URL: http://classwork.engr.oregonstate.edu:60728/

Project Step 4

Sean Miller and Sean David Brady

Sean & Sean Books: A Bookstore Run By Seans

Project Overview

With the abundance of books in publication, careful organization is required in order to maintain stock and track down difficult-to-discover reading material. *Sean & Sean Books* is a growing bookstore chain, which sold 8,000 books last year and is seeking to sell an additional 2,000 this year, with issues managing the essential data needed so that their business can flourish as their customer base expands. They currently have 220 customers in their database and seek to double that figure this year while maintaining an ease of accessing various information about those customers. Some copies of books sold out because inventory wasn't being correctly tracked in a timely manner, and *Sean and Sean Books* wants to avoid that in the future. We seek to simplify the processes and synthesize the data into easily accessible and useful information so that *Sean & Sean Books* can focus on their passion for spreading good literature to those around the globe.

Database Outline

Here are the entities that will be put in place to make a functioning database. All RELATIONSHIPS are not attributes.

Customers: Tracks the details of customers with whom we do business

- *customerID*: int, auto_increment, unique, not null, PK
- *firstName*: varchar(100), not null
- *lastName*: varchar(100), not null
- *email*: varchar(255),
- *phoneNumber:* char(10)
- RELATIONSHIPS: 1:M → **Orders**

Books: Tracks the details of various books

- *bookID:* int, auto_increment, unique, not null, PK
- *publicationDate*: date, not null
- *isbn-10*: int
- *isbn-13:* int
- *inStock*: tinyint, not null

- *price*: decimal, not null
- *inventoryQty*: int,
- *title*: varchar, not null
- *publisherID*: int, not null, FK from **Publishers**
- RELATIONSHIPS: M:M → **Orders**, M:M → **SLOCs**, **M:M → Authors, M:M → Genres**

Orders: Tracks the details of customer orders
- *orderID*: int, auto_increment, unique, not null, PK
- *orderDate*: date, not null
- *orderTime* : time, not null
- *total:* decimal(10,2), not null
- *taxRate*: decimal(6,4)
- *customerID*: int, not null FK from **Customers**
- *salesRateID*: int, not null FK from **SalesRateLocations**
- RELATIONSHIPS: M:M → **Books**, M:1 → **SalesRateLocation**, M:1 → **Customers**

SalesRateLocations: Tracks the tax rates in states and counties
- *salesRateID*: int, auto_increment, not null, unique, PK
- *taxRate*: decimal(6, 4), not null
- *county:* varchar(45), not null
- *state*: varchar(45), not null
- RELATIONSHIPS: 1:M → **Orders**

Intersection Tables

OrderItems: Intersection table for the M:M relationship between **Orders** and **Books**
- *orderItemID*: int, auto_increment, unique, not null, PK
- *orderID*: int, not null, FK from **Orders**
- *bookID*: int, not null, FK from **Books**
- *quantity:* int, not null
- *individualPrice:* decimal(10,2), not null → Price for one book
- *subtotal*: decimal(10,2), not null → *individualPrice* x *quantity*
- RELATIONSHIPS: M:1 → **Books**, M:1 → **Orders**

BookLocations: Intersection table for the M:M relation for **Books**
- *bookID*: int, not null, FK,PK from **Books**
- *slocID*: int, not null, FK,PK from **SLOCs**
- RELATIONSHIPS: M:1 → **Books**, M:1 → **SLOCS**

BookGenres: Intersection table for the M:M relationship between **Books** and **Genres**

- *genreID*: int, not null, FK,PK from **Genres**
- *bookID*, int, not null, FK, PKfrom **Books**
- RELATIONSHIPS: M:1 → **Books** M:1 → **Genres**

BookAuthors: Intersection table for the M:M relationship between **Books** and **Authors**

- *authorID*: int, not null, FK,PK from Authors
- *bookID*, int, not null, FK,PK from Books
- RELATIONSHIPS: M:1 → **Books** M:1 → **Authors**

Category Tables

Genres: Tracks the different genre types a book can have as a category table

- *genreID*: int, auto_increment, not null, unique, PK
- *genreName*: varchar, not null
- RELATIONSHIPS: M:M → **Books**

