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)

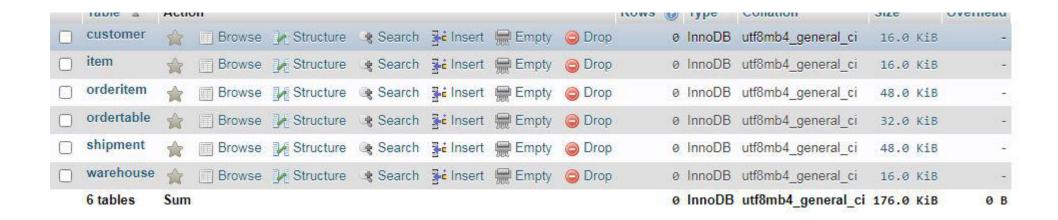
ORDER-ITEM (order #: int, item #: int, qty: int)

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.

```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.2888 seconds.)
create table customer( custNo int PRIMARY KEY, cname varchar(10), city varchar(20) )
Edit inline ] [ Edit ] [ Create PHP code ]
 MySQL returned an empty result set (i.e. zero rows). (Query took 0.1980 seconds.)
create table orderTable( orderNo int primary key, odate date, custNo int, ordAmt int, FOREIGN KEY(custNo) REFERENCES customer(custNo) on
update cascade on delete set NULL )
Edit inline ] [ Edit ] [ Create PHP code ]
 MySQL returned an empty result set (i.e. zero rows). (Query took 0.2680 seconds.)
create table item( itemNo int PRIMARY KEY, unitPrice int )
Edit inline ] [ Edit ] [ Create PHP code ]
 MySQL returned an empty result set (i.e. zero rows). (Query took 0.4220 seconds.)
create table orderItem( orderNo int, itemNo int, qty int, FOREIGN KEY(orderNo) REFERENCES orderTable(orderNo) on update cascade on
delete set NULL, FOREIGN KEY(itemNo) REFERENCES item(itemNo) on update cascade on delete set NULL )
Edit inline [ Edit ] Create PHP code ]
 MySQL returned an empty result set (i.e. zero rows). (Query took 0.2006 seconds.)
create table warehouse( warehouseNo int PRIMARY KEY, city varchar(20) )
[Edit inline][Edit][Create PHP code]
 MySQL returned an empty result set (i.e. zero rows). (Query took 0.2612 seconds.)
create table shipment( orderNo int, warehouseNo int, shipdate date, FOREIGN KEY(orderNo) REFERENCES orderTable(orderNo) on update
cascade on delete set NULL, FOREIGN KEY(warehouseNo) REFERENCES warehouse(warehouseNo) on update cascade on delete set NULL )
Edit inline ] [ Edit ] [ Create PHP code ]
```



ii. Enter at least five tuples for each relation.

```
√ 1 row inserted. (Query took 0.0472 seconds.)

 INSERT INTO `customer` VALUES (1, 'Jake', 'Bangalore')
[ Edit inline ] [ Edit ] [ Create PHP code ]

√ 1 row inserted. (Query took 0.1130 seconds.)

 INSERT INTO `customer` VALUES (2, 'Terry', 'Mumbai')
[ Edit inline ] [ Edit ] [ Create PHP code ]

√ 1 row inserted. (Query took 0.0700 seconds.)

 INSERT INTO `customer` VALUES (3, 'Charles', 'Jaipur')
[ Edit inline ] [ Edit ] [ Create PHP code ]
 1 row inserted. (Query took 0.0370 seconds.)
 INSERT INTO `customer` VALUES (4, 'Rosa', 'Guwahati')
[ Edit inline ] [ Edit ] [ Create PHP code ]

√ 1 row inserted. (Query took 0.0380 seconds.)

 INSERT INTO `customer` VALUES (5, 'Amy', 'Bangalore')
[ Edit inline ] [ Edit ] [ Create PHP code ]
 1 row inserted. (Query took 0.0330 seconds.)
 INSERT INTO `customer` VALUES (6, 'Holt', 'Mumbai')
[Edit inline] [Edit] [Create PHP code]
```

