DBMS – Project

Book Wise: A Book Shop Management System

University of Houston - Downtown

Submitted by:

- Hardi Raval

Table of Content

Index

Introduction:

This project aims to address the challenges faced by bookshops in managing diverse sets of data related to books, authors, sales, ratings, and more. By implementing a robust database management system, we strive to streamline data storage, retrieval, and analysis processes, ultimately enhancing decision-making and operational efficiency within the bookshop.

Abstract:

This project revolves around the development and implementation of a database management system tailored for a bookshop environment. The system aims to centralize and organize various datasets pertaining to books, authors, sales, ratings, and other relevant information. By leveraging the capabilities of a relational DBMS, we aim to create a cohesive platform that facilitates efficient data management, querying, and reporting. Through this project, we seek to empower bookshop owners and managers with the tools necessary to make informed decisions and optimize business operations.

Mission Statement:

Our mission is to design, develop, and deploy a comprehensive database management system that addresses the data management challenges faced by bookshops. By establishing a centralized repository for diverse datasets, our system aims to enhance data accessibility, integrity, and usability.

Mission Objectives:

1. Ensure accurate and up-to-date information on book titles, authors, editions, publishers, and other relevant details.
    1. To maintain (enter, update, delete) data on books.
    2. To maintain (enter, update, delete) data on authors.
    3. To maintain (enter, update, delete) data on book editions.
    4. To maintain (enter, update, delete) data on publishers.
    5. To maintain (enter, update, delete) data on Rating.
    6. To maintain (enter, update, delete) data on Checkouts.
    7. To maintain (enter, update, delete) data on award.
    8. To maintain (enter, update, delete) data on Sales.
    9. To maintain (enter, update, delete) data on Reviewers.

2. Develop search functionalities to retrieve specific books based on various criteria, such as title, author, genre, and publication date.
    1. To perform searches on books.
    2. To perform searches on authors.
    3. To perform searches on book editions.
    4. To perform searches on publishers.
    5. To perform searches on Rating.
    6. To perform searches on Checkouts.
    7. To perform searches on award.
    8. To perform searches on Sales.
    9. To perform searches on Reviewers.
3. Generate reports summarizing book details, such as total number of books, bestselling titles, and author distributions.
    1. To report on books.
    2. To report on authors.
    3. To report on publishers.
    4. To report on award.
    5. To report on Sales and so on.
4. Implement efficient data retrieval mechanisms, including SQL queries and indexing, to facilitate quick and accurate access to information.
5. Design the system with scalability and performance in mind to accommodate future growth and handle increasing volumes of data without compromising efficiency.

Major Users:

Data	Access Type	Book Store Manager	Book Store Supervisor	Book Store Assistant	Cashier
Books Details	Maintain	X		X	
	Query	X	X		
	Report	X	X	X	X
Author Details	Maintain	X		X	
	Query	X	X		
	Report	X	X		X
Book Edition Details	Maintain	X		X	
	Query	X	X		
	Report	X	X	X	X
Publisher Details	Maintain	X		X	
	Query	X	X		
	Report	X	X	X	X
Rating Details	Maintain	X		X	
	Query	X	X		
	Report	X	X	X	X

Checkout Details	Maintain	X		X	X
	Query	X	X		
	Report	X	X		X
Awards Detail	Maintain	X		X	
	Query	X	X		
	Report	X	X		X
Sales Details	Maintain	X			X
	Query	X			
	Report	X	X		
Reviewers Details	Maintain	X		X	
	Query	X	X		
	Report	X	X	X	X

Book Store Manager:

- **Maintain**: The manager is capable to maintaining all aspects of the bookshop database, including books, authors, editions, publishers, ratings, checkouts, awards, sales, and Reviewers. This involves adding, updating, and deleting records as necessary.
- **Query**: The manager can query the database to retrieve information about all the details like books, authors, editions, etc., for decision-making and analysis.
- **Report**: The manager generates comprehensive reports on various aspects of the bookshop, such as sales performance, inventory management, and customer trends, to monitor the overall business operations.

Book Store Supervisor:

- **Maintain**: Manager cannot do any maintaining of the data; they can make query and report to the data added by the assistant.
- **Report**: They generate reports to provide insights into sales, inventory, and other relevant metrics, assisting in decision-making and performance evaluation.

Book Store Assistant:

- **Maintain**: Assistants help in maintaining data integrity by entering and updating records related to books, authors, editions, publishers, awards, and Reviewers info as directed by the manager or supervisor.
- **Query**: They can query the database to retrieve information for assisting customers, handling inquiries, and supporting other operational tasks.
- **Report**: Assistants may generate basic reports on sales, inventory status, and customer interactions to provide relevant information for decision-making and customer service.

Cashier:

- **Maintain**: Cashiers maintain checkout details, such as processing transactions, updating payment records, and managing checkout-related information in the database.

- **Query**: They can query the database to retrieve transaction details, customer information, and other data necessary for processing sales transactions.
- **Report**: Cashiers may generate reports on daily sales, transaction summaries, and cashier performance to track sales trends and monitor cash flow.

Normalization Explanation:

1. **Book Table**

- In Book table Genre attribute having the multiple values in single columns like: Sci-Fi/ Fantasy, Mystery & thriller.
- This violates the principles of database normalization, specifically the first normal form (1NF).
- In order to resolve this issue and adhere to database normalization principles, you can create a separate column to store the genres associated with each book.
- Book Table is like below as an example:

BookID	Title	AuthID	Genre	SeriesID	Volume Number	Staff Comment
1	The Hobbit	105	Sci-Fi/Fantasy	002A	1	Prelude
2	The Da Vinci Code	106	Mystery & Thriller	NULL	NULL	Gripping
3	Jurassic Park	107	Sci-Fi	003A	1	Adventure
4	The Hobbit	105	Sci-Fi/Fantasy	002A	1	Prelude

- Book Table is like below after applying the 1NF:

BookID	Title	AuthID	Genre1	Genre2	SeriesID	Volume Number	Staff Comment
1	The Hobbit	105	**Sci-Fi**	**Fantasy**	002A	1	Prelude
2	The Da Vinci Code	106	**Mystery**	**Thriller**	NULL	NULL	Gripping
3	Jurassic Park	107	**Sci-Fi**	**NULL**	003A	1	Adventure
4	The Hobbit	105	**Sci-Fi**	**Fantasy**	002A	1	Prelude

- By allowing null values in the "Genre2" column, books with only one genre can leave the "Genre2" column empty, while books with multiple genres can populate both "Genre1" and "Genre2" columns accordingly.
- **Primary Key**: Book ID
- **Foreign Key**: AuthID (references AuthID in Author table)
- **Attributes dependency**:
  - Title depends on BookID.
  - AuthID depends on BookID.
  - **Genre1 depends on BookID.**
  - **Genre2 depends on BookID.**

  - SeriesID depends on BookID.
  - Volume Number depends on BookID.
  - Staff Comment depends on BookID.
- **1NF**: Each cell has only one data value from its attribute domain.
- **2NF**: All non-prime attributes (Title, AuthID, Genre1, Genre2, SeriesID, Volume Number, Staff Comment) are fully functionally dependent on the primary key (BookID).
- **3NF**: There are no transitive dependencies. All non-prime attributes depend directly on the primary key (BookID).
- **BCNF**: Each subset of the primary key (BookID) determines all attributes. There are no non-trivial functional dependencies of attributes on the primary key.

2. Author Table

- **Primary Key**: AuthID
- **Foreign Key**: N/A
- **Attributes dependency**:
  - First Name depends on AuthID.
  - Last Name depends on AuthID.
  - Birthday depends on AuthID.
  - Country of Residence depends on AuthID.
  - Hrs Writing per Day depends on AuthID.
- **1NF**: Each cell has only one data value from its attribute domain.
- **2NF**: All non-prime attributes (First Name, Last Name, Birthday, Country of Residence, Hrs Writing per Day) are fully functionally dependent on the primary key (AuthID).
- **3NF**: There are no transitive dependencies. All non-prime attributes depend directly on the primary key (AuthID).
- **BCNF**: Each subset of the primary key (AuthID) determines all attributes. There are no non-trivial functional dependencies of attributes on the primary key.

3. Award Table

- **Primary Key**: Award ID
- **Foreign Key**: BookID (Reference BookID in Book Table)
- **Attributes dependency**:
  - Title depends on AwardID.
  - Award Name depends on AwardID.
  - Year Won depends on AwardID.
  - Book ID depends on AwardID.
- **1NF**: Each cell has only one data value from its attribute domain.
- **2NF**: All non-prime attributes (Title, Award Name, Year Won, Book ID) are fully functionally dependent on the primary key (AwardID).

