DBMS LAB REPORT
(PROGRAM 1-10)

NAME:VAIBHAVI PATIL
USN:1BM19CS217
CSE SECTION 'A'

PROGRAM 1: INSURANCE DATABASE

Consider the Insurance database given below.
The data types are specified.
PERSON (driver_id: String, name: String, address: String)
CAR (reg_num: String, model: String, year: int)
ACCIDENT (report_num: int, accident_date: date, location: String)
OWNS (driver_id: String, reg_num: String)
PARTICIPATED (driver_id: String,reg_num: String, report_num: 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.

```
create database insudb;
use insudb;

create table PERSON(
driver_id varchar(10) NOT NULL,name varchar(10) NOT NULL,address varchar(20) NOT
NULL,primary key (driver_id)
);
create table  CAR(
Regno varchar(10) NOT NULL,model varchar(10) NOT NULL,year int NOT NULL,primary key
(Regno)
);
create table ACCIDENT(
report_number int NOT NULL,accdate date,location varchar(20),primary key (report_number)
);
create table OWNS(
driver_id varchar(10),Regno varchar(10),primary key (driver_id,Regno),foreign key (driver_id)
references PERSON (driver_id),foreign key (Regno) references CAR (regno)
);
create table PARTICIPATED(
driver_id varchar(10),Regno varchar(10),report_number int,damage_amount int,primary key
(driver_id,Regno,report_number),foreign key (driver_id) references PERSON (driver_id),foreign
key (Regno) references car (Regno),
foreign key (report_number) references ACCIDENT (report_number)
);

insert into PERSON values("A01","Richard","Srinivas nagar"),
("A02","Pradeep","Rajaji nagar"),
("A03","Smith","Ashok nagar"),
("A04","Venu","NR Colony"),
("A05","Jhon","Hanumanth nagar");
```

```
insert into CAR values("KA052250","Indica","1990"),
("KA031181","Lancer","1957"),
("KA095477","Toyota","1998"),
("KA053408","Honda","2008"),
("KA041702","Audi","2005");


insert into OWNS VALUES("A01","KA052250"),
("A02","KA053408"),
("A03","KA031181"),
("A04","KA095477"),
("A05","KA041702");

insert into ACCIDENT values("11","2003-01-01","Mysore Road"),
("12","2004-02-02","Southend circle"),
("13","2003-01-21","Bull temple road"),
("14","2008-02-17","Mysore Road"),
("15","2005-03-04","Kanakpura road");

insert into PARTICIPATED values ("A01","KA052250","11","10000"),
 ("A02","KA053408","12","50000"),
 ("A03","KA095477","13","25000"),
 ("A04","KA031181","14","3000"),
 ("A05","KA041702","15","5000");
```

select * from person;
PERSON

driver_id	name	address
A01	Richard	Srinivas nagar
A02	Pradeep	Rajaji nagar
A03	Smith	Ashok nagar
A04	Venu	NR Colony
A05	Jhon	Hanumanth nagar

person 3 ✕

select * from car;
CAR

Regno	model	year
KA031181	Lancer	1957
KA041702	Audi	2005
KA052250	Indica	1990
KA053408	Honda	2008
KA095477	Toyota	1998

Select * from owns;
OWNS

driver_id	Regno
A03	KA031181
A05	KA041702
A01	KA052250
A02	KA053408
A04	KA095477

OWNS 4 ✕

Select * from accident;

ACCIDENT

	report_number	accdate	location
▶	11	2003-01-01	Mysore Road
	12	2004-02-02	Southend circle
	13	2003-01-21	Bull temple road
	14	2008-02-17	Mysore Road
	15	2005-03-04	Kanakpura road

accident 1 ✕

Select * from participated;

PARTICIPATED

	driver_id	Regno	report_number	damage_amount
▶	A01	KA052250	11	10000
	A02	KA053408	12	50000
	A03	KA095477	13	25000
	A04	KA031181	14	3000
	A05	KA041702	15	5000

participated 5 ✕

iii)Demonstrate how you
a.Update the damage amount to 25000 for the car with a specific reg-num(example 'K
A053408') for which the accident report number was 12.

Ans : update participated set damage_amount = 25000 where report_number = 12 and Regno = "KA053408";

	driver_id	Regno	report_number	damage_amount
▶	A01	KA052250	11	10000
	A02	KA053408	12	25000
	A03	KA095477	13	25000
	A04	KA031181	14	3000
	A05	KA041702	15	5000
✱	NULL	NULL	NULL	NULL

b.Add a new accident to the database.

Ans : insert into ACCIDENT values("16","2001-11-09","Majestic circle");

report_number	accdate	location
11	2003-01-01	Mysore Road
12	2004-02-02	Southend circle
13	2003-01-21	Bull temple road
14	2008-02-17	Mysore Road
15	2005-03-04	Kanakpura road
16	2001-11-09	Majestic circle
NULL	NULL	NULL

iv)Find the total number of people who owned cars that were involved in accidents in 2008.

Ans : select count(*) from person p,accident ac,participated pa where (p.driver_id=pa.driver_id) and (ac.report_number=pa.report_number) and (accdate like "2008%");

no_of_accidents_in_2008
1

Result 10 ×

v)Find the number of accidents in which cars belonging to a specific model (example Indica)were involved.

select count(*) as accident_beloging_to_model_Indica from car c,accident ac,participated pa where(c.Regno=pa.Regno) and (ac.report_number=pa.report_number) and c.model="Indica";

accident_beloging_to_model_Indica
1

PROGRAM 2: BANKING ENTERPRISE DATABASE

Consider the following database for a banking enterprise.

Branch (branch-name: String, branch-city: String, assets: real)
BankAccount(accno: int, branch-name: String, balance: real)
BankCustomer (customer-name: String, customer-street: String, customer-city: String)
Depositer(customer-name: String, accno: int)
Loan (loan-number: int, branch-name: String, amount: real)

i. Create the above tables by properly specifying the primary keys and the foreign keys.
ii. Enter at least five tuples for each relation.

```
create database bankingdatabase1;
use bankingdatabase1;
create table BRANCH(
branch_name varchar(20) NOT NULL,branch_city varchar(20) NOT NULL,assets real,primary
key(branch_name)
);
create table BANKACCOUNT(
accno int NOT NULL,branch_name varchar(20) NOT NULL,balance real,primary key(accno),
foreign key (branch_name) references BRANCH(branch_name)on delete cascade
);

create table BANKCUSTOMER(
customer_name varchar(20) ,customer_street varchar(25) NOT NULL,customer_city
varchar(20) NOT NULL,primary key(customer_name)
);
create table DEPOSITOR(
customer_name varchar(20),
accno int,
primary key(customer_name,accno),
foreign key(customer_name) references bankcustomer(customer_name),
foreign key (accno) references BANKACCOUNT(accno)
);
create table LOAN(
loan_number int,branch_name varchar(20),amount real,primary key(loan_number),
foreign key (branch_name) references branch(branch_name)
);


insert into branch values("SBI_Chamrajpet","Banglore","50000"),
("SBI_ResidencyRoad","Banglore","10000"),
("SBI_ShivajiRoad","Bombay","20000"),
("SBI_ParlimentRoad","Delhi","10000"),
```

```sql
("SBI_Jantarmantar","Delhi","20000");
select * from branch;

insert into bankaccount values("1","SBI_Chamrajpet","2000"),
("2","SBI_ResidencyRoad","5000"),
("3","SBI_ShivajiRoad","6000"),
("4","SBI_ParlimentRoad","9000"),
("5","SBI_Jantarmantar","8000"),
("6","SBI_ShivajiRoad","4000"),
("8","SBI_ResidencyRoad","4000"),("9","SBI_ParlimentRoad","3000"),
("10","SBI_ResidencyRoad","5000"),("11","SBI_Jantarmantar","2000");
select * from bankaccount;

insert into bankcustomer values("Avinash","Bull_Temple_Road","Banglore"),
("Dinesh","Baneergatta_Road","Banglore"),
("Mohan","NationalCollege_Road","Banglore"),
("Nikil","Akbar_Road","Delhi"),
("Ravi","Prithviraj_Road","Delhi");
select * from bankcustomer;

insert into depositor values("Avinash","1"),
("Dinesh","2"),
("Nikil","4"),
("Ravi","5"),
("Avinash","8"),
("Nikil","9"),
("Dinesh","10"),
("Nikil","11");
select * from depositor;

insert into loan values("1","SBI_Chamrajpet","1000"),
("2","SBI_ResidencyRoad","2000"),
("3","SBI_ShivajiRoad","3000"),
("4","SBI_ParlimentRoad","4000"),
("5","SBI_Jantarmantar","5000");
select * from loan;
```
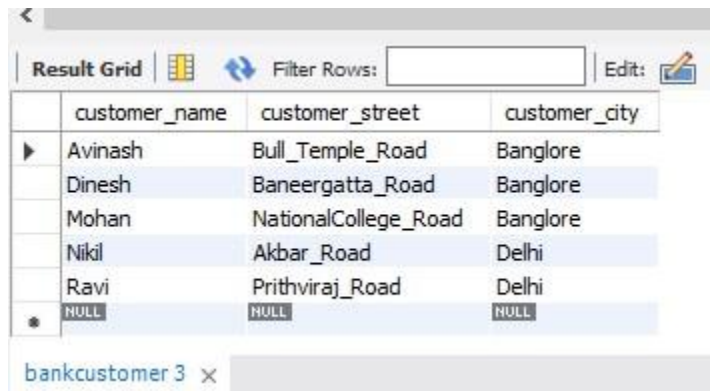
Select * from branch;
BRANCH

branch_name	branch_city	assets
SBI_Chamrajpet	Banglore	50000
SBI_Jantarmantar	Delhi	20000
SBI_ParlimentRoad	Delhi	10000
SBI_ResidencyRoad	Banglore	10000
SBI_ShivajiRoad	Bombay	20000
NULL	NULL	NULL

