**Final Project Reflection**

1. **Introduction**

Technology has allowed us to access and share information easily with the entire online world. An application exists for even the most trivial of tasks, so it was shocking when our desk research failed to find a tool that we deemed beneficial for our target audience. We were unable to find a free, semi-complete database that included books with the necessary information for library managers, educational professionals and brick and mortar stores to select and organize books and their awards. As a result, our team decided to create a database to organize books and the awards granted to them.

1. **Database Description**

The database is fully functional for our target users: book-sellers, book-readers, library managers, educational professionals, and brick and mortar stores. It is structured such that books are receiving awards, rather than their author. This is the most common case, though there are exceptions in real life. One such case is a Nobel Prize awarded to an author for a piece of literature. For our purposes, although the author was honored, the book contains the award and the author is linked to the book.

Our final tables include:

Books

* Includes the information unique and exclusive to each book and connects to all tables

Authors

* Includes the full names of authors

Book\_authors

* Linking table between books and authors. Allows for the creation of multiple books per author, as well as multiple contributors for a single book

Publishers

* Information about publishers and their locations

Awards

* Names of awards that have been received in the database

Award\_books

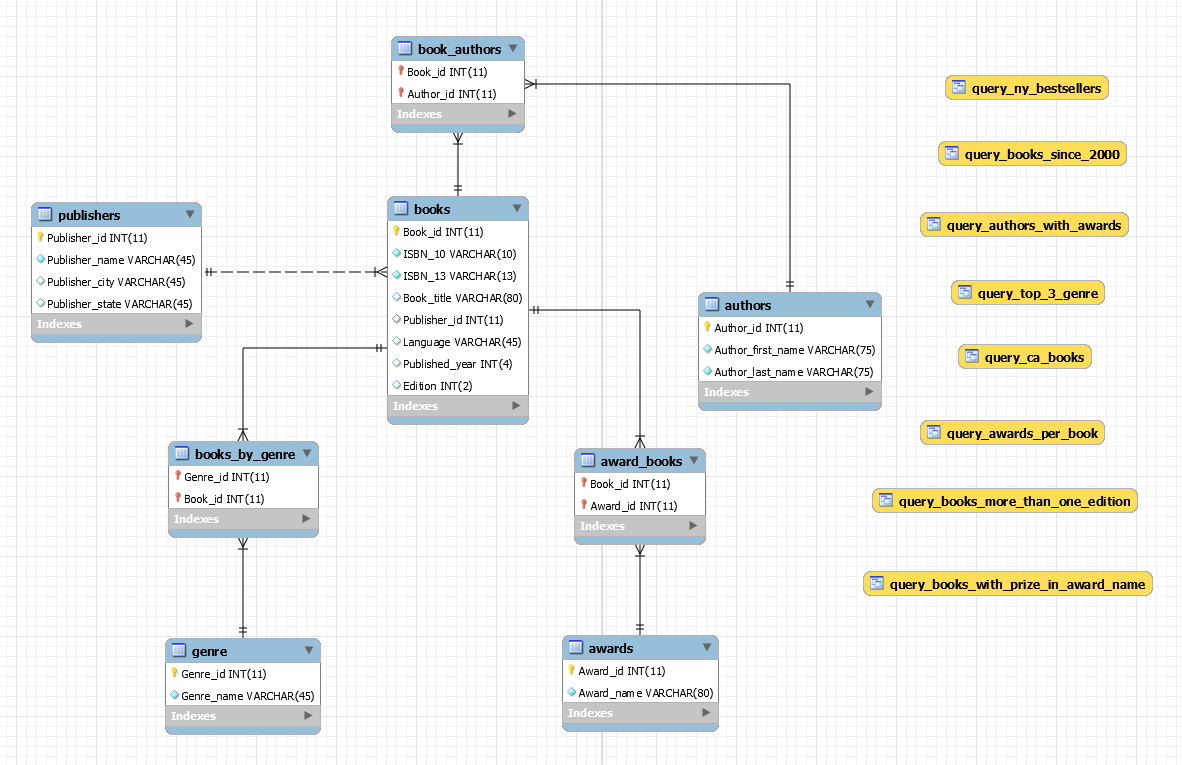
* Allows for the creation of multiple books having the same award, as well as books receiving multiple awards

Genre

* Categorization of the genre

Book\_by\_genre

* Linking table between books and genre. Allows for books to have multiple genre categories as well as for many books to have the same genre

**Logical Design  
**

During the logical design process. We arranged our data into logical relationships. We drew our entities on whiteboard with their attributes. This was our first blueprint to understanding the relationships between our entities. Furthermore, This process proved creating a database will be more challenging than we first thought. We realized some of the entities we have were really the same, and others had many to many relationships which would require a joining table.

