Heidi Ly

Samuel Zink

Website

http://flip3.engr.oregonstate.edu:9111/ website containing all other pages

Feedback: Step 1

1. Joseph Shing:

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

Yes, the website with a DB backend aims to streamline and manage information for a library. The system seems to be for library transactions and keeping track of the the items in the library and the checkout of items for each user. With the given application of this database, renewals, late returns, reservations and holds for books can also be included as either entities or attributes.

Does the overview list specific facts?

Yes, the overview specifies the number of books in the library (5,000 books) and the number of member served annually (1,000 members). Could be interesting to see some details on how many book categories there are, how many transactions per day, etc. but totally optional.

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

There are 4 entities: (1) Books, (2) Authors, (3) Members, (4) Borrowing Transaction. Each entity makes sense and can have their own respective attributes to be stored as a list.

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

Yes, the outline describes the purpose of each entity, lists the attribute datatypes and constraints, and the relationships between the entities are present.

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 relationship makes sense in which a book can be on multiple borrowing transactions and for each borrowing transaction, there can only be one book. There could be multiple rows for when a user checks out more than 1 book at one time. There needs to be a M:M relationship. You could consider making another entity, like a borrowed_books entity and use that as a junction table in which books and members would be foreign keys with a many to many relationship between those two entities. The ERD provided otherwise makes sense and is logical.

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

Make sure the capitalization is consistent through the entities. I see some attributes capitalized while others are not (i.e: Date_borrowed in the Borrowing Transaction entity). Also make sure that there are no spaces and that snake case or underscores are used for multiple words.

Overall - great job! Just some small clean up activities.

2. Kyle Pfister

I enjoyed reading your proposal for an Online Library Management System! Here are some initial thoughts:

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

The overview does a good job of describing the problem of managing a library with a collection of 5,000 books and around 1,000 annual members. It outlines how the database-driven website will facilitate tracking of books, member details, and borrowing history. However, it could be enhanced by explaining how this system improves over traditional methods or other databases.

Does the overview list specific facts?

Yes, the overview lists specific facts like the number of books and members, which helps in understanding the scale of the library. You could add something about whether you are target public libraries at cities/villages or a private libraries at like a university or college.

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

Yes, four entities are described: Books, Authors, Members, and Borrowing Transactions. Each represents a distinct idea related to library management.

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

The outline describes the purpose of each entity along with attributes and their datatypes. However, more details on some of the constraints and a deeper text explanation of relationships could be beneficial.

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

1:M relationships seem to be formulated correctly, like the one between Authors and Books. However, the proposal could benefit from a M:M relationship, maybe between Members and Books to reflect multiple borrowings? The ERD presents a logical view but can be more detailed.

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

Naming between the overview and entities/attributes is consistent. Entities are plural and attributes singular, adhering to standard conventions. The use of capitalization aids in readability and understanding.

Thanks for sharing!

3. Pierce Fleming

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

Yes. The problems are that there is no way to manage and track transactions of library members when they borrow books, there is no way to track each member's details/which books they currently have checked out, and there is no way to track the status of particular books (whether or not they are currently checked out, who has them checked out if they are checked out). The website with DB back end solves these problems.

Does the overview list specific facts?

Yes. 5,000 books, around 1,000 members are served annually.

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

Yes, there are at least four entities described (Books, Authors, Members, and Borrowing Transaction), and each one represents a single idea to be stored as a list. The Books table stores data about each book in the library like bookID, author, title, etc, and most importantly, is_checked_out. The Authors table is simple and stores a unique authorID and the author's name. The Members table stores a unique memberID, their name, email, and most importantly, checked_out to track which books a member has checked out. The Borrowing Transaction table tracks each transaction and is of course linked the the Books table and the Members table.

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

Yes. The purpose of each table is described, but it needs proofreading (e.g. "Books: Records information about product" should be Books: Records information about books). The outline lists attribute datatypes and constraints and does a good job at describing relationships between entities.

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 following 1:M relationships are correctly formatted: Books to Borrowing_Transaction, Authors to Books. There is one 1:M realtionship that is not clear/formatted properly: "Members to transactionID" which I believe should be "Members to Borrowing Transaction." As it currently stands, there is not a M:M relationship. I believe the relationship between Authors and Books should be a M:M relationship, rather than what is currently stated (many to one). A book can have many authors, and many authors can contribute to one book. The ERD presents a logical view of the database, aside from the aforementioned issues.

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

No. The naming of attributes is not consistent. Some attributes are named using camel case, some with snake case, some with pascal case. For example, in the Members table: memberID is camel case, Name is pascal case, checked_out is snake case. "Borrowing_Transaction" is also listed as a relationship in the Books table relationships description when it should be "Borrowing Transaction" (not snake case).

Recommendations

I believe Borrowing Transaction should be Borrowing Transactions (plural). I believe the relationship between Authors and Books should be a M:M relationship, rather than what is currently stated (many to one). A book can have many authors, and many authors can contribute to one book.

4. Syme Shahidi

Great work Heidi and Samuel- I enjoyed reading through your project draft!

