

Colleen Minor

CS 340: Databases

Final Project

Summer 2016

Outline

For my database, I decided to stick with my original plan of making an online multiplayer Mancala game. My final project is pretty close to what I originally outlined in my project proposal: The entities are games, users, players, pits, and endzones, and there is also a table called `player_game` that is used for various purposes.

The game is very simple and is meant to be played by two people. If you want to test it you can just make different accounts in different browsers (you can only have one session per browser at a time) and make a game in one browser, then join it the other, and play against yourself. The game is pretty simple and all you really have to know is that you drop the pits counter-clockwise, and player 1's endzone is on the right, while player 2's endzone is on the left. In the browser where you're player 1, grab from any of the 6 bottom pits on your turn, and in the browser where you're player 2, grab from any of the 6 top pits on your turn. You have to hit refresh to update the status, and don't worry if it warns you about re-submitting a post. The game ends when one of the sides's pits are all empty, and the winner is declared as whoever has the most 'seeds' in their endzone at that time. If you land in an empty pit on your own side, you get to steal the seeds of your opponent in the pit opposite of it. If your last seed lands in your endzone, you get to have a second turn.

Database Outline in Words

Players play the game. They have a primary key **id**, a **name** of either Player1 or Player2, a tinyint **turn** which is set to 1 during their turn, a foreign key **game_id** which has the constraint of being a game's "id", and a **user_id** which I use as a foreign key for a user's "userId" but I did not assign a constraint because sometimes I set it to 0 while it's waiting to be assigned to a user.

Pits have an **id** of 1-12 that's used to keep track of which pit on the board they are, a **seedCount** which holds the number of seeds they contain, and an **owner** which has the constraint of being a player's id.

Endzones have a primary key **id**, a **seedCount** which holds the number of seeds they contain, a **game_id** which I use as a foreign key for a game's id but didn't set a constraint on, and a **player_id** (that's the player who's endzone it is) which I use as a key for a player's id but I also didn't set a constraint on it. The reason I didn't set these constraints was because of issues I had importing my Mancala.sql file into my database. Come to think of it, it was probably that I would put 0 in for the sample data in my SQL file and those weren't the ids of any existing games or players.

Games have a primary key **id**, a **name**, a **turn** which gets set to either Player1 or Player2, and an int called **active** which gets set to either 0 for false or 1 for true.

