

# Assignment3\_Manisha\_Bagora

```
Drop table Assignment_tictactoe3;

set serveroutput on;

-- Create table Assignment_tictactoe3 table if it does not exist

DECLARE

    mbta NUMBER;

BEGIN

    SELECT count(*) INTO mbta FROM user_tables

        WHERE TABLE_NAME = 'Assignment_tictactoe3';

    IF mbta = 0 THEN

        EXECUTE IMMEDIATE 'CREATE TABLE Assignment_tictactoe3(

            y NUMBER,

            A CHAR,

            B CHAR,

            C CHAR

        )';

    END IF;

END;

/

-- number to column name conversion function

CREATE OR REPLACE FUNCTION numbertocolname(mb IN NUMBER)

RETURN CHAR

IS

BEGIN

    IF mb=1 THEN

        RETURN 'A';
```

```

ELSIF mb=2 THEN

    RETURN 'B';

ELSIF mb=3 THEN

    RETURN 'C';

ELSE

    RETURN '_';

END IF;

END;

/

-- procedure to display the game board

CREATE OR REPLACE PROCEDURE print_game_tic_tac_toe IS

BEGIN

    dbms_output.enable(100000000);

    dbms_output.put_line(' ');

    FOR aa in (SELECT * FROM Assignment_tictactoe3 ORDER BY Y) LOOP

        dbms_output.put_line('  ' || aa.A || ' ' || aa.B || ' ' || aa.C);

    END LOOP;

    dbms_output.put_line(' ');

END;

/

-- The Procedure to reset the game

CREATE OR REPLACE PROCEDURE game_reset IS

ii NUMBER;

BEGIN

    DELETE FROM Assignment_tictactoe3;

    FOR ii in 1..3 LOOP

        INSERT INTO Assignment_tictactoe3 VALUES (ii,'_','_','_');

    END LOOP;

    dbms_output.enable(10000);

```

```

print_game_tic_tac_toe();

dbms_output.put_line('The Tic_tac_toe game is ready to play : EXECUTE play('X', x, y);');

END;

/

-- The procedure to play game tic_tac_toe

CREATE OR REPLACE PROCEDURE play(symb IN VARCHAR2, colmb IN NUMBER, row IN NUMBER) IS
val Assignment_tictactoe3.a%type;
colo CHAR;
symb2 CHAR;
BEGIN
    SELECT numbertocolname(colmb) INTO colo FROM DUAL;
    EXECUTE IMMEDIATE ('SELECT ' || colo || ' FROM Assignment_tictactoe3 WHERE y=' || row) INTO val;
    IF val!='_' THEN
        dbms_output.enable(1000000);
        dbms_output.put_line('Its a Draw');
        END IF;
    IF val='_' THEN
        EXECUTE IMMEDIATE ('UPDATE Assignment_tictactoe3 SET ' || colo || '="' || symb || '"' WHERE y='
|| row);
        IF symb='X' THEN
            symb2:='O';
        ELSE
            symb2:='X';
        END IF;
        print_game_tic_tac_toe();
        dbms_output.put_line('Around ' || symb2 || '. to play : EXECUTE play('' || symb2 || ''', x, y);');
    ELSE
        dbms_output.enable(10000);
        dbms_output.put_line('You cannot play this game square, it is already played');
    END IF;
END;

```

```

    END IF;

END;

/

-- winning procedure

CREATE OR REPLACE PROCEDURE winner_of_the_game(symb IN VARCHAR2) IS

BEGIN

    dbms_output.enable(10000);

    print_game_tic_tac_toe();

    dbms_output.put_line('The player ' || symb || ' won !!');

    dbms_output.put_line('-----');

    dbms_output.put_line('Now Starting a new game...');

    game_reset();

END;

/

-- The column creation function

CREATE OR REPLACE FUNCTION Request_Winner_col(colnom IN VARCHAR2, symb IN VARCHAR2)

RETURN VARCHAR2

IS

BEGIN

    RETURN ('SELECT COUNT(*) FROM Assignment_tictactoe3 WHERE ' || colnom || ' = ''' || symb || ''' AND

' || colnom || ' != "_"');

END;

/

-- column query creation function

CREATE OR REPLACE FUNCTION cross_winner_request(colnom IN VARCHAR2, yvalue IN NUMBER)

RETURN VARCHAR2

IS

BEGIN

    RETURN ('SELECT ' || colnom || ' FROM Assignment_tictactoe3 WHERE y=' || yvalue);

