

## Reader Log: “Silicon Shelf”

Joshua Sears, Nick Critchfield

PHASE: CS 340: PROJECT STEP 5 DRAFT VERSION (GROUP, ON ED DISCUSSIONS)

DUE THURSDAY, 6/1/23

SUBMITTAL: .ZIP TO ED DISCUSSION, LINK POSTED TO GROUP CANVAS PAGE.

PROJECT URL: [HTTP://FLIP3.ENGR.OREGONSTATE.EDU:9036/INDEX.HTML](http://flip3.engr.oregonstate.edu:9036/index.html)

## Contents

Reader Log: “Silicon Shelf” .....	1
Overview .....	2
Database Outline.....	2
❖ Readers (Object Entity) .....	2
❖ Books (Object Entity).....	2
❖ ReadingClubs (Object Entity) .....	2
❖ ReadingLogs (Transaction Entity) .....	2
❖ ClubMembers (Composite Entity) .....	3
❖ ReadingStatus (Category Entity).....	3
Entity-Relationship Diagram (ERD).....	4
Schema.....	5
Example Data Inserted for Step 3: .....	6
Step 4 Feedback .....	8
Actions Taken/Updates from Step 4: .....	8
Step 3 Feedback .....	9
Actions Taken from Step 3 Feedback: .....	11
Step 2 Feedback .....	12
Actions Taken from Step 2 Feedback Step .....	13
Step 1 Feedback .....	14
Actions Taken from Step 1 Feedback Step .....	15
Updates from Step 1 Draft to Final .....	15
Changes made to Step 2 Draft .....	15

## Overview

Silicon Shelf is a platform for tracking personal reading lists as well as coordinating reading with friends via reading clubs. A single user can add books to their own log, manage their reading list with status updates, track dates which they started and completed each book. Users may create and join reading clubs. This allows them to add a book to all their club members' reading logs. With an anticipated 10,000 monthly active users and an average yearly enqueue rate of 10 books per user, a robust relational database is needed.

## Database Outline

### ❖ Readers (Object Entity)

- readerID: INT, AUTO\_INCREMENT, NOT NULL, PK
- name: VARCHAR(50), NOT NULL
- email: VARCHAR(50), NOT NULL
- relations:
  - M:M Books through ReadingLogs
  - M:M ReadingClubs through ClubMembers

### ❖ Books (Object Entity)

- bookID: INT, AUTO\_INCREMENT, NOT NULL, PK
- title: VARCHAR(50), NOT NULL
- author: VARCHAR(50), NOT NULL
- year: DATE()
- relations:
  - M:M Readers through ReadingLog
  - M:M ReadingClubs through ReadingLog

### ❖ ReadingClubs (Object Entity)

- clubID: INT, AUTO\_INCREMENT, NOT NULL, PK
- clubName: VARCHAR(50), NOT NULL
- relations:
  - M:M Readers through ClubMembers
  - M:M Books through ClubMembers

### ❖ ReadingLogs (Transaction Entity)

- logID: INT, AUTO\_INCREMENT, NOT NULL, PK
- readerID: FK(Readers.readerID), ON DELETE SET NULL
- bookID: FK(Books.bookID), ON DELETE SET NULL

- readingClubID: FK (ReadingClubs.clubID), DEFAULT NULL, ON DELETE SET NULL
- statusID: FK (ReadingStatus.status), ON DELETE SET NULL
- timeStamp: DATETIME, DEFAULT, (CURRENT\_TIMESTAMP)
- relations:
  - 1:M Readers
  - 1:M Books
  - 1:M ReadingClubs
  - 1:M ReadingStatus

#### ❖ ClubMembers (Composite Entity)

- clubMemberID: INT, NOT NULL, AUTO\_INCREMENT, PK
- readerID: FK(Readers.readerID), NOT NULL, ON DELETE CASCADE
- clubID: FK(ReadingClubs.clubID), NOT NULL, ON DELETE CASCADE
- relations:
  - M:1 Readers
  - M:1 ReadingClubs

