

DBMS LAB REPORT 1

Mallika Prasad

1BM19CS081

Lab 1

INSURANCE database

```
Create
database
mallika;

show databases;
use mallika;
show tables;
create table PERSON(driver_id varchar(10), name varchar(20), address varchar(15), primary
key(driver_id));
create table CAR(regno varchar(10), model varchar(20), year date, primary key(regno));
create table ACCIDENT(report_no int, adate date, location varchar(15), primary key(report_no));
create table OWNS(driver_id varchar(10), regno varchar(10), primary key(driver_id,regno), foreign
key(driver_id) references PERSON(driver_id) on delete cascade, foreign key(regno) references
CAR(regno) on delete cascade);
create table PARTICIPATED(driver_id varchar(10), regno varchar(10), report_no int, damage_amt
float, foreign key(driver_id,regno) references OWNS(driver_id,regno) on delete cascade, foreign
key(report_no) references ACCIDENT(report_no) on delete cascade);
desc PERSON;
desc CAR;
desc ACCIDENT;
desc OWNS;
desc PARTICIPATED;
insert into PERSON values(1111,'sam','k.s.layout');
insert into PERSON values(2222,'john','indiranagar');
insert into PERSON values(3333,'penny','jayanagar');
insert into PERSON values(4444,'tom','whitefield');
insert into PERSON values(5555,'paul','vijayanagar');
insert into CAR values('CN5247','maruthi','2000-01-14');
insert into CAR values('CN8569','fordicon','2000-04-25');
insert into CAR values('CN4712','zen','1999-05-06');
insert into CAR values('CN1425','maruthi','2002-11-19');
insert into CAR values('CN5678','indica','2002-06-20');
insert into ACCIDENT values(12,'02-06-02','m g road');
insert into ACCIDENT values(200,'00-12-02','double road');
insert into ACCIDENT values(300,'99-07-23','m g road');
insert into ACCIDENT values(25000,'11-06-00','residency road');
insert into ACCIDENT values(26500,'01-09-01','richmond road');
insert into OWNS values(1111,'CN5247');
```

```

insert into OWNS values(1111,'CN8569');
insert into OWNS values(2222,'CN4712');
insert into OWNS values(3333,'CN1425');
insert into OWNS values(4444,'CN5678');
insert into PARTICIPATED values(1111,'CN5247',12,20000);
insert into PARTICIPATED values(2222,'CN4712',200,500);
insert into PARTICIPATED values(3333,'CN1425',300,10000);
insert into PARTICIPATED values(4444,'CN5678',25000,2375);
insert into PARTICIPATED values(1111,'CN8569',26500,70000);
insert into PARTICIPATED values(2222,'CN4712',12,10000);
desc PERSON;
select * from PERSON;
select * from ACCIDENT;
select * from OWNS;
select * from PARTICIPATED;
update PARTICIPATED set damage_amt=25000 where report_no=12 and regno='CN5247';
insert into ACCIDENT values(27500,'08-09-01','church street');
select count(*) from accident where adate like '2008-__-__';
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';

```

OUTPUT

The screenshot shows a database management interface. The top section displays a list of SQL queries, including several INSERT statements for the PARTICIPATED table, a DESC statement for the PERSON table, and SELECT statements for PERSON, ACCIDENT, and OWNS tables. Below the queries, the 'Result Grid' is visible, showing the output of the last query (SELECT * from PERSON;). The grid contains five rows of data with columns driver_id, name, and address. The bottom of the interface shows a status bar with 'PERSON 43' and buttons for 'Apply' and 'Revert'.

driver_id	name	address
1111	sam	k.s.layout
2222	john	indiranagar
3333	penny	jayanagar
4444	tom	whitefield
5555	paul	vijayanagar
* NULL	NULL	NULL

Limit to 1000 rows

```

39 • insert into PARTICIPATED values(1111,'CN8569',26500,70000);
40 • insert into PARTICIPATED values(2222,'CN4712',12,10000);
41 • desc PERSON;
42 • select * from PERSON;
43 • select * from ACCIDENT;
44 • select * from OWNS;
45 • select * from PARTICIPATED;

```

Result Grid

report_no	adate	location
12	2001-06-02	m g road
200	2000-12-02	double road
300	1999-07-23	m g road
25000	2011-06-00	residency road
26500	2016-09-01	richmond road
27500	2008-09-01	church street
NULL	NULL	NULL

ACCIDENT 44

Apply Revert

Limit to 1000 rows

```

40 • insert into PARTICIPATED values(2222,'CN4712',12,10000);
41 • desc PERSON;
42 • select * from PERSON;
43 • select * from ACCIDENT;
44 • select * from OWNS;
45 • select * from PARTICIPATED;
46 • update PARTICIPATED set damage_amt=25000 where report_no=12 and regno='CN5247';

```

Result Grid

driver_id	regno
3333	CN1425
2222	CN4712
1111	CN5247
4444	CN5678
1111	CN8569
NULL	NULL

OWNS 45

Apply Revert

Limit to 1000 rows

```

41 • desc PERSON;
42 • select * from PERSON;
43 • select * from ACCIDENT;
44 • select * from OWNS;
45 • select * from PARTICIPATED;
46 • update PARTICIPATED set damage_amt=25000 where report_no=12 and regno='CN5247';
47 • insert into ACCIDENT values(27500,'08-09-01','church street');

```

Result Grid

driver_id	regno	report_no	damage_amt
1111	CN8569	26500	70000
4444	CN5678	25000	2375
3333	CN1425	300	10000
2222	CN4712	200	500
2222	CN4712	12	10000
1111	CN5247	12	25000

PARTICIPATED 46

Read Only

Limit to 1000 rows

```

41 • desc PERSON;
42 • select * from PERSON;
43 • select * from ACCIDENT;
44 • select * from OWNS;
45 • select * from CAR;
46 • select * from PARTICIPATED;
47 • update PARTICIPATED set damage_amt=25000 where report_no=12 and regno='CN5247';

```

Result Grid

regno	model	year
CN1425	maruthi	2002-11-19
CN4712	zen	1999-05-06
CN5247	maruthi	2000-01-14
CN5678	indica	2002-06-20
CN8569	fordicon	2000-04-25
NULL	NULL	NULL

Output

Limit to 1000 rows

```

43 • select * from ACCIDENT;
44 • select * from OWNS;
45 • select * from PARTICIPATED;
46 • update PARTICIPATED set damage_amt=25000 where report_no=12 and regno='CN5247';
47 • insert into ACCIDENT values(27500,'08-09-01','church street');
48 • select count(*) from accident where adate like '2008-__-__';
49 • 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
50

```

Result Grid

count(*)
1

Result 37

Limit to 1000 rows

```

44 • select * from OWNS;
45 • select * from PARTICIPATED;
46 • update PARTICIPATED set damage_amt=25000 where report_no=12 and regno='CN5247';
47 • insert into ACCIDENT values(27500,'08-09-01','church street');
48 • select count(*) from accident where adate like '2008-__-__';
49 • 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
50
51

