Peng Zhang & Tyson Reitenbaugh
CS 340: Introduction to Databases
March 19, 2019
Final version of project

Library Database

http://flip3.engr.oregonstate.edu:3028/

Feedback by peer reviewers (Step 6)

Review feedback by Nathan Fehlbaum:

Looks good. The only other thing I could see that would be a nit-pick would be the Books, Users, Genres, and Rentals categories do not readjust with the side of the window.

Review feedback by Samuel Judkis:

Nice job! I really like the + buttons to add multiple entries for author and genre, it works very smoothly. I was able to add blank books nand users without filling in any of the forms. I think you can fix this by making those fields required, unless there were some you wanted to leave as optional. You probably know this already since you mentioned it for duplicate books but I get an error for duplicate genre/isbn combination as well. I also noticed that I could return a book to the library with a timestamp that is earlier than when the book was checked out. This might be something you want to check for in your back end logic. The actual functionality of renting and checking out books is nice though, but I get errors when I check out or return without filling out the form. UPDATE: The update functionality seems to work well. I like you can continuously add new authors and genres, and that the multiple fields all show up in the modify form when you go to make a change. DELETE: The delete buttons themselves seem to work well, I haven't found any issues there. When there are multiple authors or genres associated with a book though, it might be nice to have a specific delete button next to each so that the user can choose to delete one directly. I was able to delete them by removing the text in the form

input and submitting but it wasn't clear if that would work, I just kind of guessed. Purely a design choice though, so up to you. It definitely seems to work as is.

Review feedback by Melody Reebs:

Hi Peng,

Everything seems to be working correctly in regards to your delete functionality. In response to Christopher's comment, I think your delete genre logic makes sense--just removing the genres from the associated books as opposed to deleting the books themselves.

Review feedback by Clarissa Gasiciel:

Hi Peng!

- Delete functionality works well for books! One suggestion is to have a delete button for each entry rather than delete via entering the book's id number. This could help prevent the user accidentally deleting the wrong book or trying to delete an entry that doesn't exist.

- Delete functionality is solid for genres as well. When I delete a genre, the book associated with it is left alone except its genre is blank. Nice job! Again, I would recommend using a button rather than entering the genre id to help prevent user error.

- Delete also works great for users. Same suggestion regarding button etc. etc.

Review feedback by Christopher Mello:

The website looks awesome! Not much at all to update.

Some Comments:

-Deleting a genre doesn't delete all books associated with that genre. Im sure

you are aware of this, however, just something to not if you wanted different functionality.

-A non-functional note is that if you slide right on most tabs, there is a large white space. It is somewhat small but can be something to look at!

Other than that, the website looks awesome!

Good job cant wait to see it in a week!

Actions based on feedback (Step 6)

- **We fixed UI for nav bar. Now it can automatically adapt to the screen size.**

Feedback by peer reviewers (Step 5)

Review feedback by Evan Linepensel:

```
Hi Tyson,

Looks good overall.   Like Sam already indicated, the update and
delete functions seem to be working as designed.

I found a couple things that weren't already mentioned:

1) When adding a book, if you put in a non-integer into the ISBN
field, it takes you to an error page, but you can't return to the
site, other than using the back button in the browser.  This is really
a nit-pick, but it would be nice to have the site display an error
while still staying on the page and letting the user know the ISBN has
to be a number.

2) When you add a book, you can give it a Genre that doesn't currently
exist in the Genre entity.  It may be nice to either a) not let the
user add a book from an unknown genre, or b) also update the genre
table with the new genre added when adding the book.

Aside from those things and what others pointed out already, the site
looks good and works well.  Good work and good luck with the remainder
of the project!
```

Review feedback by Fedor Titov:

Hi Tyson, looking good.

To lead off I want to point out that having that the way you can update a book's genre isn't the best in my opinion. The way it's a text input means you can accidentally enter a new genre without meaning to do so. This might be a choice to make it easier to enter new genres, but from my point of view it can introduce a new source of error that might not be immediately caught. For me it makes sense to just be able to create new genres from the genre page.

For incorrect input, like trying to rent a book and entering a name instead of an ID, it would be nice to have some kinda of input validation that would notify the user the problem instead of leading to an error page which you'd have to back up from manually.

Also I see you have checkout/return dates, but they aren't displayed anywhere. Perhaps they're being used in the back-end, but it's harder to see what for. It just seems weird to me to have a checkout date, yet not have it be represented by anything. Also u can return a book on a date before it's checked out.

If you add an author and you don't include an ISBN, it tacks on the author to one of the books. It seemed to only tack them onto book id 36 for some reason, and stopped after I deleted it, so I'm not sure if it'll still be a problem.

Hope these helped! Good luck on the rest of the project.

