

Justin Holley, Louis Hwang

Team 57 - Casino Experts

Project Title: Casino Management

Project Step 6 (Portfolio Assignment)

Executive Summary:

The initial project idea was to build a database which would model all the commercial aspects of a casino, including the hotel, restaurant, and gambling operations of a casino. While working on Project Step 1, we decided to just focus our entities on the gambling aspect of a casino so that the scope of the project would not be too extensive. Our entities are based on a gaming table in a casino that allows players to make bets and play in order to try to make more money than they bet.

After first implementing our entities, we didn't change any of their names or attributes that we created in the beginning of the term since we were satisfied with how we had set them up. In Project step 1, we made corrections to the way the attributes were named such as changing first_name to firstName so that there was consistency in all the naming. In step 2, we had no problems creating the DDL.SQL file and were able to implement the sample data as needed. We made sure that all foreign keys either have an action to CASCADE or SET TO NULL on a deletion based on feedback given to us. In step 3, we were able to create the HTML pages which reflected our initial database proposal well. Based on feedback we were given, we added dropdowns for foreign keys to the user interface, but other than that the reviews had been mostly positive.

Project step 4 was the most time-consuming step for us to implement since we did not have much backend experience and there were some bugs that caused certain delete and update operations to not function properly. By the time we completed this step, we managed to get everything working and only had to fix the problem where the site had to be refreshed after every delete or update. In step 5, we decided to implement add and delete functionality to all the entities and update functionality to the GamingSessions table, with the playerID foreign key being the relationship that could be set to NULL. Based on feedback given to us, we made some fixes to have the website be consistent with our outline, with one example being to make the frontend ensure that it was not possible to add a Supplier without a name entered. After adding some styling, our website is now functional and ready to be used by any user.

Project Outline:

Our database is based on the operations of a small casino and focuses on the table gaming operations of the business. The database will store the personal information of around 100 *Players*, track 10 total *Tables* at the casino, maintain a list of 20-30 *Employees*, and include a table of 5 *Suppliers* that provide equipment for maintaining gaming operations. *Games* and *Suppliers* will also serve as entities for keeping track of expenses incurred with running the casino games. Every time a game is played at the casino the *GamingSessions* (intersection table) logs the gaming session by recording the employees working the table, the players at the table, and the financial aspect of the game outcome. It is expected that each player will play an average of 2 sessions per day, meaning that around 200 sessions will be recorded in the *GamingSessions* table per day. With this data, the casino profit can be obtained by subtracting the total expenses of employee wages, game maintenance costs, and player winnings from the total revenue obtained. Casino Management will be able to use this information to analyze the casino operations to determine how to maximize casino profit.

Database Outline:

Normalization of the Database:

After review of each of our Tables and Database, our group believes the database design is in a 3NF form and no substantive transitive dependencies exist that would result in conflicts for the intent of the database.

- **Players:** Records personal information about the casino game players
 - playerId: int(11), auto increment, unique, not NULL, PK
 - firstName: varchar(55), not NULL
 - lastName: varchar(55), not NULL
 - email: varchar(55), unique
 - birthdate: date, not NULL
 - moneySpent: decimal(10,2), not NULL
 - **Relationship:** a M:M relationship between Players and Tables with playerId as a FK in a GamingSessions intersection table.

- **Employees:** Records personal information about the casino employees
 - employeeID: int(11), auto increment, unique, not NULL, PK
 - firstName: varchar, not NULL
 - lastName: varchar, not NULL
 - birthdate: date, not NULL

