

Database Systems Laboratory 11

EXCEPTION HANDLER

Exception Handler

Built-in Errors

User-defined
Exceptions

Assigning name and error
number to user-defined
exception

Chittaranjan Pradhan
School of Computer Engineering,
KIIT University

1 Exception Handler

2 Built-in Errors

3 User-defined Exceptions

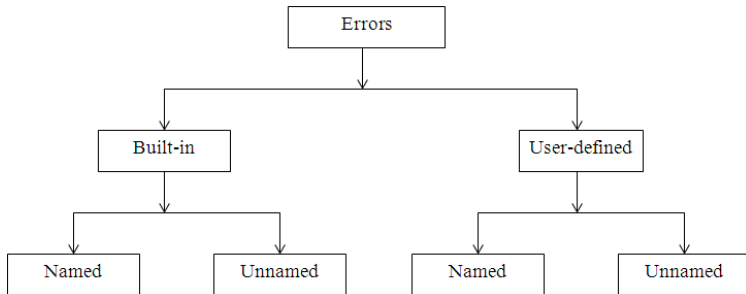
Assigning name and error number to user-defined exception

Exception Handler

Exception Handler

When an exception occurs, the control is passed to the exception handler and error handling is done. Exception handler allows performing final action before it terminates

It is an aid to develop program efficiently. Good PL/SQL program traps the anticipated error using PL/SQL statements and only unanticipated errors are diverted to exception handler



1. Built-in Errors

These errors are pre-defined and are automatically raised by Oracle whenever an error is encountered. Each error is assigned a unique number and a message

Exception Handler

Built-in Errors

User-defined
ExceptionsAssigning name and error
number to user-defined
exception

1.a. Named Errors

These pre-defined errors have been assigned names by Oracle

Error No.	Error Name	Error No.	Error Name
ORA-01403	NO_DATA_FOUND	ORA-06501	PROGRAM_ERROR
ORA-01422	TOO_MANY_ROWS	ORA-06530	ACCESS_INTO_NULL
ORA-01476	ZERO_DIVIDE	ORA-06511	CURSOR_ALREADY_OPEN
ORA-06502	VALUE_ERROR	ORA-00001	DUP_VAL_ON_INDEX
ORA-06500	STORAGE_ERROR	ORA-01001	INVALID_CURSOR
ORA-01017	LOGIN_DENIED	ORA-01722	INVALID_NUMBER
ORA-01012	NOT_LOGGED_ON	ORA-06504	ROWTYPE_MISMATCH
ORA-00051	TIMEOUT_ON_RESOURCE		

Built-in Errors...

1.b. Unnamed Errors

These pre-defined errors have been assigned error number but no name. These are referred to by the general name **OTHERS**. *OTHERS stands for all other exceptions not explicitly named*

- **SQLCODE**: returns a negative error code number
- **SQLERRM**: returns the error message associated with the error code

2	ORA-00000 to ORA-00830
3	ORA-00910 to ORA-01497
4	ORA-01500 to ORA-02098
5	ORA-02140 to ORA-04099
6	ORA-04930 to ORA-07499
7	ORA-07500 to ORA-09859
8	ORA-09870 to ORA-12100
9	ORA-12150 to ORA-12236
10	ORA-12315 to ORA-12354
11	ORA-12400 to ORA-12497
12	ORA-12500 to ORA-12699
13	ORA-12700 to ORA-19361

**Display the employee working in the dept entered by the user
(Incorrect)**

```
DECLARE
    vname EMPLOYEE.name%TYPE;
    vd EMPLOYEE.deptno%TYPE;
BEGIN
    vd: =&vd;
    SELECT name INTO vname FROM EMPLOYEE WHERE
        deptno=vd;
    DBMS_OUTPUT.PUT_LINE(vname);
END;
```