```

Result Grid

count(a.report_no)
2

Result 38

LAB 2

BOOKDETAILS database

```
use
mallika;

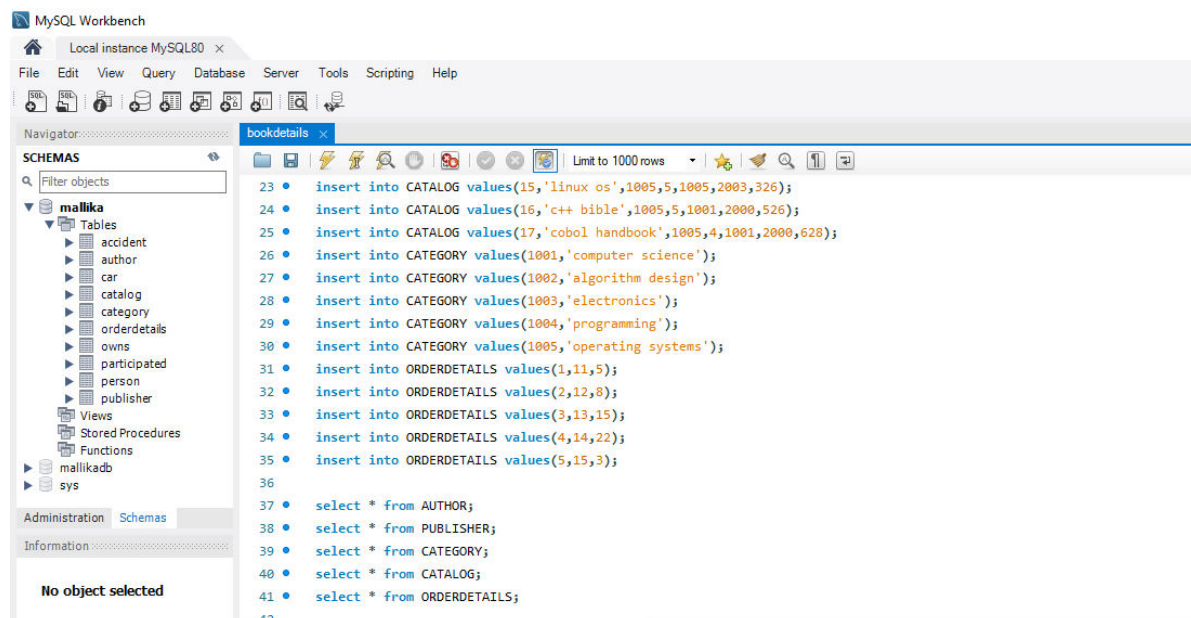
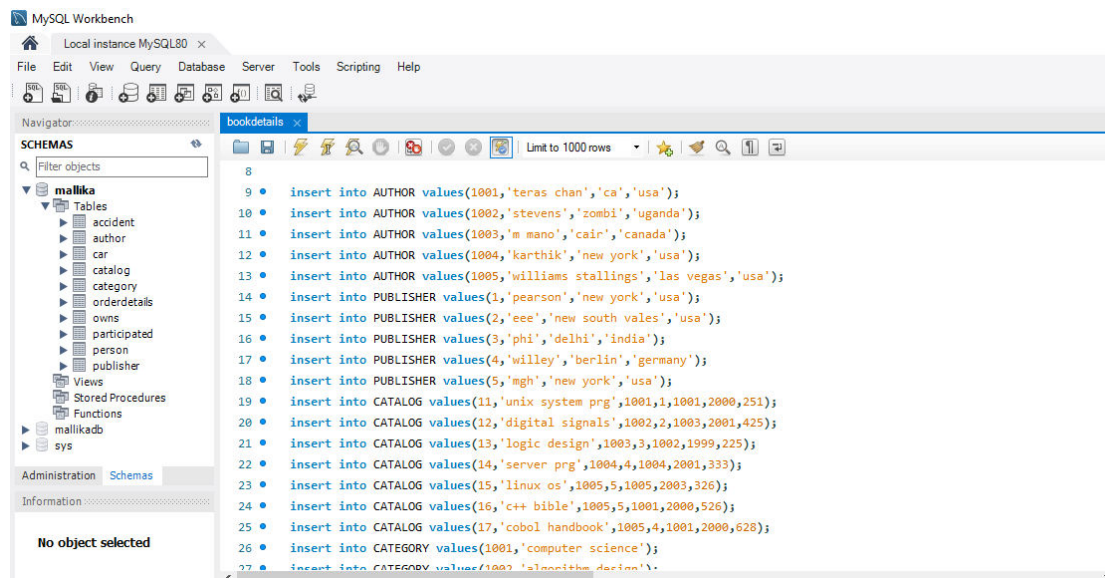
show tables;
create table AUTHOR(authorid int, name varchar(30), city varchar(30), country varchar(30), primary
key(authorid));
create table PUBLISHER(publisherid int,name varchar(30),city varchar(30), country varchar(30), primary
key(publisherid));
create table CATALOG(bookid int,title varchar(30), authorid int,publisherid int ,categoryid int, year
int, price int, primary key(bookid), foreign key(publisherid) references PUBLISHER(publisherid),
foreign key(categoryid) references CATEGORY(categoryid));
create table CATEGORY(categoryid int,description varchar(30), primary key(categoryid));
create table ORDERDETAILS(orderno int,bookid int,quantity int, primary key(orderno), foreign
key(bookid) references CATALOG(bookid));
insert into AUTHOR values(1001,'teras chan','ca','usa');
insert into AUTHOR values(1002,'stevens','zombi','uganda');
insert into AUTHOR values(1003,'m mano','cair','canada');
insert into AUTHOR values(1004,'karthik','new york','usa');
insert into AUTHOR values(1005,'williams stallings','las vegas','usa');
insert into PUBLISHER values(1,'pearson','new york','usa');
insert into PUBLISHER values(2,'eee','new south vales','usa');
insert into PUBLISHER values(3,'phi','delhi','india');
insert into PUBLISHER values(4,'willey','berlin','germany');
insert into PUBLISHER values(5,'mgh','new york','usa');
insert into CATALOG values(11,'unix system prg',1001,1,1001,2000,251);
insert into CATALOG values(12,'digital signals',1002,2,1003,2001,425);
insert into CATALOG values(13,'logic design',1003,3,1002,1999,225);
insert into CATALOG values(14,'server prg',1004,4,1004,2001,333);
insert into CATALOG values(15,'linux os',1005,5,1005,2003,326);
insert into CATALOG values(16,'c++ bible',1005,5,1001,2000,526);
insert into CATALOG values(17,'cobol handbook',1005,4,1001,2000,628);
insert into CATEGORY values(1001,'computer science');
insert into CATEGORY values(1002,'algorithm design');
insert into CATEGORY values(1003,'electronics');
insert into CATEGORY values(1004,'programming');
insert into CATEGORY values(1005,'operating systems');
insert into ORDERDETAILS values(1,11,5);
insert into ORDERDETAILS values(2,12,8);
insert into ORDERDETAILS values(3,13,15);
insert into ORDERDETAILS values(4,14,22);
insert into ORDERDETAILS values(5,15,3);
select * from AUTHOR;
select * from PUBLISHER;
select * from CATEGORY;
select * from CATALOG;
select * from ORDERDETAILS;
select a.authorid,a.name,a.city,a.country,c.price from AUTHOR a,CATALOG c where a.authorid=c.authorid
```

```

AND c.year>2000 group by c.authorid HAVING count(c.authorid)>=2;
select a.name from AUTHOR a, CATALOG c, ORDERDETAILS o where a.authorid=c.authorid and
c.bookid=o.bookid and o.quantity=(select max(quantity) from ORDERDETAILS);
update CATALOG c set c.price=1.1*price where c.publisherid=(select publisherid from PUBLISHER where
name='willey');
select p.name, c.price from PUBLISHER p, CATALOG c where p.publisherid=c.publisherid and
p.name='willey';

