**SVKM’s NMIMS**

**School of Technology Management & Engineering**

A.Y. 2023 - 24

**Course: Database Management Systems**

**Project Report**

|  |  |  |
| --- | --- | --- |
| Program | B tech CE | |
| Semester | 4 | |
| Name of the Project: | Online Bookshop Management System | |
|  | | |
| Details of Project Members |  |  |
| Batch | Roll No. | Name |
| 2 | T034 | Kalash Maheshwari |
| 2 | T069 | Yashwardhan Sant |
|  |  |  |
| Date of Submission: | | |

**Contribution of each project Members:**

|  |  |  |
| --- | --- | --- |
| Roll No. | Name: | Contribution |
| T034 | Kalash Maheshwari | 50% |
| T069 | Yashwardhan Sant | 50% |

**Github link of your project:**

**Note:**

1. Create a readme file if you have multiple files
2. All files must be properly named (R004\_DBMSProject)
3. Submit all relevant files of your work (Report, all SQL files, Any other files)
4. **Plagiarism is highly discouraged (Your report will be checked for plagiarism)**

**Rubrics for the Project evaluation:**

|  |  |
| --- | --- |
| First phase of evaluation:  Innovative Ideas (5 Marks)  Design and Partial implementation (5 Marks) | 10 marks |
| Final phase of evaluation  Implementation, presentation and viva, Self-Learning and Learning Beyond classroom | 10 marks |

**Project Report**

**Online Bookshop Management System**

**by**

**Student 1, T034: Kalash Maheshwari**

**Student 2, T069: Yashwardhan Sant**

**Course: DBMS**

**AY: 2023-24**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Topic** | **Page no.** |
| **1** | Storyline |  |
| **2** | Components of Database Design |  |
| **3** | Entity Relationship Diagram |  |
| **4** | Relational Model |  |
| **5** | Normalization |  |
| **6** | SQL Queries |  |
| **7** | Learning from the Project |  |
| **8** | Project Demonstration |  |
| **9** | Self-learning beyond classroom |  |
| **10** | Learning from the project |  |
| **8** | Challenges faced |  |
| **9** | Conclusion |  |

**I. Storyline**

The "Online Bookstore" website project is designed for online book sales. Customers can browse a large selection, search for titles, and place orders. The website manages customer information, book details, orders, publishers, and authors. It uses a relational database to track customer purchases, book availability, and author/publisher information, offering a convenient and efficient way to buy books online.

**II. Components of Database Design**

**Customer Table:**

**Attributes:**

* **customer\_id**
* **customer\_name**
* **house\_number**
* **street\_name**
* **additional\_address**
* **city**
* **state**
* **postal\_code**
* **contact**

**Author Table:**

**Attributes:**

* **author\_id**
* **author\_Fname**
* **author\_lname**
* **qualification**

**Publisher Table:**

**Attributes:**

* **publisher\_id**
* **publisher\_name**
* **country**

**Book Table:**

**Attributes:**

* **book\_id**
* **book\_title**
* **book\_price**
* **book\_status**
* **subject**

**Order\_Details Table:**

**Attributes:**

* **order\_id**
* **order\_date**
* **order\_amount**

**Order\_Book Table:**

**Attributes:**

* **order\_id**
* **book\_id**

**Makes Table:**

**Attributes:**

* **order\_id**
* **customer\_id**

**Written\_By Table:**

**Attributes:**

* **book\_id**
* **author\_id**

**Publish Table:**

**Attributes:**

* **book\_id**
* **publisher\_id**

**Relationship:**

**Book to Author Relationship:**

* Many-to-Many
* Books can have multiple authors, and authors can write multiple books.
* Mediated by the Written\_By table.

**Book to Publisher Relationship:**

* One-to-Many
* Each book is published by only one publisher, but a publisher can publish multiple books.

**Order\_Details to Customer Relationship:**

* One-to-Many
* Each order is placed by only one customer, but a customer can place multiple orders.

**Order\_Details to Book Relationship:**

* One-to-Many
* Each order can contain multiple books, and a book can be in multiple orders.

**Order\_Book to Order\_Details Relationship:**

* One-to-Many
* Each order-book entry belongs to only one order, but an order can have multiple order-book entries.

**Order\_Book to Book Relationship:**

* One-to-Many
* Each order-book entry is associated with only one book, but a book can be in multiple order-book entries.

**Makes to Order\_Details Relationship:**

* One-to-Many
* Each makes entry corresponds to only one order, but an order can have multiple makes entries.

**Makes to Customer Relationship:**

* One-to-Many
* Each makes entry corresponds to only one customer, but a customer can have multiple makes entries.

**Written\_By to Book Relationship:**

* One-to-Many
* Each written\_by entry corresponds to only one book, but a book can have multiple written\_by entries (multiple authors).

**Written\_By to Author Relationship:**

* One-to-Many
* Each written\_by entry corresponds to only one author, but an author can have multiple written\_by entries (multiple books).

