

# **DBMS LAB REPORT**

## **INTERNAL-S-1**

**USN-1BM19CS079**

**NAME-LIKITHA B**

## **DBMS LAB PROGRAM-1**

### **INSURANCE DATABASE**

#### **QUESTION-**

Consider the Insurance database given below. The primary keys are underlined and the

data types are specified.

**PERSON (driver-id #: String, name: String, address: String)**

**CAR (Regno: String, model: String, year: int)**

**ACCIDENT (report-number: int, adate: date, location: String)**

**OWNS (driver-id #: String, Regno: String)**

**PARTICIPATED (driver-id: String, Regno: String, report-number: int, damage-amount: 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. Demonstrate how you
  - a. Update the damage amount for the car with a specific Regno in the accident with report number 12 to  
25000.
  - b. Add a new accident to the database.
- iv. Find the total number of people who owned cars that involved in accidents in 2008.
- v. Find the number of accidents in which cars belonging to a specific model were involved.

## **PROGRAM CODE-**

```
1 •  create database Insurance;
2 •  show databases;
3 •  use Insurance;
4 •  show tables;
5 •  create table PERSON(driverid varchar(20),dname varchar(20),address varchar(40),primary key(driverid));
6 •  desc PERSON;
7 •  create table CAR(regno varchar(10),model varchar(10),year int ,primary key(regno));
8 •  desc CAR;
9 •  create table ACCIDENT(report_no int,adate date,location varchar(20),primary key(report_no));
10 • desc ACCIDENT;
11 •  create table OWNS(driverid varchar(10),regno varchar(10),primary key(driverid,regno),
12   foreign key(driverid) references PERSON(driverid) on delete cascade,
13   foreign key(regno) references CAR(regno) on delete cascade);
14 •  CREATE TABLE PARTICIPATED(driverid varchar(10),regno varchar(10),report_no int,
15   damage_amt float, foreign key (driverid,regno) references OWNS(driverid,regno)
16   ON DELETE CASCADE,foreign key (REPORT_NO) references ACCIDENT(REPORT_NO) ON DELETE CASCADE);
17 •  desc PARTICIPATED;
18 •  insert into PERSON values('1111','Ramu','K.S.LAYOUT');
19 •  commit;
20 •  select* FROM PERSON;
21 •  insert into PERSON values('2222','John','INDIRANAGAR');
22 •  insert into PERSON values('3333','Priya','JAYANAGAR');
23 •  insert into PERSON values('4444','Gopal','WHITEFIELD');
24 •  insert into PERSON values('5555','Latha','VIJAYNAGAR');
25 •  commit;
26 •  insert into CAR values('KA04Q2301','MARUTHI-DX', 2000);
27 •  insert into CAR values('KA05P1000', 'FORDICON', 2000);
28 •  insert into CAR values('KA03L1234',' ZEN-VXI',1999);
29 •  insert into CAR values('KA03L9999', 'MARUTHI-DX', 2002);
30 •  insert into CAR values('KA01P4020', 'INDICA-VX', 2002);
31 •  commit;
32 •  select * from CAR;
```

```

31 •    commit;
32 •    select * from CAR;
33
34 •    insert into ACCIDENT values(12,'2002-06-01', 'M G ROAD');
35 •    insert into ACCIDENT values(200, '2002-12-10', 'DOUBLEROAD');
36 •    insert into ACCIDENT values(300, '1999-07-23', 'M G ROAD');
37 •    insert into ACCIDENT values(25000, '2000-06-11', 'RESIDENCY ROAD');
38 •    insert into ACCIDENT values(26500, '2001-10-16', 'RICHMOND CIRCLE');
39 •    commit;
40 •    select * from ACCIDENT;
41
42 •    insert into OWNS values('1111','KA04Q2301');
43 •    insert into OWNS values('1111', 'KA05P1000');
44 •    insert into OWNS values('2222', 'KA03L1234');
45 •    insert into OWNS values('3333', 'KA03L9999');
46 •    insert into OWNS values('4444', 'KA01P4020');
47 •    commit;
48 •    select * from OWNS;
49
50 •    insert into PARTICIPATED values('1111', 'KA04Q2301', 12 ,20000);
51 •    insert into PARTICIPATED values('2222', 'KA03L1234', 200, 500);
52 •    insert into PARTICIPATED values('3333', 'KA03L9999', 300, 10000);
53 •    insert into PARTICIPATED values('4444', 'KA01P4020', 25000 ,2375);
54 •    insert into PARTICIPATED values('1111', 'KA05P1000', 26500 ,70000);
55
56 •    UPDATE PARTICIPATED SET DAMAGE_AMT=25000 WHERE REPORT_NO =12 AND REGNO='KA04Q2301';
57 •    select * from PARTICIPATED;
58
59 •    select count(*) from ACCIDENT where Adate like '2002-__-__';
60 •    select count(A.Report_no) from ACCIDENT A, PARTICIPATED P, CAR C where
61      A.Report_no=P.Report_no AND P.Regno=C.Regno AND C.Model="MARUTHI-DX";
62

```

---

## OUTPUT-

## SHOW TABLES:

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

sys

LAB PROGRAM-1(INSURANCE) - SQL

```
57 • select * from PARTICIPATED;
58
59 • select count(*) from ACCIDENT where Adate like '2002-%-%';
60 • select count(A.Report_no) from ACCIDENT A, PARTICIPATED P, CAR C where
A.Report_no=P.Report_no AND P.Regno=C.Regno AND C.Model="MARUTHI-0X";
62 • select*from accident;
63 • show tables;
64
65
```

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

Tables\_in\_insurance

- accident
- car
- owns
- participated
- person

Administration Schemas Information

No object selected

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

insurance

Tables

- accident
- car
- owns
- participated
- person

Views

Stored Procedures

Functions

sys

LAB PROGRAM-1(INSURANCE) - SQL

```
1 • SELECT * FROM insurance.accident;
```

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

report_no	adate	location
12	2002-06-01	M G ROAD
200	2002-12-10	DOUBLEROAD
300	1999-07-23	M G ROAD
25000	2000-06-11	RESIDENCY ROAD
26500	2001-10-16	RICHMOND CIRCLE
NULL	NULL	NULL

Administration Schemas Information

Table: accident

Columns:

report_no	int PK
adate	date
location	varchar(20)

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

insurance

Tables

accident

Columns

report\_no  
adate  
location

Indexes  
Foreign Keys  
Triggers

car  
owns  
participated  
person

Views  
Stored Procedures  
Functions

sys

Administration Schemas

Information

Column: report\_no

LAB PROGRAM-1(INSURANCE)

1 • SELECT \* FROM insurance.car;

Execute the selected portion of the script or everything, if there is no selection

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

regno	model	year
KA01P4020	INDICA-VX	2002
KA03L1234	ZEN-VXI	1999
KA03L9999	MARUTHI-OX	2002
KA04Q2301	MARUTHI-OX	2000
KA05P1000	FORDICON	2000
NULL	NULL	NULL

Result Grid Form Editor Field Types

Result Grid

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

insurance

Tables

accident

Columns

report\_no  
adate  
location

Indexes  
Foreign Keys  
Triggers

car  
owns  
participated  
person

Views  
Stored Procedures  
Functions

sys

Administration Schemas

Information

Table: owns

Columns:

drivervid regno

4444 KA01P4020

2222 KA03L1234

3333 KA03L9999

1111 KA04Q2301

1111 KA05P1000

NULL NULL

owns 3

Result Grid Form Editor Field Types

Result Grid

MySQL Workbench

Local instance MySQL80 ×

Edit View Query Database Server Tools Scripting Help

Navigator LAB PROGRAM-1(INSURANCE) ×

HEMAS

Filter objects

insurance

Tables

accident

Columns

report\_no

adate

location

Indexes

Foreign Keys

Triggers

car

owns

participated

person

Views

Stored Procedures

Functions

sys

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

driverid	regno	report_no	damage_amt
1111	KA04Q2301	12	25000
2222	KA03L1234	200	500
3333	KA03L9999	300	10000
4444	KA01P4020	25000	2375
1111	KA05P1000	26500	70000