Review feedback by Melody Reebs:

Hi Peng,

Great work so far! Nice job especially on getting the multiple authors and genres working. Here are some comments:

- **I noticed that when I tried to add a book with a duplicate ISBN, it threw an error, which I think makes sense. However, I didn't see in your outline the requirement that the ISBN field is unique**
- **Again, not sure if you're still working on it, but I didn't see any search or filter functionality**

Otherwise I was able to update a Book entity with no issue

Review feedback by Clarissa Gasiciel:

Hey Peng!

Update functionality works well for books! I like that a user can click the book's title to get update and delete functionality for that particular book. Having a button or link that specifically says "Update" might also make a good addition as clicking on the book to edit might not always be intuitive.

Actions based on feedback (Step 5)

- **Generally speaking, ISBN is not allowed to be repeated, so we added this feature to the database, we did not write in the outline because this is a well-known problem.**
- **We add a new feature that allow users search their book by title.**

Feedback by peer reviewers (Step 4)

Review feedback by Evan Linepensel:

```
Hi Tyson,

I really like the background image on your homepage.  It has a really
nice, old school library feel to it.

Here is some feedback I hope you find helpful:

When you go to another page, there isn't a "home" button to get back
to the homepage.  I would recommend adding one to the menu.
Your Books, Users, Genres, and Rental all seem to be displaying data
correctly.
It looks like you are probably still working on incorporating all your
queries into your UI, for example, you when you click on a book title
```

or ISBN, it doesn't currently go to the book info, it returns to the homepage. I image this and other things you just haven't gotten to yet, because it looks like you have queries for them in your DMQ.
When I try to add a book on the Books page, I get the following error: {"code":"ER_BAD_NULL_ERROR","errno":1048,"sqlMessage":"Column 'name' cannot be null","sqlState":"23000","index":0,"sql":"INSERT INTO authors (name) VALUES (NULL)"}

It seems like it believes the 'name' of the book is null, however, I tested it with both names "Book" and "Book Name" and got the same error. Something you probably want to look into.

I also got a similar error when trying to add an author, this time with it having issue with the 'isbn' column.

Your delete function to remove a book worked when I tested.
On the users page, the add users functions seems to work. I do recommend putting something above the join date box that shows the date format. It will still add the user if the date is in the wrong format, but it puts the date to all zeros if not formatted correctly.
Or perhaps you could consider having the date auto assigned based on when the form is filled out? That is more of a design choice, though, and really depends on the use case. You may want to add users sometime after they join, so in that case you wouldn't want the date auto assigned. Just depends on the design.

Overall, looks like a good start with implementing the queries into the UI. Good luck on your project.

-Evan

Review feedback by Fedor Titov:

Hi Tyson.

Looking good! Now Evan did a pretty good breakdown already, so I'll add on a couple of notes. Also I figure that a lot of the code for connecting to the database is incomplete, as we're in the same boat.

1. Aside from the fact that form submission leads to the homepage, your books all have hyperlink that also lead back to the homepage.

Maybe this is meant to later hyperlink to an amazon page for the book or something, but it seems hard to combine that with form submission.

2. For rentals, you have fields that reference ids of books/users. My first impression was that this is wrong, but I do suppose it's not so bad. It does run into the problem that you're forcing the person accessing to look up the id numbers ahead of time, but again it's probably fine for a student project.

3. Finally, something our group will also need to deal with: it might be usefull to add a check to make sure the input for your forms is in proper format. For example not accepting text when you need a number. Otherwise the whole site will crash.

Other than that, looks very good at this stage. Good luck at the next step!

Review feedback by Samuel Judkis:

Nice job so far, I also really like the background picture. Very comfortable looking.

I encountered the same errors as the previous reviews when trying to add new books and authors. The delete functions for both books and users worked very nicely. I was able to add a user but there didn't seem to be any restrictions for doing so. A new user entry is created even when no data is entered into the form. the row just comes up black. I'm sure this is just something you haven't gotten to yet though, I know my group had plenty of that. Depending on your design choice, you might want to restrict the join date of a new user to be only the current day or in the past. I was able to create an account with a future join date. It's possible this could be by design though, maybe if you wanted to allow users to preregister or something.

Review feedback by Melody Reebs:

Hi Peng,
Your project is really coming along! Here's some of my feedback:

- **I'm not sure why there's a separate form on the Books page to add a new author to the library? It looks like you can add authors when you add a book, so I'm not sure that the separate form is necessary**

- It looks like you are still working on functionality to add genres to the DB. Once that's set up, I'd recommend just adding a genre field to the form where you add books as opposed to a separate form on the Genres page
- I'd suggest adding something to the Books table that shows whether or not a book is checked out or not
- For the Rentals form, I'd suggest having a dropdown for the User and Book fields. That way, you don't have to look back at the other pages to get the IDs for those.
- It doesn't look like the "Add new genre to the library" or "Add new rental to the library" forms are working
- I didn't see where you are able to return a book to the library
- I didn't see any search functionality. Not sure if you're planning to add this later?

