CURSORS

|  |
| --- |
|  |
| 1)declare |
|  | depno number:=&num; |
|  | cursor c\_emp(depno number) is select ename, sal\*(110/100) inc\_sal from emp where deptno= depno; |
|  | v\_emp c\_emp%rowtype; |
|  | begin |
|  | open c\_emp(depno); |
|  | dbms\_output.put\_line('emp from depno'); |
|  | loop |
|  | fetch c\_emp into v\_emp; |
|  | exit when c\_emp%notfound; |
|  | dbms\_output.put\_line(' ename = ' || v\_emp.ename || ' sal = ' || v\_emp.inc\_sal || ' rowcount = ' || c\_emp%rowcount); |
|  | end loop; |
|  | end; |
|  | / |
|  | OUTPUT---- |
|  | Enter value for num: 10 |
|  | old 2: depno number:=&num; |
|  | new 2: depno number:=10; |
|  | emp from depno |
|  | ename = KING sal = 5500 rowcount = 1 |
|  | ename = CLARK sal = 2695 rowcount = 2 |
|  | ename = MILLER sal = 1430 rowcount = 3 |
|  |  |
|  |  |
|  |  |
|  | 2.3)declare |
|  | cursor c\_emp is select \* from emp; |
|  | v\_emp c\_emp%rowtype; |
|  | Begin |
|  | open c\_emp; |
|  | for v\_counter in 1..5 |
|  | loop |
|  | fetch c\_emp into v\_emp; |
|  | dbms\_output.put\_line(' ename = ' || v\_emp.ename || ' sal = ' || v\_emp.sal || ' job = ' || v\_emp.job); |
|  | end loop; |
|  | close c\_emp; |
|  | end; |
|  | / |
|  | OUTPUT |
|  | ename = KING sal = 13182 job = PRESIDENT |
|  | ename = BLAKE sal = 3793.35 job = MANAGER |
|  | ename = CLARK sal = 7579.65 job = MANAGER |
|  | ename = JONES sal = 5140.8 job = MANAGER |
|  | ename = MARTIN sal = 1663.75 job = SALESMAN |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  |  |
|  | 3) |
|  | Declare |
|  | cursor c\_emp is select ename,sal,job from emp where job='MANAGER'; |
|  | v\_emp c\_emp%rowtype; |
|  | Begin |
|  | open c\_emp; |
|  | for v\_counter in 1..3 |
|  | loop |
|  | fetch c\_emp into v\_emp; |
|  | dbms\_output.put\_line('ename=' || v\_emp.ename || 'sal=' ||v\_emp.sal|| 'job=' || v\_emp.job|| 'rowcount='|| c\_emp%rowcount); |
|  | end loop; |
|  | close c\_emp; |
|  | end ; |
|  |  |
|  | OUTPUT--- |
|  | ename=BLAKEsal=3793.35job=MANAGERrowcount=1 |
|  | ename=CLARKsal=7579.65job=MANAGERrowcount=2 |
|  | ename=JONESsal=5140.8job=MANAGERrowcount=3 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  |  |
|  | 4)declare |
|  | p\_job varchar2(10) := '&eno'; |
|  | cursor c\_emp(p\_job varchar2) is select \* from emp where job=p\_job; |
|  | v\_emp c\_emp%rowtype; |
|  | Begin |
|  | open c\_emp(p\_job); |
|  | dbms\_output.put\_line('emp from job'); |
|  | loop |
|  | fetch c\_emp into v\_emp; |
|  | exit when c\_emp%notfound; |
|  | dbms\_output.put\_line(' ename = ' || v\_emp.ename || ' sal = ' ||v\_emp.sal|| ' job = ' || v\_emp.job || ' rowcount = '|| c\_emp%rowcount); |
|  | end loop; |
|  | if c\_emp%isopen then |
|  | dbms\_output.put\_line('closing cursor'); |
|  | close c\_emp; |
|  | end if; |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  |  |
|  | Enter value for eno: CLERK |
|  | old 2: p\_job varchar2(10) := '&eno'; |
|  | new 2: p\_job varchar2(10) := 'CLERK'; |
|  | emp from job |
|  | ename = JAMES sal = 1264.45 job = CLERK rowcount = 1 |
|  | ename = SMITH sal = 1382.4 job = CLERK rowcount = 2 |
|  | ename = ADAMS sal = 1900.8 job = CLERK rowcount = 3 |
|  | ename = MILLER sal = 5053.1 job = CLERK rowcount = 4 |
|  | closing cursor |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | SQL> select empno,sal from emp; |
|  |  |
|  | EMPNO SAL |
|  | ---------- ---------- |
|  | 7839 5000 |
|  | 7698 2850 |
|  | 7782 2450 |
|  | 7566 2975 |
|  | 7654 1250 |
|  | 7499 1600 |
|  | 7844 1500 |
|  | 7900 950 |
|  | 7521 1250 |
|  | 7902 3000 |
|  | 7369 800 |
|  |  |
|  | EMPNO SAL |
|  | ---------- ---------- |
|  | 7788 12000 |
|  | 7876 1100 |
|  | 7934 1300 |
|  |  |
|  | 14 rows selected. |
|  |  |
|  | create trigger tr\_update\_sal after update on emp |
|  | begin |
|  | dbms\_output.put\_line('sal is triggered '); |
|  | end; |
|  | / |
|  |  |
|  | Trigger created. |
|  |  |
|  | SQL> update emp set sal=sal+1000 ; |
|  | sal is triggered |
|  |  |
|  | 14 rows updated. |
|  |  |
|  | SQL> |
|  |  |
|  | -------------------------------SQL> update emp set sal=sal+1000 where empno=7839; |
|  | sal is triggered |
|  |  |
|  | 1 row updated. |
|  |  |
|  | SQL> select sal,empno from emp; |
|  |  |
|  | SAL EMPNO |
|  | ---------- ---------- |
|  | 7000 7839 |
|  | 3850 7698 |
|  | 3450 7782 |
|  | 3975 7566 |
|  | 2250 7654 |
|  | 2600 7499 |
|  | 2500 7844 |
|  | 1950 7900 |
|  | 2250 7521 |
|  | 4000 7902 |
|  | 1800 7369 |
|  |  |
|  | SAL EMPNO |
|  | ---------- ---------- |
|  | 13000 7788 |
|  | 2100 7876 |
|  | 2300 7934 |
|  |  |
|  | 14 rows selected. |
|  |  |
|  | ----------------------------------------------------- |
|  | statement level triggers --- executed once per statement |
|  | for multiple rows ---- use (for each row) |
|  |  |
|  |  |
|  | create trigger t\_update\_sal after update on emp |
|  | for each row |
|  | Begin |
|  | dbms\_output.put\_line('sal is triggered '); |
|  | end; |
|  | / |
|  |  |
|  | update emp set sal=sal+1000 ; |
|  |  |
|  | ----------------------------------------------------- |
|  | create trigger t\_updae\_sal after update or insert or delete on emp |
|  | for each row |
|  | Begin |
|  | dbms\_output.put\_line('sal is triggered '); |
|  | end; |
|  | / |
|  | ----------------------------------------- |
|  |  |
|  | create trigger t\_updates\_sal after update or insert or delete on emp |
|  | for each row |
|  | Begin |
|  | if inserting then |
|  | dbms\_output.put\_line(' inserting '); |
|  | elsif updating then |
|  | dbms\_output.put\_line(' updating row '); |
|  | elsif deleting then |
|  | dbms\_output.put\_line(' deleting row '); |
|  | end if; |
|  | end; |
|  | / |
|  |  |
|  |  |
|  | ------------------------ |
|  | create or replace trigger tr\_update\_sal after update |
|  | on emp |
|  | for each row |
|  |  |
|  | insert into sal\_details values(:old.empno,:new.sal); |
|  | dbms\_output.put\_line('added in sal details'); |
|  | end; |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