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

Yes, the overview is detailed and describes that their website with a DB back end hopes to streamline information for a library by keeping tracks of the books, members, and members' borrowing books.

Does the overview list specific facts?

Yes, the overview mentions that the library has 5,000 books and 1,000 members each year.

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

Yes, there are 4 entities: Books, Authors, Members, and Borrowing Transaction. Each entity represents a single idea and all comes together to manage and streamline the information for a library.

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

Yes, the outline is quite detailed and provides the purpose, attribute datatypes and constraints, and relationships between entities. One thing I would recommend is to mentions the relationships in terms of entities alongside the attributes. For example, for the members entity, the relationship is described as follows, "one to many relationship with transactionID". I think it should go, "one to many relationship with Borrowing Transaction", as it follows the trend of the other relationships written in your outline.

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 correctly formulated and are between Books and Borrowing Transaction, Members and Borrowing Transaction, Books, and Authors. There is no M:M relationship present. You could maybe do a M:M relationship between books and members as members can check out many books and books can be checked out by many members (if the library has multiple copies of the same book), for example.

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

There is some inconsistency in naming. Some attributes are camelCase like authorID while some are snake_case like is_checked_out. All entities are plural, besides Borrowing Transaction, and all attributes are singular. Capitalization is a bit inconsistent as some attributes are capitalized like Name while email is not for the Members entity, for example.

5. Amadou Diallo

Great work everyone. It was neatly done!!!

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

Yes, the overview describes the problem very well. It also provides how it will be done by making it more efficient while handling thousands of books and members.

Does the overview list specific facts?

The overview specifies the number of books and members. It also provides the type of platform.

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

Yes, there are four entities listed. They each represent a single idea to be stored as a list. For example, the members' entity represents information about the members, such as their contact information.

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

Yes, the outline of the entity is well put together. There are primary keys for each entity and also foreign keys are present. Foreign keys help build a relationship with the different entities. Datatypes and constraints are also well described.

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 formulated correctly. On the other hand, there is no M;M relationship. One easy example would be members and books, as multiple members can borrow books. The ERD presents a logical view of the database, with an addition of an M:M relationship making it even more logical.

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

Throughout the document, there was consistency in both naming in the overview and the entity/attributes. All entities are plural, while attributes are singular. Capitalization was also well-used for naming.

Feedback: Step 2

1. Ngoc-Thao Ly

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

Your outline looks consistent with your updated ER diagram, but your Schema is missing some information. You're missing authorID under Books in the Schema and your outline is missing the Books_bookID foreign key under Authors, the Books_bookID and the Members_memberID foreign keys under Borrowing Transactions. Perhaps you meant to turn bookID and memberID into foreign keys instead, in which case, I'd recommend looking at 5:20 in this video: https://youtu.be/q5wFWfsS-4I?t=321

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?

Your variable names are consistent throughout the pdf, but the names themselves should be edited. There seems to be some inconsistencies in your capitalization. Some of your attributes are in snake case and others are in camel case--like under Members where there is first_name and checkedOut. It says you switched to PascalCase, but you seem to instead have some camel cases added instead by the way. I'm not sure if that was intended. For my project, I personally used camel case.

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

The schema is correctly drawn. However, I would like to mention that Authors and Books should technically have a M:N relationship, as an author can right many books, just as a book may have many authors.

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

The intersection table, BorrowingTransactions has two foreign keys called bookID and memberID to connect the two tables. As mentioned earlier, the way the foreign keys were made seem to be incorrect. I would also consider that Members could technically not have any books rented out on their account, and a book may not have ever been rented out--so it would be an optional relationship.

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

I don't see any dependency issues, but is_checked_out under Books is a confusing attribute. Is there only one copy of each book on your website? Considering that its an online website to allow renting/borrowing, it seems a little odd that a book can only be rented out by 1 person at a time. If this is intentional, it may help to clarify this in your overview. Similar issue with checkedOut under Members.

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 does not have any issues when its run on phpmyadmin.

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

The data types look correct.

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

The primary keys and foreign keys look correctly defined, but no CASCADE operations were written.

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

The relationship tables do correlate to the schema correctly.

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

All the example data is shown.

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

The SQL is organized and easy to read.

2. William Sullivan

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

There are some inconsistencies between the Schema, the ERD, and the database outline. The ERD shows authorID as a foreign key in the Books table, but this is not present in the Schema. In the Schema, the Authors table has an attribute called “Books_bookID” that is not present in the ERD. This appears to have the relationship backwards from what it is supposed to be. The Schema is drawn such that there is a 1:M relationship between Books and Authors, so each book can have multiple authors and each author can only have one book. In the BorrowingTransactions table in the Schema, there appear to be duplicate attributes. There is a bookID as well as a Books_bookID, and there is a memberID as well as a Members_memberID. These duplicates should be removed to match the ERD and the outline.

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?

