

LIS 501

Final Project Report

GroupThing

(Jared Dunn, Renata Sancken, Will Kent, Mary Gen Davies, Paul Friesen, Elin Bammerlin)

Introduction

In GroupThing's analysis of LibraryThing's structure, we were able to apply our knowledge of Entity-Relationship diagrams, metadata, and FRBR on a very real, practical level. There was constant discussion of how different elements of LibraryThing relate to others, how a database functions (and functions most efficiently), and how the elements of FRBR (Work, Expression, Manifestation, Item) were at play within the structure.

Drafting the LibraryThing's Entity-Relationship Diagram was an enjoyable challenge for GroupThing. As a group, we had to come to many agreements upon what should or should not be included in the E-R Diagram as well as how that information relates to each other. As we will elucidate further in our process below, we had to make some concessions and agreements as a group, but in the end were all satisfied with the final product. We believe that, while admittedly not exhaustive, our E-R Diagram accurately represents the LibraryThing database at a conceptual level.

Analyzing the current and possible use cases for LibraryThing was another useful way to dissect the intricacies of the LibraryThing database, what it allows and what would be possible to allow in future. Scrutinizing the database in this way, forced us to figure out how the database functions and what could be changed or augmented to make LibraryThing even more useful for its members.

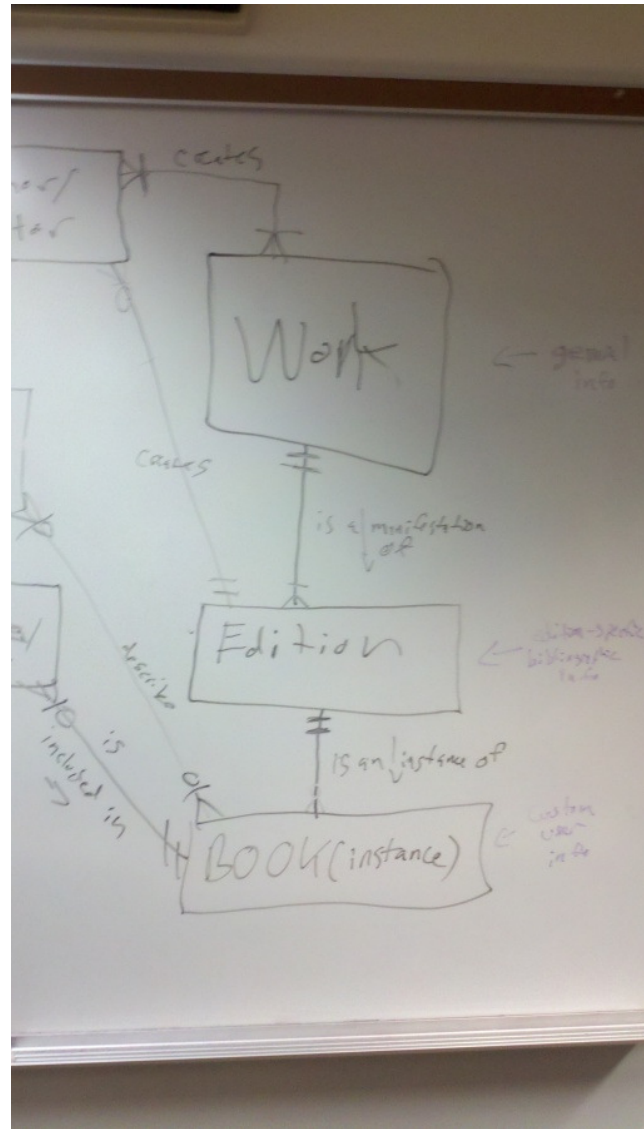
Process

November 8

Our group met after class on November 8th to discuss the E-R diagrams we had made individually. We started off with a group show and tell, where each group member explained their provisional diagrams. As a group, we discussed commonalities among diagrams as well as unique features. Paul had created the most complex diagram, which featured a diagram of the LibraryThing website itself as well as the entities of the site (such as authors and books). Everyone had book, author, tag, and user on their original diagram. Other entities various group members included were edition/copy, group, project, review, image, DVD, and CD.

