Board Game Shop Database Revisions Executive Summary

Chris Sexton & Shane Bliss

http://classwork.engr.oregonstate.edu:40586/

Throughout the project, the database design and implementation evolved in response to TA and peer feedback. Some of our early feedback emphasized the need for greater detail in describing the business' scale, and for smaller technical improvements, such as swapping data types to ones more appropriate, or modifying the allowed character limits. Our initial design was refined by splitting a confusing intersection table into two separate intersection tables, clarifying attribute names, and introducing a new Stocks table to support future product expansion beyond board games.

Further feedback led to improvements in schema clarity, such as removing redundant constraints, defaulting certain values, and ensuring consistency in naming and data types. The decision to keep some attributes and table structures were justified through the business' needs.

A major turning point in the project was the addition of intersection tables (StocksHasOrders and StocksHasRentals) to the user interface. Along with the newly added intersection tables came the addition and removal of several CRUD operations, making sure to not introduce data anomalies. Usability feedback prompted UI enhancements, including more readable column names and better layout for longer text fields within tables.

Overall, the project shifted from a board-game-only inventory to a scalable system capable of supporting a broader retail scope. The purpose of this change was to allow the business to introduce non-board-game products in the future without a complete redesign of the database. AI tools were used on occasion to brainstorm ideas, typically utilized when we ran into a block of some sort. This primarily consisted of inquiring about alternate approaches to a given problem.

Board Game Shop Database Revisions (Fixes based on Feedback from Steps 1-4)

Chris Sexton & Shane Bliss

Feedback by the TAs and peer reviewers (From Steps 1 - 5)

TA (Mahdis Safari) Step 1

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) Step 1

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) Step 1

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

TA (Mahdis Safari) Step 1

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) Step 1

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) Step 1

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

TA (Mahdis Safari) Step 1

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) Step 1*

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) Step 1

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) Step 1

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) Step 1

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) Step 1

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) Step 1

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.

TA (Mahdis Safari) Step 2

10. ERD often leaves out the finer details such as attributes and intersection tables. You did a good job not including all attributes in the ERD, but note that just the primary key is good enough. You should also consider leaving the intersection table out.

(Michael England) Step 2

11. There is no need to include "NULL" in the attribute fields that allow for null values. In SQL, all fields are, by default, allowed to have null values unless they are restricted by the "NOT NULL" clause. Removing this declaration would make it slightly easier to see which fields can't have null values when looking at the DDL.

(Michael England) Step 2

12. You mentioned that "email" can't be null and must be unique. What about a situation where a customer doesn't have an email? That may be rare these days, but as this isn't an online business, I think it could happen. Also, what about a couple where, for whatever reason, they want different accounts but share an email? I don't think this is necessarily a poor design decision, but I wanted to give you some more food for thought to confirm this is how you want the database designed.

(Michael England) Step 2

13. The "currentRented" field isn't strictly necessary in the Customers table. This field could possibly be useful if it is searched quite frequently, but it is easily accessible by acquiring a count for the customer from the Rentals table. Additionally, why does one customer have "NULL" in this field? I would think that this field would not be allowed to be null and would be set to a default value of "0" upon customer creation.

(Michael England) Step 2

14. The schema is very similar, but the order of fields in Rentals and Orders is different between the outline and the schema. The schema follows the DDL, but the outline shows the foreign key (customerID) as the second column in these two tables while the schema and DDL have these foreign keys as the last column.

(Michael England) Step 2

15. You may want to consider renaming Orders as Purchases. This is a very minor detail, but I would think that there are orders which are rentals and orders which are purposes. This doesn't make the database more functional, but I believe it would make it slightly more intuitive.

(Katlin Hopkins) Step 2

16. There are two FKs per intersection table. You might consider whether or not you want to define a primary key for these tables.

(Michael England) Step 2

The intersection tables don't have a primary key. While they can be referenced using the two foreign keys, I believe that best practice is to have a primary key for these tables also.

(Michael England) Step 2

17. I don't really understand the need for the Stocks table. Couldn't the stock for each game simply be listed as attributes in the BoardGames table? Having an extra table for this seems like an unnecessary addition of complexity when I can't see any reason these two fields ("numItem" and "numRented") can't simply be added as additional attributes for each board game. As an additional minor detail, it's not clear whether "numItem" includes the rentals or represents the number of physical copies that are currently at the store.