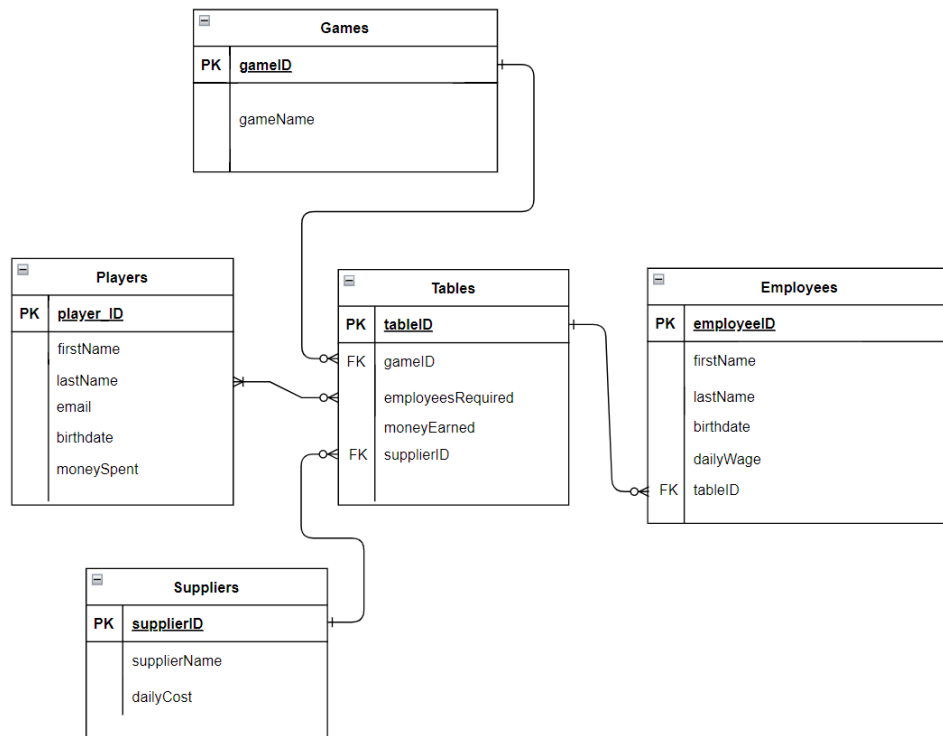
- dailyWage: decimal(10,2), not NULL
 - We specify that an employee works at one table for a day and there will be no partial days worked.
 - tableID: int(11), FK
 - **Relationship:** a M:1 relationship between Employees and Tables with tableID as a FK inside of Employees. Each employee can only work at one table, but a table can have multiple employees.
-
- **Tables:** Records information about tables at the casino including which game is played at that table, which suppliers are responsible for resupplying that table, and how much revenue the table generates. For example, one instance of a Table at the casino could be for blackjack which might need one supplier and one employee.
 - tableID: int(11), auto increment, unique, not NULL, PK
 - gameId: int(11), not Null, FK
 - employeesRequired: int(11), not NULL
 - moneyEarned: decimal(10,2) not NULL
 - supplierID: int(11), not NULL, FK
 - **Relationship:** a M:M relationship between Tables and Players with tableID as a FK in a GamingSessions intersection table.
 - **Relationship:** a 1:M relationship between Tables and Employees with tableID as a FK inside of Employees
 - **Relationship:** a M:1 relationship between Tables and Suppliers with supplierID as a FK inside of Tables
 - **Relationship:** a M:1 relationship between Tables and Games with gameId as a FK inside of Tables
-
- **Games:** Records information about the types of games at the casino. This is used to categorize the tables into the same or different categories.
 - gameId: int(11), auto increment, unique, not NULL, PK
 - gameName: varchar(55), not NULL
 - **Relationship:** a 1:M relationship between Games and Tables with gameId as a FK inside of Tables

- **Suppliers:** Records information about the suppliers responsible for providing cards, dice, and other supplies or services needed for each table of a game present at the casino.
 - supplierID: int(11), auto increment, unique, not NULL, PK
 - supplierName: varchar(55), not NULL
 - dailyCost: decimal(10,2), not NULL
 - **Relationship:** a 1:M relationship between Suppliers and Tables with supplierID as a FK inside of Tables. This is because a Supplier can help maintain multiple Tables of casino games but each Table only needs one Supplier.