Review feedback by Clarissa Gasiciel:

Hi Peng! The front end design of your site is really cute! Well done!
Some notes(though I know we're all still building and fine-tuning at this stage):
-The nav bar spills out of its container for me. I wonder if setting up some kind of flexbox situation could be helpful in keeping it contained.
-When adding a book, perhaps consider a way to make sure that I am not adding an author that already exists in the database (i.e. use a drop down menu populated with entries from the Authors table where the last entry is an option to add a new author if I can't find the one I need). I would suggest doing the same for genre to prevent duplicates.
-Consider adding the delete button into your table for each book. If a user has to type in the ISBN to delete something, they can easily make a mistake and delete the wrong book.
-I agree with Melody that adding a dropdown for the User ID and Book ID on the Rentals page would be really useful. If you go this route, I would make sure the user's name is listed alongside the id, so there is an easy way to look up a user. The same goes for the book title. A dropdown would also be useful for returning a book.
-Remember to include to search/filter functionality somewhere on your site.

All-in-all, it's shaping up nicely. Keep up the good work!

Actions based on feedback (Step 4)

- Add new author is is to facilitate the addition of more authors, if it is incomplete, of course we can use the update to complete. But we will keep it.
- Now, our database can add new genres in both book and genre page.
- We fixed the display of checked out books in rental page.
- We active the insert features for genres and new rental.
- We add a new feature to return the book.
- New features for update and delete.
- Queries are now based on ID and not ISBN too allow duplicate books, as originally intended.
- TODO: Fix date entry and re-prompt upon invalid input in any field.
- Changed delete genres by genre_id instead of genre_name.

Feedback by peer reviewers (Step 3)

Review feedback by Fedor Titov:

Hi Tyson,

Looking good, keep up the good work.

I do have a suggestion relating to how you can check if a book is currently checked out. I do having the backend implementation help you out here will save you some effort, similarly as was discussed in the Piazza post. Here's how I'd do it. Run a check for the row with the most recent last checkout date, then see if for that entry the check-in date is also filled in. Then have your backend (or you can make this one large SQL query) check the result; if the check-in date is NULL return a message that the book is checked out. Otherwise allow the book to be checked out. Hopefully

Relating to this I'd like to note that currently the way you have your checks for statement for checking out books, the if not exists statement will stop the book from being checked out if it's been ever

checked out before. This would make sense if you don't store past rental records, but from your notes that's likely not what you want.

A small tick is that you don't need the if exists for delete statements, since if the row doesn't exist, it won't delete anything anyway. Also you do have the option of having if statements in your backend language instead of in SQL because from what I can tell SQL isn't the most intuitive when it comes to if clauses.

In regards to your second question, I don't think there's a particularly good way of doing that. The closest thing is using DISTINCT, but that works only on whole rows. Perhaps there are some more advanced functions you can try looking into.

That's it for me. Hopefully at least some of this is helpful. There's just too much stuff to look through this time around.

Review feedback by Evan Linepensel:

Hi Tyson,

Thanks for submitting and thanks for explaining the “ToDo” items in your post. That helped with reviewing. Looking at your two questions, to be honest, I’m not familiar enough on the subject to provide a clear answer to either question.

That said, I like the simplicity of your UI design. I tested the 3 links and they all worked. This is more of a personal preference, but there is a large amount of white space in the bottom left corner of the main page. Perhaps consider moving some things around to have more even whitespace between different forms? Otherwise, UI looks good.

I did encounter an error when I attempted to import your DDQ using PHPMyAdmin. To be honest, I don’t know if it is something I did wrong or not. It did work if I ran each individual query to create each table. Here is what the error said:

“Error

Static analysis:

1 errors were found during analysis.

Unrecognized statement type. (near "Data" at position 0)

SQL query:

Data Definition Queries CREATE TABLE books (id int NOT NULL AUTO_INCREMENT, isbn bigint NOT NULL, title varchar(255) NOT NULL, PRIMARY KEY (id)) ENGINE=InnoDB DEFAULT CHARSET=latin1

MySQL said:

#1064 - You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'Data Definition Queries

CREATE TABLE books (
id int NOT NULL AUTO_' at line 1”

It seemed to have an issue with the syntax with the first table, but when I ran each CREATE TABLE query individually, it worked without error. I just wanted you to be aware of it.

Your DDM queries look good to me. I didn’t see any major issues there. They seemed aligned with your UI design, as well as your Database ERD, Schema, and relationship documentation.

Good work and best of luck!