CURSORS NOTES

|  |
| --- |
| DAY--2 |
|  |  |
|  | to work with multiple rows cursor is used |
|  | CURSOR DECLARE |
|  | set serveroutput on |
|  | Declare |
|  | cursor c\_emp is select \* from emp; |
|  | v\_emp c\_emp%rowtype; |
|  | Begin |
|  | open c\_emp; |
|  | for v\_counter in 1..10 |
|  | loop |
|  | fetch c\_emp into v\_emp; |
|  | dbms\_output.put\_line(' ename = ' || v\_emp.ename || ' sal = ' || v\_emp.sal); |
|  | end loop; |
|  | close c\_emp; |
|  | end; |
|  | / |
|  | ename=KINGsal=5000 |
|  | ename=BLAKEsal=2850 |
|  | ename=CLARKsal=2450 |
|  | ename=JONESsal=2975 |
|  | ename=MARTINsal=1250 |
|  | ename=ALLENsal=1600 |
|  | ename=TURNERsal=1500 |
|  | ename=JAMESsal=950 |
|  | ename=WARDsal=1250 |
|  | ename=FORDsal=3000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | --------------------------------------------------------------------------------------------------------------------- |
|  | CURSOR ATTRIBUTE - NOTFOUND |
|  | declare |
|  | cursor c\_emp is select \* from emp; |
|  | v\_emp c\_emp%rowtype; |
|  | begin |
|  | open c\_emp; |
|  | loop |
|  | fetch c\_emp into v\_emp; |
|  | exit when c\_emp%notfound; |
|  | dbms\_output.put\_line('ename=' || v\_emp.ename || 'sal=' ||v\_emp.sal|| 'rowcount='|| c\_emp%rowcount); |
|  | end loop; |
|  | close c\_emp; |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT |
|  | ename=KINGsal=5000rowcount=1 |
|  | ename=BLAKEsal=2850rowcount=2 |
|  | ename=CLARKsal=2450rowcount=3 |
|  | ename=JONESsal=2975rowcount=4 |
|  | ename=MARTINsal=1250rowcount=5 |
|  | ename=ALLENsal=1600rowcount=6 |
|  | ename=TURNERsal=1500rowcount=7 |
|  | ename=JAMESsal=950rowcount=8 |
|  | ename=WARDsal=1250rowcount=9 |
|  | ename=FORDsal=3000rowcount=10 |
|  | ename=SMITHsal=800rowcount=11 |
|  | ename=SCOTTsal=3000rowcount=12 |
|  | ename=ADAMSsal=1100rowcount=13 |
|  | ename=MILLERsal=1300rowcount=14 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | -------------------------------------------------------------------------------------------------- |
|  | CLOSING CURSOR |
|  | Declare |
|  | cursor c\_emp is select \* from emp; |
|  | v\_emp c\_emp%rowtype; |
|  | Begin |
|  | open c\_emp; |
|  | loop |
|  | fetch c\_emp into v\_emp; |
|  | exit when c\_emp%notfound; |
|  | dbms\_output.put\_line('ename=' || v\_emp.ename || 'sal=' ||v\_emp.sal|| 'rowcount='|| c\_emp%rowcount); |
|  | end loop; |
|  | if c\_emp%isopen then |
|  | dbms\_output.put\_line('closing cursor'); |
|  | close c\_emp; |
|  | end if; |
|  | end; |
|  | / |
|  |  |
|  | Op |
|  | ename=KINGsal=5000 |
|  | ename=BLAKEsal=2850 |
|  | ename=CLARKsal=2450 |
|  | ename=JONESsal=2975 |
|  | ename=MARTINsal=1250 |
|  | ename=ALLENsal=1600 |
|  | ename=TURNERsal=1500 |
|  | ename=JAMESsal=950 |
|  | ename=WARDsal=1250 |
|  | ename=FORDsal=3000 |
|  | ename=SMITHsal=800 |
|  | ename=SCOTTsal=3000 |
|  | ename=ADAMSsal=1100 |
|  | ename=MILLERsal=1300 |
|  | closing cursor |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | --------------------------------------------------------------------------------- |
|  | Declare |
|  | cursor c\_emp is select \* from emp; |
|  | v\_emp c\_emp%rowtype; |
|  | Begin |
|  | open c\_emp; |
|  | loop |
|  | fetch c\_emp into v\_emp; |
|  | exit when c\_emp%notfound; |
|  | dbms\_output.put\_line('ename=' || v\_emp.ename || 'sal=' ||v\_emp.sal|| 'rowcount='|| c\_emp%rowcount); |
|  | end loop; |
|  | if c\_emp%isopen then |
|  | dbms\_output.put\_line('closing cursor'); |
|  | close c\_emp; |
|  | end if; |
|  | open c\_emp; |
|  | loop |
|  | fetch c\_emp into v\_emp; |
|  | exit when c\_emp%notfound; |
|  | dbms\_output.put\_line('ename=' || v\_emp.ename || 'sal=' ||v\_emp.sal|| 'rowcount='|| c\_emp%rowcount); |
|  | end loop; |
|  | if c\_emp%isopen then |
|  | dbms\_output.put\_line('closing cursor'); |
|  | close c\_emp; |
|  | end if; |
|  | end; |
|  | / |
|  |  |
|  | op |
|  | ename=KINGsal=5000rowcount=1 |
|  | ename=BLAKEsal=2850rowcount=2 |
|  | ename=CLARKsal=2450rowcount=3 |
|  | ename=JONESsal=2975rowcount=4 |
|  | ename=MARTINsal=1250rowcount=5 |
|  | ename=ALLENsal=1600rowcount=6 |
|  | ename=TURNERsal=1500rowcount=7 |
|  | ename=JAMESsal=950rowcount=8 |
|  | ename=WARDsal=1250rowcount=9 |
|  | ename=FORDsal=3000rowcount=10 |
|  | ename=SMITHsal=800rowcount=11 |
|  | ename=SCOTTsal=3000rowcount=12 |
|  | ename=ADAMSsal=1100rowcount=13 |
|  | ename=MILLERsal=1300rowcount=14 |
|  | closing cursor |
|  | ename=KINGsal=5000rowcount=1 |
|  | ename=BLAKEsal=2850rowcount=2 |
|  | ename=CLARKsal=2450rowcount=3 |
|  | ename=JONESsal=2975rowcount=4 |
|  | ename=MARTINsal=1250rowcount=5 |
|  | ename=ALLENsal=1600rowcount=6 |
|  | ename=TURNERsal=1500rowcount=7 |
|  | ename=JAMESsal=950rowcount=8 |
|  | ename=WARDsal=1250rowcount=9 |
|  | ename=FORDsal=3000rowcount=10 |
|  | ename=SMITHsal=800rowcount=11 |
|  | ename=SCOTTsal=3000rowcount=12 |
|  | ename=ADAMSsal=1100rowcount=13 |
|  | ename=MILLERsal=1300rowcount=14 |
|  | closing cursor |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | ---------------------------------------------------------------------------------------------------- |
|  | CURSOR PARAMETER PASSING |
|  | Declare |
|  | cursor c\_emp(p\_deptno number) is select \* from emp where deptno=p\_deptno; |
|  | v\_emp c\_emp%rowtype; |
|  | Begin |
|  | open c\_emp(10); |
|  | dbms\_output.put\_line('emp from deptno 10'); |
|  | loop |
|  | fetch c\_emp into v\_emp; |
|  | exit when c\_emp%notfound; |
|  | dbms\_output.put\_line('ename=' || v\_emp.ename || 'sal=' ||v\_emp.sal|| 'rowcount='|| c\_emp%rowcount); |
|  | end loop; |
|  | if c\_emp%isopen then |
|  | dbms\_output.put\_line('closing cursor'); |
|  | close c\_emp; |
|  | end if; |
|  | open c\_emp(20); |
|  | dbms\_output.put\_line('emp from deptno 20'); |
|  | loop |
|  | fetch c\_emp into v\_emp; |
|  | exit when c\_emp%notfound; |
|  | dbms\_output.put\_line('ename=' || v\_emp.ename || 'sal=' ||v\_emp.sal|| 'rowcount='|| c\_emp%rowcount); |
|  | end loop; |
|  | if c\_emp%isopen then |
|  | dbms\_output.put\_line('closing cursor'); |
|  | close c\_emp; |
|  | end if; |
|  | end; |
|  | / |
|  |  |
|  | op |
|  | emp from deptno 10 |
|  | ename=KINGsal=5000rowcount=1 |
|  | ename=CLARKsal=2450rowcount=2 |
|  | ename=MILLERsal=1300rowcount=3 |
|  | closing cursor |
|  | emp from deptno 20 |
|  | ename=JONESsal=2975rowcount=1 |
|  | ename=FORDsal=3000rowcount=2 |
|  | ename=SMITHsal=800rowcount=3 |
|  | ename=SCOTTsal=3000rowcount=4 |
|  | ename=ADAMSsal=1100rowcount=5 |
|  | closing cursor |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | ----------------------------------------------------------------------------------------------------------------------------------------------------- |
|  | USING FOR LOOP IN CURSOR |
|  | Declare |
|  | cursor c\_emp(p\_deptno number) is select \* from emp where deptno=p\_deptno; |
|  | Begin |
|  | dbms\_output.put\_line('emp from deptno 10'); |
|  | for v\_emp in c\_emp(10) |
|  | loop |
|  | dbms\_output.put\_line('ename=' || v\_emp.ename || 'sal=' ||v\_emp.sal|| 'rowcount='|| c\_emp%rowcount); |
|  | end loop; |
|  | dbms\_output.put\_line('emp from deptno 20'); |
|  | for v\_emp in c\_emp(20) |
|  | loop |
|  | dbms\_output.put\_line('ename=' || v\_emp.ename || 'sal=' ||v\_emp.sal|| 'rowcount='|| c\_emp%rowcount); |
|  | end loop; |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT |
|  | emp from deptno 10 |
|  | ename=KINGsal=5000rowcount=1 |
|  | ename=CLARKsal=2450rowcount=2 |
|  | ename=MILLERsal=1300rowcount=3 |
|  | emp from deptno 20 |
|  | ename=JONESsal=2975rowcount=1 |
|  | ename=FORDsal=3000rowcount=2 |
|  | ename=SMITHsal=800rowcount=3 |
|  | ename=SCOTTsal=3000rowcount=4 |
|  | ename=ADAMSsal=1100rowcount=5 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | ---------------------------------------------------------------------------------------------------------------------------------------------- |
|  | cursor - memory which hold multiple rows |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | IMPLICIT and EXPLICIT CURSORS |
|  |  |
|  | Begin |
|  | update emp set sal=sal+1000 where deptno=10; |
|  | if SQL%found then |
|  | dbms\_output.put\_line('record updated no of rows=' || SQL%rowcount); |
|  | Else |
|  | dbms\_output.put\_line('record not updated'); |
|  | end if; |
|  | end; |
|  | / |
|  |  |
|  | record updated no of rows=3 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | ------------------------------------------------------------------------------------------------------------ |
|  | RECORD LOCKING |
|  |  |
|  | Declare |
|  | cursor c\_emp is select \* from emp for update ; |
|  | Begin |
|  | for v\_emp in c\_emp |
|  | Loop |
|  | if v\_emp.deptno=10 then |
|  | update emp set sal=sal+sal\*0.3 where current of c\_emp; |
|  | dbms\_output.put\_line(' for deptno 10 ename= ' ||v\_emp.ename || ' sal= ' || v\_emp.sal|| ' deptno = ' || v\_emp.deptno ); |
|  | elsif v\_emp.deptno=20 then |
|  | update emp set sal=sal+sal\*0.2 where current of c\_emp; |
|  | dbms\_output.put\_line(' for deptno 20 ename= ' ||v\_emp.ename || ' sal= ' || v\_emp.sal|| ' deptno = ' || v\_emp.deptno ); |
|  | elsif v\_emp.deptno=30 then |
|  | update emp set sal=sal+sal\*0.1 where current of c\_emp; |
|  | dbms\_output.put\_line(' for deptno 10 ename= ' ||v\_emp.ename || ' sal= ' || v\_emp.sal|| ' deptno = ' || v\_emp.deptno ); |
|  | end if; |
|  | end loop; |
|  | end; |
|  | / |
|  |  |
|  |  |
|  | OUTPUT---- |
|  | for deptno 10 ename= KING sal= 10140 deptno = 10 |
|  | for deptno 10 ename= BLAKE sal= 3448.5 deptno = 30 |
|  | for deptno 10 ename= CLARK sal= 5830.5 deptno = 10 |
|  | for deptno 20 ename= JONES sal= 4284 deptno = 20 |
|  | for deptno 10 ename= MARTIN sal= 1512.5 deptno = 30 |
|  | for deptno 10 ename= ALLEN sal= 1936 deptno = 30 |
|  | for deptno 10 ename= TURNER sal= 1815 deptno = 30 |
|  | for deptno 10 ename= JAMES sal= 1149.5 deptno = 30 |
|  | for deptno 10 ename= WARD sal= 1512.5 deptno = 30 |
|  | for deptno 20 ename= FORD sal= 4320 deptno = 20 |
|  | for deptno 20 ename= SMITH sal= 1152 deptno = 20 |
|  | for deptno 20 ename= SCOTT sal= 4320 deptno = 20 |
|  | for deptno 20 ename= ADAMS sal= 1584 deptno = 20 |
|  | for deptno 10 ename= MILLER sal= 3887 deptno = 10 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | SQL> SELECT DEPTNO,SAL FROM EMP; |
|  |  |
|  | DEPTNO SAL |
|  | ---------- ---------- |
|  | 10 13182 |
|  | 30 3793.35 |
|  | 10 7579.65 |
|  | 20 5140.8 |
|  | 30 1663.75 |
|  | 30 2129.6 |
|  | 30 1996.5 |
|  | 30 1264.45 |
|  | 30 1663.75 |
|  | 20 5184 |
|  | 20 1382.4 |
|  |  |
|  | DEPTNO SAL |
|  | ---------- ---------- |
|  | 20 5184 |
|  | 20 1900.8 |
|  | 10 5053.1 |
|  |  |
|  | 14 rows selected. |
|  | ------------------------------------------------------------------------------------------------------- |

