Bookstore Sales Tracker

By Mo Hudeihed and Saurav Shrestha

URL: https://web.engr.oregonstate.edu/~hudeihem/CS340/public/index.html

Feedback by peer reviewer (Step 1)

Feedback by the peer reviewer

Review 1 (Zachary Maes - TA)

Great job on your first draft group 21. You appear to have a strong foundational understanding of the concepts learned so far. Please see my feedback for each of the key questions below. I think you are mostly on the right track but need to add a few more things prior to your final submission.

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

Your overview is concise and describes the problem in a definitive way. You organize the information well and implement clear formatting methods like **bolding** and *italicization* that make entities stick out to the reader.

2.Does the overview list specific facts?

Your overview does list specific facts about the various entities like Customers and Stores. You also do a wonderful job painting the picture of how the business will operate using specific numerical data for revenue, sales volume, number of locations, and number of customers.

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

You appear to have five entities that are all clearly defined and represent a single idea. I don't foresee any issues converting these entities into actual SQL.

(Note to students: If there was an actual issue, now would be a great time to describe what you see and how the group could go about fixing it.)

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

Your outline does a great job describing each entity's purpose, attributes, and relationships. The color and **bolding** go a long way in keeping the details organized. However, I do see a few things that are missing from the outline section:

·You are missing the **AuthorsBooks** intersection table in your outline section. Please add this table to the outline along with the two FK attributes, their details, and a relationships section.

·While you describe the foreign keys in the relationships sections (and your descriptions are correct!), you should also add these attributes to the outline and show the "FK" in the same way that you added "PK". Correct this for **Invoices** and **AuthorsBooks.** Here is an example of how you might change the **Invoices** table to accomplish this:

Invoices: ...(description)...

·invoiceID: int, auto_increment, unique, not NULL, PK

·date: date, not NULL

·bookID: int, FK, not NULL

·storeID: int, FK, not NULL

·customerID: int, FK, not NULL

Relationships: (keep the current descriptions)

5.**Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?**

All one-to-many relationships have been accurately established, as demonstrated by the precise definitions of foreign keys (FK) and primary keys (PK) outlined in the relationship sections. There is also one intersection table **AuthorsBooks** that implements a many-to-many relationship between **Authors** and **Books**. The accuracy of these relationships is also reflected in the correct use of crow's foot notation within the Entity Relationship Diagram. **I recommend that you add the "FK" attributes inside of your ERD to make these relationships more clear.**

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

Good work on your attention to details. Your entity names are consistently plural, capitalized, **bold,** and follow **PascalCase**. Your attributes are singular, colored blue for clarity, and follow camelCase.

Overall, great job on this project draft! If you implement the changes above, they will go a long way in keeping your project organized. This will help streamline your coding and SQL work during future project steps. I am happy to clarify any of my suggestions above.

Review 2 (Jerome Cagado)

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

Your overview is well done. It explains the reason for why BooksRus needs a database and the overview explains the various data that will be tracked. I also like how you bolded the entities and provided color-coded to the attributes. It provides an easy ready so good on that.

2.Does the overview list specific facts?

The overview does list the specific facts about the various entities that you provided. You relate how the author's entity has a relationship with the book entity. In addition, you relate customers to the stores and the overall entities makes sense in a database.

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

You appear to have five entities that display why they are there. You also have each entity represent a single idea to be stored as a list. The only thing that I would suggest is differentiating your attributes that have the same name. For instance, under stores you have phone and you also have phone under customers. Why not make your phone under the stores entities as storePhone and customers phone as customerPhone? This is also the case for firstName, lastName which is the same attribute name for authors. Just something to think about.

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

The outline does indeed outline the entity details and describes the purpose of each, while also listing the attribute datatypes, constraints, and relationships between entities. I feel overall, you guys did a great job.

5**.Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?**

The outline does indeed have a 1:M relationship and they are correctly formulated. There is also one M:M relationship as well. The ERD makes logical sense. The only issue I have is that you technically can make it a bit more complex if you like. The authorID and bookID could technically fall in either the store or invoices. Something to consider.

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

There is consistency in the naming convention. The entities are plural and the attributes are singular. You also capitalize the entities. You guys also opted for proper naming conventions rather than brevity so good job on that.

Great job and good luck on the project!

Review 3 (Katalina Nakoa)

Great job, Group 21, here's my review of your draft:

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

The overview is very well-written and describes the problem that the website with a DB backend aims to solve. It shows the need for better sales tracking capabilities and improved inventory management for BooksRus, a bookstore chain with lots of stores across the nation.

2. Does the overview list specific facts?

Yes, the overview lists specific facts about BooksRus, such as its annual book sales volume, the number of customers served, and the annual revenue generated. These specific facts provide a clear understanding of the scale and scope of the business.