We decided for the purpose of this project to keep our group diagram simple, and decided to focus only on entities related to the Book-User relationship and not the social aspects of the LibraryThing website. We also decided to focus solely on books rather than DVDs and CDs, for simplicity's sake and because LibraryThing was initially designed for and is primarily used for books. Having made that decision, we settled upon a final list of entities to include: author, work, edition, copy, user, library (referring to all of a user's added books) and tag. We spent some time clarifying the differences between a work (for example, 1984), an edition (for example, the Plume 60th Anniversary edition), and a copy/instance (for example, Jared's copy of the 60th Anniversary edition that he added to his own LibraryThing library). We wondered if author was always a required element, thinking of books like the Bible or anonymous works like Primary Colors. However, we searched for them on LibraryThing and found that they were always listed with some kind of creator, such as "Anonymous" or the publisher of a specific edition of the Bible.

We also spent awhile discussing whether "tag" was an entity or an attribute, and what it is applied to in this context. On one hand, it seems like an attribute since it describes an entity. However, Paul and Jared made a case that it was an entity because of the way the LibraryThing website handles tags--they are stored in one database location together and are added by users to their book copies, but are also associated with authors and works. We determined that the author and work tags come from the same pool by searching for an author with only one prominent work (Harper Lee, To Kill a Mockingbird) and noting that the tag clouds for its work page were the same as the ones for her author page. Thus, we decided to treat tags as entities. Then we debated whether reviews and ratings should also be treated like tags, but decided to treat them as attributes of copies since reviews are not re-used--a book may have many unique reviews, but each review only applies to one book. One tag, however, may be applied to many items.



Brainstorming the FRBR Tree

We also began listing other attributes that we had come up with individually. We pulled up LibraryThing to look at different pages for books, editions, authors, and tags to cull all the possible attributes.

Finally, we discussed use cases. Each group member came prepared with a few use studies they had thought of before meeting. We chose three that we were excited about, that made sense to us given the diagram we had constructed, and that represented a spectrum of LibraryThing's capabilities. At the end of our meeting, we reviewed the project description on Moodle to make sure we had covered all of the requirements. We also divided up tasks for each person to prepare a draft of to share with the group before the presentation.



Brainstorming for the ER Diagram and Database Tables

November 15

After class on the 15th we discussed our 20 min presentation, the narrative we will have to turn in as well as the components of these presentations/narratives that we still have to finish. Our diagrams were essentially complete at this point, and Jared and Paul had used their knowledge of graphic design and Visio to make some visually appealing diagrams.

For the narrative we debated where the ratings attribute should belong. Some thought "edition" while other thought it would fit better as a "work" or "book." We decided that the average rating would go with the "work," although the user adds a review at the level of the "book."

We also addressed how to incorporate diagrams into the paper and how to incorporate our use cases into the diagrams. Jared volunteered to create customized diagrams showing only the parts of LibraryThing needed to discuss each use case.

We still needed to compile, polish and edit the narrative, create some slides and props for the presentation. We decided that the presentation would be interactive and include involvement from the rest of the class. We wanted to make sure everyone is engaged with our presentation, despite seeing four similar presentations. We considered creating a hologram, but it that appears to be currently unfeasible.

We also decided that in our presentation, Paul, presenting the database diagram, will follow Jared's ER diagram which means that we can further connect the class to each other with thread to illustrate the diagrams. We will need yarn and some signs for our interactive diagram.

We planned to meet to rehearse our presentation after the next class.

December 1

We met after class today and did a dry run of our project. Paul introduced a song he had written for our presentation, which everyone loved. We practiced all the different parts of our presentation (including the yarn throw). We addressed the order of the presentation and started tackling the phrasing of what we would say and how we would describe it. We wanted to make sure that we explained everything thoroughly and appropriately, keeping in mind that our audience would have done similar work but with possibly different conclusions.

A refined list of what we still need to do was drawn up. We selected BOOK and USER as entity examples, because the connection is very concrete and will work well for our class project. We still need to choose Attribute examples.