Result Grid Form Editor Field Types

Table: participated

Columns:

participated 4 × Read Only

MySQL Workbench

Local instance MySQL80 ×

Edit View Query Database Server Tools Scripting Help

Navigator LAB PROGRAM-1(INSURANCE) ×

SCHEMAS

Filter objects

insurance

Tables

accident

Columns

report\_no

adate

location

Indexes

Foreign Keys

Triggers

car

owns

participated

person

Views

Stored Procedures

Functions

sys

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

driverid	dname	address
1111	Ramu	K.S.LAYOUT
2222	John	INDIRANAGAR
3333	Priya	JAYANAGAR
4444	Gopal	WHITEFIELD
5555	Latha	VIJAYNAGAR
6666	Shyam	BBM

Result Grid Form Editor Field Types

Table: person

Columns:

person 5 × Apply Revert

## **DBMS LAB PROGRAM-2**

### **BOOK DEALER DATABASE**

#### **QUESTION-**

**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 CODE-**

```

1 •  create database book_dealer;
2 •  use book_dealer;
3 •  create table author(author_id int,name varchar(15),city varchar(15),country varchar(15),primary key(author_id));
4 •  create table publisher(publisher_id int,name varchar(15),city varchar(15),country varchar(15),primary key(publisher_id));
5 •  create table category(category_id int,description varchar(15),primary key(category_id));
6 •  create table catalog(book_id int,title varchar(15),author_id int,publisher_id int,category_id int,year int,price int,primary key(book_id),
7   foreign key(author_id)references author(author_id) on delete cascade,
8   foreign key(publisher_id)references publisher(publisher_id) on delete cascade,
9   foreign key(category_id)references category(category_id) on delete cascade);
10 •  create table order_details(order_no int,book_id int,quantity int,primary key(order_no),
11   foreign key(book_id)references catalog(book_id) on delete cascade);
12 •  insert into author values('1001','TERAS CHAIN','CA','USA');
13 •  insert into author values('1002','STEVENS','ZOMBI','UGANDA');
14 •  insert into author values('1003','M MANO','CAIR','CANADA');
15 •  insert into author values('1004','KARTHIK B.P.','NEW YORK','USA');
16 •  insert into author values('1005','WILLIAM','LAS VEGAS','USA');
17 •  commit;
18 •  select*from author;
19 •  insert into publisher values('101','PEARSON','NEW YORK','USA');
20 •  insert into publisher values('102','EEE','NEW SOUTHVALES','USA');
21 •  insert into publisher values('103','PHI','DELHI','INDIA');
22 •  insert into publisher values('104','WILLEY','BERLIN','GERMANY');
23 •  insert into publisher values('105','MGH','NEW YORK','USA');
24 •  commit;
25 •  select*from publisher;
26 •  insert into category values('1011','CSE');
27 •  insert into category values('1022','ADA');
28 •  insert into category values('1033','ELECTRONICS');
29 •  insert into category values('1044','PROGRAMMING');
30 •  insert into category values('1055','OS');
31 •  commit;

27 •  insert into category values('1022','ADA');
28 •  insert into category values('1033','ELECTRONICS');
29 •  insert into category values('1044','PROGRAMMING');
30 •  insert into category values('1055','OS');
31 •  commit;
32 •  select*from category;
33 •  insert into catalog values('11','UNIX SYSTEM','1001','101','1011','1998','235');
34 •  insert into catalog values('12','DIG ANALYSIS','1001','102','1033','1997','255');
35 •  insert into catalog values('13','LOGIC DESIGN','1002','103','1022','2001','352');
36 •  insert into catalog values('14','SERVER','1002','103','1011','2002','523');
37 •  insert into catalog values('15','LINUX OS','1003','104','1044','2003','124');
38 •  commit;
39 •  select*from catalog;
40 •  insert into order_details values('1','11','12');
41 •  insert into order_details values('1','12','2');
42 •  insert into order_details values('2','12','15');
43 •  insert into order_details values('3','13','22');
44 •  insert into order_details values('4','13','14');
45 •  insert into order_details values('5','15','7');
46 •  commit;
47 •  select*from order_details;
48
49 •  select a.author_id,a.name,a.city,a.country,c.price,year from author a join catalog c on a.author_id=c.author_id where c.year>=2000 group by a;
50
51 •  select name from author a where a.author_id=
52   select c.author_id from catalog c where book_id=(select book_id from order_details od where quantity=(select max(quantity) from order_details
53
54 •  update catalog set price=1.1*price where publisher_id=(select publisher_id from publisher where name='PEARSON');
55 •  select*from catalog;
56

```

## OUTPUT-

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS book\_dealer

Tables: author, catalog, category, order\_details, publisher

Views, Stored Procedures, Functions

sys

LAB PROGRAM-2(BOOK DEAL) SQL File 3\* author

1 • SELECT \* FROM book\_dealer.author;

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types

author_id	name	city	country
1001	TERAS CHAIN	CA	USA
1002	STEVENS	ZOMBI	UGANDA
1003	M MANO	CAIR	CANADA
1004	KARTHIK B.P.	NEW YORK	USA
1005	WILLIAM	LAS VEGAS	USA
*	NULL	NULL	NULL

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS book\_dealer

Tables: catalog

Views, Stored Procedures, Functions

sys

LAB PROGRAM-2(BOOK DEAL) SQL File 3\* catalog

1 • SELECT \* FROM book\_dealer.catalog;

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types

book_id	title	author_id	publisher_id	category_id	year	price
11	UNIX SYSTEM	1001	101	1011	1998	235
12	DIG ANALYSIS	1001	102	1033	1997	255
13	LOGIC DESIGN	1002	103	1022	2001	352
14	SERVER	1002	103	1011	2002	523
15	LINUX OS	1003	104	1044	2003	124
*	NULL	NULL	NULL	NULL	NULL	NULL

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS book\_dealer

Tables: author, catalog, category, order\_details, publisher

Views, Stored Procedures, Functions

sys

Administration Schemas Information

Schema: book\_dealer

LAB PROGRAM-2(BOOK DEALER) SQL File 3\* author catalog catalog category

1 • SELECT \* FROM book\_dealer.category;

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

category_id	description
1011	CSE
1022	ADA
1033	ELECTRONICS
1044	PROGRAMMING
1055	OS
1066	NULL

category 1 x Apply Revert

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS book\_dealer

Tables: author, catalog, category, order\_details, publisher

Views, Stored Procedures, Functions

sys

Administration Schemas Information

Schema: book\_dealer

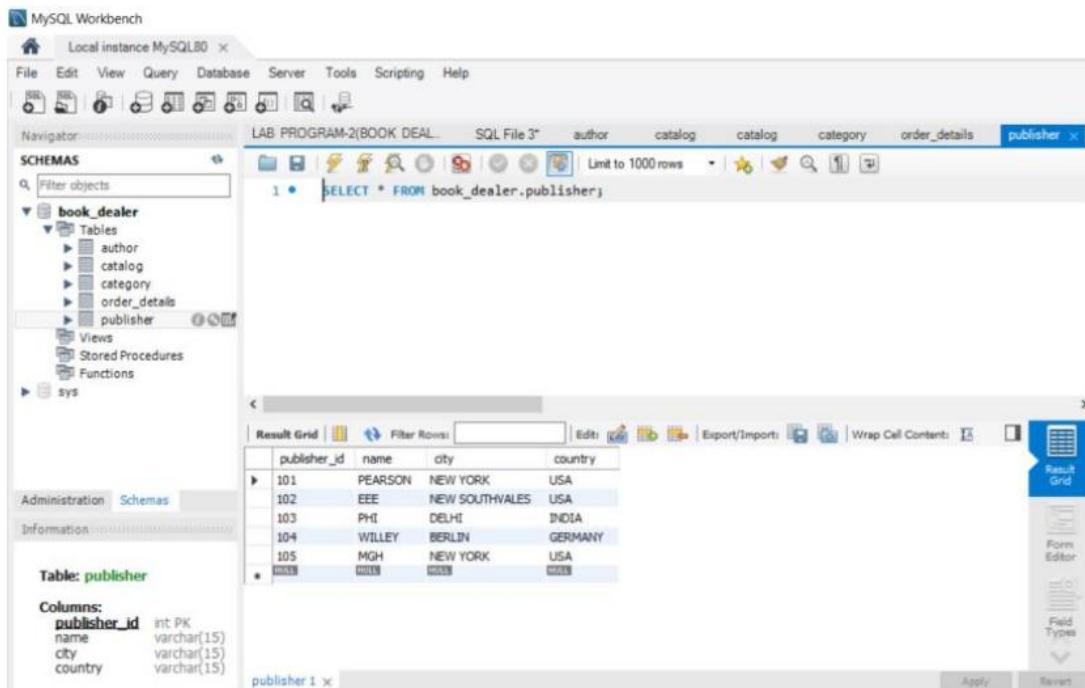
LAB PROGRAM-2(BOOK DEALER) SQL File 3\* author catalog catalog category order\_details