```

```

END;

/

-- The column test function

CREATE OR REPLACE FUNCTION Winner_col(colnom IN VARCHAR2)

RETURN CHAR

IS

    mbwin NUMBER;

    r VARCHAR2(1000);

BEGIN

    SELECT Request_Winner_col(colnom, 'X') into r FROM DUAL;

    EXECUTE IMMEDIATE r INTO mbwin;

    IF mbwin=3 THEN

        RETURN 'X';

    ELSIF mbwin=0 THEN

        SELECT Request_Winner_col(colnom, 'O') into r FROM DUAL;

        EXECUTE IMMEDIATE r INTO mbwin;

        IF mbwin=3 THEN

            RETURN 'O';

        END IF;

    END IF;

    RETURN '_';

END;

/

-- The diagonal test function of tic_tac_toe

CREATE OR REPLACE FUNCTION crosswin(tmpx IN CHAR, numcol IN NUMBER, numrow1 IN NUMBER)

RETURN CHAR

IS

    tmpvar CHAR;

    tmpxvar CHAR;

```

```

r VARCHAR2(65);
BEGIN
SELECT cross_winner_request(numbertocolname(numcol), numrow1) 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;
/
-- The test trigger if win
CREATE OR REPLACE TRIGGER winner_of_the_game_tic_toc_toe
AFTER UPDATE ON Assignment_tictactoe3
DECLARE
CURSOR cr_row1 IS
SELECT * FROM Assignment_tictactoe3 ORDER BY Y;
crlv Assignment_tictactoe3%rowtype;
tmpvar CHAR;
tmpx1 CHAR;
tmpx2 CHAR;
r VARCHAR2(40);
BEGIN
FOR crlv IN cr_row1 LOOP

```

```

-- line test

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

    winner_of_the_game(crlv.A);

    EXIT;

END IF;

-- column test

SELECT Winner_col(numbertocolname(crlv.Y)) INTO tmpvar FROM DUAL;

IF NOT tmpvar = '_' THEN

    winner_of_the_game(tmpvar);

    EXIT;

END IF;

-- diagonal test

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

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

END LOOP;

IF NOT tmpx1 = '_' THEN

    winner_of_the_game(tmpx1);

END IF;

IF NOT tmpx2 = '_' THEN

    winner_of_the_game(tmpx2);

END IF;

END;

/

--

Test Cases:

EXECUTE game_reset;

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

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

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

