Shri Ramdeobaba College of Engineering & Management, Nagpur
Electronics and Communication Engineering Department

Teachers Assessment
DATABASE MANAGEMENT SYSTEMS
ECT 355-5
B.Tech. 5th Semester EC (2023-24)

Project Report

on

Design a database for "Library Management"

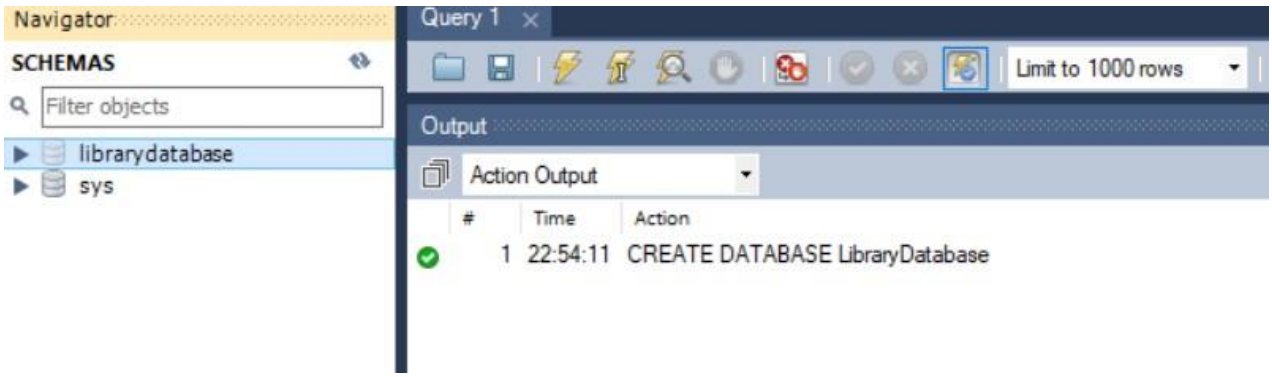
Submitted by:

Group No	Roll no	Name	Section
1	42	Narendra Dhakate	A
	76	Siddhesh Kalejwar	A
	77	Somesh Banode	A
	78	Sujal Tambe	A

Contents

1. Schema:

- Creating a database: LibraryDatabase



- Creating Tables:

Tables to be created are:

BOOK (Book_id, Title, Publisher_Name, Pub_Year)

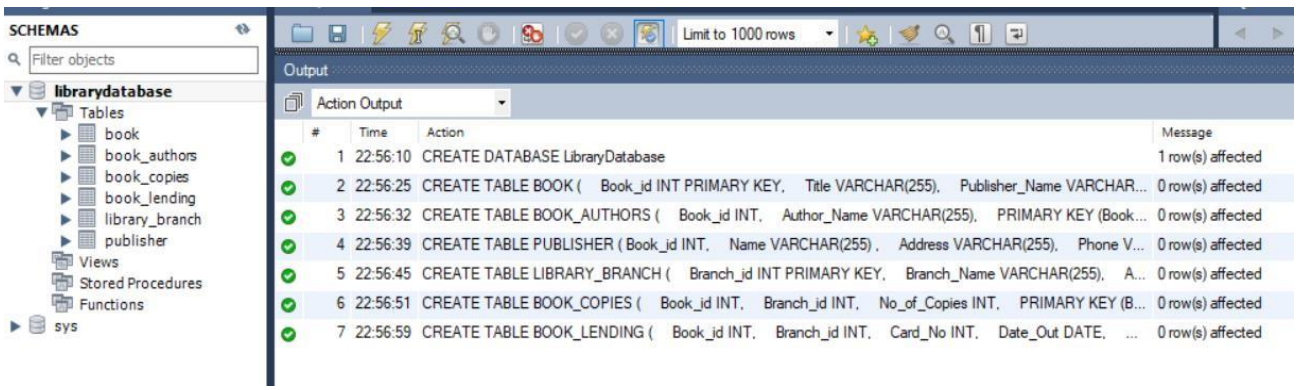
BOOK_AUTHORS (Book_id, Author_Name)

PUBLISHER (Name, Address, Phone)

BOOK_COPIES (Book_id, Branch_id, No-of_Copies)

BOOK_LENDING (Book_id, Branch_id, Card_No, Date_Out, Due_Date)

LIBRARY_BRANCH (Branch_id, Branch_Name, Address)



- **Code: --** Create BOOK table

```
CREATE TABLE BOOK (
   Book_id INT PRIMARY KEY,
   Title VARCHAR(255),
   Publisher_Name VARCHAR(255),
   Pub_Year INT
);

-- Create BOOK_AUTHORS table
CREATE TABLE BOOK_AUTHORS (
   Book_id INT,
   Author_Name VARCHAR(255),
   PRIMARY KEY (Book_id, Author_Name),
   FOREIGN KEY (Book_id) REFERENCES BOOK(Book_id)
);

-- Create PUBLISHER table
CREATE TABLE PUBLISHER (
         Book_id INT,
   Name VARCHAR(255) ,
   Address VARCHAR(255),
   Phone VARCHAR(15),
   PRIMARY KEY (Book_id, Name),
         FOREIGN KEY (Book_id) REFERENCES BOOK(Book_id)

);

-- Create LIBRARY_BRANCH table
CREATE TABLE LIBRARY_BRANCH (
   Branch_id INT PRIMARY KEY,
   Branch_Name VARCHAR(255),
   Address VARCHAR(255)
```

```
);
-- Create BOOK_COPIES table
CREATE TABLE BOOK_COPIES (
    Book_id INT,
    Branch_id INT,
    No_of_Copies INT,
    PRIMARY KEY (Book_id, Branch_id),
    FOREIGN KEY (Book_id) REFERENCES BOOK(Book_id),
    FOREIGN KEY (Branch_id) REFERENCES LIBRARY_BRANCH(Branch_id)
);

-- Create BOOK_LENDING table
CREATE TABLE BOOK_LENDING (
    Book_id INT,
    Branch_id INT,
    Card_No INT,
    Date_Out DATE,
    Due_Date DATE,
    PRIMARY KEY (Book_id, Branch_id, Card_No),
    FOREIGN KEY (Book_id) REFERENCES BOOK(Book_id),
    FOREIGN KEY (Branch_id) REFERENCES LIBRARY_BRANCH(Branch_id)
);
```

