Project Step 4 Draft

Project Title: Beaverton Community Library

Team Members:

- Anil Verman
- Alexander Lott

Website URL: https://web.engr.oregonstate.edu/~vermanan/

FEEDBACK FOR STEP 3 DRAFT:

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

    - Yes, there are select queries for every schema

- *Does the UI implement an INSERT form for at least one table in the schema?* In other words, there should be UI input fields that correspond to at least one table.
    - No inserts are found on the website
- *Does the UI have at least one DELETE for any one entity?* In other words, is there a form/link/button that will allow the deletion of a row in at least one table?
    - Yes the UI provides DELETE function for all entities
- *Does the UI have at least one DELETE that will remove things from a M:M relationship?* In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.
    - There are deletes for checkouts but no edit or delete for book genre relationship
- *Is there at least one UPDATE form in the UI for any one entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?
    - All entities have edit(Update function)
- *Is there at least one UPDATE form in the UI to modify an M:M relationship?* In other words, does the UPDATE allow the user to select a different foreign key value to update the intersection table with?
    - Yes, edits for checkouts are found however, book genre relationship cant be modified
- *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.*
    - The navagation tab might be better if it is fixed sized.

The above is all my work."

- INSERT forms have been added to the website in response to this feedback.

Blaise Skoletsky: "*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 can see a place for the select, but data is not yet populated. This needs to be implemented still.

- *Does the UI implement an INSERT form for at least one table in the schema?* In other words, there should be UI input fields that correspond to at least one table.
    - I do not see any spot for inserts in the table.
- *Does the UI have at least one DELETE for any one entity?* In other words, is there a form/link/button that will allow the deletion of a row in at least one table?
    - There is a delete section on each table, but it does not yet work.
- *Does the UI have at least one DELETE that will remove things from a M:M relationship?* In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.
    - In the checkout table, there is a delete button.
- *Is there at least one UPDATE form in the UI for any one entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?
    - Yes, there is an update button, but it does not work yet. This is on each row.
- *Is there at least one UPDATE form in the UI to modify an M:M relationship?* In other words, does the UPDATE allow the user to select a different foreign key value to update the intersection table with?
    - Yes, there is an update button on the M:M table.
- *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 like the UI, maybe add some more descriptive comments on the header for what the project is used for.

This is all my work and I did not use AI tools to formulate any of this feedback."

- UI column names were replaced with more readables aliases in response to this feedback.

Cecilia Bui: "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.

    - There is a table header for each table in the schema, but there is no data populated in the tables. Looking at the DML.sql file, the SELECT/read queries are not entirely fleshed out yet with JOIN operations to properly display all relevant attributes. However, it does seem like the team is already thinking about it, as some attributes' parent tables are aliased. Relatedly, the queries are not all syntactically correct since JOINs are not all in place (for example, the READ query under the Genres section).

Does the UI implement an INSERT form for at least one table in the schema? In other words, there should be UI input fields that correspond to at least one table.

    - There are no INSERT forms for any of the tables.

Does the UI have at least one DELETE for any one entity? In other words, is there a form/link/button that will allow the deletion of a row in at least one table?

    - There are DELETE headers on each table but no formal link/button.

Does the UI have at least one DELETE that will remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.
    - There are DELETE headers but no formal link/button on the intersection tables as well (like Reviews or Checkouts pages).

Is there at least one UPDATE form in the UI for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?
    - There is an "Edit" header where UPDATE form/links may go, but there is no visible UPDATE form yet for any entities.

Is there at least one UPDATE form in the UI to modify an M:M relationship? In other words, does the UPDATE allow the user to select a different foreign key value to update the intersection table with?
    - There are no UPDATE forms that would modify M:M relationships (like on Reviews or Checkouts). However, there is a header on the table that indicates it will be implemented.

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 appreciate the smoothed borders on the tables and the color palette. However, it does seem like the size of the tables affect the size of the header and navigation bar. I would recommend making the header and navigation bar either static or at least independent of the table sizes. To fit with the smooth theme, the links in the navigation bar could be updated to be buttons or without underlines. The tables do have aptly aliased column names, but this seems to be hard-coded rather than part of the "AS" aliasing function in SELECT queries.

