VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

Database Management Systems (22CS3PCDBM)

Submitted by

G MOHAMMED AWAIZ(1BM21CS060)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
October-2022 to Feb-2023

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "Database Management Systems (22CS3PCDBM) carried out by G MOHAMMED AWAIZ(1BM21CS060) **,** who is Bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a Database Management Systems (22CS3PCDBM) work prescribed for the said degree.

Dr. Nandhini Vineeth	Dr.Jyothi S Nayak
Assistant professor	Professor and Head
Department of CSE	Department of CSE
BMSCE, Bengaluru	BMSCE, Bengaluru

Index

1.Insurance Database

PERSON (driver_id: String, name: String, address: String)

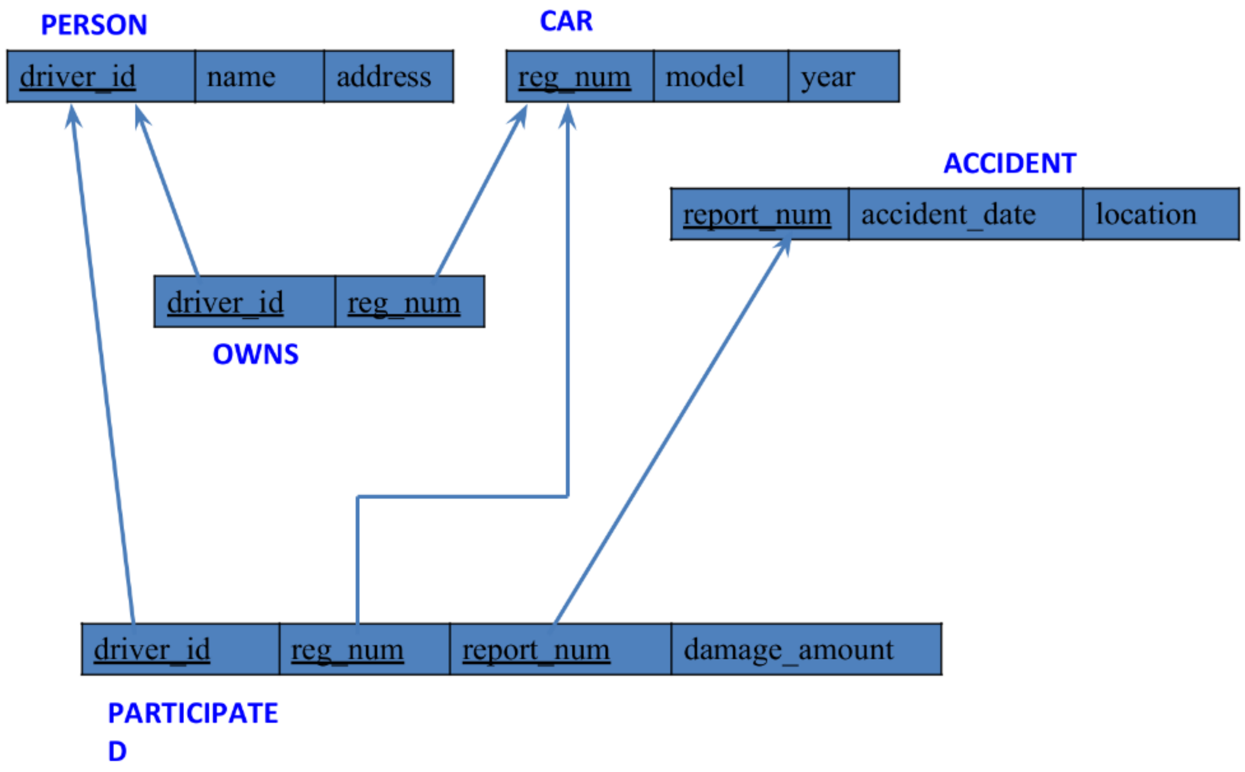
CAR (reg_num: String, model: String, year: int)

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)

- Create the above tables by properly specifying the primary keys and the foreign keys.
- Enter at least five tuples for each relation.
- Display the entire CAR relation in the ascending order of manufacturing year.
- Find the number of accidents in which cars belonging to a specific model (example 'Lancer') were involved.
- Find the total number of people who owned cars that were involved in accidents in 2008.



Creating Database

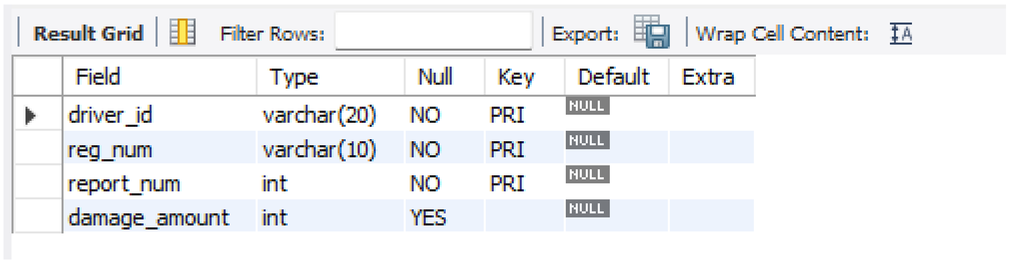
create database Awaiz_insurance;
use Awaiz_insurance;

Creating Tables

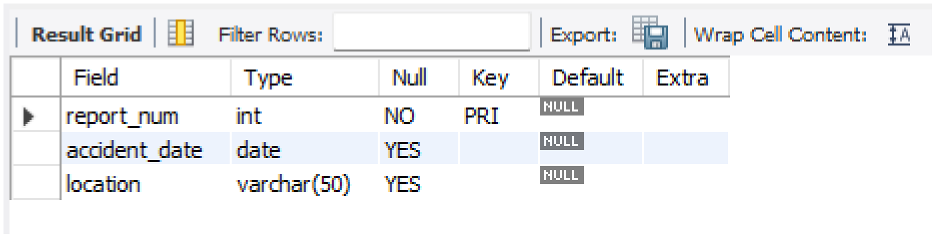
create table Person
(
driver_id varchar(20),
name varchar(20),
address varchar(30),
primary key(driver_id)
);
create table car
(
reg_num int,
model varchar(20),

```
year int,
primary key(reg_num)
);
create table owns
(
driver_id varchar(20),
reg_num int,
primary key(driver_id,reg_num),
foreign key(driver_id) references Person(driver_id),
foreign key(reg_num) references car(reg_num)
);

create table accident
(
report_num int,
accident_date date,

location varchar(30),
primary key(report_num)
);

desc accident;
create table participated
(
driver_id varchar(20),
reg_num int,
report_num int,
damage_amount int,
primary key(driver_id,reg_num,report_num),
foreign key(driver_id) references Person(driver_id),
foreign key(reg_num) references car(reg_num),
foreign key(report_num) references accident(report_num)
);
```

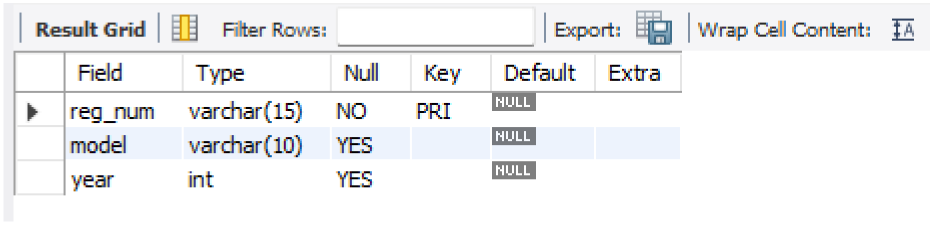
Structure of Tables



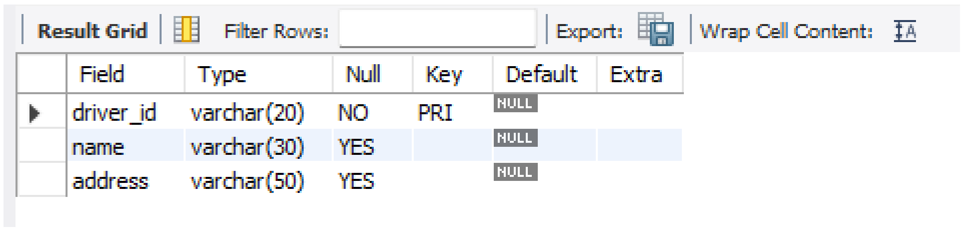
Field	Type	Null	Key	Default	Extra
driver_id	varchar(20)	NO	PRI	NULL	
reg_num	varchar(10)	NO	PRI	NULL	
report_num	int	NO	PRI	NULL	
damage_amount	int	YES		NULL	



