Group 7

Mason Griffith

Reed Stout

Kellen Brown

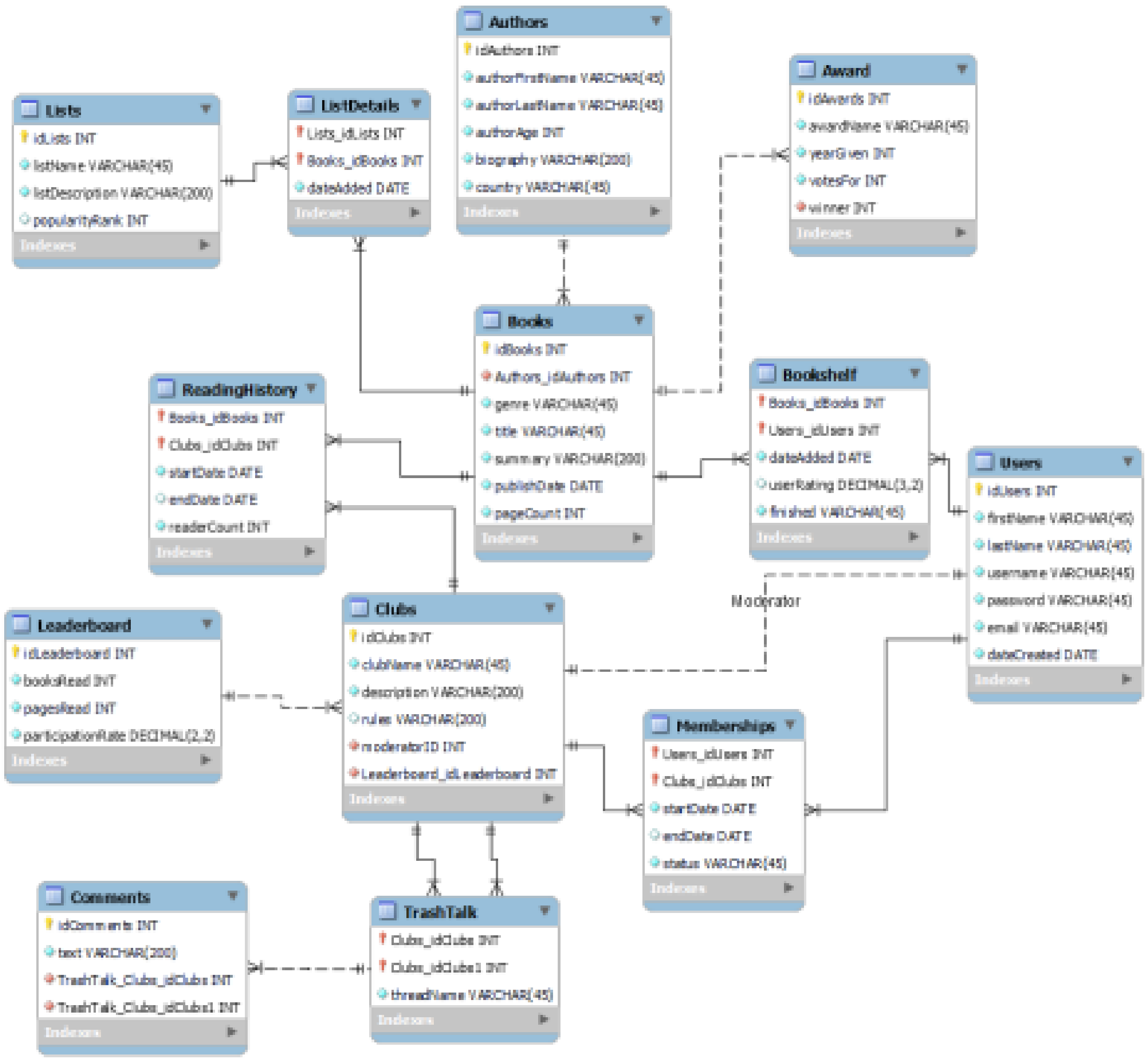
Sahil Khatri

Daniel Shin

Problem Description:

For our "reading list" mobile application we wanted to tackle the stigma of how reading is "boring and antisocial" and promote a fun and competitive culture. Current mobile applications such as Good Reads, Reading List, etc. do a great job of providing information on books, their descriptions and ratings, and clubs you can join to keep up with the reading community. We based a large amount of the framework of the model upon what we observed from using these sites. However, one thing these applications lack is an aspect of competition, and that is the value we wanted to provide in our application. For our application we created our own Leaderboard entity where book clubs are compared to each other on a few different metrics. These include the number of books they have read, total pages read, and participation rate among users in the club. All three of these combined make up the ranking on the leaderboard. Then we also created a TrashTalk entity, which stores a thread name where on the app clubs can post comments addressed to other clubs or just about their current leaderboard status. These are between two clubs at a time, but the amount of threads that can be created by a club is not limited in theory. The Comments entity stores the text of each comment which is useful for front-end development and also moderation. It also allows each thread to have multiple comments on it instead of creating a new conversation each time someone wanted to talk.

Data Model:



Relationship Explanation:

Each book has an author, and an author can write many books. Each book can also win an award, but an award can only go to one book per year. A book may be placed on many created lists, and a list may have many books on it. A user may save books to their virtual bookshelf if they are interested in them. Many books may be saved to many different user's bookshelves. Users may also be a member of many clubs. Many clubs can have many members. Clubs have reading histories consisting of many books and books can be on many club's histories. There is a leaderboard that many clubs have a place on, but there is only one leaderboard. Clubs may "talk trash" to each other via TrashTalk threads. There are many of these threads and each thread has more than one club in it. Further, each TrashTalk thread can have several comments in it, but a comment only belongs to one thread.

Data Dictionary:

Table: **Authors**

Column Name	Description	Data Type	Size	Format	Key?
idAuthors	Unique sequential number identifying each author	text	7		PK
authorFirstName	First name of each author	Text	15		
authorLastName	Last name of each author	Text	15		
authorAge	Age of each author	Text	3		
biography	Brief description of each author	Text	200		
country	Country each author is from	Text	20		

Table: **Award**

Column Name	Description	Data Type	Size	Format	Key?
idAwards	Unique sequential number identifying each award	Text	7		PK
awardName	Name of each award	Text	20		
yearGiven	Year each award was given in	Numeric	4		
votesFor	Amount of votes each winning book received	Numeric	15		
winner	Indicates each book who won an award	text	12		FK (ref. Books)

Table: **Books**

Column Name	Description	Data Type	Size	Format	Key?
idBooks	Unique sequential number identifying each book	Text	12		PK
Authors_idAuthors	Indicates the author of the book	Text	7		FK (ref Authors)
Genre	Genre of each book	Text	15		
Title	Title of each book	Text	20		
summary	Brief summary of each book	Text	200		
publishDate	Date each book was published	Date	8	YYYY MM-DD	
pageCount	Amount of pages in each book	Numeric	4		

Table: **Bookshelf**

Column Name	Description	Data Type	Size	Format	Key?
Books_idBooks	The book that is saved to a user's virtual bookshelf for future reading	Text	12		FK (ref Books)
Users_idUsers	The user who owns the virtual bookshelf	Text	7		FK (ref Users)
dateAdded	Date a book was added to a user's bookshelf	Date	8	YYYY MM-DD	
userRating	The rating each user can give to each book on their bookshelf	Numeric	3	9.99	

finished	Whether a book has been finished by the user	Text	1	'Y' if finished 'N' if not	

Table: **Comments**

Column Name	Description	Data Type	Size	Format	Key?
idComments	Unique sequential number identifying each comment	Text	12		PK
Text	The contents of each comment	Text	200		
TrashTalk_Clubs_idClubs	First club the comment is between	Text	7		FK (ref Groups)
TrashTalk_Clubs_idClubs1	Second club the comment is between	Text	7		FK (ref Groups)

Table: **Clubs**

Column Name	Description	Data Type	Size	Format	Key?
idClubs	Unique sequential number identifying each club	text	7		PK
clubName	Name of each reading club	Text	20		
description	Brief description of each club	Text	200		

rules	Any rules a club might have	Text	200		
moderatorID	The user that moderates the club	Text	7		FK (ref Users)
Leaderboard_idLeaderboard	The rank of each club on the competitive leaderboard	Numeric	4		FK (ref Leaderboard)