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

- Improved the SELECT/Read queries (added more joins, aliases for columns, etc.) in the DML in response to this feedback.

Sean Miller: "*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.
    - There is a table layout for each of their entities where data will eventually be displayed via SELECT queries. The Author and Genre tables seem to be fairly lacking in attributes, though there may not be a need to display more than a name for each of them.

- *Does the UI implement an INSERT form for at least one table in the schema?* In other words, there should be UI input fields that correspond to at least one table.
    - There is not any place in these tables to INSERT data.
- *Does the UI have at least one DELETE for any one entity?* In other words, is there a form/link/button that will allow the deletion of a row in at least one table?
    - There is a delete cell on each of the tables. Personally, I would recommend moving the 'Edit' and 'Delete' instances onto the rows themselves as this is the data that will eventually be edited or deleted.
- *Does the UI have at least one DELETE that will remove things from a M:M relationship?* In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

    - I'm guessing Checkouts is a M:M relationship and on that table there is an instance of Delete.
- *Is there at least one UPDATE form in the UI for any one entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?
    - There are cells that say 'Edit' on each of the tables.
- *Is there at least one UPDATE form in the UI to modify an M:M relationship?* In other words, does the UPDATE allow the user to select a different foreign key value to update the intersection table with?
    - There are cells that say 'Edit' on each of the tables, there is not an additional form where one could go to update specific data yet.
- *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 personally recommend that some sample data is placed into your program in order to more accurately display how data will eventually be represented once it is hooked up to the database. Naming conventions and data displayed seems good though!
- As a reviewer, clearly describe to what extent your feedback to the team was original (e.g. "all my work") or non-original (e.g. used AI tools per Code Citation Tips).
    - This is all my work and I did not use AI tools to formulate any of this feedback."

- UI was modified to display sample data in response to this feedback.

Strato Bayitaa: "Good progress so far. Keep up the great work. Please pay attention to the rubric going forward. You lost some points because: 1. File was not properly named as required: **ProjectGroupxx_Step3_DRAFT.zip** 2. Assignment was submitted late."

- Zip file name was changed to the proper format for Stage 4 of the project in response to this feedback.

FEEDBACK FOR STEP 2 DRAFT:

Strato Bayitaa: "Excellent progress! Keep up the great work."

- No changes were implemented based on this feedback, as it did not contain any recommendations or questions.

Tyler Renn: "**Does the schema follow the ERD and outline?**
Yeah, the SQL schema matches the ERD and outline exactly, all the entities, attributes, and relationships line up as expected.

- **Is the naming consistent?**
  Mostly yes, entities are plural, attributes are singular, and the naming style is consistent across the board. I did notice the capitalization of table names in the outline, but the ERD and .sql file have tables names in lowercase.

- **Is the schema easy to read?**
  The schema diagram is easy to read. It's well-organized and easy to follow, with no confusing lines or overlap. Very linear.
- **Are intersection tables properly formed?**
  Yep, all M:N relationships like `Checkouts`, `Reviews`, and `Books_has_Genres` are correctly set up with two foreign keys each.
- **Any non-normalized issues in the sample data?**
  Nope, everything looks normalized, no repeating groups or partial dependencies.
- **Is the SQL syntactically correct?**
  Yes, the SQL runs fine in phpMyAdmin with no syntax errors, tables create and data inserts as expected.
- **Are data types appropriate for each attribute?**
  For sure, text fields use `VARCHAR`, dates use `DATE`, phone numbers are stored as strings, and so on.
- **Are PKs and FKs correctly defined with CASCADEs?**
  Primary and foreign keys are properly defined, though `ON DELETE` and `ON UPDATE` are set to `NO ACTION` could consider adding `CASCADE` if deletions need to propagate.
- **Are relationship tables present in the SQL?**
  Yep, all the relationship tables mentioned in the ERD are present and implemented correctly in the SQL.
- **Is all example data inserted into the SQL?**
  Yes, the INSERTs in the SQL match exactly with the example data shown in the PDF.
- **Is the SQL well-structured and commented?**
  The SQL is clean and readable, though it looks like it was exported from phpMyAdmin, so it's clear and organized.

Well done!

This is my own work"

- Table names in the schema and SQL file have been capitalized in response to this feedback.

