

### *Bookstore: -*

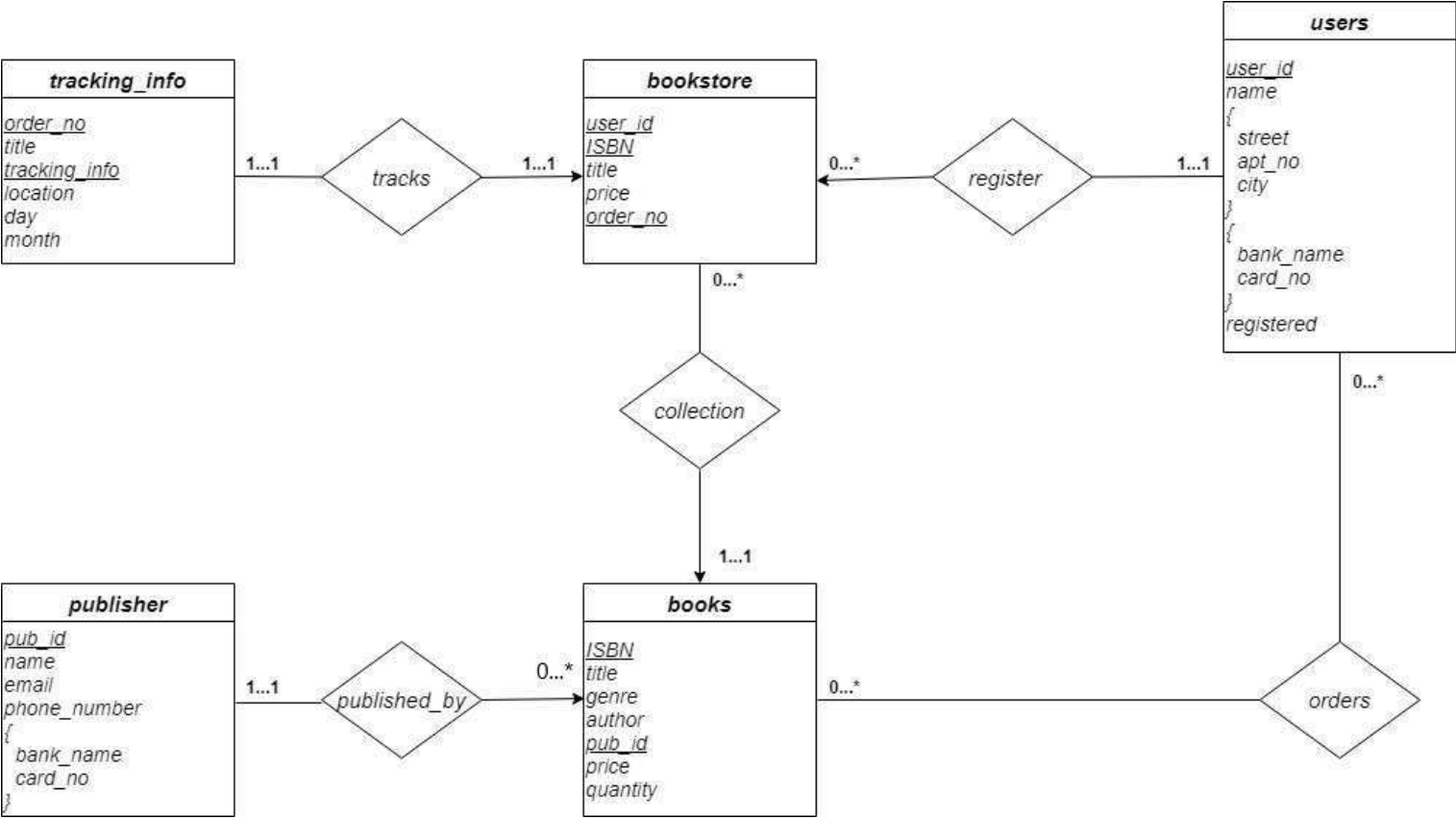
- A bookstore user interface that uses CRUD operations to interact with a postgresSQL server in order to manipulate data.
- For more about how to run the program, refer to the Implementation section
- 

