

Book Database

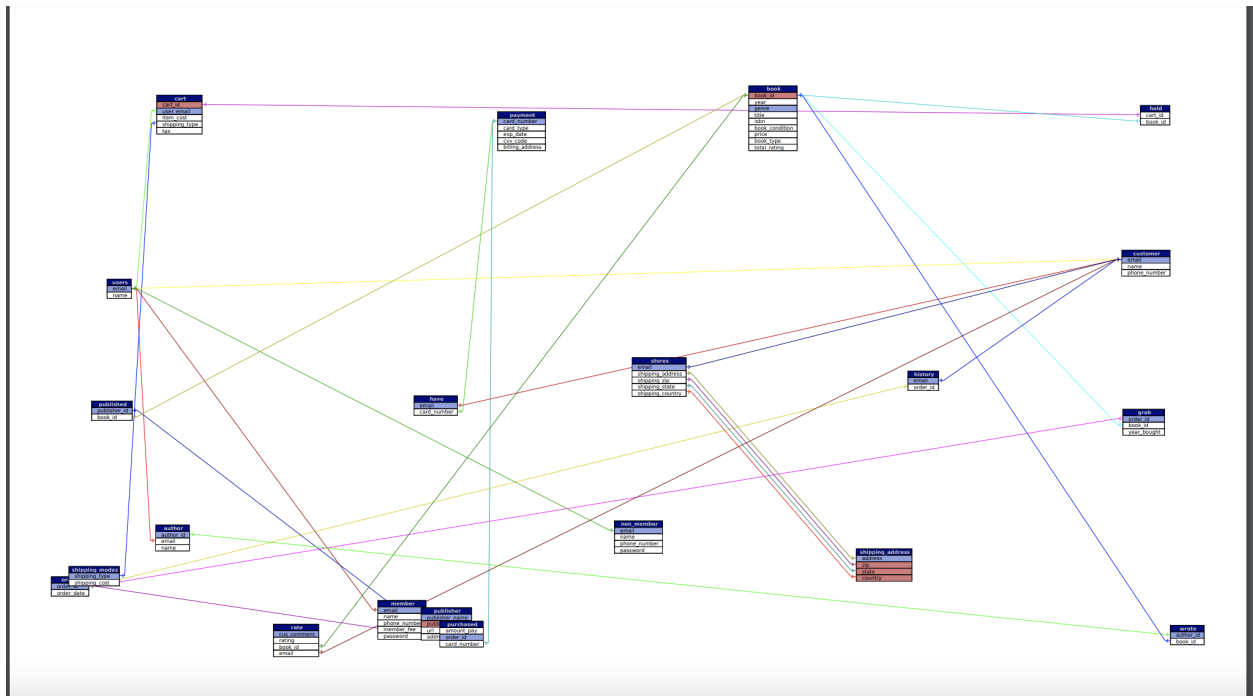
Spring 2022

Group 7: Zack Pelley, Likitha Shrinivas Gurdalwar

Abstract: This book database project included three phase:

- phase 1 is to design a conceptual model of the database and draw an ER diagram that can capture the information needed for this database
- phase 2 is to create and populate the database
- phase 3 (added later)

Phase 1:



ER Diagram

This ER Diagram satisfied the following queries:

1. There is a super user who can modify all contents of the database.
2. Publishers can add new books.
3. Customers can register, and login with username and password, and update their own information. Customers may become members after paying membership fee.
4. Customers may make purchases of books. Customers should be able to check their order information and purchase history.
5. Customers should be able to search by keyword.
6. Different types of books may have different delivery method.
7. Customers may give books they have purchased a rating between 1 and 5 and a comment.

