Reader Log: “Silicon Shelf”

Joshua Sears, Nick Critchfield

PHASE: CS 340: PROJECT STEP 1 FINAL VERSION (GROUP, ON ED DISCUSSIONS)

DUE MONDAY, 4/24/23

SUBMITTAL: SUBMIT PDF TO CANVAS GROUP PAGE

Contents

Overview

Silicon Shelf is a platform for tracking personal reading lists as well as coordinating reading with friends. A database-driven website will allow users to manage reading queues and recommend books to other users. Users can add books to their personal reading log, ranking them by priority and tracking completion status. Additionally, book club functionality allows users to recommend books to everyone else in each club. 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) Details of a reader (user) who may read books, queue books to be read, participate in a book club, and recommend books.
    - `readerID: INT, AUTO_INCREMENT, not NULL, PK`
    - `name: VARCHAR(50), not NULL`
    - `email: VARCHAR(50), not NULL`
    - relations:
        - M:M `Books` through `ReadingLog`
        - M:M `ReadingClubs` through `ClubMembers`
        - M:M `Books` through `ReadQueues`
        - 1:M `Recommendations`
- ❖ `Books` (Object Entity) Details of a book.
    - `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 `Readers` through `ReadQueues`
        - 1:M `Recommendations`
- ❖ `ReadingClubs` (Object Entity) Must be coordinated by one Reader, may specify a maximum roster size. Enrollment in a club's roster serves as an interface to populate Recommendations or ReadQueues to groups of Readers.
    - `clubID: INT, AUTO_INCREMENT, not NULL, PK`
    - `clubName: VARCHAR(50), not NULL`
    - relations:
        - M:M `Readers` through `ClubMembers`
- ❖ `ReadingLogs` (Transaction Entity) Records the status of a Book being read by a reader.

- logID: INT, AUTO_INCREMENT, not NULL, PK
- readerID: FK(Readers.readerID), not NULL
- bookID: FK(Books.bookID), not NULL
- status: VARCHAR(50)
- dateStarted: DATE()
- dateCompleted: DATE()
- relations:
  - 1:M Readers
  - 1:M Books

❖ Recommendations (Transaction Entity) Using Recommendations, a reader may see a list of suggested books by different readers.

- recommendedBy: FK(Readers.readerID), not NULL
- recommendedTo: FK(Readers.readerID), not NULL
- bookID: FK(BookID), not NULL
- recommendationAccepted: BOOL
- relations:
  - M:1 Books
  - M:1 Readers

