Application Development Case Study

Diane Harrell

For IST 423 Spring 2024

Contents

Background…….. 2

Project Scope / Parts……..3

Challenges……4

Benefits……5

Appendix A - ER Diagrams……6

Appendix B - SQL Queries……7

Appendix C - Data Dictionary……9

Background

Beyond Borders and Books is a local Bookstore with an attached cafe. The bookstore sells a wide quantity of books, and the cafe sells both breakfast and lunch dishes. They reached out to us to ask us to build a database that will include but is not limited to the following information:

- Book Information (such as Author, Title, Publish Date)
- Author Information (such as Name, Date of Birth)
- Book Sales Information
- Cafe Items
- Cafe Ingredients
- Cafe Allergens
- Cafe Ingredients Sales Information
- Staff
- Staff Hours
- Staff Schedules

This company is having trouble keeping up with all of their information in its current state, and believes a new database will help to alleviate some of the struggle, and keep records in one place. Previously, BBBooks was using an outdated database to maintain its information, which did not include the staff and sales section of the cafe inside of the bookstore. Unfortunately, the old database is unusable and we will need to restart.

Project Scope / Parts

For this database I planned to create a database that would allow the company to easily view their information, with the ability to create schedules and the ability to reorder supplies from within the database - or at least be able to tell what they needed to rebuy from a glance. I also intended to make the database have the capability of showing what books are in stock, along with displaying information about each book within the bookstore.

The intended purpose of this database is to better organize BBBooks inventory and information, and leave them with the ability to more neatly and efficiently organize their information simply.

I intended to build this database within SQL and leave the company with the ability to edit the database, while also having a viewer only interface to ensure only individuals with the proper clearance could edit the database. This stops any unintentional editing, and also limits the amount of traffic inside of the actual database interface.

This plan was accomplished by creating a series of tables, linked together with primary and foreign keys that displayed all of the information needed. The project was completed within SQL, and is able to be freely edited to add more books, ingredients, schedules, etc.

Challenges

Challenge 1:

The main challenge was figuring out where to start. Many of the tables are linked together in many different ways, and it was important to figure out which tables needed to be created before others, and how to maintain all of the information.

Solution:

To combat this problem I created an ER Diagram which helped me to visualize what needed to be done and when.

Challenge 2:

I ran into a few issues in which I forgot to add a foreign key specification to the table. I then had to figure out the code for adding in extra bits of information to SQL tables.

Solution:

To solve this I had to use the following code:

```
ADD CONSTRAINT fk_salesid
FOREIGN KEY(salesid) REFERENCES Sales (salesid);
```

This code was altered a few times to fit my needs (See Appendix B 1, 2, and 3)

