

讲师: collen7788@126.com

PLSQL程序设计

本章目标

1

PL/SQL程序结构及组成

2

PL/SQL的应用

最简单的PL/SQL程序

❖ 在屏幕上打印” Hello World!”

●Java程序

```
public static void main(String[] args){  
    System.out.println("Hello World");  
}
```

●PL/SQL程序

```
SQL> declare  
2   begin  
3       dbms_output.put_line('Hello World');  
4   end;  
5   /
```

注意：如果要在屏幕上输出信息，需要将serveroutput开关打开
set serveroutput on

PL/SQL程序结构及组成

❖ 什么是PL/SQL?

- PL/SQL (Procedure Language/SQL)
- PLSQL是Oracle对sql语言的过程化扩展
- 指在SQL命令语言中增加了过程处理语句（如分支、循环等），使SQL语言具有过程处理能力。

PL/SQL的提出

❖ 例1. 为职工长工资，每人长**10%**的工资。

```
Update emp set sal=sal*1.1
```

- SQL优点
- 交互式非过程化；
- 数据操纵功能强；
- 自动导航语句简单；
- 调试容易使用方便。

❖ 例2: 按职工的职称长工资,总裁长**1000**元,经理长**800**元, 其他人员长**400**元。

- 把SQL语言的数据操纵能力与过程语言的数据处理能力结合起来，使得PLSQL面向过程但比过程语言简单、高效、灵活和实用。

❖ Plsql(oracle), Transact-sql(SQL server)

PL/SQL程序结构

declare

说明部分 （变量说明，光标申明，例外说明）

begin

语句序列 （DML语句） ...

exception

例外处理语句

End;

/

变量和常量说明

❖ 说明变量 (char, varchar2, date, number, boolean, long)

```
var1      char(15) ;  
married   boolean :=true ;  
psal      number(7,2);  
my_name   emp.ename%type;  
emp_rec   emp%rowtype;
```

说明变量名、数据类型和长度后用分号结束说明语句。

引用型变量，即my_name的类型与emp表中ename列的类型一样

记录型变量

❖ 记录变量分量的引用：

- emp_rec.ename:='ADAMS';

IF语句

```
1.  IF 条件 THEN 语句1;  
    语句2;  
    END IF;
```

```
2.  IF 条件 THEN 语句序列1;  
    ELSE 语句序列2;  
    END IF;
```

```
3.  IF 条件 THEN 语句;  
    ELSIF 语句 THEN 语句;  
    ELSE 语句;  
    END IF;
```


IF语句示例:

❖ 判断用户输入的数字。

❖ 提示:

- 从键盘输入:

accept num prompt '请输入一个数字';

- 得到键盘输入的值:

pnum number := #

循环语句

```
WHILE total <= 25000  
LOOP  
...  
total := total + salary;  
END LOOP;
```

```
Loop  
EXIT [when 条件];  
.....  
End loop
```

```
FOR I IN 1 .. 3  
LOOP  
语句序列 ;  
END LOOP ;
```

循环语句示例：

❖ 输出数字**1~10**。

示例： 按员工的工种长工资,总裁**1000**元, 经理长**800**元其, 他人员长**400**元。

```
1 set serveroutput on;
2
3 declare
4   ptitle varchar(20); --记录员工的工种
5 begin
6   select job into ptitle from emp; --得到员工的工种
7   --判断员工的工种
8   if ptitle = 'PRESIDENT' then update emp set sal = sal + 1000;
9   elsif ptitle = 'MANAGER' then update emp set sal = sal + 800;
10  else update emp set sal = sal + 400;
11  end if;
12 end;
13 /
```

• 问题1: 返回多行

• 问题2: 没有指定where条件

光标(Cursor)==ResultSet

❖ 说明光标语法:

CURSOR 光标名 [(参数名 数据类型[,参数名 数据类型]...)]
IS SELECT 语句;

❖ 用于存储一个查询返回的多行数据

```
cursor c1 is select ename from emp;
```

❖ 打开光标: **open c1;** (打开光标执行查询)

❖ 取一行光标的值: **fetch c1 into pename;** (取一行到变量中)