Phase 2: consisting the following files
DB2.sql contain the code to create table:

- users
- author
- customer
- member
- non-member
- publisher
- orders
- payment
- book
- shipping address
- shipping modes
- cart
- grab
- purchased
- rate
- published
- wrote
- history
- have
- stores
- hold

bookstore.html contain the code to

- let users search for best seller
- let users login/signup (if don't have an account yet)
- let users check for all books that is in the database
- let users search for a specific book by name or books that written by a specific authors
- let users add book into their cart

Bookstore.php contain the code to

- After user login from bookstore.html, bookstore.php validates the user if they are in the database
- Bookstore.php also display the book database(bookid, published, genre, title, isbn, condition, price, type, total)

bestseller.php contain the code to

- display the bestselling book using query:

```
select * from book where book_id in
```

```
(select b.book from
```

```
(select max(s.cou), s.book as book, s.years from
```

```
(select count(g.book_id) as cou,g.book_id as book, year(o.order_date) as years from  
grab g, orders o where
```

```
g.order_id = o.order_id group by years,book)
as s group by s.years) as b where b.years = '$year_provided')
```

- display the best selling book by year

```
select * from book where book_id in
(select b.book_id from
(select max(s.cou), s.book_id as book, s.years from
(select count(g.book_id) as cou, g.book_id as book, year(o.order_date) as years from
grab g, orders o where
g.order_id = o.order_id group by years,book)
as s group by s.years) as b)
```

task4_searchbook.php

- using "SELECT * FROM book WHERE title = '\$search_term'" to search for book by the title of the book that user has inserted
- if the user insert the author name then use "SELECT * FROM book WHERE book_id in (SELECT book_id from wrote WHERE author_id in (SELECT author_id from author WHERE name = '\$search_term'))" to look book for book by using the author name

review_of_book.php

- if the user enter a non existing book -> display all of the book that is in the database using queries "select * from book"
- if the user enter a book that is in the database then display the book title and rating using queries "select * from rate where book_id in (select book_id from book where title = '\$book_title')";

login.php

- if user already have an account, they can login using their email and password -> after user login successfully, they will be prompted to the bookstore.php under that user email section
- if an account does not exist -> click on create will take user to create_account.php

catalog.php contain the code to

- Display the book database using query:
select * from book where user can add and purchase the books from there

task1.php

- using queries "INSERT INTO users (email, name) values ('\$email', '\$name')"; in order to add the the user email and name into super class users
- using queries INSERT INTO customer (email, name, phone_number) values ('\$email', '\$name', \$phone_number) to add the user email, name, phone number into customer table - subclass of users
- generate the cart_id (each user have their own cart_id) by using random
- insert the user info in member or non-member of table accordingly

create_account.php

- Form the user to input their name, password, phone number, select membership so that task1.php can handle adding new user in the database

publish.php

- Form the user to input Publisher ID, Author ID, Title, Genre, ISBN, and select the type of the new book so that when the user click on add book, the browser will form task2.php to handle the adding book in the database
- Form the user to input publisher ID, title, new price to update the price of the book. The browser will form update.php to handle updating price of the book in the database

task2.php

- check if the book is already exist, if not then insert the new book in the database

```
"INSERT INTO `book` (`book_id`, `year`, `genre`, `title`,  
`isbn`, `book_condition`, `price`, `book_type`, `total_rating`)  
VALUES ('$bid', '$year', '$bgenre', '$btitle', '$isbn', 'new',  
'$money', '$type', '0.0')";
```
- set default price for ebook = \$14.99, hardcover = 24.99, paperback = \$19.99

update.php

- using query "update book set price = '\$new_price' where book_id in(select book_id from published where book_id in (select book_id from book where title = '\$title') and publisher_id = '\$a_id')"; to update the price of the book that is input in publish.php

superuser.php

- only admin@gmail.com have access to superuser.php
- after login into bookstore.php, admin can click on Secret Super User Button to change the shipping type or cost of the shipping
- after adding new shipping method and price, the browser will form admin to change_ship.php-where the shipping is handle in the database

change_ship.php

- query "insert into shipping_modes(shipping_type, shipping_cost) values('\$type', '\$cost')"; let the admin added a new shipping method and price
- query "update shipping_modes set shipping_cost = '\$cost' where shipping_type = '\$type'"; let the admin update the price of the existing shipping method