$\leftarrow T \rightarrow$				▼ custNo	cname	city
	<i>⊘</i> Edit	∄ ċ Copy	Delete	1	Jake	Bangalore
		≩≟ Copy	Delete	2	Terry	Mumbai
	@ Edit	≟ Copy	Delete	3	Charles	Jaipur
	Edit	≩ ≟ Copy	Delete	4	Rosa	Guwahati
	Edit	≩ ≟ Copy	Delete	5	Amy	Bangalore
	🧷 Edit	≩- Copy	Delete	6	Holt	Mumbai

```
1 row inserted. (Query took 0.0678 seconds.)
 INSERT INTO `ordertable` VALUES (1,'2021-06-01',1,1000)
[ Edit inline ] [ Edit ] [ Create PHP code ]
 1 row inserted. (Query took 0.1014 seconds.)
 INSERT INTO 'ordertable' VALUES (2,'2021-06-02',1,1500)
[ Edit inline ] [ Edit ] [ Create PHP code ]
 1 row inserted. (Query took 0.0562 seconds.)
 INSERT INTO `ordertable` VALUES (3,'2021-06-03',3,1600)
[ Edit inline ] [ Edit ] [ Create PHP code ]
 1 row inserted. (Query took 0.0412 seconds.)
 INSERT INTO 'ordertable' VALUES (4,'2021-06-04',2,1700)
[ Edit inline ] [ Edit ] [ Create PHP code ]
 1 row inserted. (Query took 0.0532 seconds.)
 INSERT INTO 'ordertable' VALUES (5,'2021-06-05',4,1200)
[ Edit inline ] [ Edit ] [ Create PHP code ]
 1 row inserted. (Query took 0.0618 seconds.)
 INSERT INTO `ordertable` VALUES (6,'2021-06-06',5,1100)
[ Edit inline ] [ Edit ] [ Create PHP code ]
 1 row inserted. (Query took 0.0400 seconds.)
 INSERT INTO 'ordertable' VALUES (7,'2021-06-07',2,1890)
[ Edit inline ] [ Edit ] [ Create PHP code ]
 1 row inserted. (Query took 0.2150 seconds.)
 INSERT INTO `ordertable` VALUES (8,'2021-06-08',4,1999)
Console [ Edit ] [ Create PHP code ]
```