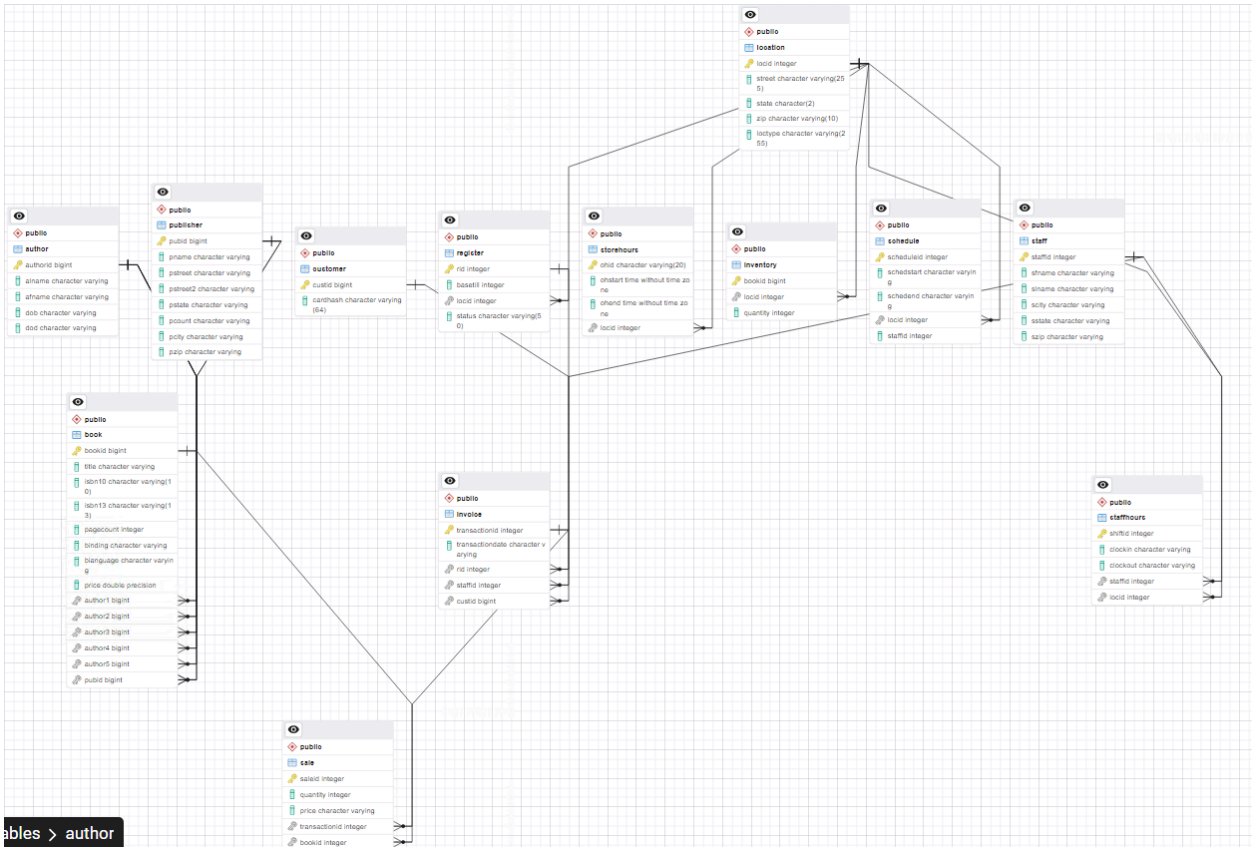
Benefits

This database positively benefits BBBooks by giving them a place in which they are able to have efficient and streamlined operations that let the business quickly adapt to new information like changing market conditions, along with changing clientele. One of the main benefits of this database is that it has improved inventory management. The database now keeps track of all ingredients within the cafe, along with how many are left and when they should reorder them.

The database also helps with capturing sales data, which helps keep track of which items within the store are selling faster than others, and which books the store could consider “popular”. This gives them a data-driven advantage, allowing the company to decide which books they should put on sale, and where to position books within the store.

It also makes it easier for the bookstore to manage a large staff. The database processes payroll, helps make schedules, and ensures all staff are getting fair hours. The database also keeps each individual employee's information safely tucked away in case someone with security access to the database needs to access it.

Appendix A - ER Diagrams



Appendix B - SQL Queries

1

Query Query History Scratch Pad ×

```
1  ADD CONSTRAINT fk_salesid
2  FOREIGN KEY(salesid) REFERENCES Sales (salesid);
```

2

Query Query History Scratch Pad ×

```
1  ALTER TABLE sale
2  ADD CONSTRAINT fk_transactionid
3  FOREIGN KEY(transactionid) REFERENCES invoice(transactionid);
```

3

Query | Query History | Scratch Pad

```
1  ALTER TABLE book
2  ADD CONSTRAINT fk_pubid
3  FOREIGN KEY (pubid) REFERENCES publisher(pubid);
4
```

Appendix C - Data Dictionary

Field Name:	OHID	**Field Type:**	PK
Field Description:	OHID is the operation hours ID. It serves as the primary key for StoreHours		
References:			

Field Name:	LOCID	**Field Type:**	PK
Field Description:	LOCID is the location ID. It serves as the primary key for Location		
References:	Location (PK) StoreHours (FK) Register (FK) Inventory (FK) Schedule (FK) StaffHours (FK)		

Field Name:	RID	**Field Type:**	PK
Field Description:	RID is the Register ID. It serves as the primary key for Register		
References:	Register (PK) Invoice (FK)		

Field Name:	CustID	**Field Type:**	PK
Field Description:	CustID is the customer ID. It serves as the primary key for Customer		
References:	Customer (PK) Invoice (FK)		

Field Name:	ScheduleID	**Field Type:**	PK
Field Description:	ScheduleID is the Schedule ID. It serves as the primary key for Schedule		
References:	Schedule (PK)		

Field Name:	ShiftID	**Field Type:**	PK
Field Description:	ShiftID is the Shift ID. It serves as the primary key for Staff Hours		
References:	Shift Hours (PK)		

Field Name:	TransactionID	**Field Type:**	PK
Field Description:	TransactionID is the Transaction ID. It serves as the primary key for Invoice		
References:	Invoice (PK) Sale (FK)		

Field Name:	StaffID	**Field Type:**	PK
Field Description:	StaffID is the Staff ID. It serves as the primary key for Staff.		
References:	Staff (PK) StaffHours(FK) Schedule (FK)		

Field Name:	SaleID	**Field Type:**	PK
Field Description:	Sale ID is the Sale ID. It serves as the primary key for sale		
References:	Sale (PK)		

Field Name:	BookID	**Field Type:**	PK

Field Description:	BookID is the Book ID. It serves as the primary key for Book
References:	Book(PK) Inventory (FK) Sale (FK)

Field Name:	AuthorID	**Field Type:**	PK
Field Description:	AuthorID is the Author ID. It serves as the primary key for Author.		
References:	Author (PK) Book (FK)		