**Scenario 1: Handle Exceptions During Fund Transfers Between Accounts**

**Solution:**

sql

CopyEdit

CREATE OR REPLACE PROCEDURE SafeTransferFunds (

p\_from\_account IN NUMBER,

p\_to\_account IN NUMBER,

p\_amount IN NUMBER

) IS

v\_from\_balance NUMBER;

v\_to\_balance NUMBER;

BEGIN

-- Check the balance of the source account

SELECT Balance INTO v\_from\_balance

FROM Accounts

WHERE AccountID = p\_from\_account;

-- Check if there are sufficient funds

IF v\_from\_balance < p\_amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds in source account.');

END IF;

-- Check the balance of the destination account

SELECT Balance INTO v\_to\_balance

FROM Accounts

WHERE AccountID = p\_to\_account;

-- Perform the fund transfer

UPDATE Accounts

SET Balance = v\_from\_balance - p\_amount

WHERE AccountID = p\_from\_account;

UPDATE Accounts

SET Balance = v\_to\_balance + p\_amount

WHERE AccountID = p\_to\_account;

-- Commit the transaction

COMMIT;

-- Log success message

DBMS\_OUTPUT.PUT\_LINE('Transfer successful.');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

-- Handle case where one of the accounts does not exist

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: Account not found.');

WHEN OTHERS THEN

-- Handle other unforeseen errors

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END SafeTransferFunds;

/

**Possible Outputs:**

1. **Successful Transfer** (if everything goes smoothly):

plaintext

CopyEdit

Transfer successful.

1. **Insufficient Funds** (if the source account has less than the transfer amount):

plaintext

CopyEdit

Error: Insufficient funds in source account.

1. **Account Not Found** (if either source or destination account does not exist):

plaintext

CopyEdit

Error: Account not found.

1. **Other Errors** (in case of unexpected errors like database issues):

plaintext

CopyEdit

Error: ORA-XXXXX: some other error message

**Scenario 2: Manage Errors When Updating Employee Salaries**

**Solution:**

sql

CopyEdit

CREATE OR REPLACE PROCEDURE UpdateSalary (

p\_employee\_id IN NUMBER,

p\_percentage IN NUMBER

) IS

v\_current\_salary NUMBER;

BEGIN

-- Fetch the current salary of the employee

SELECT Salary INTO v\_current\_salary

FROM Employees

WHERE EmployeeID = p\_employee\_id;

-- Update the salary based on the percentage

UPDATE Employees

SET Salary = v\_current\_salary \* (1 + p\_percentage / 100)

WHERE EmployeeID = p\_employee\_id;

-- Commit the transaction

COMMIT;

-- Log success message

DBMS\_OUTPUT.PUT\_LINE('Salary updated successfully.');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

-- Handle case where the employee does not exist

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: Employee ID does not exist.');

WHEN OTHERS THEN

-- Handle other unforeseen errors

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END UpdateSalary;

/

**Possible Outputs:**

1. **Successful Salary Update** (if the employee exists):

plaintext

CopyEdit

Salary updated successfully.

1. **Employee Not Found** (if the employee does not exist):

plaintext

CopyEdit

Error: Employee ID does not exist.

1. **Other Errors** (in case of unexpected errors):

plaintext

CopyEdit

Error: ORA-XXXXX: some other error message

**Scenario 3: Ensure Data Integrity When Adding a New Customer**

**Solution:**

sql

CopyEdit

CREATE OR REPLACE PROCEDURE AddNewCustomer (

p\_customer\_id IN NUMBER,

p\_name IN VARCHAR2,

p\_dob IN DATE,

p\_balance IN NUMBER

) IS

BEGIN

-- Check if the customer already exists

SELECT 1 INTO v\_dummy

FROM Customers

WHERE CustomerID = p\_customer\_id;

-- Raise an error if the customer already exists

RAISE\_APPLICATION\_ERROR(-20002, 'Customer with the same ID already exists.');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

-- If customer doesn't exist, insert the new customer

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (p\_customer\_id, p\_name, p\_dob, p\_balance, SYSDATE);

-- Commit the transaction

COMMIT;

-- Log success message

DBMS\_OUTPUT.PUT\_LINE('Customer added successfully.');

WHEN OTHERS THEN

-- Handle other unforeseen errors

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END AddNewCustomer;

/

**Possible Outputs:**

1. **Successful Customer Addition** (if the customer does not already exist):

plaintext

CopyEdit

Customer added successfully.

1. **Customer Already Exists** (if the customer ID already exists in the Customers table):

plaintext

CopyEdit

Error: Customer with the same ID already exists.

1. **Other Errors** (in case of unexpected errors):

plaintext

CopyEdit

Error: ORA-XXXXX: some other error message