←T→				orderNo	odate	custNo	ordAmt
	@ Edit	⊒- ċ Copy	Delete	1	2021-06-01	1	1000
	Edit	≩ сору	Delete	2	2021-06-02	1	1500
		≩ € Copy	Delete	3	2021-06-03	3	1600
	<i>⊘</i> Edit	≩- Copy	Delete	4	2021-06-04	2	1700
		≩	Delete	5	2021-06-05	4	1200
		∄ € Copy	Delete	6	2021-06-06	5	1100
		≩ € Copy	Delete	7	2021-06-07	2	1890
	Edit	∄ ċ Copy	Delete	8	2021-06-08	4	1999

```
INSERT INTO 'item' VALUES (1,1000)
[ Edit inline ] [ Edit ] [ Create PHP code ]

√ 1 row inserted. (Query took 0.1390 seconds.)

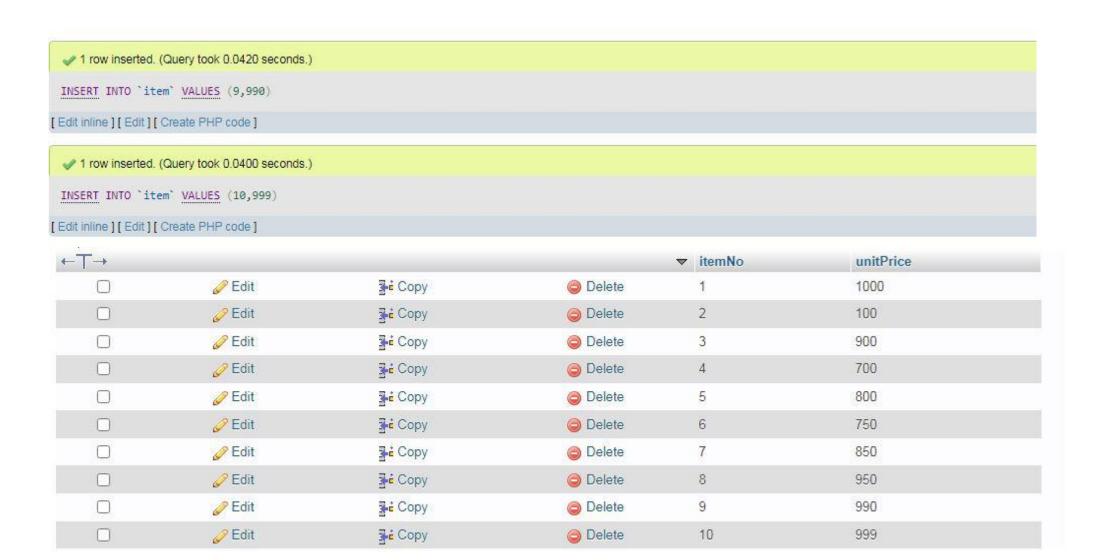
 INSERT INTO `item` VALUES (2,100)
[ Edit inline ] [ Edit ] [ Create PHP code ]
 1 row inserted. (Query took 0.1210 seconds.)
 INSERT INTO 'item' VALUES (3,900)
[ Edit inline ] [ Edit ] [ Create PHP code ]
 1 row inserted. (Query took 0.0312 seconds.)
 INSERT INTO 'item' VALUES (4,700)
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ 1 row inserted. (Query took 0.0426 seconds.)

 INSERT INTO 'item' VALUES (5,800)
[ Edit inline ] [ Edit ] [ Create PHP code ]
 1 row inserted. (Query took 0.0540 seconds.)
 INSERT INTO 'item' VALUES (6,750)
[ Edit inline ] [ Edit ] [ Create PHP code ]

√ 1 row inserted. (Query took 0.0320 seconds.)

 INSERT INTO 'item' VALUES (7,850)
[ Edit inline ] [ Edit ] [ Create PHP code ]
 1 row inserted. (Query took 0.0320 seconds.)
 INSERT INTO 'item' VALUES (8,950)
[ Edit inline ] [ Edit ] [ Create PHP code ]
```



```
INSERT INTO `orderitem` VALUES (1,9,1)
[ Edit inline ] [ Edit ] [ Create PHP code ]
  1 row inserted. (Query took 0.0864 seconds.)
 INSERT INTO `orderitem` VALUES (2,8,1)
[ Edit inline ] [ Edit ] [ Create PHP code ]
  1 row inserted. (Query took 0.0366 seconds.)
 INSERT INTO `orderitem` VALUES (3,7,1)
[ Edit inline ] [ Edit ] [ Create PHP code ]
  1 row inserted. (Query took 0.0156 seconds.)
 INSERT INTO `orderitem` VALUES (4,6,1)
[ Edit inline ] [ Edit ] [ Create PHP code ]
  1 row inserted. (Query took 0.0844 seconds.)
 INSERT INTO `orderitem` VALUES (5,5,1)
[ Edit inline ] [ Edit ] [ Create PHP code ]

√ 1 row inserted. (Query took 0.0156 seconds.)

 INSERT INTO `orderitem` VALUES (6,4,1)
[ Edit inline ] [ Edit ] [ Create PHP code ]
  1 row inserted. (Query took 0.0446 seconds.)
 INSERT INTO `orderitem` VALUES (7,3,1)
[ Edit inline ] [ Edit ] [ Create PHP code ]
  1 row inserted. (Query took 0.0260 seconds.)
 INSERT INTO `orderitem` VALUES (8,10,1)
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

orderNo	itemNo	qty
1	9	1
2	8	1
3	7	1
4	6	1
5	5	1
6	4	1
7	3	1
8	10	1
8	1	1
7	2	8

```
√ 1 row inserted. (Query took 0.0230 seconds.)

 INSERT INTO `warehouse` VALUES (1, 'Bengaluru')
[ Edit inline ] [ Edit ] [ Create PHP code ]

√ 1 row inserted. (Query took 0.1322 seconds.)