- **Adding Data to the table :**

Book_id	Title	Publisher_Name	Pub_Year
2	Database Design for Mere Mortals	Addison-Wesley	2013
3	Python Crash Course	No Starch Press	2015
4	Clean Code	Prentice Hall	2008
5	Data Science for Beginners	Packt	2020
6	SQL Performance Explained	Markus Winand	2014
7	JavaScript: The Good Parts	O'Reilly Media	2008
8	Designing Data-Intensive Applications	O'Reilly Media	2017
9	The Pragmatic Programmer	Addison-Wesley	1999
10	Eloquent JavaScript	No Starch Press	2018
11	Head First Python	O'Reilly Media	2016
12	The Clean Coder	Prentice Hall	2011
13	Learning SQL	O'Reilly Media	2009
14	Fluent Python	O'Reilly Media	2015
15	Data Mining: Concepts and Techniques	Morgan Kaufm...	2006
NULL	NULL	NULL	NULL

INSERT INTO BOOK VALUES
(1, 'The Art of SQL', 'O"Reilly Media', 2009),
(2, 'Database Design for Mere Mortals', 'Addison-Wesley', 2013),
(3, 'Python Crash Course', 'No Starch Press', 2015),
(4, 'Clean Code', 'Prentice Hall', 2008),
(5, 'Data Science for Beginners', 'Packt', 2020),
(6, 'SQL Performance Explained', 'Markus Winand', 2014),
(7, 'JavaScript: The Good Parts', 'O"Reilly Media', 2008),
(8, 'Designing Data-Intensive Applications', 'O"Reilly Media', 2017),
(9, 'The Pragmatic Programmer', 'Addison-Wesley', 1999),
(10, 'Eloquent JavaScript', 'No Starch Press', 2018),
(11, 'Head First Python', 'O"Reilly Media', 2016),
(12, 'The Clean Coder', 'Prentice Hall', 2011),
(13, 'Learning SQL', 'O"Reilly Media', 2009),
(14, 'Fluent Python', 'O"Reilly Media', 2015),
(15, 'Data Mining: Concepts and Techniques', 'Morgan Kaufmann', 2006);

INSERT INTO BOOK_AUTHORS VALUES
(1, 'Peter Robson'),
(1, 'Mary Jones'),
(2, 'Michael Hernandez'),
(3, 'Eric Matthes'),
(4, 'Robert C. Martin'),
(2, 'Emily Davis'),
(3, 'Alex Turner'),
(4, 'Sarah Miller'),
(5, 'Chris Wilson'),
(6, 'Lisa Johnson'),
(5, 'John Doe'),
(6, 'Markus Winand'),
(7, 'Douglas Crockford'),
(8, 'Martin Kleppmann'),
(9, 'Andy Hunt'),
(10, 'Marijn Haverbeke'),
(11, 'Paul Barry'),
(12, 'Robert C. Martin'),
(13, 'Alan Beaulieu'),
(14, 'Luciano Ramalho'),
(15, 'Jiawei Han');

Book_id	Author_Name
1	Mary Jones
1	Peter Robson
2	Emily Davis
2	Michael Hernandez
3	Alex Turner
3	Eric Matthes
4	Robert C. Martin
4	Sarah Miller
5	Chris Wilson
5	John Doe
6	Lisa Johnson
6	Markus Winand
7	Douglas Crockford
8	Martin Kleppmann
9	Andy Hunt
10	Marijn Haverbeke
11	Paul Barry
12	Robert C. Martin
13	Alan Beaulieu
14	Luciano Ramalho
15	Jiawei Han
NULL	NULL

INSERT INTO PUBLISHER VALUES
(1, 'O"Reilly Media', '123 Tech St', '555-1234'),
(2, 'Addison-Wesley', '456 Book Ave', '555-5678'),
(3, 'No Starch Press', '789 Code Ln', '555-9012'),
(4, 'Prentice Hall', '101 Learn Dr', '555-3456'),
(5, 'Packt', '222 Tech Blvd', '555-6789'),
(6, 'Markus Winand', '333 SQL Ln', '555-2345'),
(7, 'Morgan Kaufmann', '444 Data St', '555-7890'),
(8, 'Manning Publications', '555 Code St', '555-4321'),
(9, 'Wrox', '666 Learn Blvd', '555-8765'),
(1, 'Hachette Book Group', '454 Knowledge Ln', '555-1122'),
(5, 'HarperCollins Publishers', '565 Reading Blvd', '555-3344'),
(8, 'Simon & Schuster', '676 Literature Dr', '555-5566'),
(9, 'Random House', '787 Story Ave', '555-7788'),
(4, 'Macmillan Publishers', '898 Novel St', '555-9900'),
(10, 'Apress', '777 Tech Dr', '555-2109'),
(11, 'Microsoft Press', '888 Book Ln', '555-5432'),
(12, 'Pragmatic Bookshelf', '999 Data Ave', '555-0987'),
(13, 'Pearson', '121 SQL Blvd', '555-6543'),
(14, 'Springer', '232 Tech St', '555-9876'),
(15, 'Cengage Learning', '343 Learn Ave', '555-3210');