There are inconsistencies. The naming appears to be consistent across the outline, ERD, and Schema (other than duplicates that were mentioned above). All tables are plural and capitalized. However, the use of snake case versus camel case is not consistent in the attribute naming. Some names are snake case, such as is_checked_out, first_name, and last_name. But then others are camel case, like dateBorrowed and checkedOut. There doesn't seem to be any pattern that is followed, it's just random on what kind of casing is used.

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

The Schema is generally easy to read, but as noted before, there are issues with it that end up making it a bit confusing to understand what is supposed to be happening. There are duplicate attributes, foreign keys missing, foreign keys that are there that shouldn't be, and relationships that are drawn backwards.

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

There is no intersection table and there is currently no M:N relationship drawn in the Schema. It is unclear which relationship is supposed to be many to many. There is a line in the outline for the Books table that reads, "many to many relationships with books implemented as FK in books." This line is in the Books table and says there is a many to many relationship with the books table. I am not sure what this is trying to say. There is also a sentence in the fixes based on feedback section that says, "The other update given by peers was to add a many to many relationships between the books and authors which have been implemented." However, there is not a many to many relationship between books and authors. It is drawn as a 1:M between books and authors.

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

The only potential normalization issue I see is in the Members table, where there is an attribute called “checked_out”. It is unclear exactly what this attribute is supposed to be referencing. I’m not sure if it is supposed to be saying how many books a member has checked out or something else, and I don’t know how or when it is updated. Also, in the outline, this attribute is labeled as auto_increment, so it will end up being a unique integer for each member. I believe this attribute needs to be made more clear what it is referencing.

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, the SQL file is syntactically correct. I was able to run the SQL file on my database with no errors, and all tables were created with sample data populated.

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

In general, the data types are appropriate. In the Members table, the checkedOut attribute is not set to auto_increment, whereas it is listed as that in the outline. I believe it’s best to not have that set for this attribute, but you may just want to update that in the outline. In the BorrowingTransactions table, the transactionID, bookID, and memberID have a unique constraint on the combination of the three. However, this seems redundant because the transactionID will already be unique on its own by being the primary key of the table. I believe this constraint can be removed.

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

All of the primary keys are correctly defined. I believe the foreign keys are correctly defined as well, however they do not all match the Schema. In the SQL file, authorID is a foreign key in the Books table. This seems to be the correct way to set it up, but this foreign key is missing from the Schema. The Schema should be updated to reflect this. There are no cascade operations declared, so users will be unable to delete any rows that are referenced in child tables. For example, if you wanted to delete a book because you were removing it from the library or it was lost, you would be unable to if there were any transactions that occurred with that book. It may make more sense to include a constraint for “ON DELETE CASCADE” so that you can delete books from the database. This will then delete any borrowing transactions that included that book to ensure the integrity of the database. Consider which cascades should be implemented for the other tables as well.

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

There are currently no relationship tables in SQL or in the Schema.

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

Yes, all data from the PDF is inserted.

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

It is well structured but it does not have many comments.

3. Alejandro Hernandez

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

There are a few inconsistencies here, mainly the lack of an authorID in the Books table as well as duplicate foreign keys in the BorrowingTransactions table. It is worth noting that the sql code does follow the outline correctly but the schema that was included is missing and including a few pieces of incorrect information. There is also a bookID foreign key that should not be present in the Authors table.

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?

a) The naming convention is not very consistent, as there is use of snake_case and camel case all throughout

b) All entities are plural and each attribute is singular

c) The capitalization is consistent for the entities and it seems there was an effort to make all the attributes lowercase except for the ISBN attribute in the Books table. There is definitely consistency in this aspect if the schema is ignored, but the schema introduces an uppercase first letter in many of the attributes.

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

Yes, the schema is easy to read and understand, and there are no lines that cross in the diagram. The relationships between each entity are clear as well.

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

I did not find an intersection table present.

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

There is a possible issue with the checkedOut attribute in the Members table, as there doesn't seem to be a clear reason why it would be necessary, given you can check a member's transactions and see if they are currently in possession of a book. If it is an attribute that keeps count of how many books the member has checked out, then it should not be set to auto_increment

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, the syntax follows the conventions that we have learned thus far, and the import files did not cause any issues.

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

The data types are appropriate although, I personally think an integer (such as tinyInt) would be a better data type for is_checked_out and make the value 1 or 0, rather than a varchar.

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

Yes, the appropriate syntax is always used when defining primary keys as well as the foreign key constraints in each table. These match up with the schema. There are no CASCADE operations declared.

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

All tables are present that are referenced in the erd/schema but no intersection tables are present.

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

Yes, all the data is inserted in the SQL.

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

Yes, the code is easy to follow, although it would be nice if each operation had a short description of what the table represents (not the insert operations, just the create).

4. Brian James

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

No, there are inconsistencies between the Schema, the ERD, and the database outline. The Schema deviates from the ERD by missing the authorID foreign key in the Books table and introducing an attribute called “Books_bookID” in the Authors table, which is not present in the

ERD. Additionally, there are duplicate attributes and missing foreign keys in the BorrowingTransactions table.