1 • SELECT \* FROM book\_dealer.order\_details;

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

order_no	book_id	quantity
1	11	12
*	NULL	NULL

order\_details 1 x Apply Revert



## DBMS LAB PROGRAM-3

### ORDER PROCESSING DATABASE

#### QUESTION-

II. 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.**

**PROGRAM CODE-**

```

1 •  create database Order_processing;
2 •  use Order_processing;
3 •  CREATE TABLE CUSTOMER
4  (
5     cust_no int,
6     cname VARCHAR(15),
7     city VARCHAR(15),
8     PRIMARY KEY(cust_no)
9 );
10 •  CREATE TABLE ORDERS(
11     order_no int,
12     odate date,
13     cust_no int,
14     foreign key(cust_no) references CUSTOMER(cust_no) on delete cascade,
15     ord_Amt int,
16     primary key(order_no)
17 );
18 •  create table ITEM (
19     item_no int,
20     unit_price int,
21     primary key(item_no)
22 );
23 •  create table ORDER_ITEM (
24     order_no int,
25     item_no int,
26     qty int,
27     foreign key(order_no) references ORDERS(order_no) on delete cascade,
28     foreign key(item_no) references ITEM(item_no) on delete SET NULL
29 );
30 •  create table WAREHOUSE(
31     warehouseno int,
32     city varchar(30),
33     primary key(warehouseno)
34 );
35 •  create table SHIPMENT(
36     order_no int,
37     warehouseno int,
38     ship_date date,
39     foreign key(order_no) references ORDERS(order_no) on delete cascade,
40     foreign key(warehouseno) references WAREHOUSE(warehouseno) on delete cascade
41 );
42 •  show tables;
43 •  insert into CUSTOMER(cust_no,cname,city)values(771,'PUSHPA K','BANGALORE');
44 •  insert into CUSTOMER(cust_no,cname,city)values(772,'SUNAN','MUMBAI');
45 •  insert into CUSTOMER(cust_no,cname,city)values(773,'SOURAV','CALICUT');
46 •  insert into CUSTOMER(cust_no,cname,city)values(774,'LAILA','HYDERABAD');
47 •  insert into CUSTOMER(cust_no,cname,city)values(775,'FAIZAL','BANGALORE');
48 •  COMMIT;
49 •  desc CUSTOMER;
50 •  SELECT *FROM CUSTOMER;
51 •  insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(111,'22-01-02',771,18000);
52 •  insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(112,'30-07-02',774,6000);
53 •  insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(113,'03-04-03',775,9000);
54 •  insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(114,'03-11-03',775,29000);
55 •  insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(115,'10-12-03',773,29000);
56 •  insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(116,'19-08-04',772,56000);
57 •  insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(117,'10-09-04',771,20000);
58 •  insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(118,'20-11-04',775,29000);
59 •  insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(119,'13-02-05',774,29000);
60 •  insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(120,'13-10-05',775,29000);

```

```

59 • insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(119,'13-02-05',774,29000);
60 • insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(120,'13-10-05',775,29000);
61 • COMMIT;
62 • desc ORDERS;
63 • SELECT *FROM ORDERS;
64 • insert into ITEM(item_no,unit_price)values(5001,503);
65 • insert into ITEM(item_no,unit_price)values(5002,750);
66 • insert into ITEM(item_no,unit_price)values(5003,150);
67 • insert into ITEM(item_no,unit_price)values(5004,600);
68 • insert into ITEM(item_no,unit_price)values(5005,890);
69 • COMMIT;
70 • desc ITEM;
71 • SELECT *FROM ITEM;
72 • insert into ORDER_ITEM(order_no,item_no,qty)values(111,5001,50);
73 • insert into ORDER_ITEM(order_no,item_no,qty)values(112,5003,20);
74 • insert into ORDER_ITEM(order_no,item_no,qty)values(113,5002,50);
75 • insert into ORDER_ITEM(order_no,item_no,qty)values(114,5005,60);
76 • insert into ORDER_ITEM(order_no,item_no,qty)values(115,5004,90);
77 • insert into ORDER_ITEM(order_no,item_no,qty)values(116,5001,10);
78 • insert into ORDER_ITEM(order_no,item_no,qty)values(117,5003,80);
79 • insert into ORDER_ITEM(order_no,item_no,qty)values(118,5005,50);
80 • insert into ORDER_ITEM(order_no,item_no,qty)values(119,5002,10);
81 • insert into ORDER_ITEM(order_no,item_no,qty)values(120,5004,45);
82 • COMMIT;
83 • desc ORDER_ITEM;
84 • SELECT *FROM ORDER_ITEM;
85 • insert into WAREHOUSE(warehouseno,city)values(1,'DELHI');
86 • insert into WAREHOUSE(warehouseno,city)values(2,'BOMBAY');
87 • insert into WAREHOUSE(warehouseno,city)values(3,'CHENNAI');
88 • insert into WAREHOUSE(warehouseno,city)values(4,'BANGALORE');
89 • insert into WAREHOUSE(warehouseno,city)values(5,'BANGALORE');

91 • insert into WAREHOUSE(warehouseno,city)values(7,'BOMBAY');
92 • insert into WAREHOUSE(warehouseno,city)values(8,'CHENNAI');
93 • insert into WAREHOUSE(warehouseno,city)values(9,'DELHI');
94 • insert into WAREHOUSE(warehouseno,city)values(10,'BANGALORE');
95 • COMMIT;
96 • desc WAREHOUSE;
97 • SELECT *FROM WAREHOUSE;
98 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(111,1,'10-02-02');
99 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(112,5,'10-09-02');
100 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(113,8,'10-02-03');
101 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(114,3,'10-12-03');
102 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(115,9,'19-01-04');
103 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(116,1,'20-09-04');
104 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(117,5,'10-09-04');
105 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(118,7,'30-11-04');
106 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(119,7,'30-04-05');
107 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(120,6,'21-12-05');
108 • COMMIT;
109 • desc SHIPMENT;
110 • SELECT *FROM SHIPMENT;
111 /*Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column
112 is the total numbers of orders by the customer and the last column is the average order amount for that customer.*/
113 • SELECT C.CNAME as CUSTNAME, COUNT(*) as no_of_orders,AVG(O.ord_Amt) as AVG_ORDER_AMT FROM CUSTOMER C,
114 ORDERS O WHERE C.cust_no=O.cust_no GROUP BY C.CNAME;
115 /*List the order# for orders that were shipped from all warehouses that the company has in a specific city.*/
116 • SELECT order_no FROM WAREHOUSE W, SHIPMENT S WHERE W.warehouseno=S.warehouseno AND CITY='BANGALORE';
117 /*Demonstrate how you delete item# 10 from the ITEM table and make that field null in the ORDER_ITEM table.*/
118 • delete from ITEM where item_no=5005;
119 • select *from ITEM;
120 • select *from ORDER_ITEM;

