Exemples PACKAGES

--POINTEUR de CURSEUR

CREATE OR REPLACE PACKAGE **PK\_emp** IS

TYPE t\_emp IS REF CURSOR RETURN emp%rowtype;

END;

/

CREATE OR REPLACE PROCEDURE **p\_emp**

(v\_deptno IN emp.deptno%TYPE, c\_emp IN OUT **PK\_emp**.t\_emp) IS

BEGIN

OPEN c\_emp FOR SELECT \* FROM emp WHERE deptno = v\_deptno;

END ;

/

DECLARE

c\_emp **PK\_emp**.t\_emp;

v\_emp emp%rowtype;

BEGIN

**p\_emp** (10,c\_emp);

LOOP

FETCH c\_emp INTO v\_emp;

EXIT WHEN c\_emp%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_emp.ename||' : '||v\_emp.sal);

END LOOP;

CLOSE c\_emp;

END;

/

--Gestion d’une exception

create table t1(nom varchar2(5));

create or replace PACKAGE PK\_exception IS

trop exception;

pragma exception\_init(trop,-12899);

end;

/

begin

insert into t1 values('MARTIN');

commit;

exception

when PK\_exception.trop then

dbms\_output.put\_line('Valeur trop grande');

end;

/

--déclaration de curseur

Create or replace package PK\_emp IS

cursor c\_emp is select \* from emp;

end;

/

create or replace procedure p\_emp is

begin

for i in pk\_emp.c\_emp loop

dbms\_output.put\_line(i.ename);

end loop;

end;

/

exec p\_emp;

--package avec type et curseur

CREATE OR REPLACE PACKAGE gestion\_dept IS

TYPE T\_DEP IS RECORD(num dept.deptno%type,

nom dept.dname%type);

CURSOR C\_DEP RETURN T\_DEP;

PROCEDURE liste\_dept;

END;

/

CREATE OR REPLACE PACKAGE BODY gestion\_dept IS

CURSOR C\_DEP RETURN T\_DEP IS SELECT deptno,dname FROM dept;

PROCEDURE liste\_dept IS

BEGIN

FOR i IN C\_DEP LOOP

dbms\_output.put\_line(i.num||' : '||i.nom);

END LOOP;

END;

END;

/

show errors

EXECUTE gestion\_dept.liste\_dept;

--si la comm est supérieure à la comm max alors erreur

CREATE OR REPLACE PACKAGE PK\_EX1 IS

comm\_std number:=200;

PROCEDURE annule\_comm(nouv\_comm number);

END PK\_EX1;

/

CREATE OR REPLACE PACKAGE BODY PK\_EX1 IS

FUNCTION validation(comm number) RETURN BOOLEAN IS

max\_comm emp.comm%type;

BEGIN

select max(comm) INTO max\_comm

from emp;

RETURN (comm between 0.0 and max\_comm);

END validation;

PROCEDURE annule\_comm(nouv\_comm number) IS

BEGIN

IF validation(nouv\_comm) then

comm\_std:=nouv\_comm;

ELSE

raise\_application\_error(-20210,'Mauvaise comm');

END IF;

DBMS\_OUTPUT.PUT\_LINE(comm\_std);

END annule\_comm;

END PK\_EX1;

/

select max(comm) from emp;

exec PK\_EX1.annule\_comm(2000);

--Pour retourner un employé par le numéro de ligne

CREATE OR REPLACE PACKAGE PK\_EX2 IS

TYPE emp\_table\_type IS TABLE OF emp%ROWTYPE

INDEX BY BINARY\_INTEGER;

PROCEDURE employes(emploi OUT emp\_table\_type);

END PK\_EX2;

/

CREATE OR REPLACE PACKAGE BODY PK\_EX2 IS

PROCEDURE employes(emploi OUT emp\_table\_type) IS

i binary\_integer:=0;

BEGIN

FOR x IN (SELECT \* FROM emp) LOOP

emploi(i):=x;

i:=i+1;

END LOOP;

END employes;

END PK\_EX2;

/

DECLARE

emploi PK\_EX2.emp\_table\_type;

BEGIN

PK\_EX2.employes(emploi);

dbms\_output.put\_line('Emp 2: '||emploi(0).ename);

END;

/

--Pour Business Objects par exemple

create or replace package PK\_emp IS

type emp\_row\_type IS REF CURSOR return emp%rowtype;

end;

/

create or replace procedure empdept

(return\_rows\_cursor IN OUT PK\_emp.emp\_row\_type,

dept\_parameter IN emp.deptno%type) IS

begin

open return\_rows\_cursor FOR select \* from emp

where emp.deptno=dept\_parameter;

end;

/