Reader Log: "Silicon Shelf"

Joshua Sears, Nick Critchfield

Contents

Executive Summary

This project started with a hefty seven entities in the first draft that offered multiple features for would-be users. In this initial model, we had a core three object entities: Readers, Books, and Reading Clubs. The other four entities each offered a "feature" to the users: a Club Members entity to register for Reading Clubs, a Reading Log to log reading activity, a Read Queue to order your reading list, and a Recommendation table to provide book recommendations to other users.

We reanalyzed our approach to this project going into step two and decided to focus on our core objects with just two "features" instead of four. This reduced our initial implementation to five entities. Because of how our Reading Log entity took shape, we also needed to add back a sixth small entity to pass the 3 normal form standards: a category entity for Reading Statuses. Additionally, in step two, we trimmed down attributes on our intersection tables (Club Members and Reading Log) to just the essential data point and foreign keys.

Initially, this project did not have a high-level ERD, instead it only had a more technical schema. To include the higher-level overview of the project, we reverse engineered from our schema the ERD now included.

The web implementation steps focused on creating an "administrator's interface" for managing the database. This included full CRUD operations for all our entities, except for operations which we considered *tampering* with transactional data in our two transaction (intersection) tables.

A key part of our web implementation was deciding on technologies we would utilize in addition to the core stack of Node and MySQL. We decided against using a templating engine such as Handlebars. This way we would be in full control of DOM tree manipulations in our JavaScript.

Midway through web implementation, we needed to fully flush out how we were handling DELETE scenarios for M:N relationships. We added CASCADE constraints to our Club Members intersection table and SET NULL constraints to the Reading Log intersection table.

Our UI was enhanced by integrating the following feedback: adding a secondary navigation bar to allow navigating from entity to entity, loading display tables automatically, and adjusting datatypes for multiple attributes that were causing issue when attempting to add or modify data. After seeing other implementations, we incorporated ORDER BY clauses in our SELECT statements to make reviewing data in tables and drop menus easier for users.

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: VARCHAR(10)`
    - 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`
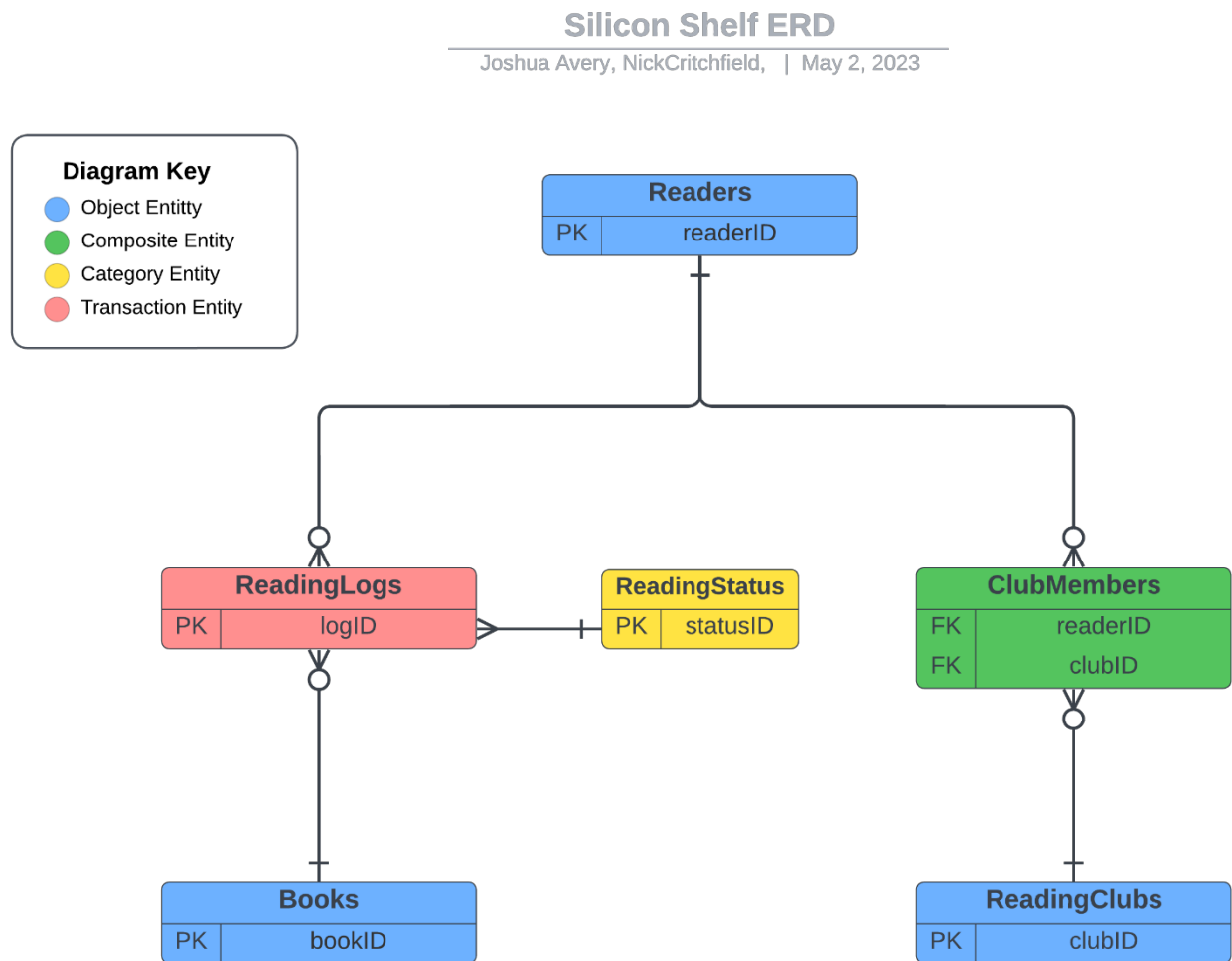 - **1:M** `ReadingClubs`
  - **1:M** `ReadingStatus`