Authors: Tracks the different authors types a book can have as a category table

- *authorID*: int, auto_increment, not null, unique, PK
- *firstName*: varchar(80), not null
- *middleName*: varchar(80),
- *lastName*: varchar(80),
- *fullName*: varchar(243), not null,
- RELATIONSHIPS: M:M → **Books**

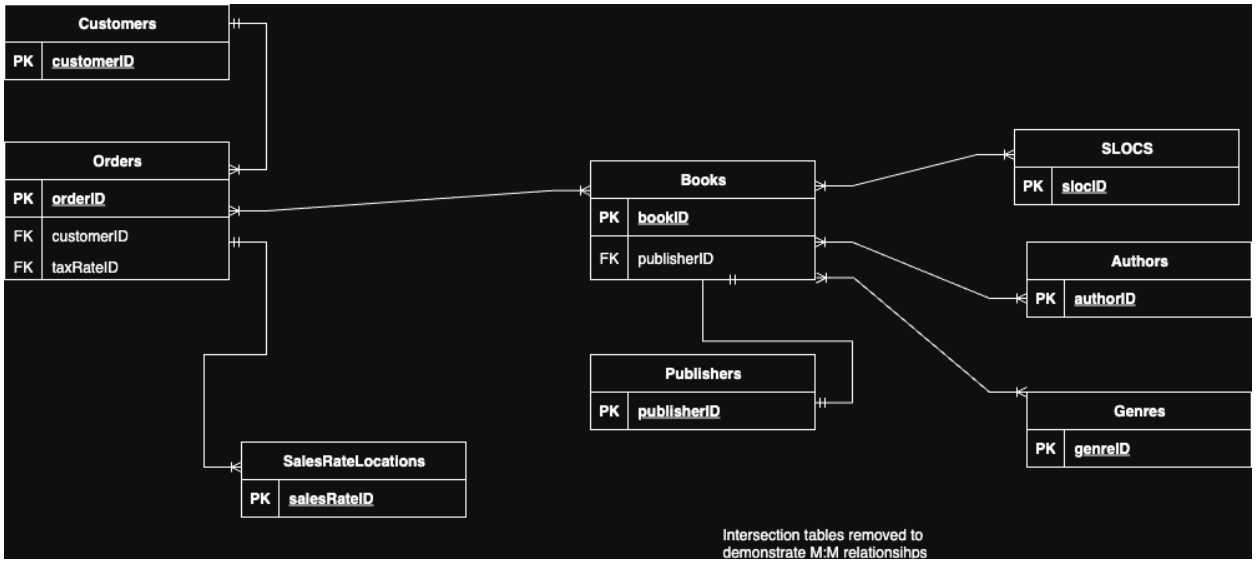
Publishers: Tracks the different publishers a book can have as a category table

- *publisherID*: int, auto_increment, not null, unique, PK
- *publisherName*: varchar, not null
- RELATIONSHIPS: 1:M → **Books**