 INSERT INTO `warehouse` VALUES (2, 'Mumbai')
[ Edit inline ] [ Edit ] [ Create PHP code ]

√ 1 row inserted. (Query took 0.0366 seconds.)

 INSERT INTO `warehouse` VALUES (3, 'Delhi')
[ Edit inline ] [ Edit ] [ Create PHP code ]

√ 1 row inserted. (Query took 0.0488 seconds.)

 INSERT INTO `warehouse` VALUES (4, 'Chennai')
[ Edit inline ] [ Edit ] [ Create PHP code ]

√ 1 row inserted. (Query took 0.0156 seconds.)

 INSERT INTO `warehouse` VALUES (5, 'Delhi')
[ Edit inline ] [ Edit ] [ Create PHP code ]

√ 1 row inserted. (Query took 0.0376 seconds.)

 INSERT INTO `warehouse` VALUES (6, 'Bengaluru')
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

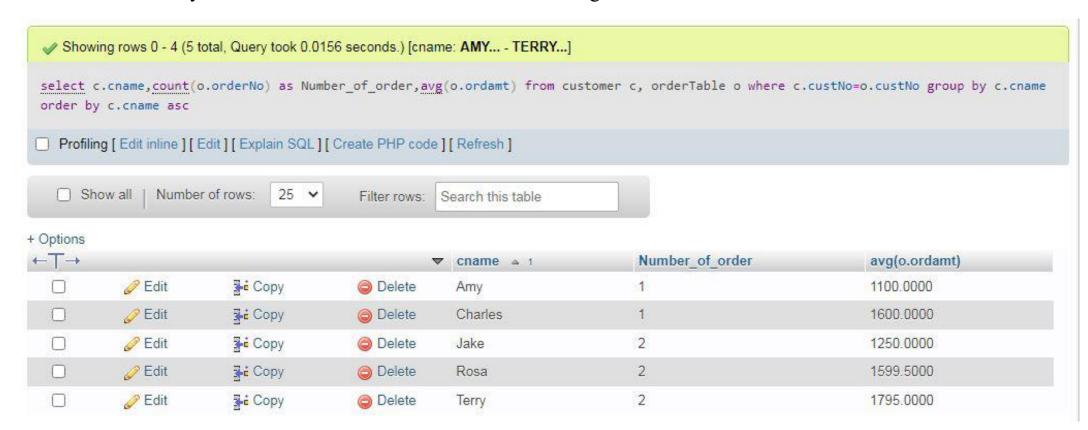
+T→			~	warehouseNo	city
	@ Edit	≩ € Copy	Delete	1	Bengaluru
	<i>⊘</i> Edit	≩≟ Copy	Delete	2	Mumbai
	@ Edit	≩ Copy	Delete	3	Delhi
		≩≟ Copy	Delete	4	Chennai
	<i> Edit</i>	≩≟ Copy	Delete	5	Delhi
		≩≟ Copy	Delete	6	Bengaluru

```
1 row inserted. (Query took 0.0420 seconds.)
 INSERT INTO `shipment` VALUES (1,5,'2021-06-08')
[ Edit inline ] [ Edit ] [ Create PHP code ]
  1 row inserted. (Query took 0.1022 seconds.)
 INSERT INTO `shipment` VALUES (2,2,'2021-06-08')
[ Edit inline ] [ Edit ] [ Create PHP code ]
  1 row inserted. (Query took 0.0400 seconds.)
 INSERT INTO `shipment` VALUES (3,6,'2021-06-08')
[ Edit inline ] [ Edit ] [ Create PHP code ]
  1 row inserted. (Query took 0.0250 seconds.)
 INSERT INTO `shipment` VALUES (4,1,'2021-06-08')
[ Edit inline ] [ Edit ] [ Create PHP code ]
  1 row inserted. (Query took 0.0540 seconds.)
 INSERT INTO `shipment` VALUES (5,6,'2021-06-08')
[ Edit inline ] [ Edit ] [ Create PHP code ]
  1 row inserted. (Query took 0.0350 seconds.)
 INSERT INTO `shipment` VALUES (6,5,'2021-06-08')
[ Edit inline ] [ Edit ] [ Create PHP code ]
  1 row inserted. (Query took 0.0410 seconds.)
 INSERT INTO `shipment` VALUES (7,4,'2021-06-08')
[ Edit inline ] [ Edit ] [ Create PHP code ]
  1 row inserted. (Query took 0.0360 seconds.)
 TNCCOT TNTO `shipment` VALUES (8,3,'2021-06-08')
```