We also need to add a few more slides to the presentation and submit polished explanations of the various parts of the project for the narrative. Mary Gen volunteered to finish the PowerPoint. Renata offered to make butterflies out of arts and crafts supplies.

December 4

We met one last time to rehearse for our presentation. Will and Paul decided to share the task of presenting the database diagram, and spent some time discussing how to best present that information. We set our final presentation order:

Renata: Introduction
Jared: E-R diagram
Paul and Will: database diagram
Mary Gen: Tagmash use case
Jared: Custom metadata use case
Will: Margin note use case
Elin: Conclusion
All: Paul's Yesterday song
Q&A

We went through our presentation once, taking pauses to clarify and make suggestions for each other. We added a few slides to the PowerPoint so that Will could follow-up on Paul's part of the presentation with a highlighted slide. We also added the lyrics to Paul's song to a few slides to the audience could follow along. We then did a "dress rehearsal" of our presentation with no stops. It came in a little longer than 20 minutes but we felt that when people actually present, they tend to talk faster and our presentation would be fine when we were actually giving it to the class.

We decided to dress in black and white for the presentation to give our group a cohesive look. We made sure that someone was specifically responsible for bringing the PowerPoint presentation and props to class. We also discussed a strategy for hiding Paul's guitar, so the song would be a surprise. We finished the evening feeling very prepared for and excited about the presentation.

Entity-Relation Diagram

The E-R diagram covers how the database is structured and how data is stored, moves through it, and is altered and recombined to meet the needs of LibraryThing and its users. We have limited ourselves in this diagram primarily to the interaction of users with books, and concentrated the most on the FRBR-like handling of book data and what kinds of functionality this allows and how.

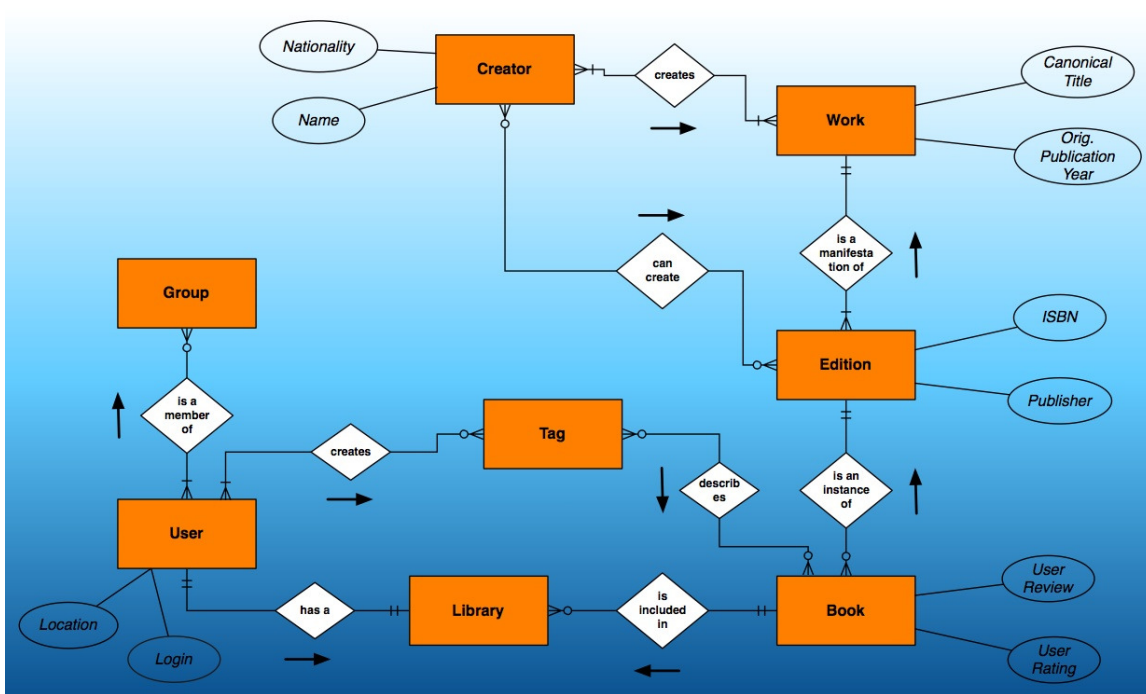
We started with the "User" entity for this diagram, and worked our way outwards from there. "Groups" was included as a sort of placeholder for or gateway to the social aspects of the site. "Library" is also a placeholder of sorts, and masks quite a bit of complexity in terms of how collections, privacy, and related issues are handled.