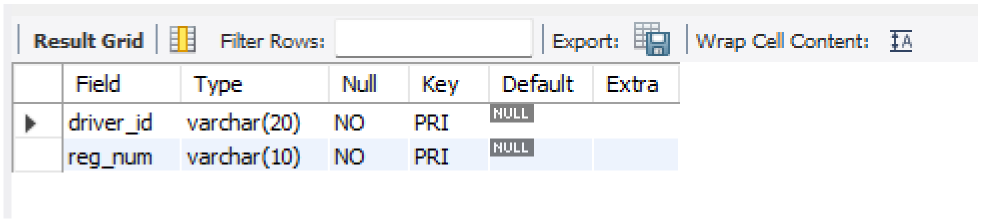
Field	Type	Null	Key	Default	Extra
report_num	int	NO	PRI	NULL	
accident_date	date	YES		NULL	
location	varchar(50)	YES		NULL	



Field	Type	Null	Key	Default	Extra
reg_num	varchar(15)	NO	PRI	NULL	
model	varchar(10)	YES		NULL	
year	int	YES		NULL	



Field	Type	Null	Key	Default	Extra
driver_id	varchar(20)	NO	PRI	NULL	
name	varchar(30)	YES		NULL	
address	varchar(50)	YES		NULL	

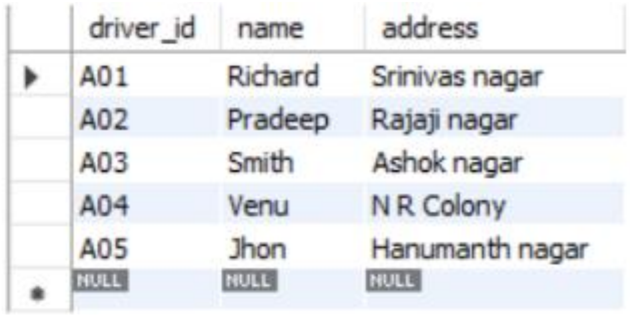


Field	Type	Null	Key	Default	Extra
driver_id	varchar(20)	NO	PRI	NULL	
reg_num	varchar(10)	NO	PRI	NULL	

Inserting Values

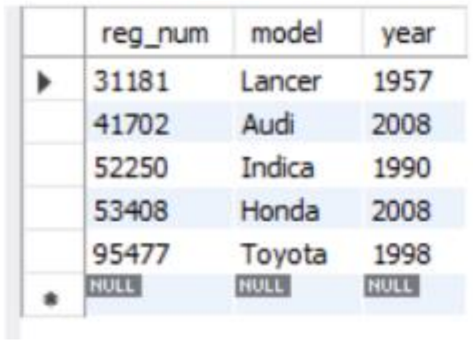
```
insert into Person values("A01","Richard","Srinivas nagar");
insert into Person values("A02","Pradeep","Rajaji nagar");
insert into Person values("A03","Smith","Ashok nagar");
insert into Person values("A04","Venu","N R Colony ");
insert into Person values("A05","Jhon","Hanumanth nagar");


insert into car values(052250,"Indica",1990);
insert into car values(031181,"Lancer",1957);
insert into car values(095477,"Toyota",1998);
insert into car values(053408,"Honda",2008);
insert into car values(041702,"Audi",2008);


insert into owns values("A01",052250);
insert into owns values("A02",031181);
insert into owns values("A03",095477);
insert into owns values("A04",053408);
insert into owns values("A05",041702);


insert into accident values(11,'2003-01-01','Mysore road');
insert into accident values(12,'2004-02-02','South end circle');
insert into accident values(13,'2003-01-21','Bull temple end');
insert into accident values(14,'2008-02-17','Mysore road');
insert into accident values(15,'2004-03-05','Kanakapura road');


insert into participated values('A01','052250',11,10000);
insert into participated values('A02','053408',12,50000);
insert into participated values('A03','095477',13,25000);
insert into participated values('A04','031181',14,3000);
insert into participated values('A05','041702',15,5000);
```

select * from Person;



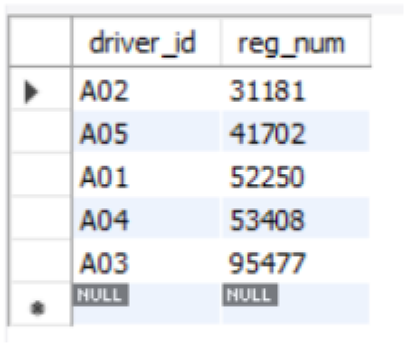
	driver_id	name	address
▶	A01	Richard	Srinivas nagar
	A02	Pradeep	Rajaji nagar
	A03	Smith	Ashok nagar
	A04	Venu	N R Colony
	A05	Jhon	Hanumanth nagar
*	NULL	NULL	NULL

select * from car;



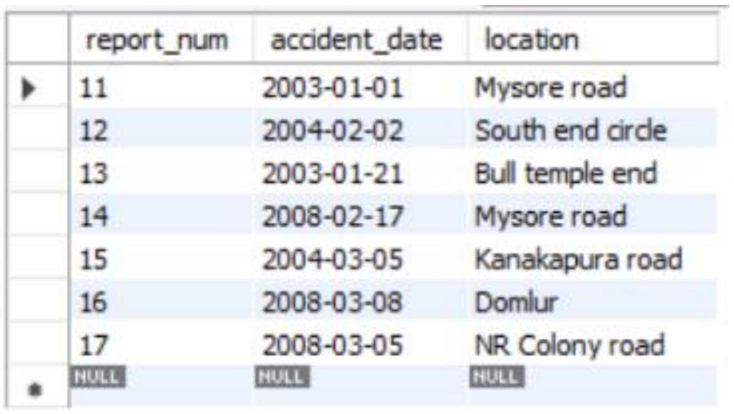
	reg_num	model	year
▶	31181	Lancer	1957
	41702	Audi	2008
	52250	Indica	1990
	53408	Honda	2008
	95477	Toyota	1998
*	NULL	NULL	NULL

select * from owns;



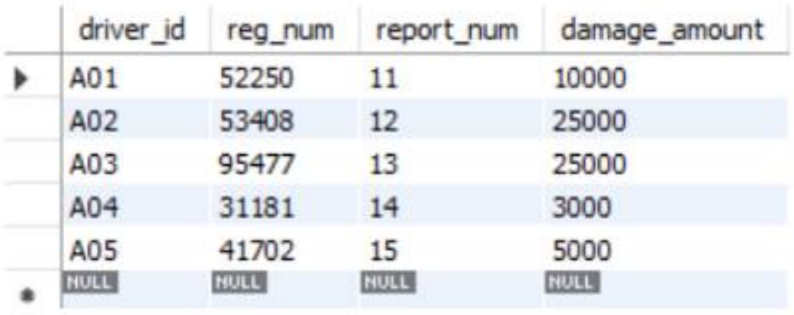
	driver_id	reg_num
▶	A02	31181
	A05	41702
	A01	52250
	A04	53408
	A03	95477
*	NULL	NULL

select * from accident;



report_num	accident_date	location
11	2003-01-01	Mysore road
12	2004-02-02	South end circle
13	2003-01-21	Bull temple end
14	2008-02-17	Mysore road
15	2004-03-05	Kanakapura road
16	2008-03-08	Domlur
17	2008-03-05	NR Colony road
NULL	NULL	NULL

select * from participated;



driver_id	reg_num	report_num	damage_amount
A01	52250	11	10000
A02	53408	12	25000
A03	95477	13	25000
A04	31181	14	3000
A05	41702	15	5000
NULL	NULL	NULL	NULL

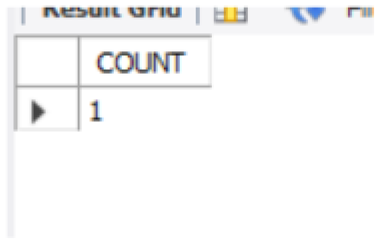
Queries

1. Update the damage amount to 25000 for the car with a specific reg-num (example '053408') for which the accident report number was 12

```
update participated
set damage_amount=25000
where reg_num='053408' and report_num=12;
```