Table: **Leaderboard**

Column Name	Description	Data Type	Size	Format	Key?
idLeaderboard	Unique sequential number identifying a club's place on the leaderboard	Numeric	4		PK
booksRead	Number of books read by that rank on the board	Numeric	5		
pagesRead	Number of pages read by that rank on the board	Numeric	10		
participationRate	Proportion of users that are active in the club compared to total members	Numeric	2	.99	

Table: **ListDetails**

Column Name	Description	Data Type	Size	Format	Key?
Lists_idLists	List being described	Text	7		FK (ref Lists)

Books_idBooks	Book on the list being described	Text	12		FK (ref Books)
dateAdded	Date the book was added to the list	Date	8	YYYY-M M DD	

Table: **Lists**

Column Name	Description	Data Type	Size	Format	Key?
idLists	Unique sequential number identifying each list	Text	7		PK
listName	Name of each list	Text	20		
listDescription	Brief description of each list	Text	200		
popularityRank	Ranking of how popular the list is on the app	Numeric	4		

Table: **Memberships**

Column Name	Description	Data Type	Size	Format	Key?
Users_idUsers	User that is a member of the club	Text	7		FK (ref Users)
Clubs_idClubs	Club the user is a member of	Text	7		Fk (ref clubs)
startDate	Date the membership started	Date	8	YYYY MM-DD	
endDate	Date the membership ended (if it has)	Date	8	YYYY MM-DD	
status	If the membership is active or not	Text	8	‘active’ or ‘inactive’	

Table: **ReadingHistory**

Column Name	Description	Data Type	Size	Format	Key?
Books_idBooks	The book that is included in a club's reading history	Text	12		FK (ref Books)
Clubs_idClubs	The club that is keeping track of the books it reads	Text	7		FK (ref clubs)
startDate	Date the club started reading the book	Date	8	YYYY MM-DD	
endDate	Date the club stopped reading the book	Date	8	YYYY MM-DD	
readerCount	Amount of members of the club that read the book	Numeric	7		

Table: **TrashTalk**

Column Name	Description	Data Type	Size	Format	Key?
Clubs_idClubs	First club participating in the Trash Talk thread	Text	7		FK (ref Groups)
Clubs_idClubs1	Second club participating in the Trash Talk thread	Text	7		FK (ref Groups)
threadName	Name of the thread of conversation between two clubs	Text	20		

Table: **Users**

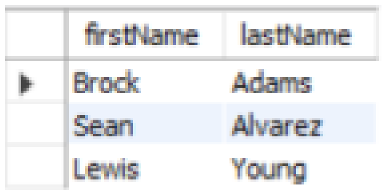
Column	Description	Data	Size	Format	Key?

Name		Type			
idUsers	Unique sequential number identifying each user	Text	7		PK
firstName	First name of each user	Text	15		
lastName	Last name of each user	Text	15		
username	Username a user chooses	Text	15		
password	Password a user chooses to access their account	Text	15		
email	Email address of each user	Text	20		
dateCreated	Date each user's account was created	Date	8	YYYY-M M DD	

Queries:

TP_Q1: What are the full names (first and last) of users who have read a book with the word "the" in the title? Order by last name alphabetically.

```
select firstName, lastName from Users
join Bookshelf on Users.idUsers = Bookshelf.Users_idUsers
join Books on Bookshelf.Books_idBooks = Books.idBooks
where title regexp 'the'
order by lastName;
```



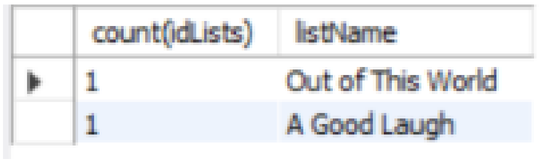
```
firstName  lastName
Brock      Adams
Sean       Alvarez
Lewis      Young
```

This would be useful to see what users are reading and filter it by a keyword if someone was doing research to help cater books to specific users.

TP_Q2: What lists have a book on them that won an award before 2020 and how many on each? select count(idLists), listName from Lists join ListDetails

on Lists.idLists = ListDetails.Lists_idLists

```
join Books on ListDetails.Books_idBooks = Books.idBooks
join Award on Books.idBooks = Award.winner
where Award.yearGiven < 2020
group by listName ;
```

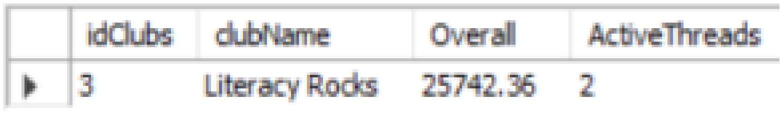


count(idLists)	listName
1	Out of This World
1	A Good Laugh

This would be useful if someone was looking for award winning books on lists from a specific time period or that were not given awards in the past couple years.

