Error Handling

1. CREATE OR REPLACE PROCEDURE SafeTransferFunds(

p\_from\_account\_id NUMBER,

p\_to\_account\_id NUMBER,

p\_amount NUMBER

) AS

v\_from\_balance NUMBER;

v\_to\_balance NUMBER;

BEGIN

SELECT balance INTO v\_from\_balance FROM accounts WHERE account\_id = p\_from\_account\_id;

SELECT balance INTO v\_to\_balance FROM accounts WHERE account\_id = p\_to\_account\_id;

IF v\_from\_balance >= p\_amount THEN

UPDATE accounts SET balance = v\_from\_balance - p\_amount WHERE account\_id = p\_from\_account\_id;

UPDATE accounts SET balance = v\_to\_balance + p\_amount WHERE account\_id = p\_to\_account\_id;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Funds transferred successfully.');

ELSE

ROLLBACK; -- Rollback the transaction

DBMS\_OUTPUT.PUT\_LINE('Insufficient funds for transfer.');

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

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

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An error occurred. Transaction rolled back.');

ROLLBACK; -- Rollback for any other exceptions

END SafeTransferFunds;

/

1. CREATE OR REPLACE PROCEDURE UpdateSalary(

p\_employee\_id NUMBER,

p\_percentage\_increase NUMBER

) AS

v\_current\_salary NUMBER;

v\_new\_salary NUMBER;

BEGIN

SELECT salary INTO v\_current\_salary FROM employees WHERE employee\_id = p\_employee\_id;

v\_new\_salary := v\_current\_salary \* (1 + p\_percentage\_increase / 100);

UPDATE employees SET salary = v\_new\_salary WHERE employee\_id = p\_employee\_id;

DBMS\_OUTPUT.PUT\_LINE('Salary updated successfully for Employee ' || p\_employee\_id || '. New salary: $' || v\_new\_salary);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Employee ' || p\_employee\_id || ' not found.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An error occurred while updating the salary.');

END UpdateSalary;

/

1. CREATE OR REPLACE PROCEDURE AddNewCustomer(

p\_customer\_id NUMBER,

p\_customer\_name VARCHAR2,

p\_customer\_email VARCHAR2

) AS

BEGIN

SELECT COUNT(\*) INTO v\_existing\_count FROM customers WHERE customer\_id = p\_customer\_id;

IF v\_existing\_count > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Customer with ID ' || p\_customer\_id || ' already exists.');

ELSE

INSERT INTO customers (customer\_id, name, email) VALUES (p\_customer\_id, p\_customer\_name, p\_customer\_email);

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

END IF;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An error occurred while adding the customer.');

END AddNewCustomer;

/