

## **DBMS -LAB RECORD**

**NAME: SAQUIB NAUSHAD**

**USN: 1BM19CS144**

**DEPT: CSE**

**SEM: 4**

**SECTION: C**

**COURSE NAME: DATABASE MANAGEMENT SYSTEMS**

**LAB\_BATCH:C-3**

## **PROGRAM-6**

### **PROGRAM 6. ORDER PROCESSING DATABASE**

Consider the following relations for an Order Processing database application in a company.

**CUSTOMER** (CUST #: int, cname: String, city: String)

**ORDER** (order #: int, odate: date, cust #: int, ord-Amt: int)

**ITEM** (item #: int, unit-price: int)

**ORDER-ITEM** (order #: int, item #: int, qty: int)

**WAREHOUSE** (warehouse #: int, city: String)

**SHIPMENT** (order #: int, warehouse #: int, ship-date: date)

i. Create the above tables by properly specifying the primary keys and the foreign keys and the foreign keys.

ii. Enter at least five tuples for each relation.

iii. Produce a listing: CUSTNAME, #oforders, AVG\_ORDER\_AMT, where the middle column is the total

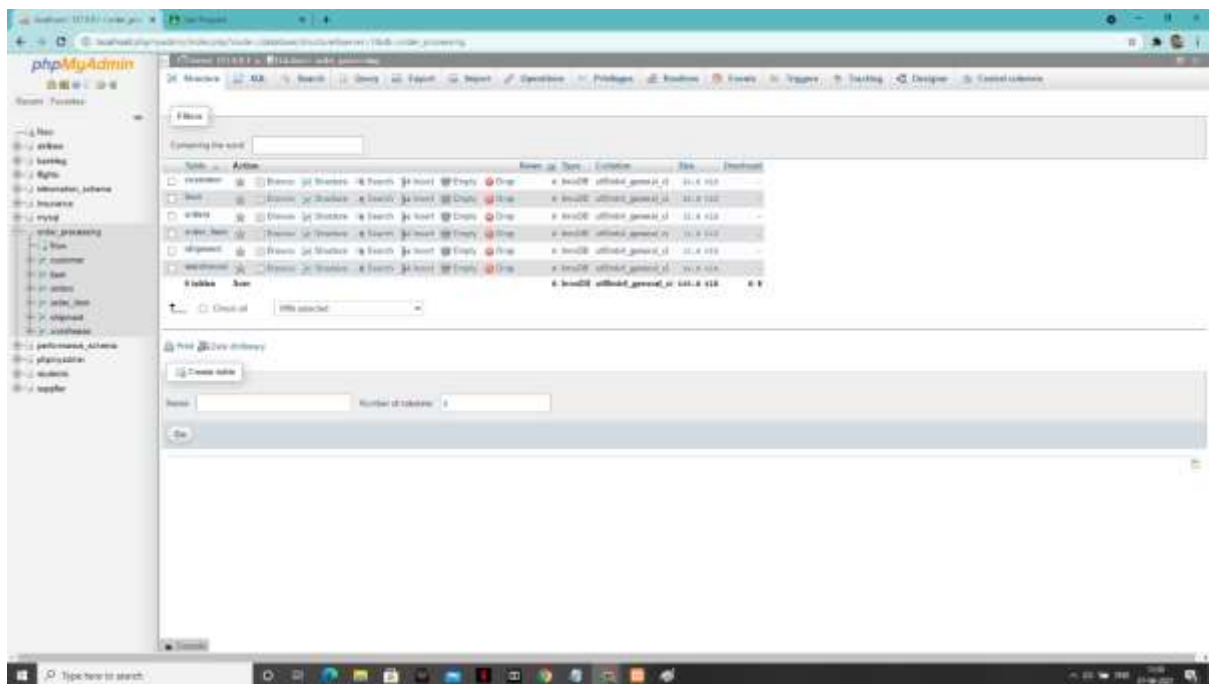
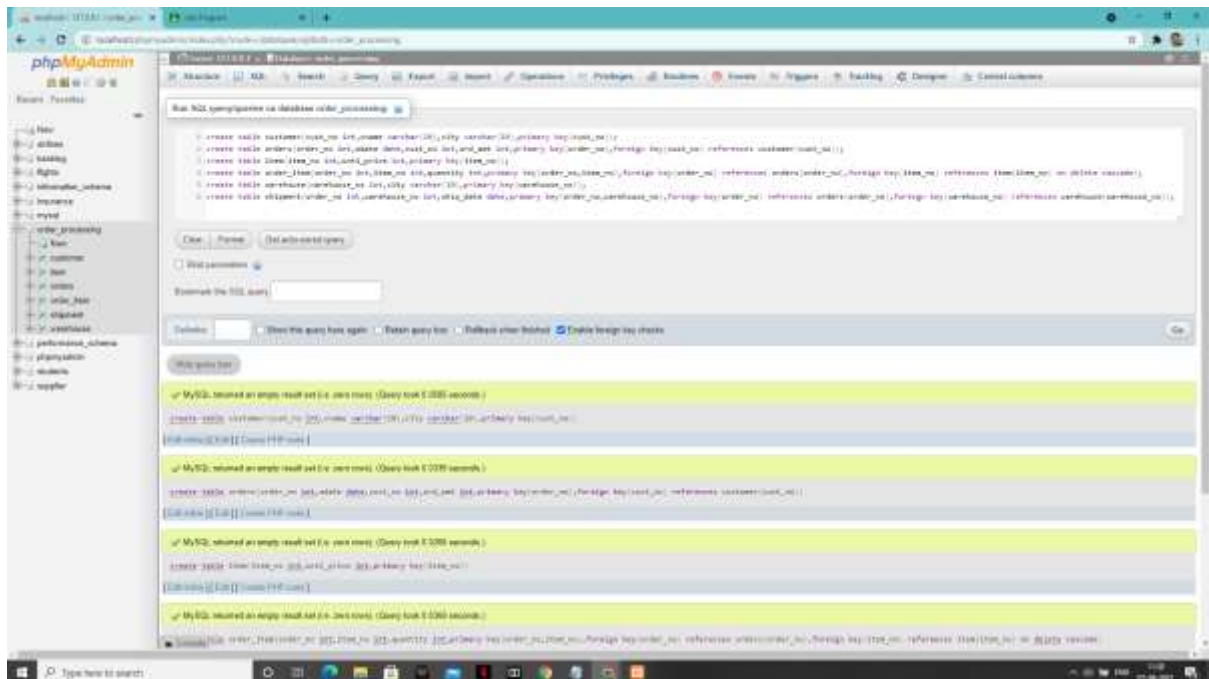
numbers of orders by the customer and the last column is the average order amount for that customer.

iv. List the order# for orders that were shipped from all warehouses that the company has in a specific city.

v. Demonstrate how you delete item# 10 from the ITEM table and make that field null in the ORDER\_ITEM

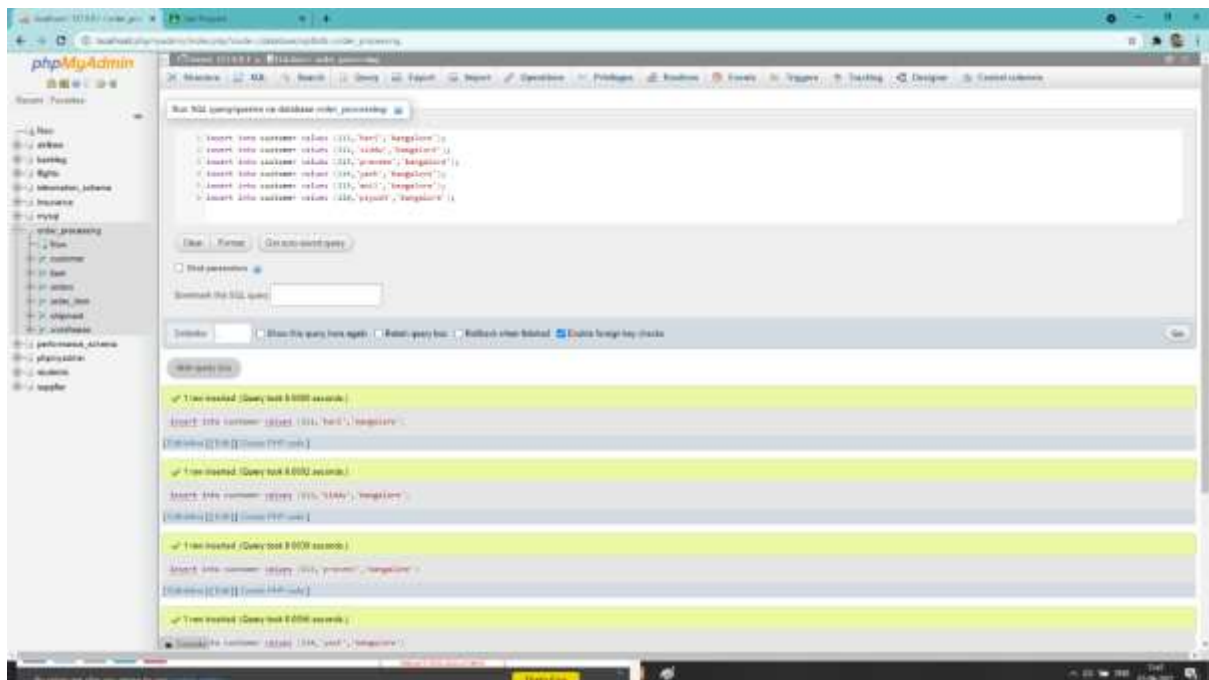
table.

## Create table:-

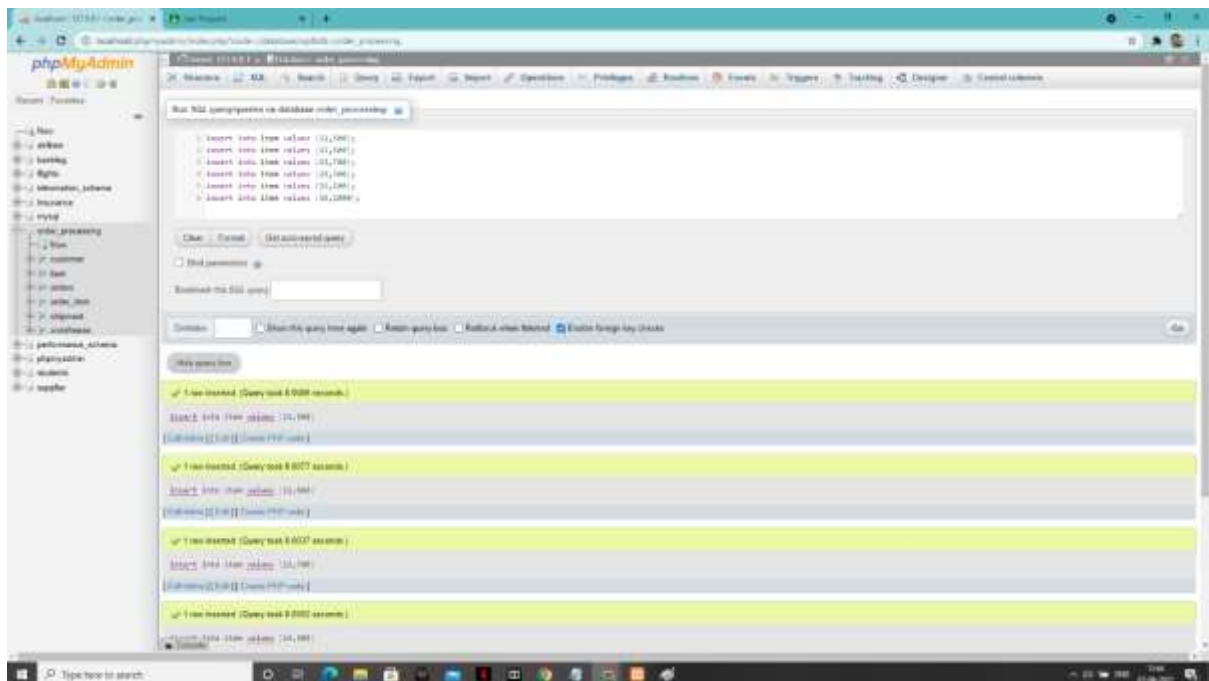


## 2) Enter tuples for each relation.

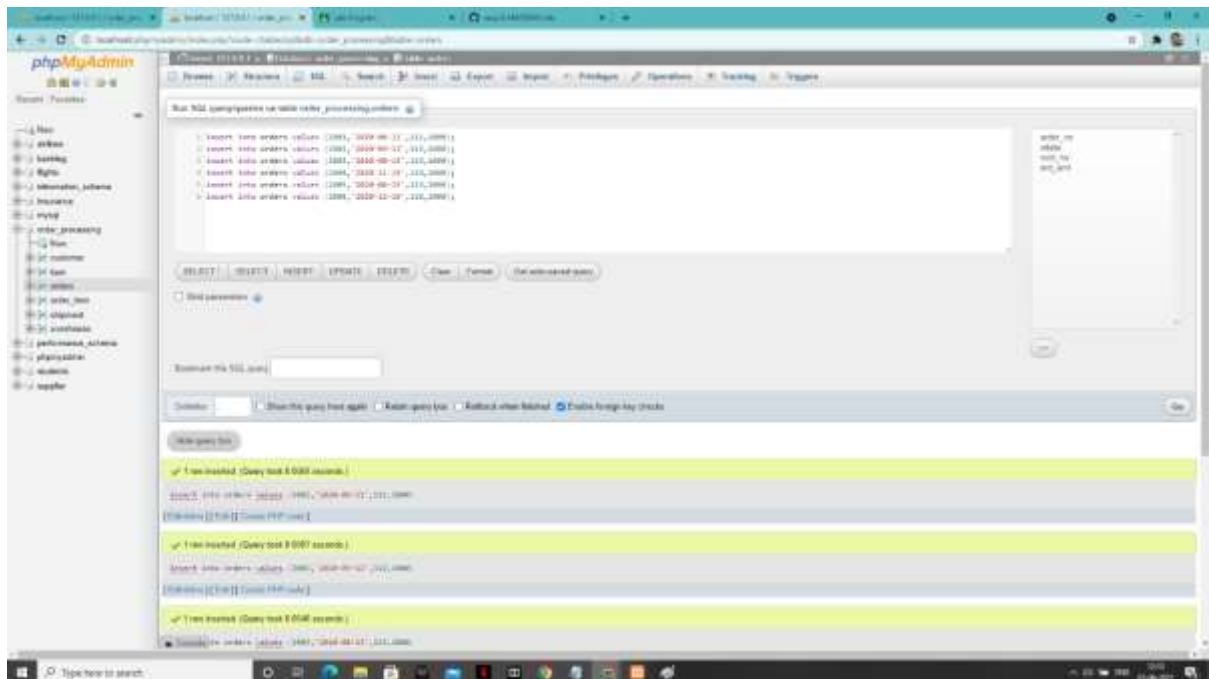
‘Customer’ table:



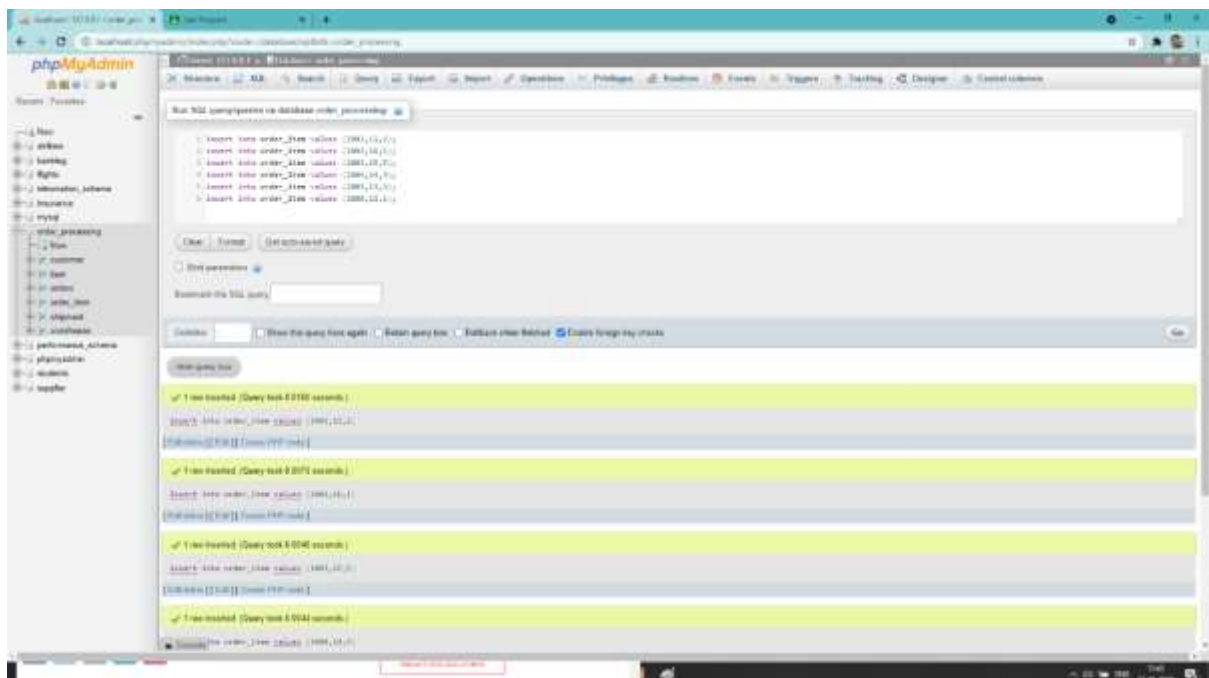
Item' table:



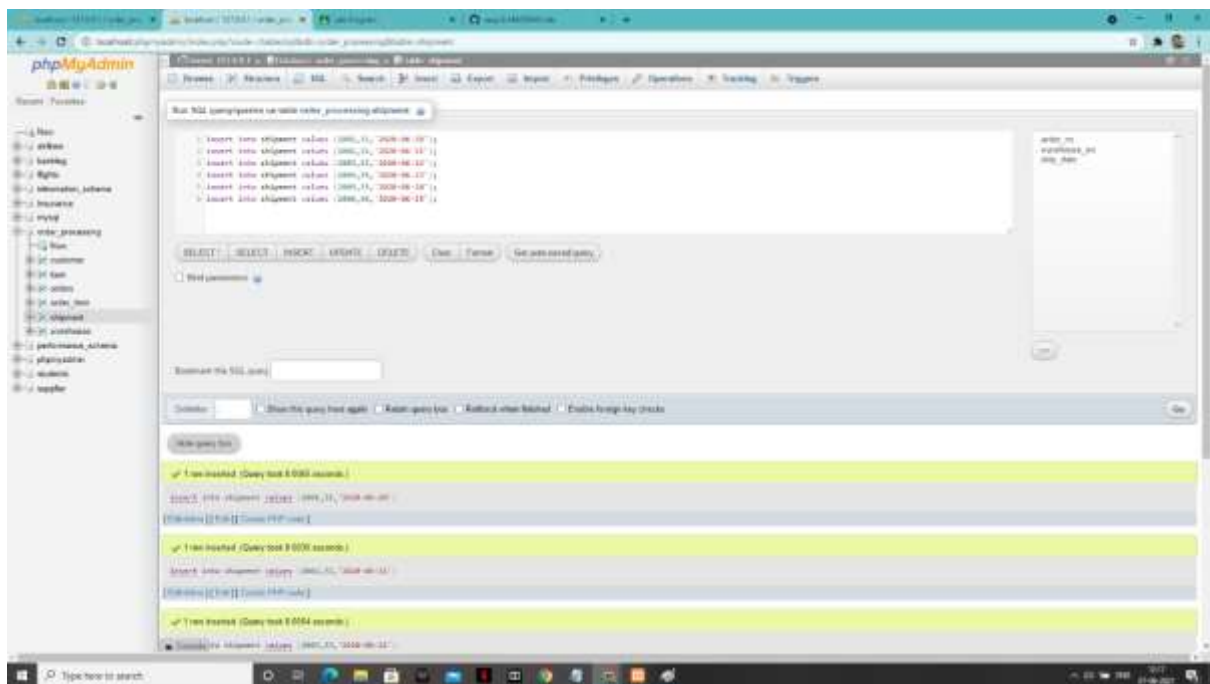
## ‘Order’ table:



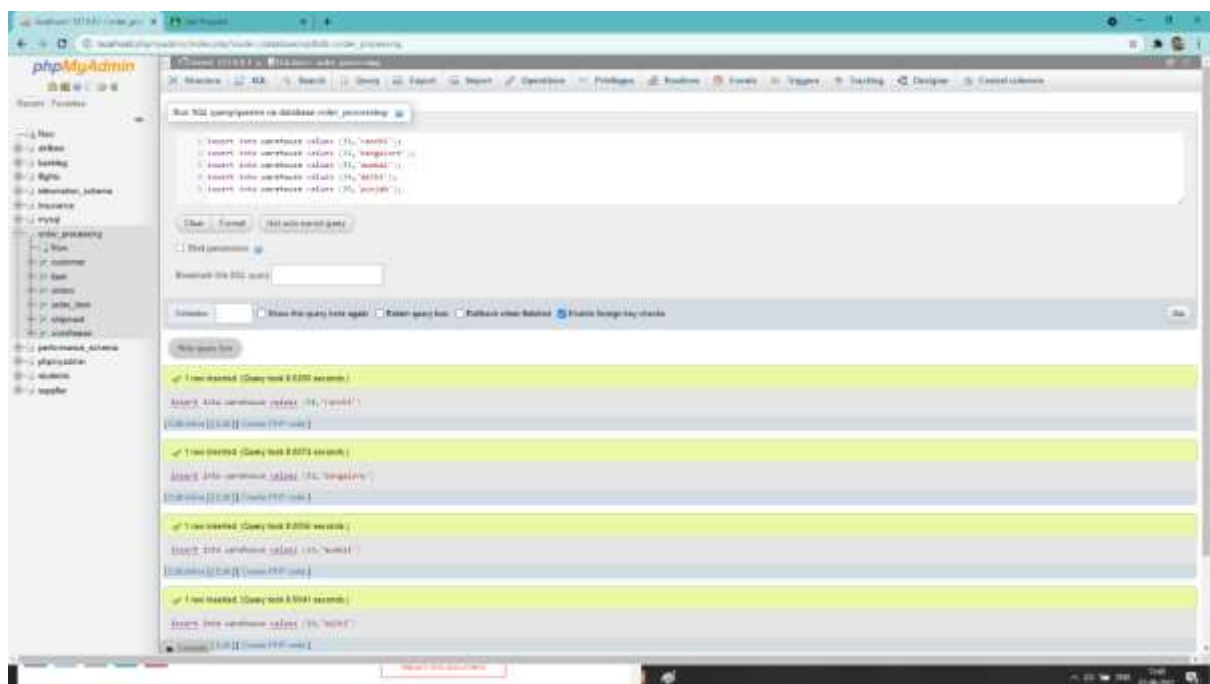
## Order\_item value: -



## Shipment table:

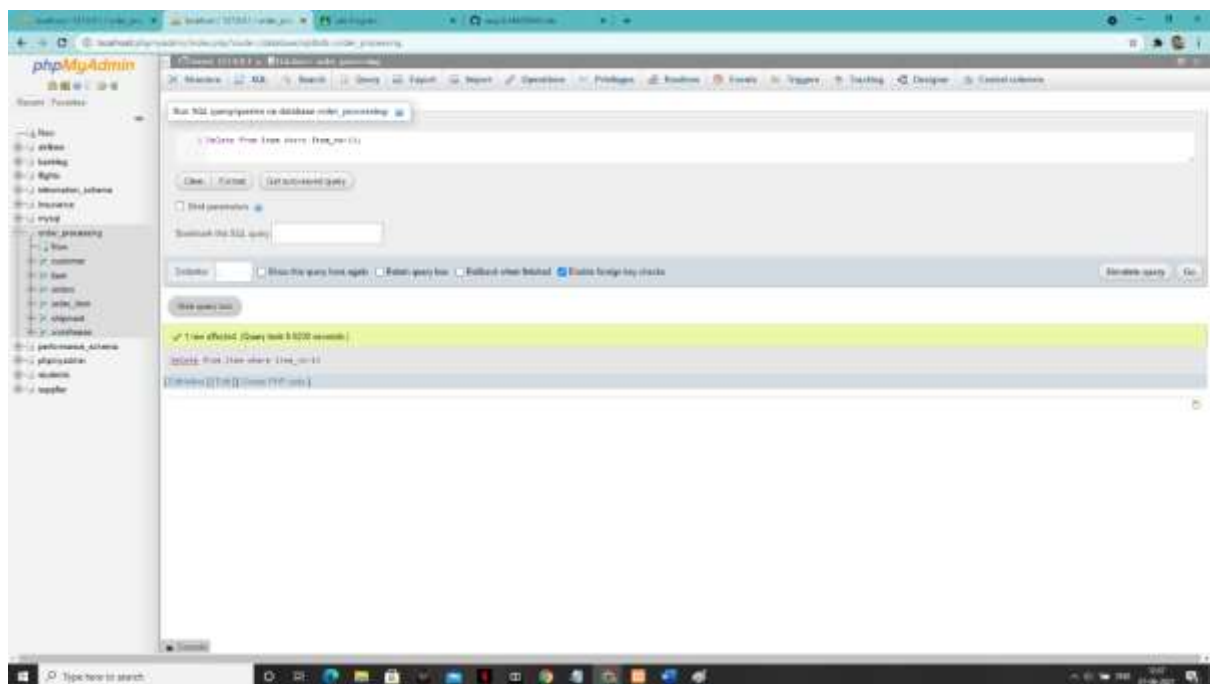


## Warehouse table:-



[illegible]

3:-





## **PROGRAM -7**

### **PROGRAM 7 :- BOOK DEALER DATABASE**

**The following tables are maintained by a book dealer:**

**AUTHOR(author-id: int, name: String, city: String, country: String)**

**PUBLISHER(publisher-id: int, name: String, city: String, country: String)**

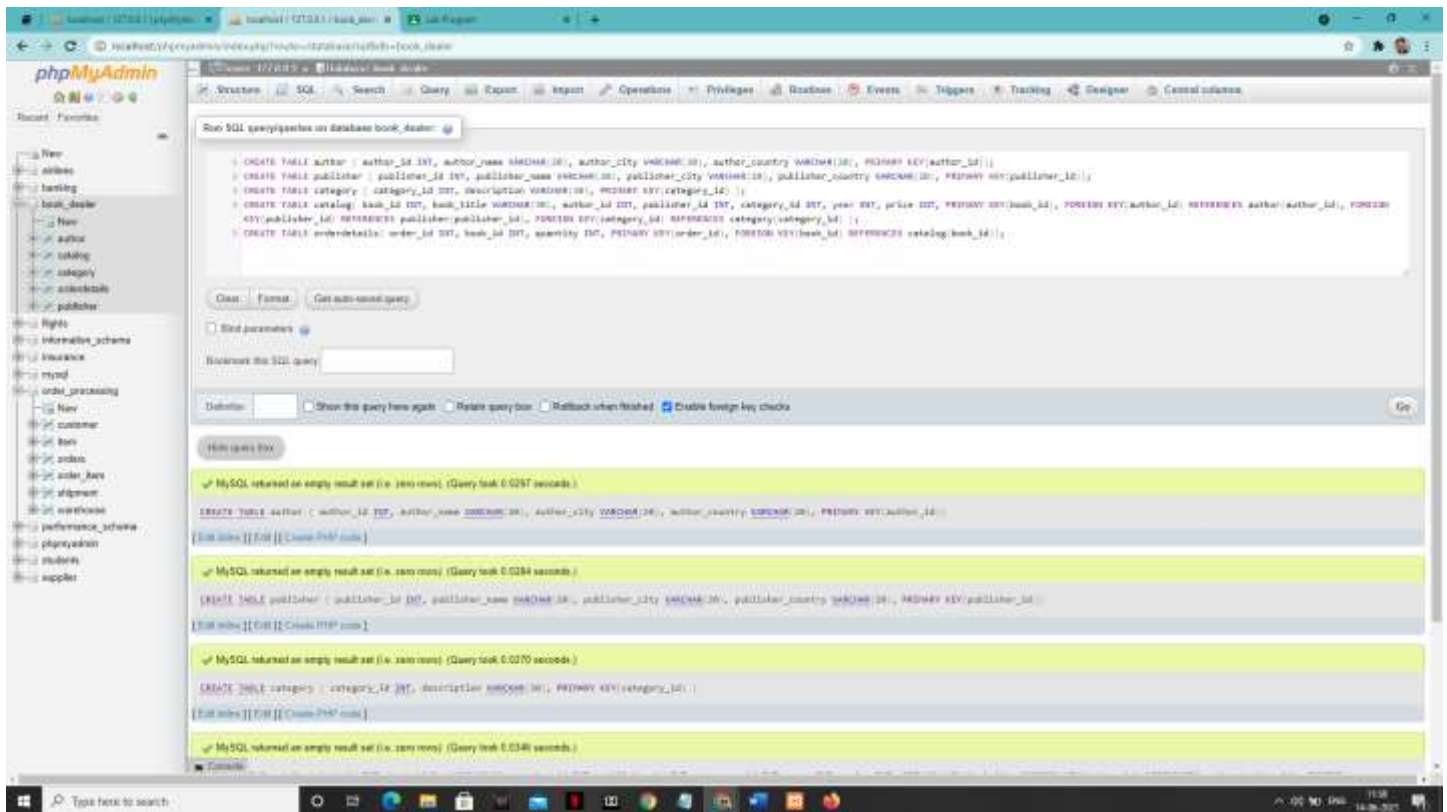
**CATALOG (book-id: int, title: String, author-id: int, publisher-id: int, category-id: int, year: int, price: int)**

**CATEGORY(category-id: int, description: String)**

**ORDER-DETAILS(order-no: int, book-id: int, quantity: int)**

- i. Create the above tables by properly specifying the primary keys and the foreign keys.**
- ii. Enter at least five tuples for each relation.**
- iii. Give the details of the authors who have 2 or more books in the catalog and the price of the books in the catalog and the year of publication is after 2000.**
- iv. Find the author of the book which has maximum sales.**
- v. Demonstrate how you increase the price of books published by a specific publisher by 10%..**

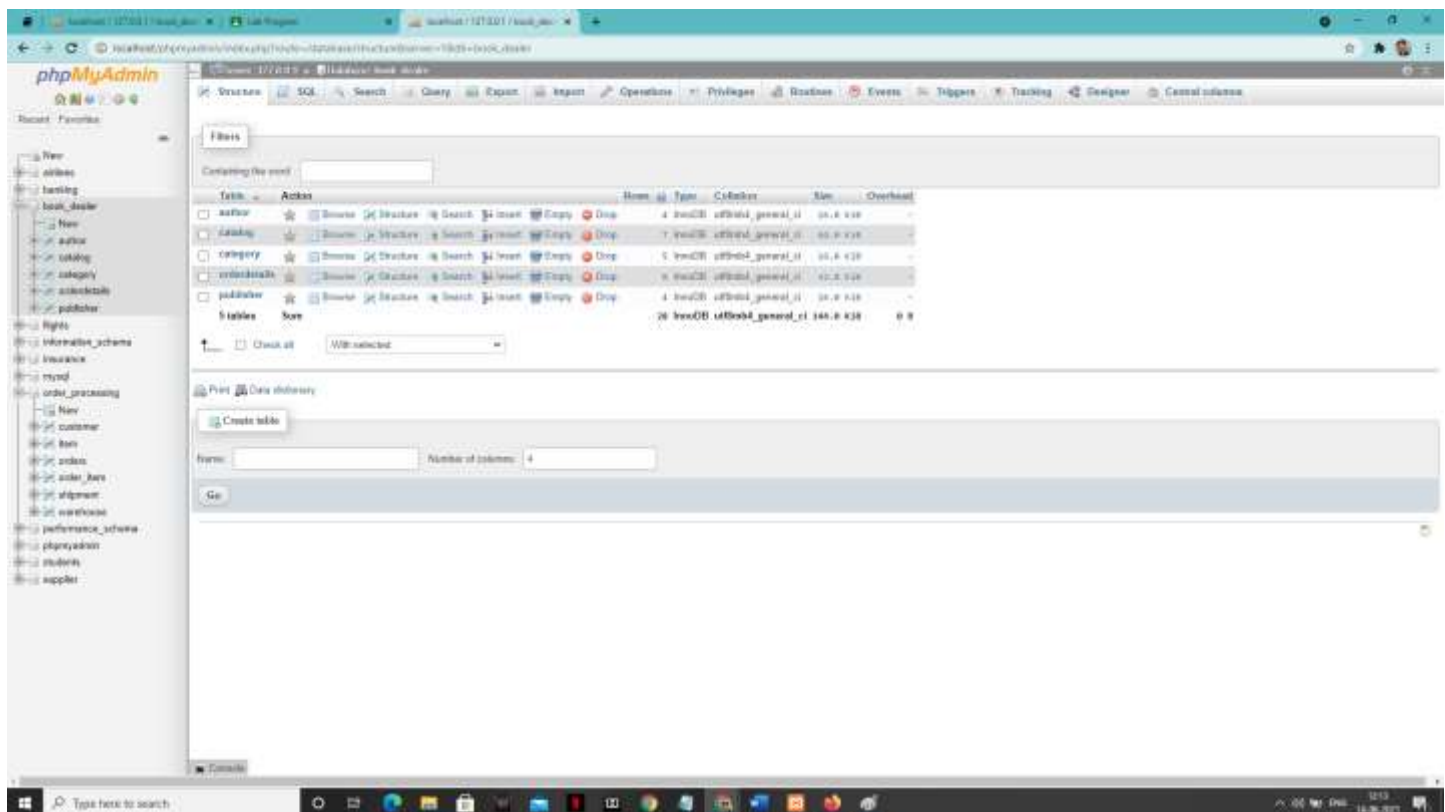
## Create table:-



The screenshot shows the phpMyAdmin interface with the SQL tab selected. The database 'book\_dealer' is chosen. The SQL editor contains the following queries:

```
CREATE TABLE author ( author_id INT, author_name VARCHAR(20), author_city VARCHAR(20), author_country VARCHAR(20), PRIMARY KEY(author_id));
CREATE TABLE publisher ( publisher_id INT, publisher_name VARCHAR(20), publisher_city VARCHAR(20), publisher_country VARCHAR(20), PRIMARY KEY(publisher_id));
CREATE TABLE category ( category_id INT, description VARCHAR(20), PRIMARY KEY(category_id));
CREATE TABLE book ( book_id INT, book_title VARCHAR(20), author_id INT, publisher_id INT, category_id INT, year INT, price INT, PRIMARY KEY(book_id), FOREIGN KEY(author_id) REFERENCES author(author_id), FOREIGN KEY(publisher_id) REFERENCES publisher(publisher_id), FOREIGN KEY(category_id) REFERENCES category(category_id));
CREATE TABLE orderdetails ( order_id INT, book_id INT, quantity INT, PRIMARY KEY(order_id), FOREIGN KEY(book_id) REFERENCES book(book_id));
```

The 'Go' button is clicked, and the results show four successful queries, each returning an empty result set. The console at the bottom shows the execution of each query.



