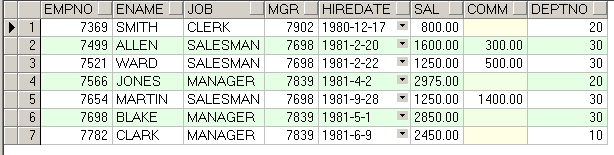
## PL/SQL程序框架基本结构:

Declare --声明变量

Begin --执行代码

End; --结束

## 下图为emp表内容，后面例子主要使用此表：



## 嵌入SQL语句

### 嵌入select语句

嵌入select语句，通过输入的雇员编码(empno)查询雇员的姓名(ename)和工资(sal)

declare

v\_ename emp.ename%type;

v\_sal emp.sal%type;

begin

select emp.ename,emp.sal into v\_ename,v\_sal from emp

where emp.empno=&no;

dbms\_output.put\_line('员工姓名：' || v\_ename);

dbms\_output.put\_line('工资：' || v\_sal);

end;

note: dbms\_output.put\_line() 相当于java的 system.out.println()，

|| 相当于 +

& 表示该值为输入值

### 嵌入insert语句

编写添加员工信息的PL/SQL程序，员工编号(empno)和员工名称(ename)使用参数接受输入

declare

v\_empno emp.empno%type;

v\_ename emp.ename%type;

begin

v\_empno:=&empno;

v\_ename:=&ename;

insert into emp(empno,ename) values (v\_empno,v\_ename);

end;

Note: := 为赋值符，相当于Java中的 =

### 嵌入update语句

编写PL/SQL程序更新根据给定的员工编码(empno)的emp表的员工的工资(sal)，员工编号和工资使用参数输入

declare

v\_empno emp.empno%type;

v\_sal emp.sal%type;

begin

v\_empno:=&empno;

v\_sal:=&sal;

update emp set sal=v\_sal where empno=v\_empno;

end;

### 嵌入delete语句

编写PL/SQL程序删除emp表的员工，员工号(empno)使用参数输入

declare

v\_empno emp.empno%type;

begin

v\_empno:=&empno;

delete from emp where emp.empno=v\_empno;

end;

## PL/SQL控制结构

### 条件语句

#### if条件

依据输入一雇员编号(empno),雇员岗位(job)增加雇员工资(sal)，president更新1000，manager增加500，其他雇员增加200

declare

v\_job emp.job%type;

v\_empno emp.empno%type := &empno;

v\_sal emp.sal%type;

begin

select job,sal into v\_job,v\_sal from emp where empno=v\_empno;

if v\_job=lower('president') then

v\_sal := v\_sal + 1000;

elsif v\_job=lower('manager') then

v\_sal := v\_sal + 500;

else

v\_sal := v\_sal + 200;

end if;

update emp set sal=v\_sal where empno=v\_empno;

end;

#### Case

例：

declare

v\_case number := &v\_case;

begin

case v\_case

when 1 then

dbms\_output.put\_line('It is fun!!!');

when 2 then

dbms\_output.put\_line('It is awesome!!!');

when 3 then

dbms\_output.put\_line('It is great!!!');

end case;

end;

### 循环控制语句

#### 基本循环

使用基本循环计算1-n的累计和

declare

v\_sum int := 0;

n int := &n;

i int := 0;

begin

loop

i := i+1;

v\_sum := v\_sum + i;

exit when i>=n;

end loop;

dbms\_output.put\_line(v\_sum);

end;

#### While循环

使用while循环计算1-n的累计和

declare

v\_sum int := 0;

i int := 0;

n int := &n;

begin

while i<n loop

i := i+1;

v\_sum := v\_sum+i;

end loop;

dbms\_output.put\_line(v\_sum);

end;

#### for循环

使用for循环计算1-n的累计和

方法一：

declare

v\_sum int := 0;

n int := &n;

begin

for i in 1..n loop

v\_sum := v\_sum+i;

end loop;

dbms\_output.put\_line(v\_sum);

end;

方法二：

declare

v\_sum int := 0;

n int := &n;

begin

for i in reverse 1..n loop

v\_sum := v\_sum+i;

end loop;

dbms\_output.put\_line(v\_sum);

end;

#### 嵌套循环和标号

外层循环i取值1-100，内层循环j取值1-100，当i\*j的值为500时，中断内层循环，当i\*j的值为1000时，中断外层循环

declare

ret int;

i int;

j int;

begin

<<outer>> --loop语句的标签

for i in 1..100 loop --1...100

<<inner>>

for j in 1..100 loop

ret:=i\*j;

exit outer when ret=1000;

exit when ret=500;

end loop inner;

dbms\_output.put\_line(ret);

end loop outer;

dbms\_output.put\_line(ret);

end;

## PL/SQL复合数据类型

### PL/SQL记录(存储单行数据)

定义记录变量emp\_record\_type用于存储指定雇员姓名(ename)和工资(sal)

常规定义：

declare

type emp\_record\_type is record(

ename emp.ename%type,

sal emp.sal%type

);

v\_record emp\_record\_type;

begin

select ename,sal into v\_record

from emp where empno=7369;

dbms\_output.put\_line(v\_record.ename);

dbms\_output.put\_line(v\_record.sal);

end;

使用%rowtype定义：

declare

v\_record emp%rowtype;

begin

select \* into v\_record from emp where empno=7369;

dbms\_output.put\_line(v\_record.ename);

dbms\_output.put\_line(v\_record.sal);

end;

### PL/SQL记录表

使用记录表存储emp表的员工姓名（ename）工资（sal）

常规:

declare

type emp\_record\_type is record(

ename emp.ename%type,

sal emp.sal%type

);

type emp\_record\_table is table of emp\_record\_type

index by binary\_integer;

v\_table emp\_record\_table;

begin

select ename,sal bulk collect into v\_table from emp;

for i in 1..v\_table.count loop

dbms\_output.put\_line(v\_table(i).ename || ': ' || v\_table(i).sal);

end loop;

end;

使用%rowtype:

declare

type emp\_record\_table is table of emp%rowtype

index by binary\_integer;

v\_table emp\_record\_table;

begin

select \* bulk collect into v\_table from emp;

for i in 1..v\_table.count loop

dbms\_output.put\_line(v\_table(i).ename || ': ' || v\_table(i).sal);

end loop;

end;