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?

a) There are inconsistencies in naming between the overview, outline, ER, and schema entity/attributes. While table names are consistent, attribute naming lacks consistency in terms of casing, with a mix of snake case and camel case used randomly.

b) Entities are consistently named in plural form, and attributes in singular form.

c) Capitalization for naming is consistent, with PascalCase for table names and camelCase for attribute names.

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

The Schema is generally easy to read, but inconsistencies and missing elements make it confusing. Duplicate attributes, missing foreign keys, and backwards relationships detract from its clarity.

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

No, there is no intersection table and no clear many-to-many relationship drawn in the Schema. The relationship between Books and Authors appears to be drawn as a 1:M relationship instead of a many-to-many relationship as intended.

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

There is a potential normalization issue with the "checkedOut" attribute in the Members table, as its purpose and update mechanism are unclear. Additionally, the redundancy of unique constraints in the BorrowingTransactions table may indicate a non-normalized design.

Is the SQL file syntactically correct?

Yes, the SQL file is syntactically correct and can be executed without errors.

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

In general, the data types are appropriate. However, there are inconsistencies such as the "checkedOut" attribute not being set to auto_increment as indicated in the outline. Additionally, redundant unique constraints in the BorrowingTransactions table should be removed.

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

While primary keys are correctly defined, there are inconsistencies with foreign keys between the SQL file and the Schema. The SQL file correctly includes the authorID foreign key in the Books table, which is missing in the Schema. No CASCADE operations are declared.

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

No, there are no relationship tables present in either the SQL or the Schema.

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

Yes, all example data from the PDF is correctly inserted into the tables.

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

The SQL is well structured but lacks sufficient comments for clarity and understanding. More comments should be added to explain the purpose of each table and attribute.

Feedback Step 3

Christopher Jones

- **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.**

- The UI appropriately utilizes SELECT queries for Books and Members tables, displaying corresponding data with correct SELECT queries but don't see tables for Authors or BorrowingTransactions.

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

- Did not see SELECT queries that utilize search/filter.

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.

- Input fields for INSERT are present for Books and Members tables, aligning with their respective attributes, no input fields for Authors and BorrowingTransactions.

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, AuthorID is added in Books but no intersection table.

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.

- DELETE present but not for 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 UPDATE functionality for example of member allows updates to first_name, last_name, etc.

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 NULLable relationship.

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.

- Splitting fields like name into first/last would definitely be a great idea to avoid having to parse and separate the values upon retrieval.

Howard Wang

·**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.**

Books and Members has the corresponding table header for the proper SELECT queries. Some example data would be helpful in the Books table, but the SELECT query is correct. No tables for Authors or BorrowingTransactions.

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

It does not look like there were any search/filter SELECT queries.

·**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 are UI input fields to INSERT for Books and Members with their corresponding attributes. Need fields for Authors and BorrowingTransactions.

·**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).**

AuthorID added in Books but no intersection tables were made in the HTML or the SQL

·**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 DELETE is present for a 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?

UPDATE for member is present, allowing members to update their first_name, last_name, and email

·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 no NULLable relationships present. A lot of the attributes appear to be NOT NULL.

·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.

Adding some CSS to the tables could help with visual clarity of the displayed data. The update form should include first name and last name instead of using one input field for both as that is what the attributes specify.

Trevor DePalatis

- *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.

Every table in the schema in the SQL has a corresponding SELECT. However, the outline includes a table that is not represented in the code, BookAuthors.

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

I did not see any. The selects mainly brought the basic information from the tables.

- *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 are only inserts for Members and Books. There needs to be ones for Authors and Borrowing Transactions added.

- *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, the insert for Books adds the author ID, but there is no code written anywhere for the table that facilitates 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.

There are deletes for Books and Members, however because the M:M table is not here, it will not perform in the way that is required.

- *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 Members that can update first name, last name, and email.

- *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.

This feature was missing. The only updatable functionality current was for non-null selections.

- *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.*
- It may be helpful to be able to delete a book simply by pressing a button connected to the book itself, rather than having to type in the book ID in a separate category.
- Currently for Add Member, there is only one text entry for the name, while in the ddl you have a first name and a last name variable. It would be a good idea to have two boxes to match for first and last name.

Brandon Tsai

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.

    - I found SELECT statements for Members, Books, Authors, and BorrowingTransactions but it looks like there is one missing from BookAuthors. Looks like memberID was also mispelled in the first SELECT statement just to let yall know so it doesn't break anything when you run the sql file

- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*
    - It looks like the SELECT statements are currently only used to display the table data, it doesn't look like there are any search/filter SELECTs.
- *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.
    - Currently there are only INSERTs for the Members and Books 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).
    - Yes, the INSERT statement for books adds the author_id FK. There doesn't seem to be an INSERT for the M:M table at the moment.
- *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 are DELETE statements for Members and Books, but the M:M table doesn't have an ON DELETE CASCADE statement on its FKs.