```

## OUTPUT-

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: order\_processing ×

SCHEMAS

Filter objects

book\_dealer

Tables

- author
- catalog
- category
- order\_details
- publisher

Views

Stored Procedures

Functions

sys

Limit to 1000 rows

40     foreign key(warehouse) references WAREHOUSE(warehouse) on delete cascade  
41     );  
42 •     show tables;  
43 •     insert into CUSTOMER(cust\_no,cname,city)values(771,'PUSHPA K','BANGALORE');  
44 •     insert into CUSTOMER(cust\_no,cname,city)values(772,'SUMAN','MUMBAI');  
45 •     insert into CUSTOMER(cust\_no,cname,city)values(773,'SOURAV','CALICUT');  
46 •     insert into CUSTOMER(cust\_no,cname,city)values(774,'LAILA','HYDERABAD');  
47 •     insert into CUSTOMER(cust\_no,cname,city)values(775,'FAIZAL','BANGALORE');  
48 •     COMMIT;

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

Tables\_in\_order\_processing

- customer
- item
- order\_item
- orders
- shipment
- warehouse

Result 18 ×

Read Only

Result Grid

Form Editor

Field Types

No object selected

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas

order\_processing

```

48 •    COMMIT;
49 •    desc CUSTOMER;
50 •    SELECT *FROM CUSTOMER;
51 •    insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(111,'22-01-02',771,18000);
52 •    insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(112,'30-07-02',774,6000);
53 •    insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(113,'03-04-03',775,9000);
54 •    insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(114,'03-11-03',775,29000);
55 •    insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(115,'10-12-03',773,29000);
56 •    insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(116,'19-08-04',772,56000);

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types

	cust_no	cname	city
▶	771	PUSHPA K	BANGALORE
	772	SUMAN	MUMBAI
	773	SOURAV	CALICUT
	774	LAILA	HYDERABAD
	775	FAIZAL	BANGALORE
*	NULL	NULL	NULL

Administration Schemas Information No object selected

CUSTOMER 19 X

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas

order\_processing

```

56 •    insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(116,'19-08-04',772,56000);
57 •    insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(117,'10-09-04',771,20000);
58 •    insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(118,'20-11-04',775,29000);
59 •    insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(119,'13-02-05',774,29000);
60 •    insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(120,'13-10-05',775,29000);
61 •    COMMIT;
62 •    desc ORDERS;
63 •    SELECT *FROM ORDERS;
64 •    insert into ITEM(item_no,unit_price)values(5001,503);

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types

	order_no	odate	cust_no	ord_Amt
▶	111	2022-01-02	771	18000
	112	2030-07-02	774	6000
	113	2003-04-03	775	9000
	114	2003-11-03	775	29000
	115	2010-12-03	773	29000
	116	2019-08-04	772	56000
	117	2010-09-04	771	20000
	118	2020-11-04	775	29000
	119	2013-02-05	774	29000
	120	2013-10-05	775	29000
*	NULL	NULL	NULL	NULL

Administration Schemas Information No object selected

ORDERS 20 X

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS book\_dealer

Tables author catalog category order\_details publisher

Views

Stored Procedures

Functions

sys

order\_processing

Limit to 1000 rows

64 • insert into ITEM(item\_no,unit\_price)values(5001,503);  
65 • insert into ITEM(item\_no,unit\_price)values(5002,750);  
66 • insert into ITEM(item\_no,unit\_price)values(5003,150);  
67 • insert into ITEM(item\_no,unit\_price)values(5004,600);  
68 • insert into ITEM(item\_no,unit\_price)values(5005,890);  
69 • COMMIT;  
70 • desc ITEM;  
71 • SELECT \*FROM ITEM;  
72 • insert into ORDER\_ITEM(order\_no,item\_no,qty)values(111,5001,50);

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

item_no	unit_price
5001	503
5002	750
5003	150
5004	600
5005	890

No object selected

ITEM 21 x Apply Reset

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS book\_dealer

Tables author catalog category order\_details publisher

Views

Stored Procedures

Functions

sys

order\_processing

Limit to 1000 rows

80 • insert into ORDER\_ITEM(order\_no,item\_no,qty)values(119,5002,10);  
81 • insert into ORDER\_ITEM(order\_no,item\_no,qty)values(120,5004,45);  
82 • COMMIT;  
83 • desc ORDER\_ITEM;  
84 • SELECT \*FROM ORDER\_ITEM;  
85 • insert into WAREHOUSE(warehouseno,city)values(1,'DELHI');  
86 • insert into WAREHOUSE(warehouseno,city)values(2,'BOMBAY');  
87 • insert into WAREHOUSE(warehouseno,city)values(3,'CHENNAI');  
88 • insert into WAREHOUSE(warehouseno,city)values(4,'BANGALORE');

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

order_no	item_no	qty
111	5001	50
112	5003	20
113	5002	50
114	5005	60
115	5004	90
116	5001	10
117	5001	15
118	5003	50
119	5002	10
120	5004	45

No object selected

ORDER\_ITEM 22 x Read Only

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: order\_processing

SCHEMAS

book\_dealer

- Tables
  - author
  - catalog
  - category
  - order\_details
  - publisher
- Views
- Stored Procedures
- Functions

sys

Limit to 1000 rows

```

96 • desc WAREHOUSE;
97 • SELECT *FROM WAREHOUSE;
98 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(111,1,'10-02-02');
99 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(112,5,'10-09-02');
100 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(113,8,'10-02-03');
101 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(114,3,'10-12-03');
102 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(115,9,'19-01-04');
103 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(116,1,'20-09-04');
104 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(117,5,'10-09-04');
105 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(118,7,'30-11-04');
106 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(119,7,'30-04-05');
107 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(120,6,'21-12-05');
108 • COMMIT;
109 • desc SHIPMENT;
110 • SELECT *FROM SHIPMENT;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types | Apply | Reset |

warehouseno	city
1	DELHI
2	BOMBAY
3	CHELNNAI
4	BANGALORE
5	BANGALORE
6	DELHI
7	BOMBAY
8	CHELNNAI
9	DELHI
10	BANGALORE
11	MUMBAI

WAREHOUSE 23 x

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: order\_processing

SCHEMAS

book\_dealer

- Tables
  - author
  - catalog
  - category
  - order\_details
  - publisher
- Views
- Stored Procedures
- Functions

sys

Limit to 1000 rows

```

102 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(115,9,'19-01-04');
103 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(116,1,'20-09-04');
104 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(117,5,'10-09-04');
105 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(118,7,'30-11-04');
106 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(119,7,'30-04-05');
107 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(120,6,'21-12-05');
108 • COMMIT;
109 • desc SHIPMENT;
110 • SELECT *FROM SHIPMENT;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Field Types | Read Only |

order_no	warehouseno	ship_date
111	1	2010-02-02
112	5	2010-09-02
113	8	2010-02-03
114	3	2010-12-03
115	9	2019-01-04
116	1	2020-09-04
117	5	2010-09-04
118	7	2030-11-04
119	7	2030-04-05
120	6	2021-12-05

SHIPMENT 24 x

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

order\_processing

```

106 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(119,7,'30-04-05');
107 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(120,6,'21-12-05');
108 • COMMIT;
109 • desc SHIPMENT;
110 • SELECT *FROM SHIPMENT;
111 /*Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column
112 is the total numbers of orders by the customer and the last column is the average order amount for that
113 • SELECT C.CNAME as CUSTNAME, COUNT(*) as no_of_orders,AVG(O.ord_Amt) as AVG_ORDER_AMT FROM CUSTOMER C,
114 ORDERS O WHERE C.cust_no=O.cust_no GROUP BY C.CNAME];
115
116
117
118
119