(Katlin Hopkins) Step 2

18. Is it possible for one board game to be multiple genres? If yes, how will that work in the BoardGames and Genres tables? if not, you should be good!

(Katlin Hopkins) Step 2

19. Yes, the Schema matches the ER diagram and the database outline. The only discrepancy is the relationships under BoardGames, as the connection to Genres is missing. This relationship is however covered in the Genres table relationship section.

(Katlin Hopkins) Step 2

20. The Schema has numPlayer listed as INT, but the outline, example data and SQL statements have it as varchar. Seems like you only need to update the Schema screenshot. The rest of the data types seem appropriate.

(Katlin Hopkins) Step 2

21. A few of the IDs were different between the example data and the sql queries. I thought this might be because we were auto generating IDs for the stockID, orderID, and rentalID. However, the issue is in the intersection tables - you have hardcoded an ID lookup based on a unique id that will be auto generated by the program. Maybe consider how you can/how you need to adjust this for the next few steps of the project.

(Michael England) Step 2

22. The SQL is well-structured although lightly commented. I like the inclusion of the descriptions for each table, but it might also be helpful to list the relationships between tables in the comments as well. This makes it easier to see these connections at a glance.

NOTE: Did not receive TA feedback for step 3

(Lucy Cheng) Step 3

23. Yes, there is a SELECT statement for every table. Data from each table is displayed on a different page. But not in the intersection table. There's no intersection table on the UI.

(Lucy Cheng) Step 3

The intersection table is not shown on the website, so I cannot update it - Lucy Cheng.

(Lucy Cheng) Step 3

I tried to delete a row in the Stocks table, but it did not delete it, and the StockHasOrders table is not on the website, so I believe it does not delete corresponding rows in the StockHasOrders table.

(Lucy Cheng) Step 3

You should add intersection tables on your website, and some AS aliases for your attribute name will make the data more readable.

(Berenice Rubalcaba) Step 3

Possibly, If im not mistaken I dont think I saw an explicit DELETE function that would remove things from a M:M relationship.

(Berenice Rubalcaba) Step 3

I noticed some tables were mentioned on the DDL that werent on the UI such as StocksHasRentals and StocksHasOrders. Adding those to the UI might clear up some confusion.

(Gage Shinneman) Step 3

Since the UI does not feature intersection tables, it does not appear that the UI has a DELETE that allows the user to remove things from a M:M relationship, unless I am misunderstanding the question.

(Gage Shinneman) Step 3

Since the UI does not feature intersection tables, it does not appear that the UI has a UPDATE that allows the user to select a different foreign key value to

update the intersection table with, again, unless I am misunderstanding the question.

(Natalie Nascimento) Step 3

Update forms specify specific data types on M:M tables so that they are only altered at that table's location and not any other attributes on other tables.

(Gage Shinneman) Step 3

24. My main suggestion would be to finish replacing column names from the schema with aliases to replace camel case and shortened names such as 'numRented'.

(Natalie Nascimento) Step 3

25. I would suggest, for the Genres page, to limit the table size so that the description is not dominating the whole screen. It seems quite oppressive now, but if the text were not stretch over the size of the page, I think it would make for better UI design. Also, I would recommend making the "Description" input boxes tops to be level with the top of its corresponding text, so that it doesn't look so strange next to the "Genre Name" text box. Otherwise, I feel the implementation of the queries and their UI to be satisfactory to the aims of the project.

No comments from the TA for Step 3 Final.

(Kiana Shim) Step 4

26. The UI is well made as it's easy to read and utilize. The only minor change I would suggest is to make the RESET button's styling more consistent with the rest of the web design.

(Kiana Shim) Step 4

27. As for feedback, I think implementing a helper procedure would be useful in not convoluting the data pool. Also, as you point out, you likely won't need a DELETE button on the StocksHasOrders page, but do still implement the Cascade operation (ON DELETE CASCADE).

(Holly Porter) Step 4

28. I think using a trigger or procedure to increase stock by 1 every time an order is canceled would work so long as there were safeguards against situations like say somehow ordering an out of stock item, then deleting the order and having the stock increase anyway, for instance. Other approaches might be using "SELECT count(stockID) FROM StocksHaveOrders" to get the number of orders for an item and subtracting that from a periodic total inventory to get the current available stock, or using an entity for ordered items for which you could change the status of an item to "to be delivered" when an order is placed to excluding it from available stock counts, and then using an ON DELETE constraint to reset its status to null when the associated order is deleted.