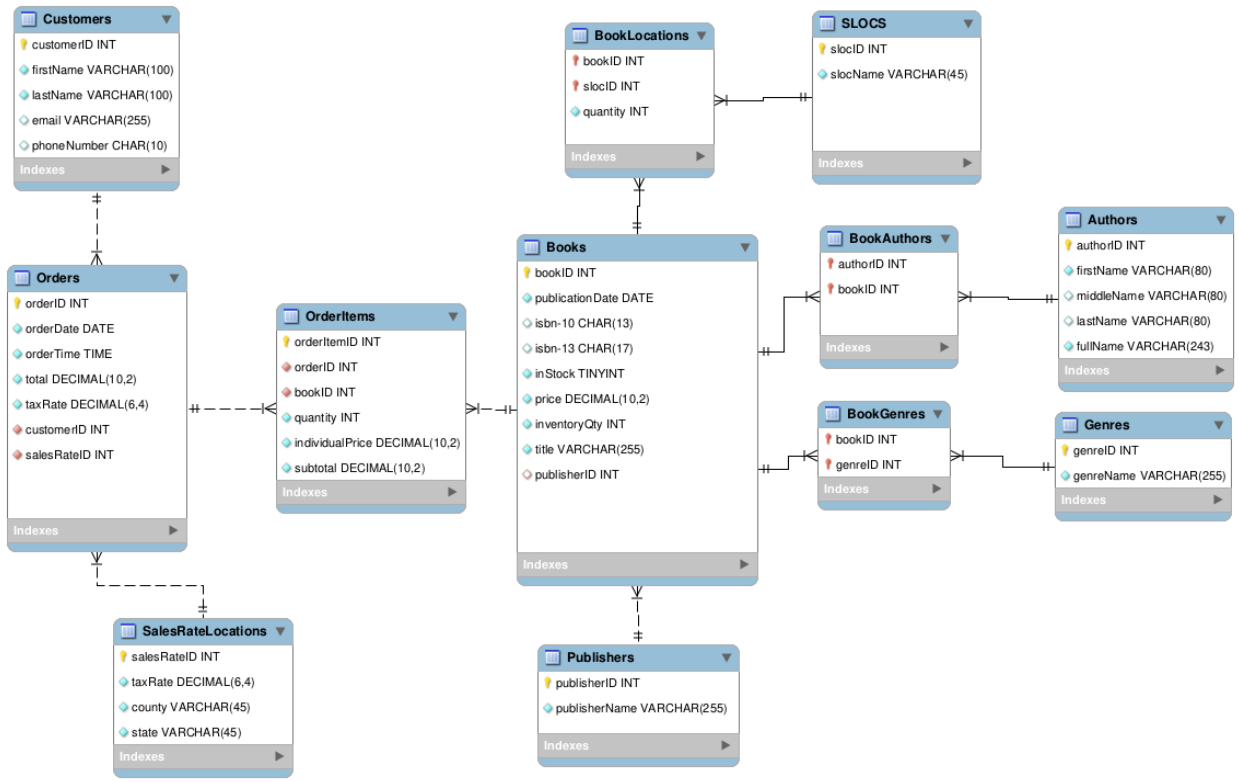
SLOCs: Tracks the storage locations as a category table

- *slocID*: int, auto_increment, not null, PK
- *slocName*: varchar(45), not null
- RELATIONSHIPS: M:M → **Books**

Entity Relationship Diagram



Schema Diagram



Example Data

Customers

- firstName = ‘Reggie’, lastName = ‘Reggerson’, email = ‘regreg@reg.com’, phoneNumber = 3333888902
- firstName = ‘Gail’, lastName = ‘Nightingstocks’, email = ‘gailsmail@gmail.com’, phoneNumber = 2295730384
- firstName = ‘Filipe’, lastName = ‘Redsky’, email = ‘filipe@hotmail.com’, phoneNumber = 5649836590

Orders

- orderDate = Oct. 01, 2025, orderTime = 21:12:11, total = $45.61, taxRate = *select statement to find taxRate for Polk, IA*, customerID = *select statement to find customerID for Reg*, salesRateID = *select statement to find salesRateID for Polk, IA*
- orderDate = Nov. 05, 2025, orderTime = 23:01:42, total = $61.21, taxRate = *select statement to find taxRate for Jerome, ID*, customerID = *select statement to find customerID for Gail*, salesRateID = *select statement to find salesRateID for Jerome, ID*

- orderDate = Dec. 09, 2025, orderTime = 08:24:24, total = $112.09, taxRate = *select statement to find taxRate for San Francisco, CA*, customerID = *select statement to find customerID for Filipe*, salesRateID = *select statement to find salesRateID for San Francisco, CA*

SalesRateLocations

- taxRate = 4.2, county = 'Polk', state = 'Iowa'
- taxRate = 5.1, county = 'Jerome', state = 'Idaho',
- taxRate = 8.625, county = 'San Francisco', state = 'California'

OrderItems

- orderID = *select statement for proper order*, bookID = *select statement for Beloved*, quantity = 2, individualPrice = *select statement for Beloved price*, subtotal = *calculation from qty x individualPrice*
- orderID = *select statement for proper order*, bookID = *select statement for Inherent Vice*, quantity = 1, individualPrice = *select statement for Inherent Vice price*, subtotal = *calculation from qty x individualPrice*
- orderID = *select statement for proper order*, bookID = *select statement for Good Omens*, quantity = 3, individualPrice = *select statement Good Omens price*, subtotal = *calculation from qty x individualPrice*