    - I think it would help to use member_id in the WHERE clause for the Members DELETE statement, as it would save the trouble of having to make an extra subquery to get the first and last name when deleting a member.
- *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 Members table.
- *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, looks like this is missing.
- *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 think it would help to split the name input into first name and last name, in case that causes issues for you guys in the future when trying to retrieve the form data and having to separate the full name into first/last names.
    - Would help to add a dropdown menu in the Add Book section for authors.
    - I think setting the borders for the tables displaying data would be a good improvement as it's a bit hard to tell which values correspond to the headers.

Fixes based on Feedback from Previous Steps:

Step 1 fixes: Throughout the reviews we were given all of them credited as fundamental ideas and structure being solid. The only revisions we were seriously given were on naming conventions which have been remedied.
Upgrades to the Draft version:

- Fixed Formatting: Ensured consistent and uniform formatting for improved readability
    - Plural names for entities and entities use Snake_Case
    - Attributes are all underscore and use camelCase

- Modified relationships to accurately reflect many:many associations between books and authors, refining the relational dynamics within the database.

Step 2 fixes: The feedback we got from step 2 was very helpful as it identified many mistakes we did not catch.

Consistency in Naming Conventions:

- Updated ERD, Schema, and Outline so entities are PascalCase and entities are camelCase

Alignment of Schema with ER Diagram and Outline:

- Updated schema so that authorID was not missing under Books
- Updated outline so foreign keys, bookID and memberID are correct
- Updated BorrowingTransactions to have correct entities and foreign keys for bookID and memberID

Normalization Issues:

- Deleted checkedOut since storing the number of books checked out directly in the Members table can lead to potential update anomalies and redundancy, it's better to calculate this value dynamically when needed using a query

SQL and Comments:

- Updated SQL to reflect all ERD and Schema changes and made more detailed comments

Relationship Tables and Sample Data:

- It was pointed out that authors and books have a many to many relationship. So we created an intersection table to add to our outline, ERD, and schema
- Updated all relationship descriptions to accurately reflect foreign keys and tables
- We decided to leave BorrowingTransactions foreign key alone as it is not a intersection table and does not represent a many to many relationship, it's more of a transactional record, it just maintains the relationships between Books and Members
- We added an intersection table BooksAuthors to represent the many to many relationship between books and authors. Each record in this table pairs a book with its author(s), establishing a direct relationship between the two entities.

Step 3 fixes:

Response to Feedback - Updates and Actions

The vast majority of the feedback was on the SQL. Part of it being naming conventions, syntax errors, and forgetfulness. All of these have been remedied. We changed conventions to camel case, fixed the syntax errors, and added the corresponding queries we forgot to include. One thing we forgot to do was add NULL values for a relationship. We didn't have one present so we added publishers with a one to many relationship with books to remedy this. We've updated our UI to make it more user friendly.

Project Outline - Updated

1. Online Library Management System
   Our proposed website, driven by a database, aims to streamline and manage information related to books, authors, members, and borrowing transactions for a library with a collection of 5,000 books and serves around 1,000 members annually. The system will facilitate efficient tracking of books, member details, and borrowing history. Members can borrow up to multiple books and each book can be associated with an author. The database-driven website will enhance the library's operations by providing a centralized platform for managing these entities and their relationships.

Database Outline

a. Books: Records information about product
   i. bookID: int, auto_increment, unique, not NULL, pk
   ii. title: varchar(120), not NULL
   iii. ISBN: int(13), unique, not Null
   iv. publisherID: int
   v. Relationship:
      - one to many relationship with Borrowing_Transactions (bookID foreign key in BorrowingTransactions)
      - many to many relationships with Authors (implemented via BookAuthors intersection table with bookID implemented as foreign key

b. Publishers: Publisher table stores information about publishers.
   i. publisherID: Int auto_increment not NULL, pk
   ii. name: varchar(120) not NULL,
   iii. website: varchar(120) not NULL
      - One-to-many relationship with Books: Each publisher can publish multiple books. Implemented via the publisherID foreign key in the Books table, referencing the publisherID primary key in the Publishers table.

c. Authors: Records data about the author of the books

   i. authorID: int, auto_increment, unique, not Null, pk

   ii. firstName: varchar(50)

   iii. lastName: varchar(50)

   iv. Relationship:

      - many to many relationship with Books implemented via BookAuthors intersection table with authorID implemented as foreign key

d. BookAuthors: Records the relationship between books and their authors. Each record represents a pairing of a book with its respective author(s).

   i. bookID: int, fk, pk

   ii. authorID: int, fk, pk

   iii. Relationship:

      - many to many relationship with Books (Implemented via the bookID foreign key in the BookAuthors table, referencing the bookID primary key in the Books table)
      - many to many relationship with Authors (Implemented via the authorID foreign key in the BookAuthors table, referencing the authorID primary key in the Authors table.)

