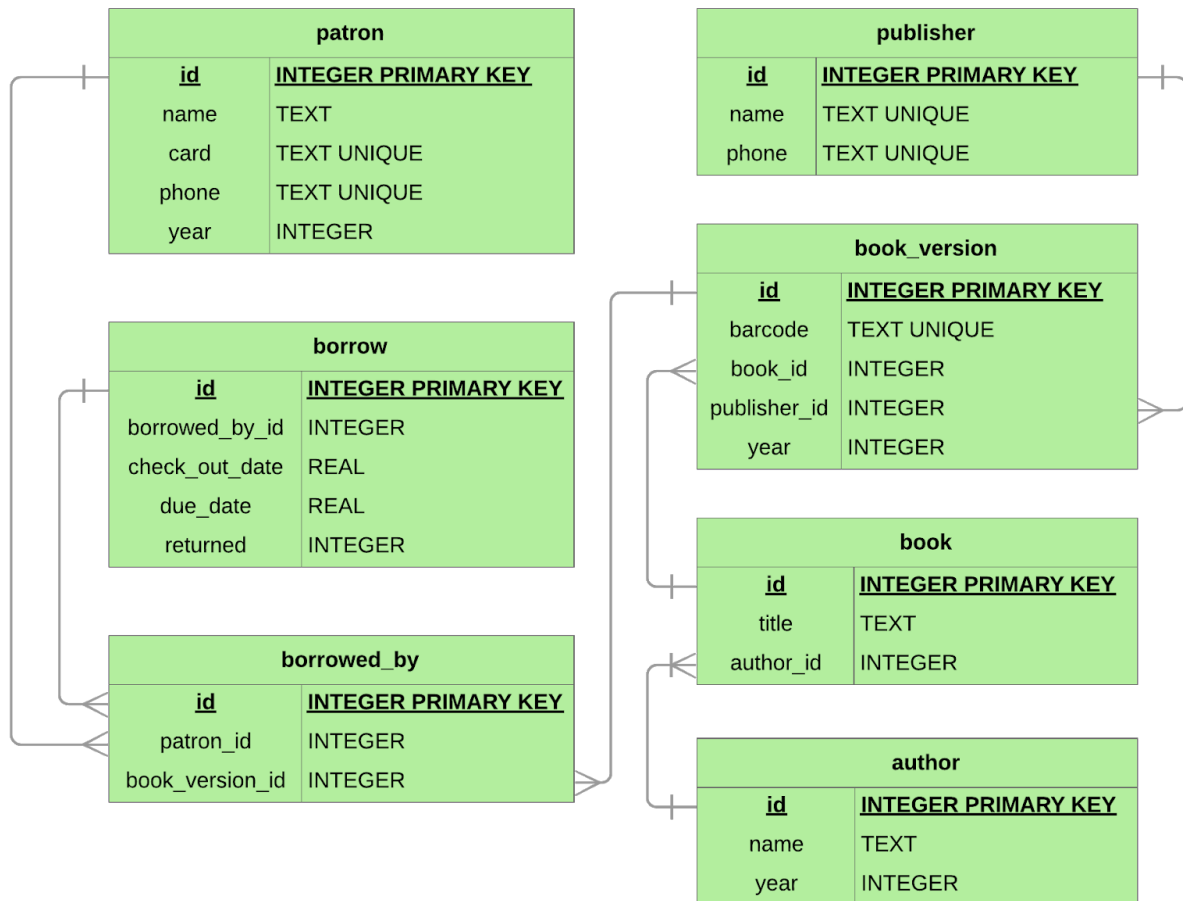


PA2 REPORT

SHANGYU ZHANG

ER Diagram (4NF)



Clarification

In all the tables, `id` is created to be the primary key for better performance on DB query.

Table `patron`

Some patrons share the exact same names but different cards, phones and joined years. I prefer to consider them as different persons. Since only few patrons have the same name, 4NF is not adopted in this table.

"Yanira Wheeler"	"4727498535203287930603641"	" +1-(996)-216-1891"	"2003"
"Yanira Wheeler"	"1195094781883833145765974"	"248-054-7644"	"2015"

"Yanira Wheeler"	"2357667128359497769894125"	"1-(102)-891-7197"	"1997"
"Zane Irwin"	"2816080189446391139436214"	"980.467.8497"	"1994"
"Zane Irwin"	"2896552070462588416602920"	"1-302-958-5583"	"1986"

For each patron, the card and phone number are unique. The joined year is determined by the card too. So, the `patron` table contains name, card, phone and joined year. The card and phone are unique columns.

Table `author`

No authors share the same name. And an author's birth year depends on himself/herself. So, `author` table contains id, name and birth year.