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

There are five entities described: Authors, Books, Invoices, Stores, and Customers. Each entity represents a single idea and the details provided for each entity contribute to a good understanding of the data to be stored.

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

The outline of entity details is thorough and well-organized. It effectively describes the purpose of each entity, lists attribute datatypes and constraints and describes relationships between entities. The use of relationships and the inclusion of intersection tables (like AuthorsBooks) for M:M relationships shows a good use of database design principles.

5. Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

Yes, the 1:M relationships are correctly formulated, and there is one M:M relationship between Authors and Books, implemented with the AuthorsBooks intersection table. The Entity Relationship Diagram could be improved by adding FK attributes to make the relationships clearer, also suggested by other reviewers.

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

Yea, there is consistency in naming conventions. Entities are consistently plural, attributes are singular, and capitalization follows a pattern. It seems like readability was prioritized and maintains a structure.

Overall, you did a good job in creating a draft for the bookstore sales tracker. H. Keep up the good work!

Review 4 (Alma Valenzuela)

Hi Team 21, overall good job on your draft! Here is my feedback:

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

The overview does a good job describing the problem that is to be solved through the use of a database. The bookstore chain, BooksRus, is in need of a database to tracks sales to improve inventory management. I like that the entities and relationships of the database were included in the overview because it helps me better understand how the database will help BooksRus.

Does the overview list specific facts?

Yes, the overview states the name of a bookstore chain and mentions how many books they sell a year, the number of customers they have and the revenue generated. It also mentions the entities and relationships of the database.

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

All of the entities are described. They each represent a single idea and the description explains the purpose of having them included in the database.

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

Each entity has attributes and relationships. The attributes include the datatypes and constraints. The relationships between the entities are clearly outlined through the use of bullet points. The different font styles make it clear to understand what the entities vs attributes are.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

The 1:M relationships are properly formulated and explained and there is at least one M:M relationship. I noticed that the relationships described the foreign keys of the entities but the entities themselves did not have the foreign keys as attributes. For example, Invoices does not have the FK bookID in the outline. I am not sure the reasoning for this but it would be useful to include the foreign keys in the entity outline. The ERD is missing the AuthorsBooks intersection table but, other than this, the ERD represents the outline well and appropriately shows the relationships between the entities.

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

There is consistency in the naming between the entities and attributes. Entities are plural and the attributes are singular. The attributes are written in camelCase.

Canvas Draft Assignment Feedback (Dylan Majewski - ULA)

This is a great idea for a database and it seems like you have a firm understanding how of you want to implement it! The ERD provided gives more than it needs. For a better explanation, see the following canvas page:
https://canvas.oregonstate.edu/courses/1946034/pages/erd-vs-schema?module_item_id=23809319

Feedback by peer reviewer (Step 2)

Joshua Samuel 2d

Hi Group 21! Excellent job on your Step 2 draft! The schema for your database is well-structured and easy to follow. I did encounter a small issue with the *Invoices* table when testing your DDl.sql in MariaDB and described this in greater detail below. Please let me know if I can clarify anything. Good luck with your project going forward!

Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?
Yes. Your schema diagram appears to align exactly with your database outline as well as your ERD. All entities and attributes in the database outline and ERD are present in the schema diagram.

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 to all. All names used throughout your document are consistent between the overview, outline, ERD, and schema diagram. By review of your entity and attribute names, entities are appropriately plural and PascalCase, and attributes are singular and camelCase.

Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?
Yes. Your schema diagram is nicely laid out across the page. Relationships between entities are easy to discern, and no relationship lines are crossed.

Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?
Yes. There is one intersection table: *AuthorsBooks.* This facilitates a M:N relationship between *Books* and *Authors*. This relationship is properly depicted in your ERD with crow's foot notation, and in the schema diagram with relationship lines connecting bookID and authorID attributes to the *AuthorsBooks* table.

Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?
No issues. In review of your provided example data, I do not appreciate any non-normalization issues. None of the attributes in your tables appear to be only partially dependent on the PK. There are no apparent dependencies of non-prime attributes on other non-prime attributes. Taken together, all your tables appear to be in 3NF.

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!)
Yes for the most part. I did encounter an issue when trying to test your DDL.sql file in MariaDB. It was not able to create the *Invoices* table and threw the error `ERROR 1005 (HY000) at line 69 in file: 'DDL.sql': Can't create table \`cs340_samueljo\`.\`Invoices\` (errno: 150 "Foreign key constraint is incorrectly formed)`. After some testing, it looks like the problem is related to the code at line 74, where you define customerID as NOT NULL, but then in the constraint statement at line 92, you set customerID to NULL when it is deleted. Removing the "NOT NULL" declaration from customerID on line 74 fixed this problem for me.