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

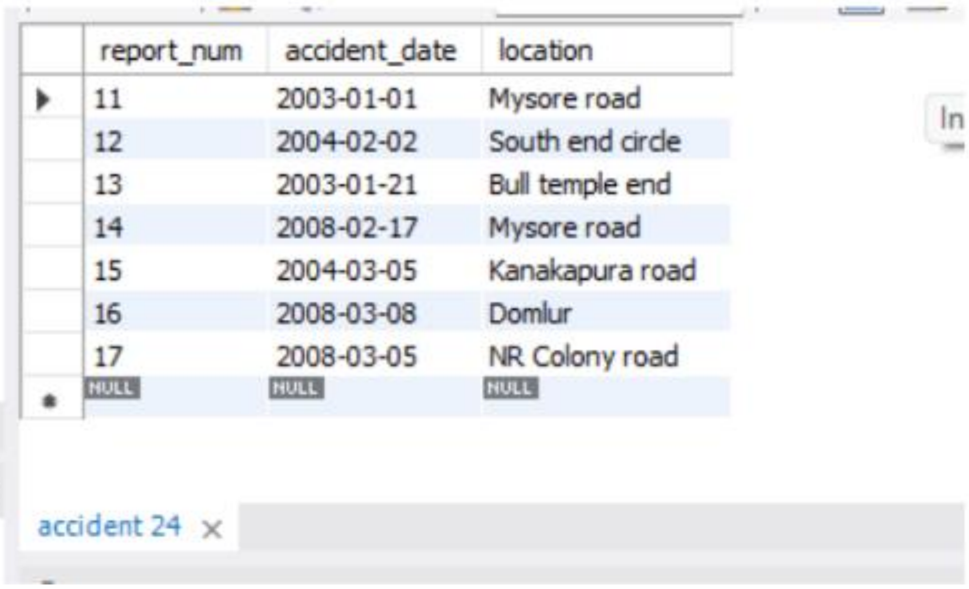
```
select count(distinct driver_id) COUNT
from participated a, accident b
where a.report_num=b.report_num and b.accident_date like '%08%';
```



	COUNT
▶	1

3.Add a new accident to the database.

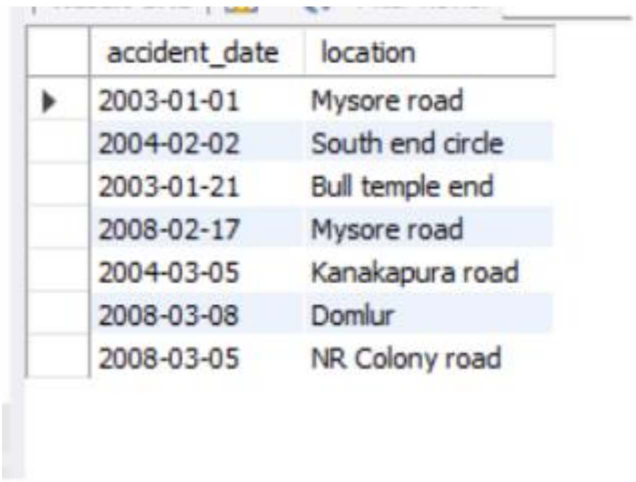
```
insert into accident values(16,'2008-03-08',"Domlur");

select * from accident;
```



	report_num	accident_date	location
▶	11	2003-01-01	Mysore road
	12	2004-02-02	South end circle
	13	2003-01-21	Bull temple end
	14	2008-02-17	Mysore road
	15	2004-03-05	Kanakapura road
	16	2008-03-08	Domlur
	17	2008-03-05	NR Colony road
●	NULL	NULL	NULL

4.Display Accident date and location

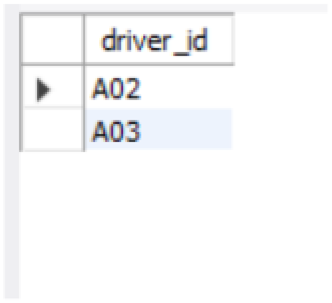
select accident_date, location from accident;



	accident_date	location
▶	2003-01-01	Mysore road
	2004-02-02	South end circle
	2003-01-21	Bull temple end
	2008-02-17	Mysore road
	2004-03-05	Kanakapura road
	2008-03-08	Domlur
	2008-03-05	NR Colony road

5.Display driver id who did accident with damage amount greater than or equal to Rs.25000

select driver_id from participated where damage_amount>=25000;

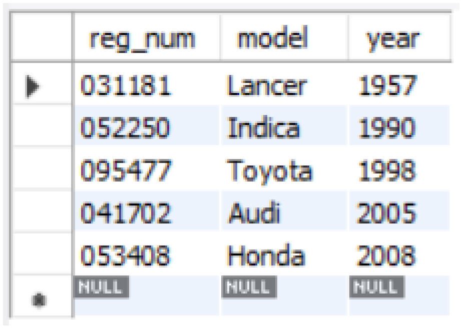


driver_id
A02
A03

2.More Queries on Insurance Database

1.Display the entire CAR relation in the ascending order of
manufacturing year.

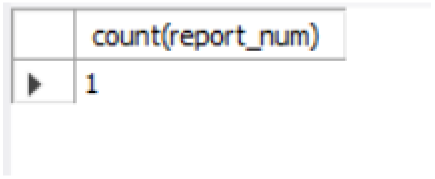
select * from car order by year asc;



	reg_num	model	year
▶	031181	Lancer	1957
	052250	Indica	1990
	095477	Toyota	1998
	041702	Audi	2005
	053408	Honda	2008
*	NULL	NULL	NULL

2.Find the number of accidents in which cars belonging to a specific model (example 'Lancer')
were involved.

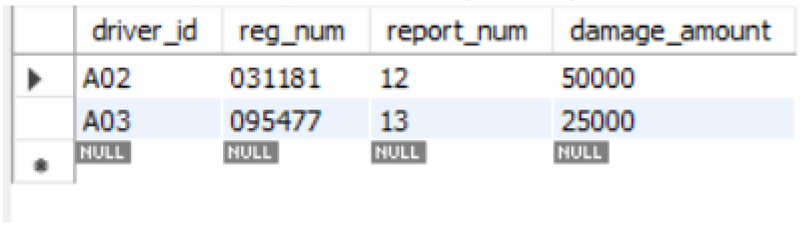
select count(report_num) from car c, participated p where c.reg_num=p.reg_num and c.model='Lancer';



	count(report_num)
▶	1

3.LIST THE ENTIRE PARTICIPATED RELATION IN THE DESCENDING ORDER OF DAMAGE
AMOUNT.

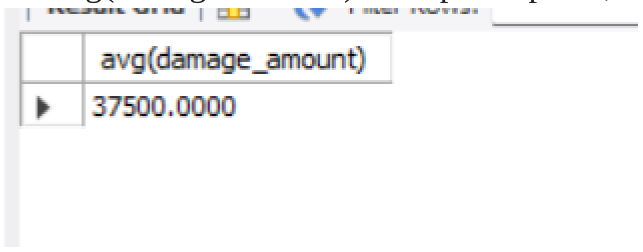
select * from participated order by damage_amount desc;



driver_id	reg_num	report_num	damage_amount
A02	031181	12	50000
A03	095477	13	25000
NULL	NULL	NULL	NULL

4.FIND THE AVERAGE DAMAGE AMOUNT

select avg(damage_amount) from participated;



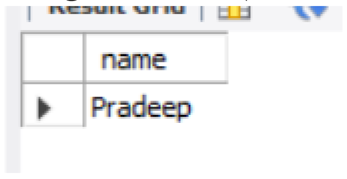
avg(damage_amount)
37500.0000

5.DELETE THE TUPLE WHOSE DAMAGE AMOUNT IS BELOW THE AVERAGE DAMAGE AMOUNT

delete from participated
where damage_amount < (select p.damage_amount from(select avg(damage_amount) as damage_amount from participated) p);

6.LIST THE NAME OF DRIVERS WHOSE DAMAGE IS GREATER THAN THE AVERAGE DAMAGE
AMOUNT.

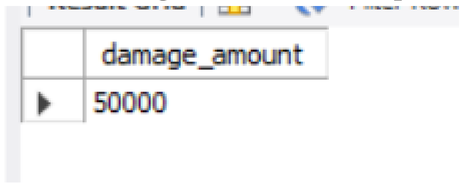
select name from person,participated where person.driver_id=participated.driver_id and damage_amount>(select avg(damage_amount) from participated);



	name
▶	Pradeep

7.FIND MAXIMUM DAMAGE AMOUNT.

select damage_amount from participated having max(damage_amount);



	damage_amount
▶	50000

3.Bank Database

Branch (branch-name: String, branch-city: String, assets: real)

