


# Databases for Book Management

Naila Fatima  
nfatima3@gatech.edu  
Group ID: I4

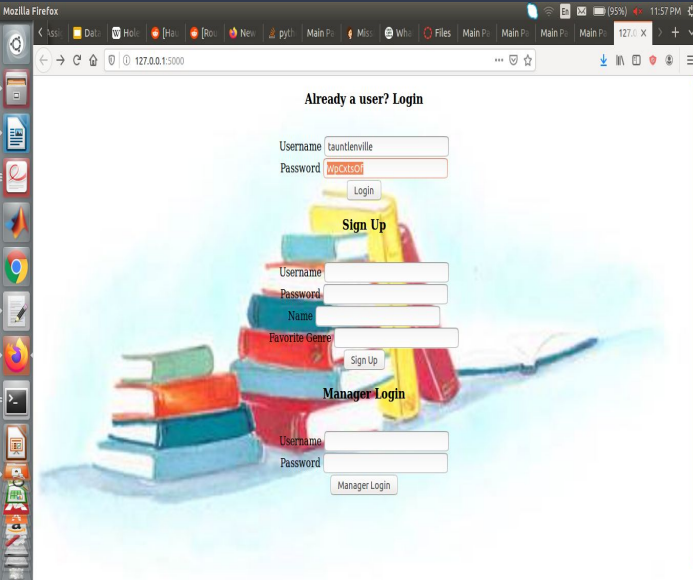
A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the slide.

# Overall Goal

The goal of the project is to create a meaningful application while improving my understanding of databases.

I have created an application which makes it easier for a user to find 'interesting' books. Users are allowed to search for books using different criteria as well as rate them after reading them. If a user finds an interesting book, they can order the book from a bookstore of their choice. .

# Functionality (1)



## Functionality of the System

The application will be able to perform the following functions:

1. Allow users to login: Users will have to use a username and password in order to login to their accounts. The username should be unique for each user.
2. Find books by a particular author: Given an author, all books by that author should be retrieved.
3. Find books of a particular genre: Given a particular genre, all books belonging to that genre should be retrieved.
4. Allow users to rate a book that they have read: Each user will be allowed to rate a book that they have read. They can give a particular book only one rating- either 1,2,3,4 or 5. These ratings will help the application recommend books to users.
5. Find the highest rated book of an author: Find a book written by a particular author which has the highest average rating. This can be considered to be the most-liked book of the author.
6. Recommend a book to the user based on the books that they have read and liked or if they are a new reader, recommend a book of their favorite genre: Each user will be asked to specify a favorite genre (only one). If they are a new reader (if they have not added any books that they have read), they will be recommended a book belonging to their favorite genre. If a user has books which they have read, they will be recommended the highest rated book belonging to the genre which the user has read most.
7. Find a bookstore which carries a particular book: Given a particular book title, retrieve information about the bookstore which carries the book (if in stock) as well as its price.
8. Find the most “critically-acclaimed” book: For all the books in the database, find the book which has the most critical acclaim. The score for each book is calculated as  
$$\text{score} = \text{Number of copies sold} + \text{Number of 5-ratings}$$

# Functionality (2)

The book with the highest score will be assumed to be the most critically acclaimed.

9. Find all books ordered by a particular user.

10. Find books by authors who have won a certain prize.

11. Find the book with the maximum orders.

12. Allow the managers of a bookstore to approve the pending orders made by the employees of the store.

# Users

Users of this application will include frequent readers as well as people who want to develop the habit of reading. It is a well known fact that reading is a beneficial activity- it stimulates the mind while improving concentration. For readers, choosing a book from the thousands of available choices is often a daunting task. It is likely that the availability of a book recommendation system will make it easier for readers to select a book by providing them with a list of books which satisfy specific conditions.

Since my application allows users to order books from the bookstores of their choice, the managers of the bookstores can also use this application to view the orders that their bookstore has received. The application also allows them to place orders to a supplier.

# Databases

Tables\_in\_DB\_PROJECT

Author  
Book\_Selling  
Books  
Books\_Read  
Bookstore  
Inventory  
Manager  
Orders  
User

The relation schemas are shown below:

## User

<u>Username</u>	Password	Name	Favorite Genre
-----------------	----------	------	----------------

## Books

<u>ISBN Code</u>	Title	Genre	Price	Author
------------------	-------	-------	-------	--------

*Note:* ISBN code is a unique identifier for a book.

## Books Read

<u>Username</u>	ISBN Code	Rating	Author
-----------------	-----------	--------	--------

## Author

<u>Name</u>	Hugo Award	Man Booker Prize	Pulitzer Prize
-------------	------------	------------------	----------------

## Bookstore

<u>StoreID</u>	Name	Address
----------------	------	---------

## Orders

<u>OrderID</u>	Username	StoreID	ISBN Code
----------------	----------	---------	-----------

## Book Selling

StoreID	ISBN Code	Number of copies available	Number of copies sold
---------	-----------	----------------------------	-----------------------

*Note:* This table relates the books with the bookstores which sell them.