```

Output



MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

malika

Tables

- accident
- author
- car
- catalog
- category
- orderdetails
- owns
- participated
- person
- publisher

Views

Stored Procedures

Functions

malikadb

sys

Administration Schemas

Information

No object selected

bookdetails

Limit to 1000 rows

```

34 • insert into ORDERDETAILS values(4,14,22);
35 • insert into ORDERDETAILS values(5,15,3);
36
37 • select * from AUTHOR;
38 • select * from PUBLISHER;
39 • select * from CATEGORY;
40 • select * from CATALOG;
41 • select * from ORDERDETAILS;
42
43 • select a.authorid,a.name,a.city,a.country,c.price from AUTHOR a,CATALOG c where a.authorid=c.authorid AND c.year>2000;
44 • select a.name from AUTHOR a, CATALOG c, ORDERDETAILS o where a.authorid=c.authorid and c.bookid=o.bookid and o.quantity>0;
45 • update CATALOG c set c.price=1.1*price where c.publisherid=(select publisherid from PUBLISHER where name='willey');

```

Result Grid

authorid	name	city	country	price
----------	------	------	---------	-------

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

malika

Tables

- accident
- author
- car
- catalog
- category
- orderdetails
- owns
- participated
- person
- publisher

Views

Stored Procedures

Functions

malikadb

sys

Administration Schemas

Information

No object selected

bookdetails

Limit to 1000 rows

```

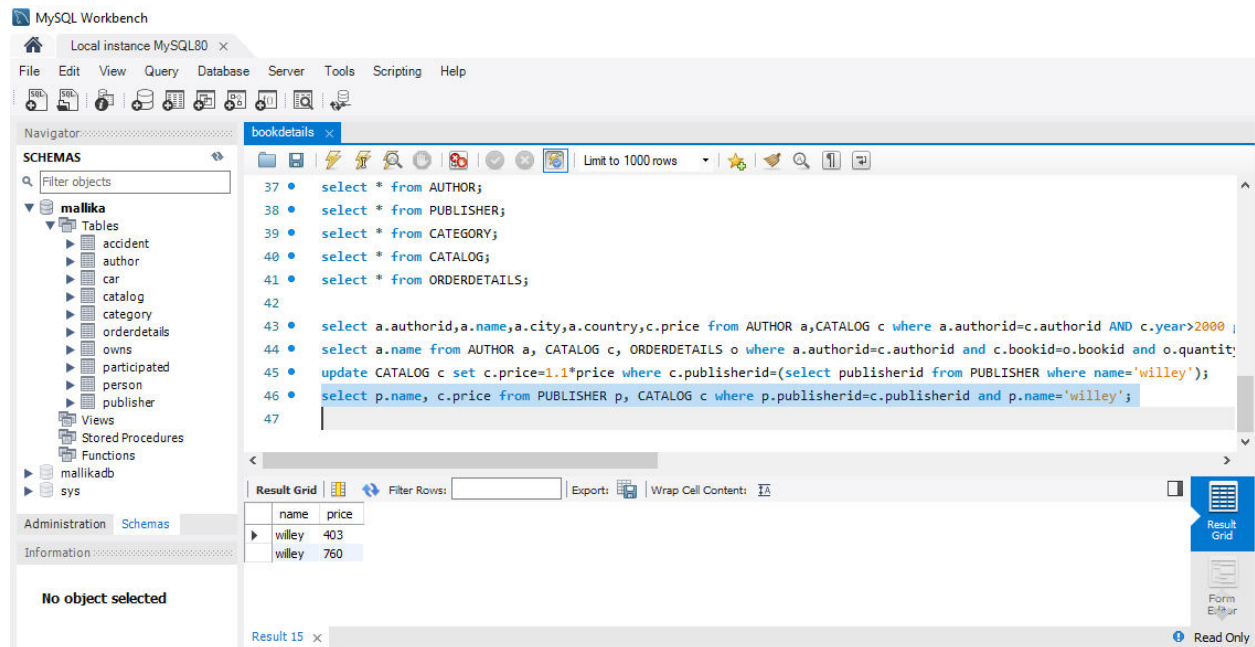
35 • insert into ORDERDETAILS values(5,15,3);
36
37 • select * from AUTHOR;
38 • select * from PUBLISHER;
39 • select * from CATEGORY;
40 • select * from CATALOG;
41 • select * from ORDERDETAILS;
42
43 • select a.authorid,a.name,a.city,a.country,c.price from AUTHOR a,CATALOG c where a.authorid=c.authorid AND c.year>2000;
44 • select a.name from AUTHOR a, CATALOG c, ORDERDETAILS o where a.authorid=c.authorid and c.bookid=o.bookid and o.quantity>0;
45 • update CATALOG c set c.price=1.1*price where c.publisherid=(select publisherid from PUBLISHER where name='willey');
46 • select p.name, c.price from PUBLISHER p, CATALOG c where p.publisherid=c.publisherid and p.name='willey';

```

Result Grid

name
karthik

Result 14



LAB 3

use

mallika;

```
create table CUSTOMER(custno int, cname varchar(30), city varchar(30), primary key(custno));
create table ORDERS(orderno int,odate date,custno int,ordamt int, primary key(orderno), foreign
key(custno) references CUSTOMER(custno));
create table ITEM(itemno int,unitprice int, primary key(itemno));
create table ORDERITEM(orderno int,itemno int, qty int, foreign key(orderno) references
ORDERS(orderno) ON DELETE SET NULL ON UPDATE CASCADE, foreign key(itemno) references ITEM(itemno) ON
DELETE SET NULL ON UPDATE CASCADE);
create table WAREHOUSE(warehouseno int,city varchar(30), primary key(warehouseno));
create table SHIPMENT(orderno int,warehouseno int,shipdate date, foreign key(orderno) references
ORDERS(orderno), foreign key(warehouseno) references WAREHOUSE(warehouseno));
INSERT INTO CUSTOMER VALUES(771,'PUSHPA K','BANGALORE');
INSERT INTO CUSTOMER VALUES(772,'SUMAN','MUMBAI');
INSERT INTO CUSTOMER VALUES(773,'SOURAV','CALICUT');
INSERT INTO CUSTOMER VALUES(774,'LAILA','HYDERABAD');
INSERT INTO CUSTOMER VALUES(775,'FAIZAL','BANGALORE');
INSERT INTO ORDERS VALUES(111,'2002-01-22',771,18000);
INSERT INTO ORDERS VALUES(112,'2002-07-30',774,6000);
INSERT INTO ORDERS VALUES(113,'2003-04-03',775,9000);
INSERT INTO ORDERS VALUES(114,'2003-11-03',775,29000);
INSERT INTO ORDERS VALUES(115,'2003-12-10',773,29000);
INSERT INTO ORDERS VALUES(116,'2004-08-19',772,56000);
INSERT INTO ORDERS VALUES(117,'2004-09-10',771,20000);
INSERT INTO ORDERS VALUES(118,'2004-11-20',775,29000);
INSERT INTO ORDERS VALUES(119,'2005-02-13',774,29000);
INSERT INTO ORDERS VALUES(120,'2005-10-13',775,29000);
INSERT INTO ITEM VALUES(5001,503);
```

