

CARLETON UNIVERSITY
OTTAWA, ON

COMP3005 - FALL 2022

SECTION B

Project Report

Authors:
Justin Sward

Student Number:
100683372

Date Submit:
December 11, 2022

Contents

1	Introduction	3
2	Conceptual Design	4
2.1	Entities and Relationships	5
2.1.1	Book	5
2.1.2	Publisher	5
2.1.3	Author	5
2.1.4	Genre	5
2.1.5	Order_Manifest	5
2.1.6	Orders	6
2.1.7	Useraccounts	6
2.1.8	Commissions	6
2.2	Assumptions	6
3	Reduction to Relation Schemas	7
4	Normalization of Relation Schemas	8
4.1	Functional Dependencies	8
4.1.1	BOOK	8
4.1.2	BOOK_AUTHORS	8
4.1.3	AUTHOR	8
4.1.4	GENRE	8
4.1.5	PUBLISHER	8
4.1.6	COMMISSIONS	9
4.1.7	ORDER_MANIFEST	9
4.1.8	ORDERS	9
4.1.9	USERACCOUNTS	9
5	Database Schema Diagram	10
6	Implementation	11
6.1	Overview and Instructions	11
6.2	Screenshots	12
6.3	Code Snippets	12
7	Github Repository	17
8	Appendix I: Availability	17

List of Figures

1	ER diagram of the database for the Look Inna Book bookstore.	4
2	Schema diagram of the database for the Look Inna Book bookstore.	7
3	Final schema of the database for the Look Inna Book bookstore.	10

4	Screen capture of the Home page.	12
5	Screen capture of the Search page.	13
6	Screen capture of the Orders page.	14
7	Screen capture of the Admin page via an administrator account.	14
8	Screen capture of the AddBook page via an administrator account.	15
9	Screen capture of the Users page via an administrator account via an administrator account. . .	15
10	Screen capture of the AddUser page via an administrator account.	16
11	Screen capture of the database being queried in SQLite3 in a terminal.	16

1 Introduction

This report details the solution to the final project for COMP3005.

This project involves the design and creation of a database-based application for the purpose of creating a fictional bookstore called Look Inna Book.

The design requirements are specified by the project instructions for the course.

2 Conceptual Design

Figure 1, below, shows the overall entity-relationship diagram for the conceptual design of the database for the bookstore application.