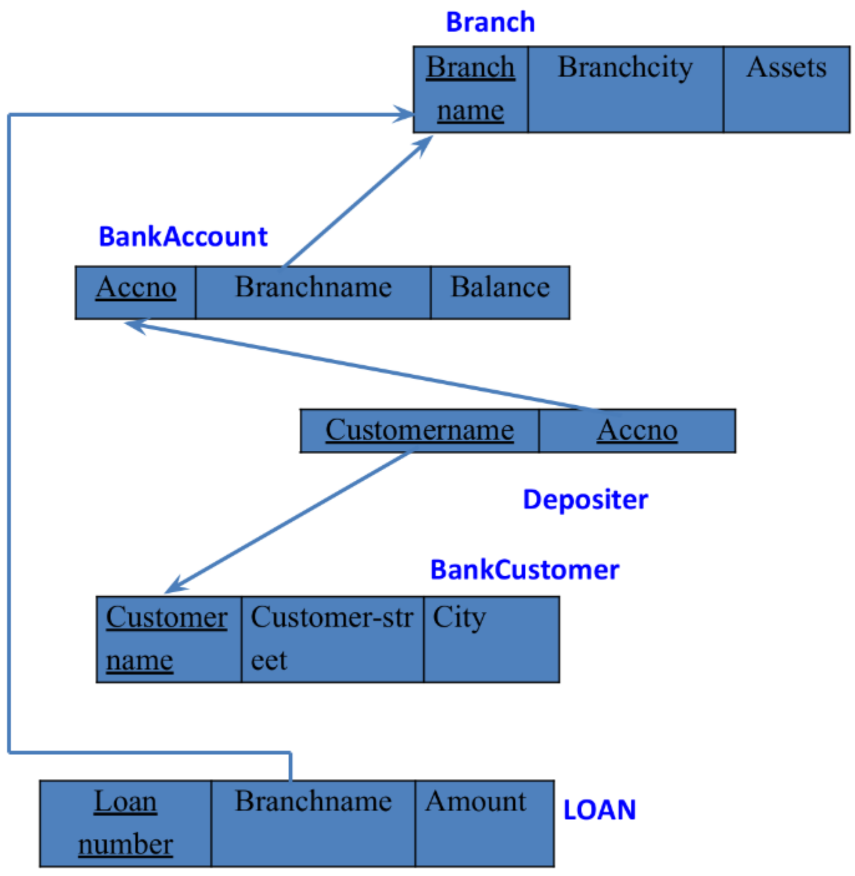
BankAccount(accno: int, branch-name: String, balance: real)

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)



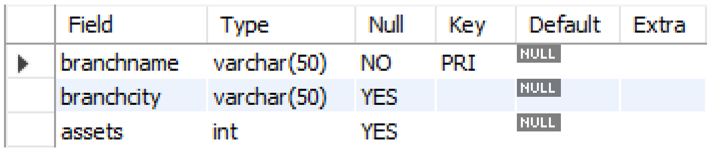
Creating Database

```
create database bank;
use bank;
```

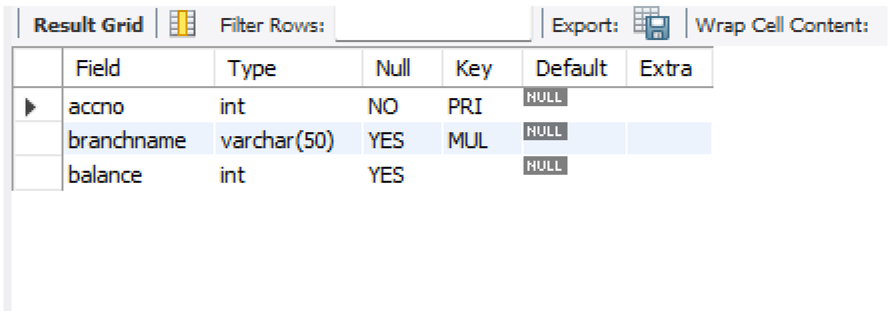
Creating Tables

```
create table branch
( branchname varchar(50), branchcity varchar(50), assets int,
primary key(branchname) );

create table bankaccount
( accno int, branchname varchar(50), balance int,
primary key(accno),
foreign key(branchname) references branch(branchname));

create table bankcustomer
(customername varchar(50), customerstreet varchar(50), city varchar(50),
primary key(customername));

create table depositer
( customername varchar(50), accno int,
primary key(accno),
foreign key(accno) references bankaccount(accno),
foreign key(customername) references bankcustomer(customername));

create table loan
( loannumber int, branchname varchar(50), amount int,
foreign key(branchname) references branch(branchname));
```

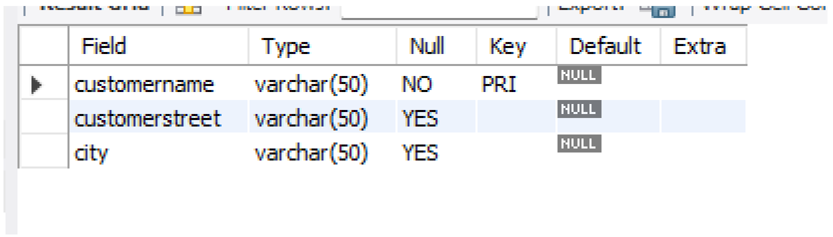
Structure of Tables



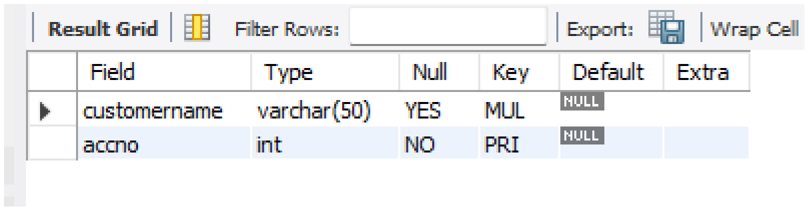
Field	Type	Null	Key	Default	Extra
branchname	varchar(50)	NO	PRI	NULL	
branchcity	varchar(50)	YES		NULL	
assets	int	YES		NULL	



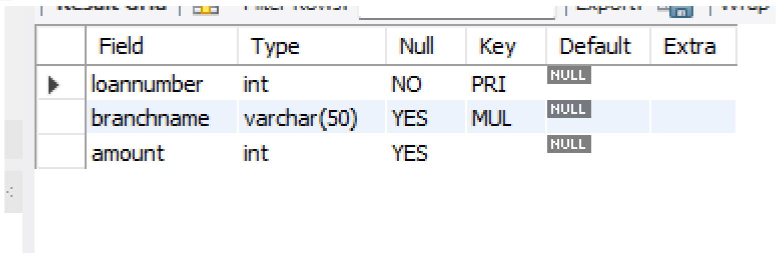
Field	Type	Null	Key	Default	Extra
accno	int	NO	PRI	NULL	
branchname	varchar(50)	YES	MUL	NULL	
balance	int	YES		NULL	



Field	Type	Null	Key	Default	Extra
customername	varchar(50)	NO	PRI	NULL	
customerstreet	varchar(50)	YES		NULL	
city	varchar(50)	YES		NULL	



Field	Type	Null	Key	Default	Extra
customername	varchar(50)	YES	MUL	NULL	
accno	int	NO	PRI	NULL	



Field	Type	Null	Key	Default	Extra
loannumber	int	NO	PRI	NULL	
branchname	varchar(50)	YES	MUL	NULL	
amount	int	YES		NULL	

Inserting Values

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

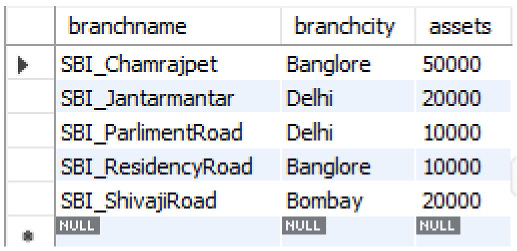
```
insert into bankaccount values(1,"SBI_Chamrajpet",2000);
insert into bankaccount values(2,"SBI_ResidencyRoad",5000);
insert into bankaccount values(3,"SBI_ShivajiRoad",6000);
insert into bankaccount values(4,"SBI_ParlimentRoad",9000);
insert into bankaccount values(5,"SBI_Jantarmantar",8000);
insert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

```
insert into bankcustomer values("Avinash","Bull_Temple_Road","Banglore");
insert into bankcustomer values("Dinesh","Bannergatta_Road","Banglore");
insert into bankcustomer values("Mohan","NationalCollege_Road","Banglore");
insert into bankcustomer values("Nikil","Akbar_Road","Delhi");
insert into bankcustomer values("Ravi","Prithviraj_Road","Delhi");
```

```
insert into depositer values("Avinash",1);
insert into depositer values("Dinesh",2);
insert into depositer values("Nikil",4);
insert into depositer values("Ravi",5);
insert into depositer values("Avinash",8);
insert into depositer values("Nikil",9);
insert into depositer values("Dinesh",10);
insert into depositer values("Nikil",11);
```

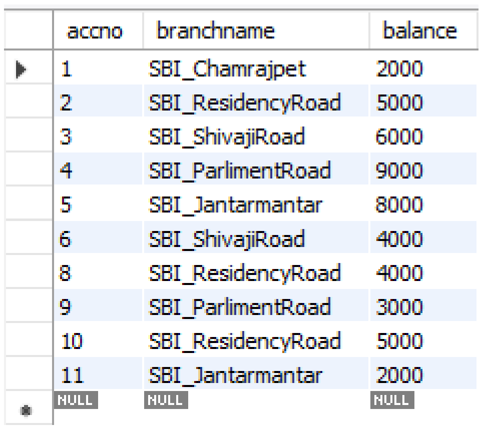
```
insert into loan values(1,"SBI_Chamrajpet",1000);
insert into loan values(2,"SBI_ResidencyRoad",2000);
insert into loan values(3,"SBI_ShivajiRoad",3000);
insert into loan values(4,"SBI_ParlimentRoad",4000);
insert into loan values(5,"SBI_Jantarmantar",5000);
```