```

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

CUSTNAME	no_of_orders	AVG_ORDER_AMT
PUSHPA K	2	19000.0000
SUMAN	1	56000.0000
SOURAV	1	29000.0000
LAILA	2	17500.0000
FAIZAL	4	24000.0000

No object selected

Result 25

Read Only

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

order\_processing

```

111 /*Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column
112 is the total numbers of orders by the customer and the last column is the average order amount for that
113 • SELECT C.CNAME as CUSTNAME, COUNT(*) as no_of_orders,AVG(O.ord_Amt) as AVG_ORDER_AMT FROM CUSTOMER C,
114 ORDERS O WHERE C.cust_no=O.cust_no GROUP BY C.CNAME];
115 /*List the orders for orders that were shipped from all warehouses that the company has in a specific c
116 • SELECT order_no FROM WAREHOUSE W, SHIPMENT S WHERE W.warehouseno=S.warehouseno AND CITY='BANGALORE';
117 /*Demonstrate how you delete item# 10 from the ITEM table and make that field null in the ORDER_ITEM ta
118 • delete from ITEM where item_no=5005;
119 • select *from ITEM;

```

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

item_no	unit_price
5001	503
5002	750
5003	150
5004	600
5005	NULL

No object selected

ITEM 27

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar has sections for Administration, Schemas, and Information, with 'No object selected' currently highlighted. The main area contains a query editor titled 'order\_processing' with the following SQL code:

```

112      is the total numbers of orders by the customer and the last column is the average order amount for that ^
113 •   SELECT C.CNAME as CUSTNAME, COUNT(*) as no_of_orders, AVG(O.ord_Amt) as AVG_ORDER_AMT FROM CUSTOMER C,
114     ORDERS O WHERE C.cust_no=O.cust_no GROUP BY C.CNAME;
115 /*List the order# for orders that were shipped from all warehouses that the company has in a specific c
116 •   SELECT order_no FROM WAREHOUSE W, SHIPMENT S WHERE W.warehouseno=S.warehouseno AND CITY='BANGALORE';
117 /*Demonstrate how you delete item# 10 from the ITEM table and make that field null in the ORDER_ITEM ta
118 •   delete from ITEM where item_no=5005;
119 •   select *from ITEM;
120 •   select *from ORDER_ITEM;

```

Below the code is a 'Result Grid' showing the following data:

order_no	item_no	qty
111	5001	50
112	5003	20
113	5002	50
114	NULL	60
115	5004	90
116	5001	10
117	5003	80
118	NULL	50
119	5002	10
120	5004	45

The bottom status bar indicates 'ORDER\_ITEM 28' and 'Read Only'.

## DBMS LAB PROGRAM-4

### BANKING DATABASE

#### QUESTION-

Consider the following database for a banking enterprise.

**BRANCH** (branch-name: String, branch-city: String, assets: real)

**ACCOUNTS** (accno: int, branch-name: String, balance: real)

**DEPOSITOR** (customer-name: String, customer-street: String,  
customer-city: String)

**LOAN** (loan-number: int, branch-name: String, amount: real)

**BORROWER (customer-name: String, loan-number: 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) Find all the customers who have at least two accounts at the Main branch.
- iv) Find all the customers who have an account at all the branches located in a specific city.
- v) Demonstrate how you delete all account tuples at every branch located in a specific city.
- vi) Generate suitable reports.
- vii) Create suitable front end for querying and displaying the results.

**PROGRAM CODE-**

```
1 ●  create database banking;
2 ●  use banking;
3
4 ●  create table branch(
5     branch_name varchar(30) primary key,
6     branch_city varchar(30),
7     assets real);
8
9 ●  create table accounts(
10    accno int primary key,
11    branch_name varchar(30),
12    balance real,
13    foreign key (branch_name) references branch(branch_name) on delete cascade on update cascade);
14
15 ●  create table customer(
16    customer_name varchar(30) primary key,
17    customer_street varchar(20),
18    customer_city varchar(20));
19
20 ●  create table depositor(
21    customer_name varchar(30),
22    accno int,
23    primary key(customer_name ,accno),
24    foreign key (accno) references accounts(accno) on delete cascade on update cascade,
25    foreign key (customer_name) references customer(customer_name) on delete cascade on update
26    cascade);
27
28 ●  create table loan(
29    loan_number int primary key,
30    branch_name varchar(30),
31    amount real.
```

```

30 |     branch_name varchar(30),
31 |     amount real,
32 |     foreign key (branch_name) references branch(branch_name)
33 | );
34 |
35 • ◎ create table borrower (
36 |     customer_name varchar(30),
37 |     loan_number int,
38 |     primary key(customer_name, loan_number),
39 |     foreign key (customer_name) references customer(customer_name) on delete cascade on update cascade,
40 |     foreign key (loan_number) references loan(loan_number) on delete cascade on update cascade);
41 •   show tables;
42 •   insert into branch(branch_name,branch_city,assets) values
43 |     ('A','Bangalore',190000),
44 |     ('B','Bangalore',200000),
45 |     ('C','Delhi',235344),
46 |     ('D','Chennai',1050560),
47 |     ('E','Chennai',678909);
48 •   select *from branch;
49 |
50 •   insert into accounts(accno,branch_name,balance) VALUES
51 |     (1001,'A',10000),
52 |     (1002,'B',5000),
53 |     (1003,'C',7500),
54 |     (1004,'D',50000),
55 |     (1005,'D',75000),
56 |     (1006,'E',560),
57 |     (1007,"B",500),
58 |     (1008,"B",1500);
59 •   select *from accounts;
60 |
61 •   insert into customer(customer_name,customer_street,customer_city) VALUES
62 |     ("Ravi","Dasarahalli","Bangalore"),
63 |     ("Shyam","Indiranagar","Delhi"),
64 |     ("Seema","Vasanthagar","Chennai"),
65 |     ("Arpita","Church Street","Bangalore"),
66 |     ("Vinay","MG Road","Chennai");
67 •   select *from customer;
68 |
69 •   insert into depositor(customer_name,accno) VALUES
70 |     ("Ravi",1001),
71 |     ("Ravi",1002),
72 |     ("Shyam",1003),
73 |     ("Seema",1004),
74 |     ("Seema",1005),
75 |     ("Arpita",1006),
76 |     ("Vinay",1007),
77 |     ("Vinay",1008);
78 •   select *from depositor;
79 |
80 |
81 •   insert into loan(loan_number,branch_name,amount) VALUES
82 |     (001,'A',10000),
83 |     (002,'B',25000),
84 |     (003,'B',250000),
85 |     (004,'C',5000),
86 |     (005,'E',90000);
87 •   select *from loan;
88 |
89 •   insert into borrower(customer_name,loan_number) VALUES

```

```
88
89 • insert into borrower(customer_name,loan_number) VALUES
90     ("Arpita",001),
91     ("Ravi",002),
92     ("Arpita",003),
93     ("Shyam",004),
94     ("Vinay",005);
95 • select *from borrower;
96
97 /*iii. Find all the customers who have at least two accounts at the Main branch */
98
99 • select customer_name from depositor
100 join accounts on depositor.accno = accounts.accno where accounts.branch_name = "D"
101 group by depositor.customer_name having count(depositor.customer_name) >=2;
102
103 /* iv. Find all the customers who have an account at all the branches located in a specific city.*/
104
105 • select customer_name from depositor
106 join accounts on accounts.accno = depositor.accno
107 join branch on branch.branch_name = accounts.branch_name
108 where branch.branch_city = "Bangalore"
109 GROUP BY depositor.customer_name
110 having count(DISTINCT branch.branch_name) = (SELECT COUNT(branch_name)
111 FROM branch
112 WHERE branch_city = 'Bangalore');
113
114 /*v. Demonstrate how you delete all account tuples at every branch located in a specific city.*/
115 • delete from accounts where branch_name in
116 (select branch_name from branch where branch_city="Delhi");
117 • select *from accounts;
```

## OUTPUT-

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator order\_processing SQL File 11\* ×

