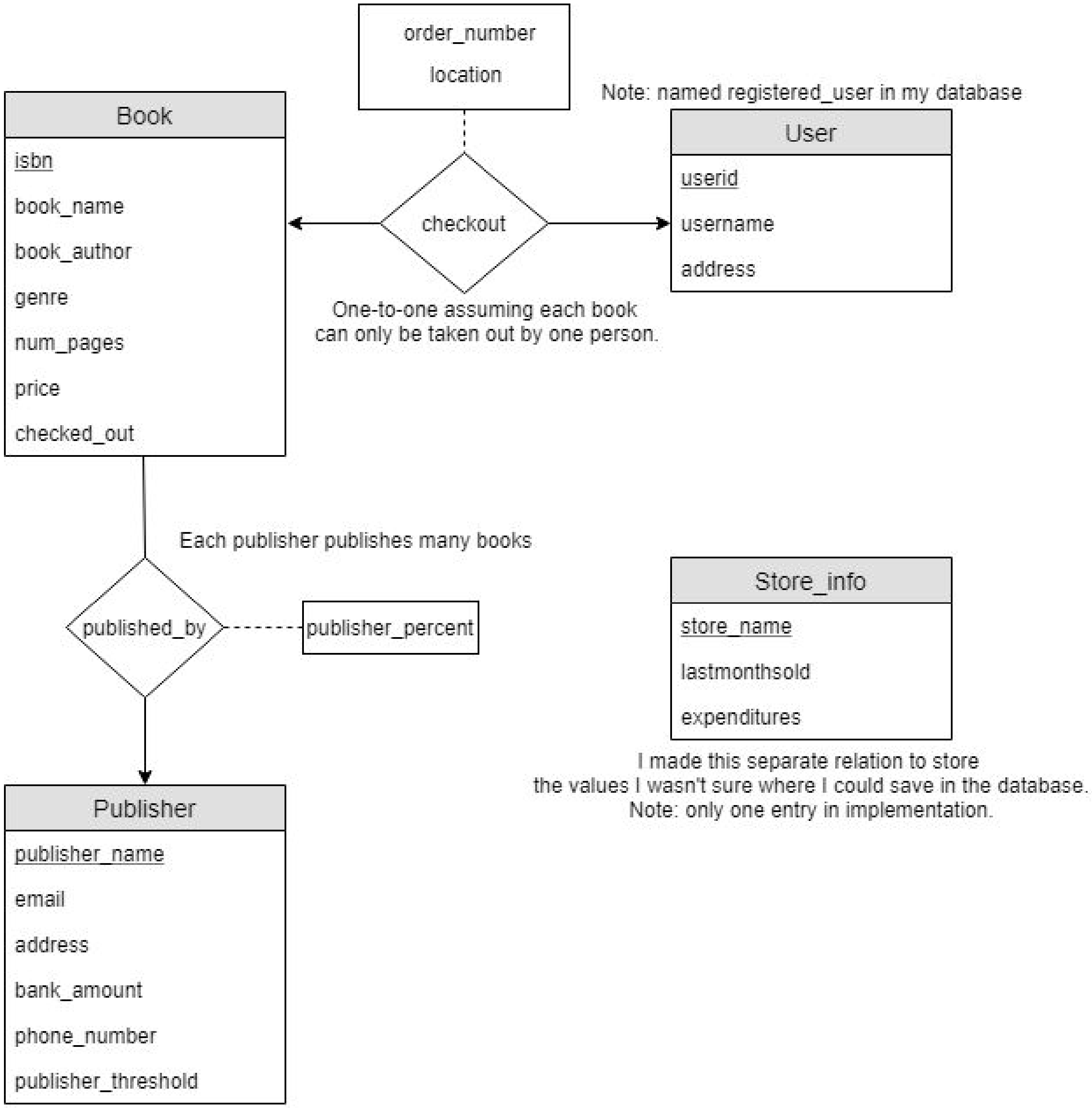
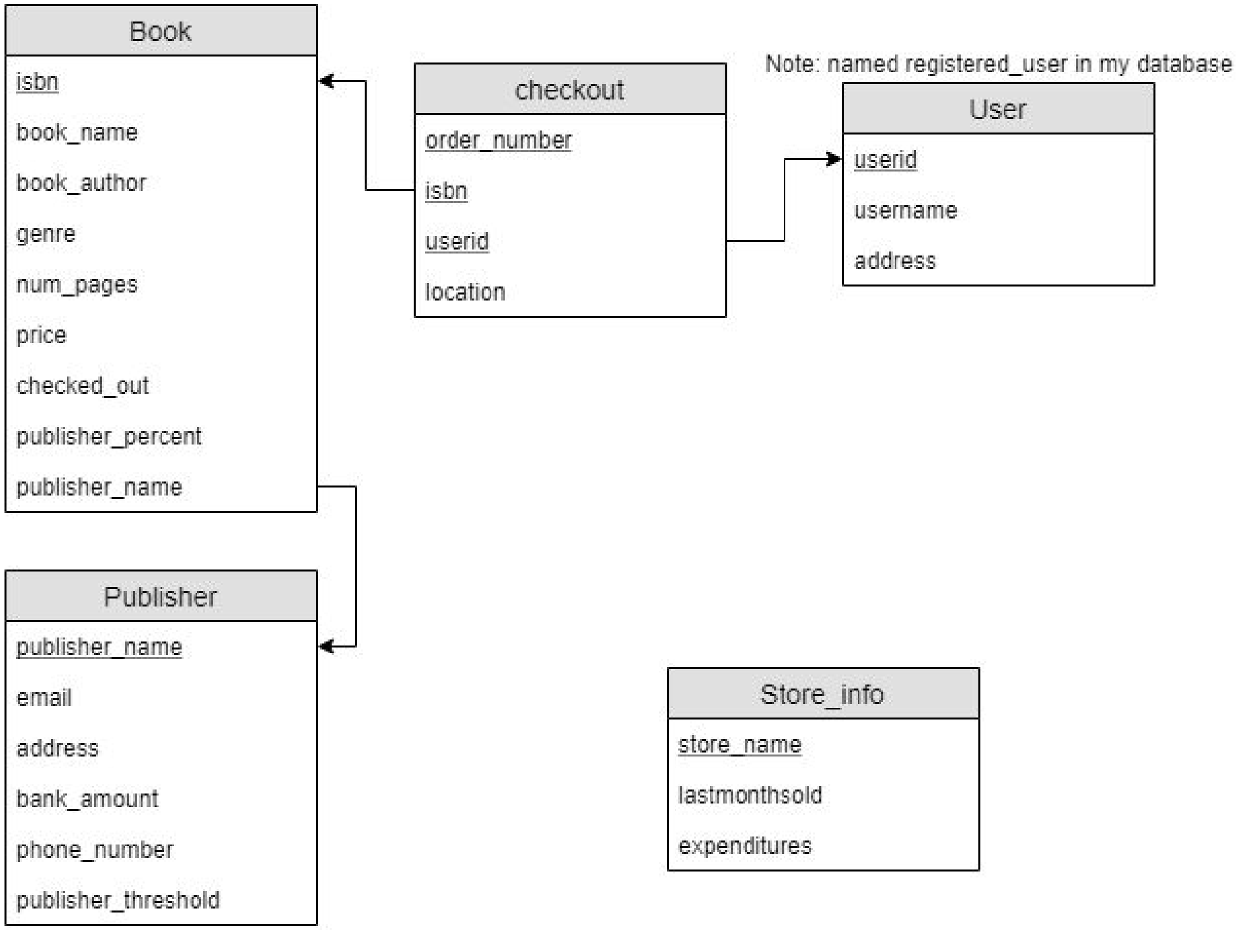
Conceptual Design (ER-Diagram)