## Inventory Supply Orders

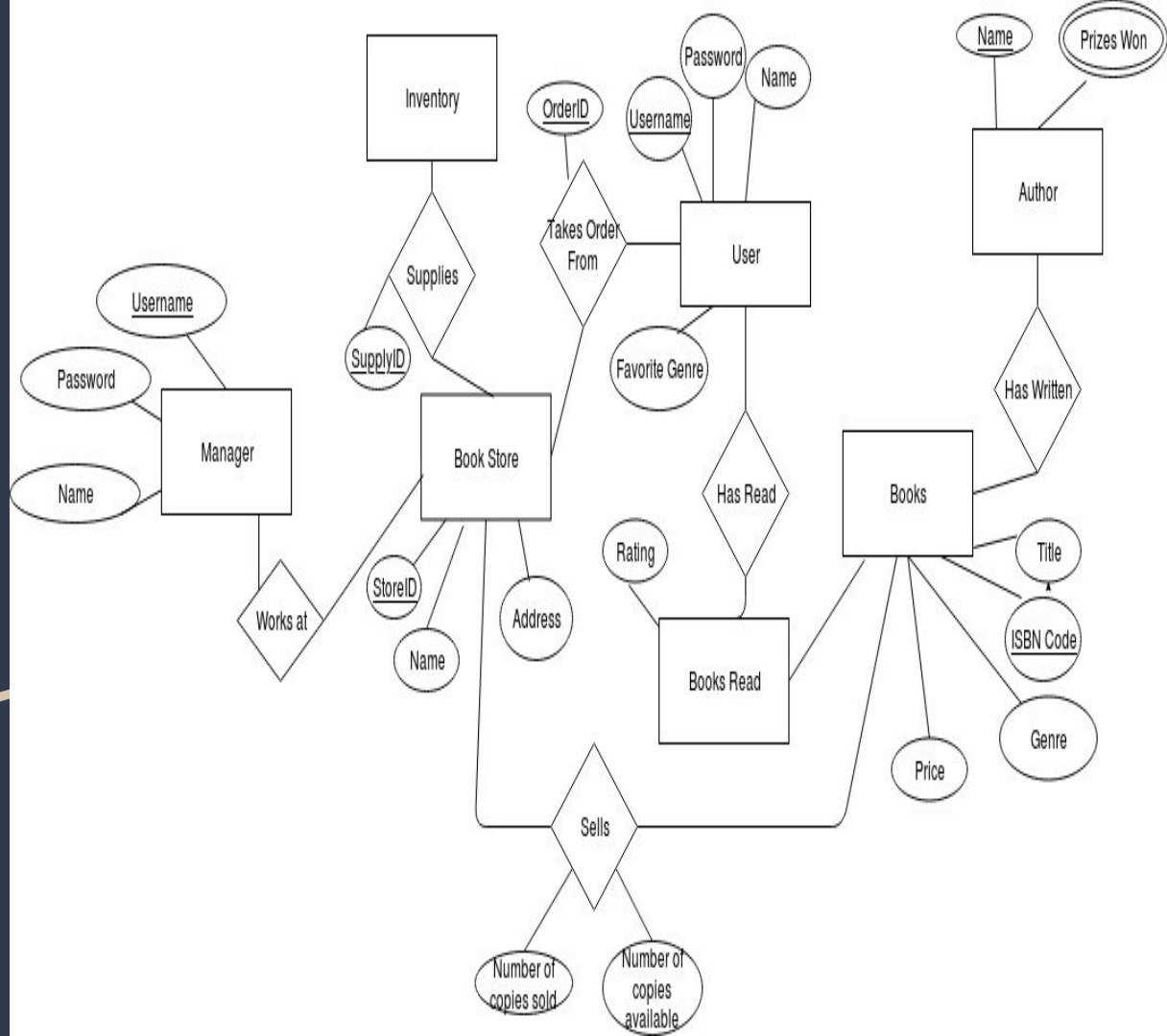
<u>SupplyID</u>	StoreID	ISBN Code
-----------------	---------	-----------

*Note:* We assume that the inventory always has a supply of the books.

# Enhanced Entity Relation (EER) diagram

**Entities:** Inventory, Book Store, Books Read, User, Books, Author, Manager

**Relationships:** Supplies, Sells, Takes Order From, Views, Has Read, Has Written, Works At



# Implementation

## Difficulties faced while creating the application

I was able to implement all the functions that I had mentioned in my interim report. The application was implemented by using the Flask library to create HTML web-pages through Python. The flask-mysqldb library was used to connect the MySQL database to the HTML web-pages. The data was randomly generated by using the Random library of Python and was inserted into the MySQL databases by using the mysql.connector library.

Since my relation schema was minimal (to minimize redundancy), I had to make use of a lot of JOIN conditions. I also faced difficulties when using aggregate functions in the subqueries since there was grouping involved (for example, when we try to find the book with maximum orders). For certain functionalities, I made use of the IN clause and ORDER BY clauses.

Formatting the HTML web-pages was also difficult as I had to try several different styling options (such as CSS) to make the web-pages user friendly.



# What I learnt by doing this project

## What additional features I could have added

- Learnt how to map an EER model to a relational model while minimizing redundancy.
  - Learnt how to formulate queries using JOINS and aggregate functions.
  - Learnt how to connect HTML web-pages to a database in order to show the retrieved results.
- 
- Could have expanded the list of awards considered when finding books written by award-winning authors. I have just considered 3.
  - Could have added a 'Delete Account' feature.
  - Could have added an 'Employee' entity for the bookstore. The employees would order the books while the manager would approve the pending orders.