Table publisher

No publishers share the same name nor phone numbers. So, `publisher` table contains id, name and phone. Name is unique. Phone is unique.

Table book

There are books with the same title but different authors. Two books called 'Power Electronics' are two different books written by two different authors, Rashid Muhammad and Mohan Ned. So, table `book` is unique on the pair of title and author_id. Since one author wrote one or more books, like Palkhivala has written two books, 'We the Nation' and 'We the people', author_id in table `book` has a one or many to one relationship to the id in table `author`.

Table book_version

Why do we need this table? Can we put barcode, publisher_id and published year to table `book`? If it is a 3NF schema, yes we can. But for a 4NF schema, we can't. Because even the books with the same titles and authors still have different publishers and barcodes. No multivalued dependencies are allowed in 4NF. Thus, I created table `book_version` including barcode, book_id, publisher_id and published year. The book_id has a many to one relationship to the id in table `book`. The publisher_id has a many to one relationship to the id in table `publisher`.

Table borrowed_by

One patron can borrow many books, and one book can be borrowed by many patrons. Therefore, it is a many to many relationship between book_version_id and patron_id. The book_version_id has a many to one relationship to the id in table `book_version`. The patron_id has a many to one relationship to the id in table `patron`. Table `borrowed_by` is to present the

relationship of patrons and books. Pair of book_version_id and patron_id is unique. Besides, one patron can borrow the same book for many times. Therefore, I created another table borrow to reduce multivalued data. Otherwise there will be multiple rows of the same patron_id and book_version_id pairs with different check out dates.

Table borrow

Since one patron can borrow the same book for several times, I created a primary key id in this table to identify each check out. The borrowed_by_id has a many to one relationship to the id in table `borrowed_by`. Column returned can only be 1 or 0.

1NF Schema

The original csv is already a 1NF schema.

The candidate key should be (card_barcode, book_barcode, check_out_date).

data	
patron	TEXT
year_joined	INTEGER
<u>card_barcode</u>	<u>TEXT</u>
patron_phone	TEXT
<u>book_barcode</u>	<u>TEXT</u>
returned	INTEGER
title	TEXT
year	INTEGER
author	REAL
author_year	INTEGER
publisher	REAL
publisher_phone	TEXT
<u>check_out_date</u>	<u>REAL</u>
due_date	REAL

2NF Schema

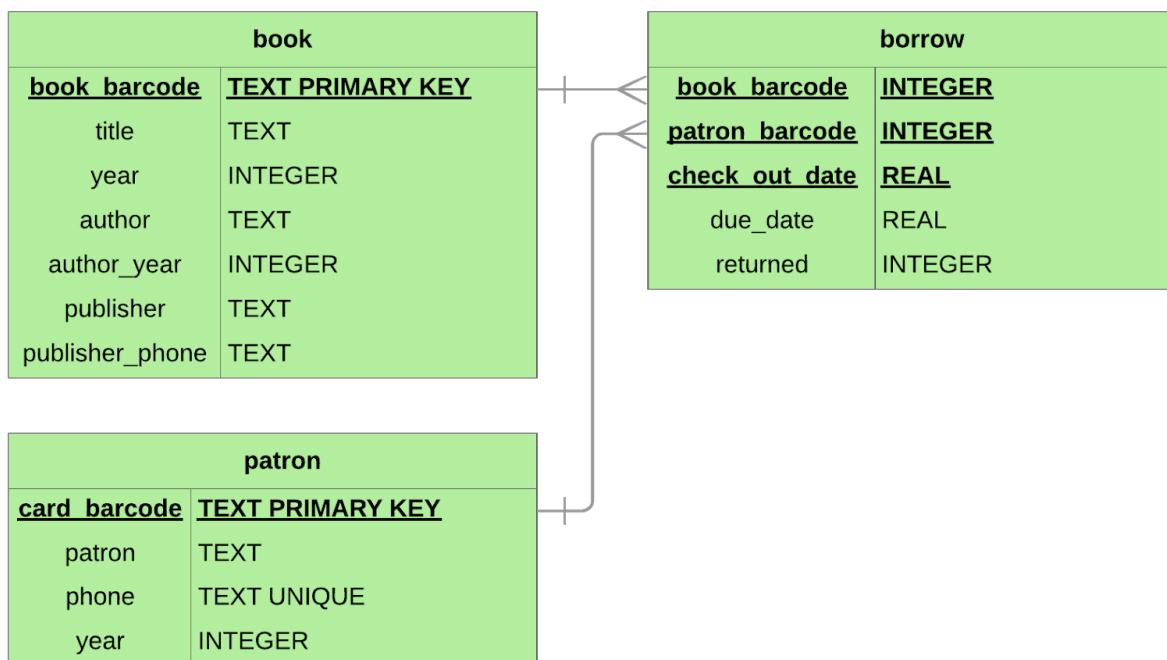
To remove the partial dependencies in the 1NF schema, I separated the 1NF schema into three tables.

