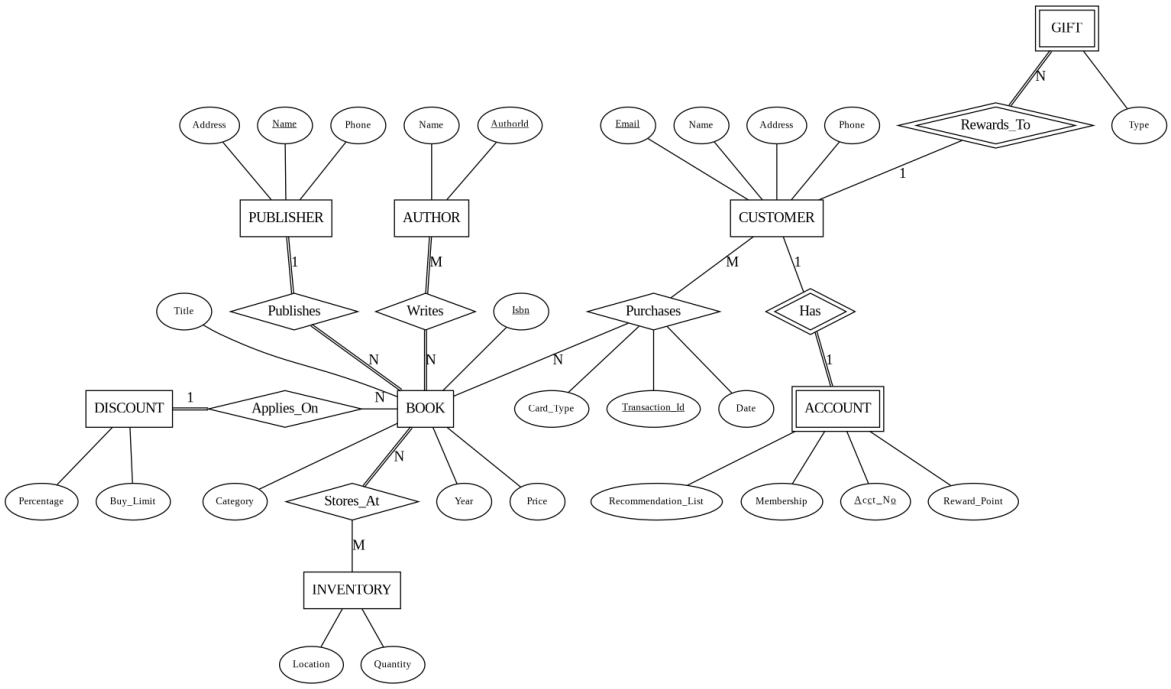
The Final Report

Wally Yang, Tina Liang, Wei Zong, Chuwen Sun, Xingyu Yan

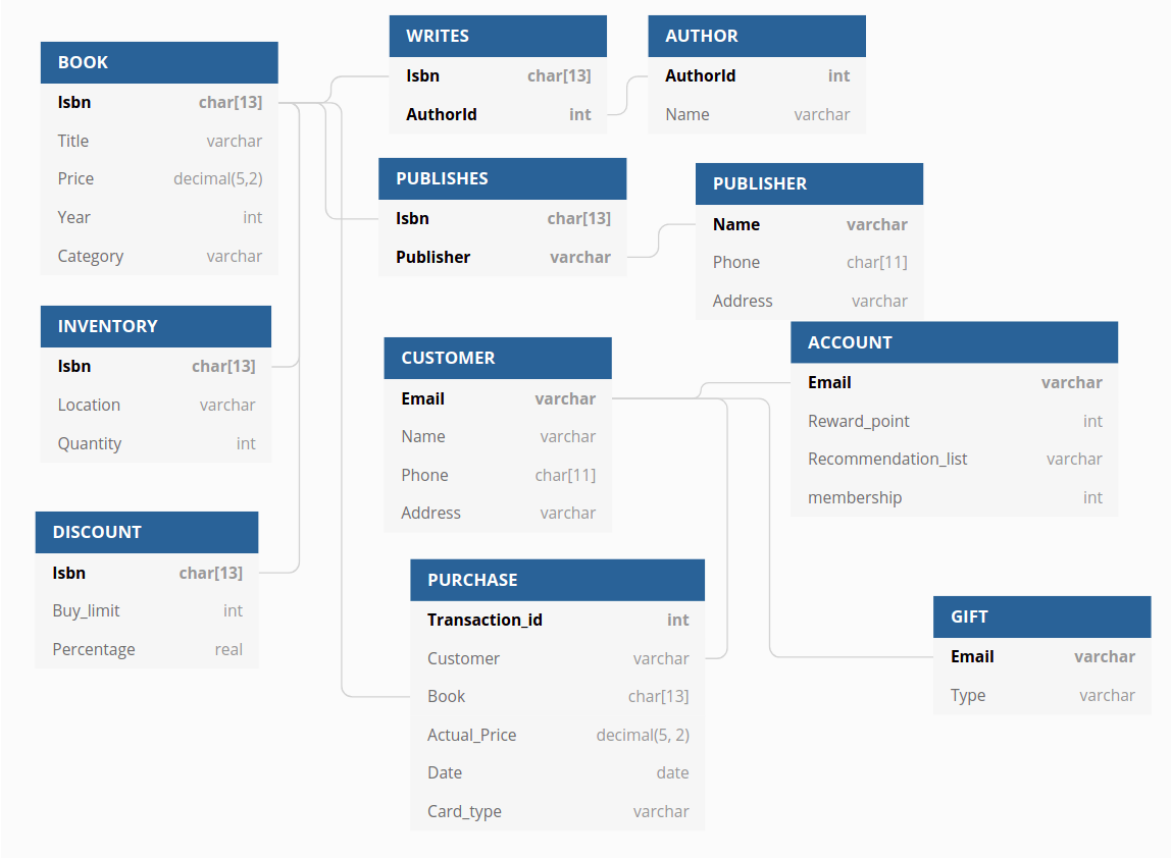
April 20, 2020

0.1 Database Description

1. ER-model of the Database Design



2. Relational Schema for the Database



The bold field refers to the primary keys of the relational schema

3. Levels of normalization for each table: All tables achieve BCNF.

4. Indices for the Database

   We chose the Tree-based index for our BOOK table since the tree-based index is good for looking up values based on range tests. It will speed up our queries when we want to retrieve the book based on the range of the year or the range of the price. Also it is not too bad for looking up values based on equality tests, so it also can slightly speed up our queries when we are trying to look up books based on the titles, categories, authors, and publishers.

5. Views for the Database

   - View A

     Description: This view is able to show all the titles and their dates of purchase made by each customer. And this could be useful to make book recommendations for a customer by looking at his or her purchase history.

Relational algebra expression:

$$R1 \leftarrow PURCHASE \bowtie_{Customer=Email} Customer$$

$$R2 \leftarrow BOOK \bowtie_{Isbn=Book} R1$$

$$Result \leftarrow \pi_{Name,\ Title,\ Date} R2$$

```
CREATE VIEW CUSTOMER_P AS
  SELECT C.Name, B.Title, P.Date
    FROM BOOK AS B, PURCHASE AS P, CUSTOMER AS C
   WHERE B.Isbn = P.Book AND P.Customer = C.Email
```

Sample output:

Luqman Finnegan	OCP: Oracle9i Certification Kit	07/01/16
Phebe Christian	SQL Server 2000 for Experienced DBA's	09/16/18