Aside from the *Invoices* table issue described above, all your tables are created as expected and are populated with the appropriate INSERT data when I execute SELECT * FROM queries on them.

In the SQL, are the data types appropriate considering the description of the attribute in the database outline?
Yes. All data types declared in your DDL.sql match those documented in your database outline.

In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?
Yes to the first question; some issues with the second. All the primary and foreign keys appearing in your schema diagram are appropriately defined in your DDL.sql file. As for the CASCADE operations, these all work as expected except for the issue with customerID in the *Invoices* table which I described above.

In the SQL, are relationship tables present when compared to the ERD/Schema?
Yes. All the tables and relationships are consistent with what is depicted in your ERD/Schema when I view the diagram myself in phpMyAdmin.

In the SQL, is all example data shown in the PDF INSERTED?
Yes. In my testing, I was able to write SELECT * FROM queries for all your tables and get the same results depicted in your example data.

Is the SQL well-structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?
Yes. Your SQL is well-structured and easy to follow. It is exported from MySQL, but this does not affect readability, as comments have been added for clarity and the structure of the code is great.

♡ Reply ···

S Shuyao Zeng 23h

Hi Mo and Saurav, this is an awesome design! You have pretty clear orders and ways of presenting. Here are some ideas that I think your draft can follow to become even better.

- **Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?**
- Yes, the schema does present the physical model that follows the database outline and the ER logical diagram. The only thing to mention is that the relationships shown in the schema do not highlight the mandatory and optional lines. If there is a way to show the differences, I suggest you still do that like how the ERD diagram is designed.
- **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. All the names of overview, database outline, ERD and schema entities and attributes keep to be consistent. All the entities are plural and PascalCase. All the attributes are singular and camelCase.
- **Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?**
- Yes. It is conspicuously titled and is placed in the correct order. The name of the tables are attached, and all attributes are listed as the database outline lists them out. The relationship lines are not crossed, and it is easy to tell the differences between two sides of the 1:M relationship lines. It will be better if the primary keys and foreign keys can be labeled so the readers do not need to search about them above.
- **Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?**
- Yes. There is one intersection table named AuthorsBooks. It mediates the M:M relationship between Authors and Books. There are two foreign keys named bookID and authorID inside the table. They are the primary keys inside the Books table and Authors table. It builds two 1:M relationships between Books and AuthorsBooks, Authors and AuthorsBooks. However, the primary key named authorBookID seems useless. Since this intersection table has no exact meaning besides being a connection of Authors and Books. Two foreign keys should be enough.
- **Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?**
- There are no non-normalized issues. None of the attributes in the tables are partially dependent on primary keys. Non-primary attributes also have no absolute dependence on other non-primary attributes. It seems like all the tables are in 3NF.
- **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!)**
- Most parts are correct. The only table that can not be created correctly is Invoices. I use the VS Code to test it and believe that there must be something wrong with the foreign key customerID in the Invoices table. You have the NOT NULL at line 74, but later use ON DELETE SET NULL at line 92, which means that the customerID will be set to null when it is deleted. I suggest you remove the NOT NULL at line 74 so it will not be inconsistent. All the other tables are successfully created and the data are inserted with no mistake. I can get the output tables by inputting SELECT * FROM Tables.
- **In the SQL, are the data types appropriate considering the description of the attribute in the database outline?**
- All data types are appropriate. You can use tinyint or different sizes of varchar instead to make it more accurate, but the current version is already good.
- **In the SQL, are the primary and foreign keys correctly defined when compared to the Schema?**

- Yes. There is one intersection table named AuthorsBooks. It mediates the M:M relationship between Authors and Books. There are two foreign keys named bookID and authorID inside the table. They are the primary keys inside the Books table and Authors table. It builds two 1:M relationships between Books and AuthorsBooks, Authors and AuthorsBooks. However, the primary key named authorBookID seems useless. Since this intersection table has no exact meaning besides being a connection of Authors and Books. Two foreign keys should be enough.
- **Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?**
- There are no non-normalized issues. None of the attributes in the tables are partially dependent on primary keys. Non-primary attributes also have no absolute dependence on other non-primary attributes. It seems like all the tables are in 3NF.
- **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!)**
- Most parts are correct. The only table that can not be created correctly is Invoices. I use the VS Code to test it and believe that there must be something wrong with the foreign key customerID in the Invoices table. You have the NOT NULL at line 74, but later use ON DELETE SET NULL at line 92, which means that the customerID will be set to null when it is deleted. I suggest you remove the NOT NULL at line 74 so it will not be inconsistent. All the other tables are successfully created and the data are inserted with no mistake. I can get the output tables by inputting SELECT * FROM Tables.
- **In the SQL, are the data types appropriate considering the description of the attribute in the database outline?**
- All data types are appropriate. You can use tinyint or different sizes of varchar instead to make it more accurate, but the current version is already good.
- **In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?**
- Yes, all the primary keys and foreign keys are correctly defined. And most CASCADE operations are appropriately declared except for the customerID attribute and Invioices table. It is a very small error. Everything else seems fine.
- **In the SQL, are relationship tables present when compared to the ERD/Schema?**
- Yes, all the tables including the intersection all contain the same attributes as the ERD and schema show. All the data types assigned to different attributes are all correct. The primary keys and foreign keys are set the same way. And the relationships between different tables keep to be consistent in both the SQL, the ERD and schema.
- **In the SQL, is all example data shown in the PDF INSERTED?**
- Yes. I can get all the other example data except for the Invoices table by inputting SELECT * FROM Tables. After I fix the issue of customerID and Invoices, the output of the Invoices table also shows up which means that it is inserted successfully. So basically you have all your insertion sections done well.
- **Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?**
- Yes. It is very well structured and can be easily exported from MySQL. The comments are clear. They divide the whole file into several parts for tables, the intersection table, and data insert. There is a very high degree of completion.

