

### Assignment 3: Haarika Sai Katlaparthi

set serveroutput on;

**-- Creation of the Assignement3 table if it does not exist**

```
DECLARE
  num1 NUMBER;
BEGIN
  SELECT count(*) INTO num1 FROM user_tables
    WHERE TABLE_NAME = 'Assignement3';
  IF num1 = 0 THEN
    EXECUTE IMMEDIATE 'CREATE TABLE Assignement3(
      y NUMBER,
      A CHAR,
      B CHAR,
      C CHAR
    )';
  END IF;
END;
/
```

**-- Number to column name conversion function**

```
CREATE OR REPLACE FUNCTION ColName(n1 IN NUMBER)
RETURN CHAR
IS
BEGIN
  IF n1=1 THEN
    RETURN 'A';
  ELSIF n1=2 THEN
    RETURN 'B';
  ELSIF n1=3 THEN
    RETURN 'C';
  ELSE
    RETURN '_';
  END IF;
END;
/
```

**-- Procedure to display the game board**

```
CREATE OR REPLACE PROCEDURE show_game IS
BEGIN
  dbms_output.enable(1000000);
  dbms_output.put_line(' ');
  FOR ll in (SELECT * FROM Assignement3 ORDER BY Y) LOOP
    dbms_output.put_line('  ' || ll.A || ' ' || ll.B || ' ' || ll.C);
  END LOOP;
  dbms_output.put_line(' ');
END;
```

/

**-- Game reset procedure**

```
CREATE OR REPLACE PROCEDURE reset_game IS
a1 NUMBER;
BEGIN
  DELETE FROM Assignement3;
  FOR a1 in 1..3 LOOP
    INSERT INTO Assignement3 VALUES (a1,'_','_','_');
  END LOOP;
  dbms_output.enable(1000000);
  show_game();
  dbms_output.put_line('The game is ready to play : EXECUTE play("X", x, y);');
END;
/
```

**-- Procedure to play**

```
CREATE OR REPLACE PROCEDURE play(s1 IN VARCHAR2, colnum IN NUMBER, rownumm IN NUMBER) IS
val Assignement3.a%type;
col1 CHAR;
s2 CHAR;
BEGIN
  SELECT ColName(colnum) INTO col1 FROM DUAL;
  EXECUTE IMMEDIATE ('SELECT ' || col1 || ' FROM Assignement3 WHERE y=' || rownumm) INTO val;
  IF val!='_' THEN
    dbms_output.enable(1000000);
    dbms_output.put_line('Its a tie');
  END IF;
  IF val='_' THEN
    EXECUTE IMMEDIATE ('UPDATE Assignement3 SET ' || col1 || '=' || s1 || ' WHERE y=' ||
rownumm);
    IF s1='X' THEN
      s2:='O';
    ELSE
      s2:='X';
    END IF;
    show_game();
    dbms_output.put_line('Around ' || s2 || ' . to play : EXECUTE play('' || s2 || '', x, y);');
  ELSE
    dbms_output.enable(1000000);
    dbms_output.put_line('You cannot play this square, it is already played');
  END IF;
END;
/
```

**-- Procedure to declare win**

```
CREATE OR REPLACE PROCEDURE winner(s1 IN VARCHAR2) IS
BEGIN
    dbms_output.enable(1000000);
    show_game();
    dbms_output.put_line('The player ' || s1 || ' won !!');
    dbms_output.put_line('-----');
    dbms_output.put_line('Starting a new game...');
    reset_game();
END;
/
```

**-- Column query creation function**

```
CREATE OR REPLACE FUNCTION wincol(numcol IN VARCHAR2, s1 IN VARCHAR2)
RETURN VARCHAR2
IS
BEGIN
    RETURN ('SELECT COUNT(*) FROM Assignment3 WHERE ' || numcol || ' = ''' || s1 || ''' AND ' || numcol
    || ' != "_"');
END;
/
```

