

Project Step 4 Draft Version: Design HTML Interface

Group 7: Kimchi Taco

Junhyeok Jeong

Joelle Perez

Site URL: <http://web.engr.oregonstate.edu/~jeongju/beaver-books/>

DDL file name: beaver_books_DDL_queries.sql

DML file name: beaver_books_DML_queries.sql

a) Project Outline and Database Outline, ERD and Schema Updated Version

1. Overview

Our project is an online bookstore called “Beaver Books”. The “Beaver Books” usually have about 30,000 sales for a year by Oregon State University students and Corvallis people. Our web-app formulated book databases from Amazon Books. The website sequence is mainly from login screen to bookstore web-app with one-user interface. For example, a user logged in successfully, then the user can search and buy books from the book list. In the end, the user can interact with the shopping cart before purchasing. To populate the bookstore’s book DB and book search-bar, we will use ‘Python Scrapy’ to get the book information like ISBN, author, publisher, year, and price from Amazon Books.

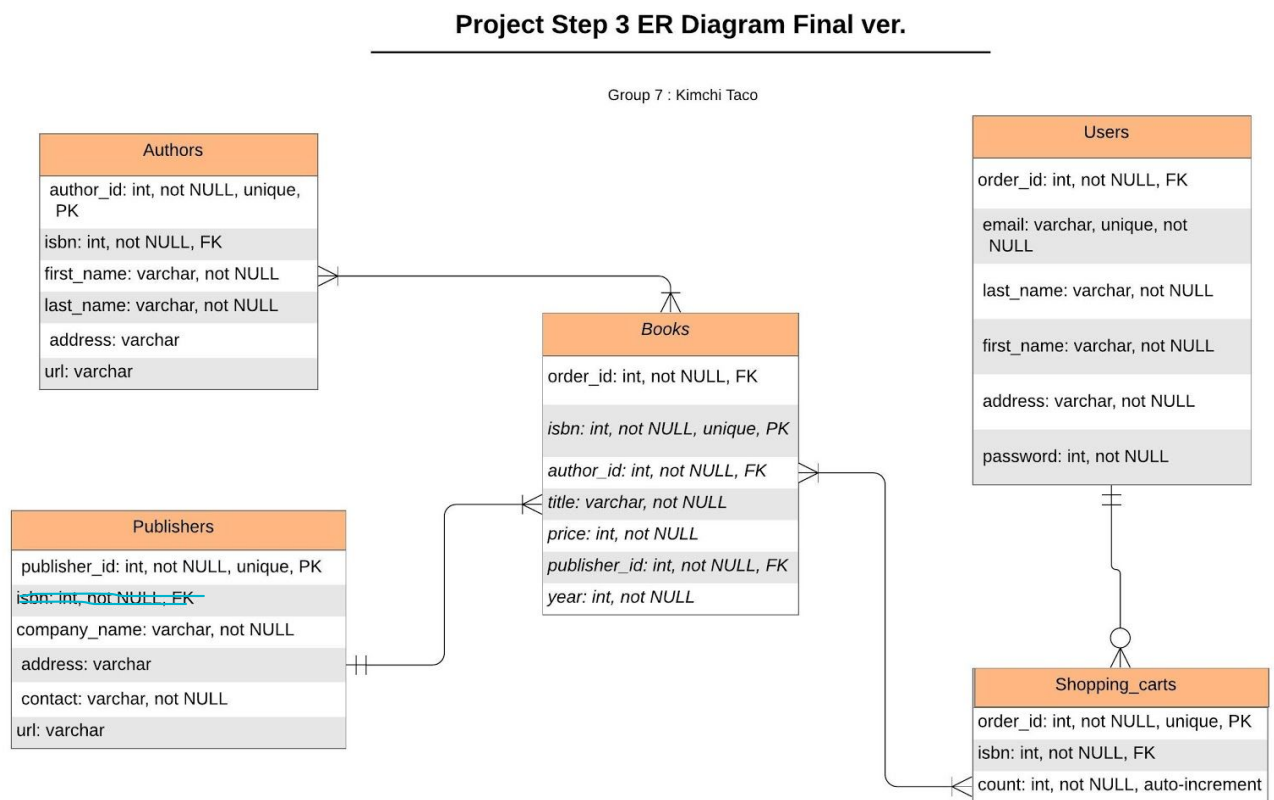
2. Project Outline and Database Outline:

- Users: saves the user information who uses our website.
 - order_id: int, not NULL, FK
 - email: varchar, unique; only one email per user, not NULL
 - last_name: varchar, not NULL
 - first_name: varchar, not NULL
 - address: varchar, not NULL
 - password: varbinary, not NULL

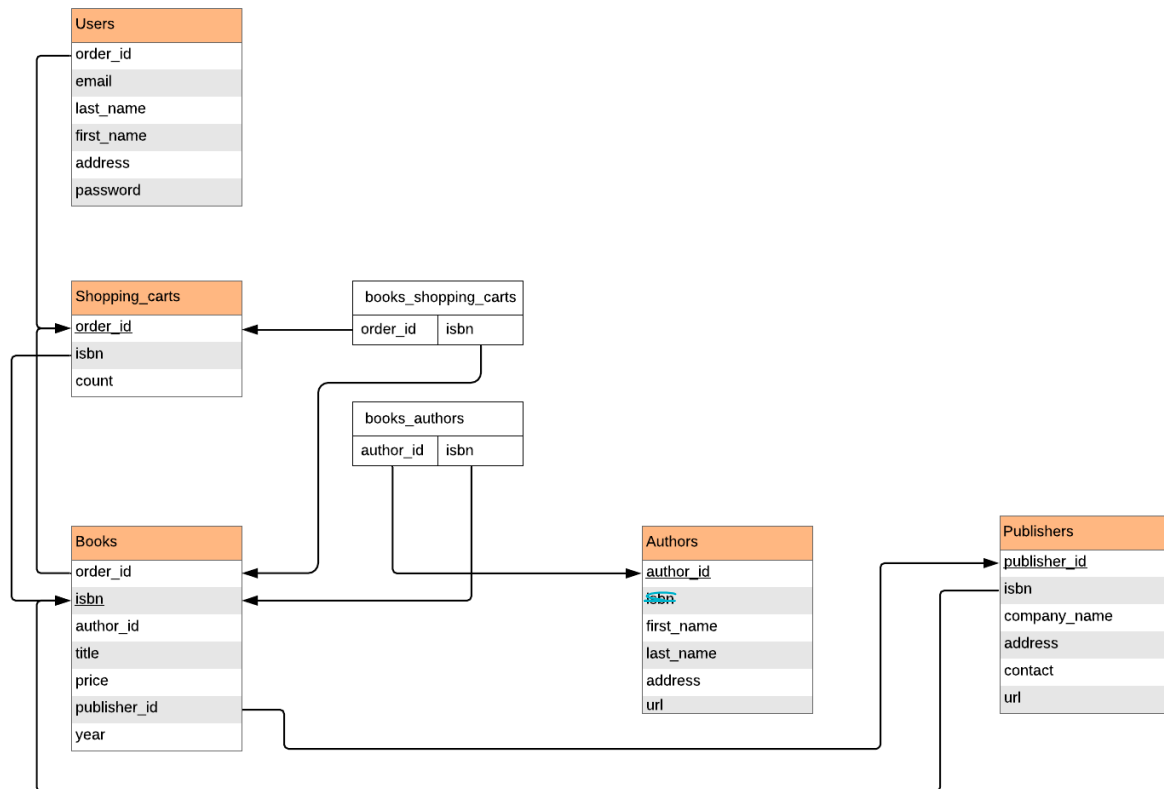
- relationship: a 1:M relationship between User and Shopping Cart is implemented with Email to find their orders.
- Shopping_carts: saves the user's chosen books before purchasing them.
 - order_id: int, not NULL, unique, PK
 - isbn: int, not NULL, FK
 - count: int, not NULL, auto-increment
 - Relationships:
 - a 1:M relationship with Books is implemented with order_id and ISBN to check books for certain users.
 - a M:M relationship with Users is implemented with order_id to check certain users.
- Books: saves the details of the book.
 - order_id: int, not NULL, FK
 - isbn: int, not NULL, unique, PK
 - author_id: int, not NULL, FK
 - title: varchar, not NULL
 - price: int, not NULL
 - publisher_id: int, not NULL, FK
 - year: int, not NULL
 - relationships:
 - a M:1 relationship with Publishers is implemented with publisher_id and isbn to find the publisher of the book.
 - a M:M relationship with Authors is implemented with author_id and isbn to find the authors of the book.
 - a M:M relationship with Shopping_carts is implemented with isbn and order_id to find the ordered user
- Authors: keeps the information about the author.
 - author_id: int, not NULL, unique, PK
 - isbn: int, not NULL, FK
 - first_name: varchar, not NULL
 - last_name: varchar, not NULL
 - address: varchar
 - url: varchar
 - relationship: a M:M relationship between Authors and Books is implemented with isbn and author_id to find books under that authors.

