

吴老师教学讲义

ORACLE®

忽然抚尺一下，群响毕绝。撤屏视之，一人、一桌、一椅、一扇、一抚尺而已



吴 青

QQ: 16910735

wuqing_bean@126.com

<http://blog.sina.com/accpwulaoshi>

Oracle PL/SQL 编程

1. 什么是 PL/SQL

PL/SQL (procedural language/sql)是 oracle 在标准的 sql 语言上的扩展。PL/SQL 可以定义变量和常量，允许使用条件判断和循环，还有异常处理。PL/SQL 可以像其它编程语言一样，对数据库进行编程。

我们可以在 Oracle 中使用 PL/SQL 编写过程（也叫存储过程），函数，触发器，它们是在 Oracle 中执行，函数和过程可以在使用其它语言编写的程序中调用。

2. PL/SQL 块

块(block)是 PL/SQL 的基本程序单元 编写 PL/SQL 程序实际就是编写 PL/SQL 块。

2.1 块的结构

PL/SQL 块由三个部分组成：定义部分，执行部分，例外部分。

- 定义部分用于定义常量，变量，游标，例外，复杂数据类型等。
- 执行部分用于实现应用模块功能，该部分包含了要执行的 PL/SQL 语句和 SQL 语句
- 例外部分用于处理执行部分可能出现的运行错误，也就是异常处理

结构如下：

```
DECLARE
/*
*定义部分—定义常量，变量，复杂数据类型，游标
*/
BEGIN
/*
*执行部分—PL/SQL 语句和 SQL 语句
*/
EXCEPTION
/*
*例外处理部分（异常处理）--处理运行错误
*/
END; /*块结束标记*/
```

其中, **DECLARE** 部分是可选的, **BEGIN** 部分是必须的, 而 **EXCEPTION** 部分是可选的, 而 **END** 则是 PL/SQL 块的结束标记。需要注意的是, **DECLARE, BEGIN, EXCEPTION** 后面没有分号(;), 而 **END** 后是必须带有分号(;)。

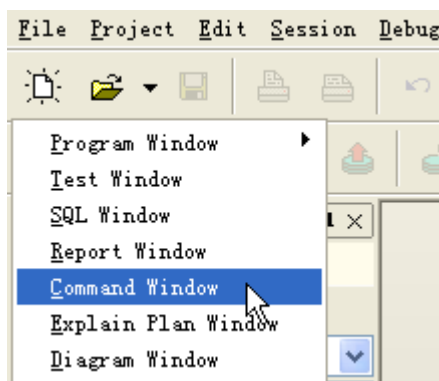
2.2 注释

单行注释: -- (两个横杠)

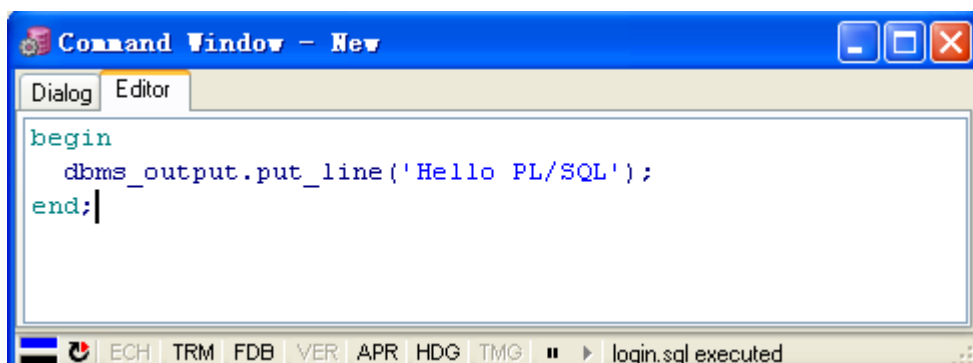
多行注释: /* 注释内容 */

2.3 例子

打开 PLSQL Developer 工具, 我们编写 PL/SQL 主要使用这个工具来开发。使用 SCOTT 用户登录。打开命令行窗口:



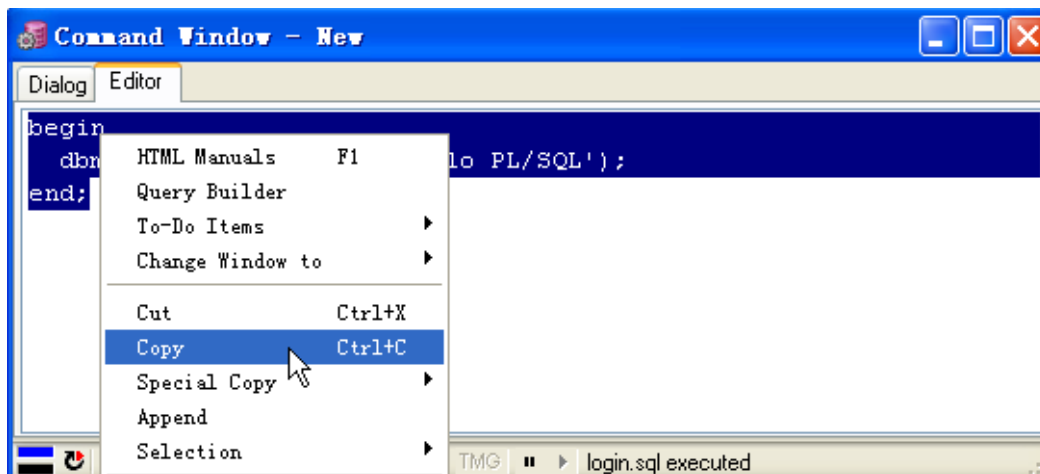
然后在 edit 窗口选项卡中输入如下代码:



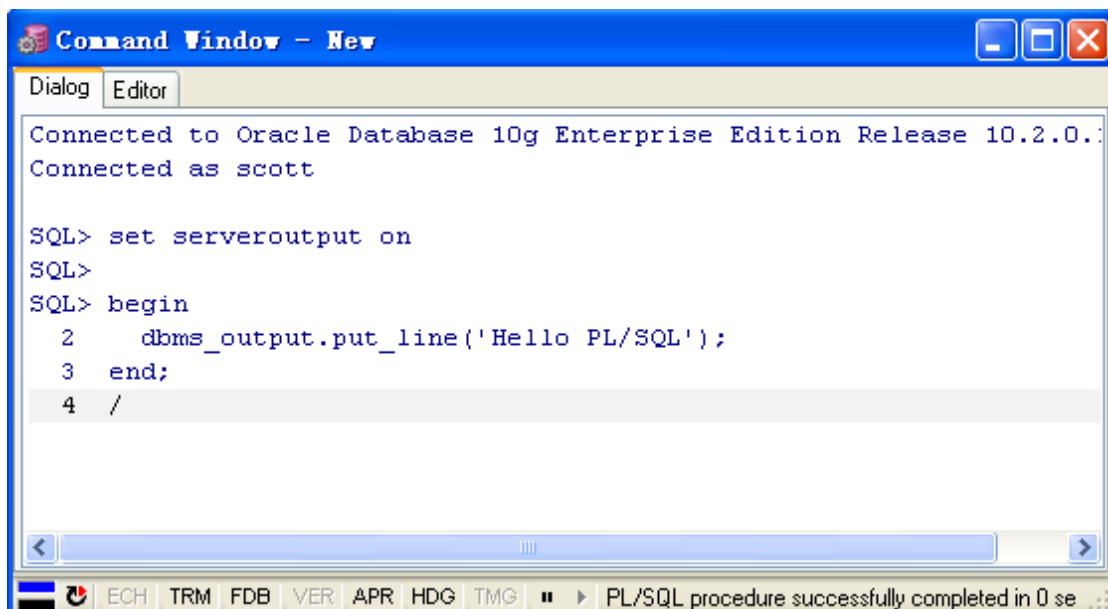
代码解析:这是一个 PL/SQL 块, 这个块省略了 **DECLARE** 部分和 **EXCEPTION** 部分, **dbms_output** 是 Oracle 中的包, 这个包中有一个名称为 **put_line** 的存储过程, 这个存储过程的作用是将信息输出到控制台上。关于 Oracle 包的概念后面会讲到。这里向控制台上输出的信息, 默认情况下, 向控制台输出的信息是不会显示出来的, 需要我们更改环境变量 **serveroutput** 的设置, 将其设置为 **on** 即可, 即:

```
set serveroutput on
```