Overall perfect work! I enjoy reading your draft a lot. I hope the feedback is useful for you.

♡ Reply ···

Olivia Cruz 16h

Hello Group 21!

- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?
    - Yes, the schema follows your outline and ER diagram. It was easy to follow and I like how you included the crow's notation for the optional 0 or 1 in some relationships.
- 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?
    - All naming between the outline, ER and schema is consistent..
- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?
    - Yes, the schema is linear so it's easy to read and understand each relationship.
- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?
    - Yes, the intersection table is properly formed with 2 FKs:
        - bookID
        - authorID
- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?
    - The data sample tables in the document have good mock data. The insertion of NULL for the authorID in the AuthorsBooks table was a great inclusion to represent that a value can be unknown.
- 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!)
    - The SQL file did not fully load all of the tables and sample data in PhPMyAdmin.
        - tables : Invoices, AuthorsBooks
        - All sample data was not loaded
- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?
    - Yes, all of the data types are appropriate.
        - The use of the constraint, UNIQUE, for some data types will prevent confusion and ensure proper communication.

- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?
    - Yes, the primary and foreign keys are correctly defined.
    - One cascade option was used in the AuthorBooks table.
- In the SQL, are relationship tables present when compared to the ERD/Schema?
    - Yes, all tables are present in SQL file.
- In the SQL, is all example data shown in the PDF INSERTED?
    - It was hard to tell since the SQL did not load.
- Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?
-
    - Yes, the SQL is well structured and commented.

Great job Group 21! I enjoyed reviewing your project and look forward to your future work. I can tell that a lot of time and effort has been put into your project.

♡ Reply ···

J Jenny Zhong 14h

- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?
    - Yes the schema is accurate to the database outline and follows ER logical diagram exactly.
- 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, naming conventions are consistent throughout the overview and diagrams. Entities are plural and attributes are singular. Capitalization follows camel case.
- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?
    - No relationship lines are crossed. Diagram is clear and easy to read. Relationship lines are clear.
- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?
    - Yes intersection tables are properly formed such as Invoices and AuthorsBooks.
- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?
    - Good mock data is provided for the database. No non-normalized issues. None of the attributes in the tables are partially dependent on primary keys.
- 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!)
    - The SQL file did not execute all statements. Not all data was loaded to the tables.
- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?
    - Yes, price is in decimal (12,2) and all other choices for varchar attributes are well-chosen.
- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?
    - Yes, PK and FK are defined correctly. A cascade option is used in AuthorsBooks
- In the SQL, are relationship tables present when compared to the ERD/Schema?
    - Yes relationship tables are present when compared to the ERD
- In the SQL, is all example data shown in the PDF INSERTED?
    - No, not all example data was inserted to the tables.
- Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?
    - Yes, the SQL code is well structured and commented.

♡ Reply ···

Feedback by peer reviewer (Step 3)

Sonny Box 4d

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

The UI is shown sample data from each table.

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

There is not a way yet to SELECT a list of custom properties, only the obvious ones so far. For example, you guys could do "Top Authors" by ordering the count of the most owned books from each customer, keeping track of authors and ordering high to low.

Does the UI implement an INSERT for every table in the schema?

Yes for sure, they are located conveniently beneath the populated tables.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?

Yea, in the AuthorsBooks table, we can see a M:M at play, joining the bookID and authorID when a subquery returns the correct title or author.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?

I think the DELETE is set up as a cascade so when an author is deleted so do their books from AuthorsBooks.

Is there at least one UPDATE for any one entity?

Yes we can update a book!

Is at least one relationship NULLable?