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

Yes – The ER diagram and schema diagram match with the one created by the DDL file.

- 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 the naming appears to be consist. Entities are named with plural and their attributes are singular. Camel case is used when necessary and ID is at the end for each ID. They also consistently use the same datatypes for similar attributes.

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

The ER Diagram is easy to read – no crossed lines and all attributes are easily read.

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

The intersection table “books_has_genres” has 2 foreign keys: bookID and genreID which form the primary key for the table. In the description for “Reviews” and “Checkouts”, these two tables are also described as intersection tables. They both contain at least 2 foreign keys : memberID and bookID for both. They each also have their own seperate primary keys: reviewID and checkoutID.

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

I don’t know if it could become problematic. If a member were to checkout multiple books at one time (assuming this is allowed) - would there need to be a seperate checkoutID generated for each one?

- 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 ran without any errors in my CS 340 database.

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

The datatypes are appropriate and consistent. They used varchar for name, email, title, etc. Which consistent amounts for attributes that several entities had. Int for all the ID and smallint for values that should be boolean or singular digits.

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

Author defines primary and foreign keys on lines 189-235. The operations are to be declared correctly.

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

The relationship tables: checkouts, reviews and books_has_genres were all declared and populated. Additional their cascade operation were declared as well.

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

Yes all data is inserted – checkouts data is copied twice in pdf.

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

The DDL is well structured and includes comments before adding tables or inserting data into tables. Comments also indicate when primary/foreign keys are being declared and constraints on tables are being created.

All work above is my own."

- After careful consideration, no changes were made in response to this feedback. The reviewer raised a pertinent question about whether or not multiple books being checked out together could be made to share a checkoutID, but our current plan is for the checkoutID of each book in that multitude to remain unique, as the alternative would likely make things trickier to manage by necessitating an additional intersection table. If the need to investigate books that were checked out together arises, they could be feasibly found via their checkout dates/timestamps, instead.

Zain Ali: "Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?
    - I noticed that the tables in the diagram are all in lowercase, but in the outline, I saw that the first letter is in uppercase.
    - I also did not see the table Books_has_Genres in 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?
    - The diagram has all tables in lowercase letters, but in the outline, the first character is uppercase.
- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

    - Overall, the Schema Diagram was easy to read, but the image was a little blurry; Since I zoomed in, it was easy to read.
- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?
    - Yes, it seems that the intersections are formed properly.
- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?
    - The sample data did not suggest any non-normalized issues.
- 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!)
    - After running the SQL file, there were no syntactical errors. Everything seems to work correctly.
- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?
    - The data types are appropriate considering the description of the attributes in the database outline
- 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, but I did notice that the primary keys are not auto_increment in the SQL file, but they are in the outline.
- In the SQL, are relationship tables present when compared to the ERD/Schema?
    - Yes, the relationship tables are present.
- In the SQL, is all example data shown in the PDF INSERTED?
    - Yes, all the example data is shown in the PDF, but the data is just screenshots of the queries rather than typed out in a table.
- Is the SQL well-structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?
    - Yes, the query is well-structured and commented.

All of this work is original.”

- An explicit entry for the Books_has_Genres intersection table has been added to the database outline in response to this feedback.

Jared Staiert: “Hi group 53!

- Your draft looks awesome and your database setup .sql file worked without issue on my end. I don't really have much to add, other than resolving the disparity between table case in the outline /ERD and schema / .sql file.

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

- The schema matches what's in the "cs340_vermanan.sql" file. A quick scan shows all tables with correct fields and proper primary / foreign key creation.

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 is consistency in the naming scheme, but as my classmates have already mentioned: The Outline and ERD both have the table names capitalized (as Books) while the schema diagram and .sql file have the tables in lowercase (as books).
- I suspect that the word processor just capitalized the first word in the outline (happened to me, was annoying!), and the same with whatever software was used for the ERD. Probably easier to just change the .sql file and regenerate the schema.

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

- The schema is reasy to read, though in the PDF it's (in my opinion) a little small.

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

- The intersection table "books_has_genres" is correctly formed, with two FK's and facilitates M:N relationship.

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

- The sample data doesn't seem to suggest an normalization issue to me,.

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 file is syntactically correct, I was able to execute the file without issues on my end.