-Evan

Review feedback by Melody Reebs:

Hi Peng,

Great work! Here are some of my comments:

- For your SELECT query to show books available in the library, I think you may have an issue when joining to the rentals table, since if there are multiple rentals associated with that book, it'll produce duplicate rows. I

know you are still figuring out how to consolidate the authors and genres, so just something additional to think about (DMQ, Line 189)
- When removing a book from library, since you want to delete all rentals associated with that book, you don't need the WHERE constraint for the user (DMQ, Lines 100 & 102)
- What if user rents the same book more than once? How does the return query know which one to update? (DMQ, Line 155-160)
- Can ISBNs start with a 0? If so, does a string data type make sense instead of an int? (DDQ, Line 6)

Review feedback by Clarissa Gasiciel:

Hey Peng!

Your database is shaping up really nicely! Some considerations:

I like the layout of your page, but I also wonder if there is too much going on for a single interface. Dividing certain categories into separate pages might make it more readable/user friendly (i.e. a page to add/delete user, one to add/delete books, authors, and genres, and one to manage rentals). This would also give you the option to display the other books/genres/authors already in the database. I noticed that you used the UNIQUE keyword in your DDQ (good thinking!) for genres, but consider how you might prevent a user from making multiple entries of the same authors or books.

Out of curiosity, what was behind the decision to change the author's name to a single name rather than first and last as separate entries? What will happen if a user searches an author's last name only?

In your DDQ, I noticed that there wasn't a DROP TABLE IF EXISTS [table name] statement before creating your tables. This statement appears in our sample MySQL files. I'm not entirely sure why it seems to be best practice, but perhaps look into it more and consider adding it before creating your tables.

Actions based on feedback (Step 3)

1. From the **Review feedback by Fedor Titov**, a few “if” statements were removed from some queries due to their redundancy. If something does not exists, it will not be deleted, so checking for it only slowed down the query. Some statements were intentionally kept as they are integral to confirm prior to adding, updating, or removing data.
2. From the **Review feedback by Fedor Titov**, the ‘Rent out a book from the library’ query was fixed to check for a rental of the book without a return date (indicating it is still checked out).
3. From the **Review feedback by Evan Linepensel**, no change was required for the data definition queries. The error encountered was a failure on PhpMyAdmin’s part to ignore comments indicated by “--” or “/* … */” and is resolved by uploading the .sql file with character set ANSI while telling phpMyAdmin that the file uses character set UTF-8.
4. From the **Review feedback by Melody Reebs**, the query that displays available books will now show distinct rows that aren’t duplicated as a result of multiple completed rentals in the past.
5. From the **Review feedback by Melody Reebs,** when user removing a book from the library, we only need to get the book id -- fixed DMQ.
6. From the **Review feedback by Melody Reebs,** the return query was fixed to specifically change the rental that has not yet been returned by checking for a NULL date-in. This would have been database-breaking had it not been caught.
7. From the **Review feedback by Melody Reebs,** the ISBNs usually begin with 97…, so we will still use bigint type for that.
8. From the **Review feedback by Clarissa Gasiciel,** in order to show the available books in the library, we added a rental table and book table in the HTML.
9. From the **Review feedback by Evan Linepensel and Clarissa Gasiciel**, the layout of HTML is too much things in a page. We fixed the page into 4 subpages and added tables for features.

Feedback by peer reviewers (Step 2)

Review feedback by Melody Reebs:

Hi Peng, Here are some comments:

It looks like this database is meant to reflect "real-time" borrowing status, as opposed to also keeping track of historic data? If so, keep in mind that when implementing, this will require you to update/replace previous rental entries when those corresponding books are borrowed again. Otherwise, you might want to take the approach of keeping the historic data, as I think implementation would be similar in terms of complexity; you'd essentially just allow books to be associated with multiple rentals, and then create a new rental entity for each new rental of that book

In your relationship outline, you state that "Each book may belong to 1 and only 1 rental." However, based on your ER diagram, it's possible that a book does not belong to a rental, so this should be "at most 1 rental"

You're missing a * next to id field for the users entity in your schema
I really like your secondary view of the schema; I'd suggest adding arrows to show which foreign keys are referencing which primary keys

Review feedback by Clarissa Gasiciel:

Hi Peng! Your proposal is really well put together. Good work! Just some comments/questions:

Outline: I was curious about your decision not to add an id to the genre table. I read your reasoning, and I get what you're saying. Since our class examples all included ids for entities, however, I wonder if having an id integer might be best practice.

Relationships: I am also curious about your decision that an author need not be affiliated with a book to be included in your database. Was there a particular reason you both chose to go that route? Since the intent of the database is to keep track of the books the library has, is there a benefit to listing an author who does not have a book at that library?

I also read in your description that a book does not need to belong to a genre. Your ER diagram, however, shows that a book should belong to at least 1. Make sure that these two things are consistent. I like the idea of requiring a book to belong to at least one genre as books generally can be assigned to at least one.