In setting out to model how book data works in the database, we thought both about the FRBR model and the different types of data and uses on the site and how the two might correspond. We noted that users can edit almost every piece of data related to their books, and that thus this data must have its own home. We also noted that each book had an overarching "Work" page, which has lots of information about the work itself, some of it user-provided, and some of it aggregated from other parts of the site. Finally, we noted that the site aggregates information for each different edition of a work, and uses that data to initially populate a user's books. Even though there aren't pages for each individual edition on the site, the data is still there, and must live somewhere. Thus we decided on a three-tier approach to entities in the FRBR model of the book data: "Book" for the editable data on individual user's books, "Edition" for the aggregated bibliographic data from places like Amazon and the LOC, and Work for user-submitted data on the work, as well as machine-created data like the combined ratings, tags, reviews, etc. that work their way up the chain from all of the individual user's books.

This structure allows the site to both meet user's needs for cataloging flexibility and its own needs for a canonical dataset as a union catalog and recommendation engine. The division of the data allows information associated with each of these functions to be stored independently and recombined and altered only when it meets the needs of the site and the users.

Additionally, we decided that “Creator” must be a flexible entity that accounts for multiple roles and relationships. A “Creator” can be individually responsible for a “Work” as an artistic entity, and also contribute to an edition as a translator, editor, co-author, etc.

Finally, after some debate, we decided to keep “Tags” as an entity to themselves for efficiency and compartmentalization reasons, because having them as an attribute under users or books would require searching those entire tables to generate tag clouds or search by tags, while having a “Tag” entity with foreign keys pointing to users and works would allow you to simply search the “Tags” table for those relationships.