**-- Column query creation function**

```
CREATE OR REPLACE FUNCTION wincross(numcol IN VARCHAR2, yvalue IN NUMBER)
RETURN VARCHAR2
IS
BEGIN
    RETURN ('SELECT ' || numcol || ' FROM Assignment3 WHERE y=' || yvalue);
END;
/
```

**-- column test function**

```
CREATE OR REPLACE FUNCTION wincoll(numcol IN VARCHAR2)
RETURN CHAR
IS
    winnum NUMBER;
    r VARCHAR2(100);
BEGIN
    SELECT wincol(numcol, 'X') into r FROM DUAL;
    EXECUTE IMMEDIATE r INTO winnum;
    IF winnum=3 THEN
        RETURN 'X';
    ELSIF winnum=0 THEN
```

```

SELECT wincol(numcol, 'O') into r FROM DUAL;
EXECUTE IMMEDIATE r INTO winnum;
IF winnum=3 THEN
    RETURN 'O';
END IF;
END IF;
RETURN ' _';
END;
/

```

#### **-- Diagonal test function**

```

CREATE OR REPLACE FUNCTION wincross1(tmpx IN CHAR, numcol IN NUMBER, numrow IN NUMBER)
RETURN CHAR
IS
    tmpvar CHAR;
    tmpxvar CHAR;
    r VARCHAR2(100);
BEGIN
    SELECT wincross(ColName(numcol), numrow) 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;
/

```

#### **-- Test trigger if we win**

```

CREATE OR REPLACE TRIGGER iswinner1
AFTER UPDATE ON Assignment3
DECLARE
    CURSOR cr_ligne IS
        SELECT * FROM Assignment3 ORDER BY Y;
    crlv Assignment3%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

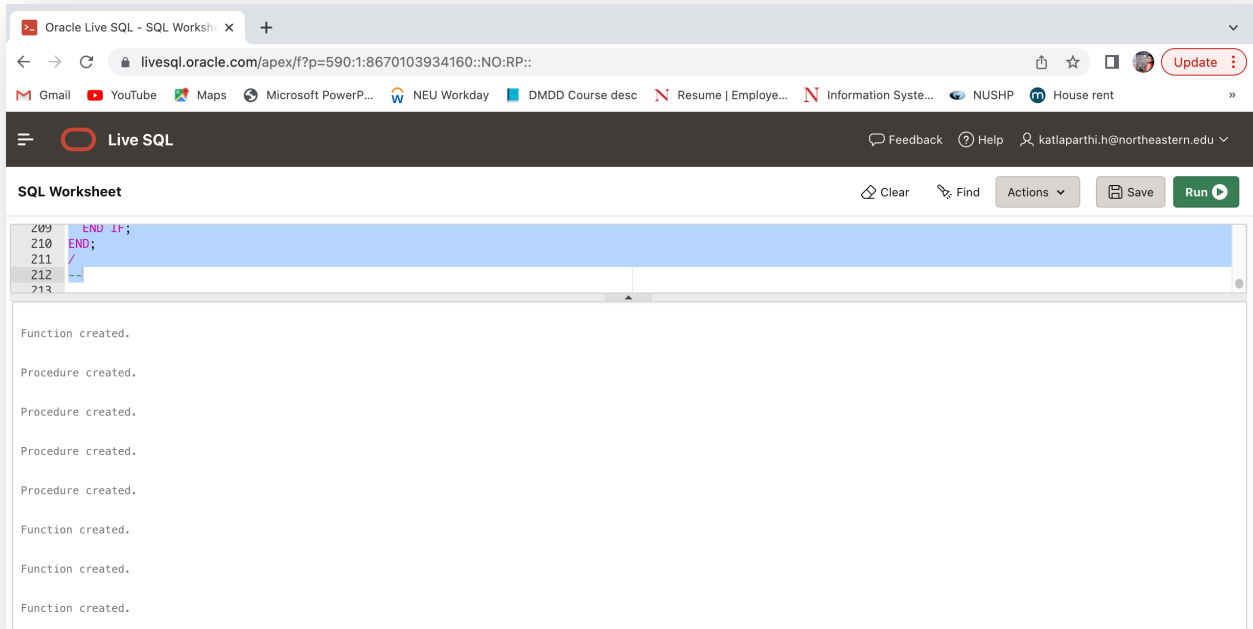
```

```

    winner(crlv.A);
    EXIT;
END IF;
-- column test
SELECT wincoll(ColName(crlv.Y)) INTO tmpvar FROM DUAL;
IF NOT tmpvar = '_' THEN
    winner(tmpvar);
    EXIT;
END IF;
-- diagonal test
SELECT wincross1(tmpx1, crlv.Y, crlv.Y) INTO tmpx1 FROM dual;
SELECT wincross1(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;
/
--