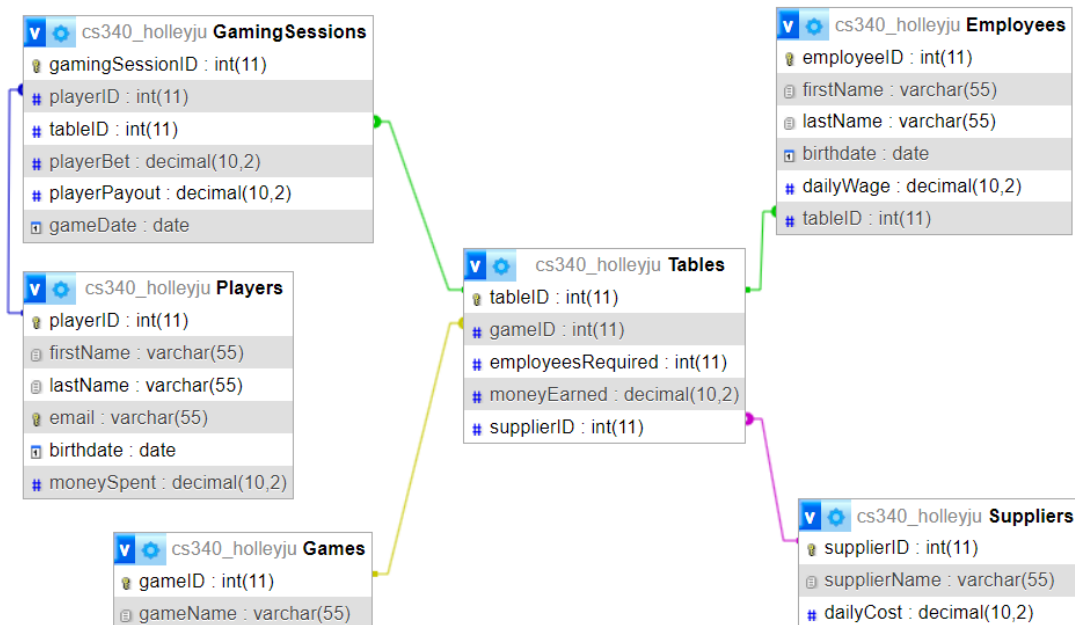
Intersection Table for the M:M Relationship that exists between Tables and Players (Not shown in ERD to display the M:M relationship between Tables and Players, however it is shown in the Schema):

- **Gaming Sessions:** This intersection table is developed to accurately track each game that is played at the casino by tracking which player and at what table in the casino the game session occurred at. This table tracks the bet and any payout that occurs at the completion of the game, as well as what date the game occurred on.
 - gamingSessionID: int(11), auto increment, unique, not NULL, PK
 - playerId: int(11), FK (From Players table)
 - tableID: int(11) not Null, FK (From Tables table)
 - playerBet: decimal(10,2) Not Null
 - playerPayout: decimal(10,2) Default 0 Not Null
 - gameDate: date not Null

Entity-Relationship Diagram:



Schema:



Sample Data:

The example data for each table is shown below.

Players:

playerID	firstName	lastName	email	birthdate	moneySpent
1	John	Jay	jayj@hello.com	1990-01-02	0.00
2	Michael	Smith	smithm@hello.com	1995-06-10	0.00
3	Bob	Cabrera	cabrerab@hello.com	1997-10-15	0.00
4	Ashley	Brown	browna@hello.com	1992-11-13	0.00
5	Mary	Jane	janem@hello.com	2001-07-09	0.00

Games:

gameID	gameName
1	Blackjack
2	Poker
3	Roulette

Suppliers:

supplierID	supplierName	dailyCost
1	Card Game Supplier	100.00
2	Slots Supplier	200.00
3	Roulette Supplier	150.00

Tables:

tableID	gameID	employeesRequired	moneyEarned	supplierID
1	1	2	0.00	1
2	1	2	0.00	1
3	2	1	0.00	1
4	3	1	0.00	3

Employees:

employeeID	firstName	lastName	birthdate	dailyWage	tableID
1	Lou	Gehrig	1988-01-13	102.00	1
2	George	Costanza	2001-04-17	105.00	1
3	Brian	Green	1999-12-31	130.00	2
4	Nancy	Drew	1986-03-29	170.00	3