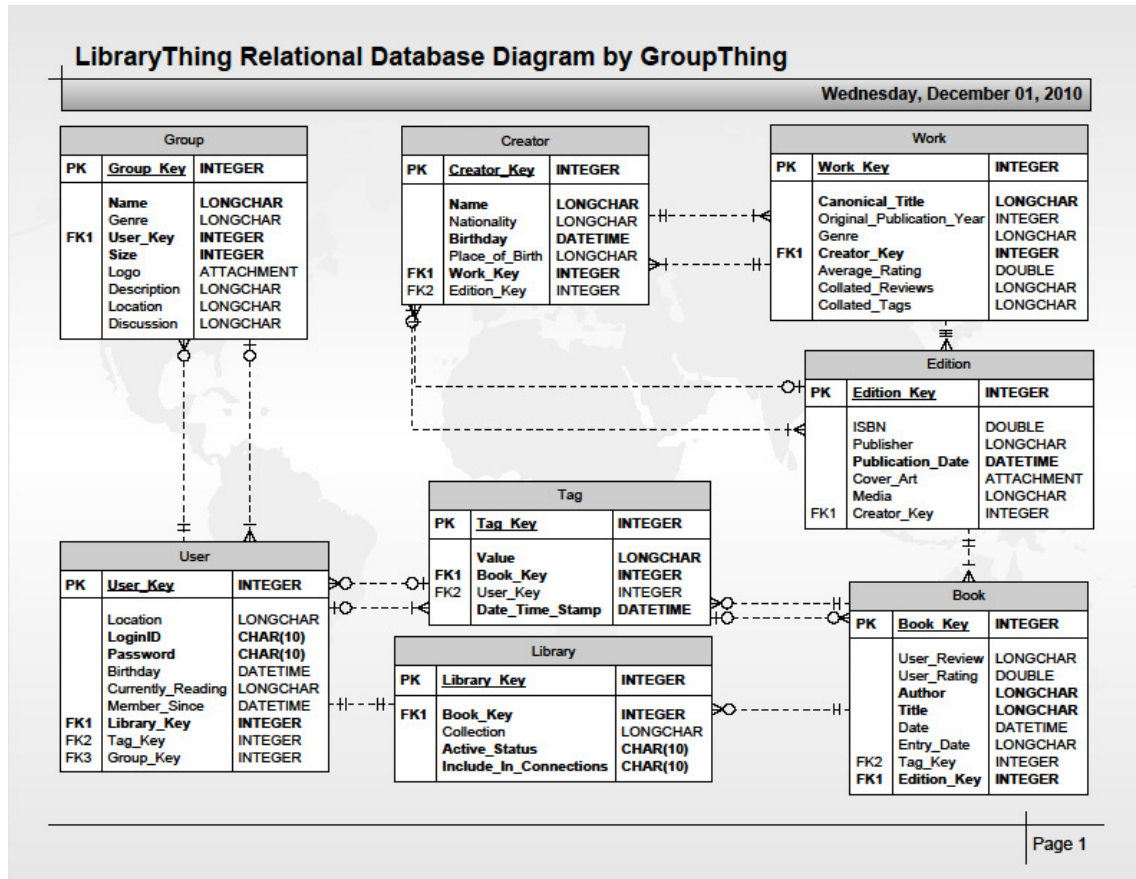


Database Tables

We came up with our detailed entity relationship diagram by modeling it after the simple entity relationship diagram. Using Microsoft Visio lent itself to several other decisions with this diagram. Attributes were easily added. We chose a primary key for each table as well as several important attributes. Our goal was not an exhaustive list, but attributes that highlighted the significance of each entity. On our diagram, bold text signifies attributes required for the entity.

Adding cardinality was problematic. We could not connect the tables exactly the way we wanted using one join line so we ended up using two relationship joins for each table. After consulting with a more experienced database wiz, we found out that Visio was actually trying to force us to use junction tables. However, since we did not get into that kind of detail with this project, we simply settled on two lines for joining each table. Foreign keys were automatically added when tables were joined together.

Adding data types was easy, especially since several of the members of the group were quite familiar with data type terminology. We selected attributes and chose the desired data type from a drop down list. Using Visio, it was easy to make a professional looking entity relationship diagram.



Use Case #1 - Custom Metadata for Each Book in a User's Library

A common user need is the ability to customize their environment and tools to fit their personal needs. Where users are interacting with canonical datasets, this need presents unique challenges. Such is the case with LibraryThing, which has missions both at the user level (as a personal catalog) and at the site level (as a union catalog of sorts and a recommendation engine).

Use case #1 is the ability to create custom data for every field needed to catalog each individual book in a user's collection. LibraryThing initially compiles and provides metadata for a given book from a variety of sources such as Amazon and the LOC, but that data is only complete for newer books, and falls far short of being able to completely and accurately catalog all books that the site's users might wish to.

If a user has a book that predates the ISBN/electronic data era, or an unusual edition of a newer book that doesn't exactly match any in the sources LibraryThing draws upon, the

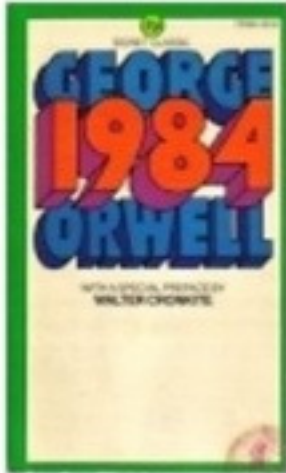
user is then stuck with incomplete or inaccurate data in their personal catalog, and LibraryThing's effectiveness as a personal catalog is degraded.

However, if they simply let the users change the existing data in the database to match their own books, then the canonical data that describes the much more common versions of those books is altered and lost, and Librarything's effectiveness as a recommendation engine and union catalog is degraded.

