

**1.** 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.

### Insurance Database

Create database insurance;

use insurance,;

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);  
show tables;  
insert into person values('1111','Ramu','K.S.Layout');  
insert into person values('2222','John','Indiranagar');  
insert into person values('3333','Priya','Jayanagar');  
insert into person values('4444','Gopal','Whitefield');  
insert into person values('5555','Latha','Vijaynagar');  
commit;  
select * from person;  
insert into car values('KA04Q2301','MARUTHI-DX','2000-11-10');
```

```
insert into car values('KA05P1000','FORDICON','2000-12-20');
insert into car values('KA03L1234','ZEN-VXI','1999-05-22');
insert into car values('KA03L9999','MARUTH-DX ','2002-04-12');
insert into car values('KA01P4020','INDICA-VX','2002-01-03');
commit;
select * from car;

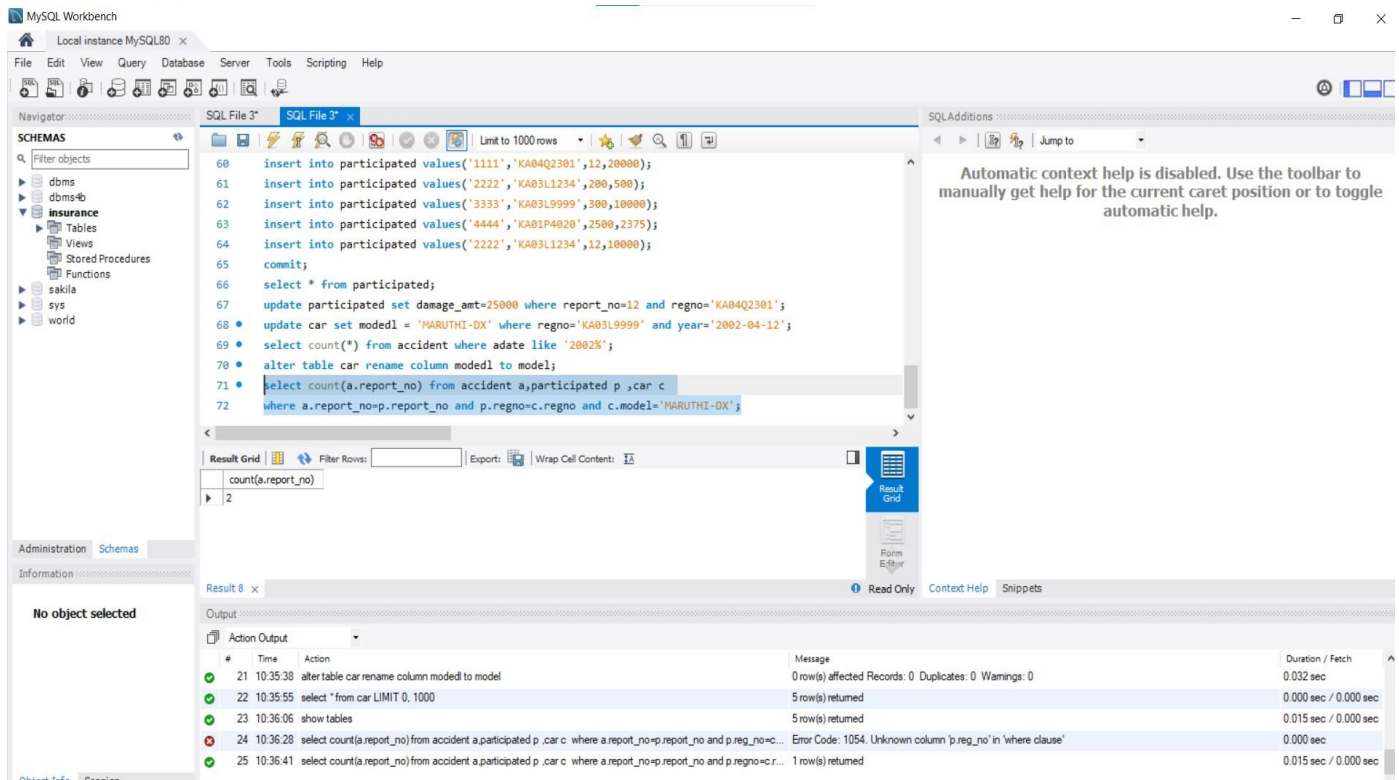
insert into accident values(12,'2002-06-01','MG road');
insert into accident values(200,'2002-12-10','Double road');
insert into accident values(300,'1999-07-23','MG road');
insert into accident values(2500,'2000-06-11','Residency road');
insert into accident values(2600,'2001-10-16','Richmond road');
commit;
select * from accident;