DAY1 PLSQL

|  |
| --- |
| 1) set serveroutput on |
|  | Declare |
|  | 2 i number:=&num1; |
|  | 3 j number:=&num2; |
|  | 4 addition number; |
|  | 5 begin |
|  | 6 addition:=i+j; |
|  | 7 dbms\_output.put\_line('addition of number is' ||addition); |
|  | 8 end; |
|  | 9 / |
|  | Enter value for num1: 25 |
|  | old 2: i number:=&num1; |
|  | new 2: i number:=25; |
|  | Enter value for num2: 25 |
|  | old 3: j number:=&num2; |
|  | new 3: j number:=25; |
|  | addition of number is50 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | ----------------------------------------------------------------------------- |
|  |  |
|  | 2)set serveroutput on |
|  | Declare |
|  | 2 v\_name varchar2(20); |
|  | 3 v\_sal number; |
|  | 4 v\_empid number:=&eno; |
|  | 5 begin |
|  | 6 select last\_name,salary into v\_name,v\_sal from employees |
|  | 7 where employee\_id=v\_empid; |
|  | 8 dbms\_output.put\_line('name='||v\_name); |
|  | 9 dbms\_output.put\_line('salary='||v\_sal); |
|  | 10 end; |
|  | 11 / |
|  | Enter value for eno: 100 |
|  | old 4: v\_empid number:=&eno; |
|  | new 4: v\_empid number:=100; |
|  | name=King |
|  | salary=24000 |
|  | ----------------------------------------------------------------------- |
|  | 3)declare |
|  | 2 Rec\_emp employees%rowtype; |
|  | 3 v\_empid number:=&eno; |
|  | 4 begin |
|  | 5 select \* into rec\_emp from employees |
|  | 6 where employee\_id=v\_empid; |
|  | 7 dbms\_output.put\_line('name= '||Rec\_emp.last\_name); |
|  | 8 dbms\_output.put\_line('salary= '||Rec\_emp.salary); |
|  | 9 end; |
|  | 10 / |
|  | Enter value for eno: 101 |
|  | old 3: v\_empid number:=&eno; |
|  | new 3: v\_empid number:=101; |
|  | name= Kochhar |
|  | salary= 17000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | -------------------------------------------------------------------- |
|  |  |
|  | 4) variablename tablename.column%type |
|  | declare |
|  | 2 v\_name employees.last\_name%type; |
|  | 3 v\_sal employees.salary%type; |
|  | 4 v\_empid number:=&eno; |
|  | 5 begin |
|  | 6 select last\_name,salary into v\_name,v\_sal from employees |
|  | 7 where employee\_id=v\_empid; |
|  | 8 dbms\_output.put\_line('name= '||v\_name); |
|  | 9 dbms\_output.put\_line('salary= '||v\_sal); |
|  | 10 end; |
|  | 11 / |
|  | Enter value for eno: 100 |
|  | old 4: v\_empid number:=&eno; |
|  | new 4: v\_empid number:=100; |
|  | name= King |
|  | salary= 24000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | ---------------------------------------------------------------------- |
|  | 5) if condition |
|  | declare |
|  | 2 a number:=&num1; |
|  | 3 begin |
|  | 4 if (mod(a,2)=0) then |
|  | 5 dbms\_output.put\_line(a||' is even number'); |
|  | 6 else |
|  | 7 dbms\_output.put\_line(a||' is odd number'); |
|  | 8 end if; |
|  | 9 end; |
|  | 10 / |
|  | Enter value for num1: 25 |
|  | old 2: a number:=&num1; |
|  | new 2: a number:=25; |
|  | 25 is odd number |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | SQL> r |
|  | 1 declare |
|  | 2 a number:=&num1; |
|  | 3 begin |
|  | 4 if (mod(a,2)=0) then |
|  | 5 dbms\_output.put\_line(a||' is even number'); |
|  | 6 else |
|  | 7 dbms\_output.put\_line(a||' is odd number'); |
|  | 8 end if; |
|  | 9\* end; |
|  | Enter value for num1: 22 |
|  | old 2: a number:=&num1; |
|  | new 2: a number:=22; |
|  | 22 is even number |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | -------------------------------------------------------------------------- |
|  | 6) |
|  | set verify off |
|  | declare |
|  | a number :=&num1; |
|  | b number :=&num2; |
|  | c number :=&num3; |
|  | begin |
|  | if (a>b and a>c) then |
|  | dbms\_output.put\_line(a||' is max number'); |
|  | else if (b>c) then |
|  | dbms\_output.put\_line(b||' is max number'); |
|  | else |
|  | dbms\_output.put\_line(c||' is max number'); |
|  | end if; |
|  | end; |
|  | / |
|  | -------------------------------------------------------------------------------- |
|  |  |
|  | 7)SQL> declare |
|  | 2 i number:=1; |
|  | 3 begin |
|  | 4 loop |
|  | 5 exit when (i>9); |
|  | 6 dbms\_output.put\_line( |
|  | 7 i:=i+1; |
|  | 8 end loop; |
|  | 9 end; |
|  | 10 / |
|  | op----123456789 |
|  |  |
|  | ------------------------------------------------------------------- |
|  | 8) while loop |
|  | 1 declare |
|  | 2 i number:=1; |
|  | 3 begin |
|  | 4 while(i<=5) |
|  | 5 loop |
|  | 6 dbms\_output.put\_line(i); |
|  | 7 i:=i+1; |
|  | 8 end loop; |
|  | 9 end; |
|  | 10 / |
|  | op---12345 |
|  | ------------------------------------------------------------------ |
|  | 9) for loop |
|  | Begin |
|  | for i in 1..6 |
|  | Loop |
|  | dbms\_output.put\_line(i); |
|  | end loop; |
|  | end; |
|  | / |
|  | op—123456 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | ------------------------------------------------------------------- |
|  | 10) begin |
|  | 2 for i in reverse 1..6 |
|  | 3 loop |
|  | 4 dbms\_output.put\_line(i); |
|  | 5 end loop; |
|  | 6 end; |
|  | 7 / |
|  | op--- 654321 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | ---------------------------------------------------------------------- |
|  | 11)SQL> variable maxsal number |
|  | SQL> print maxsal |
|  |  |
|  | MAXSAL |
|  | ---------- |
|  |  |
|  |  |
|  | SQL> begin |
|  | 2 select max(salary) into :maxsal from employees; |
|  | 3 dbms\_output.put\_line('max salary is= '||:maxsal); |
|  | 4 end; |
|  | 5 / |
|  | max salary is= 24000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | SQL> variable maxsal number |
|  | SQL> print maxsal |
|  |  |
|  | MAXSAL |
|  | ---------- |
|  | 24000 |
|  | ------------------------------------------------------------------------------------ |
|  |  |
|  | 12) nesting |
|  |  |
|  | <<outer>> |
|  | Declare |
|  | Str |
|  | ---------------------------------------------------------------------------------------- |

EXCEPTION ASSIGNMNET

|  |
| --- |
|  |
| 1)declare |
|  | sal1 number :=&salary; |
|  | v\_ename varchar2(250); |
|  | Begin |
|  | select ename into v\_ename from emp where sal=sal1 ; |
|  | dbms\_output.put\_line(' name= ' || v\_ename); |
|  | insert into messages values('name of employee is '|| v\_ename); |
|  | Exception |
|  | when too\_many\_rows then |
|  | insert into messages values(' more than one employee with salary of '|| sal1); |
|  | when no\_data\_found then |
|  | insert into messages values(' No employee with salary of' || sal1); |
|  | when others then |
|  | insert into messages values(' some other error occured'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT |
|  | 1)Enter value for salary: 1500 |
|  | old 2: sal1 number :=&salary; |
|  | new 2: sal1 number :=1500; |
|  | name= TURNER |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | SQL> select \* from messages; |
|  |  |
|  | ERR\_MESSAGE |
|  | -------------------------------------------------------------------------------- |
|  | name of employee is TURNER |
|  |  |
|  | 2)Enter value for salary: 3000 |
|  | old 2: sal1 number :=&salary; |
|  | new 2: sal1 number :=3000; |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | SQL> SELECT \* FROM MESSAGES; |
|  |  |
|  | ERR\_MESSAGE |
|  | -------------------------------------------------------------------------------- |
|  | more than one employee with salary of 3000 |
|  | name of employee is TURNER |
|  |  |
|  | 3)Enter value for salary: 21563 |
|  | old 2: sal1 number :=&salary; |
|  | new 2: sal1 number :=21563; |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | SQL> SELECT \* from messages; |
|  |  |
|  | ERR\_MESSAGE |
|  | -------------------------------------------------------------------------------- |
|  | No employee with salary of21563 |
|  | more than one employee with salary of 3000 |
|  | name of employee is TURNER |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | 2) declare |
|  | dept\_no number :=&number; |
|  | dname varchar2(14):='&name'; |
|  | dloc varchar2(13) := '&location'; |
|  | duplicate\_value exception; |
|  | pragma exception\_init(duplicate\_value, -00001); |
|  | Begin |
|  | insert into dept values(dept\_no,dname,dloc); |
|  | exception |
|  | when duplicate\_value then |
|  | dbms\_output.put\_line('dept id already exists ,enter some other value'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Enter value for number: 50 |
|  | old 2: dept\_no number :=&number; |
|  | new 2: dept\_no number :=50; |
|  | Enter value for name: chidy |
|  | old 3: dname varchar2(14):='&name'; |
|  | new 3: dname varchar2(14):='chidy'; |
|  | Enter value for location: Chennai |
|  | old 4: dloc varchar2(13) := '&location'; |
|  | new 4: dloc varchar2(13) := 'chennai'; |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | SQL> select \* from dept; |
|  |  |
|  | DEPTNO DNAME LOC |
|  | ---------- -------------- ------------- |
|  | 50 chidy Chennai |
|  | 10 ACCOUNTING NEW YORK |
|  | 20 RESEARCH DALLAS |
|  | 30 SALES CHICAGO |
|  | 40 OPERATIONS BOSTON |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 3)declare |
|  | v\_name varchar2(10); |
|  | duplicate\_president exception; |
|  | pragma exception\_init(duplicate\_president,-01422); |
|  | Begin |
|  | select job into v\_name from emp where job='CLERK'; |
|  | dbms\_output.put\_line('one president exists'); |
|  | Exception |
|  | when duplicate\_president then |
|  | dbms\_output.put\_line('more than one president'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | one president exists |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | 3.2)declare |
|  | v\_name varchar2(10); |
|  | duplicate\_CLERK exception; |
|  | pragma exception\_init(duplicate\_CLERK,-01422); |
|  | Begin |
|  | select job into v\_name from emp where job='CLERK'; |
|  | dbms\_output.put\_line('one CLERK exists'); |
|  | Exception |
|  | when duplicate\_CLERK then |
|  | dbms\_output.put\_line('more than one CLERK'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | more than one CLERK |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 4. declare |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

