

Sanjivani Rural Education Society's College of Engineering, Kopargaon
Department of Electronics and Computer Engineering
TITLE: Experiment Write-up (EW)

EXPERIMENT NO.: 08

TITLE: Write a PL SQL code to implement exception handling

LEARNING OBJECTIVES:

1. To study the fundamental concepts of database management.
2. To learn the basic issues of transaction processing and concurrency control.
3. To learn a powerful, flexible and scalable general-purpose distributed database.

An error occurs during the program execution is called Exception in PL/SQL.

PL/SQL facilitates programmers to catch such conditions using exception block in the program and an appropriate action is taken against the error condition.

There are two type of exceptions:

- System-defined Exceptions
- User-defined Exceptions

Syntax for exception handling:

```
DECLARE
    <declarations section>
BEGIN
    <executable command(s)>
EXCEPTION
    <exception handling goes here >
    WHEN exception1 THEN
        exception1-handling-statements
    WHEN exception2 THEN
        exception2-handling-statements
    WHEN exception3 THEN
        exception3-handling-statements
    .....
    WHEN others THEN
        exception3-handling-statements
END;
```

Prepared by

Mr. N. I. Bhopale
(Subject Teacher)

Verified by

Dr. B. S. Agarkar
(HOD, Deptt. of ECE)

Sanjivani Rural Education Society's College of Engineering, Kopargaon
Department of Electronics and Computer Engineering
TITLE: Experiment Write-up (EW)

Example of exception handling

SELECT* FROM COUSTOMERS;

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	23	Allahabad	20000
2	Suresh	22	Kanpur	22000
3	Mahesh	24	Ghaziabad	24000
4	Chandan	25	Noida	26000
5	Alex	21	Paris	28000
6	Sunita	20	Delhi	30000

```
DECLARE
  c_id customers.id%type := 8;
  c_name customers.name%type;
  c_addr customers.address%type;
BEGIN
  SELECT name, address INTO c_name, c_addr
    FROM customers
   WHERE id = c_id;
  DBMS_OUTPUT.PUT_LINE ('Name: '|| c_name);
  DBMS_OUTPUT.PUT_LINE ('Address: ' || c_addr);
EXCEPTION
  WHEN no_data_found THEN
    dbms_output.put_line('No such customer!');
  WHEN others THEN
    dbms_output.put_line('Error!');
END;
/
```

RESULT: No such customer!
PL/SQL procedure successfully completed.

The above program should show the name and address of a customer as result whose ID is given. But there is no customer with ID value 8 in our database, so the program raises the run-time exception NO_DATA_FOUND, which is captured in EXCEPTION block.

Syntax for raising an exception:

Prepared by

Mr. N. I. Bhopale
(Subject Teacher)

Verified by

Dr. B. S. Agarkar
(HOD, Deptt. of ECE)

Sanjivani Rural Education Society's College of Engineering, Kopargaon
Department of Electronics and Computer Engineering
TITLE: Experiment Write-up (EW)

```
DECLARE
    exception_name EXCEPTION;
BEGIN
    IF condition THEN
        RAISE exception_name;
    END IF;
EXCEPTION
    WHEN exception_name THEN
        statement;
END;
```

Syntax for user define exceptions

```
DECLARE
my-exception EXCEPTION;
```

Exception	Oracle Error	SQL Code	Description
ACCESS_INTO_NULL	06530	-6530	It is raised when a NULL object is automatically assigned a value.
CASE_NOT_FOUND	06592	-6592	It is raised when none of the choices in the "WHEN" clauses of a CASE statement is selected, and there is no else clause.
COLLECTION_IS_NULL	06531	-6531	It is raised when a program attempts to apply collection methods other than exists to an uninitialized nested table or varray, or the program attempts to assign values to the elements of an uninitialized nested table or varray.
DUP_VAL_ON_INDEX	00001	-1	It is raised when duplicate values are attempted to be stored in a column with unique index.
INVALID_CURSOR	01001	-1001	It is raised when attempts are made to make a cursor operation that is not allowed, such as closing an unopened cursor.

Prepared by

Mr. N. I. Bhopale
(Subject Teacher)

Verified by

Dr. B. S. Agarkar
(HOD, Deptt. of ECE)

Sanjivani Rural Education Society's College of Engineering, Kopargaon
Department of Electronics and Computer Engineering
TITLE: Experiment Write-up (EW)

INVALID_NUMBER	01722	-1722	It is raised when the conversion of a character string into a number fails because the string does not represent a valid number.
LOGIN_DENIED	01017	-1017	It is raised when a program attempts to log on to the database with an invalid username or password.
NO_DATA_FOUND	01403	+100	It is raised when a select into statement returns no rows.
NOT_LOGGED_ON	01012	-1012	It is raised when a database call is issued without being connected to the database.
PROGRAM_ERROR	06501	-6501	It is raised when PL/SQL has an internal problem.
ROWTYPE_MISMATCH	06504	-6504	It is raised when a cursor fetches value in a variable having incompatible data type.
SELF_IS_NULL	30625	-30625	It is raised when a member method is invoked, but the instance of the object type was not initialized.
STORAGE_ERROR	06500	-6500	It is raised when PL/SQL ran out of memory or memory was corrupted.
TOO_MANY_ROWS	01422	-1422	It is raised when a SELECT INTO statement returns more than one row.
VALUE_ERROR	06502	-6502	It is raised when an arithmetic, conversion, truncation, or size-constraint error occurs.
ZERO_DIVIDE	01476	1476	It is raised when an attempt is made to divide a number by zero.

```

DECLARE
    a int;
    b int;
    c int;
BEGIN
    a := 11;