```
select * from branch;
```



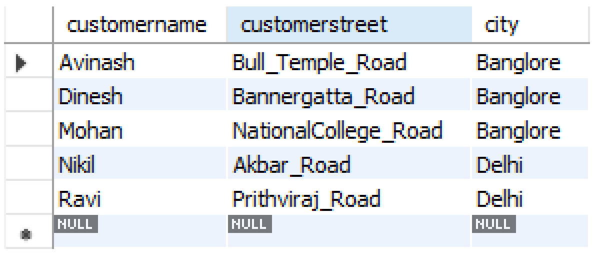
branchname	branchcity	assets
SBI_Chamrajpet	Banglore	50000
SBI_Jantarmantar	Delhi	20000
SBI_ParlimentRoad	Delhi	10000
SBI_ResidencyRoad	Banglore	10000
SBI_ShivajiRoad	Bombay	20000
NULL	NULL	NULL

```
select * from bankaccount;
```



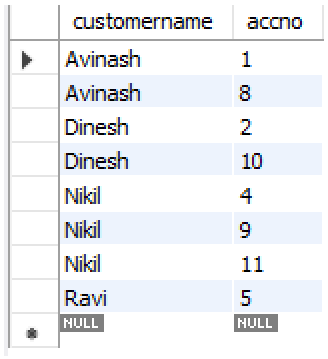
accno	branchname	balance
1	SBI_Chamrajpet	2000
2	SBI_ResidencyRoad	5000
3	SBI_ShivajiRoad	6000
4	SBI_ParlimentRoad	9000
5	SBI_Jantarmantar	8000
6	SBI_ShivajiRoad	4000
8	SBI_ResidencyRoad	4000
9	SBI_ParlimentRoad	3000
10	SBI_ResidencyRoad	5000
11	SBI_Jantarmantar	2000
NULL	NULL	NULL

select * from bankcustomer;



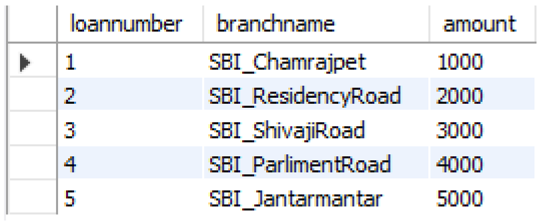
customername	customerstreet	city
Avinash	Bull_Temple_Road	Banglore
Dinesh	Bannergatta_Road	Banglore
Mohan	NationalCollege_Road	Banglore
Nikil	Akbar_Road	Delhi
Ravi	Prithviraj_Road	Delhi
NULL	NULL	NULL

select * from depositer;



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

select * from loan;

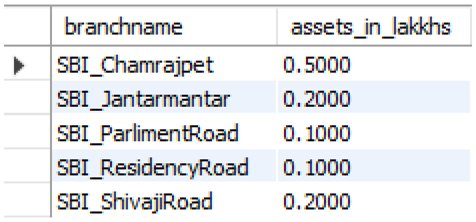


loannumber	branchname	amount
1	SBI_Chamrajpet	1000
2	SBI_ResidencyRoad	2000
3	SBI_ShivajiRoad	3000
4	SBI_ParlimentRoad	4000
5	SBI_Jantarmantar	5000

Queries

1.Display the branch name and assets from all branches in lakhs of rupees and rename the assets column to 'assets in lakhs'.

select branchname,assets/100000 as assets_in_lakkhs from branch;



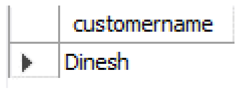
branchname	assets_in_lakkhs
SBI_Chamrajpet	0.5000
SBI_Jantarmantar	0.2000
SBI_ParlimentRoad	0.1000
SBI_ResidencyRoad	0.1000
SBI_ShivajiRoad	0.2000

2.Find all the customers who have at least two accounts at the same branch (ex. SBI_ResidencyRoad).

select d.customername from bankaccount b, depositer d

where b.branchname="SBI_ResidencyRoad" and b.accno=d.accno

group by d.customername having count(d.accno)>=2;

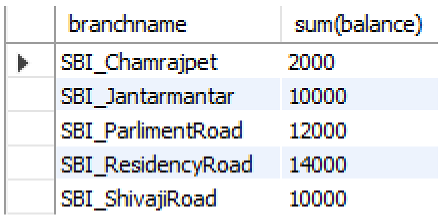


customername
Dinesh

3.CREATE A VIEW WHICH GIVES EACH BRANCH THE SUM OF THE AMOUNT OF ALL THE LOANS AT THE BRANCH.

```
create view sum_of_loan
as select branchname,sum(balance)
from bankaccount
group by branchname;
```

```
select * from sum_of_loan;
```



branchname	sum(balance)
SBI_Chamrajpet	2000
SBI_Jantarmantar	10000
SBI_ParlimentRoad	12000
SBI_ResidencyRoad	14000
SBI_ShivajiRoad	10000

4.More Queries on Bank Database

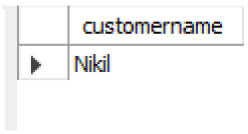
1.Find all the customers who have an account at all the branches located in a specific city (Ex. Delhi).

```
insert into bankaccount values(12,"SBI_MatriMarg",2000);
insert into branch
values("SBI_MatriMarg","Delhi",200000); insert into
depositer values("Nikil",12);

create table borrower(customername varchar(50), loannumber int,
foreign key(customername) references bankcustomer(customername),
foreign key(loannumber) references loan(loannumber));

insert into borrower
values("Avinash",1),("Dinesh",2),("Mohan",3),("Nikil",4),("Ravi",5);
```

Queries

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

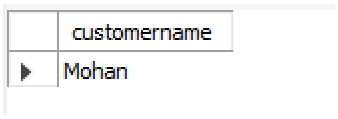
```
select d.customername from branch b, depositer d, bankaccount ba where
b.branchcity='Delhi' and d.accno=ba.accno and b.branchname=ba.branchname
group by d.customername having count(customername)>1;
```



	customername
▶	Nikil

2.Find all customers who have a loan at the bank but do not have an account.select distinct b.customername from borrower b, depositer d where b.Customername not in(

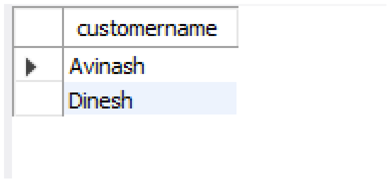
```
                    select d.customername from loan l,depositer d, borrower b where
                    l.loannumber=b.loannumber and
d.customername=b.customername );
```



	customername
▶	Mohan

3.Find all customers who have both an account and a loan at the Bangalore branch

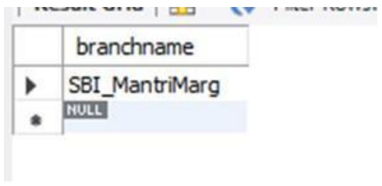
```
select distinct d.customername from depositer d
where d.customername in(
                select d.customername from branch br,depositer d, bankaccount ba
                where br.branchcity="Banglore" and br.branchname=ba.branchname
                and ba.accno=d.accno and d.customername in(
select customername from borrower));
```



```
customername
Avinash
Dinesh
```

4.Find the names of all branches that have greater assets than all branches located in Bangalore.

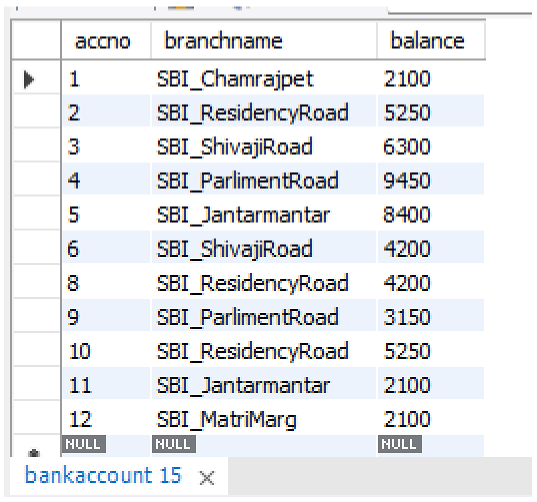
```
select b.branchname from branch b
where b.assets> all (
select sum(b.assets) from branch b
               where b.branchcity='Banglore' );
```