e. Members: Records information about members at library

   i. memberID: int, auto_increment, unique, not Null, pk

   ii. firstName: varchar(50), not NULL

   iii. lastName: varchar(50), not NULL

   iv. email: varchar(100), not NULL

   v. Relationship: one to many relationship with BorrowingTransactions (memberID foreign key in BorrowingTransactions table)

f. BorrowingTransactions: Records information about transactions, links members to books

   i. transactionID: int, auto_increment, unique, not Null, pk

   ii. bookID: int, unique, not Null, fk

   iii. memberID, int, unique, not Null, fk

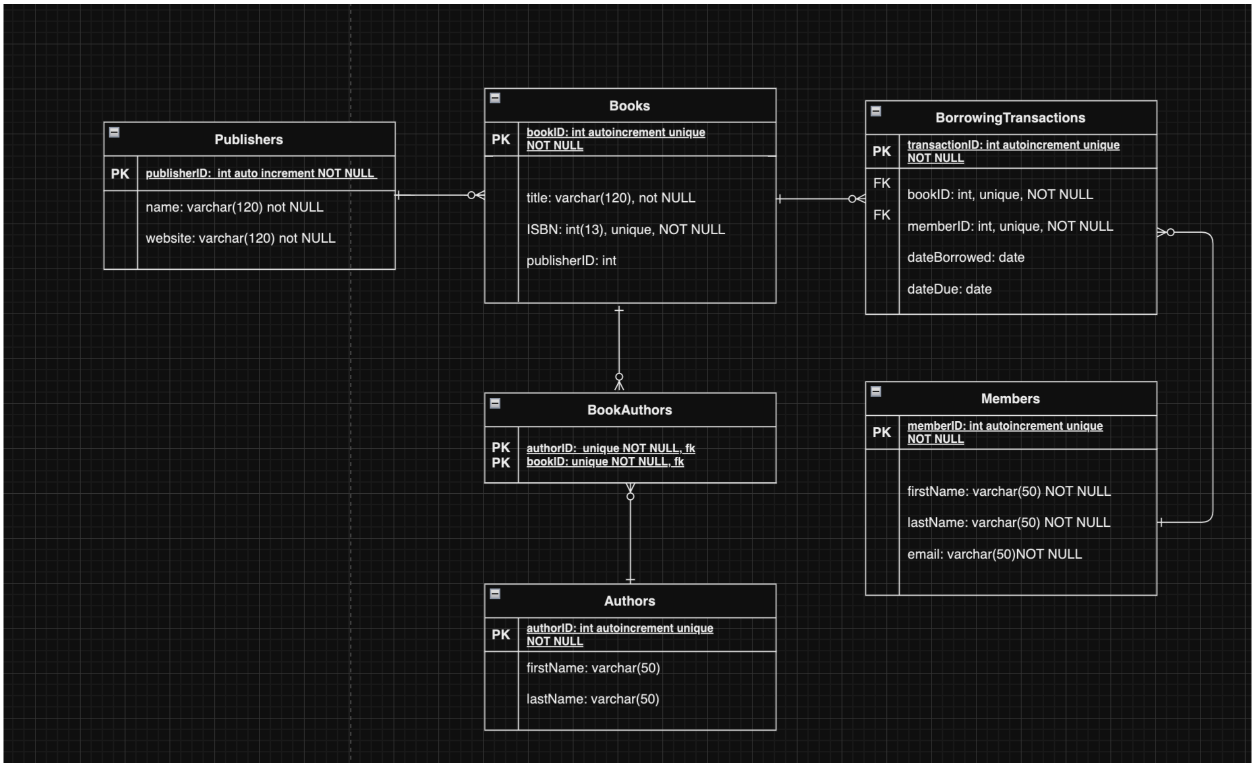
   iv. dateBorrowed: date

   v. dateDue: date

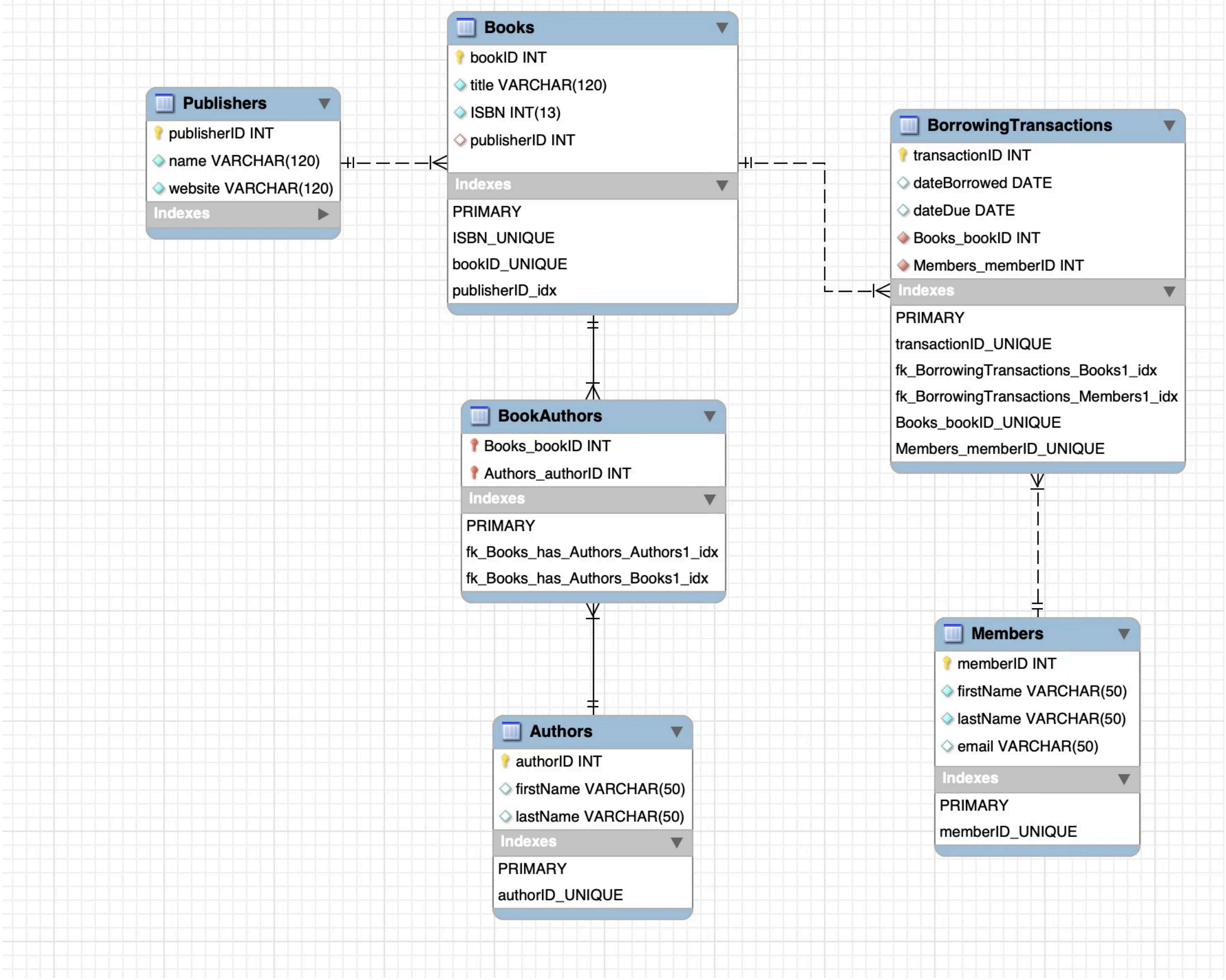
   vi. Relationship:

      - many to one relationship with Books (implemented with bookID foreign key in BorrowingTransactions)
      - many to one relationship with Members (implemented with memberID foreign key in BorrowingTransactions)

ERD - Updated:


Publishers
PK
publisherID: int auto increment NOT NULL
name: varchar(120) not NULL
website: varchar(120) not NULL
Books
PK
bookID: int autoincrement unique NOT NULL
title: varchar(120), not NULL
ISBN: int(13), unique, NOT NULL
publisherID: int
BorrowingTransactions
PK