Select * from bankaccount;
BANKACCOUNT

accno	branch_name	balance
1	SBI_Chamrajpet	2000
2	SBI_ResidencyRoad	5000
3	SBI_ShivajiRoad	6000
4	SBI_ParlimentRoad	9000
5	SBI_Jantarmantar	8000
6	SBI_ShivajiRoad	4000
8	SBI_ResidencyRoad	4000
9	SBI_ParlimentRoad	3000
10	SBI_ResidencyRoad	5000
11	SBI_Jantarmantar	2000
NULL	NULL	NULL
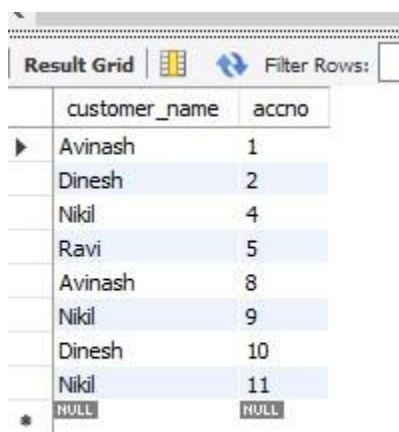
Select * from bankcustomer;
BANKCUSTOMER

customer_name	customer_street	customer_city
Avinash	Bull_Temple_Road	Banglore
Dinesh	Baneergatta_Road	Banglore
Mohan	NationalCollege_Road	Banglore
Nikil	Akbar_Road	Delhi
Ravi	Prithviraj_Road	Delhi
NULL	NULL	NULL

bankcustomer 3 ✕

Select * from depositor;
DEPOSITOR

customer_name	accno
Avinash	1
Dinesh	2
Nikil	4
Ravi	5
Avinash	8
Nikil	9
Dinesh	10
Nikil	11
NULL	NULL

Select * from loan;
LOAN

loan_number	branch_name	amount
1	SBI_Chamrajpet	1000
2	SBI_ResidencyRoad	2000
3	SBI_ShivajiRoad	3000
4	SBI_ParlimentRoad	4000
5	SBI_Jantarmantar	5000
NULL	NULL	NULL

loan 1 ✕

iii. Find all the customers who have at least two accounts at the *Main* branch (ex. SBI_ResidencyRoad).

select d.customer_name from depositor d ,bankaccount a where a.accno=d.accno and a.branch_name="SBI_ResidencyRoad" group by d.customer_name having count(*)>=2;

customer_name
Dinesh

iv. Find all the customers who have an account at *all* the branches located in a specific city (Ex. Delhi).

select d.customer_name from bankaccount a, depositor d,branch b where d.accno=a.accno and b.branch_name=a.branch_name and b.branch_city="Delhi" group by d.customer_name having count(distinct b.branch_name)=(select count(branch_name) from branch where branch_city="Delhi");

customer_name
Nikil

v. Demonstrate how you delete all account tuples at every branch located in a specific city (Ex. Bombay).

delete from bankaccount where branch_name in(select branch_name from branch where branch_city="Bombay");

accno	branch_name	balance
1	SBI_Chamrajpet	2000
2	SBI_ResidencyRoad	5000
4	SBI_ParlimentRoad	9000
5	SBI_Jantarmantar	8000
8	SBI_ResidencyRoad	4000
9	SBI_ParlimentRoad	3000
10	SBI_ResidencyRoad	5000
11	SBI_Jantarmantar	2000
NULL	NULL	NULL

bankaccount 8 ×

PROGRAM 3: SUPPLIER DATABASE

Consider the following schema:

SUPPLIERS(<u>sid: integer</u>, sname: string, address: string)

PARTS(<u>pid: integer</u>, pname: string, color: string)

CATALOG(<u>sid: integer, pid</u>: integer, cost: real)

The Catalog relation lists the prices charged for parts by Suppliers.

```
create database  supplierdb1;
use supplierdb1;

create table suppliers(
sid int,
sname varchar(20),
address varchar(25),
primary key(sid)
);

create table parts(
pid int,
pname varchar(20),
color varchar(20),
primary key(pid)
);

create table catalog(
sid int,
pid int,
cost real,
primary key(sid,pid),
foreign key (sid)references suppliers (sid),
foreign key (pid)references parts (pid));


insert into suppliers values ("10001","Acme Widget","Banglore");
insert into suppliers values ("10002","Johns","Kolkata");
```
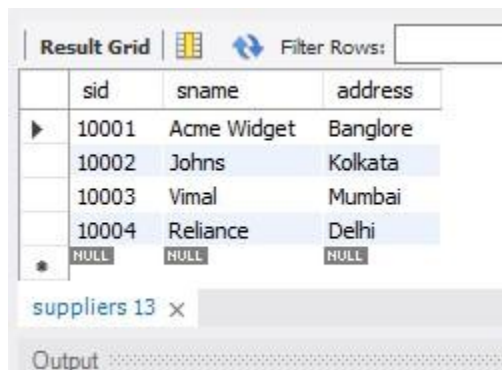
insert into suppliers values ("10003","Vimal","Mumbai");
insert into suppliers values ("10004","Reliance","Delhi");

insert into parts values ("20001","Book","Red");
insert into parts values ("20002","Pen","Red");
insert into parts values ("20003","Pencil","Green");
insert into parts values ("20004","Mobile","Green");
insert into parts values ("20005","Charger","Black");

insert into catalog values ("10001","20001","10");
insert into catalog values ("10001","20002","10");
insert into catalog values ("10001","20003","30");
insert into catalog values ("10001","20004","10");
insert into catalog values ("10001","20005","10");
insert into catalog values ("10002","20001","10");
insert into catalog values ("10002","20002","20");
insert into catalog values ("10003","20003","30");
insert into catalog values ("10004","20003","40");
insert into catalog values ("10003","20002","10");

Select * from suppliers;
SUPPLIERS
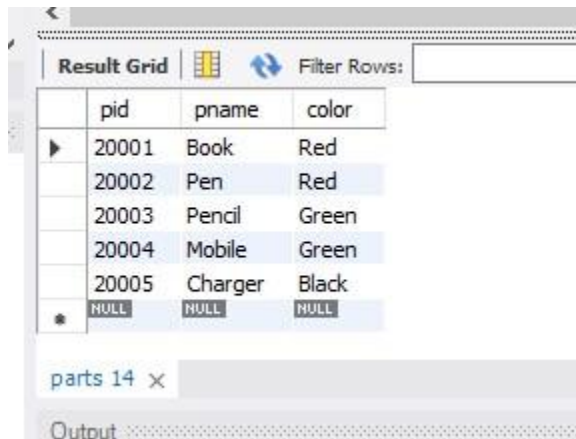
sid	sname	address
10001	Acme Widget	Banglore
10002	Johns	Kolkata
10003	Vimal	Mumbai
10004	Reliance	Delhi
NULL	NULL	NULL

suppliers 13 ×

Output

Select * from parts;
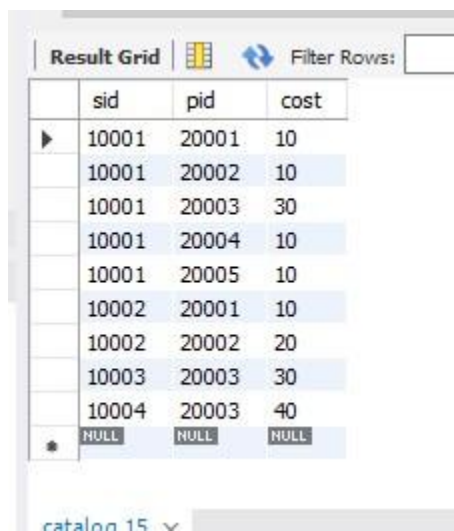
PARTS

pid	pname	color
20001	Book	Red
20002	Pen	Red
20003	Pencil	Green
20004	Mobile	Green
20005	Charger	Black
NULL	NULL	NULL

parts 14 ×

Output

Select * from catalog;

CATALOG

sid	pid	cost
10001	20001	10
10001	20002	10
10001	20003	30
10001	20004	10
10001	20005	10
10002	20001	10
10002	20002	20
10003	20003	30
10004	20003	40
NULL	NULL	NULL

catalog 15 ×

i)Find the pnames of parts for which there is some supplier.

select distinct P.pname from Parts P,Catalog C where P.pid=C.pid and exists(select 'X' from Catalog where pid=P.pid);

pname
▶ Book
Pen
Pencil
Mobile
Charger

ii) Find the snames of suppliers who supply every part.

select S.sname, S.sid from suppliers S where S.sid IN (select C.sid from catalog C group by C.sid having count(distinct C.pid) = (select count(pid) from parts));

Result Grid		Filter Rows

sname
▶ Acme Widget

suppliers 3 ✕

iii)Find the snames of suppliers who supply every red part.