SCHEMAS

Filter objects

book\_dealer

- Tables
- Views
- Stored Procedures
- Functions

order\_processing

sys

SQL Editor

```
53     (1003, 'C', 7500),
54     (1004, 'D', 50000),
55     (1005, 'D', 75000),
56     (1006, 'E', 560),
57     (1007, "B", 500),
58     (1008, "B", 1500);
59 •   select *from accounts;
60
61 •   insert into customer(customer_name,customer_street,customer_city) VALUES
```

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

acco	branch_name	balance
1001	A	10000
1002	B	5000
1004	D	50000
1005	D	75000
1006	E	560
1007	B	500
1008	B	1500
NULL	NULL	NULL

Administration Schemas Information

No object selected

Result Grid

Form Editor

Field Types

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

order\_processing

```

41 • Insert into customer(customer_name,customer_street,customer_city) VALUES
42 ("Arpit","Church Street","Bangalore"),
43 ("Ravi","Deshpali","Bangalore"),
44 ("Sena","Vasanthnagar","Chennai"),
45 ("Vipika","Church Street","Bangalore"),
46 ("Way","MG Road","Chennai")
47 • Select * from customer
48
49 • Insert into depositor(customer_name,acno) VALUES

```

Result Grid

customer_name	customer_street	customer_city
Arpit	Church Street	Bangalore
Ravi	Deshpali	Bangalore
Sena	Vasanthnagar	Chennai
Vipika	Infranagar	Dehi
Way	MG Road	Chennai

customer 12

Output

Action Output

- 22 17:46:42 select customer\_name from depositor join accounts on depositor.acno = accounts.acno where accounts.acno = 10001; 1 rows(affected)
- 23 17:46:42 select customer\_name from depositor join accounts on accounts.acno = depositor.acno join branch on bran...
- 24 17:46:42 delete from accounts where branch\_name in (select branch\_name from branch where branch.city='Dehi')
- 25 17:46:42 select \* from accounts LIMIT 0, 1000
- 26 17:50:15 select \* from accounts LIMIT 0, 1000
- 27 17:51:12 select \* from customer LIMIT 0, 1000

Object Info Session

Type here to search

1751 21-05-2021

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

order\_processing

```

45 ('C', 'Chennai', 1234544),
46 ('C', 'Chennai', 1098989),
47 ('C', 'Chennai', 4567888),
48 • Select * from branch
49
50 • Insert into accounts(acno,branch_name,balance) VALUES
51 (1001, 'A', 10000),
52 (1002, 'B', 30000),
53 (1003, 'C', 70000),
54

```

Result Grid

branch_name	branch_city	assets
A	Banglore	10000
B	Banglore	30000
C	Dehi	225244
D	Chennai	1032560
E	Chennai	678908

branch 12

Output

Action Output

- 23 17:46:42 select customer\_name from depositor join accounts on depositor.acno = accounts.acno join branch on bran...
- 24 17:46:42 delete from accounts where branch\_name in (select branch\_name from branch where branch.city='Dehi')
- 25 17:46:42 select \* from accounts LIMIT 0, 1000
- 26 17:50:15 select \* from accounts LIMIT 0, 1000
- 27 17:51:12 select \* from customer LIMIT 0, 1000
- 28 17:52:40 select \* from branch LIMIT 0, 1000

Object Info Session

Type here to search

1752 21-05-2021

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigation

SCHEMAS

- book\_dealer
- order\_processing
- sys

SQL Editor

SQL File 1

```
27  * ("Viney",1000)
28  * select * from depositor
29
30
31  * insert into loan(loan_number,branch_name,amount) VALUES-
32  * ('001', 'A',10000),
33  * ('002', 'B',20000),
34  * ('003', 'C',20000),
35  * ('004', 'C',5000),
```

Result Grid

customer_name	account
Ravi	1001
Suresh	1002
Seema	1004
Aryam	1006
Viney	1007
Viney	1008
Shivani	1009

Administrator Schemas Information

No object selected

Execution Time: 17:53:36

Action Output

- 24 17:46:42 delete from accounts where branch\_name in (select branch\_name from branch where branch\_id='Deli')
- 25 17:46:42 select from accounts LIMIT 5, 1000
- 26 17:50:15 select from accounts LIMIT 5, 1000
- 27 17:51:12 select from customer LIMIT 5, 1000
- 28 17:52:40 select from branch LIMIT 5, 1000
- 29 17:53:36 select from depositor LIMIT 5, 1000

Object Info Session

Type here to search

17:53

21-05-2021

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigation

SCHEMAS

- book\_dealer
- order\_processing
- sys

SQL Editor

SQL File 1

```
83  * ("001", "C",10000)
84  * ("002", "B",20000)
85  * select * from loan
86
87  * insert into borrower(customer_name,loan_number) VALUES-
88  * ("Ravi",1001),
89  * ("Seema",1002),
90  * ("Aryam",1006),
91  * ("Viney",1008),
```

Result Grid

loan_number	branch_name	amount
1	A	10000
2	B	20000
3	B	20000
4	C	5000
5	C	50000
6	D	50000

Administrator Schemas Information

No object selected

Execution Time: 17:54:08

Action Output

- 25 17:46:42 select from accounts LIMIT 5, 1000
- 26 17:50:15 select from accounts LIMIT 5, 1000
- 27 17:51:12 select from customer LIMIT 5, 1000
- 28 17:52:40 select from branch LIMIT 5, 1000
- 29 17:53:36 select from depositor LIMIT 5, 1000
- 30 17:54:08 select from loan LIMIT 5, 1000

Object Info Session

Type here to search

17:54

21-05-2021

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, Help, and a toolbar with various icons. The left sidebar has sections for Navigator, Schemas, and Administration, with 'order\_processing' selected under Schemas. The main area is a query editor titled 'SQL File 11' containing the following SQL code:

```
/* ALL. Find all the customers who have at least two accounts at the main branch */
10
11
12 • select customer_name from depositor
13 join accounts on depositor.acno = accounts.acno where accounts.branch_name = 'Main'
14 group by depositor.customer_name having count(depositor.customer_name) >= 2
15
16 /* i.e. Find all the customers who have an account at all the branches located in a specific city. */
17
18 • select customer_name from depositor
19
20 Result Grid | Filter Rows: Report: Wrap Cell Content: □
21 > customer_name
22 > Search
```

The results grid shows one row with the value 'customer\_name'. Below the results grid is a toolbar with icons for New, Open, Save, Paste, Print, Copy, and Paste. The bottom of the screen displays the Windows taskbar with various pinned icons and the system clock showing 21:05:2021.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the schema tree under 'order\_processing'. The central area has a 'SQL' tab open with the following SQL code:

```
100 GROUP BY depositor.customer_name  
101 having count(HISTORIES branch.branch_name) = (SELECT COUNT(branch_name)  
102 FROM branch  
103 WHERE branch_city = "Bangalore")  
104 /* Demonstrate how you delete all account tuples at every branch located in a specific city.*/  
105 * delete from accounts where branch_name in  
106 (select branch_name from branch where branch_city='Delhi')  
107 * select *from accounts
```

The results grid shows the following data:

account_id	branch_name	balance
1001	A	10000
1002	B	5000
1004	D	50000
1005	D	70000
1006	E	500
1007	E	500
1008	B	1000
1009	BBB	1000

The bottom pane shows the 'Object Info' and 'Session' tabs, and a search bar at the bottom.

The screenshot shows the MySQL Workbench interface with the following details:

- File Edit View Query Database Server Tools Scripting Help**
- Navigator**: Shows the schema **book\_deal** which contains **accounts**, **branch**, **customer**, **depositor**, and **loan**.
- SQL Editor**: Contains the following SQL code:

```
use order_processing;
insert into branch(branch_name,branch_city,assets) values
('Citi', 'Bangalore', 100000),
('Citi', 'Bangalore', 200000),
('Citi', 'Delhi', 1255344);
```
- Result Grid**: Shows the results of the last query:

branch_name	branch_city	assets
Citi	Bangalore	100000
Citi	Bangalore	200000
Citi	Delhi	1255344
- Object Info** and **Session** tabs are visible at the bottom.
- Status Bar**: Duration / Fetch: 0.000 sec / 0.000 sec

# **DBMS LAB PROGRAM-5**

# STUDENT ENROLLMENT DATABASE

# **QUESTION-**

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.

