DEPARTMENT OF COMPUTER SYSTEMS ENGINEERING
MEHRAN UNIVERSITY OF ENGINEERING & TECHNOLOGY, JAMSHORO
Database Management Systems (4th Semester) 18CS

Lab Experiment 12

Roll No: **Date of Conduct:**

Submission Date: **Grade Obtained:**

Problem Recognition (0.3)	Completeness & accuracy (0.4)	Timeliness (0.3)	Score (1.0)

Objective: **To explore Exception Handling in PL/SQL.**

Tools: MySql, Oracle.

Introduction:

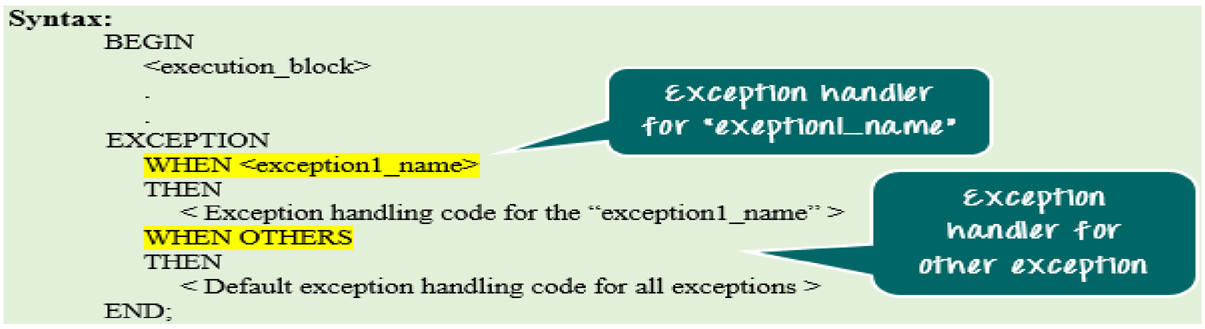
An exception occurs when the PL/SQL engine encounters an instruction which it cannot execute due to an error that occurs at run-time. These errors will not be captured at the time of compilation and hence these needed to handle only at the run-time.

For example: if PL/SQL engine receives an instruction to divide any number by '0', then the PL/SQL engine will throw it as an exception. The exception is only raised at the run-time by the PL/SQL engine.

Exceptions will stop the program from executing further, so to avoid such condition, they need to be captured and handled separately. This process is called as Exception-Handling, in which the programmer handles the exception that can occur at the run time.

Exception-Handling Syntax:

Exceptions are handled at the block, level, i.e., once if any exception occurs in any block then the control will come out of execution part of that block. The exception will then be handled at the exception handling part of that block.


Syntax:

```
BEGIN
    <execution_block>
    .
    .
EXCEPTION
    WHEN <exception1_name>
    THEN
        < Exception handling code for the "exception1_name" >
    WHEN OTHERS
    THEN
        < Default exception handling code for all exceptions >
END;
```

Exception handler for 'exeption1_name'
Exception handler for other exception

Types of Exception:

1. Predefined Exceptions
2. User-defined Exception

Predefined Exceptions: Oracle has predefined some common exception. These exceptions have a unique exception name and error number. These exceptions are already defined in the 'STANDARD' package in Oracle. In code, we can directly use these predefined exception name to handle them.

User-defined Exception: In Oracle, other than the above-predefined exceptions, the programmer can create their own exception and handle them. They can be created at a subprogram level in the declaration part. These exceptions are visible only in that subprogram. The exception that is defined in the package specification is public exception, and it is visible wherever the package is accessible.