```

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

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

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

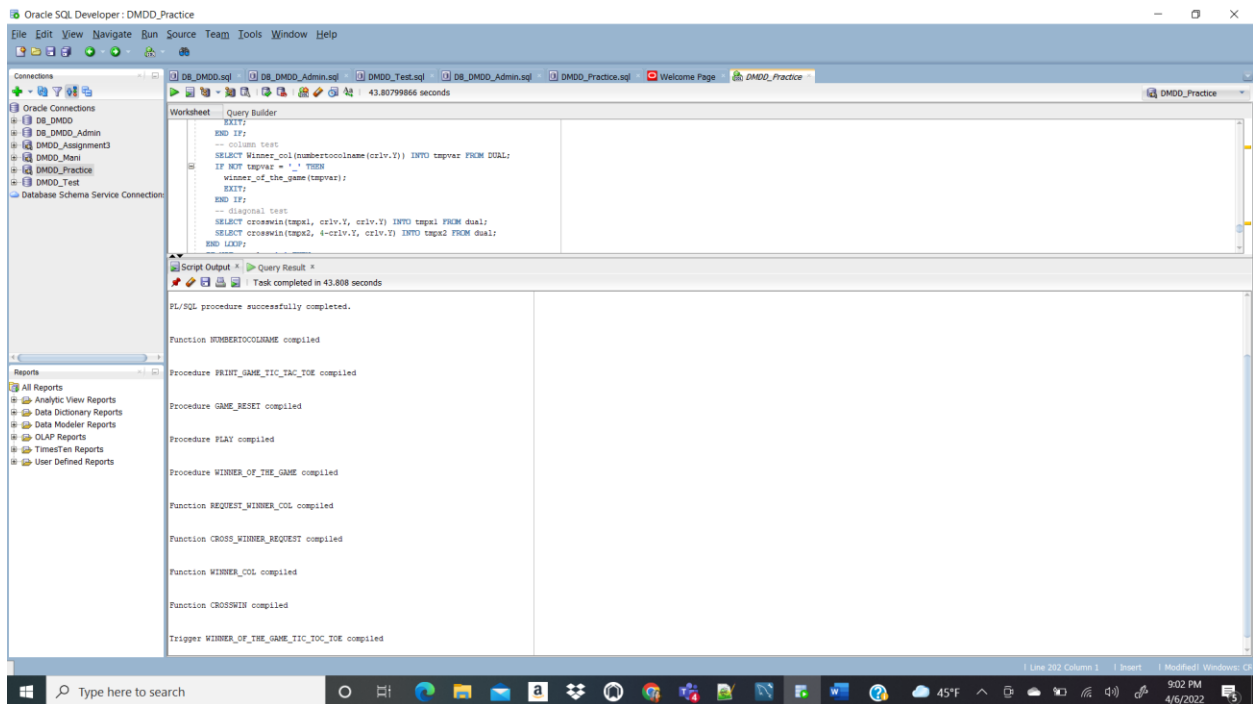
EXECUTE play('X', 2, 3);

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

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

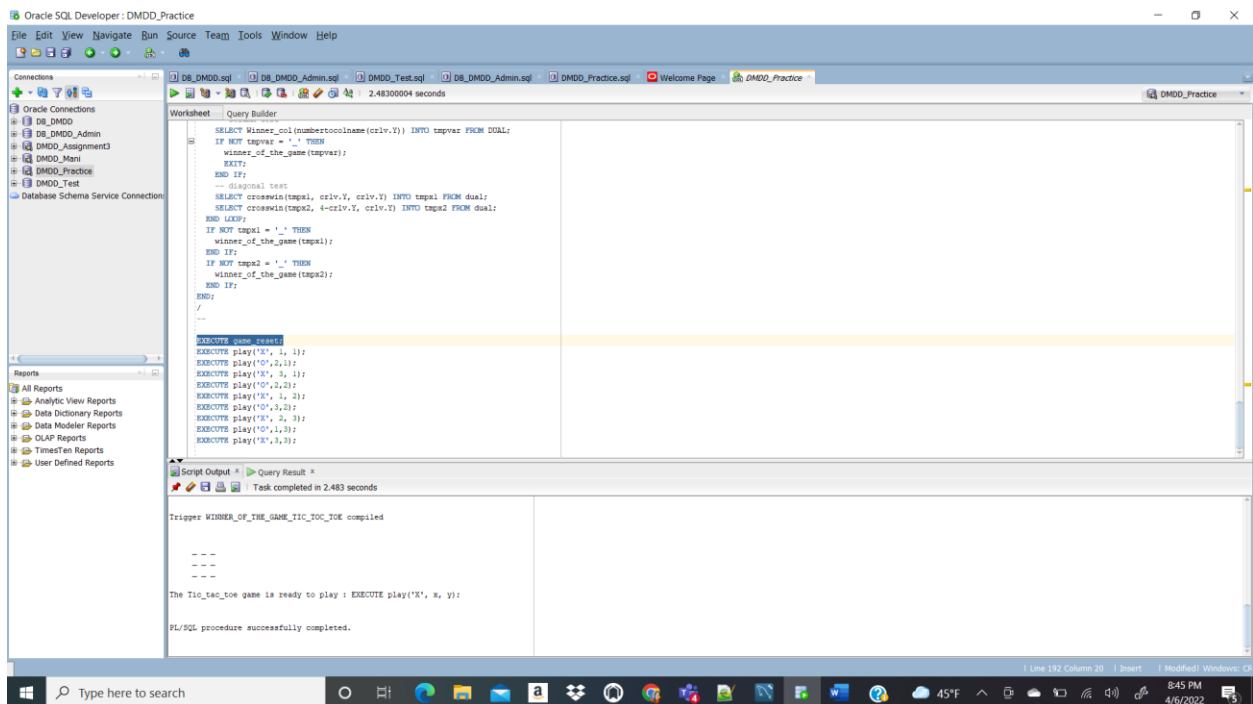
EXECUTE play('X',3,3);