insert into owns values('1111','KA04Q2301');
insert into owns values('1111','KA05P1000');
insert into owns values('2222','KA03L1234');
insert into owns values('3333','KA03L9999');
insert into owns values('4444','KA01P4020');
commit;
select * from owns;

insert into participated values('1111','KA04Q2301',12,20000);
insert into participated values('2222','KA03L1234',200,500);
insert into participated values('3333','KA03L9999',300,10000);
insert into participated values('4444','KA01P4020',2500,2375);
insert into participated values('2222','KA03L1234',12,10000);
commit;
select * from participated;

update participated set damage_amt=25000 where report_no=12 and regno='KA04Q2301';
```

select count(\*) from accident where adate like '2002%';



2. 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)

- Create the above tables by properly specifying the primary keys and the foreign keys.
- Enter at least five tuples for each relation.
- 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.
- Find the author of the book which has maximum sales.
- Demonstrate how you increase the price of books published by a specific publisher by 10%.

## BookDealer Database

```
create database BookDealer;;

show databases;

use BookDealer;

CREATE TABLE AUTHOR(
AUTHOR_ID INT,
ANAME VARCHAR(30),
ACITY VARCHAR(30),
ACOUNTRY VARCHAR(30),
PRIMARY KEY(AUTHOR_ID));

CREATE TABLE PUBLISHER(
PUBLISHER_ID INT,
PNAME VARCHAR(20),
PCITY VARCHAR(30),
PCOUNTRY VARCHAR(30),
PRIMARY KEY(PUBLISHER_ID));

CREATE TABLE CATALOG(
BOOK_ID INT,
TITLE VARCHAR(30),
AUTHOR_ID INT,
PUBLISHER_ID INT,
CATEGORY_ID INT,
CYEAR INT,
PRICE INT,
PRIMARY KEY(BOOK_ID),
FOREIGN KEY(AUTHOR_ID) REFERENCES AUTHOR(AUTHOR_ID) ON DELETE CASCADE,
FOREIGN KEY(PUBLISHER_ID) REFERENCES PUBLISHER(PUBLISHER_ID) ON DELETE CASCADE,
FOREIGN KEY(CATEGORY_ID) REFERENCES CATEGORY(CATEGORY_ID) ON DELETE CASCADE);

CREATE TABLE CATEGORY(
CATEGORY_ID INT,
```

```
DESCRIPTION VARCHAR(30),
PRIMARY KEY(CATEGORY_ID));

CREATE TABLE ORDER_DETAILS(
ORDER_NO INT,
BOOK_ID INT,
QUANTITY INT,
PRIMARY KEY(ORDER_NO),
FOREIGN KEY(BOOK_ID) REFERENCES CATALOG(BOOK_ID) ON DELETE CASCADE);

SHOW TABLES;

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 B.P.','NEW YORK','USA');
insert into AUTHOR values(1005,'WILLIAM STALLINGS','LAS VEGAS','USA');

COMMIT;

select * from AUTHOR;

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','NEW DELHI','INDIA');
insert into PUBLISHER values(4,'WILLEY','BERLIN','GERMANY');
insert into PUBLISHER values(5,'MGH','NEW YORK','USA');

COMMIT;

select * from PUBLISHER;

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');

COMMIT;
```

```

select * from CATEGORY;

INSERT INTO CATALOG VALUES(11,'UNIX System Prg',1001,1,1001,2000,251);

insert into catalog values(12,'DS',1002,2,1003, 2001 ,425);

insert into catalog values(13,'LD',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 HB',1005,4,1001, 2000 ,658);

COMMIT;

select * from catalog;

insert into order_details values(1,11,5);

insert into order_details values(2,12,8);

insert into order_details values(3,13,15);

insert into order_details values(4,14,22);

insert into order_details values(5,15,3);

insert into order_details values(12,17,10);

commit;

select * from order_details;

SELECT A.ANAME FROM AUTHOR A,CATALOG C,ORDER_DETAILS O
WHERE A.AUTHOR_ID=C.AUTHOR_ID AND C.BOOK_ID=O.BOOK_ID
AND O.QUANTITY=(SELECT MAX(QUANTITY) FROM ORDER_DETAILS);

SELECT AUTHOR_ID, ANAME, ACITY, AOUNTRY FROM AUTHOR
WHERE AUTHOR_ID IN(SELECT AUTHOR_ID FROM CATALOG
WHERE cYEAR = 2000 GROUP BY AUTHOR_ID HAVING COUNT(AUTHOR_ID) >=2 );