Books

- publicationDate = Aug. 4, 2008, isbn-10 = 143126850, isbn-13 = 978143126850, inStock = 5, price = $15.99, inventoryQty = 5, title = 'Inherent Vice', publisherID = *select statement to find publisherID for Penguin Books*
- publicationDate = Sept. 1, 1978, isbn-10 = 1400033416, isbn-13 = 9781400033416, inStock = 7, price = $17.99, inventoryQty = 7, title = 'Beloved', publisherID = *select statement to find publisherID for Vintage International*
- publicationDate = Nov. 8, 1984, isbn-10 = 0670691992, isbn-13 = 9780670691999, inStock = 1, price = $18.99, inventoryQty = 6, title = 'The Talisman', publisherID = *select statement to find publisherID for Viking Press*
- publicationDate = Nov. 28, 2006, isbn-10 = 0060853980, isbn-13 = 9780060853983, inStock = 1, price = $16.99, inventoryQty = 8, title = 'Good Omens', publisherID = *select statement to find publisherID for William Morrow*

Publishers

- publisherName = 'Penguin Books'
- publisherName = 'Vintage International'
- publisherName = 'Viking Press'
- publisherName = 'William Morrow'

BookLocations

- bookID = *select statement to find bookID for Beloved*, slocID = *select statement to find slocID for Orchard,* quantity = 8

- bookID = *select statement to find bookID for Inherent Vice*, slocID = *select statement to find slocID for SunWillow,* quantity = 12
- bookID = *select statement to find bookID for Good Omens*, slocID = *select statement to find slocID for Whiskey Pines,* quantity = 3

SLOCs

- slocName = 'Orchard'
- slocName = 'Sunwillow'
- slocName = 'Whiskey Pines'

BookAuthors

- authorID = *select statement to find authorID for Morrison*, bookID = *select statement to find bookID for Beloved*
- authorID = *select statement to find authorID for Pynchon*, bookID = *select statement to find bookID for Inherent Vice*
- authorID = *select statement to find authorID for Gaiman*, bookID = *select statement to find bookID for Good Omens*

BookGenres

- genreID = *select statement to find genreID for Historical Fiction*, bookID = *select statement to find bookID for Beloved*
- genreID = *select statement to find genreID for Postmodern Fiction*, bookID = *select statement to find bookID for Inherent Vice*
- genreID = *select statement to find genreID for Science Fiction*, bookID = *select statement to find bookID for Good Omens*

Authors

- firstName = 'Toni', lastName = 'Morrison'
- firstName = 'Thomas', lastName = 'Pynchon'
- firstName = 'Stephen', middleName = 'Edwin', lastName = 'King'
- firstName = 'Peter', lastName = 'Straub'
- firstName = 'Neil', middleName = 'Richard', lastName = 'Gaiman'
- firstName = 'Terry', lastName = 'Pratchett'

Genres

- genreName = 'Historical Fiction'
- genreName = 'Postmodern Fiction'
- genreName = 'Horror Fiction',
- genreName = 'Science Fiction',
- genreName = 'Fantasy Fiction'

Citations

1. https://app.diagrams.net/. Used 7/26/25. Draw.io. Application used for creating the Entity-Relationship Diagram.

2. MySQL Workbench forward engineering feature utilized to create initial DDL.sql document. Used 7/26/25.

FEEDBACK SECTION

Feedback from Step One

Rafael Carlos Ayala

“I think this project has a solid idea behind it. The problem is clear and they’re building a system to help an online bookstore, Sean & Sean Books, organize their growing amount of customer, book, and order information.
They’ve included plenty of specific details that show they’ve thought through how the database will work. There are clear entities like Customers, Books, Orders, TaxRates, and Publishers, along with supporting tables like States, Counties, and SLOCs for storage locations. I like how they’ve included things like Genres and Authors through intersection tables to handle the many-to-many relationships with Books.
They meet the requirement of at least four entities, and each one represents a simple, clear list: customers, books, orders, etc. The relationships between tables are laid out well, and the intersection tables handle the many-to-many connections properly. The way they’ve set up primary keys, foreign keys, and constraints is clear and follows good structure.
One small suggestion is to stay consistent with naming conventions. Right now, there’s a mix of snake_case and PascalCase, especially with the intersection tables. It would be good to pick one style and stick with it across the whole design. I’d also consider renaming SLOCs. I understand it stands for storage locations, but it’s not obvious at first glance so a clearer name would help with readability.
For the BookAuthors intersection table, it’s great that they set up the many-to-many relationship between Books and Authors. One thing to think about is whether the extra idBookAuthor as a primary key is necessary. In cases like this, it’s often cleaner to use a composite primary key of (bookID, authorID) instead. It helps prevent duplicate entries and reflects the relationship more directly. I don't think it's wrong how they did it, just something to consider. Also, I noticed a small typo in the BookGenres table description. The primary key field is listed as idBookAuthor instead of something like idBookGenre, so they’ll probably want to fix that for clarity.
Overall, the database is structured well and lines up with what the project is trying to solve. The relationships make sense, and the idea feels realistic for a business like this. My feedback is original and based on my own understanding of the project.”

