

Exception Handling

In PL/SQL, a warning or error condition is called an exception, which arises during program execution. Run-time errors or exception arise from design faults, coding mistakes, hardware failure and many other sources. When an error occurs, an exception is raised, i.e. the normal execution stops and the control transfers to the exception handling part of the PL/SQL block.

Exceptions are the identifiers in PL/SQL which may be raised during the execution of a block to terminate its main body of actions. A block will always terminate when an exception is raised, but you may specify an 'Exception Handler' to perform final actions before the block terminates.

PL/SQL exceptions consist following three,

1. Exception Type
2. Error Code
3. Error Message

TYPES OF EXCEPTION

Exceptions can be of two types :-

- (1) Predefined Exception
- (2) User-defined Exception

PL/SQL Predefined Exception :-

They are internally defined by runtime system. It is raised implicitly whenever PL/SQL program violates an ORACLE rule or exceeds a system-dependent limit. Every Oracle error has a number, but exceptions must be handled by name. So, PL/SQL predefined some common ORACLE errors as exceptions.

For example, if you try to divide a number by zero then PL/SQL raises an exception called ZERO_DIVIDE and if SELECT can not find a record then PL/SQL raises exception NO_DATA_FOUND.

How to handle exception ??

When PL/SQL raises a predefined exception, the program is aborted by displaying error message. But if program is to handle exception raised by PL/SQL then we have to use Exception handling part of the PL/SQL block.

Exception handling part is used to specify the statements to be executed when an exception occurs. Control is transferred to exception handling part whenever an

exception occurs. After the exception handler is executed, control is transferred to next statement in the enclosing block.

Syntax :-

```
WHEN <exception-identifier> THEN <action>;
```

Where <actions> may be one or more PL/SQL or SQL statements, each terminated by semi-colons.

Example :-

```
DECLARE
    N NUMBER;
BEGIN
    N:=10/0;
EXCEPTION
    WHEN ZERO_DIVIDE THEN
        DBMS_OUTPUT.PUT_LINE (' ZERO DIVIDE ERROR' );
END;
/
```

PL/SQL User Defined Exception

PL/SQL user defined exception to make your own exception. PL/SQL give you control to make your own exception base on oracle rules. User define exception must be declare yourself and RAISE statement to raise explicitly.

How to Define Exception

(1) Declare exception :-You must have to declare user define exception name in DECLARE block.

```
user_define_exception_name EXCEPTION;
```

(2) RAISE exception :- RAISE statement to raised defined exception name and control transfer to a EXCEPTION block.

```
RAISE user_define_exception_name;
```

(3) Handling an Exception Condition :- In PL/SQL EXCEPTION block add WHEN condition to implement user-defined action.

```
BEGIN
```

```
-----
```

```
-----
```

```
EXCEPTION
```

```
WHEN user_define_exception_name THEN
```

```
    User-defined statements (action) will be taken;
```

```
END;
```