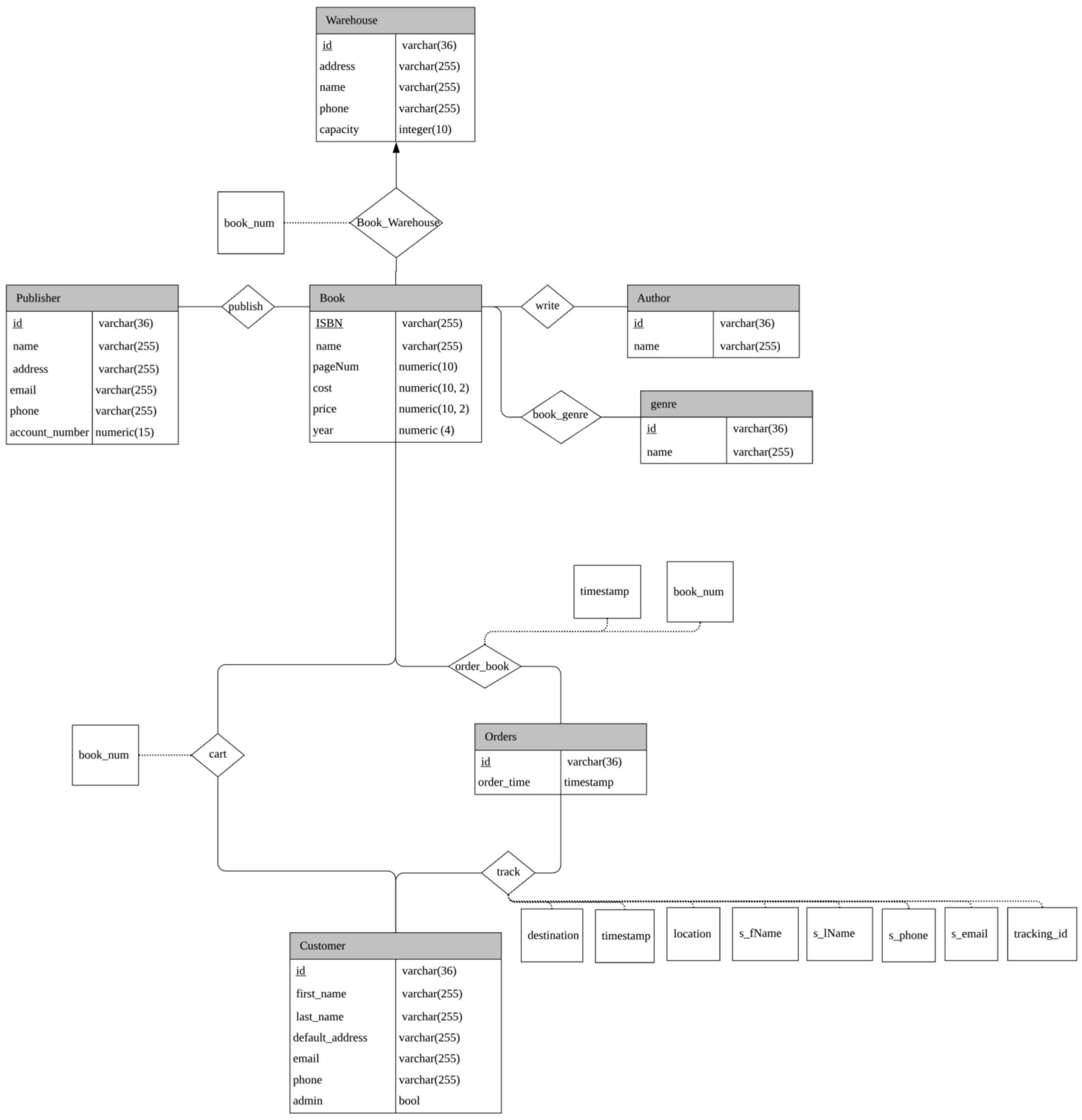
Book Store

Haowei Tu

1. ER Diagram:



About the ER diagram:

1. There is only one warehouse to do all the shipping
2. Every book has a list of “genre”s instead of one genre
3. Every book has a list of “author”s instead of one author
4. Every book has a list of “publisher”s instead of one publisher
5. Items in the shopping cart will be stored in the database, which means if you add some books in the cart and log out, when you come back logged in, those books in the cart will not disappear.
6. If you entered different information at the time of checking out, it will not affect all the default information stored in customer, an order will be created based on the information you entered at the time of checking out
7. Everything else is a “many-to-many” relationship except for the warehouse
8. In most cases, there is only one tracking in one order
9. The “customer” entity in the database contains two kinds of accounts, one is the admin account, the ‘admin’ attribute has a value of ‘true’ for admin accounts. The other one is the customer account, the ‘admin’ attribute of customer accounts has a value of ‘null’. Logging in to the admin account will enter the admin management page, logging into the customer account will enter the shopping page.