transactionID: int autoincrement unique NOT NULL
FK
bookID: int, unique, NOT NULL
FK
memberID: int, unique, NOT NULL
dateBorrowed: date
dateDue: date
BookAuthors
PK
PK
authorID: unique NOT NULL, fk
bookID: unique NOT NULL, fk
Members
PK
memberID: int autoincrement unique NOT NULL
firstName: varchar(50) NOT NULL
lastName: varchar(50) NOT NULL
email: varchar(50)NOT NULL
Authors
PK
authorID: int autoincrement unique NOT NULL
firstName: varchar(50)
lastName: varchar(50)

Schema - Updated:



Books

bookID INT

title VARCHAR(120)

ISBN INT(13)

publisherID INT

Indexes

PRIMARY

ISBN_UNIQUE

bookID_UNIQUE

publisherID_idx

Publishers

publisherID INT

name VARCHAR(120)

website VARCHAR(120)

Indexes

BorrowingTransactions

transactionID INT

dateBorrowed DATE

dateDue DATE

Books_bookID INT

Members_memberID INT

Indexes

PRIMARY

transactionID_UNIQUE

fk_BorrowingTransactions_Books1_idx

fk_BorrowingTransactions_Members1_idx

Books_bookID_UNIQUE

Members_memberID_UNIQUE

BookAuthors

Books_bookID INT

Authors_authorID INT

Indexes

PRIMARY

fk_Books_has_Authors_Authors1_idx

fk_Books_has_Authors_Books1_idx

Members

memberID INT

firstName VARCHAR(50)

lastName VARCHAR(50)

email VARCHAR(50)

Indexes

PRIMARY

memberID_UNIQUE

Authors

authorID INT

firstName VARCHAR(50)