ER Diagram: This also kind of goes with relationships. It seems that a rental can only contain one book. However, once that book is

returned, can the book be rented again? If so, that might make that relationship one-to-many and change your design and implementation.

Overall, nice job!

Review feedback by Jesus Madrigales

I think your group did a fantastic job. I like your layout. I just have a couple comments.

In the relationships part, you say each book may belong to 0+ genres and each genre may hold 0+ books. I feel like a book should have at least one genre. In fact, in your ER diagram you list say each book may belong to 1+ genres. I suppose you can have genres in your database with zero books associated with it, but I think if it is going to be in the database then there should be at least one book in it.

Your ER diagram says that the relationship between books and rentals is zero to one but the relationship says one and only one.

Schema was great. I liked how you gave a secondary view.

Review feedback by Evan Linepensel

Hi Tyson. Thanks for submitting your Step 2 for review and also thank you for reviewing my submission. I like the idea of a library for your project. I also find it helpful that you indicated what changes you made from Step 1 to Step 2.

The project outline is straightforward conceptually, however, it seemed some of the wording could be reworded to made it a little easier to read. I'd recommend reading the paragraph over and massage the wording so it flows a little better. Not a major issue, though. Still got the point across. For example, this sentence: "This is the database that record users borrow their book in the library, include what the books name, the date of borrowed and returned." It may sound a little better written this way: "This is the database that records users borrowed books from the library, including what the book's name, the date borrowed, and date returned." There may be even better ways to word it, but I hope this gives you an idea.

Your next section, listing the entities and their attributes and constraints, is straightforward and well organized. One thing to note, which I also got feedback from the TA in Step 1 of my project, is that DBs don't have a "string" data type. They have things like

"varchar" and other types that you would use instead. I also like that you left the old Author and Genre attributes, but just crossed out and with a note on why you changed it. That allows the reader to understand your thought process.

The next sections for Relationships and the ERD also look good and make sense to me. The final section for Schema doesn't seem to include a full schema diagram that I believe is require for the project. You have the tables in a diagram, but you would also need arrows to between the different keys used throughout the database. It was, however, useful that you explained the schema with some sentences and showed the tables.

Overall, it looks like a good draft. Best of luck on your project.

Review feedback by Fedor Titov

Hi Tyson.

Your database is looking great! I can't find anything major required to fix, save for maybe using arrows to connect your schema, but that's just to conform more to the stylistics of the class. And yeah, strings should be char/varchar, though that's a quick fix and not really a major problem.

Without anything too big, I'll nitpick a bit. I find the way genres are included to be a bit odd. They only have one column and are only connected to the books table, so it almost seems they should just be another column in books. I can see that you separated out genres because books can fit under several genres at a time, but even if that's the case, most libraries and bookstores usually just pick one genre and assign it to the book, otherwise they'd have a hell of a time sorting books by genre. Even if you wanted to keep multiple genres, you could still implement it as multiple columns in books such as main genre/ secondary genre, etc. If you do want to keep them, may I suggest connecting them to the Users table? After all you did write in the description that it might be helpful to track which users like which genres.

The other minor thing is that when you have a relationship table that connects a many-to-many relation, you don't really need a unique ID, at least from my understanding. Usually the combination of unique ID from the first table combined with the unique ID from the second table is enough. Though there's nothing wrong with keeping them either.

Finally some of the participation seems a bit wonky. I don't know that I'd call an author with zero books an author. I guess the library could list an author even if they don't have any of his books, but that also sounds weird. Same thing for genres, why would you have a genre if you don't own a single in that genre.

That's it. Mostly small stuff that you might not even need to change, I'd say. Good luck on the rest of it!

Review feedback by Samuel Judkis

I like the project idea.

I'm a bit confused by the books/rentals/users relationships though. You say a book is related to at most one rental but what about after a rental is completed? I would expect that the book would then be available again to be associated with another rental but the relationship doesn't allow that. Unless the database only acts as a record of currently active rentals, where a rental is removed once it has been completed? It seems to me like this could be a one to many, where one rental has one book, but a book can be related to many rentals over time. In terms of a functioning library too, it might be interesting to have some sort of quantity attribute attached to the books. That way, a librarian or other person could check if the book is available to rent.

I mentioned something similar in my other review, but it might also be good to have an integer primary key for genres. A similar issue came up in my own project, and my partner explained that storing the integer key can save a lot of memory space compared to storing strings. Many books will have the same genre, so each genre string will be stored many times in the books_genres table. By having an int primary key, you are able to only store the genre name once in the genres table, and use the int id as a reference to it, which saves memory. I'm no expert but my partner explained this and it made sense.

Your schema looks very good, I think it's just missing the arrows connecting the primary and foreign keys.

