Table:**

`Accounts(account_id NUMBER, balance NUMBER)`

---

### **Procedure Code:**

```
CREATE OR REPLACE PROCEDURE TransferFunds(  
    from_acc IN NUMBER,  
    to_acc IN NUMBER,  
    amt IN NUMBER  
) AS  
    insufficient_balance EXCEPTION;  
    v_balance NUMBER;  
BEGIN  
    SELECT balance INTO v_balance FROM Accounts WHERE account_id =  
    from_acc;  
  
    IF v_balance < amt THEN  
        RAISE insufficient_balance;  
    END IF;
```

```
UPDATE Accounts
SET balance = balance - amt
WHERE account_id = from_acc;
```

```
UPDATE Accounts
SET balance = balance + amt
WHERE account_id = to_acc;
```

```
DBMS_OUTPUT.PUT_LINE('Funds transferred from ' || from_acc || ' to ' ||
to_acc);
EXCEPTION
  WHEN insufficient_balance THEN
    DBMS_OUTPUT.PUT_LINE('Error: Not enough balance in account ' ||
from_acc);
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Unexpected error occurred.');
```

END;  
/

### **To Run:**

```
BEGIN
  TransferFunds(1001, 1002, 500);
END;
/
```

---

### **Expected Output (If balance is enough):**

Funds transferred from 1001 to 1002

### **Expected Output (If balance is not enough):**

Error: Not enough balance in account 1001

---