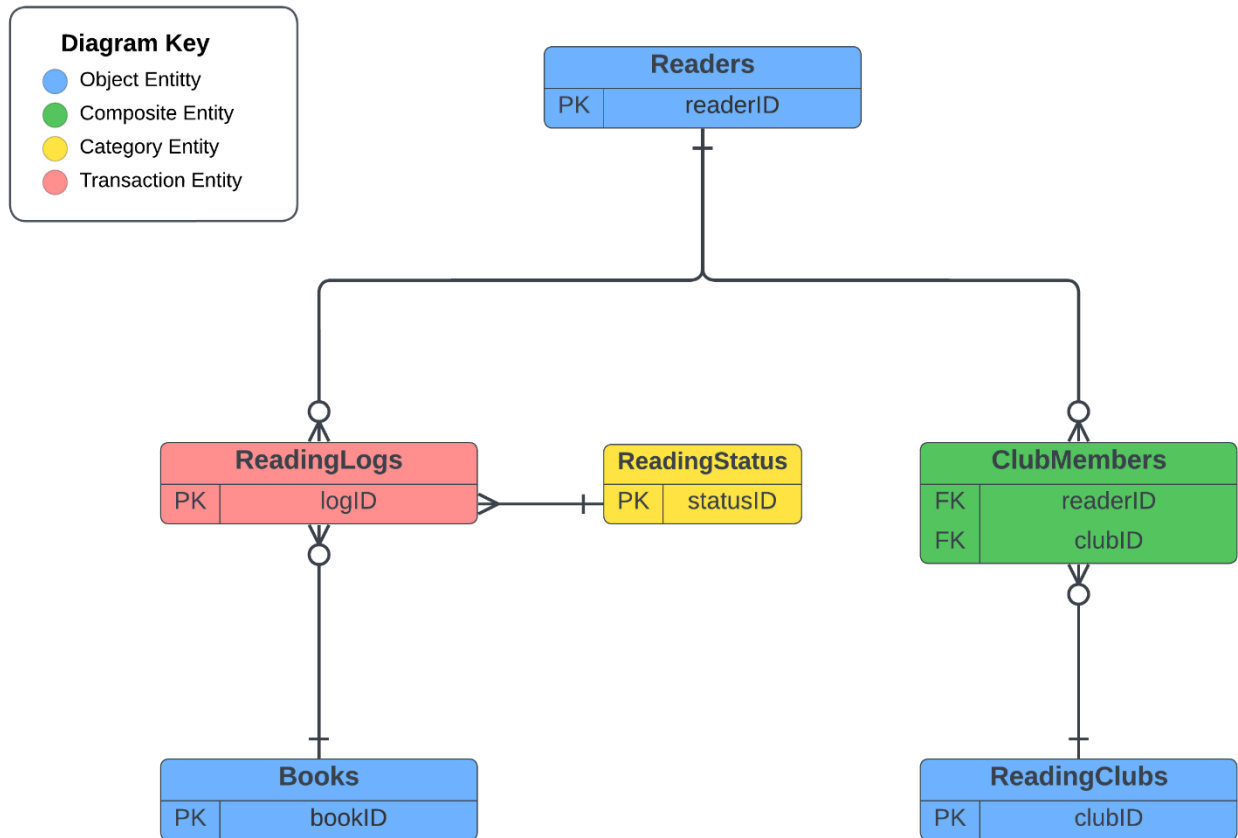
#### ❖ ReadingStatus (Category Entity)

