DECLARE

nbta NUMBER;

BEGIN

SELECT count(\*) INTO nbta FROM user\_tables

WHERE TABLE\_NAME = 'MORPION';

IF nbta = 0 THEN

EXECUTE IMMEDIATE 'CREATE TABLE morpion(

y NUMBER,

A CHAR,

B CHAR,

C CHAR

)';

END IF;

END;

/

CREATE OR REPLACE FUNCTION nbToColName(nb IN NUMBER)

RETURN CHAR

IS

BEGIN

IF nb=1 THEN

RETURN 'A';

ELSIF nb=2 THEN

RETURN 'B';

ELSIF nb=3 THEN

RETURN 'C';

ELSE

RETURN '\_';

END IF;

END;

/

CREATE OR REPLACE PROCEDURE print\_game

IS

BEGIN

dbms\_output.put\_line(' ');

dbms\_output.enable(10000);

FOR ll in (SELECT \* FROM morpion ORDER BY Y) LOOP

dbms\_output.put\_line(' ' || ll.A || ' ' || ll.B || ' ' || ll.C);

END LOOP;

dbms\_output.put\_line(' ');

END;

/

CREATE OR REPLACE PROCEDURE reset\_game IS

ii NUMBER;

BEGIN

DELETE FROM morpion;

FOR ii in 1..3 LOOP

INSERT INTO morpion VALUES (ii,'\_','\_','\_');

END LOOP;

dbms\_output.enable(10000);

print\_game();

dbms\_output.put\_line('The game is ready to play');

END;

/

CREATE OR REPLACE PROCEDURE play(symbole IN VARCHAR2, ligne IN NUMBER, colonb IN NUMBER) IS

val morpion.a%type;

colo CHAR;

symbole2 CHAR;

BEGIN

SELECT nbToColName(colonb) INTO colo FROM DUAL;

EXECUTE IMMEDIATE ('SELECT ' || colo || ' FROM morpion WHERE y=' || ligne) INTO val;

IF val='\_' THEN

EXECUTE IMMEDIATE ('UPDATE morpion SET ' || colo || '=''' || symbole || ''' WHERE y=' || ligne);

IF symbole='X' THEN

symbole2:='O';

ELSE

symbole2:='X';

END IF;

print\_game();

dbms\_output.put\_line('Now ' || symbole2 || '''s turn. to play ');

ELSE

dbms\_output.enable(10000);

dbms\_output.put\_line('You cannot play this square, it is already played');

END IF;

END;

/

execute reset\_game;

CREATE OR REPLACE PROCEDURE winner(symbole IN VARCHAR2) IS

BEGIN

dbms\_output.enable(10000);

print\_game();

dbms\_output.put\_line('The player ' || symbole || ' '||'won !!');

dbms\_output.put\_line('---------------------------------------');

dbms\_output.put\_line('Launch of a new part...');

reset\_game();

END;

/

EXECUTE play('X', 1, 3);

EXECUTE play('O', 2, 1);

EXECUTE play('X', 2, 2);

EXECUTE play('O', 2, 3);

EXECUTE play('X', 3, 1);

CREATE OR REPLACE FUNCTION wincol\_request(nomcol IN VARCHAR2, symbole IN VARCHAR2)

RETURN VARCHAR2

IS

BEGIN

RETURN ('SELECT COUNT(\*) FROM morpion WHERE ' || nomcol || ' = '''|| symbole ||''' AND ' || nomcol || ' != ''\_''');

END;

/

CREATE OR REPLACE FUNCTION wincross\_request(nomcol IN VARCHAR2, yvalue IN NUMBER)

RETURN VARCHAR2

IS

BEGIN

RETURN ('SELECT '|| nomcol ||' FROM morpion WHERE y=' || yvalue);

END;

/

CREATE OR REPLACE FUNCTION wincol(nomcol IN VARCHAR2)

RETURN CHAR

IS

nbwin NUMBER;

r VARCHAR2(56);

BEGIN

SELECT wincol\_request(nomcol, 'X') into r FROM DUAL;

EXECUTE IMMEDIATE r INTO nbwin;

IF nbwin=3 THEN

RETURN 'X';

ELSIF nbwin=0 THEN

SELECT wincol\_request(nomcol, 'O') into r FROM DUAL;

EXECUTE IMMEDIATE r INTO nbwin;

IF nbwin=3 THEN

RETURN 'O';

END IF;

END IF;

RETURN '\_';

END;

/

CREATE OR REPLACE FUNCTION wincross(tmpx IN CHAR, numcol IN NUMBER, numligne IN NUMBER)

RETURN CHAR

IS

tmpvar CHAR;

tmpxvar CHAR;

r VARCHAR2(56);

BEGIN

SELECT wincross\_request(nbToColName(numcol), numligne) INTO r FROM DUAL;

IF tmpx IS NULL THEN

EXECUTE IMMEDIATE (r) INTO tmpxvar;

ELSIF NOT tmpx = '\_' THEN

EXECUTE IMMEDIATE (r) INTO tmpvar;

IF NOT tmpx = tmpvar THEN

tmpxvar := '\_';

END IF;

ELSE

tmpxvar := '\_';

END IF;

RETURN tmpxvar;

END;

/

CREATE OR REPLACE TRIGGER iswinner

AFTER UPDATE ON morpion

DECLARE

CURSOR cr\_ligne IS

SELECT \* FROM morpion ORDER BY Y;

crlv morpion%rowtype;

tmpvar CHAR;

tmpx1 CHAR;

tmpx2 CHAR;

r VARCHAR2(40);

BEGIN

FOR crlv IN cr\_ligne LOOP

-- line test

IF crlv.A = crlv.B AND crlv.B = crlv.C AND NOT crlv.A='\_' THEN

winner(crlv.A);

EXIT;

END IF;

-- colon test

SELECT wincol(nbToColName(crlv.Y)) INTO tmpvar FROM DUAL;

IF NOT tmpvar = '\_' THEN

winner(tmpvar);

EXIT;

END IF;

-- diagonal test

SELECT wincross(tmpx1, crlv.Y, crlv.Y) INTO tmpx1 FROM dual;

SELECT wincross(tmpx2, 4-crlv.Y, crlv.Y) INTO tmpx2 FROM dual;

END LOOP;

IF NOT tmpx1 = '\_' THEN

winner(tmpx1);

END IF;

IF NOT tmpx2 = '\_' THEN

winner(tmpx2);

END IF;

END;

/

EXECUTE reset\_game;

EXECUTE play('X', 1, 3);

EXECUTE play('O', 2, 1);

EXECUTE play('X', 2, 2);

EXECUTE play('O', 2, 3);

EXECUTE play('X', 3, 1);