**Practical No : 07**

**Handling Exceptions in PL/SQL**

1. **User defined exception**

**B) System Defined Exception:**

**Name :Sibaraj Lenka**

**Class : FY.Bsc.IT(A)**

**Roll No : 58**

**Subject : PL/SQL**

**Sign :**

**A)User defined exception**

**CREATE TABLE**

CREATE TABLE account\_58(account\_id NUMBER PRIMARY KEY, name VARCHAR2(100), balance NUMBER);

INSERT INTO account\_58(account\_id, name, balance) VALUES(1,'Sibaraj', 50000);

**Input:**

Set serveroutput on;

DECLARE

insufficient\_balance EXCEPTION; -- Step 1: Declare the exception

balance NUMBER := 50000; -- Assume initial balance is 500

withdraw\_amount NUMBER := 600; -- Attempt to withdraw 600

BEGIN

-- Check if withdrawal exceeds balance

IF withdraw\_amount > balance THEN

RAISE insufficient\_balance; -- Step 2: Raise the exception

ELSE

balance := balance - withdraw\_amount;

DBMS\_OUTPUT.PUT\_LINE('Withdrawal successful. Remaining balance: ' || balance);

END IF;

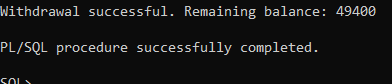
EXCEPTION

WHEN insufficient\_balance THEN -- Step 3: Handle the exception

DBMS\_OUTPUT.PUT\_LINE('Error: Insufficient balance for withdrawal.');

END;

**Output:**

****

**B) System Defined Exception:**

Example 1: Handling ZERO\_DIVIDE

**Input :**

BEGIN

-- Attempt division by zero

DECLARE

num1 NUMBER := 10;

num2 NUMBER := 0;

result NUMBER;

BEGIN

result := num1 / num2; -- This will raise ZERO\_DIVIDE

EXCEPTION

WHEN ZERO\_DIVIDE THEN

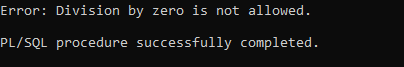
DBMS\_OUTPUT.PUT\_LINE('Error: Division by zero is not allowed.');

END;

END;

/

**Output :**

****

Example 2: Handling Multiple Exceptions

**Input :**

BEGIN

DECLARE

num1 NUMBER := 10;

num2 NUMBER := 0;

result NUMBER;

BEGIN

result := num1 / num2; -- Raises ZERO\_DIVIDE

EXCEPTION

WHEN ZERO\_DIVIDE THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Division by zero.');

WHEN OTHERS THEN

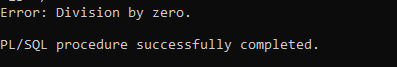
DBMS\_OUTPUT.PUT\_LINE('An unknown error occurred.');

END;

END;

/

**Output :**

****