**EXCEPTION HANDLING IN PL/SQL**

**EXCEPTION:** Runtime errors are called an Exception. If at any time an error occurs in the PL/SQL block at that time PL/SQL block execution is stopped and oracle returns an error message.

To continue the program execution and to display user friendly message exception needs to be handle exception include exception block in PL/SQL.

Exceptions are classified into two types. Those are

1. System/Pre Defined Exception
2. User Defined Exception

**Syntax:**

Declare

< Variables, cursor, user define exception>;

Begin

<statements…………..>;

Exception

When <Exception Name> then

<Error Statements…….>;

End;

1. **System/Pre-Defined Exception:**

These are defined by ORACLE by default. Whenever runtime error is occurred in PL/SQL then we use an appropriate pre defined exception in the program.

**SOME PRE-DEFINED EXCEPTIONS:**

1. No\_data\_found
2. Too\_many\_rows
3. Zero\_divide
4. Invalid \_cursor
5. Cursor\_already\_open……etc

**No\_data\_found:** Whenever PL/SQL block carry the SELECT…..INTO clause and also if required data not available in a table then ORACLE server returns an exception.

Ex: ORA-1403: no data found

To handle this exception oracle provided “No\_data\_found” exception.

**Ex:**

DECLARE TENAME VARCHAR2(20);TSAL NUMBER(10);

BEGIN

SELECT ENAME,SAL INTO TENAME,TSAL FROM EMPLOYEE WHERE EID=&EID;

DBMS\_OUTPUT.PUT\_LINE(TENAME||','||TSAL);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('RECORD IS NOT FOUND');

END;

/

**Too\_many\_rows:** When SELECT…..INTO clause try to return more than one value or one row then oracle server returns an error.

EX: ORA-1422: Exact fetch returns more than requested number of rows.

To handle for this error oracle provide “Too\_many\_rows” exception.

**EX:**

DECLARE TSAL NUMBER(10);

BEGIN

SELECT SAL INTO TSAL FROM EMPLOYEE;

DBMS\_OUTPUT.PUT\_LINE(TSAL);

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('FETCHING MORE THAN ONE');

END;

/

**Zero\_divide:-** In oracle when we are tried to perform division with zero then oracle return an error.

ORA-1476: divisor is equal to zero.

To handle for this error oracle provide “Zero\_divide” exception

**EX:**

DECLARE X NUMBER(10);Y NUMBER(10);Z NUMBER(10);

BEGIN

X:=&X;

Y:=&Y;

Z:=X/Y;

DBMS\_OUTPUT.PUT\_LINE('RESULT:-'||Z);

EXCEPTION

WHEN ZERO\_DIVIDE THEN

DBMS\_OUTPUT.PUT\_LINE('SECOND NUMBER SHOULD NOT BE ZERO');

END;

**/**

**Invalid cursor**: When we are not opening the cursor but we are try to perform operations on cursor then ORACLE returns an error.

EX: ORA-1001: invalid cursor

To handle this error oracle provide “Invalid\_cursor” exception.

EX:

declare

cursor c1 is select \* from employee;

teid number(10);tename varchar2(20);tsal number(10);tage number(10);

begin

fetch c1 into teid,tename,tsal,tage;

dbms\_output.put\_line(teid||' '||tename||' '||tsal||' '||tage);

close c1;

exception

when invalid\_cursor then

dbms\_output.put\_line('First you must open the cursor');

end;

**/**

**Cursor\_already\_open**: Before reopening the cursor we must close the cursor properly otherwise ORACLE returns an error i.e.

EX: ORA-6511: Cursor\_already\_open

To handle this error oracle provide ‘Cursor\_already\_open’ exception.

**EX**:

declare

cursor c1 is select \* from employee;

teid number(10);tename varchar2(20);tsal number(10);tage number(10);

begin

open c1;

loop

fetch c1 into teid,tename,tsal,tage;

exit when c1%notfound;

dbms\_output.put\_line(teid||' '||tename||' '||tsal||' '||tage);

end loop;

open c1;

exception

when cursor\_already\_open then

dbms\_output.put\_line('we must close the cursor before reopen');

end;

**SQLCODE & SQLERRM:** PL/SQL provides following built-in properties which are used in error handling.

SQLCODE returns error code.

SQLERRM returns error message.

Ex:

DECLARE

X NUMBER(10);

Y NUMBER(20);

Z NUMBER(10);

BEGIN

X:=&X;

Y:=&Y;

Z:=X/Y;

DBMS\_OUTPUT.PUT\_LINE(Z);

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE(SQLCODE);

DBMS\_OUTPUT.PUT\_LINE(SQLERRM);

END;

**OUTPUT:**

Enter value for x: 10

Enter value for y: 2

5

Enter value for x: 10

Enter value for y: 0

-1476---------error code

ORA-01476: divisor is equal to zero------error message