UPDATE CATALOG SET PRICE=(PRICE+PRICE*0.1) WHERE PUBLISHER_ID=5;

```

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- book
- bookdealer
  - Tables
  - Views
  - Stored Procedures
  - Functions
- dbms
- dbms4b
- insurance
- sakila
- sys
- world

SQL File 3\* SQL File 3\* SQL File 5\* x SQL File 6\*

Limit to 1000 rows

```

66 • insert into catalog values(17,'coba1 HB',1005,4,1001, 2000 ,658);
67 • COMMIT;
68 • select * from catalog;
69 • insert into order_details values(1,11,5);
70 • insert into order_details values(2,12,8);
71 • insert into order_details values(3,13,15);
72 • insert into order_details values(4,14,22);
73 • insert into order_details values(5,15,3);
74 • insert into order_details values(12,17,10);
75 • commit;
76 • select * from order_details;
77 • SELECT A.ANAME FROM AUTHOR A,CATALOG C,ORDER_DETAILS O
78 • WHERE A.AUTHOR_ID=C.AUTHOR_ID AND C.BOOK_ID=O.BOOK_ID

```

Result Grid

ANAME
KARTHIK B.P.

Administration Schemas

Information

Result 2 x Read Only

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- book
- bookdealer
  - Tables
  - Views
  - Stored Procedures
  - Functions
- dbms
- dbms4b
- insurance
- sakila
- sys
- world

SQL File 3\* SQL File 3\* SQL File 5\* x SQL File 6\*

Limit to 1000 rows