Yes if a customer is deleted from the Customers table, the customerID in the Invoices table will be set to NULL.

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

I think the UI is good, maybe make the header a bit smaller as it is a bit massive, then you guys can fit it all in one line, or separate it evenly (like 4 then 3 instead of 6 then 1) so it could work fine on mobile.

Good sh*t.

♡ Reply ···

Joel Gilger 5d

- *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, the UI utilizes a select for every table. Books, Authors , Stores, Customers, Invoices and Author Books all contain select queries.

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

There is a select query in the DML.sql file that selects bookID, title from Books used to populate the books dropdown menu, but that is not implemented on the UI.

- *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, Books, authors, customers, stores, and invoices all contain insert options on the UI/ For exmaple the customer page has an "add a customer" section.

- *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).

There is no insertion into the AuthorBooks on each insertion which populates the intersection table for the M:M relationship. I do see on the UI for the Add a book section there is the option to add 1 or 2 authors, but what if it was a text book written by a team of people. I also am not seeing an input value on the sql side for insertion of book pertaining to Author.

- *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 a delete for Authors and Customers and but I believe that the Authors delete needs to also affect the AuthorsBooks table to adjust the M:M relationship.

- *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, There is an update for the Books entity. But I would recommend adding an Edit feature in the Browse section of books to be able to click on it to auto populate the book details in the update section.

- *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.

According to the Outline on the pdf. no relationships are nullable. Everything currently in the database requires input.

- *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.*

I would recommend adjusting the insertion of books to be able handle more than just 2 authors. I would also add a button for the edit feature to auto populate the form for updating a book because as of now everything has to be input manually again which could lead to more typos and errors.

Christopher Poon 6d

- *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, each table has a select query that displays in their UI.
- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*
    - Yes, according to their DML.sql their select Invoices utilizes some where statements to filter by Book ID.
- *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.
    - The only one that does not have it is the intersection table AuthorBooks; however, I am not entirely sure if it needs one?
- *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).
    - Yes, adding books will update authorBooks in the M:M relationship.
- *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 did not see a cascade among the deletes so it might have some delete anomalies.
- *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, updating a book will update the entity.
- *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.
    - Yes, invoices table allows for nullable customers so in theory it could happen if needed.
- *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.*
    - In terms of UI, good work! Everything looks great. It's very simple and easy to read!

Reply ···

Kee Ching (Anson) Poon 6d

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

- Yes, the UI utilizes a SELECT for every table in the schema. Navigating to the Books, Authors, Customers, Stores, Invoices, and AuthorsBooks page, a table shows list of all entries for each entity which demostrates the use of SELECT all.

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

- There are SELECTs in the SQL file that displays all entries of a table, but I don't think there is a SELECT with search/filtering capability with dynamically populated list. To meet this requirement, I suggest adding various search features that for example, lets user search books in a specific year using a query as such: SELECT title, publicationYear,...., etc FROM Books WHERE publicationYear = := publicationYearInput.

Does the UI implement an INSERT for every table in the schema?

- Yes, the UI implement an INSERT for every table in the schema using forms. Attributes for each entities on the schema can be inserted using these forms.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?

- For the M:M relationship (i.e. Authors and Books with AuthorsBooks), I don't think INSERTs into either table add the corresponding FK attributes. From the SQL file, INSERT into Authors involves the firstName, lastName, and INSERt into Books involves the title, yearOfPublication, price. So it seems that they don't insert rows in the intersection table AuthorsBooks.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?

- There are DELETEs for deleting entries from every table. For example: DELETE FROM Customers WHERE customerID = (:customerID); shows that the database can delete a customer through the customer ID.

Is there at least one UPDATE for any one entity?

- Yes, there is an UPDATE for the Books table. However, in the SQL query file, books are updated using the bookID (i.e. UPDATE Books SET title = :title, yearOfPublication = :yearOfPublication, price = :price WHERE bookID = (:bookID);), but in the website, users have to input the attributes (title, author 1, author 2) in order to update the book. I am not exactly sure how will the database be structured, but this might be a discrepancy that requires fixing.

Is at least one relationship NULLable?

- It doesn't appear that any relationships are NULLable. Even though the attribute authorID in AuthorsBooks insersection table does not have a NOT NULL property, authorID itself is NOT NULL and auto incremented in the Authors table. To achieve NULLable relationship, an attribute has to be NULLable initially, for example, when inserting a customer into the Customer table, the database could be set up so that phone number or email is not required.

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.

- I notice that when updating an entry, the update forms requires user to re-type all attributes which might not be the most intuitive way to updating because this could be error-proned. Using an entry point that user can easily input (i.e. Book ID) to get all the pre-populated attributes for that book to be updated, will help make the process more straighforward. Otherwise, I think the UI looks great, I like the color palette and design!