LibraryThing solved this dilemma by creating extra levels of abstraction to handle the interaction of user and canonical data, along the lines of FRBR. We have chosen to call these three levels the Work, the Edition, and the Book. The Work is mostly a container that provides a place to name and distinguish the creative work in question as a distinct entity and to collate information about all of the different editions that are manifestations of it. The Editions are where the canonical cataloging data for each manifestation lives, with commercial information such as publisher, publication date, physical dimensions, and so on stored here.

To protect that information and still allow the user to change it for their own items, the Book entity was created, with attributes corresponding to the metadata fields (such as title, author, cover art, publication date, tags, etc.) from the Work and Edition levels. Many of these are initially populated from the existing data under Work/Edition, but the user can now edit these to suit their own needs.

Most of these edits stay at the book/user level and this protects the integrity of the canonical data in Work/Edition. However, some edits that are vital to the social aspects of the site, such as tags, reviews, and ratings, are allowed to travel up the chain and are collated at the level of the work. This structure allows data to be used, created, edited, aggregated, and to move between the different levels of abstraction while maintaining the integrity of each different type and intended use of the data. Users can now make all the changes to their cataloging data that they desire, but only the changes that Librarything can make use of are allowed to propagate to the rest of the database.



1984

by George Orwell

Members	Reviews	Popular
36,193	467	10

- Main page
- Edit book**
- Details
- Change cover
- Member reviews (467)
- Recommendations
- Descriptions (7)
- Members
- Conversations (687)

Work details | Book details

Cancel Save

Title **1983** !?

Author **Orwell, George**
Last, First

Tags **20th-century, british, dystopia, epistemolog**
Separate with commas, like "history, milita
current tags: 20th-century, british, dystopi
language, modernist, novels, ontology, ov
technology, totalitarianism, trust, violence,

Collections Your library
show all edit collections

Use Case #2 – Margin Notes

After analyzing LibraryThing our group was impressed by the breadth of their collection and their emphasis on a collective conscious understanding of books and literature. One way that LibraryThing could enhance this aspect of its user experience would be if LibraryThing created a feature that allowed users to "write in the margins" electronically of the books they read. Since LibraryThing does not have eBooks yet, we thought it would make sense if the "margins" section (not unlike a Facebook wall) were attached to the book. It would relate to a book much like a tag would. It would describe a book which is an instance of an edition which is a manifestation of a work. This way, the margins could be uniquely generated, but could be findable through multiple editions of that work. Also like a tag, margin notes would contain their own sets of attributes (user, date, rating, text, duration, references).

Users could take advantage of this feature in a number of ways. People could create discussions around the same passage, analyze overlooked passages, compare and contrast the same passages through different lenses, and record their reactions/feelings around

passages. We think that this would bring a new dimension to the works and increase the communal feeling of LibraryThing.