TP_Q3: All Bark No Bite Club- Clubs in more than 1 trash talking thread with a below average Overall Club Score

```
select idClubs, clubName, pagesRead*participationRate as Overall, Count(threadname) as
ActiveThreads from Clubs
Join Leaderboard on Leaderboard.idLeaderboard=
Clubs.Leaderboard_idLeaderboard Join TrashTalk on Clubs.idCLubs =
TrashTalk.Clubs_idClubs
Group by idCLubs
Having ActiveThreads > 1 and Overall < 26000;
```

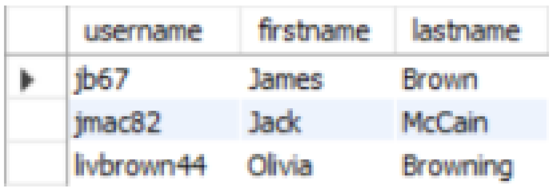


idClubs	clubName	Overall	ActiveThreads
3	Literacy Rocks	25742.36	2

This query is helpful in that it can establish what groups need to up their performance to back up the trash talk they've been putting up. This will not only encourage the bad groups to up their work but to give the better groups firepower to dominate the thread. A huge aspect of our angle of the app is promoting competition and this is another way to facilitate that.

TP_Q4: Users who haven't added a book in over a year

```
select username, firstname, lastname from Users
Left Join Bookshelf on Bookshelf.Users_idUsers = Users.idUsers
WHERE NOT EXISTS (select datediff(Current_date(), Bookshelf.dateAdded) from
Bookshelf Where Bookshelf.Users_idUsers = Users.idUsers
and datediff(Current_date(), Bookshelf.dateAdded) < 365);
```

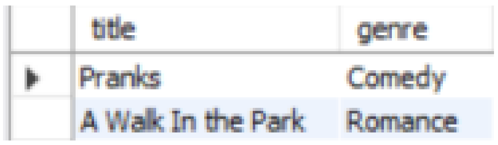


username	firstname	lastname
jb67	James	Brown
jmac82	Jack	McCain
livbrown44	Olivia	Browning

This query finds what Users haven't read a book in the past year. This not only allows friends to encourage other friends to continue reading but also for back end managers to remind users to use the app.

TP_Q5: List the titles and genres of the books whose authors' first name are Jim or Emma.

```
SELECT title, genre FROM Books
JOIN Authors ON Books.Authors_idAuthors = Authors.idAuthors
WHERE authorFirstName IN('Jim', 'Emma');
```

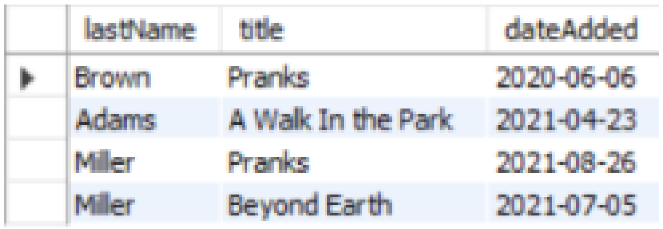


title	genre
Pranks	Comedy
A Walk In the Park	Romance

This query is useful to find a book that is written by a certain author. For example if a user really enjoys reading books that are written by two certain authors, this query could be useful to filter and locate the books that are written by those authors.