**Physical Database Design**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Column** | **Data Type** | **Notes** |
| authors | Author\_id  Author\_first\_name  Author\_last\_name | INT(11)  VARCHAR(75)  VARCHAR(75) | Primary Key |
| awards | Award\_id  Award\_name | INT(11)  VARCHAR(80) | Primary Key |
| genre | Genre\_id  Genre\_name | INT(11)  VARCHAR(45) | Primary Key |
| publishers | Pubnlisher\_id  Publisher\_name  Publisher\_city  Publisher\_state | INT(11)  VARCHAR(45)  VARCHAR(45)  VARCHAR(45) | Primary Key |
| books | Book\_id  ISBN\_10  ISBN\_13  Book\_title  Publisher\_id  Language  Published\_year  Edition | INT(11)  VARCHAR(10)  VARCHAR(13)  VARCHAR(80)  INT(11)  VARCHAR(45)  INT(4)  INT(2) | Primary Key  Foreign Key |
| book\_authors | Book\_id  Author\_id | INT(11)  INT(11) | Foreign Key  Foreign Key |
| award\_books | Book\_id  Award\_id | INT(11)  INT(11) | Foreign Key  Foreign Key |
| books\_by\_genre | Genre\_id  Book\_id | INT(11)  INT(11) | Foreign Key  Foreign Key |

**Sample Data**

While designing the database we decided to create our sample data manually so we would not run into any issue while importing the data. We divided the task of creating an excel sheet with all the data needed by assigning 10 books per team member. This method gave us a total of 60 books. Our sample data contains 8 tables with 3 of them being a joining table. Books, publishers, genre, award, authors, books\_by\_genre, award\_book, book\_authors. Once all the required data was inputted into our excel sheet we made sure that all of our data types matched the datatypes we designed the database with. Next, we created our ids, this step was a critical step in our project because if we made any mistake while creating the ids we would get a wrong output from our queries and incorrect links in our joining tables.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| View Name | Req. A | Req. B | Req. C | Req. D | Req. E |
| query\_awards\_per\_book | X |  | X | X |  |
| query\_books\_since\_2000 | X | X |  | X |  |
| query\_top\_3\_genre | X |  | X | X |  |
| query\_CA\_books | X | X |  |  | X |
| query\_NY\_bestsellers | X | X |  | X |  |
| query\_books\_more\_than\_one\_edition | X | X |  |  |  |
| query\_authors\_with\_awards | X |  | X | X |  |
| query\_books\_with\_prize\_in\_award\_name | X | X |  | X | X |

**Views / Queries**

On our first run through, we wrote queries mainly revolving around our “books” table as our primary source of data. All of our queries were very similar to one another and appeared as “top books in every genre,” “top books for every publisher,” and “top books for every state.” Although we fulfilled all of the view/query requirements, the relationships between all of our look-up tables were not being utilized. This was an oversight caused by our naivety in thinking our database could only be used in a few different ways.

After speaking with the professor and Jonathan Bui (AMP), we were able to refine our queries through two different process strategies. The first was using our ERD as a basis for our queries and making at least one connection between every table. Instead of just connecting books with each connecting look-up table or “limb”, we began to make connections between the tables such as “top genre published in each state” and “award-winning authors for each genre.” These are questions we did not consider that could even be gathered by our database. The second strategy that Jonathan suggested to us was to make queries based on the viewpoint of an author. We had considered views from library or book store managers but did not think about the perspective of a competing author looking for market research. One query that could satisfy this thinking is “top author or genre in each state.” Authors can look at other books for sources of inspiration in the same state as them through this database.

1. **Changes From Original Design**Our original design included many more tables and involved using the Amazon Books API We dropped the idea of using an API realizing that this was a quantity of data unmanageable during the time we had to complete our project. This idea would have also potentially cost a team member several hundred dollars, as Amazon implemented a data cap on their information which results in a fee if exceeded.

We reduced the tables after realizing that we had many cases of redundancy. Audiobooks, digital books/ebooks, children’s books, and printed books were all congregated into a single entity since we concluded that, for our purposes, these were all instances of the same content no different than a paperback versus a hardcover book. Because we no longer have a distinction between formats of each book, Amazon’s shipping information such as weight, length, width, height, are no longer necessary. Subsequently, information such as cover image, illustrators, narrators, and length of the book in hours was removed due to the format specific nature of those values. Price is no longer included, as this varies between book formats and seller.

1. **Lessons Learned**

As observed by Team ARES, our peer reviewers, our first lesson was just how long of a process this could be. We were told by their team that we were biting off more than we could chew by planning to use an API. Not only did this plan prove to be financially inaccessible to the group, but it would have involved a great amount of cleaning to fit our needs and would result in a plethora of inutile data.

We struggled initially with the connections between our tables and understanding the logical functionality and connections of our database. We discussed our table relationships with Professor Duffy, and she helped us better understand the way our linking tables work. The linking tables allowed for repeat award issuance and books with multiple awards, authors with multiple books and books with multiple contributors, and a single book to be under multiple genre categories as well as a single genre to be assigned to many books.

Following the redesign of our database, we lost touch with the solutions that the database could provide. As mentioned previously, we consulted with our AMP Jonathan Bui, who helped us come up with new questions for our queries by shifting our perspective to the other components involved in the database, such as authors looking for market research or states looking for their literary demographic information.

One of the discoveries we had while working with MYSQL Workbench is: if there is any dashes within a table name, the command would not be executed. Therefore, we have modified the table names ISBN-10 and ISBN-13 to put underscore between the letters and number instead of dashes. The modification has resolved the issue that at first we weren’t aware of.

1. **Potential Future Work**If we were to continue working on this database, ideally we would use Amazon’s API to source our data. This would allow us to have a much larger and richer database. As we learn more about our user needs, we could potentially include more queries to accommodate them.

In the very near future, another possible extension of our database may involve virtual reality. It would be interesting if we could visualize our database in a virtual environment where you could see all the tables, as well as how they connect to one another. The books table could show the covers of each book, where as you may one day be able to skim through the authors table that has profile pictures for all authors. This type of experience could also lend itself to teaching database management in a different way. Instead of using a computer to create an ERD diagram, virtual reality could add a physical aspect where students can interact with the way a database is organized.