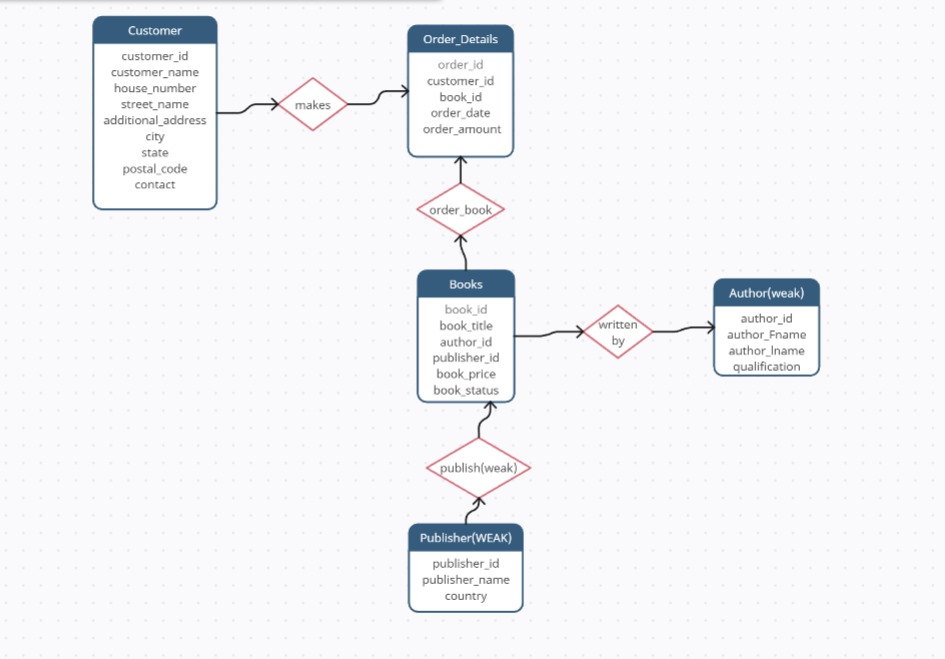
**Publish to Book Relationship:**

* One-to-Many
* Each publish entry corresponds to only one book, but a book can have multiple publish entries (multiple publishers).

**Publish to Publisher Relationship:**

* One-to-Many
* Each publish entry corresponds to only one publisher, but a publisher can have multiple publish entries (publishes multiple books).

**III. Entity Relationship Diagram**



**IV. Relational Model**

1. Order\_book:

Attributes:

Order\_id (Primary Key, Foreign Key referencing Order\_Details)

BookId (Primary Key, Foreign Key referencing Book)

Connected Tables: Books,Order\_Details

1. Makes:

Attributes:

Order\_id (Primary Key, Foreign Key referencing Order\_Details)

customer\_id (Primary Key, Foreign Key referencing customer)

Connected Tables: Books, customer

1. Written\_By:

Attributes:

author\_id (Primary Key, Foreign Key referencing Author)

BookId (Primary Key, Foreign Key referencing Book)

Connected Tables: Books, Author

1. Publish:

Attributes:

publisher\_id (Primary Key, Foreign Key referencing Publisher)

BookId (Primary Key, Foreign Key referencing Book)

Connected Tables: Books, Publisher

**V. Normalization**

**Customer Table:**

**Attributes:**

* **customer\_id**
* **customer\_name**
* **house\_number**
* **street\_name**
* **additional\_address**
* **city**
* **state**
* **postal\_code**

**Normal Forms:**

**First Normal Form (1NF): All attributes are atomic.**

**Second Normal Form (2NF): No partial dependencies exist.**

**Third Normal Form (3NF): No transitive dependencies exist.**

**Conclusion: The Customer table is already in 3NF.**

**Author Table:**

**Attributes:**

* **author\_id**
* **author\_Fname**
* **author\_lname**
* **qualification**

**Normal Forms:**

**First Normal Form (1NF): All attributes are atomic.**

**Second Normal Form (2NF): No partial dependencies exist.**

**Third Normal Form (3NF): No transitive dependencies exist.**

**Conclusion: The Author table is already in 3NF**

**Publisher Table:**

**Attributes:**

* **publisher\_id**
* **publisher\_name**
* **country**

**Normal Forms:**

**First Normal Form (1NF): All attributes are atomic.**

**Second Normal Form (2NF): No partial dependencies exist.**

**Third Normal Form (3NF): No transitive dependencies exist.**

**Conclusion: The Publisher table is already in 3NF**

**Book Table:**

**Attributes:**

* **book\_id**
* **book\_title**
* **book\_price**
* **book\_status**
* **subject**