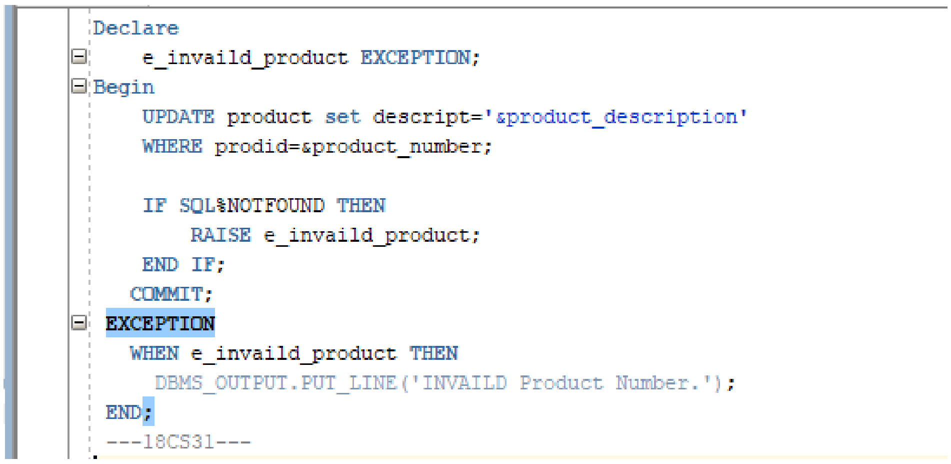
```
DECLARE
<exception_name> EXCEPTION;
BEGIN
<Execution block>
EXCEPTION
WHEN <exception_name> THEN
<Handler>
END;
```

- In the above syntax, the variable 'exception_name' is defined as 'EXCEPTION' type.
- This can be used as in a similar way as a predefined exception.

Lab Task

1. Write a PL/SQL block that updates description of a product and raises a user-defined exception when that product is not found.

Task:



```
Declare
    e_invaild_product EXCEPTION;
Begin
    UPDATE product set descript='&product_description'
    WHERE prodid=&product_number;

    IF SQL%NOTFOUND THEN
        RAISE e_invaild_product;
    END IF;
   COMMIT;
 EXCEPTION
   WHEN e_invaild_product THEN
     DBMS_OUTPUT.PUT_LINE('INVAILD Product Number.');
 END;
 ---18CS31---
```

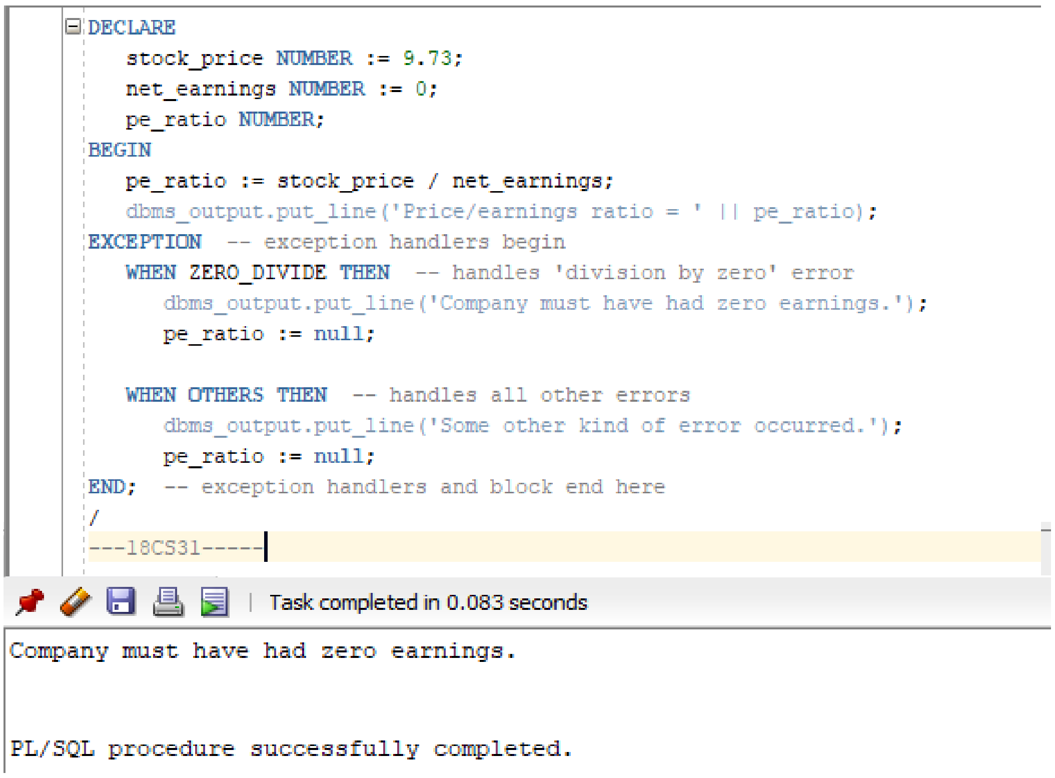
Output:

```
Invalid product number.


PL/SQL procedure successfully completed.
```