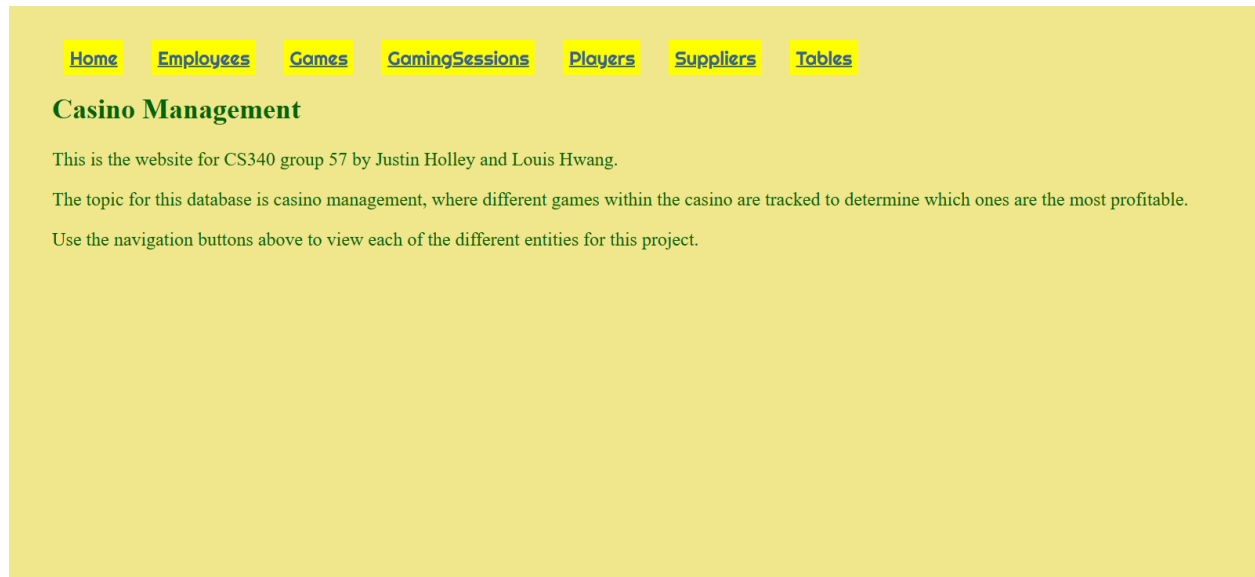
GamingSessions (Intersection table)

gamingSessionID	playerID	tableID	playerBet	playerPayout	gameDate
1	1	1	100.00	50.00	2024-01-03
2	1	1	200.00	250.00	2024-01-03
3	2	3	300.00	80.00	2024-01-03
4	3	3	200.00	150.00	2024-01-04
5	4	2	50.00	100.00	2024-01-05

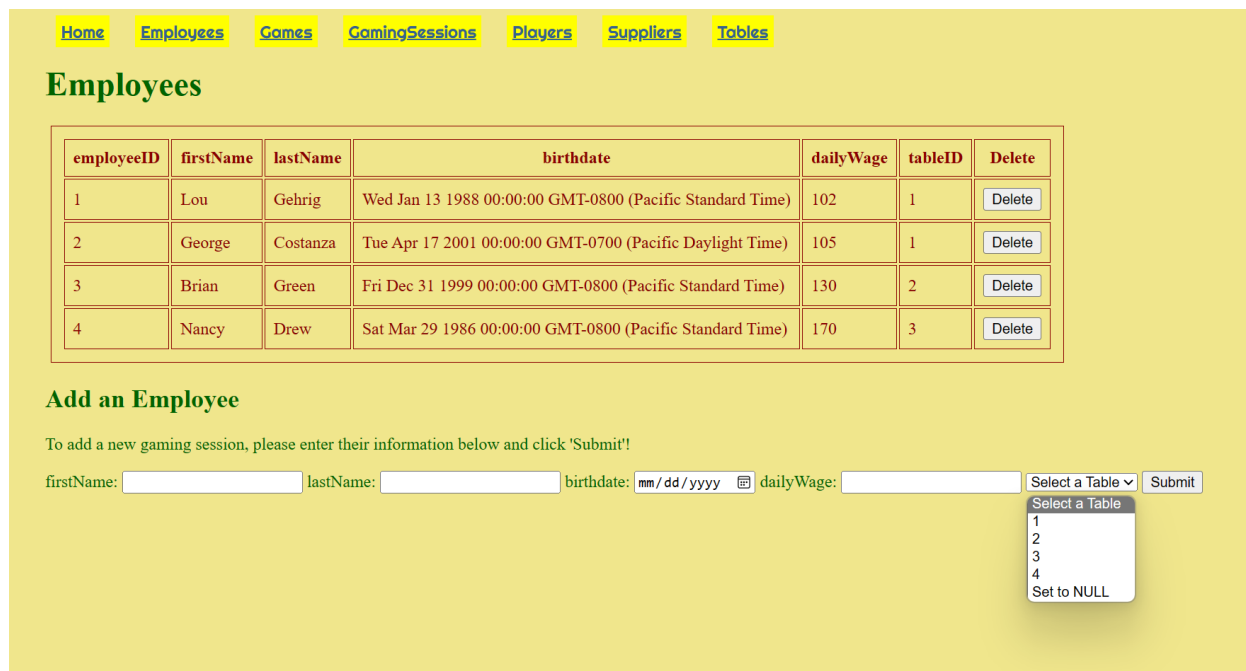
UI Screen Captures:

The screen captures below show each of the UI pages on our website.