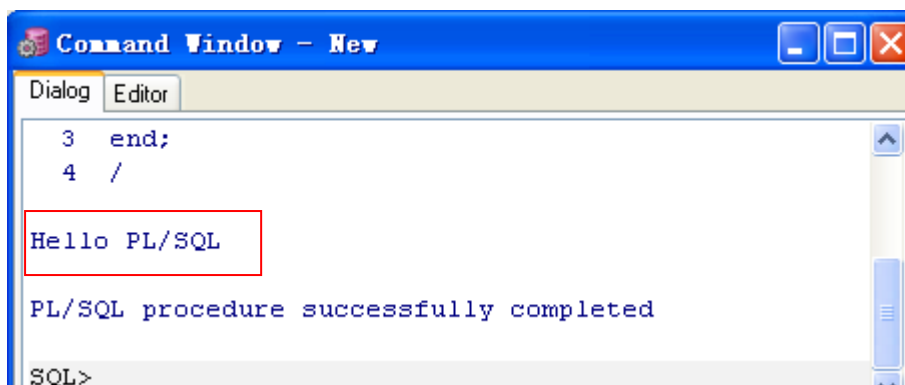
接着复制这段代码:



将复制的代码粘贴到 Dialog 选项卡中,然后在最后面添加一个“/”,这个“/”所代表的意义是执行上面的块:



结果:



3. 定义变量

很显然，定义变量应该在块结构的 declare 部分来定义。

3.1 命名规范

在了解如何定义变量之前，首先了解 PL/SQL 代码编写的规则：

- 定义变量，使用 “v_” 作为前缀 v_sal
- 常量使用 “c_” 开头 如 c_rate
- 定义游标使用 “_cursor” 为后缀
- 定义异常使用 “e_” 为前缀

3.2 定义变量语法

变量或者常量名称 [CONSTANT] 数据类型 [NOT NULL] [:=初始值]

说明:CONSTANT 用于定义常量，定义常量时，必须指定值，并且值不能更改。NOT NULL 用于强制初始化变量，不能为 null，指定 NOT NULL 的时候，必须为变量提供初始值。如果没有为变量指定初始值，那么变量的初始值为 null

如：

```
v_name varchar2(20);  
v_sal number(6,2);  
c_tax_rate CONSTANT number(3,2):=5.5;
```

3.3 例子

```
declare  
  v_ename varchar2(5);  
  v_sal number(6,2);  
  v_eno number(4);  
begin  
  v_eno:=7788; --为v_eno赋值  
  select ename,sal into v_ename,v_sal  
  from emp where empno=v_eno;  
  dbms_output.put_line('雇员名称'||v_ename);  
  dbms_output.put_line('雇员工资'||v_sal);  
end;
```

代码解析:

为变量赋值使用这样的形式: `v_eno:=7788;` 将查询出的结果赋值为变量:
`select ename,sal into v_ename,v_sal from emp where empno=v_eno;`

3.4 使用%TYPE 属性

当定义 PL/SQL 变量存放列值的时候,必须确保变量使用合适的数据类型和长度,否则在运行的时候出现 PL/SQL 运行错误。为了避免这种不必要的错误,可以使用%TYPE 属性定义变量。当使用%TYPE 属性定义变量时,它会按照数据库列或者其他变量来确定新变量的类型和长度:

```
declare
  v_ename emp.ename%type;
  v_sal emp.sal%type;
  v_eno emp.empno%type;
begin
  v_eno:=7788; --为v_eno赋值
  select ename,sal into v_ename,v_sal
  from emp where empno=v_eno;
  dbms_output.put_line('雇员名称'||v_ename);
  dbms_output.put_line('雇员工资'||v_sal);
end;
```

4. 流程控制语法

PL/SQL 中也可以使用判断,循环。

4.1 if 条件语句

```
if 条件 then
  代码;
[elseif 条件 then
  代码;]
[else
  代码]
end if;
```

例子: 取得员工号为 7788 的员工的工资, 如果工资低于 2000, 则为其增加 200 元

```
declare
    v_sal emp.sal%type;
    v_eno emp.empno%type;
begin
    v_eno:=7788;
    select sal into v_sal from emp where empno=v_eno;
    if v_sal<2000 then
        update emp set sal=v_sal+200 where empno=v_eno;
    end if;
end;
```

例子:如果雇员编号为 7788 的雇员补助不是 0, 则在原来的基础上增加 100, 如果补助为 0 或者 null, 则设置其补助为 200

```
declare
    v_comm emp.comm%type;
    v_eno emp.empno%type;
begin
    v_eno:=7788;
    select comm into v_comm from emp where empno=v_eno;
    if v_comm<>0 then
        update emp set comm=v_comm+100 where empno=v_eno;
    else
        update emp set comm=200 where empno=v_eno;
    end if;
end;
```

4.2 case 语句

case 条件选择符

when 条件 then 代码;

when 条件 then 代码;

...

end case;

