ONLINE BOOK LIBRARY is a project for the course CSC 4710. A database which can be used to keep track of authors, publishers, books, readers, and Category information and demonstrate the relationships between each of them.

The project will be written using MySQL and will also include Entity -Relationship schematic diagrams

**Online Book Library Project**

**CSC 4710 Database Systems Spring 2021**

**Saahil Karnik**

**Table of Contents**

PHASE 1 1

Requirements Analysis 1

Data Requirements1

Relationship Cardinality3

Functional Requirements 3

Phase 2 5

Conceptual Design5

Phase 3 7

Relational Design7

Phase 4 8

Data Dictionary 8

Phase 5 10

Tables10

Queries14

Phase 6 20

Views20

Triggers21

Conclusion 23

# PHASE 1

01

## Introduction

Online Library Management System maintains the information about the books present in the library, their authors, the members of library to whom books are issued, library staff etc. As this is very difficult to organize manually, maintenance of all this information is a very complex task. Due to recent advances in technology, organization of an Online Library has become much simple.

My current project is a database of an ‘Online Book Library’ and I dedicate this project to my fellow book-readers. This database can be used to keep track of authors, publishers, books, readers, and issue information and demonstrate the relationships between them. The project will be written using MySQL and will also include Entity -Relationship schematic diagrams.

The Online Library Management system has been designed to computerize and automate the operations performed over the information about the members, book issues and returns and all other operations. This computerization of library reduces the workload of management as most of the manual work done is reduced.

## Requirements Analysis

### Data Requirements

The online database library is made up of different sections as follows:

Author: This section contains data about the personal information of the author like his full name, email and contact number. The Author ID serves as the primary key.

Publisher: This section contains data about the publisher information like Publisher ID, Publisher name, contact address and the no. of books published by that publisher.

02

Category: The section contains information about different category for books like Science, Crafting, Computer etc. It contains Category Id and Category Name. Each category has a subcategory.

Subcategory: This will contain information about subcategory of books. Eg: Category of Crafting is subdivided into subcategories such as Painting, Beading, Fabric decorating etc. Subcategory contains id and name of the book.

Book: This section is the main data of the system. Each book has many attributes like ISBN, Title, price, page count, edition, its availability, publication year etc. Additionally, it will be associated with Category Id, Author Id and Publisher Id to help quicker searching and generating efficient query.

Reader: This section has information about reader like id, name, sex, age, email, address, phone number etc. It also has the date the reader registers in the online library.

Book\_reader: This will have information about reader id and book ISBN. It serves to map a particular book in the online library to the reader who has enrolled in that library.

The Entity and attributes can be referred in the Table 1 below:

|  |  |
| --- | --- |
| **Table 1: Entity and Attributes** | |
| Entity | Attribute |
| Author | AuthorID, FName, LName, Email, Phone |
| Publisher | PublisherID, PName, Email, Books\_published, Address, Phone |
| Reader | Reader Id, FName, LName, Age, Address, Phone, Email, Sex, Date\_of\_registration |
| **Table 1: Entity and Attributes-ctd.** | |
| Book | ISBN, Availability, Book\_Title, Price, Page\_count, Edition, Category Id, Author\_Id, Publisher\_Id, Year\_of\_publication |
| Subcategory | Subcat\_id, subcat\_name, cat\_id |
| Category | Cat\_id, cat\_name |
| Book\_reader | Bookread\_Id, ISBN, reader\_id |

### 

03

### Relationship Cardinality:

* Author can write 1 or many book (1:M)
* Publisher publishes 1 or many book (1:M)
* Reader reads 0 or many book (0:M)
* Category has 1 or many subcategory(1:M)
* Subcategory has only 1 category (1:1)
* Subcategory has 0 or many books(0:M)
* Book has only one subcategory (1:1)

### Functional requirements:

**Search book:** search a list of books which are matching the search criteria sorted by book title, category, ISBN, publisher, author.