Actions based on feedback (Step 2)

1. From the **Review feedback by Melody Reebs,** we fixed the relationship between books and rentals in order to show historical data; the relationship change reflects that one book

can have 0 or many rentals so that we can keep track of past rentals.

2. From the **Review feedback by Melody Reebs,** we got the mistakes from schema, we missed a * next to id field for the users. I add change to that.
3. From the **Review feedback by Clarissa Gasiciel,** we didn't add id to genre filed because one genre have only one name. However, feedback from Samuel Judkis convinced a change to be made for the sake of memory, so the genre table now uses an id as a primary key.
4. From the **Review feedback by Clarissa Gasiciel,** we removed author and genres from book entity because each book may belong to multiple authors or genres. And we just represented it in many-to-many relationship tables at the bottom.
5. From the **Review feedback by Clarissa Gasiciel,** we got a mistake from relationship between book and genre. We changed to a book can have one or more genres.
6. From the **Review feedback by Jesus Madrigales,** all the problems are same as above, we have changed it.
7. From the **Review feedback by Evan Linespel**, some of the grammar in the introduction has been fixed to be a bit more clear.
8. From the **Review feedback by Evan Linespel**, mentions of strings were replaced with varchars to accurately portray how some attributes will be represented. This was also mentioned by Fedor Titov.
9. From the **Review feedback by Evan Linespel**, arrows were added to the visual schema to indicate the source of foreign keys. This was also mentioned by Fedor Titov and Samuel Judkis.
10. From the **Review feedback by Evan Linespel**, no change will be made to the book and genre relationship. A book will still be able to have multiple genres, such as sci-fi and horror.
11. From the **Review feedback by Evan Linespel**, the only use of a unique id in a relationship table is in the case of the rentals entity. The id is required as the same user may borrow the same book again in the future, and the library will now support historical data. No change is made.
12. From the **Review feedback by Evan Linespel**, no change will be made to the inclusion of authors without books. This is to give feedback to a user that may browse the library searching for a particular author. If the library recognizes an author and shows no books available, the user has a better confirmation rather than assuming a database error or misspelling. Additionally, no change will be made to genres as there is no negative impact to

the database for a genre to not have a relationship with any books.

13. From the **Review feedback by Samuel Judkis**, the relationship between books and rentals has been overhauled. Now, a book may belong to multiple rentals, but careful implementation will be made so that a rental cannot be made if the book belongs to a rental without a return date. The outline, ERD, and schema have been updated to reflect this.
14. From the **Review feedback by Samuel Judkis**, quantity may be of some value, but is obtainable through various queries. For the moment, no change will be made unless it proves to be more efficient to constantly update a quantity available in the future.
15. From the **Review feedback by Samuel Judkis**, the genre table will now include a unique id to save memory in the books_genres relationship table.

Upgrades to the draft version

Small changes are shown throughout the document. Otherwise, the biggest changes that were influenced by the peer reviews are as follows:

1. Books can now belong to many rentals so that historical data can be produced and kept. To ensure a book is not rented out if it is unavailable, restrictions will be made such that a rental cannot be created if the book_id belongs to a rental without a return_date.
2. Genres now have an id to save memory in the books_genres table.
3. The ERD and Schema visuals have been updated to reflect changes, and have a new look to them.
4. Tables books_authors and books_genres have had book_id attributes replaced with a non-foreign-key isbn. The attribute id has been removed and the primary key is the remaining two attributes.

Project Outline

This database will represent a library. It will hold users, books, and other entities to track various rentals. It will be capable of storing historical data on rentals that can indicate time of borrow and return, and place restrictions to ensure a book cannot be rented out if it is unavailable.

Entities

- Books: each book that the library has in its possession or is rented out.
    - ID (required): the primary key, as some books could have the same name, author, or both. It is an integer.
    - ISBN (required): the serial number of the book. Useful for looking up the specific edition. It is an integer.
    - Title (required): the name of the book. It is a varchar.
    - ~~Author (required): the author of the book. It is a string.~~
    - ~~Genre (optional): the genre(s) that the book belongs to. It is one or more strings.~~
- Authors: the authors who may have some books in the library.
    - ID (required): the primary key, as some authors may have the same name. It is an integer.
    - ~~First Name (required): the author's first name. It is a varchar.~~
    - ~~Last Name (required): the author's last name. It is a varchar.~~
    - Name (required): the author's name. It is a varchar.
- Genres: the genres to which a book may belong to. Useful for users with a preference to which the library or user may search.
    - ID (required): the primary key, as relationships involving genre can save memory with its use.
    - Name (required): the name of the genre. It is a unique varchar.
- Users: the people who use the library, who may also be referred to as the borrowers.
    - ID (required): the primary key, as any number of other attributes could be the same. It is an integer.
    - First Name (required): the user's first name. It is a varchar.
    - Last Name (required): the user's last name. It is a varchar.
    - Date Joined (required): the date in which the user has joined the library. May be useful in later implementation