(Holly Porter) Step 4

29. Regarding the delete buttons, (if I've understood correctly) it seems like being able to delete from the intersection table instead of the Orders table would allow removing specific items from an order without canceling the whole order. I can see the argument for keeping it in there but I guess it all comes back to the whole scenario and how the store customarily handles these kinds of things.

(Niko Bransfield) Step 4

30. I did have one minor issue with the UI: whenever I wanted to update a Customer, I always needed to fill in all of the attributes on the form—even ones I wasn't changing. To help prevent input errors, perhaps the update form could allow the text input fields to be "optional", or maybe it could automatically fill in existing data whenever a Customer is selected to edit.

TA (Mahdis Safari) Step 4

31. DDL.SQL in PL/SQL to RESET data is missing.

(Jonas Field-Patton) Step 5

32. One idea is auto filling the update form with the values of the record that is being updated. I think this would be easier on the user, but it is certainly not essential.

(Zachary Krodel) Step 5

33. I did notice that you can add letters to the Phone Number field while creating or updating Customers. Data validation is not technically required for this assignment, though, just something I noticed.

(Anthony Li) Step 5

34. Some input validation might be needed for types and in case of SQL injections, but that is honestly not something significant to worry about for this class.

Actions based on the feedback (From Steps 1 - 5)

TA (Mahdis Safari) Step 1
(Fidella Wu) 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.

TA (Mahdis Safari) Step 1

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

TA (Mahdis Safari) Step 1

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

TA (Mahdis Safari) Step 1

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.

TA (Mahdis Safari) Step 1

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.

TA (Mahdis Safari) Step 1

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.

(Fidella Wu) Step 1

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.

(Grant Wu) Step 1

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.

(Fidella Wu) Step 1

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

TA (Mahdis Safari) Step 2

10. Thank you for the feedback! The ERD table has been refined to leave out the intersection tables and only include the primary keys.

(Michael England) Step 2

11. This was included based on personal preference, but you are correct that it is unnecessary. NULL will be removed for better clarity.

(Michael England) Step 2

12. We have made no changes on this matter. The purpose of email being required is that an email can be a unique identifying factor for customers to hold on to without requiring their customer ID number.

(Michael England) Step 2

13. Our reasoning for keeping currentRented is that the company deems it useful to know how many rented items a customer currently has. It also provides some basic structure as to limiting the amount of rented items a customer could have at any given time. As for the second part of the question, good catch! We initially had our scope that the business was primarily for selling games, but we have since changed the business to be more of an even split between board game sales and rentals. We felt it was unnecessary to populate currentRented with 0 in the past as it would have been a lot less common to have rental data attached to a customer at all. We have now changed it to default to 0 as it makes more sense with the current project.

(Michael England) Step 2

14. While this would improve readability, we deemed that it was an unnecessary change. The small amount of added clarity would be strictly for the developers of the project.

(Michael England) Step 2

15. While this change may improve readability for some, we deemed that it was an unnecessary change.

(Katlin Hopkins) Step 2
(Michael England) Step 2

16. The intersection tables don't have primary keys because we did not see a purpose for them in our planning, at least with our current knowledge of databases. It could be that it would be useful to have a primary key for these tables, but as we aren't sure what that use could be, we have chosen to not include them at this time.

(Michael England) Step 2

17. The purpose of having an additional table for Stocks is so that the company can introduce additional non-board game items smoothly into the database. For example, they may end up selling card sleeves at some point. This would introduce a new table which defines a new type of item that links up to the Stocks table.

(Katlin Hopkins) Step 2

18. It is not currently possible. We decided that each board game would receive a single primary genre. It could be in the future that this business may want to include multiple genres (or even subgenres), but this granularity isn't entirely necessary for the purpose of the business (eg. sorting the store by game genre).

(Katlin Hopkins) Step 2

19. While this may add clarity, we did not want to introduce redundancy in the explanations for our relationships. We figured that the redundancy may make the relationship explanations more confusing to read.

(Katlin Hopkins) Step 2

20. Thank you for noticing, it has been changed in the Schema screenshot.

(Katlin Hopkins) Step 2

21. We recognize that this change will be useful in the future, but we have decided to keep the intersection table data for now as it makes sense in the context of the assignment.