**Normal Forms:**

**First Normal Form (1NF): All attributes are atomic.**

**Second Normal Form (2NF): No partial dependencies exist.**

**Third Normal Form (3NF): No transitive dependencies exist.**

**Conclusion: The Book table is already in 3NF**

**Order\_Details Table:**

**Attributes:**

* **order\_id**
* **order\_date**
* **order\_amount**

**Normal Forms:**

**First Normal Form (1NF): All attributes are atomic.**

**Second Normal Form (2NF): No partial dependencies exist.**

**Third Normal Form (3NF): No transitive dependencies exist.**

**Conclusion: The Order\_Details table is already in 3NF**

**Order\_Book Table:**

**Attributes:**

* **order\_id**
* **book\_id**

**Normal Forms:**

**First Normal Form (1NF): All attributes are atomic.**

**Second Normal Form (2NF): No partial dependencies exist.**

**Third Normal Form (3NF): No transitive dependencies exist.**

**Conclusion: The Order\_Book table is already in 3NF**

**Makes Table:**

**Attributes:**

* **order\_id**
* **customer\_id**

**Normal Forms:**

**First Normal Form (1NF): All attributes are atomic.**

**Second Normal Form (2NF): No partial dependencies exist.**

**Third Normal Form (3NF): No transitive dependencies exist.**

**Conclusion: The Makes table is already in 3NF**

**Written\_By Table:**

**Attributes:**

* **book\_id**
* **author\_id**

**Normal Forms:**

**First Normal Form (1NF): All attributes are atomic.**

**Second Normal Form (2NF): No partial dependencies exist.**

**Third Normal Form (3NF): No transitive dependencies exist.**

**Conclusion: The Written\_By table is already in 3NF**

**Publish Table:**

**Attributes:**

* **book\_id**
* **publisher\_id**

**Normal Forms:**

**First Normal Form (1NF): All attributes are atomic.**

**Second Normal Form (2NF): No partial dependencies exist.**

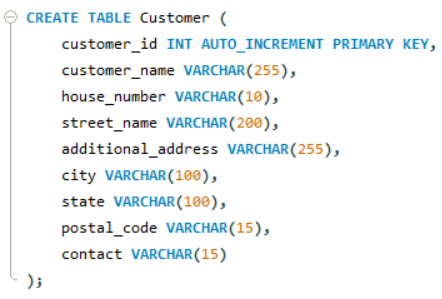
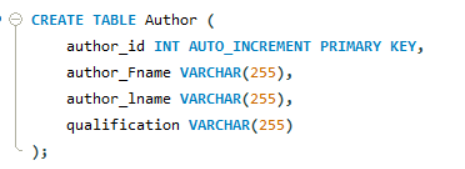
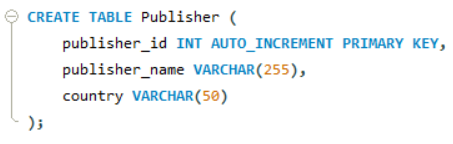
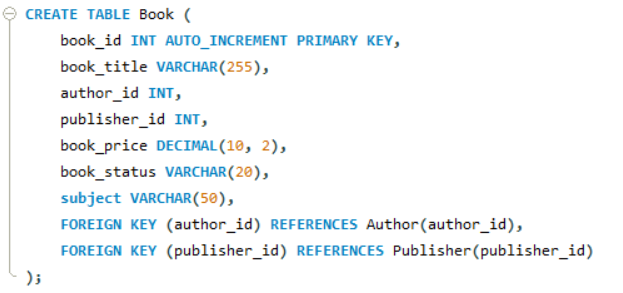
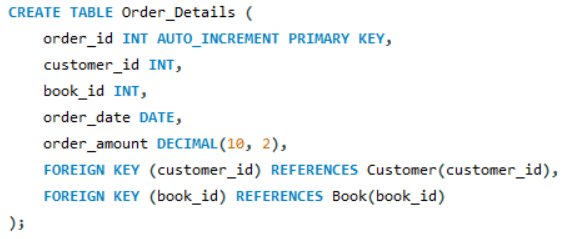
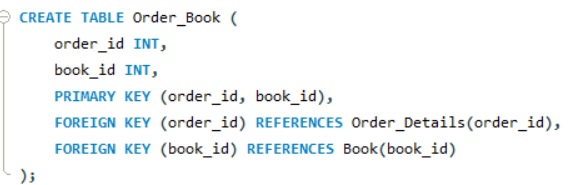
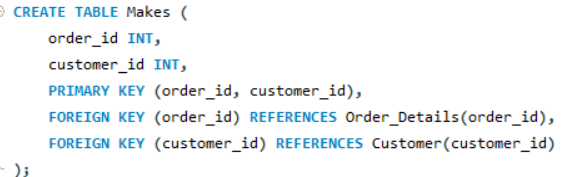
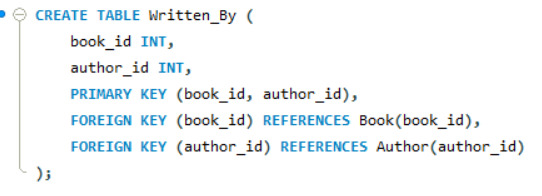
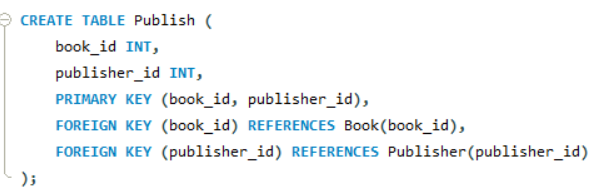
**Third Normal Form (3NF): No transitive dependencies exist.**

