IOP Conf. Series: Journal of Physics: Conf. Series **1213** (2019) 052008 doi:10.1088/1742-6596/1213/5/052008

Traditional Book Stores Industry Reforming Based on the New Management System

Tianyi Luo

Major in Electrical and Computer Engineer, College of Engineering, The Ohio State University 43201

Abstract: The Book Stores industry has had its pages ruffled over the near half decade. Despite a strengthening domestic economy and rising per capita disposable income, the industry continues to experience decline. The business had negative annual growth since 2013, and there is no sign of stopping this trending. It is highly duo to two main factors, the change of consumer purchasing preference and external competitors rising.
When such a traditional business decided to embrace this new "Internet+" era by opening a new online store to reseize the market share, it needs to adopt a new management system containing technologies such as big data and cloud computing which have performed a significant role in this era. At the same time, to keep pace with the new requirements, both physical and online store needed a simple information management system and database to support their inventory and sales operations. This paper aims to set and design the database to support the operations of this industry.

1.Introduction

The librarian used to be a muscular labor job. They are responsible for recording every book information by hand-writing and keeping an index of all the books. Due to the innovation of database, the work load of the librarian has been tremendously left. The Big data is designed to help people record and possibly use a large sequence of information. A complete database must have the ability to gather, organize and process numbers of information, and be conveniently accessible for users. Thus, the database for the Bit & Books should include as much detail information as possible. Based on the requirements given, a list of entities is modeled in this database. And to better describe the entity, each table includes a list of associated attributes and the level of normalization.

2 Individual Components

(1)Author (First Name, Last Name)

This entity contains a list of all authors related to the database. It may also contain authors who have not written any books in the database. It has two attributes which both of them are the Primary Key, the identity which will be used to different from others, the author's first name(VARCHAR) and last name(VARCHAR). The first name of the author includes middle name or middle initial if applicable. Both entities will be storage as VARCHAR datatype in the database.

(2)Books (ISBN #, Title, Price, Year, Rating, Genre, Quantity)

This entity contains a list of all books that are currently in the database. There are six attributes used to define each one book from the database. The most important one is ISBN (VARCHAR). Since every book's ISBN is issued to be independent, ISBN is set to be the Primary Key for this entity. Title(VARCHAR), price(DOUBLE), and year (INTEGER) are three other attributes relates to the

book entity that none of them can be NULL. What's more, quantity (INTEGER) is the number of books that exist in the database (for sale). It cannot be null be could equal to zero. To organize the database to be more convenient to use by online costumer, there is another attribute, genre (VARCHAR). This is the genre the book most belongs to. Each book can only have one genre in the database, or it can be null, meaning undefined or no genre is available.

(3)Cart (Cart ID#)

This entity contains a list of ID #s for customer carts. It only contains one attribute, cart ID (INTEGER). This is the unique ID of the cart which is also set to be Primary key.

(4)Customer (Cust ID #, First Name, Last Name, Street, City, State, Zip)

With the function to place an order online, each user would have an individual account to access and purchase any book they prefer. This entity contains a list of customers who bought or will buy books. It includes seven attributes, Customer ID(INTEGER), First Name(VARCHAR), Last Name (VARCHAR), and four attributes for four different parts of information about mailing address (VARCHAR). The customer ID is the uniquely identifying ID number and the Primary key used to define this entity. Customer's name is recorded as First Name and Last Name in different attributes of this entity. Also, the database also keep track of customer's mailing address by dividing into four attributes of the customer entity which are Street (VARCHAR), City (VARCHAR), State (VARCHAR), and Zip (VARCHAR).

(5)Publisher (Pub Name)

This entity contains a list of publishers, identified by publisher's name, PubName (VARCHAR), which is also the Primary key.

(6)Review (Rev ID #, Rating, Comment, Timestamp, ISBN*, Cust ID #*)

This entity contains a list of all reviews of books in the database, and each review would have a uniquely identifying ID number, RevID (INTEGER), as the Primary Key of this entity. The review also includes Rating (INTEGER), Comment (VARCHAR), and Timestamp (DATETIME), and two foreign keys, ISBN (VARCHAR) and CustID (INTEGER), to reference the book being reviewed and the customer who made the review.

(7)Transactions (Order ID #, Time, Cust ID #*)

This entity contains a list of transactions and the customer who ordered them. The OrderID is an attribute which is the uniquely identifying ID# for the transaction and the Primary key of this entity. Timestamp (DATETIME) is another attribute that record the time the transaction took place. CustID (INTEGER) register the customer who made the transaction that references Customer entity.

(8)Warehouse (W ID #, Street, City, State, Zip, Name)

This entity contains a list of warehouses where books are stored. WarehouseID (INTEGER) is the uniquely identifying ID# for the warehouse and the Primary key for this entity. Other information about the warehouse are recorded by five individual attributes such as Name (VARCHAR), Street (VARCHAR), City (VARCHAR), State (VARCHAR), and Zip (VARCHAR). Those attributes register the name and the location of each warehouse.