**View Book:** This function will map all necessary information from different table and display all the information about book like book name, author name, publisher name, etc.

**Search books by reader age**: This functionality is implemented using view to display book Read by reader of age group like list of books read by age of person 20 to 30

**Book Published by year:** This functionality is implemented using viewto Display book published between year specific year like 2000 and 2021 with Author name.

**Reader’s City wise book:** This function is implemented to Display book read by reader from specific address like Delhi city. This requirement will be implemented with view.

**Search book by author starting name:** This functional requirement is used to search any book written by author whose name starting like author name starting with G.

**Update Total Book published:** This function requirement will be performed by trigger automatically after new book is inserted; it will automatically update book published total by 1 of that particular publisher.

**Delete Subcategory:** This function requirement will be performed by trigger automatically when any category will be deleted all subcategory will be deleted also.

**Book Price Difference:** This function requirement will be performed by trigger automatically when price of the book is updated, it will print total increased in the price.

04

**Issue book:** This function will map the information about books issued or downloaded by reader. It will map book ISBN and reader id so you can also search book downloaded by reader.

**Manage author:** This function is used to insert, update, or delete any details of author’s personal information. It can also view all the author’s personal information.

**Manage publisher:** This function is basically used to insert, update, delete, update, and search or view the record of publisher. It will perform all DML operations of various column like name, email, phone no etc.

**Manage Category**: This function is basically needed to insert, update and delete category of book like Computer is one category. Search category is also performed by this function

**Manage Subcategory**: This function is basically needed to insert, update and delete subcategory of book like Networking, Internet can be subcategory of Computer Category. Search subcategory is also performed by this function.

**Manage book**: This is the most important function. It is used to manage book registration, deletion, updating of book information like book title, page, availability, subcategory type of book, author, publisher of the book.

**Registration reader**: This function is basically used to register user by entering name, address, age, phone no, email.

# 

# PHASE 2

05

## Conceptual DesignDiagram Description automatically generated

06

I have used crow-foot notation to represent the ER schema instead of Min max notation. Kindly consider the below legend for reference.



# PHASE 3

07

## Relational Design

A picture containing text

Description automatically generated

Here I have mapped the above ER schema to the relational data model. In the above diagram, underlined and bold attributes are the primary keys while FK = Foreign key.