```

Executed all procedures.



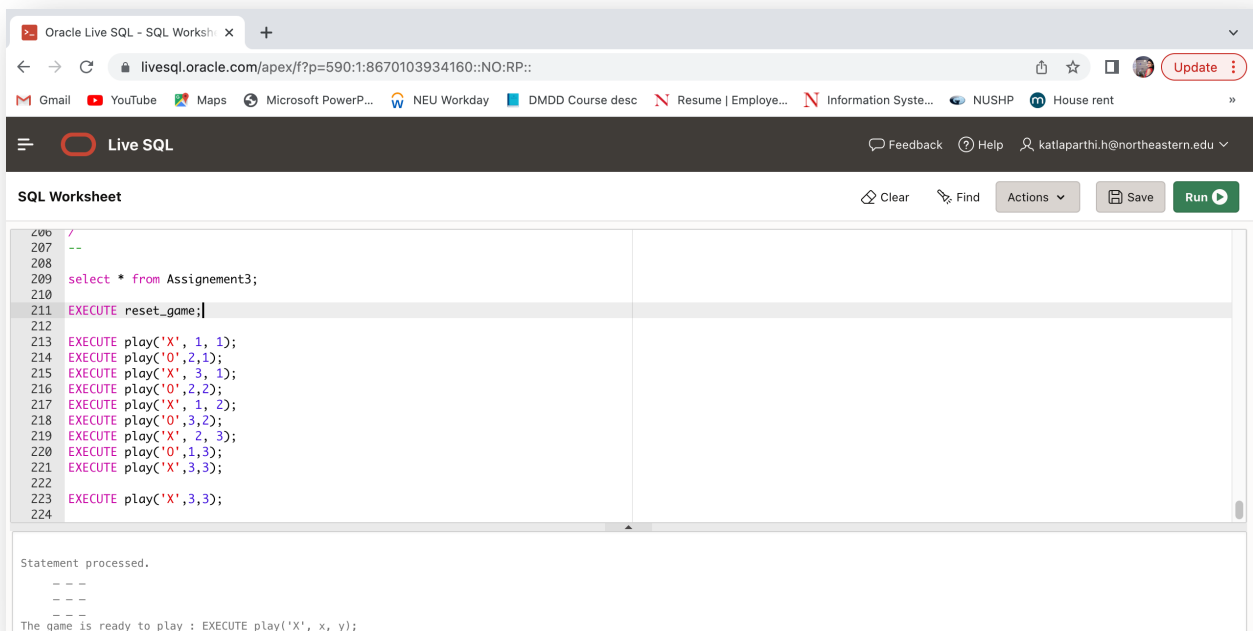
The screenshot shows the Oracle Live SQL interface. The top navigation bar includes the 'Live SQL' logo, a 'Feedback' link, a 'Help' link, and a user profile for 'katlaparthi.h@northeastern.edu'. The main area is titled 'SQL Worksheet' and contains a code editor with the following SQL code:

```
209 END IF;  
210 END;  
211 /  
212 --  
213
```

Below the code editor, the output area displays the following messages:

```
Function created.  
Procedure created.  
Procedure created.  
Procedure created.  
Procedure created.  
Function created.  
Function created.  
Function created.
```

