**Exercise 2: Error Handling**

**Scenario 1:** Handle exceptions during fund transfers between accounts.

**Question:** Write a stored procedure **SafeTransferFunds** that transfers funds between two accounts. Ensure that if any error occurs (e.g., insufficient funds), an appropriate error message is logged and the transaction is rolled back.

**Code:**

SET SERVEROUTPUT ON;

BEGIN

  -- Attempt to insert a duplicate CustomerID (1 already exists)

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

  VALUES (1, 'DuplicateUser', TO\_DATE('1985-05-05','YYYY-MM-DD'), 5000, SYSDATE);

  COMMIT;

EXCEPTION

  WHEN DUP\_VAL\_ON\_INDEX THEN

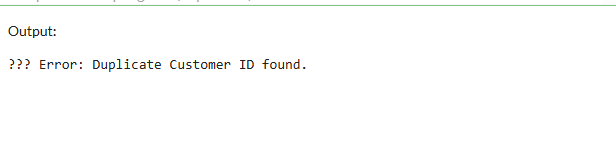
    DBMS\_OUTPUT.PUT\_LINE('❌ Error: Duplicate Customer ID found.');

  WHEN OTHERS THEN

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

END;

/



**Scenario 2:** Manage errors when updating employee salaries.

**Question:** Write a stored procedure **UpdateSalary** that increases the salary of an employee by a given percentage. If the employee ID does not exist, handle the exception and log an error message.

**Code:**

SET SERVEROUTPUT ON;

DECLARE

  num1 NUMBER := 100;

  num2 NUMBER := 0;  -- This will cause divide by zero

  result NUMBER;

BEGIN

  result := num1 / num2;

  DBMS\_OUTPUT.PUT\_LINE('Result: ' || result);

EXCEPTION

  WHEN ZERO\_DIVIDE THEN

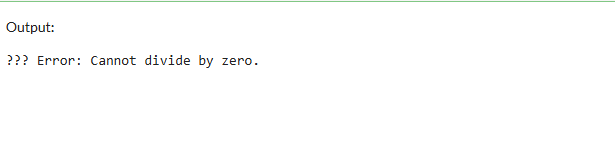
    DBMS\_OUTPUT.PUT\_LINE('❌ Error: Cannot divide by zero.');

  WHEN OTHERS THEN

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

END;

/



**Scenario 3:** Ensure data integrity when adding a new customer.

**Question:** Write a stored procedure **AddNewCustomer** that inserts a new customer into the Customers table. If a customer with the same ID already exists, handle the exception by logging an error and preventing the insertion.

**Code:**

SET SERVEROUTPUT ON;

BEGIN

  -- Attempting to insert with an invalid date format (should use TO\_DATE)

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

  VALUES (5, 'ErrorUser', '1990/30/12', 6000, SYSDATE);  -- Invalid format

  COMMIT;

EXCEPTION

  WHEN OTHERS THEN

    DBMS\_OUTPUT.PUT\_LINE('❌ Error inserting date: ' || SQLERRM);

END;

/