**Conclusion: The Publish table is already in 3NF**

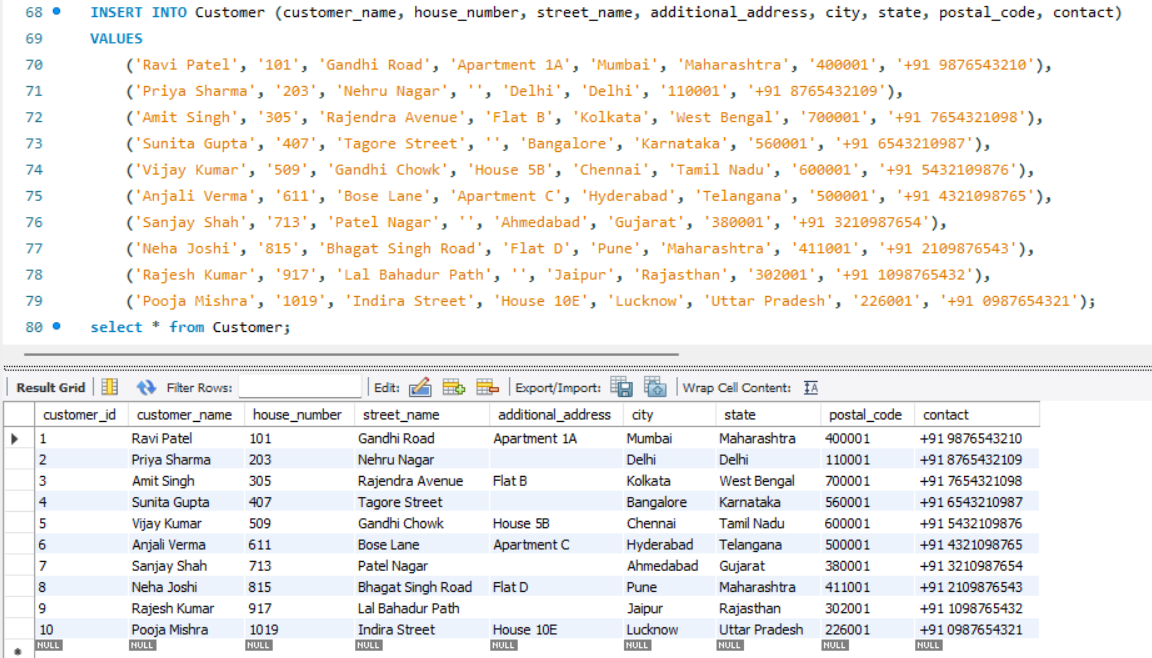
**VI. SQL Queries**

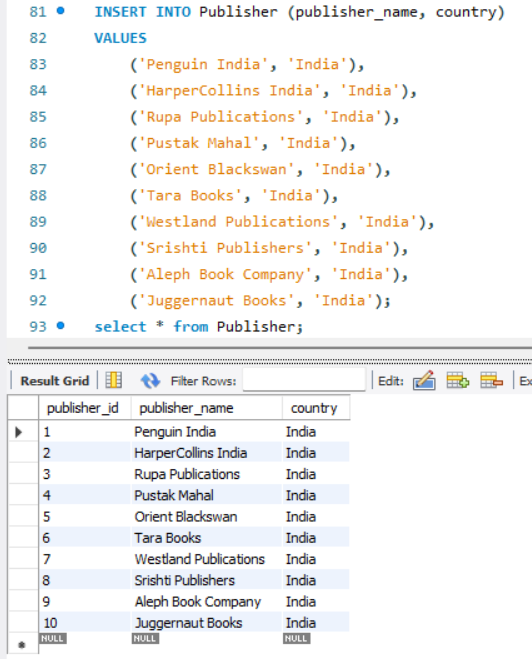
Using a DBMS software (SQLite3 or MySQL or any other of your choice):

