# Board Game Shop Database Revisions (Fixes based on Feedback from Step 1)

Chris Sexton & Shane Bliss

## Feedback by the TAs and peer reviewers (From Step 1)

---

*TA (Mahdis Safari)*

1. It's clear and to the point. You explain the business needs and how a database helps. Consider briefly mentioning how many games are in the collection or what kind of customer traffic the shop gets—just to give a bit more scope.

*(Fidella Wu)*

The overview mentions that the business currently has a problem of being unable to keep track of all the board games and which ones are rented or purchased. It mentions the fact that there are hundreds of records coming in weekly. I think that this could be a bit more specific since hundreds is a bit vague, as it is a big range of about 100s to 900s.

*TA (Mahdis Safari)*

2. No need to include unique or not NULL—those are automatically implied by PK.

*TA (Mahdis Safari)*

3. It’s better to store phone numbers as VARCHAR, since phone numbers can have special characters (like +, –, or parentheses), and leading zeroes might get dropped with an INT.

*TA (Mahdis Safari)*

4. What does currentRental: INT represent? If it’s meant to track a single current rental, that might be too limiting (what if they rent multiple games?). If it's a flag or foreign key, clarify its purpose or rename it.

   *TA (Mahdis Safari)*

5. Consider using constraints on dates (e.g., due dates for rentals) and statuses (returned vs. overdue).

   *TA (Mahdis Safari)*

6. Consider splitting the intersection table—combining rentalID and orderID can be confusing since rentals and purchases are separate actions. Separate tables would keep things clearer.

   (*Grant Wu)*
   On the same note, you may benefit from splitting the intersection table that contains rentalID, OrderID, and boardgame ID into 2 tables, as it would not make sense for a customer to rent the same board game that another customer has bought.

   *(Michael Russel)*
   One thing that might help is fixing the intersection table. Right now, both rentalID and orderID are in one table. That might be confusing because a customer wouldn’t rent and buy a game at the same time. It could be easier to have two different tables: one for rentals and one for orders.

   *(Fidella Wu)*

7. One suggestion for the datatype is increasing the 45 character limit for the board game name, since there might be ones with longer names.

*(Grant Wu)*

8. The only attributes I'm confused about are the num item and num rented attributes under the board game item. This makes it sound like each board game can have multiple copies. If this is the case, I think another entity should be created that gives each individual board game a unique id. For example, if we have multiple copies of Settlers of Catan, they should have a board game ID that identifies it to Settlers of Catan, but another ID that specifies each board game. This way we can keep track of which copy a customer has rented or bought, otherwise customers could rent the same board game ID at the same time, which doesn't make sense.

   *(Michael Russell)*
   Also, for the board games, you have numItem and numRented. This makes it seem like you're tracking how many copies you have. But what if someone rents the same game at the same time? It might be better to give each copy its own ID so you know precisely which copy was rented or bought.

   *(Fidella Wu)*
9. There is a minor inconsistency issue in the FK naming in BoardGamesHasRentalsOrders when comparing the database outline and ERD, since the ERD says BoardGames_boardGameID, but the description says boardGameID.

## Actions based on the feedback (From Step 1)

---

1. We agree that the overview needed some more specificity in terms of scale. We have added details specifying the amount of transactions the business sees in an average week.

2. We agree that it is unnecessary to include these additional constraints for primary keys. We have changed this formatting across the document.

3. We are now aware of the pitfalls in storing phone numbers as integers. We have now changed the data type to VARCHAR.

4. CurrentRental was created to represent the number of rented out board games by a single customer. The purpose of this was to integrate some limit on the amount of rentals a single customer can have at any given time. To make this more clear, we have changed the attribute name to CurrentRented.

5. We aren't aware of what other constraints may be useful for dates or statuses, so we have made no changes to them at this time.

6. We agree that it would be more clear to split BoardGamesHasRentalsOrders into two different intersection tables as each table can represent a separate action.

7. After this suggestion we scoured the internet for board games with long names and came to the conclusion that 45 characters is far too low. We have now updated it to 200, which should cover for games that have exceptionally long names.