- **3NF**: There are no transitive dependencies. All non-prime attributes depend directly on the primary key (AwardID).
- **BCNF**: Each subset of the primary key (AwardID) determines all attributes. There are no non-trivial functional dependencies of attributes on the primary key.

4. Checkout Table

- **Primary Key**: CheckoutID
- **Foreign Key**: BookID (Reference BookID in Book Table)
- **Attributes dependency**:
  - BookID depends on CheckoutID.
  - CheckoutMonth depends on CheckoutID.
  - Number of Checkouts depends on CheckoutID.
  - Title depends on AwardID.
- **1NF**: Each cell has only one data value from its attribute domain.
- **2NF**: All non-prime attributes (BookID, CheckoutMonth, Number of Checkouts) are fully functionally dependent on the primary key (CheckoutID).
- **3NF**: There are no transitive dependencies. All non-prime attributes depend directly on the primary key (CheckoutID).
- **BCNF** Each subset of the primary key (CheckoutID) determines all attributes. There are no non-trivial functional dependencies of attributes on the primary key.

5. Book Edition Table

- **Primary Key**: ISBN
- **Foreign Key**
  - BookID (Reference BookID in Book Table)
  - PubID (references PubID in Publisher table)
- **Attributes dependency**:
  - BookID depends on ISBN.
  - Format depends on ISBN.
  - PubID depends on ISBN.
  - Publication Date depends on ISBN.
  - Pages depends on ISBN.
  - Print Run Size (k) depends on ISBN.
  - Price depends on ISBN.
- **1NF**: Each cell has only one data value from its attribute domain.
- **2NF**: All non-prime attributes (BookID, Format, PubID, Publication Date, Pages, Print Run Size (k), Price) are fully functionally dependent on the primary key (ISBN).
- **3NF**: There are no transitive dependencies. All non-prime attributes depend directly on the primary key (ISBN).
- **BCNF** Each subset of the primary key (ISBN) determines all attributes. There are no non-trivial functional dependencies of attributes on the primary key.

6. Publisher Table
   - **Primary Key**: PubID
   - **Foreign Key:** N/A
   - **Attributes dependency**:
     - Publishing House depends on PubID.
     - City depends on PubID.
     - State depends on PubID.
     - Country depend on PubID.
     - Year Established depends on PubID.
     - Marketing Spend depends on PubID.
   - **1NF**: Each cell has only one data value from its attribute domain.
   - **2NF** All non-prime attributes (Publishing House, City, State, Country, Year Established, Marketing Spend) are fully functionally dependent on the primary key (PubID).
   - **3NF**: There are no transitive dependencies. All non-prime attributes depend directly on the primary key (PubID).
   - **BCNF** Each subset of the primary key (PubID) determines all attributes. There are no non-trivial functional dependencies of attributes on the primary key.

7. Rating Table
   - **Primary Key**: RatingID
   - **Foreign Key:**
     - BookID (references BookID in Book table)
     - ReviewerID (references ReviewerID in Reviewer table)
   - **Attributes dependency**:
     - BookID depends on RatingID.
     - Rating depends on RatingID.
     - ReviewerID depends on RatingID.
   - **1NF**: Each cell has only one data value from its attribute domain.
   - **2NF** All non-prime attributes (BookID, Rating, ReviewerID) are fully functionally dependent on the primary key (RatingID).
   - **3NF**: There are no transitive dependencies. All non-prime attributes depend directly on the primary key (RatingID).
   - **BCNF** Each subset of the primary key (RatingID) determines all attributes. There are no non-trivial functional dependencies of attributes on the primary key.

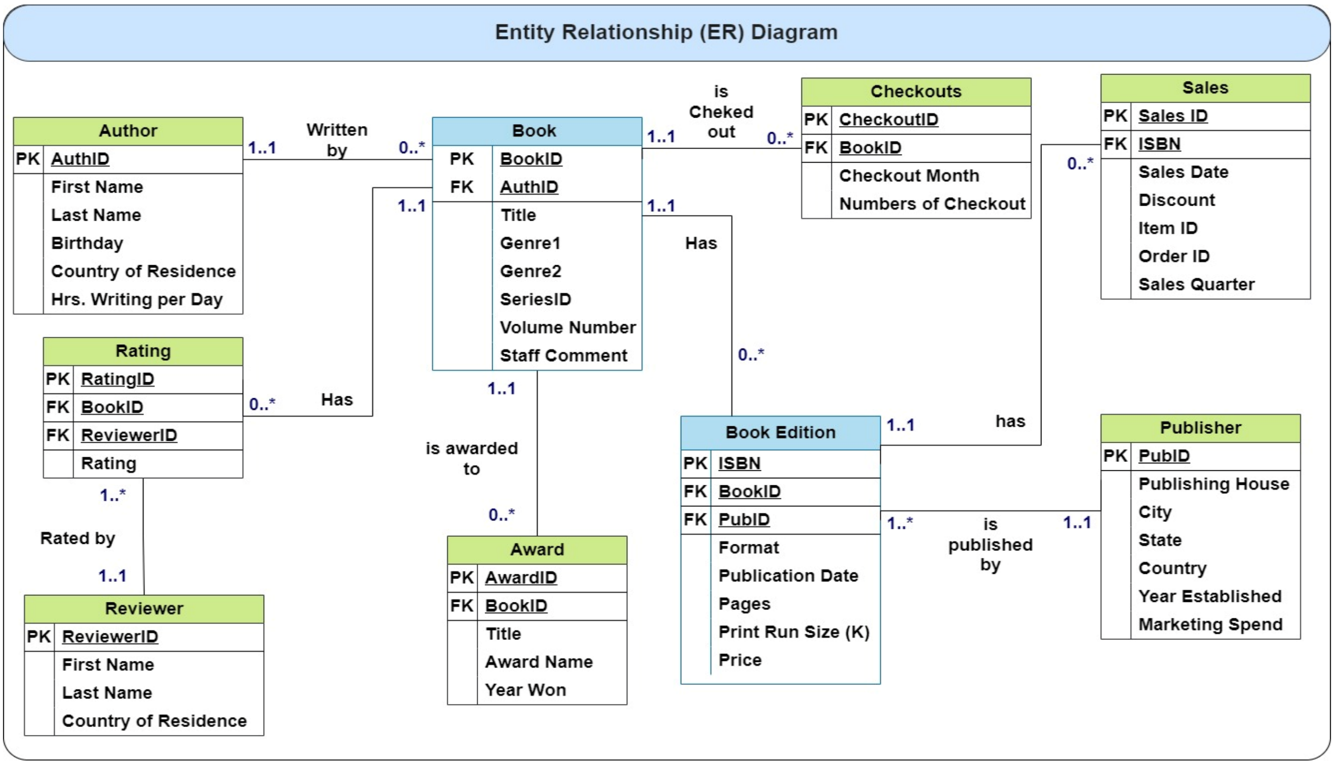
8. Reviewer Table

- **Primary Key**: ReviewerID
- **Foreign Key:** N/A
- **Attributes dependency**:
    - First Name depends on ReviewerID.
    - Last Name depends on ReviewerID.
    - Country of Residence depends on ReviewerID.
- **1NF**: Each cell has only one data value from its attribute domain.
- **2NF** All non-prime attributes (First Name, Last Name, Country of Residence) are fully functionally dependent on the primary key (ReviewerID).
- **3NF**: There are no transitive dependencies. All non-prime attributes depend directly on the primary key (ReviewerID).
- **BCNF** Each subset of the primary key (ReviewerID) determines all attributes. There are no non-trivial functional dependencies of attributes on the primary key.

9. Sales Table

- **Primary Key**: SalesID
- **Foreign Key:** N/A
- **Attributes dependency**:
    - Sale Date depends on SalesID.
    - ISBN depends on SalesID.
    - Discount depends on SalesID.
    - ItemID depends on SalesID.
    - OrderID depends on SalesID.
    - Quarter depends on SalesID.
- **1NF**: Each cell has only one data value from its attribute domain.
- **2NF:** All non-prime attributes (Sale Date, ISBN, Discount, ItemID, OrderID, Quarter) are fully functionally dependent on the primary key (SalesID).
- **3NF**: There are no transitive dependencies. All non-prime attributes depend directly on the primary key (SalesID).
- **BCNF** Each subset of the primary key (SalesID) determines all attributes. There are no non-trivial functional dependencies of attributes on the primary key.

ER Diagram:



- Above ER Diagram is reviewed, the comments provided on the phase 2 submission and carefully considered the suggestions for improvement.
- With the inclusion of the new table, the ER diagram now encompasses a total of 9 tables, meeting the minimum requirement set forth for the project.

