

Trick!-Crack!-Ouch! How to build a TicTacToe game in SQL

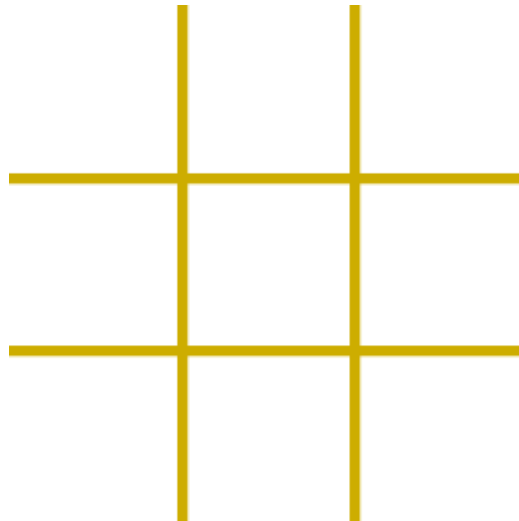
by Group 5 | Sep 20, 2019 | Group 5, HackDay Bogota 2019 | 0 comments

Building games are milestones on the programmer's way of life and that's our challenge on our first Hack Day for cohort 8 at Holberton School. As Bogota campus' group five we've chosen SQL as the programming language to build the TicTacToe game. So, in this post, we'll tell the tale about this game in this language.

Before to go:

What is TicTacToe?

A traditional paper and pencil game for two users taking turns to write 'X' or 'O' into a 3×3 grid. The first user which places three of his/her mark in horizontal, vertical or diagonal line will be the winner. Below you'll find a graphical explanation:



TicTacToe on SQL? Really?

Really. It is true that we could have chosen a(nother) programming language, but as a programmer, you have imagination and knowledge about how your preferred

programming language works as two powerful tools, so it is possible to use your language features to build things.

Why not and why SQL?

SQL was initially developed at IBM, its first apparition was in 1974 and, properly speaking, it's a domain-specific, query language, intended for handling data on relational database management systems: an approach for managing data in which information is organized into finite ordered sequences of elements (also know as tuples), stored in unique tables which consist of unique columns with unique datatype for each one, and rows containing values for each attribute. But SQL and its tools are powerful and given this condition, SQL is a declarative programming language (it indicates the way it has to be run but it doesn't provide the steps to implement it). Using SQL and its tools' power, it's possible to play TicTacToe with tables in a database.

Building and playing TicTacToe on SQL: The How-To document

Intended audience:

Users of all backgrounds. This document is intended to be fully descriptive.

Purpose:

Explain in detail how to build TicTacToe with SQL as a declarative programming language.

Prerequisites:

- Mysql Server 5.7.x

Overview:

For building this version of TicTacToe it is necessary to set a database containing two tables: one of them will handle the grid of the game and another one will establish the player's turn. Besides, it is necessary to build three functions: a first one to set the players' movements, a second one to check if there is a victory checking the recently updated values on the grid and a third one to erase all values on it.

Steps:

1. Create a database.
2. Create a grid table attending the 3×3 structure with four columns (three for the x-axis and the remaining one for the y-axis) In our case, the data type expected for each one of

the columns is string and the values of the three rows were set as NULL by default. After that, insert three rows filling the field intended to be the y-axis column.

3. Create a turn table with a unique column with string as its datatype. After that, insert a row using the mark for the first player. In our case, we chose 'X'.

4. Define functions (the player_move procedure): SQL powers as declarative programming language lies in the procedure feature, useful to set routines for different cases. In our case, we ask the user for three values, the mark and the column – row combination to store the first one. We also set conditions for checking the 'X' or 'O' strings as the only input as marks and for checking the right column and rows input. After that, we check if turns (warning if the entered turn is not the right one) and we also check if the column- row value is still NULL (every value can be updated only once). If all the previous conditions were met, the mark will be stored in the designated position and the value of the turn table will be changed for the other player.

```
-- Check if move is validated
SET R1 = 0;
IF 'A' = p_column THEN
    SET R1 = (SELECT COUNT(*) FROM TicTacToe WHERE A IS NULL AND ID = p_row);
END IF;
IF 'B' = p_column THEN
    SET R1 = (SELECT COUNT(*) FROM TicTacToe WHERE B IS NULL AND ID = p_row);
END IF;
IF 'C' = p_column THEN
    SET R1 = (SELECT COUNT(*) FROM TicTacToe WHERE C IS NULL AND ID = p_row);
END IF;
```

