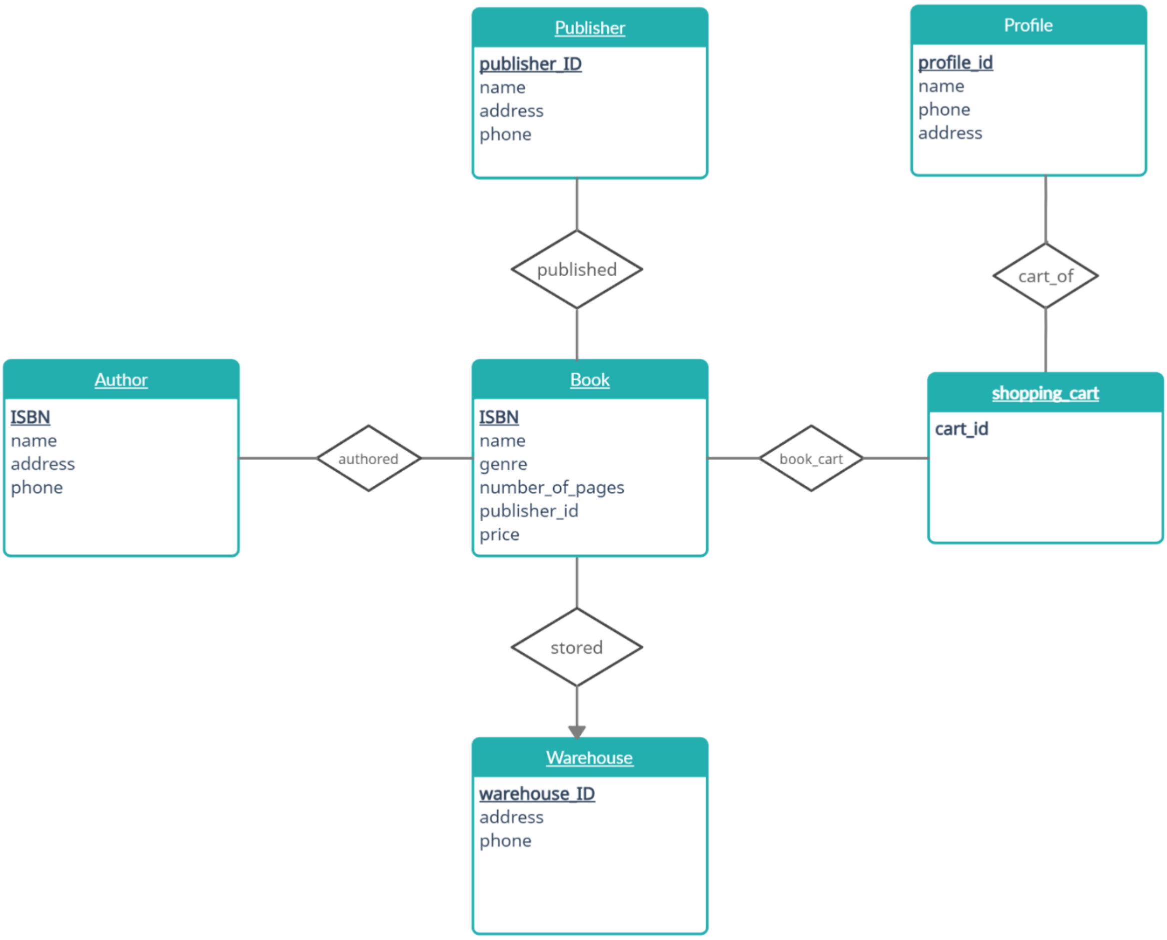
COMP 3005 Final Project Report

Online Bookstore

Prepared by:
Gift Akinpelu 101151897
Martin Macharia 101170692

Conceptual Design



Our conceptual design is based entirely on the information provided in the problem statement. The main assumption needed to run the database is that there is only one warehouse, as detailed in the problem statement. Otherwise, the design is solely based on the requirements in the problem statement with no other assumptions added to it.

Reduction to Relation Schemas

Publisher(publisher_id, name, address, phone)
Published(published_id, ISBN)
Author(ISBN, name, address, phone)
Authored(ISBN)
Book(ISBN, name, genre, number_of_pages, publisher_id,price)
Stored(ISBN, warehouse_ID)
Warehouse(warehouse_ID, address, phone)
Profile (profile_id, name, phone, address)
Cart_of(profile_id)
Book_cart(ISBN)
Shopping cart(cart_id)

Normalization of Relation Schemas

All the relations are in 3NF because each table has just one F.D therefore there is no extraneous attribute(each attribute is important and non-redundant). Therefore, I just need to check for BCNF

//author
author(ISBN,name, address, phone);
F ={
ISBN -> name, address, phone
}
The relation is in good normal form in BCNF because the author_id is a superkey.
ISBN is in BCNF because [ISBN]+ = Author_ID, name, address, phone

//publisher
publisher(publisher_id,name, address, phone);
F = {
Publisher_id -> name, address, phone
}
The relation is in good normal form in BCNF because the publisher_id is a superkey.
Publisher_ID is in BCNF because [Publisher_ID]+ = Publisher_ID, name, address, phone

//Book
book(ISBN, name, genre, number_of_pages,publisher_id, price);
F = {
ISBN -> name, genre, number_of_pages,publisher_id,price
}
The relation is in good normal form in BCNF because the ISBN is a superkey
ISBN is in BCNF because [ISBN]+ = ISBN, name,genre, number_of_pages, price