The screenshot shows the phpMyAdmin interface with the Structure tab selected. The database 'book\_dealer' is chosen. The table structure for 'book\_dealer' is displayed:

Table	Action	Rows	Type	Collation	Size	Overhead
author	Structure   Search   Insert   Copy   Drop	4	InnoDB	utf8mb4_general_ci	32.0 x 10	-
category	Structure   Search   Insert   Copy   Drop	7	InnoDB	utf8mb4_general_ci	32.0 x 10	-
book	Structure   Search   Insert   Copy   Drop	5	InnoDB	utf8mb4_general_ci	32.0 x 10	-
orderdetails	Structure   Search   Insert   Copy   Drop	6	InnoDB	utf8mb4_general_ci	32.0 x 10	-
publisher	Structure   Search   Insert   Copy   Drop	4	InnoDB	utf8mb4_general_ci	32.0 x 10	-
<b>Tables</b>	<b>Size</b>	<b>26</b>	<b>InnoDB</b>	<b>utf8mb4_general_ci</b>	<b>344.0 x 30</b>	<b>0.0</b>

The 'Tables' section shows a 'Size' of 26 and an 'Overhead' of 0.0. The 'Check all' button is visible. Below the table structure, the 'Create table' button is highlighted.

## 2) Enter tuples for each relation.

‘Author’ table:

The screenshot shows the phpMyAdmin interface with the 'book\_dealer' database selected. The 'SQL' tab is active, and an INSERT query has been executed. The query is: `INSERT INTO author (author_id,author_name,author_city,author_country) VALUES (1801,'Lalit Kumar','Mumbai','India'), (1802,'Chetan Bhagat','Mumbai','India'), (1803,'Shard Kumar Verma','Delhi','India'), (1804,'Neil Singh','Mumbai','India');`. The execution result shows '4 rows affected. (Query took 0.0000 seconds)'. Below the result, the same query is displayed in a code block.

The screenshot shows the phpMyAdmin interface with the 'book\_dealer' database selected. The 'Browse' tab is active, displaying the 'author' table. The table contains 4 rows of data. The columns are: author\_id, author\_name, author\_city, and author\_country. The data rows are: (1801, Lalit Kumar, Mumbai, India), (1802, Chetan Bhagat, Mumbai, India), (1803, Shard Kumar Verma, Delhi, India), and (1804, Chetan Bhagat, Mumbai, India). The interface also shows options for displaying the table, such as 'Show all', 'Number of rows', and 'Filter rows'.

author_id	author_name	author_city	author_country
1801	Lalit Kumar	Mumbai	India
1802	Chetan Bhagat	Mumbai	India
1803	Shard Kumar Verma	Delhi	India
1804	Chetan Bhagat	Mumbai	India

## 'Publisher' table:

The screenshot shows the phpMyAdmin interface with the 'publisher' table selected in the left sidebar. The main panel displays the 'SQL' tab with an executed INSERT query. The query text is: `INSERT INTO publisher (publisher_id,publisher_name,publisher_city,publisher_country) VALUES (200,'Shreshth','Mumbai','India'), (201,'Neha','Bangalore','India'), (202,'Saba','Pune','India'), (203,'Ravi','Jaipur','India');`. The execution status shows '4 rows inserted (Query took 0.003 seconds)'. Below the query, the raw SQL is displayed: `INSERT INTO publisher (publisher_id,publisher_name,publisher_city,publisher_country) VALUES (200,'Shreshth','Mumbai','India'), (201,'Neha','Bangalore','India'), (202,'Saba','Pune','India'), (203,'Ravi','Jaipur','India');`. The interface includes various toolbars for database management and a bottom status bar showing the system time as 12:46 on 14-09-2021.

The screenshot shows the phpMyAdmin interface with the 'publisher' table selected. The main panel displays the 'Table' tab, showing the table structure and data. The table has four columns: `publisher_id`, `publisher_name`, `publisher_city`, and `publisher_country`. The data is displayed in a table with four rows, each corresponding to one of the inserted records. The interface includes various toolbars for table management and a bottom status bar showing the system time as 12:54 on 14-09-2021.

	<code>publisher_id</code>	<code>publisher_name</code>	<code>publisher_city</code>	<code>publisher_country</code>
<input type="checkbox"/>	200	Shreshth	Mumbai	India
<input type="checkbox"/>	201	Neha	Bangalore	India
<input type="checkbox"/>	202	Saba	Pune	India
<input type="checkbox"/>	203	Ravi	Jaipur	India

## ‘Catalog’ table:

The screenshot shows the phpMyAdmin interface with the 'catalog' table selected. A SQL query is entered in the 'Run SQL query' box and executed. The query inserts seven rows into the 'catalog' table. The results are displayed below the query box, showing 7 rows inserted.

```

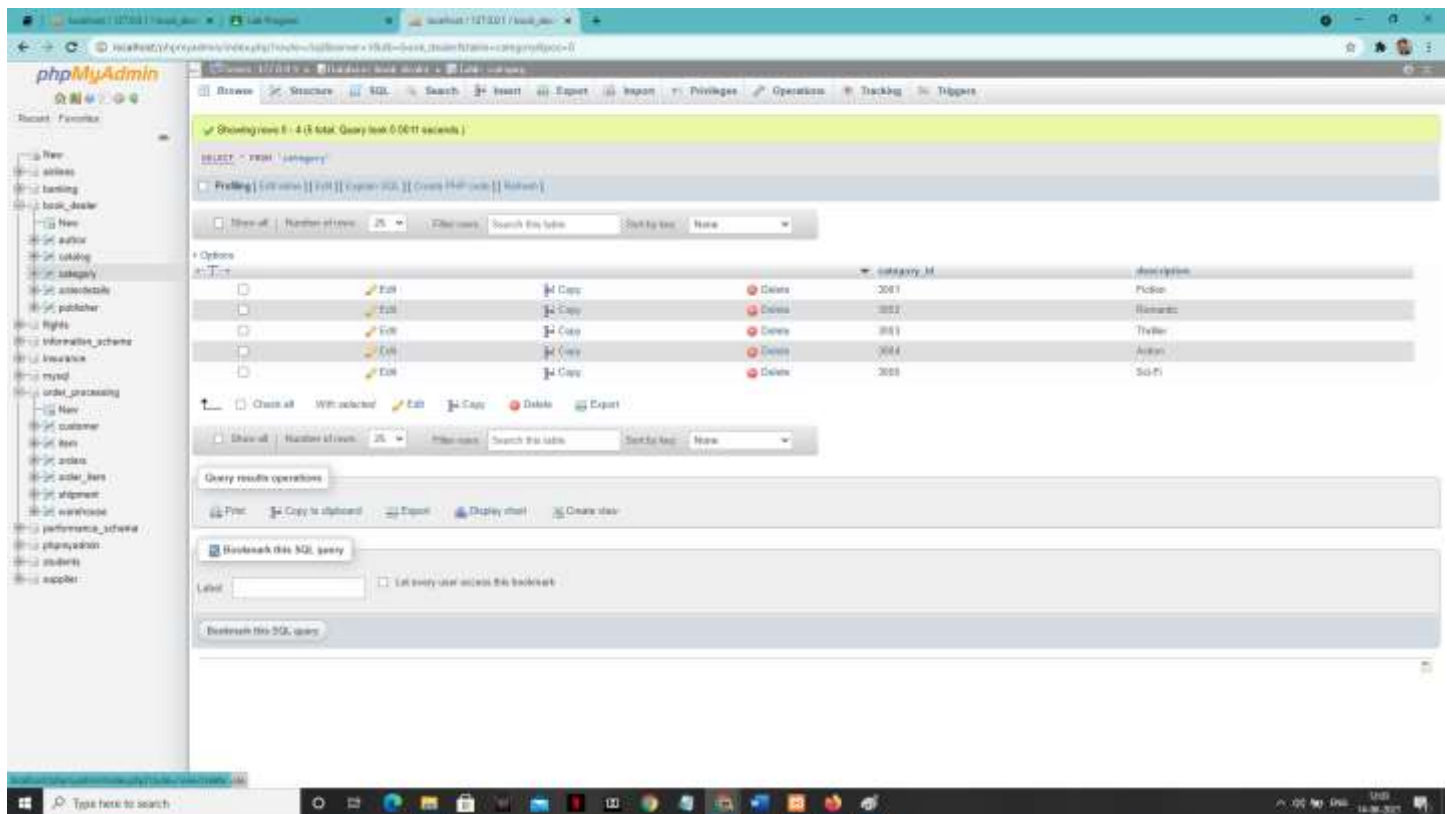
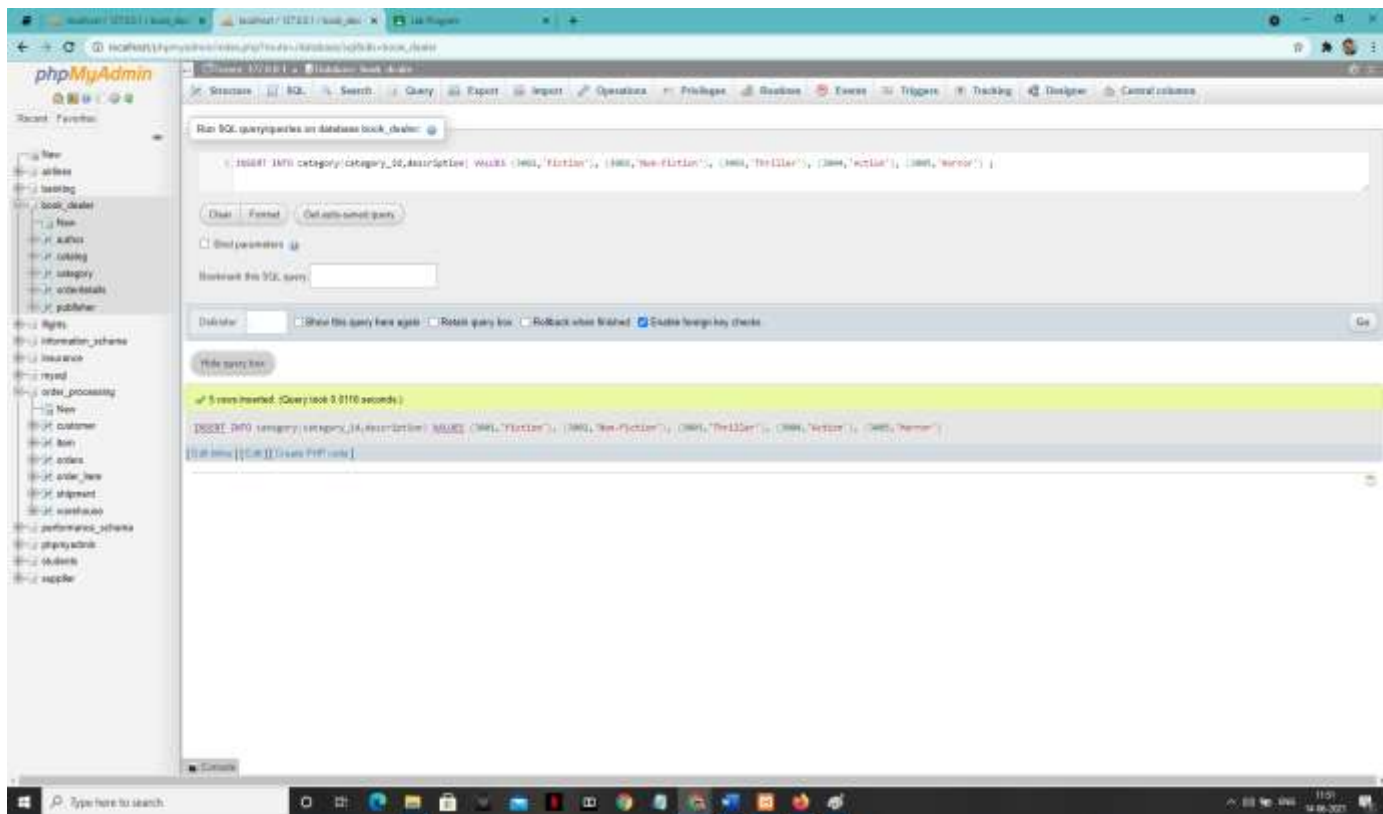
INSERT INTO catalog(book_id,book_title,author_id,publisher_id,category_id,year,price) VALUES ('HP and Order of the Phoenix',2001,2001,2001,2001,2001,2001), ('The Hunger',2001,2001,2001,2001,2001,2001), ('HP and Order of Phoenix',2001,2001,2001,2001,2001,2001), ('The Hunger',2001,2001,2001,2001,2001,2001), ('HP and Order of Phoenix',2001,2001,2001,2001,2001,2001), ('The Hunger',2001,2001,2001,2001,2001,2001), ('HP and Order of Phoenix',2001,2001,2001,2001,2001,2001);
  
```

7 rows inserted (Query took 0.004 seconds)

The screenshot shows the phpMyAdmin interface with the 'catalog' table selected. The table structure is displayed, showing columns: book\_id, book\_title, author\_id, publisher\_id, category\_id, year, and price. The table contains 7 rows of data.

book_id	book_title	author_id	publisher_id	category_id	year	price
2001	HP and Order of the Phoenix	2001	2001	2001	2001	2001
2001	The Hunger	2001	2001	2001	2001	2001
2001	HP and Order of Phoenix	2001	2001	2001	2001	2001
2001	The Hunger	2001	2001	2001	2001	2001
2001	HP and Order of Phoenix	2001	2001	2001	2001	2001
2001	The Hunger	2001	2001	2001	2001	2001
2001	HP and Order of Phoenix	2001	2001	2001	2001	2001

‘Category’ value: -





## ‘Orderdetails’ table:

The screenshot shows the phpMyAdmin interface with the 'orderdetails' table selected in the database 'book\_store'. The SQL query editor displays the following query:

```
INSERT INTO orderdetails (order_id,book_id,quantity) VALUES (5001,4001,5), (5001,4001,7), (5001,4001,15), (5004,4004,11), (5005,4005,9), (5006,4006,8), (5007,4007,2), (5008,4008,3);
```

The query has been executed successfully, resulting in 8 rows inserted. The status bar indicates '8 rows inserted (Query took 0.0107 seconds)'.

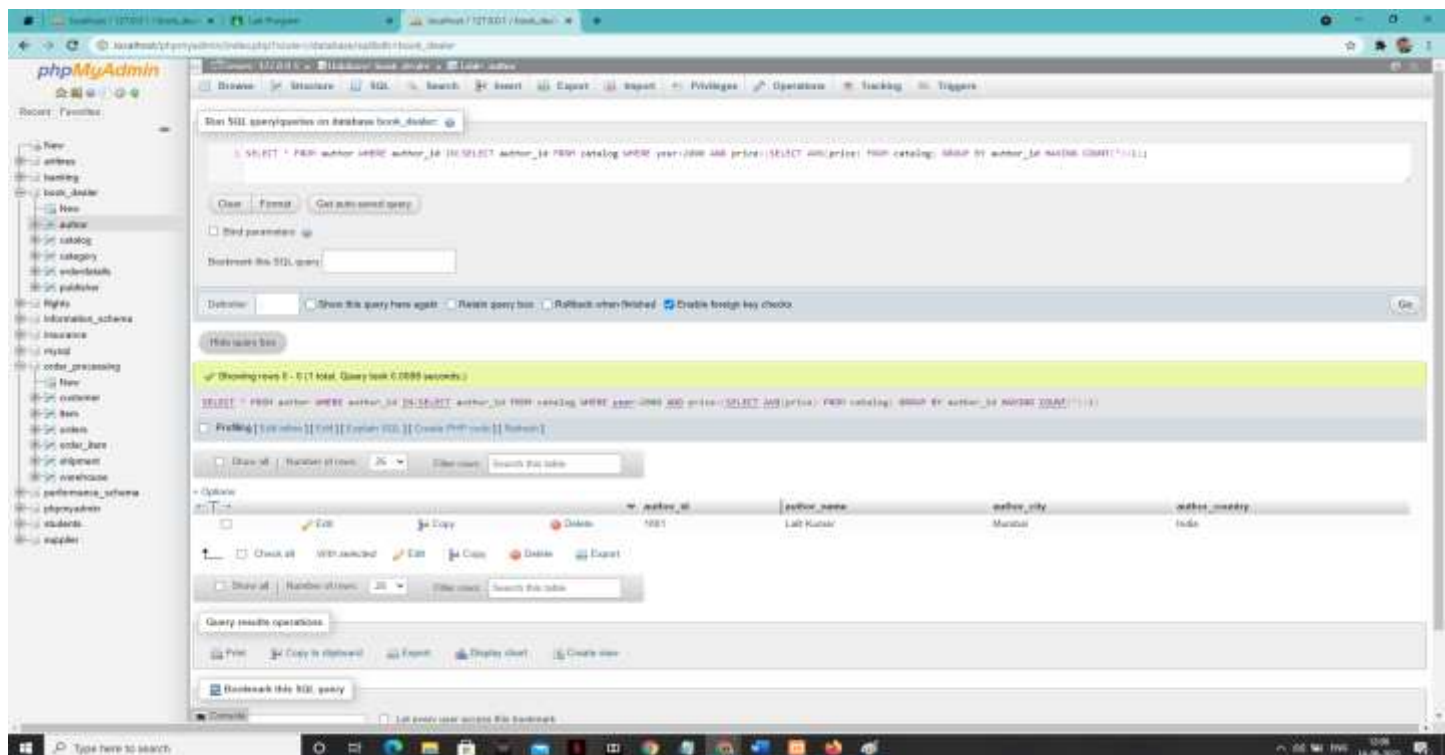
The screenshot shows the phpMyAdmin interface displaying the data of the 'orderdetails' table. The table has 8 rows of data. The columns are 'order\_id', 'book\_id', and 'quantity'.