select S.sname from suppliers S where S.sid IN (select C.sid from catalog c where not exists (select P.pid from parts P where P.color="red"and(not exists(select C1.sid from catalog C1 where C1.sid=C.sid and C1.pid=P.pid))));
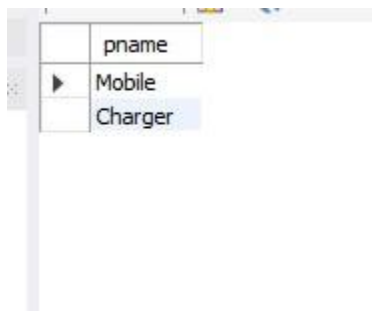
sname
▶ Acme Widget
Johns

suppliers 4 ✕

iv)Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.
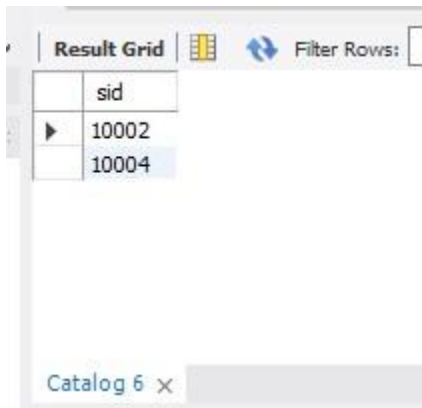
select P.pname from parts P,catalog C,suppliers S where P.pid=C.pid and C.sid=S.sid and S.sname="Acme Widget"and not exists(select * from catalog c1,Suppliers s1 where P.pid=C1.pid and C1.sid=S1.sid and S1.sname<>"Acme Widget");

pname
▶ Mobile
Charger

v) Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part)

select distinct C.sid from Catalog C where C.cost>(select avg(C1.cost) from catalog C1 where C1.pid=C.pid);
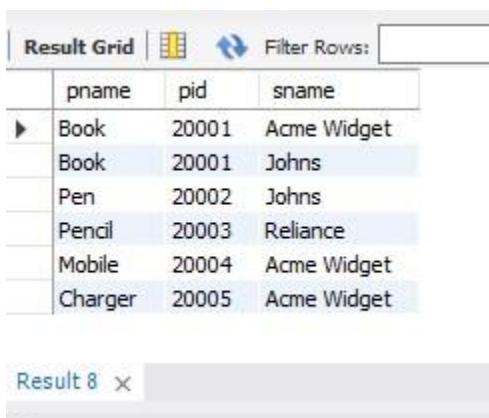
sid
10002
10004

Catalog 6 ×

vi) For each part, find the sname of the supplier who charges the most for that part.

Select P.pname, P.pid ,S.sname from parts p,Suppliers S,catalog C where C.pid=P.pid and C.sid=S.sid and C.cost = (select max(C1.cost) from catalog C1 where C1.pid=P.pid);

pname	pid	sname
Book	20001	Acme Widget
Book	20001	Johns
Pen	20002	Johns
Pencil	20003	Reliance
Mobile	20004	Acme Widget
Charger	20005	Acme Widget

Result 8 ×

PROGRAM 4: STUDENT FACULTY DATABASE

Consider the following database for
student enrollment for course :

STUDENT(snum: integer, sname:string, major: string, lvl: string, age: integer)

CLASS(cname: string, meetsat: time, room: string, fid: integer)

ENROLLED(snum: integer, cname:string)

FACULTY(fid: integer, fname:string, deptid: integer)

The meaning of these relations is straightforward; for example, Enrolled has one record
per student-class pair such that the student is enrolled in the class. Level(lvl) is a two
character
code with 4 different values (example: Junior: JR etc)

Write the following queries in SQL.
No duplicates should be printed in any of the answers.

Select * from student;

snum	sname	major	lvl	age
▶ 1	jhon	CS	Sr	19
2	Smith	CS	Jr	20
3	Jacob	CV	Sr	20
4	Tom	CS	Jr	20
5	Rahul	CS	Jr	20
6	Rita	CS	Sr	21
* NULL	NULL	NULL	NULL	NULL

student 13 ×

select * from faculty;

fid	fname	deptid
▶ 11	Harish	1000
12	MV	1000
13	Mira	1001
14	Shiva	1002
15	Nupur	1000
* NULL	NULL	NULL

faculty 14 ×

select * from class;

cname	meetsat	room	fid
Class1	2012-11-15 10:15:16	R1	14
Class10	2012-11-15 10:15:16	R128	14
Class2	2012-11-15 10:15:20	R2	12
Class3	2012-11-15 10:15:25	R3	11
Class4	2012-11-15 20:15:20	R4	14
Class5	2012-11-15 20:15:20	R3	15
Class6	2012-11-15 13:20:20	R2	14
Class7	2012-11-15 10:10:10	R3	14
NULL	NULL	NULL	NULL

select * from enrolled;

snum	cname
1	class1
2	class1
1	class10
3	class3
4	class3
5	class4
NULL	NULL

i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by the name "Shiva".

select s.sname from STUDENT s, CLASS c, ENROLLED e where s.snum=e.snum and c.cname = e.cname and c.fid =(select fid from FACULTY where fname = "Shiva") and s.lvl ="Jr";

sname
Smith
Rahul

ii. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.

SELECT C.cname FROM Class C
WHERE C.room = "R128"
or C.cname IN (SELECT E.cname FROM Enrolled E GROUP BY E.cname HAVING count(E.snum) >= 5);

	cname
▶	Class 10
*	NULL

iii. Find the names of all students who are enrolled in two classes that meet at the same Time.

select distinct s.sname from student s where s.snum in
(select e1.snum from enrolled e1, enrolled e2, class c1, class c2
where e1.snum = e2.snum and e1.cname != e2.cname and e1.cname = c1.cname
and e2.cname = c2.cname and c1.meetsat = c2.meetsat);

	sname
▶	jhon

iv. Find the names of faculty members who teach in every room in which some class is taught.

select f.fname,c.fid from faculty f,class c where f.fid=c.fid group by c.fid having count(c.fid)=(select count(distinct room)from class);

Result Grid		
	fname	fid
▶	Shiva	14

v. Find the names of faculty members for whom the combined enrollment of the courses that they teach less than five.

select distinct f.fname from faculty f where 5>(select COUNT(e.snum) from Class c, enrolled e where c.cname = e.cname and c.fid = f.fid);

fname
▸ Harish
MV
Mira
Shiva
Nupur

vi. Find the names of students who are not enrolled in any class.

select distinct s.sname from student s
where s.snum not in(select e.snum from enrolled e);

sname
▸ Rita

vii. For each age value that appears in Students, find the level value that appears most
select s.age ,s.lvl from student s group by s.age having s.lvl in (select s1.lvl from student s1 where s1.age = s.age group by s1.age having count(*)>= all(select s2.lvl from student s2 where s2.age = s1.age group by s2.age));

age	lvl
▸ 19	Sr
20	Jr
21	Sr

PROGRAM 5: AIRLINE FLIGHT DATABASE

Consider the following database that keeps track of airline flight information:

FLIGHTS(flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price: integer)

AIRCRAFT(aid: integer, aname: string, cruisingrange: integer)

CERTIFIED(eid: integer, aid: integer)

EMPLOYEES(eid: integer, ename: string, salary: integer)

Note that the Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly.

Write each of the following queries in SQL.

```
create database airlinedb;
use airlinedb;

create table FLIGHTS(
flno int,
fromplace varchar(20),
toplace  varchar(20),
distance integer,
departs datetime,
arrives datetime,
price int,
primary key(flno)
);
create table AIRCRAFT(
aid int,
aname varchar(20),
cruisingrange int,
primary key (aid)
);
create table EMPLOYEES(
eid int,
ename varchar(20),
salary int,
primary key (eid)
);

create table certified(
eid int,
```

```
aid int,
foreign key (eid) references employees(eid),
foreign key (aid) references aircraft(aid)
);


insert into aircraft values
("101","747","3000"),("102","Boeing","900"),("103","647","800"),("104","Dreamliner","10000"),("1
05","Boeing","3500"),("106","707","1500"),("107","Dream","120000");
select * from aircraft;

insert into flights values("101","Banglore","Delhi","2500","05/05/13 07.15.31","05/05/13
07.15.31","5000"),("102","Banglore","Lucknow","3000","05/05/13 07.15.31","05/05/13
11.15.31","6000"),("103","Lucknow","Delhi","500","05/05/13 12.15.31","05/05/13
17.15.31","3000"),
("107","Banglore","Frankfrut","8000","05/05/13 07.15.31","05/05/13 22.15.31","60000"),
("104","Banglore","Frankfrut","8500","05/05/13 07.15.31","05/05/13 23.15.31","75000"),
("105","Kolkata","Delhi","3400","05/05/13 07.15.31","05/05/13 09.15.31","7000");
insert into flights values("106","Delhi","kolkata","4000","05/05/13 09.15.31","06/05/13
06.00.00","2000");
select * from flights;

insert into employees values
("701","A","50000"),("702","B","100000"),("703","C","150000"),("704","D","90000"),("705","E","40
000"),("706","F","60000"),("707","G","90000");
select * from employees;

insert into certified
values("701","101"),("701","102"),("701","106"),("701","105"),("702","104"),("703","104"),("704","
104"),("702","107"),("703","107"),("704","107"),("702","101"),("703","105"),("704","105"),("705","1
03");
```

select * from aircraft;

aid	aname	cruisingrange
101	747	3000
102	Boeing	900
103	647	800
104	Dreamliner	10000
105	Boeing	3500
106	707	1500
107	Dream	120000
NULL	NULL	NULL

aircraft 6 ×

select * from flights;

flno	fromplace	toplace	distance	departs	arrives	price
101	Banglore	Delhi	2500	2005-05-13 07:15:31	2005-05-13 07:15:31	5000
102	Banglore	Lucknow	3000	2005-05-13 07:15:31	2005-05-13 11:15:31	6000
103	Lucknow	Delhi	500	2005-05-13 12:15:31	2005-05-13 17:15:31	3000
104	Banglore	Frankfrut	8500	2005-05-13 07:15:31	2005-05-13 23:15:31	75000
105	Kolkata	Delhi	3400	2005-05-13 07:15:31	2005-05-13 09:15:31	7000
106	Delhi	kolkata	4000	2005-05-13 09:15:31	2006-05-13 06:00:00	2000
107	Banglore	Frankfrut	8000	2005-05-13 07:15:31	2005-05-13 22:15:31	60000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

flights 7 ×

select * from employees;

	eid	ename	salary
▶	701	A	50000
	702	B	100000
	703	C	150000
	704	D	90000
	705	E	40000
	706	F	60000
	707	G	90000
*	NULL	NULL	NULL
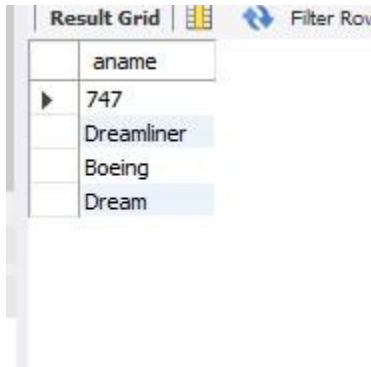
select * from certified;