8. We do not believe it is necessary to change these attributes in such a way that makes each individual item within the stock unique. Within their unique ID, the board games are individual items, but when it comes to the stock of those items, representing each as a unique entity seems unnecessary. Through these suggestions we have come to the understanding that including an additional table between BoardGames and the intermediary tables called Stocks would prove useful. It individualizes the BoardGames table while allowing for further expansion of product tables other than BoardGames, should the store expand to additional items in the future.

9. Good catch! We have now hopefully fixed all consistency issues.

## Upgrades to the Draft version (From Step 1)

---

We have added in a new table called Stocks. This table will be useful to the business if they decide to include products other than board games, such as card sleeves or comic books.

## Upgrades to the Draft version (Step 2)

---

We have made the email attribute in the Customers table unique since it may be helpful to identify the customer based on the email of the customer. The genre name was made unique since there can be no more than one genre with the same name. The numPlayer attribute in table BoardGames was changed to a VARCHAR data type instead of an INT data type as the number of players can vary in a board game.

# Board Game Shop Database
# (Project Outline and Database Outline)

Chris Sexton & Shane Bliss

## Overview

A local business that sells and lends board games has grown in size and is in need of a system to keep track of their transactions. The business is now receiving 400 transactions weekly. Of the 400 transactions, an estimated 50% of the transactions are rentals and the remaining transactions are orders. The local business does not have a way to keep track of the number of board games rented, bought, and in stock currently. Additionally, the business would like to have the price of the game, number of players for the game, and information about the customers such as first and last name, email, and phone number. A database will allow this business to keep track of this information and provide real time access.

## Database Outline

- **Customers:** Records the details of the Customers that are doing business.
    - customerID: INT, auto_increment, PK
    - firstName: VARCHAR(45), not NULL
    - lastName: VARCHAR(45), not NULL
    - email: VARCHAR(200), not NULL, UNIQUE
    - phoneNumber: VARCHAR(30)
    - currentRented: INT

  Relationship: The first relationship is a 1:M relationship between Customers and Rentals with customerID as a FK inside of Rentals. The second relationship is a 1:M relationship between Customers and Orders with customerID as a FK inside of Orders.

- **Rentals:** Records the board games that are rented.
    - rentalID: INT, auto_increment, PK
    - customerID: INT, not NULL, FK
    - rentalDate: DATE, not NULL
    - returnDate: DATE

  Relationship: A M:M relationship between Rentals and Stocks with rentalID as a FK inside the intersection table and stockID as a FK inside the intersection table.

- **Orders:** Records the board games that are bought.
    - orderID: INT, auto_increment, PK
    - customerID: INT, not NULL, FK
    - orderDate: DATE, not NULL

  Relationship: A M:M relationship between Orders and Stocks with orderID as a FK inside the intersection table and stockID as a FK inside the intersection table.

- **StocksHasRentals:** Intersection table between Rentals and Stocks.
    - stockID: INT, not NULL, FK
    - rentalID: INT, not NULL, FK

  Relationship: An intersection table that facilitates the M:M relationship between Rentals and Stocks. The rentalID FK and stockID FK are used to facilitate the transition.

- **StocksHasOrders:** Intersection table between Orders and Stocks.
    - stockID: INT, not NULL, FK
    - orderID: INT, not NULL, FK

  Relationship:  An intersection table that facilitates the M:M relationship between Orders and Stocks. The orderID FK and stockID FK are used to facilitate the transition.

- **Stocks:** Records the amount of a board game in stock.
    - stockID: INT, auto_increment, PK
    - boardGameID: INT, not NULL, FK
    - numItem: INT
    - numRented: INT

  Relationship: A 1:1 relationship between Stocks and BoardGames with boardGameID as a FK inside the table. Stocks is a part of two M:M relationships between Rentals and Orders.