+ Options

orderNo	warehouseNo	shipdate
1	5	2020-06-08
2	2	2020-06-08
3	6	2020-06-08
4	1	2020-06-08
5	6	2020-06-08
6	5	2020-06-08
7	4	2020-06-08
8	3	2020-06-08

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.



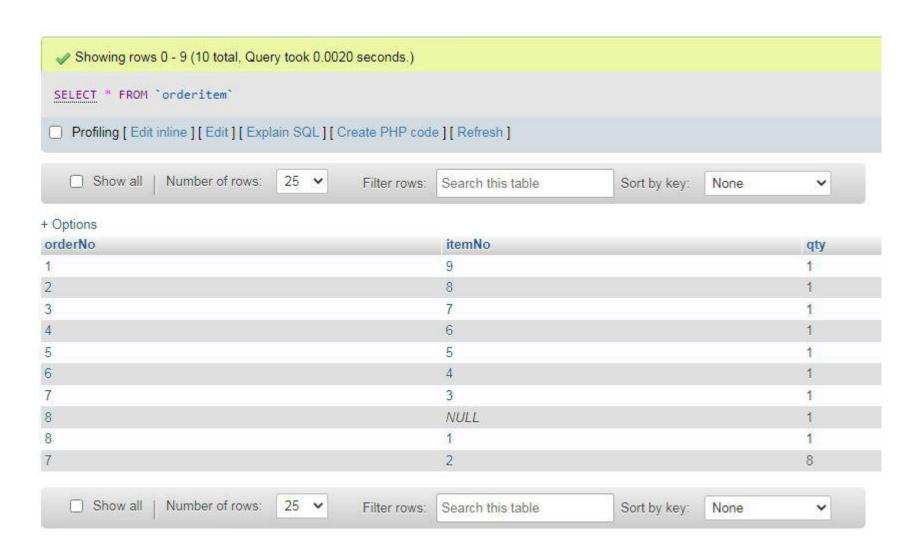
iv. List the order# for orders that were shipped from all warehouses that the company has in a specific city.



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

On delete set null used at the end of foreign key definiton while creating order item table hence the item number deleted will automatically be set to null in order item delete from item where itemNo=10

[Edit inline][Edit][Create PHP code]



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)

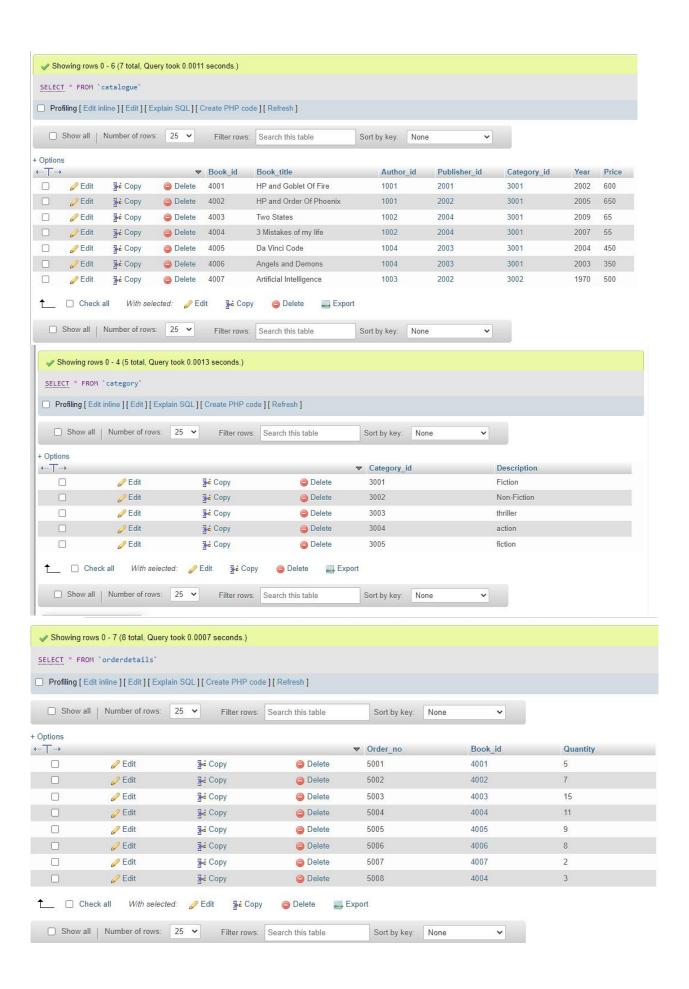
CATALOG (book-id: int, title: String, author-id: int, publisher-id: int, category-id: int, year: int, price: int)