Francis Truong

“First off, I wanted to say I enjoyed the naming going on with the bookstore. The overview did a great job of describing the issue as a growing online bookstore needing to manage data as its customer base grows.

One note though, there isn't any mention of the scope and scale to be expected, with an estimate of how many books, number of daily processes, and how many people or customers are expected. That could be listed in the future to better understand the scale of the database needed for this bookstore.
I see that there are a lot of entities with customers, books, orders, tax rates, genres, authors, publishers, countries, states, and SLOCs. Each does represent a single idea. I respect how detailed is too scale the project is at this scale. The relationships look good with more than one M:M relationship.
One small minor detail I noticed for the datatypes is that some of the varchars have a size to them, and some do not, which doesn't quite line up with the consistency of it. If it were possible for books can have multiple genres, there could be an intersection table with many books going to many genres. I also noticed in the ER diagram that genres with books and authors with books could have a M:M relationship with a possible intersection implementation.
The naming is consistent with entities being plural and the attributes of the table being singular. We also used camel casing for our attributes.
Overall, the database and overview are very detailed and strong. I agree with the first review,w it feels quite realistic for a business like this. My feedback is original and based on my understanding of this project."

Juliet Daryl Consignado

"The name of the bookstore is fun and witty. The overview provides context for the business problem and how a database could help scale the business. It could use more detailed information on how entities of the proposed website with DB backend could capture their intended information and how that would specifically contribute to a solution to the business problem.
Additionally, the overview could benefit from stating facts like we sold 1,000 books last year but were sold out for many copies due to not ordering enough inventory for 50 titles that were in high demand.
There are at least four entities that represented a single idea, and I appreciate that the category and intersection tables are well thought out.
The database outline provides a description of each entity, its attributes and their datatypes, and also the relationships between entities. However, the constraints are missing from the outline, and I think the constraints for each table would provide more insight into the intention of each entity.
The relationship between entities looked correctly formulated and there are multiple M:M relationships with intersection tables. The ERD does provide a logical view of the database. One suggestion would be to add constraints to each entity so it's easier to understand the flow between entities.

There is consistency in naming conventions between entities as plural and attributes as singular. Attributes follow a camel case capitalization structure.
My review of this project outline is all my work without the use of outside sources."

Madelyn Lazar

"Hi team 2! This is a great first step! Here is some feedback: +Great job using consistent naming conventions, +You have the minimum table requirements and enough M:N relationships, +Your relationships are all handled correctly and listed correctly in your outline. -This project tends to get complicated and big pretty fast. You currently have 14 tables which is quite a lot of relationships to handle. I highly suggest just keeping the core necessary tables to make it simpler, especially as we move through future steps and continue to add functionality. However if you are up for the challenge, go for it! -I notice that bookID is listed as varchar(45) in the BookLocations table in the outline, but is listed as INT in your ERD. Make sure everything is consistent and matches. -CustomerID is listed twice in the Orders table in your outline. -For future steps, make sure to have any citations you use. Remember that this includes class starter code or any programs you used (i.e. REACT starter code, Flask starter code, etc.). If you used no resources, list at the bottom."

Madelyn Lazar

"Hi team, great work responding to feedback! One thing to note: I see you still have 'optional' tables left in your outline and ERD which is a little bit misleading. If you currently don't have those tables in your database structure, then please delete them for clarities sake. I highly suggest simplify your database to cut down on the amount of relationships you have to handle, but if you are up for the challenge feel free to leave them in. Whichever option you choose, only the tables you are currently using and are included in your database structure should be there."

Fixes Based on Feedback from Step One

The helpful feedback we received on ED from fellow students encouraged us to add measurable, specific figures into the overview, which we did include. Rafael also pointed out a couple minor typos and style conventions which we missed, which were modified for the final draft.