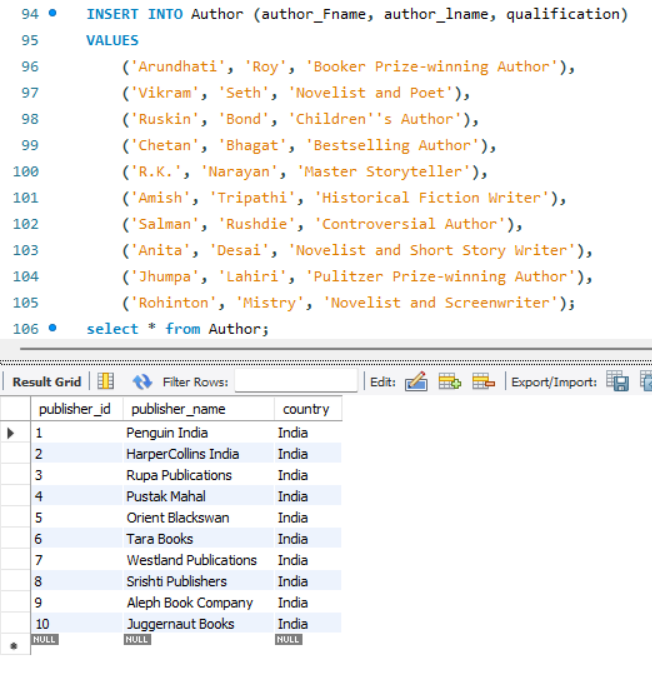
* Create the tables

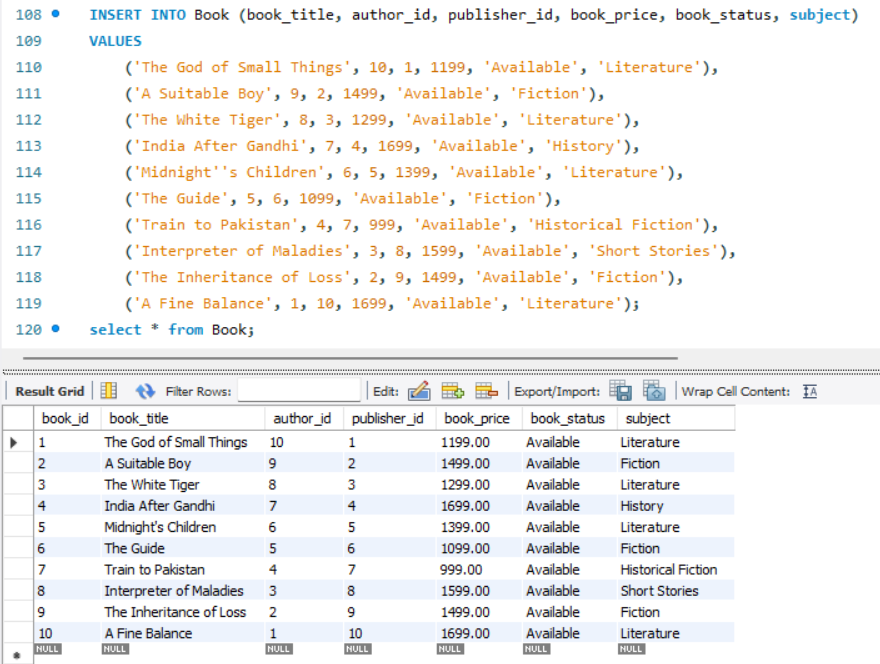


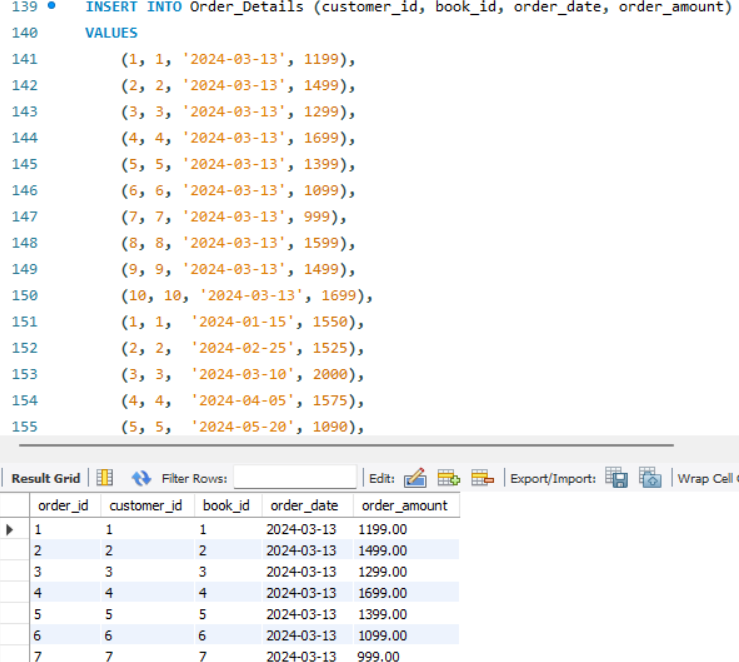
* Populate the tables (insert some meaningful data, at least 10 tuples for each relation)

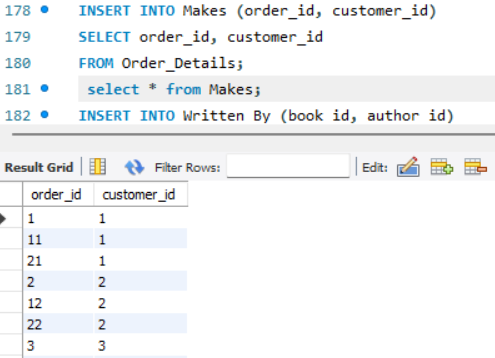
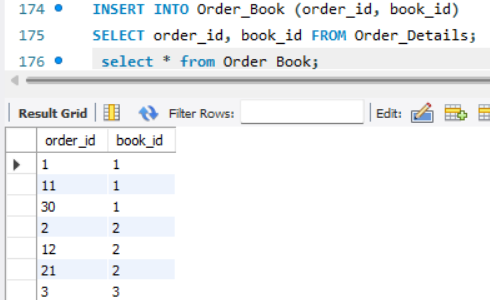
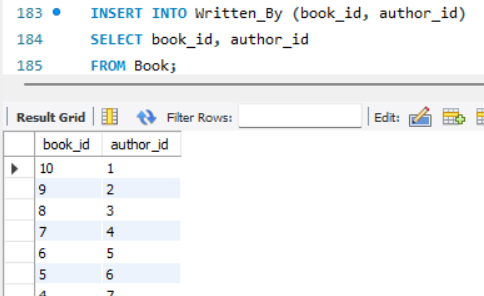
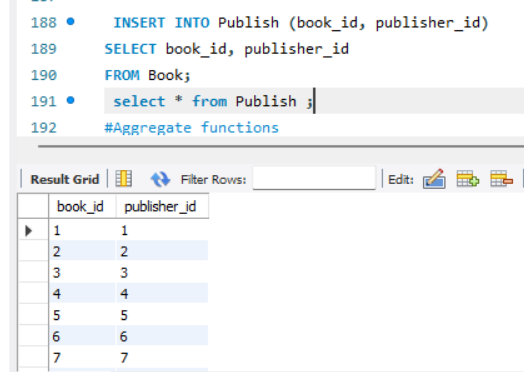