❖ `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`
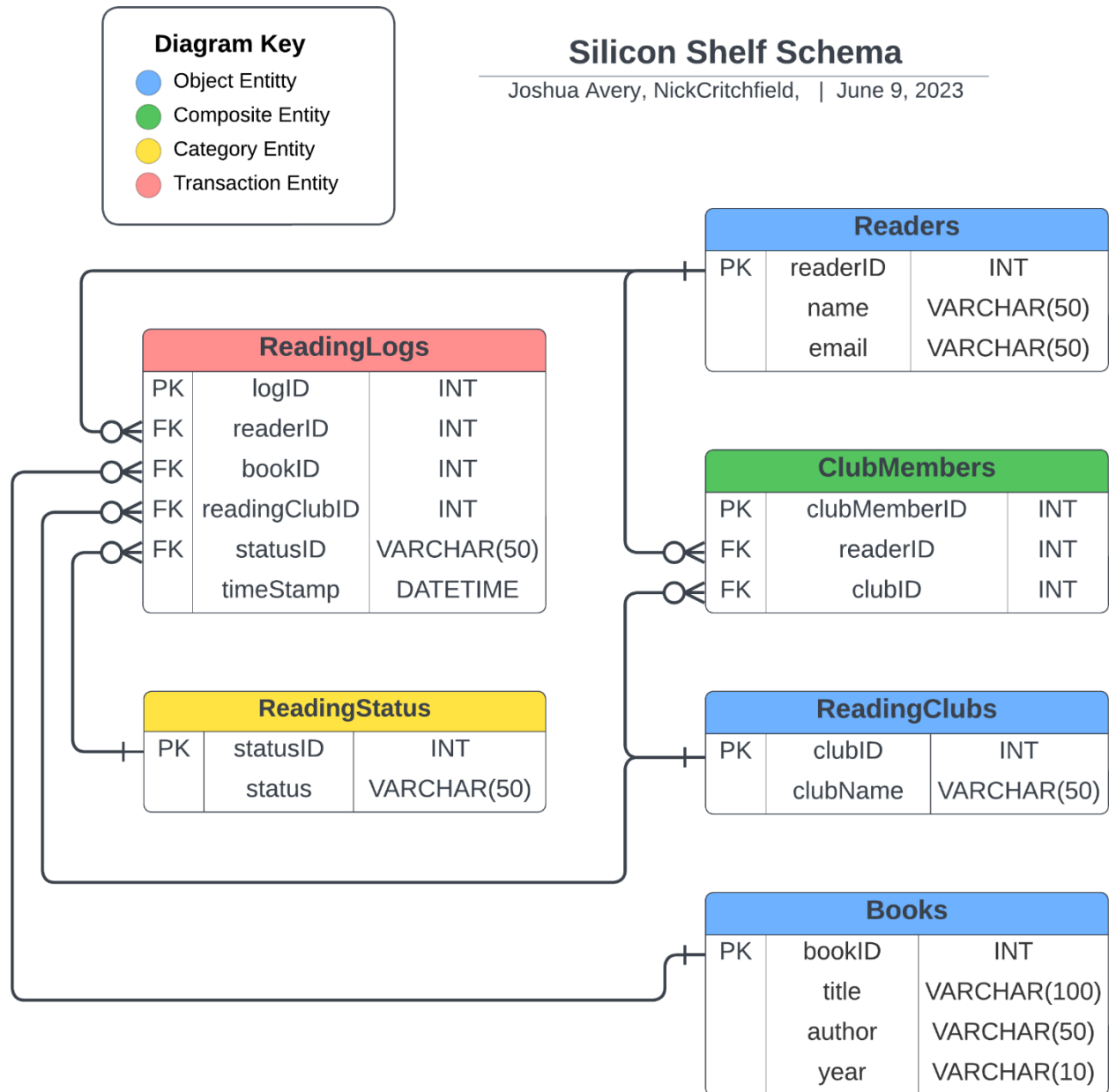 - **M:1** `ReadingClubs`