EXCEPTION NOTES

|  |
| --- |
|  |
| PREDEFINED EXCEPTIONS |
|  | 1) |
|  | declare |
|  | v\_sal emp.sal%type; |
|  | begin |
|  | select sal into v\_sal from emp where empno=1; |
|  | dbms\_output.put\_line('sal=' || v\_sal); |
|  | exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line(' empno is not available'); |
|  | end; |
|  | / |
|  | OUTPUT |
|  | empno is not available |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 2) |
|  | Declare |
|  | v\_sal emp.sal%type; |
|  | z number; |
|  | Begin |
|  | z:=10/0; |
|  | select sal into v\_sal from emp where empno=1; |
|  | dbms\_output.put\_line('sal=' || v\_sal); |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line(' empno is not available'); |
|  | when zero\_divide then |
|  | dbms\_output.put\_line(' dont try to divide by zero'); |
|  | end; |
|  | / |
|  | OUTPUT |
|  | dont try to divide by zero |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 3) |
|  | Declare |
|  | v\_sal emp.sal%type; |
|  | var number(3); |
|  | z number; |
|  | Begin |
|  | var:=10000; |
|  | z:=10/0; |
|  | select sal into v\_sal from emp where empno=1; |
|  | dbms\_output.put\_line('sal=' || v\_sal); |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line(' empno is not available'); |
|  | when zero\_divide then |
|  | dbms\_output.put\_line(' dont try to divide by zero'); |
|  | when others then |
|  | dbms\_output.put\_line(' exception raised'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | exception raised |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 4) |
|  | Declare |
|  | cursor c\_emp is select \* from emp; |
|  | v\_emp c\_emp%rowtype; |
|  | Begin |
|  | for v\_counter in 1..10 |
|  | loop |
|  | fetch c\_emp into v\_emp; |
|  | dbms\_output.put\_line(' ename = ' || v\_emp.ename || ' sal = ' || v\_emp.sal); |
|  | exception |
|  | when invalid\_cursor then |
|  | dbms\_output.put\_line(' cursor is not opened or closed ') |
|  | end loop; |
|  | close c\_emp; |
|  | end; |
|  | / |
|  |  |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 5) declare |
|  | v\_sal emp.sal%type; |
|  | var number(3); |
|  | z number; |
|  | Begin |
|  | var:=10000; |
|  | z:=10/0; |
|  | select sal into v\_sal from emp where empno=1; |
|  | dbms\_output.put\_line('sal=' || v\_sal); |
|  | Exception |
|  | when no\_data\_found or value\_error then |
|  | dbms\_output.put\_line(' empno is not available or value error'); |
|  | when zero\_divide then |
|  | dbms\_output.put\_line(' dont try to divide by zero'); |
|  | when others then |
|  | dbms\_output.put\_line(' exception raised'); |
|  | end; |
|  | / |
|  |  |
|  |  |
|  | OUTPUT: |
|  | empno is not available or value error |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 6) |
|  | Begin |
|  | Declare |
|  | v\_sal emp.sal%type; |
|  | var number(3); |
|  | z number; |
|  | Begin |
|  | select sal into z from emp; |
|  | var:=10000; |
|  | z:=10/0; |
|  | select sal into v\_sal from emp where empno=1; |
|  | dbms\_output.put\_line('sal=' || v\_sal); |
|  | end; |
|  | Exception |
|  | when no\_data\_found or value\_error then |
|  | dbms\_output.put\_line(' empno is not available or value error'); |
|  | when zero\_divide then |
|  | dbms\_output.put\_line(' dont try to divide by zero'); |
|  | when too\_many\_rows then |
|  | dbms\_output.put\_line(' multiple rows fetched'); |
|  | when others then |
|  | dbms\_output.put\_line(' exception raised'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT |
|  | multiple rows fetched |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 7)begin |
|  | Declare |
|  | v\_sal emp.sal%type; |
|  | var number(3):=10000; |
|  | z number; |
|  | Begin |
|  | select sal into z from emp; |
|  | z:=10/0; |
|  | select sal into v\_sal from emp where empno=1; |
|  | dbms\_output.put\_line('sal=' || v\_sal); |
|  | end; |
|  | Exception |
|  | when no\_data\_found or value\_error then |
|  | dbms\_output.put\_line(' empno is not available or value error'); |
|  | when zero\_divide then |
|  | dbms\_output.put\_line(' dont try to divide by zero'); |
|  | when too\_many\_rows then |
|  | dbms\_output.put\_line(' multiple rows fetched'); |
|  | when others then |
|  | dbms\_output.put\_line(' exception raised'); |
|  | end; |
|  | / |
|  | OUTPUT |
|  | empno is not available or value error |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 8) |
|  | Declare |
|  | var number(3); |
|  | z number; |
|  | Begin |
|  | Begin |
|  | var:=10000; |
|  | dbms\_output.put\_line(' var= '|| var); |
|  | Exception |
|  | when no\_data\_found or value\_error then |
|  | dbms\_output.put\_line(' empno is not available or value error'); |
|  | end; |
|  |  |
|  | z:=10/0; |
|  | dbms\_output.put\_line(' after z'); |
|  | Exception |
|  | when zero\_divide then |
|  | dbms\_output.put\_line(' dont try to divide by zero'); |
|  |  |
|  | when others then |
|  | dbms\_output.put\_line(' exception raised'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT |
|  | empno is not available or value error |
|  | dont try to divide by zero |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 9) |
|  | Declare |
|  | var number(3); |
|  | z number; |
|  | Begin |
|  | Begin |
|  | var:=10000; |
|  | dbms\_output.put\_line(' var= '|| var); |
|  | Exception |
|  | when zero\_divide then |
|  | dbms\_output.put\_line(' empno is not available or value error'); |
|  | end; |
|  |  |
|  | z:=10/0; |
|  | dbms\_output.put\_line(' after z'); |
|  | Exception |
|  | when zero\_divide then |
|  | dbms\_output.put\_line(' dont try to divide by zero'); |
|  |  |
|  | when others then |
|  | dbms\_output.put\_line(' exception raised'); |
|  | end; |
|  | / |
|  |  |
|  | exception raised |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | exception code with name |
|  | 10) |
|  |  |
|  | A)SELECT \* FROM EMP WHERE DEPTNO=10; |
|  |  |
|  | OUTPUT |
|  | EMPNO ENAME JOB MGR HIREDATE SAL COMM |
|  | ---------- ---------- --------- ---------- --------- ---------- ---------- |
|  | DEPTNO |
|  | ---------- |
|  | 7839 KING PRESIDENT 17-NOV-81 5000 |
|  | 10 |
|  |  |
|  | 7782 CLARK MANAGER 7839 09-JUN-81 2450 |
|  | 10 |
|  |  |
|  | 7934 MILLER CLERK 7782 23-JAN-82 1300 |
|  | 10 |
|  |  |
|  | B)begin |
|  | delete dept where deptno=10; |
|  | dbms\_output.put\_line('dept 10 is deleted'); |
|  | end; |
|  |  |
|  | OUTPUT |
|  | Begin |
|  | \* |
|  | ERROR at line 1: |
|  | ORA-02292: integrity constraint (P2007\_18.EMP\_FOREIGN\_KEY) violated – child |
|  | record found |
|  | ORA-06512: at line 2 |
|  |  |
|  | C)declare |
|  | childfound exception; |
|  | pragma exception\_init(childfound,-02292); |
|  | Begin |
|  | delete dept where deptno=10; |
|  | dbms\_output.put\_line('dept 10 is deleted'); |
|  | Exception |
|  | when childfound then |
|  | dbms\_output.put\_line(' emp existing in dept 10'); |
|  | end; |
|  |  |
|  | OUTPUT |
|  | emp existing in dept 10 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | USERDEFINED EXCEPTION |
|  |  |
|  | Declare |
|  | marks number; |
|  | wrong\_marks exception; |
|  | Begin |
|  | marks:=&marks; |
|  | if marks>100 then |
|  | raise wrong\_marks; |
|  | end if; |
|  | dbms\_output.put\_line('ur marks =' || marks); |
|  | Exception |
|  | when wrong\_marks then |
|  | dbms\_output.put\_line(' please enter proper marks which is >=100'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Enter value for marks: 200 |
|  | old 5: marks:=&marks; |
|  | new 5: marks:=200; |
|  | please enter proper marks which is >=100 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | OUTPUT: |
|  | Enter value for marks: 19 |
|  | old 5: marks:=&marks; |
|  | new 5: marks:=19; |
|  | ur marks =19 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | \*\*\*\*enter employee id and amount if it matches the employee salary return salary else return exception\*\*\*\* |
|  |  |
|  | Declare |
|  | empid number :=&num; |
|  | amt number := &amount; |
|  | not\_available exception; |
|  | v\_sal number; |
|  | Begin |
|  | select sal into v\_sal from emp where empno=empid; |
|  | if amt>v\_sal then |
|  | raise not\_available; |
|  | end if; |
|  | dbms\_output.put\_line('amount given is'); |
|  | Exception |
|  | when not\_available then |
|  | dbms\_output.put\_line(' enter the amount less than or equal to sal'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Enter value for num: 7844 |
|  | old 2: empid number :=&num; |
|  | new 2: empid number :=7844; |
|  | Enter value for amount: 2500 |
|  | old 3: amt number := &amount; |
|  | new 3: amt number := 2500; |
|  | enter the amount less than or equal to sal |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | TO DISPLAY ERROR CODE AND MESSAGE |
|  |  |
|  | declare |
|  | var number(3); |
|  | Begin |
|  | var:=10000; |
|  | dbms\_output.put\_line(' var =' || var); |
|  | Exception |
|  | when value\_error then |
|  | dbms\_output.put\_line(' error code = '|| SQLCODE || ' message= ' || SQLERRM); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | error code = -6502 message= ORA-06502: PL/SQL: numeric or value error: number |
|  | precision too large |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  |  |
|  |  |