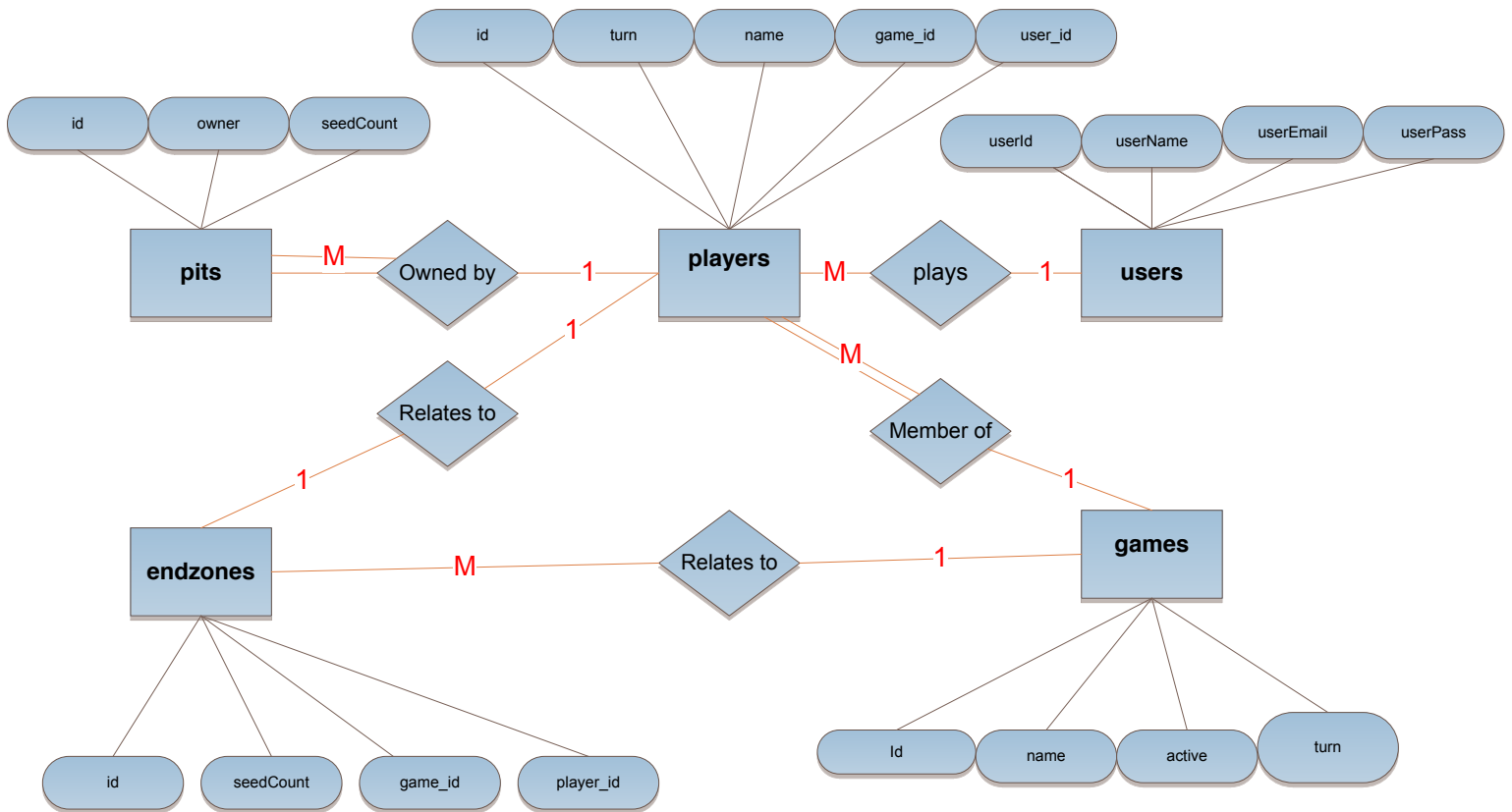
players_games is a table of rows that have a primary key **id**, a **player_id** which has the constraint that it must be a player's id, and **game_id** which has the constraint that it must be a game's id.

Users have a primary key **userId**, a **userName**, a **userEmail**, and a **userPass**. I actually didn't implement users until after I made the game. I got the code for how to make users and use sessions from here:

<http://www.codingcage.com/2015/01/user-registration-and-login-script-using-php-mysql.html>

and then I added the **user_id** to **players** to keep track of what user they belonged to. Before that I used filler ids for games and players of 1 and 2.

ER Diagram of Database



Now, you may be looking at this diagram and thinking, “where is the many-to-many relationship?” But there is one: It isn’t really **players** that play the games, it’s **users**. Users can play as many games as they want, and games have multiple (2) users. The player table ultimately serves as a user_game table, and is an entity that exists to bridge that gap. If it were a simple one-to-many relationship, I could just add a user_id field to each game or game_id field to each user and saved myself a lot of trouble.