## **PROGRAM CODE-**

```

1 ● CREATE DATABASE STUDENTENROLLMENT;
2
3 ● USE STUDENTENROLLMENT;
4
5 ● ○ CREATE TABLE STUDENT(
6     REG_NO VARCHAR(30),
7     SNAME VARCHAR(30),
8     MAJOR VARCHAR(30),
9     BDATE DATE,
10    PRIMARY KEY(REG_NO)
11 );
12
13
14 ● ○ CREATE TABLE COURSE(
15     COURSE_ID INT,
16     CNAME VARCHAR(30),
17     DEPT VARCHAR(30),
18    PRIMARY KEY(COURSE_ID)
19 );
20
21
22 ● ○ CREATE TABLE ENROLL(
23     REG_NO VARCHAR(30),
24     COURSE_ID INT,
25     SEM INT,
26     MARKS INT,
27     FOREIGN KEY(REG_NO) REFERENCES STUDENT(REG_NO) ON DELETE CASCADE ON UPDATE CASCADE,
28     FOREIGN KEY(COURSE_ID) REFERENCES COURSE(COURSE_ID) ON DELETE CASCADE ON UPDATE CASCADE
29 );
30
31 ● ○ CREATE TABLE BOOK_ADOPTION(
32     COURSE_ID INT,

```

```

32     COURSE_ID INT,
33     SEM INT,
34     BOOK_ISBN INT,
35     PRIMARY KEY(BOOK_ISBN),
36     FOREIGN KEY(COURSE_ID) REFERENCES COURSE(COURSE_ID) ON DELETE CASCADE ON UPDATE CASCADE
37 );
38
39 ● ○ CREATE TABLE TEXT(
40     BOOK_ISBN INT,
41     BOOK_TITLE VARCHAR(30),
42     PUBLISHER VARCHAR(30),
43     AUTHOR VARCHAR(30),
44     FOREIGN KEY(BOOK_ISBN) REFERENCES BOOK_ADOPTION(BOOK_ISBN) ON DELETE CASCADE ON UPDATE CASCADE
45 );
46 ● show tables;
47 ● INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('CS01', 'RAM', 'DS', '1986-03-12');
48 ● INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('IS02', 'SMITH', 'USP', '1987-12-23');
49 ● INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('EC03', 'AHMED', 'SNS', '1985-04-17');
50 ● INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('CS03', 'SNEHA', 'DBMS', '1987-01-01');
51 ● INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('TC05', 'AKHILA', 'EC', '1986-10-06');
52 ● SELECT * FROM STUDENT;
53
54 ● INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (11, 'DS', 'CS');
55 ● INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (22, 'USP', 'IS');
56 ● INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (33, 'SNS', 'EC');
57 ● INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (44, 'DBMS', 'CS');
58 ● INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (55, 'EC', 'TC');
59 ● SELECT * FROM COURSE;
60
61 ● INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('CS01', 11, 4, 85);
62 ● INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('IS02', 22, 6, 80);
63 ● INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('EC03', 33, 2, 80);

```

```

63 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('EC03', 33, 2, 80);
64 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('CS03', 44, 6, 75);
65 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('TC05', 55, 2, 8);
66 • SELECT * FROM ENROLL;
67
68 • INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (11,4,1);
69 • INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (11,4,2);
70 • INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (44,6,3);
71 • INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (44,6,4);
72 • INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (55,2,5);
73 • INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (22,6,6);
74 • INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (55,2,7);
75 • SELECT * FROM BOOK_ADOPTION;
76
77 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (1, 'DS and C', 'Princeton', 'Padma Reddy');
78 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (2, 'Fundamentals of DS', 'Princeton', 'Gosse');
79 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (3, 'Fundamentals of DBMS', 'Princeton', 'Navathe');
80 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (4, 'SQL', 'Princeton', 'Foley');
81 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (5, 'Electronic circuits', 'TMH', 'Elmasri');
82 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (6, 'Adv unix prog', 'TMH', 'Stevens');
83 • SELECT * FROM TEXT;
84
85 -- Demonstrate how you add a new text book to the database and make this book be adopted by some department.
86 • INSERT INTO TEXT VALUES(7, "TREES & GRAPHS", "PRINCETON", "SADGE");
87 • INSERT INTO BOOK_ADOPTION VALUES(11, 4, 8);
88
89 • SELECT * FROM BOOK_ADOPTION;
90
91 • SELECT * FROM TEXT;
92
93 -- 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
94 • SELECT C.COURSE_ID,T.BOOK_ISBN,T.BOOK_TITLE FROM TEXT T,COURSE C,BOOK_ADOPTION B WHERE T.BOOK_ISBN=B.BOOK_ISBN AND

```

```

82 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (6, 'Adv unix prog', 'TMH', 'Stevens');
83 • SELECT * FROM TEXT;
84
85 -- Demonstrate how you add a new text book to the database and make this book be adopted by some department.
86 • INSERT INTO TEXT VALUES(7, "TREES & GRAPHS", "PRINCETON", "SADGE");
87 • INSERT INTO BOOK_ADOPTION VALUES(11, 4, 8);
88
89 • SELECT * FROM BOOK_ADOPTION;
90
91 • SELECT * FROM TEXT;
92
93 -- 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
94 • SELECT C.COURSE_ID,T.BOOK_ISBN,T.BOOK_TITLE FROM TEXT T,COURSE C,BOOK_ADOPTION B WHERE T.BOOK_ISBN=B.BOOK_ISBN AND
95 B.COURSE_ID=C.COURSE_ID AND C.DEPT='CS' AND (SELECT COUNT(B.BOOK_ISBN) FROM BOOK_ADOPTION B WHERE
96 C.COURSE_ID=B.COURSE_ID)>=2 ORDER BY T.BOOK_TITLE;
97
98
99 -- List any department that has all its adopted books published by a specific publisher.
100 • SELECT DISTINCT C.DEPT
101     FROM COURSE C
102     WHERE C.DEPT IN
103     ( SELECT C.DEPT
104         FROM COURSE C,BOOK_ADOPTION B,TEXT T
105         WHERE C.COURSE_ID=B.COURSE_ID
106         AND T.BOOK_ISBN=B.BOOK_ISBN
107         AND T.PUBLISHER='Princeton'
108         AND C.DEPT NOT IN
109         (SELECT C.DEPT
110             FROM COURSE C,BOOK_ADOPTION B,TEXT T
111             WHERE C.COURSE_ID=B.COURSE_ID
112             AND T.BOOK_ISBN=B.BOOK_ISBN
113             AND T.PUBLISHER != 'Princeton');

```

## OUTPUT-

order\_processing Banker Database SQL File 4\* | Limit to 1000 rows

```
81 •    INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (5, 'Electronic circuits', 'TMH', 'El^
82 •    INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (6, 'Adv unix prog', 'TMH', 'Stevens'
83 •    SELECT * FROM TEXT;
84
85      -- Demonstrate how you add a new text book to the database and make this book be adopted by some depa
86 •    INSERT INTO TEXT VALUES(7, "TREES & GRAPHS", "PRINCETON", "SADGE");
87 •    INSERT INTO BOOK_ADOPTION VALUES(11, 4, 8);
88
89 •    SELECT * FROM BOOK_ADOPTION;
```

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

COURSE_ID	SEM	BOOK_ISBN
11	4	1
11	4	2
44	6	3
44	6	4
55	2	5
22	6	6
55	2	7
11	4	8
HULL	HULL	HULL

BOOK\_ADOPTION 11 x

Result Grid  
Form Editor  
Field Types

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator order\_processing Banker Database SQL File 4\* ×