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

- The data types are appropriate and match the description / schema / outline.

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

- The primary and foreign keys are properly defined in alter table statements after insert. However, CASCADE on UPDATE and DELETE are both set to (I presume) default value NO ACTION.

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

- Relationship tables are present vis-a-vis the ERD/Schema.

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

- All examples shown in PDF are included.

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

- The sql is well-structured and commented, though it appears to be a dump from phpMyAdmin. It is well readable, though as a matter of personal preference I like to declare primary and foreign keys before inserting data.

All of the comments above are my own.

Hope you have a great rest of the quarter!

Jared Staiert"

- The Books, Books_has_Genres, and Reviews tables have had CASCADE statements added to their constraints in response to this feedback.

FEEDBACK FOR STEP 1 DRAFT:

Strato Bayitaa: "Great work, Group 53! Your submission is commendable. However, you lost some points due to the absence of key numerical details. For instance, it would strengthen your analysis to include estimates such as the number of books loaned out weekly or monthly, as well as an approximation of the library's total book collection. I encourage you to incorporate such data in future submissions for a more complete and compelling presentation. Keep it up!"

- Numerical estimates for the number of total patrons, their monthly book checkout rate, the number of books in the library, the number of genres, and the number of authors have been added to the overview in response to this feedback.

Lauren Gliane: "Hi Anil and Alexander, Your project overview gave an insightful narrative to the situation your database will be used for. The ERD is organized and is easy to understand. Just wondering why all relationships between your entities are optional. Besides that, your entities and attributes are well named and the relationships between them are logical. Good job!"

- The Genres-Books and Authors-Books relationships have been made mandatory in response to this feedback.

Kyle Davis: "Great job with this, here is my review:

- Your overview describes the problem at hand, which is that the Beaverton Library is experiencing a surge of new patrons because of new improvements thanks to unexpected funding.
- To align with the Project 1 Draft requirements, I think it would help to provide actual figures, such as how many new patrons are frequenting the library as opposed to pre-expansion.
- At least 4 entities are described, and include: Members, Books, Genres, Authors.
- The outline of entities does correctly describe the purpose of each and includes attribute data types and constraints.
- 1:M relationships seem to be correctly formulated, and there are two intersections present that properly conveys an M:M relationship.
- The biggest room for improvement can be found in the consistency of your naming conventions. For example, some of your attributes are correctly defined in PascalCase, however, there are several instances where you revert to snake_case (see Books Genre_idgenre AND authors_idauthor). Furthermore, I think convention usually suggests that ID is at the end of the attribute name, not the beginning."

- Attribute naming scheme has been changed to camelCase in response to this feedback.
- Additionally, all names of the form "id[stuff]" have been changed to the form "[stuff]ID".

Julie Kim: "Hello Anil Verman and Alexander Lott, Hello Anil Verman and Alexander Lott,

Your overview describes the problem. The Beaverton Library recently is experiencing an unexpected surge of library patrons due to recent improvements and expansions, and its pen-and-paper tracking system needs to be replaced and updated to meet the needs. Through the new database driven website, the library's operations will be easily managed, recorded and provide better service to Beaverton Library's patrons.

The overview lists specific facts:

• Hundreds of unique library patrons

• Thousands of books available

• Multiple books can be checked out per member simultaneously

• Each book can only be checked out by one member at a time

• Books are organized by genre (one genre per book)

• System tracks checkout status, due dates, and member reviews

• Reviews use a 1-5 rating scale

There are four entities described: Members, Books, Genres, Authors. Also, there are two intersection tables: Checkout, Reviews. Each entity represents a single idea and can be stored as an individual list.

Most entity details describe their purpose and list attribute datatypes correctly. However, constraints for each entity are missing.

One suggestion for the attributes is the phone in Members. Instead of integer, it should be handled as string type using Varchar since being an integer can easily result in errors (some people prefer to put phone numbers with '-', some numbers start with digit 0).

1:M relationships are correctly formulated. There are two 1:M relationships defined: genre – books, and Authors - books.

There are also two M:M relationships: Checkout, Review. One suggestion I have is to add more description about these relationships. ERD presents a logical view of the database.