```
branchname
SBI_MantriMarg
NULL
```

5.Update the Balance of all accounts by 5%

```
update bankaccount set balance=(balance+(balance*0.05));
select * from bankaccount
```

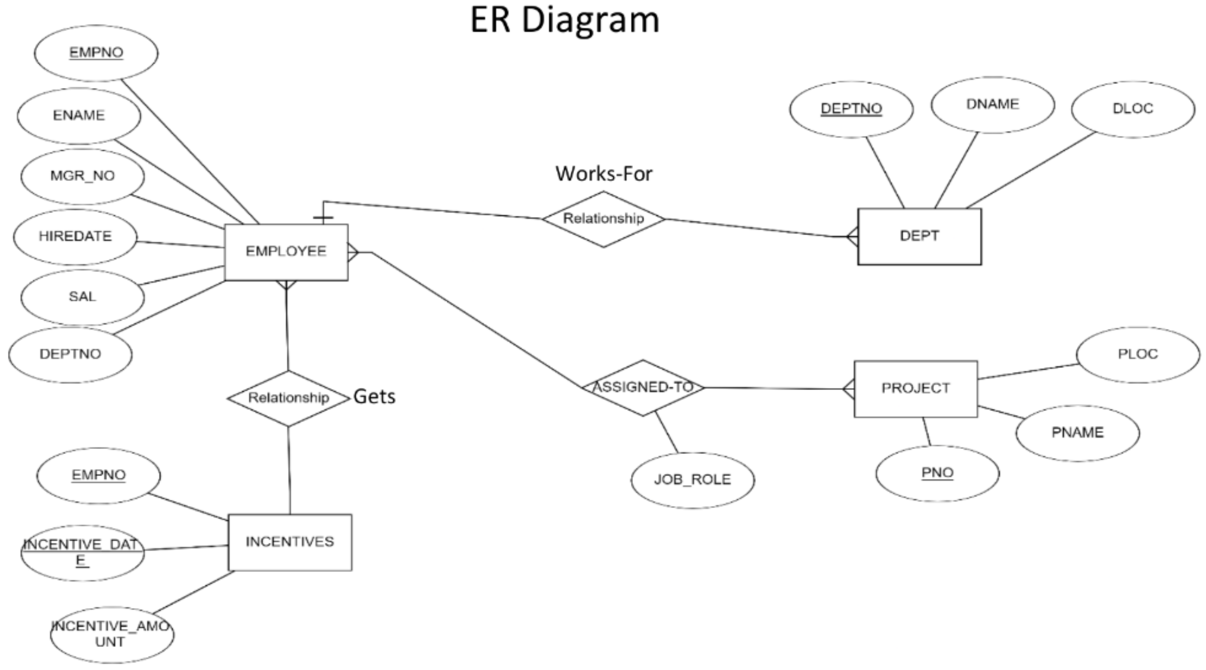


accno	branchname	balance
1	SBI_Chamrajpet	2100
2	SBI_ResidencyRoad	5250
3	SBI_ShivajiRoad	6300
4	SBI_ParlimentRoad	9450
5	SBI_Jantarmantar	8400
6	SBI_ShivajiRoad	4200
8	SBI_ResidencyRoad	4200
9	SBI_ParlimentRoad	3150
10	SBI_ResidencyRoad	5250
11	SBI_Jantarmantar	2100
12	SBI_MatriMarg	2100
NULL	NULL	NULL

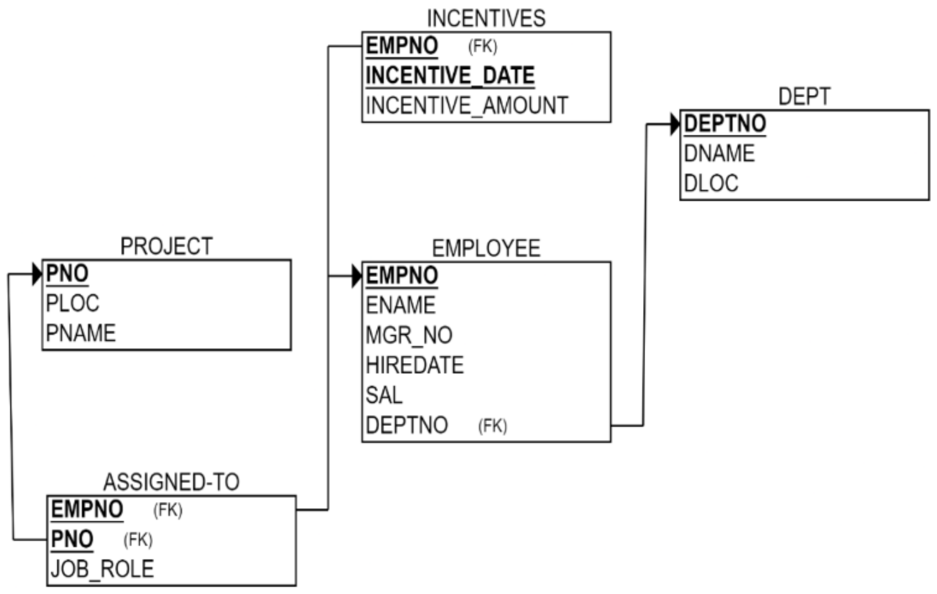
bankaccount 15

5.Employee Database

ER Diagram



Schema Diagram



Creating Database

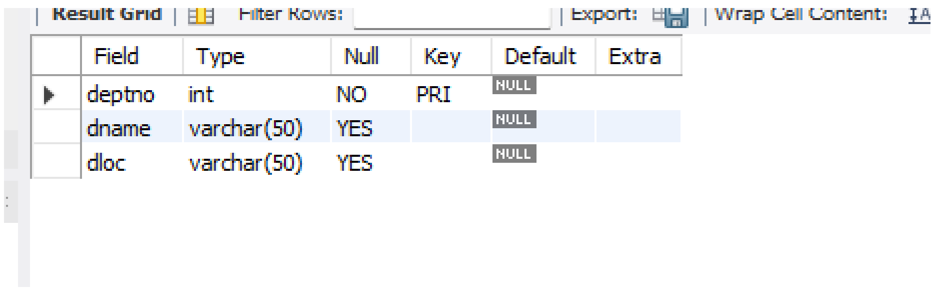
```
create database employee;
use employee;
```

Creating Tables

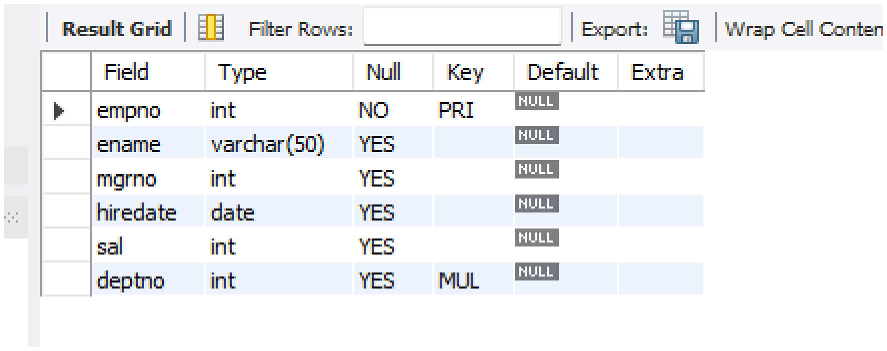
```
create table dept
( deptno int, dname varchar(50), dloc varchar(50),
 primary key(deptno));

create table employee
( empno int, ename varchar(50), mgrno int, hiredate date, sal int,
deptno int,  primary key(empno), foreign key(deptno) references
dept(deptno)  on update cascade on delete cascade);

create table incentive
( empno int, incentivedate date, incentiveamount int,
 primary key(incentivedate),
 foreign key(empno) references employee(empno)
 on update cascade on delete cascade);

create table project
(pno int, ploc varchar(50), pname varchar(50),
primary key(pno));
```

