

```

CREATE OR REPLACE PROCEDURE SafeTransferFunds (
    p_from_account_id IN NUMBER,
    p_to_account_id IN NUMBER,
    p_amount IN NUMBER
) AS
    insufficient_funds EXCEPTION;
    PRAGMA EXCEPTION_INIT(insufficient_funds, -20001);
BEGIN
    -- Check if the from account has enough balance

    DECLARE
        v_balance NUMBER;
    BEGIN
        SELECT Balance INTO v_balance FROM Accounts WHERE AccountID = p_from_account_id;

        IF v_balance < p_amount THEN
            RAISE insufficient_funds;
        END IF;
    END;

    -- Perform the transfer

    UPDATE Accounts
    SET Balance = Balance - p_amount
    WHERE AccountID = p_from_account_id;

    UPDATE Accounts
    SET Balance = Balance + p_amount
    WHERE AccountID = p_to_account_id;

    COMMIT;

    DBMS_OUTPUT.PUT_LINE('Transfer completed successfully.');
```

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: ' || SQLERRM);

END SafeTransferFunds;

```

```

CREATE OR REPLACE PROCEDURE UpdateSalary (

    p_employee_id IN NUMBER,

    p_percentage IN NUMBER

) AS

    employee_not_found EXCEPTION;

    PRAGMA EXCEPTION_INIT(employee_not_found, -20001);

BEGIN

    -- Check if the employee exists

    DECLARE

        v_salary NUMBER;

    BEGIN

        SELECT Salary INTO v_salary FROM Employees WHERE EmployeeID = p_employee_id;

    EXCEPTION

        WHEN NO_DATA_FOUND THEN

            RAISE employee_not_found;

    END;

    -- Update the salary

    UPDATE Employees

    SET Salary = Salary + (Salary * p_percentage / 100)

    WHERE EmployeeID = p_employee_id;

    COMMIT;

```

```

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

EXCEPTION

```

    WHEN employee_not_found THEN

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

WHEN OTHERS THEN

```

        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
```

END UpdateSalary;

CREATE OR REPLACE PROCEDURE AddNewCustomer (

```

    p_customer_id IN NUMBER,
    p_name IN VARCHAR2,
    p_dob IN DATE,
    p_balance IN NUMBER
) AS
```

```

    customer_already_exists EXCEPTION;

    PRAGMA EXCEPTION_INIT(customer_already_exists, -20001);
```

BEGIN

```

    -- Check if the customer already exists

    DECLARE

        v_existing_id NUMBER;

    BEGIN

        SELECT CustomerID INTO v_existing_id FROM Customers WHERE CustomerID = p_customer_id;

        RAISE customer_already_exists;
```

EXCEPTION

```

    WHEN NO_DATA_FOUND THEN

        NULL;
```

END;


```

    -- Insert the new customer

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

```
VALUES (p_customer_id, p_name, p_dob, p_balance, SYSDATE);
```

```
COMMIT;
```

```
DBMS_OUTPUT.PUT_LINE('New customer added successfully.');
```

```
EXCEPTION
```

```
WHEN customer_already_exists THEN
```

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

```
WHEN OTHERS THEN
```

```
    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
```

```
END AddNewCustomer;
```