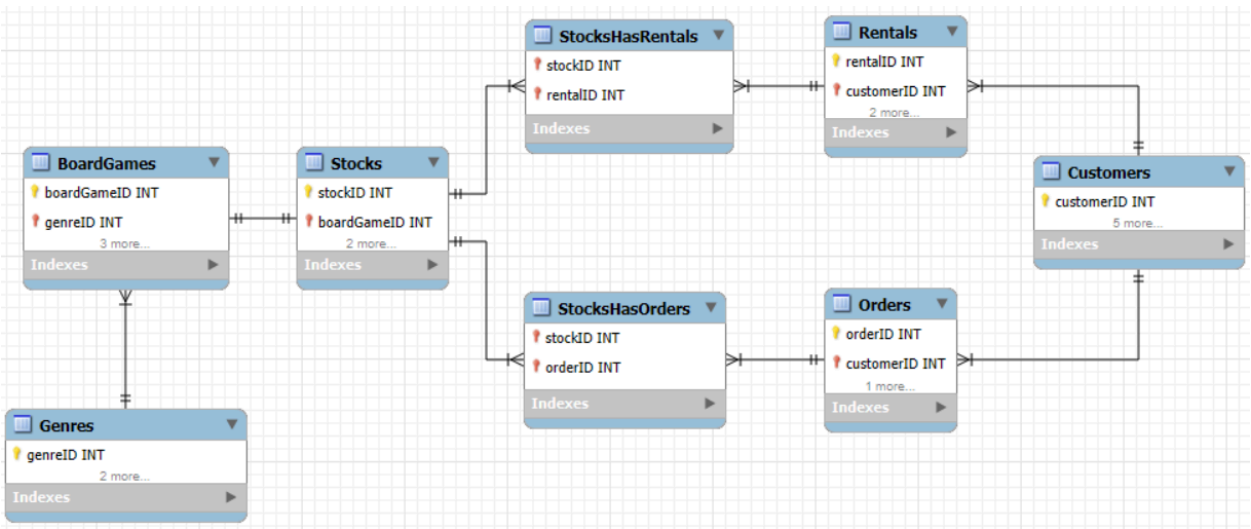
- **BoardGames:** Records the details about the board games.
    - boardGameID: INT, auto_increment, PK
    - genreID: INT, not NULL, FK
    - gameName: VARCHAR(200), not NULL
    - numPlayer: VARCHAR(10), not NULL
    - gamePrice: DECIMAL

  Relationship: A 1:1 relationship between Stocks and BoardGames with boardGameID as a FK inside the table.

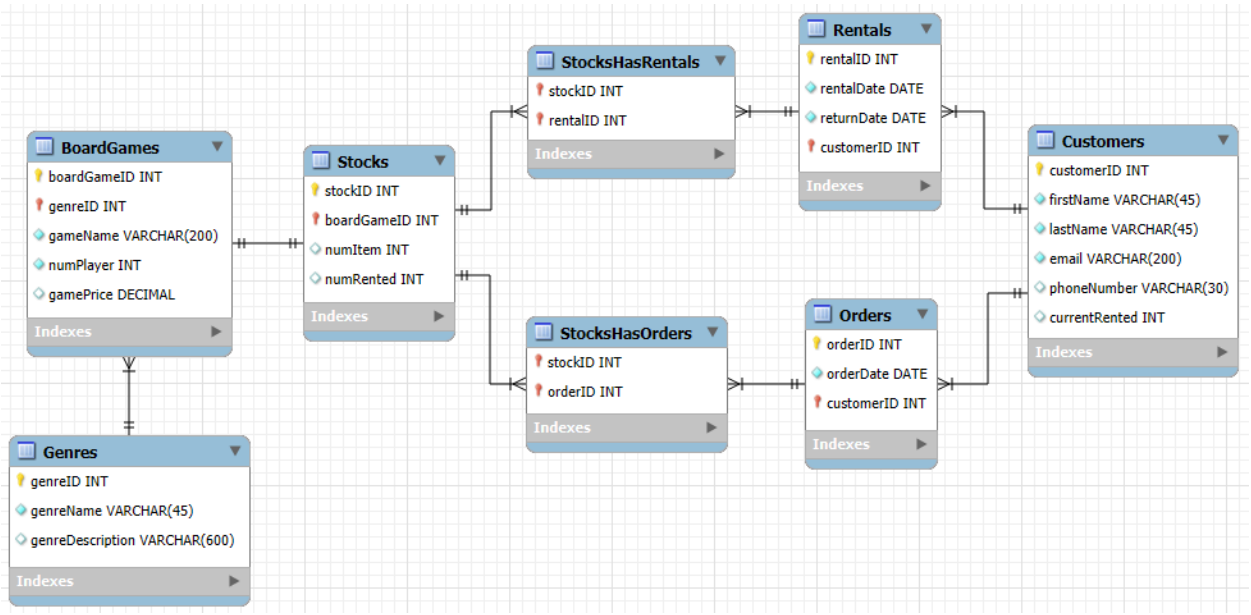
- **Genres:** Genres associated with the board games.
    - genreID: INT, auto_increment, PK
    - genreName: VARCHAR(45), not NULL, UNIQUE
    - genreDescription: VARCHAR(600)

  Relationship: A 1:M relationship between Genres and BoardGames with genreID as the FK inside of BoardGames.

