I'm re-submitting this because I misread the directions at first and I was going to do a simpler version, but I see I need a many-to-many relationship and at least four entities, which I lacked.

I will be making a database for the western version of the board game Mancala ("Kalah").

https://en.wikipedia.org/wiki/Kalah

In Kalah, two players face opposite sides of a long board which contains 6 small pits on both long sides (each player facing one long side), and a big end-zone on either short side. At the start of the game, each pit contains 4 "seeds." Like this:

```
   <--- North
------------------------
 4   4   4   4   4   4

 0               0   <— end zone

 4   4   4   4   4   4
------------------------
       South --->
```

The goal is to have the most seeds into your end-zone when the game ends ("your" end-zone is the end-zone that's to the right of you). On your turn, you pick up all of the seeds in one of your pits ("your" pits are the six closest to you) and you drop them one-by-one in each of the pits as you pass them (moving counter-clocking), including both end-zones. If your final seed lands in your end-zone, you get to go again. If your final seed lands in an empty pit of yours, you get to take all of the seeds in the pit opposite that pit and put them in your end-zone.

This is for a website that would allow users to create and join many Kalah games.

The entities I will have in my database:

Users— This is people who are registered with an account in the database to play the game.

Games—Games have up to two users who are playing the game, and two player entities to represent the user during gameplay.

Players— players have a user that's playing them, a name (which is either player1 or player2), 1 game, 6 pits (p1-p6), an end-zone, and a bool "turn."

Pits— They will have a seed-count and a name from p1-p6

End-zones— They will have a seed-count, and a name of either p1Endzone or p2Endzone

The relationships:

Users are members of games. They can be in as many games as they want, and each game will eventually have multiple (2) users as users join. This is a many-to-many relationship. (I believe that two is enough to qualify as many, seeing as it is multiple).

A player (the entity a user has for particular game) belongs to a user. However, a user has many players, as they play different games. This is a one-to-many relationship.

Players also belong to a game. This is a one-to-many relationship, seeing as a game has multiple players, but the player has only one game.

Pits belong to players. a pit can only belong to one player, but a player has six pits. This is a one-to-many relationship.

End zones belong to players. An end-zone can only belong to one player, and each player has only one end-zone. This is a one-to-one relationship.