# 

# PHASE 4

08

## Data Dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TABLE** | **Attribute** | **Data Type** | **Primary Key** | **Foreign Key** | **Constraints** |
| AUTHOR | Author\_ID | VARCHAR(6) | YES |  | NOT NULL |
| AUTHOR | Fname | VARCHAR(30) |  |  | NOT NULL |
| AUTHOR | Lname | VARCHAR(30) |  |  |  |
| AUTHOR | Email | VARCHAR(50) |  |  | NOT NULL UNIQUE |
| AUTHOR | Phoneno | BigINT |  |  |  |
| PUBLISHER | Pname | VARCHAR(25) |  |  | NOT NULL |
| PUBLISHER | Address | VARCHAR(100) |  |  |  |
| PUBLISHER | Publisher\_ID | VARCHAR(6) | YES |  | NOT NULL |
| PUBLISHER | Books\_published | INT |  |  |  |
| PUBLISHER | Phone | INT |  |  |  |
| PUBLISHER | Email | VARCHAR(50) |  |  | NOT NULL UNIQUE |
| CATEGORY | Cat\_id | VARCHAR(6) | YES |  | NOT NULL |
| CATEGORY | Cat\_name | VARCHAR(30) |  |  | NOT NULL |
| SUB\_CATEGORY | Subcat\_id | VARCHAR(6) | YES |  | NOT NULL |
| SUB\_CATEGORY | Subcat\_name | VARCHAR(30) |  |  | NOT NULL |
| SUB\_CATEGORY | Cat\_id | VARCHAR(6) |  | YES | NOT NULL |
| BOOK | ISBN | VARCHAR(10) | YES |  | NOT NULL |
| BOOK | Availability | VARCHAR(15) |  |  |  |
| BOOK | Book\_title | VARCHAR(100) |  |  | NOT NULL |
| BOOK | Price | BIGINT |  |  |  |
| BOOK | Path | VARCHAR(50) |  |  | NOT NULL |
| BOOK | Page\_count | INT |  |  | >0 |
| BOOK | Year | INT |  |  | >0 |
| BOOK | Edition | VARCHAR(25) |  |  |  |
| BOOK | Author\_ID | VARCHAR(25) |  | YES | NOT NULL |
| BOOK | Publisher\_ID | INT |  | YES | NOT NULL |
| BOOK | Subcat\_id | VARCHAR(6) |  | YES | NOT NULL |
| READER | Reader\_ID | VARCHAR(6) | YES |  | NOT NULL |
| READER | Address | VARCHAR(100) |  |  |  |
| READER | Fname | VARCHAR(30) |  |  | NOT NULL |
| READER | Lname | VARCHAR(30) |  |  |  |
| READER | Phone | INT |  |  |  |
| READER | Email | VARCHAR(50) |  |  | NOT NULL UNIQUE |
| **TABLE** | **Attribute** | **Data Type** | **Primary Key** | **Foreign Key** | **Constraints** |
| READER | Sex | CHAR |  |  |  |
| READER | Date\_of\_registration | TIME |  |  |  |
| READER | Age | INT |  |  | NOT NULL |
| Book\_Reader | br\_id | VARCHAR(10) |  |  |  |
| Book\_Reader | ISBN | VARCHAR(10) |  | YES | NOT NULL |
| Book\_Reader | Reader\_ID | VARCHAR(6) |  | YES | NOT NULL |
| Book\_Reader | Read\_date | DATE |  |  | NOT NULL |

09

# PHASE 5

10

## Implementation

This section displays all the tables as well as the queries. The tables are Author Table, Category Table, Subcategory Table, Publisher Table, Reader Table, Book reader Table and Book Table. The Queries are of different types – Regular(RQ), Nested(NQ) , Join (JQ) and Aggregate function (AFQ). All queries and tables are depicted in the format of code and output.

## Tables

### Author Table

CREATE table author

(

author\_id VARCHAR(6) NOT NULL,

fname VARCHAR(30) NOT NULL,

lname VARCHAR(30),

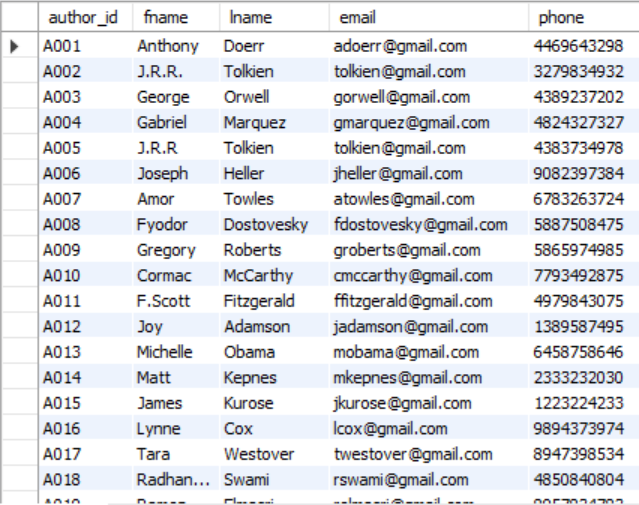
email VARCHAR(30) NOT NULL,

phone Bigint(11),

CONSTRAINT author\_pk PRIMARY KEY (author\_id)

);

Output:



### Category Table

11

CREATE table category

(

cat\_id VARCHAR(6) NOT NULL,

cat\_name VARCHAR(30) NOT NULL,

CONSTRAINT cat\_pk PRIMARY KEY (cat\_id)

);

Output:

### 

### Sub Category Table

CREATE table sub\_category

(

subcat\_id VARCHAR(6) NOT NULL,

subcat\_name VARCHAR(30) NOT NULL,

cat\_id VARCHAR(6) NOT NULL,

CONSTRAINT subcat\_pk PRIMARY KEY (subcat\_id),

FOREIGN KEY (cat\_id) REFERENCES category(cat\_id)

);

Output:

### 

### Publisher Table

CREATE table publisher

(

publisher\_id VARCHAR(6) NOT NULL,

pname VARCHAR(30),

email VARCHAR(30),

books\_published INT (10),

address VARCHAR(30),

phone BIGINT(30),

CONSTRAINT publisher\_pk PRIMARY KEY (publisher\_id)

12

);

Output

### 

### Reader Table

CREATE table reader -- reader table

(

reader\_id VARCHAR(6) NOT NULL,

fname VARCHAR(30),

lname VARCHAR(15),

age INT(3),

address VARCHAR(15),

phone VARCHAR (30),

email VARCHAR(25),

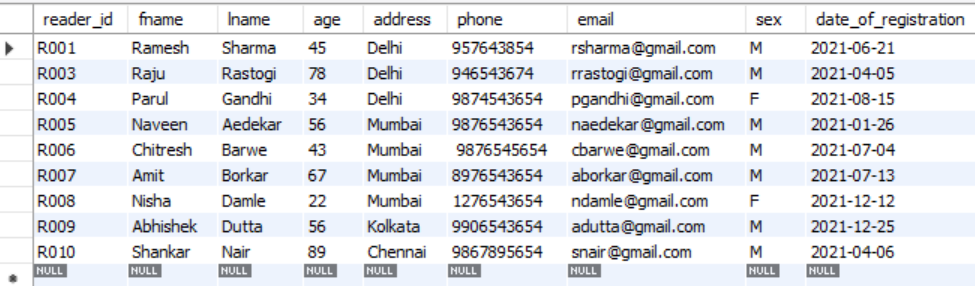
sex CHAR,

date\_of\_registration date,

CONSTRAINT reader\_pk PRIMARY KEY (reader\_id)

);

Output:



### Book Table

CREATE table book ( -- book table

ISBN VARCHAR(10) NOT NULL,

availibility VARCHAR(60),

book\_title VARCHAR(40) NOT NULL,

price INT(3) NOT NULL,

page\_count INT (5),

edition VARCHAR(10) NOT NULL,

13

subcat\_id VARCHAR (30) NOT NULL,

year\_of\_publication INT(4),

author\_id VARCHAR(10) NOT NULL,

publisher\_id VARCHAR(10) NOT NULL,

reader\_id VARCHAR (10) NOT NULL,

CONSTRAINT book\_pk PRIMARY KEY (ISBN),

CONSTRAINT author\_id FOREIGN KEY (author\_id) REFERENCES author (author\_id),

CONSTRAINT publisher\_id FOREIGN KEY (publisher\_id) REFERENCES publisher(publisher\_id),

CONSTRAINT reader\_id FOREIGN KEY (reader\_id) REFERENCES reader (reader\_id) );

