Wentian Yao

101104458

Final project

Conceptual design

For this database, I have designed a 9 entity database,

For the Book store entity, contains bookstore id, bookstore name and physical address, for user to search.

There is also a entity book contains books infos, like ISBN, title, genre author etc...

A bookstore have a relationship with book called contains and can be accessed from bookstore for user to search.

Publisher entity contains publisher infos, like ID, name, email, address and bank account to transfer,

Author of book is also an entity which contains author's ID, name and email.

In order to purchase, there is a shopping cart entity called busket, (basket*),contains basket ID and books in it, have relationship with book named have and user named user's busket to identify who owns the basket.

While checking out, there is an entity named order infos which contains order ID to searc, customer id, busket id, the shipping address, date of purchase and total amount of the purchase. Orders are reported to the employer of the book store which is an entity contains employer;s infos, which are Id, names, email, phone number and password to login the database.

Shipping info is also an entity which contains the shipping id, status of shipping (pending or out for shipping, or packaging etc..), and estimate time of arrival.

The user entity have users ID, infos to login(username and password), contact infos(phone, email, address). Users need to register to bookstore database so there is a relationship between user and bookstore named register.
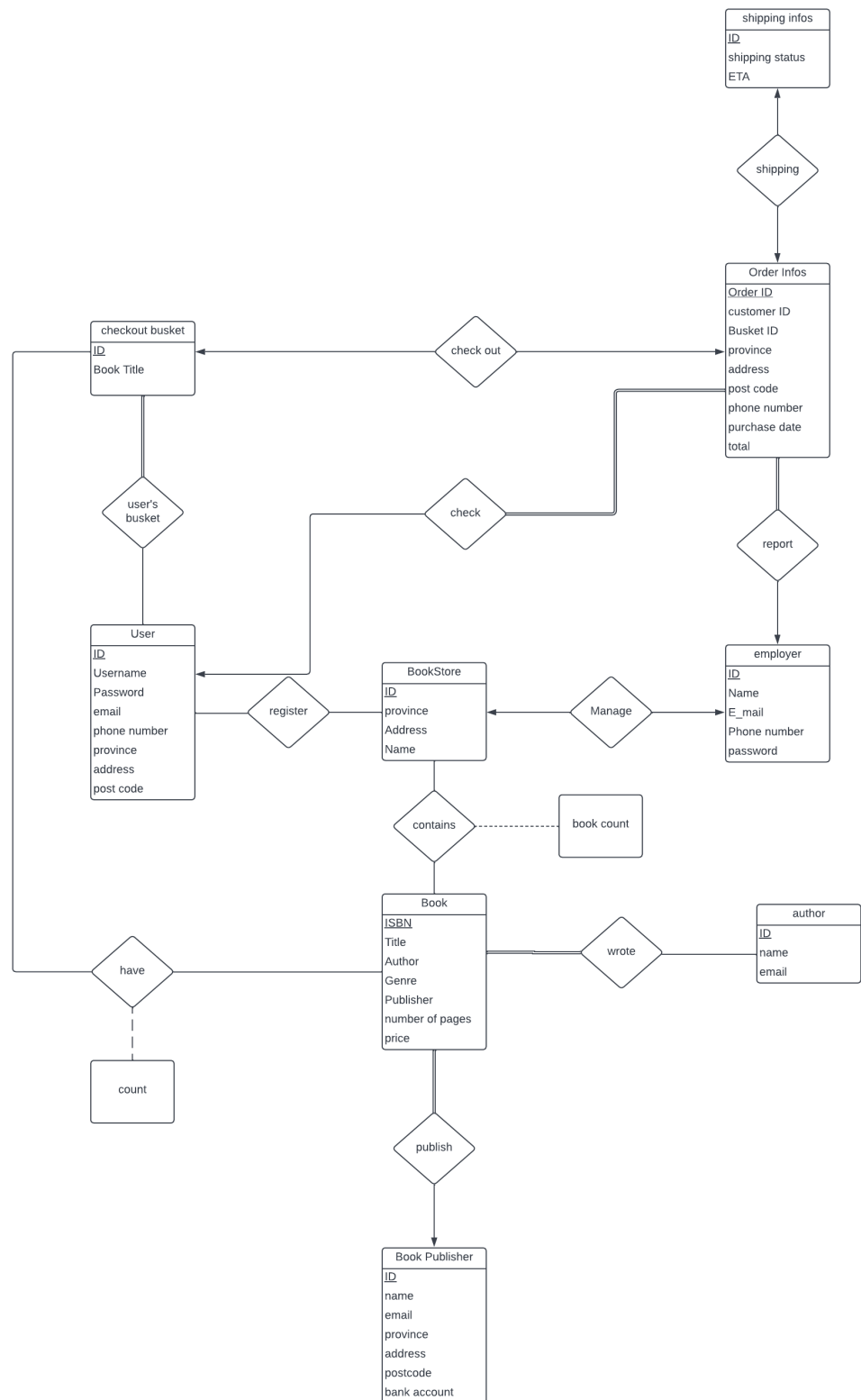
For searching books with genre, the database should access each book it have and check if book's genre matches user input's genre and return the result.