Our implementation for checking if the column-row provided is still NULL

5. Define functions (the check_victory procedure): After the player move function runs, this check victory function will evaluate the recently updated values on the grid table to determine if there's a line of one of the marks in the horizontal, vertical or diagonal way. If that's the case, it will be printed who's the winner. Otherwise, it will print that the game is still running.

6. Define functions (the reset_board procedure): After a winner is declared, or in case of a tie, you can reset all the values on the grid to the default NULL ones using this function. You can find our full implementation [here](#).

How to play TicTacToe on MySQL:

After setting all the tables and functions, you'll be able to start playing calling the `player_move` function, sending the mark of the player who has the turn as the first field, the desired column on as the second input and the desired row as the final one. You can check the values in the grid table every time `player_move` is executed or using :

*SELECT * from gridtablename;*

Screenshots

```
MariaDB [game]> CALL ttt_PlayerMove('X','A',3);
+-----+
| CONCAT('This turn belongs to player ', (SELECT turn FROM ttt_PlayerTurn), '!!') |
+-----+
| This turn belongs to player 0! |
+-----+
1 row in set (0.00 sec)

+-----+
| ID | A | B | C | Result |
+-----+
| 1 | X | NULL | NULL | Player X is victorious! |
| 2 | X | 0 | 0 | Player X is victorious! |
| 3 | X | NULL | NULL | Player X is victorious! |
+-----+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [game]> CALL ttt_PlayerMove('O','B',2);
+-----+
| CONCAT('Good, ', p_move, ' have played!') |
+-----+
| Good, 0 have played! |
+-----+
1 row in set (0.00 sec)

+-----+
| ID | A | B | C | Result |
+-----+
| 1 | X | NULL | NULL | Game is still ongoing |
| 2 | X | 0 | 0 | Game is still ongoing |
| 3 | NULL | NULL | NULL | Game is still ongoing |
+-----+
3 rows in set (0.11 sec)

Query OK, 0 rows affected (0.11 sec)
```

```

display all possible states. (y or n)
MariaDB [game]> CALL ttt_ResetBoard();
Query OK, 0 rows affected (0.09 sec)

MariaDB [game]> CALL ttt_PlayerMove('X','A',1);
+-----+
| CONCAT('Good, ', p_move, ' have played!') |
+-----+
| Good, X have played!                        |
+-----+
1 row in set (0.00 sec)

+-----+-----+-----+-----+-----+
| ID | A   | B   | C   | Result                |
+-----+-----+-----+-----+-----+
| 1  | X   | NULL | NULL | Game is still ongoing |
| 2  | NULL | NULL | NULL | Game is still ongoing |
| 3  | NULL | NULL | NULL | Game is still ongoing |
+-----+-----+-----+-----+-----+
3 rows in set (0.10 sec)

Query OK, 0 rows affected (0.10 sec)

```

Tip:

To run this program it is required to log in into MySQL as a user with all permissions to write and read. Otherwise, we recommend trying this with the root user.

Resources:

[Lifewire: Definition of database relation](#)

[Techopedia: Relational model](#)

[Wikipedia: SQL](#)

[Puppet: Declarative language: modeling instead of scripting](#)

[Wikipedia: Tuple](#)

[Wikipedia: TicTacToe](#)

[The Apache Forrest Project: How to write a How-to](#)

[Code Review: SQL Tic-Tac-Toe attempt](#)

[Youtube: What is Database & SQL?](#)

Recent Posts

Trick!-Crack!-Ouch! How to build a TicTacToe game in SQL

TicTacToe with Elixir Language

How to write a Tic Tac Toe in Scala?

TicTacToe in R – Awesome !!!

Recent Comments

Archives

September 2019

Categories

Group 5

HackDay Bogota 2019



Designed by **Elegant Themes** | Powered by **WordPress**