(Michael England) Step 2

22. While we agree it could be helpful to list the relationships between tables in comments, we felt it was unnecessary to add as we have the ERD and Schema diagrams to facilitate that knowledge. Additionally, if any changes were made to the diagrams, then changes would also need to be made to the code comments, which doesn't seem like an efficient system to us at this time. A comment on the values section of the file has been added for clarity.

(Lucy Cheng) Step 3
(Berenice Rubalcaba) Step 3
(Gage Shinneman) Step 3
(Natalie Nascimento) Step 3

23. Thanks everyone for pointing out that the intersection tables were left off of the UI. We appreciate the feedback. Below are all the changes made that should answer all the questions pertaining to each person:

StockHasOrders and StocksHasRentals intersection tables have been added to the UI, each with their own webpage. Each webpage has a DELETE and UPDATE form.

(Lucy Cheng) Step 3

24. Thank you, Gage. Column names have been adjusted to be more user friendly.

(Natalie Nascimento) Step 3

25. Thank you, Natalie. The Genres page's table size has been adjusted so the description is not dominating the screen. The description input box requires more text, and for now, we are leaving the boxes as is.

(Kiana Shim) Step 4

26. Thank you for the suggestion, Kiana. As far as the RESET button goes, we believe having the button colored makes sense since this button should only be pressed in the event the database needs to be reset. Offsetting the color to the same as the rest of the page could have a user accidentally press the button. In the future, the RESET button may be set to a different color depending on how far along the website gets.

(Kiana Shim) Step 4

27. Thank you for the suggestion, Kiana. We are planning on implementing a helper function for PLSQL file. We removed the CRUD operations from the tables to make the database less confusing. The other tables will update these tables as necessary.

(Holly Porter) Step 4

28. This functionality is something we plan to implement in the final version of the project. Thank you for the feedback!

(Holly Porter) Step 4

29. We don't currently see a purpose to including the delete button for the intersection tables either, so we have removed them for now. In their previous iteration they could only be used to tamper with the database. We have removed them for now and may add them back later if necessary.

(Niko Bransfield) Step 4

30. We agree that this change would make the update forms quicker to interface with. We have put this on our list of things to address and will implement it if time allows.

TA (Mahdis Safari) Step 4

31. We have now added RESET to the DDL.SQL file.

(Jonas Field-Patton) Step 5

32. Thank you for the suggestion! We decided to not implement autofill on update forms due to time constraints.

(Zachary Krodel) Step 5

(Anthony Li) Step 5

Thank you for the suggestions. Data validation would definitely be on our list of fine-tuning opportunities had we been given more time on the project. Unfortunately, data validation did not make the final cut for the submission.

Upgrades to the Draft/Final 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/Final version (From 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.

Upgrades to the Draft/Final version (From Step 3)

No upgrades outside of current scope at this time.

Upgrades to the Draft/Final version (From Step 4)

No upgrades outside of current scope at this time.

Upgrades to the Draft/Final version (From Step 5)

A default ‘none’ genre was made to ensure deleting a genre does not cascade to deleting a board game, stock, and the corresponding row on the intersection tables. If a board game row has a foreign key to genre, and that genre is deleted, the row’s genre will be moved to “none”.

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.