例子:

```
declare
  v_sal emp.sal%type;
  v_ename emp.ename%type;
  v_eno emp.empno%type;
begin
  v_eno:=7788;
  select ename,sal into v_ename,v_sal from emp where empno=v_eno;
  case
    when v_sal<1000 then
      update emp set comm=100 where ename=v_ename;
    when v_sal<2000 then
      update emp set comm=80 where ename=v_ename;
    when v_sal<6000 then
      update emp set comm=50 where ename=v_ename;
    end case;
end;
```

4.3 基本循环语句

loop

代码;

exit [when 条件]

end loop;

例子: 打印 110

```
declare
  v_i number:=1;
begin
  loop
    dbms_output.put_line(v_i);
    exit when v_i>=10;
    v_i:=v_i+1;
  end loop;
end;
```


4.4 while 循环

while 条件 loop

代码;

end loop;

例子:打印从 1 到 10

```
declare
  v_i number:=1;
begin
  while v_i<=10 loop
    dbms_output.put_line(v_i);
    v_i:=v_i+1;
  end loop;
end;
```

4.5 for 循环

for i in reverse 1..10 loop

代码

end loop;

reverse 表示每次递减, 循环变量从 10 开始,到 1 结束

例子:输出 从 1 到 10:

```
declare
  v_i number;
begin
  for v_i in 1..10 loop
    dbms_output.put_line(v_i);
  end loop;
end;
```

例子:输出 从 10 到 1:

```
declare
  v_i number;
begin
  for v_i in reverse 1..10 loop
    dbms_output.put_line(v_i);
  end loop;
end;
```

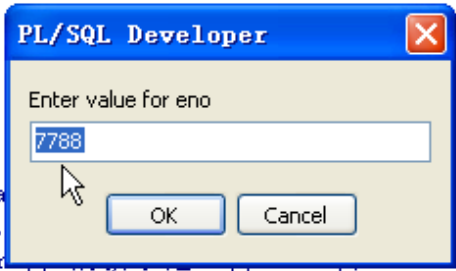
5. 例外（异常处理）

看下面例子:

```
declare
  v_sal emp.sal%type;
  v_ename emp.ename%type;
  v_eno emp.empno%type;
begin
  v_eno:=&eno;
  select ename,sal into v_ename,v_sal from emp where empno=v_eno;
  dbms_output.put_line('测试.....');
  dbms_output.put_line(v_ename||'的薪水是:'||v_sal);
end;
```

&eno 在执行的时候会弹出一个对话框，让输入一个值，这个值给 eno,这个 eno 事先不用定义,如下图，如果输入 7788

```
SQL> declare
  2   v_sal emp.sal%type;
  3   v_ename emp.ename%type;
  4   v_eno emp.empno%type;
  5   begin
  6     v_eno:=&eno;
  7     select ename,sal into v_ename,v_sal from emp where empno=v_eno;
  8     dbms_output.put_line('测试.....');
  9     dbms_output.put_line(v_ename||'的薪水是:'||v_sal);
 10   end;
 11   /
```



则执行结果是:

```
测试.....
SCOTT的薪水是:3000

PL/SQL procedure successfully completed
```

但是如果输入一个不存在的员工编号:7777,则程序就会抛出异常，“测试...”这句话不会输出。因为员工编号不存在，那么姓名和薪水自然也不存在，将不存在的值放入变量 v_ename 和 v_eno 程序就会抛出异常：

```
ORA-01403: 未找到数据
ORA-06512: 在 line 8
```

抛出异常之后，后面的代码不会执行，此时可以使用异常处理，修改代码为:

```
declare
    v_sal emp.sal%type;
    v_ename emp.ename%type;
    v_eno emp.empno%type;
begin
    v_eno:=&eno;
    select ename,sal into v_ename,v_sal from emp where empno=v_eno;

    dbms_output.put_line('测试.....');

    dbms_output.put_line(v_ename || '的薪水是:' || v_sal);

    exception
    when no_data_found then

        dbms_output.put_line('员工编号有误!');
end;
```

这里的 no_data_found 是系统定义好的异常类型，常用预定义例外：

| 例外 | 说明 |
|---------------------|--------------|
| CURSOR_ALREADY_OPEN | 游标已经打开 |
| DUP_VAL_ON_INDEX | 向唯一索引中插入重复数据 |
| INVALID_CURSOR | 非法游标操作 |
| INVALID_NUMBER | 字符向数字转换失败 |
| NO_DATA_FOUND | 没有找到数据 |
| PROGRAM_ERROR | 程序错误 |
| TOO_MANY_ROWS | 返回多行 |
| VALUE_ERROR | 数值转换错误 |
| ZERO_DIVIDE | 被零除 |