- Publishers: keeps the information about the publisher.
 - publisher_id: int, not NULL, unique, PK
 - company_name: varchar, not NULL
 - address: varchar
 - contact: varchar, not NULL
 - url: varchar
 - relationship: a 1:M relationship between Publishers and Books is implemented with publisher_id and isbn to find books under that publisher.

3. Updated Entity-Relationship Diagram:



4. Updated Schema:



Feedback by the peer reviewer

We have 4 reviews from other students on Piazza

James Belknap, section 400

1. Does the overview describe what problem is to be solved by a website with DB back end?

- Yes the overview did a good job describing what the problem is. It is going to be a website to purchase books, and uses a database to populate the books information with price. So it acts as a bookstore. I do like how the overview explained a walkthrough for the user visiting the site. Though after reading the shopping cart entity I have become confused. So, it's just an order history? If they decide to buy a book will they be directed to amazon, since you are using amazonBooks as a database? If the site is actually selling books, the shoppingCart entity description "saves the books ordered by the user" to me doesn't reflect that.

2. Does the overview list specific facts?

- Yes, the overview lists that they will be using python scrappy, which will help them get the information for their database from amazon books. This information is used directly with a user being able to see this information when looking up books, or looking at the book lists. A question that arose when I read it was who/where the book lists were coming from? Is the user supposed to populate the booklist for future purchases, are the book lists coming from you guys, or are the booklists coming from amazon best sellers in categories? Also is the site carrying out the purchase or is the purchase happening through amazon?

3. Are at least four entities described and does each one represent a single idea to be stored as a list?

- Yes they have 5 entities, and each entity corresponds to its own table/list. The entities are users, shopping carts, books, authors, and publishers.

4. Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

- Yes, the outline does describe the purpose of each entity, as well as their attributes. Though the shopping cart entity confuses me because the description makes it seem like an order history. Also, I think this would be a M:M relationship with books. Books can be a part of many different order histories/shopping carts, and a shopping cart can have many books. I think maybe this relationship could be explored better. If implemented this would also need a third schema table to showcase the M:M relationship. All the attributes have a data type, which I think is important and nice to see. I also feel like the outline only describes the relationships one way, but not the other way around. I think including it will help visualize things better.

5. Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

- Yes, they have 1 M:M relationship between authors and books. Though in the outline they only describe the M:M when it comes to authors writing several books. They don't explore the M:M relationship between books and authors. They also don't show a 3rd table in the schema linking the M:M relationship, which is needed for a M:M relationship to be represented properly. I stated earlier, but I think a M:M relationship should be explored between books and shopping_cart.

6. Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

- Yes they do having consistency with their naming. All their entities are properly capitalized and plural. Although their shopping cart entity does not follow proper naming requirements. Their shouldn't be a space. It should either be Shopping_carts or shoppingCarts. Their attributes are also all consistent. They are singular and do not have spacing unlike the entity names. They however also do not follow proper naming requirements in that they are not lowercase. This is a problem we ran into during our first grading. This can also be seen in the attributes in the schema.

Name: Darwin Yee, Section: 401

1. Does the overview describe what problem is to be solved by a website with DB back end? Does the overview list specific facts?

- The overview has enough details about the problem and an overview of the functions of the Database for the bookstore. But as our grader suggested, a specific fact about how many people the website will be serving may be needed.

2. Are at least four entities described and does each one represent a single idea to be stored as a list?

- It has 5 entities and each represents a single idea.

3. Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

- The function of the Shopping Carts is a little confusing to me. It doesn't work like an ordinary shopping cart from conventional websites. It may be better to rename it to OrderHistory.
- For the Books table, isn't ISBN unique for each book? I think you can use ISBN as the PK for Books instead of Order_ID.

4. Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

- There is one M:M relationship between Authors and Books. The 1:M relationships are correctly formulated.
- According to the class lecture, the Schema should have arrows pointing from a FK to the PK of another table.
- An extra table, "Authors_Books", may be needed for the ER and the Schema to correctly represent the M:M relationship.

5. Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

- It would be better to change the "Shopping carts" to "Shopping_carts" for consistency.

Name: David Lee, Section: 401

1. Does the overview describe what problem is to be solved by a website with DB back end?

- The overview was able to describe the requirements for the project properly by displaying a need for a basic search/purchase system for books within an online bookstore. Being a simple point of sales system, requiring a login based on customer data, it is able to serve as a database that tracks the outbound of sales.

2. Does the overview list specific facts?

- As stated in the outline, we can clearly define a single customer that can have multiple shopping carts that contain multiple book orders, as well as associating books to their respective author and publisher information. One problem I had difficulty comprehending is the scale of the project in terms of user count as well as an issue with inventory management. Does the store have a warehouse with an unlimited quantity of products (books)? In my personal opinion, a proper method of tracking inventory inbound could improve this system.

3. Are at least four entities described and does each one represents a single idea to be stored as a list?

- The outline clearly defines each entity's function via five different tables. (Users, Shopping carts, Books, Authors, Publishers)

4. Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

- The entities and attributes are defined clearly with their purposes and relationships. The primary keys and foreign keys relationship could have been defined in the outline (before the diagrams). It is stated that multiple attributes follow a VARCHAR data type. As an addendum, with VARCHAR not having defined its string length, it may cause an error at both table creation time as well as associating data during joins (due to data being stored in different data sizes). The size of the data type should be clearly defined. The shopping cart entity seems to be static and not dynamic as seen in normal e-commerce systems.

5. Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

- There exists an M:M relationship between Books and Authors. With an M:M relationship, it would be proper to implement a joint table, Books_Authors.
- There exists a 1:M relationship between Users & Shopping carts, Shopping carts & Books and Publishers & Books. One confusing aspect I see is between Shopping carts and Books. It seems that Books are uniquely stored in shopping carts when it is possible by logic that the same type of book can be associated with multiple shopping carts and users. A key to defining the same book in different carts (when the store has more than one of the same books) could be a convention used to improve the system.

6. Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

- The entity names follow a capitalization rule that is uniform in the outline, with it being plural as required by our lectures; however, in the diagram, there are attributes that start with a lower case (not uniform in the diagram). I would have preferred all attribute names to be lowercased and entity names are capitalized or vice-versa (for distinction - personal preference). The name of the Shopping carts entity should be changed so that space is not used within the name.

name : Martin Delgado, Section: ???