### Execution of all Procedure and Triggers:





## Case: Reset Game



### Test Cases:

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

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

***EXECUTE playAA('X', 3, 1);***

***EXECUTE playAA('O',2,2);***

***EXECUTE play1('X', 1, 2);***

***EXECUTE play1('O',3,2);***

***EXECUTE play1('X', 2, 3);***

***EXECUTE play1('O',1,3);***

***EXECUTE play1('X',3,3);***

**Case 1: if the position is reachable, and the game will continue for at least one more move.**

```
SELECT Winner_col(numbercolumnname(crv.Y)) INTO tmpvar FROM DUAL;
IF NOT tmpvar = '-' THEN
  winner_of_the_game(tmpvar);
EXIT;
END IF;
-- diagonal test
SELECT crosswin(tmp1, crv.Y, crv.Y) INTO tmp1 FROM dual;
SELECT crosswin(tmp2, 4-crv.Y, crv.Y) INTO tmp2 FROM dual;
END LOOP;
IF NOT tmp1 = '-' THEN
  winner_of_the_game(tmp1);
END IF;
IF NOT tmp2 = '-' THEN
  winner_of_the_game(tmp2);
END IF;
END;
/
--
EXECUTE game_reset;
EXECUTE play('0',1,1);
EXECUTE play('0',2,1);
EXECUTE play('0',2,3);
EXECUTE play('X',1,2);
EXECUTE play('0',3,2);
EXECUTE play('0',3,3);
EXECUTE play('0',1,3);
EXECUTE play('X',3,3);
```

Script Output: Query Result

Task completed in 0.113 seconds

PL/SQL procedure successfully completed.

X - -  
- - -  
- - -

Around 0. to play : EXECUTE play('0', x, y);

PL/SQL procedure successfully completed.

```
SELECT Winner_col(numbercolumnname(crv.Y)) INTO tmpvar FROM DUAL;
IF NOT tmpvar = '-' THEN
  winner_of_the_game(tmpvar);
EXIT;
END IF;
-- diagonal test
SELECT crosswin(tmp1, crv.Y, crv.Y) INTO tmp1 FROM dual;
SELECT crosswin(tmp2, 4-crv.Y, crv.Y) INTO tmp2 FROM dual;
END LOOP;
IF NOT tmp1 = '-' THEN
  winner_of_the_game(tmp1);
END IF;
IF NOT tmp2 = '-' THEN
  winner_of_the_game(tmp2);
END IF;
END;
/
--
EXECUTE game_reset;
EXECUTE play('0',1,1);
EXECUTE play('X',3,3);
EXECUTE play('0',3,1);
EXECUTE play('0',2,2);
EXECUTE play('0',1,2);
EXECUTE play('X',2,3);
EXECUTE play('0',1,3);
EXECUTE play('X',3,3);
```

Script Output: Query Result

Task completed in 0.109 seconds

PL/SQL procedure successfully completed.