order_id	book_id	quantity
5001	4001	5
5001	4001	7
5001	4001	15
5004	4004	11
5005	4005	9
5006	4006	8
5007	4007	2
5008	4008	3

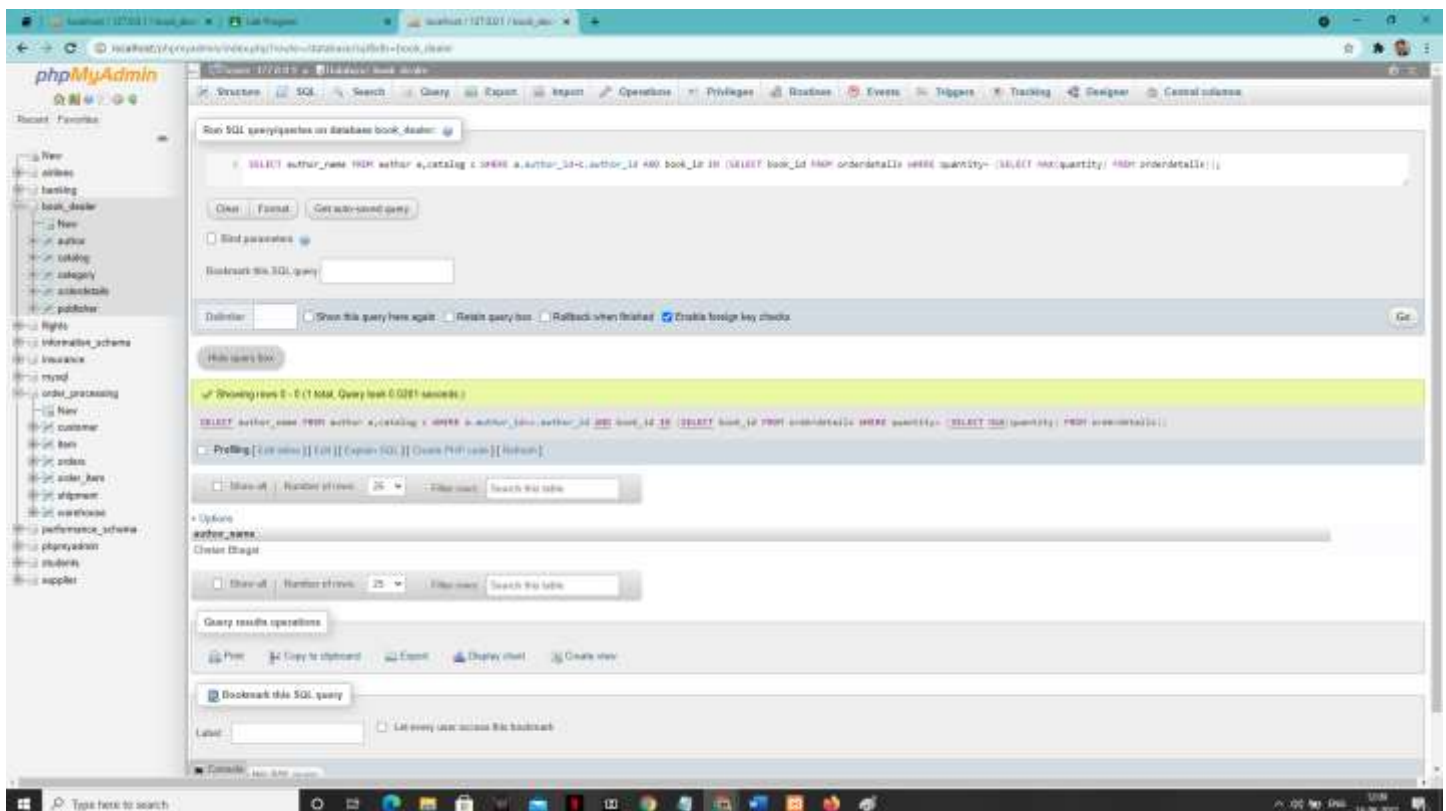
The status bar indicates 'Showing rows 0 - 7 (8 total. Query took 0.0011 seconds)'.

# SOLUTION

1-



2:-





3:-

The screenshot shows the phpMyAdmin interface with the 'book\_store' database selected. A SQL query has been executed successfully, returning 1 row. The query is: `UPDATE catalog SET price=1*price WHERE publisher_id IN (SELECT publisher_id FROM publisher WHERE publisher_name='Shrohart');`

Query result:

```

1 row affected (Query took 0.0072 seconds)

UPDATE catalog SET price=1*price WHERE publisher_id IN (SELECT publisher_id FROM publisher WHERE publisher_name='Shrohart')
1 row in set (0.00) [Click for SQL]
  
```

The screenshot shows the phpMyAdmin interface displaying a table of book data. The table has columns: book\_id, book\_title, publisher\_id, publisher\_name, category\_id, year, and price. There are 6 rows of data displayed.

book_id	book_title	publisher_id	publisher_name	category_id	year	price
4001	HP and Global Office	3001	3001	2002	2002	600
4002	For Student	3002	3002	2003	2003	800
4003	HP and Order Of Process	3003	3003	2004	2004	80
4004	3 Minutes of my life	3004	3004	2005	2005	95
4005	Artificial Intelligence	3005	3005	2006	2006	450
4006	Angels and Demons	3006	3006	2007	2007	300

## **PROGRAM-8**

### **PROGRAM 8 :- STUDENT ENROLLMENT DATABASE**

Consider the following database of student enrollment in courses and books adopted for each course.

**STUDENT** (regno: String, name: String, major: String, bdate: date)

**COURSE** (course #: int, cname: String, dept: String)

**ENROLL** (regno: String, cname: String, sem: int, marks: int)

**BOOK\_ADOPTION** (course #: int, sem: int, book-ISBN: int)

**TEXT**(book-ISBN:int, book-title:String, publisher:String, author:String)

i. Create the above tables by properly specifying the primary keys and the foreign keys.

ii. Enter at least five tuples for each relation.

iii. Demonstrate how you add a new text book to the database and make this book be adopted by some

department.

iv. Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses

offered by the 'CS' department that use more than two books.

v. List any department that has all its adopted books published by a specific publisher.

## Create table:-

The screenshot shows the phpMyAdmin interface with the SQL tab selected. The database 'student\_warehouse' is chosen. The SQL query area contains the following commands:

```
CREATE TABLE student (regno VARCHAR(11), regn VARCHAR(10), regn VARCHAR(10), data DATE, PRIMARY KEY (regno));
CREATE TABLE course (regno VARCHAR(11), regn VARCHAR(10), regn VARCHAR(10), PRIMARY KEY (regno), FOREIGN KEY (course) REFERENCES student (regno));
CREATE TABLE enroll (regno VARCHAR(11), courseno INT, enr INT(11), enr INT(11), PRIMARY KEY (regno, courseno), FOREIGN KEY (courseno) REFERENCES course (courseno));
CREATE TABLE book (book_id VARCHAR(11), book_title VARCHAR(10), publisher VARCHAR(10), author VARCHAR(10), PRIMARY KEY (book_id));
CREATE TABLE book_adoption (courseno INT, enr INT(11), book_id VARCHAR(11), PRIMARY KEY (courseno, book_id), FOREIGN KEY (courseno) REFERENCES course (courseno), FOREIGN KEY (book_id) REFERENCES book (book_id));
```

Below the query area, the execution results are shown for each table creation:

- MySQL returned an empty result set (0 rows affected) (Query took 0.0273 seconds)
- MySQL returned an empty result set (0 rows affected) (Query took 0.0265 seconds)
- MySQL returned an empty result set (0 rows affected) (Query took 0.0332 seconds)
- MySQL returned an empty result set (0 rows affected) (Query took 0.0243 seconds)

The screenshot shows the phpMyAdmin interface with the Structure tab selected for the 'book\_adoption' table. The table structure is displayed as follows:

Table	Actions	Scan	Size	Type	Collation	Size	Overhead
book_adoption	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	1 row	11.0 K	MySQL	utf8mb4_general_ci	11.0 K	0
enroll	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	1 row	11.0 K	MySQL	utf8mb4_general_ci	11.0 K	0
student	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	1 row	11.0 K	MySQL	utf8mb4_general_ci	11.0 K	0
book	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	1 row	11.0 K	MySQL	utf8mb4_general_ci	11.0 K	0
<b>1 tables</b>	<b>Sum</b>					44.0 K	0

Below the table structure, the 'Create table' button is visible, along with a 'Number of columns' field set to 4.

2) Enter tuples for each relation.

‘STUDENT’ table:

The screenshot shows the phpMyAdmin interface for a database named 'student'. The left sidebar lists various databases and tables, including 'student' and 'student\_enrollment'. The main panel displays the 'STUDENT' table structure, which has columns: 'id', 'name', 'age', 'sex', 'email', and 'phone'. The table contains 5 rows of data. Below the table, there are options to 'Query results operations' and 'Bookmark this SQL query'.

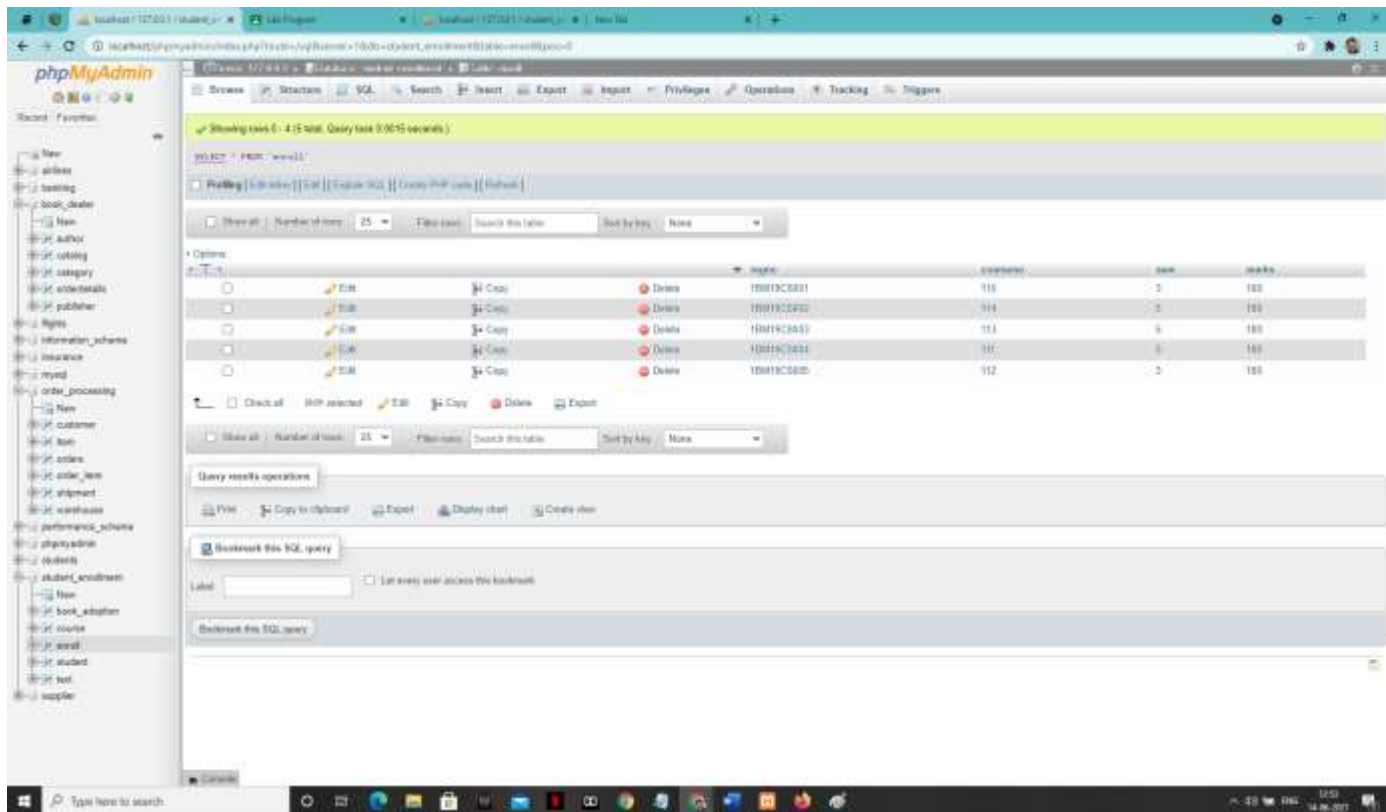
id	name	age	sex	email	phone
1001	John	20	M	john@163.com	13800131200
1002	Jane	19	F	jane@163.com	13800131201
1003	Mike	21	M	mike@163.com	13800131202
1004	Emily	22	F	emily@163.com	13800131203
1005	David	23	M	david@163.com	13800131204

‘COURSE’ table:

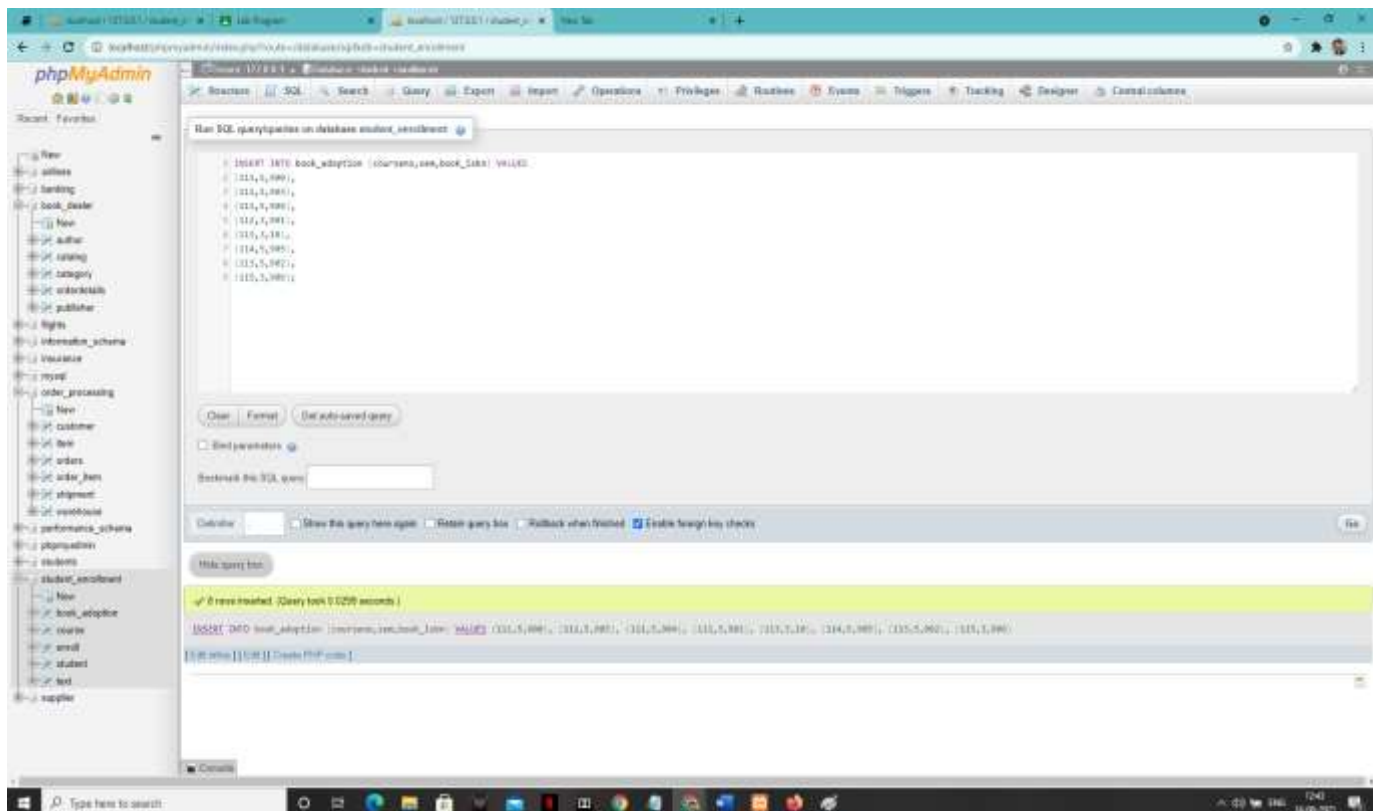
The screenshot shows the phpMyAdmin interface for a database named 'student'. The left sidebar lists various databases and tables, including 'student' and 'student\_enrollment'. The main panel displays the 'COURSE' table structure, which has columns: 'id', 'name', 'credits', 'prerequisites', and 'instructor'. The table contains 5 rows of data. Below the table, there are options to 'Query results operations' and 'Bookmark this SQL query'.

id	name	credits	prerequisites	instructor
1001	Math	3		John
1002	Physics	4	1001	Jane
1003	Chemistry	3	1001	Mike
1004	Biology	3	1001	Emily
1005	History	3		David