The screenshot shows the LibraryThing website interface. At the top, the 'LibraryThing' logo is displayed in a large, stylized font. Below the logo is a navigation bar with buttons for 'Home', 'Profile', 'Your books', 'Add books', 'Talk', 'Groups', 'Local', 'More', and 'Margins'. A secondary navigation bar includes 'Users', 'Text', 'Comments', 'Duration', 'Reviews', 'Statistics/Memes', 'Tags', and 'References'. The main content area is titled 'THE WONDERFUL WIZARD OF OZ'. On the left side, there are five comment boxes, each with a header 'Comments groupthing | December 6, 2010 6:38om'. The first comment reads: 'I think the imagery here is a lot like *The Grapes of Wrath* with everything being burned and the blistered paint. I wonder if he read this before writing that book?'. The second comment says: 'This part of the book is slow!!'. The third comment is a longer one: 'For class we will be analyzing Baum's commentary on the economy. For class, you should pay attention particularly to the second and third paragraphs of this page. From there start your essays which should include excerpts from your journal entries. Also note the character descriptions and how they may or may not parallel characters Dorothy meets in Oz. See if there are any historical characters from the era that meet these descriptions.'. The fourth comment states: 'I agree with groupthing's comment about Baum being an avid socialist. If you look at the utopic parts of the narrative, you will clearly see that a centralized order and hierarchy is in control (see: *Oz, description* and *Kansas, description*). The descriptions here characterize this flawlessly.'. The fifth comment says: 'Look the most interesting part of the book is not what'. On the right side, the text of the book is displayed in a serif font. The visible text includes: 'little cracks running through it. Even the grass was not green, for the sun had burned the tops of the long blades until they were the same gray color to be seen everywhere. Once the house had been painted, but the sun blistered the paint and the rains washed it away, and now the house was as dull and gray as everything else. When Aunt Em came there to live she was a young, pretty wife. The sun and wind had changed her, too. They had taken the sparkle from her eyes and left them a sober gray; they had taken the red from her cheeks and lips, and they were gray also. She was thin and gaunt, and never smiled, now. When Dorothy, who was an orphan, first came to her, Aunt Em had been so startled by the child's laughter that she would scream and press her hand upon her heart whenever Dorothy's merry voice reached her ears; and she still looked at the little girl with wonder that she could find anything to laugh at. Uncle Henry never laughed. He worked hard from morning till night and did not know what joy was. He was gray also, from his long beard to his rough boots, and he looked stern and solemn, and rarely spoke. It was Toto that made Dorothy laugh, and saved her from growing as gray as her other surroundings. Toto was not gray; he was a little black dog, with long, silky hair and small black eyes that twinkled merrily on either side of his funny, wee nose. Toto played all day long, and Dorothy played with him, and loved him dearly. Today, however, they were not playing. Uncle Henry sat upon the

Use Case #3 - Tagmash

The tags allow for many different use cases for LibraryThing users. One use, in particular, is for teachers to be able to find relevant books related to very specific lesson plans or projects. For instance, should a teacher need to find children's book about butterflies, the teacher can enter both terms ("butterflies" and "children's") as a search under "tags." The resulting "Tagmash" retrieves all the books containing both tags.

The database is able to retrieve these results by performing a search for both the desired terms ("butterflies" and "children's") within the tag entity. The resulting tagmash would display all items in the database that are attached to both tags, thereby producing the desired search result for the teacher. Because "tags" itself is an entity, not an attribute

of another entity, it is possible to be able to search within “tags” easily, pulling in the other entities that are attached to those particular values of “butterflies” and “children’s”.

The screenshot shows the LibraryThing website interface. At the top, there is a navigation bar with links for Home, Profile, Your books, Add books, Talk, Groups, Local, More, and Zeitgeist. A search bar is located in the top right corner. The main content area is titled "Tagmash: butterflies, children's". Below this title, there are sections for "Mashing tags" which lists related tags like "butterflies" and "children's" with their respective counts and descriptions. A "Top works (248)" section lists several books with their authors. To the right, there is a "Try another tagmash?" section with a search input and a "Mash" button. Below that is a "Related tags" section showing a grid of tags such as "animals", "board book", "bugs", "butterflies", "caterpillar", "caterpillars", "children", "children's", "children's books", "children's literature", "classic", "counting", "Eric Carle", "fiction", "insects", "nature", "Newbery", "non-fiction", "picture book", "poetry", "school", "science", "short stories", "spring", and "young adult". At the bottom right, there is a "Related tagmashes" section listing combinations of tags and their counts, such as "butterflies, picture book (141)", "butterflies, children (138)", "butterflies, fiction (101)", "butterflies, children, fiction (92)", "butterflies, non-fiction (81)", "children's, insects (68)", "insects, picture book (59)", "animals, bugs (58)", and "bugs, insects (57)".

Conclusion

As this narrative has highlighted, this was a collaborative process for GroupThing. A few take-away lessons from our process include how we ultimately decided that different entities relate to each other, how these relations uphold and break the FRBR framework, and how the details of this organizational system work (attributes, data types, keys, etc.).

From this analysis we were able to debate concepts introduced to us in class and apply them to a real world scenario. Regardless of how accurate we were to actually guessing how LibraryThing is organized, every member of our group has a better understanding of how LibraryThing organizes data and how users access it, and how databases work to facilitate the organization, access, and manipulation of information.