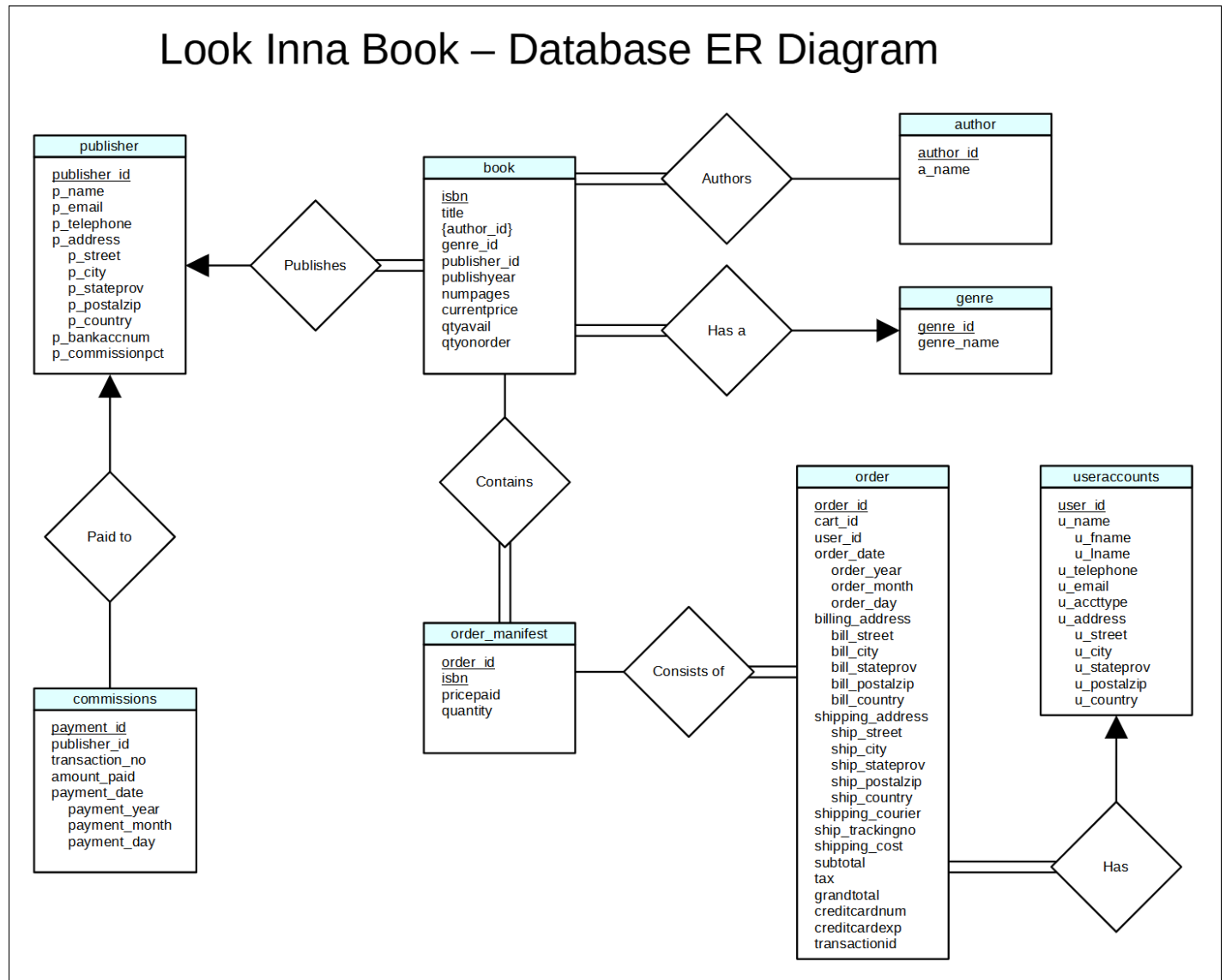


Figure 1: ER diagram of the database for the Look Inna Book bookstore.

The premise of this design is explained in the following subsections.

2.1 Entities and Relationships

2.1.1 Book

The Book entity contains the information regarding each book available in the bookstore. The primary key is the ISBN, which is the International Standard Book Number, unique to each book in print. Using this as the primary key guarantees it will be unique to a single book and is a globally-used identifier for each book.

Books must have one publisher relationship, one genre, and at least one author.

Many books can be added to a cart and many carts can contain any one book.

2.1.2 Publisher

This entity contains the information for each of the book publishers. This includes the publisher name, address, contact information, as well as the bank account number and commission rate to be used for processing periodic commission payments. Each book must have one publisher that publishes the book, but a publisher can publish many books.

2.1.3 Author

The Author entity contains the name information to be stored for each author. By using a unique author ID, searching can be made more quickly as the search process for integers is much quicker than text string matching. Each book must have at least one author and the author can write many books.

2.1.4 Genre

The Genre entity provides the name information of the genre ID used in the books. Each book must have a genre associated with it and genres can relate to many books.

2.1.5 Order_Manifest

The Order_Manifest is where the information on the items purchased in each order, quantity purchased, and amount paid are stored. The price of the books can change so the price paid by the user is recorded.

This relation only has a few important attributes, but it can be combined with other relations to have extremely detailed information that can be used to generate various detailed sales reports.