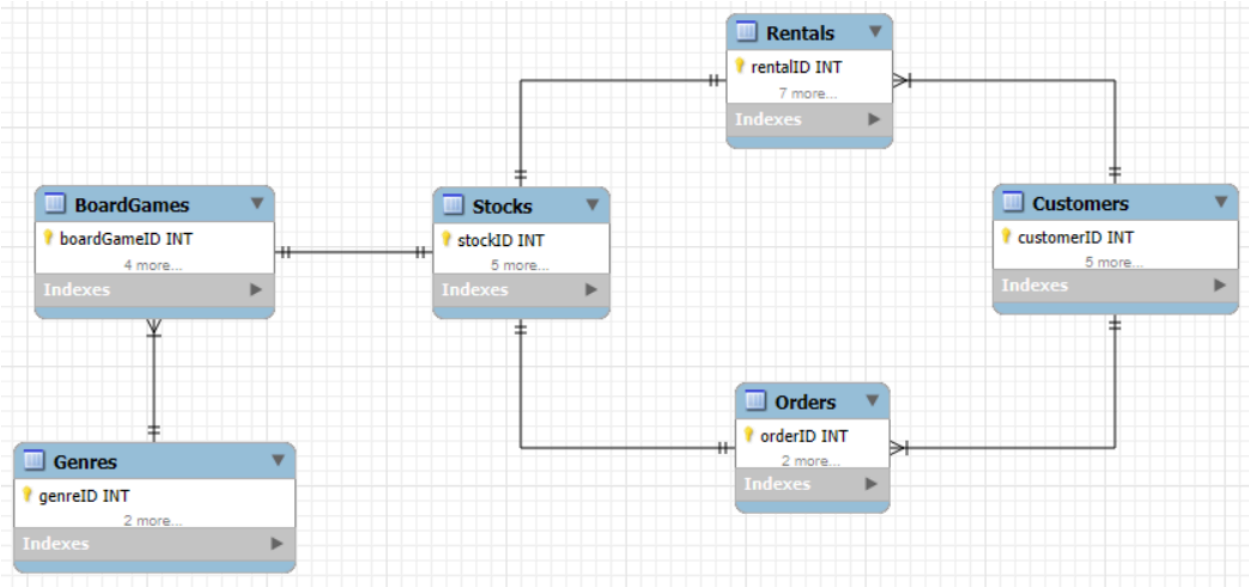
- **BoardGames:** Records the details about the board games.
    - boardGameID: INT, auto_increment, PK
    - genreID: INT, not NULL, FK
    - gameName: VARCHAR(200), not NULL
    - numPlayer: VARCHAR(10), not NULL
    - gamePrice: DECIMAL

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

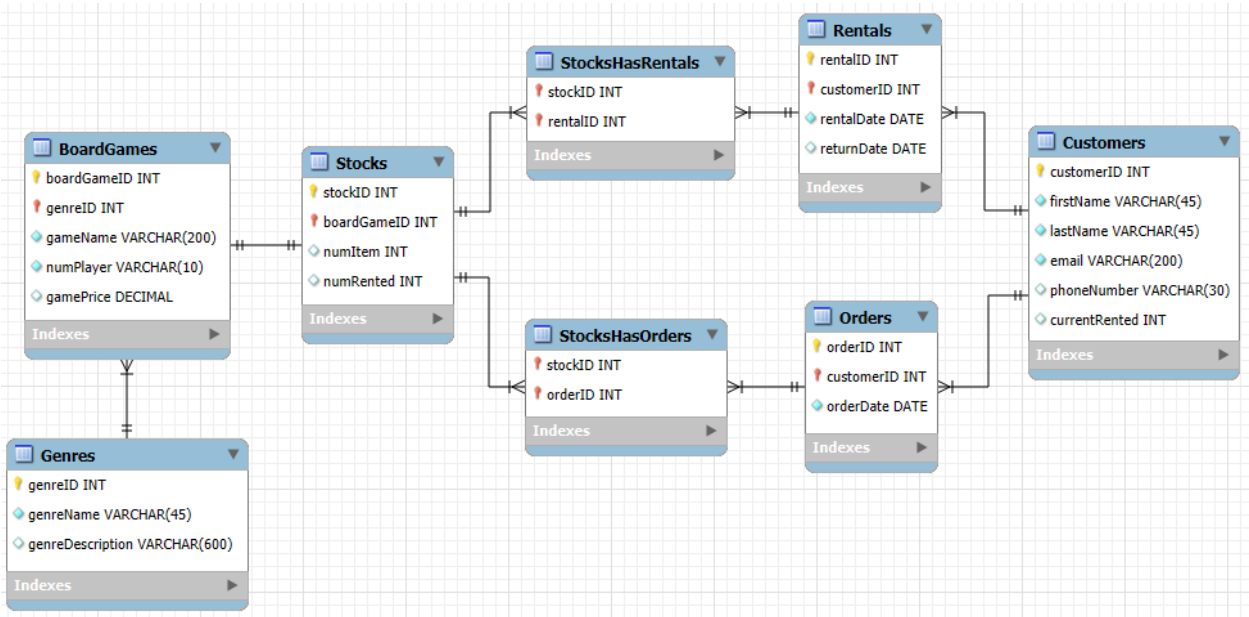
- **Genres:** Genres associated with the board games.
    - genreID: INT, auto_increment, PK
    - genreName: VARCHAR(45), not NULL, UNIQUE
    - genreDescription: VARCHAR(600)