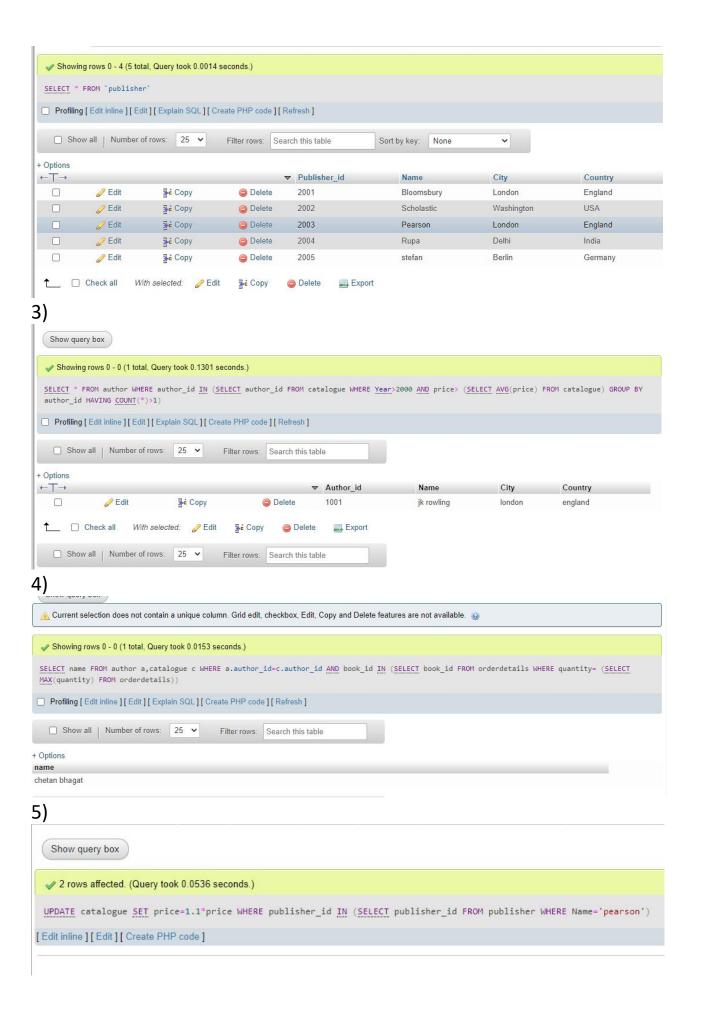
CATEGORY(category-id: int, description: String)

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%.







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.

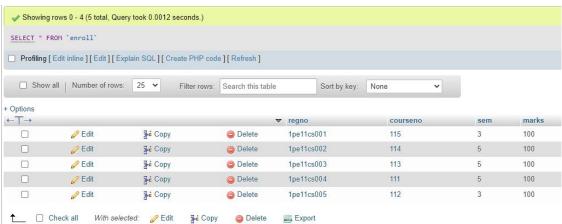
BOOK ADOPTION



COUSRE



ENROLL

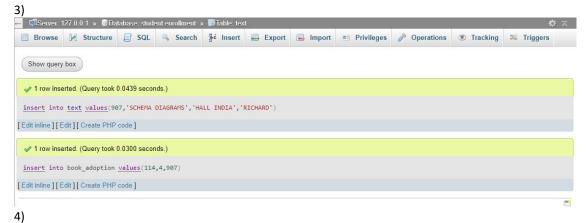


STUDENT



TEXT





✓ Showing rows 0 - 2 (3 total, Query took 0.0148 seconds.) [book_title: DATABASE SYSTEMS... - SCHEDULING...]