## Entity-Relationship Diagram




## Schema



## Example Data

---

### Genres

| genreID | genreName | genreDescription |
|---|---|---|
| 1 | Strategy | A strategy game is a game in which the players' decision-making skills have a high significance in determining the outcome. Strategy games often require decision tree analysis, or probabilistic estimation in the case of games with chance elements. Strategy games include abstract games, with artificial rules and little or no theme, and simulations (including wargames), with rules designed to emulate and reproduce a real or fictional scenario. |
| 2 | Thematic | Thematic Games contain a strong theme which drives the overall game experience, creating a dramatic story ("narrative") similar to a book or action movie. This type of game often features player to player direct conflict (with the chance of elimination), dice rolling, and plastic miniatures. |
| 3 | Family | Family games are often created with a varied demographic in mind, so anyone aged 8-80 can play. The themes of these games can vary, but overall they tend to have a simple game play structure with clear and easy to understand rules that can be learnt and explained in a short amount of time. They allow everyone to join in for a fun game night. |
| 4 | Dexterity | Dexterity games often compete players' physical reflexes and co-ordination as a determinant of overall success. |
| 5 | Cards | Card Games use cards as its sole or central component. There are stand-alone card games, in which all the cards necessary for gameplay are purchased at once. |

Genre descriptions are attributed to BoardGameGeek.

### BoardGames

| boardGameID | genreID | gameName | numPlayer | gamePrice |
|---|---|---|---|---|
| 1 | 1 | Brass: Birmingham | 2-4 | 69.99 |
| 2 | 2 | Pandemic Legacy: Season 1 | 2-4 | 71.99 |
| 3 | 4 | KLASK | 2 | 59.99 |
| 4 | 1 | Wingspan | 1-5 | 59.99 |

**Stocks**

| stockID | boardGameID | numItem | numRented |
|---|---|---|---|
| 1 | 1 | 50 | 21 |
| 2 | 2 | 32 | 10 |
| 3 | 3 | 79 | 5 |
| 4 | 4 | 12 | 12 |

**StocksHasRentals**

| stockID | rentalID |
|---|---|
| 2 | 1 |
| 2 | 2 |
| 1 | 3 |
| 4 | 4 |

**StocksHasOrders**

| stockID | orderID |
|---|---|
| 3 | 1 |
| 4 | 2 |
| 1 | 3 |
| 2 | 4 |
| 3 | 5 |

**Rentals**

| rentalID | rentalDate | returnDate | customerID |
|---|---|---|---|
| 1 | 2025-04-21 | 2025-04-24 | 1 |
| 2 | 2025-04-22 | NULL | 2 |
| 3 | 2025-04-23 | 2025-04-31 | 2 |
| 4 | 2025-04-25 | NULL | 4 |

**Orders**

| orderID | orderDate | customerID |
|---|---|---|
| 1 | 2025-04-21 | 1 |
| 2 | 2025-04-22 | 2 |
| 3 | 2025-04-23 | 3 |
| 4 | 2025-04-24 | 4 |
| 5 | 2025-04-29 | 3 |

**Customers**

| customer ID | firstName | lastName | email | phoneNumber | current Rented |
|---|---|---|---|---|---|
| 1 | Shane | Bliss | shaneB@gmail.com | 1-428-733-5028 | 0 |
| 2 | Chris | Sexton | chrSexton@gmail.com | 593-343-7490 | 1 |
| 3 | Michael | Curry | mCurry@gmail.com | NULL | NULL |
| 4 | Danielle | Safonte | dSafonte@oregonstate.edu | 593-893-5493 | 1 |

## Citations

---

BoardGameGeek. (n.d.). https://boardgamegeek.com/

All other work is original and completed by Chris Sexton and Shane Bliss.