One question I have is, can books have multiple genres? Like mystery/romance, history/mystery. Then there will also be M:M relationship. Same for books-authors since books can have multiple authors (co-authored books, anthologies, etc.). The current design limits each book to exactly one genre and one author, which might be limiting for real-world library operations, though I understand this keeps the design simpler.

Overall, naming conventions (plurals, capitalizations, simplicity) should be improved. For example, authors_idauthor should be simplified to Author_id and Genre_idgenre should be Genre_id for consistency and clarity

Most entities are in plural except Checkout.

Otherwise, great work!"

- The Genres-Books relationship is now M:M as a result of this feedback.

Evan Fiddler: "Hi Anil and Alexander,

Good idea overall and I appreciated the detailed overview!

I only have a few small-ish thoughts that are more *nits* than anything.

Naming

The pattern that you used for foreign keys seems to be <related table>_<related id>. This isn't inherently a bad way to name things, but a few things:

- Your tables are plural but your FK to the genres table on the books table is inconsistent:
    - CURRENT: genre_idgenre
    - CORRECTED: genres_idgenre
    - This is the only actual *mistake* I see in the naming
- If you maintain the lowercase styling, consider introducing an underscore into all of your attribute names when multi-word:
    - CURRENT: idbook
    - OPTION 1: book_id <-easier to read
    - OPTION 2: bookID <- would require more changes to the entire system but is also easier to read
    - WHY?: This allows you to be more concise when using these as foreign keys (instead of books_idbook you could simply have book_id and we all know what it's pointed at)

~~My only other thought has to do with the reviews entity and the checkout entity. Consider that a member and book are both connected (M:N) over these through tables but that neither are connected. Therefore, there's never an explicit relationship ship between a review and checkout. Maybe this is desired, but also consider that a patron may want to re-review the book after the second or third time they check it out.~~

Feel free to read what I was starting to write above (which is now crossed out) and maybe you can see the thought-process I had. But as I was writing, I began to prefer the way you built the review entity and the checkoutentity instead.

Lastly, consider expanding the books entity to contain more attributes like year, ISBN, publisher (which could be a FK to a Publishers entity), page_count, etc. Check out Amazon or bn.com to see what other values are listed on the details section of a book.

Otherwise, your overview addresses a real issue with facts, you've described 6 entities in-total to solve that problem, and your relationship details are correctly formulated (including the correct usage of M:M on the intersection tables).

Overall, good work, it's laid out well and I can see how it could expand over time without much issue."

- "Year" and "ISBN" attributes added to the Books entity in response to this feedback.

Overview:

Beaverton Community Library is a relatively small, yet beloved public library situated within the quaint town of Beaverton, OR. Due to an unexpected clerical error, funding earmarked for the expansion of a much more prosperous library in the state capital was serendipitously delivered to Beaverton Community Library instead, allowing it to finally expand its book catalogue and renovate its computer systems as the populace of Beaverton demanded. However, these much-anticipated updates to the library's facilities have caused a surge of new patrons to materialize, to the point that the library's pen-and-paper member tracking system can no longer track them all. In order to avoid losing track of its members' activities and its books, and in order to ensure that Beaverton Community Library's tracking system could be scaled up to meet the needs of its ever-expanding clientele, the local government ordered the development of a database through which the library's operations may be easily managed and recorded.

The database setup outlined below is intended to be a relatively simple, yet robust one capable of tracking the library's hundreds of unique patrons (roughly 350 patrons in total who collectively check out about 2,200 books per month) and the almost 8,000 books that the library has available for them to read. Each member can have multiple books checked out at the same time, but each book can only be checked out by one member at a time. Additionally, since the library's books are organized by genre (the library carries ~20 different genres), each book must be assigned at least one genre by the database; each book also has a corresponding author (the library carries works from about 6,000 different authors) who is tracked by the database in a separate table. The database also has dedicated tables for tracking both the checkout status (i.e. which members checked out which books, the books' due dates, their checkout dates, etc.) of the library's books and the reviews (i.e. ratings on a scale of 1 to 5) that the library's members have submitted for them, allowing both librarians and patrons to get a sense of which books are garnering the most traffic.

