**WEEK-2**

**MODULE -3**

**PL/SQL PROGRAMMING**

**SUPERSET ID:6407550**

**Exercise:2-Error Handling**

-- Create two accounts

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

VALUES (3, 1, 'Savings', 50, SYSDATE); -- Low balance account

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

VALUES (4, 2, 'Checking', 1000, SYSDATE);

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

VALUES (7, 'Duplicate Don', TO\_DATE('1991-01-01', 'YYYY-MM-DD'), 5000, SYSDATE);

**Scenario:1**

CREATE OR REPLACE PROCEDURE SafeTransferFunds(

p\_FromAccountID IN NUMBER,

p\_ToAccountID IN NUMBER,

p\_Amount IN NUMBER

) IS

v\_FromBalance NUMBER;

BEGIN

SELECT Balance INTO v\_FromBalance FROM Accounts WHERE AccountID = p\_FromAccountID;

IF v\_FromBalance < p\_Amount THEN

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

END IF;

UPDATE Accounts SET Balance = Balance - p\_Amount WHERE AccountID = p\_FromAccountID;

UPDATE Accounts SET Balance = Balance + p\_Amount WHERE AccountID = p\_ToAccountID;

COMMIT;

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

EXCEPTION

WHEN OTHERS THEN

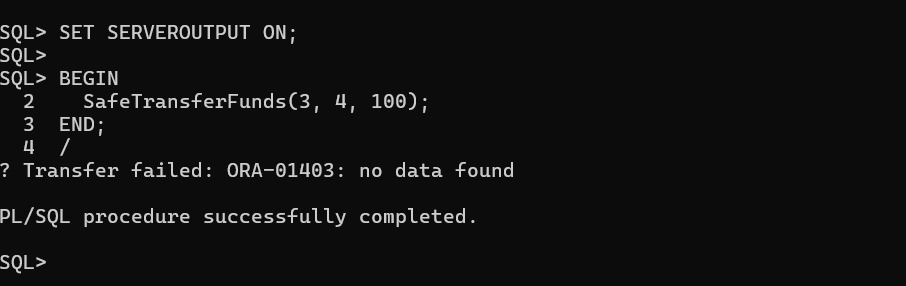
ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('❌ Transfer failed: ' || SQLERRM);

END;

/

**OUTPUT:**

****

**Scenario:2**

CREATE OR REPLACE PROCEDURE UpdateSalary(

p\_EmployeeID IN NUMBER,

p\_Percent IN NUMBER

) IS

v\_OldSalary NUMBER;

BEGIN

SELECT Salary INTO v\_OldSalary FROM Employees WHERE EmployeeID = p\_EmployeeID;

UPDATE Employees

SET Salary = Salary + (Salary \* p\_Percent / 100)

WHERE EmployeeID = p\_EmployeeID;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('✅ Salary updated for Employee ID ' || p\_EmployeeID);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('❌ Error: Employee ID ' || p\_EmployeeID || ' not found.');

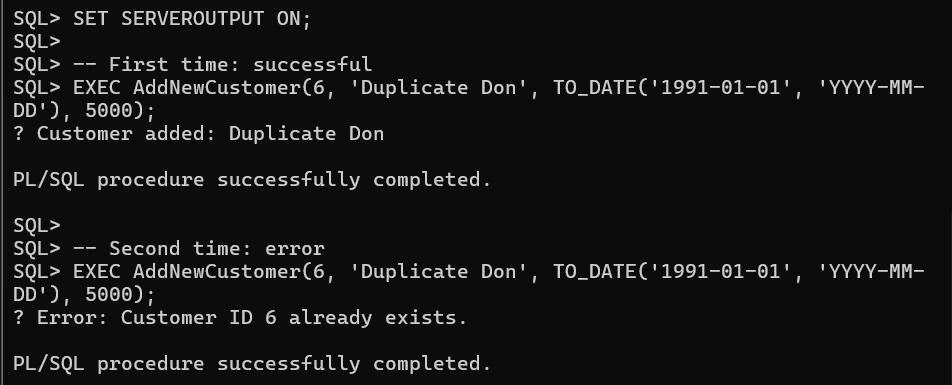
WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('❌ Unexpected error: ' || SQLERRM);

END;

/

**OUTPUT:**

****

**Scenario:3**

CREATE OR REPLACE PROCEDURE AddNewCustomer(

p\_CustomerID IN NUMBER,

p\_Name IN VARCHAR2,

p\_DOB IN DATE,

p\_Balance IN NUMBER

) IS

BEGIN

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

VALUES (p\_CustomerID, p\_Name, p\_DOB, p\_Balance, SYSDATE);

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('✅ Customer added: ' || p\_Name);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

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

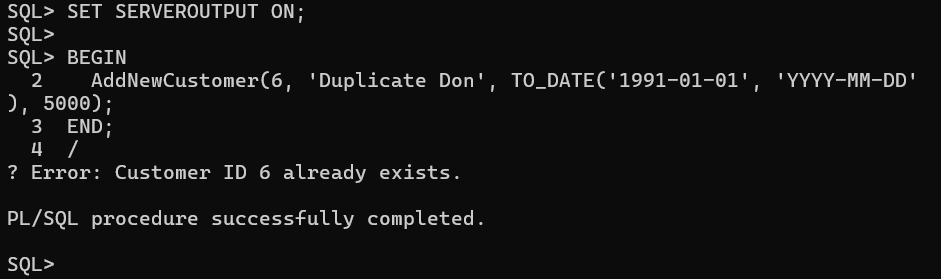
WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('❌ Unexpected error: ' || SQLERRM);

END;

/

**OUTPUT:**

****