  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

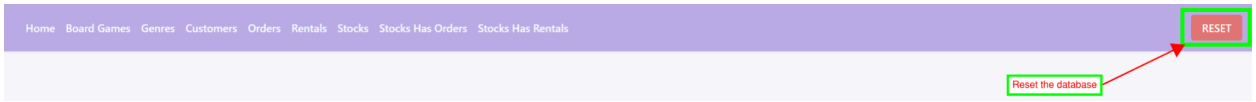
Customers

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

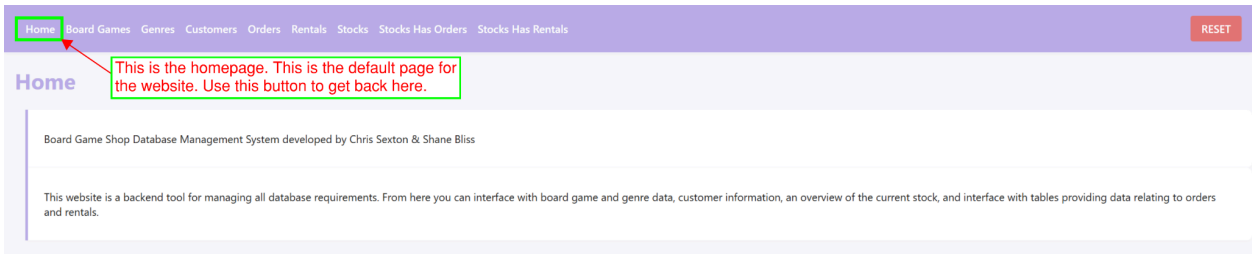
UI Screenshots

DELETE/CREATE RESET Button

(Inc. RESET DB)