For tracking shipping status, book store can access employee->order infos-> shipping infos and track both order id and shipping id, but order ID is more recommended.

As assumed there is only one warehouse, I don't list it in the database, which Can be added as an entity with address and name, have relationship with books called "stored in", and all of these books points to the same warehouse.

For adding or remove books in the store, the employer can access to the bookstore database and do dql.

ER Diagram

shipping infos
- ID
- shipping status
- ETA

◇ shipping

Order Infos
- Order ID
- customer ID
- Busket ID
- province
- address
- post code
- phone number
- purchase date
- total

checkout busket
- ID
- Book Title

◇ check out

◇ user's busket

◇ check

◇ report

employer
- ID
- Name
- E_mail
- Phone number
- password

User
- ID
- Username
- Password
- email
- phone number
- province
- address
- post code

◇ register

BookStore
- ID
- province
- Address
- Name

◇ Manage

◇ contains ------ book count

Book
- ISBN
- Title
- Author
- Genre
- Publisher
- number of pages
- price

◇ wrote

author
- ID
- name
- email

◇ have

count

◇ publish

Book Publisher
- ID
- name
- email
- province
- address
- postcode
- bank account

Reduction to Relation Schemas

User {<u>User_ID</u>, Username, password, email, phone number, province, address, post code}

Bookstore {<u>ID</u>, employer_id, name, province, address, Postcode }

Book {<u>ISBN</u>, Title, Genre, Publisher_ID, number of pages, rate of transfer}

Publisher { <u>ID</u>, publisher name, email, Bank account, province, address, post code}

Author {<u>ID</u>, name, email}

Employer {<u>ID</u>, Name, email, phone number, password}

Busket {<u>Busket ID</u>, <u>User_ID</u>}

Order infos{<u>Order ID</u>, Employer_ID, User ID, Busket ID, province, address, postcode, phone number, purchase date, total}

Shipping infos {<u>shipping id</u>, shipping status, ETA}

Relations:

Wrote {<u>ISBN, Author_ID</u>}

Register {<u>User_Id, Bookstore_ID</u>}

Contains {<u>ISBN, Bookstore_ID</u>, amount}

Have {<u>Busket_ID, ISBN</u>, amount}

Normalization:

1.

R = {ISBN, Title, Genre, Publisher name, number of pages, rate of transfer, ID, email,

Bank account, province, address, post code }

F = {

    ISBN -> Title, Genre, Publisher_ID, number of pages, rate of transfer

    Publisher_ID-> publisher name, email, Bank account, province, address, post code

}

Publisher_ID = ID

So,

R1:Book = {ISBN, Title, Genre, Publisher_ID, number of pages, rate of transfer, Price}

R2:Publisher= {Publisher_ID, name, email, Bank Account, province, address, postcode}

R1 n R2 -> R2

Good normal form

2.

R = {ISBN, Title, Genre, Publisher_ID, number of pages, rate of transfer }

F = {ISBN -> Title, Genre, Publisher_ID, number of pages, rate of transfer}

Good normal form

3.

R = {ID, name, email}

F = {ID->name, email}

ID is super key, good normal form

4.

R = Publisher {<u>ID</u>, publisher_name, email, Bank account, province, address, post code}

F = {ID->publisher_name, email, Bank account, province, address, post code }

ID is super key, good normal form

5.

R= User {<u>User_ID</u>, Username, password, email, phone number, province, address, post code}

F = {User_ID-> Username, password, email, phone number, province, address, post code}

User_ID is super key and good Normal form

6.

Bookstore {<u>Bookstore_ID</u>, employer_id, name, province, address, Postcode,}

Employer {<u>ID</u>, Name, email, phone number, password}

F = {Bookstore_ID -> employer_id, name, province, address, Postcode

ID-> Name, email, phone number, password

ID = employer_id}

Bookstore n Employer ->{employer_id}

Database Schema

Author
- Author_ID
- Name
- Email

Wrote
- ISBN
- Author_ID

Book
- ISBN
- Title
- Genre
- Publisher_ID
- Number_of_pages
- Rate_of_transfer

Publisher
- Publisher_ID
- Name
- Email
- Bank_Account
- Province
- Address
- Postcode

Have
- ISBN
- Busket_Number
- Amount

Contains
- ISBN
- Bookstore_ID
- amount

Busket
- Busket_ID
- User_ID

Bookstore
- Bookstore_ID
- Employer_ID
- Name
- Province
- Address
- Postcode

Register
- Bookstore_ID
- User_ID

Employer
- Employer_ID
- Name
- Email
- Phone_Number
- Password

User
- User_ID
- UserName
- Password
- Email
- Phone_Number
- Province
- Address
- Postcode

Shipping_Info
- Shipping_ID
- Order_ID
- Status
- ETA

Order
- Order_ID
- Employer_ID
- User_ID
- Busket_ID
- Phone Number
- Province
- Address
- Postcode
- Date
- Total