Relationship between tables:

- **Author & Book**: An author can write zero or one book (1..1) and a book can be written by one and only one author (1..1).
- **Book & Checkout**: A book can be checked out zero or many times (0..M) and a checkout can be for one and only one book (1..1).
- **Book & Rating**: A book can have zero or many ratings (0..M) and a rating can be for one and only one book (1..1).
- **Book & Award**: A book can win zero or many awards (0..M) and an award can be given to one or many books (M..N).
- **Book & Publisher**: A book is published by one and only one publisher (1..1) and a publisher can publish many books (1..M).
- **Book edition & sales**: A Book Edition can have zero or many (0..M) Sales records, but each Sale record belongs to only one (1..1) Book Edition.
- **Book edition & Publisher**: One publisher can publish one or many (1..M) book editions. Each book edition can have only one (1..1) Publisher.
- **Rating & Reviewer**: Each reviewer can review one or many (1..M) books, and each book can have only one rating from a reviewer (1…1).

Actors:

1. **Manager:**

   Role Description: Oversees all operations within the bookstore. Managers are responsible for strategic planning, managing staff, and making key business decisions. They access comprehensive data across all sections of the database to monitor performance and implement improvements.

2. **Supervisor:**

   Role Description: Acts as an intermediary between the managerial staff and the operational staff, such as cashiers and assistants. Supervisors manage day-to-day activities, ensure compliance with store policies, and handle minor administrative tasks. They frequently interact with sections of the database related to employee performance and operational efficiency.

3. **Cashier:**

   Role Description: Handles all customer transactions, including sales and returns. Cashiers need access to sales and checkout sections of the database to process purchases, apply discounts, and manage cash flows. They play a critical role in customer service and the accuracy of financial transactions.

4. **Assistant:**

   Role Description: Supports various store functions, from stocking shelves to assisting customers and helping with inventory management. Assistants use the database primarily for inventory checks, updating stock levels, and sometimes for basic customer queries about book availability and specifications.

Use Cases:

1. Books Details Use Cases:

1. Insert Book Operation:
   - Actor: Book Store User
   - Data: Book ID, Title, Author ID, Genre1, Genre2, SeriesID, Volume Number, Staff Comment
     - Steps:
       - (Assistant/ Manager) User navigates to the "Books" section.
       - (Assistant/ Manager) User selects "Add New Book."
       - (Assistant/ Manager) User fills in book details.
       - (Assistant/ Manager) User submits to add the new book.
   - Insert Operation: Add a new book.

     INSERT INTO Book (BookID, Title, AuthID, Genre1, Genre2, SeriesID, VolumeNumber, StaffComment) VALUES (1, 'New Book Title', 'XX999', 'Fantasy', NULL, 'BS-CHP', 1, 'This is a new book.');

2. Delete Book Operation:
   - Actor: Book Store User
   - Data: Book ID
     - Steps:
       - User navigates to the "Books" section.
       - User selects the book to delete.
       - User confirms deletion.
   - Delete Operation: Remove a book.

     DELETE FROM Book WHERE BookID = 3;

3. Update Book Operation:
   - Actor: Book Store User
   - Data Book ID, Title, Author ID, Genre1, Genre2, SeriesID, Volume Number, Staff Comment
     - Steps:
       - User navigates to the "Books" section.
       - User selects the book to update.
       - User modifies book details.
       - User saves changes.
   - Update Operation: Associate a book with a different author.

     UPDATE Book SET Genre1 = 'Science’ WHERE BookID = 'AY135';

4. Aggregate Book Operation
   - Actor: Book Store User
   - Data Needed: Book ID
   - Purpose: To determine the average rating of a specific book to gauge its reception and quality based on user reviews.
     - Steps:
       - User navigates to the "Ratings" section.
       - User selects "View Average Rating."
       - User enters or selects a Book ID for which the average rating is needed.
       - User submits the query to process.
       - System calculates and displays the average rating for the specified book.
   - Aggregate Query**:** Find the average rating for a book.
     SELECT Genre1, COUNT(*) AS TotalBooks FROM Book GROUP BY Genre1;

2. Author Details Use Cases:

5. Insert Author Operation:
   - Actor: Book Store User
   - Data: First Name, Last Name, Birthday, Country of Residence, Hours Writing per Day
     - Steps:
       - User navigates to "Authors" section.
       - User selects "Add New Author."
       - User enters author details.
       - User submits to add the new author.
   - **INSERT INTO Author (AuthID, FirstName, LastName, Birthday, CountryOfResidence, HrsWritingPerDay) VALUES (1, 'Jane', 'Doe', '1975-05-20', 'USA', 2.0);**

6. Delete Author Operation:
   - Actor: Book Store User
   - Data: Author ID
     - Steps:
       - User navigates to "Authors" section.
       - User selects the author to delete.
       - User confirms deletion.
   - **DELETE FROM Author WHERE AuthID = 4;**

7. Update Author Operation:
   - Actor: Book Store User
   - Data: Author ID, First Name, Last Name, Birthday, Country of Residence, Hours Writing per Day
     - Steps:
       - User navigates to "Authors" section.
       - User selects the author to update.
       - User modifies author details.
       - User saves changes.
   - **UPDATE Author SET HrsWritingPerDay = 4.0 WHERE AuthID = 2;.**

8. Aggregate Author Operation
   - Actor: Book Store User
   - Purpose: Determine the total number of authors by each country.
     - Steps:
       - User accesses the "Authors" section.
       - User selects the "View Count of Author by Country" option.
   - System displays a list of authors with their respective book counts by each country.
   - **SELECT CountryOfResidence, COUNT(*) AS NumAuthors
FROM Author
GROUP BY CountryOfResidence;**

3. Books Edition Details Use Cases:

9. Insert Book Edition Operation:
   - Actor: Book Store User
   - Data: ISBN, Book ID, Format, Publication ID, Publication Date, Pages, Print Run Size (k), Price
     - Steps:
       - User navigates to "Book Editions" section.
       - User selects "Add New Edition."
       - User fills in edition details.
       - User submits to add the new edition.
   - **INSERT INTO BookEdition VALUES ('ISBN-106', 105, 203, 'Ebook', '2023-01-10', NULL, NULL, 15.99);**

10. Delete Book Edition Operation:
    - Actor: Book Store User
    - Data: ISBN
      - Steps:
        - User navigates to "Book Editions" section.
        - User selects the edition to delete.
        - User confirms deletion.

- **DELETE FROM BookEdition**
  WHERE ISBN = 'ISBN-106';

11. Update Book Edition Operation:
    - Actor: Book Store User
    - Data: ISBN, Book ID, Format, Publication ID, Publication Date, Pages, Print Run Size (k), Price
        - Steps:
            - User navigates to "Book Editions" section.
            - User selects the edition to update.
            - User modifies edition details.
            - User saves changes.
    - **UPDATE BookEdition**
      SET Price = 17.99 WHERE ISBN = 'ISBN-105';

12. Aggregate Query Operation: Calculate Average Price of Book Editions
    - Actor: Book Store User
    - Purpose: To determine the average price of all book editions, which helps in pricing strategies and market analysis.
        - Steps:
            - User navigates to the "Book Editions" section.
            - User selects "View Average Price."
    - System calculates and displays the average price of all book editions.
    - **SELECT AVG(Price) FROM BookEdition;**

4. Publisher Details Use Cases:

13. Insert Publisher Operation:
    - Actor: Book Store User
    - Data: Publisher Name, City, State, Country, Year Established, Marketing Spend
        - Steps:
            - User navigates to "Publishers" section.
            - User selects "Add New Publisher."
            - User enters publisher details.
            - User submits to add the new publisher.
    - **INSERT INTO Publisher VALUES (206, 'Scholastic', 'New York', 'NY', 'USA', 1920, 300000);**

14. Delete Publisher Operation:
    - Actor: Book Store User
    - Data: Publisher ID
        - Steps:
            - User navigates to "Publishers" section.
            - User selects the publisher to delete.
            - User confirms deletion.
    - **DELETE FROM Publisher WHERE PubID = 206;**

15. Update Publisher Operation:
    - Actor: Book Store User
    - Data: Publisher ID, Publisher Name, City, State, Country, Year Established, Marketing Spend
        - Steps:
            - User navigates to "Publishers" section.
            - User selects the publisher to update.
            - User modifies publisher details.
            - User saves changes.
    - **UPDATE Publisher SET MarketingSpend = 350000 WHERE PubID = 2001;**

16. Aggregate Query Operation: Count Publishers by Country
    - Actor: Book Store User
    - Purpose: To determine the number of publishers based in a specific country, aiding in regional market analysis and operational planning.
        - Steps:
            - User navigates to the "Publishers" section.
            - User selects "Count Publishers by Country."
            - User specifies the country of interest, such as "USA."
            - System calculates and displays the number of publishers located in the specified country.
    - **SELECT COUNT(*) FROM Publisher WHERE Country = 'USA';**

5. Rating Details Use Cases:

17. Insert Rating Operation:
    - Actor: Book Store User
    - Data: Book ID, Reviewer ID, Rating, Rating ID
        - Steps:
            - User navigates to “Ratings” section.
            - User selects “Add New Rating.”

