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

```sql

CREATE OR REPLACE PROCEDURE SafeTransferFunds (

p\_from\_account\_id IN NUMBER,

p\_to\_account\_id IN NUMBER,

p\_amount IN NUMBER

) IS

BEGIN

-- Start a new transaction

SAVEPOINT start\_transaction;

-- Check if sufficient funds are available

DECLARE

v\_balance NUMBER;

BEGIN

SELECT balance INTO v\_balance

FROM accounts

WHERE account\_id = p\_from\_account\_id

FOR UPDATE;

IF v\_balance < p\_amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds for transfer.');

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Source account not found.');

END;

-- Perform the fund transfer

BEGIN

UPDATE accounts

SET balance = balance - p\_amount

WHERE account\_id = p\_from\_account\_id;

UPDATE accounts

SET balance = balance + p\_amount

WHERE account\_id = p\_to\_account\_id;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

-- Roll back to the savepoint in case of error

ROLLBACK TO start\_transaction;

RAISE\_APPLICATION\_ERROR(-20003, 'Transfer failed: ' || SQLERRM);

END;

END;

/

```

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

```sql

CREATE OR REPLACE PROCEDURE UpdateSalary (

p\_employee\_id IN NUMBER,

p\_percentage IN NUMBER

) IS

BEGIN

-- Update the employee salary

BEGIN

UPDATE employees

SET salary = salary \* (1 + p\_percentage / 100)

WHERE employee\_id = p\_employee\_id;

IF SQL%ROWCOUNT = 0 THEN

RAISE\_APPLICATION\_ERROR(-20004, 'Employee ID does not exist.');

END IF;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

-- Log the error

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

ROLLBACK;

END;

END;

/

```

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

```sql

CREATE OR REPLACE PROCEDURE AddNewCustomer (

p\_customer\_id IN NUMBER,

p\_customer\_name IN VARCHAR2,

p\_balance IN NUMBER

) IS

BEGIN

-- Insert the new customer

BEGIN

INSERT INTO customers (customer\_id, customer\_name, balance)

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

COMMIT;

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

-- Log the error for duplicate customer ID

DBMS\_OUTPUT.PUT\_LINE('Error: Customer ID already exists.');

ROLLBACK;

WHEN OTHERS THEN

-- Log other possible errors

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

ROLLBACK;

END;

END;

/

```