❖ `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

Diagram Key
- Object Entitty
- Composite Entity
- Category Entity
- Transaction Entity

Readers

PK	readerID

ReadingLogs

PK	logID

ReadingStatus

PK	statusID

ClubMembers

FK	readerID
FK	clubID

Books

PK	bookID

ReadingClubs

PK	clubID

Schema

Diagram Key
- 🔵 Object Entitty
- 🟢 Composite Entity
- 🟡 Category Entity
- 🔴 Transaction Entity

Silicon Shelf Schema
Joshua Avery, NickCritchfield, | June 9, 2023

Readers

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

ReadingLogs

PK	logID	INT
FK	readerID	INT
FK	bookID	INT
FK	readingClubID	INT
FK	statusID	VARCHAR(50)
	timeStamp	DATETIME

ClubMembers

PK	clubMemberID	INT
FK	readerID	INT
FK	clubID	INT

ReadingStatus

PK	statusID	INT
	status	VARCHAR(50)

ReadingClubs

PK	clubID	INT
	clubName	VARCHAR(50)

Books

PK	bookID	INT
	title	VARCHAR(100)
	author	VARCHAR(50)
	year	VARCHAR(10)

Example Data

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 |
|      2 | Gray's Anatomy          | Henry Gray      | 1858 |
|      3 | The Bell Jar            | Sylvia Plath    | 1963 |
|      4 | Finite and Infinite Games | James P. Carse  | 1986 |
+--------+-------------------------+-----------------+------+
```

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

Screen Captures of UI

Home Page with information about the present state of the application for administrators and development aim for users.
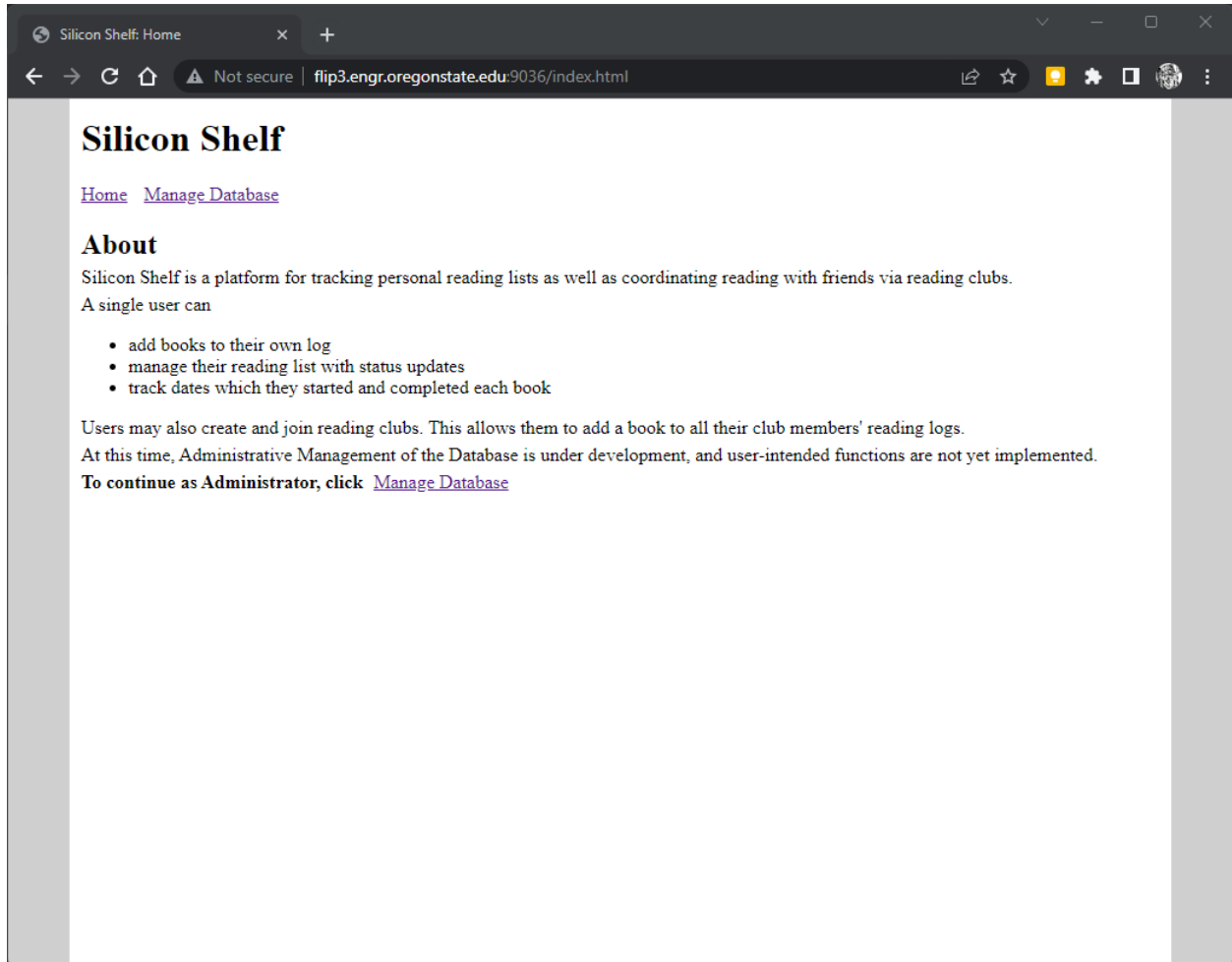
Administrator landing page with diagram and links to all entities.

Books entity: CREATE / RETRIEVE / UPDATE / DELETE

Books is in a M:N relationship with Readers through the ReadingLog intersection table. Note that both UPDATE and DELETE are present here.

ClubMembers entity: CREATE / RETRIEVE / DELETE

Silicon Shelf > Manage Database > ClubMembers

Home   Manage Database

Entities: Books   ClubMembers   ReadingClubs   ReadingLogs   Readers   ReadingStatus

ClubMembers

Readers registered to ReadingClubs.
Click delete to remove record from database.
Add a transaction by selecting required fields from the drop down.

clubMemberID	name	clubName	