```
create table assignedto
( empno int,pno int, jobrole varchar(50),
foreign key(empno) references employee(empno),
foreign key(pno) references project(pno)
on update cascade on delete cascade);
```

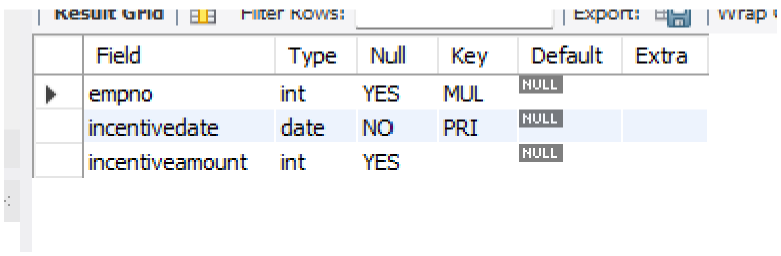
Structure of Tables



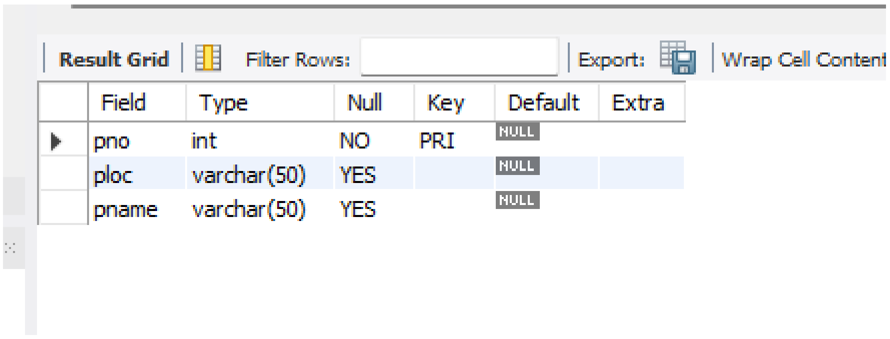
Field	Type	Null	Key	Default	Extra
deptno	int	NO	PRI	NULL	
dname	varchar(50)	YES		NULL	
dloc	varchar(50)	YES		NULL	



Field	Type	Null	Key	Default	Extra
empno	int	NO	PRI	NULL	
ename	varchar(50)	YES		NULL	
mgrno	int	YES		NULL	
hiredate	date	YES		NULL	
sal	int	YES		NULL	
deptno	int	YES	MUL	NULL	

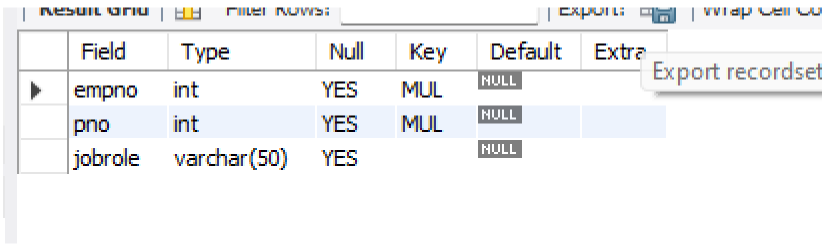


Field	Type	Null	Key	Default	Extra
empno	int	YES	MUL	NULL	
incentivedate	date	NO	PRI	NULL	
incentiveamount	int	YES		NULL	



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

Field	Type	Null	Key	Default	Extra
pno	int	NO	PRI	NULL	
ploc	varchar(50)	YES		NULL	
pname	varchar(50)	YES		NULL	



Field	Type	Null	Key	Default	Extra
empno	int	YES	MUL	NULL	
pno	int	YES	MUL	NULL	
jobrole	varchar(50)	YES		NULL	

Inserting Values

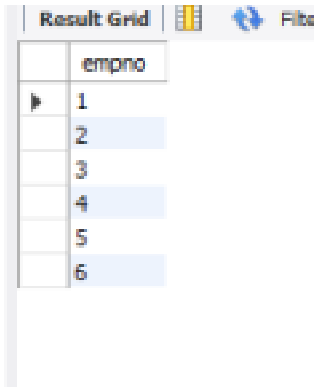
```
insert into dept values

(1,"Admin","Banglore"),
(2,"Sales","Bangolre"),

(3,"Finance","Hyderbad"),

(4,"Marketing","Mysore"),

(5,"Shipping","Hyderbad");

insert into dept values(6,"Purchasing","Mysore");


insert into employee values(1,"Avinash",3,"2000-02-

14",25000,1), (2,"Balaji",3,"1999-05-11",31000,3),

(3,"Dinesh",NULL,"1992-01-26",46000,2),

(4,"Chandan",3,"2001-05-21",28000,4),

(5,"Aravind",2,"1998-09-22",17000,5);

insert into employee values(6,"Amal",3,"2003-02-14",25000,6);


insert into incentive values(1,"2005-03-

23",5000), (3,"2001-08-23",50000),

(5,"2011-04-02",1500);
```

```
insert into project
values(11,"Banglore","Documentation"),
(12,"Banglore","Selling"),
(13,"Hyderbad","Accounting"),
(14,"Mysore","Advertising"),
(15,"Hyderbad","Transportation");
insert into project
values(16,"Mysore","Purchasing"); insert into
project values(17,"Hubli","Presentatiom");


insert into assignedto
values(1,11,"Administration"),
(2,12,"Salesman"),
(3,13,"Accounts"),
(4,14,"Advertising"),
(5,15,"Transporting");
insert into assignedto values(6,16,"Purchasing");
```

Queries

1.Retrieve the employee numbers of all employees who work on project located in Banglore, Hyderbad, or Mysore

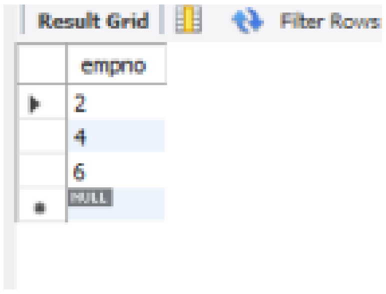
```
select empno from assignedto e where e.pno=any(select p.pno from project p
where  ploc="Banglore" or ploc="Hyderbad" or ploc="Mysore");
```



```
empno
1
2
3
4
5
6
```

2.Get Employee ID's of those employees who didn't receive incentives

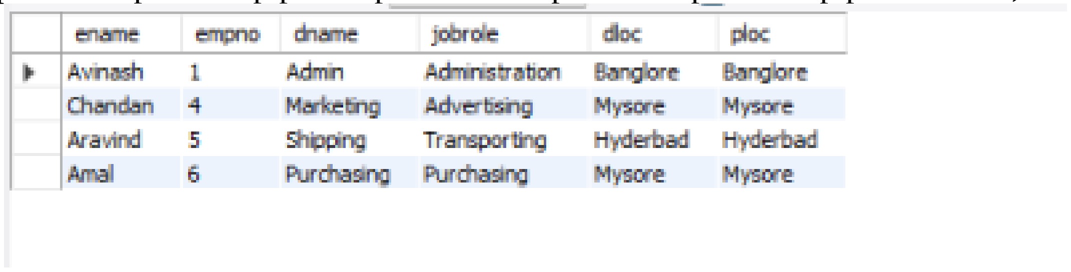
```
select e.empno from employee e
where e.empno not in
 (select i.empno from incentive i);
```