1. Does the overview describe what problem is to be solved by a website with DB back end?

- Yes the overview did a good job describing what the problem is. It is going to be a website to purchase books, and uses a database to populate the books information with price. So it acts as a bookstore. I do like how the overview explained a walkthrough for the user visiting the site. Though after reading the shopping cart entity I have become confused. So, it's just an order history? If they decide to buy a book will they be directed to amazon, since you are using amazonBooks as a database? If the site is actually selling books, the shoppingCart entity description "saves the books ordered by the user" to me doesn't reflect that.

2. Does the overview list specific facts?

- Yes, the overview lists that they will be using python scrapy, which will help them get the information for their database from amazon books. This information is used directly with a user being able to see this information when looking up books, or looking at the book lists. A question that arose when I read it was who/where the book lists were coming from? Is the user supposed to populate the booklist for future purchases, are the book lists coming from you guys, or are the booklists coming from amazon best sellers in categories? Also is the site carrying out the purchase or is the purchase happening through amazon?

3. Are at least four entities described and does each one represent a single idea to be stored as a list?

- Yes they have 5 entities, and each entity corresponds to its own table/list. The entities are users, shopping carts, books, authors, and publishers.

4. Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

- Yes ,the outline does describe the purpose of each entity, as well as their attributes. Though the shopping car entity confuses me because the description makes it seem like an order history. Also, I think this would be a M:M relationship with books. Books can be a part of many different order histories/shopping carts, and a shopping cart can have many books. I think maybe this relationship could be explored better. If implemented this would also need a third schema table to showcase the M:M relationship. All the attributes have a data type, which I think is important and nice to see. I also feel like the outline only describes the relationships

one way, but not the other way around. I think including it will help visualize things better.

5. Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

- Yes, they have 1 M:M relationship between authors and books. Though in the outline they only describe the M:M when it comes to authors writing several books. They don't explore the M:M relationship between books and authors. They also don't show a 3rd table in the schema linking the M:M relationship, which is needed for a M:M relationship to be represented properly. I stated earlier, but I think a M:M relationship should be explored between books and shopping_cart.

6. Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

- Yes they do having consistency with their naming. All their entities are properly capitalized and plural. Although their shopping cart entity does not follow proper naming requirements. Their shouldn't be a space. It should either be Shopping_carts or shoppingCarts. Their attributes are also all consistent. They are singular and do not have spacing unlike the entity names. They however also do not follow proper naming requirements in that they are not lowercase. This is a problem we ran into during our first grading. This can also be seen in the attributes in the schema.

Feedback by the Step 2 Final Version Grader

- (1) Your schema is missing arrows.
- (2) The foreign keys of your M:M relationship (book_author) should be connected to the primary keys.
- (3) Instead of storing the total_price of all the books of the shopping cart, total_price could be calculated by returning the sum of the price of all the books within the shopping cart.

b) Fixes based on Feedback from Previous Steps:

Fixes based on peer reviewers

1. In the overview, the reviewers were confused by 'One-user interaction'. Our team's intention of the term was that our online bookstore will provide single-user interface like checking logged-in user information and shopping cart like normal online shopping malls (Amazon, ebay). We changed the term and put specific examples to make future reviewers understand.
2. In the 'Project Outline and Database Outline', a reviewer can't understand the description of the "Shopping_Carts" entity because it says "saves the books ordered by the user". We noticed that is a problem. Therefore, we needed to change it.
3. In the ER diagram and schema, a reviewer mentioned that we didn't use the proper naming requirements for the attributes. Thus, we changed them to lowercase names.
4. We're disregarding changing the shopping cart and books relationship to an M:M because we only want one shopping cart per user rather than many shopping carts.
5. Reviewers suggested adding more specific information on websites like main customer and sales.
6. We set Primary Keys and Foreign Keys based on the suggestions from reviewers because it could make a well-structured database and prevent collision.

Fixes based on grader

1. Remove 'total_price' attribute in Shopping_carts
2. Added arrows to schema.
3. M:M relationship book_author FKs are now connected to PKs.

Feedback by the Peer Reviewer (Project 3 Review)

We have 3 feedbacks.