        - User enters rating details.
        - User submits to add the new rating.
- **INSERT INTO Rating VALUES (6, 102, 4.0, 6);**

18. Delete Rating Operation:

- Actor: Book Store User
- Data: Rating ID
    - Steps:
        - User navigates to "Ratings" section.
        - User selects the rating to delete.
        - User confirms deletion.
- **DELETE FROM Rating WHERE RatingID = 6;**

19. Update Rating Operation:

- Actor: Book Store User
- Data: Rating ID, Book ID, Rating, Reviewer ID, Review ID
    - Steps:
        - User navigates to "Ratings" section.
        - User selects the rating to update.
        - User modifies rating details.
        - User saves changes.
- **UPDATE Rating SET Rating = 4.5 WHERE RatingID = 5;**

20. Aggregate Query Operation: Determine the number of ratings for each book

- Actor: Book Store User
- Purpose: To calculate the number of ratings for each book
    - Steps:
        - User navigates to the "Ratings" section.
        - User selects "View Ratings for each book."
- System computes and displays the average rating for each book in the database.
- **SELECT BookID, COUNT(*) AS NumRatings**
FROM Rating
GROUP BY BookID;

6. <u>Check-Out Details Use Cases:</u>

21. Insert Checkout Operation:

- Actor: Book Store User
- Data: Book ID, Checkout Month, Number of Checkouts
    - Steps:
        - User navigates to "Checkouts" section.

        - User selects "Add New Checkout."
        - User enters checkout details.
        - User submits to add the new checkout.
- **INSERT INTO Checkouts VALUES (6, 104, '2023-01-01', 1);**

22. Delete Checkout Operation:

- Actor: Book Store User
- Data: Checkout ID
    - Steps:
        - User navigates to "Checkouts" section.
        - User selects the checkout to delete.
        - User confirms deletion.
- **DELETE FROM Checkouts WHERE CheckoutID = 6;**

23. Update Checkout Operation:

- Actor: Book Store User
- Data: Checkout ID, Book ID, Checkout Month, Number of Checkouts
    - Steps:
        - User navigates to "Checkouts" section.
        - User selects the checkout to update.
        - User modifies checkout details.
        - User saves changes.
- **UPDATE Checkouts SET NoOfCheckouts = 2 WHERE CheckoutID = 5;**

24. Aggregate Query Operation: Average number of checkouts per book

- Actor: Book Store User
- Purpose: To calculate the Average number of checkouts per book.
    - Steps:
        - User navigates to the "Checkouts" section.
        - User selects "View Avg Checkouts for a Book."
- System calculates and displays the total number of checkouts for the specified book.
- **SELECT AVG(NoOfCheckouts) AS AvgCheckoutsPerBook FROM Checkouts;**

7. Awards Details Use Cases:

25. Insert Award Operation:

- Actor: Book Store User
- Data: Title, Award Name, Year Won
    - Steps:
        - User navigates to "Awards" section.
        - User selects "Add New Award."
        - User enters award details.

        - User submits to add the new award.
- **INSERT INTO Award VALUES (6, 102, 'Best Novel', 'National Book Award', 2022);**

26. Delete Award Operation:
    - Actor: Book Store User
    - Data: Award ID
        - Steps:
            - User navigates to "Awards" section.
            - User selects the award to delete.
            - User confirms deletion.
    - **DELETE FROM Award WHERE AwardID = 6;**

27. Update Award Operation:
    - Actor: Book Store User
    - Data: Award ID, Title, Award Name, Year Won
        - Steps:
            - User navigates to "Awards" section.
            - User selects the award to update.
            - User modifies award details.
            - User saves changes.
    - **UPDATE Award SET YearWon = 2021 WHERE AwardID = 5;**

28. Aggregate Query Operation: Count Awards by Year
    - Actor: Book Store User
    - Purpose: To determine the number of awards given out each year, useful for analyzing trends and the success of various titles over time.
        - Steps:
            - User navigates to the "Awards" section.
            - User selects "View Awards by Year."
    - System computes and displays the total number of awards distributed each year, providing insights into the frequency and distribution of accolades.
    - **SELECT YearWon, COUNT(*) AS TotalAwards**
FROM Award
GROUP BY YearWon
ORDER BY YearWon;

8. Sales Details Use Cases:

29. Insert Sales Operation:

- Actor: Book Store User
- Data: Sale Date, ISBN, Discount, Item ID, Order ID
    - Steps:
        - User navigates to "Sales" section.
        - User selects "Add New Sale."
        - User enters sale details.
        - User submits to add the new sale.
- **INSERT INTO Sales VALUES (6, 'ISBN-103', '2023-01-15', 10, 2, 6001, 1);**

30. Delete Sales Operation:

- Actor: Book Store User
- Data: Sale ID
    - Steps:
        - User navigates to "Sales" section.
        - User selects the sale to delete.
        - User confirms deletion.
- **DELETE FROM Sales WHERE SalesID = 6;**

31. Update Sales Operation:

- Actor: Book Store User
- Data: Sale ID, Sale Date, ISBN, Discount, Item ID, Order ID
    - Steps:
        - User navigates to "Sales" section.
        - User selects the sale to update.
        - User modifies sale details.
        - User saves changes.
- **UPDATE Sales SET Discount = 15 WHERE SalesID = 5;**

32. Aggregate operation: Calculate the total sales amount for each quarter.

- Actor: Book Store User
- Purpose: To determine the Calculate the total sales amount for each quarter.
    - Steps:
        - User navigates to the "Sales" section.
        - User selects "Total Books Sold each quarter."
- System calculates and displays the cumulative number of books sold on each quarter.
- **SELECT SalesQuarter, SUM(ItemID) AS TotalSalesAmount**
  FROM Salesf
  GROUP BY SalesQuarter;

9. Reviewer Details Use Cases:

33. Insert Reviewer Info Operation:

- Actor: Book Store User
- Data: Reviewer ID, First Name, Last Name, Country of Residence
  - Steps:
    - User navigates to "Reviewer" section.
    - User selects "Add New Reviewer".
    - User enters Reviewer details.
    - User submits to add the new Reviewer info.

-
  ```
  INSERT INTO Reviewer (ReviewerID, FirstName, LastName, CountryOfResidence)
  VALUES (102, 'John', 'Smith', 'United States');
  ```

34. Delete Reviewer Info Operation:

- Actor: Book Store User
- Data: Reviewer ID
  - Steps:
    - User navigates to " Reviewer" section.
    - User selects the Reviewer to delete.
    - User confirms deletion.

-
  ```
  DELETE FROM Reviewer WHERE ReviewerID = 110;
  ```

35. Update Reviewer Info Operation:

- Actor: Book Store User
- Data: Reviewer ID, First Name, Last Name, Country of Residence
  - Steps:
    - User navigates to " Reviewer" section.
    - User selects the Reviewer to update.
    - User modifies Reviewer details.
    - User saves changes.

-
  ```
  UPDATE Reviewer
  SET CountryOfResidence = 'Italy'
  WHERE ReviewerID = 105;
  ```

36. Aggregate Query Operation: Count of Reviewers by Country

- Actor: Book Store User
- Purpose: To tally the number of books per genre.
  - Steps:
    - User navigates to " Reviewer" section.
    - User selects " Count of Reviewers by country".
    - System displays total books per genre.

-
  ```
  SELECT CountryOfResidence, COUNT(*) AS TotalReviewers
  FROM Reviewer GROUP BY CountryOfResidence;
  ```

10.Join Query (Relationship) Details Use Cases:

37. Author and Book

- Use Case: List All Authors with Their Books
- Actor: Book Store User
- Purpose: To display a list of all authors along with the books they have authored by book IDs.
    - Steps:
        - User navigates to the "Authors" section.
        - User selects "View Authors and Their Books."
- System displays a list of authors alongside their book IDs, facilitating tracking and identification without revealing book titles.

- Joint Query between Author and Book to list authors and their respective book IDs
-
  ```
  SELECT Author.AuthID, Author.FirstName, Author.LastName, Book.BookID
  FROM Author
  JOIN Book ON Author.AuthID = Book.AuthID;
  ```

38. Book and Rating

- Use Case: List All Books with Their Average Ratings
- Actor: Book Store User
- Purpose: To display a list of all books along with their average ratings using only book IDs to maintain confidentiality or simplicity in reporting.
    - Steps:
        - User navigates to the "Books" section.
        - User selects "View Books with Average Ratings."
- System calculates and displays each book's ID along with its average rating, facilitating assessment of popularity and reception without revealing book titles.