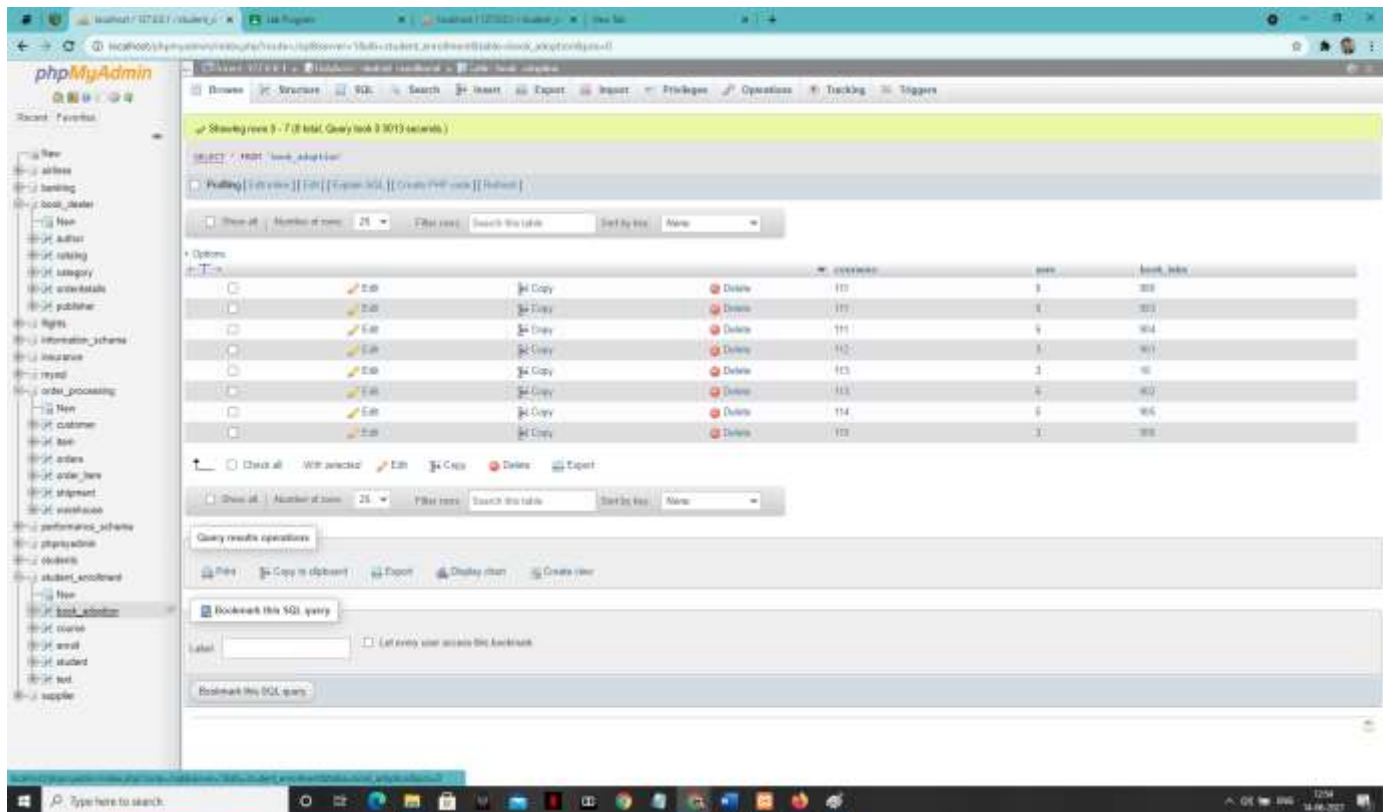




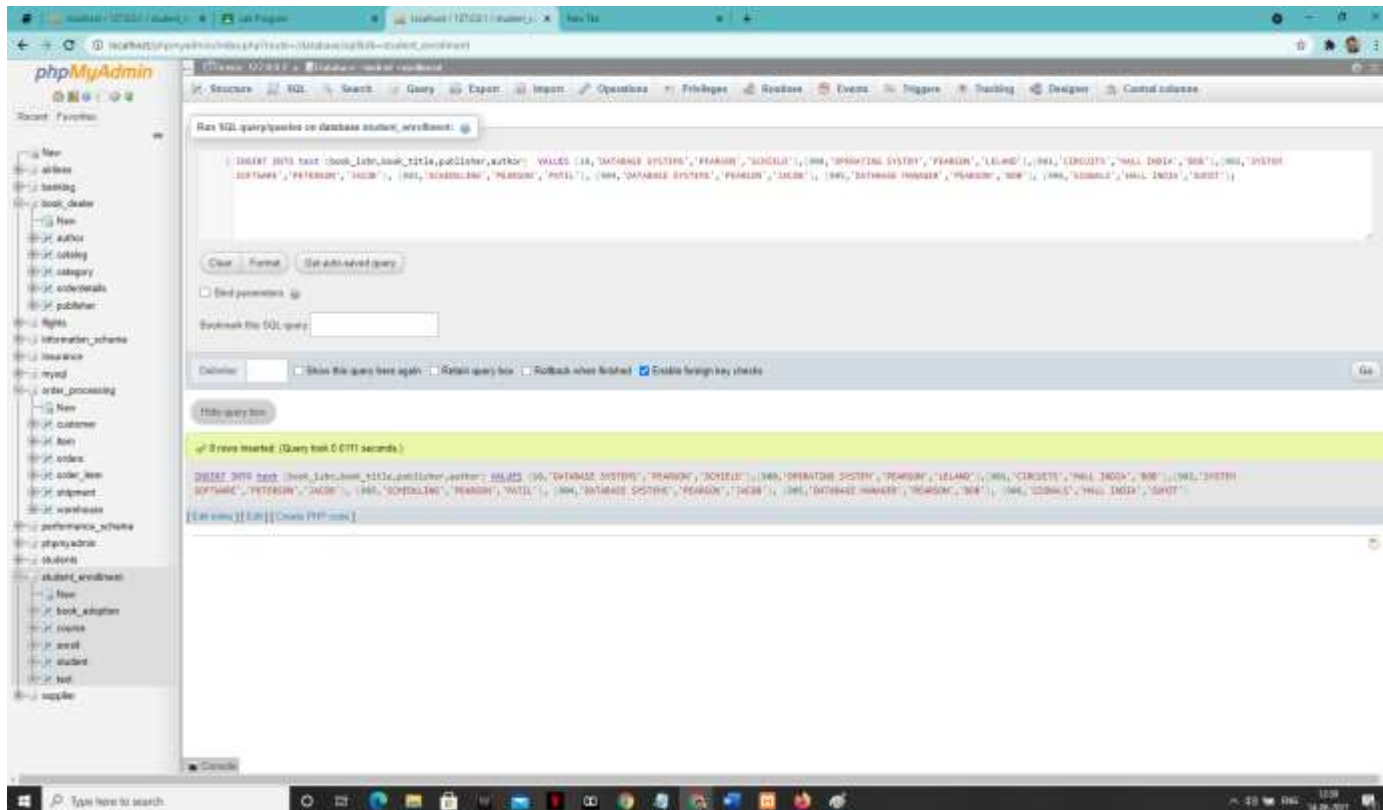
‘BOOK ADOPTION’ value: -

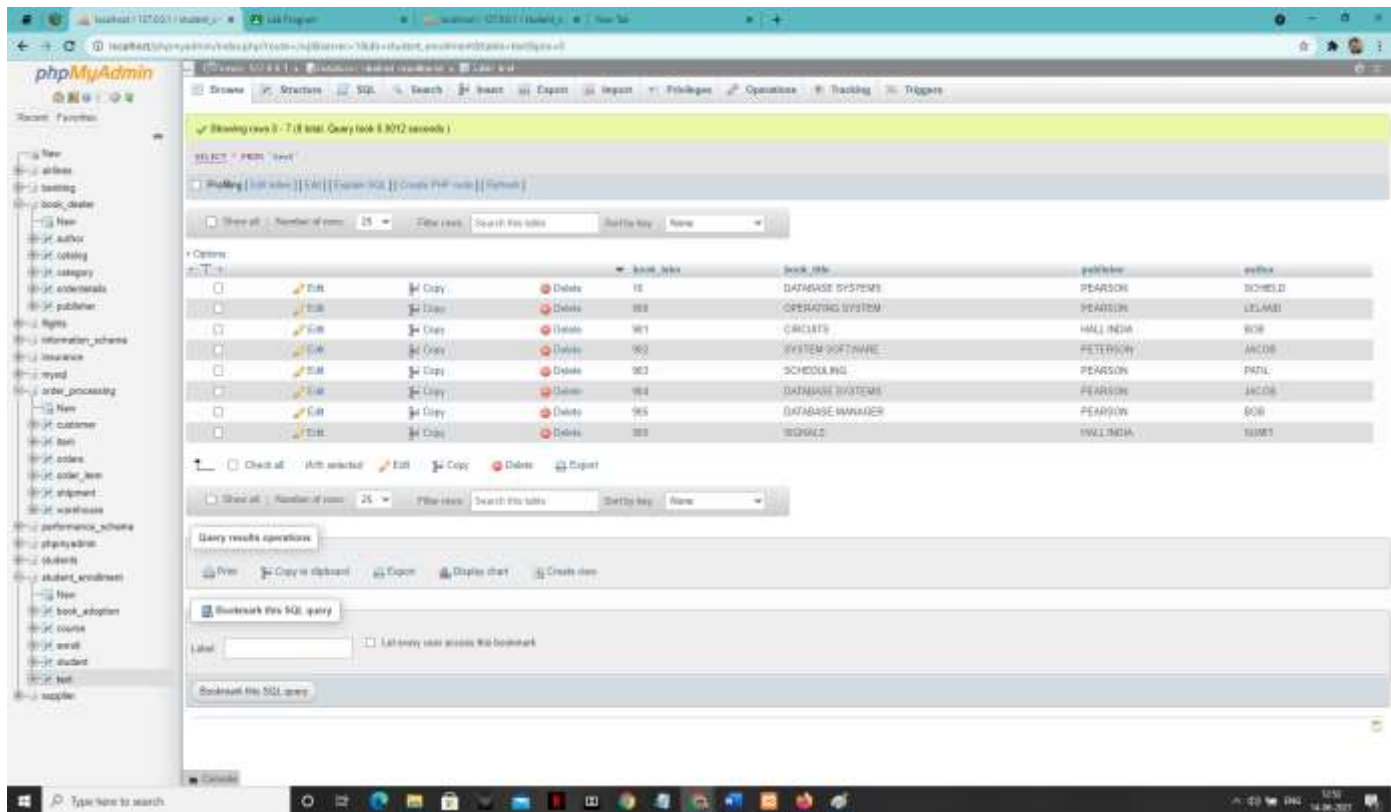






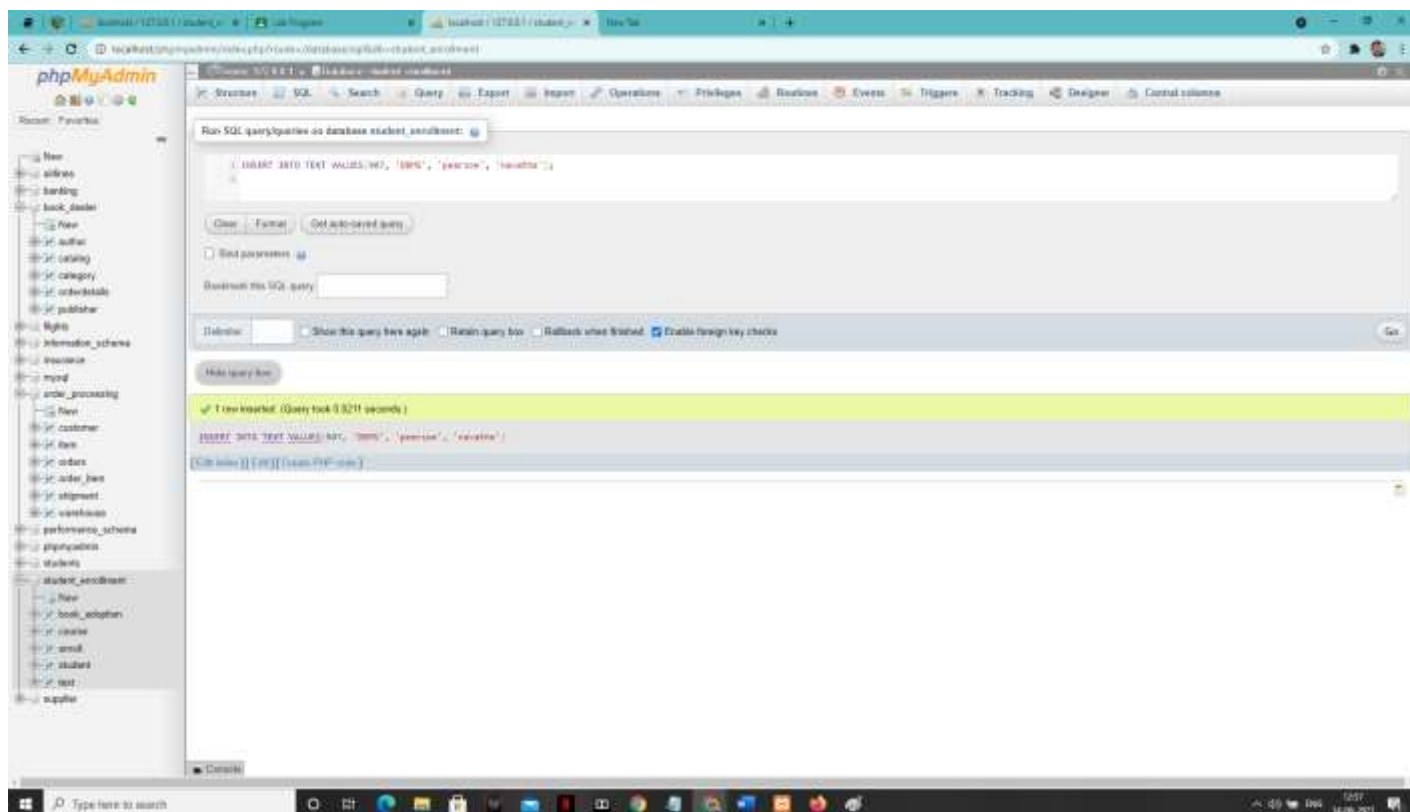
‘TEXT’ table:





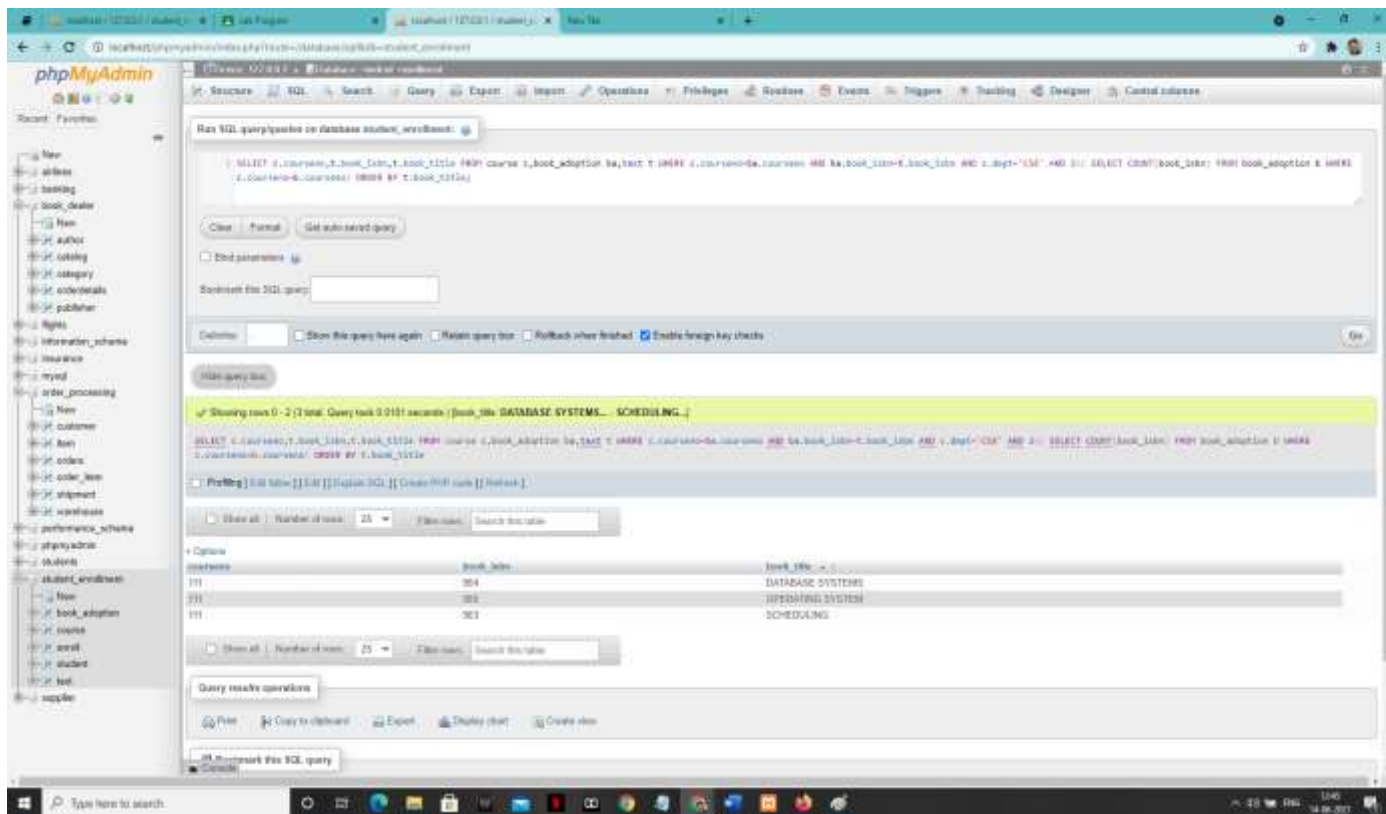
## SOLUTION

1:-

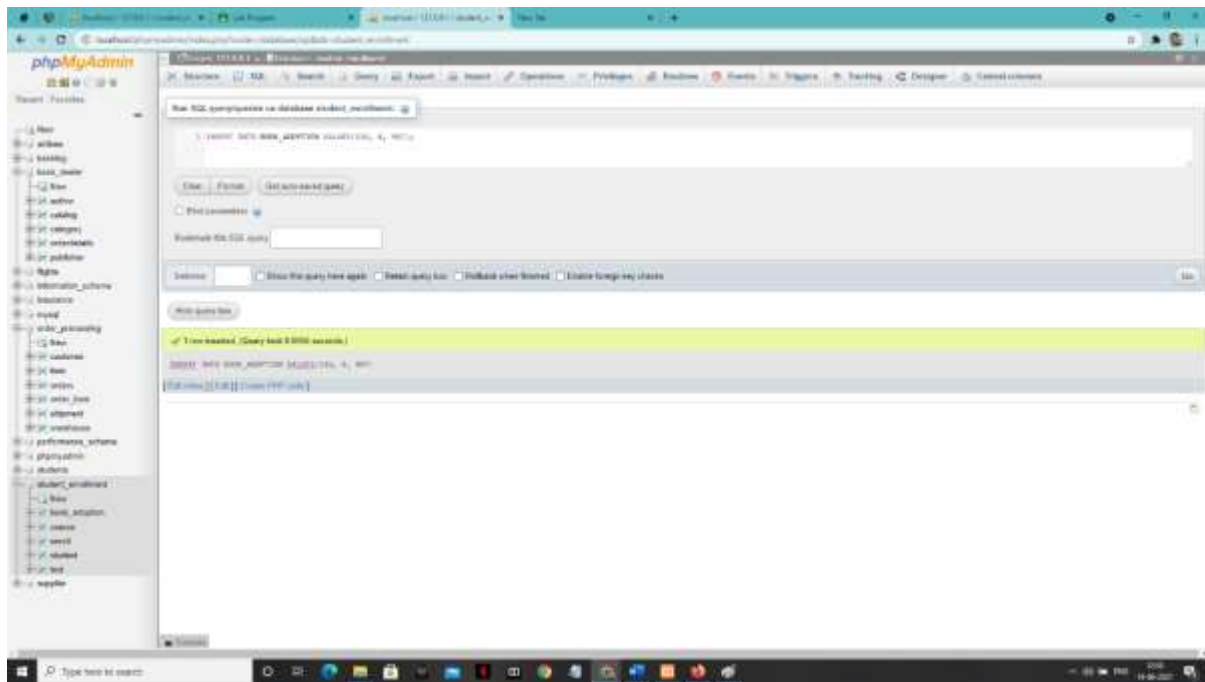








The screenshot shows the phpMyAdmin interface with the 'test' database selected. The 'SQL' tab is active, and a query has been executed successfully. The query is: `INSERT INTO TEST VALUES(999, '9999', '999999', '99999999')`. The result shows 1 row inserted, taking 0.0271 seconds. The left sidebar shows the database structure with tables like test, test2, and test3.



## PROGRAM-9

### PROGRAM 9 :- MOVIE DATABASE

Consider the schema for Movie Database:

**ACTOR**(Act\_id, Act\_Name, Act\_Gender)

**DIRECTOR**(Dir\_id, Dir\_Name, Dir\_Phone)

**MOVIES**(Mov\_id, Mov\_Title, Mov\_Year, Mov\_Lang, Dir\_id)

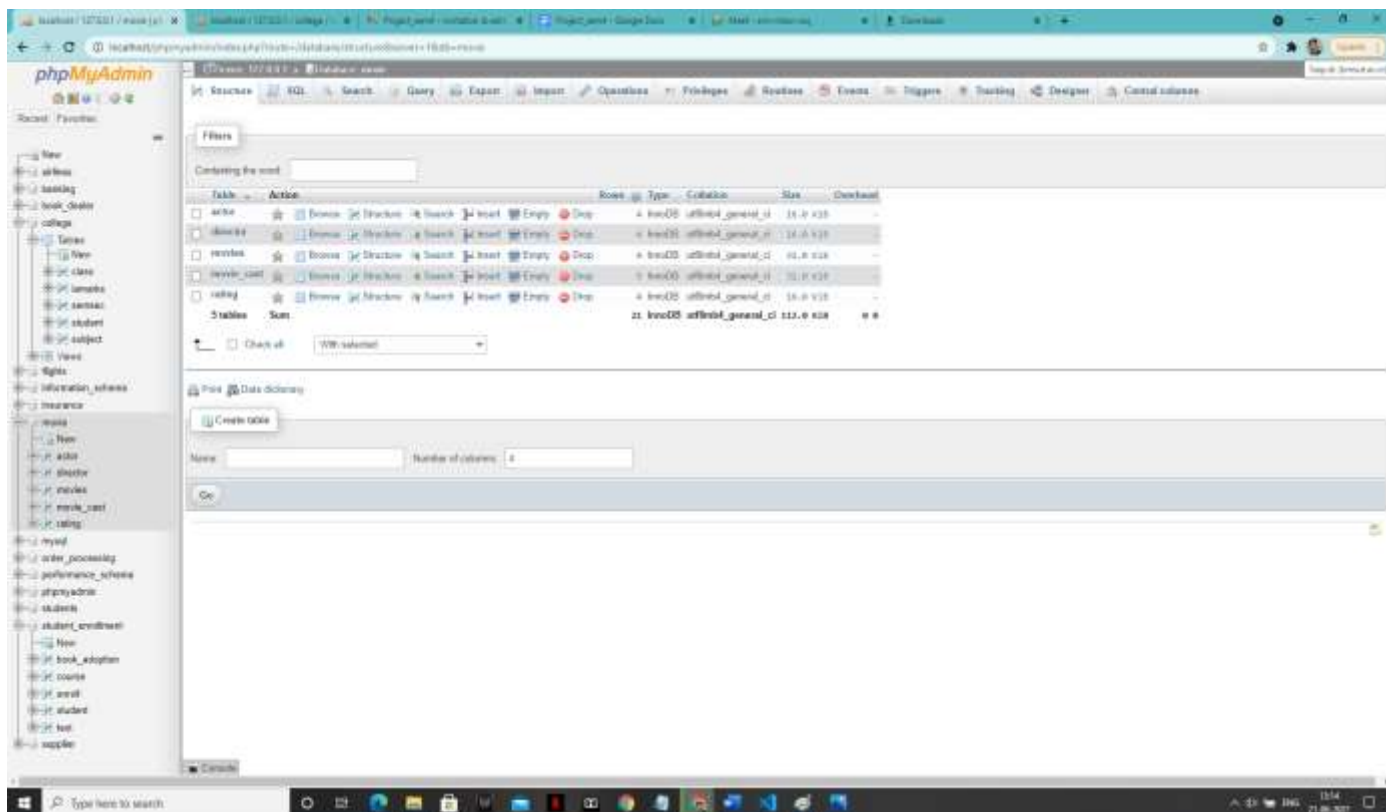
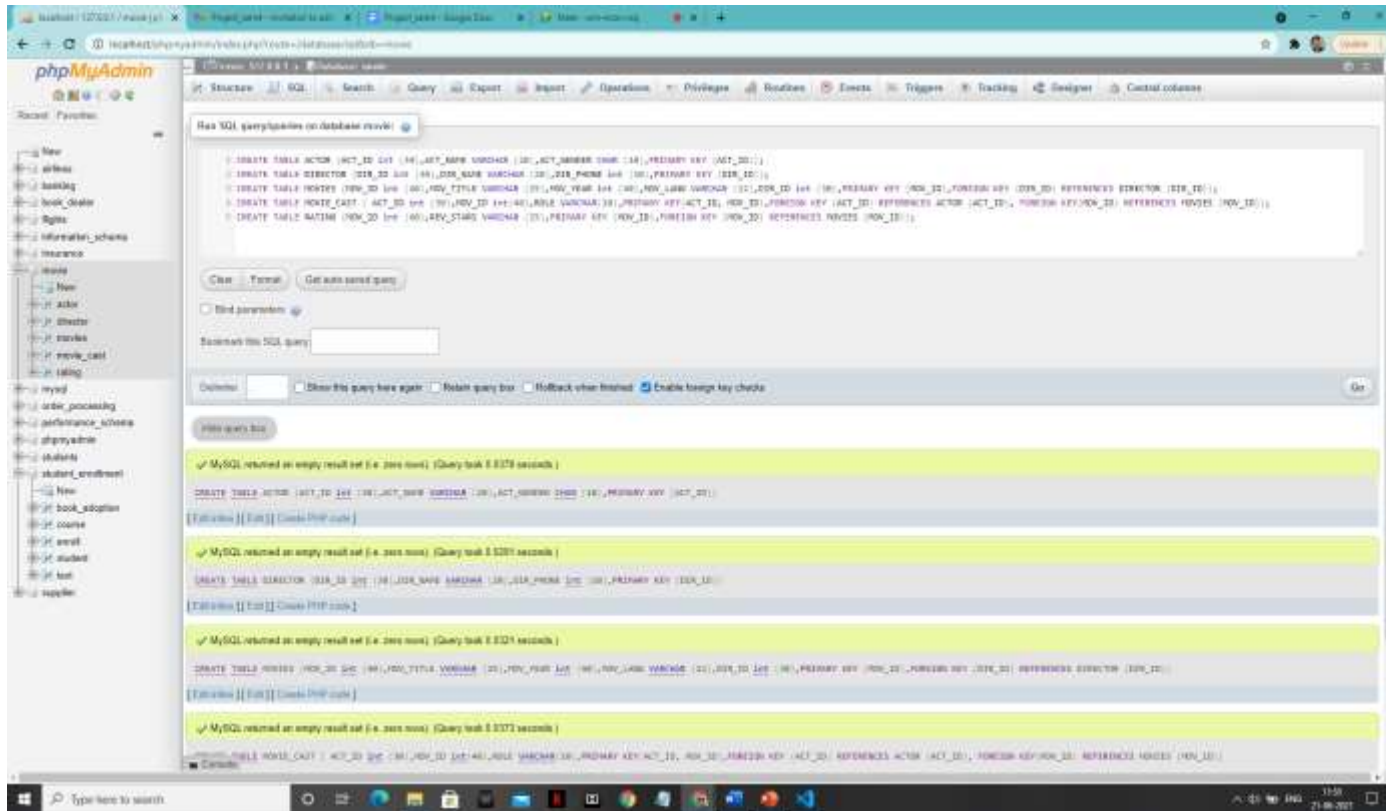
**MOVIE\_CAST**(Act\_id, Mov\_id, Role)

**RATING**(Mov\_id, Rev\_Stars)

Write SQL queries to

- i. List the titles of all movies directed by 'Hitchcock'.
- ii. Find the movie names where one or more actors acted in two or more movies.
- iii. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).
- iv. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.
- v. Update rating of all movies directed by 'Steven Spielberg' to 5.

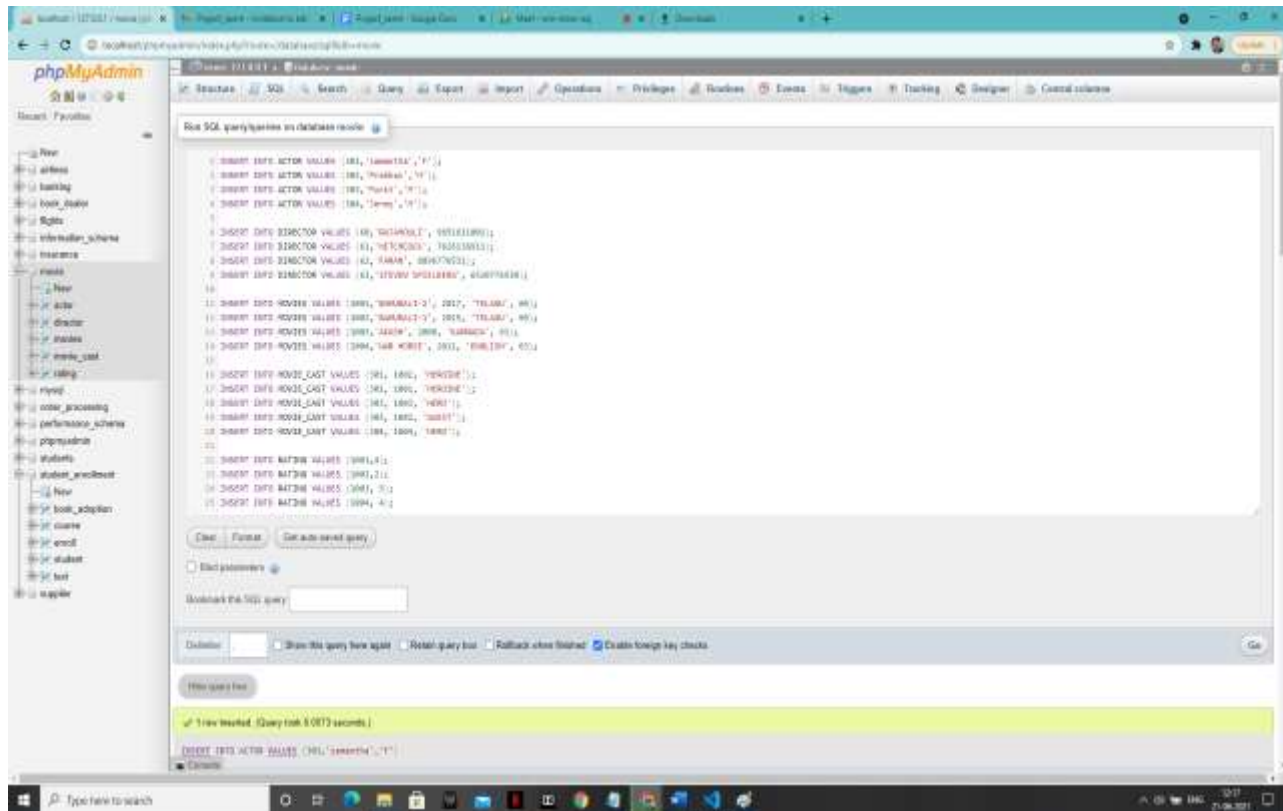
**Create table:-**



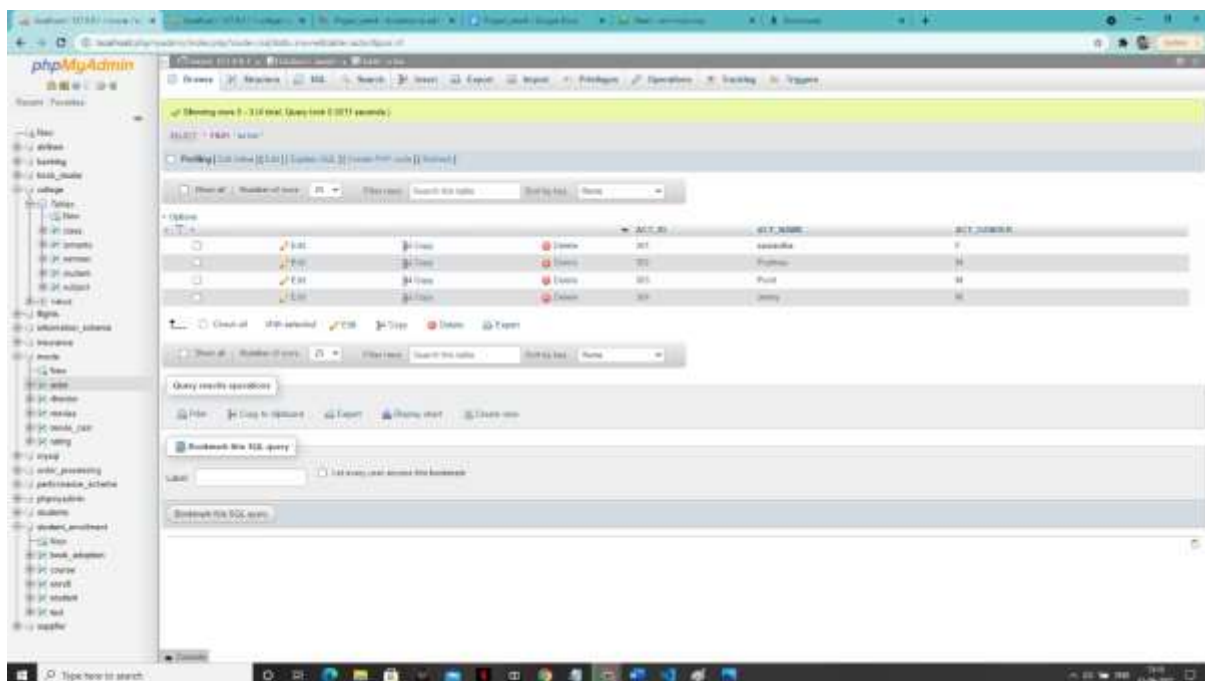


## 2) Enter tuples for each relation:-

TABLES VALUES:-



‘ACTOR’ table:



## ‘DIRECTOR’ table:

The screenshot shows the phpMyAdmin interface with the 'DIRECTOR' table selected. The table structure is as follows:

DBL_ID	DIR_NAME	DIR_PHONE
83	RAJAWALE	2187403647
91	WITTHICOOK	2187403647
82	PARIAN	2187403647
83	STEVEN SPEELING	2187403647

The interface also shows the SQL query: `SELECT * FROM `director`` and various options for table operations.

## ‘MOVIE’ table:

The screenshot shows the phpMyAdmin interface with the 'MOVIE' table selected. The table structure is as follows:

MOVIE_ID	MOVIE_TITLE	MOVIE_YEAR	MOVIE_LENGTH	DBL_ID
1991	Backstage	2011	78	83
1992	Backstage 2	2016	78	91
1993	Backstage	2016	78	82
1994	Backstage	2011	78	83

The interface also shows the SQL query: `SELECT * FROM `movie`` and various options for table operations.