```

73 • insert into order_details values(5,15,3);
74 • insert into order_details values(12,17,10);
75 • commit;
76 • select * from order_details;
77 • SELECT A.ANAME FROM AUTHOR A,CATALOG C,ORDER_DETAILS O
78 • WHERE A.AUTHOR_ID=C.AUTHOR_ID AND C.BOOK_ID=O.BOOK_ID
79 • AND O.QUANTITY=(SELECT MAX(QUANTITY) FROM ORDER_DETAILS);
80
81 • SELECT AUTHOR_ID, ANAME, ACITY, A.COUNTRY FROM AUTHOR
82 • WHERE AUTHOR_ID IN(SELECT AUTHOR_ID FROM CATALOG
83 • WHERE cYEAR = 2000 GROUP BY AUTHOR_ID HAVING COUNT(AUTHOR_ID) >=2 );
84
85 • UPDATE CATALOG SET PRICE=(PRICE+PRICE*0.1) WHERE PUBLISHER_ID=5;

```

Result Grid

BOOK_ID	TITLE	AUTHOR_ID	PUBLISHER_ID	CATEGORY_ID	CYEAR	PRICE
11	UNIX System Prg	1001	1	1001	2000	251
12	DS	1002	2	1003	2001	425
13	LD	1003	3	1002	1999	225
14	server prg	1004	4	1004	2001	333
15	linux os	1005	5	1005	2003	359
16	c++ bible	1005	5	1001	2000	579

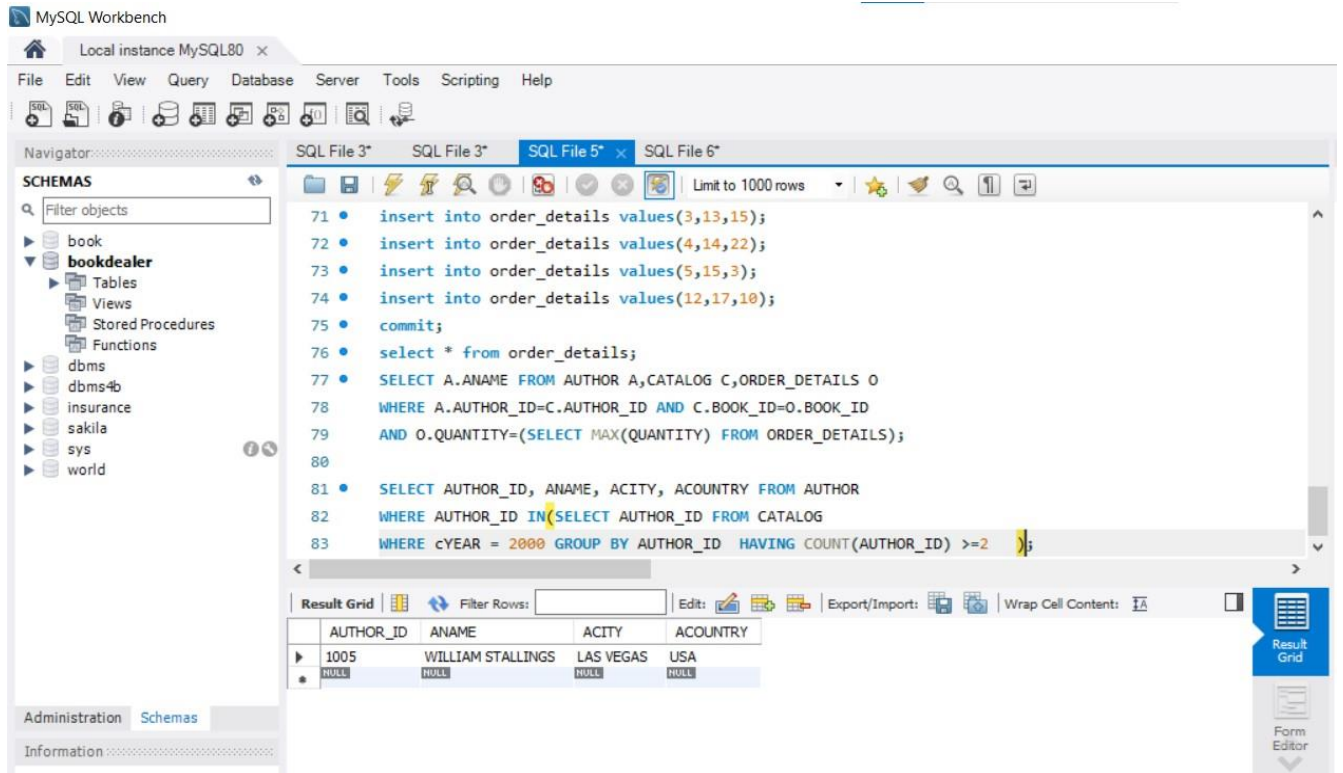
Administration Schemas

Information

No object selected

catalog 5 x Apply





- I. 3. Consider the following relations for an Order Processing database application in a company.
- II.
- III. **CUSTOMER** (CUST #: int, cname: String, city: String)
- IV. **ORDER** (order #: int, odate: date, cust #: int, ord-Amt: int)
- V. **ITEM** (item #: int, unit-price: int)
- VI. **ORDER-ITEM** (order #: int, item #: int, qty: int)
- VII. **WAREHOUSE** (warehouse #: int, city: String)
- VIII. **SHIPMENT** (order #: int, warehouse #: int, ship-date: date)
- IX.
- x. i) Create the above tables by properly specifying the primary keys and the foreign keys and the foreign
- XI. keys.
- XII. ii) Enter at least five tuples for each relation.
- XIII.
- XIV. iii) Produce a listing: CUSTNAME, #oforders, AVG\_ORDER\_AMT, where the middle column is the total

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

XVI.

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

XVIII.

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

XX. table.

XXI.

## Orderprocessing Database

```
create database orderprocessingDB;
use orderprocessingDB;
create table customer(
cust_no int,
cname varchar(20),
city varchar(30),
primary key(cust_no));
create table order1(
order_no int,
odate date,
cust_no int,
order_amt int,
primary key(order_no),
foreign key(cust_no) references customer(cust_no) on delete cascade);
create table item(
item_no int,
price int,
primary key(item_no));
```

```
create table order_item(  
order_no int,  
item_no int,  
qty int,  
foreign key(item_no) references item(item_no) on delete SET NULL,  
foreign key(order_no) references order1(order_no) on delete cascade);  
  
drop table order_item;  
  
create table warehouse(  
warehouse_no int,  
city varchar(20),  
primary key(warehouse_no));  
  
create table shipment(  
order_no int,  
warehouse_no int,  
ship_date date,  
foreign key(order_no) references order1(order_no) on delete cascade,  
foreign key(warehouse_no) references warehouse(warehouse_no) on delete cascade);  
  
show tables;  
  
insert into customer values(771,'PUSHPA K','BANGALORE');  
insert into customer values(772,'MUMBAI','MUMBAI');  
insert into customer values(773,'SOURAV','CALICUT');  
insert into customer values(774,'LAILA','HYDERABAD');  
insert into customer values(775,'FAIZAL','BANGALORE');  
  
commit;  
  
select * from customer;  
  
insert into order1 values(111,'2002-01-22',771,18000);  
insert into order1 values(112,'2002-07-30',774,6000);  
insert into order1 values(113,'2003-04-03',775,9000);  
insert into order1 values(114,'2003-11-03',775,29000);
```