**Display the employee working in the dept entered by the user
(Partially Correct)**

```
DECLARE
    vname EMPLOYEE.name%TYPE;
    vd EMPLOYEE.deptno%TYPE;
BEGIN
    vd: =&vd;
    SELECT name INTO vname FROM EMPLOYEE WHERE
        deptno=vd;
    DBMS_OUTPUT.PUT_LINE(vname);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No such dept with employee');
END;
```

Exception Handler

Built-in Errors

User-defined
Exceptions

Assigning name and error
number to user-defined
exception

Display the employee working in the dept entered by the user
(Correct)

```
DECLARE
    vname EMPLOYEE.name%TYPE;
    vd EMPLOYEE.deptno%TYPE;
BEGIN
    vd: =&vd;
    SELECT name INTO vname FROM EMPLOYEE WHERE
        deptno=vd;
    DBMS_OUTPUT.PUT_LINE(vname);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No such dept with employee');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(SQLCODE||' '||SQLERRM);
END;
```

Exception Handler

Built-in Errors

User-defined
Exceptions

Assigning name and error
number to user-defined
exception

User-defined Exceptions

2. User-defined Exceptions

These are explicitly declared and raised by user

2.a. User-defined Named Exceptions

- Step-1: declare the exception in the DECLARE section
- Step-2: raise the exception in the execution section of the program with **RAISE** statement
- Step-3: write handler for the exception

2.b. User-defined Unnamed Exceptions

Oracle provides a feature wherein we can RAISE exceptions by our own assigned error number and message without specifying a name to it

Error numbers from -20000 to -20999 can be used for user-defined exceptions

User-defined Exceptions...

Write a PL/SQL block which will display the commission of the inputted empno. Handle the exception for negative commission

```
DECLARE
    invalid_comm EXCEPTION;
    vcomm Employee.comm%TYPE;
    veno Employee.empno%TYPE;
BEGIN
    veno: =&veno;
    SELECT comm INTO vcomm FROM Employee WHERE empno=veno;
    IF vcomm<0 THEN
        RAISE invalid_comm;
    ELSE
        DBMS_OUTPUT.PUT_LINE(vcomm);
    END IF;
EXCEPTION
    WHEN invalid_comm THEN
        DBMS_OUTPUT.PUT_LINE('Commission is negative');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No such id');
END;
```

User-defined Exceptions...

Write a PL/SQL block which will display the commission of the inputted empno. Handle the exception for negative commission (User-defined Unnamed Exceptions)

```
DECLARE
    vcomm Employee.comm%TYPE;
    veno Employee.empno%TYPE;
BEGIN
    veno: =&veno;
    SELECT comm INTO vcomm FROM Employee WHERE empno=veno;
    IF vcomm<0 THEN
        RAISE_APPLICATION_ERROR(-20000, 'Commission is negative');
    ELSE
        DBMS_OUTPUT.PUT_LINE(vcomm);
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No such id');
END;
```

Assigning name and error number to user-defined exception

Assigning name and error number to user-defined exception

A user-defined exception can be assigned a name and an error number by using **PRAGMA** pre-compiler directive. This directive binds the specified error number to a user-defined exception name

You can use more than one **PRAGMA EXCEPTION_INIT** directives

The syntax is:

```
exceptionname EXCEPTION;  
PRAGMA EXCEPTION_INIT(exceptionname, errorcode);
```

Assigning name & error number to user-defined exception...

```
DECLARE
    vcomm Employee.comm%TYPE;
    veno Employee.empno%TYPE;
    Invalid_comm EXCEPTION;
    PRAGMA EXCEPTION_INIT(Invalid_comm, -20000);
BEGIN
    veno := &veno;
    SELECT comm INTO vcomm FROM Employee WHERE empno=veno;
    IF vcomm<0 THEN
        RAISE Invalid_comm;
    ELSE
        DBMS_OUTPUT.PUT_LINE(vcomm);
    END IF;
EXCEPTION
    WHEN Invalid_comm THEN
        DBMS_OUTPUT.PUT_LINE(SQLERRM||' '||'Negative commission');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No such id');
END;
```