Série 3

/\*Laisser la ligne suivante : sert pour l'affichage de toutes les commandes\*/

SET SERVEROUTPUT ON;

/\* Ex1 \*/

DECLARE

v\_nb\_emp NUMBER(3);

BEGIN

SELECT COUNT(empno) INTO v\_nb\_emp FROM emp;

DBMS\_OUTPUT.PUT\_LINE('Le nombre total d''employés est : '||v\_nb\_emp);

END;\*/

/\* Ex2 \*/

/\* a \*/

ACCEPT p\_nb PROMPT 'Entrer le numéro d''employé'

DECLARE

sal\_moy\_emp emp.sal%TYPE;

v\_job emp.job%TYPE;

BEGIN

SELECT job INTO v\_job FROM emp WHERE empno=&p\_nb;

SELECT avg(sal) INTO sal\_moy\_emp FROM emp WHERE job=v\_job;

DBMS\_OUTPUT.PUT\_LINE('Moyenne des salaires des '||v\_job||': '||sal\_moy\_emp||'€');

COMMIT;

END;

/\* b\*/

ACCEPT p\_nb PROMPT 'Entrer le numéro d''employé'

DECLARE

v\_moy\_sal emp.sal%TYPE;

v\_job emp.job%TYPE;

v\_sal emp.sal%TYPE;

BEGIN

SELECT job INTO v\_job FROM emp WHERE empno=&p\_nb;

SELECT avg(sal) INTO v\_moy\_sal FROM emp WHERE job=v\_job;

SELECT sal INTO v\_sal FROM emp WHERE empno=&p\_nb;

IF v\_sal<v\_moy\_sal THEN

UPDATE emp

SET sal=v\_moy\_sal

WHERE empno=&p\_nb;

ELSE

UPDATE emp

SET sal=v\_sal\*1.1

WHERE empno=&p\_nb;

END IF;

COMMIT;

END;

/\* Ex3 \*/

SET SERVEROUTPUT ON;

DEFINE p\_num=5;

DECLARE

v\_fact NUMBER(1) :=1;

v\_num &p\_num%TYPE :=&p\_num;

BEGIN

IF &p\_num=0 THEN

DBMS\_OUTPUT.PUT\_LINE('Factorielle de 0 est: 1');

ELSE

FOR i IN 1..&p\_num LOOP

v\_fact:=v\_fact\*&p\_num;

END LOOP;

END IF;

DBMS\_OUTPUT.PUT\_LINE('Factorielle de '||v\_num||' est '||v\_fact);

COMMIT;

END; /\* Pas complet \*/

/\* Ex4 \*/

CREATE TABLE amorti (

Durée NUMBER(2) CONSTRAINT amorti\_pk PRIMARY KEY,

Somme\_due NUMBER(8),

Remb\_An NUMBER(6),

Intérêt NUMBER(5),

Annuité NUMBER(6) );

SET serveroutput on

SET verify off

ACCEPT somme prompt 'Entrez la somme empruntée :'

ACCEPT duree prompt 'Entrez la durée de l''emprunt en années :'

ACCEPT taux prompt 'Entrez le taux d''intérêt en % :'

DECLARE

v\_remb\_an amorti.somme\_due%type := &somme/&duree;

v\_interet amorti.intérêt%type;

v\_annuite amorti.annuité%type;

v\_som amorti.somme\_due%type;

BEGIN

v\_som := &somme;

FOR i IN 1..&duree LOOP

v\_interet := (v\_som \* &taux)/100;

v\_annuite := v\_remb\_an + v\_interet;

INSERT INTO amorti

VALUES (i, v\_som, v\_remb\_an, v\_interet, v\_annuite);

v\_som := v\_som - v\_remb\_an;

END LOOP

COMMIT;

END;

SELECT \* FROM amorti;

DROP TABLE amorti;

SET VERIFY ON

/\* Ex5.a-b \*/

SET SERVEROUTPUT ON;

ACCEPT p\_nb PROMPT 'Numéro à multiplier';

DECLARE

TYPE multi IS TABLE OF INTEGER(3) INDEX BY BINARY\_INTEGER;

v\_multi multi;

BEGIN

FOR i IN 1..9 LOOP

v\_multi(i) := &p\_nb \* i;

DBMS\_OUTPUT.PUT\_LINE(i || '\*&p\_nb=' || v\_multi(i));

END LOOP;

END;

/\* Ex6 \*/

--pas encore fonctionnel

CREATE TABLE amorti (

Durée NUMBER(2) CONSTRAINT amorti\_pk PRIMARY KEY,

Somme\_due NUMBER(8),

Remb\_An NUMBER(6),

Intérêt NUMBER(5),

Annuité NUMBER(6) );

SET serveroutput on

SET verify off

ACCEPT somme prompt 'Entrez la somme empruntée :'

ACCEPT duree prompt 'Entrez la durée de l''emprunt en années :'

ACCEPT taux prompt 'Entrez le taux d''intérêt en % :'

DECLARE

v\_remb\_an amorti.somme\_due%type := &somme/&duree;

v\_interet amorti.intérêt%type;

v\_annuite amorti.annuité%type;

v\_som amorti.somme\_due%type;

BEGIN

v\_som := &somme;

FOR i IN 1..&duree LOOP

v\_interet := (v\_som \* &taux)/100;

v\_annuite := v\_remb\_an + v\_interet;

INSERT INTO amorti

VALUES (i, v\_som, v\_remb\_an, v\_interet, v\_annuite);

v\_som := v\_som - v\_remb\_an;

END LOOP

COMMIT;

--

DBSM\_OUTPUT.PUT\_LINE(rpad('Duree', 7) || rpad('Intérêt', 10) || 'Annuité');

FOR enr\_amorti IN (SELECT \* FROM amorti) LOOP

DBSM\_OUTPUT.PUT\_LINE(rpad(enr\_amorti.durée, 7) || rpad(enr\_amorti.intérêt, 10) || enr\_amorti.annuité);

END LOOP;

--

END;

SELECT \* FROM amorti;

DROP TABLE amorti;

SET VERIFY ON

/\* Ex7.a\*/