Home Page:



Employees Page: Create, Read, and Delete functionalities have been implemented. The tableID foreign key for Employees is NULLable.



Games Page: Create, Read, and Delete functionalities have been implemented.

[Home](#) [Employees](#) [Games](#) [GamingSessions](#) [Players](#) [Suppliers](#) [Tables](#)

Games

gameID	gameName	Delete
1	Blackjack	<input type="button" value="Delete"/>
2	Poker	<input type="button" value="Delete"/>
3	Roulette	<input type="button" value="Delete"/>

Add a Game

To add a new game, please enter their information below and click 'Submit'!

gameName:

Players Page: Create, Read, and Delete functionalities have been implemented. In this screenshot, playerID #6 will be deleted later to show that GamingSessions can handle the task of deleting from an M:N relationship.

[Home](#) [Employees](#) [Games](#) [GamingSessions](#) [Players](#) [Suppliers](#) [Tables](#)

Players

playerID	firstName	lastName	email	birthdate	moneySpent	Delete
1	John	Jay	jayj@hello.com	Tue Jan 02 1990 00:00:00 GMT-0800 (Pacific Standard Time)	0	<input type="button" value="Delete"/>
2	Michael	Smith	smithm@hello.com	Sat Jun 10 1995 00:00:00 GMT-0700 (Pacific Daylight Time)	0	<input type="button" value="Delete"/>
3	Bob	Cabrera	cabrera@hello.com	Wed Oct 15 1997 00:00:00 GMT-0700 (Pacific Daylight Time)	0	<input type="button" value="Delete"/>
4	Ashley	Brown	browna@hello.com	Fri Nov 13 1992 00:00:00 GMT-0800 (Pacific Standard Time)	0	<input type="button" value="Delete"/>
5	Mary	Jane	janem@hello.com	Mon Jul 09 2001 00:00:00 GMT-0700 (Pacific Daylight Time)	0	<input type="button" value="Delete"/>
6	a	b	s@as.com	Tue Mar 07 2000 00:00:00 GMT-0800 (Pacific Standard Time)	241	<input type="button" value="Delete"/>

Add a Player

To add a new gaming session, please enter their information below and click 'Submit'!

firstName: lastName: email: birthdate: moneySpent:

GamingSessions (Intersection Table) Page: Create, Read, Update and Delete have been implemented for this table. In the screenshot below, gamingSessionID #7 has playerID 6, which will be used to test a delete cascade relationship.

[Home](#) [Employees](#) [Games](#) [GamingSessions](#) [Players](#) [Suppliers](#) [Tables](#)

Gaming Sessions

gamingSessionID	playerID	tableID	playerBet	playerPayout	gameDate	Delete
1	1	1	100	50	Wed Jan 03 2024 00:00:00 GMT-0800 (Pacific Standard Time)	Delete
2	1	1	200	250	Wed Jan 03 2024 00:00:00 GMT-0800 (Pacific Standard Time)	Delete
3	2	3	300	80	Wed Jan 03 2024 00:00:00 GMT-0800 (Pacific Standard Time)	Delete
4	3	3	200	150	Thu Jan 04 2024 00:00:00 GMT-0800 (Pacific Standard Time)	Delete
5	4	2	50	100	Fri Jan 05 2024 00:00:00 GMT-0800 (Pacific Standard Time)	Delete
7	6	2	111	333	Fri Mar 01 2024 00:00:00 GMT-0800 (Pacific Standard Time)	Delete

Add a Gaming Session

To add a new gaming session, please enter their information below and click 'Submit'!

Select a Player Select a Table playerBet: playerPayout: gameDate:

Update a Gaming Session

To update a gaming session, fill out the below information and click 'Submit'!

Select a Gaming Session ID Select a Player Select a Table playerBet: playerPayout: gameDate:

After player #6 is deleted from the Players table, the row with gamingSessionID #7 has playerID set to NULL since the part underlined in green is blank, as shown below.