- Joint Query between Book and Rating to find the average rating for each book by ID
-
  ```
  SELECT Book.BookID, AVG(Rating.Rating) AS AverageRating
  FROM Book
  JOIN Rating ON Book.BookID = Rating.BookID
  GROUP BY Book.BookID;
  ```

39. Book and Award

- Use Case: List All Award-Winning Books with Award Details
- Actor: Book Store User
- Purpose: To display a list of all award-winning books using book IDs, along with details of the awards they have won, to understand their achievements without revealing the titles.
    - Steps:
    - User navigates to the "Awards" section.
    - User selects "View Award-Winning Books."
- System calculates and displays each book's ID along with its corresponding award details such as the name of the award and the year won.

- Joint Query between Book and Award to display books (by ID) and their corresponding awards
- **SELECT Book.BookID, Award.AwardName, Award.YearWon**
  FROM Book
  JOIN Award ON Book.BookID = Award.BookID;

40. Publisher and Book Edition

- Use Case: List All Publishers with Their Book Editions
- Actor: Book Store User
- Purpose: To display a list of all publishers along with the details of the book editions they have published, using ISBNs as identifiers to maintain a focus on logistics and cataloging.
    - Steps:
    - User navigates to the "Publishers" section.
    - User selects "View Publishers and Their Editions."
- System calculates and displays each publisher's details along with the ISBNs of the book editions they have published.

- Joint Query between Publisher and BookEdition to show publisher names and the book editions (by ISBN) they have published.
- **SELECT Publisher.PubID, Publisher.PublishingHouse, BookEdition.ISBN**
  FROM Publisher
  JOIN BookEdition ON Publisher.PubID = BookEdition.PubID;

41. Book and Book Edition
    - Query: List all books and each of their editions' details.
    - Use Case: List All Books and Their Editions
    - Actor: Book Store User
    - Purpose: To display a list of all books using book IDs and provide detailed information about each of their editions, such as ISBN, format, and publication date, aiding in inventory management and catalog organization.
        - Steps:
            - User navigates to the "Books" section.
            - User selects "View Books and Their Editions."
    - System calculates and displays each book's ID along with the details of its editions.

    - <u>Joint Query between Book and BookEdition to list all editions for each book by ID</u>
    - **SELECT Book.BookID, BookEdition.ISBN, BookEdition.Format, BookEdition.PublicationDate FROM Book**
JOIN BookEdition ON Book.BookID = BookEdition.BookID;

42. Book Edition and Sales
    - Use Case: List Sales Details for Each Book Edition
    - Actor: Book Store User
    - Purpose: To display detailed sales data for each book edition, using ISBNs to uniquely identify editions, which is essential for sales analysis and inventory management.

        - Steps:
        - User navigates to the "Sales" section.
        - User selects "View Sales Details by Edition."
    - System calculates and displays sales data for each book edition, including sales date, item ID, and discount, all indexed by ISBN.

    - <u>Joint Query between BookEdition and Sales to retrieve sales data for each book edition by ISBN</u>

    - **SELECT BookEdition.ISBN, Salesf.SalesDate, Salesf.ItemID, Salesf.Discount**
FROM BookEdition
JOIN Salesf ON BookEdition.ISBN = Salesf.ISBN;

43. Book & Checkouts

- Use Case: Association between books and their checkouts.
- Actor: Book Store User
- Purpose: To display a list of checkouts for each book.
    - Steps:
        - Librarian navigates to the "Checkouts" section.
        - Librarian selects "View Checkouts for Books."
- The system displays a list of checkouts for each book, showing the number of times each book has been checked out, facilitating tracking of book circulation.
- Join Query for Book table along with the checkout details
- **SELECT Book.BookID, Book.Title, Book.AuthID, Book.Genre1, Book.Genre2, Book.SeriesID, Book.[Volume Number], Book.[Staff Comment], Checkout.CheckoutMonth, Checkout.[Number of Checkouts]**
  FROM Book
  LEFT JOIN Checkout ON Book.BookID = Checkout.BookID;

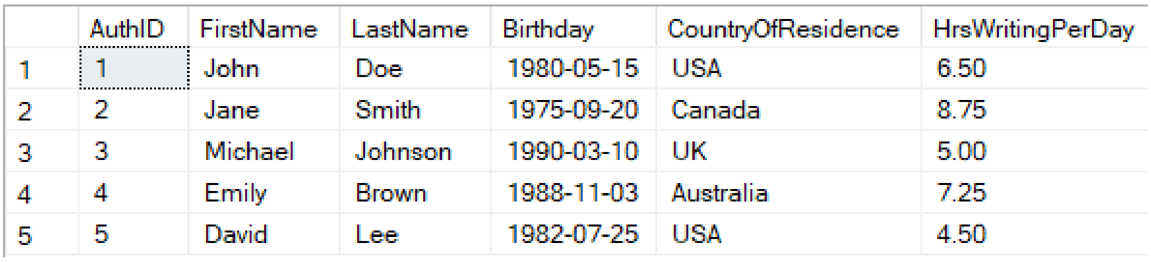
44. Ratings & Reviewers

- Use Case: To display a list of ratings provided by reviewers along with their information.
- Actor: Book Store User
- Purpose: To display a list of ratings provided by reviewers along with their information.
    - Steps:
        - User navigates to the "Ratings" section.
        - User selects "View Ratings and Reviewers."
- The system displays a list of ratings given by reviewers for each book, along with the reviewer's information (First Name, Last Name, Country of Residence), facilitating understanding of the ratings without revealing reviewer identifiers.
- Join Query for list of ratings provided by reviewers along with their info.
- **SELECT Rating.BookID, Book.Description, Reviewer.FirstName, Reviewer.LastName, Reviewer.CountryOfResidence, Rating.Rating**
  FROM Rating
  JOIN Reviewer ON Rating.ReviewerID = Reviewer.ReviewerID;

All Table Data:

1. Author Table

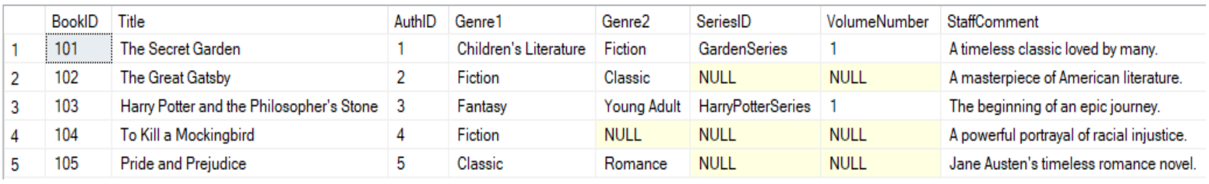
Select * from Author;



	AuthID	FirstName	LastName	Birthday	CountryOfResidence	HrsWritingPerDay
1	1	John	Doe	1980-05-15	USA	6.50
2	2	Jane	Smith	1975-09-20	Canada	8.75
3	3	Michael	Johnson	1990-03-10	UK	5.00
4	4	Emily	Brown	1988-11-03	Australia	7.25
5	5	David	Lee	1982-07-25	USA	4.50

2. Book Table

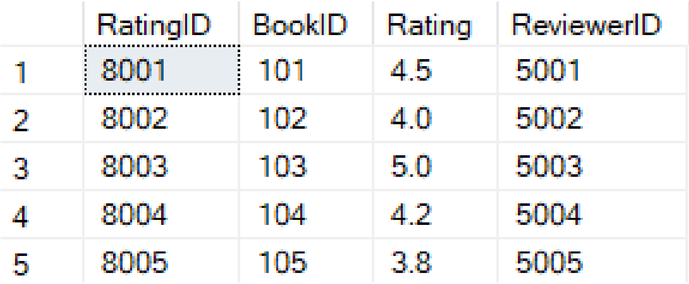
Select * from Book;



	BookID	Title	AuthID	Genre1	Genre2	SeriesID	VolumeNumber	StaffComment
1	101	The Secret Garden	1	Children's Literature	Fiction	GardenSeries	1	A timeless classic loved by many.
2	102	The Great Gatsby	2	Fiction	Classic	NULL	NULL	A masterpiece of American literature.
3	103	Harry Potter and the Philosopher's Stone	3	Fantasy	Young Adult	HarryPotterSeries	1	The beginning of an epic journey.
4	104	To Kill a Mockingbird	4	Fiction	NULL	NULL	NULL	A powerful portrayal of racial injustice.
5	105	Pride and Prejudice	5	Classic	Romance	NULL	NULL	Jane Austen's timeless romance novel.