X 0 -  
- - -  
- - -

Around X. to play : EXECUTE play('X', x, y);

PL/SQL procedure successfully completed.

Oracle SQL Developer: DMDD\_Practice

File Edit View Navigate Run Source Team Tools Window Help

Connections: DB\_DMDD.sql, DB\_DMDD\_Admin.sql, DMDD\_Test.sql, DB\_DMDD\_Admin.sql, DMDD\_Practice.sql, Welcome Page, DMDD\_Practice

Oracle Connections: DB\_DMDD, DB\_DMDD\_Admin, DMDD\_Assignment3, DMDD\_Man, DMDD\_Practice, DMDD\_Test, Database Schema Service Connection

Worksheet: Query Builder

```

EXECUTE game_test;
EXECUTE play('X', 1, 1);
EXECUTE play('O', 2, 1);
EXECUTE play('X', 3, 1);
EXECUTE play('O', 2, 2);
EXECUTE play('X', 4, 2);
EXECUTE play('O', 3, 2);
EXECUTE play('X', 2, 3);
EXECUTE play('O', 1, 3);
EXECUTE play('X', 3, 3);

```

Script Output: Query Result

Task completed in 2.24 seconds

```

--
--
--
The Tic_tac_toe game is ready to play : EXECUTE play('X', x, y);

PL/SQL procedure successfully completed.

X --
--
--
Around O. to play : EXECUTE play('O', x, y);

PL/SQL procedure successfully completed.

X O --
--
--
Around X. to play : EXECUTE play('X', x, y);

PL/SQL procedure successfully completed.

X O X
--
--
--
Around O. to play : EXECUTE play('O', x, y);

```

Reports: All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, User Defined Reports

Windows Taskbar: 9:06 PM 4/6/2022

Oracle SQL Developer: DMDD\_Practice

File Edit View Navigate Run Source Team Tools Window Help

Connections: DB\_DMDD.sql, DB\_DMDD\_Admin.sql, DMDD\_Test.sql, DB\_DMDD\_Admin.sql, DMDD\_Practice.sql, Welcome Page, DMDD\_Practice

Oracle Connections: DB\_DMDD, DB\_DMDD\_Admin, DMDD\_Assignment3, DMDD\_Man, DMDD\_Practice, DMDD\_Test, Database Schema Service Connection

Worksheet: Query Builder

```

EXECUTE game_test;
EXECUTE play('X', 1, 1);
EXECUTE play('O', 2, 1);
EXECUTE play('X', 3, 1);
EXECUTE play('O', 2, 2);
EXECUTE play('X', 1, 2);
EXECUTE play('O', 3, 2);
EXECUTE play('X', 2, 3);
EXECUTE play('O', 1, 3);
EXECUTE play('X', 3, 3);

```

Script Output: Query Result

Task completed in 2.24 seconds

```

X O X
O --
--
Around X. to play : EXECUTE play('X', x, y);

PL/SQL procedure successfully completed.

X O X
X O --
--
--
Around O. to play : EXECUTE play('O', x, y);

PL/SQL procedure successfully completed.

X O X
X O O
--
--
Around X. to play : EXECUTE play('X', x, y);

PL/SQL procedure successfully completed.

X O X
X O O
X --
--
Around O. to play : EXECUTE play('O', x, y);

```

Reports: All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, User Defined Reports

Windows Taskbar: 9:07 PM 4/6/2022

Oracle SQL Developer: DMDD\_Practice

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Oracle Connections
  - DB\_DMDD
  - DB\_DMDD\_Admin
  - DMDD\_Assignment3
  - DMDD\_Man
  - DMDD\_Practice
  - DMDD\_Test
  - Database Schema Service Connection