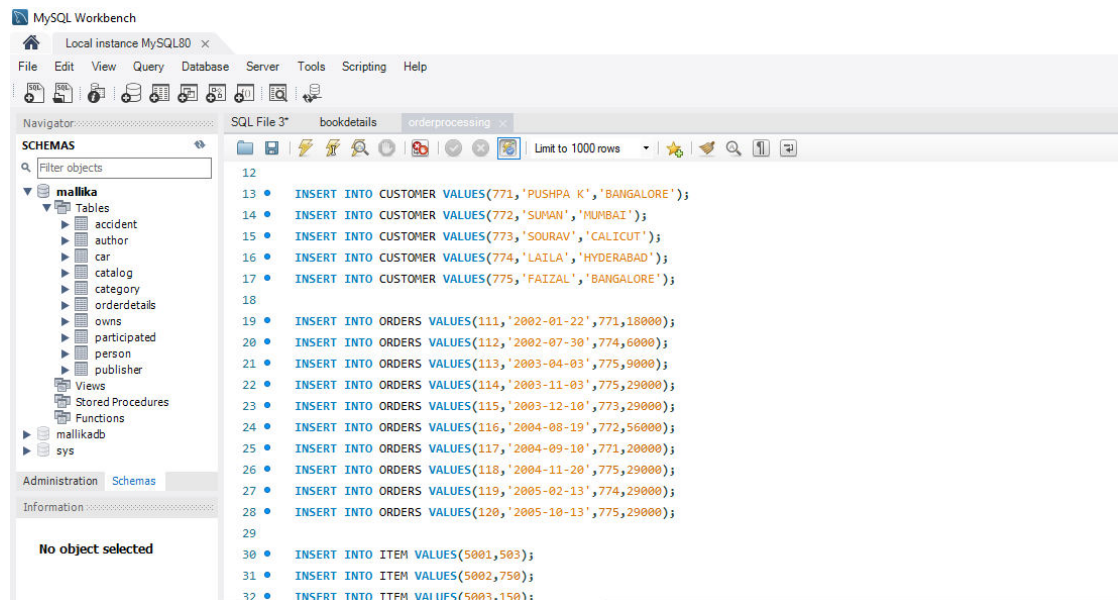
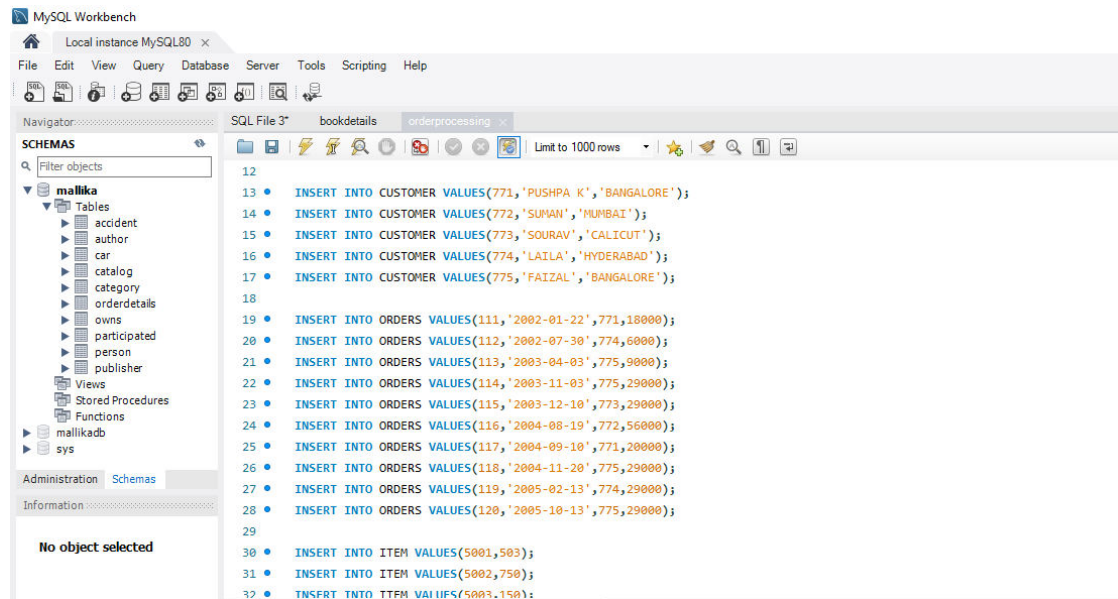