PROCEDURES AND FUNCTIONS

|  |
| --- |
|  |
| PROCEDURES AND FUNCTIONS |
|  |  |
|  | 1)create or replace procedure display\_message as |
|  | begin |
|  | dbms\_output.put\_line('hello all'); |
|  | end; |
|  | / |
|  | OUTPUT |
|  | SQL> execute display\_message |
|  | hello all |
|  |  |
|  | OUTPUT |
|  |  |
|  | SQL> BEGIN |
|  | 2 DISPLAY\_MESSAGE; |
|  | 3 END; |
|  | 4 / |
|  | hello all |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 2) |
|  | create or replace procedure display\_message as |
|  | v\_sal emp.sal%type; |
|  | Begin |
|  | select sal into v\_sal from emp where empno=7369; |
|  | dbms\_output.put\_line(' sal = ' || v\_sal); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Procedure created. |
|  |  |
|  | SQL> execute display\_message; |
|  | sal = 800 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | OUTPUT: |
|  | SQL> begin |
|  | 2 display\_message; |
|  | 3 end; |
|  | 4 / |
|  | sal = 800 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | EXCEPTION IN Procedure |
|  | 3).A) create or replace procedure display\_message as |
|  | v\_sal emp.sal%type; |
|  | Begin |
|  | select sal into v\_sal from emp where empno=7; |
|  | dbms\_output.put\_line(' sal = ' || v\_sal); |
|  | end; |
|  |  |
|  | OUTPUT: |
|  | SQL> execute display\_message; |
|  | BEGIN display\_message; END; |
|  |  |
|  | \* |
|  | ERROR at line 1: |
|  | ORA-01403: no data found |
|  | ORA-06512: at "P2007\_18.DISPLAY\_MESSAGE", line 4 |
|  | ORA-06512: at line 1 |
|  |  |
|  | B)create or replace procedure display\_message as |
|  | v\_sal emp.sal%type; |
|  | Begin |
|  | select sal into v\_sal from emp where empno=7; |
|  | dbms\_output.put\_line(' sal = ' || v\_sal); |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('empno not available'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Procedure created. |
|  |  |
|  | SQL> execute display\_message; |
|  | empno not available |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | IF CONDITION |
|  |  |
|  | 4) create or replace procedure display\_message as |
|  | v\_sal emp.sal%type; |
|  | Begin |
|  | select sal into v\_sal from emp where empno=7839; |
|  | dbms\_output.put\_line(' sal = ' || v\_sal); |
|  | if v\_sal>20000 then |
|  | dbms\_output.put\_line(' greater than 20000'); |
|  | Else |
|  | dbms\_output.put\_line(' less than 20000'); |
|  | end if; |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('empno not available'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  |  |
|  | Procedure created. |
|  |  |
|  | SQL> execute display\_message |
|  | sal = 5000 |
|  | less than 20000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | PARAMETER IN PROCEDURE |
|  | READ ONLY (IN) DEFAULT |
|  | 5) |
|  | create or replace procedure get\_sal\_info (p\_empno number) as |
|  | v\_sal emp.sal%type; |
|  | Begin |
|  | select sal into v\_sal from emp where empno=p\_empno; |
|  | dbms\_output.put\_line(' sal = ' || v\_sal); |
|  | if v\_sal>20000 then |
|  | dbms\_output.put\_line(' greater than 20000'); |
|  | Else |
|  | dbms\_output.put\_line(' less than 20000'); |
|  | end if; |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('empno not available'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | SQL> execute get\_sal\_info(7844); |
|  | sal = 1500 |
|  | less than 20000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | OUTPUT: |
|  | SQL> execute get\_sal\_info(7788); |
|  | sal = 3000 |
|  | less than 20000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | OUTPUT: |
|  | SQL> begin |
|  | 2 get\_sal\_info(7876); |
|  | 3 end; |
|  | 4 / |
|  | sal = 1100 |
|  | less than 20000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | OUTPUT: |
|  | SQL> declare |
|  | 2 v\_empno emp.empno%type:=7499; |
|  | 3 begin |
|  | 4 get\_sal\_info(v\_empno); |
|  | 5 end; |
|  | 6 / |
|  | sal = 1600 |
|  | less than 20000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | 6) |
|  | create or replace procedure get\_emp\_info (p\_deptno number) as |
|  | cursor c\_emp(p\_dept number) is select \* from emp where deptno=p\_dept; |
|  | Begin |
|  | dbms\_output.put\_line('emp from deptno '); |
|  | for v\_emp in c\_emp(p\_deptno) |
|  | loop |
|  | dbms\_output.put\_line('ename=' || v\_emp.ename || 'sal=' || v\_emp.sal); |
|  | end loop; |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('empno not available'); |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | Procedure created. |
|  |  |
|  | SQL> execute get\_emp\_info(10); |
|  | emp from deptno |
|  | ename=KINGsal=5000 |
|  | ename=CLARKsal=2450 |
|  | ename=MILLERsal=1300 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | WRITE ONLY (OUT): |
|  | 7). |
|  | create or replace procedure get\_sal\_info (p\_empno number,p\_sal out number) as |
|  | Begin |
|  | select sal into p\_sal from emp where empno=p\_empno; |
|  | dbms\_output.put\_line(' sal = ' || p\_sal); |
|  | if p\_sal>20000 then |
|  | dbms\_output.put\_line(' greater than 20000'); |
|  | Else |
|  | dbms\_output.put\_line(' less than 20000'); |
|  | end if; |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('empno not available'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Procedure created. |
|  |  |
|  | Declare |
|  | var\_sal emp.sal%type; |
|  | Begin |
|  | get\_sal\_info(7788, var\_sal); |
|  | dbms\_output.put\_line(' sal from caller block = ' ||var\_sal); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | sal = 3000 |
|  | less than 20000 |
|  | sal from caller block = 3000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | IN - DEFAULT ,no need to mention parameter as in, read only, reads variable value expressions. |
|  | OUT- mention out along with parameter, write only, writes only variable |
|  | IN OUT - mention in out along with parameter, read and write, writes only variable |
|  | 8). |
|  | A)create or replace procedure get\_sal\_info (p\_empno number,p\_sal in out number) as |
|  | Begin |
|  | dbms\_output.put\_line(' sal in procedure before modification = ' || p\_sal); |
|  | select sal into p\_sal from emp where empno=p\_empno; |
|  | dbms\_output.put\_line(' sal = ' || p\_sal); |
|  | if p\_sal>20000 then |
|  | dbms\_output.put\_line(' greater than 20000'); |
|  | Else |
|  | dbms\_output.put\_line(' less than 20000'); |
|  | end if; |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('empno not available'); |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | Procedure created. |
|  |  |
|  |  |
|  | Declare |
|  | var\_sal emp.sal%type := 1000; |
|  | Begin |
|  | get\_sal\_info(7788, var\_sal); |
|  | dbms\_output.put\_line(' sal from caller block = ' ||var\_sal); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | sal in procedure before modification = 1000 |
|  | sal = 3000 |
|  | less than 20000 |
|  | sal from caller block = 3000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | B)create or replace procedure get\_sal\_info (p\_empno number,p\_sal out number) as |
|  | Begin |
|  | dbms\_output.put\_line(' sal in procedure before modification = ' || p\_sal); |
|  | select sal into p\_sal from emp where empno=p\_empno; |
|  | dbms\_output.put\_line(' sal = ' || p\_sal); |
|  | if p\_sal>20000 then |
|  | dbms\_output.put\_line(' greater than 20000'); |
|  | Else |
|  | dbms\_output.put\_line(' less than 20000'); |
|  | end if; |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('empno not available'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Procedure created. |
|  |  |
|  | Declare |
|  | var\_sal emp.sal%type := 1000; |
|  | Begin |
|  | get\_sal\_info(7788, var\_sal); |
|  | dbms\_output.put\_line(' sal from caller block = ' ||var\_sal); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | sal in procedure before modification = |
|  | sal = 3000 |
|  | less than 20000 |
|  | sal from caller block = 3000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | FUNCTIONS - RETURNS A VALUE, |
|  | PROCEDURE - DOESNT RETURN , TO RETURN USING OUT |
|  |  |
|  | 1) create or replace function get\_sal (p\_empno number) return number as |
|  | v\_sal emp.sal%type; |
|  | Begin |
|  | select sal into v\_sal from emp where empno=p\_empno; |
|  | dbms\_output.put\_line(' sal = ' || v\_sal); |
|  | if v\_sal>20000 then |
|  | dbms\_output.put\_line(' greater than 20000'); |
|  | Else |
|  | dbms\_output.put\_line(' less than 20000'); |
|  | end if; |
|  | return v\_sal; |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('empno not available'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Function created. |
|  |  |
|  | Declare |
|  | var\_sal emp.sal%type; |
|  | Begin |
|  | var\_sal:=get\_sal (7788); |
|  | dbms\_output.put\_line(' sal from caller block = ' ||var\_sal); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | sal = 3000 |
|  | less than 20000 |
|  | sal from caller block = 3000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | select query can be used in functions not in procedures |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | FUNCTION INTO PROCEDURE |
|  |  |
|  | create or replace function get\_sal (p\_empno number) return number as |
|  | v\_sal emp.sal%type; |
|  | Begin |
|  | select sal into v\_sal from emp where empno=p\_empno; |
|  | dbms\_output.put\_line(' sal = ' || v\_sal); |
|  | if v\_sal>20000 then |
|  | dbms\_output.put\_line(' greater than 20000'); |
|  | Else |
|  | dbms\_output.put\_line(' less than 20000'); |
|  | end if; |
|  | return v\_sal; |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('empno not available'); |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | Function created. |
|  |  |
|  | Declare |
|  | var\_sal emp.sal%type; |
|  | Begin |
|  | var\_sal:=get\_sal (7788); |
|  | dbms\_output.put\_line(' sal from caller block = ' ||var\_sal); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | sal = 7000 |
|  | less than 20000 |
|  | sal from caller block = 7000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | ------------------------------------------------------------ |
|  | select get\_sal(7788) from dual; |
|  |  |
|  | OUTPUT: |
|  | GET\_SAL(7788) |
|  | ------------- |
|  | 7000 |
|  |  |
|  | sal = 7000 |
|  | less than 20000 |
|  | -------------------------------------------------------------- |
|  | create or replace procedure sal\_dis( p\_empno emp.empno%type) as |
|  | v\_sal emp.sal%type; |
|  | Begin |
|  | v\_sal:=get\_sal(p\_empno); |
|  | if v\_sal>10000 then |
|  | update emp set sal=sal+1000 where empno=p\_empno; |
|  | Else |
|  | update emp set sal=sal+2000 where empno=p\_empno; |
|  | end if; |
|  | end; |
|  | / |
|  | --------------------------------------- |
|  | OR(Calling Function In IF statement) |
|  |  |
|  | create or replace procedure sal\_dis( p\_empno emp.empno%type) as |
|  | Begin |
|  | if get\_sal(p\_empno)>10000 then |
|  | update emp set sal=sal+1000 where empno=p\_empno; |
|  | Else |
|  | update emp set sal=sal+2000 where empno=p\_empno; |
|  | end if; |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  |  |
|  | Procedure created. |
|  |  |
|  | ---> |
|  | execute sal\_dis(7788); |
|  | OUTPUT: |
|  | sal = 7000 |
|  | less than 20000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | ----> |
|  | execute sal\_dis(7788); |
|  | OUTPUT: |
|  | sal = 9000 |
|  | less than 20000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | FUNCTION RETURN BOOLEAN VALUE |
|  | CAN BE CALLED INSIDE IF |
|  |  |
|  | create or replace function get\_sal (p\_empno number) return boolean as |
|  | v\_sal emp.sal%type; |
|  | Begin |
|  | select sal into v\_sal from emp where empno=p\_empno; |
|  | dbms\_output.put\_line(' sal = ' || v\_sal); |
|  | if v\_sal>20000 then |
|  | dbms\_output.put\_line(' greater than 20000'); |
|  | return true; |
|  | Else |
|  | dbms\_output.put\_line(' less than 20000');: |
|  | return false; |
|  | end if; |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('empno not available'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Function created. |
|  |  |
|  | ANONYMOUS BLOCK |
|  |  |
|  | Begin |
|  | if get\_sal(7788) then |
|  | dbms\_output.put\_line('from caller greater than 20000'); |
|  | Else |
|  | dbms\_output.put\_line('from caller less than 20000'); |
|  | end if; |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | sal = 12000 |
|  | less than 20000 |
|  | from caller less than 20000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  |  |