2.1 Joint Components

(1)Books_Author(ISBN*, FirstName*, LastName*)

This table contains a list of books by ISBN and the author(s) who wrote them. A book or author may be listed more than once, meaning a book can have multiple authors, or an author can write multiple books.

(2)Books_Cart (Cart ID #*, ISBN*)

This table contains a list of carts and the books they contain. A cart may contain more than one book, and the same book may be in different carts.

(3)Books_Publisher(ISBN*, PubName*)

This table contains a list that connects each book to a publisher.

(4)Books_Transactions (ISBN*, Order ID #*)

This table contains a list of transactions, and the books they contain. Each tuple means the book is

in the order.

(5)Books_Warehouse(ISBN*, W ID #*)

This table contains a list of warehouses and the books they contain. A book may be more than one warehouse, and a warehouse may have more than one of the same books.

(6)Cust_Cart (Cust ID #*, Cart ID #*)

This table contains a list matching each cart to a customer. Customers may have multiple carts.

3.Normalization

This section describes the level of normalization achieved for each table in the database. All the tables, entities, are designed to have the highest normal form been set as 3NF, the third normal form, which helps database minimizing the storage costs.

Also all the tables are in BCNF (Boyce–Codd normal form) form, thus all non-key attributes are fully dependent on the key. Table such as Author, Publisher, Cart, and all the joint tables would not have any functional dependency inside the entity. Other tables would have all the non-key attributes depend on Primary Key attribute.The team did not use any indexes when they created the database. However, for future queries, it might be useful to index books by their ISBN to increase performance.

4.Database Schema

4.1Relation

To create a fully functional database, the system should include an ER (Entity Relationship) diagram and the relational schemas for the database describes what each view produces, provides sample output for the views, and gives sample SQL statements for transaction.

This section describes the database that the team developed and documents the relationship among entities in the database.

4.2 Entity Relationship Model

After discussing the project requirements, the team constructed the ER diagram shown on the following page in Figure 1.