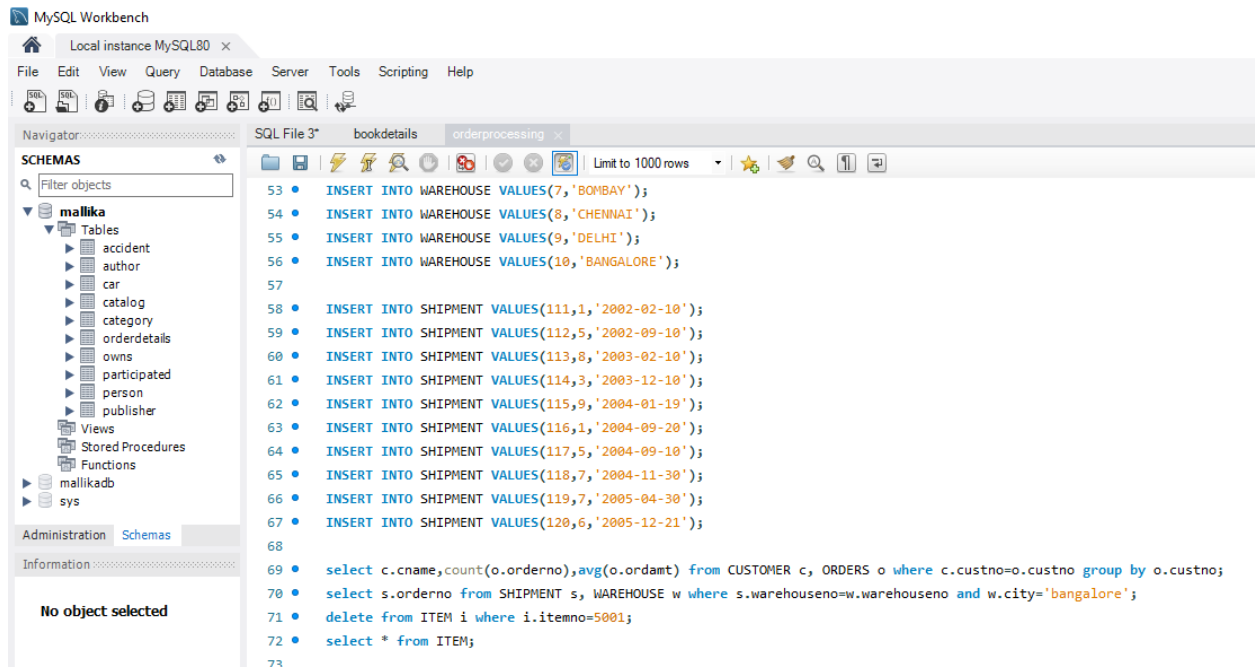
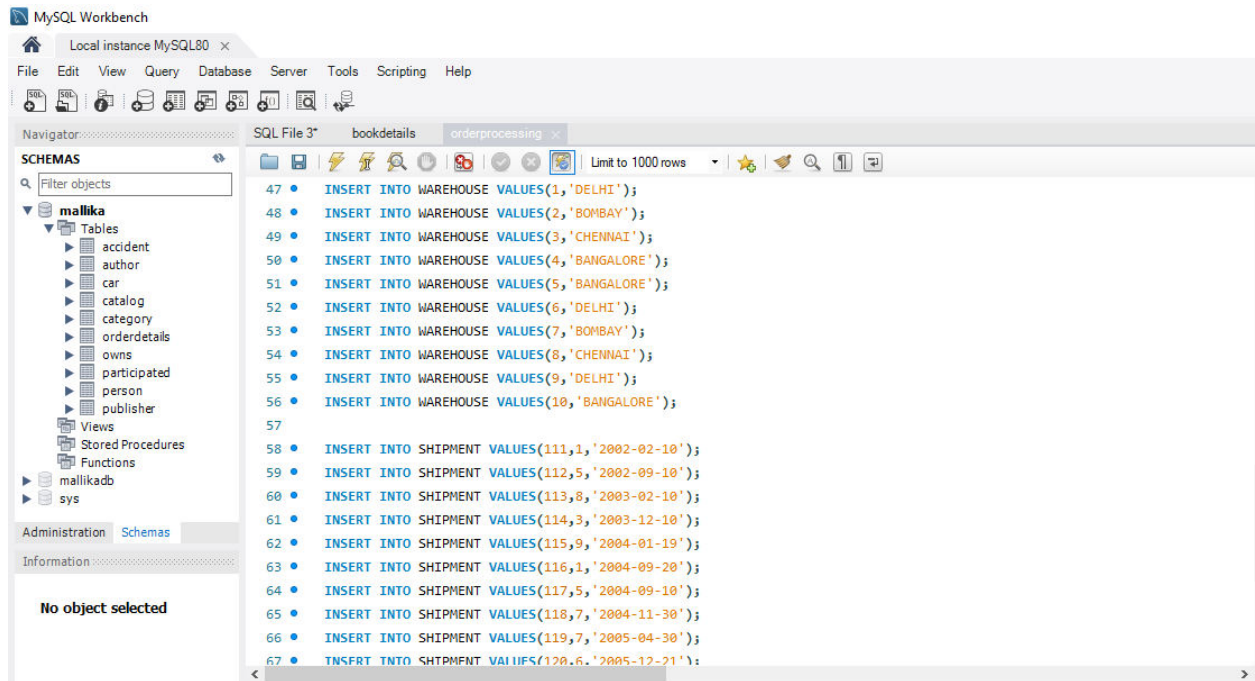
```

INSERT INTO ITEM VALUES(5002,750);
INSERT INTO ITEM VALUES(5003,150);
INSERT INTO ITEM VALUES(5004,600);
INSERT INTO ITEM VALUES(5005,890);
INSERT INTO ORDERITEM VALUES(111,5001,50);
INSERT INTO ORDERITEM VALUES(112,5003,20);
INSERT INTO ORDERITEM VALUES(113,5002,50);
INSERT INTO ORDERITEM VALUES(114,5005,60);
INSERT INTO ORDERITEM VALUES(115,5004,90);
INSERT INTO ORDERITEM VALUES(116,5001,10);
INSERT INTO ORDERITEM VALUES(117,5003,80);
INSERT INTO ORDERITEM VALUES(118,5005,50);
INSERT INTO ORDERITEM VALUES(119,5002,10);
INSERT INTO ORDERITEM VALUES(120,5004,45);
INSERT INTO WAREHOUSE VALUES(1,'DELHI');
INSERT INTO WAREHOUSE VALUES(2,'BOMBAY');
INSERT INTO WAREHOUSE VALUES(3,'CHENNAI');
INSERT INTO WAREHOUSE VALUES(4,'BANGALORE');
INSERT INTO WAREHOUSE VALUES(5,'BANGALORE');
INSERT INTO WAREHOUSE VALUES(6,'DELHI');
INSERT INTO WAREHOUSE VALUES(7,'BOMBAY');
INSERT INTO WAREHOUSE VALUES(8,'CHENNAI');
INSERT INTO WAREHOUSE VALUES(9,'DELHI');
INSERT INTO WAREHOUSE VALUES(10,'BANGALORE');
INSERT INTO SHIPMENT VALUES(111,1,'2002-02-10');
INSERT INTO SHIPMENT VALUES(112,5,'2002-09-10');
INSERT INTO SHIPMENT VALUES(113,8,'2003-02-10');
INSERT INTO SHIPMENT VALUES(114,3,'2003-12-10');
INSERT INTO SHIPMENT VALUES(115,9,'2004-01-19');
INSERT INTO SHIPMENT VALUES(116,1,'2004-09-20');
INSERT INTO SHIPMENT VALUES(117,5,'2004-09-10');
INSERT INTO SHIPMENT VALUES(118,7,'2004-11-30');
INSERT INTO SHIPMENT VALUES(119,7,'2005-04-30');
INSERT INTO SHIPMENT VALUES(120,6,'2005-12-21');
select c.cname,count(o.orderno),avg(o.ordamt) from CUSTOMER c, ORDERS o where c.custno=o.custno group
by o.custno;
select s.orderno from SHIPMENT s, WAREHOUSE w where s.warehouseno=w.warehouseno and
w.city='bangalore';
delete from ITEM i where i.itemno=5001;
select * from ITEM;

```

OUTPUT





MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

malika

Tables

accident

author

car

catalog

category

orderdetails

owns

participated

person

publisher

Views

Stored Procedures

Functions

malikadb

sys

Administration Schemas

Information

No object selected

SQL File 3* bookdetails orderprocessing

Limit to 1000 rows

```

61 • INSERT INTO SHIPMENT VALUES(114,3,'2003-12-10');
62 • INSERT INTO SHIPMENT VALUES(115,9,'2004-01-19');
63 • INSERT INTO SHIPMENT VALUES(116,1,'2004-09-20');
64 • INSERT INTO SHIPMENT VALUES(117,5,'2004-09-10');
65 • INSERT INTO SHIPMENT VALUES(118,7,'2004-11-30');
66 • INSERT INTO SHIPMENT VALUES(119,7,'2005-04-30');
67 • INSERT INTO SHIPMENT VALUES(120,6,'2005-12-21');
68
69 • select c.cname,count(o.orderno),avg(o.ordamt) from CUSTOMER c, ORDERS o where c.custno=o.custno group by o.custno;
70 • select s.orderno from SHIPMENT s, WAREHOUSE w where s.warehouseno=w.warehouseno and w.city='bangalore';
71 • delete from ITEM i where i.itemno=5001;