Database Schema

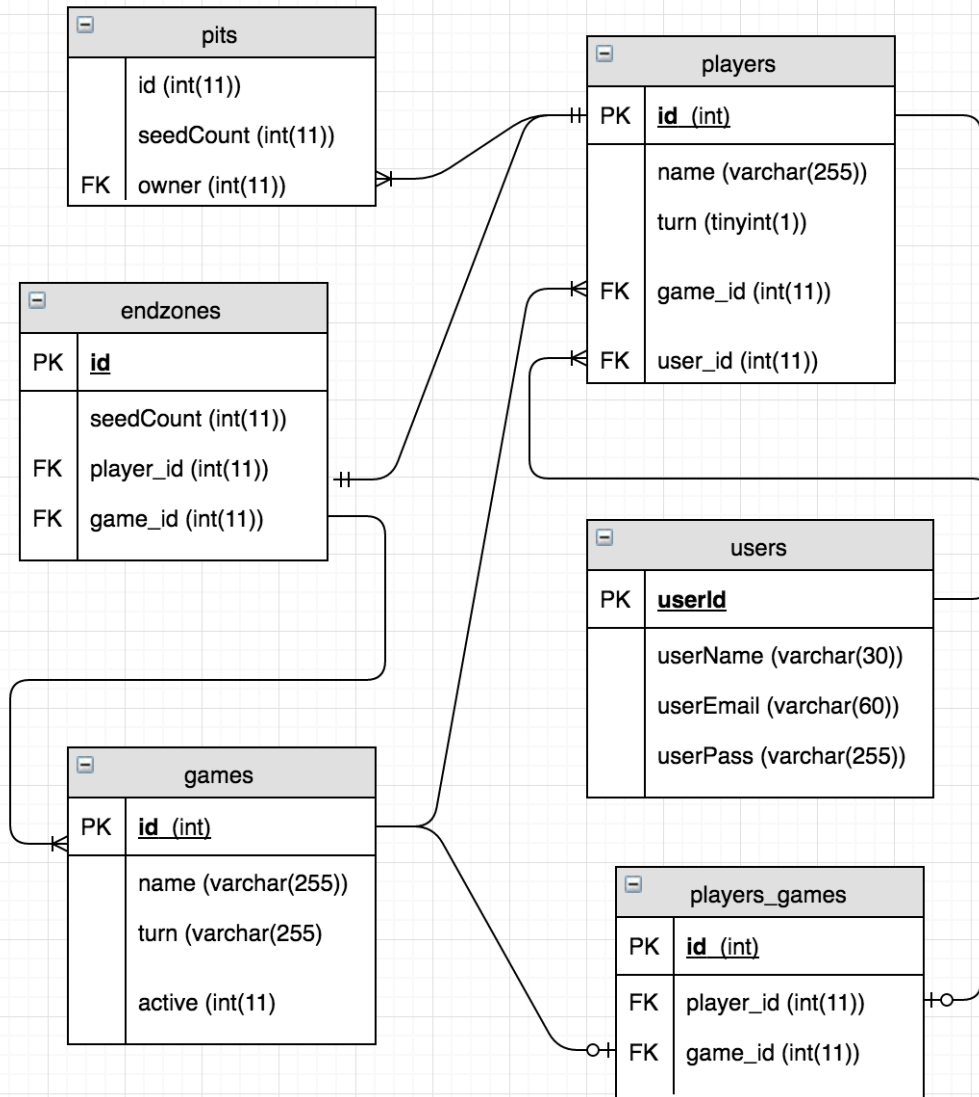


Table Creation Queries

First, here are the queries that I used to create the five tables that made the base game:

```
--  
  
-- Table structure for table `endzone`  
  
--  
  
DROP TABLE IF EXISTS `endzones`;  
  
/*!40101 SET @saved_cs_client = @@character_set_client */;  
/*!40101 SET character_set_client = utf8 */;  
  
CREATE TABLE `endzones` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `seedCount` int(11) NOT NULL default 0,  
  `game_id` int(11) NOT NULL default 0,  
  `player_id` int(11) NOT NULL default 0,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;  
  
/*!40101 SET character_set_client = @saved_cs_client */;  
  
--
```

-- Dumping data for table `endzone`

--

LOCK TABLES `endzones` WRITE;

/*!40000 ALTER TABLE `endzones` DISABLE KEYS */;

INSERT INTO `endzones` VALUES (1, 0,1,1),(2, 0,1,1);

/*!40000 ALTER TABLE `endzones` ENABLE KEYS */;

UNLOCK TABLES;

DROP TABLE IF EXISTS `players`;