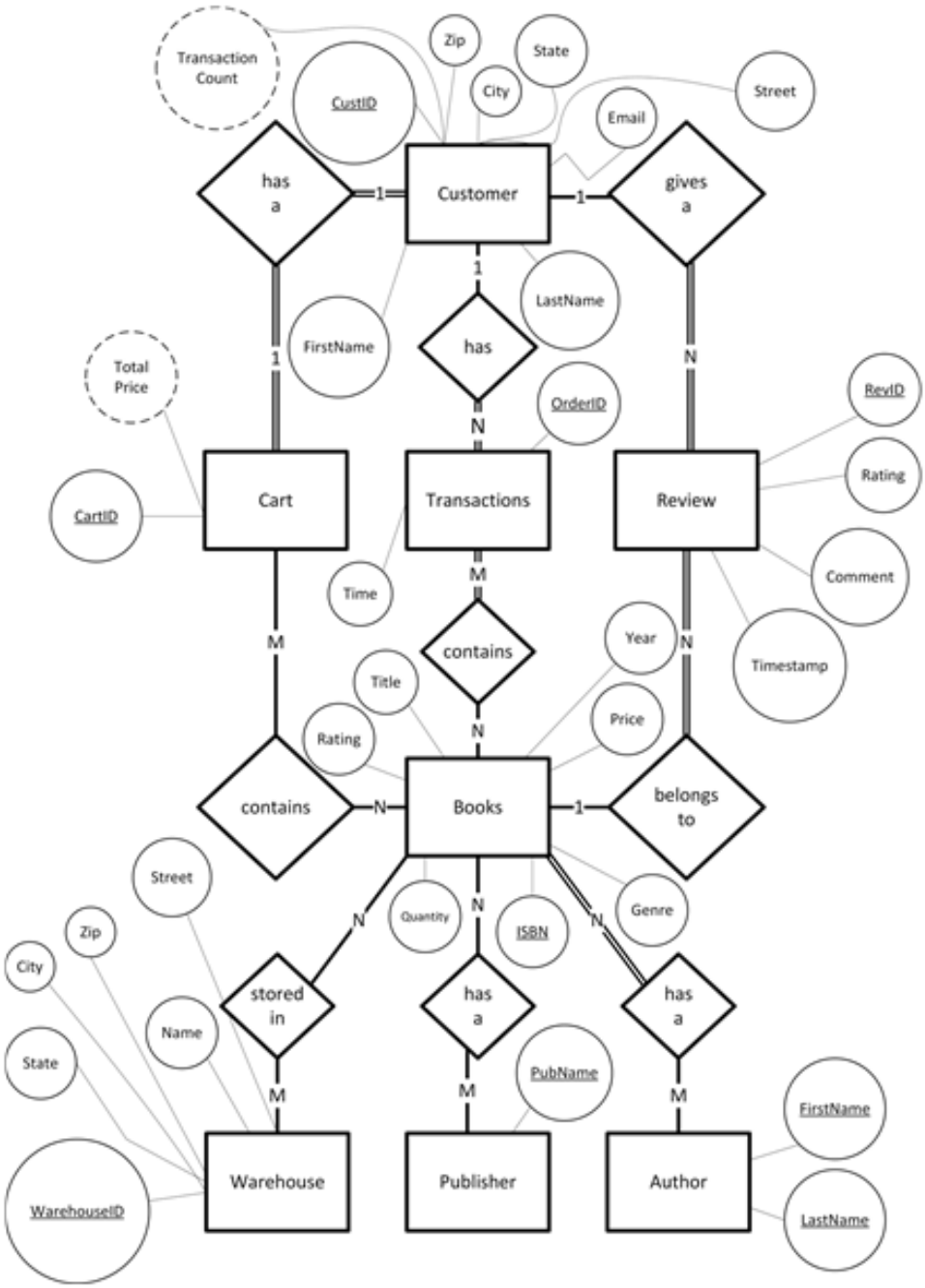

Figure 1: ER diagram of the bookstore database

4.3 Views.

Besides the tables and entities modeled in the database, the system would include some useful interfaces, views in SQL database, to better present sets of stored queries on the data collection.

(1)Books List by Costumer Rating.This view shows the average ratings of every book in the database. The entries in the table are sorted by rating in ascending order. This is useful for giving customers a list of the highest rated books. The relational algebra and SQL statement to create this view is shown below.

A1 ← (Books⋈ $_{ISBN = ISBN}$ Reviews)

All_Ratings ← $\pi_{\text{Title, } \textit{f}\text{AVERAGE Rating (A1)}}$ (A1)

```
CREATE VIEW All_Ratings AS
SELECT B.Title, AVG(R.Rating) AS Rating
FROM Books AS B, Review AS R
WHERE B.ISBN = R.ISBN
GROUP BY B.Title
ORDER BY Rating DESC;
```

(2)Books Inside the Shopping Cart.This view returns a list of books and their associated customer cart id. This is useful for viewing what a customer has added in their cart to buy. The relational algebra to create this view is shown below and the equivalent SQL statement is shown on the following page.

C1 ← (Books$\bowtie_{\text{ISBN = ISBN}}$ Books_Cart)

C2 ← (C1$\bowtie_{\text{CartID = CartID}}$ Cart)

Books_In_Cart ← $\pi_{\text{Title, Price, ISBN, Quantity, CartID}}$ (C3)

```
CREATE VIEW Books_In_Cart AS
SELECT B.Title, B.Price, B.ISBN, B.Quantity, C.CartID
FROM Books AS B, Cart AS C, Books_Cart AS BC
WHERE B.ISBN = BC.ISBN AND C.CartID = BC.CartID;
```

(3)All Books.This view returns a list of all the books sorted in alphabetical order by title. This can be useful for listing books on the website. The relational algebra and the equivalent SQL statement to create this view is shown below.

```
CREATE VIEW All_Books AS
SELECT B.Title, B.Price, B.Genre, B.ISBN
FROM Books AS B
ORDER BY B.Title ASC;
```

All_Books ← $\pi_{\text{Title, Price, Genre, ISBN}}$ (Books)

(4)All Authors.This view returns a list of all the authors sorted in alphabetical order by last name. This can be useful for listing authors on the website. The relational algebra and the equivalent SQL statement to create this view is shown below.

```
CREATE VIEW All_Authors AS
SELECT A.FirstName, A.LastName
FROM Author AS A
ORDER BY A.LastName ASC;
```

All_Authors ← $\pi_{\text{FirstName, LastName}}$ (Author)

5.Conclusion

The digital database designed for the Bits & Book has the functionality to collect and organize information of every book. The data collection of the system includes information about author, book publisher, book basic information, and inventory status. It would help the either online store or the offline shop efficiently finds and locates any book the customer desired at any time. Also, it brings extra ability to present the top seller or recommend relative books to consumers.

References

[1] Munawaroh, R. S. , & Jannah, M. . (2011). Making starmoonqueen book stores online website

using database mysql and php. Microbiology, 69(3), 266-270.
[2] Yu-Guang, Y. , Min, L. I. , & Peng-Fei, X. . (2007). Establishment of on-line bookstore system and realization of links function of its database. Journal of Changchun Normal University.
[3] Xiao-Wei, L. I. , & Jian, Z. . (2016). The design and implementation of information management system for medium and small bookstore. Computer Knowledge & Technology.
[4] Gufeng Z , Xiangdong L , Yeqin Z . Establishment and realization of a web-based open system for management of crop pests and diseases[J]. Acta Phytophylacica Sinica, 2007, 34(6):585-591.
[5] Zhou M H , Liu H B , Wu W . Development of a web-based geographic information system for crop pests and diseases management[C]// IEEE International Conference on Information Management & Engineering. IEEE, 2010.
[6] Timo G , Swanson T M . On Biology and Technology: The Economics of Managing Biotechnologies[J]. SSRN Electronic Journal, 2003.