A) **Fixes based on Feedback from Step 3:**

- Updated the INSERT statement for adding a book and updating the book in the DML file. This update to the INSERT statement will also add/update the data in AuthorsBooks table by adding the corresponding FK's (bookID and AuthorID).
- Added Query on the DML file so when the user selects a book to update, it will auto populate the current data for that particular book.
- Added ON DELETE CASCADE constraint to customerID FK in Invoice table. When a customer is deleted from Customers table, respective invoices for the customer are also deleted from the Invoices table.
- Added client-side input validations for all pages.
- Configured UPDATE for the books page to populate the input form and update a book by its ID.

Suggestions we decided not to act on:

"There is not a way yet to SELECT a list of custom properties, only the obvious ones so far. For example, you guys could do "Top Authors" by ordering the count of the most owned books from each customer, keeping track of authors and ordering high to low."

"There are SELECTs in the SQL file that displays all entries of a table, but I don't think there is a SELECT with search/filtering capability with dynamically populated list. To meet this requirement, I suggest adding various search features that for example, lets user search books in a specific year using a query as such: SELECT title, publicationYear,...., etc FROM Books WHERE publicationYear = := publicationYearInput."

It appears that there may be some confusion regarding the project's drop-down/search requirement. The TA's response in this ED discussion post (https://edstem.org/us/courses/49750/discussion/4389869) suggests that there are two acceptable methods for implementation. Our interpretation aligns with one of these methods, where we have successfully implemented dynamic drop-downs featuring a list of authors for book addition. We believe that our approach fulfills the project requirements.

"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.
The only one that does not have it is the intersection table AuthorBooks; however, I am not entirely sure if it needs one?"

The intersection table (AuthorsBooks) is used to manage the relationship between authors and books. The entries are automatically generated/updated when users add a book. Instead of

requiring users to manually manage the entries in the AuthorsBooks table, our UI can automatically handle the updating of records based on the authors selected for a particular book. This approach also simplifies the user experience.

"I would recommend adjusting the insertion of books to be able handle more than just 2 authors."

The majority of the books in our stores adhere to the two-author limit. If there is a significant shift in our dataset, we can enhance our program to accommodate such cases.

Fixes based on Feedback from Step 2:

- Modified customerID foreign key in the **Invoices** table from having NOT NULL constraint to be nullable, as it's set to NULL upon deletion.

Suggestions we decided not to act on:

"The only thing to mention is that the relationships shown in the schema do not highlight the mandatory and optional lines. If there is a way to show the differences, I suggest you still do that like how the ERD diagram is designed."
The database structure and relationship details are clearly described in the database outline and follow the project requirements.

"It will be better if the primary keys and foreign keys can be labeled so the readers do not need to search about them above."

The primary and foreign keys have corresponding symbols in the schema, making them easy to recognize. Primary keys have the "key" symbol to the left. Foreign keys have the same label but don't have the "key" symbol.

"the primary key named authorBookID seems useless. Since this intersection table has no exact meaning besides being a connection of Authors and Books. Two foreign keys should be enough."

Our reason for including auto-incrementing authorBookID as a primary key for the table is because it provides a unique identifier for each record in the intersection table. This makes it easier to reference the relationships, i.e., simplifying JOIN operations. Similarly, if the primary key of the intersection table is a composite key, it might be inconvenient to work with in certain scenarios. Thus, we decided to keep authorBookID as a primary key to simplify any future queries we might have to write.

Fixes based on Feedback from Step 1:

- Added "**AuthorsBooks**" intersection table to the outline, including the two FK attributes "authorID" and "bookID" with their data types and constraints, and a relationship section.

- Added FK to denote foreign keys to the attributes "bookID", "storeID" and "customerID" in the outline of "**Invoices**" table.

- Simplified the ER diagram by modeling it without any attributes. This allowed for basic understanding and logical view of the relationships between entities without going into the details of their properties.

One of the suggestion we decided not to act on was for the following:

"The only thing that I would suggest is differentiating your attributes that have the same name. For instance, under stores you have phone and you also have phone under customers. Why not make your phone under the stores entities as storePhone and customers phone as customerPhone? This is also the case for firstName, lastName which is the same attribute name for authors. Just something to think about."

This was because the attributes are implied by their tables, making their names more concise and their meanings more intuitive. Additionally, since these attributes serve the same purpose (name and contact information) in different tables, it is a good practice to be consistent with the attribute names to make the database easier to understand, manage, and maintain.

Another suggestion we decided not to act on was for the following:

"*The accuracy of these relationships is also reflected in the correct use of crow's foot notation within the Entity Relationship Diagram. I recommend that you add the "FK" attributes inside of your ERD to make these relationships more clear."*

This was because we decided to make our Entity Relationship Diagram much simpler, as described in "Actions based on feedback" section above. Since the diagram focused on just the entities without specifying any attributes, there was no need to add "FK" attributes inside the ERD. These details will be included in our schema diagram in the later part of the project.