```

Result Grid

cname	count(o.orderno)	avg(o.ordamt)
PUSHPA K	2	19000.0000
LAILA	2	17500.0000
FAIZAL	4	24000.0000
SOURAV	1	29000.0000
SUMAN	1	56000.0000

Result Grid

Form Editor

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

malika

Tables

accident

author

car

catalog

category

orderdetails

owns

participated

person

publisher

Views

Stored Procedures

Functions

malikadb

sys

Administration Schemas

Information

No object selected

SQL File 3* bookdetails orderprocessing

Limit to 1000 rows

```

62 • INSERT INTO SHIPMENT VALUES(115,9,'2004-01-19');
63 • INSERT INTO SHIPMENT VALUES(116,1,'2004-09-20');
64 • INSERT INTO SHIPMENT VALUES(117,5,'2004-09-10');
65 • INSERT INTO SHIPMENT VALUES(118,7,'2004-11-30');
66 • INSERT INTO SHIPMENT VALUES(119,7,'2005-04-30');
67 • INSERT INTO SHIPMENT VALUES(120,6,'2005-12-21');
68
69 • select c.cname,count(o.orderno),avg(o.ordamt) from CUSTOMER c, ORDERS o where c.custno=o.custno group by o.custno;
70 • select s.orderno from SHIPMENT s, WAREHOUSE w where s.warehouseno=w.warehouseno and w.city='bangalore';
71 • delete from ITEM i where i.itemno=5001;
72 • select * from ITEM;

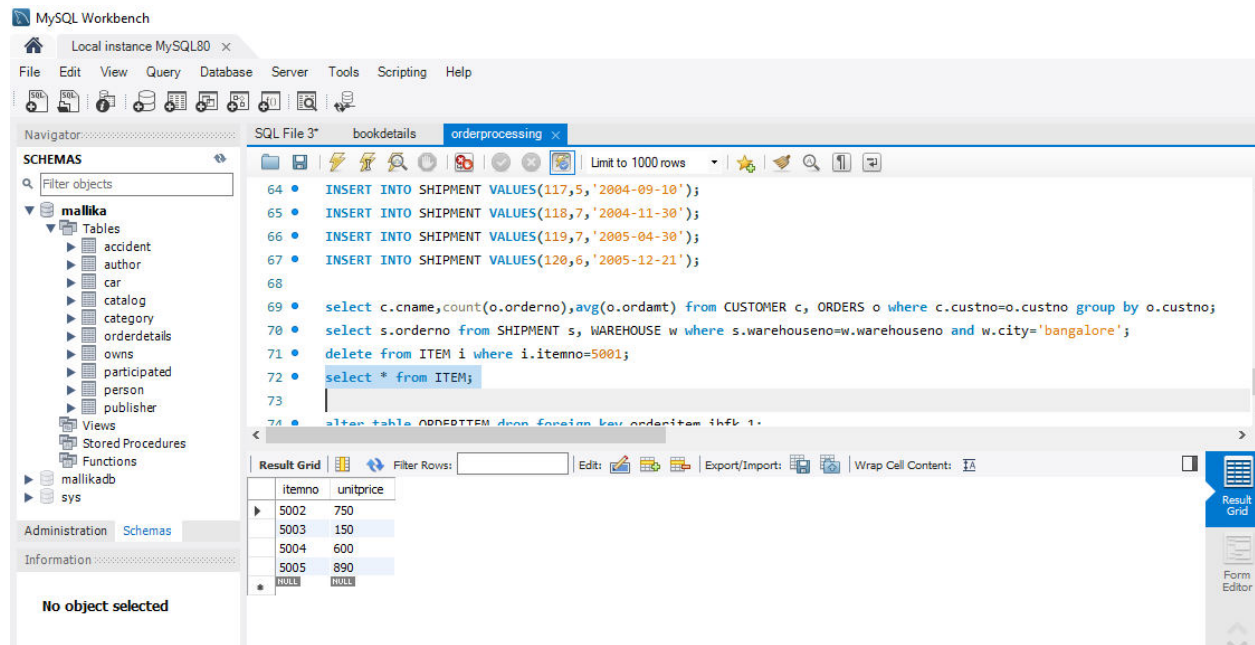
```

Result Grid

orderno
112
117

Result Grid

Form Editor



LAB 4

BANKDETAILS database

use

mallika;

```

create table branch( branch_name varchar(30) primary key, branch_city varchar(30), assets real);
create table accounts( accno int primary key, branch_name varchar(30), balance real,foreign key
(branch_name) references branch(branch_name) on delete cascade on update cascade);
create table customers( customer_name varchar(30) primary key,customer_street
varchar(20),customer_city varchar(20));
create table depositor(customer_name varchar(30),accno int,primary key(customer_name ,accno),foreign
key (accno) references accounts(accno) on delete cascade on update cascade,foreign key (customer_name)
references customers(customer_name) on delete cascade on update cascade);
create table loan( loan_number int primary key,branch_name varchar(30),amount real,foreign key
(branch_name) references branch(branch_name));
create table borrower (customer_name varchar(30),loan_number int,primary key(customer_name,
loan_number),foreign key (customer_name) references customers(customer_name) on delete cascade on
update cascade,foreign key (loan_number) references loan(loan_number) on delete cascade on update
cascade);
show tables;
insert into branch(branch_name,branch_city,assets) values('A','Bangalore',190000),
('B','Bangalore',200000),
('C','Delhi',235344),
('D','Chennai',1050560),
('E','Chennai',678909);
select *from branch;
insert into accounts(accno,branch_name,balance) VALUES
(1001,'A',10000),
(1002,'B',5000),

```

```

(1003,'C',7500),
(1004,'D',50000),
(1005,'D',75000),
(1006,'E',560),
(1007,"B",500),
(1008,"B",1500);
select *from accounts;
insert into customers(customer_name,customer_street,customer_city) VALUES
("Ravi","Dasarahalli","Bangalore"),
("Shyam","Indiranagar","Delhi"),
("Seema","Vasantnagar","Chennai"),
("Arpita","Church Street","Bangalore"),
("Vinay","MG Road","Chennai");
select *from customer;
insert into depositor(customer_name,accno) VALUES
("Ravi",1001),
("Ravi",1002),
("Shyam",1003),
("Seema",1004),
("Seema",1005),
("Arpita",1006),
("Vinay",1007),
("Vinay",1008);
select *from depositor;
insert into loan(loan_number,branch_name,amount) VALUES
(001,'A',10000),
(002,'B',25000),
(003,'B',250000),
(004,'C',5000),
(005,'E',90000);
select *from loan;
insert into borrower(customer_name,loan_number) VALUES
("Arpita",001),
("Ravi",002),
("Arpita",003),
("Shyam",004),
("Vinay",005);
select *from borrower;
select customer_name from depositor
join accounts on depositor.accno = accounts.accno where accounts.branch_name = "D"
group by depositor.customer_name having count(depositor.customer_name) >=2;
select customer_name from depositor
join accounts on accounts.accno = depositor.accno
join branch on branch.branch_name = accounts.branch_name
where branch.branch_city = "Bangalore"
GROUP BY depositor.customer_name
having count(DISTINCT branch.branch_name) = (SELECT COUNT(branch_name)
FROM branch
WHERE branch_city = 'Bangalore'));
delete from accounts where branch_name in
(select branch_name from branch where branch_city="Delhi");