```
insert into order1 values(115,'2003-12-10',773,29000);
insert into order1 values(116,'2004-08-19',772,56000);
insert into order1 values(117,'2004-09-10',771,20000);
insert into order1 values(118,'2004-11-20',775,29000);
insert into order1 values(119,'2005-02-13',774,29000);
insert into order1 values(120,'2005-10-13',775,29000);
commit;
select * from order1;
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);
commit ;
select * from item;
insert into order_item values(111,5001,50);
insert into order_item values(112,5003,20);
insert into order_item values(113,5002,50);
insert into order_item values(114,5005,60);
insert into order_item values(115,5004,90);
insert into order_item values(116,5001,10);
insert into order_item values(117,5003,80);
insert into order_item values(118,5005,50);
insert into order_item values(119,5002,10);
insert into order_item values(120,5004,45);
commit;
select * from order_item;
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');
commit;
```

```
select * from warehouse;

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');
commit;

select * from shipment;
```

```
select C.cname,COUNT(O.order_no),AVG(O.order_amt)
FROM CUSTOMER C,ORDER1 O WHERE C.cust_no=O.cust_no GROUP BY O.cust_no;
```

```
select o.order_no from order1 o, warehouse w, shipment s where
o.order_no=s.order_no and s.warehouse_no=w.warehouse_no and w.city='BOMBAY';
```

delete from item where item\_no=5003 ;

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- book
- bookdealer
  - Tables
  - Views
  - Stored Procedures
  - Functions
- dbms
- dbms4b
- insurance
- sakila
- sys
- world

SQL File 3\* SQL File 3\* SQL File 5\* SQL File 6\* x

```
94 • insert into shipment values(118,7,'2004-11-30');
95 • insert into shipment values(119,7,'2005-04-30');
96 • insert into shipment values(120,6,'2005-12-21');
97 • commit;
98 • select * from shipment;
99
100 • select C.cname,COUNT(O.order_no),AVG(O.order_amt)
101 FROM CUSTOMER C,ORDER1 O WHERE C.cust_no=O.cust_no GROUP BY O.cust_no;
102
103 • select o.order_no from order1 o, warehouse w, shipment s where
104 o.order_no=s.order_no and s.warehouse_no=w.warehouse_no and w.city='BOMBAY';
105
106 • delete from item where item_no=5003 ;
```

Result Grid

order_no	item_no	qty
111	5001	50
112	5001	20
114	5005	60
115	5004	90
116	5001	10
117	5001	80

Administration Schemas

Information

order\_item 28 x

Read Only

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- book
- bookdealer
  - Tables
  - Views
  - Stored Procedures
  - Functions
- dbms
- dbms4b
- insurance
- sakila
- sys
- world

SQL File 3\* SQL File 3\* SQL File 5\* SQL File 6\* x