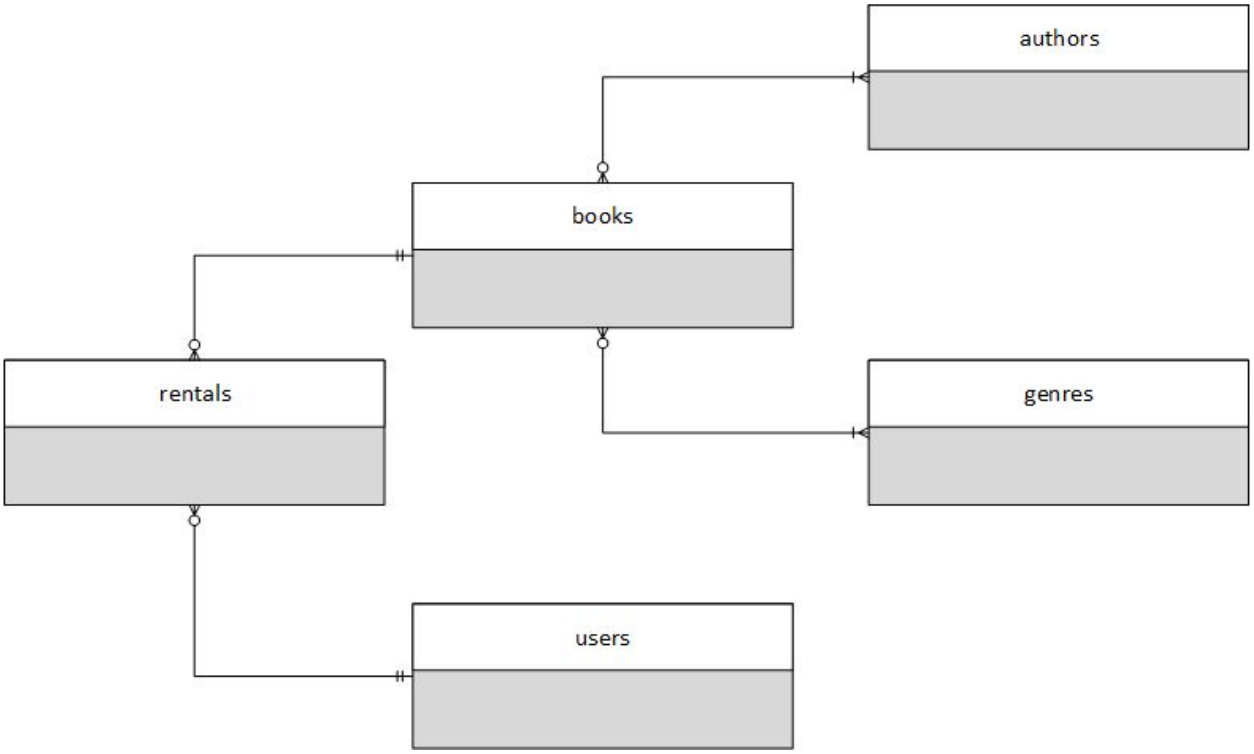
for other attributes, such as maximum number of rentals or borrowing period. It is a date.

- Rentals: the relationship of books and users, the position of books should be available and lent.
    - ID (required): the primary key, every rental ID should be different. It is an integer.
    - Book ID (required): the primary key, as some books could have the same name, author, or both. It is an integer and another entity.
    - User ID (required): the primary key, as some authors may have the same name. It is an integer and another entity.
    - Date Checked Out (required): the date that user borrowed. It should be a date as a data record in database.
    - Date Returned (optional): the date that user returned their books, it should be a date as a data record in database.