2. Write a simple PL/SQL code block, to demonstrate the use of Named Exception Handler for division by zero error.

Task:



```
DECLARE
   stock_price NUMBER := 9.73;
   net_earnings NUMBER := 0;
   pe_ratio NUMBER;
BEGIN
   pe_ratio := stock_price / net_earnings;
   dbms_output.put_line('Price/earnings ratio = ' || pe_ratio);
EXCEPTION  -- exception handlers begin
   WHEN ZERO_DIVIDE THEN  -- handles 'division by zero' error
      dbms_output.put_line('Company must have had zero earnings.');
      pe_ratio := null;

   WHEN OTHERS THEN  -- handles all other errors
      dbms_output.put_line('Some other kind of error occurred.');
      pe_ratio := null;
END;  -- exception handlers and block end here
/
---18CS31-----
```

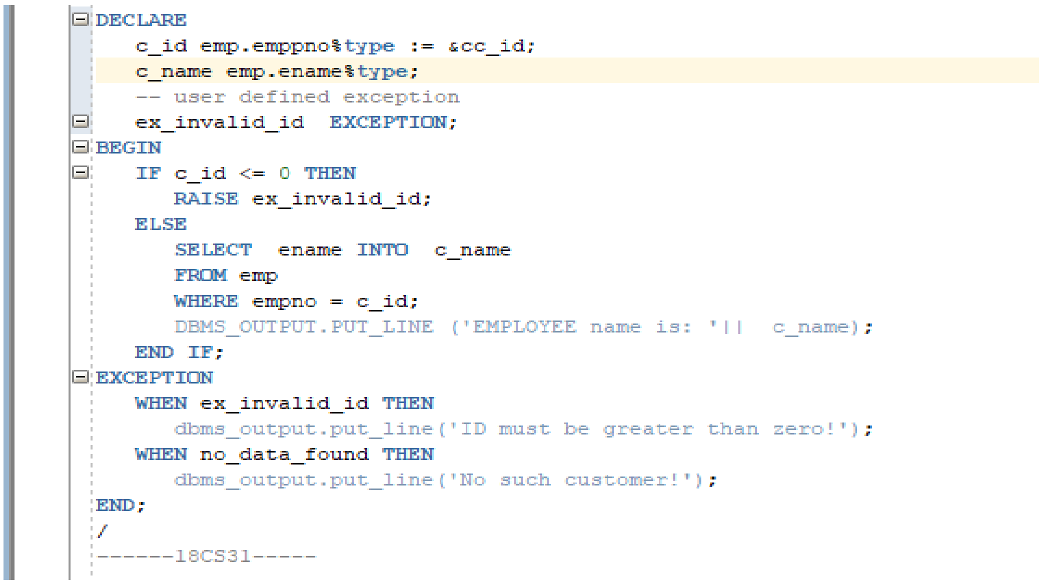
Task completed in 0.083 seconds

```
Company must have had zero earnings.


PL/SQL procedure successfully completed.
```

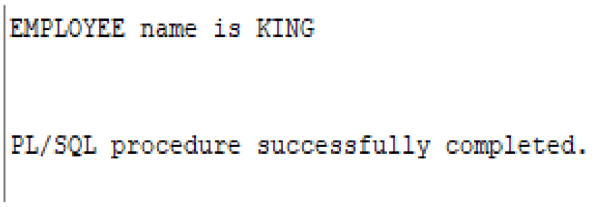
3. Write a PL/SQL block for displaying information of a customer ID, when the user enters an invalid ID, raise the exception invalid_id.

Task:



```
DECLARE
    c_id emp.emppno%type := &cc_id;
    c_name emp.ename%type;
    -- user defined exception
    ex_invalid_id  EXCEPTION;
BEGIN
    IF c_id <= 0 THEN
        RAISE ex_invalid_id;
    ELSE
        SELECT  ename INTO  c_name
        FROM emp
        WHERE empno = c_id;
        DBMS_OUTPUT.PUT_LINE ('EMPLOYEE name is: '||  c_name);
    END IF;
EXCEPTION
    WHEN ex_invalid_id THEN
        dbms_output.put_line('ID must be greater than zero!');
    WHEN no_data_found THEN
        dbms_output.put_line('No such customer!');
END;
/
------18CS31-----
```

Output:



```
EMPLOYEE name is KING


PL/SQL procedure successfully completed.
```