Result Grid | Filter Rows:

	eid	aid
▶	701	101
	701	102
	701	106
	701	105
	702	104
	703	104
	704	104
	702	107
	703	107
	704	107
	702	101
	703	105
	704	105
	705	103

certified 9 ✕

Output

Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000

select distinct A.aname from aircraft A where A.aid in (select C.aid from certified C,Employees E where C.eid=E.eid and not exists (select * from employees E1 where E1.eid=E.eid and E1.salary<80000));
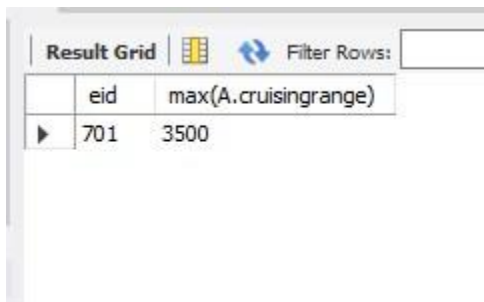
aname
747
Dreamliner
Boeing
Dream

For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.

select C.eid ,max(A.cruisingrange) from certified C,aircraft A where C.aid=A.aid group by C.eid having count(*)>3;
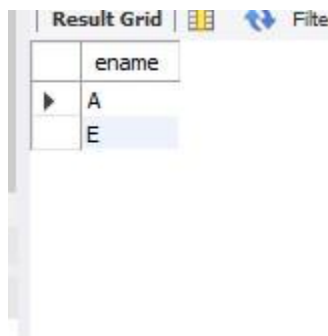
eid	max(A.cruisingrange)
701	3500

Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.
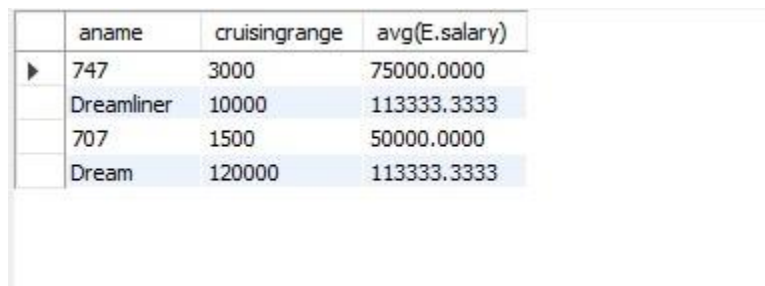
select distinct E.ename from employees E where E.salary <(select min(F.price) from flights F where F.fromplace="Banglore" and F.toplace="Frankfrut");

ename
A
E

For all aircraft with cruisingrange over 1000 Kms, find the name of the aircraft and the average salary of all pilots certified for this aircraft.

select A.aname,A.cruisingrange,avg(E.salary) from aircraft A,employees E,Certified C where C.eid=E.eid and C.aid=A.aid group by A.aname having A.cruisingrange>1000;

aname	cruisingrange	avg(E.salary)
747	3000	75000.0000
Dreamliner	10000	113333.3333
707	1500	50000.0000
Dream	120000	113333.3333

Find the names of pilots certified for some Boeing aircraft.

select distinct E.ename from employees E,certified C,aircraft A where E.eid = C.eid and C.aid =A .aid and A.aname LIKE "Boeing";
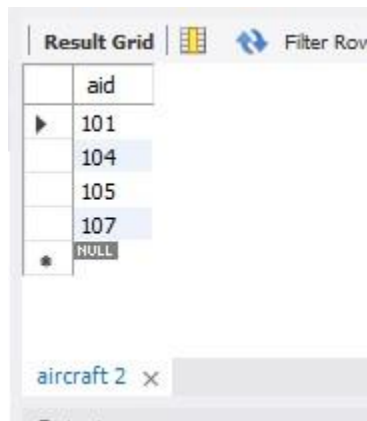
ename
A
C
D

Result 1 ✕

Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi

select A.aid from aircraft A where A.cruisingrange > (select min(F.distance) from flights F where F.fromplace = "Banglore" and F.toplace = "Delhi");
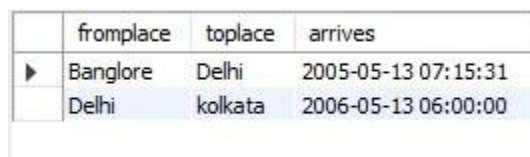
aid
101
104
105
107
NULL

aircraft 2 ×

A customer wants to travel from Bangalore to Kolkata New with no more than two changes of flight. List the choice of departure times from Madison if the customer wants to arrive in Kolkata by 6 p.m.

select f.fromplace,f.toplace,f.arrives from flights f where(f.fromplace="Banglore" and f.toplace=(select fromplace from flights where toplace="kolkata"))or f.toplace="kolkata";

fromplace	toplace	arrives
Banglore	Delhi	2005-05-13 07:15:31
Delhi	kolkata	2006-05-13 06:00:00

PROGRAM 6 : ORDER DATABASE

Consider the following schema for Order Database:

SALESMAN (*Salesman_id, Name, City, Commission*)
CUSTOMER (*Customer_id, Cust_Name, City, Grade, Salesman_id*)
ORDERS (*Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id*)

1. Count the customers with grades above Bangalore's average.

2. Find the name and numbers of all salesmen who had more than one customer.

3. List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)

4. Create a view that finds the salesman who has the customer with the highest order of a day.

5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

```
create database orderdb1;
use orderdb1;

create table salesman(
salesman_id varchar(20),
salesman_name varchar(20),
salesman_city varchar(20),
commission  varchar(20),
primary key(salesman_id)
);

create table customer(
customer_id varchar(20),
customer_name varchar(20),
customer_city varchar(20),
grade varchar(20),
salesman_id varchar(20),
primary key(customer_id),
foreign key(salesman_id) references salesman(salesman_id) on delete set null);

create table orders(
ord_no int,
purchase_amt double,
ord_date date,
```

```sql
customer_id varchar(20),
salesman_id varchar(20),
foreign key(salesman_id) references salesman(salesman_id) on delete cascade,
foreign key(customer_id) references customer(customer_id) on delete cascade
);

insert into salesman values("1000","JHON","BANGLORE","25%"),
("2000","RAVI","BANGLORE","20%"),
("3000","KUMAR","MYSORE","15%"),
("4000","SMITH","DELHI","30%"),
("5000","HARSHA","HYDRABAD","15%");
select * from salesman;

insert into customer values("10","PREETHI","BANGLORE","100","1000"),
("11","VIVEK","MANGLORE","300","1000"),
("12","BHASKAR","CHENNAI","400","2000"),
("13","CHETHAN","BANGLORE","200","2000"),
("14","MAMTHA","BANGLORE","400","3000");

select * from customer;

insert into orders values("50","5000","17-05-04","10","1000"),
("51","450","17-01-20","10","2000"),
("52","1000","17-02-24","13","2000"),
("53","3500","17-04-13","14","3000"),
("54","550","17-03-09","12","2000");
select * from orders;
insert into salesman values("1000","JHON","BANGLORE","25%"),
("2000","RAVI","BANGLORE","20%"),
("3000","KUMAR","MYSORE","15%"),
("4000","SMITH","DELHI","30%"),
("5000","HARSHA","HYDRABAD","15%");
select * from salesman;

insert into customer values("10","PREETHI","BANGLORE","100","1000"),
("11","VIVEK","MANGLORE","300","1000"),
("12","BHASKAR","CHENNAI","400","2000"),
("13","CHETHAN","BANGLORE","200","2000"),
("14","MAMTHA","BANGLORE","400","3000");

select * from customer;

insert into orders values("50","5000","17-05-04","10","1000"),
("51","450","17-01-20","10","2000"),
```

("52","1000","17-02-24","13","2000"),
("53","3500","17-04-13","14","3000"),
("54","550","17-03-09","12","2000");
select * from orders;

select * from salesman;

salesman_id	salesman_name	salesman_city	commission
1000	JHON	BANGLORE	25%
2000	RAVI	BANGLORE	20%
3000	KUMAR	MYSORE	15%
4000	SMITH	DELHI	30%
5000	HARSHA	HYDRABAD	15%
NULL	NULL	NULL	NULL

salesman 1 ×

select * from customer;

customer_id	customer_name	customer_city	grade	salesman_id
10	PREETHI	BANGLORE	100	1000
11	VIVEK	MANGLORE	300	1000
12	BHASKAR	CHENNAI	400	2000
13	CHETHAN	BANGLORE	200	2000
14	MAMTHA	BANGLORE	400	3000
NULL	NULL	NULL	NULL	NULL

customer 2 ×

select * from orders;

ord_no	purchase_amt	ord_date	customer_id	salesman_id
50	5000	2004-05-17	10	1000
51	450	2020-01-17	10	2000
52	1000	2024-02-17	13	2000
53	3500	2013-04-17	14	3000
54	550	2009-03-17	12	2000

orders 3 ×

1. Count the customers with grades above Bangalore's average.
```
select grade,count(distinct customer_id)
from customer group by grade
having grade > (select    avg(grade)
from customer where customer_city ="BANGLORE");
```

grade	count(distinct customer_id)
300	1
400	2

2. Find the name and numbers of all salesmen who had more than one customer.
```
select salesman_id ,salesman_name
from salesman S
where 1 <(select count(*)
from customer where salesman_id = S.salesman_id);
```

salesman_id	salesman_name
1000	JHON
2000	RAVI
NULL	NULL

3. List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)
```
select salesman.salesman_id ,salesman_name,customer_name,commission
from salesman,customer
where salesman_city = customer_city
union
select salesman_id,salesman_name ,'NO MATCH FOUND',commission from salesman
where not salesman_city = any(select customer_city from customer)order by 2 desc;
```

salesman_id	salesman_name	customer_name	commission
4000	SMITH	NO MATCH FOUND	30%
2000	RAVI	PREETHI	20%
2000	RAVI	CHETHAN	20%
2000	RAVI	MAMTHA	20%
3000	KUMAR	NO MATCH FOUND	15%
1000	JHON	PREETHI	25%
1000	JHON	CHETHAN	25%
1000	JHON	MAMTHA	25%
5000	HARSHA	NO MATCH FOUND	15%