/*!40101 SET @saved_cs_client = @@character_set_client */;

/*!40101 SET character_set_client = utf8 */;

CREATE TABLE `players` (

 `id` int(11) NOT NULL AUTO_INCREMENT,

 `name` varchar(255) NOT NULL,

 `turn` tinyint(1) not null DEFAULT 0,

 `game_id` int(11) DEFAULT NULL,

 `user_id` int(11) DEFAULT NULL,

 KEY `game_id` (`game_id`),

 CONSTRAINT `mancala_player_ibfk_3` FOREIGN KEY (`game_id`) REFERENCES `games` (`id`),

 PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;

/*!40101 SET character_set_client = @saved_cs_client */;

LOCK TABLES `players` WRITE;

```

/*!40000 ALTER TABLE `players` DISABLE KEYS */;

INSERT INTO `players` VALUES (1,'Player1',1, 1,1),(2,'Player2', 0, 1,1);

/*!40000 ALTER TABLE `players` ENABLE KEYS */;

UNLOCK TABLES;


DROP TABLE IF EXISTS `pits`;

/*!40101 SET @saved_cs_client = @@character_set_client */;

/*!40101 SET character_set_client = utf8 */;

CREATE TABLE `pits` (
  `id` int(11) NOT NULL,
  `seedCount` int(11) NOT NULL default 4,
  `owner` int(11) DEFAULT NULL,
  KEY `owner` (`owner`),
  CONSTRAINT `mancala_pit_ibfk_1` FOREIGN KEY (`owner`) REFERENCES `players` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;


LOCK TABLES `pits` WRITE;

/*!40000 ALTER TABLE `pits` DISABLE KEYS */;

INSERT INTO `pits` VALUES (1,4,1),(2,4,1),(3,4,1),(4,4,1),(5,4,1),(6,4,1);

INSERT INTO `pits` VALUES (7,4,2),(8,4,2),(9,4,2),(10,4,2),(11,4,2),(12,4,2);

/*!40000 ALTER TABLE `pits` ENABLE KEYS */;

UNLOCK TABLES;


DROP TABLE IF EXISTS `games`;

/*!40101 SET @saved_cs_client = @@character_set_client */;

/*!40101 SET character_set_client = utf8 */;

```

```
CREATE TABLE `games` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) NOT NULL,  
  `turn` varchar(255) NOT NULL default 'Player1',  
  `active` int(11) NOT NULL default 0,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;
```

```
LOCK TABLES `games` WRITE;  
  
/*!40000 ALTER TABLE `games` DISABLE KEYS */;  
  
INSERT INTO `games` VALUES (1,'Fun Game','Player1', 0);  
  
/*!40000 ALTER TABLE `games` ENABLE KEYS */;  
  
UNLOCK TABLES;
```

```
DROP TABLE IF EXISTS `players_games`;  
  
/*!40101 SET @saved_cs_client = @@character_set_client */;  
  
/*!40101 SET character_set_client = utf8 */;  
  
CREATE TABLE `players_games` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `player_id` int(11) NOT NULL,  
  `game_id` int(11) NOT NULL,  
  CONSTRAINT `players_games_ibfk_1` FOREIGN KEY (`player_id`) REFERENCES `players` (`id`),  
  CONSTRAINT `players_games_ibfk_2` FOREIGN KEY (`game_id`) REFERENCES `games` (`id`),  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;
```



```
LOCK TABLES `players_games` WRITE;

/*!40000 ALTER TABLE `players_games` DISABLE KEYS */;

INSERT INTO `players_games` VALUES (1,1,1);

INSERT INTO `players_games` VALUES (2,2,1);

/*!40000 ALTER TABLE `players_games` ENABLE KEYS */;

UNLOCK TABLES;
```