Database Outline:

- **Members:** records the details of Members of this library
    - memberID: int, auto_increment, not NULL, PK
    - name: varchar, not NULL
    - email: varchar, not NULL
    - phone: varchar, not NULL
    - Relationships:
        - M:M relationship between Members (optional) and Books (optional) is implemented with the intersection table Reviews
        - M:M relationship between Members (optional) and Books (optional) is implemented with the intersection table Checkouts

- **Books:** records the details of Books at this library
  - bookID: int, auto_increment, not NULL, PK
  - title: varchar, not NULL
  - authorID: int, not NULL
  - year: int, not NULL
  - isbn: varchar, not NULL
  - Relationships:
    - M:M relationship between Books (optional) and Members (optional) is implemented with the intersection table Reviews
    - M:M relationship between Books (optional) and Members (optional) is implemented with the intersection table Checkouts
    - M:M relationship between Books (mandatory) and Genres (optional) is implemented with the intersection table Books_has_Genres (a book can have multiple genres)
    - M:1 relationship between Books (mandatory) and Authors (mandatory, multiple books can be of the same author)

- **Genres:** lists all Genres of books
  - genreID: int, auto_increment, not NULL, PK
  - description: varchar, not NULL
  - Relationships:
    - M:M relationship between Books (mandatory) and Genres (optional) is implemented with the intersection table Books_has_Genres (a book can have multiple genres)

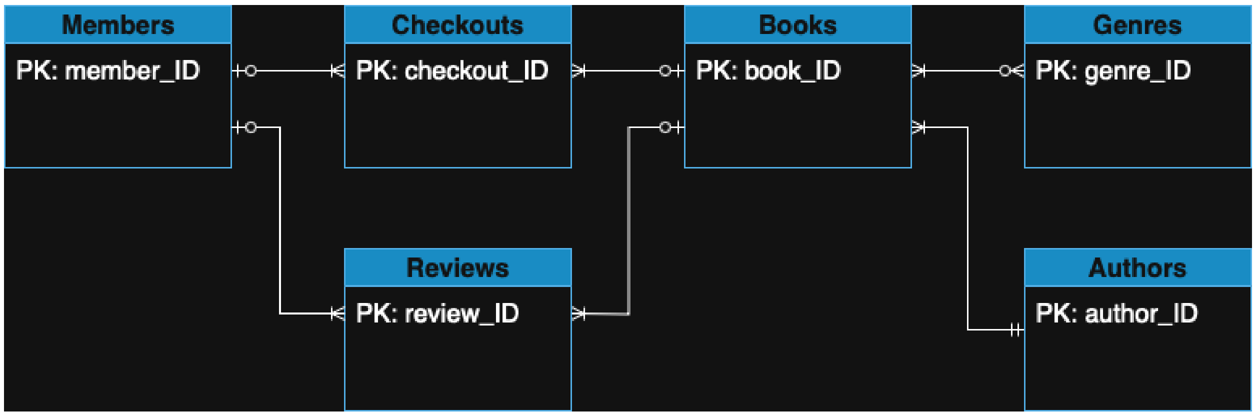
- **Books_has_Genres:** lists which books are associated with which genres, and vice versa
  - bookID: int, auto_increment, not NULL
  - genreID: int, auto_increment, not NULL
  - Relationships:
    - Mediates an M:M relationship between Books (mandatory) and Genres (optional)

- **Authors:** lists all Authors of books
  - authorID: int, auto_increment, not NULL, PK
  - name: varchar, not NULL
  - Relationships:
    - 1:M relationship between Authors (mandatory) and Books (mandatory, multiple books can be of the same author)