SELECT c.courseno,t.book_isbn,t.book_title FROM course c,book_adoption ba,text t WHERE c.courseno=ba.courseno AND ba.book_isbn=t.book_isbn AND c.dept='CSE' AND 2<(SELECT COUNT(book_isbn) FROM book_adoption b WHERE c.courseno=b.courseno) ORDER BY t.book_title

**The course of the course of th

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all Number of rows:	25 V Filter rows: Search this table	e
- Options		
courseno	book_isbn	book_title 🔺 1
111	904	DATABASE SYSTEMS
111	900	OPERATING SYS
111	903	SCHEDULING

5)



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)

ATING(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.

ACTOR

	~		ACTID	ACTNAME	ACTGENDER	
@ Edit	⊒ Copy	Delete	301	anuskha	f	
	≩ ≟ Copy	Delete	302	prabhas	m	
<i> </i>	⊒ Copy	Delete	303	punith	m	
	≩ ≟ Copy	Delete	304	sudeep	m	
@ Edit	≩	Delete	305	akshay	m	
	Ø Edit Ø Edit Ø Edit	② Edit 3ª É Copy	Pedit Pedit Pedit	Ø Edit ☐ Delete ☐ Delete ☐ 303 ☐ Delete ☐ Delete	Ø Edit ☐ Edit	Ø Edit ☐ Edit ☐ Copy ☐ Delete ☐ Sopy ☐ Delete ☐ Delete ☐ Sopy ☐ Delete ☐ Delete ☐ Sopy ☐ Delete ☐ Delete ☐ Delete ☐ Delete ☐ Delete ☐ Sopy ☐ Delete ☐ De

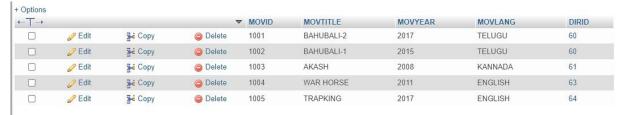
DIRECTOR

+ Options						
←T→				▼ DIRID	DIRNAME	DIRPHONE
	@ Edit	≩ copy	Delete	60	RAJAMOULI	919191919
	<i>⊘</i> Edit	≩ å Copy	Delete	61	нітснсоск	818181818
	@ Edit	≩ å Copy	Delete	62	FARAN	717171717
	Edit	₫ Copy	Delete	63	STEVEN SPIELBERG	616161616
		3- Copy	Delete	64	MANIRATHNAM	919191919

MOVIECAST

- Options							
$\leftarrow T \rightarrow$				▼ ACTID	MOVID	ROLE	
	<i></i> € Edit	≩ Copy	Delete	301	1001	HEROINE	
		∄ ċ Copy	Delete	301	1002	HEROINE	
		Z Copy	Delete	303	1002	GUEST	
	<i>⊘</i> Edit	≩≟ Copy	Delete	303	1003	HERO	
	<i>❷</i> Edit	3 - Copy	Delete	304	1004	HERO	

MOVIES



RATING



3)

ACTNAME

anuskha





MOVYEAR

2017

MOVTITLE

BAHUBALI-2



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) IAMARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)

Write SQL queries to

- 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 '1BI15CS101' in all subjects.