order_number
location
Note: named registered_user in my database
Book
isbn
book_name
book_author
genre
num_pages
price
checked_out
checkout
User
userid
username
address
One-to-one assuming each book
can only be taken out by one person.
Each publisher publishes many books
published_by
publisher_percent
Store_info
store_name
lastmonthsold
expenditures
I made this separate relation to store
the values I wasn't sure where I could save in the database.
Note: only one entry in implementation.
Publisher
publisher_name
email
address
bank_amount
phone_number
publisher_threshold

Reduction to Relation Schemas

book(isbn, book_name, book_author, genre, num_pages, price, checked_out, publisher_percent, publisher_name)
publisher(publisher_name, email, address, bank_amount, phone_number, publisher_threshold)
registered_user(userid, username, address)
checkout(order_number, isbn, userid, location)
store_info(store_name, lastmonthsold, expenditures)

Schema Diagram



Note: named registered_user in my database

Book
isbn
book_name
book_author
genre
num_pages
price
checked_out
publisher_percent
publisher_name

checkout
order_number
isbn
userid
location

User
userid
username
address

Publisher
publisher_name
email
address
bank_amount
phone_number
publisher_threshold

Store_info
store_name
lastmonthsold
expenditures

Normalization of Relation Schemas

To prove my relation schemas are in good normal form, I'll show the tests of each relation schema and how it is in Third Normal Form (with appropriate assumptions for functional dependencies).

Store_info Relation

Functional dependencies

1. *store_name → lastmonthsold, expenditures* (Assuming given the store name we can determine the info about the store)

The store_info relation is in third normal form because the only functional dependency has a superkey on the left hand side.

Publisher Relation

Functional dependencies

1. *publisher_name → email, address, bank_amount, phone_number, publisher_threshold* (Given the name can determine all info about the publisher)
2. *email → publisher_name* (assuming you can determine name if you know the email - can google it and find results)
3. *phone_number → publisher_name* (assuming you can determine name if you know the phone number)
4. *address → publisher_name* (assuming you can determine name if you know the address)

The publisher relation is in third normal form because each of the functional dependencies has a superkey on the left-hand side (can determine all other attributes). This is because knowing the publisher_name allows you to determine every other attribute, and the other functional dependencies allow you to determine the publisher_name.

Book relation

Functional dependencies

1. *isbn → book_name, book_author, genre, num_pages, price, checked_out, publisher_name, publisher_percent* (assuming all info about the book can be searched up with the ISBN)
2. *book_name → author, genre, num_pages, publisher*
3. *author, genre, num_pages, publisher → isbn* (Assuming with the help of the publisher, can find the ISBN of the book with the other info on the book)

Dependency 1 is a superkey because isbn determines all attributes for books.

Dependency 2 is a superkey because…

$(book\_name)^{+} = author, genre, num\_pages, publisher$

$author, genre, num\_pages, publisher -> isbn : (book\_name)^{+} = book\_name, book\_author, genre, num\_pages, price, checked\_out, publisher\_name, publisher\_percent$

It determines isbn through dependency 3 which determines all attributes for books.

Dependency 3 is a superkey because it determines isbn which determines all attributes for books.

Therefore the book relation is in third normal form.

Registered_user relation

Functional dependencies

1. *userid → username, address*

Note: *username → userid* is not a functional dependency because you can't figure out your ID using your username, as the ids are stored internally.

The registered_user relation is in third normal form because the only functional dependency has a superkey on the left hand side.

Checkout relation

Functional dependencies

1. *order_number → isbn, userid, location*

Note: *userid → order_number* is not a functional dependency because you're only shown your order number once when completing your order.

The checkout relation is in third normal form because the only functional dependency has a superkey on the left hand side.