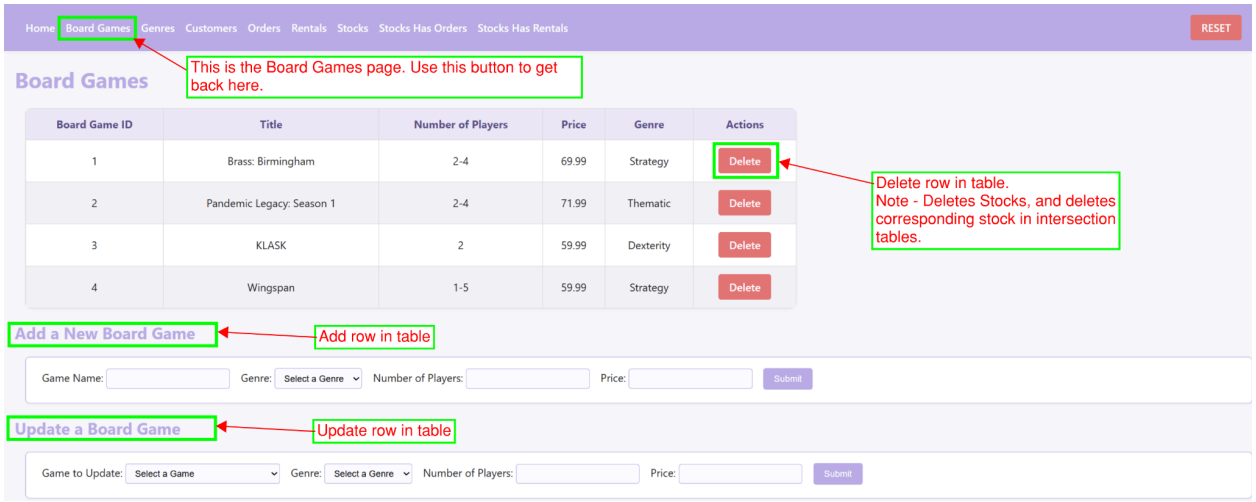


Home page

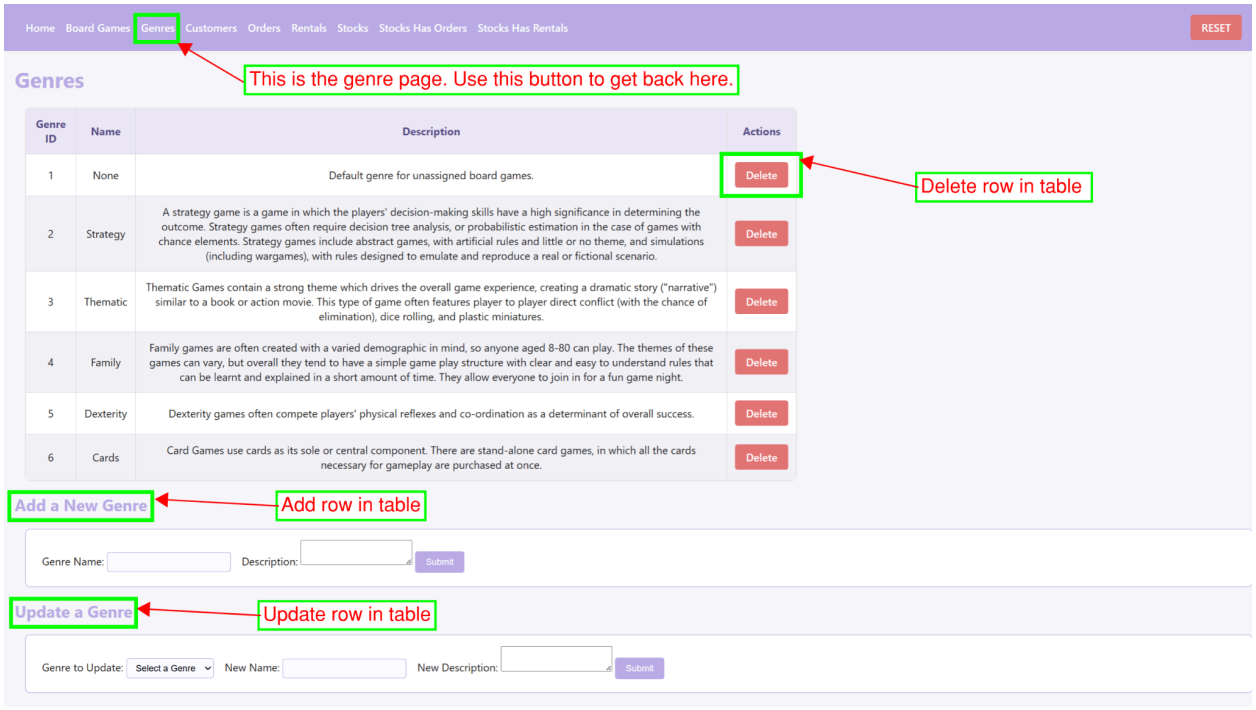


CREATE/READ/UPDATE/DELETE Board Games page

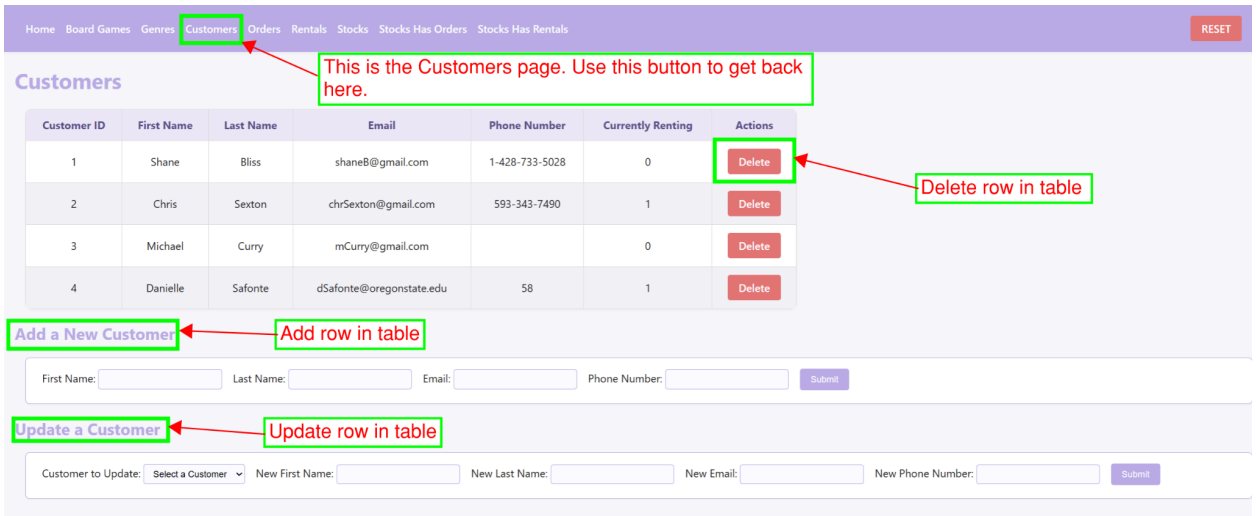
(inc. Deletes to the M:N, this cascades to the intersection tables)