2.1.6 Orders

An order combines the cart for the current session with the user account purchasing the books and billing and shipping addresses to fulfill the order. The shipping information (carrier, tracking number) is also stored in the orders, as well as the date of the order. The date can be used to look up the sales information by selecting all of the orders for a given time period.

An order only exists if a cart with the items included for the order exists, and it must have a user account associated with it.

2.1.7 Useraccounts

The Useraccounts entity stores the information for each user account. This is used to process the books from the cart into an order. The user's email address must be unique, but the phone number may not be (such as the case of multiple users with a shared home telephone). Note that the address is not stored on file for the user and must be provided at the check-out to complete the order. User account type is used to determine the privileges associated with the account; administrator accounts can order more books, add or remove books from the inventory or catalogue, and update the information of any user account. They can also generate reports using the order information.

2.1.8 Commissions

This is a record of all of the commission transactions made. The process is to be automated but a record is kept of all of the transactions for the financial records.

2.2 Assumptions

The following assumptions were included in this design:

- The cart will be based on the current user session
- The bookstore uses only one shipping courier
- Only one ISBN is registered to each book (there are potential exceptions to this in practice)

3 Reduction to Relation Schemas

Figure 2, below, shows the database schema after being reduced from the ER-diagram from the previous section.

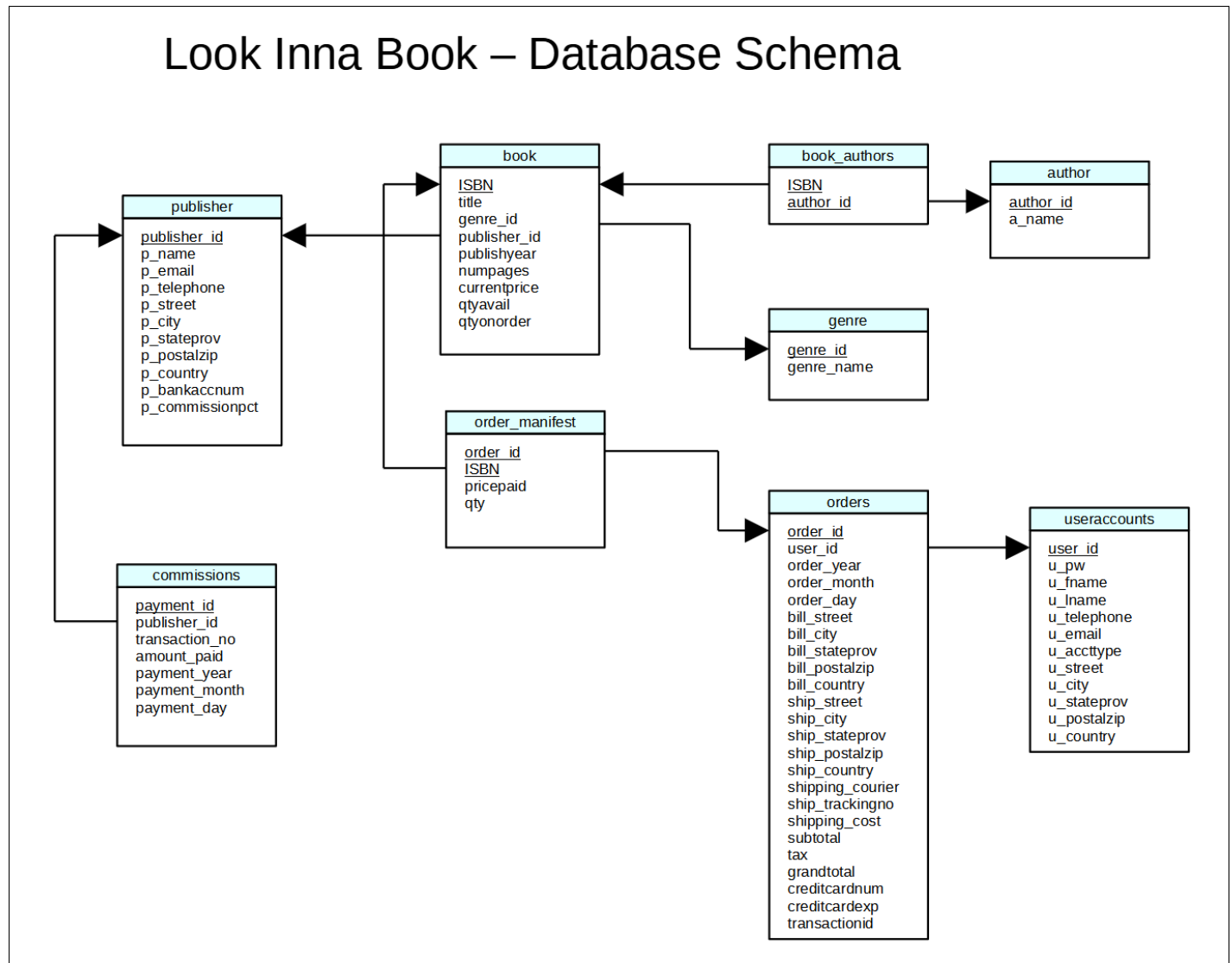


Figure 2: Schema diagram of the database for the Look Inna Book bookstore.

4 Normalization of Relation Schemas

4.1 Functional Dependencies

4.1.1 BOOK

$\text{ISBN} \rightarrow \text{title, genre_id, publisher_id, publishyear, numpages, currentprice, qtyavail, qtyonorder}$

This is already in BCNF as each attribute in the table depends on the key and only the key (The LHS is a superkey for the functional dependency for this relation).

4.1.2 BOOK_AUTHORS