Reset game



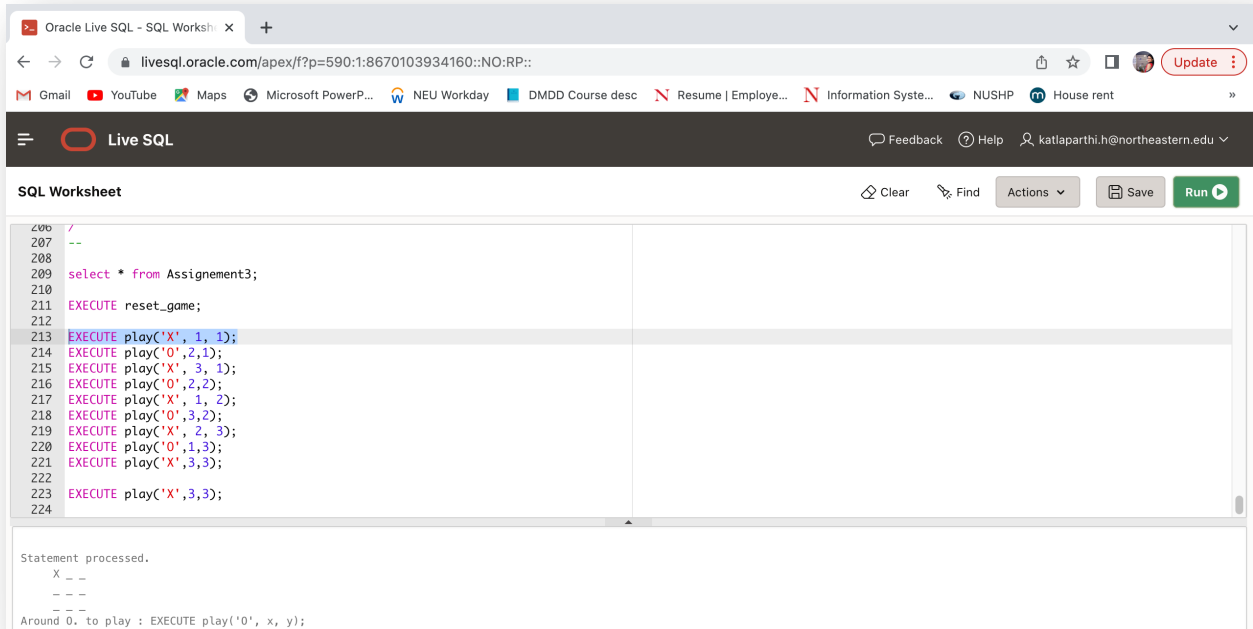
The screenshot shows the Oracle Live SQL interface. The top navigation bar is the same as the previous screenshot. The main area is titled 'SQL Worksheet' and contains a code editor with the following SQL code:

```
206 /  
207 --  
208  
209 select * from Assignment3;  
210  
211 EXECUTE reset_game;|  
212  
213 EXECUTE play('X', 1, 1);  
214 EXECUTE play('O', 2, 1);  
215 EXECUTE play('X', 3, 1);  
216 EXECUTE play('O', 2, 2);  
217 EXECUTE play('X', 1, 2);  
218 EXECUTE play('O', 3, 2);  
219 EXECUTE play('X', 2, 3);  
220 EXECUTE play('O', 1, 3);  
221 EXECUTE play('X', 3, 3);  
222  
223 EXECUTE play('X', 3, 3);  
224
```

Below the code editor, the output area displays the following messages:

```
Statement processed.  
---  
---  
---  
The game is ready to play : EXECUTE play('X', x, y);
```

