

由于种种原因，程序在运行时会发生错误，这时系统会显示错误信息，并异常终止程序运行。一个完善的程序应能预见可能发生并需要处理的错误，并编写处理代码，而不是简单的异常终止。

异常是Oracle系统程序执行过程中的一种特殊状态，是由程序发生错误或用户有目的地故意引发的。异常处理代码通常放在PL/SQL的exception代码块中，根据异常产生的机制和原理，可将Oracle异常分为两大类。

1 预定义异常

1.1 预定义异常介绍

Oracle系统提供的，可在PL/SQL中使用的预定义异常，对于这种异常，用户无须再定义，它们由Oracle自动引发。

1.2 常见的预定义异常

错误名称	错误代码	错误含义
CURSOR_ALREADY_OPEN	ORA_06511	试图打开已经打开的游标
INVALID_CURSOR	ORA_01001	试图使用没有打开的游标
DUP_VAL_ON_INDEX	ORA_00001	保存重复值到唯一索引约束的列
ZERO_DIVIDE	ORA_01476	发生除数为零的除法错误
INVALID_NUMBER	ORA_01722	试图对无效字符进行数值转换
ROWTYPE_MISMATCH	ORA_06504	主变量和游标的类型不兼容
VALUE_ERROR	ORA_06502	转换、截断或算术运算发生错误
TOO_MANY_ROWS	ORA_01422	select...into...语句返回多行数据
NO_DATA_FOUND	ORA_01403	select...into...语句返回零行数据
TIMEOUT_ON_RESOURCE	ORA_00051	等待资源时发生超时错误
TRANSACTION_BACKED_OUT	ORA_00060	由于死锁，提交失败
STORAGE_ERROR	ORA_06500	发生内存错误
PROGRAM_ERROR	ORA_06501	发生PL/SQL内部错误
NOT_LOGGED_ON	ORA_01012	试图操作未连接的数据库
LOGIN_DENIED	ORA_01017	在连接时提供了无效用户名或口令

1.3 示例

1.3.1 展示系统预定义异常

```

DECLARE
  v_empno scott.emp.empno%TYPE;
  v_ename scott.emp.ename%TYPE;
  v_job   scott.emp.job%TYPE;
BEGIN
  SELECT empno, ename, job
     INTO v_empno, v_ename, v_job
    FROM scott.emp
   WHERE empno = 8888;

  dbms_output.put_line('EmpNO:' || v_empno || ' EName:' || v_ename ||
    ' Job:' || v_job);

EXCEPTION
  WHEN no_data_found THEN
    dbms_output.put_line('No data .');
END;

```

1.3.2 展示错误信息

```

DECLARE
  v_n1 NUMBER := 88;
  v_n2 NUMBER := 0;

  v_result NUMBER;
BEGIN
  v_result := v_n1 / v_n2;

  dbms_output.put_line('Result is : ' || v_result);

EXCEPTION
  WHEN OTHERS THEN
    dbms_output.put_line('SQL Code is ' || SQLCODE);
    dbms_output.put_line('SQL Err is ' || SQLERRM);

```

END;

2 自定义异常

2.1 自定义异常介绍

自己定义异常类型，可以在声明部分定义新的异常类型，定义的语法：

```
错误名 EXCEPTION;
```

用户定义的错误不能由系统来触发，必须由程序显式的触发，触发语法：

```
RAISE 错误名;
```

RAISE也可以用来引发模拟系统错误，比如，RAISE ZERO_DIVIDE将引发模拟除零错误（除数为零）。

使用RAISE_APPLICATION_ERROR函数也可以引发异常。该函数要传递两个参数，第一个参数是用户自定义的错误编号，第二个参数是用户自定义的错误信息。使用该函数的异常编号应该在20000和20999之间。

2.2 示例

2.2.1 使用RAISE_APPLICATION_ERROR函数引发系统错误

```
DECLARE
  v_no NUMBER(10) := 8888;
BEGIN
  INSERT INTO scott.emp (empno, ename) VALUES (v_no, 'Jims');
  IF v_no < 0 THEN
    ROLLBACK;
    raise_application_error(-20001, '员工No不能小于0');
  ELSIF v_no > 8000 THEN
    ROLLBACK;
    raise_application_error(-20002, '员工No不能大于8000');
  ELSE
    COMMIT;
  END IF;
```

END;

2.2.2 自定义异常

DECLARE

exp_user_err EXCEPTION;

v_ename scott.emp.ename%TYPE;

BEGIN

BEGIN

SELECT ename INTO v_ename FROM scott.emp WHERE empno = 8888;

EXCEPTION

WHEN OTHERS THEN

RAISE exp_user_err;

END;

dbms_output.put_line('ENAME is : ' || v_ename);

EXCEPTION

WHEN exp_user_err THEN

dbms_output.put_line('There is no emp data .');

WHEN OTHERS THEN

dbms_output.put_line('SQL Code is ' || SQLCODE);

dbms_output.put_line('SQL Err is ' || SQLERRM);

END;