‘MOVIE\_CAST’ value: -

The screenshot shows the phpMyAdmin interface with the 'MOVIE\_CAST' table selected. The table structure is as follows:

Field	Type	Length	Null	Key	Extra
CAST_ID	INT	11	NO	PRIMARY	
MOVIE_ID	INT	11	NO	INDEX	
ROLE	VARCHAR	45	NO		

The table contains 4 rows of data:

CAST_ID	MOVIE_ID	ROLE
301	1001	HEROINE
301	1002	HEROINE
303	1003	GUEST
303	1003	HERO

‘RATING’ table:

The screenshot shows the phpMyAdmin interface with the 'RATING' table selected. The table structure is as follows:

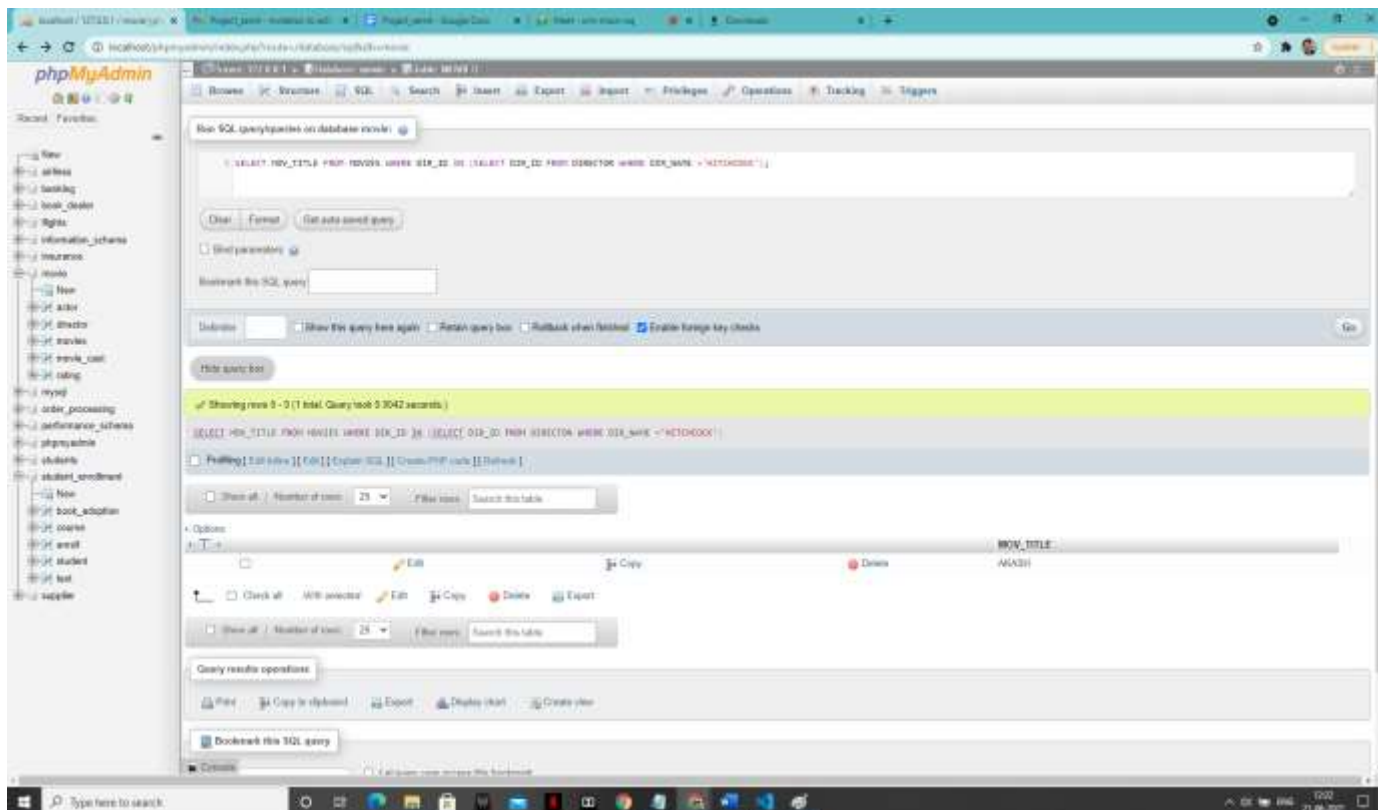
Field	Type	Length	Null	Key	Extra
MOVIE_ID	INT	11	NO	INDEX	
RAT_STARS	INT	1	NO		

The table contains 4 rows of data:

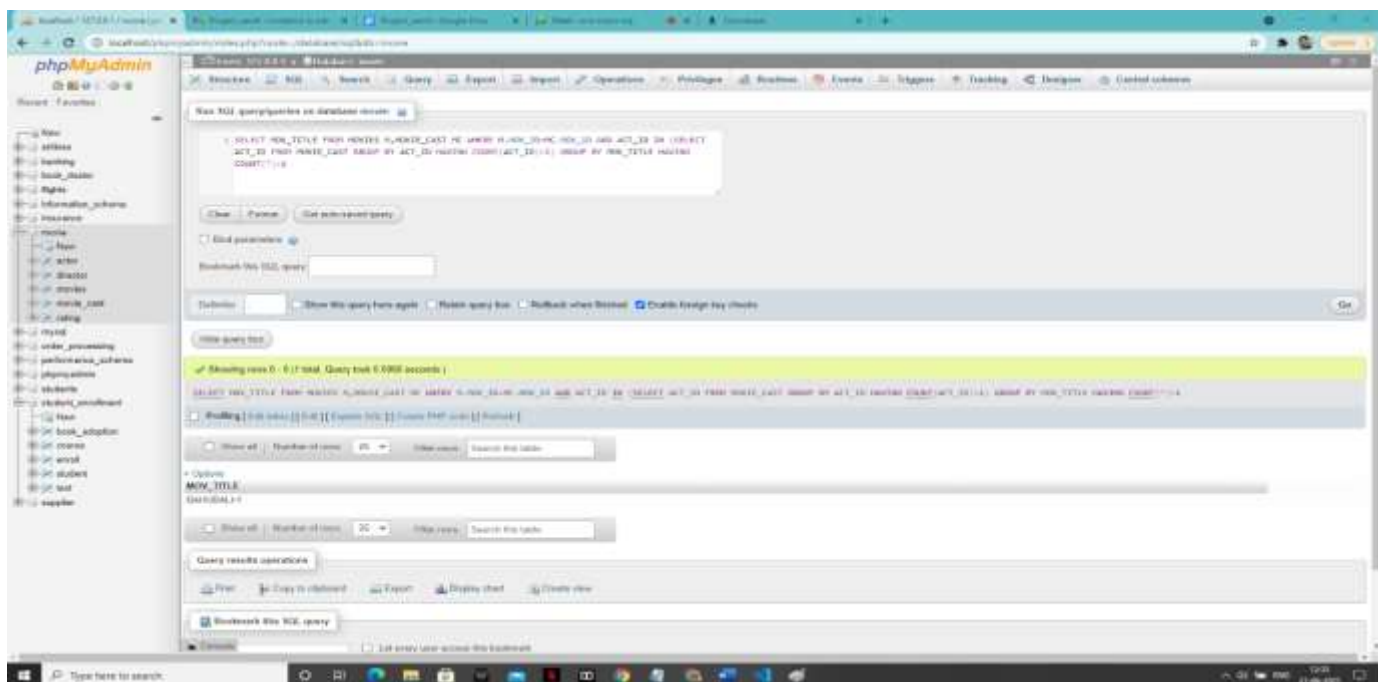
MOVIE_ID	RAT_STARS
1001	4
1002	5
1003	5
1004	5



**1:-**



**2:-**



3:-

The screenshot shows the phpMyAdmin interface with a SQL query executed. The query is: `SELECT MOV_YEAR, MOV_TITLE, MOV_YEAR FROM MOVIES WHERE MOV_YEAR < 2000 ORDER BY MOV_YEAR ASC`. The result shows a single row with MOV\_YEAR: 2017 and MOV\_TITLE: SARFAROSH.

SQL Query:

```
SELECT MOV_YEAR, MOV_TITLE, MOV_YEAR FROM MOVIES WHERE MOV_YEAR < 2000 ORDER BY MOV_YEAR ASC
```

Query Results:

MOV_YEAR	MOV_TITLE
2017	SARFAROSH

4:

The screenshot shows the phpMyAdmin interface with a SQL query executed. The query is: `SELECT MOV_YEAR, MOV_TITLE, MOV_YEAR FROM MOVIES WHERE MOV_YEAR < 2000 ORDER BY MOV_YEAR ASC`. The result shows a single row with MOV\_YEAR: 2017 and MOV\_TITLE: SARFAROSH.

SQL Query:

```
SELECT MOV_YEAR, MOV_TITLE, MOV_YEAR FROM MOVIES WHERE MOV_YEAR < 2000 ORDER BY MOV_YEAR ASC
```

Query Results:

MOV_YEAR	MOV_TITLE
2017	SARFAROSH

5:-

The screenshot shows the phpMyAdmin interface with the SQL tab selected. A query is entered in the text area: `UPDATE ratings SET REV_STARS=5 WHERE MOV_ID IN (SELECT MOV_ID FROM MOVIES WHERE MOV_ID IN (SELECT MOV_ID FROM DIRECTOR WHERE DIR_NAME = 'STEVEN SPIELBERG'));`. The query is executed, and a message states: "1 row affected (Query took 0.0271 seconds)". The query is also displayed in the SQL history below.

The screenshot shows the phpMyAdmin interface with the SQL tab selected. A query is entered in the text area: `SELECT * FROM ratings;`. The query is executed, and a message states: "Showing rows 6 - 9 (4 total. Query took 0.0012 seconds)". The query is also displayed in the SQL history below. Below the history, there is a table of results with columns MOV\_ID and REV\_STARS.

MOV_ID	REV_STARS
1001	4
1002	3
1003	5
1004	5

## **PROGRAM-10**

### **PROGRAM 10:- COLLEGE DATABASE**

**Consider the schema for College Database:**

**STUDENT(USN, SName, Address, Phone, Gender)**

**SEMSEC(SSID, Sem, Sec)**

**CLASS(USN, SSID)**

**SUBJECT(Subcode, Title, Sem, Credits)**

**MARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)**

- i. List all the student details studying in fourth semester 'C' section.**
- ii. Compute the total number of male and female students in each semester and in each section.**
- iii. Create a view of Test1 marks of student USN '22' in all subjects.**
- iv. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.**
- v. Categorize students based on the following criterion: If FinalIA = 17 to 20 then CAT = 'Outstanding' If FinalIA = 12 to 16 then CAT = 'Average' If FinalIA < 12 then CAT = 'Weak' Give these details only for 8th semester A, B, and C section students.**

## Create table:-

The screenshot shows the phpMyAdmin interface with the SQL tab selected. The database 'college' is chosen from the left sidebar. The SQL editor contains five CREATE TABLE statements, each followed by a 'Go' button. The execution results show that each statement was successful, returning an empty result set (0 rows).

```
CREATE TABLE STUDENT ( ID VARCHAR (10) PRIMARY KEY, NAME VARCHAR (20), ADDRESS VARCHAR (50), PHONE INT (10), SEX CHAR (10));
```

```
CREATE TABLE EMPLOYEE ( EMP VARCHAR (10) PRIMARY KEY, EMP_ID VARCHAR (10), SEX CHAR (10), DOB DATE (10));
```

```
CREATE TABLE CLASS ( ID VARCHAR (10), SSN VARCHAR (10), PRIMARY KEY (ID, SSN), FOREIGN KEY (SSN) REFERENCES STUDENT (ID), FOREIGN KEY (SSN) REFERENCES EMPLOYEE (SSN));
```

```
CREATE TABLE SUBJECT ( SUBJECT VARCHAR (20), TITLE VARCHAR (20), SEM INT (2), CREDITS INT (2), PRIMARY KEY (SUBJECT));
```

```
CREATE TABLE SECTION ( ID VARCHAR (10), SUBJECT VARCHAR (20), SSN VARCHAR (10), TEST1 INT (20), TEST2 INT (20), FINALIA INT (20), PRIMARY KEY (ID, SUBJECT, SSN), FOREIGN KEY (SUBJECT, SSN) REFERENCES STUDENT (SUBJECT, SSN), FOREIGN KEY (SSN) REFERENCES EMPLOYEE (SSN));
```

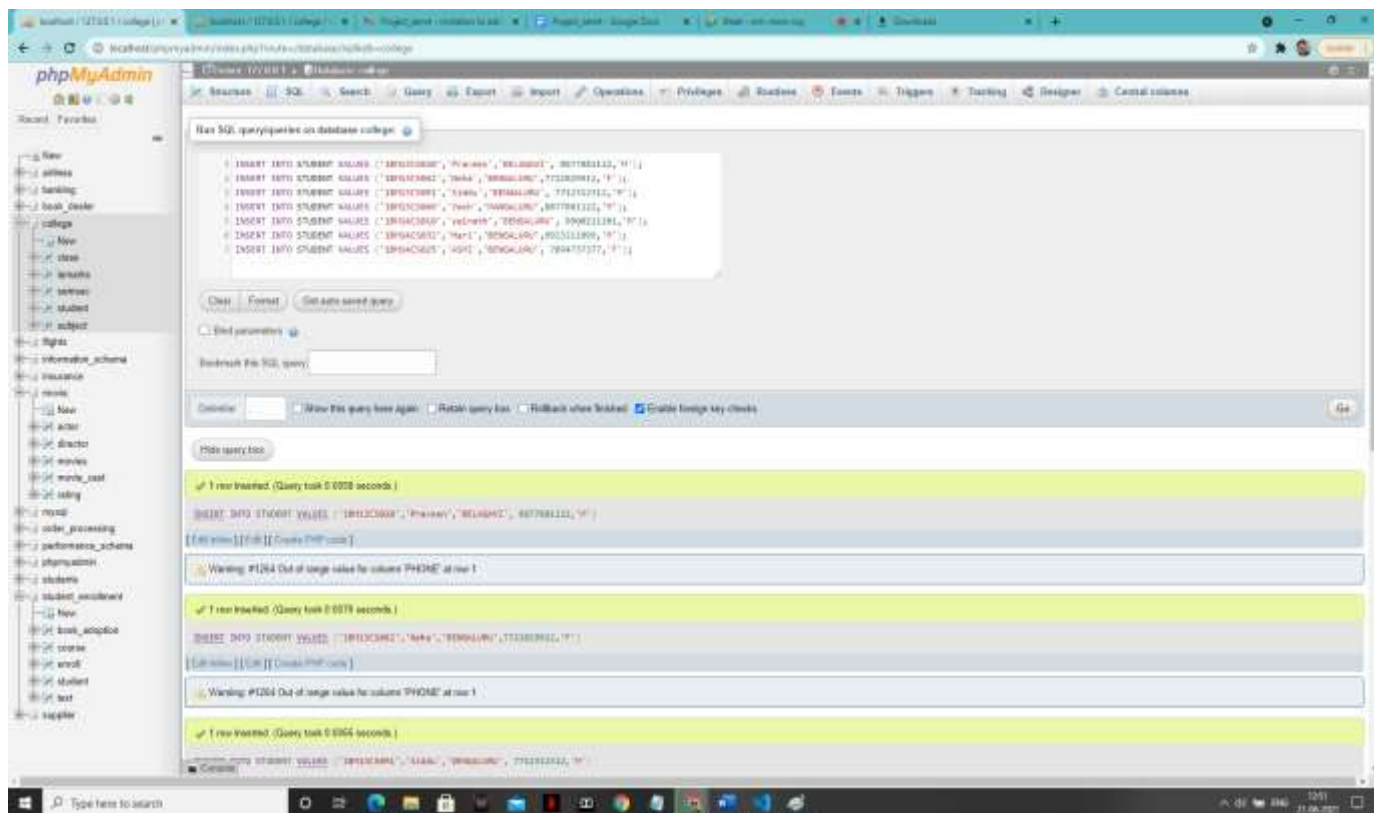
The screenshot shows the phpMyAdmin interface with the Table tab selected. The database 'college' is chosen from the left sidebar. The 'class' table is selected, and its structure is displayed. The table has 12 columns and 1 row. The columns are: ID, SSN, SEX, DOB, ADDRESS, PHONE, TEST1, TEST2, FINALIA, and 7 other columns. The table is a MyISAM table with a primary key on (ID, SSN).

