- **Scenario 1**

```
CREATE OR REPLACE PROCEDURE SafeTransferFunds (
    p_from_account_id IN NUMBER,
    p_to_account_id   IN NUMBER,
    p_amount          IN NUMBER
)
IS
    insufficient_funds EXCEPTION;
    v_from_balance NUMBER;
BEGIN
    SELECT Balance INTO v_from_balance FROM Accounts WHERE AccountID = p_from_account_id FOR UPDATE;

    IF v_from_balance < p_amount THEN
        RAISE insufficient_funds;
    END IF;

    -- Deduct from sender
    UPDATE Accounts
    SET Balance = Balance - p_amount,
        LastModified = SYSDATE
    WHERE AccountID = p_from_account_id;

    -- Add to receiver
    UPDATE Accounts
    SET Balance = Balance + p_amount,
        LastModified = SYSDATE
    WHERE AccountID = p_to_account_id;

    INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)
    VALUES (Transactions_seq.NEXTVAL, p_from_account_id, SYSDATE, p_amount, 'Transfer-OUT');
    INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)
    VALUES (Transactions_seq.NEXTVAL, p_to_account_id, SYSDATE, p_amount, 'Transfer-IN');

    COMMIT;

EXCEPTION
    WHEN insufficient_funds THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Transfer Failed: Insufficient funds.');

    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Transfer Failed: ' || SQLERRM);
END;
/
```

- **Scenario 2**

```
CREATE OR REPLACE PROCEDURE UpdateSalary (
   p_employee_id IN NUMBER,
   p_percentage  IN NUMBER
)
IS
   v_salary Employees.Salary%TYPE;
BEGIN
   -- Check existence
   SELECT Salary INTO v_salary FROM Employees WHERE EmployeeID = p_employee_id;

   -- Update salary
   UPDATE Employees
   SET Salary = Salary + (Salary * p_percentage / 100)
   WHERE EmployeeID = p_employee_id;

   DBMS_OUTPUT.PUT_LINE('Salary updated successfully.');

EXCEPTION
   WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('Error: Employee ID ' || p_employee_id || ' does not exist.');

   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Error updating salary: ' || SQLERRM);
END;
/
```

- **Scenario 3**

```
CREATE OR REPLACE PROCEDURE AddNewCustomer (
   p_customer_id   IN NUMBER,
   p_name          IN VARCHAR2,
   p_dob           IN DATE,
   p_balance       IN NUMBER
)
IS
BEGIN
   -- Try to insert
   INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)
   VALUES (p_customer_id, p_name, p_dob, p_balance, SYSDATE);

   DBMS_OUTPUT.PUT_LINE('Customer added successfully.');

EXCEPTION
   WHEN DUP_VAL_ON_INDEX THEN
      DBMS_OUTPUT.PUT_LINE('Error: Customer ID ' || p_customer_id || ' already exists.');

   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Error adding customer: ' || SQLERRM);
END;
/
```