Output

### 

### Book Reader Table

### CREATE table book\_reader (

### ISBN VARCHAR(10) ,

### reader\_id VARCHAR(6) ,

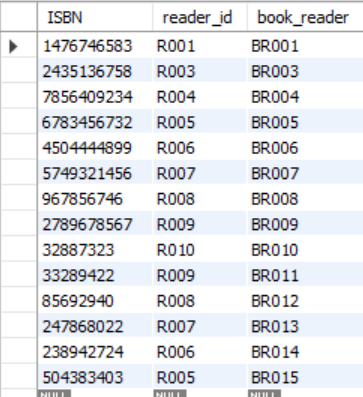
### book\_reader VARCHAR(8) NOT NULL,

### CONSTRAINT book\_reader\_pk PRIMARY KEY (book\_reader),

### FOREIGN KEY (ISBN) REFERENCES book(ISBN),

### FOREIGN KEY (reader\_id) REFERENCES reader(reader\_id));

Output



14

## Queries

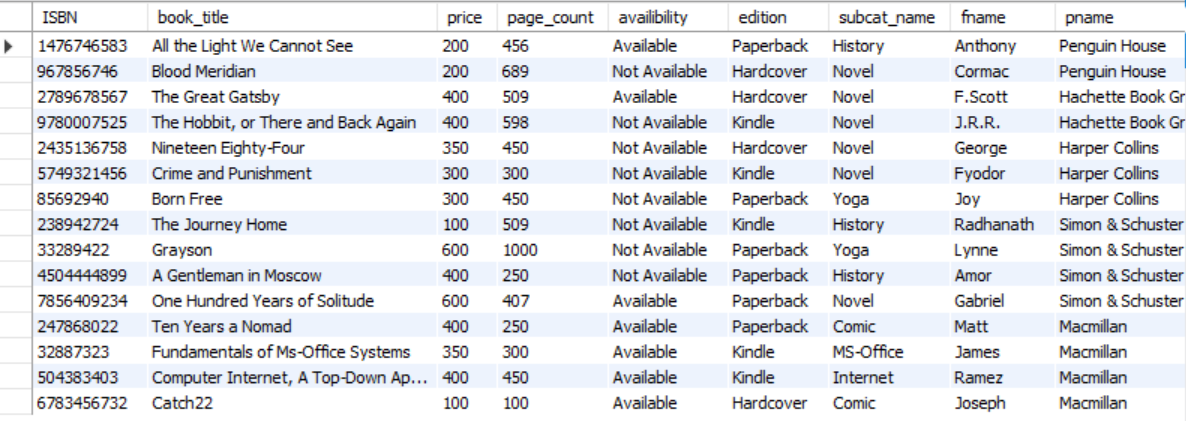
### Regular Query (RQ)

**RQ 1**

**Display the book details with Author name,Publisher name and subcategory**

SELECT ISBN,book\_title,price,page\_count,availibility,edition,s.subcat\_name,a.fname,p.pname FROM book b,sub\_category s, author a, publisher p WHERE a.author\_id=b.author\_id and p.publisher\_id=b.publisher\_id and s.subcat\_id=b.subcat\_id;

Output:

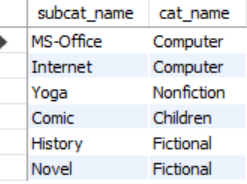


**RQ2**

**Display subcategory and category name**

SELECT subcat\_name,cat\_name FROM sub\_category s, category c WHERE s.cat\_id=c.cat\_id; -- query #2

Output:



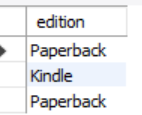
15

**RQ3**

**Find the edition type of the book such that the subcategory ID of the book is ‘S005’**.

SELECT edition FROM book WHERE subcat\_id = 'S005'

Output:



### Nested Query (NQ)

**NQ1**

**Find the edition type of the book such that the category ID of the book is ‘C005’**.

SELECT edition,book\_title

FROM book

WHERE subcat\_id IN

(

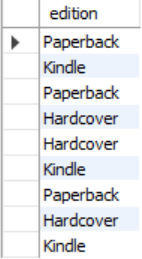
SELECT subcat\_id

FROM sub\_category

WHERE cat\_id = 'C005'

);

Output:



**NQ 2**

**Find the year when a book was published whose author’s email id is ‘tolkien@gmail.com’.**

SELECT year\_of\_publication

FROM book

16

WHERE author\_id IN(

SELECT author\_id

FROM author

WHERE email = 'tolkien@gmail.com'

);

Output:

### 

**NQ 3**

**Display the publisher details with maximum book published**.

SELECT PNAME,email,books\_published FROM `publisher` WHERE books\_published IN (SELECT MAX(books\_published)FROM publisher )