PROCEDURE AND FUNCTION ASSIGNMENT

|  |
| --- |
| 1) |
|  | A)create or replace procedure USER\_QUERY\_EMP (p\_empno number,p\_sal out number,p\_job out varchar2) as |
|  | Begin |
|  | select sal,job into p\_sal,p\_job from emp where empno=p\_empno; |
|  | dbms\_output.put\_line(' sal = ' || p\_sal || ' job = ' || p\_job); |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('empno not available'); |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | Procedure created. |
|  |  |
|  | B)declare |
|  | var\_sal emp.sal%type; |
|  | var\_job emp.job%type; |
|  | Begin |
|  | USER\_QUERY\_EMP(7788, var\_sal,var\_job); |
|  | dbms\_output.put\_line(' sal from caller block = ' ||var\_sal); |
|  | dbms\_output.put\_line(' job from caller block = ' ||var\_job); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | sal = 3000 job = ANALYST |
|  | sal from caller block = 3000 |
|  | job from caller block = ANALYST |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | 2) |
|  | create or replace function USER\_ANNUAL\_COMP (p\_eno number, p\_sal out number, p\_comm out number) return number as |
|  | compensation number; |
|  | Begin |
|  | select sal,comm into p\_sal,p\_comm from emp where empno=p\_eno; |
|  | dbms\_output.put\_line(' sal = ' || p\_sal || ' comm = ' || p\_comm); |
|  | select sal, comm ,(sal + NVL(comm,0))\*12 into p\_sal, p\_comm, compensation from emp where empno=p\_eno; |
|  | dbms\_output.put\_line(' compensation = ' || compensation); |
|  | return compensation; |
|  | exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('empno not available'); |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | Function created |
|  |  |
|  |  |
|  | Declare |
|  | var\_sal emp.sal%type; |
|  | var\_comm emp.comm%type; |
|  | annual number; |
|  | Begin |
|  | annual:=USER\_ANNUAL\_COMP(7521, var\_sal ,var\_comm); |
|  | dbms\_output.put\_line(' annual = ' || annual); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | sal = 1250 comm = 500 |
|  | compensation = 21000 |
|  | annual = 21000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 3)create or replace function user\_valid\_deptno(p\_dno number) return boolean as |
|  | dep\_no number; |
|  | ct number; |
|  | Begin |
|  | select count(\*) into ct from emp where deptno=p\_dno; |
|  | if ct>0 then |
|  | dbms\_output.put\_line('dept exists'); |
|  | return true; |
|  | Else |
|  | dbms\_output.put\_line('dept not exists'); |
|  | return false; |
|  | end if; |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('error'); |
|  | end; |
|  | / |
|  |  |
|  |  |
|  |  |
|  | create or replace procedure show\_strength(p\_deptno number) as |
|  | ect number; |
|  | Begin |
|  | if user\_valid\_deptno(p\_deptno) then |
|  | select count(\*) into ect from emp where deptno=p\_deptno; |
|  | dbms\_output.put\_line(' enp no ' || ect); |
|  | Else |
|  | dbms\_output.put\_line(' dept no not found '); |
|  | end if; |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  |  |
|  | Procedure created. |
|  |  |
|  | Begin |
|  | show\_strength(10); |
|  | end; |
|  | / |
|  | dept exists |
|  | enp no 3 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | begin |
|  | show\_strength(50); |
|  | end; |
|  | / |
|  | dept not exists |
|  | dept no not found |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 4) |
|  |  |
|  | create or replace procedure SHOW\_RECORDS(p\_join\_date date) as |
|  | cursor c1 is select \* from emp where hiredate>p\_join\_date; |
|  | begin |
|  | dbms\_output.put\_line(' Employees joined after ' || p\_join\_date); |
|  | dbms\_output.put\_line(' ename ' || ' job ' || ' salary ' || 'department '); |
|  | dbms\_output.put\_line('----------------------------------------------------------------'); |
|  | for v1 in c1 |
|  | loop |
|  | dbms\_output.put\_line( v1.ename ||' '|| v1.job ||' '|| v1.sal || ' ' || v1.deptno); |
|  | end loop; |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Procedure created. |
|  | --------------------------------------------- |
|  | declare |
|  | join\_date date :='&date'; |
|  | begin |
|  | SHOW\_RECORDS(join\_date); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Enter value for date: 17-DEC-81 |
|  | old 2: join\_date date :='&date'; |
|  | new 2: join\_date date :='17-DEC-81'; |
|  | Employees joined after 17-DEC-81 |
|  | ename job salary department |
|  | ---------------------------------------------------------------- |
|  | SCOTT ANALYST 12000 20 |
|  | ADAMS CLERK 1100 20 |
|  | MILLER CLERK 1300 10 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 5) |
|  | create or replace procedure ADD\_EMPLOYEE(job\_id varchar2, manager\_id number, hire\_date date, salary number, commission number,department\_id number)as |
|  | select dbms\_random.random from dual; |
|  |  |
|  |  |

PACKAGE

|  |
| --- |
|  |
| PACKAGES |
|  |  |
|  | 1). PACKAGE CREATION: |
|  | create or replace package emp\_pack as |
|  | procedure getempinfo(p\_empno emp.empno%type); |
|  | function getsal(p\_empno emp.empno%type) return number; |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | Package created. |
|  | ---------------------------------------------------------------- |
|  | PACKAGE BODY: |
|  |  |
|  | create or replace package body emp\_pack as |
|  |  |
|  | procedure getempinfo(p\_empno emp.empno%type) as |
|  | v\_name emp.ename%type; |
|  | begin |
|  | select ename into v\_name from emp where empno=p\_empno; |
|  | dbms\_output.put\_line('from getempinfo proc ename = ' || v\_name); |
|  | exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('invalid empno'); |
|  | end; |
|  |  |
|  | function getsal(p\_empno emp.empno%type) return number as |
|  | v\_sal emp.sal%type; |
|  | begin |
|  | select sal into v\_sal from emp where empno=p\_empno; |
|  | dbms\_output.put\_line('from getsal function sal = ' || v\_sal); |
|  | return v\_sal; |
|  | exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('invalid empno'); |
|  | end; |
|  | end; |
|  |  |
|  | OUTPUT: |
|  | Package body created. |
|  | ------------------------------------------------------------- |
|  | PROCEDURE EXECUTION: |
|  |  |
|  | execute emp\_pack.getempinfo(7788); |
|  |  |
|  | OUTPUT: |
|  | from getempinfo proc ename = SCOTT |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | ------------------------------------------------------------- |
|  | FUNCTION EXECUTION: |
|  |  |
|  | declare |
|  | v\_forsal emp.sal%type; |
|  | begin |
|  | v\_forsal:=emp\_pack.getsal(7788); |
|  | dbms\_output.put\_line('sal from caller= '|| v\_forsal); |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | from getsal function sal = 12000 |
|  | sal from caller= 12000 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 2) |
|  | create or replace package emp\_pack as |
|  | procedure getempinfo(p\_empno emp.empno%type); |
|  | function getsal(p\_empno emp.empno%type) return number; |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Package created. |
|  |  |
|  | ------------------------------------------------------------------ |
|  | create or replace package body emp\_pack as |
|  | procedure pri\_proc as |
|  | begin |
|  | dbms\_output.put\_line('in private proc pri\_proc'); |
|  | end; |
|  | procedure getempinfo(p\_empno emp.empno%type) as |
|  | v\_name emp.ename%type; |
|  | begin |
|  | pri\_proc; |
|  | select ename into v\_name from emp where empno=p\_empno; |
|  | dbms\_output.put\_line('from getempinfo proc ename = ' || v\_name); |
|  | exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('invalid empno'); |
|  | end; |
|  | function getsal(p\_empno emp.empno%type) return number as |
|  | v\_sal emp.sal%type; |
|  | begin |
|  | select sal into v\_sal from emp where empno=p\_empno; |
|  | dbms\_output.put\_line('from getsal function sal = ' || v\_sal); |
|  | return v\_sal; |
|  | exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('invalid empno'); |
|  | end; |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | Package body created. |
|  | -------------------------------------------------- |
|  | execute emp\_pack.getempinfo(7788); |
|  |  |
|  | OUTPUT: |
|  | in private proc pri\_proc |
|  | from getempinfo proc ename = SCOTT |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | 3)create or replace package emp\_pack as |
|  | g\_var number:=100; |
|  | procedure getempinfo(p\_empno emp.empno%type); |
|  | function getsal(p\_empno emp.empno%type) return number; |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | Package created. |
|  | ------------------------------------------------------ |
|  | create or replace package body emp\_pack as |
|  | procedure pri\_proc as |
|  | begin |
|  | dbms\_output.put\_line('in private proc pri\_proc'); |
|  | end; |
|  | procedure getempinfo(p\_empno emp.empno%type) as |
|  | v\_name emp.ename%type; |
|  | begin |
|  | pri\_proc; |
|  | select ename into v\_name from emp where empno=p\_empno; |
|  | dbms\_output.put\_line('from getempinfo proc ename = ' || v\_name); |
|  | exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('invalid empno'); |
|  | end; |
|  | function getsal(p\_empno emp.empno%type) return number as |
|  | v\_sal emp.sal%type; |
|  | begin |
|  | select sal into v\_sal from emp where empno=p\_empno; |
|  | dbms\_output.put\_line('from getsal function sal = ' || v\_sal); |
|  | return v\_sal; |
|  | exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('invalid empno'); |
|  | end; |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | Package body created. |
|  | ------------------------------------------------------------ |
|  | begin |
|  | dbms\_output.put\_line(' val = '|| emp\_pack.g\_var); |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | val = 100 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 4)create or replace package emp\_pack as |
|  | g\_var number:=100; |
|  | procedure getempinfo(p\_empno emp.empno%type); |
|  | function getsal(p\_empno emp.empno%type) return number; |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT |
|  | Package created. |
|  | ------------------------------------------------ |
|  |  |
|  | create or replace package body emp\_pack as |
|  | pri\_var number:=122; |
|  | procedure pri\_proc as |
|  | Begin |
|  | dbms\_output.put\_line('in private proc pri\_proc'); |
|  | end; |
|  | procedure getempinfo(p\_empno emp.empno%type) as |
|  | v\_name emp.ename%type; |
|  | Begin |
|  | pri\_proc; |
|  | select ename into v\_name from emp where empno=p\_empno; |
|  | dbms\_output.put\_line('from getempinfo proc ename = ' || v\_name); |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('invalid empno'); |
|  | end; |
|  | function getsal(p\_empno emp.empno%type) return number as |
|  | v\_sal emp.sal%type; |
|  | Begin |
|  | select sal into v\_sal from emp where empno=p\_empno; |
|  | dbms\_output.put\_line('from getsal function sal = ' || v\_sal); |
|  | return v\_sal; |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('invalid empno'); |
|  | end; |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | Package body created. |
|  | ------------------------------------------------------------- |
|  | Begin |
|  | dbms\_output.put\_line(' val = '|| emp\_pack.pri\_var); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | ERROR at line 2: |
|  | ORA-06550: line 2, column 44: |
|  | PLS-00302: component 'PRI\_VAR' must be declared |
|  | ORA-06550: line 2, column 1: |
|  | PL/SQL: Statement ignored |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | 5) |
|  | create or replace package emp\_pack as |
|  | g\_var number:=100; |
|  | procedure getempinfo(p\_empno emp.empno%type); |
|  | function getsal(p\_empno emp.empno%type) return number; |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | Package created. |
|  | ------------------------------- |
|  | create or replace package body emp\_pack as |
|  | pri\_var number:=122; |
|  | procedure pri\_proc as |
|  | Begin |
|  | dbms\_output.put\_line('in private proc pri\_proc'); |
|  | end; |
|  | procedure getempinfo(p\_empno emp.empno%type) as |
|  | v\_name emp.ename%type; |
|  | Begin |
|  | pri\_proc; |
|  | select ename into v\_name from emp where empno=p\_empno; |
|  | dbms\_output.put\_line('from getempinfo proc ename = ' || v\_name); |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('invalid empno'); |
|  | end; |
|  | function getsal(p\_empno emp.empno%type) return number as |
|  | v\_sal emp.sal%type; |
|  | Begin |
|  | select sal into v\_sal from emp where empno=p\_empno; |
|  | dbms\_output.put\_line('from getsal function sal = ' || v\_sal); |
|  | return v\_sal; |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('invalid empno'); |
|  | end; |
|  | Begin |
|  | dbms\_output.put\_line('initialization'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Package body created. |
|  | ------------------------------------------------- |
|  | A)begin |
|  | dbms\_output.put\_line(' val = '|| emp\_pack.g\_var); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Initialization |
|  | val = 100 |
|  |  |
|  |  |
|  | B)begin |
|  | dbms\_output.put\_line(' val = '|| emp\_pack.g\_var); |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | val = 100 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 6) |
|  | create or replace package emp\_pack as |
|  | g\_var number; |
|  | procedure getempinfo(p\_empno emp.empno%type); |
|  | function getsal(p\_empno emp.empno%type) return number; |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | Package created. |
|  | --------------------------------------------------------------------- |
|  | create or replace package body emp\_pack as |
|  | pri\_var number:=122; |
|  | procedure pri\_proc as |
|  | Begin |
|  | dbms\_output.put\_line('in private proc pri\_proc'); |
|  | end; |
|  | procedure getempinfo(p\_empno emp.empno%type) as |
|  | v\_name emp.ename%type; |
|  | Begin |
|  | pri\_proc; |
|  | select ename into v\_name from emp where empno=p\_empno; |
|  | dbms\_output.put\_line('from getempinfo proc ename = ' || v\_name); |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('invalid empno'); |
|  | end; |
|  | function getsal(p\_empno emp.empno%type) return number as |
|  | v\_sal emp.sal%type; |
|  | Begin |
|  | select sal into v\_sal from emp where empno=p\_empno; |
|  | dbms\_output.put\_line('from getsal function sal = ' || v\_sal); |
|  | return v\_sal; |
|  | Exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line('invalid empno'); |
|  | end; |
|  | Begin |
|  | dbms\_output.put\_line('initialization'); |
|  | g\_var:=1; |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Package body created. |
|  | ------------------------------------------------------------------- |
|  | Begin |
|  | dbms\_output.put\_line(' val = '|| emp\_pack.g\_var); |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | Initialization |
|  | val = 1 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | Begin |
|  | dbms\_output.put\_line(' val = '|| emp\_pack.g\_var); |
|  | end; |
|  | / |
|  | val = 1 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | FUNCTION OVERLOADING |
|  |  |
|  | create or replace package operations as |
|  | function add(var1 number,var2 number) return number; |
|  | function add(var1 number,var2 number,var3 number) return number; |
|  | end; |
|  |  |
|  | OUTPUT: |
|  | Package created. |
|  | ---------------------------------------------------------------------- |
|  | create or replace package body operations as |
|  | function add(var1 number,var2 number) return number as |
|  | result number; |
|  | Begin |
|  | result:=var1+var2; |
|  | dbms\_output.put\_line( 'add with 2 para result = ' || result ); |
|  | return result; |
|  | end; |
|  | function add(var1 number,var2 number,var3 number) return number as |
|  | result number; |
|  | Begin |
|  | result:=var1+var2+var3; |
|  | dbms\_output.put\_line( 'add with 3 para result = ' || result ); |
|  | return result; |
|  | end; |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Package body created |
|  | ------------------------------------------------------------------------ |
|  | Declare |
|  | res number; |
|  | Begin |
|  | res:= operations.add(10,20); |
|  | dbms\_output.put\_line(' add of 2 = ' || res); |
|  | res:= operations.add(100,200,300); |
|  | dbms\_output.put\_line(' add of 3 = ' || res); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | add with 2 para result = 30 |
|  | add of 2 = 30 |
|  | add with 2 para result = 600 |
|  | add of 3 = 600 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | META DATA |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