```
88 • insert into shipment values(112,5,'2002-09-10');
89 • insert into shipment values(113,8,'2003-02-10');
90 • insert into shipment values(114,3,'2003-12-10');
91 • insert into shipment values(115,9,'2004-01-19');
92 • insert into shipment values(116,1,'2004-09-20');
93 • insert into shipment values(117,5,'2004-09-10');
94 • insert into shipment values(118,7,'2004-11-30');
95 • insert into shipment values(119,7,'2005-04-30');
96 • insert into shipment values(120,6,'2005-12-21');
97 • commit;
98 • select * from shipment;
99
100 • select C.cname,COUNT(O.order_no),AVG(O.order_amt)
```

Result Grid

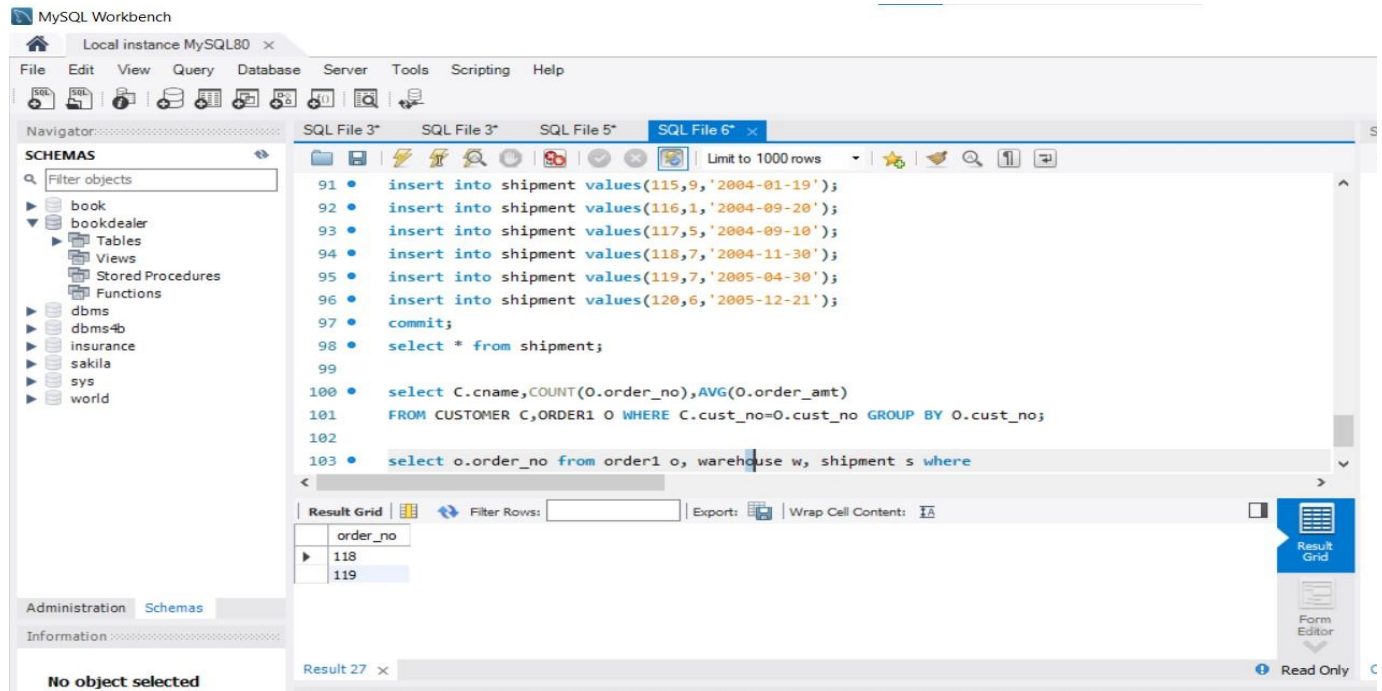
cname	COUNT(O.order_no)	AVG(O.order_amt)
PUSHPA K	2	19000.0000
LAILA	2	17500.0000
FAIZAL	4	24000.0000
SOURAV	1	29000.0000
MUMBAI	1	56000.0000

Administration Schemas

Information

Result 26 x

Read Only



4. 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.

## Banking\_enterprise Database

```
create database Banking_enterprise;
```

```
use Banking_enterprise;
```

```
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 customer(  
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 customer(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 customer(customer_name) on delete cascade on update  
cascade,  
foreign key (loan_number) references loan(loan_number) on delete cascade on update cascade);
```

```
insert into branch(branch_name,branch_city,assets) values  
( 'A','Bangalore',190000),  
( 'B','Bangalore',200000),  
( 'C','Delhi',235344),  
( 'D','Chennai',1050560),  
( 'E','Chennai',678909);
```

```
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);
```

```
insert into customer(customer_name,customer_street,customer_city) VALUES
```

```
("Ravi","Dasarahalli","Bangalore"),  
("Shyam","Indiranagar","Delhi"),  
("Seema","Vasantnagar","Chennai"),  
("Arpita","Church Street","Bangalore"),  
("Vinay","MG Road","Chennai");
```

```
insert into depositor(customer_name,accno) VALUES
```

```
("Ravi",1001),  
("Ravi",1002),  
("Shyam",1003),  
("Seema",1004),  
("Seema",1005),  
("Arpita",1006),  
("Vinay",1007),  
("Vinay",1008);
```

```
insert into loan(loan_number,branch_name,amount) VALUES
```

```
(001,'A',10000),  
(002,'B',25000),  
(003,'B',250000),  
(004,'C',5000),
```

```
(005,'E',90000);
```

```
insert into borrower(customer_name,loan_number) VALUES
```

```
("Arpita",001),
```

```
("Ravi",002),
```

```
("Arpita",003),
```

```
("Shyam",004),
```

```
("Vinay",005);
```

```
SELECT * FROM BRANCH;
```

```
SELECT * FROM ACCOUNTS;
```

```
SELECT * FROM DEPOSITOR;
```

```
SELECT * FROM LOAN;
```

```
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");
```

# MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Filter objects

- banking
- book
- bookdealer
- dbms
- dbms4b
- insurance
- orderprocessingdb
- sakila
- sys
- world

SQL File 3\* SQL File 3\* SQL File 5\* SQL File 6\* SQL File 6\* SQL File 7\* x

Limit to 1000 rows