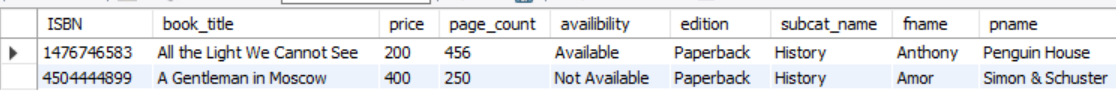
Output:



**NQ 4**

**Display All the Book with Paperback edition and from Sub category History**

SELECT ISBN,book\_title,price,page\_count,availibility,edition,s.subcat\_name,a.fname,p.pname FROM book b,sub\_category s, author a, publisher p WHERE a.author\_id=b.author\_id and p.publisher\_id=b.publisher\_id and s.subcat\_id=b.subcat\_id and edition='Paperback' and b.subcat\_id in (SELECT subcat\_id FROM sub\_category WHERE subcat\_name='History');



### Join Query (JQ)

**JQ1**

**Create a common table between author and book**

SELECT \*

FROM author

INNER JOIN book ON author.author\_id = book.author\_id;

17

Output:



**JQ2**

**Create a common table between sub\_category and book**

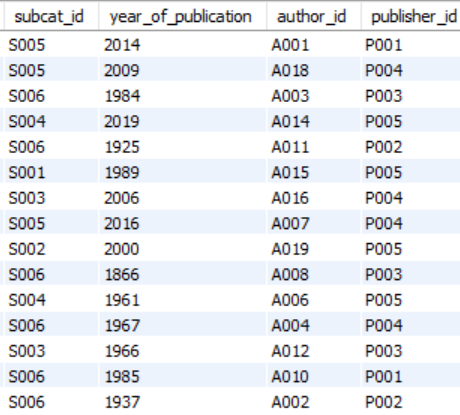
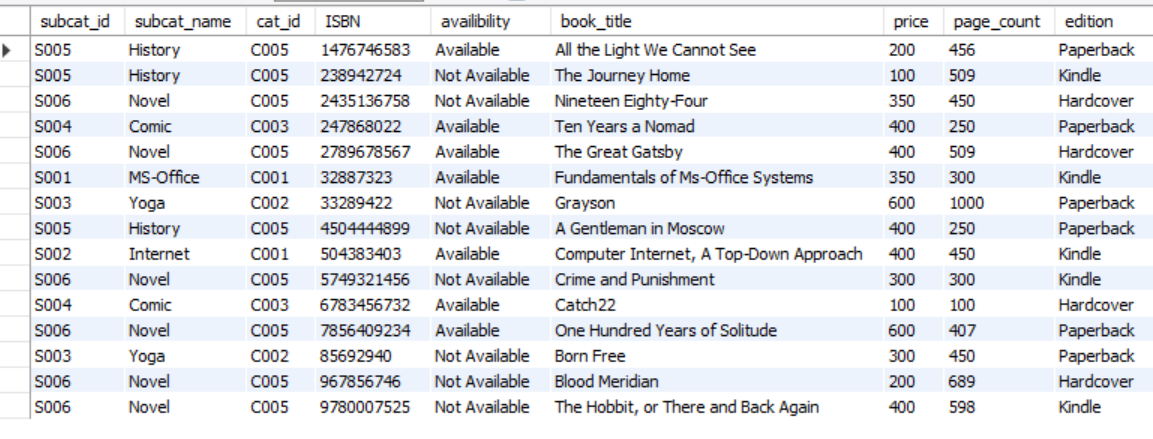
SELECT \* -- Join query #2

FROM sub\_category

RIGHT OUTER JOIN book ON sub\_category.subcat\_id = book.subcat\_id;

Output:

18



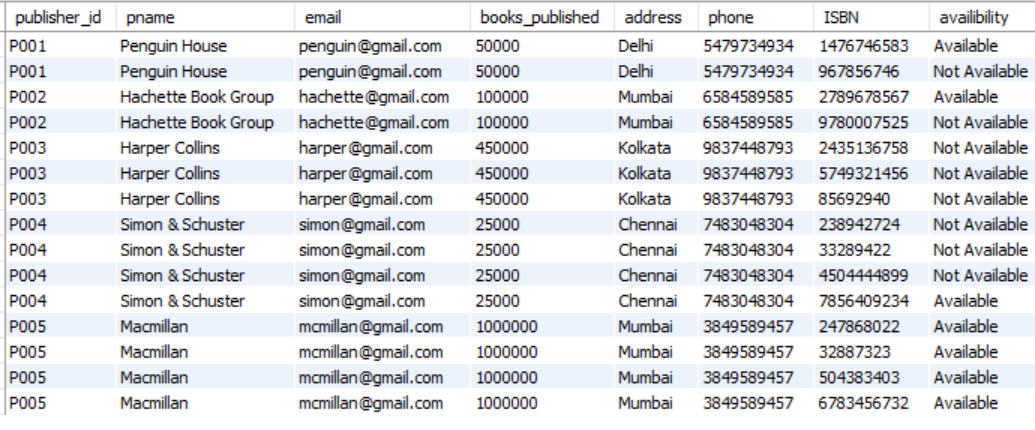
**JQ3**

**Create a common table publisher and book**

SELECT \* -- Join query #3

FROM publisher

LEFT OUTER JOIN book ON publisher.publisher\_id = book.publisher\_id;



### 

19

### Aggregate Function Query

**AFQ1**

**Find the total cost, minimum cost, maximum cost and average cost of all the books**.

SELECT SUM(price) AS Total\_Cost, -- Aggregate Functions query

MIN(price) AS Minimum\_Cost,

MAX(price) AS Maximum\_Cost,

AVG(price) AS Average\_Cost

FROM book;

Output:



### 

# PHASE 6

20

## Views: -

Following are 4 views that have been generated.

### View to Display book published between year 2000 and 2021 with Author name

**V1**

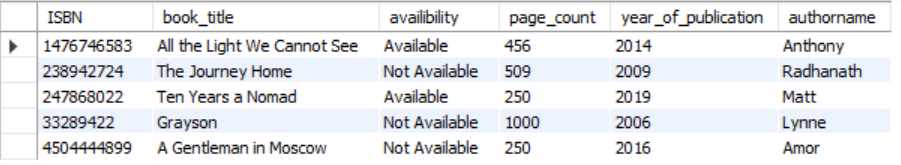
CREATE VIEW v1\_bookpub AS

SELECT ISBN,book\_title,availibility,page\_count,year\_of\_publication,fname AS authorname

FROM book b,author a WHERE year\_of\_publication<2021 and year\_of\_publication>2000 and b.author\_id in (SELECT a.author\_id FROM author); -- View #1

SELECT \* FROM v1\_bookpub;

Output:

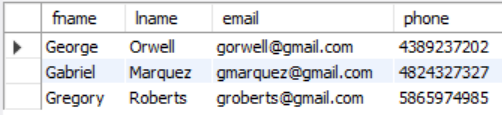


### View to Display Author name starting with G

**V2**

CREATE VIEW v2\_authordetail AS SELECT fname,lname,email,phone FROM author WHERE fname like 'G%'; -- View #2

SELECT \* FROM v2\_authordetail;Output:



### View to Display book read by reader from Delhi city

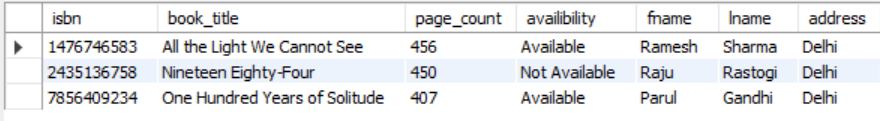
**V3**

CREATE VIEW v3\_bookreader AS SELECT b.isbn,book\_title,page\_count,availibility,r.fname,r.lname,r.address FROM book b, reader r, book\_reader br WHERE b.isbn=br.ISBN and r.reader\_id=br.reader\_id and r.reader\_id in (SELECT reader\_id FROM reader WHERE address='Delhi'); -- View #3