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^{+}

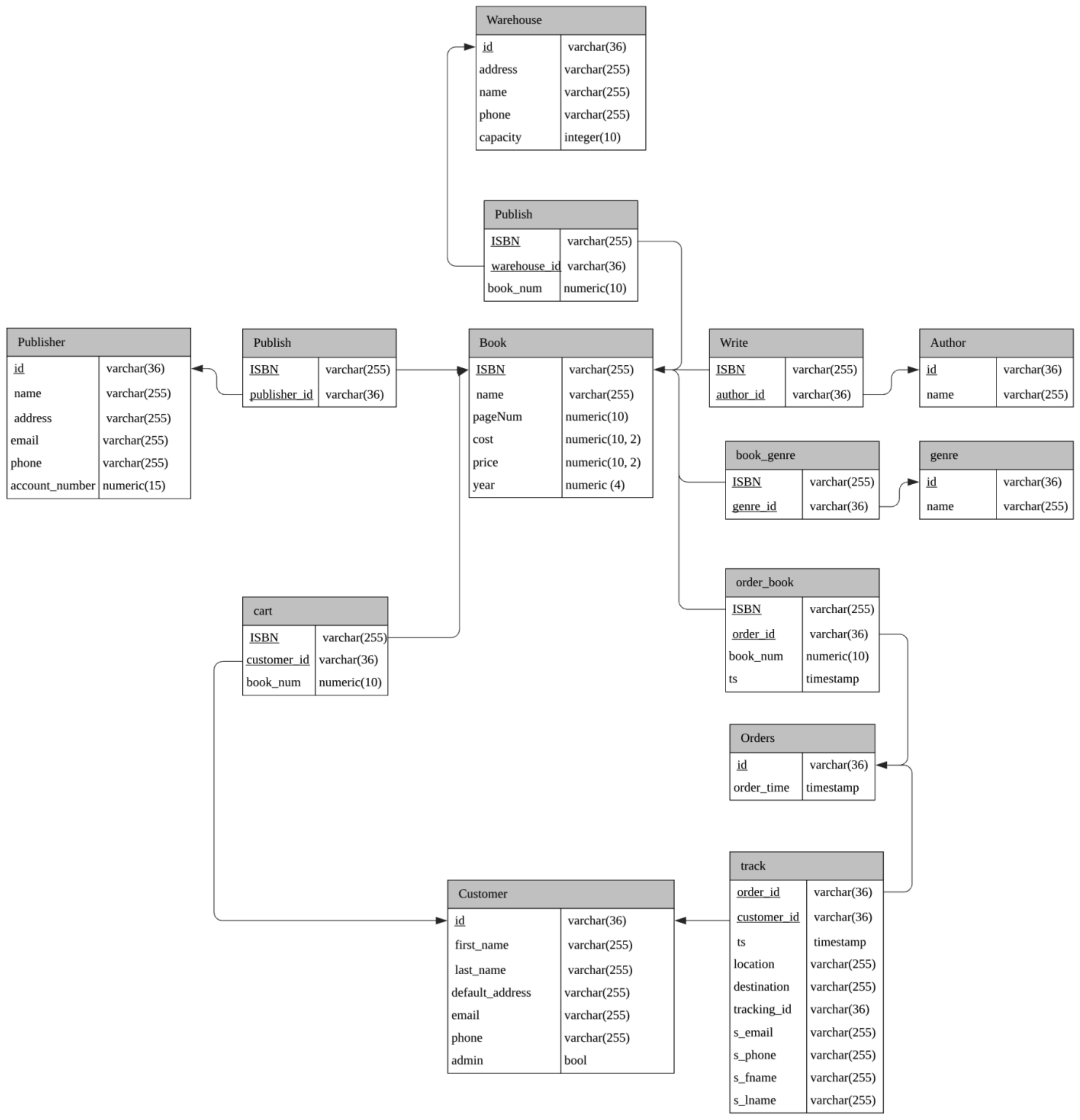
Oi, Ci --> Tt, Tl, Td, Ti, Tse, Tsp, Tsf, Tsl

Bi, Oi --> BOt, BOn

Bi, Ci --> BCn

}

4. Schema Diagram



Implementation:

The application is web-based using Javascript, Node.js, Express framework, MongoDB, PostgreSQL, although two databases being used, the MongoDB only contains the login session data, 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 you are searching a book 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 from “J. K. Rowling”. However, searching by genre and ISBN only supports exact search for precision. Sldo, when you enter a book detail page (click on book URL), the similar books by the same author will be randomly selected and shown to you at the bottom. (you will see a new one by refresh)

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:

I have already did the demonstration with a TA, however, if you are still going to run the program on your device, here are some instructions.

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

The .sql file is created by pg_dump, you can load the database directly by the step 1 above, (\i BookStore.sql), and then everything is imported.