A couple minor suggestions we decided not to include were Rafael's recommendation to omit PKs in some of our intersection tables. We think this is a helpful and necessary component to include in these entities. Francis noted that in our Database Outline, some of the varchars do not specify a number. We have decided to keep these non-specific at the moment since the sizes of some of these attributes are still uncertain and will become more clear as we progress through the project. Juliet also recommended we added a constraints description to each of the entities. We

think the constraints are made clear through the descriptions, attributes, and relationships included in each entity and are not in need of their own explanation under each entity.

We fixed a couple typos in the Database Outline and streamlined our naming conventions that were initially slightly inconsistent. We also added specific figures for the bookstore in our overview. We took Rafael's advice to standardize the key naming as our intersection tables now have the same naming convention as the regular tables. We also took Madelyn's advice regarding a couple tweaks in our attributes and added a citation section at the end of the document.

Finally, given Madelyn's advice, we removed the optional notation around certain tables and have a fully committed and clear schema. We have opted not to simplify and feel we are up to the challenge of a more robust database.

Feedback from Step Two

Toni Orban

"I can tell how much work you all put into this. You've got a lot of entities, way to go above and beyond. I hope my feedback helps and I look forward to seeing how it turns out.

- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

For the Books table the relationship with OrderItems this is shown as 1:1 in the schema when the outline describes a 1:M relationship between books and OrderItems.
The outline description for Books does not describe the intersection table (BookLocations) that is used to manage the M:N relationship between Books and SLOCS, the intersection table (BookAuthors) used for the M:N relationship between Books and Authors, or the intersection BookGenres which is used for the M:N relationship between Books and Genres.
For the Orders table the relationship with TaxRates is describes as M:1 in the outline, but is shown as 1:1 in the ERD and Schema.
The outline describes the relationship between Books and SLOCs to be M:N, but then the intersection table has the relationship between BookLocation to be 1:1 which is how it is presented in the schema. This would mean that each location can only have 1 book.
The outline describes the relationship between BookGenres and Genres to be M:1, but the schema has it as 1:1
The outline describes the relationship between BookAuthors and Authors to be 1:1 and the schema also reflects this, but the ERD states that the relationship between Books and Authors is M:N, which would indicate that the intersection table should show a M:1 relationship between BookAuthors and Authors.
The outline states that the relationship between Publishers and Books is 1:M, but the schema and ERD show 1:1

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

Yes.

- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

Yes.

- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

Yes.

- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

I wonder if it's necessary to have Counties, States, and TaxRates as three different entities. Counties and States represent the same idea, being location and TaxRate is dependent on location. I would consider combining these three entities into one: SalesLocation or SalesTaxRate, which can have attributes of county, state, and taxRate. This would help to adhere to the 1NF of each entity having it's own idea and the 3NF principal of partial dependencies. There is also an attribute in Orders called taxRate, which appears to be redundant.

- Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)

I had a bit of a hard time verifying this because I could not run it in my database, due to the code specifically referencing cs240_milles29 throughout. I did still look for any errors and could not find any. I'm unsure if this reference is going to cause issues with the TA's or others reviewing your work though.

- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?

In the Customers table described in the outline the attribute for phoneNumber is described as NOT NULL, while the SQL and the schema show it as NULL.
In the Books table described in the outline the attribute for isbn-10 and isbn-13 are described as NOT NULL, while the SQL and the schema show it as NULL.

- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?

Yes.

- In the SQL, are relationship tables present when compared to the ERD/Schema?

Yes.

- In the SQL, is all example data shown in the PDF INSERTED?

Yes.

- Is the SQL well-structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?

The code has comments that clearly delineate each section.

- As a reviewer, clearly describe to what extent your feedback to the team was original (e.g. "all my work") or non-original (e.g. used AI tools per Code Citation Tips).

This was all my work."

Madalyn Lazar

"Hi team 2! Great work in step 2!

+Your schema and ERD look well structured
+you did a good job responding to feedback from your peers and I.