Name: Yihao Wu, Section: N/A

Does the UI utilize a **SELECT** for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

No. I only saw a registration email page and a login email page. But don't see on a list of Books in the schema diagrams, the Authors, Publishers and Shopping_carts things on the UI. I guess you should implement the login interface first, and then the book. However, the assignment requires that we implement in the UI all entities contained in the schema diagram. I suggest you start implementing them ahead of time.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

No. I don't see any on the SELECT utilize a search/filter with a dynamically populated the list of properties. I suggest you implement this requirement in the Books entity in the schema diagram.

Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.

Yes. I see that you implemented INSERT in the USERS entity. You let the user INSERT their personal registration information and store the user's personal information in the database. It's a good choice.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

No. I don't see any associated FK attributes in your Users entity, and any M: M relationships. Maybe you do, but I haven't seen it in your UI yet.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

No. I see in the schema diagram that you're going to have DELETE on your UI. But I haven't seen anything about DELETE in the UI so far. I guess you're going to make a DELETE between your Authors and Books. I think that would be a good approach.

Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

Yes. I'm sure you're going to use UPDATE in user registration because you're going to UPDATE the user's information to allow the user to log in to the account they just signed up for. And, I'm also sure that you're going to use UPDATE when you add the entities of Authors and Books.

Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.

No. I don't see anything about NULLable in your UI right now. But I think you can use NULLable between Authors or Publishers. Because I think when you add books, sometimes you don't have Authors or Publishers. I hope my ideas will help you.

Do you have any other suggestions for the team to help with their HTML UI?

My advice is that you can refer to the advice I have given you, but they are only for reference, and only my personal thoughts, does not mean that my thoughts are correct. Before the next assignment is submitted, I hope you can implement INSERT, DELETED, UPDATE and SEARCH. And it can bring more explicit UI experience to users. I look forward to your final work.

Name: Louis Adams, Section: 401

1. Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and display them.

I do think the UI is designed to utilize a SELECT for all of the tables. I'm not sure the above reviewer noticed that you have to login using the given login credentials to see the rest of the site. Almost all of the data is displayed on the one page, Available Books. I do think this could be considered to be "a single query to join all tables and display them" because there is really only the one page that has data for Books, Authors, and Publishers. There is a search bar which is not functional yet but this could be used to SELECT from different tables. The user does have the ability to SELECT the user on the homepage.

2. Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

There is a search bar so it does seem like this is being planned for selecting user specified Books/Authors/Publishers. There could be different user accounts which would presumably SELECT whichever Shopping_carts that user may have associated with their account.

3. Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.

You can enter an email address and create a new account so I believe there will be an INSERT for Users. Adding books to your cart will basically be an INSERT for Shopping_carts. I'm not sure how an INSERT would be implemented for Books, Authors, and Publishers. You may want to add some kind of an admin account that has the ability to add Books, Authors, and Publishers. Although even then it seems like Books would be added and Authors and Publishers would be added through the foreign key.

4. Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

No, I don't think there are any INSERTs that will also add the corresponding FK attributes. The M:M relationship is between Books and Authors and I don't see any way to INSERT into the Books or Authors table. It seems like the Books table would just be pre-populated with data. There are some issues with FKs that need to be dealt with as well.

5. Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

There is the ability to DELETE a book from the shopping cart. It seems like the way this is set up is that the Shopping_carts entity is really an order of a one book and a quantity. So a shopping cart cannot hold more than one kind of book. This could work but Shopping_carts should probably be named something like cart_books to be less confusing. I would consider making the Books - Shopping_carts relationship your M:M relationship with an intermediary entity of cart_books which would keep track of the quantity of each book in the shopping cart.

6. Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

Yes, I believe Shopping_carts could be updated by adding books to the user's cart. With the current UI, I don't think the other tables could be updated.

7. Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.

I don't see any relationship that is NULLable currently. There is a potential NULLable relationship between Books and Publishers I suppose. Authors who self-publish would have books with no publisher.

