

Oracle概念:异常和游标Oracle认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/587/2021_2022_Oracle_E6_A6_82_E5_c102_587107.htm 异常和游标管理 游标：用来查询数据库，获取记录集合(结果集)的指针，可以让开发者一次访问一行结果集，在每条结果集上作操作。分类：静态游标：

分为显式游标和隐式游标。REF游标：是一种引用类型，类似于指针。显式游标：CURSOR 游标名(参数)[返回值类型] IS Select 语句 生命周期：1. 打开游标(OPEN)：解析，绑定。。不会从数据库检索数据 2. 从游标中获取记录(FETCH INTO)：执行查询，返回结果集。通常定义局域变量作为从游标获取数据的缓冲区。 3. 关闭游标(CLOSE)完成游标处理，用户不能从游标中获取行。还可以重新打开。

选项：参数和返回类型 set serveroutput on declare cursor

```
emp_cur ( p_deptid in number) is 0select * from employees where department_id = p_deptid. l_emp employees%rowtype. begin dbms_output.put_line( ' Getting employees from department 30 ' ). open emp_cur(30). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' || l_emp.last_name). end loop. close emp_cur. dbms_output.put_line( ' Getting employees from department 90 ' ). open emp_cur(90). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' ||
```

emp_cur (p_deptid in number) is 0select * from employees where department_id = p_deptid. l_emp employees%rowtype. begin dbms_output.put_line(' Getting employees from department 30 '). open emp_cur(30). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line(' Employee id ' || l_emp.employee_id || ' is '). dbms_output.put_line(l_emp.first_name || ' ' || l_emp.last_name). end loop. close emp_cur. dbms_output.put_line(' Getting employees from department 90 '). open emp_cur(90). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line(' Employee id ' || l_emp.employee_id || ' is '). dbms_output.put_line(l_emp.first_name || ' ' ||

```
emp_cur ( p_deptid in number) is 0select * from employees where department_id = p_deptid. l_emp employees%rowtype. begin dbms_output.put_line( ' Getting employees from department 30 ' ). open emp_cur(30). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' || l_emp.last_name). end loop. close emp_cur. dbms_output.put_line( ' Getting employees from department 90 ' ). open emp_cur(90). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' ||
```

```
emp_cur ( p_deptid in number) is 0select * from employees where department_id = p_deptid. l_emp employees%rowtype. begin dbms_output.put_line( ' Getting employees from department 30 ' ). open emp_cur(30). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' || l_emp.last_name). end loop. close emp_cur. dbms_output.put_line( ' Getting employees from department 90 ' ). open emp_cur(90). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' ||
```

```
emp_cur ( p_deptid in number) is 0select * from employees where department_id = p_deptid. l_emp employees%rowtype. begin dbms_output.put_line( ' Getting employees from department 30 ' ). open emp_cur(30). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' || l_emp.last_name). end loop. close emp_cur. dbms_output.put_line( ' Getting employees from department 90 ' ). open emp_cur(90). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' ||
```

```
emp_cur ( p_deptid in number) is 0select * from employees where department_id = p_deptid. l_emp employees%rowtype. begin dbms_output.put_line( ' Getting employees from department 30 ' ). open emp_cur(30). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' || l_emp.last_name). end loop. close emp_cur. dbms_output.put_line( ' Getting employees from department 90 ' ). open emp_cur(90). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' ||
```

```
emp_cur ( p_deptid in number) is 0select * from employees where department_id = p_deptid. l_emp employees%rowtype. begin dbms_output.put_line( ' Getting employees from department 30 ' ). open emp_cur(30). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' || l_emp.last_name). end loop. close emp_cur. dbms_output.put_line( ' Getting employees from department 90 ' ). open emp_cur(90). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' ||
```

```
emp_cur ( p_deptid in number) is 0select * from employees where department_id = p_deptid. l_emp employees%rowtype. begin dbms_output.put_line( ' Getting employees from department 30 ' ). open emp_cur(30). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' || l_emp.last_name). end loop. close emp_cur. dbms_output.put_line( ' Getting employees from department 90 ' ). open emp_cur(90). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' ||
```

```
emp_cur ( p_deptid in number) is 0select * from employees where department_id = p_deptid. l_emp employees%rowtype. begin dbms_output.put_line( ' Getting employees from department 30 ' ). open emp_cur(30). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' || l_emp.last_name). end loop. close emp_cur. dbms_output.put_line( ' Getting employees from department 90 ' ). open emp_cur(90). loop fetch emp_cur into l_emp. exit when emp_cur%notfound. dbms_output.put_line( ' Employee id ' || l_emp.employee_id || ' is ' ). dbms_output.put_line(l_emp.first_name || ' ' ||
```

l_emp.last_name). end loop. close emp_cur. end. / 隐式游标：不用明确建立游标变量，分两种：1. 在PL/SQL中使用DML语言，使用ORACLE提供的名为SQL的隐式游标 2. CURSOR FOR LOOP，用于for loop 语句。1 举例：declare begin 0update departments set department_name=department_name. --where 1=2. dbms_output.put_line(' 0update ' || sql%rowcount || ' records '). end. / 2 举例：declare begin for my_dept_rec in (0select department_name, department_id from departments) loop dbms_output.put_line(my_dept_rec.department_id || ' : ' || my_dept_rec.department_name). end loop. end. / 游标属性：
%FOUND：变量最后从游标中获取记录的时候，在结果集中找到了记录。 %NOTFOUND：变量最后从游标中获取记录的时候，在结果集中没有找到记录。 %ROWCOUNT：当前时刻已经从游标中获取的记录数量。 %ISOPEN：是否打开。
Declare Cursor emps is Select * from employees where rownum<6 order by 1. Emp employees%rowtype. Row number :=1. Begin Open emps. Fetch emps into emp. Loop If emps%found then Dbms_output.put_line(' Looping over record ' ||row|| ' of ' || emps%rowcount). Fetch emps into emp. Row := row 1. Elself emps%notfound then Exit. ---exit loop, not IF End if. End loop. If emps%isopen then Close emps. End if. End. / 显式和隐式游标的区别：尽量使用隐式游标，避免编写附加的游标控制代码(声明，打开，获取，关闭)，也不需要声明变量来保存从游标中获取的数据。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com