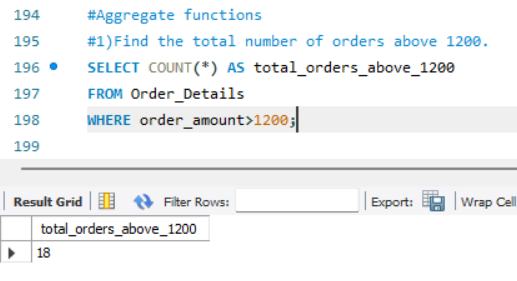


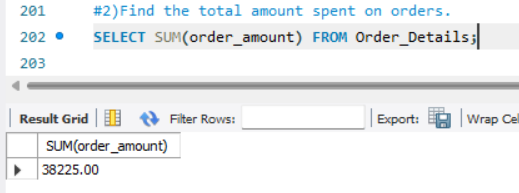


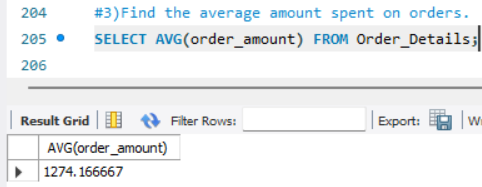


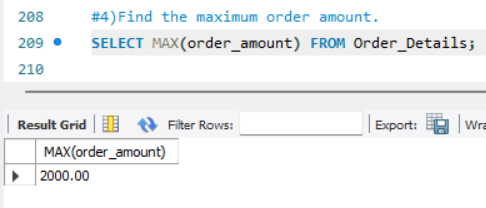
  

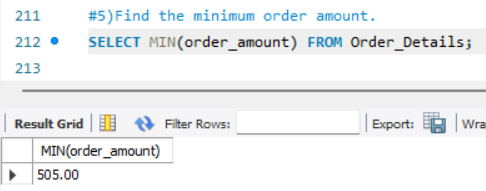
* Run SQL queries (minimum 20) covering **all concepts** learned in the class