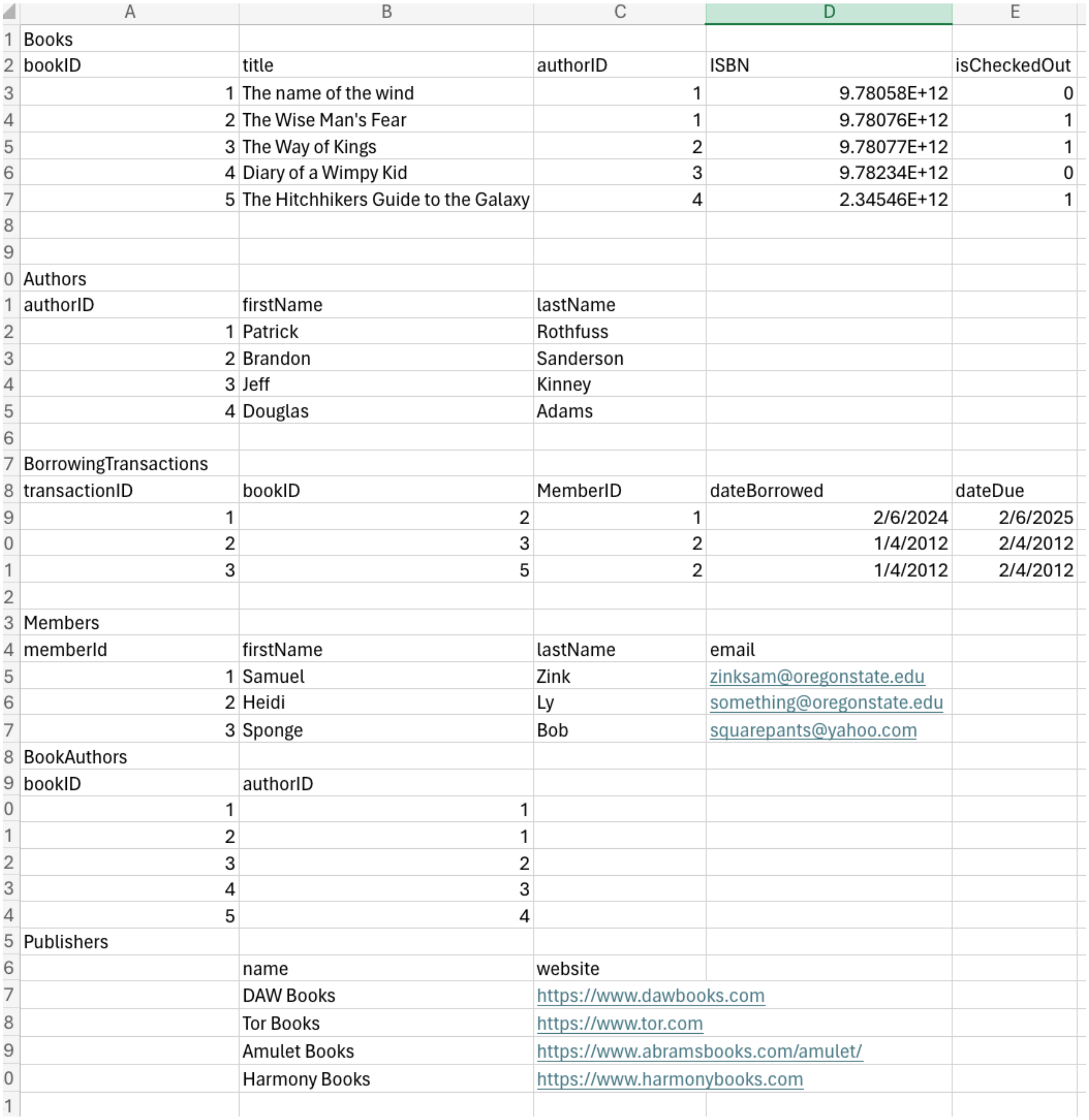
lastName VARCHAR(50)

Indexes

PRIMARY

authorID_UNIQUE

Sample Date - Updated:



	A	B	C	D	E
1	Books				
2	bookID	title	authorID	ISBN	isCheckedOut
3	1	The name of the wind	1	9.78058E+12	0
4	2	The Wise Man's Fear	1	9.78076E+12	1
5	3	The Way of Kings	2	9.78077E+12	1
6	4	Diary of a Wimpy Kid	3	9.78234E+12	0
7	5	The Hitchhikers Guide to the Galaxy	4	2.34546E+12	1
8					
9					
0	Authors				
1	authorID	firstName	lastName		
2	1	Patrick	Rothfuss		
3	2	Brandon	Sanderson		
4	3	Jeff	Kinney		
5	4	Douglas	Adams		
6					
7	BorrowingTransactions				
8	transactionID	bookID	MemberID	dateBorrowed	dateDue
9	1	2	1	2/6/2024	2/6/2025
0	2	3	2	1/4/2012	2/4/2012
1	3	5	2	1/4/2012	2/4/2012
2					
3	Members				
4	memberId	firstName	lastName	email	
5	1	Samuel	Zink	zinksam@oregonstate.edu	
6	2	Heidi	Ly	something@oregonstate.edu	
7	3	Sponge	Bob	squarepants@yahoo.com	
8	BookAuthors				
9	bookID	authorID			
0	1	1			
1	2	1			
2	3	2			
3	4	3			
4	5	4			
5	Publishers				
6		name	website		
7		DAW Books	https://www.dawbooks.com		
8		Tor Books	https://www.tor.com		
9		Amulet Books	https://www.abramsbooks.com/amulet/		
0		Harmony Books	https://www.harmonybooks.com		
1					