

- **Scenario 1**

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

 FOR acc IN (

 SELECT AccountID, Balance

 FROM Accounts

 WHERE AccountType = 'Savings'

)

 LOOP

 UPDATE Accounts

 SET

 Balance = acc.Balance + (acc.Balance * 0.01),

 LastModified = SYSDATE

 WHERE AccountID = acc.AccountID;

 END LOOP;

END;

/

- **Scenario 2**

```
CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (  
    p_Department IN Employees.Department%TYPE,  
    p_BonusPercent IN NUMBER  
) AS  
BEGIN  
    UPDATE Employees  
    SET Salary = Salary+(Salary*p_BonusPercent/100)  
    WHERE Department = p_Department;  
  
    DBMS_OUTPUT.PUT_LINE('Bonus of ' || p_BonusPercent || '% applied to department: ' || p_Department);  
  
END;  
  
/  
  
SET SERVEROUTPUT ON;  
  
BEGIN  
    UpdateEmployeeBonus('Sales', 10);  
END;  
  
/  
  
SET SERVEROUTPUT ON;  
  
BEGIN  
    UpdateEmployeeBonus('Sales', 20);  
END;  
  
/
```

- **Scenario 3**

```
CREATE SEQUENCE Transactions_seq
```

```
START WITH 1
```

```
INCREMENT BY 1
```

```
NOCACHE
```

```
NOCYCLE;
```

```
CREATE OR REPLACE PROCEDURE TransferFunds (
```

```
    p_FromAccountID IN NUMBER,
```

```
    p_ToAccountID IN NUMBER,
```

```
    p_Amount IN NUMBER
```

```
)
```

```
IS
```

```
    v_FromBalance NUMBER;
```

```
    v_Now DATE := SYSDATE;
```

```
BEGIN
```

```
    SELECT Balance INTO v_FromBalance
```

```
    FROM Accounts
```

```
    WHERE AccountsID = p_fromAccountID
```

```
    FOR UPDATE;
```

```
    IF v_FromBalance < p_Amount THEN
```

```
        RAISE_APPLICATION_ERROR(-20001, 'Insufficient balance in source account.');
```

```
    END IF;
```

```
    UPDATE Accounts
```

```
    SET Balance = Balance - p_Amount,
```

```
        LastModified = v_NOW
```

```
    WHERE AccountID = p_FromAccountID;
```

```
UPDATE Accounts
SET Balance = Balance + p_Amount,
    LastModified = v_Now
WHERE AccountID = p_ToAccountID;
```

```
INSERT INTO Transactions(
    TransactionID,
    AccountID,
    TransactionDate,
    Amount,
    TransactionType
) VALUES (
    Transactions_seq.NEXTVAL,
    p_ToAccountID,
    v_Now,
    -p_Amount,
    'DEBIT'
);
```

```
INSERT INTO Transactions (
    TransactionID,
    AccountID,
    TransactionDate,
    Amount,
    TransactionType
) VALUES (
    Transactions_seq.NEXTVAL,
    p_ToAccountID,
    v_Now,
    p_Amount,
    'CREDIT'
);
```

```
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END;
/
```

@Usage

```
BEGIN
    TransferFunds(1001, 1003, 2000);
END;
```

```
SELECT AccountID, CustomerID, AccountType, Balance, LastModified
FROM Accounts
WHERE AccountID IN (1001, 1003);
```

```
SELECT TransactionID, AccountID, TransactionDate, Amount, TransactionType
FROM Transactions
WHERE AccountID IN (1001, 1003)
ORDER BY TransactionDate DESC;
```