❖ 关闭光标: **close c1;**(关闭游标释放资源)

❖ 注意: 上面的**pename**必须与**emp**表中的**ename**列类型一致:

- 定义: **pename emp.ename%type;**

演示: 使用游标查询员工姓名和工资, 并打印

示例： 按员工的工种长工资,总裁**1000**元, 经理长**800**元其, 他人员长**400**元。

```
1 declare
2   cursor c1 is select empno,empjob from emp; --定义光标保存查询结果
3   pno emp.empno% TYPE;
4   pjob emp.empjob% TYPE;
5 begin
6   open c1;--打开游标,即执行查询
7
8   --循环开始
9   loop
10    fetch c1 into pno, pjob; --取一条记录中的员工编号和工种,并付给pno pjob
11    --循环退出条件
12    exit when c1%notfound;
13
14    --判断员工工种,执行加薪
15    if pjob = 'PRESIDENT' then update emp set sal = sal + 1000 where empno = pno;
16    elsif pjob = 'MANAGER' then update emp set sal = sal + 800 where empno = pno;
17    else update emp set sal = sal + 400 where empno = pno;
18    end if;
19
20  end loop;
21
22  close c1;--关闭游标
23  commit; --提交修改
24 end;
25 /
```

1

2

执行select语句并把查询的结果写c1

3

取一条记录处理

4

判断是否取到数据

带参数的光标

```
cursor c2(jobc varchar2)
is
  select ename, sal from emp
 where job=jobc;
```

执行语句:

```
Open c2('clerk');
```

带参数的光标示例：

- ❖ 写一段**PL/SQL**程序，为部门号为**10**的员工涨工资。

例外

- ❖ 例外是程序设计语言提供的一种功能，用来增强程序的健壮性和容错性。

Oracle的异常处理

❖ 系统定义例外

- No_data_found (没有找到数据)
- Too_many_rows (select ...into语句匹配多个行)
- Zero_Divide (被零除)
- Value_error (算术或转换错误)
- Timeout_on_resource (在等待资源时发生超时)

❖ 用户定义的例外

❖ 演示：系统定义例外（被**0**除）

演示：用户定义例外及处理例外

```
DECLARE  
My_job  char(10);  
v_sal   emp.sal%type;  
No_data exception;  
cursor c1 is select  
distinct job from emp  
order by job;
```

```
begin  
open c1;  
Fetch c1 into v_job;  
IF c1%notFOUND then raise no_data;  
end if;  
...  
EXCEPTION  
WHEN no_data THEN insert into emp  
values('fetch语句没有获得数据或数据已  
经处理完');  
END;
```

用户定义例外及处理例外（续）

❖ 在**declare**节中定义例外

- `out_of exception ;`

❖ 在可行语句中引起例外

- `raise out_of ;`

❖ 在**Exception**节处理例外

- `when Out_of then ...`

实例1：统计每年入职的员工个数。

❖ 使用**PL/SQL** 重写。

Total	1980	1981	1982	1987
14	1	10	1	2

实例2：（使用**scott**用户下的**emp**表）

- ❖ 为员工涨工资。从最低工资调起每人长**10%**，但工资总额不能超过**5**万元,请计算长工资的人数和长工资后的工资总额，并输出长工资人数及工资总额。
- ❖ 可能用到的**SQL**语句：
 - `select empno,sal from emp order by sal ;`
 - `select sum(sal) into s_sal from emp;`

实例3:

- ❖ 用**PL/SQL**语言编写一程序，实现按部门分段（**6000**以上、**(6000, 3000)**、**3000**元以下）统计各工资段的职工人数、以及各部门的工资总额（工资总额中不包括奖金），参考如下格式：

部门	小于3000数	3000-6000	大于6000	工资总额
10	2	1	0	8750
20	3	2	0	10875
30	6	0	0	9400
40	0	0	0	0

实例3： （续）

❖ 提示： 可以创建一张新表用于保存数据

```
create table msg1  
(deptno number,  
emp_num1 number,  
emp_num2 number,  
emp_num3 number,  
sum_sal number);
```


讲师: collen7788@126.com

Thank you