```



```
select *from accounts;
```

Output

```
Query 1 x SQL File 2
Limit to 1000 rows
12 • insert into branch(branch_name,branch_city,assets) values('A','Bangalore',190000),
13 ('B','Bangalore',200000),
14 ('C','Delhi',235344),
15 ('D','Chennai',1050560),
16 ('E','Chennai',678909);
17 • select *from branch;
18
19 • insert into accounts(accno,branch_name,balance) VALUES
20 (1001,'A',10000),
21 (1002,'B',5000),
22 (1003,'C',7500),
23 (1004,'D',50000),
24 (1005,'D',75000),
25 (1006,'E',560),
26 (1007,'B',500),
27 (1008,'B',1500);
28 • select *from accounts;
29
30 • insert into customers(customer_name,customer_street,customer_city) VALUES
31 ("Ravi","Dasarahalli","Bangalore"),
32 ("Shyam","Indiranagar","Delhi"),
33 ("Seema","Vasantnagar","Chennai"),
```

```
Query 1 x SQL File 2
Limit to 1000 rows
12 • insert into branch(branch_name,branch_city,assets) values('A','Bangalore',190000),
13 ('B','Bangalore',200000),
14 ('C','Delhi',235344),
15 ('D','Chennai',1050560),
16 ('E','Chennai',678909);
17 • select *from branch;
18
19 • insert into accounts(accno,branch_name,balance) VALUES
20 (1001,'A',10000),
21 (1002,'B',5000),
22 (1003,'C',7500),
23 (1004,'D',50000),
24 (1005,'D',75000),
25 (1006,'E',560),
26 (1007,'B',500),
27 (1008,'B',1500);
28 • select *from accounts;
29
30 • insert into customers(customer_name,customer_street,customer_city) VALUES
31 ("Ravi","Dasarahalli","Bangalore"),
32 ("Shyam","Indiranagar","Delhi"),
33 ("Seema","Vasantnagar","Chennai"),
```

```
Query 1 x SQL File 2
Limit to 1000 rows

29
30 • insert into customers(customer_name,customer_street,customer_city) VALUES
31 ("Ravi","Dasarahalli","Bangalore"),
32 ("Shyam","Indiranagar","Delhi"),
33 ("Seema","Vasantnagar","Chennai"),
34 ("Arpita","Church Street","Bangalore"),
35 ("Vinay","MG Road","Chennai");
36 • select *from customer;
37
38 • insert into depositor(customer_name,accno) VALUES
39 ("Ravi",1001),
40 ("Ravi",1002),
41 ("Shyam",1003),
42 ("Seema",1004),
43 ("Seema",1005),
44 ("Arpita",1006),
45 ("Vinay",1007),
46 ("Vinay",1008);
47 • select *from depositor;
48
49
```

```
Query 1 x SQL File 2
Limit to 1000 rows

47 • select *from depositor;
48
49
50 • insert into loan(loan_number,branch_name,amount) VALUES
51 (001,'A',10000),
52 (002,'B',25000),
53 (003,'B',250000),
54 (004,'C',5000),
55 (005,'E',90000);
56 • select *from loan;
57
58 • insert into borrower(customer_name,loan_number) VALUES
59 ("Arpita",001),
60 ("Ravi",002),
61 ("Arpita",003),
62 ("Shyam",004),
63 ("Vinay",005);
64 • select *from borrower;
65
66
```

Query 1 x SQL File 2

Limit to 1000 rows

```
65
66
67
68 • select customer_name from depositor
69   join accounts on depositor.accno = accounts.accno where accounts.branch_name = "D"
70   group by depositor.customer_name having count(depositor.customer_name) >=2;
71
```

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

customer_name
Seema

Query 1 x SQL File 2

Limit to 1000 rows

```
74 • select customer_name from depositor
75   join accounts on accounts.accno = depositor.accno
76   join branch on branch.branch_name = accounts.branch_name
77   where branch.branch_city = "Bangalore"
78   GROUP BY depositor.customer_name
79   having count(DISTINCT branch.branch_name) = (SELECT COUNT(branch_name)
80   FROM branch
81   WHERE branch_city = 'Bangalore');
```

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

customer_name
Ravi

Result 3 x Read

Output

Query 1 x SQL File 2

Limit to 1000 rows

```

80 FROM branch
81 WHERE branch_city = 'Bangalore');
82
83 • delete from accounts where branch_name in
84 (select branch_name from branch where branch_city="Delhi");
85 • select *from accounts;

```

Result Grid

	accno	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

LAB 5

STUDENTENROLLMENT database

use

mallika;

```

create table STUDENT(regno varchar(30),name varchar(30),major varchar(30),bdate date, primary
key(regno));
create table COURSE(courseno int,cname varchar(30),dept varchar(30), primary key(courseno));
create table ENROLL(regno varchar(30),courseno int, marks int,sem int, foreign key(regno) references
STUDENT(regno) on delete cascade on update cascade, foreign key(courseno) references COURSE(courseno)
on delete cascade on update cascade);
create table BOOK_ADOPTION(courseno int,sem int, bookisbn int, foreign key(courseno) references
COURSE(courseno) on delete cascade on update cascade, foreign key(bookisbn) references
TEXTS(bookisbn));
create table TEXTS(bookisbn int,booktitle varchar(30),publisher varchar(30),author varchar(30),
primary key(bookisbn));
insert into STUDENT values('CS01','RAM','DS','1986-03-12');
insert into STUDENT values('IS02','SMITH','USP','1987-12-23');
insert into STUDENT values('EC03','AHMED','SNS','1985-04-17');
insert into STUDENT values('CS03','SNEHA','DBMS','1987-01-01');
insert into STUDENT values('TC05','AKHILA','EC','1986-10-06');
insert into COURSE values(11,'DS','CS');
insert into COURSE values(22,'USP','IS');
insert into COURSE values(33,'SNS','EC');
insert into COURSE values(44,'DBMS','CS');
insert into COURSE values(55,'EC','TC');
insert into ENROLL values('CS01',11,4,85);
insert into ENROLL values('IS02',22,6,80);