8. Do you have any other suggestions for the team to help with their HTML UI?

Overall, I think this HTML UI has a nice design and layout but there are some real issues with keys and some other fundamental design problems. As I mentioned above it would seem to me that your M:M relationship should really be between Books and Shopping_Carts with an intermediary such as cart_books to keep track of the quantity of each book in the cart. I'm not sure a M:M relationship would really be necessary between Books and Authors. Yes, books can have more than one author but for the purposes of this assignment you could probably say that each book has one author. Also, after looking at your ER diagram and schema I believe there are issues with the primary and foreign keys. There is no primary key listed for Users. email would be the obvious choice since it is a unique identifier. Since Users - Shopping_carts is a 1:M relationship you should have the foreign key in Shopping_carts, not the other way around. So you could have email as a FK in Shopping_carts, which links each shopping cart with a user. order_id should not be the attribute of a user since a user may have many orders. Also, order_id should not be an attribute of Books because books may be ordered many times. Also, isbn should not be an attribute of Publishers or Authors. An author cannot have an isbn nor can a publisher. You can keep both isbn and author_id as the FKs in the intermediary entity book_author if you want to keep that M:M relationship. For Publishers, you have publisher_id as the FK in Books. You don't need a FK in Publishers. I think it's really essential to get these issues with keys figured out soon before we start populating our databases with data and running queries.

Name: Esther Park, Section: 401

Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

Currently, there only seems to be one universal search bar. A universal search bar seems complex for an introductory SQL course, so if this was the team's intention then somewhat yes but for the purposes of this project no there is not a SELECT specifically for every table in the schema.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

Again, there is only one universal search bar on every web page. It is uncertain what the search bar is for, especially when it is on the "about" page or "carts" page. Perhaps the team can add a page for authors and a page for publishers and implement a filter bar there to either search for a certain author or publisher.

Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.

There is an option to probably insert a book into the shopping cart and perhaps insert a new account for when a user registers. A possible idea that I've seen on other websites is an admin webpage, where the team can manually insert/add/delete books, publishers and authors to the tables.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products,

INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

Depending on the approach, an insert feature for admin could fulfill this requirement of the M:M relationship. Although rare, a book can have more than one author, and those authors can author many books.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

Currently I see a delete function in the shopping car page, where a user can delete a book from their cart. I can see this deleting from the shopping_carts table, but not removing from the publishes, authors and books tables.

Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

Shopping_carts can be updated for when a user adds a book to their carts. If the team adds an admin page, there would be more opportunities for this update feature.

Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.

There are currently no NULLable relationships that I see. To fulfill this requirement, during account registration there could be optional values such as favorite genre, author, book, etc that could be NULLable.

Do you have any other suggestions for the team to help with their HTML UI?

UI is nice and clean.

Actions based on the feedback (Project 3)

- Yihao Wu
 - Yihao recommended adding a specific UI of our website, but we think she misunderstood or missed our instructions about how to login with a temporary account.
 - Although she couldn't get into any other htmls than register.html and login.html, she gave advice about the population INSERT, DELETE, UPDATE, and SEARCH for sql table. To be honest, we didn't feel satisfied

about connecting our schema and ER diagram to the website. Therefore, we planned to add more specifications on the website.

- Louis Adams
 - Louis recommended adding an admin account to interact with sql DB and to manipulate the shop page.
 - He gave some advice on M:M relationship because he thinks applying M:M relationship between 'Books' and 'Shopping_carts' would be more reasonable.
- Esther Parks
 - An admin page to where the team can insert/add/delete books.
 - Instead of a global search bar, we can have specific search bars for each page such as the shop page to look for certain books.
 - Search for books can have optional titles and authors.

Upgrades to the Draft version (Project 3)

- Added admin.html and created tables for interacting with sql DB. (website)
- Created an admin account for interacting with admin.html and sql DB. (website)
- Added M:M relationship between 'Books' and 'Shopping_carts' (ER diagram and Schema)
- Added search bars on the shop page for to search for books specified on authors or titles, either one are optional.