TP_Q6: List the title and the date added of books in users' bookshelves that have a Science Fiction, comedy, or romance genre.

```
SELECT Users.lastName, title, dateAdded FROM Bookshelf
JOIN Books ON Bookshelf.Books_idBooks = Books.idBooks
join Users on Bookshelf.Users_idUsers = Users.idUsers
WHERE genre IN ('Science Fiction', 'Comedy', 'Romance')
AND EXISTS (SELECT * FROM Bookshelf WHERE Bookshelf.Books_idBooks =
Books.idBooks);
```

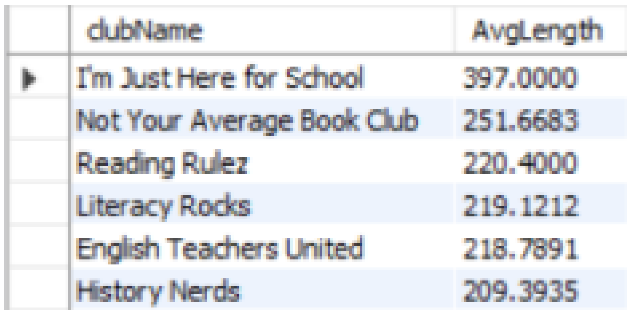


lastName	title	dateAdded
Brown	Pranks	2020-06-06
Adams	A Walk In the Park	2021-04-23
Miller	Pranks	2021-08-26
Miller	Beyond Earth	2021-07-05

This would be useful in order to see what books are being added to users' bookshelves that follow a specific genre.

TP_Q7: What is the average length of the books that each club has read? Display results from longest to shortest average.

```
select clubName, (pagesRead/booksRead) as AvgLength from Clubs
join Leaderboard on Leaderboard.idLeaderboard=Clubs.Leaderboard_idLeaderboard
order by AvgLength desc
```

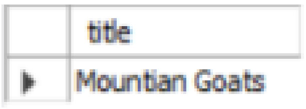


clubName	AvgLength
I'm Just Here for School	397.0000
Not Your Average Book Club	251.6683
Reading Rulez	220.4000
Literacy Rocks	219.1212
English Teachers United	218.7891
History Nerds	209.3935

This would be useful to see the strategies of each club for climbing the leaderboard. Are they going for more pages read with longer books or do they just want to power through more books overall?

TP_Q8: What books are on a list before 2021 and have lower than average rating?

```
select title from ListDetails
join Books on ListDetails.Books_idBooks=Books.idBooks
join Bookshelf on Bookshelf.Books_idBooks=Books.idBooks
where ListDetails.dateAdded<'2021-01-01' and
Bookshelf.userRating<(select avg(Bookshelf.userRating) from
Bookshelf);
```



title
Mountian Goats

This would be useful to see which books that are on lists may be outdated as their rating is below average. It would allow us to see which books may need to be taken off lists.

TP_Q9: Books on a user's bookshelf that is above the average length of all books on bookshelves.

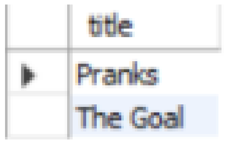
```
SELECT DISTINCT(title)

FROM Bookshelf
JOIN Books ON idBooks = Books_idBooks

WHERE pageCount > ( SELECT AVG(pageCount) FROM Books JOIN Bookshelf ON
```

Books_idBooks = idBooks

WHERE idBooks = Books_idBooks);



```
title
Pranks
The Goal
```

This would be useful in seeing which longer books people are actually interested in. At least interested in enough to have saved them for later reference.

TP_Q10: Which author has the highest rated book?

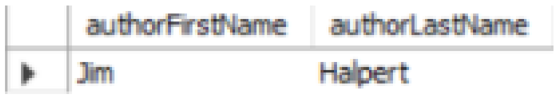
SELECT authorFirstName, authorLastName

FROM Authors

JOIN Books ON idAuthors = Authors_idAuthors

JOIN Bookshelf ON Books_idBooks = idBooks

WHERE userRating = (SELECT MAX(userRating) FROM Bookshelf);



```
authorFirstName   authorLastName
Jim               Halpert
```

This would be useful to know if you were looking for critically acclaimed authors as many people choose their books that way. Also, this allows us to see which authors we want to have more of their work on our app.

Query Matrix:

	Query 1	Query 2	Query 3	Query 4	Query 5	Query 6	Query 7	Query 8	Query 9	Query 10
Multi-Table Join	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
Subquery						**x**		**x**		**X**
Correlated Subquery				**x**					**x**	

Group By		**x**								
Group By with Having			**x**							
Order By	**x**						**x**			
In or Not In					**x**	**X**				
Built in Function / Calculated Field		**x**					**x**	**x**	**x**	**x**
Regexp	**x**									
Not Exists				**x**		**x**				

Database Name: ts_29701_7