Limit to 1000 rows

```

81 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (5, 'Electronic circuits', 'TMH', 'Elmasri')
82 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (6, 'Adv unix prog', 'TMH', 'Stevens')
83 • SELECT * FROM TEXT;
84
85 -- Demonstrate how you add a new text book to the database and make this book be adopted by some department
86 • INSERT INTO TEXT VALUES(7, "TREES & GRAPHS", "PRINCETON", "SADGE");
87 • INSERT INTO BOOK_ADOPTION VALUES(11, 4, 8);
88
89 • SELECT * FROM BOOK_ADOPTION;

```

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

	BOOK_ISBN	BOOK_TITLE	PUBLISHER	AUTHOR
1	DS and C	Princeton	Padma Reddy	
2	Fundamentals of DS	Princeton	Godse	
3	Fundamentals of DBMS	Princeton	Navathe	
4	SQL	Princeton	Foley	
5	Electronic circuits	TMH	Elmasri	
6	Adv unix prog	TMH	Stevens	
7	TREES & GRAPHS	PRINCETON	SADGE	

TEXT 12 × Read Only

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator order\_processing Banker Database SQL File 4\* ×

Limit to 1000 rows

```

89 • SELECT * FROM BOOK_ADOPTION;
90
91 • SELECT * FROM TEXT;
92
93 -- Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for
94 • SELECT C.COURSE_ID, T.BOOK_ISBN, T.BOOK_TITLE FROM TEXT T, COURSE C, BOOK_ADOPTION B WHERE T.BOOK_ISBN=B.BOOK_ISBN AND C.COURSE_ID=B.COURSE_ID AND C.DEPT="CS" AND (SELECT COUNT(B.BOOK_ISBN) FROM BOOK_ADOPTION B WHERE
95 C.COURSE_ID=B.COURSE_ID)>=2 ORDER BY T.BOOK_TITLE;
96
97

```

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

	BOOK_ISBN	BOOK_TITLE	PUBLISHER	AUTHOR
1	DS and C	Princeton	Padma Reddy	
2	Fundamentals of DS	Princeton	Godse	
3	Fundamentals of DBMS	Princeton	Navathe	
4	SQL	Princeton	Foley	
5	Electronic circuits	TMH	Elmasri	
6	Adv unix prog	TMH	Stevens	
7	TREES & GRAPHS	PRINCETON	SADGE	

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

order\_processing Banker Database SQL File 4\*

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

```
89 • SELECT * FROM BOOK_ADOPTION;
90
91 • SELECT * FROM TEXT;
92
93 -- Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for
94 • SELECT C.COURSE_ID,T.BOOK_ISBN,T.BOOK_TITLE FROM TEXT T,COURSE C,BOOK_ADOPTION B WHERE T.BOOK_ISBN=B.BO
95 B.COURSE_ID=C.COURSE_ID AND C.DEP="CS" AND (SELECT COUNT(B.BOOK_ISBN) FROM BOOK_ADOPTION B WHERE
96 C.COURSE_ID=B.COURSE_ID)>=2 ORDER BY T.BOOK_TITLE;
97
```

Administration Schemas Information

No object selected

Result 14 ×

Read Only

COURSE_ID	BOOK_ISBN	BOOK_TITLE
11	1	DS and C
44	3	Fundamentals of DBMS
11	2	Fundamentals of DS
44	4	SQL

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SQL File 5\*

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

```
185 WHERE C.COURSE_ID=B.COURSE_ID
186 AND T.BOOK_ISBN=B.BOOK_ISBN
187 AND T.PUBLISHER='Princeton')
188 AND C.DEP NOT IN
189 (SELECT C.DEP
190 FROM COURSE C,BOOK_ADOPTION B,TEXT T
191 WHERE C.COURSE_ID=B.COURSE_ID
192 AND T.BOOK_ISBN=B.BOOK_ISBN
193 AND T.PUBLISHER != 'Princeton'))
```

Administration Schemas Information

No object selected

DEPT
CS

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- banking
- book\_dealer
- order\_processing

SQL File 5\* ×

```

57 • INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (44, 'DBMS', 'CS');
58 • INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (55, 'EC', 'TC');
59 • SELECT * FROM COURSE;
60
61 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('CS01', 11, 4, 85);
62 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('IS02', 22, 6, 80);
63 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('EC03', 33, 2, 80);
64 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('CS03', 44, 6, 75);
65 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('TC05', 55, 2, 8);

```

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

COURSE_ID	CNAME	DEPT
11	DS	CS
22	USP	IS
33	SNS	EC
44	DBMS	CS
55	EC	TC
*	HULL	HULL

COURSE 11 ×

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- banking
- book\_dealer
- order\_processing

SQL File 5\* ×

```

41     BOOK_TITLE VARCHAR(30),
42     PUBLISHER VARCHAR(30),
43     AUTHOR VARCHAR(30),
44     FOREIGN KEY(BOOK_ISBN) REFERENCES BOOK_ADOPTION(BOOK_ISBN) ON DELETE CASCADE ON UPDATE CASCADE
45   );
46 • show tables];
47 • INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('CS01', 'RAM', 'DS', '1986-03-12');
48 • INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('IS02', 'SMITH', 'USP', '1987-12-23');
49 • INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('EC03', 'AHMED', 'SNS', '1985-04-17');

```

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

Tables_in_studenterrollment
book_adoption
course
enroll
student
text

Result 12 ×

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- banking
- book\_dealer
- order\_processing

SQL File 5\* ×

```

49 • INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('EC03', 'AHMED', 'SNS', '1985-04-17');
50 • INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('CS03', 'SNEHA', 'DBMS', '1987-01-01');
51 • INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('TC05', 'AIKHILA', 'EC', '1986-10-06');
52 • SELECT * FROM STUDENT;
53
54 • INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (11, 'DS', 'CS');
55 • INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (22, 'USP', 'IS');
56 • INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (33, 'SNS', 'EC');
57 • INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (44, 'DBMS', 'CS');

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types |

REG_NO	SNAME	MAJOR	BDATE
CS01	RAM	DS	1986-03-12
CS03	SNEHA	DBMS	1987-01-01
EC03	AHMED	SNS	1985-04-17
IS02	SMITH	USP	1987-12-23
TC05	AIKHILA	EC	1986-10-06

Administration Schemas Information No object selected

STUDENT 13 ×

Output

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- banking
- book\_dealer
- order\_processing

SQL File 5\* ×

```

65 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('TC05', 55, 2, 11);
66 • SELECT * FROM ENROLL;
67
68 • INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (11,4,1);
69 • INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (11,4,2);
70 • INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (44,6,3);
71 • INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (44,6,4);
72 • INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (55,2,5);
73 • INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (22,6,6);

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Field Types |

REG_NO	COURSE_ID	SEM	MARKS
CS01	11	4	85
IS02	22	6	80
EC03	33	2	80
CS03	44	6	75
TC05	55	2	8

Administration Schemas Information No object selected

ENROLL 14 ×

Read Only

File Server Tools Scripting Help

SQL File 5\* ×

```
73 • INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (22,6,6);
74 • INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (55,2,7);
75 • SELECT * FROM BOOK_ADOPTION;
76
77 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (1, 'DS and C', 'Princeton', 'Padma R
78 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (2, 'Fundamentals of DS', 'Princeton'
79 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (3, 'Fundamentals of DBMS', 'Princeto
80 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (4, 'SQL', 'Princeton', 'Foley');
81 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (5, 'Electronic circuits', 'TMH', 'El
```

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

	COURSE_ID	SEM	BOOK_ISBN
▶	11	4	1
	11	4	2
	44	6	3
	44	6	4
	55	2	5
	22	6	6
	55	2	7
	11	4	8
*	HULL	HULL	HULL

Result Grid  
Form Editor  
Field Types