

Oracle数据库游标使用大全 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/167/2021_2022_Oracle_E6_95_B0_E6_c102_167084.htm

SQL是用于访问ORACLE数据库的语言，PL/SQL扩展和加强了SQL的功能，它同时引入了更强的程序逻辑。PL/SQL支持DML命令和SQL的事务控制语句

。DDL在PL/SQL中不被支持，这就意味作在PL/SQL程序块中不能创建表或其他任何对象。较好的PL/SQL程序设计是

在PL/SQL块中使用象DBMS_SQL这样的内建包或执

行EXECUTE IMMEDIATE命令建立动态SQL来执行DDL命令

，PL/SQL编译器保证对象引用以及用户的权限。下面我们将

讨论各种用于访问ORACLE数据库的DDL和TCL语句。查询

SELECT语句用于从数据库中查询数据，当在PL/SQL中使用

SELECT语句时，要与INTO子句一起使用，查询的返回值

被赋予INTO子句中的变量，变量的声明是在DECLCARE中

。SELECT INTO语法如下：SELECT

[DISTICT|ALL]{*|column[,column,...]} INTO

(variable[,variable,...] |record) FROM {table|(sub-query)}[alias]

WHERE..... PL/SQL中SELECT语句只返回一行数据。如果

超过一行数据，那么就要使用显式游标（对游标的讨论我们将在后面进行），INTO子句中要有与SELECT子句中相同列

数量的变量。INTO子句中也可以是记录变量。%TYPE属性

在PL/SQL中可以将变量和常量声明为内建或用户定义的数据

类型，以引用一个列名，同时继承他的数据类型和大小。这种

动态赋值方法是非常有用的，比如变量引用的列的数据类型

和大小改变了，如果使用了%TYPE,那么用户就不必修改代

码，否则就必须修改代码。例：v_empno

```
SCOTT.EMP.EMPNO%TYPE. v_salary EMP.SALARY%TYPE. 不但列名可以使用%TYPE,而且变量、游标、记录，或声明的常量都可以使用%TYPE。这对于定义相同数据类型的变量非常有用。 DECLARE V_A NUMBER(5):=10. V_B V_A%TYPE:=15. V_C V_A%TYPE. BEGIN DBMS_OUTPUT.PUT_LINE (V_A=||V_A||V_B=||V_B||V_C=||V_C). END SQL>/ V_A=10 V_B=15 V_C= PL/SQL procedure successfully completed. SQL>
```

其他DML语句 其它操作数据的DML语句是:INSERT

、UPDATE、DELETE和LOCK TABLE,这些语句在PL/SQL中的语法与在SQL中的语法相同。我们在前面已经讨论过DML语句的使用这里就不再重复了。在DML语句中可以使用任何在DECLARE部分声明的变量，如果是嵌套块，那么要注意变量的作用范围。例：CREATE OR REPLACE PROCEDURE

```
FIRE_EMPLOYEE (pempno in number) AS v_ename  
EMP.ENAME%TYPE. BEGIN SELECT ename INTO v_ename  
FROM emp WHERE empno=p_empno. INSERT INTO  
FORMER_EMP(EMPNO,ENAME) VALUES
```

```
(p_empno,v_ename). DELETE FROM emp WHERE  
empno=p_empno. UPDATE former_emp SET  
date_0deleted=SYSDATE WHERE empno=p_empno.
```

```
EXCEPTION WHEN NO_DATA_FOUND THEN
```

```
DBMS_OUTPUT.PUT_LINE(Employee Number Not Found!).
```

END DML语句的结果 当执行一条DML语句后，DML语句的结果保存在四个游标属性中，这些属性用于控制程序流程或者了解程序的状态。当运行DML语句时，PL/SQL打开一个内

建游标并处理结果，游标是维护查询结果的内存中的一个区域，游标在运行DML语句时打开，完成后关闭。隐式游标只使用SQL%FOUND,SQL%NOTFOUND,SQL%ROWCOUNT三个属性.SQL%FOUND,SQL%NOTFOUND是布尔值，SQL%ROWCOUNT是整数值。SQL%FOUND和SQL%NOTFOUND在执行任何DML语句前SQL%FOUND和SQL%NOTFOUND的值都是NULL,在执行DML语句后，SQL%FOUND的属性值将是：. TRUE :INSERT . TRUE :DELETE和UPDATE，至少有一行被DELETE或UPDATE. . TRUE :SELECT INTO至少返回一行当SQL%FOUND为TRUE时,SQL%NOTFOUND为FALSE。SQL%ROWCOUNT在执行任何DML语句之前，SQL%ROWCOUNT的值都是NULL,对于SELECT INTO语句，如果执行成功，SQL%ROWCOUNT的值为1,如果没有成功，SQL%ROWCOUNT的值为0，同时产生一个异常NO_DATA_FOUND. SQL%ISOPEN SQL%ISOPEN是一个布尔值，如果游标打开，则为TRUE,如果游标关闭，则为FALSE.对于隐式游标而言SQL%ISOPEN总是FALSE，这是因为隐式游标在DML语句执行时打开，结束时就立即关闭。