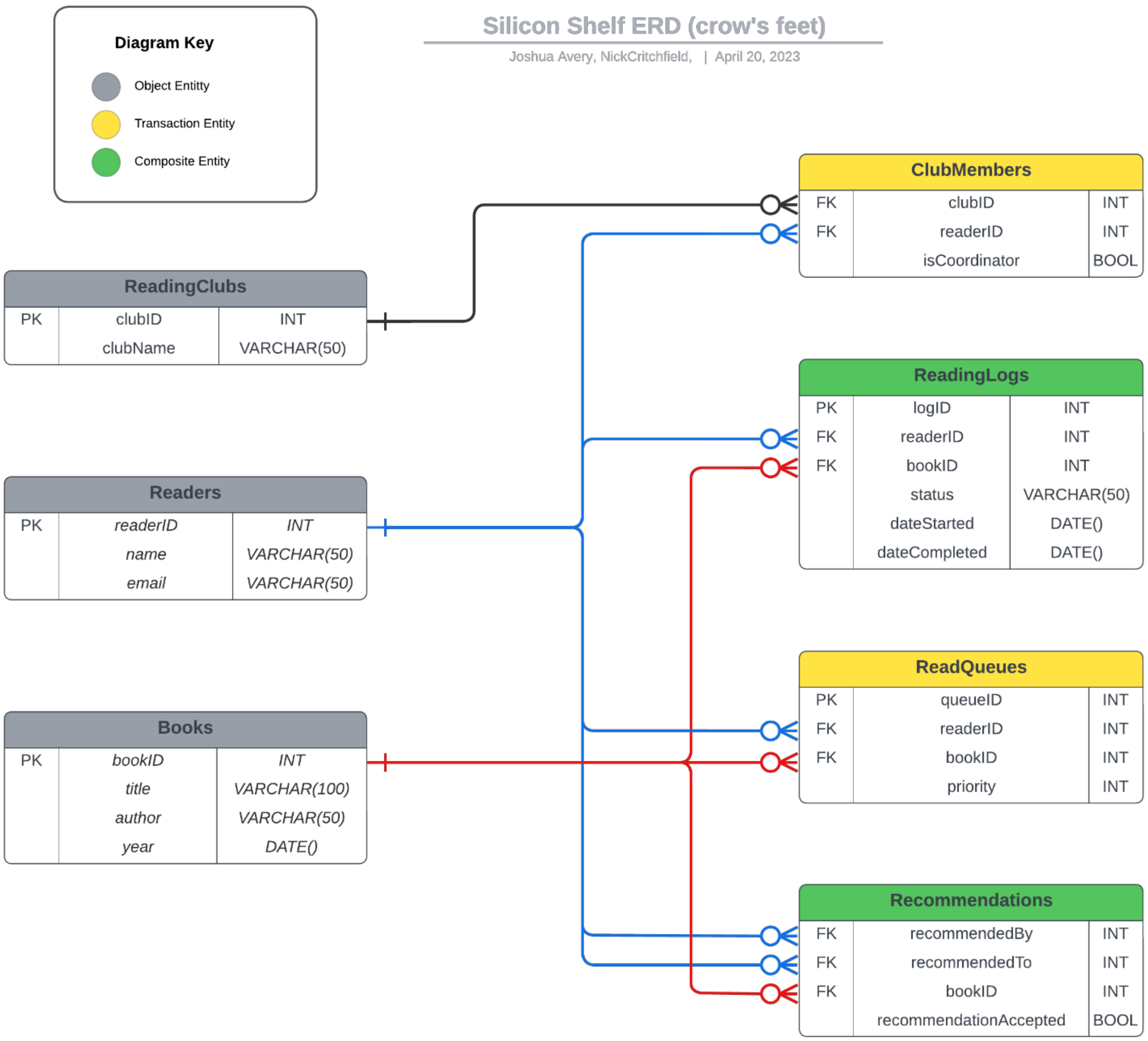
❖ ClubMembers (Composite Entity) Records membership in a club.

- clubID: FK(ReadingClubs.clubID), not NULL
- readerID: FK(Readers.readerID), not NULL
- isCoordinator: BOOL
- relations:
  - M:1 Readers
  - M:1 ReadingClubs

❖ ReadQueues (Composite Entity) Records books in a queue.

- queueID: INT, not NULL, PK
- readerID: FK(Readers.readerID), not NULL
- bookID: FK(Books.bookID) , not NULL
- priority: INT, default = 1
- relations:
  - M:1 Readers
  - M:1 Books

Entity-Relationship Diagram (ERD)



Silicon Shelf ERD (crow's feet)

Joshua Avery, NickCritchfield, | April 20, 2023

Diagram Key

- ● Object Entitty
- ● Transaction Entity
- ● Composite Entity

ClubMembers

FK	clubID	INT
FK	readerID	INT
	isCoordinator	BOOL

ReadingClubs

PK	clubID	INT
	clubName	VARCHAR(50)

ReadingLogs

PK	logID	INT
FK	readerID	INT
FK	bookID	INT
	status	VARCHAR(50)
	dateStarted	DATE()
	dateCompleted	DATE()

Readers

PK	*readerID*	*INT*
	name	*VARCHAR(50)*
	email	*VARCHAR(50)*

ReadQueues

PK	queueID	INT
FK	readerID	INT
FK	bookID	INT
	priority	INT

Books

PK	*bookID*	*INT*
	title	*VARCHAR(100)*
	author	*VARCHAR(50)*
	year	*DATE()*

Recommendations

FK	recommendedBy	INT
FK	recommendedTo	INT
FK	bookID	INT
	recommendationAccepted	BOOL

Peer Review Feedback

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. ”

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 a s 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 then 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.“

Actions Taken from Feedback

- 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.

Upgrades from Draft to Final

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