SELECT \* FROM v3\_bookreader;

Output:

21



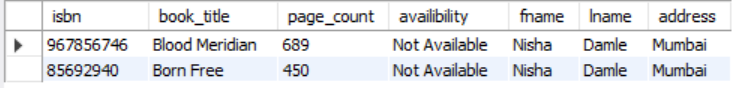
### View to Display book Read by reader of age group 20 to 30

**V4**

CREATE VIEW v4\_agebook AS SELECT b.isbn,book\_title,page\_count,availibility,r.fname,r.lname,r.address FROM book b, reader r, book\_reader br WHERE b.isbn=br.ISBN and r.reader\_id=br.reader\_id and r.reader\_id in (SELECT reader\_id FROM reader WHERE age>=20 and age<=30); -- View #4

SELECT \* FROM v4\_agebook;

Output:



## Trigger:

Following are triggers that have been generated.

### Trigger to Update total book published by one when new book is entered

**T1**

Delimiter ///

CREATE TRIGGER update\_publisher\_total

AFTER INSERT ON Book

FOR EACH ROW

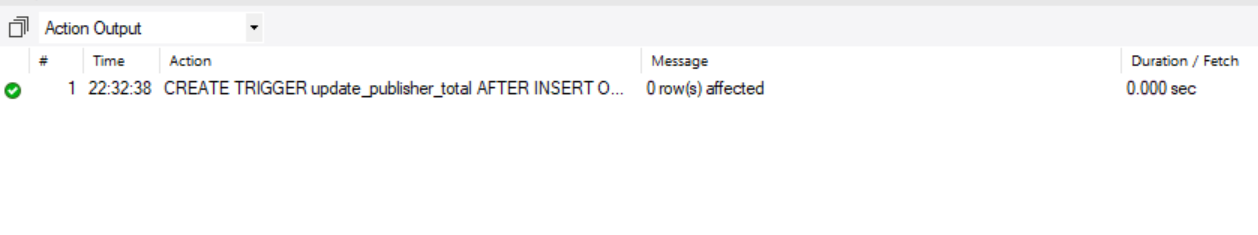
BEGIN

update publisher set books\_published=books\_published+1 where publisher\_id=new.publisher\_id

;END; -- Trigger #1

///

Output:

;END;///

22

### Trigger to delete sub category when category is deleted

**T2**

Delimiter ///

CREATE TRIGGER trig\_delsubcat

BEFORE DELETE ON category

FOR EACH ROW

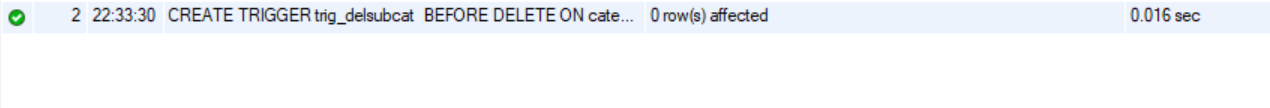
BEGIN

DELETE from sub\_category WHERE cat\_id=old.cat\_id

;END; -- Trigger #2

///

Output:



### 

# CONCLUSION

23

Thus we have created the project “Online Book Library” which serves to keep track of books in circulation and also maintain records of readers, publishers and authors.

This program can also help to generate various information and few examples are given below.

1. The demographic profile of Readers.
2. List of books read by set of age group.

3.     The preferences of Authors as envisaged by Readers

4.     Planning of addition of books based on Reader preferences

5.     Determine category of Books based on Reader preferences"

6. Different edition type of books available