Table	Actions	Rows	Type	Collation	Size	Used
class	Structure   Search   Insert   Empty   Drop	1	MyISAM	utf8mb4_general_ci	12.8 K B	-
employee	Structure   Search   Insert   Empty   Drop	1	MyISAM	utf8mb4_general_ci	10.8 K B	-
section	Structure   Search   Insert   Empty   Drop	1	MyISAM	utf8mb4_general_ci	14.8 K B	-
subject	Structure   Search   Insert   Empty   Drop	1	MyISAM	utf8mb4_general_ci	11.8 K B	-
student	Structure   Search   Insert   Empty   Drop	1	MyISAM	utf8mb4_general_ci	11.8 K B	-
5 tables	Sum	5	MyISAM	utf8mb4_general_ci	128.8 K B	0 B

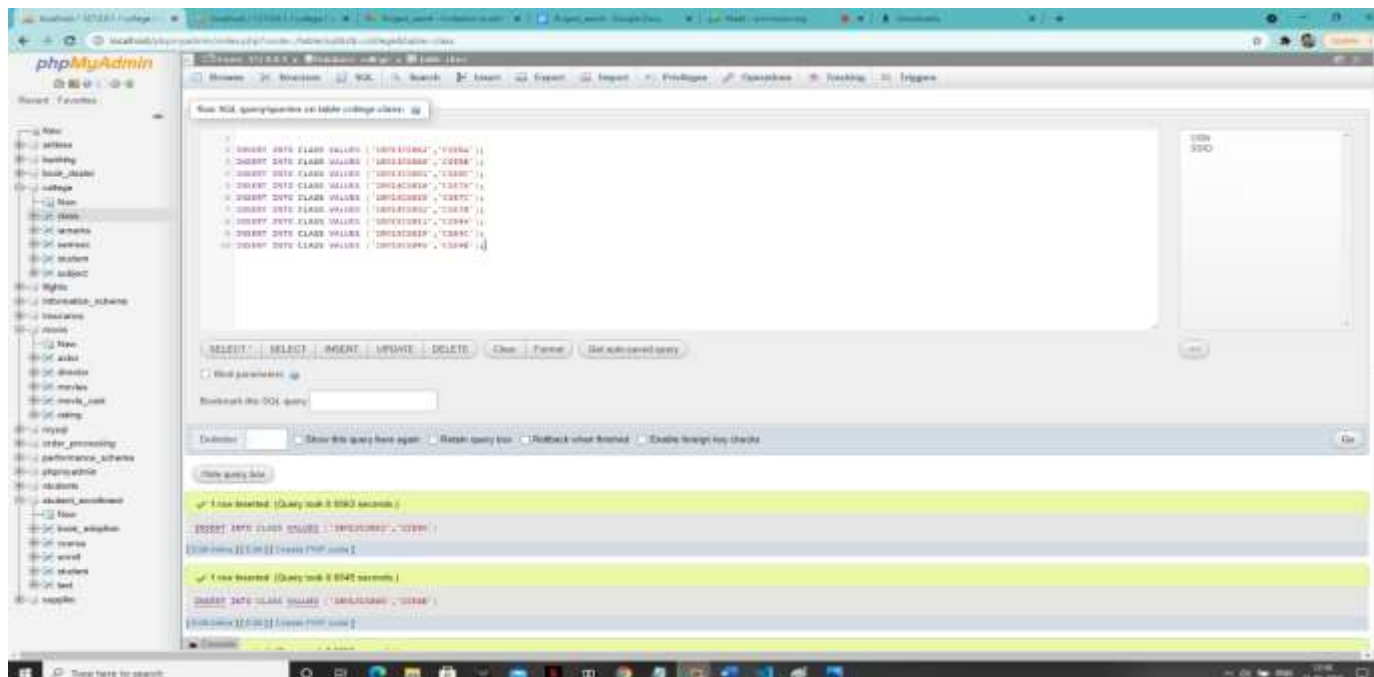


2) Enter tuples for each relation.

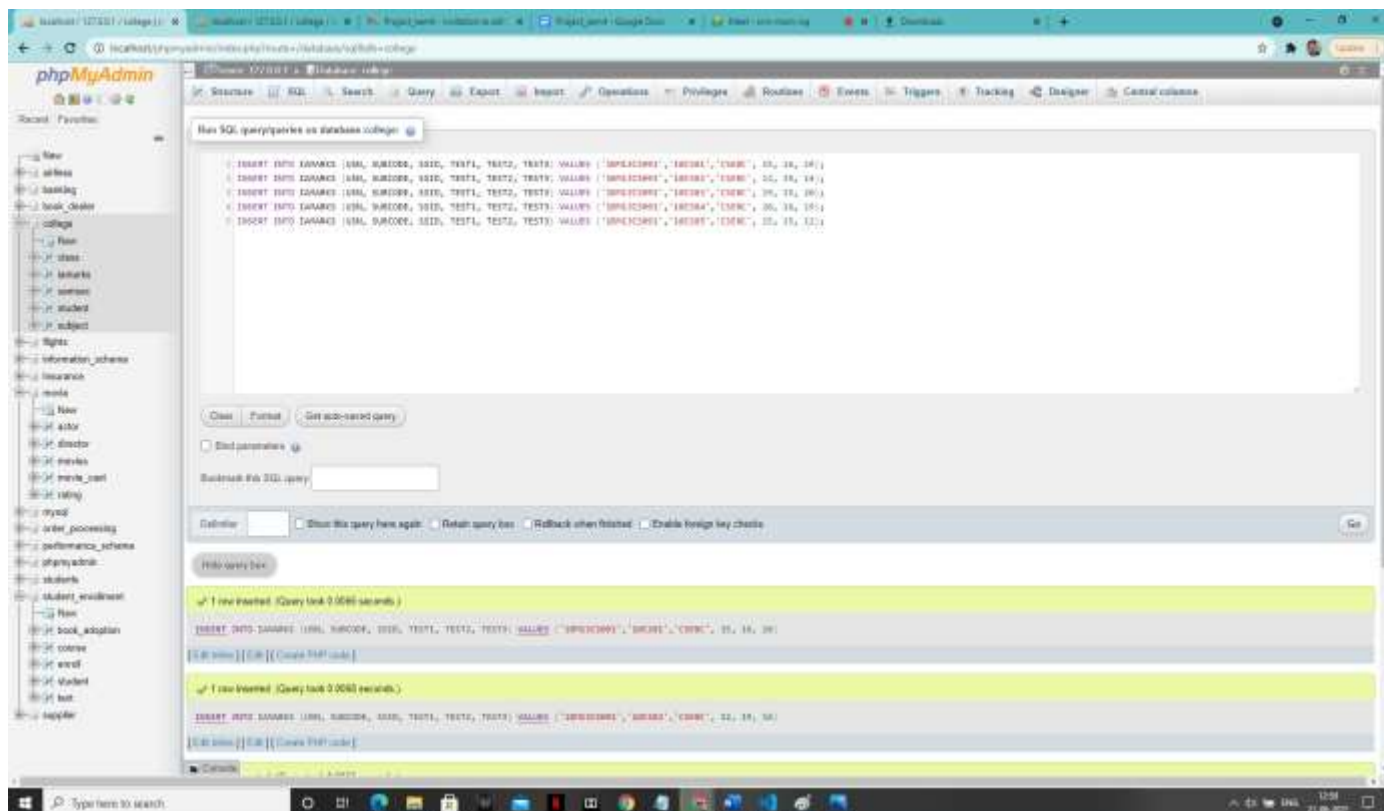
‘STUDENT’ table:



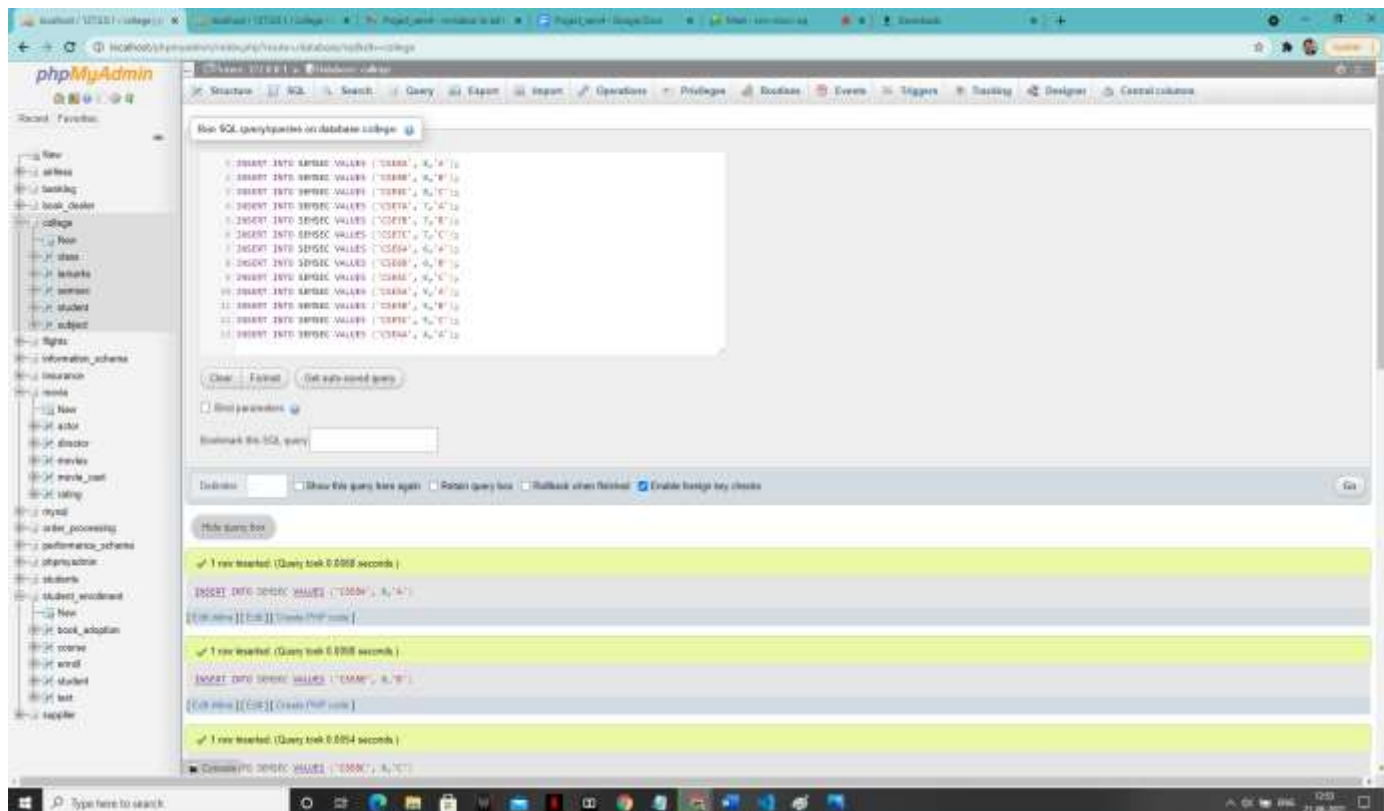
‘Class’ table:



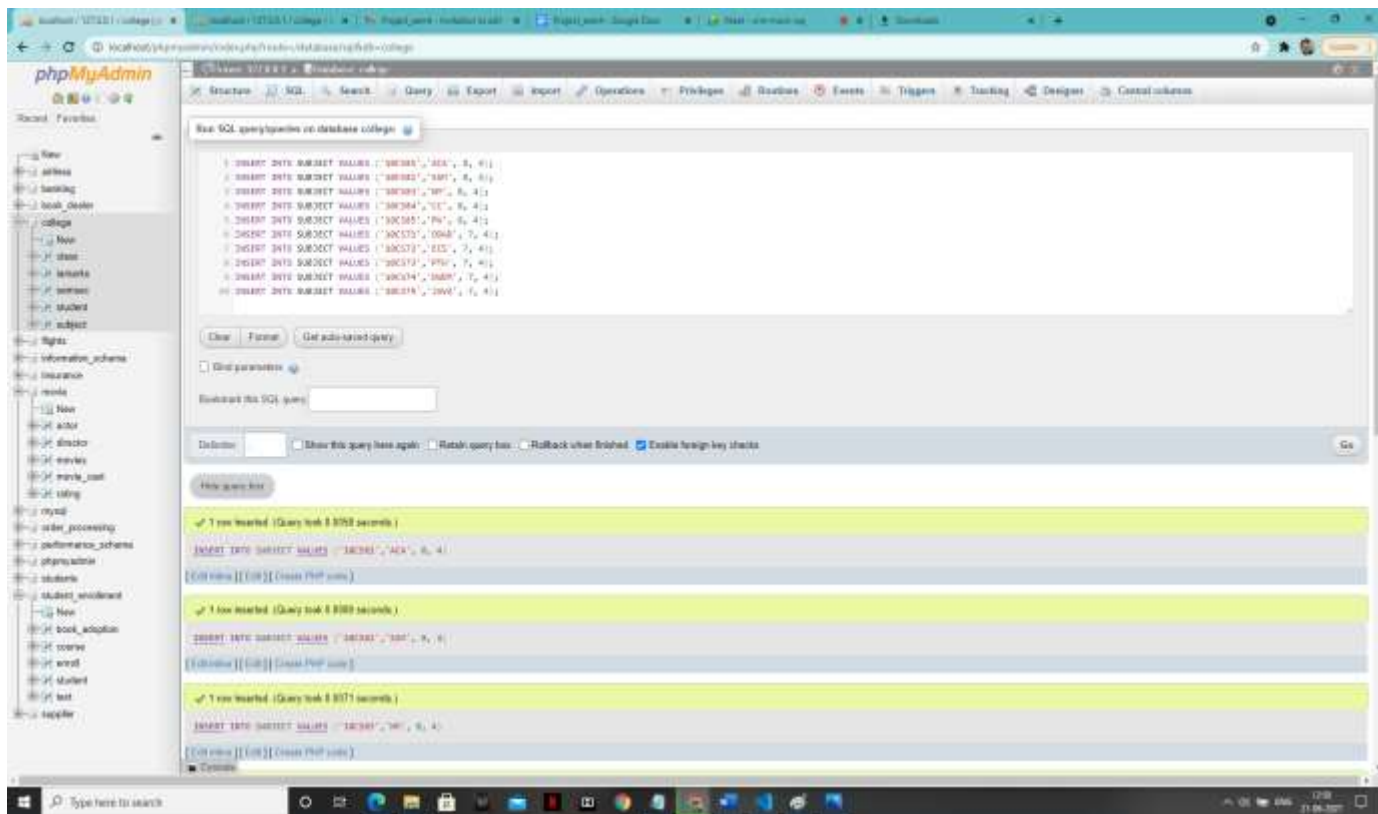
‘Marks’ table:



‘semsec’ value: -

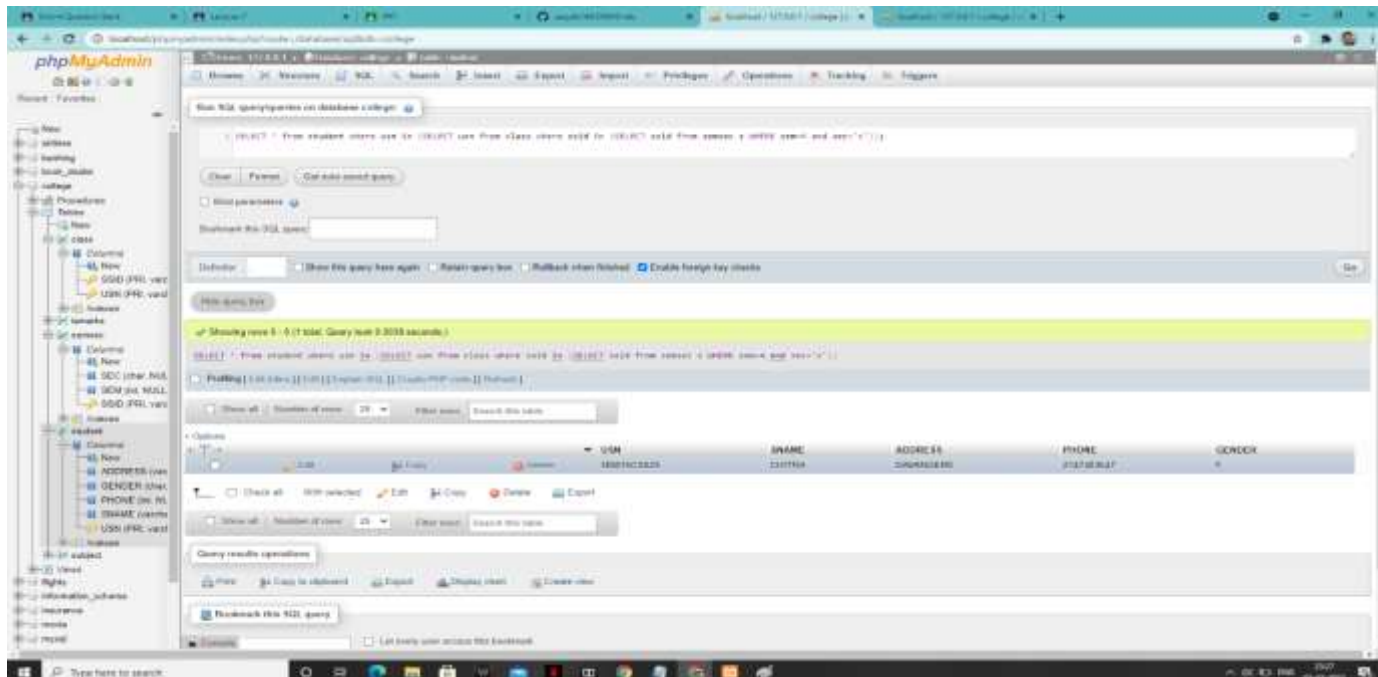


‘subject’ table:



SOLUTION

1:-





2:-

The screenshot shows the phpMyAdmin interface with a SQL query entered in the query box. The query is:

```
SELECT SS.SEP, SS.DET, S.SEMESTER, COUNT(*) AS COUNT
FROM STUDENT S, SEMESTER SS, CLASS C
WHERE S.SEP = C.SEP AND
SS.SEP = C.SEP
GROUP BY SS.SEP, SS.DET, S.SEMESTER
ORDER BY ID;
```

The results are displayed in a table with the following columns: SEP, DET, GENDER, and COUNT. The data is as follows:

SEP	DET	GENDER	COUNT
A	1	M	1
B	1	M	1
C	1	F	1
A	2	F	1
B	2	M	1
C	2	M	1

3:-

The screenshot shows the phpMyAdmin interface with a SQL query entered in the query box. The query is:

```
CREATE VIEW STUDENT_TEST1_VIEW AS SELECT TEST1, SUM(CASE WHEN STUDENT_TEST1_VIEW = 'PASS' THEN 1 ELSE 0 END) AS PASS_COUNT
FROM STUDENT_TEST1_VIEW;
```

The results are displayed in a table with the following columns: TEST1 and PASS\_COUNT. The data is as follows:

TEST1	PASS_COUNT
1	1
2	1
3	1



**5:-**