-Your sample data section is lacking. You should have at a MINIMUM 3 rows of data per table. most of your tables, I only see 2 rows.
-Your DDL file is not testable because you hardcoded your username prefixes in your file. It should be a generic file so it can be tested by anyone. For example instead of CREATE TABLE cs340_milles29.Publishers (...), you should have something like this: CREATE TABLE Publishers (...). So make sure to fix this on the next step, so I can run your file and check"

Fixes Based on Feedback from Step Two

Toni offered lots of helpful feedback. They noted a discrepancy between our outline and schema in several relationships between Books and OrderItems, Orders and TaxRates, SLOCS and BookLocations, BookGenres and Genres, BookAuthors and Authors, and Publishers and Books. Largely, we had issues with how our schema was representing relationships with intersection tables. These issues have been resolved and are now accurately represented.

They also noted an inconsistency in how we listed relationships in our outline involving intersection tables. We opted to remove notes of relationships between intersection tables in our object and category table sections and simply represent the M:M relationships which are facilitated by the intersection tables.

They suggested that we consolidate three of our tables into one for the purposes of normalization. We have decided this was a good suggestion and have implemented in our new schema and outline. They also suggested that there was a redundancy by having taxRate as an attribute in Orders. We consider this not to be a redundancy because if tax rates in a certain location change, this way the order has a record of what the tax rate was when the order was processed.

Finally, they pointed out a couple of inconsistencies of attributes listed as NULL in our SQL and schema and NOT NULL in the outline. These inconsistencies have been resolved.

We also noticed upon further review that we neglected to add some of our sample data in the SQL into our documentation. This has been updated and they now reflect each other accurately.

Madalyn noted that we had too few insert statements per table and that our DDL file was hard coded with a database so they were unable to test it. We have rectified these issues in the next step.

Feedback from Step Three

Rafael Carlos Ayala

"**1. Does the UI utilize a SELECT for every table in the schema?**
Yes. The application displays data from each table in the schema individually.
2. Does the UI implement an INSERT form for at least one table in the schema?
Yes. The UI includes input forms that allow users to insert new data into at least one of the tables.
3. Does the UI have at least one DELETE for any one entity?
Yes. There is functionality in place that allows users to delete individual rows from at least one table.
4. Does the UI have at least one DELETE that will remove things from a M:M relationship?
Yes. Deleting entries like an order properly cascades to remove related records in the junction table.
5. Is there at least one UPDATE form in the UI for any one entity?
Yes. The site provides an interface to update data for specific records in a table.
6. Is there at least one UPDATE form in the UI to modify an M:M relationship?
Yes. The app allows users to update entries in an intersection table by modifying foreign key references.
7. Do you have any other suggestions for the team to help with their HTML UI?
The UI is clean and well-organized, and the homepage summary of each table is a helpful touch. One suggestion: instead of having dropdowns in the nav bar for each CRUD action (like "view books," "add books," etc.), it might be more user-friendly to allow a single click on the nav option (e.g., "Books") to take the user to a page that includes all the relevant actions in one place.

This would reduce the number of clicks and streamline navigation for users who just want to quickly access a table view.

8. As a reviewer, clearly describe to what extent your feedback to the team was original (e.g. "all my work") or non-original (e.g. used AI tools per Code Citation Tips).

All my work."

Erin Cook

"Misc:

·Your website is super cool! I like how you have the summaries of each field on the front page.

·The navigation across the top – it might be more visually appealing if there were a more equal number on each line? Totally optional, function is more important!

Does the UI utilize a SELECT for every table in the schema?

·Yes, very impressive considering how many tables there are in the schema!

Does the UI implement an INSERT form for at least one table in the schema?

· Yes, there appear to be INSERT forms for every entity.

Does the UI have at least one DELETE for any one entity?

· Yes, there appears to be DELETE for every entity

Does the UI have at least one DELETE that will remove things from a M:M relationship?

· All entities appear to have DELETE functionality, including the ones in M:M relationships. The code comments give information about which entities can and can't be fully deleted depending on the "ON DELETE" settings for tables with FKs.

Is there at least one UPDATE form in the UI for any one entity?

· Every entity appears to have an UPDATE functionality

Is there at least one UPDATE form in the UI to modify an M:M relationship?

· Yes, every entity has an associated update form, including those in M:M relationships.

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

· Really excellent webpage, the design is very nice. I love all the icons! I sorta feel like I would like some more color/styling, but also it's really elegant being super clean like it is. In any case that's a "when everything's fully working" thing 😊. Great job team 2!