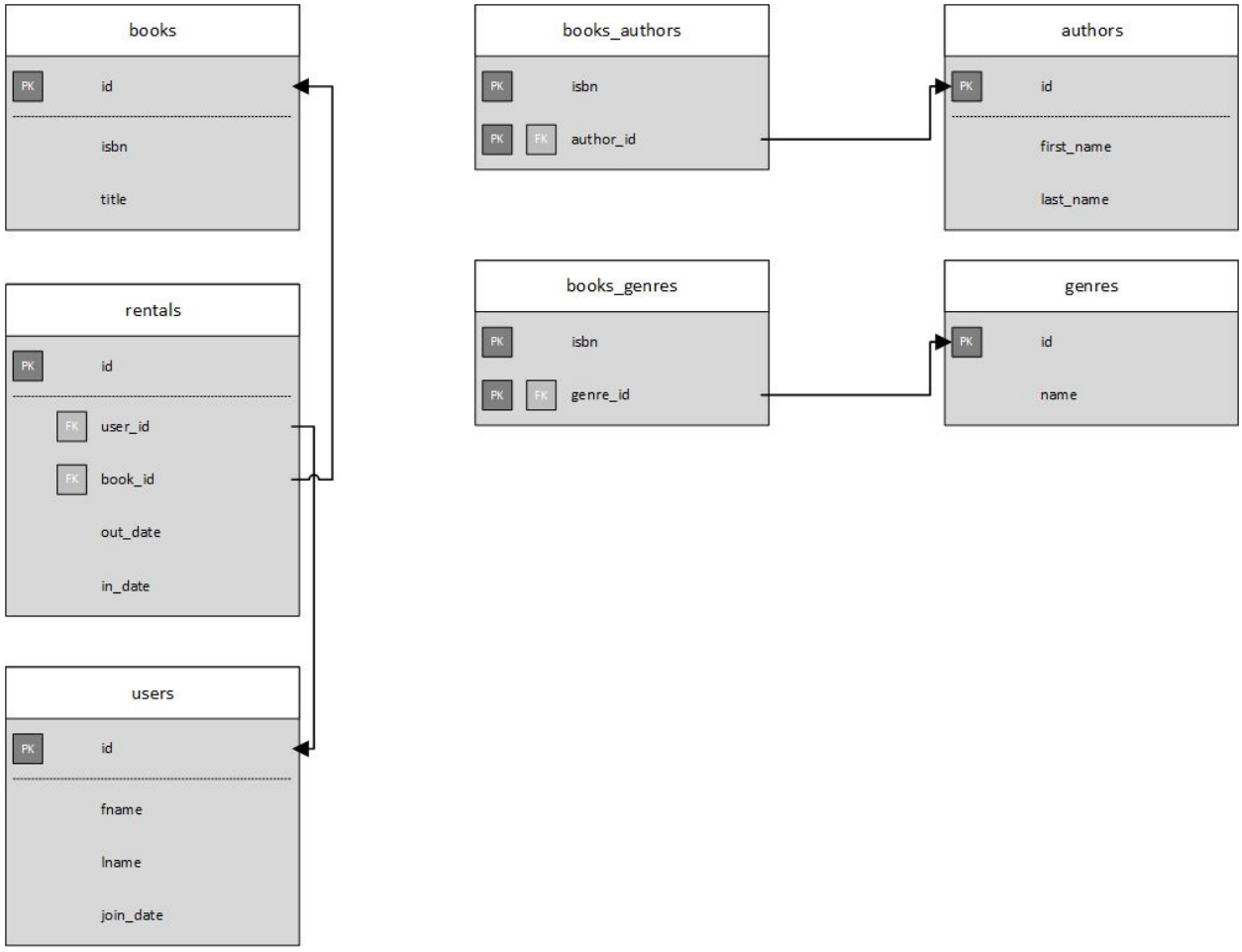
Relationships

- Books and Authors (many-to-many)
    - Each book may belong to 1+ authors.
    - Each author may hold 0+ books.
    - Books >0---|< Authors
- Books and Genres (many-to-many)
    - Each book may belong to 1+ genres.
    - Each genre may hold 0+ books.
    - Books >0---|< Genres
- Books and Rentals (one-to-many)
    - Each book may belong to any number of rentals. (Changed from one-to-one relationship to allow historical data.)
    - Each rental may hold 1 and only 1 book.
    - Books ||---0< Rentals
- Users and Rentals (one-to-many)
    - Each user may own 0+ rentals.
    - Each rental may belong to 1 and only 1 user.
    - Users ||---0< Rentals

Entity Relationship Diagram



Schema



Legend	
*	Primary key
[%]	Indicates attribute is a foreign key (examples: [b], [a], [u])
(0)	Optional field

This legend applies to the tables on the following pages.

books		
id *	isbn	title

Each book may belong to multiple authors or genres, so there is no column for either entity. Instead, relationship tables will be used.

authors		
id *	first_name	last_name

Authors may have the same first and last name, requiring an id as a primary key.

genres	
id *	name

Though no two genres will share the same name, an incrementing integer will save memory in the books_genre relationship table.

users			
id *	first_name	last_name	join_date

Another many-to-many relationship will present itself in the form of rentals, so users will not contain columns relating to their checked out items.

rentals				
id *	user_id [u]	book_id [b]	out_date	in_date (0)

The most important relationship table, equivalent to books_users. A user may rent the same book multiple times, so a rental id is required as a primary key.

books_authors	
isbn *	author_id * [a]

A many-to-many relationship table that isn’t quite that, preventing multiple entries that hold the same information while allowing smoother queries.

books_genres	
isbn *	genre_name * [g]

Another many-to-many relationship table that isn’t quite that.