Book_id	Name	Address	Phone
1	Hachette Book Group	454 Knowledge Ln	555-1122
1	O'Reilly Media	123 Tech St	555-1234
2	Addison-Wesley	456 Book Ave	555-5678
3	No Starch Press	789 Code Ln	555-9012
4	Macmillan Publishers	898 Novel St	555-9900
4	Prentice Hall	101 Learn Dr	555-3456
5	HarperCollins Publishers	565 Reading Blvd	555-3344
5	Packt	222 Tech Blvd	555-6789
6	Markus Winand	333 SQL Ln	555-2345
7	Morgan Kaufmann	444 Data St	555-7890
8	Manning Publications	555 Code St	555-4321
8	Simon & Schuster	676 Literature Dr	555-5566
9	Random House	787 Story Ave	555-7788
9	Wrox	666 Learn Blvd	555-8765
10	Apress	777 Tech Dr	555-2109
11	Microsoft Press	888 Book Ln	555-5432
12	Pragmatic Bookshelf	999 Data Ave	555-0987
13	Pearson	121 SQL Blvd	555-6543
14	Springer	232 Tech St	555-9876
15	Cengage Learning	343 Learn Ave	555-3210
NULL	NULL	NULL	NULL

INSERT INTO LIBRARY_BRANCH VALUES
(1, 'Main Library', '123 Main St'),
(2, 'Branch A', '456 Branch Ave'),
(3, 'Branch B', '789 Branch Ln'),
(4, 'Branch C', '101 Branch Dr'),
(5, 'Branch D', '202 Branch Blvd'),
(6, 'Branch E', '303 Extension St'),
(7, 'Branch F', '404 Annex Ave'),
(8, 'Branch G', '505 Library Ln'),
(9, 'Branch H', '606 Reading Blvd'),
(10, 'Branch I', '707 Knowledge Dr'),
(11, 'Branch J', '808 Fiction Ave'),
(12, 'Branch K', '909 Story Ln'),
(13, 'Branch L', '010 Novel Blvd'),
(14, 'Branch M', '121 Literature Dr'),
(15, 'Branch N', '232 Academic Ave');

Branch_id	Branch_Name	Address
1	Main Library	123 Main St
2	Branch A	456 Branch Ave
3	Branch B	789 Branch Ln
4	Branch C	101 Branch Dr
5	Branch D	202 Branch Blvd
6	Branch E	303 Extension St
7	Branch F	404 Annex Ave
8	Branch G	505 Library Ln
9	Branch H	606 Reading Blvd
10	Branch I	707 Knowledge Dr
11	Branch J	808 Fiction Ave
12	Branch K	909 Story Ln
13	Branch L	010 Novel Blvd
14	Branch M	121 Literature Dr
15	Branch N	232 Academic Ave
NULL	NULL	NULL

INSERT INTO BOOK_COPIES VALUES
(1, 1, 10),
(1, 2, 5),
(2, 1, 8),
(3, 2, 12),
(4, 1, 15),
(5, 1, 20),
(6, 3, 7),
(7, 2, 10),
(8, 1, 18),
(9, 3, 5),
(10, 2, 13),
(11, 1, 9),
(12, 1, 11),
(13, 3, 6),
(14, 2, 14),
(15, 1, 22);

Book_id	Branch_id	No_of_Copies
1	1	10
1	2	5
2	1	8
3	2	12
4	1	15
5	1	20
6	3	7
7	2	10
8	1	18
9	3	5
10	2	13
11	1	9
12	1	11
13	3	6
14	2	14
15	1	22
NULL	NULL	NULL

INSERT INTO BOOK_LENDING VALUES
(1, 1, 1, '2023-01-01', '2023-02-01'),
(1, 2, 4, '2023-02-01', '2023-03-01'),
(2, 1, 3, '2023-03-01', '2023-04-01'),
(3, 2, 2, '2023-04-01', '2023-05-01'),
(4, 1, 23, '2017-01-01', '2017-05-01'),
(5, 1, 5, '2017-02-01', '2023-07-01'),
(6, 3, 1, '2017-03-01', '2023-08-01'),
(7, 2, 1, '2017-04-01', '2017-06-30'),
(8, 1, 45, '2023-09-01', '2023-10-01'),
(9, 3, 789, '2023-10-01', '2023-11-01'),
(10, 2, 7, '2017-05-01', '2017-06-27'),
(11, 1, 234, '2023-12-01', '2024-01-01'),
(12, 1, 567, '2024-01-01', '2024-02-01'),
(13, 3, 890, '2017-05-01', '2017-05-15'),
(14, 2, 123, '2024-03-01', '2024-04-01'),
(15, 1, 456, '2024-03-01', '2024-04-01');

Book_id	Branch_id	Card_No	Date_Out	Due_Date
1	1	1	2023-01-01	2023-02-01
1	2	4	2023-02-01	2023-03-01
2	1	3	2023-03-01	2023-04-01
3	2	2	2023-04-01	2023-05-01
4	1	23	2017-01-01	2017-05-01
5	1	5	2017-02-01	2023-07-01
6	3	1	2017-03-01	2023-08-01
7	2	1	2017-04-01	2017-06-30
8	1	45	2023-09-01	2023-10-01
9	3	789	2023-10-01	2023-11-01
10	2	7	2017-05-01	2017-06-27
11	1	234	2023-12-01	2024-01-01
12	1	567	2024-01-01	2024-02-01
13	3	890	2017-05-01	2017-05-15
14	2	123	2024-03-01	2024-04-01
15	1	456	2024-03-01	2024-04-01
NULL	NULL	NULL	NULL	NULL

Answers to SQL Queries:

-- 1. Retrieve details of all books in the library:

```
SELECT B.Book_id, B.Title, B.Publisher_Name, BA.Author_Name, BC.No_of_Copies,
LB.Branch_Name
FROM BOOK B
JOIN BOOK_AUTHORS BA ON B.Book_id = BA.Book_id
JOIN BOOK_COPIES BC ON B.Book_id = BC.Book_id
JOIN LIBRARY_BRANCH LB ON BC.Branch_id = LB.Branch_id;
```

Book_id	Title	Publisher_Name	Author_Name	No_of_Copies	Branch_Name
1	The Art of SQL	O'Reilly Media	Mary Jones	10	Main Library
1	The Art of SQL	O'Reilly Media	Peter Robson	10	Main Library
1	The Art of SQL	O'Reilly Media	Mary Jones	5	Branch A
1	The Art of SQL	O'Reilly Media	Peter Robson	5	Branch A
2	Database Design for Mere Mortals	Addison-Wesley	Emily Davis	8	Main Library
2	Database Design for Mere Mortals	Addison-Wesley	Michael Hernandez	8	Main Library
3	Python Crash Course	No Starch Press	Alex Turner	12	Branch A
3	Python Crash Course	No Starch Press	Eric Matthes	12	Branch A
4	Clean Code	Prentice Hall	Robert C. Martin	15	Main Library
4	Clean Code	Prentice Hall	Sarah Miller	15	Main Library
5	Data Science for Beginners	Packt	Chris Wilson	20	Main Library
5	Data Science for Beginners	Packt	John Doe	20	Main Library
6	SQL Performance Explained	Markus Winand	Lisa Johnson	7	Branch B
6	SQL Performance Explained	Markus Winand	Markus Winand	7	Branch B
7	JavaScript: The Good Parts	O'Reilly Media	Douglas Crockford	10	Branch A
8	Designing Data-Intensive Applica...	O'Reilly Media	Martin Kleppmann	18	Main Library
9	The Pragmatic Programmer	Addison-Wesley	Andy Hunt	5	Branch B
10	Eloquent JavaScript	No Starch Press	Marijn Haverbeke	13	Branch A
11	Head First Python	O'Reilly Media	Paul Barry	9	Main Library
12	The Clean Coder	Prentice Hall	Robert C. Martin	11	Main Library
13	Learning SQL	O'Reilly Media	Alan Beaulieu	6	Branch B
14	Fluent Python	O'Reilly Media	Luciano Ramalho	14	Branch A
15	Data Mining: Concepts and Tech...	Morgan Kaufm...	Jiawei Han	22	Main Library

-- 2. Get the particulars of borrowers who have borrowed more than 3 books from Jan 2017 to Jun 2017:

```
SELECT Card_No, COUNT(*) AS Books_Borrowed
FROM BOOK_LENDING
WHERE Date_Out BETWEEN '2017-01-01' AND '2017-06-30'
GROUP BY Card_No
HAVING COUNT(*) > 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Card_No	Books_Borrowed
1	4

-- 3. Delete a book in the BOOK table and update related tables:

```
-- Delete a book with Book_id = 5
DELETE FROM BOOK WHERE Book_id = 5;
select * from book;
-- Update the title of the book with Book_id = 5
UPDATE BOOK SET Title = 'New Title' WHERE Book_id = 3;
select * from book;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Book_id	Title	Publisher_Name	Pub_Year
1	The Art of SQL	O'Reilly Media	2009
2	Database Design for Mere Mortals	Addison-Wesley	2013
3	Python Crash Course	No Starch Press	2015
4	Clean Code	Prentice Hall	2008
6	SQL Performance Explained	Markus Winand	2014
7	JavaScript: The Good Parts	O'Reilly Media	2008
8	Designing Data-Intensive Applications	O'Reilly Media	2017
9	The Pragmatic Programmer	Addison-Wesley	1999
10	Eloquent JavaScript	No Starch Press	2018
11	Head First Python	O'Reilly Media	2016

Result Grid | Filter Rows: | Edit: | Export/Import:

Book_id	Title	Publisher_Name	Pub_Year
1	The Art of SQL	O'Reilly Media	2009
2	Database Design for Mere Mortals	Addison-Wesley	2013
3	New Title	No Starch Press	2015
4	Clean Code	Prentice Hall	2008
6	SQL Performance Explained	Markus Winand	2014
7	JavaScript: The Good Parts	O'Reilly Media	2008
8	Designing Data-Intensive Applications	O'Reilly Media	2017
9	The Pragmatic Programmer	Addison-Wesley	1999
10	Eloquent JavaScript	No Starch Press	2018
11	Head First Python	O'Reilly Media	2016