PACKAGE ASSIGNMENT

|  |
| --- |
|  |
| 1) |
|  | create or replace package MANAGE\_EMP\_PACK as |
|  | procedure HIRE\_EMP ( p\_empno number, p\_ename varchar2, p\_job varchar2); |
|  | procedure FIRE\_EMP ( p\_empno emp.empno%type, p\_ename emp.ename%type, p\_job emp.job%type); |
|  | v\_insert\_cnt varchar2(250); |
|  | v\_delete\_cnt varchar2(250); |
|  | function VALIDATE\_EMP (p\_empno emp.empno%type)return boolean; |
|  | end; |
|  | / |
|  |  |
|  |  |
|  | create or replace package body MANAGE\_EMP\_PACK as |
|  | procedure HIRE\_EMP ( p\_empno emp.empno%type, p\_ename emp.ename%type, p\_job emp.job%type ); |
|  |  |
|  |  |
|  |  |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 2) |
|  | create or replace package MY\_EMP\_PACK as |
|  | function GET\_AVG\_SAL(p\_ename varchar2) return number; |
|  | function GET\_AVG\_SAL(p\_empno number) return number; |
|  | end; |
|  |  |
|  | OUTPUT: |
|  | Package created. |
|  | --------------------------------------------------------------------------------------- |
|  | create or replace package body MY\_EMP\_PACK as |
|  | function GET\_AVG\_SAL(p\_ename varchar2) return number as |
|  | v\_avg number; |
|  | v\_dept number; |
|  | begin |
|  | select avg(sal),deptno into v\_avg,v\_dept from emp where deptno=v\_dept; |
|  | dbms\_output.put\_line(' average of sal = '||v\_avg); |
|  | return v\_avg; |
|  | exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line(' empno not available '); |
|  | end; |
|  | function GET\_AVG\_SAL(p\_empno number ) return number as |
|  | v\_avg number; |
|  | v\_dept number; |
|  | begin |
|  | select avg(sal),deptno into v\_avg,v\_dept from emp where deptno=v\_dept; |
|  | dbms\_output.put\_line(' average of sal = '|| v\_avg); |
|  | return v\_avg; |
|  | exception |
|  | when no\_data\_found then |
|  | dbms\_output.put\_line(' empno not available '); |
|  | end; |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | Package body created. |
|  |  |
|  | --------------------------------------------------------------------------- |
|  | declare |
|  | res number; |
|  | begin |
|  | res:= MY\_EMP\_PACK.GET\_AVG\_SAL('BLAKE'); |
|  | dbms\_output.put\_line(' salary of blake = ' || res); |
|  | res:= MY\_EMP\_PACK.GET\_AVG\_SAL(7844); |
|  | dbms\_output.put\_line(' salary of 7844 = ' || res); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  | average of sal = 2850 |
|  | salary of blake = 2850 |
|  | average of sal = 1500 |
|  | salary of 7844 = 1500 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 2)SQL> create or replace package MY\_EMP\_PACK as |
|  | 2 function GET\_AVG\_SAL(p\_ename varchar2) return number; |
|  | 3 function GET\_AVG\_SAL(p\_empno number) return number; |
|  | 4 end; |
|  | 5 / |
|  |  |
|  | Package created. |
|  |  |
|  | SQL> create or replace package body MY\_EMP\_PACK as |
|  | 2 function GET\_AVG\_SAL(p\_ename varchar2) return number as |
|  | 3 v\_avg number; |
|  | 4 begin |
|  | 5 select avg(sal) into v\_avg from emp where ename=p\_ename; |
|  | 6 dbms\_output.put\_line(' average of sal = '||v\_avg); |
|  | 7 return v\_avg; |
|  | 8 exception |
|  | 9 when no\_data\_found then |
|  | 10 dbms\_output.put\_line(' empno not available '); |
|  | 11 end; |
|  | 12 |
|  | 13 function GET\_AVG\_SAL(p\_empno number ) return number as |
|  | 14 v\_avg number; |
|  | 15 begin |
|  | 16 select avg(sal) into v\_avg from emp where empno=p\_empno; |
|  | 17 dbms\_output.put\_line(' average of sal = '|| v\_avg); |
|  | 18 return v\_avg; |
|  | 19 exception |
|  | 20 when no\_data\_found then |
|  | 21 dbms\_output.put\_line(' empno not available '); |
|  | 22 end; |
|  | 23 end; |
|  | 24 / |
|  |  |
|  | Package body created. |
|  |  |
|  |  |
|  |  |
|  | SQL> declare |
|  | 2 res number; |
|  | 3 begin |
|  | 4 res:= MY\_EMP\_PACK.GET\_AVG\_SAL('BLAKE'); |
|  | 5 dbms\_output.put\_line(' salary of blake = ' || res); |
|  | 6 res:= MY\_EMP\_PACK.GET\_AVG\_SAL(7844); |
|  | 7 dbms\_output.put\_line(' salary of 7844 = ' || res); |
|  | 8 end; |
|  | 9 / |
|  | average of sal = 2850 |
|  | salary of blake = 2850 |
|  | average of sal = 1500 |
|  | salary of 7844 = 1500 |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

ORDBMS

|  |
| --- |
|  |
| OBJECT RELATIONAL DATABASE SYSTEM: (ORDBMS) |
|  |  |
|  | create or replace type emp\_address as object |
|  | (street varchar2(20), city varchar2(20), country varchar2(20)); |
|  | OUTPUT: |
|  | Type created. |
|  |  |
|  | create table person (pid number, pname varchar2(20), |
|  | address emp\_address); |
|  | OUTPUT: |
|  | Table created. |
|  |  |
|  | insert into person values(1,'karthik' , emp\_address('ayanavaram', 'chennai', 'India')); |
|  | OUTPUT: |
|  | 1 row created. |
|  |  |
|  | insert into person values(2,'vijay' , emp\_address('avadi', 'chennai', 'India')); |
|  | OUTPUT: |
|  | 1 row created. |
|  |  |
|  | select \* from person |
|  | OUTPUT: |
|  | PID PNAME |
|  | ---------- -------------------- |
|  | ADDRESS(STREET, CITY, COUNTRY) |
|  | -------------------------------------------------------------------------------- |
|  | 1 karthik |
|  | EMP\_ADDRESS('ayanavaram', 'chennai', 'India') |
|  |  |
|  | 2 vijay |
|  | EMP\_ADDRESS('avadi', 'chennai', 'India') |
|  |  |
|  |  |
|  | select p.address.city from person p where pid=2; |
|  | OUTPUT: |
|  | ADDRESS.CITY |
|  | -------------------- |
|  | chennai |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | update person set address=emp\_address('annanagar','chennai city' ,'India') |
|  | where pid=2; |
|  | OUTPUT: |
|  | 1 row created. |
|  |  |
|  | select p.address.city from person p where pid=2; |
|  | OUTPUT: |
|  | ADDRESS.CITY |
|  | -------------------- |
|  | chennai city |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | PLSQL QUERY |
|  |  |
|  | declare |
|  | add emp\_address; |
|  | begin |
|  | add:=emp\_address(' aaa ', ' bbb ' ,' ccc '); |
|  | update person set address = add where pid=1; |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | select p.address.city from person p where pid=1; |
|  | OUTPUT: |
|  | ADDRESS.CITY |
|  | -------------------- |
|  | bbb |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

