**DMDD ASSIGNMENT 3**

**Submitted by: K SAI SUPRAJA**

**NUID: 002191644**

**EMAIL: kosuri.sa@northeastern.edu**

**TIC TAC TOE PL/ SQL CODE**

---Creation of TIC-TAC-TOE table

CREATE TABLE TicTacToe

(

ID NUMBER,

A CHAR,

B CHAR,

C CHAR

)

--- number to column name conversion function

CREATE OR REPLACE FUNCTION numberToColumn (num IN NUMBER)

RETURN VARCHAR

IS

BEGIN

IF

num =1 THEN

RETURN 'A';

ELSIF

num =2 THEN

RETURN 'B';

ELSIF num =3 THEN

RETURN 'C';

ELSE

RETURN '\_';

END IF;

END;

-- game board Procedure to display

CREATE OR REPLACE PROCEDURE gamePrint

IS

BEGIN

dbms\_output.enable(10000);

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

FOR j in (SELECT \* FROM TicTacToe ORDER BY ID)

LOOP

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

END LOOP;

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

END;

/

-- game reset procedure

CREATE OR REPLACE PROCEDURE gameRestart

IS

p NUMBER;

BEGIN

DELETE FROM TicTacToe;

FOR p in 1..3 LOOP

INSERT INTO TicTacToe VALUES (p,'\_','\_','\_');

END LOOP;

dbms\_output.enable(10000);

gamePrint();

dbms\_output.put\_line('Begin the game : EXECUTE gameplay (''X'', x, y);');

END;

/

-- playing game procedure

CREATE OR REPLACE PROCEDURE gameplay

(symbol IN VARCHAR2,

columnNumber IN NUMBER,

line IN NUMBER)

IS

val TicTacToe.a%type;

column CHAR;

symbolx CHAR;

BEGIN

SELECT numberToColumn (columnNumber) INTO column FROM DUAL;

EXECUTE IMMEDIATE ('SELECT ' || column || ' FROM TicTacToe WHERE ID =' || line) INTO val;

IF val='\_' THEN

EXECUTE IMMEDIATE ('UPDATE TicTacToe SET ' || column || '=''' || symbol || ''' WHERE ID=' || line);

IF symbol ='X' THEN

symbolx:='O';

ELSE

symbolx:='X';

END IF;

gamePrint ();

dbms\_output.put\_line('enter ' || symbolx || '. To start Playing game : EXECUTE gamePlay (''' || symbolx || ''', x, y);');

ELSE

dbms\_output.enable(10000);

dbms\_output.put\_line('The square is completed...play other square');

END IF;

END;

/

--- declare game winner procedure

CREATE OR REPLACE PROCEDURE winner(symbol IN VARCHAR2) IS

BEGIN

dbms\_output.enable(10000);

gamePrint();

dbms\_output.put\_line('Player ' || symbol || ' \*\*\*won the game\*\*\*');

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

dbms\_output.put\_line('Restart game... to play new game');

gameRestart ();

END;

/

--- column wise win function

CREATE OR REPLACE FUNCTION colwin(columnNumber IN VARCHAR2, symbol IN VARCHAR2)

RETURN VARCHAR2

IS

BEGIN

RETURN ('SELECT COUNT (\*) FROM TicTacToe WHERE ' || columnNumber || ' = '''|| symbol

||''' AND ' || columnNumber || ' != ''\_''');

END;

/

----- row wise win function

CREATE OR REPLACE FUNCTION diagonal\_win(columnNumber IN VARCHAR2,

val IN NUMBER)

RETURN VARCHAR2

IS

BEGIN

RETURN ('SELECT '|| columnNumber ||' FROM TicTacToe WHERE Id=' || val);

END;

/

-- column test function

CREATE OR REPLACE FUNCTION columnwin(columnNumber IN VARCHAR2)

RETURN CHAR

IS

numberwin NUMBER;

n VARCHAR2(100);

BEGIN

SELECT colwin (columnNumber, 'X') into n FROM DUAL;

EXECUTE IMMEDIATE n INTO numberwin ;

IF numberwin =3 THEN

RETURN 'X';

ELSIF numberwin =0 THEN

SELECT colwin (columnNumber, 'O') into n FROM DUAL;

EXECUTE IMMEDIATE n INTO numberwin;

IF numberwin =3 THEN

RETURN 'O';

END IF;

END IF;

RETURN '\_';

END;

/

--Horizontal win test function

CREATE OR REPLACE FUNCTION diagonalwin(tmpx IN CHAR, numbercol IN NUMBER, number\_row IN NUMBER)

RETURN CHAR

IS

Tmpvar CHAR;

tmpxvar CHAR;

n VARCHAR2(56);

BEGIN

SELECT diagonal\_win (numberToColumn (numbercol), number\_row) INTO n FROM DUAL;

IF tmpx IS NULL THEN

EXECUTE IMMEDIATE (n) INTO tmpxvar;

ELSIF NOT tmpx = '\_' THEN

EXECUTE IMMEDIATE (n) INTO tmpvar;

IF NOT tmpx = tmpvar THEN

tmpxvar := '\_';

END IF;

ELSE

tmpxvar := '\_';

END IF;

RETURN tmpxvar;

END;

/

-- test if a player won using trigger

CREATE OR REPLACE TRIGGER playerWon

AFTER UPDATE ON TicTacToe

DECLARE

CURSOR cr\_line IS

SELECT \* FROM TicTacToe ORDER BY Id;

crlv TicTacToe%rowtype;

tmpvar CHAR;

tmpxy CHAR;

tmpxz CHAR;

n VARCHAR2(40);

BEGIN

FOR crlv IN cr\_line LOOP

-- line test

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

winner(crlv.A);

EXIT;

END IF;

-- test column

SELECT columnwin(numberToColumn(crlv.Id)) INTO tmpvar FROM DUAL;

IF NOT tmpvar = '\_' THEN

winner(tmpvar);

EXIT;

END IF;

-- function to test horizontally

SELECT diagonalwin(tmpxy, crlv.Id, crlv.Id) INTO tmpxy FROM dual;

SELECT diagonalwin(tmpxz, 4-crlv.Id, crlv.Id) INTO tmpxz FROM dual;

END LOOP;

IF NOT tmpxy = '\_' THEN

winner (tmpxy);

END IF;

IF NOT tmpxz = '\_' THEN

winner (tmpxz);

END IF;

END;

/

EXECUTE gameRestart;

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

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

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

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

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

TEST CASE: 1 - Player “X” Wins

MOVES

EXECUTE gameRestart;

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

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

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

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

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

Graphical user interface

Description automatically generated with medium confidence

TEST CASE : 2 – POSITION IS RECHABLE

MOVES

EXECUTE gameRestart;

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

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

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

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

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

The below screen shot explains that when a step is reachable it’s prompted on the screen

A picture containing text

Description automatically generated

Test Case: 3

EXECUTE gameRestart;

EXECUTE gameplay('O', 1, 3);

EXECUTE gameplay('X', 2, 1);

EXECUTE gameplay('O', 2, 2);

EXECUTE gameplay('X', 2, 3);

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

Graphical user interface, application

Description automatically generated