Worksheet Query Builder

```
EXECUTE game_test;
EXECUTE play('X', 1, 1);
EXECUTE play('O', 2, 1);
EXECUTE play('X', 3, 1);
EXECUTE play('O', 2, 2);
EXECUTE play('X', 1, 2);
EXECUTE play('O', 3, 2);
EXECUTE play('X', 2, 3);
EXECUTE play('O', 1, 3);
EXECUTE play('X', 3, 3);
```

Script Output Query Result

Task completed in 2.24 seconds

PL/SQL procedure successfully completed.

```
X O X
X O O
--
```

Around X. to play : EXECUTE play('X', x, y):

PL/SQL procedure successfully completed.

```
X O X
X O O
_ X _
```

Around O. to play : EXECUTE play('O', x, y):

PL/SQL procedure successfully completed.

```
X O X
X O O
O X _
```

Around X. to play : EXECUTE play('X', x, y):

PL/SQL procedure successfully completed.

Reports

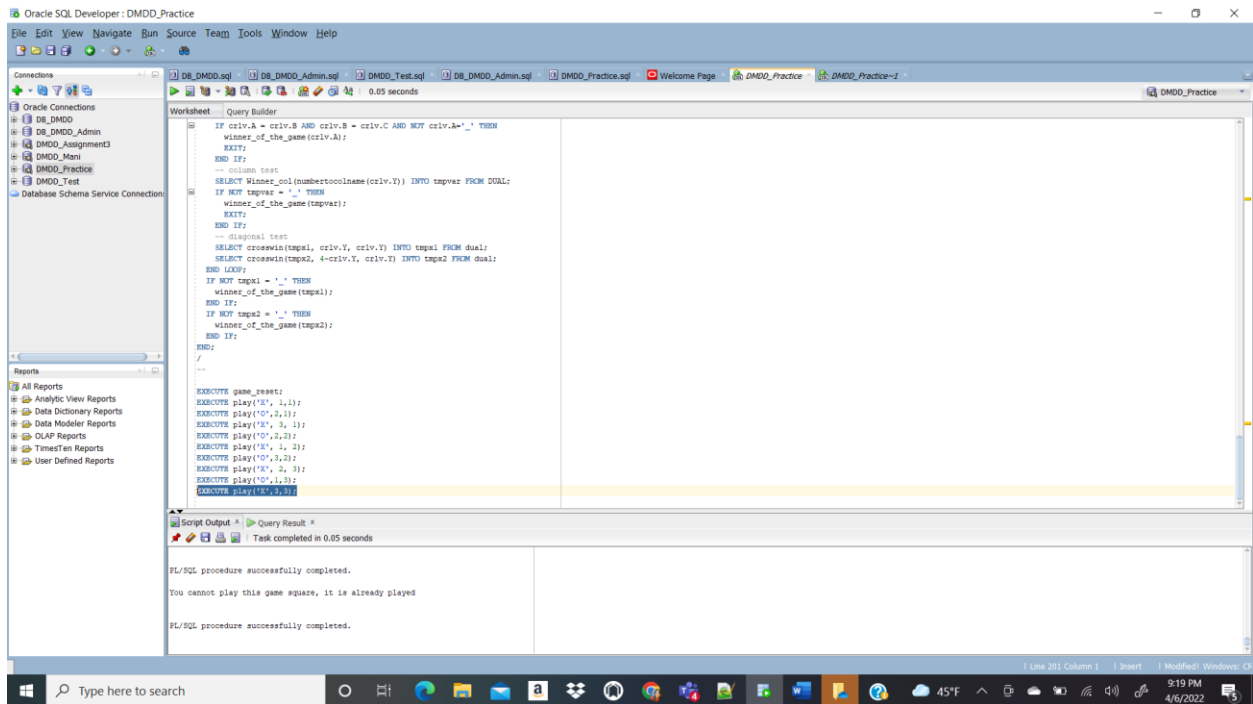
- All Reports
- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimesTen Reports
- User Defined Reports