4. Create a view that finds the salesman who has the customer with the highest order of a day.

create view best_salesman as select b.ord_date ,a.salesman_id,a.salesman_name
from salesman a,orders b where a.salesman_id=b.salesman_id
and b.purchase_amt=(select max(purchase_amt) from orders c
where c.ord_date=b.ord_date);

select * from best_salesman;

ord_date	salesman_id	salesman_name
2004-05-17	1000	JHON
2020-01-17	2000	RAVI
2024-02-17	2000	RAVI
2013-04-17	3000	KUMAR
2009-03-17	2000	RAVI

best_salesman 12 ×

LAB PROGRAM 7 : BOOK DATABASE

BOOK (Book_id, Title, Publisher_Name, Pub_Year)
BOOK_AUTHORS (Book_id, Author_Name)
PUBLISHER (Name, Address, Phone)
BOOK_COPIES (Book_id, Branch_id, No-of_Copies)
BOOK_LENDING (Book_id, Branch_id, Card_No, Date_Out, Due_Date)
LIBRARY_BRANCH (Branch_id, Branch_Name, Address)

```
create database bookdb1;
use bookdb1;

CREATE TABLE PUBLISHER
(NAME VARCHAR(20) PRIMARY KEY,
PHONE VARCHAR(10),
ADDRESS VARCHAR(20));

CREATE TABLE BOOK(
BOOK_ID INTEGER PRIMARY KEY,
TITLE VARCHAR(20),
PUB_YEAR VARCHAR(20),
PUBLISHER_NAME VARCHAR(20),
FOREIGN KEY (PUBLISHER_NAME) REFERENCES PUBLISHER (NAME) ON DELETE
CASCADE);

CREATE TABLE BOOK_AUTHORS(
AUTHOR_NAME VARCHAR(20),
BOOK_ID INTEGER,
PRIMARY KEY (BOOK_ID, AUTHOR_NAME),
FOREIGN KEY (BOOK_ID) REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE );

CREATE TABLE LIBRARY_BRANCH (
BRANCH_ID INTEGER PRIMARY KEY,
BRANCH_NAME VARCHAR(20),
ADDRESS VARCHAR(20));


CREATE TABLE BOOK_COPIES
(
NO_OF_COPIES INTEGER,
BOOK_ID INTEGER,
BRANCH_ID INTEGER,
PRIMARY KEY (BOOK_ID, BRANCH_ID),
FOREIGN KEY (BOOK_ID) REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE,
FOREIGN KEY (BRANCH_ID) REFERENCES LIBRARY_BRANCH (BRANCH_ID) ON
DELETE CASCADE);

CREATE TABLE CARD
(CARD_NO INTEGER PRIMARY KEY);
```

```sql
CREATE TABLE BOOK_LENDING(
DATE_OUT DATE,
DUE_DATE DATE,
BOOK_ID INTEGER,
BRANCH_ID INTEGER,
CARD_NO INTEGER,
PRIMARY KEY (BOOK_ID, BRANCH_ID, CARD_NO),
FOREIGN KEY (BOOK_ID) REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE,
FOREIGN KEY (CARD_NO) REFERENCES CARD (CARD_NO) ON DELETE CASCADE,
FOREIGN KEY (BRANCH_ID) REFERENCES LIBRARY_BRANCH (BRANCH_ID) ON
DELETE CASCADE);

INSERT INTO PUBLISHER VALUES ("MCGRAW-HILL", "9989076587", "BANGALORE"),
 ("PEARSON", "9889076565", "NEWDELHI"),
 ("RANDOM HOUSE", "7455679345", "HYDRABAD"),
 ("HACHETTE LIVRE", "8970862340", "CHENNAI"),
("GRUPO PLANETA","7756120238","BANGALORE");


INSERT INTO BOOK VALUES ("1","DBMS","JAN-2017", "MCGRAW-HILL"),
 ("2","ADBMS","JUN-2016", "MCGRAW-HILL"),
 ("3","CN","SEP-2016", "PEARSON"),
 ("4","CG","SEP-2015","GRUPO PLANETA"),
 ("5","OS","MAY-2016","PEARSON");

 INSERT INTO BOOK_AUTHORS VALUES ("NAVATHE","1"),
 ("NAVATHE","2"),
 ("TANENBAUM","3"),
 ("EDWARD ANGEL","4"),
 ("GALVIN","5");

 INSERT INTO LIBRARY_BRANCH VALUES ("10","RR NAGAR","BANGALORE"),
("11","RNSIT","BANGALORE"),
 ("12","RAJAJI NAGAR", "BANGALORE"),
 ("13","NITTE","MANGALORE"),
 ("14","MANIPAL","UDUPI");

INSERT INTO BOOK_COPIES
VALUES("10","1","10"),("5","1","11"),("2","2","12"),("5","2","13"),("7","3","14"),("1","5","10"),("3","4
","11");

INSERT INTO CARD VALUES ("100"),("101"),("102"),("103"),("104");

INSERT INTO BOOK_LENDING VALUES ("17-01-01","17-06-01","1","10","101"),
 ("17-01-17","17-03-17","3", "14", "101"),
 ("17-02-21","17-04-21", 2, 13, 101),
 ("17-03-15","17-07-15", "4", "11", "101"),
 ("17-04-12","17-05-12", "1", "11", "104");
```

SELECT * FROM PUBLISHER;

	NAME	PHONE	ADDRESS
▶	GRUPO PLANETA	7756120238	BANGALORE
	HACHETTE LIVRE	8970862340	CHENNAI
	MCGRAW-HILL	9989076587	BANGALORE
	PEARSON	9889076565	NEWDELHI
	RANDOM HOUSE	7455679345	HYDRABAD
*	NULL	NULL	NULL

PUBLISHER 3 ✕

SELECT * FROM BOOK;

	BOOK_ID	TITLE	PUB_YEAR	PUBLISHER_NAME
▶	1	DBMS	JAN-2017	MCGRAW-HILL
	2	ADBMS	JUN-2016	MCGRAW-HILL
	3	CN	SEP-2016	PEARSON
	4	CG	SEP-2015	GRUPO PLANETA
	5	OS	MAY-2016	PEARSON
*	NULL	NULL	NULL	NULL

BOOK 4 ✕

SELECT * FROM BOOK_AUTHORS;

	AUTHOR_NAME	BOOK_ID
▶	NAVATHE	1
	NAVATHE	2
	TANENBAUM	3
	EDWARD ANGEL	4
	GALVIN	5
*	NULL	NULL

BOOK_AUTHORS 5 ✕

SELECT * FROM LIBRARY_BRANCH;

	BRANCH_ID	BRANCH_NAME	ADDRESS
▶	10	RR NAGAR	BANGALORE
	11	RNSIT	BANGALORE
	12	RAJAJI NAGAR	BANGALORE
	13	NITTE	MANGALORE
	14	MANIPAL	UDUPI
*	NULL	NULL	NULL

LIBRARY_BRANCH 6 ✕

SELECT * FROM BOOK_COPIES;

NO_OF_COPIES	BOOK_ID	BRANCH_ID
10	1	10
5	1	11
2	2	12
5	2	13
7	3	14
3	4	11
1	5	10
NULL	NULL	NULL

BOOK_COPIES 7 ✕

SELECT * FROM CARD;

CARD_NO
100
101
102
103
104
NULL

CARD 8 ✕

SELECT * FROM BOOK_LENDING;

DATE_OUT	DUE_DATE	BOOK_ID	BRANCH_ID	CARD_NO
2017-01-01	2017-06-01	1	10	101
2017-04-12	2017-05-12	1	11	104
2017-02-21	2017-04-21	2	13	101
2017-01-17	2017-03-17	3	14	101
2017-03-15	2017-07-15	4	11	101
NULL	NULL	NULL	NULL	NULL

BOOK_LENDING 9 ✕

I. Retrieve details of all books in the library – id, title, name of publisher, authors, number of   copies in each branch, etc.
SELECT B.BOOK_ID, B.TITLE, B.PUBLISHER_NAME,
A.AUTHOR_NAME,C.NO_OF_COPIES,L.BRANCH_ID
FROM BOOK B, BOOK_AUTHORS A, BOOK_COPIES C, LIBRARY_BRANCH L
WHERE B.BOOK_ID=A.BOOK_ID AND B.BOOK_ID=C.BOOK_ID AND
L.BRANCH_ID=C.BRANCH_ID;

BOOK_ID	TITLE	PUBLISHER_NAME	AUTHOR_NAME	NO_OF_COPIES	BRANCH_ID
1	DBMS	MCGRAW-HILL	NAVATHE	10	10
1	DBMS	MCGRAW-HILL	NAVATHE	5	11
2	ADBMS	MCGRAW-HILL	NAVATHE	2	12
2	ADBMS	MCGRAW-HILL	NAVATHE	5	13
3	CN	PEARSON	TANENBAUM	7	14
4	CG	GRUPO PLANETA	EDWARD ANGEL	3	11
5	OS	PEARSON	GALVIN	1	10

II Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017
SELECT CARD_NO FROM BOOK_LENDING
WHERE DATE_OUT BETWEEN "2017-01-01" AND "2017-07-01"
GROUP BY CARD_NO
HAVING COUNT(*)>3;

CARD_NO
101

III Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.
DELETE FROM BOOK
WHERE BOOK_ID=3;