## **Conceptual Design**

### Assumptions made: -

- The bookstore entity set acts as the order history for store, where entries are added when users order books
- We are also assuming that there is just one bookstore
- The cardinality between Bookstore and User is Many to One, where the Bookstore can have 0 to Many Users but each user can only be related to one bookstore.
- The cardinality between the Bookstore and Books is Many To One, where the Bookstore can have 0 to Many books while each book can only be related to one bookstore (in this specific case, since we are assuming only one bookstore exists).
- The cardinality between Bookstore and Tracking\_info is one to one, where each order in the bookstore will have only one tracking information and each tracking information per order relates just to one order in the book store,
- The cardinality between Books and Users is Many to Many, where Books may be ordered by many Users and Users can order many books.
- The cardinality between Publisher and Books is one to many where each publisher can publish many books but a book can only be published by one publisher

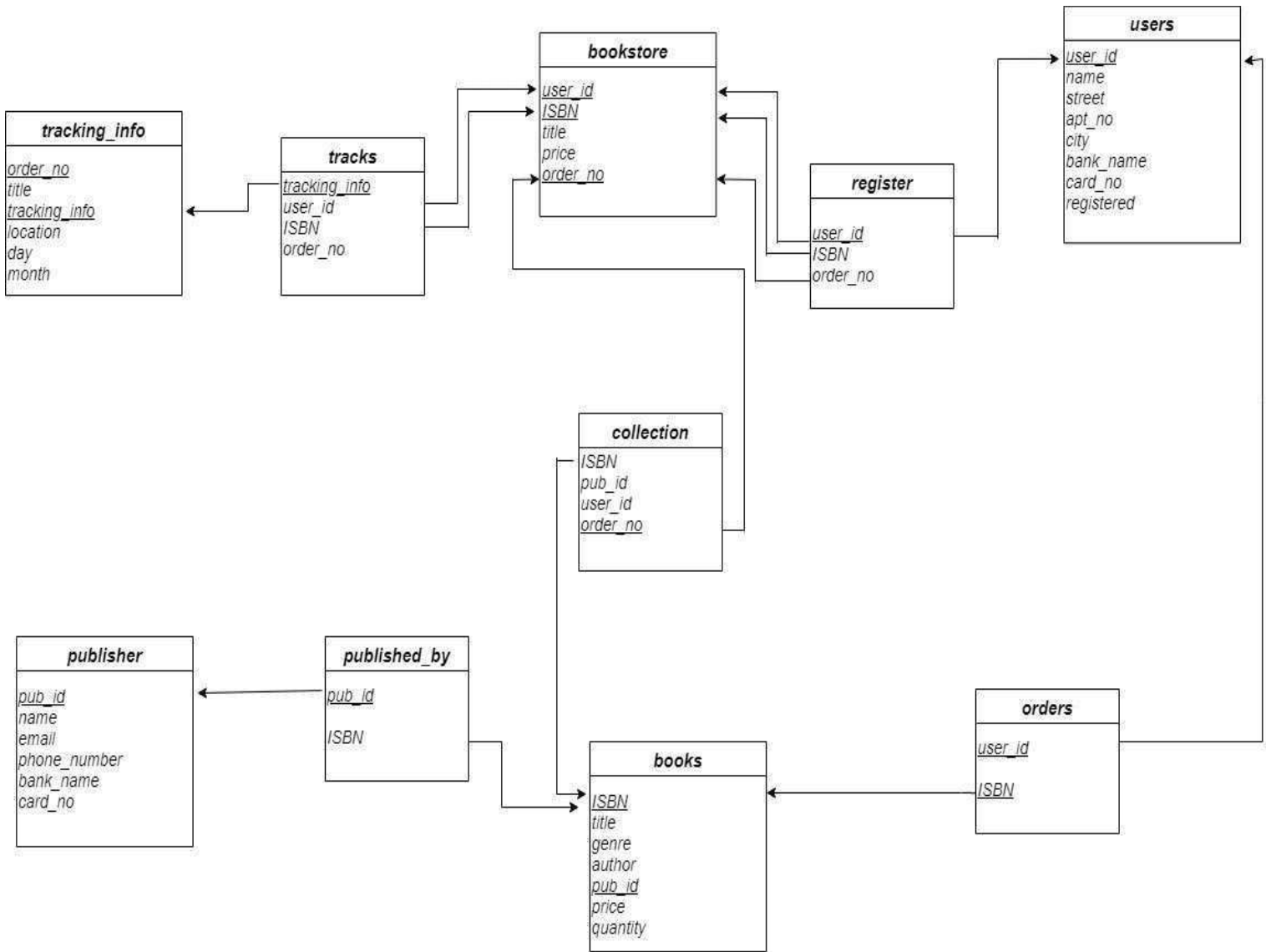
ER Diagram: -



**Reduction to Relation schemas**

bookstore( user\_id, ISBN, title, price, order\_no) user( user\_id, name, street, apt\_no, city, bank\_name, card\_no, registered)  
books( ISBN, title, genre, author, pub\_id, price, quantity) publisher( pub\_id, name, email, phone\_no, bank\_name, card\_no)  
tracking\_info( tracking\_no, title, order\_no, location, day, month)  
  
orders( user\_id, ISBN) register( user\_id, ISBN, order\_no) tracks( tracking\_no, ISBN, user\_id, order\_no) collection( ISBN, pub\_id, user\_id, order\_no) published\_by(pub\_id, ISBN)

**Database Schema Diagram**



## Implementation

I decided to use JavaScript, html, pug, node.js and express to turn this project into a web application. In this section I will talk about how the user can interact with my application.

Pre requisites :-

- To run my program, download all the files in my github repo.
- The user must have node.js and a web browser (I used Google Chrome) installed to run my application
- To install all my dependencies, open up a terminal where the project folder is located and type npm init