- status: INT, NOT NULL, AUTO\_INCREMENT PK
- status: VARCHAR(50), NOT NULL

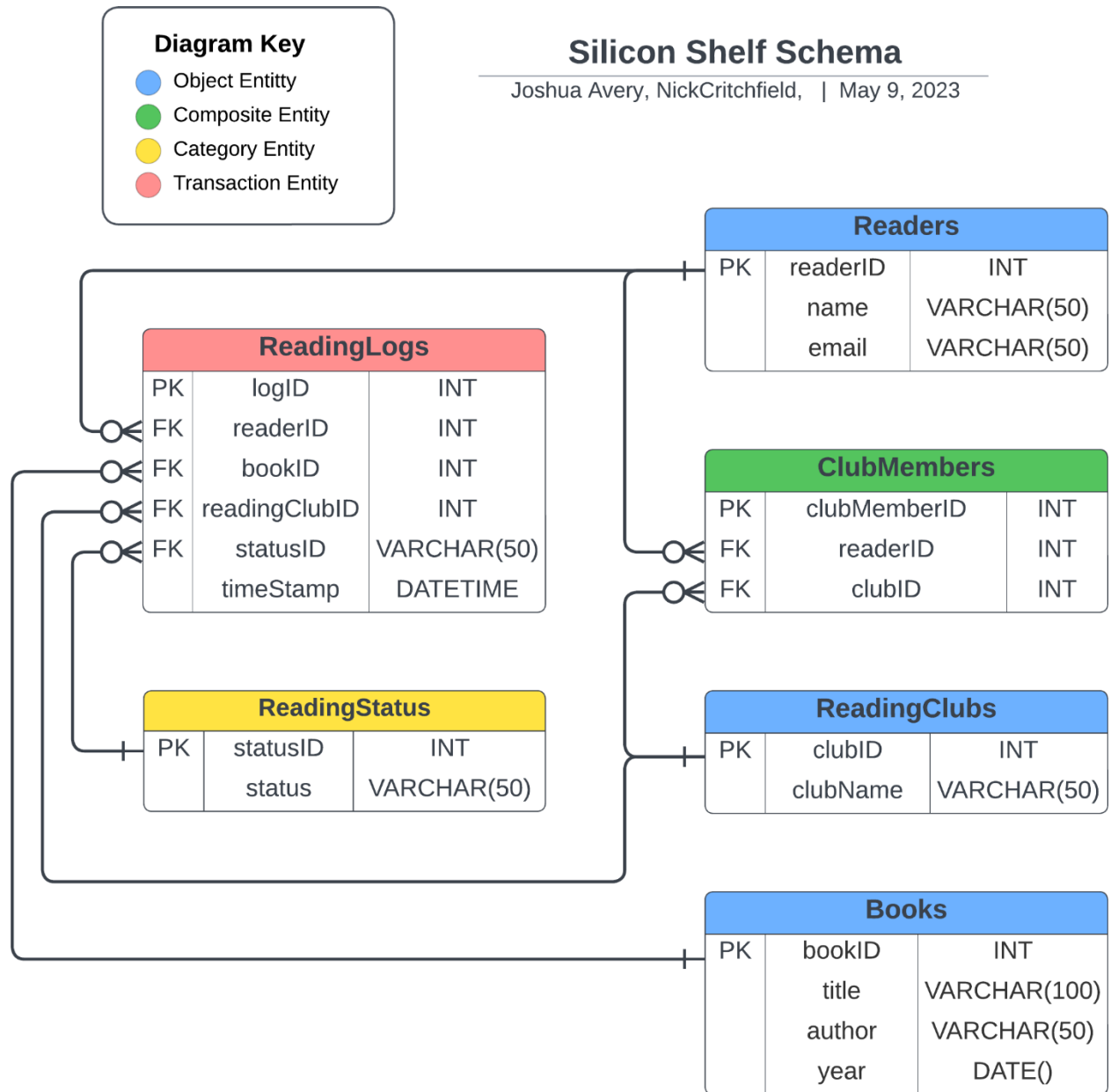
## Entity-Relationship Diagram (ERD)

### Silicon Shelf ERD

Joshua Avery, NickCritchfield, | May 2, 2023



## Schema



## Example Data Inserted for Step 3:

Sample data for Readers:

readerID	name	email
1	Joseph McReading	j.mcreads@notreal.com
2	Sarah Jessica Booker	sjb@supercom.net
3	Mr. Adultman	doing.business@thebusinessfactory.com
4	Daniel Abraham	daniel.abraham@ghmail.com
5	Ro Himbo	rohimbo@derpinout.com
6	Jeroshi Yoshi	jer.yo@jeryo.cnet
7	Samuel Hackins	samuel.hackins@g.mail.com

Sample data for Books:

bookID	title	author	year
1	Leviathan Wakes	James S.A. Corey	2011-00-00
2	Gray's Anatomy	Henry Gray	1858-00-00
3	The Bell Jar	Sylvia Plath	1963-00-00
4	Finite and Infinite Games	James P. Carse	1986-00-00