$\text{ISBN, author_id} \rightarrow \text{ISBN, author_id}$

This is already in BCNF as each attribute in the table depends on the key and only the key. The functional dependency is trivial as the superkey makes up the only attributes in this relation.

4.1.3 AUTHOR

$\text{author_id} \rightarrow \text{auth_fname, auth_mname, auth_lname}$

This is already in BCNF as each attribute in the table depends on the key and only the key (The LHS is a superkey for the functional dependency for this relation).

4.1.4 GENRE

$\text{genre_id} \rightarrow \text{genre_name}$

This is already in BCNF as each attribute in the table depends on the key and only the key (The LHS is a superkey for the functional dependency for this relation).

4.1.5 PUBLISHER

$\text{publisher_id} \rightarrow \text{p_name, p_email, p_telephone, p_street, p_city, p_stateprov, p_postalzip, p_country, p_bankaccnum, p_commissionpct}$

This is already in BCNF as each attribute in the table depends on the key and only the key (The LHS is a superkey for the functional dependency for this relation).

4.1.6 COMMISSIONS

payment_id \rightarrow publisher_id, transaction_no, amount_paid, date_paid

This is already in BCNF as each attribute in the table depends on the key and only the key (The LHS is a superkey for the functional dependency for this relation).

4.1.7 ORDER_MANIFEST

order_id, ISBN \rightarrow pricepaid, qty

This is already in BCNF as each attribute in the table depends on the key and only the key (The LHS is a superkey for the functional dependency for this relation).

4.1.8 ORDERS

order_id \rightarrow cart_id, user_id, order_year, order_month, order_day, bill_street, bill_city, bill_stateprov, bill_postalzip, bill_country, ship_street, ship_city, ship_stateprov, ship_postalzip, ship_country, shipping_courier, ship_tracking_no, shipping_cost, subtotal, tax, grandtotal, creditcardnum, creditcardexp, transactionid

This is already in BCNF as each attribute in the table depends on the key and only the key.

4.1.9 USERACCOUNTS

user_id \rightarrow u_fname, u_lname, u_telephone, u_email, u_accttype, u_street, u_city, u_stateprov, u_postalzip, u_country

This is already in BCNF as each attribute in the table depends on the key and only the key (The LHS is a superkey for the functional dependency for this relation).