事务控制语句 事务是一个工作的逻辑单元可以包括一个或多个DML语句，事物控制帮助用户保证数据的一致性。如果事务控制逻辑单元中的任何一个DML语句失败，那么整个事务都将回滚，在PL/SQL中用户可以明确地使用COMMIT、ROLLBACK、SAVEPOINT以及SET TRANSACTION语句。COMMIT语句终止事务，永久保存数据库的变化，同时释放所有LOCK,ROLLBACK终止现行事务释放所有LOCK，但不保存数据库的任何变化,SAVEPOINT用于设置中间点，当事务

调用过多的数据库操作时，中间点是非常有用的，SET TRANSACTION用于设置事务属性，比如read-write和隔离级等。显式游标 当查询返回结果超过一行时，就需要一个显式游标，此时用户不能使用0select into语句。PL/SQL管理隐式游标，当查询开始时隐式游标打开，查询结束时隐式游标自动关闭。显式游标在PL/SQL块的声明部分声明，在执行部分或异常处理部分打开，取数据,关闭。下表显示了显式游标和隐式游标的差别：表1 隐式游标和显式游标 [[The No.1 Picture.]]

使用游标 这里要做一个声明，我们所说的游标通常是指显式游标，因此从现在起没有特别指明的情况，我们所说的游标都是指显式游标。要在程序中使用游标，必须首先声明游标。

声明游标 语法：CURSOR cursor_name IS 0select_statement. 在PL/SQL中游标名是一个未声明变量，不能给游标名赋值或用于表达式中。例：DELCLARE CURSOR C_EMP IS SELECT empno,ename,salary FROM emp WHERE salary>2000 ORDER BY ename. BEGIN 在游标定义中SELECT语句中不一定非要表可以是视图，也可以从多个表或视图中选择的列，甚至可以使用*来选择所有的列。打开游标 使用游标中的值之前应该首先打开游标，打开游标初始化查询处理。打开游标的语法是：OPEN cursor_name cursor_name是在声明部分定义的游标名。例：OPEN C_EMP. 关闭游标 语法：CLOSE cursor_name 例：CLOSE C_EMP. 从游标提取数据 从游标得到一行数据使用FETCH命令。每一次提取数据后，游标都指向结果集的下一行。语法如下：FETCH cursor_name INTO variable[,variable,...] 对于SELECT定义的游标的每一列，FETCH变量列表都应该有一个变量与之相对应，变量的类

型也要相同。例：SET SERVEROUTPUT ON DECLARE v_ename EMP.ENAME%TYPE. v_salary EMP.SALARY%TYPE. CURSOR c_emp IS SELECT ename,salary FROM emp. BEGIN OPEN c_emp. FETCH c_emp INTO v_ename,v_salary. DBMS_OUTPUT.PUT_LINE(Salary of Employee|| v_ename ||is|| v_salary). FETCH c_emp INTO v_ename,v_salary. DBMS_OUTPUT.PUT_LINE(Salary of Employee|| v_ename ||is|| v_salary). FETCH c_emp INTO v_ename,v_salary. DBMS_OUTPUT.PUT_LINE(Salary of Employee|| v_ename ||is|| v_salary). CLOSE c_emp. END 这段代码无疑是非常麻烦的，如果有多行返回结果，可以使用循环并用游标属性为结束循环的条件，以这种方式提取数据，程序的可读性和简洁性都大为提高，下面我们使用循环重新写上面的程序：SET SERVEROUTPUT ON DECLARE v_ename EMP.ENAME%TYPE. v_salary EMP.SALARY%TYPE. CURSOR c_emp IS SELECT ename,salary FROM emp. BEGIN OPEN c_emp. LOOP FETCH c_emp INTO v_ename,v_salary. EXIT WHEN c_emp%NOTFOUND. DBMS_OUTPUT.PUT_LINE(Salary of Employee|| v_ename ||is|| v_salary). END 记录变量 定义一个记录变量使用TYPE命令和%ROWTYPE，关于%ROWTYPE的更多信息请参阅相关资料。记录变量用于从游标中提取数据行，当游标选择很多列的时候，那么使用记录比为每列声明一个变量要方便得多。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