And here is the one I used for the users table:

```
CREATE TABLE IF NOT EXISTS `users` (
  `userId` int(11) NOT NULL AUTO_INCREMENT,
  `userName` varchar(30) NOT NULL,
  `userEmail` varchar(60) NOT NULL,
  `userPass` varchar(255) NOT NULL,
  PRIMARY KEY (`userId`),
  UNIQUE KEY `userEmail` (`userEmail`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

General Use Queries

Here I have screenshotted a selection of queries that I used in my project.

Select queries:

```
/******* SELECT QUERIES *****/
/*Getting the player names (Player1 and Player2) and the username
of the user playing them to print in board.php*/
"SELECT players.name, users.userName FROM players
JOIN players_games
ON players.id=players_games.player_id
AND players_games.game_id = [game id]
JOIN users
ON players.user_id = users.userId"

/*Getting the seedCount of a particular pit in a game in MakeMove.php*/
"SELECT pits.seedCount
FROM pits
WHERE pits.id=[id of pit]
AND pits.owner
IN ([id of player1 in the game], [id of player 2 in the game])"

/*Getting the total of player 1's seedCount to compare to player 2's
and determine the winner, in makeMove.php*/
"SELECT SUM(endzones.seedCount) FROM endzones WHERE endzones.player_id = [player 1 id]"

/*Getting the player_ids attached to a particular user in gamesUserIsPlayingPage.php*/
"SELECT players.id FROM `players` WHERE players.user_id=$user_id"

/* Getting the game id's in player_games based on a player id,
in order to print all of the games that a user is attached to in gamesUserIsPlayingPage.php*/
"SELECT players_games.game_id FROM `players_games` WHERE players_games.player_id=$player_id"

/*Selecting id of newly created game in createGameAction*/
"SELECT games.id FROM `games` WHERE games.name = '[name that user chose]'"

```

Insert queries:

```

/***** INSERT QUERIES *****/

/*Creating a new game: All of these happen in createGamePageAction.php, in this order*/
"INSERT INTO `games`(name, turn, active) VALUES ('[name chosen by user]', 'Player1', '0')"

"INSERT INTO `players`(name, turn, game_id, user_id) VALUES ('Player1', 1, $game_id, 0)"

"INSERT INTO `endzones`(seedCount, game_id, player_id) VALUES (0, $game_id, $player_id)"

/* $player_id is the id of the newly created player that was found through a SELECT query:*/
"INSERT INTO `pits` VALUES
(1,'pit1',4,$player_id),(2,'pit2',4,$player_id),(3,'pit3',4,$player_id),
(4,'pit4',4,$player_id),(5,'pit5',4,$player_id),(6,'pit6',4,$player_id);"

/* $game_id was retrieved through a SELECT query after the game was created*/
"INSERT INTO `players_games`(player_id, game_id) VALUES ($player_id, $game_id)"

/* $user_id is the id of the user currently logged in, who is creating the game*/
"INSERT INTO `players`(name, turn, game_id, user_id) VALUES ('Player2', 0, $game_id, $user_id)"

```

Update queries:

```

/***** UPDATE QUERIES *****/

/*Here are the 2 queries that are ran when the a player is added to a pre-existing game.
When a game is created, the player 2's user_id is set to 0, so to add a user you just have to
find the player in that game who's user_id is 0 and update it to the user's id. */
"UPDATE players
JOIN players_games
ON players_games.player_id = players.id
AND players_games.game_id = $gameId
SET user_id = $user_id
WHERE user_id = 0 "

/*And then you run this one to set the game to active.*/
"UPDATE games SET active = 1 WHERE id = $gameId "

/*Updating player 1's endzone when a seed lands in it*/
"UPDATE endzones SET seedCount = seedCount + 1 WHERE player_id = $player_1_id"

/*Updating player 1's endzone when a seed lands in it*/
"UPDATE pits
SET seedCount = seedCount + 1
WHERE pits.id = [id of pit that is being updated]
AND pits.owner
IN ([id of player 1 in this game], [id of player 2 in this game])"

```

I also included the page that these screenshots were taken from in my project as sample.php.

Thanks for the great course, it's been fun. :)