5 Database Schema Diagram

Figure 3, below, shows the final database schema after being normalized. No changes were required since each relation was already in BCNF (Boyce-Codd Normal Form).

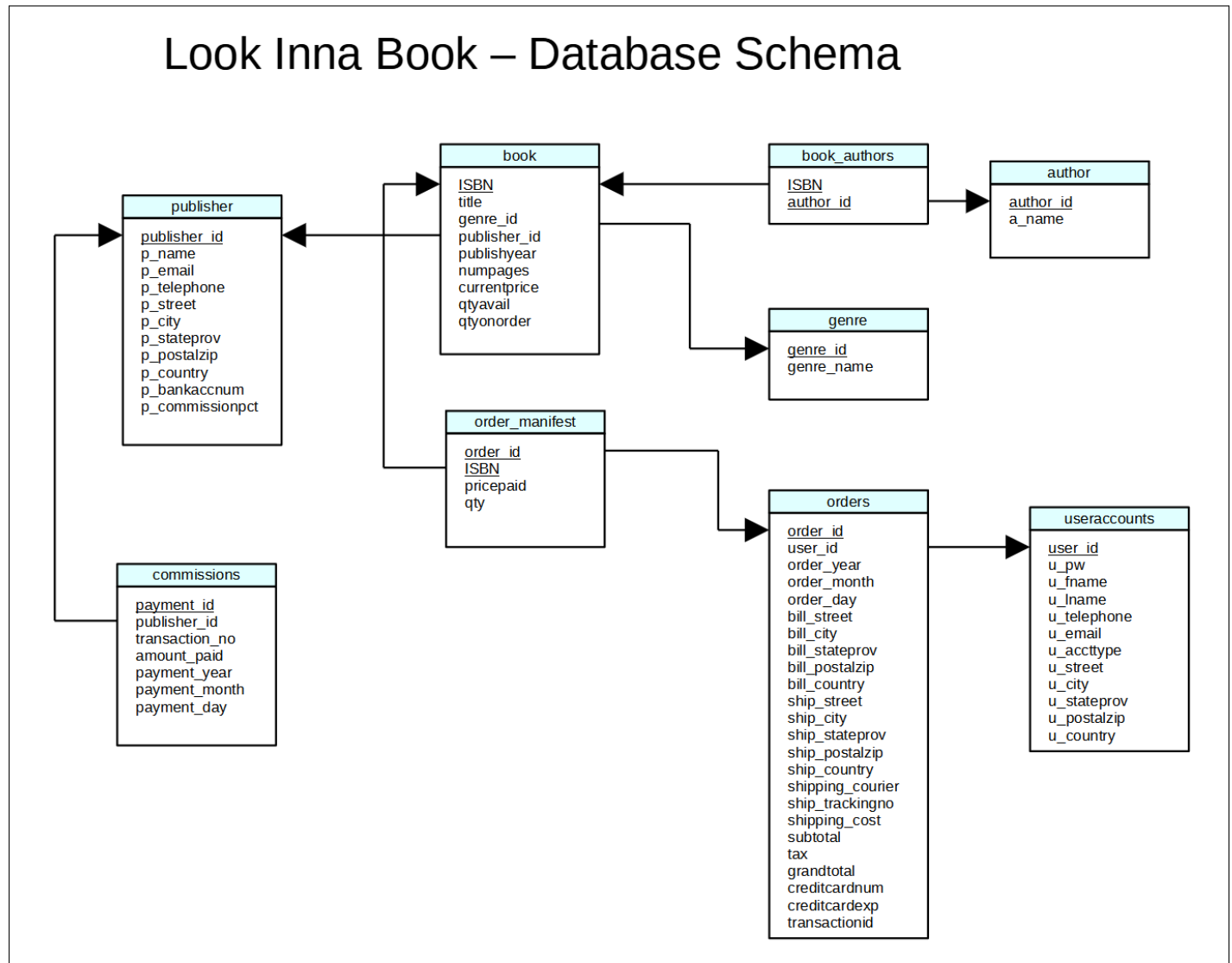


Figure 3: Final schema of the database for the Look Inna Book bookstore.