```

```

insert into ENROLL values('EC03',33,2,80);
insert into ENROLL values('CS03',44,6,75);
insert into ENROLL values('TC05',55,2,8);
insert into TEXTS values(1,'DS AND C','PRINCETON','PADMA REDDY');
insert into TEXTS values(2,'FUNDAMENTALS OF DS','PRINCETON','GODSE');
insert into TEXTS values(3,'FUNDAMENTALS OF DBMS','PRINCETON','NAVATHE');
insert into TEXTS values(4,'SQL','PRINCETON','FOLEY');
insert into TEXTS values(5,'ELECTRONIC CIRCUITS','TMH','ELMASRI');
insert into TEXTS values(6,'ADV UNIX PROG','TMH','STEVENS');
insert into BOOK_ADOPTION values(11,4,1);
insert into BOOK_ADOPTION values(11,4,2);
insert into BOOK_ADOPTION values(44,6,3);
insert into BOOK_ADOPTION values(44,6,4);
insert into BOOK_ADOPTION values(55,2,5);
insert into BOOK_ADOPTION values(22,6,6);
insert into TEXTS values(7,'GET STARTED WITH C','PRINCETON','THOMSON');
insert into BOOK_ADOPTION values(55,4,7);
select * from TEXTS;
select * from BOOK_ADOPTION;
select c.courseno,b.bookisbn,t.booktitle from BOOK_ADOPTION b,TEXTS t,COURSE c
where c.courseno=b.courseno
and b.bookisbn=t.bookisbn
and c.dept='CS'
group by b.courseno having count(b.courseno)>=2 order by t.booktitle;
select distinct c.dept from COURSE c, BOOK_ADOPTION b, TEXTS t
where c.courseno=b.courseno
and b.bookisbn=t.bookisbn
and t.publisher='PRINCETON';

```

Output

```

1 • use mallika;
2
3 • create table STUDENT(regno varchar(30),name varchar(30),major varchar(30),bdate date, primary key(regno));
4 • create table COURSE(courseno int,cname varchar(30),dept varchar(30), primary key(courseno));
5 • create table ENROLL(regno varchar(30),courseno int, marks int,sem int, foreign key(regno) references STUDENT(regno) c
6 • create table BOOK_ADOPTION(courseno int,sem int, bookisbn int, foreign key(courseno) references COURSE(courseno) on c
7 • create table TEXTS(bookisbn int,booktitle varchar(30),publisher varchar(30),author varchar(30), primary key(bookisbn)
8
9 • insert into STUDENT values('CS01','RAM','DS','1986-03-12');
10 • insert into STUDENT values('IS02','SMITH','USP','1987-12-23');
11 • insert into STUDENT values('EC03','AHMED','SNS','1985-04-17');
12 • insert into STUDENT values('CS03','SNEHA','DBMS','1987-01-01');
13 • insert into STUDENT values('TC05','AKHILA','EC','1986-10-06');
14
15 • insert into COURSE values(11,'DS','CS');
16 • insert into COURSE values(22,'USP','IS');
17 • insert into COURSE values(33,'SNS','EC');
18 • insert into COURSE values(44,'DBMS','CS');
19 • insert into COURSE values(55,'EC','TC');
20
21 • insert into ENROLL values('CS01',11,4,85);
22 • insert into ENROLL values('IS02',22,6,80);
23 • insert into ENROLL values('EC03',33,2,80);
24 • insert into ENROLL values('CS03',44,6,75);
25 • insert into ENROLL values('TC05',55,2,8);

```

```

19 • insert into COURSE values(55,'EC','TC');
20
21 • insert into ENROLL values('CS01',11,4,85);
22 • insert into ENROLL values('IS02',22,6,80);
23 • insert into ENROLL values('EC03',33,2,80);
24 • insert into ENROLL values('CS03',44,6,75);
25 • insert into ENROLL values('TC05',55,2,8);
26
27 • insert into TEXTS values(1,'DS AND C','PRINCETON','PADMA REDDY');
28 • insert into TEXTS values(2,'FUNDAMENTALS OF DS','PRINCETON','GOOSE');
29 • insert into TEXTS values(3,'FUNDAMENTALS OF DBMS','PRINCETON','NAVATHE');
30 • insert into TEXTS values(4,'SQL','PRINCETON','FOLEY');
31 • insert into TEXTS values(5,'ELECTRONIC CIRCUITS','TMH','ELMASRI');
32 • insert into TEXTS values(6,'ADV UNIX PROG','TMH','STEVENS');
33
34 • insert into BOOK_ADOPTION values(11,4,1);
35 • insert into BOOK_ADOPTION values(11,4,2);
36 • insert into BOOK_ADOPTION values(44,6,3);
37 • insert into BOOK_ADOPTION values(44,6,4);
38 • insert into BOOK_ADOPTION values(55,2,5);
39 • insert into BOOK_ADOPTION values(22,6,6);
40

```

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: bankdetails SQL File 2" x

Limit to 1000 rows

SCHEMAS

Filter objects

malika

Tables

- accident
- author
- car
- catalog
- category
- customer
- item
- orderdetails
- orderitem
- orders
- owns

Administration Schemas

Information

Schema: malika

```

40
41 • insert into TEXTS values(7,'GET STARTED WITH C','PRINCETON','THOMSON');
42 • insert into BOOK_ADOPTION values(55,4,7);
43 • select * from TEXTS;
44 • select * from BOOK_ADOPTION;
45
46 • select c.courseno,b.bookisbn,t.booktitle from BOOK_ADOPTION b,TEXTS t,COURSE c

```

Result Grid

bookisbn	booktitle	publisher	author
4	SQL	PRINCETON	FOLEY
5	ELECTRONIC CIRCUITS	TMH	ELMASRI
6	ADV UNIX PROG	TMH	STEVENS
7	GET STARTED WITH C	PRINCETON	THOMSON
* NULL	NULL	NULL	NULL

TEXTS 13 x

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

malika

Tables

- accident
- author
- car
- catalog
- category
- customer
- item
- orderdetails
- orderitem
- orders
- owns

Administration Schemas

Information

Schema: malika

bankdetails SQL File 2*

Limit to 1000 rows

```

40
41 • insert into TEXTS values(7,'GET STARTED WITH C','PRINCETON','THOMSON');
42 • insert into BOOK_ADOPTION values(55,4,7);
43 • select * from TEXTS;
44 • select * from BOOK_ADOPTION;
45
46 • select c.courseno,b.bookisbn,t.booktitle from BOOK_ADOPTION b,TEXTS t,COURSE c

```

Result Grid

courseno	sem	bookisbn
44	6	3
44	6	4
55	2	5
22	6	6
55	4	7

BOOK ADOPTION 14

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

malika

Tables

- accident
- author
- car
- catalog
- category
- customer
- item
- orderdetails
- orderitem
- orders
- owns

Administration Schemas

Information

Schema: malika

bankdetails SQL File 2*

Limit to 1000 rows

```

45
46 • select c.courseno,b.bookisbn,t.booktitle from BOOK_ADOPTION b,TEXTS t,COURSE c
47 where c.courseno=b.courseno
48 and b.bookisbn=t.bookisbn
49 and c.dept='CS'
50 group by b.courseno having count(b.courseno)>=2 order by t.booktitle;
51

```

Result Grid

courseno	bookisbn	booktitle
11	1	DS AND C
44	3	FUNDAMENTALS OF DBMS

Result 15

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

malika

Tables

- accident
- author
- car
- catalog
- category
- customer
- item
- orderdetails
- orderitem
- orders
- owns

Administration Schemas

Information

Schema: malika

bankdetails SQL File 2* x

Limit to 1000 rows

```
49 and c.dept='CS'
50 group by b.courseno having count(b.courseno)>=2 order by t.booktitle;
51
52 select distinct c.dept from COURSE c, BOOK_ADOPTION b, TEXTS t
53 where c.courseno=b.courseno
54 and b.bookisbn=t.bookisbn
55 and t.publisher='PRINCETON';
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content: ☒

dept
CS
TC

Result 16 x