Charlie Dolan	The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling	07/20/18
Kiya Mcguire	How To Do Everything with Your Tablet PC	01/26/19
Amal Terrell	Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations	06/15/17

- View B

  Description: This view is able to show the total number of books purchased by each customer. And this could be useful to see if this customer deserves a gift by making a certain amount of purchases in this store.

  Relational algebra expression:

$$R1 \leftarrow PURCHASE \bowtie_{Customer=Email} Customer$$

$$Result \leftarrow {}_{Customer}\mathcal{F}_{COUNT\ Book}(R1)$$

```
CREATE VIEW CUSTOMER_N AS
  SELECT P.Customer, COUNT(Book)
    FROM PURCHASE AS P, CUSTOMER AS C
   WHERE P.Customer = C.Email
   GROUP BY P.Customer
```

Sample output:

Ahmed.12@osu.edu	1
Christian.2@osu.edu	1
Dolan.3@osu.edu	1
Finnegan.1@osu.edu	1
Firth.9@osu.edu	1

6. Sample Transactions for the Database

- Transaction A

  Description: The customer adds a book to a order and update the book quantity in the inventory

```sql
BEGIN TRANSACTION NEW_P;

  INSERT INTO PURCHASE
  VALUES (22,
          'Finnegan.1@osu.edu',
          '782140661',
          104.97,
          DATE(),
          'AMEX');

UPDATE Inventory
   SET Quantity = Quantity - 1
  WHERE Isbn = '782140661' AND Location = 'warehouse';

COMMIT;
```

- Transction B

  Description: A certain amount(10) of books(Isbn: 616601654) transmitted from one inventory(warehouse) to another(in-sotre)

```sql
BEGIN TRANSACTION MOVE;

UPDATE Inventory
   SET Quantity = Quantity + 10
 WHERE Isbn = '782140661' AND Location = 'warehouse';

UPDATE Inventory
   SET Quantity = Quantity - 10
 WHERE Isbn = '782140661' AND Location = 'in-store';

COMMIT;
```