//Warehouse
warehouse(warehouse_id, address, phone)
F = {
Warehouse_id -> address, phone
}
The relation is in good normal form in BCNF because the Warehouse_id is a superkey
Warehouse_id is in BCNF because [Warehouse_id]+ = Warehouse_id, address, phone

//Profile
profile(profile_id, name, address, phone)
F = {
profile_id -> name, address, phone
}
The relation is in good normal form in BCNF because the User_id is a superkey
profile_id is in BCNF because [profile_id]+ = profile_id, name, address, phone

//Authored
Authored(ISBN)
F= {

ISBN -> ISBN
}
The relation is in good normal form in BCNF because it has one functional dependency which is trival. ie. ISBN is a subset of ISBN

//Published
Published(published_id, ISBN)
F= {
published_id, ISBN -> ISBN
}
The relation is in good normal form in BCNF because it has one functional dependency which is trival. ie. ISBN is a subset of publisher_id and ISBN

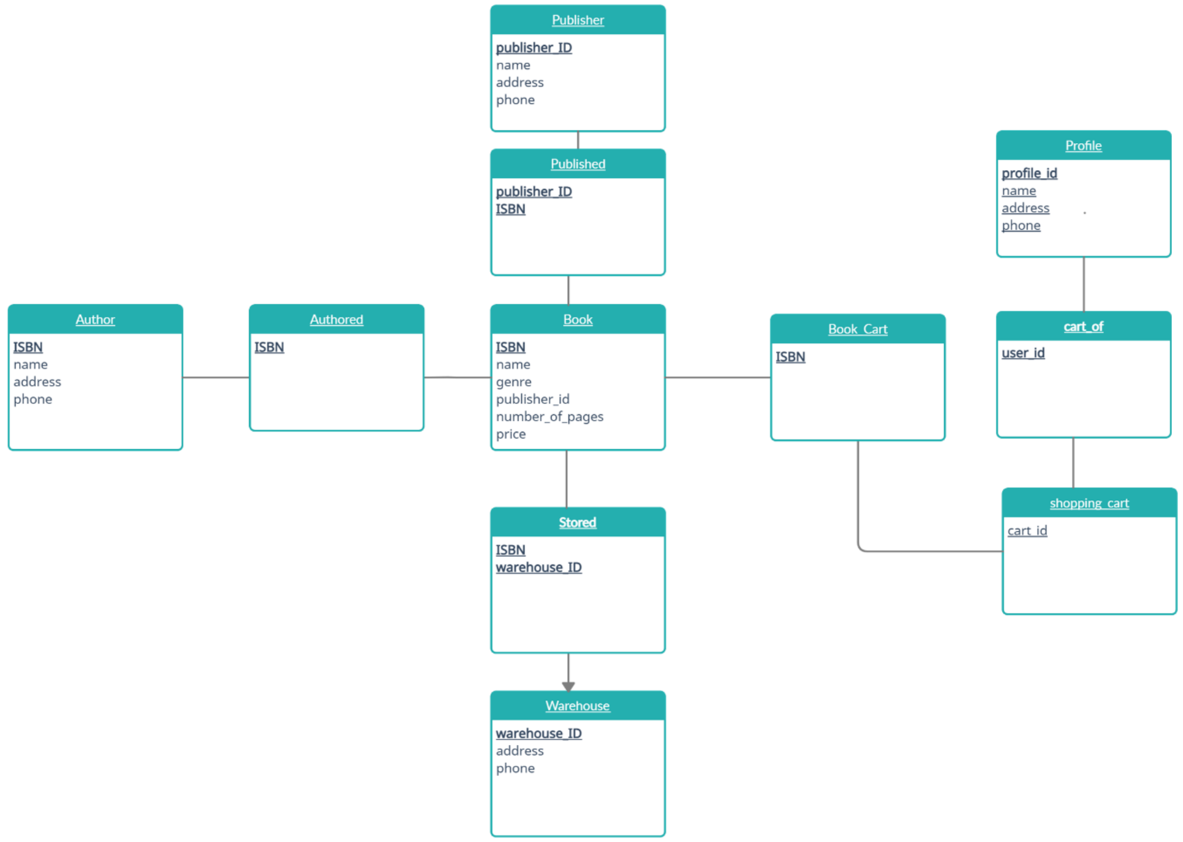
//stored
stored(ISBN, warehouse_id)
F = {
ISBN -> warehouse_id
}
The relation is in good normal form in BCNF because the ISBN is a superkey
ISBN is in BCNF because [ISBN]+ = ISBN, Warehouse_id

//Book_Cart
Book_Cart(ISBN)
F= {
ISBN -> ISBN
}
The relation is in good normal form in BCNF because it has one functional dependency which is trival. ie. ISBN is a subset of ISBN

//Cart_of
Cart_of(profile_id)
F= {
profile_id -> profile_id
}
The relation is in good normal form in BCNF because it has one functional dependency which is trival. ie. profile_id is a subset of profile_id

//shopping_cart
shopping_cart(cart_id)
F= {
cart_id -> cart_id
}
The relation is in good normal form in BCNF because it has one functional dependency which is trival. ie. cart_id is a subset of cart_id

Database Schema Diagram



Implementation

We were not able to fully implement the online bookstore application. However, the vision behind it was that we would connect to our SQL database using a get request in books.js. After running our DDL and DML scripts, we would have made our database available to be connected to. Then, our get request in books.js would find all the books that have been uploaded and send them to our /books webpage. Similarly, our /books/add page would add a book to the database following the parameters we have, and to the /books webpage as well.

GitHub Repository

Kindly follow this link:

https://github.com/Marsh90/OnlineBookstore

Appendix

Time availability:

- 11:00am
- 12:00pm
- 01:00pm