SELECT * FROM BOOK;

BOOK_ID	TITLE	PUB_YEAR	PUBLISHER_NAME
1	DBMS	JAN-2017	MCGRAW-HILL
2	ADBMS	JUN-2016	MCGRAW-HILL
4	CG	SEP-2015	GRUPO PLANETA
5	OS	MAY-2016	PEARSON
NULL	NULL	NULL	NULL

SELECT * FROM BOOK_AUTHORS;

AUTHOR_NAME	BOOK_ID
NAVATHE	1
NAVATHE	2
EDWARD ANGEL	4
GALVIN	5
NULL	NULL

SELECT * FROM BOOK_COPIES;

	NO_OF_COPIES	BOOK_ID	BRANCH_ID
▶	10	1	10
	5	1	11
	2	2	12
	5	2	13
	3	4	11
	1	5	10
＊	NULL	NULL	NULL

SELECT * FROM BOOK_LENDING;

	DATE_OUT	DUE_DATE	BOOK_ID	BRANCH_ID	CARD_NO
▶	2017-01-01	2017-06-01	1	10	101
	2017-04-12	2017-05-12	1	11	104
	2017-02-21	2017-04-21	2	13	101
	2017-01-17	2017-03-17	3	14	101
	2017-03-15	2017-07-15	4	11	101
＊	NULL	NULL	NULL	NULL	NULL

BOOK_LENDING 9 ×

IV Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.
CREATE VIEW YEAR_OF_PUBLICATION AS SELECT PUB_YEAR
FROM BOOK;

SELECT * FROM YEAR_OF_PUBLICATION;

	PUB_YEAR
▶	JAN-2017
	JUN-2016
	SEP-2016
	SEP-2015
	MAY-2016

YEAR_OF_PUBLICATION 2 ×

V Create a view of all books and its number of copies that are currently available in the Library.
CREATE VIEW BOOKS_AVAILABLE_IN_LIBRARY AS
SELECT B.BOOK_ID, B.TITLE, C.NO_OF_COPIES
FROM BOOK B, BOOK_COPIES C, LIBRARY_BRANCH L
WHERE B.BOOK_ID=C.BOOK_ID
AND C.BRANCH_ID=L.BRANCH_ID;

SELECT * FROM BOOKS_AVAILABLE_IN_LIBRARY;

BOOK_ID	TITLE	NO_OF_COPIES
1	DBMS	10
1	DBMS	5
2	ADBMS	2
2	ADBMS	5
3	CN	7
4	CG	3
5	OS	1

BOOKS_AVAILABLE_IN_LIBRAR... ✕

LAB PROGRAM 8 : STUDENT ENROLLMENT

Consider the following database of student enrollment in courses & books adopted for each course.
STUDENT (regno: string, name: string, major: string, bdate:date)
COURSE (course #:int, cname:string, dept:string)
ENROLL (regno:string, course#:int, sem:int, marks:int)
BOOK _ ADOPTION (course# :int, sem:int, book-ISBN:int)
TEXT (book-ISBN:int, book-title:string, publisher:string, author:string)

Database applications laboratory GCEM DEPARTMENT OF CSE Page - 5 - 5th semester
 i. Create the above tables by properly specifying the primary keys and the foreign keys.
ii. Enter at least five tuples for each relation.


```
create database studentenrollment;
use studentenrollment;

CREATE TABLE STUDENT(
reg_no VARCHAR(20),
name VARCHAR(20),
major VARCHAR(20),
bdate DATE,
PRIMARY KEY(reg_no));

CREATE TABLE COURSE(
course_no INT,
cname VARCHAR(20),
dept VARCHAR(20),
PRIMARY KEY (course_no));

CREATE TABLE ENROLL(
reg_no VARCHAR(15),
course_no INT,
sem INT,
marks INT,
PRIMARY KEY (reg_no,course_no),
FOREIGN KEY (reg_no) REFERENCES student (reg_no),
FOREIGN KEY (course_no) REFERENCES course (course_no));

CREATE TABLE TEXT(
book_isbn INT,
book_title VARCHAR(20),
publisher VARCHAR(20),
author VARCHAR(20),
PRIMARY KEY (book_isbn));
```

```sql
CREATE TABLE BOOK_ADOPTION(
course_no INT,
sem INT,
book_isbn INT,
PRIMARY KEY (course_no,book_isbn),
FOREIGN KEY (course_no) REFERENCES course (course_no),
FOREIGN KEY (book_isbn) REFERENCES text(book_isbn));

INSERT INTO STUDENT (reg_no,name,major,bdate) VALUES ('1pe11cs001','a','sr',19931230),
("1pe11cs002","b","sr","19930924"),
("1pe11cs003","c","sr","19931127"),
("1pe11cs004","d","sr","19930413"),
("1pe11cs005","e","jr","19940824");

INSERT INTO COURSE(course_no,cname,dept) VALUES
("111","OS","CSE"),
("112","EC","CSE"),
("113","SS","ISE"),
("114","DBMS","CSE"),
("115","SIGNALS","ECE");

INSERT INTO TEXT(book_isbn,book_title,publisher,author) VALUES
("10","DATABASE SYSTEMS","PEARSON","SCHIELD"),
("900","OPERATING SYS","PEARSON","LELAND"),
("901","CIRCUITS","HALL INDIA","BOB"),
("902","SYSTEM SOFTWARE","PETERSON","JACOB"),
("903","SCHEDULING","PEARSON","PATIL"),
("904","DATABASE SYSTEMS","PEARSON","JACOB"),
("905","DATABASE MANAGER","PEARSON","BOB"),
("906","SIGNALS","HALL INDIA","SUMIT");

INSERT INTO BOOK_ADOPTION(course_no,sem,book_isbn) VALUES
("111","5","900"),
("111","5","903"),
("111","5","904"),
("112","3","901"),
("113","3","10"),
("114","5","905"),
("113","5","902"),
("115","3","906");

INSERT INTO  ENROLL(reg_no,course_no,sem,marks) VALUES
("1pe11cs001","115","3","100"),
("1pe11cs002","114","5","100"),
("1pe11cs003","113","5","100"),
("1pe11cs004","111","5","100"),
("1pe11cs005","112","3","100");
```

SELECT * FROM STUDENT;

reg_no	name	major	bdate
1pe11cs001	a	sr	1993-12-30
1pe11cs002	b	sr	1993-09-24
1pe11cs003	c	sr	1993-11-27
1pe11cs004	d	sr	1993-04-13
1pe11cs005	e	jr	1994-08-24
NULL	NULL	NULL	NULL

STUDENT 6 ✕

SELECT * FROM COURSE;

course_no	cname	dept
111	OS	CSE
112	EC	CSE
113	SS	ISE
114	DBMS	CSE
115	SIGNALS	ECE
NULL	NULL	NULL

COURSE 7 ✕

SELECT * FROM TEXT;

book_isbn	book_title	publisher	author
10	DATABASE SYSTEMS	PEARSON	SCHIELD
900	OPERATING SYS	PEARSON	LELAND
901	CIRCUITS	HALL INDIA	BOB
902	SYSTEM SOFTWARE	PETERSON	JACOB
903	SCHEDULING	PEARSON	PATIL
904	DATABASE SYSTEMS	PEARSON	JACOB
905	DATABASE MANAGER	PEARSON	BOB
906	SIGNALS	HALL INDIA	SUMIT
NULL	NULL	NULL	NULL

TEXT 8 ✕

SELECT * FROM ENROLL;

reg_no	course_no	sem	marks
1pe11cs001	115	3	100
1pe11cs002	114	5	100
1pe11cs003	113	5	100
1pe11cs004	111	5	100
1pe11cs005	112	3	100
NULL	NULL	NULL	NULL

ENROLL 9 ×

SELECT * FROM BOOK_ADOPTION;

course_no	sem	book_isbn
111	5	900
111	5	903
111	5	904
112	3	901
113	3	10
113	5	902
114	5	905
115	3	906
NULL	NULL	NULL

BOOK_ADOPTION 10 ×

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

INSERT INTO TEXT(book_isbn,book_title,publisher,author) VALUES("907","COMPUTER NETWORKS","PEARSON","FORTAN");
INSERT INTO BOOK_ADOPTION(course_no,sem,book_isbn) VALUES("111","3","907");

SELECT * FROM TEXT;

	book_isbn	book_title	publisher	author
▶	10	DATABASE SYSTEMS	PEARSON	SCHIELD
	900	OPERATING SYS	PEARSON	LELAND
	901	CIRCUITS	HALL INDIA	BOB
	902	SYSTEM SOFTWARE	PETERSON	JACOB
	903	SCHEDULING	PEARSON	PATIL
	904	DATABASE SYSTEMS	PEARSON	JACOB
	905	DATABASE MANAGER	PEARSON	BOB
	906	SIGNALS	HALL INDIA	SUMIT
	907	COMPUTER NETWORKS	PEARSON	FORTAN
*	NULL	NULL	NULL	NULL

TEXT 1 ✕

SELECT * FROM BOOK_ADOPTION;

	course_no	sem	book_isbn
▶	111	5	900
	111	5	903
	111	5	904
	111	3	907
	112	3	901
	113	3	10
	113	5	902
	114	5	905
	115	3	906
*	NULL	NULL	NULL

BOOK_ADOPTION 2 ✕

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.