```

Prepared by

Mr. N. I. Bhopale
(Subject Teacher)

Verified by

Dr. B. S. Agarkar
(HOD, Deptt. of ECE)

Sanjivani Rural Education Society's College of Engineering, Kopargaon
Department of Electronics and Computer Engineering
TITLE: Experiment Write-up (EW)

```
b := 0;  
c := a/b;  
dbms_output.put_line('RESULT=' || c);  
EXCEPTION  
when ZERO_DIVIDE then  
    dbms_output.put_line('Division by 0 is not possible');  
END;
```

Output:

Statement processed.

Division by 0 is not possible

Numbered Exception Handling

In oracle, some of the pre-defined exceptions are numbered in the form of **four integers preceded by a hyphen symbol**. To handle such exceptions we should assign a name to them before using them.

This can be done by using the **Pragma exception technique** in which a numbered exception handler is bound to a name. For this purpose, we use a keyword in PL/SQL program and write a statement that binds a name to a numbered exception using the following syntax and this statement is written in the **DECLARE** section of program:

```
pragma exception_init(exception_name, exception _number);
```

where, `pragma exception_init(case doesn't matter)` is a keyword indicating Pragma exception technique with two arguments:

- **exception_name**, which is a user-defined name given to a predefined numbered exception if it occurs.
- **exception_number**, is the number allotted to the exception by oracle.

For example:

ROLLNO	SNAME	AGE	COURSE
11	Anu	20	BSC
12	Asha	21	BCOM
13	Arpit	18	BCA
14	Chetan	20	BCA

Prepared by

Mr. N. I. Bhopale
(Subject Teacher)

Verified by

Dr. B. S. Agarkar
(HOD, Deptt. of ECE)

Sanjivani Rural Education Society's College of Engineering, Kopargaon
Department of Electronics and Computer Engineering
TITLE: Experiment Write-up (EW)

15	Nihal	19	BBA
----	-------	----	-----

Code:

```
DECLARE
    sno student.rollno%type;
    snm student.sname%type;
    s_age student.age%type;
    cr student.course%type;
    -- Exception name declared below
    already_exist EXCEPTION;
    -- pragma statement to provide name to numbered exception
    pragma exception_init(already_exist, -1);

BEGIN
    sno:=&rollno;
    snm:='&sname';
    s_age:=&age;
    cr:='&course';
    INSERT into student values(sno, snm, s_age, cr);
    dbms_output.put_line('Record inserted');

    EXCEPTION
        WHEN already_exist THEN
            dbms_output.put_line('Record already exist');
END;
```

Output:

```
Enter the value for sno:11
Enter the value for snm:heena
Enter the value for s_age:20
Enter the value for cr:bsc
Record already exist
PL/SQL procedure successfully completed.
```

User-defined Exception

Prepared by

Mr. N. I. Bhopale
(Subject Teacher)

Verified by

Dr. B. S. Agarkar
(HOD, Deptt. of ECE)

Sanjivani Rural Education Society's College of Engineering, Kopargaon
Department of Electronics and Computer Engineering
TITLE: Experiment Write-up (EW)

ROLLNO	SNAME	Total_Courses
11	Anu	2
12	Asha	1
13	Arpit	3
14	Chetan	1

Code:

```
DECLARE
    sno student.rollno%type;
    snm student.sname%type;
    crno student.total_course%type;
    invalid_total EXCEPTION;

BEGIN
    sno := &rollno;
    snm := '&sname';
    crno:=total_courses;
    IF (crno > 3) THEN
        RAISE invalid_total;
    END IF;
    INSERT into student values(sno, snm, crno);
    EXCEPTION
        WHEN invalid_total THEN
            dbms_output.put_line('Total number of courses cannot be more than 3');
END;
```

Output:

```
Enter the value for sno:15
Enter the value for snm:Akash
Enter the value for crno:5
Total number of courses cannot be more than 3
PL/SQL procedure successfully completed.
```

Prepared by

Mr. N. I. Bhopale
(Subject Teacher)

Verified by

Dr. B. S. Agarkar
(HOD, Deptt. of ECE)

Sanjivani Rural Education Society's College of Engineering, Kopargaon
Department of Electronics and Computer Engineering
TITLE: Experiment Write-up (EW)

NOTE : Please ensure that you also add the Industrial Problem (2) in your submission/document along with the existing content.

References for Theory:

- Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", MGH
- Connally T, Begg C., "Database Systems", Pearson Education
- Raghurama Krishan, "Database Management Systems", McGrawHill
- S.K.Singh, "Database Systems : Concepts, Design and Application", Pearson

OUTPUT: _____

CONCLUSION: _____

Prepared by

Mr. N. I. Bhopale
(Subject Teacher)

Verified by

Dr. B. S. Agarkar
(HOD, Deptt. of ECE)