6 Implementation

To be completed!

6.1 Overview and Instructions

The application was implemented using a Python web server application via the Flask web framework and the Jinja web templating engine.

The backing database was implemented using MySQL running on a host server. The Python web server application communicates directly with the database via the API that communicates with the MySQL server.

The web app itself uses HTML5 to render the data and interact with the user through a web browser, with CSS used to stylize and format the interface.

To run the Python application, the virtual environment has to be invoked (must be installed). From the working directory with the Python main.py application, enter the following commands in the terminal:

- `virtualenv env`
- `source env/bin/activate`
- `python.py`

This will run the backing engine that hosts the web service and communicates with the backing server. The web application can be accessed in the web browser by going to the appropriate IP address and port configured on your system (ie. 127.0.0.1:5000).

6.2 Screenshots

6.3 Code Snippets

InnaBookStore

[Home](#) [Search](#) [Cart](#) [Orders](#) [Account](#) [Admin](#)

Log In

Current User: 0

[userlogin](#)


No account yet? [Register here!](#)

About Us

The InnaBookStore is created for the COMP3005 final project to experiment with using a database-backed user application.

The table below provides the contact information for the author of the web app.

Student	Contact
Justin Sward	justinsward@gmail.com



2022 - Carleton University

Figure 4: Screen capture of the Home page.

InnaBookStore

[Home](#)[Search](#)[Cart](#)[Orders](#)[Account](#)[Admin](#)

Book Search

Search Results

ISBN	Title	Genre	Publisher	Year	Pages	Price	Qty. Available	Qty. On Order
123123123	Test 123	1	2	1999	55	21.0	0	5
9780553211764	A Tale of Two Cities	6	1	1989	369	14.99	5	0
9780553250251	The Catcher in the Rye	6	2	1964	214	17.99	1	0
9780812978001	The Law of Dreams	2	1	2007	416	9.99	10	5
9781853260049	Great Expectations	2	3	1992	395	22.99	7	0

2022 - Carleton University

Figure 5: Screen capture of the Search page.

InnaBookStore

Home

Search

Cart

Orders

Account

Admin

Orders

Order Number	Customer ID	Year	Month	Day	Country	Total	Courier	Tracking Number	Transaction ID
1	4	2022	11	30	Canada	39.53	UPS	0009387478CA1	054985
2	3	2022	12	1	United States	77.94	FedEx	893HF8398	858875054

Figure 6: Screen capture of the Orders page.

InnaBookStore					
Home	Search	Cart	Orders	Account	Admin
Administrative Controls					
ADD BOOKS TO REPOSITORY					
MODIFY EXISTING BOOKS					
VIEW USER ACCOUNTS					
ADD NEW USER					
VIEW SALES					
VIEW COMMISSION PAYMENTS					

Figure 7: Screen capture of the Admin page via an administrator account.

The screenshot shows the InnaBookStore Admin interface. At the top is a navigation bar with links: Home, Search, Cart, Orders, Account, and Admin (highlighted in purple). Below the navigation bar is the 'Add Book' section. It contains a form with the following fields:

ISBN	ISBN
Title	Title
Genre ID	Genre ID
Publisher ID	Publisher ID
Year	Year
# of Pages	# of Pages
Price	Price
Inventory Qty.	Inventory
Qty. Ordered	On-Order
Add Book	

Figure 8: Screen capture of the AddBook page via an administrator account.