Sample data for ReadingClubs:

clubID	clubName
1	Tequila Mockingbird
2	Gone with the Gin
3	Nihilists Anonymous

Sample data for ReadingStatus:

statusID	status
1	Enqueued
2	Reading
3	Finished

Sample data for ClubMembers:

clubMemberID	readerID	clubID	isCoordinator
1	1	1	0
2	6	1	0
3	3	1	0
4	4	3	1
5	5	3	0

Sample data for ReadingLogs:

logID	readerID	bookID	readingClubID	statusID	timeStamp
1	1	2	1	1	2023-05-02 10:22:38
2	6	1	NULL	3	2023-05-02 10:22:38
3	4	1	3	2	2023-05-02 10:22:38
4	2	3	NULL	1	2023-05-02 10:22:38
5	5	1	3	2	2023-05-02 10:22:38

## Step 4 Feedback

We did not receive any official peer review/ instructor comments during step 4. We did receive a comment with the following:

Comment by Taylor Jordan

Regarding point 3, I think your 2 options would be either setting the child elements in the DDL to ON DELETE SET NULL or ON DELETE RESTRICT (might want to double check the syntax, but those 2 options, in any case), depending on if you want to force the DB admin to delete dependents before parents (the latter case), or if you're OK with Orphaned FK elements when parents are deleted (which can potentially be adopted by a new parent with an update later). If you set them to null, the FK's will just be replaced by null values in the impacted table.

You then just need to make sure any logic that processes those attributes on the page can handle null possibilities (which led to some crashes on our own page until we added in some conditionals to handle them).

## Actions Taken/Updates from Step 4:

- The following actions were taken to resolve our issue with deleting data with FK constraints:
  - Updated ClubMembers constraints: Both FKs set to ON DELETE CASCADE
  - Updated ReadingLogs constraints: All FKs set to ON DELETE SET NULL
    - NOT NULL removed from corresponding attributes on this entity. These are now reflected in the Database Overview in this document.
      - readerID
      - bookID
      - readingClubID
      - statusID



### Step 3 Feedback

#### Peer Review by David Emmons

“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 appears to be select queries for each entry based off of the schema, though there is no sample data currently.

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

There appear to be drop-down search options but none are currently populated / implemented with sample data.

Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.

The UI appears to have an insert on each entity.

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).

I could be wrong but it does not appear that inserting is currently adding FK attributes, each insert appears to be acting as a stand-alone, you would need to manually insert the values into your 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.

Every entity appears to be delete-able moving forward (the word delete exists in every entity table), though currently the delete option is non-functional and does not provide a form.

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?

Every entity appears to be updateable, though there are currently no ways to select the thing you want to update, and no sample data is provided.

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.

I am not completely sure, though I would assume it would be possible for ReadingClubs to be optional for Readers? Possible that will be a Nullable relationship moving forward.

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.

Being able to directly navigate between entity pages would be convenient, and so would having the current entity page you are on labelled as such. Drop down menus / sample data being implemented would make it easier to imagine what the website will look like going forward, and might change you opinions on how you want the tables structured as well."

#### Instructor Review by Michael Curry

"Dear group 36, nice work getting your UI deployed to flip. Here is some feedback to assist you.

Does the UI utilize a SELECT for every table in the schema? In the DML.SQL I do see SELECT queries for each entity. However, the UI mockup does not have any sample data. I do see sample data in the DDL.SQL so my assumption is that it will be coming, but for now all I see is an empty grid.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties? I do see empty drop down boxes, for example ReadingStatus has an empty drop down for Select statusID. One caution I would give is that it is not generally acceptable to expect the user to remember IDs, so we prefer instead that you use a select to populate a drop down with a user-friendly reading status (e.g. "Enqueued") instead of an ID. Additionally, I do not see any SELECT statements in the DML.SQL that have comments which note that they are selecting for drop downs which further supports my assumption that this requirement is missing.

Does the UI implement an INSERT for every table in the schema? Yes I see an insert form on each entity, good work!

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? I see the word delete on every entity, which suggests you may be planning to add an operation for each one. This is an assumption as there are no links or buttons yet. If correct, there are more deletes than specified by the CS 340 Project Guide which is awesome but only implementing ClubMembers would seem to be required.