3. Rating Table

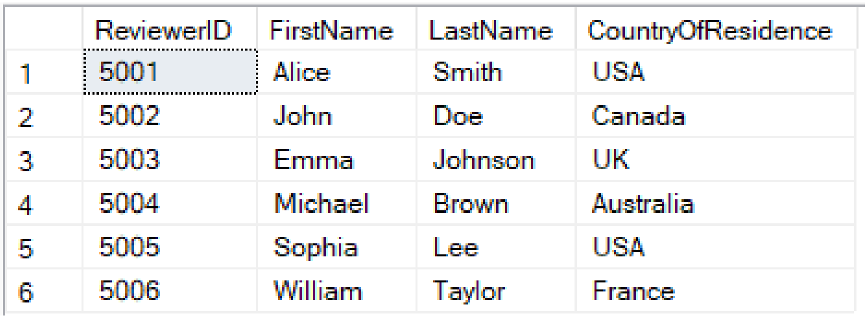
Select * from Rating;



	RatingID	BookID	Rating	ReviewerID
1	8001	101	4.5	5001
2	8002	102	4.0	5002
3	8003	103	5.0	5003
4	8004	104	4.2	5004
5	8005	105	3.8	5005

4. Reviewer Table

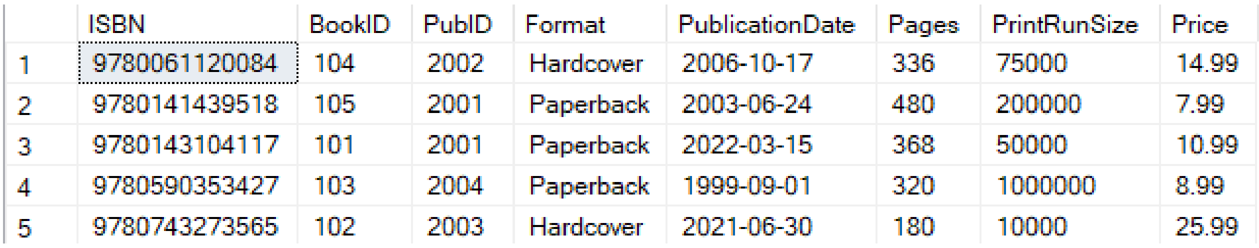
Select * from Reviewer;



	ReviewerID	FirstName	LastName	CountryOfResidence
1	5001	Alice	Smith	USA
2	5002	John	Doe	Canada
3	5003	Emma	Johnson	UK
4	5004	Michael	Brown	Australia
5	5005	Sophia	Lee	USA
6	5006	William	Taylor	France

5. Book Edition Table

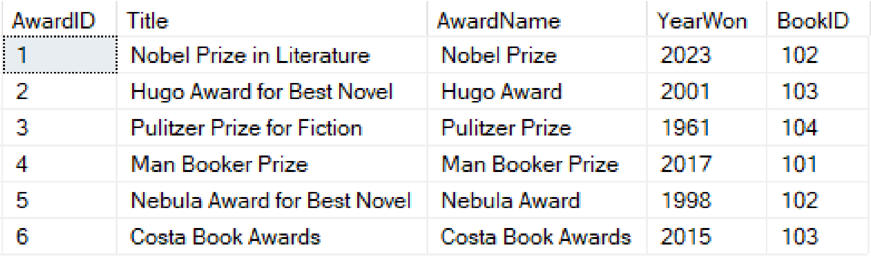
Select * from BookEdition;



	ISBN	BookID	PubID	Format	PublicationDate	Pages	PrintRunSize	Price
1	9780061120084	104	2002	Hardcover	2006-10-17	336	75000	14.99
2	9780141439518	105	2001	Paperback	2003-06-24	480	200000	7.99
3	9780143104117	101	2001	Paperback	2022-03-15	368	50000	10.99
4	9780590353427	103	2004	Paperback	1999-09-01	320	1000000	8.99
5	9780743273565	102	2003	Hardcover	2021-06-30	180	10000	25.99

6. Award Table

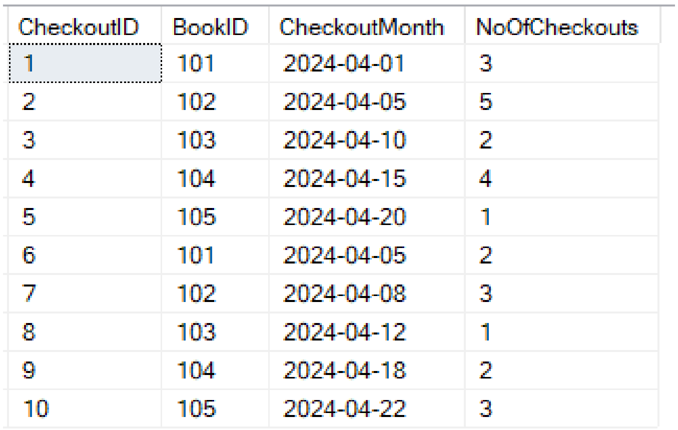
Select * from Award;



AwardID	Title	AwardName	YearWon	BookID
1	Nobel Prize in Literature	Nobel Prize	2023	102
2	Hugo Award for Best Novel	Hugo Award	2001	103
3	Pulitzer Prize for Fiction	Pulitzer Prize	1961	104
4	Man Booker Prize	Man Booker Prize	2017	101
5	Nebula Award for Best Novel	Nebula Award	1998	102
6	Costa Book Awards	Costa Book Awards	2015	103

7. Checkout Table

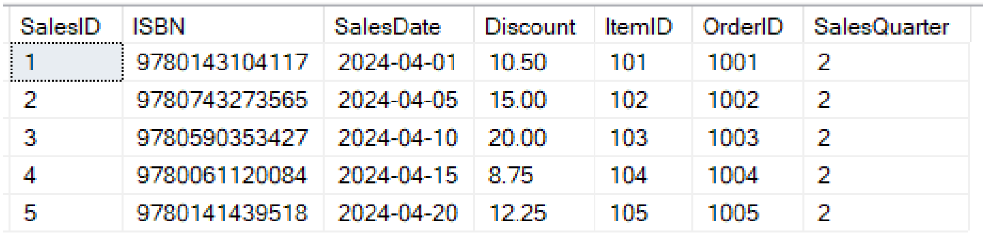
Select * from Checkouts;



CheckoutID	BookID	CheckoutMonth	NoOfCheckouts
1	101	2024-04-01	3
2	102	2024-04-05	5
3	103	2024-04-10	2
4	104	2024-04-15	4
5	105	2024-04-20	1
6	101	2024-04-05	2
7	102	2024-04-08	3
8	103	2024-04-12	1
9	104	2024-04-18	2
10	105	2024-04-22	3

8. Sales Table

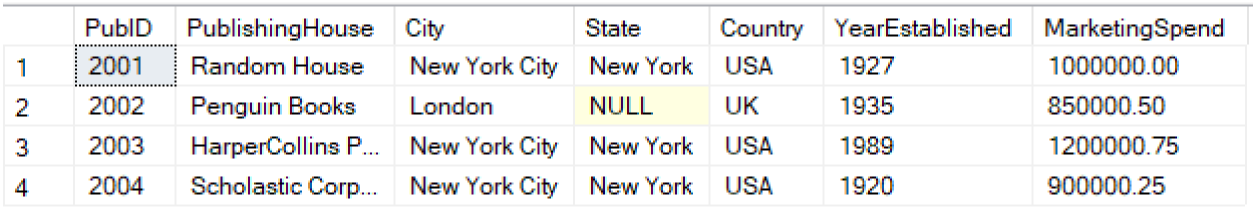
Select * from salesf;



SalesID	ISBN	SalesDate	Discount	ItemID	OrderID	SalesQuarter
1	9780143104117	2024-04-01	10.50	101	1001	2
2	9780743273565	2024-04-05	15.00	102	1002	2
3	9780590353427	2024-04-10	20.00	103	1003	2
4	9780061120084	2024-04-15	8.75	104	1004	2
5	9780141439518	2024-04-20	12.25	105	1005	2

9. Publisher Table

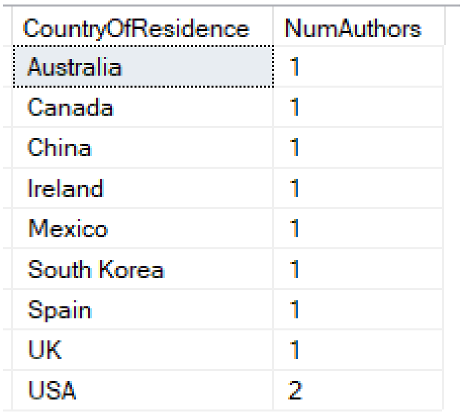
Select * from Publisher;