history.php

- after click on order history in bookstore.php, browser will form user to history.php where it display the book info and the quantity of books that the user has purchased
- using query `"select count(book_id) as quantity from grab where order_id in (select order_id from history where email = '$email') group by book_id";` to get the book count
- using query `"select * from book where book_id in (select book_id from grab where order_id in (select order_id from history where email = '$email'))";` to get all the book info that the user has purchased

logout.php

- in bookstore.php has a logout section where user can click to logout
- logout will end the current email session and take user back to bookstore.html

rating.php

- after click on rate book order in bookstore.php, the browser will form user to rating.php
- let user rate book that they have purchased by clicking rate this book button. The browser will prompt user to process_review.php

process_review.php

- using `"select * from rate where email = '$email' and book_id = '$book_id'";` to check if the book is already rated by current user
- if the book has not rated then the browser will prompt the user to added_review.php to handle adding rating into database
- if the book is already rated by the user then the browser will reload with the message "You already Rated this book silly"

view_cart.php

- after add book to cart in bookstore.php, view_cart.php display the quantity of the book that is selected
- also let the user modify their order. If the user selects the book they want to remove and click remove, view_cart will reload itself to display the order that has already been modified. That work is handle in remove.php
- if the user done modifying then by clicking confirm purchase, browser will take the user to choose_shipping.php to choose the shipping method so that the confirm_purchase.php can calculate total of the order

choose_shipping.php

- let user choose the method of shipping that the admin has inserted and updated

remove.php

- using query `"delete from hold where book_id = '$book' and cart_id = '$cart' ";` to remove the book that the user has selected in review_cart.php

confirm_purchase.php

- confirm purchase prompt the user input their shipping address and payment information.
- display the book that is in cart by `"select book_id from hold where cart_id = '$cart'"`
- calculating the total including tax and shipping
- after click on purchase, browser will form user to order.php which will handle the final step of purchasing the books

ordered.php

- save the user card info into payment table using query `"insert into payment(card_number, card_type, exp_date, cvv_code, billing_address) values ('$card_number','$card_type','$exp_date','$cvv_code','$card_address')";`
- save the user card info into have relationship using query `"insert into have(email, card_number) values ('$email', '$card_number')"`
- generating a random order_id to save that order in the orders table
- insert the order info into grab relationship `"insert into grab (order_id, book_id, year_bought) values ('$o_id','$book','$date')";`
- remove the order from hold table using `"delete from hold where cart_id = '$cart'";`

account.php

- in bookstore.php, click on account will take the user to account.php where the user can update membership-browser will prompt user to update_member_account.php or update_nonmember_account.php
- user can update phone number-browser will prompt user to update_info.php
- user can update password-browser will prompt update_info.php

update_info.php

- using query `"select * from customer where email = '$email' and phone_number = '$old_num'";` to check if the old phone number enter is in the database system
- using query `"Update customer set phone_number = '$new_phone' where email = '$email' and phone_number = '$old_num'";` to update from old number to new number
- using query `"select * from member where email = '$email' and password = '$old_pass'";` to check if the old password is in the database system

- using "Update non_member set password = '\$new_password' where email = '\$email' and password = '\$old_pass'"; to update the new password

update_member_account.php

- in account.php, if user select change member (non membership, premium membership, cancel membership), their current member status will change accordingly and update_member_account is where this task will be handle
- query "update member set member_fee = '20.00' where email = '\$email'"; will help changing current member status to normal member who will pay 20.00 monthly
- query "update member set member_fee = '50.00' where email = '\$email'"; will help changing current member status to premium member who will pay 50.00 monthly
- query "insert INTO non_member (email,name,phone_number,password) values ('\$email' , '\$name', '\$phone_number', '\$password')"; will help changing current member status to cancel member

update_nonmember_account.php

- update_nonmember_account.php helps insert a nonmember user to member in the database system
- "insert INTO member (email,name,phone_number,member_fee,password) values ('\$email' , '\$name', '\$phone_number', '20.00', '\$password')"; take the information from nonmember to member table