The candidate key in table `patron` is card_barcode.

The candidate key in table `book` is book_barcode.

The candidate key in table `borrow` is (book_barcode, patron_barcode, check_out_date).

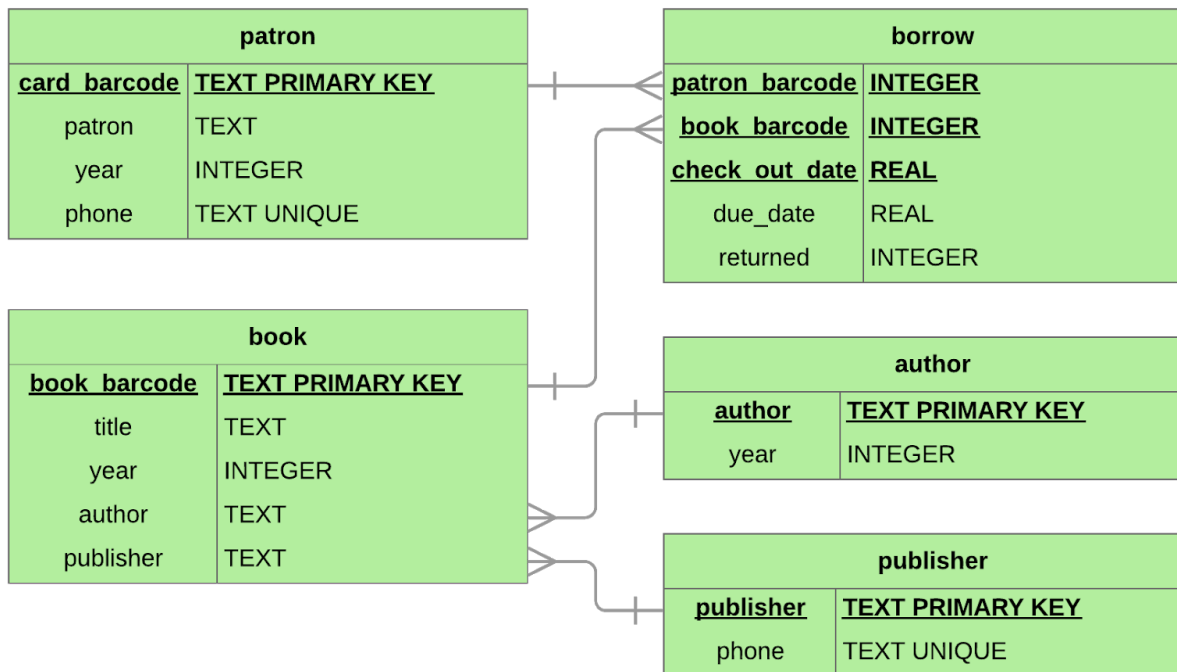
Now, each non-key column in each table depends and only depends on its candidate key.



3NF Schema

In table `book` of the 2NF schema, author_year actually is determined by author. And publisher_phone is determined by the publisher.

In 3NF schema, no transition dependencies are allowed in any tables. So, we have to separate author and publisher from table `book`.



Functional Dependencies

Under my observation of the data, I have the conclusions like the following.

author => author year

The author name is unique and the birth year is determined by the author.

publisher => publisher phone

The number of distinct publisher and publisher phone is the same. Therefore, both publisher and publisher phone are unique. Both could be a key. However, considering the real world, I prefer to use publisher as the key. Publisher phone is depended on publisher.

card barcode => patron name, patron phone, year joined

The number of distinct card barcode is more than those of patron name, patron phone and year joined. Therefore, card barcode is the key here.

book barcode => title, author, publisher, published year

The number of distinct book barcode is more than those of title, author, publisher and published year. Books with the same title may be different books written by different authors. Books with the same title and same author may have the different publishers and published years. Therefore, only the book barcode can be the key here.

card barcode, book barcode, check out date => due, returned

Card barcode and book barcode cannot decide the check out date, due date and returned or not. Because, the same patron can borrow the same book for many times. Therefore, card barcode, book barcode and check out date is the key here.

Based on transition rules, we can get:

book barcode => title, author, author year, publisher, published year, publisher phone

Since we have:

card barcode => patron name, patron phone, year joined

card barcode, book barcode, check out date => due, returned

Based on union rules, we can get:

card barcode, book barcode, check out date => patron name, patron phone, year joined, title, author, author year, publisher, published year, publisher phone, due, returned

We can tell that **(card barcode, book barcode, check out date)** determine all the other columns. Therefore, the candidate key is **(card barcode, book barcode, check out date)**.