	PubID	PublishingHouse	City	State	Country	YearEstablished	MarketingSpend
1	2001	Random House	New York City	New York	USA	1927	1000000.00
2	2002	Penguin Books	London	NULL	UK	1935	850000.50
3	2003	HarperCollins P...	New York City	New York	USA	1989	1200000.75
4	2004	Scholastic Corp...	New York City	New York	USA	1920	900000.25

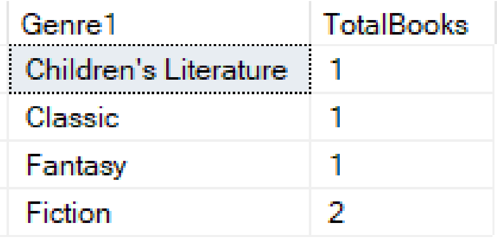
Aggregate Query for All Table:

1. **Author Table**
    - SELECT CountryOfResidence, COUNT(*) AS NumAuthors
      FROM Author
      GROUP BY CountryOfResidence;



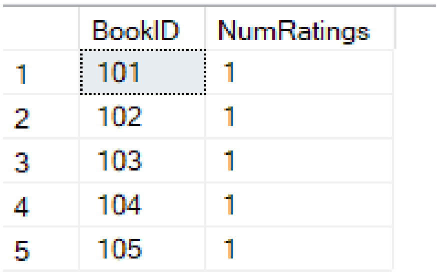
CountryOfResidence	NumAuthors
Australia	1
Canada	1
China	1
Ireland	1
Mexico	1
South Korea	1
Spain	1
UK	1
USA	2

2. **Book Table**
    - SELECT Genre1, COUNT(*) AS TotalBooks FROM Book GROUP BY Genre1;



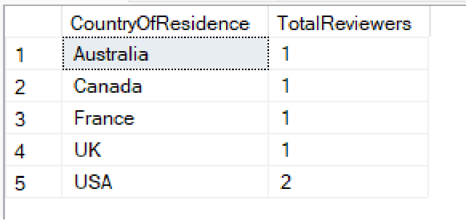
Genre1	TotalBooks
Children's Literature	1
Classic	1
Fantasy	1
Fiction	2

3. **Rating Table**
    - SELECT BookID, COUNT(*) AS NumRatings
      FROM Rating
      GROUP BY BookID;



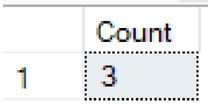
	BookID	NumRatings
1	101	1
2	102	1
3	103	1
4	104	1
5	105	1

4. **Reviewer Table**
   - SELECT CountryOfResidence, COUNT(*) AS TotalReviewers
     FROM Reviewer GROUP BY CountryOfResidence;



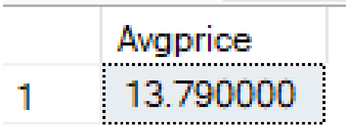
	CountryOfResidence	TotalReviewers
1	Australia	1
2	Canada	1
3	France	1
4	UK	1
5	USA	2

5. **Publisher Table**
   - SELECT COUNT(*) as Count FROM Publisher WHERE Country = 'USA';



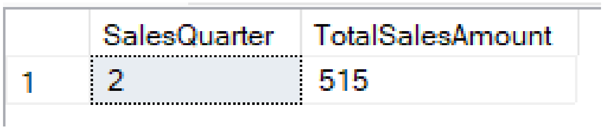
	Count
1	3

6. **Book Edition Table**
   - SELECT AVG(Price) as Avgprice FROM BookEdition;



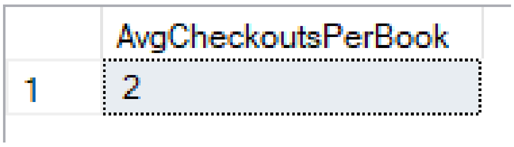
	Avgprice
1	13.790000

7. **Sales Table**
   - SELECT SalesQuarter, SUM(ItemID) AS TotalSalesAmount
     FROM Salesf
     GROUP BY SalesQuarter;



	SalesQuarter	TotalSalesAmount
1	2	515

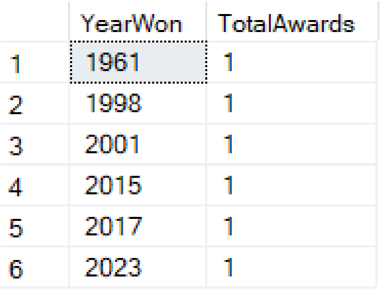
8. **Checkout Table**
   - SELECT AVG(NoOfCheckouts) AS AvgCheckoutsPerBook FROM Checkouts;



	AvgCheckoutsPerBook
1	2

9. Award Table

-
  ```
  SELECT YearWon, COUNT(*) AS TotalAwards
  FROM Award
  GROUP BY YearWon
  ORDER BY YearWon;
  ```

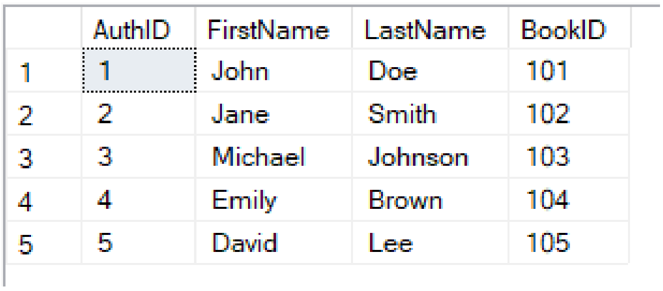


	YearWon	TotalAwards
1	1961	1
2	1998	1
3	2001	1
4	2015	1
5	2017	1
6	2023	1

Join Query for All Table:

1. Author & Book Relationship

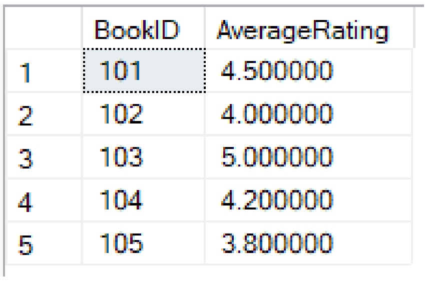
-
  ```
  SELECT Author.AuthID, Author.FirstName, Author.LastName, Book.BookID
  FROM Author
  JOIN Book ON Author.AuthID = Book.AuthID;
  ```



	AuthID	FirstName	LastName	BookID
1	1	John	Doe	101
2	2	Jane	Smith	102
3	3	Michael	Johnson	103
4	4	Emily	Brown	104
5	5	David	Lee	105

2. Book & Rating Relationship

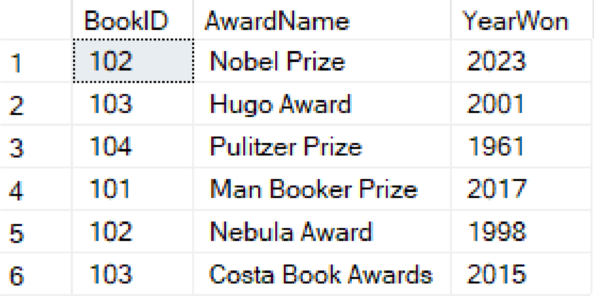
-
  ```
  SELECT Book.BookID, AVG(Rating.Rating) AS AverageRating
  FROM Book
  JOIN Rating ON Book.BookID = Rating.BookID
  GROUP BY Book.BookID;
  ```



	BookID	AverageRating
1	101	4.500000
2	102	4.000000
3	103	5.000000
4	104	4.200000
5	105	3.800000

3. Book & Award Relationship

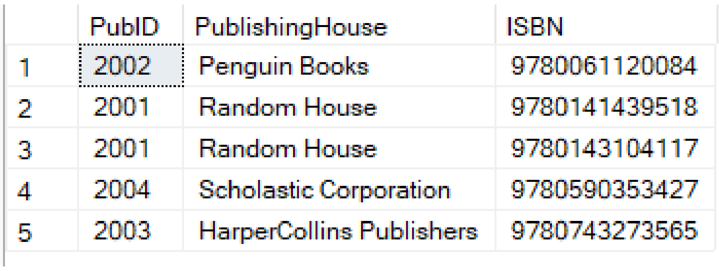
- SELECT Book.BookID, Award.AwardName, Award.YearWon
  FROM Book
  JOIN Award ON Book.BookID = Award.BookID;



	BookID	AwardName	YearWon
1	102	Nobel Prize	2023
2	103	Hugo Award	2001
3	104	Pulitzer Prize	1961
4	101	Man Booker Prize	2017
5	102	Nebula Award	1998
6	103	Costa Book Awards	2015