Table class:

∇	USN	SSID
	1RN13CS020	CSE8A
	1RN13CS062	CSE8A
	1RN13CS066	CSE8B
	1RN13CS091	CSE8C
	1RN14CS010	CSE7A
	1RN14CS025	CSE7A
	1RN14CS032	CSE7A
	1RN15CS011	CSE4A
	1RN15CS029	CSE4A
	1RN15CS045	CSE4B
	1RN15CS091	CSE4C
	1RN16CS045	CSE3A
	1RN16CS088	CSE3B
	1RN16CS122	CSE3C

Table iamarks

USN	SUBCODE	SSID	TEST1	TEST2	TEST3	FINALIA
1RN13CS091	10CS81	CSE8C	15	16	18	NULL
1RN13CS091	10CS82	CSE8C	12	19	14	NULL
1RN13CS091	10CS83	CSE8C	19	15	20	NULL
1RN13CS091	10CS84	CSE8C	20	16	19	NULL
1RN13CS091	10CS85	CSE8C	15	15	12	NULL







Table semsec

Table 30	111300		
	SSID	SEM	SEC
	CSE1C	1	С
	CSE2A	2	A
	CSE2B	2	В
	CSE2C	2	С
	CSE3A	3	A
	CSE3B	3	В
	CSE3C	3	С
	CSE4A	4	A
	CSE4B	4	В
	CSE4C	4	С
	CSE5A	5	A
	CSE5B	5	В
	CSE5C	5	С
	CSE6A	6	A
	CSE6B	6	В
	CSE6C	6	C
	CSE7A	7	A
	CSE7B	7	В
	CSE7C	7	С
	CSE8A	8	A
	CSE8B	8	В
	CSE8C	8	C

Table student

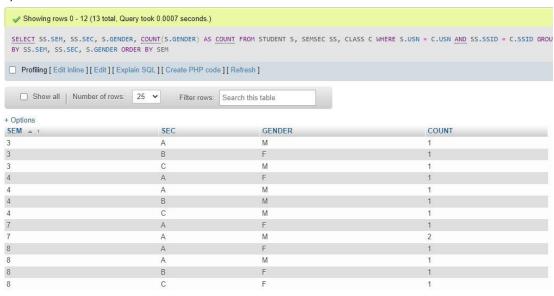
USN	SNAME	ADDRESS	PHONE	GENDER
1RN13CS020	AKSHAY	BELAGAVI	2147483647	M
1RN13CS062	SANDHYA	BENGALURU	2147483647	F
1RN13CS066	SUPRIYA	MANGALURU	2147483647	F
1RN13CS091	TEESHA	BENGALURU	2147483647	F
1RN14CS010	ABHAY	BENGALURU	2147483647	M
1RN14CS025	ASMI	BENGALURU	2147483647	F
1RN14CS032	BHASKAR	BENGALURU	2147483647	M
1RN15CS011	AJAY	TUMKUR	2147483647	M
1RN15CS029	CHITRA	DAVANGERE	2147483647	F
1RN15CS045	JEEVA	BELLARY	2147483647	M
1RN15CS091	SANTOSH	MANGALURU	2147483647	M
1RN16CS045	ISMAIL	KALBURGI	2147483647	M
1RN16CS088	SAMEERA	SHIMOGA	2147483647	F
1RN16CS122	VINAYAKA	CHIKAMAGALUR	2147483647	M

Table subject

Export Import	Privileges	<i>→</i> Operations	36 Triggers	
SUBCODE 10CS71	TITLE	SEM 7	CREDITS 4	
10CS72	ECS	7	4	
10CS73	PTW	7	4	
10CS74	DWDM	7	4	
10CS75	JAVA	7	4	
10CS76	SAN	7	4	
10CS81	ACA	8	4	
10CS82	SSM	8	4	
10CS83	NM	8	4	
10CS84	CC	8	4	
10CS85	PW	8	4	
15CS31	M3	3	4	
15CS32	ADE	3	4	
15CS33	DSA	3	4	
15CS34	со	3	4	
15CS35	USP	3	3	
15CS36	DMS	3	3	
15CS41	M4	4	4	
15CS42	SE	4	4	
15CS43	DAA	4	4	
15CS44	MPMC	4	4	
15CS45	000	4	3	
150946	DC .	И	2	

1)





3)

Show query box

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0668 seconds.)

CREATE VIEW STU_TEST1_MARKS_VIEW AS SELECT TEST1, SUBCODE FROM IAMARKS WHERE USN = '1BI15CS101'

[Edit inline] [Edit] [Create PHP code]