Is there at least one UPDATE for any one entity? Yes, here too I see there are more updates than specified by the CS 340 Project Guide which is awesome but only implementing ClubMembers would seem to be required. Again, it is not generally acceptable to expect the user to remember IDs, so we prefer instead that you use a select to populate a drop down with a user-friendly value e.g. book title, instead of an ID.

Is at least one relationship NULLable? Hard to tell at this stage. I see CASCADE operations in the DDL, so this could possibly fulfill the requirement. I also see empty drop down boxes on the ClubMembers which could be used to set the FK to NULL.

Do you have any other suggestions for the team to help with their HTML UI? It was hard to tell what entity I was on, I suggest adding a title to the page to name the entity, e.g Books, Members, etc. It was also one additional click to have to navigate back to the Manage Database page and then to the next entity instead of putting a menu across the top of the page that allowed me to navigate to each entity with one click.

Overall great work, I think that you can easily address my feedback and I look forward to seeing your project develop further.”

#### Actions Taken from Step 3 Feedback:

- Updates to Static HTML pages:
  - Readers: Removed readerID selector element from update form, added edit column to table
  - ReadingClubs: Removed clubID selector element from update form, added edit column to table
  - Books: Removed bookID selector element from update form, added edit column to table
  - ReadingLogs: Removed update/delete functionality entirely, changed selectors in create form to fk join attribute names
  - ClubMembers: Removed update functionality entirely, changed selectors in create form to fk join attribute names
  - ReadingStatus: Removed statusID selector element from update form, added edit column to table
  - Added Page Headers under Nav Pages for all 6 entity pages.
- Updates to DML.sql
  - Added SELECT queries to populate update forms (all tables)
  - Removed update/delete queries for ReadingLogs, removed update query for ClubMembers
- Updates to DDL.sql
  - Made clubIDs nullable on delete in ClubMembers table

## Step 2 Feedback

Peer Review by Colin Maloney:

"Nice work Nick and Joshua! Feedback below.

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

Yes, a) the flow of primary keys is mirrored from the ERD and b) the schema appears as described in the outline. One piece of feedback on the outline is that I believe the appropriate syntax for many-to-many is M:N not M:M, so would suggest updating that.

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, there is consistency between the illustrations and outline, and plurality is consistent as well. Regarding the use of capitalization, it appears that the attributes are camelCase but the table titles are PascalCase. Perhaps this is intentional, but worth noting.

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

The schema is very easy to read. Nicely done including color coding as well!

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

Yes, ClubMembers is an example of an intersection table which is properly formed as displays a many-to-many relationship.

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

Based on the sample data, I do not see any normalization issues or partial dependencies. It appears to be in 3nf based on my understanding of the project.

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 functional and syntactically correct.

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

Yes - ID types are integers, strings are varchar, and binary attributes are Boolean.

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

The usage of keys does match the schema. CASCADE operators are correctly defined in the clubMember table.

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

Yes, the relationship tables outlined in the schema are all present in the SQL.

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

Yes, all of the example data is represented by insert statements in the SQL file.

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

Yes, the sections are clearly defined and annotated (i.e., `/* populate ClubMembers*/`), including a heading and even optional print statements at the bottom."

Peer Review by Mark Marrero:

"First, I like the idea as a reader, I can appreciate this kind of database. I'm going to focus primarily on the schema and your SQL file. Your schema is clean. I can easily follow the links from entity to entity and the intersection tables where they are needed. I like the color coding and key included. It improved the readability of the schema!

Your schema seems free of typos or name mismatches in capitalization and such so that is good! It also looks like it matches up to the outline and ERD. I noticed your schema has the ERD symbols and connections, I don't know if that required or needed, but I liked seeing them. As far as my understanding of normalization goes, I think it is normalized and accurate to 3nf.

Your SQL file is very nice and runs well, and I also enjoyed your sample tables. It matches up to the ERD and schema so far so good.

I did not see any syntax errors, and your data types are fine. Your inserts went well, I particularly enjoyed the Mr. Adultman in Readers. I also noticed a good book in the Books table, Leviathan Wakes. Your keys are present and correct. Your cascades are present in the clubMember table and look right.