- Transaction C

  Description: Customer redeem 100 reward points to a keychain

```sql
BEGIN TRANSACTION NEW_GIFT;

UPDATE ACCOUNT
   SET Reward_point =  Reward_point - 100
 WHERE Email = 'Finnegan.1@osu.edu';

INSERT INTO GIFT
VALUES ('Finnegan.1@osu.edu', 'keychain');

COMMIT;
```

0.2 User Manual

1. Database Description

Table	Entity	Attribute	Data type	Description	Constraints
BOOK	book	Isbn	CHAR(13)	The unique identifier of the book; The primary key	NOT NULL; fixed length of 13
		Title	VARCHAR(100)	Title of the book	NOT NULL; Up to length 100
		Author	VARCHAR(100)	Author of the book	NOT NULL; Up to length 100
		Price	DECIMAL(5,2)	Price of the book	NOT NULL; total 5 digits, two digits after the decimal point
		Year	INT	Year of the book got published	NOT NULL
		Category	VARCHAR(100)	Category of the book	NOT NULL; Up to length 100
INVENTORY	inventory	Isbn	CHAR(13)	The foreign key references to BOOK table; the primary key	NOT NULL; fixed length of 13
		Location	VARCHAR(100)	Location of the inventory	NOT NULL; Up to length 100
		Quantity	INT	the number of books in inventory	
DISCOUNT	discount	Isbn	CHAR(13)	The foreign key references to BOOK table	NOT NULL; fixed length of 13
		Buy_limit	INT	Purchase limit to apply the discount	
		Percentage	REAL	The discount applied to the book	NOT NULL
CUSTOMER	customer	Email	VARCHAR(100)	The unique identifier of the customer; The primary key	NOT NULL; Up to length 100
		Name	VARCHAR(100)	The name of the customer	NOT NULL; Up to length 100
		Phone	CHAR(10)	The phone of the customer	fixed length of 10
		Address	VARCHAR(100)	The address of the customer	NOT NULL; Up to length 100
AUTHOR	author	AuthorId	INT	The unique identifier of the author; The primary key	NOT NULL
		Name	VARCHAR(100)	The name of the author	NOT NULL
PUBLISHER	publisher	Name	VARCHAR(100)	The unique identifier of the publisher; The primary key	NOT NULL; Up to length 100
		Phone	CHAR(10)	The phone of the publisher	fixed length of 10
		Address	VARCHAR(100)	The address of the publisher	NOT NULL; Up to length 100
GIFT	gift	Email	VARCHAR(100)	The foreign key references to CUSTOMER table; The primary key	NOT NULL; Up to length 100
		Type	VARCHAR	The type of the gift	
ACCOUNT	account	Account_no	INT	The unique identifier of the account; The primary key	NOT NULL
		Email	VARCHAR(100)	The foreign key references to CUSTOMER table	NOT NULL; Up to length 100
		Reward_point	INT	The reward point of the account	
		Recommendation_list	VARCHAR(500)	The recommendation list for the customer	Up to length 500
		membership	INT	The level of the membership	
WRITES	author, book	Isbn	CHAR(13)	The foreign key references to BOOK table; One of the primary key	NOT NULL; fixed length of 13
		Authorid	INT	The foreign key references to AUTHOR table; One of the primary key	
PUBLISHES	book, publisher	Isbn	CHAR(13)	The foreign key references to BOOK table	NOT NULL
		Publisher	VARCHAR(100)	The foreign key references to PUBLISHER table	NOT NULL
PURCHASE	customer, book	Transaction_id	INT	The unique identifier of the purchase; The primary key	NOT NULL
		Customer	VARCHAR(100)	The foreign key references to CUSTOMER table	NOT NULL; Up to length 100
		Book	CHAR(13)	The foreign key references to BOOK table	NOT NULL
		Date	DATE	The date of purchase	
		Card_type	VARCHAR(100)	The type of the card that customer use to purchase	Up to length 100
		Actural_Price	DECIMAL(5,2)	The price of the transaction	NOT NULL; total 5 digits, two digits after the decimal point

2. Sample SQL Queries

   (a) Find the titles of all books by Pratchett that cost less than $10

   Return a table that contains the title of that books written by Pratchett and cost less than $10

(b) Find the titles of all books by Pratchett that cost less than $10

(c) Find the titles and ISBNs for all books with less than 5 copies in stock

(d) Give all the customers who purchased a book by Pratchett and the titles of Pratchett books they purchased

(e)