

Exercise 2: Error Handling

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;  
    v_to_balance NUMBER;  
BEGIN  
    -- Check if the from account has sufficient funds  
    SELECT balance INTO v_from_balance FROM accounts WHERE account_id = p_from_account_id;  
    IF v_from_balance < p_amount THEN  
        RAISE insufficient_funds;  
    END IF;  
    -- Perform the transfer  
    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 the transaction  
    COMMIT;  
EXCEPTION  
    WHEN insufficient_funds THEN  
        ROLLBACK;  
        DBMS_OUTPUT.PUT_LINE('Error: Insufficient funds in account ' || p_from_account_id);  
    WHEN OTHERS THEN  
        ROLLBACK;  
        DBMS_OUTPUT.PUT_LINE('Error: An unexpected error occurred during the fund transfer.');
```

END SafeTransferFunds;
/

Scenario 2:

```
CREATE OR REPLACE PROCEDURE UpdateSalary (  
    p_employee_id IN NUMBER,  
    p_percentage IN NUMBER  
) IS  
    v_salary employees.salary%TYPE;  
BEGIN  
    -- Check if the employee exists and update salary  
    SELECT salary INTO v_salary FROM employees WHERE employee_id = p_employee_id;  
    UPDATE employees SET salary = salary * (1 + p_percentage / 100) WHERE employee_id =  
    p_employee_id;
```

```

-- Commit the transaction

COMMIT;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        DBMS_OUTPUT.PUT_LINE('Error: Employee ID ' || p_employee_id || ' does not exist.');
```

WHEN OTHERS THEN

```

        ROLLBACK;

        DBMS_OUTPUT.PUT_LINE('Error: An unexpected error occurred while updating the salary.');
```

END UpdateSalary;

/

Scenario 3:

```

CREATE OR REPLACE PROCEDURE AddNewCustomer (

    p_customer_id IN NUMBER,

    p_name IN VARCHAR2,

    p_age IN NUMBER,

    p_balance IN NUMBER

) IS

    customer_exists EXCEPTION;

BEGIN

    -- Check if the customer already exists

    BEGIN

        SELECT customer_id INTO v_customer_id FROM customers WHERE customer_id =
p_customer_id;

        RAISE customer_exists;

    EXCEPTION

        WHEN NO_DATA_FOUND THEN

            -- Customer does not exist, proceed with insertion

            INSERT INTO customers (customer_id, name, age, balance)

            VALUES (p_customer_id, p_name, p_age, p_balance);

            COMMIT;

    END;
```

EXCEPTION

WHEN customer_exists THEN

DBMS_OUTPUT.PUT_LINE('Error: Customer ID ' || p_customer_id || ' already exists.');

WHEN OTHERS THEN

ROLLBACK;

DBMS_OUTPUT.PUT_LINE('Error: An unexpected error occurred while adding the new customer.');

END AddNewCustomer;

/