4. Publisher & Book Edition Relationship

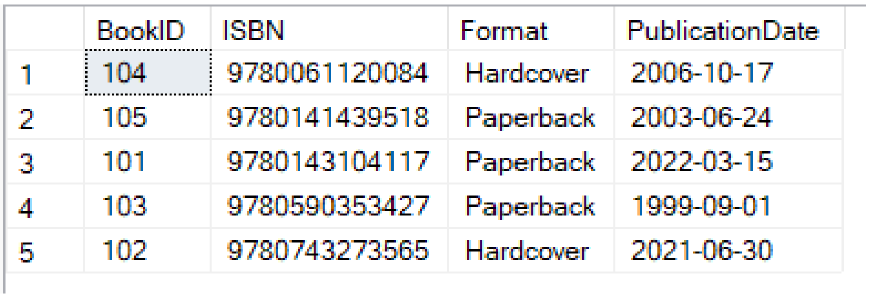
- SELECT Publisher.PubID, Publisher.PublishingHouse, BookEdition.ISBN
  FROM Publisher
  JOIN BookEdition ON Publisher.PubID = BookEdition.PubID;



	PubID	PublishingHouse	ISBN
1	2002	Penguin Books	9780061120084
2	2001	Random House	9780141439518
3	2001	Random House	9780143104117
4	2004	Scholastic Corporation	9780590353427
5	2003	HarperCollins Publishers	9780743273565

5. Book & Book Edition Relationship

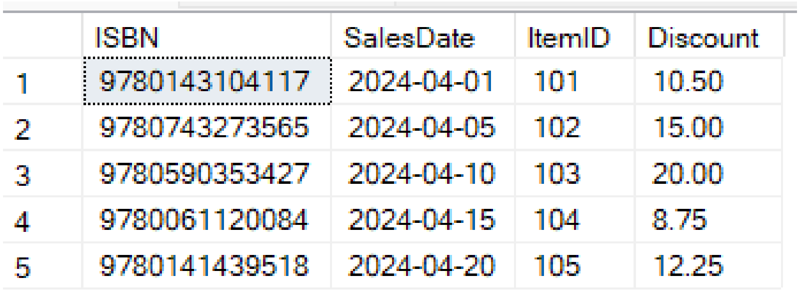
- SELECT Book.BookID, BookEdition.ISBN, BookEdition.Format, BookEdition.PublicationDate
  FROM Book
  JOIN BookEdition ON Book.BookID = BookEdition.BookID;



	BookID	ISBN	Format	PublicationDate
1	104	9780061120084	Hardcover	2006-10-17
2	105	9780141439518	Paperback	2003-06-24
3	101	9780143104117	Paperback	2022-03-15
4	103	9780590353427	Paperback	1999-09-01
5	102	9780743273565	Hardcover	2021-06-30

6. Book Edition & Sales Relationship

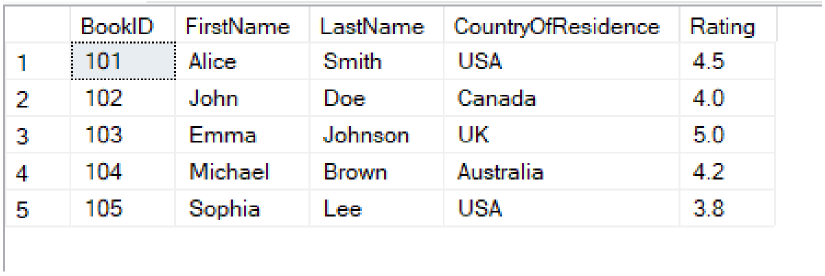
- SELECT BookEdition.ISBN, Salesf.SalesDate, Salesf.ItemID, Salesf.Discount
  FROM BookEdition
  JOIN Salesf ON BookEdition.ISBN = Salesf.ISBN;



	ISBN	SalesDate	ItemID	Discount
1	9780143104117	2024-04-01	101	10.50
2	9780743273565	2024-04-05	102	15.00
3	9780590353427	2024-04-10	103	20.00
4	9780061120084	2024-04-15	104	8.75
5	9780141439518	2024-04-20	105	12.25

7. Rating & Reviewers Relationship

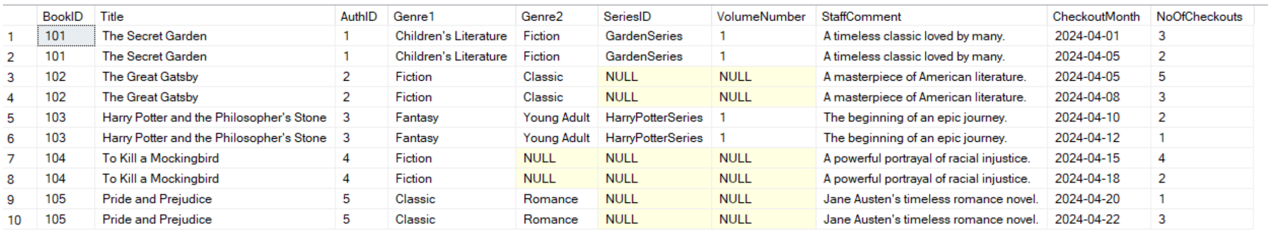
- SELECT Rating.BookID, Reviewer.FirstName, Reviewer.LastName, Reviewer.CountryOfResidence, Rating.Rating
  FROM Rating
  JOIN Reviewer ON Rating.ReviewerID = Reviewer.ReviewerID;



	BookID	FirstName	LastName	CountryOfResidence	Rating
1	101	Alice	Smith	USA	4.5
2	102	John	Doe	Canada	4.0
3	103	Emma	Johnson	UK	5.0
4	104	Michael	Brown	Australia	4.2
5	105	Sophia	Lee	USA	3.8

8. Book & Checkout Relationship

- SELECT Book.BookID, Book.Title, Book.AuthID, Book.Genre1, Book.Genre2, Book.SeriesID, Book.VolumeNumber, Book.StaffComment, Checkouts.CheckoutMonth, Checkouts.NoOfCheckouts
  FROM Book
  LEFT JOIN Checkouts ON Book.BookID = Checkouts.BookID;



	BookID	Title	AuthID	Genre1	Genre2	SeriesID	VolumeNumber	StaffComment	CheckoutMonth	NoOfCheckouts
1	101	The Secret Garden	1	Children's Literature	Fiction	GardenSeries	1	A timeless classic loved by many.	2024-04-01	3
2	101	The Secret Garden	1	Children's Literature	Fiction	GardenSeries	1	A timeless classic loved by many.	2024-04-05	2
3	102	The Great Gatsby	2	Fiction	Classic	NULL	NULL	A masterpiece of American literature.	2024-04-05	5
4	102	The Great Gatsby	2	Fiction	Classic	NULL	NULL	A masterpiece of American literature.	2024-04-08	3
5	103	Harry Potter and the Philosopher's Stone	3	Fantasy	Young Adult	HarryPotterSeries	1	The beginning of an epic journey.	2024-04-10	2
6	103	Harry Potter and the Philosopher's Stone	3	Fantasy	Young Adult	HarryPotterSeries	1	The beginning of an epic journey.	2024-04-12	1
7	104	To Kill a Mockingbird	4	Fiction	NULL	NULL	NULL	A powerful portrayal of racial injustice.	2024-04-15	4
8	104	To Kill a Mockingbird	4	Fiction	NULL	NULL	NULL	A powerful portrayal of racial injustice.	2024-04-18	2
9	105	Pride and Prejudice	5	Classic	Romance	NULL	NULL	Jane Austen's timeless romance novel.	2024-04-20	1
10	105	Pride and Prejudice	5	Classic	Romance	NULL	NULL	Jane Austen's timeless romance novel.	2024-04-22	3

Conclusion:

This project aims to improve bookshop operations by implementing a robust database management system. With this system, we can efficiently store, retrieve, and analyse diverse datasets, enhancing overall efficiency and decision-making processes. The system ensures data integrity, supports searches, generates reports, and facilitates quick data retrieval. By involving bookshop staff at all levels, we empower them with the necessary tools to manage operations effectively and adapt to changing business requirements.

References:

- Ramakrishnan, R., & Gehrke, J. (2003). Database Management Systems (3rd ed.)
- Connolly, T. M., & Begg, C. E. (2014). Database Systems: A Practical Approach to Design, Implementation, and Management (6th ed.).
- Elmasri, R., & Navathe, S. B. (2015). Fundamentals of Database Systems (7th ed.).
- Date, C. J. (2003). An Introduction to Database Systems (8th ed.). Addison-Wesley.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). Database System Concepts (6th ed.). McGraw-Hill.

These references provide valuable insights into database management principles, design methodologies, and practical approaches to implementing database systems. They serve as foundational resources for understanding and developing the database management system outlined in this project.