Running the application: -

- Initializing the database: -
  - For this you will require the contents stored in the SQL directory.
  - Start pgadmin and create a database called “bookstore” with owner postgres.
  - On my machine, the database password was “1234” , port = 8000 and host = “localhost”
  - Open up the query tool for the database and insert the DDL statements found in the DDLbookstore.sql file into the tool and run it. This will create our tables
  - After that, copy the insert queries from the bookstore\_data.sql file into the query tool and run it. This will insert books and publishers into our database.
- The Server : -
  - Open up a terminal and type “node B\_server.js”
  - This should be enough to start the server
  - Go to your web browser and type “<http://localhost:3000/>” in your address bar
  - At this point, you should see the homepage which looks like this (check screenshot below)
  - We can now interact with our application.

[Home](#) [Create User](#)

Username:

## Welcome to BestBooks

Search a book by typing it's title, ISBN, genre or author name

Our Library :

[Harry Potter and the Philosophers Stone by J.K Rowling](#)

[Hunger Games by Suzanne Collins](#)

[Champion by Marie Lu](#)

[Prodigy by Marie Lu](#)

[Legend by Marie Lu](#)

[Game of thrones One by George Martin](#)

[Romeo and Juliet by W.Shakespeare](#)

[Merchant of Venice by W.Shakespeare](#)

[Oliver Twist by Charles Dickens](#)

[Hamlet by W.Shakespeare](#)

[To Sir With Love by Braithwaite](#)

Searching for books: -

- In order to search for books in the database, the user must type into the search bar the title, ISBN, genre or author of the book they are looking for. The search is case sensitive, so the user must type in the data accurately.
- Any user can search for a book, but in order to view the books details and add the book to your cart, a user must be logged in

Logging In, Logging out and Registering: -

- In order to order books, the first thing a user must do is create a profile, which is done by clicking "Create your profile" at the top of the page.
- This will redirect you to this page:

[Home](#) [Create User](#)

Username:

## Create your profile:

Name:

Street:

Apt. Number:

City:

Bank Name:

Credit Card Number:

- 
- Once you have entered your details, the user must click sign up.
- The server checks the user table in the database to see if the user has already been created. If the user exists within the table, the user is directed to a page with the error “this user already exists please go back” at which point the user has to click the back button and enter proper details.
- If the user does not exist, the user is added to the table and redirected to the home page.
- The user can add multiple entries into the user table, however this does not mean the user is logged in which means he still cant make orders.
- In order to make orders, the user must type his username into the login bar at the top of the home page.
- When the user hits login, the server checks to see if the user exists within the database, and if the user does not, they are redirected to a page with the following error “This user has not been created please go back”.
- If the user exists, two things happen :
  - The registered column is updated for that user to true, meaning the user is now registered.
  - The user is added to an object on the server that mimics a browser session.
- The home page now looks likes this:

[Home](#)

Hello Vegeta : -

[Logout/Unregister](#)

## Welcome to BestBooks

Search a book by typing it's title, ISBN, genre or author name

Our Library :

[Harry Potter and the Philosophers Stone by J.K Rowling](#)

[Hunger Games by Suzanne Collins](#)

[Champion by Marie Lu](#)

[Prodigy by Marie Lu](#)

[Legend by Marie Lu](#)

[Game of thrones One by George Martin](#)

[Romeo and Juliet by W.Shakespeare](#)

[Merchant of Venice by W.Shakespeare](#)

[Oliver Twist by Charles Dickens](#)

[Hamlet by W.Shakespeare](#)

[To Sir With Love by Braithwaite](#)

- 
- The user Vegeta can now view book profiles and add them to their cart
- When the user logs out, the user is removed from the session object on the server and the specified user's registered column is updated in the database from true to false. This means that they would have to log in again in order to add books to their cart.
- If the browser window is closed, the user still exists in the database but must be logged in again in order to purchase books.

Buying Books: -

- In order to buy a book a user must be logged in
- Once logged in and registered, a user now can view the book information and has access to the order button
- When the order button is clicked, the book data is added to a cart object on the server and is redirected to the home page which now looks like this :
-



[Home](#)

Hello bruce : -

[Logout/Unregister](#)

[Show Cart](#)

## Welcome to BestBooks

Search a book by typing it's title, ISBN, genre or author name

Our Library :

[Harry Potter and the Philosophers Stone by J.K Rowling](#)

[Champion by Marie Lu](#)

[Prodigy by Marie Lu](#)

[Legend by Marie Lu](#)

[Game of thrones One by George Martin](#)

[Romeo and Juliet by W.Shakespeare](#)

[Merchant of Venice by W.Shakespeare](#)

[Oliver Twist by Charles Dickens](#)

[To Sir With Love by Braithwaite](#)

[Hamlet by W.Shakespeare](#)

[Hunger Games by Suzanne Collins](#)

- 
- A show Cart link is displayed, when clicked it looks like this
- 

[Home](#)

Hello bruce : -

[Logout/Unregister](#)

[Show Cart](#)

## Your cart: -

- Harry Potter and the Philosophers Stone for 45.50 dollars

[Order Now](#)

- A user can now view all his/her orders
- If the user clicks “Order now”, the server creates loops through the cart and does the following
  - Creates a tracking number
  - Enters the order into the tracking\_info table first
  - Enters the order into the bookstore (which as mentioned earlier acts as the order history for the store so when the user logs out or exits the app, the orders and tracking info still remain)
  - Decreases the quantity of specified books at the store
    - Redirects the user back to the home page with an empty cart
- Multiple orders in cart shown below

[Home](#)

Hello bruce : -

[Logout/Unregister](#)

[Show Cart](#)

## Your cart: -

- Harry Potter and the Philosophers Stone for 45.50 dollars
- Prodigy for 55.49 dollars
- Legend for 55.49 dollars
- Prodigy for 55.49 dollars

[Order Now](#)

<https://github.com/brucefernandes/Bookstore Project/>