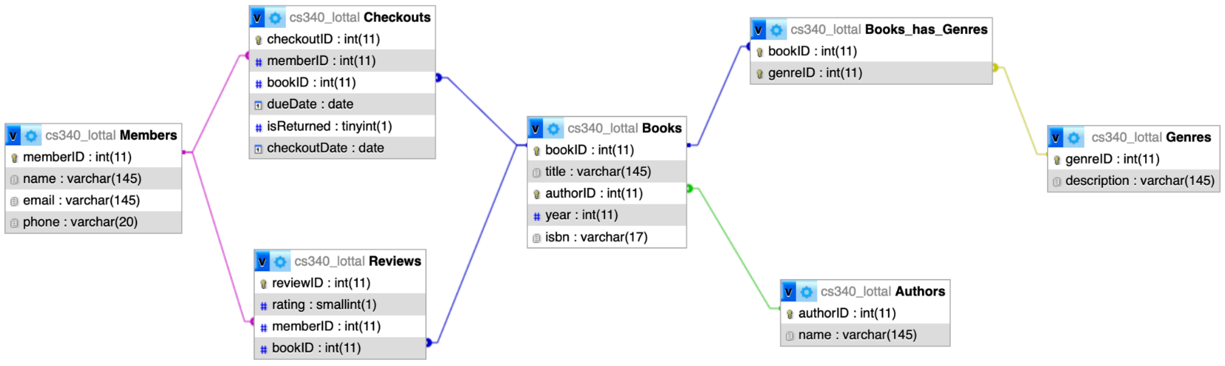
- **Checkouts:** records which member checked out which book. Also records the check-out date, due date, and whether or not the book has been returned.
    - checkoutID: int, auto_increment, not NULL, PK
    - memberID: int, not NULL
    - bookID: int, not NULL
    - dueDate: date, not NULL
    - isReturned: tinyint, not NULL
    - checkoutDate: date, not NULL
    - Relationships:
        - This is an intersection table between Books and Members
        - M:1 relationship between Checkouts (mandatory) and Members (optional)
        - M:1 relationship between Checkouts (mandatory) and Books (optional)

- **Reviews:** records which members rated which book. The rating is a number between 1 and 5.
    - reviewID: int, auto_increment, not NULL, PK
    - rating: smallint, not NULL
    - memberID: int, not NULL
    - bookID: int, not NULL
    - Relationships:
        - This is an intersection table between Books and Members
        - M:1 relationship between Reviews (mandatory) and Members (optional)
        - M:1 relationship between Reviews (mandatory) and Books (optional)

Entity-Relationship Diagram:



Schema Diagram:



Example Data:

```
INSERT INTO `authors` (`authorID`, `name`) VALUES
(1, 'George Orwell'),
(2, 'Andrew Hunt'),
(3, 'J.R.R. Tolkien');
```

```
INSERT INTO `books` (`bookID`, `title`, `authorID`, `year`, `isbn`) VALUES
(1, '1984', 1, 1949, '9780451524935'),
(2, 'The Pragmatic Programmer', 2, 1999, '9780201616224'),
(3, 'The Hobbit', 3, 1937, '9780547928227');
```

```
INSERT INTO `books_has_genres` (`bookID`, `genreID`) VALUES
(1, 1),
(2, 2),
(3, 3);
```

```
INSERT INTO `checkouts` (`checkoutID`, `memberID`, `bookID`, `dueDate`,
`isReturned`, `checkoutDate`) VALUES
(1, 1, 1, '2025-08-01', 0, '2025-07-15'),
(2, 2, 2, '2025-08-05', 1, '2025-07-10'),
(3, 3, 3, '2025-08-10', 0, '2025-07-20');
```

```
INSERT INTO `checkouts` (`checkoutID`, `memberID`, `bookID`, `dueDate`,
`isReturned`, `checkoutDate`) VALUES
(1, 1, 1, '2025-08-01', 0, '2025-07-15'),
(2, 2, 2, '2025-08-05', 1, '2025-07-10'),
(3, 3, 3, '2025-08-10', 0, '2025-07-20');
```

```sql
INSERT INTO `genres` (`genreID`, `description`) VALUES
(1, 'Dystopian'),
(2, 'Software Development'),
(3, 'Fantasy');
```

```sql
INSERT INTO `members` (`memberID`, `name`, `email`, `phone`) VALUES
(1, 'Alice Smith', 'alice@example.com', '123-456-7890'),
(2, 'Bob Johnson', 'bob@example.com', '987-654-3210'),
(3, 'Carol Davis', 'carol@example.com', '555-123-4567');
```

```sql
INSERT INTO `reviews` (`reviewID`, `rating`, `memberID`, `bookID`) VALUES
(1, 5, 1, 1),
(2, 4, 2, 2),
(3, 5, 3, 3);
```

Citations:

- Used draw.io to create the Entity-Relationship diagram
- Used phpMyAdmin to create the schema diagram