CREATE/READ/UPDATE/DELETE Genres page



CREATE/READ/UPDATE/DELETE Customers page

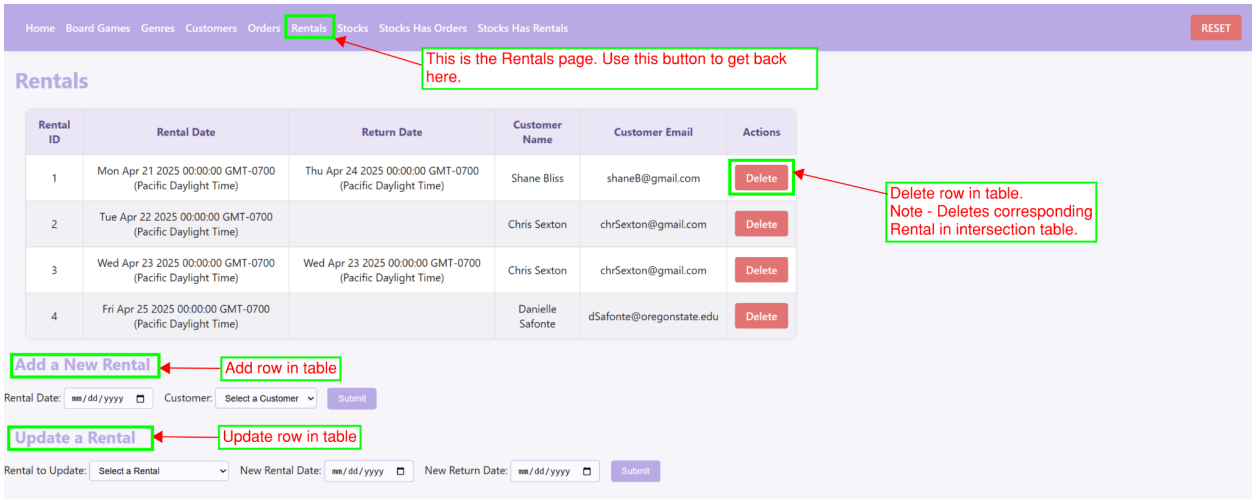


CREATE/READ/UPDATE/DELETE Orders page

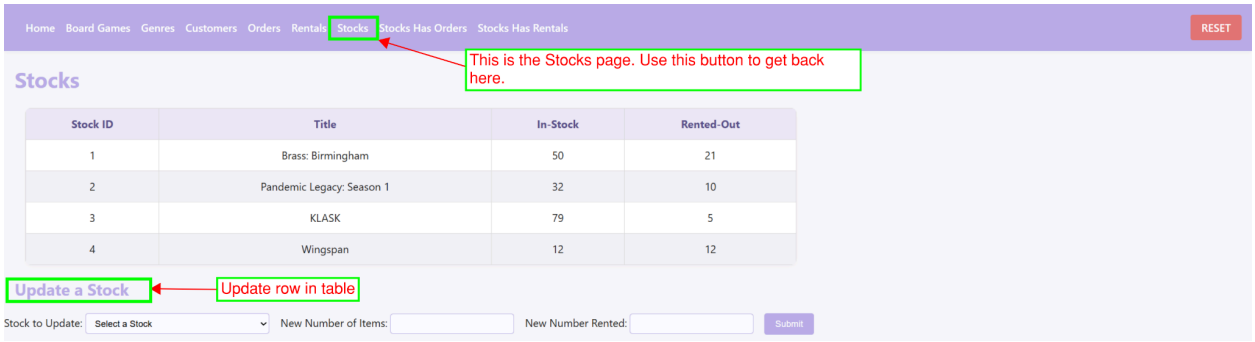
(Inc. Update of a M:N)

CREATE/READ/UPDATE/DELETE Rentals page

(Inc. Update of a M:N)



READ/UPDATE Stocks page



READ/UPDATE Stocks Has Orders page

READ/UPDATE Stocks Has Rentals page

Citations

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

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