[Home](#) [Employees](#) [Games](#) [GamingSessions](#) [Players](#) [Suppliers](#) [Tables](#)

Gaming Sessions

gamingSessionID	playerID	tableID	playerBet	playerPayout	gameDate	Delete
1	1	1	100	50	Wed Jan 03 2024 00:00:00 GMT-0800 (Pacific Standard Time)	Delete
2	1	1	200	250	Wed Jan 03 2024 00:00:00 GMT-0800 (Pacific Standard Time)	Delete
3	2	3	300	80	Wed Jan 03 2024 00:00:00 GMT-0800 (Pacific Standard Time)	Delete
4	3	3	200	150	Thu Jan 04 2024 00:00:00 GMT-0800 (Pacific Standard Time)	Delete
5	4	2	50	100	Fri Jan 05 2024 00:00:00 GMT-0800 (Pacific Standard Time)	Delete
7		2	111	333	Fri Mar 01 2024 00:00:00 GMT-0800 (Pacific Standard Time)	Delete

Add a Gaming Session

To add a new gaming session, please enter their information below and click 'Submit'!

Select a Player Select a Table playerBet: playerPayout: gameDate:

Update a Gaming Session

To update a gaming session, fill out the below information and click 'Submit'!

Select a Gaming Session ID Select a Player Select a Table playerBet: playerPayout: gameDate:

Finally, in GamingSessions, it is possible to use the update function to make the playerID NULL, as shown in the drop down menu in the screenshot below.

[Home](#) [Employees](#) [Games](#) [GamingSessions](#) [Players](#) [Suppliers](#) [Tables](#)

Gaming Sessions

gamingSessionID	playerID	tableID	playerBet	playerPayout	gameDate	Delete
1	1	1	100	50	Wed Jan 03 2024 00:00:00 GMT-0800 (Pacific Standard Time)	<button>Delete</button>
2	1	1	200	250	Wed Jan 03 2024 00:00:00 GMT-0800 (Pacific Standard Time)	<button>Delete</button>
3	2	3	300	80	Wed Jan 03 2024 00:00:00 GMT-0800 (Pacific Standard Time)	<button>Delete</button>
4	3	3	200	150	Thu Jan 04 2024 00:00:00 GMT-0800 (Pacific Standard Time)	<button>Delete</button>
5	4	2	50	100	Fri Jan 05 2024 00:00:00 GMT-0800 (Pacific Standard Time)	<button>Delete</button>
7		3	211	111	Tue Mar 05 2024 00:00:00 GMT-0800 (Pacific Standard Time)	<button>Delete</button>

Add a Gaming Session

To add a new gaming session, please enter their information below and click 'Submit'!

Select a Player

Select a Table

playerBet:

playerPayout:

gameDate: mm/dd/yyyy

Submit

Update a Gaming Session

To update a gaming session, fill out the below information and click 'Submit'!

Select a Gaming Session ID

Select a Player

1: John Jay
2: Michael Smith
3: Bob Cabrera
4: Ashley Brown
5: Mary Jane
Set to NULL


Select a Table

playerBet:

playerPayout:

gameDate: mm/dd/yyyy

Submit



Suppliers Page: Create, Read, and Delete functionalities have been implemented, as shown below. Note that deleting a supplier from this table will delete the rows in the Tables entity page that had the deleted supplier.

[Home](#) [Employees](#) [Games](#) [GamingSessions](#) [Players](#) [Suppliers](#) [Tables](#)

Suppliers

supplierID	supplierName	dailyCost	Delete
1	Card Game Supplier	100	<button>Delete</button>
2	Slots Supplier	200	<button>Delete</button>
3	Roulette Supplier	150	<button>Delete</button>

Add a Supplier

To add a new supplier, please enter their information below and click 'Submit'!

supplierName: dailyCost: Submit

Tables Page: Create, Read, and Delete functionalities have been implemented, as shown below.

[Home](#) [Employees](#) [Games](#) [GamingSessions](#) [Players](#) [Suppliers](#) [Tables](#)

Tables

tableID	gameID	employeesRequired	moneyEarned	supplierID	Delete
1	1	2	0	1	<button>Delete</button>
2	1	2	0	1	<button>Delete</button>
3	2	1	0	1	<button>Delete</button>
4	3	1	0	3	<button>Delete</button>

Add a Table

To add a new table, please enter their information below and click 'Submit'!

Select a Game employeesRequired: moneyEarned: Select a Supplier Submit