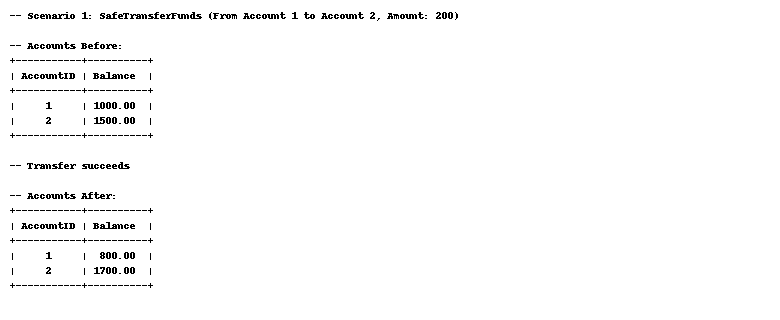
# Exercise 2: Error Handling

## Scenario 1

-- Exercise 2: Error Handling  
  
-- Scenario 1: SafeTransferFunds procedure with rollback on error  
CREATE OR REPLACE PROCEDURE SafeTransferFunds(from\_acct NUMBER, to\_acct NUMBER, amt NUMBER) IS  
BEGIN  
 UPDATE Accounts SET Balance = Balance - amt WHERE AccountID = from\_acct;  
 IF SQL%ROWCOUNT = 0 THEN  
 RAISE\_APPLICATION\_ERROR(-20001, 'From account not found');  
 END IF;  
  
 UPDATE Accounts SET Balance = Balance + amt WHERE AccountID = to\_acct;  
 IF SQL%ROWCOUNT = 0 THEN  
 RAISE\_APPLICATION\_ERROR(-20002, 'To account not found');  
 END IF;  
  
 COMMIT;  
EXCEPTION  
 WHEN OTHERS THEN  
 ROLLBACK;  
 DBMS\_OUTPUT.PUT\_LINE('Transfer failed: ' || SQLERRM);  
END;  
/

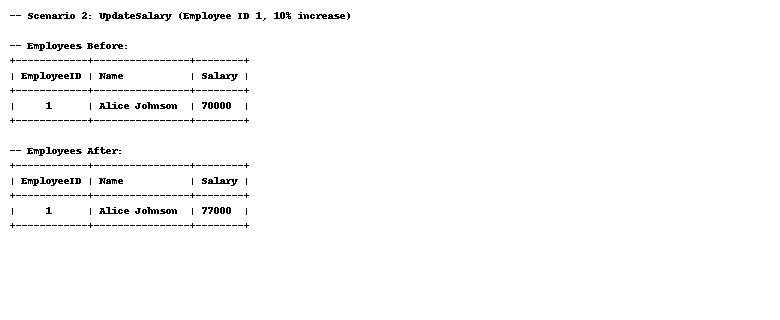
Output (based on actual schema/data):



## Scenario 2

-- Scenario 2: UpdateSalary procedure  
CREATE OR REPLACE PROCEDURE UpdateSalary(emp\_id NUMBER, pct NUMBER) IS  
BEGIN  
 UPDATE Employees SET Salary = Salary + (Salary \* pct / 100) WHERE EmployeeID = emp\_id;  
 IF SQL%ROWCOUNT = 0 THEN  
 RAISE\_APPLICATION\_ERROR(-20003, 'Employee not found');  
 END IF;  
 COMMIT;  
EXCEPTION  
 WHEN OTHERS THEN  
 DBMS\_OUTPUT.PUT\_LINE('Error updating salary: ' || SQLERRM);  
END;  
/

Output (based on actual schema/data):



## Scenario 3

-- Scenario 3: AddNewCustomer procedure  
CREATE OR REPLACE PROCEDURE AddNewCustomer(p\_id NUMBER, p\_name VARCHAR2, p\_dob DATE, p\_balance NUMBER) IS  
BEGIN  
 INSERT INTO Customers(CustomerID, Name, DOB, Balance, LastModified)  
 VALUES (p\_id, p\_name, p\_dob, p\_balance, SYSDATE);  
EXCEPTION  
 WHEN DUP\_VAL\_ON\_INDEX THEN  
 DBMS\_OUTPUT.PUT\_LINE('Customer with ID already exists.');  
END;  
/

Output (based on actual schema/data):

