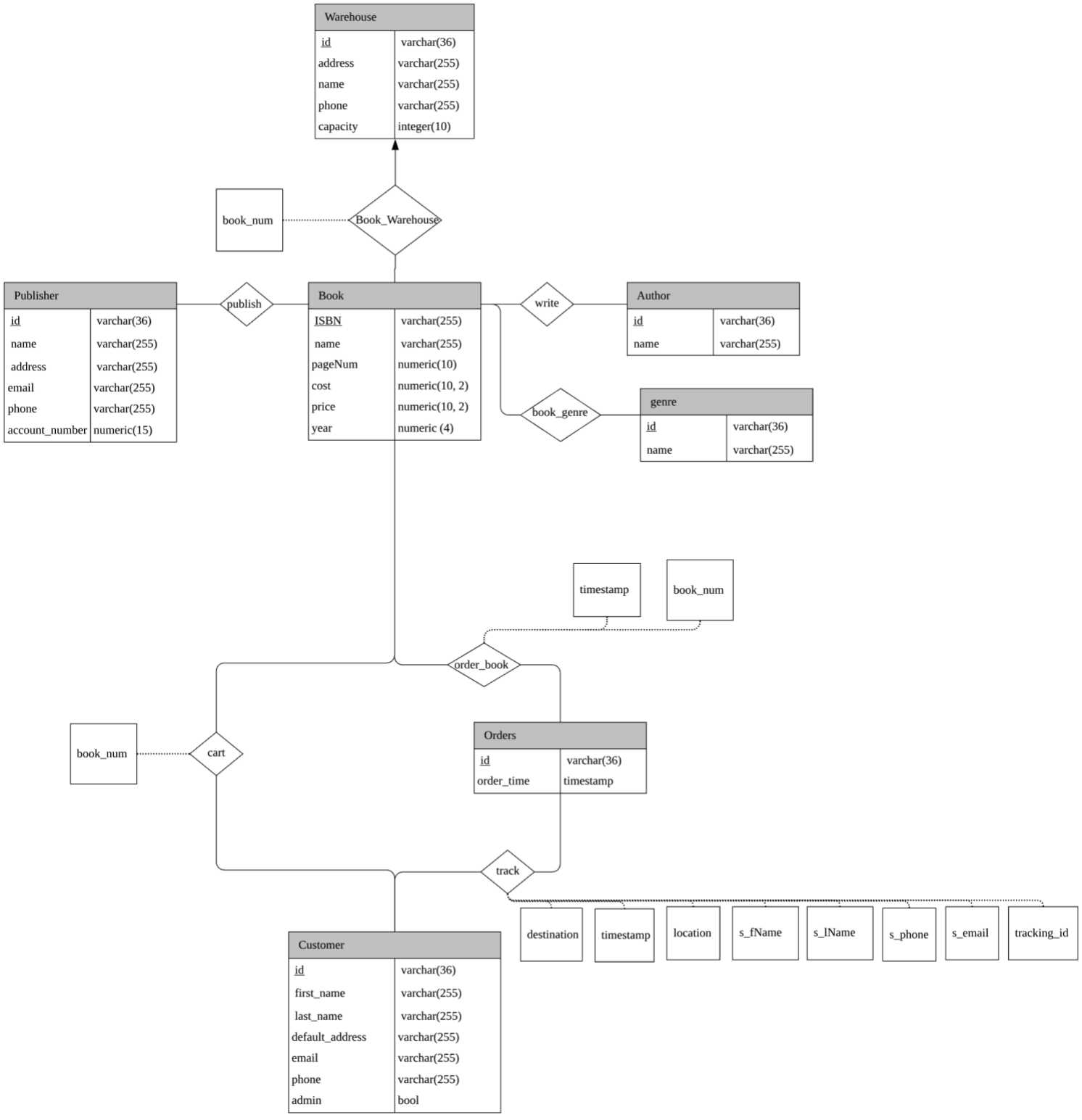
Book Store

Haowei Tu

1. ER Diagram:



2. Relation Schemas:

Book(ISBN, name, pageNum, cost, price, year)

Warehouse(id, address, name, phone, capacity)

Author(id, name)

Publisher(id, name, address, email, phone, accout_number)

Genre(id, name)

Book_genre(ISBN, genre_id)

Write(ISBN, author_id)

Publish(ISBN, publisher_id)

Orders(id, order_time)

Customer(id, first_name, last_name, default_address, email, phone, admin)

Track(order_id, customer_id, timestamp, location, destination, tracking_id, s_email, s_phone, s_fname, s_lname)

Order_book(ISBN, order_id, timestamp, book_num)

Cart(ISBN, customer_id, book_num)

3. Normalization (the order matches the above for readability):

Relations:

F = {

Bi --> Bn, Bpn, Bc, Bp, By = Bi_+

Wi --> Wa, Wn, Wp, Wc = Wi_+

Ai --> An = Ai_+

Pi --> Pn, Pa, Pe, Pp, Pan = Pi_+

Gi --> Gn = Gi_+

Oi --> Ot = Oi_+

Ci --> Cf, Cl, Cd, Ce, Cp, Ca = Ci_+

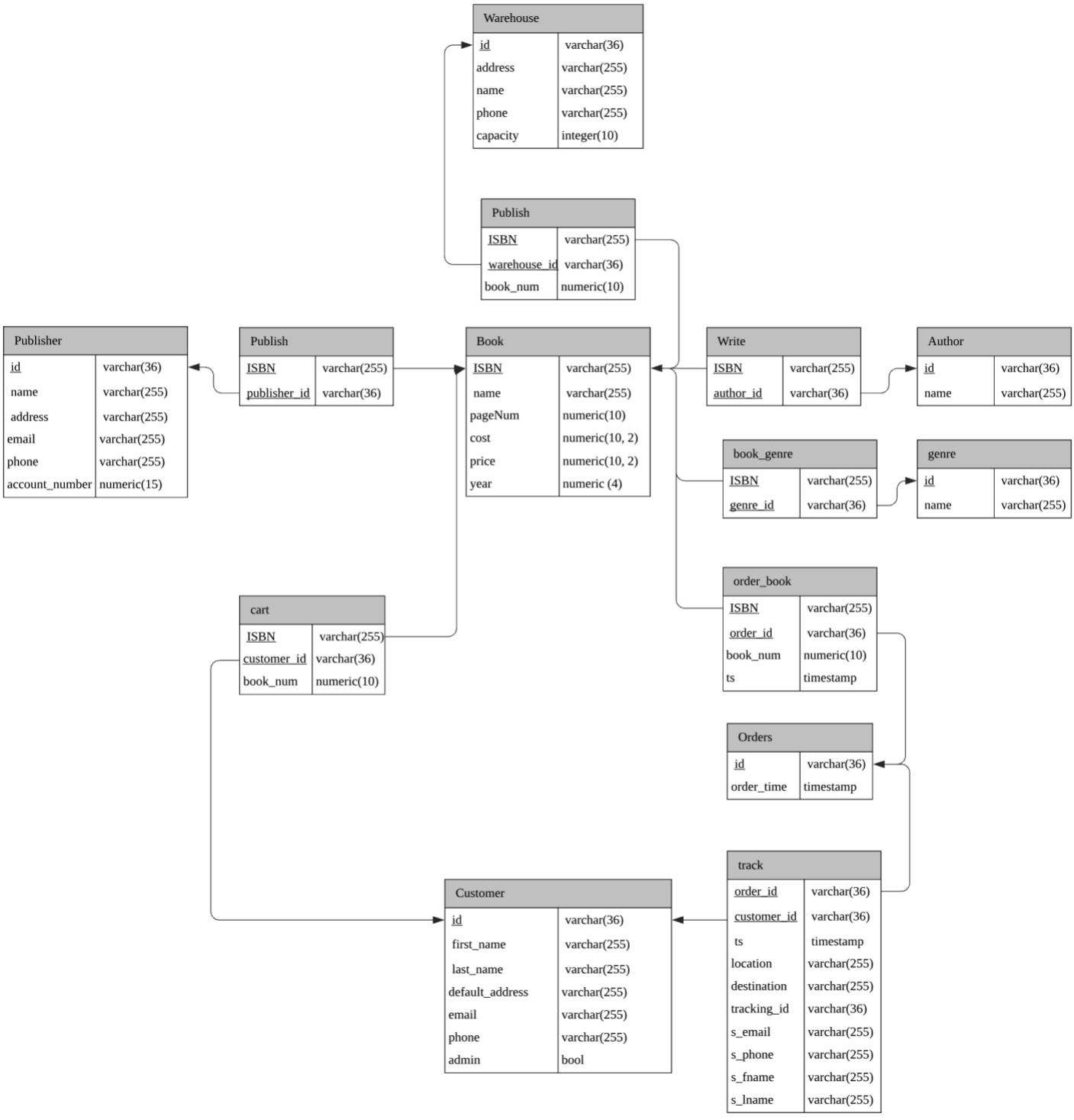
Oi, Ci --> Tt, Tl, Td, Ti, Tse, Tsp, Tsf, Tsl

Bi, Oi --> BOt, BOn

Bi, Ci --> BCn

}

4. Schema Diagram



Implementation:

The application is web-based using Javascript, Node.js, Express framework, MongoDB, PostgreSQL, the MongoDB contains the login session data and all of the actual bookstore information is stored in PostgreSQL database. Node.js uses a package called “pg” to interact with PostgreSQL.

Full Structure Diagram:

https://ibb.co/Bfp0MT6

backup link:

https://imgur.com/a/B7fijPS

The diagram provides a full illustration of every router and every function of the program with images.

Additionaly, when a book is searched by name, the search is case-insensitive and supports fuzzy search not limited to prefix and suffix. For example, if you search “potter”, it will show you all series of “Harry Potter”. Same goes for search by “author”, if you search “rowling”, it will give you all books of “J. K. Rowling”. However, searching by genre and ISBN only supports exact search for precision. Also, when you enter a book detail page (click on book URL), the recommended books by the same author will be randomly selected and shown to you at the bottom. (refresh to get new recommendations)

Also note that if the stock of a book is less than 50, it will set the new stock as the sales of this book from last month, however, if the sales from last month are zero, it will be set to 100.

Instructions for running the program:

To run the program, you need to:

1. Download PostgreSQL, create the database such the user is “postgres”, the password is “123456”, the port number “5432” and create a new database named “BookStore”, and then open SQL shell (psql), connect to the BookStore database (\connect BookStore), load the data from the BookStore.sql (\i BookStore.sql), finall leave the PostgreSQL running before launching the program
2. If you don’t have MongoDB on your device, download and install, use the default setting and have it running before launching the program.

   Windows: https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/

   Mac: https://docs.mongodb.com/manual/tutorial/install-mongodb-on-os-x/

3. Open the terminal / powershell / shell on your device, set the directory to /Proj, run “npm init”, and then “node server.js”, if there is no error message, then you are good to go.

GitHub:

https://github.com/winds-13/BookStore