## Case 1: Started Game



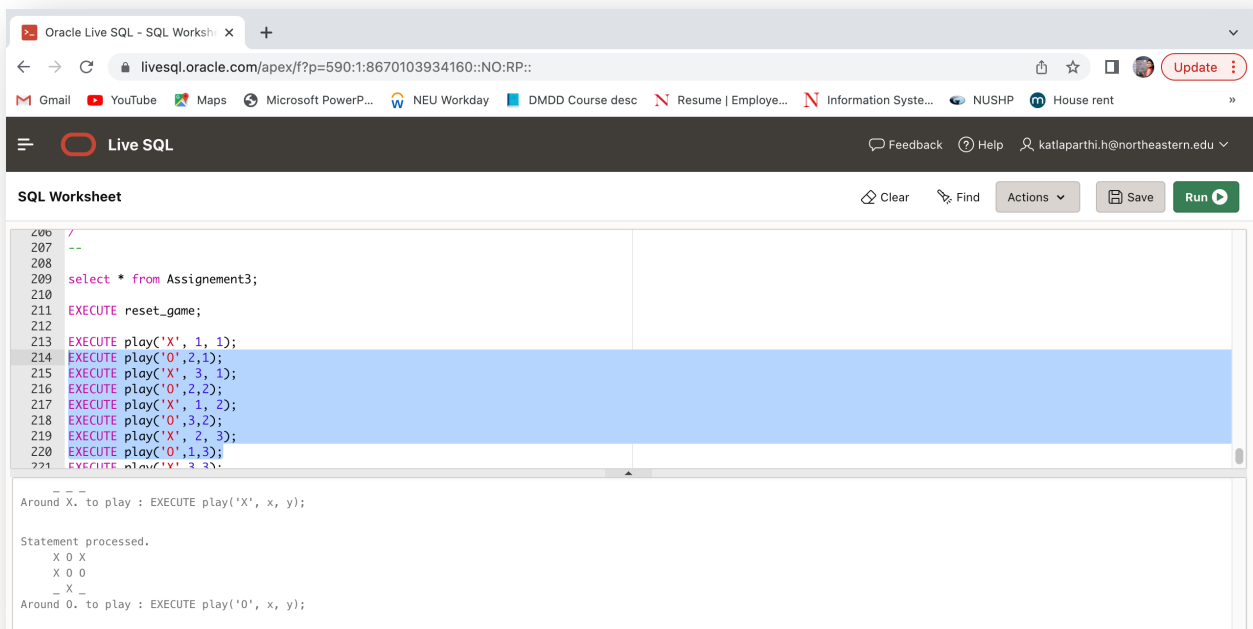
```
206 /
207 --
208
209 select * from Assignment3;
210
211 EXECUTE reset_game;
212
213 EXECUTE play('X', 1, 1);
214 EXECUTE play('0', 2, 1);
215 EXECUTE play('X', 3, 1);
216 EXECUTE play('0', 2, 2);
217 EXECUTE play('X', 1, 2);
218 EXECUTE play('0', 3, 2);
219 EXECUTE play('X', 2, 3);
220 EXECUTE play('0', 1, 3);
221 EXECUTE play('X', 3, 3);
222
223 EXECUTE play('X', 3, 3);
224
```