Line 200 Column 23 | Smart | Modified: Windows: CP

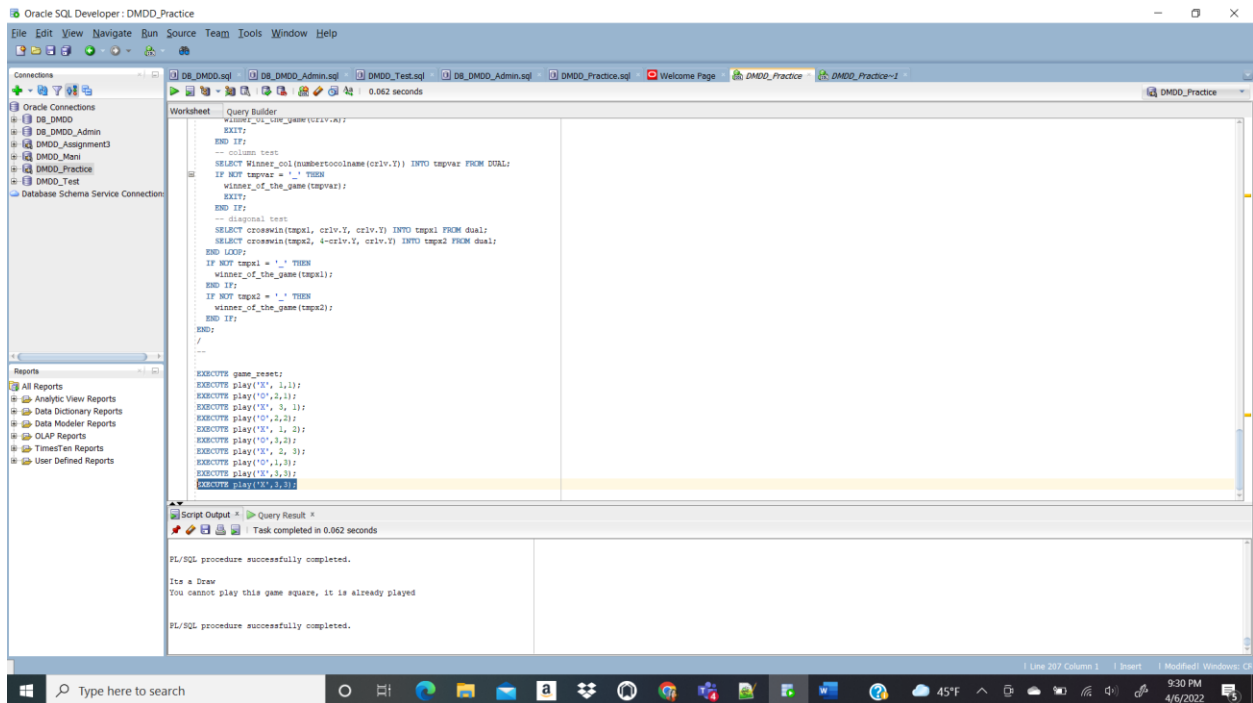
Type here to search

9:07 PM 4/6/2022

### Case 3: If Position is not reachable



### Case: If the Game has been Drawn



## If one of the player Won:

SQL Worksheet

ClearFindActionsSaveRun

```
209 END IF;
210 END;
211 /
212 --
213
214 select * from Assignment3;
215
216 EXECUTE reset_game;
217
218 EXECUTE play('X', 1, 1);
219 EXECUTE play('O', 2, 1);
220 EXECUTE play('X', 3, 1);
221 EXECUTE play('O', 2, 2);
222 EXECUTE play('X', 1, 2);
223 EXECUTE play('O', 2, 3);
224
225
226
227
```

Statement processed.  
X O X  
X O \_  
\_ O \_  
The player O won !!