6	Ro Himbo	Nihilists Anonymous	Delete
3	Jeroshi Yoshi	Nihilists Anonymous	Delete
5	Daniel Abraham	Nihilists Anonymous	Delete
2	Jeroshi Yoshi	Tequila Mockingbird	Delete
4	Mr. Adultman	Tequila Mockingbird	Delete
1	Joseph McReading	Tequila Mockingbird	Delete

Create
Select name [▼]
Select clubName [▼]
Submit

ReadingClubs entity: CREATE / RETRIEVE / UPDATE / DELETE

ReadingClubs is in a M:N relationship with Readers through the ClubMembers intersection table. Note that both UPDATE and DELETE are present here.

ReadingLogs entity: CREATE / RETRIEVE

Silicon Shelf > Manage Database > ReadingLogs

Home   Manage Database

Entities: Books   ClubMembers   ReadingClubs   ReadingLogs   Readers   ReadingStatus

ReadingLogs

Click delete to remove record from database.

Add a transaction by selecting required fields from the drop down.

logID	name	title	clubName	status	timeStamp
1	Joseph McReading	Gray's Anatomy	Tequila Mockingbird	Enqueued	2023-06-09T20:04:19.000Z
2	Jeroshi Yoshi	Leviathan Wakes		Finished	2023-06-09T20:04:19.000Z
3	Daniel Abraham	Leviathan Wakes	Nihilists Anonymous	Reading	2023-06-09T20:04:19.000Z
4	Sarah Jessica Booker	The Bell Jar		Enqueued	2023-06-09T20:04:19.000Z
5	Ro Himbo	Leviathan Wakes	Nihilists Anonymous	Reading	2023-06-09T20:04:19.000Z

Create

Select name [▼]

Select title [▼]

Select clubName [▼]

Select status [▼]

[Submit]

Readers entity: CREATE/ RETRIEVE / UPDATE / DELETE

Readers is in a M:N relationship with Books through the ReadingLog intersection table and with ReadingClubs through the ClubMembers intersection table. Note that both UPDATE and DELETE are present here.

ReadingStatus entity: CREATE / RETRIEVE / UPDATE / DELETE

Silicon Shelf > Manage Database > ReadingStatus

Home   Manage Database

Entities: Books   ClubMembers   ReadingClubs   ReadingLogs   Readers   ReadingStatus

ReadingStatus

Click delete to remove record from database.
Click edit to populate update form to begin editing.

statusID	status		
1	Enqueued	Edit	Delete
3	Finished	Edit	Delete
2	Reading	Edit	Delete

Create
status [category]
[Submit]

Update
statusID [ ]
status [category]
[Submit]