Statement processed.

```
X --
-- --
-- --
```

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

## Case 2: Game on



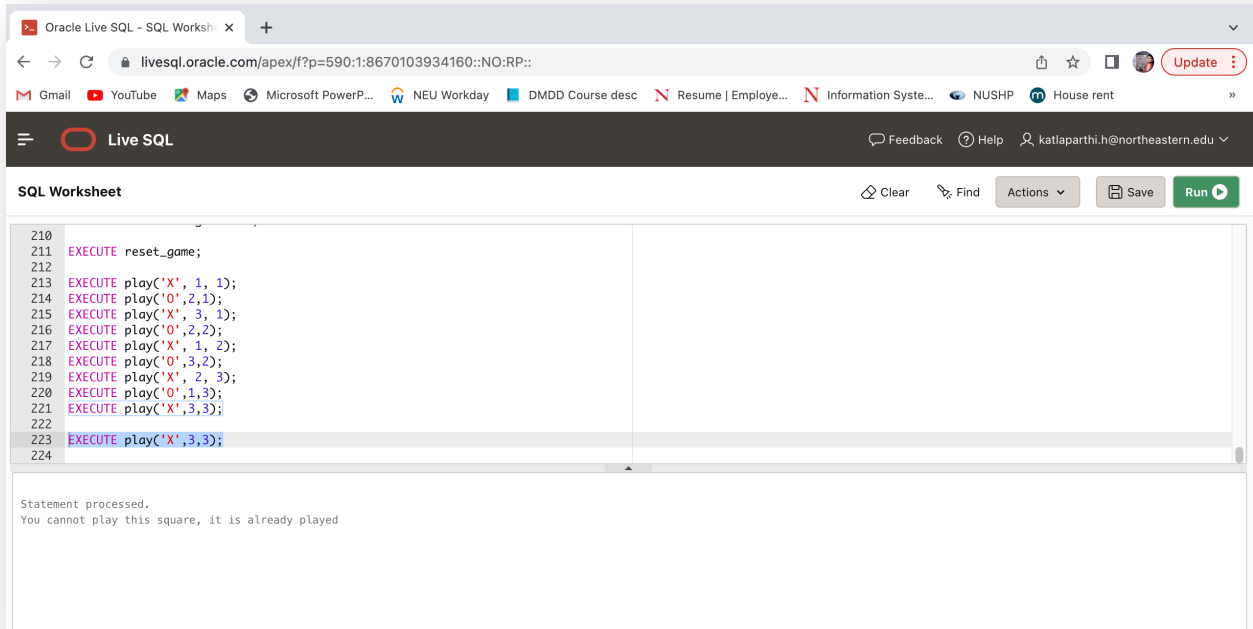
```
206 /
207 --
208
209 select * from Assignment3;
210
211 EXECUTE reset_game;
212
213 EXECUTE play('X', 1, 1);
214 EXECUTE play('0', 2, 1);
215 EXECUTE play('X', 3, 1);
216 EXECUTE play('0', 2, 2);
217 EXECUTE play('X', 1, 2);
218 EXECUTE play('0', 3, 2);
219 EXECUTE play('X', 2, 3);
220 EXECUTE play('0', 1, 3);
221 EXECUTE play('X', 3, 3);
222
```

Statement processed.

```
X 0 X
X 0 0
- X -
```

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

### Case 3: Cannot reach



The screenshot shows the Oracle Live SQL web interface. The browser address bar displays `livesql.oracle.com/apex/f?p=590:1:8670103934160::NO:RP::`. The page title is "Live SQL". The "SQL Worksheet" section contains the following SQL code:

```
210  
211 EXECUTE reset_game;  
212  
213 EXECUTE play('X', 1, 1);  
214 EXECUTE play('O', 2, 1);  
215 EXECUTE play('X', 3, 1);  
216 EXECUTE play('O', 2, 2);  
217 EXECUTE play('X', 1, 2);  
218 EXECUTE play('O', 3, 2);  
219 EXECUTE play('X', 2, 3);  
220 EXECUTE play('O', 1, 3);  
221 EXECUTE play('X', 3, 3);  
222  
223 EXECUTE play('X', 3, 3);  
224
```

Below the code, a message states: "Statement processed. You cannot play this square, it is already played".

### Case 4: Tie

```
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);
```



Oracle Live SQL - SQL Worksheet

livesql.oracle.com/apex/f?p=590:1:8670103934160::NO:RP::

Live SQL

SQL Worksheet

Clear Find Actions Save Run

```
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', 3, 2);  
224 EXECUTE play('X', 2, 3);  
225 EXECUTE play('O', 1, 3);  
226 EXECUTE play('X', 3, 3);  
227  
228  
229
```

Statement processed.  
Its a tie

## Case 5: One of the player Won

```
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', 2, 3);
```

Oracle Live SQL - SQL Worksheet

livesql.oracle.com/apex/f?p=590:1:8670103934160::NO:RP::

Live SQL

SQL Worksheet

Clear Find Actions Save Run

```
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 0 won !!