-- 4.Partition the BOOK table based on the year of publication:

```
-- Retrieve books published in a specific year :
SELECT *
FROM BOOK
WHERE Pub_Year = 2008;
-- Assuming  to create a new table :
CREATE TABLE BOOK_BTW_99AND09 AS SELECT * FROM BOOK WHERE Pub_Year
BETWEEN 1999 AND  2009;
-- Create a view partitioning the BOOK table based on the year of publication
CREATE VIEW BOOK_PARTITION_BY_YEAR AS
SELECT *
FROM BOOK
ORDER BY Pub_Year;
select * from BOOK_PARTITION_BY_YEAR;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

Book_id	Title	Publisher_Name	Pub_Year
4	Clean Code	Prentice Hall	2008
7	JavaScript: The Good Parts	O'Reilly Media	2008
NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Book_id	Title	Publisher_Name	Pub_Year
1	The [illegible]	[illegible]	2009
4	Clear[illegible]	[illegible]	2008
7	JavaScript: The Good Parts	O'Reilly Media	2008
9	The Pragmatic Programmer	Addison-Wesley	1999
13	Learning SQL	O'Reilly Media	2009
15	Data Mining: Concepts and Techniques	Morgan Kaufmann	2006

Refresh data re-executing the original query

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Book_id	Title	No_of_Copies
1	The Art of SQL	10
1	The Art of SQL	5
2	Database Design for Mere Mortals	8
3	New Title	12
4	Clean Code	15
6	SQL Performance Explained	7
7	JavaScript: The Good Parts	10
8	Designing Data-Intensive Applications	18
9	The Pragmatic Programmer	5
10	Eloquent JavaScript	13
11	Head First Python	9
12	The Clean Coder	11
13	Learning SQL	6
14	Fluent Python	14
15	Data Mining: Concepts and Techniques	22