```

100 group by depositor.customer_name having count(depositor.customer_name) >=2;
101
102 • select customer_name from depositor
103 join accounts on accounts.accno = depositor.accno
104 join branch on branch.branch_name = accounts.branch_name
105 where branch.branch_city = "Bangalore"
106 GROUP BY depositor.customer_name
107 having count(DISTINCT branch.branch_name) = (SELECT COUNT(branch_name)
108 FROM branch
109 WHERE branch_city = 'Bangalore');
110
111 • delete from accounts where branch_name in
112 (select branch_name from branch where branch_city="Delhi");

```

Result Grid

customer_name
Ravi

Administration Schemas

Information

Result 7 x Read Only

# MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Filter objects

- banking
- book
- bookdealer
- dbms
- dbms4b
- insurance
- orderprocessingdb
- sakila
- sys
- world

SQL File 3\* SQL File 3\* SQL File 5\* SQL File 6\* SQL File 6\* SQL File 7\* x

Limit to 1000 rows

```

93 • SELECT * FROM ACCOUNTS;
94 • SELECT * FROM DEPOSITOR;
95 • SELECT * FROM LOAN;
96 • SELECT * FROM BORROWER;
97
98 • select customer_name from depositor
99 join accounts on depositor.accno = accounts.accno where accounts.branch_name = "D"
100 group by depositor.customer_name having count(depositor.customer_name) >=2;
101
102 • select customer_name from depositor
103 join accounts on accounts.accno = depositor.accno
104 join branch on branch.branch_name = accounts.branch_name
105 where branch.branch_city = "Bangalore"

```

Result Grid

customer_name
Seema

Administration Schemas

Information

Result 6 x Read Only

**5. 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)

- viii. Create the above tables by properly specifying the primary keys and the foreign keys.
- ix. Enter at least five tuples for each relation.
- x. Demonstrate how you add a new text book to the database and make this book be adopted by some department.
- xi. 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.
- xii. List any department that has *all* its adopted books published by a specific publisher.

## Student Enrollment Database

```
create database student_enrollment1;

use student_enrollment1;

create table student(
regno varchar(20),
sname varchar(30),
major varchar(20),
bdate date,
primary key(regno));

create table course(
course_no int,
cname varchar(30),
```

```
dept varchar(30),
primary key(course_no));
create table enroll(
regno varchar(30),
course_no int,
sem int,
marks int,
foreign key(regno) references student(regno) on delete cascade on update cascade,
foreign key(course_no) references course(course_no) on delete cascade on update cascade);
create table book_adoption(
course_no int,
sem int,
book_ISBN int,
foreign key(book_ISBN) references text(book_ISBN) on delete cascade on update cascade,
foreign key(course_no) references course(course_no) on delete cascade on update cascade);
create table text(
book_ISBN int,
book_title varchar(30),
publisher varchar(30),
author varchar(30),
primary key(book_ISBN)
);
drop table book_adoption;
drop table text;
show tables;
insert into student(regno,sname,major,bdate) values
("CS01","RAM","DS","1986-03-12"),
("IS02","SMITH","USP","1987-12-23"),("EC03","AHMED","SNS","1985-04-17"),
("CS03","SNEHA","DBMS","1987-01-01"),("TC05","AKHILLA","EC","1986-10-16");
```

insert into course(course\_no,cname,dept) values

(11,"DS","CS"),(22,"USP","IS"),(33,"SNS","EC"),(44,"DBMS","CS"),(55,"EC","TC");

insert into enroll(regno,course\_no,sem,marks) values

("CS01",11,4,85),("IS02",22,6,80),("EC03",33,4=2,80),("CS03",44,6,75),("TC05",55,2,8);

insert into book\_adoption(course\_no,sem,book\_ISBN) values

(11,4,1),(11,4,2),(44,6,3),(44,6,4),(55,2,5),(22,6,6);

Insert into text value(1,"DS AND C","PRINCETON","PADMA REDDY");

insert into text value(2,"FUNDAMENTALS OF DS","PRINCETON","GODSE");

insert into text value(3,"FUNDAMENTALS OF DBMS","PRINCETON","NAVATHE");

insert into text value(4,"SQL","PRINCETON","FOLEY");

insert into text value(5,"ELECTRONIC CIRCUITS","TMH","ELMASRI");

insert into text value(6,"ADV UNIX PROG","TMH","STEVENS");

select \* from student;

select \* from course;

select \* from enroll;

select \* from book\_adoption;

select \* from text;

insert into text value(7,"FUNDAMENTALS OF EC","IIT","PANDEY");

insert into book\_adoption value(55,2,7);

select distinct c.dept from course c,text t,book\_adoption b where t.book\_ISBN=b.book\_ISBN and  
b.course\_no=c.course\_no and t.publisher="PRINCETON";