This review is entirely my own work."

Robert Handler

“Hello Group 2,

Does the UI utilize a SELECT for every table in the schema?

Yes the UI shows data from each table in the schema separately.

Does the UI implement an INSERT form for at least one table in the schema?

Yes, many of the tables have insert forms in the UI (Books, customers, orders, etc.)

Does the UI have at least one DELETE for any one entity?

Yes, many of the items have a delete button (Books, customers, orders, etc.)

Does the UI have at least one DELETE that will remove things from a M:M relationship?

Yes, the UI has a delete, like on the orders page, which will have the proper ON DELETE RESTRICT, and ON UPDATE CASCADE.

Is there at least one UPDATE form in the UI for any one entity?

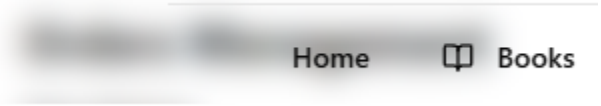
Yes there are many UPDATE forms on the UI (Books, customers, etc.)

Is there at least one UPDATE form in the UI to modify an M:M relationship?

Yes, there are are UPDATE forms in the UI to modify an M:M relationship like the orders page.

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

I am a big fan of how the navigation bar remains at the top, with the main page content showing blurred in the background when you scroll far enough. One suggestion might be to increase the amount of blurring. There are some situations where the nav bar buttons are harder to read because of the background.

- 

As a reviewer, clearly describe to what extent your feedback to the team was original (e.g. "all my work") or non-original (e.g. used AI tools per Code Citation Tips).

- All my work.

Shengdong Li

“The theming and everything looks really unified and clean! A lot of the CRUD functionality is included, including inserts, edits, and deletions.

- *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.
    - Yes, there's a SELECT for every table in the schema. Some of the namings of the table were changed, I'm assuming for readability, such as in the UI the Sales Rates was named differently than the SalesRateLocation table in the schema, and the Locations table was named a bit different than the SLOCs
- *Does the UI implement an INSERT form for at least one table in the schema?* In other words, there should be UI input fields that correspond to at least one table.
    - Yes, the table implements UI input fields and insert works for pretty much every single table!
- *Does the UI have at least one DELETE for any one entity?* In other words, is there a form/link/button that will allow the deletion of a row in at least one table?
    - Yes, the DELETE functionality is fully implemented, a neat-looking trash icon attached to every row, for intersection tables
- *Does the UI have at least one DELETE that will 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.
    - Yes, when I deleted a book Order, it deleted the entry in the Orders table, but did not delete the actual book
- *Is there at least one UPDATE form in the UI 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, clicking the edit icon next to each row shows a row where I can input new values and edit things.

- *Is there at least one UPDATE form in the UI to modify an M:M relationship?* In other words, does the UPDATE allow the user to select a different foreign key value to update the intersection table with?
    - Yes!
- *Do you have any other suggestions for the team to help with their HTML UI? For example, using AS aliases to replace obscure column names such as fname with First Name.*
    - Some pointer events (i.e., finger on hover) would be helpful in realizing whether or not certain events were clickable, but overall the responsiveness of the UI to different sizes, and the nice header-navigation for each table made the app to use.
- As a reviewer, clearly describe to what extent your feedback to the team was original (e.g. "all my work") or non-original (e.g. used AI tools per Code Citation Tips).
    - All my work”

Madalyn Lazar
“Team 2, this is awesome!
I'm really impressed by how much work you have already implemented!

+DDL runs smoothly
+all tables represented in UI
+UI looks extremely readable and clean
+DML looks good and includes all necessary SELECTS

-I noticed you’ve included alternate delete options in your DML (one using the primary key like bookAuthorID, and another using a composite key like bookID + authorID). This is totally fine for prototyping, but moving forward, choose one approach that matches how your frontend will function. You only need to keep the one you're actually using.”

Fixes Based on Feedback from Step Three

Madalyn mentioned that we had some unnecessary DELETE statements in our DML. We deleted those using the primary key of the intersection table and opted to use the DELETE statements with the composite key. Rafael suggested we remove the dropdown menu from every entity listed in the navbar, which we did do. Erin, Rober, and Shengdong all offered a few minor styling preferences. At the time, we have opted not to add these into our frontend, though we may if time allows before the final project is due.