Lastly,

"The only issue I have is that you technically can make it a bit more complex if you like. The authorID and bookID could technically fall in either the store or invoices. Something to consider."

We decided not to include authorID and bookID inside the stores table because authors and books are two separate entities with their own attributes. This would also lead to redundancy because author and book information could be repeated for each instance of a store. Similarly, authorID and booksID attributes are already in the invoices table as foreign keys. Our main goal of the database is to minimize data redundancy and dependency, which would not be achieved by following this suggestion.

Upgrades to the Draft version

Apart from the actions taken based on the feedback above, there were no design changes made to the project.

B) Project Outline and Database Outline:

Overview

BooksRus, a bookstore chain with an annual book sales volume of 2,000 books, serves approximately 1,500 customers, and generates $100,000 in revenue annually. An online sales tracker system, utilizing a relational database, is created to enhance the chain's sales tracking capabilities and improve inventory management. This system will efficiently record *Invoices*, capturing details of *Books* purchased by the *Customers* across its 5 *Stores* located throughout the United States. Additionally, the database schema accommodates scenarios of multiple *Authors* per book, multiple purchases for a single book, and the association of purchases with specific customers and stores.

Database Outline

- **Authors:** Records the details of authors of the books in the bookstore. An author can write more than one book.
  - `authorID`: int, auto_increment, unique, not NULL, PK
  - `firstName`: varchar(255), not NULL
  - `lastName`: varchar(255), not NULL
  - Relationship(s):
    - M:M relationship between **Authors** and **Books** is implemented with `authorID` as a FK inside **AuthorsBooks** (intersection table).

- **Books:** Records the details of the books in the bookstore. A book can have more than one author.
  - `bookID`: int, auto_increment, unique, not NULL, PK
  - `title`: varchar(255), not NULL
  - `yearOfPublication`: date, not NULL
  - `price`: decimal(12, 2), not NULL
  - Relationship(s):
    - M:M relationship between **Books** and **Authors** is implemented with `bookID` as a FK inside **AuthorsBooks** (intersection table).
    - 1:M relationship between **Books** and **Invoices** is implemented with `bookID` as a FK inside **Invoices**.

- **Invoices:** Records transaction details between the bookstores and customers. Only one book can be purchased in one transaction (invoiceID).
  - `invoiceID`: int, auto_increment, unique, not NULL, PK
  - `date`: date, not NULL
  - `bookID`: int, not NULL, FK
  - `storeID`: int, not NULL, FK
  - `customerID`: int, not NULL, FK
  - Relationship(s):
    - 1:M relationship between **Invoices** and **Books** is implemented with `bookID` as a FK inside **Invoices**.
    - 1:M relationship between **Invoices** and **Stores** is implemented with `storeID` as a FK inside **Invoices**.
    - 1:M relationship between **Invoices** and **Customers** is implemented with `customerID` as a FK inside **Invoices**.

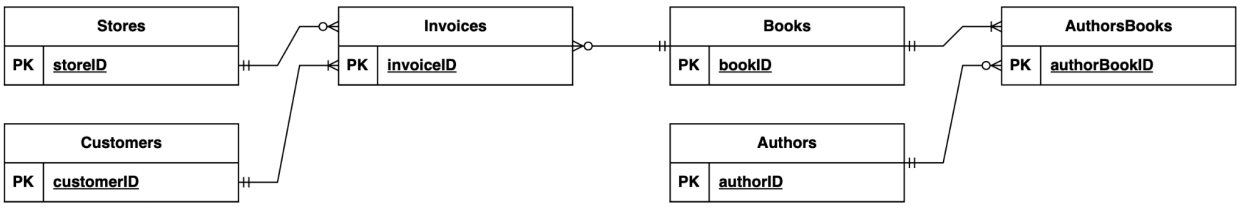
- **Stores:** Records the details of the bookstore locations.
  - `storeID`: int, auto_increment, unique, not NULL, PK
  - `name`: varchar(255), not NULL
  - `phone`: varchar(30), unique, not NULL
  - `address`: varchar(255), not NULL
  - Relationship(s):
    - 1:M relationship between **Stores** and **Invoices** is implemented with `storeID` as a FK inside **Invoices**.