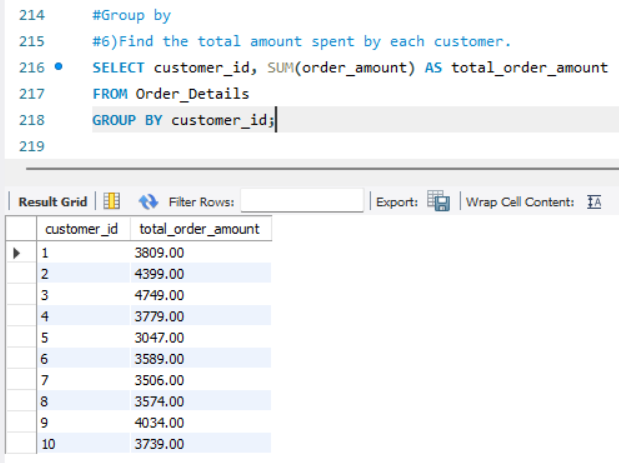
****

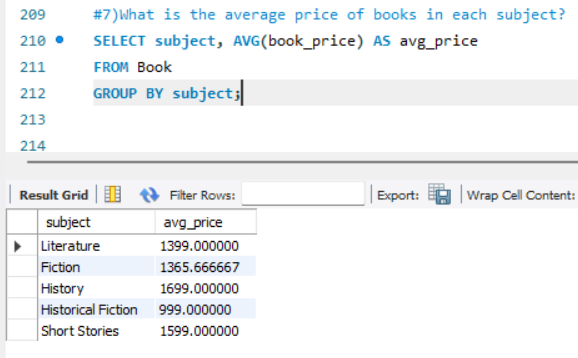
****

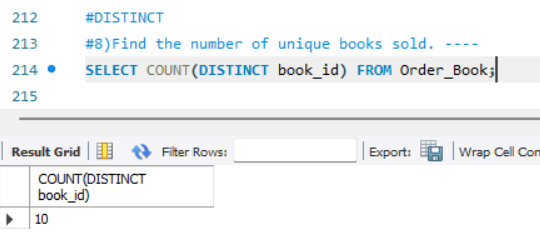
****

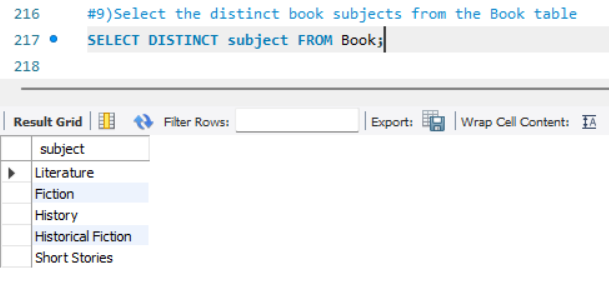
****

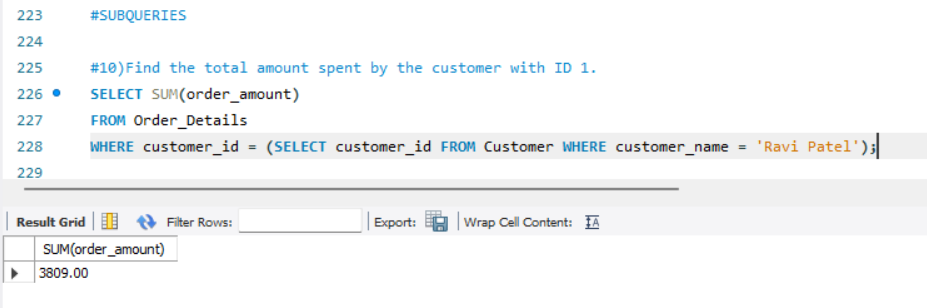
****

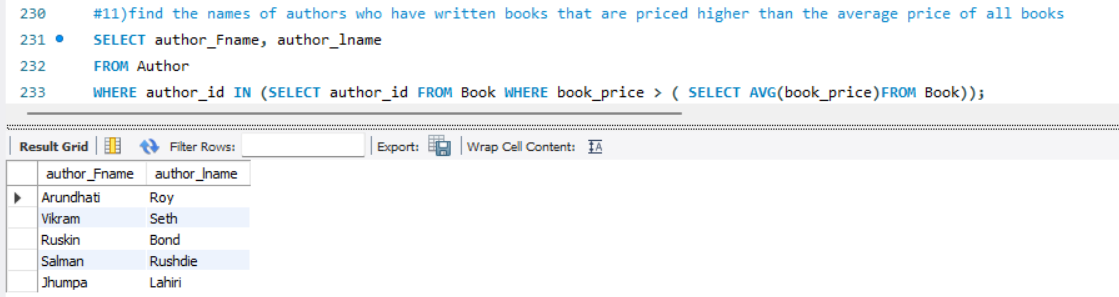
****

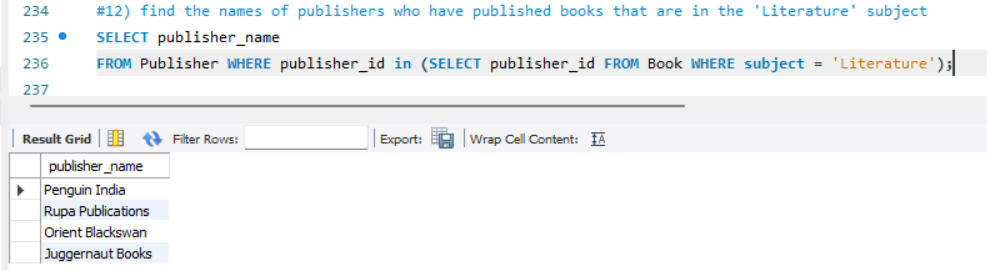
****

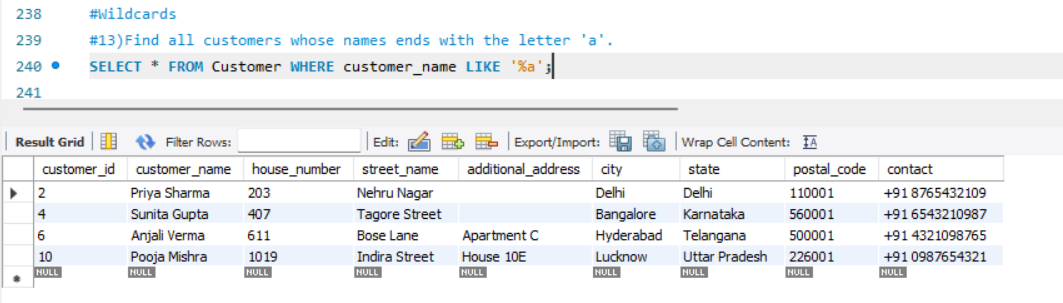
****

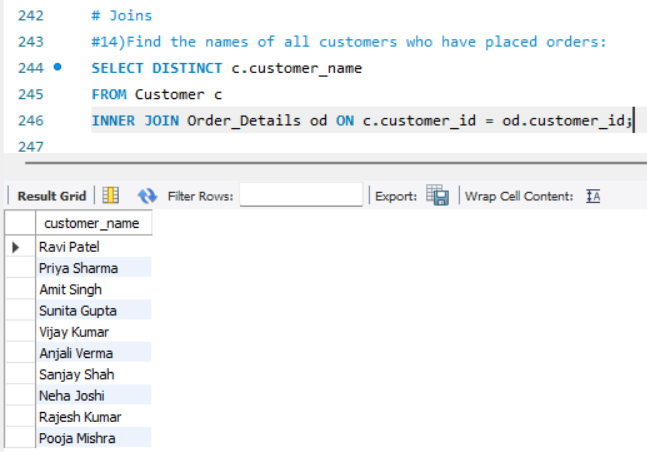
****

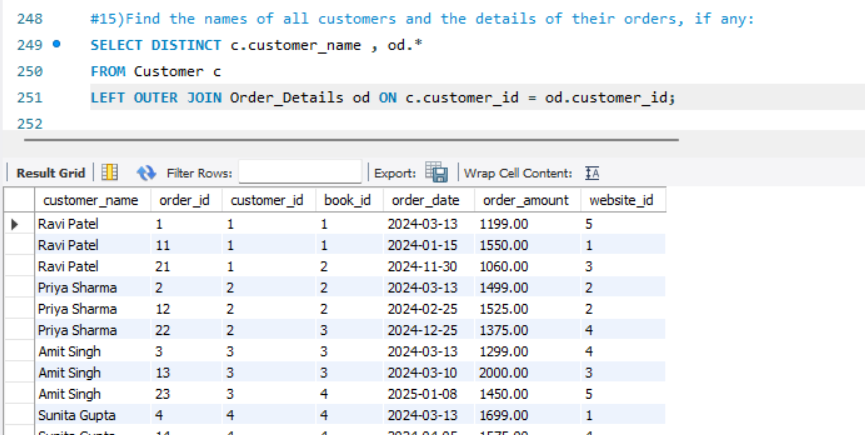
****

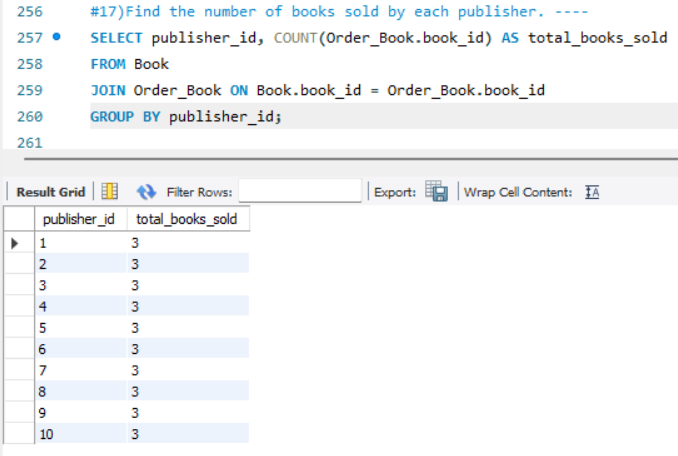
****

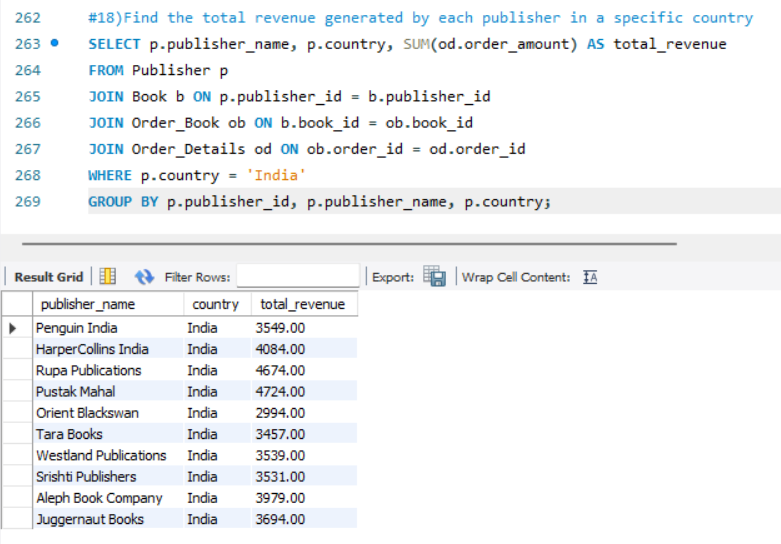
****

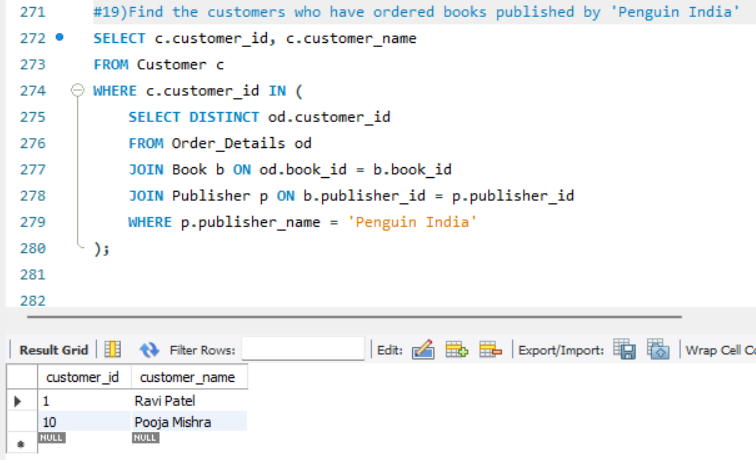
****

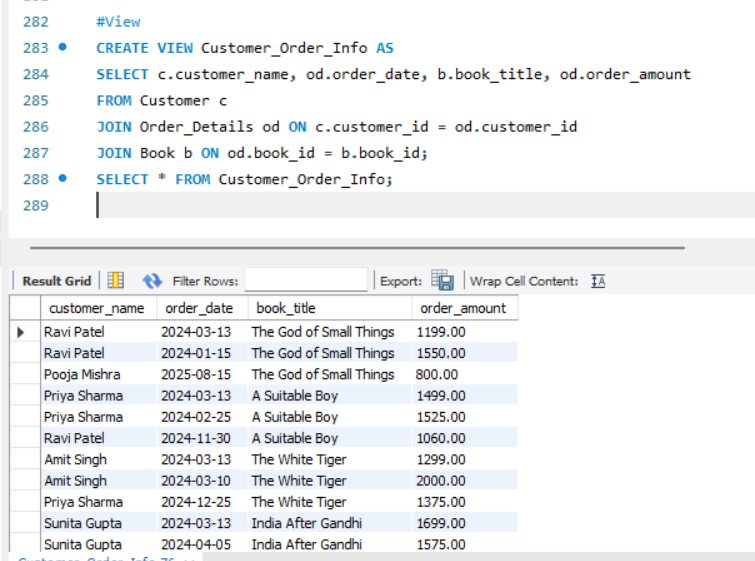
****

****

****

****

****



**VI. Project demonstration**

* Tools/software/ libraries used

MySQL Workbench

Creately

**VII. Self -Learning beyond classroom**

We have created a detailed database schema for an online book store and populated it with some sample data. Additionally, you've written SQL queries to perform various operations on this data, such as aggregation, grouping, subqueries, joins, and creating views. This setup can serve as a foundation for building an online book store system.

If you have any specific questions or if there's anything else you

**VIII. Learning from the Project**

This project provides hands-on learning in database management, covering design, normalization, SQL querying, and business logic implementation. By creating tables like Customer, Author, and Book, and executing operations such as data insertion and querying, learners gain practical experience. The project's simulation of an online book store demonstrates real-world application, fostering skills applicable to software development, data engineering, and database administration.

**IX. Challenges Faced**

Database Design Complexity: Designing a database schema that accurately represents the relationships between various entities while minimizing redundancy can be challenging, especially for complex systems.

Data Consistency: Ensuring data consistency and integrity, particularly when dealing with multiple related tables and foreign key constraints, requires careful attention to detail.

**X. Conclusion**

This project offers a hands-on learning experience in database management, covering schema design, SQL querying, and business logic implementation. Challenges such as query optimization and data consistency highlight the complexities of real-world database scenarios. Through problem-solving and continuous learning, learners develop valuable skills applicable across software development, data engineering, and database administration domains. This project serves as a foundation for understanding the role of databases in modern applications and fosters growth in database management expertise.