The screenshot shows the InnaBookStore Admin interface. At the top is a navigation bar with links: Home, Search, Cart, Orders, Account, and Admin (highlighted in purple). Below the navigation bar is the 'User Accounts' section. It contains two links: [ADD NEW USER](#) and [MODIFY EXISTING NEW USER](#). Below these links is a table of user accounts:

User ID	First Name	Last Name	Telephone	Email Address	Street	City	State/Province	Postal/zip	Country
1	Store	Owner	6132555411	justin@carleton.ca	None	None	None	None	Canada
2	Ed	Smith	4165552500	edsmith@yahoo.biz	None	None	None	None	Canada
3	Amar	Khan	3035558473	amar@khan.ca	2 Richmond Road	Los Angeles	California	71058	United States
4	Rachelle	Louis	5196825099	rl@abcinc.ca	10-387 Rue De La Fontaine	Montreal	Quebec	H8C3G0	Canada
5	Frank	Vincent	4165558877	frank@frank.frank	52 York Ave	Toronto	Ontario	J9P 6W7	Canada
6	Greg	Adams	8723673	ga@ga.ga	None	None	None	None	Mexico
7	Lionel	Richie	3035552849	lionel@richie.lr	None	None	None	None	United States
8	Joel	Essen	5555555555	joel@essen.biz.org.co.uk	None	None	None	None	Scotland

Figure 9: Screen capture of the Users page via an administrator account via an administrator account.

InnaBookStore

[Home](#)
[Search](#)
[Cart](#)
[Orders](#)
[Account](#)
[Admin](#)

Add User Account

Password	<input type="password"/>
First Name	<input type="text"/>
Last Name	<input type="text"/>
Email Address	<input type="text"/>
Telephone	<input type="text"/>
Country	<input type="text"/>
<input type="button" value="Add User"/>	

Figure 10: Screen capture of the AddUser page via an administrator account.

```

SQLite version 3.40.0 2022-11-16 12:10:08
Enter ".help" for usage hints.
sqlite> .header on
sqlite> .mode box
sqlite> select * from book;
  
```

isbn	title	genre_id	publisher_id	publishyear	numpages	currentprice	qtyavail	qtyonorder
123123123	Test 123	1	2	1999	55	21.0	0	5
9780553211764	A Tale of Two Cities	6	1	1989	369	14.99	5	0
9780553250251	The Catcher in the Rye	6	2	1964	214	17.99	1	0
9780812978001	The Law of Dreams	2	1	2007	416	9.99	10	5
9781853260049	Great Expectations	2	3	1992	395	22.99	7	0

```

sqlite> select * from useraccounts;
  
```

u_id	u_pw	u_fname	u_lname	u_telephone	u_email	u_accttype	u_street	u_city	u_stateprov	u_postalzip	u_country
1	admin	Store	Owner	6132555411	justin@carleton.ca	1					Canada
2	123	Ed	Smith	4165552500	edsmith@yahoo.biz	0					Canada
3	qwe	Amar	Khan	3035558473	amar@khan.ca	0	2 Richmond Road	Los Angeles	California	71058	United States
4	asd	Rachelle	Louis	5196825099	rl@abcinc.ca	0	10-387 Rue De La Fontaine	Montreal	Quebec	H8C3G0	Canada
5	qwe	Frank	Vincent	4165558877	frank@frank.frank	0	52 York Ave	Toronto	Ontario	J9P 6W7	Canada
6	111	Greg	Adams	8723673	ga@ga.ga	0					Mexico
7	222	Lionel	Richie	3035552849	lionel@richie.lr	0					United States
8	333	Joel	Essen	5555555555	joel@essen.biz.org.co.uk	0					Scotland

```

sqlite>
  
```

Figure 11: Screen capture of the database being queried in SQLite3 in a terminal.

7 Github Repository

The code used for the implementation of the database program can be found in the following online GitHub repository:

`https://github.com/JustinSward/bookstore`

8 Appendix I: Availability

I am available later in the day on Monday but as of the time of writing this, I have not yet received notice of what time I am to demonstrate.