-- 5.Create a view of all books and their number of copies currently available in the Library:

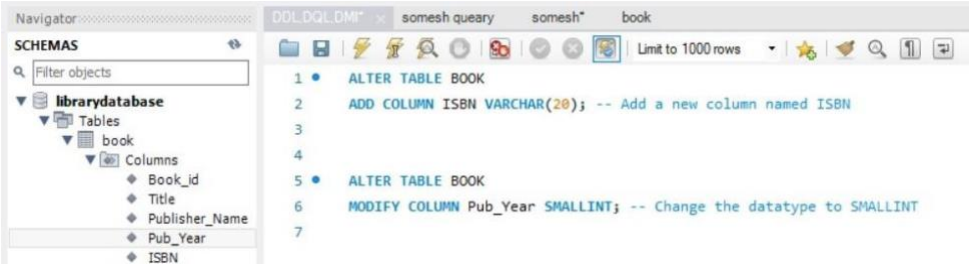
```
CREATE VIEW AVAILABLE_BOOKS AS
SELECT B.Book_id, B.Title, BC.No_of_Copies
FROM BOOK B
JOIN BOOK_COPIES BC ON B.Book_id = BC.Book_id;
```

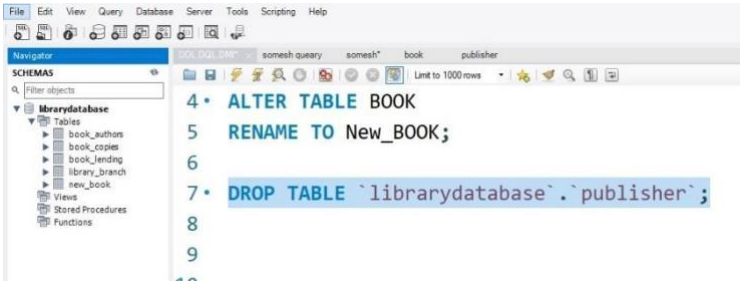
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

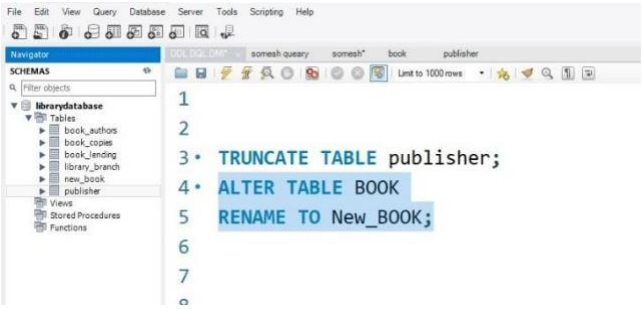
Book_id	Title	Publisher_Name	Pub_Year
9	The Pragmatic Programmer	Addison-Wesley	1999
15	Data Mining: Concepts and Techniques	Morgan Kaufmann	2006
4	Clean Code	Prentice Hall	2008
7	JavaScript: The Good Parts	O'Reilly Media	2008
1	The Art of SQL	O'Reilly Media	2009
13	Learning SQL	O'Reilly Media	2009
12	The Clean Coder	Prentice Hall	2011
2	Database Design for Mere Mortals	Addison-Wesley	2013
6	SQL Performance Explained	Markus Winand	2014
3	New Title	No Starch Press	2015
14	Fluent Python	O'Reilly Media	2015
11	Head First Python	O'Reilly Media	2016
8	Designing Data-Intensive Applications	O'Reilly Media	2017
10	Eloquent JavaScript	No Starch Press	2018

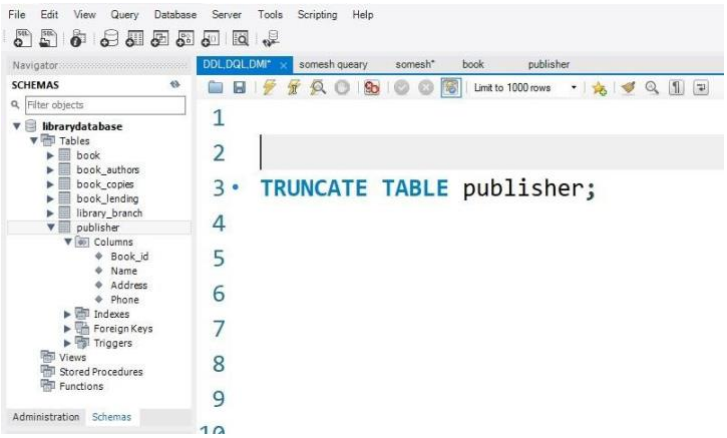
2014

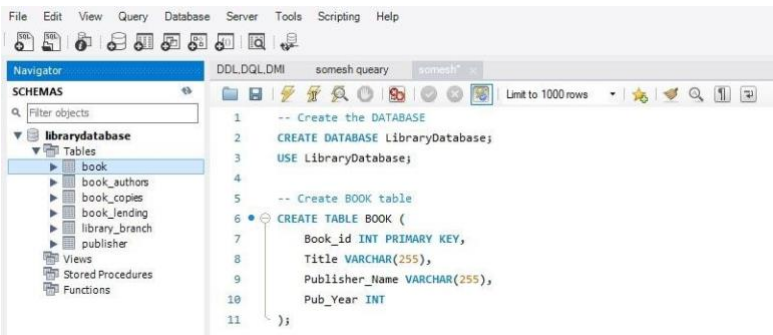
2. Implementing DDL commands in SQL:



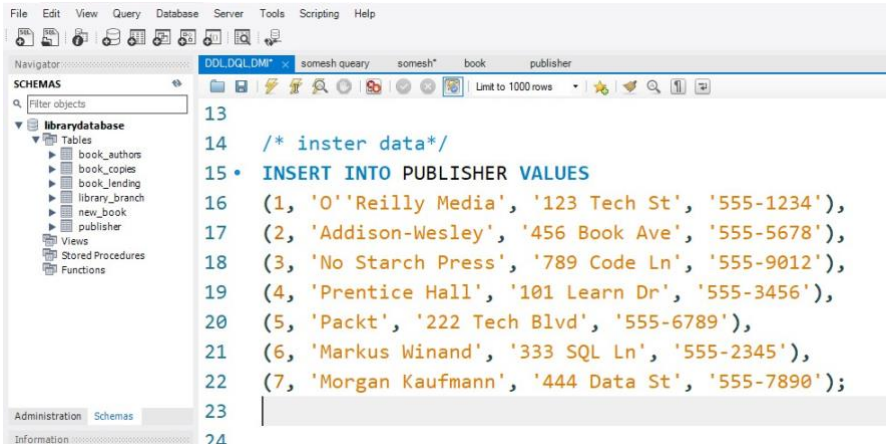


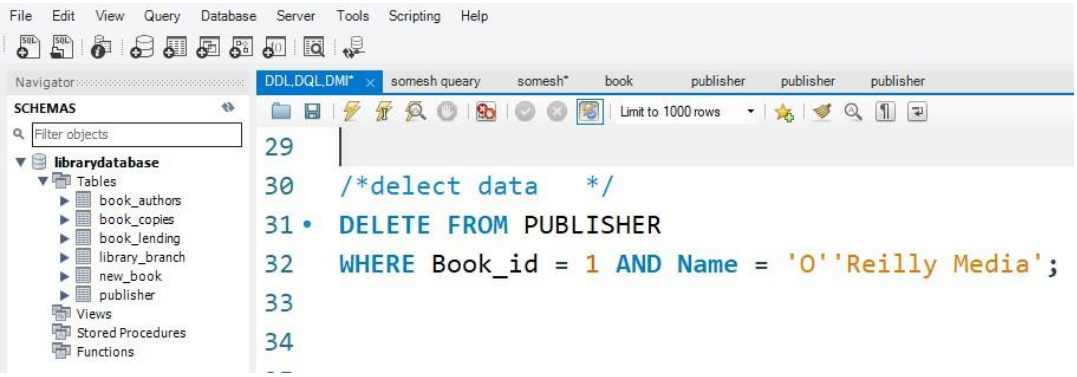


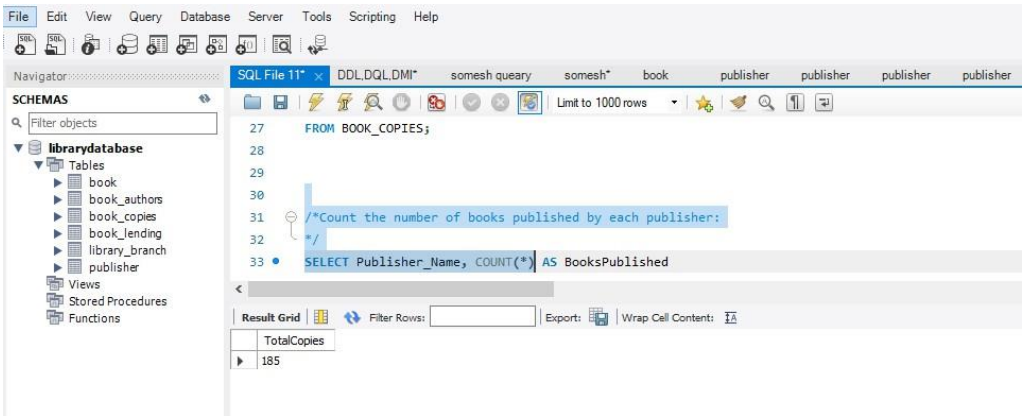


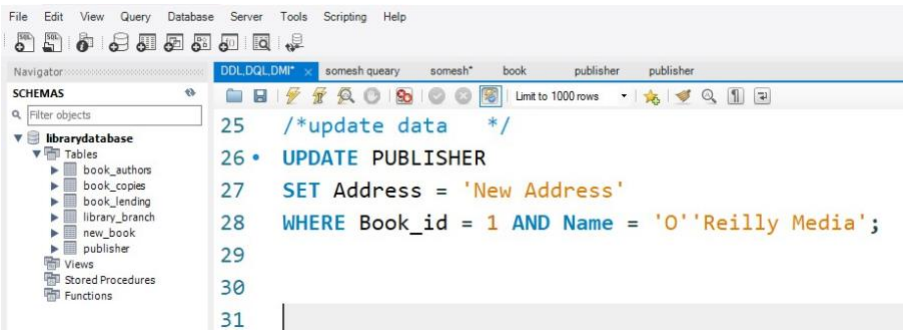


3. Implementing DML commands in SQL:









4. Implementing DQL commands in SQL:

1. Retrieve all columns from the BOOK table:

```
SELECT *
FROM BOOK;
```

2. Retrieve specific columns from the BOOK table:

```
SELECT Book_id, Title, Publisher_Name
FROM BOOK;
```

3. Filter books published by a specific publisher:

```
SELECT *
FROM BOOK
WHERE Publisher_Name = 'O''Reilly Media';
```

4. Retrieve distinct publisher names from the BOOK table:

```
SELECT DISTINCT Publisher_Name
FROM BOOK;
```

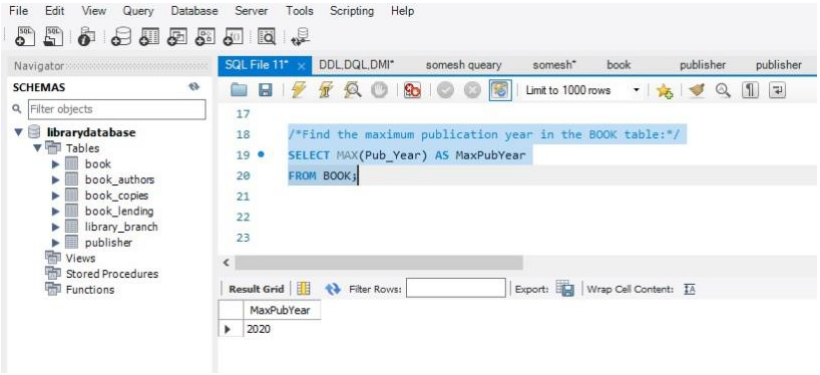
5. Order books by publication year in descending order:

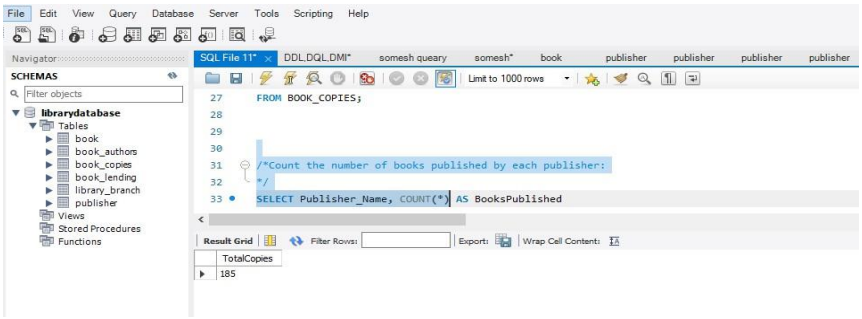
```
SELECT *
FROM BOOK
ORDER BY Pub_Year DESC;
```

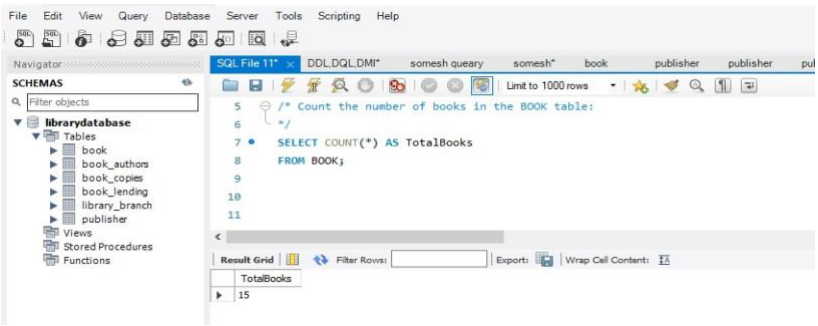
6. Count the number of books published each year:

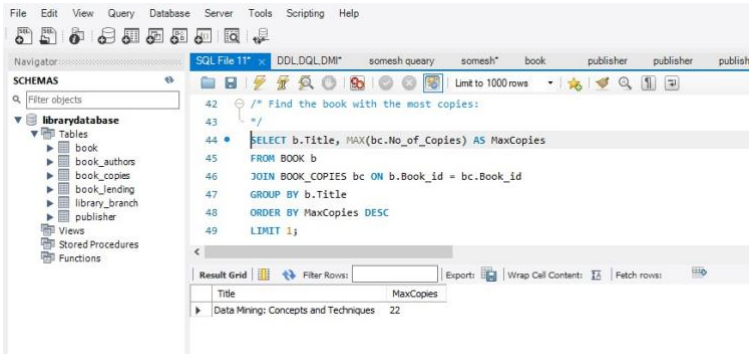
```
SELECT Pub_Year, COUNT(*) AS BooksPublished
FROM BOOK
GROUP BY Pub_Year;
```

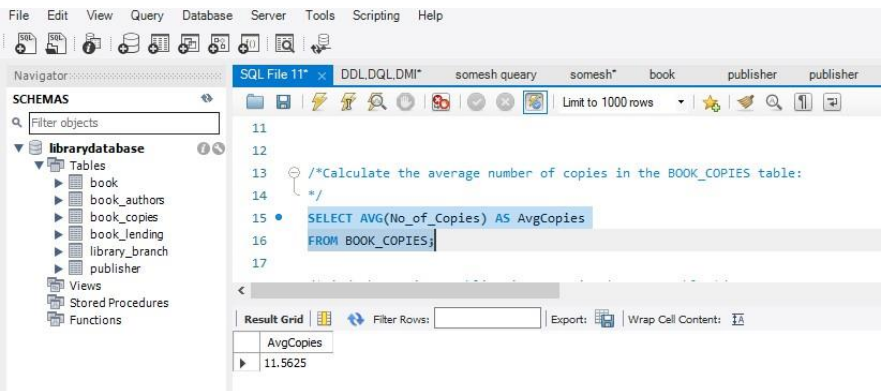
5. Implementing AGGEREATION commands in SQL:



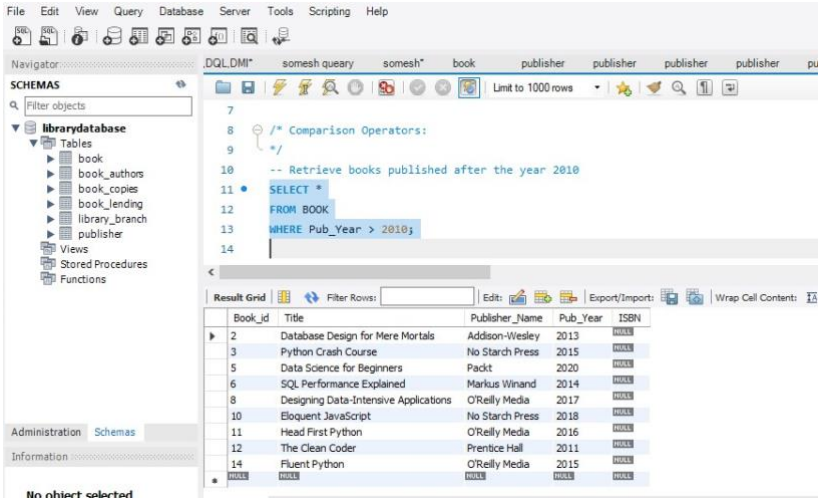


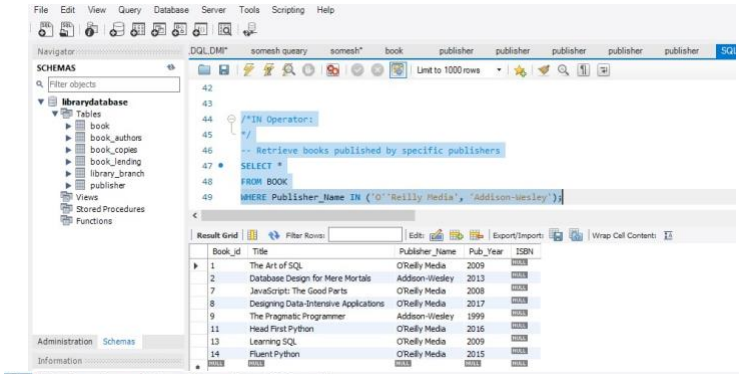


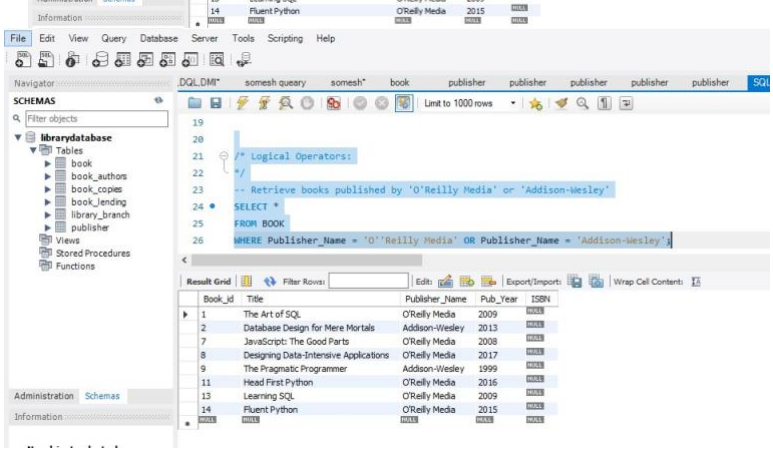


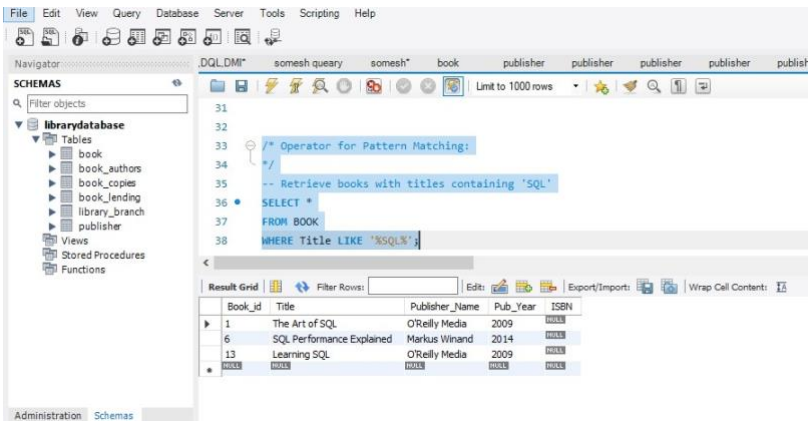


6. Implementing various types of operation in SQL:









7. Implement various types of joins:

```
-- 1. Inner Join
SELECT *
FROM NEW_BOOK
INNER JOIN DETAILS ON NEW_BOOK.Book_id = DETAILS.Branch_id;


-- 2.Left Join (or Left Outer Join)
SELECT *
FROM NEW_BOOK
LEFT JOIN DETAILS ON NEW_BOOK.Book_id = DETAILS.Branch_id;

-- 3.Right Join (or Right Outer Join)
SELECT *
FROM NEW_BOOK
RIGHT JOIN DETAILS ON NEW_BOOK.Book_id = DETAILS.Branch_id;

-- 4.Full Join (or Full Outer Join)
SELECT *
FROM NEW_BOOK
LEFT JOIN DETAILS ON NEW_BOOK.Book_id = DETAILS.Branch_id
UNION
SELECT *
FROM NEW_BOOK
RIGHT JOIN DETAILS ON NEW_BOOK.Book_id = DETAILS.Branch_id;

-- 5.Natural Join
SELECT *
FROM NEW_BOOK
NATURAL JOIN DETAILS;
```