```
SELECT C.course_NO,T.book_ISBN,T.book_title FROM TEXT T,COURSE C,BOOK_ADOPTION B WHERE  
T.BOOK_ISBN=B.BOOK_ISBN AND B.COURSE_NO=C.COURSE_NO AND C.DEPT="CS" AND  
(SELECT COUNT(B.BOOK_ISBN) FROM BOOK_ADOPTION B WHERE C.COURSE_NO=B.COURSE_NO)>=2  
ORDER BY T.BOOK_TITLE;
```

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- banking
- banking\_enterprise
- book
- bookdealer
- dbms
- dbms4b
- insurance
- orderprocessingdb
- sakila
- student\_enrollment
- student\_enrollment1**
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys
- world

Administration Schemas

Information

SQL File 3\* SQL File 3\* SQL File 5\* SQL File 6\* SQL File 6\* SQL File 7\* SQL File 8\* x

Limit to 1000 rows

```
58
59 • select * from student;
60 • select * from course;
61 • select * from enroll;
62 • select * from book_adoption;
63 • select * from text;
64
65 • insert into text value(7,"FUNDAMENTALS OF EC","IIT","PANDEY");
66 • insert into book_adoption value(55,2,7);
67
68 • select distinct c.dept from course c,text t,book_adoption b where t.book_ISBN=b.book_ISBN and
69 b.course_no=c.course_no and t.publisher="PRINCETON";
70
```

Result Grid

	course_no	sem	book_ISBN
	11	4	2
	44	6	3
	44	6	4
	55	2	5
	22	6	6
	55	2	7

book\_adoption 10

mysql workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- banking
- banking\_enterprise
- book
- bookdealer
- dbms
- dbms4b
- insurance
- orderprocessingdb
- sakila
- student\_enrollment
- student\_enrollment1**
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys
- world

Administration Schemas

Information

SQL File 3\* SQL File 3\* SQL File 5\* SQL File 6\* SQL File 6\* SQL File 7\* SQL File 8\* x

Limit to 1000 rows

```
57
58
59 • select * from student;
60 • select * from course;
61 • select * from enroll;
62 • select * from book_adoption;
63 • select * from text;
64
65 • insert into text value(7,"FUNDAMENTALS OF EC","IIT","PANDEY");
66 • insert into book_adoption value(55,2,7);
67
68 • select distinct c.dept from course c,text t,book_adoption b where t.book_ISBN=b.book_ISBN and
69 b.course_no=c.course_no and t.publisher="PRINCETON";
```

Result Grid

	dept
	CS

Result 11

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- banking
- banking\_enterprise
- book
- bookdealer
- dbms
- dbms4b
- insurance
- orderprocessingdb
- sakila
- student\_enrollment
- student\_enrollment1
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys
- world

SQL File 3\* SQL File 3\* SQL File 5\* SQL File 6\* SQL File 6\* SQL File 7\* SQL File 8\* x

Limit to 1000 rows

```
63 • select * from text;
64
65 • insert into text value(7,"FUNDAMENTALS OF EC","IIT","PANDEY");
66 • insert into book_adoption value(55,2,7);
67
68 • select distinct c.dept from course c,text t,book_adoption b where t.book_ISBN=b.book_ISBN and
69 b.course_no=c.course_no and t.publisher="PRINCETON";
70
71 • SELECT C.course_NO,T.book_ISBN,T.book_title FROM TEXT T,COURSE C,BOOK_ADOPTION B WHERE
72 T.BOOK_ISBN=B.BOOK_ISBN AND B.COURSE_NO=C.COURSE_NO AND C.DEPT="CS" AND
73 (SELECT COUNT(B.BOOK_ISBN) FROM BOOK_ADOPTION B WHERE C.COURSE_NO=B.COURSE_NO)>=2 ORDER BY T.BOOK_TI
74
75
```

Result Grid

Filter Rows: Export: Wrap Cell Content: I A

	course_NO	book_ISBN	book_title
▶	11	1	DS AND C
	44	3	FUNDAMENTALS OF DBMS
	11	2	FUNDAMENTALS OF DS
	44	4	SQL

Administration Schemas

Information

Result 12 x Read Only