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

course_no	book_isbn	book_title
111	907	COMPUTER NETWORKS
111	904	DATABASE SYSTEMS
111	900	OPERATING SYS
111	903	SCHEDULING

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

SELECT c.dept
FROM course c, book_adoption ba
WHERE c.course_no=ba.course_no
GROUP BY c.dept
HAVING count(ba.book_isbn)=(SELECT count(ba2.book_isbn)
FROM TEXT t,book_adoption ba2,course c2
WHERE t.book_isbn=ba2.book_isbn AND c2.course_no=ba2.course_no AND
t.publisher='HALL INDIA' AND c2.dept=c.dept);

dept
ECE

PROGRAM 9 : MOVIE DATABASE

Consider the schema for Movie Database:

ACTOR (*Act_id, Act_Name, Act_Gender*)
DIRECTOR (*Dir_id, Dir_Name, Dir_Phone*)
MOVIES (*Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id*)
MOVIE_CAST (*Act_id, Mov_id, Role*)
RATING (*Mov_id, Rev_Stars*)

```
CREATE DATABASE MOVIEDB;
USE MOVIEDB;
CREATE TABLE ACTOR (
ACT_ID INT,
ACT_NAME VARCHAR (20),
ACT_GENDER CHAR (1),
PRIMARY KEY (ACT_ID));

CREATE TABLE DIRECTOR (
DIR_ID INT,
DIR_NAME VARCHAR (20),
DIR_PHONE LONG,
PRIMARY KEY (DIR_ID));

CREATE TABLE MOVIES (
MOV_ID INT,
MOV_TITLE VARCHAR (25),
MOV_YEAR INT,
MOV_LANG VARCHAR (12),
DIR_ID INT,
PRIMARY KEY (MOV_ID),
FOREIGN KEY (DIR_ID) REFERENCES DIRECTOR (DIR_ID));

CREATE TABLE MOVIE_CAST (
ACT_ID INT,
MOV_ID INT,
ROLE VARCHAR(10),
PRIMARY KEY (ACT_ID, MOV_ID),
FOREIGN KEY(ACT_ID) REFERENCES ACTOR(ACT_ID) ON DELETE CASCADE,
FOREIGN KEY(MOV_ID) REFERENCES MOVIES(MOV_ID) ON DELETE CASCADE);

CREATE TABLE RATING (
MOV_ID INT,
REV_STARS VARCHAR (25),
PRIMARY KEY (MOV_ID),
```

FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID));

INSERT INTO ACTOR VALUES ("301","ANUSHKA","F"),
              ("302","PRABHAS","M"),
              ("303","PUNITH","M"),
              ("304","JERMY","M");

INSERT INTO DIRECTOR VALUES ("60","RAJAMOULI", "8751611001"),
              ("61","HITCHCOCK", "7766138911"),
              ("62","FARAN", "9986776531"),
              ("63","STEVEN SPIELBERG", "8989776530");

INSERT INTO MOVIES VALUES ("1001","BAHUBALI-2", "2017","TELAGU", "60"),
              ("1002","BAHUBALI-1", "2015", "TELAGU", "60"),
              ("1003","AKASH", "2008", "KANNADA", "61"),
              ("1004","WAR HORSE", "2011", "ENGLISH", "63");

INSERT INTO MOVIE_CAST VALUES ("301","1002", "HEROINE"),
              ("301","1001", "HEROINE"),
              ("303", "1003", "HERO"),
              ("303", "1002", "GUEST"),
              ("304", "1004", "HERO");

INSERT INTO RATING VALUES ("1001","4"),
              ("1002", "2"),
              ("1003","5"),
              ("1004", "4");

SELECT * FROM ACTOR;

	ACT_ID	ACT_NAME	ACT_GENDER
▶	301	ANUSHKA	F
	302	PRABHAS	M
	303	PUNITH	M
	304	JERMY	M
*	NULL	NULL	NULL

ACTOR 2 ✕

SELECT * FROM DIRECTOR;

DIR_ID	DIR_NAME	DIR_PHONE
60	RAJAMOULI	8751611001
61	HITCHCOCK	7766138911
62	FARAN	9986776531
63	STEVEN SPIELBERG	8989776530
NULL	NULL	NULL

DIRECTOR 3 ×

SELECT * FROM MOVIES;

MOV_ID	MOV_TITLE	MOV_YEAR	MOV_LANG	DIR_ID
1001	BAHUBALI-2	2017	TELAGU	60
1002	BAHUBALI-1	2015	TELAGU	60
1003	AKASH	2008	KANNADA	61
1004	WAR HORSE	2011	ENGLISH	63
NULL	NULL	NULL	NULL	NULL

MOVIES 4 ×

SELECT * FROM MOVIE_CAST;

ACT_ID	MOV_ID	ROLE
301	1001	HEROINE
301	1002	HEROINE
303	1002	GUEST
303	1003	HERO
304	1004	HERO
NULL	NULL	NULL

MOVIE_CAST 2 ×

SELECT * FROM RATING;

MOV_ID	REV_STARS
1001	4
1002	2
1003	5
1004	4
NULL	NULL

RATING 6 ×

1. List the titles of all movies directed by 'Hitchcock'.

SELECT MOV_TITLE
FROM MOVIES
WHERE DIR_ID IN (SELECT DIR_ID
FROM DIRECTOR
WHERE DIR_NAME = 'HITCHCOCK');

MOV_TITLE
▶ AKASH

2. Find the movie names where one or more actors acted in two or more movies.
SELECT MOV_TITLE
FROM MOVIES M, MOVIE_CAST MV
WHERE M.MOV_ID=MV.MOV_ID AND ACT_ID IN (SELECT ACT_ID
FROM MOVIE_CAST GROUP BY ACT_ID
HAVING COUNT(ACT_ID)>1)
GROUP BY MOV_TITLE
HAVING COUNT(*)>1;

MOV_TITLE
▶ BAHUBALI-1

3. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).

SELECT ACT_NAME, MOV_TITLE, MOV_YEAR
FROM ACTOR A
JOIN MOVIE_CAST C
ON A.ACT_ID=C.ACT_ID
JOIN MOVIES M
ON C.MOV_ID=M.MOV_ID
WHERE M.MOV_YEAR NOT BETWEEN 2000 AND 2015;

ACT_NAME	MOV_TITLE	MOV_YEAR
▶ ANUSHKA	BAHUBALI-2	2017

4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.

SELECT MOV_TITLE,MAX(REV_STARS)
FROM MOVIES M,RATING R
WHERE M.MOV_ID=R.MOV_ID
GROUP BY MOV_TITLE
HAVING COUNT(*)>=1
ORDER BY MOV_TITLE;

MOV_TITLE	MAX(REV_STARS)
AKASH	5
BAHUBALI-1	2
BAHUBALI-2	4
WAR HORSE	4

5. Update rating of all movies directed by 'Steven Spielberg' to 5

UPDATE RATING
SET REV_STARS=5
WHERE MOV_ID IN(SELECT MOV_ID FROM MOVIES
WHERE DIR_ID IN(SELECT DIR_ID
FROM DIRECTOR
WHERE DIR_NAME = 'STEVEN SPIELBERG'));

SELECT * FROM RATING;

MOV_ID	REV_STARS
1001	4
1002	2
1003	5
1004	5
NULL	NULL

LAB PROGRAM 10 : COLLEGE DATABASE

Consider the schema for College Database:
STUDENT (*USN, SName, Address, Phone, Gender*)
SEMSEC (*SSID, Sem, Sec*)
CLASS (*USN, SSID*)
SUBJECT (*Subcode, Title, Sem, Credits*)
IAMARKS (*USN, Subcode, SSID, Test1, Test2, Test3, FinalIA*)