ORDBMS ASSIGNMENT

|  |
| --- |
|  |
| 1)create or replace type MARKS\_TYPE as object |
|  | (Subject varchar2(25),midTermMarks number(3), annualmarks number(3)); |
|  | / |
|  | OUTPUT: |
|  | Type created. |
|  |  |
|  | create table MYMARKS(name varchar2(25), marks MARKS\_TYPE); |
|  | OUTPUT: |
|  | Table created |
|  |  |
|  | insert into MYMARKS values( ' vishwa ' , MARKS\_TYPE( ' sql ' ,80,85)); |
|  | insert into MYMARKS values( ' ashok ' , MARKS\_TYPE( ' plsql ' ,70,75)); |
|  | insert into MYMARKS values( ' jeeva ' , MARKS\_TYPE( ' java ' ,70,95)); |
|  |  |
|  | OUTPUT: |
|  | 1 row created. |
|  | 1 row created. |
|  | 1 row created. |
|  |  |
|  | select \* from MYMARKS; |
|  | OUTPUT: |
|  | NAME |
|  | ------------------------- |
|  | MARKS(SUBJECT, MIDTERMMARKS, ANNUALMARKS) |
|  | -------------------------------------------------------------------------------- |
|  | vishwa |
|  | MARKS\_TYPE(' sql ', 80, 85) |
|  |  |
|  | ashok |
|  | MARKS\_TYPE(' plsql ', 70, 75) |
|  |  |
|  | jeeva |
|  | MARKS\_TYPE(' java ', 70, 95) |
|  |  |
|  | select m.marks.subject from MYMARKS m where name=' ashok '; |
|  | OUTPUT: |
|  | MARKS.SUBJECT |
|  | ------------------------- |
|  | plsql |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 2)create table students (studentName varchar2(25), rollno number(3) primary key, score MARKS\_TYPE); |
|  | OUTPUT: |
|  | Table created |
|  |  |
|  | insert into students values( ' vinodh ',25 , MARKS\_TYPE( 'J2E' ,93,95)); |
|  | insert into students values( ' ajay ' ,27 ,MARKS\_TYPE( 'ANGULAR JS' ,96,92)); |
|  | insert into students values( ' ruhid ' ,36 ,MARKS\_TYPE( 'HTML' ,97,80)); |
|  | OUTPUT: |
|  | 1 row created. |
|  | 1 row created. |
|  | 1 row created. |
|  |  |
|  | select s.score.midTermMarks , s.score.annualmarks from students s where rollno=27; |
|  | OUTPUT: |
|  | SCORE.MIDTERMMARKS SCORE.ANNUALMARKS |
|  | ------------------ ----------------- |
|  | 96 92 |
|  |  |
|  | select s.score.annualmarks+5 from students s where rollno=36; |
|  | OUTPUT: |
|  |  |
|  | S.SCORE.ANNUALMARKS+5 |
|  | --------------------- |
|  | 85 |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | 3) |
|  | create or replace type ICECREAM\_TYPE as object |
|  | (flovorNo number(2), fname varchar(10), cost number(3), Hastopping varchar(1)); |
|  | / |
|  | OUTPUT: |
|  | Type created. |
|  |  |
|  | create table menu( menuno number(2) primary key , maincourse varchar(15) not null , desert ICECREAM\_TYPE , Colddrink varchar(6)); |
|  | OUTPUT: |
|  | Table created |
|  |  |
|  |  |
|  | insert into menu values( 12, ' BIRIYANI ' ,ICECREAM\_TYPE( 20 , ' VANILLA ', 50, 'Y'), ' COKE '); |
|  | insert into menu values( 13, ' NOODLES ' ,ICECREAM\_TYPE( 30 , ' BADAM ', 60, 'N'), ' PEPSI' ); |
|  | insert into menu values( 14, 'FRIEDRICE' ,ICECREAM\_TYPE( 40 , 'CHOCOLATE', 50, 'Y'), 'LIMCA'); |
|  | insert into menu values( 15, ' PAROTA ' ,ICECREAM\_TYPE( 50 , ' PISTA ', 50, 'N'), ' FANTA' ); |
|  | insert into menu values( 16, 'CHAPATHI' ,ICECREAM\_TYPE( 60 , 'MIXING', 50, 'Y'),' LASSI'); |
|  |  |
|  | OUTPUT: |
|  | 1 row created. |
|  | 1 row created. |
|  | 1 row created. |
|  | 1 row created. |
|  | 1 row created. |
|  |  |
|  | select \* from menu; |
|  |  |
|  | OUTPUT: |
|  |  |
|  | MENUNO MAINCOURSE |
|  | ---------- --------------- |
|  | DESERT(FLOVORNO, FNAME, COST, HASTOPPING) |
|  | -------------------------------------------------------------------------------- |
|  | COLDDR |
|  | ------ |
|  | 12 BIRIYANI |
|  | ICECREAM\_TYPE(20, ' VANILLA ', 50, 'Y') |
|  | COKE |
|  |  |
|  | 13 NOODLES |
|  | ICECREAM\_TYPE(30, ' BADAM ', 60, 'N') |
|  | PEPSI |
|  |  |
|  | MENUNO MAINCOURSE |
|  | ---------- --------------- |
|  | DESERT(FLOVORNO, FNAME, COST, HASTOPPING) |
|  | -------------------------------------------------------------------------------- |
|  | COLDDR |
|  | ------ |
|  |  |
|  | 14 FRIEDRICE |
|  | ICECREAM\_TYPE(40, 'CHOCOLATE', 50, 'Y') |
|  | LIMCA |
|  |  |
|  | 15 PAROTA |
|  | ICECREAM\_TYPE(50, ' PISTA ', 50, 'N') |
|  |  |
|  | MENUNO MAINCOURSE |
|  | ---------- --------------- |
|  | DESERT(FLOVORNO, FNAME, COST, HASTOPPING) |
|  | -------------------------------------------------------------------------------- |
|  | COLDDR |
|  | ------ |
|  | FANTA |
|  |  |
|  | 16 CHAPATHI |
|  | ICECREAM\_TYPE(60, 'MIXING', 50, 'Y') |
|  | LASSI |
|  |  |
|  |  |
|  | select i.desert.cost , i.desert.fname from menu i where i.desert.flovorno=20; |
|  |  |
|  | OUTPUT: |
|  | DESERT.COST DESERT.FNA |
|  | ----------- ---------- |
|  | 50 VANILLA |
|  |  |
|  | update menu i set i.desert.cost=i.desert.cost+20, maincourse='fish' where menuno=12; |
|  |  |
|  | OUTPUT: |
|  | 1 row updated. |
|  | select \* from menu; |
|  |  |
|  | MENUNO MAINCOURSE |
|  | ---------- --------------- |
|  | DESERT(FLOVORNO, FNAME, COST, HASTOPPING) |
|  | -------------------------------------------------------------------------------- |
|  | COLDDR |
|  | ------ |
|  | 12 fish |
|  | ICECREAM\_TYPE(20, ' VANILLA ', 90, 'Y') |
|  | COKE |
|  |  |
|  | 13 NOODLES |
|  | ICECREAM\_TYPE(30, ' BADAM ', 60, 'N') |
|  | PEPSI |
|  |  |
|  | MENUNO MAINCOURSE |
|  | ---------- --------------- |
|  | DESERT(FLOVORNO, FNAME, COST, HASTOPPING) |
|  | -------------------------------------------------------------------------------- |
|  | COLDDR |
|  | ------ |
|  |  |
|  | 14 FRIEDRICE |
|  | ICECREAM\_TYPE(40, 'CHOCOLATE', 50, 'Y') |
|  | LIMCA |
|  |  |
|  | 15 PAROTA |
|  | ICECREAM\_TYPE(50, ' PISTA ', 50, 'N') |
|  |  |
|  | MENUNO MAINCOURSE |
|  | ---------- --------------- |
|  | DESERT(FLOVORNO, FNAME, COST, HASTOPPING) |
|  | -------------------------------------------------------------------------------- |
|  | COLDDR |
|  | ------ |
|  | FANTA |
|  |  |
|  | 16 CHAPATHI |
|  | ICECREAM\_TYPE(60, 'MIXING', 50, 'Y') |
|  | LASSI |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | TO ADD VALUES |
|  | create or replace procedure formodify(p\_empid number, p\_phone in out phone\_no1) as |
|  | var number; |
|  | begin |
|  | var:=p\_phone.count; |
|  | p\_phone.extend(1); |
|  | p\_phone(var+1):=abc1('demo','demoPL'); |
|  | update emp\_v1 set phone where empid=p\_empid; |
|  | dbms\_output.put\_line('done'); |
|  | end; |
|  |  |
|  |  |
|  |  |
|  | declare |
|  | t\_phone phone\_no1; |
|  | begin |
|  | select phone into t\_phone from emp\_v1 where empid=1; |
|  | formodify(1,t\_phone); |
|  | end; |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | NESTED |
|  |  |
|  | create type vehicle as object(vid number, vname varchar2(20)); |
|  | / |
|  |  |
|  | create type vehicle\_tab as table of vehicle; |
|  | / |
|  |  |
|  |  |
|  | create table person\_id(pid number, pname varchar2(20), vehicle\_details vehicle\_tab) |
|  | nested table vehicle\_details store as vehicle\_details\_tab; |
|  |  |
|  | insert into person\_id values(1, ' ariv ' , vehicle\_tab(vehicle(1, ' ferrari '), vehicle (2, ' BMW '))) |
|  |  |
|  | OUTPUT: |
|  | SQL> select \* from person\_id; |
|  |  |
|  | PID PNAME |
|  | ---------- -------------------- |
|  | VEHICLE\_DETAILS(VID, VNAME) |
|  | -------------------------------------------------------------------------------- |
|  | 1 ariv |
|  | VEHICLE\_TAB(VEHICLE(1, ' ferrari '), VEHICLE(2, ' BMW ')) |
|  |  |
|  | FOR UPDATING: |
|  |  |
|  | declare |
|  | new\_vehicle vehicle\_tab; |
|  | begin |
|  | new\_vehicle:=vehicle\_tab(vehicle(1,'ROYAL ENFIELD'), vehicle(2,' DUCATI ')); |
|  | update person\_id set vehicle\_details=new\_vehicle where pid=1; |
|  | dbms\_output.put\_line('updated'); |
|  | end; |
|  | / |
|  |  |
|  | OUTPUT: |
|  |  |
|  | updated |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | SQL> select \* from person\_id; |
|  |  |
|  | PID PNAME |
|  | ---------- -------------------- |
|  | VEHICLE\_DETAILS(VID, VNAME) |
|  | -------------------------------------------------------------------------------- |
|  | 1 ariv |
|  | VEHICLE\_TAB(VEHICLE(1, 'ROYAL ENFIELD'), VEHICLE(2, ' DUCATI ')) |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | insert into person\_id values(2, ' VINAY ' , vehicle\_tab(vehicle(1, ' ALTO '), vehicle (2, ' PUNTO '))) |
|  |  |
|  |  |
|  | declare |
|  | l\_name person\_id.pname%type; |
|  | l\_vehicle person\_id.vehicle\_details%type; |
|  | cursor get\_data is select pname,vehicle\_details from person\_id; |
|  | begin |
|  | open get\_data; |
|  | loop |
|  | fetch get\_data into l\_name,l\_vehicle; |
|  | exit when get\_data%notfound; |
|  | dbms\_output.put\_line('person name = ' || l\_name || 'and vehicles are '); |
|  | for x in 1..l\_vehicle.count |
|  | loop |
|  | dbms\_output.put\_line(l\_vehicle(x).vname); |
|  | end loop; |
|  | end loop; |
|  | end; |
|  | / |
|  | OUTPUT: |
|  | person name = ariv and vehicles are |
|  | ROYAL ENFIELD |
|  | DUCATI |
|  | person name = VINAY and vehicles are |
|  | ALTO |
|  |  |
|  | PUNTO |
|  |  |
|  | PL/SQL procedure successfully completed. |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | update the (select vehicle\_details from person\_id where pid=2) set vname=' yamaha ' where vid=1; |
|  |  |
|  | OUTPUT: |
|  |  |
|  | SQL> select \* from person\_id where pid=2; |
|  |  |
|  | PID PNAME |
|  | ---------- -------------------- |
|  | VEHICLE\_DETAILS(VID, VNAME) |
|  | -------------------------------------------------------------------------------- |
|  | 2 VINAY |
|  | VEHICLE\_TAB(VEHICLE(1, ' yamaha '), VEHICLE(2, ' |
|  | PUNTO ')) |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | insert into the (select vehicle\_details from person\_id where pid=2) values (3, ' BAJAJ '); |
|  |  |
|  | OUTPUT: |
|  | 1 row created. |
|  |  |
|  | SQL> select \* from person\_id where pid=2; |
|  |  |
|  | PID PNAME |
|  | ---------- -------------------- |
|  | VEHICLE\_DETAILS(VID, VNAME) |
|  | -------------------------------------------------------------------------------- |
|  | 2 VINAY |
|  | VEHICLE\_TAB(VEHICLE(1, ' yamaha '), VEHICLE(2, ' |
|  | PUNTO '), VEHICLE(3, ' BAJAJ ')) |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | delete from the (select vehicle\_details from person\_id where pid=2) where vid=1; |
|  |  |
|  | OUTPUT: |
|  |  |
|  | SQL> select \* from person\_id where pid=2; |
|  |  |
|  | PID PNAME |
|  | ---------- -------------------- |
|  | VEHICLE\_DETAILS(VID, VNAME) |
|  | -------------------------------------------------------------------------------- |
|  | 2 VINAY |
|  | VEHICLE\_TAB(VEHICLE(2, ' |
|  | PUNTO '), VEHICLE(3, ' BAJAJ ')) |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  |  |