Andrew Assoh

Gather Requirements for Online Book Store:

To gather information for my project, I looked up any bookstores in New York that had websites. I contacted a few of them through email to learn how they keep track of their information. I reached out to Strand Book Store, Astoria Bookshop, and Book Culture, and I received some useful information. The staff shared details about the types of information they track and the reasons behind it. For example, lifetime sales and rate of sales are used to determine if books are in demand. This helps them decide whether to order more copies of a book or return it. Some stores divide their sections by genres, some by sub-genres, and some by region. Based on this, I decided to divide my sections by sub-genre. They also keep track of customer information, vendor and publisher information, invoices, and more.

My database includes tables to manage books, authors, customers, sales, and inventory. The Books table stores information like title, ISBN, binding, list price, publish date, and links to a publisher and section. The Authors table has each author's first and last name. The BookAuthors table connects books and authors since a book can have more than one author.

Publishers and Vendors each have a name and ID. The Inventory table tracks which vendor supplied each book, the stock quantity, date received, and lifetime sales. Sections are used to organize books by sub-genre.

The Customers table includes customer names, shipping, and billing addresses. Invoices track purchases using a unique order number and connect to customers. InvoiceItems links invoices to the books being purchased.

Business rules: ISBNs and order numbers must be unique, books must be either hardcover or paperback binding, publish dates can't be in the future, and invoices must include at least one book. Customers must have names and addresses, and books are grouped by sub-genres.

Document Process:

<u>Creating my database:</u>

- I used the lookup wizard to create a set of limited options for the Binding and SectionName fields. The Binding field has paperback and hardcover options. The SectionName field includes different sub-genres. The sub-genres I used were based on the examples given to me by the staff that I emailed.

- I made sure to name fields clearly, such as using AuthorFirstName and CustomerFirstName, to avoid confusion between similar fields across tables.

- I created a junction table for books and authors to represent a many-to-many relationship. A book can have many authors, and an author can write many books. BookID and AuthorID are a composite key for the table, and are foreign keys referencing other tables.

- The junction table Invoice items creates the many-to-many relationship between books and invoices. One book can be in many invoices, and an invoice can have many books. InvoiceID and BookID are a composite key and are foreign keys referencing other tables.

- I created a query to calculate the rate of sale.

- The last step was exporting each table into MySQL. First, I created a database in MySQL, then I used ODBC to connect it, and lastly, I exported each table into MySQL. In the schema, I was able to right-click on each table, choose "Send to SQL" editor, then choose "Create Statement." This generated the CREATE TABLE scripts based on my database. The statements didn't have primary and foreign key constraints, so I added those as well. Lastly, I needed to change the Binding and SectionName datatypes to enum so I could have a list of predetermined values similar to my Access database.

<u>Challenges:</u>

- Creating relationships: I ran into an error that said "no unique index found for referenced field of the primary key table" when trying to create relationships. I resolved this by making sure every primary key was indexed and set to no duplicates. When I used "Show All Relationships," some tables were disconnected, so I decided to manually create them by dragging each primary key to its corresponding foreign key.

- Using ISBN: I initially wanted to use ISBN as a unique identifier, but it needed to be set to short text rather than auto number because ISBNs can have dashes. I believe that this contributed to my difficulty with making the relationships work, so I decided to create a new primary key field called BookID.

- Calculating rate of sale: I initially tried to make RateOfSale a calculated field, but I kept getting errors. Instead, I created a query that calculates the monthly rate of sale by dividing LifetimeSales by the number of months between DateReceived and the current date. I used the DateDiff() function: RateOfSale: [LifetimeSales] / DateDiff("m", [DateReceived], Date())

- Entering sample data: I was having trouble entering sample data into the database. I was getting the error "can't change because it's related to another table." To work around this, I needed to fill out the sample data in the independent tables that don't depend on another table's primary key. I then entered foreign key values that already existed when entering

data into the dependent tables. These tables were Authors, Publishers, Sections, Vendors, and Customers. The next step was to add data to the Books and Invoices tables. Lastly, I added to the BookAuthors, Inventory, and InvoiceItems tables. For each record, I needed to add foreign key data before I could fill out the rest of the fields.