- **Customers:** Records the details of the bookstores' customers information.
  - `customerID`: int, auto_increment, unique, not NULL, PK
  - `firstName`: varchar(255), not NULL

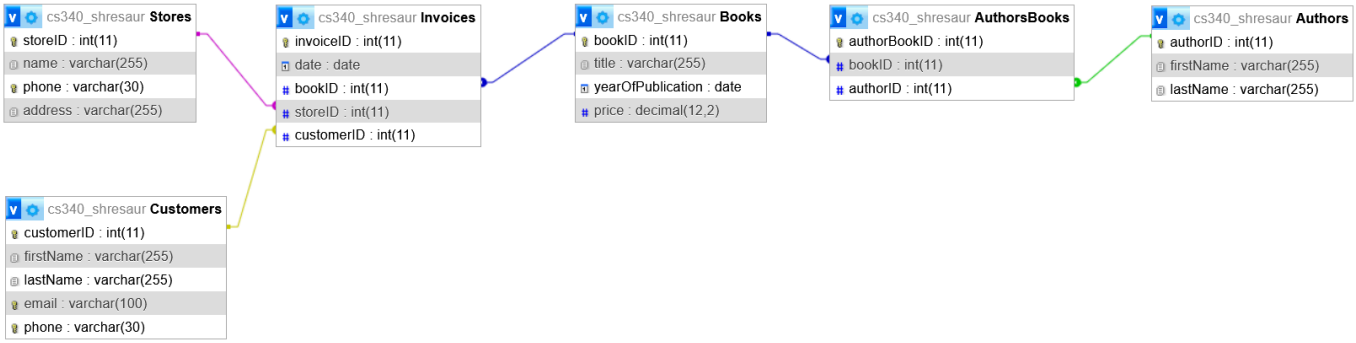
- `lastName`: varchar(255), not NULL
- `email`: varchar(100), not NULL
- `phone`: varchar(30), unique, not NULL
- Relationship(s):
  - 1:M relationship between **Customers** and **Invoices** is implemented with `customerID` as a FK inside **Invoices**.

- **AuthorsBooks:** Records the details of each author from **Authors** and the books they've written from **Books**.
  - `authorBookID`: int, auto_increment, not NULL, PK
  - `authorID`: int, not NULL, FK
  - `bookID`: int, not NULL, FK
  - Relationship(s):
    - 1:M relationship between **AuthorsBooks** and **Authors** is implemented with `authorID` as a FK inside **AuthorsBooks** (intersection table).
    - 1:M relationship between **AuthorsBooks** and **Books** is implemented with `bookID` as a FK inside **AuthorsBooks** (intersection table).

C) ER Diagram:



D) SCHEMA:



E) **EXAMPLE DATA:**

Stores table:

storeID	name	phone	address
1	Metropolitan Stories	402-547-6787	1 Broad Court, Irmo, SC 29063
2	Literary Books	542-888-2167	121 Main Lane, Pataskala, OH 43062
3	Bookshelf Boutique	901-325-4321	9906 W. Union Street, Burke, VA 22015

Customers table:

customerID	firstName	lastName	email	phone
1	Patricia	Carlson	pcarlson@gmail.com	667-222-4521
2	Selena	Lozada	slozada@gmail.com	701-548-1944
3	Michael	Karlsson	mkarlsson@gmail.com	402-666-1234
4	Sergio	Hernandez	shernandez@gmail.com	801-951-3574

Authors table:

authorID	firstName	lastName
1	Edgar	Patterson
2	William	Fitzgerald
3	George	Wolf
4	Mike	Lang

Books table:

bookID	title	yearOfPublication	price
1	Linear Algebra	2005	80.00
2	Discrete Mathematics	1992	90.99
3	Organic Chemistry	2020	120.00
4	Statistics	2021	115.00

AuthorsBooks table:

authorBookID	bookID	authorID
1	1	1
2	1	2
3	2	2
4	3	3
5	4	NULL

M:M relationship between Authors and Books. A book can have one or many authors (the book with bookID = 1 has two authors (authorID = 1 and authorID = 2). An author can have

zero books (authorID = 4 has no books in their name at the store) or many books (authorID = 2 has two books written by them, bookID = 1 and bookID = 2).

Similarly, if an author is deleted from Authors table, the authorID will be set to NULL in AuthorsBooks table. ON DELETE SET NULL operation was used to achieve this.

Invoices table:

invoiceID	date	bookID	storeID	customerID
1	2021-02-25	1	1	1
2	2021-03-19	2	1	1
3	2023-10-10	2	2	2
4	2022-02-05	1	1	4

1:M relationship between Customers and Invoice table. One customer can buy multiple books (customerID = 1 bought bookID = 1 and bookID = 2).

1:1 relationship between Invoice and Customers. Each invoice is associated with one customer.

1:M relationship between Books and Invoices. One book can be purchased by multiple customers (bookID = 2 is purchased by customerID = 1 and customerID = 2).

1:M relationship between Stores and Invoices. One store can sell multiple books (storeID = 1 sold bookID = 1 and bookID = 2).