CODE
```
CREATE DATABASE COLLEGEDATABASEENTRY;
USE COLLEGEDATABASEENTRY;

CREATE TABLE STUDENT (
USN VARCHAR (10),
SNAME VARCHAR (25),
ADDRESS VARCHAR (25),
PHONE LONG,
GENDER CHAR (1),
PRIMARY KEY (USN));


CREATE TABLE SEMSEC (
SSID VARCHAR (5),
SEM INT,
SEC CHAR (1),
PRIMARY KEY (SSID));
select * from semsec;

CREATE TABLE CLASS (
USN VARCHAR (10),
SSID VARCHAR (5),
PRIMARY KEY (USN, SSID),
FOREIGN KEY (USN) REFERENCES STUDENT (USN),
FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID));
select * from class;

CREATE TABLE SUBJECT (
SUBCODE VARCHAR (8),
TITLE VARCHAR (20),
SEM INT,
CREDITS INT,
PRIMARY KEY (SUBCODE));
select * from subject;

CREATE TABLE IAMARKS (
USN VARCHAR (10),
SUBCODE VARCHAR (8),
SSID VARCHAR (5),
```

```sql
TEST1 INT,
TEST2 INT,
TEST3 INT,
FINALIA INT,
PRIMARY KEY (USN, SUBCODE, SSID),
FOREIGN KEY (USN) REFERENCES STUDENT (USN),
FOREIGN KEY (SUBCODE) REFERENCES SUBJECT (SUBCODE),
FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID));
select * from iamarks;

INSERT INTO STUDENT VALUES ('1RN13CS020','AKSHAY','BELAGAVI', 8877881122,'M'),
                ('1RN13CS062','SANDHYA','BENGALURU', 7722829912,'F'),
                ('1RN13CS091','TEESHA','BENGALURU', 7712312312,'F'),
                ('1RN13CS066','SUPRIYA','MANGALURU', 8877881122,'F'),
                ('1RN14CS010','ABHAY','BENGALURU', 9900211201,'M'),
                ('1RN14CS032','BHASKAR','BENGALURU', 9923211099,'M'),
                ('1RN14CS025','ASMI','BENGALURU', 7894737377,'F'),
                ('1RN15CS011','AJAY','TUMKUR', 9845091341,'M');


INSERT INTO STUDENT VALUES ('1RN15CS029','CHITRA','DAVANGERE', 7696772121,'F'),
                ('1RN15CS045','JEEVA','BELLARY', 9944850121,'M'),
                ('1RN15CS091','SANTOSH','MANGALURU', 8812332201,'M'),
                ('1RN16CS045','ISMAIL','KALBURGI', 9900232201,'M'),
                ('1RN16CS088','SAMEERA','SHIMOGA', 9905542212,'F'),
                ('1RN16CS122','VINAYAKA','CHIKAMAGALUR', 8800880011,'M');



INSERT INTO SEMSEC VALUES ('CSE8A', 8,'A'),
                ('CSE8B', 8,'B'),
                ('CSE8C', 8,'C'),
                ('CSE7A', 7,'A'),
                ('CSE7B', 7,'B'),
                ('CSE7C', 7,'C'),
                ('CSE6A', 6,'A'),
                ('CSE6B', 6,'B'),
                ('CSE6C', 6,'C'),
                ('CSE5A', 5,'A'),
                ('CSE5B', 5,'B'),
                ('CSE5C', 5,'C'),
                ('CSE4A', 4,'A'),
                ('CSE4B', 4,'B'),
                ('CSE4C', 4,'C'),
                ('CSE3A', 3,'A'),
                ('CSE3B', 3,'B'),
                ('CSE3C', 3,'C'),
                ('CSE2A', 2,'A'),
                ('CSE2B', 2,'B'),
                ('CSE2C', 2,'C'),
                ('CSE1A', 1,'A'),
```

```sql
                    ('CSE1B', 1,'B'),
                    ('CSE1C', 1,'C');

INSERT INTO CLASS VALUES ('1RN13CS020','CSE8A'),
 ('1RN13CS062','CSE8A'),
 ('1RN13CS066','CSE8B'),
 ('1RN13CS091','CSE8C'),
 ('1RN14CS010','CSE7A'),
 ('1RN14CS025','CSE7A'),
 ('1RN14CS032','CSE7A'),
 ('1RN15CS011','CSE4A'),
 ('1RN15CS029','CSE4A'),
 ('1RN15CS045','CSE4B'),
 ('1RN15CS091','CSE4C'),
 ('1RN16CS045','CSE3A'),
 ('1RN16CS088','CSE3B'),
 ('1RN16CS122','CSE3C');


INSERT INTO SUBJECT VALUES ('10CS81','ACA', 8, 4),
 ('10CS82','SSM', 8, 4),
 ('10CS83','NM', 8, 4),
 ('10CS84','CC', 8, 4),
 ('10CS85','PW', 8, 4),
 ('10CS71','OOAD', 7, 4),
 ('10CS72','ECS', 7, 4),
 ('10CS73','PTW', 7, 4),
 ('10CS74','DWDM', 7, 4),
 ('10CS75','JAVA', 7, 4),
 ('10CS76','SAN', 7, 4),
 ('15CS51', 'ME', 5, 4),
 ('15CS52','CN', 5, 4),
 ('15CS53','DBMS', 5, 4),
 ('15CS54','ATC', 5, 4),
 ('15CS55','JAVA', 5, 3),
 ('15CS56','AI', 5, 3),
 ('15CS41','M4', 4, 4),
 ('15CS42','SE', 4, 4),
 ('15CS43','DAA', 4, 4),
 ('15CS44','MPMC', 4, 4),
 ('15CS45','OOC', 4, 3),
 ('15CS46','DC', 4, 3),
 ('15CS31','M3', 3, 4),
 ('15CS32','ADE', 3, 4),
 ('15CS33','DSA', 3, 4),
 ('15CS34','CO', 3, 4),
 ('15CS35','USP', 3, 3),
 ('15CS36','DMS', 3, 3);


INSERT INTO IAMARKS VALUES ('1RN13CS091','10CS81','CSE8C', 15, 16, 18,17),
```

('1RN13CS091','10CS82','CSE8C', 12, 19, 14,17),
('1RN13CS091','10CS83','CSE8C', 19, 15, 20,20),
('1RN13CS091','10CS84','CSE8C', 20, 16, 19,20),
('1RN13CS091','10CS85','CSE8C', 15, 15, 12,15);

SELECT * FROM STUDENT;

	USN	SNAME	ADDRESS	PHONE	GENDER
▶	1RN13CS020	AKSHAY	BELAGAVI	8877881122	M
	1RN13CS062	SANDHYA	BENGALURU	7722829912	F
	1RN13CS066	SUPRIYA	MANGALURU	8877881122	F
	1RN13CS091	TEESHA	BENGALURU	7712312312	F
	1RN14CS010	ABHAY	BENGALURU	9900211201	M
	1RN14CS025	ASMI	BENGALURU	7894737377	F
	1RN14CS032	BHASKAR	BENGALURU	9923211099	M
	1RN15CS011	AJAY	TUMKUR	9845091341	M
	1RN15CS029	CHITRA	DAVANGERE	7696772121	F
	1RN15CS045	JEEVA	BELLARY	9944850121	M
	1RN15CS091	SANTOSH	MANGALURU	8812332201	M
	1RN16CS045	ISMAIL	KALBURGI	9900232201	M
	1RN16CS088	SAMEERA	SHIMOGA	9905542212	F
	1RN16CS122	VINAYAKA	CHIKAMAG…	8800880011	M
*	NULL	NULL	NULL	NULL	NULL

student 7 ×

SELECT * FROM CLASS;

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

SELECT * FROM SEMSEC;

SSID	SEM	SEC
CSE1A	1	A
CSE1B	1	B
CSE1C	1	C
CSE2A	2	A
CSE2B	2	B
CSE2C	2	C
CSE3A	3	A
CSE3B	3	B
CSE3C	3	C
CSE4A	4	A
CSE4B	4	B
CSE4C	4	C
CSE5A	5	A
CSE5B	5	B
CSE5C	5	C
CSE6A	6	A
CSE6B	6	B
CSE6C	6	C
CSE7A	7	A
CSE7B	7	B
CSE7C	7	C
CSE8A	8	A
CSE8B	8	B
CSE8C	8	C
NULL	NULL	NULL

semsec 8 ×

SELECT * FROM SUBJECT;

SUBCODE	TITLE	SEM	CREDITS
▶ 10CS71	OOAD	7	4
10CS72	ECS	7	4
10CS73	PTW	7	4
10CS74	DWDM	7	4
10CS75	JAVA	7	4
10CS76	SAN	7	4
10CS81	ACA	8	4
10CS82	SSM	8	4
10CS83	NM	8	4
10CS84	CC	8	4
10CS85	PW	8	4
15CS31	M3	3	4
15CS32	ADE	3	4
15CS33	DSA	3	4
15CS34	CO	3	4
15CS35	USP	3	3
15CS36	DMS	3	3
15CS41	M4	4	4
15CS42	SE	4	4
15CS43	DAA	4	4

15CS44	MPMC	4	4
15CS45	OOC	4	3
15CS46	DC	4	3
15CS51	ME	5	4
15CS52	CN	5	4
15CS53	DBMS	5	4
15CS54	ATC	5	4
15CS55	JAVA	5	3
15CS56	AI	5	3
* NULL	NULL	NULL	NULL

subject 10 ×

1. List all the student details studying in fourth semester 'C' section.

SELECT S.*, SS.SEM, SS.SEC
FROM STUDENT S, SEMSEC SS, CLASS C
WHERE S.USN = C.USN AND
SS.SSID = C.SSID AND
SS.SEM = 4 AND SS.SEC='C';

USN	SNAME	ADDRESS	PHONE	GENDER	SEM	SEC
▶ 1RN15CS091	SANTOSH	MANGALURU	8812332201	M	4	C

2. Compute the total number of male and female students in each semester and in each section.

SELECT SS.SEM, SS.SEC, S.GENDER, COUNT(S.GENDER) AS COUNT
FROM STUDENT S, SEMSEC SS, CLASS C
WHERE S.USN = C.USN AND
SS.SSID = C.SSID
GROUP BY SS.SEM, SS.SEC, S.GENDER
ORDER BY SEM;

SEM	SEC	GENDER	COUNT
3	A	M	1
3	B	F	1
3	C	M	1
4	A	F	1
4	A	M	1
4	B	M	1
4	C	M	1
7	A	F	1
7	A	M	2
8	A	F	1
8	A	M	1
8	B	F	1
8	C	F	1

3. Create a view of Test1 marks of student USN '1BI15CS101' in all subjects.

CREATE VIEW STUDENT_TEST1_MARKS_VIEW
AS
SELECT TEST1, SUBCODE
FROM IAMARKS
WHERE USN = '1RN13CS091';

SELECT * FROM STUDENT_TEST1_MARKS_VIEW;

Result Grid | Filter Rows:

TEST1	SUBCODE
15	10CS81
12	10CS82
19	10CS83
20	10CS84
15	10CS85

STUDENT_TEST1_MARKS_VIE... ×

4. Categorize students based on the following criterion:
If FinalIA = 17 to 20 then CAT = 'Outstanding'
If FinalIA = 12 to 16 then CAT = 'Average'
If FinalIA< 12 then CAT = 'Weak'
Give these details only for 8th semester A, B, and C section students.

SELECT S.USN,S.SNAME,S.ADDRESS,S.PHONE,S.GENDER,
(CASE
WHEN IA.FINALIA BETWEEN 17 AND 20 THEN 'OUTSTANDING'
WHEN IA.FINALIA BETWEEN 12 AND 16 THEN 'AVERAGE'
ELSE 'WEAK'
END) AS CAT
FROM STUDENT S, SEMSEC SS, IAMARKS IA, SUBJECT SUB
WHERE S.USN = IA.USN AND
SS.SSID = IA.SSID AND
SUB.SUBCODE = IA.SUBCODE AND
SUB.SEM = 8;

USN	SNAME	ADDRESS	PHONE	GENDER	CAT
1RN13CS091	TEESHA	BENGALURU	7712312312	F	OUTSTANDING
1RN13CS091	TEESHA	BENGALURU	7712312312	F	OUTSTANDING
1RN13CS091	TEESHA	BENGALURU	7712312312	F	OUTSTANDING
1RN13CS091	TEESHA	BENGALURU	7712312312	F	OUTSTANDING
1RN13CS091	TEESHA	BENGALURU	7712312312	F	AVERAGE