This is some good work and some clean SQL writing. It almost makes me feel like I should go back and rewrite my scripts to make them look nicer. Great success!"

TA Feedback by Katie Russell

"Great work group 36! Your outline, erd, schema and ddl all look great. I really like the upgraded visuals for the erd and the new schema. A small note, your example data for your intersection tables should properly show the M:M relationship. For example, Clubs to Readers M:M, you are currently showing that a Club can have many readers, but you also have to show that a reader can belong to many clubs. given (readers, clubs) --> (1,1), (1,3), (2,3) is sufficient to show this. Also in the future, please submit your document as a PDF."

### Actions Taken from Step 2 Feedback Step

- Submitting as PDF not docx.
- Reordered project document to better fit assignment specifications.
- Added new insert to DDL.sql to show that a reader may belong to many clubs.

## Step 1 Feedback

### Peer Review by Jacob Ogle:

"Hey Nice and Josh, great idea on the database and I think this will turn into a great project.

I really think this is a well thought out system and should be pretty fun to design a web application around. I think using a database as a priority queue will be interesting to implement. Would a reader be allowed to set multiple different books with similar priorities or would the application allow certain rankings.

For suggestions, they are kind of nit-picky but they're just suggestions 😊. I would possibly bump the varchar sizes up a bit. For example, in the ReadingClubs entity it might be nice to have the varchar size set to something like (50). Same with name, email. etc. "

### Peer Review by John Lofgren:

- "Does the overview describe what problem is to be solved by a website with DB back end?
  - Yes. It is a website for tracking books and the features it will have directly contribute to that goal.
- Does the overview list specific facts?
  - Yes! I think it was sparse on detailing the "numbers" each feature would have, but the diagram later on does a great job of showing how the DB is going to work.
- 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 and match requirements.
- Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?
  - Yes it meets all requirements.
- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?
  - Yes they are.
- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
  - Yes naming conventions are standardized. Only inconsistency is email address is listed on the ERD, but nowhere else and it does not follow the camelcase used elsewhere.

Overall: The intro could probably be more descriptive of your overall plan, but it is a nitpick. I think that your ERD is great and really pulls your whole plan together. When I saw it and then went back and read your descriptions it all clicked for me. The only variable I didn't get is maxRosterSize. More from a user standpoint than a database. If its a reading club and you want it to be exclusive, then the Coordinator can approve or disapprove applicants, and if they don't care if its exclusive, then I am not sure they would care about the maxroster."

### TA Feedback by Amelia Kawasaki

“Hey team, nice step 1 draft. I would recommend providing more detail in your project outline and also revising your ERD. The ERD should be less of a technical blueprint (aka a schema diagram) and more of a high-level logical overview of the database entities and their relationships. ERD diagrams often leave out the finer details such as attributes and intersection tables as its main focus is to demonstrate the rough structure and interconnectivity of the database tables. I would also consider removing some of the extra intersection tables from your project. I worry that implementing so many of these tables further in the quarter will be incredibly time consuming. I enjoy the effort you both put into the draft but I don't want you to have to spend all of your time on this class in the future!”

### Actions Taken from Step 1 Feedback Step

- Removed `maxRosterSize` attribute from `ReadingClubs` entity per John Lofgren.
- Increased `VARCHAR` size on all attributes from 32 to 50 per Jacob Ogle.
- Corrected `email address` on ERD to `email` for naming consistency per John Lofgren.
- Expanded Overview to include more details regarding functionality.

### Updates from Step 1 Draft to Final

- Additional proof read, spelling correct.
- Formatted text.
- Added Table of Contents to top of draft.
- Page layout edits.
- Scaled ERD.

### Changes made to Step 2 Draft

- Identified and removed redundancies:
  - Removed `Recommendations` Entity
  - Removed `ReadingQueues` Entity
  - Added `ReadingStatus` category Entity
  - Updated relationships on all entities
- Added default values for some key attributes
- Added `clubID` FK to `ReadingLogs` entity
- Added `isActive` to `ClubMembers` entity
- Reverse-engineered ERD from Schema submitted in Step 1 (per TA feedback)