```
empno
2
4
6
NULL
```

3.Write a SQL query to find the employees name, number, dept, job_role, department location and project location who are working for a project location same as his/her department location.

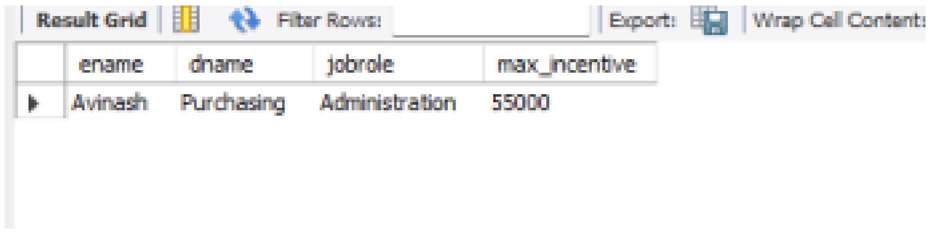
select e.ename ename, e.empno empno, d.dname dname, a.jobrole jobrole, d.dloc dloc, p.ploc ploc from project p, dept d, employee e, assignedto a
where e.empno=a.empno and p.pno=a.pno and e.deptno=d.deptno and p.ploc=d.dloc;



ename	empno	dname	jobrole	dloc	ploc
Avinash	1	Admin	Administration	Banglore	Banglore
Chandan	4	Marketing	Advertising	Mysore	Mysore
Aravind	5	Shipping	Transporting	Hyderbad	Hyderbad
Amal	6	Purchasing	Purchasing	Mysore	Mysore

4. Find the employee name, dept name and job_role of an employee who received max incentive in year 2005

select e.ename, d.dname, a.jobrole, max(i.incentiveamount) max_incentive from employee e, dept d, incentive i, assignedto a where incentivedate between "2005-01-01" and "2005-12-31";

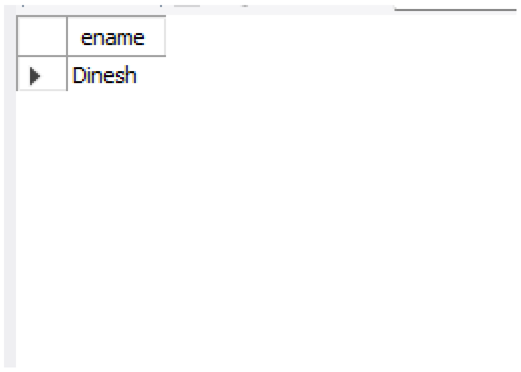


ename	dname	jobrole	max_incentive
Avinash	Purchasing	Administration	55000

6.More Queries on Employee Database

1.List the name of the managers with the maximum employees select e.ename from employee e,employee f

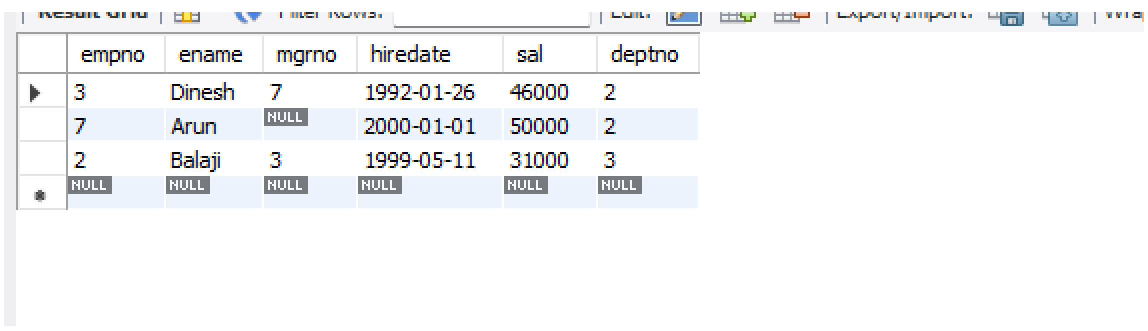
```
 where e.empno=f.mgrno
 group by e.empno
 having count(*)=(select
 max(mycount) from
 (select count(*) mycount
 from employee
 group by mgrno) a);
```



	ename
▶	Dinesh

2.Display those managers name whose salary is more than average salary of his employee.

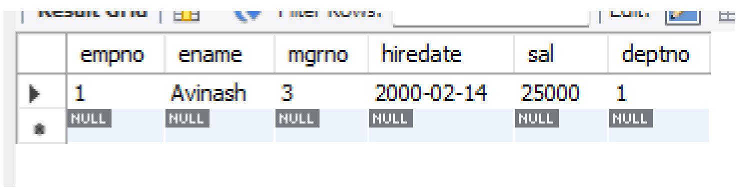
```
    select *
   from employee m
   where m.empno in
       (select mgrno
     from employee)
     and m.sal>(select avg(n.sal)
  from employee n
  where n.mgrno=m.empno);
```



	empno	ename	mgrno	hiredate	sal	deptno
▶	3	Dinesh	7	1992-01-26	46000	2
	7	Arun	NULL	2000-01-01	50000	2
	2	Balaji	3	1999-05-11	31000	3
●	NULL	NULL	NULL	NULL	NULL	NULL

3.Find the employee details who got second maximum incentive in 2005.

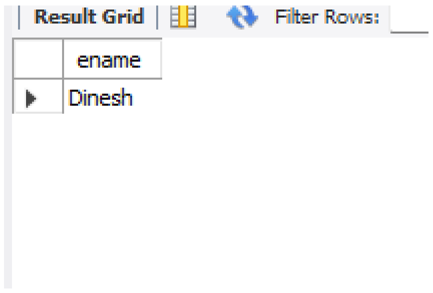
select * from employee where empno=
(select iii.empno from incentive iii
where iii.incentiveamount=
(select max(ii.incentiveamount) from incentive ii
where ii.incentiveamount<(select max(i.incentiveamount) from
incentive i where i.incentivedate between "2005-01-01" and "2005-12-31")
and incentivedate between "2005-01-01" and "2005-12-31"));



	empno	ename	mgrno	hiredate	sal	deptno
▶	1	Avinash	3	2000-02-14	25000	1
●	NULL	NULL	NULL	NULL	NULL	NULL

4.Display those employees who are working in the same department where his manager is working.

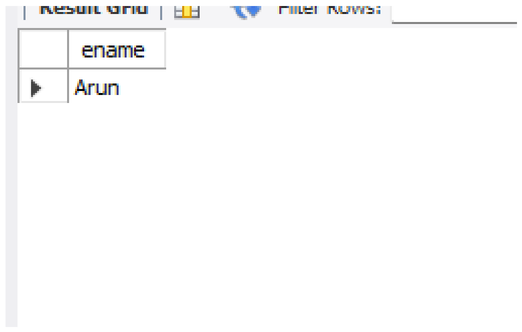
```
select e.ename from employee e
where e.deptno=(select f.deptno from employee f
                                        where e.mgrno=f.empno);
```


Result Grid | Filter Rows:

```
ename
Dinesh
```

5.Find the name of the second top level managers of each department.

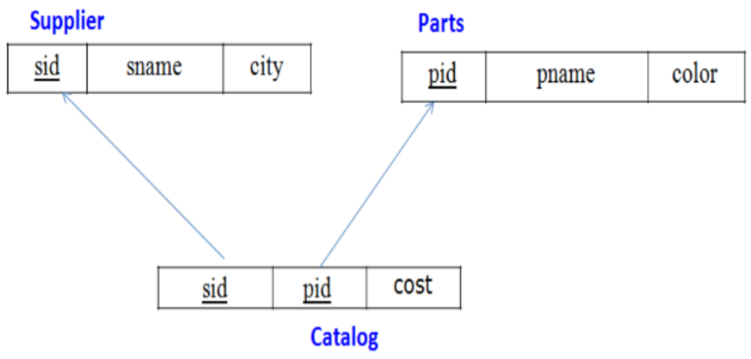
```
select ename from employee where empno in(select distinct mgrno
from employee
where empno in (select distinct mgrno
from employee
where empno in(select distinct mgrno from employee)));
```



```
ename
Arun
```

7.Supplier Database

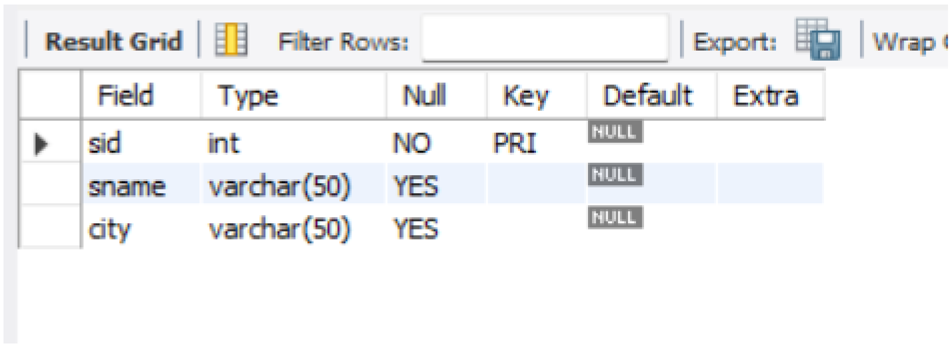
Schema Diagram



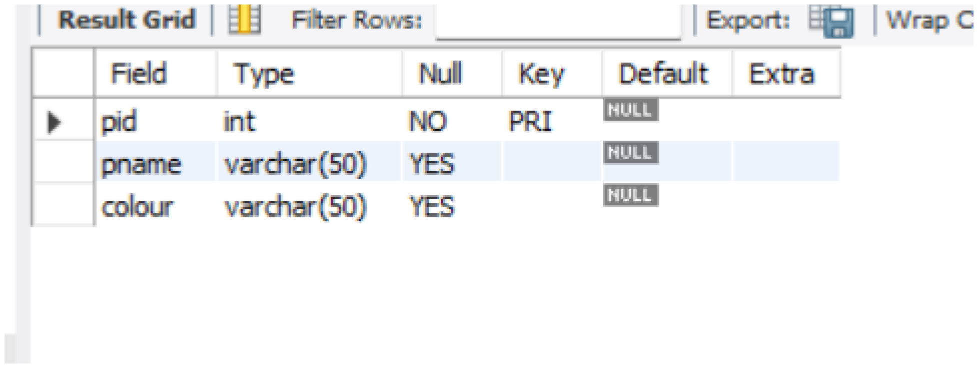
Creating Tables

```
create table supplier
(sid int, sname varchar(50), city
 varchar(50), primary key(sid));

 create table parts
 (pid int, pname varchar(50), colour
 varchar(50), primary key(pid));

create table
catalog (sid int,
pid int, cost int,
 foreign key(sid) references
 supplier(sid), foreign key(pid)
 references parts(pid));
```

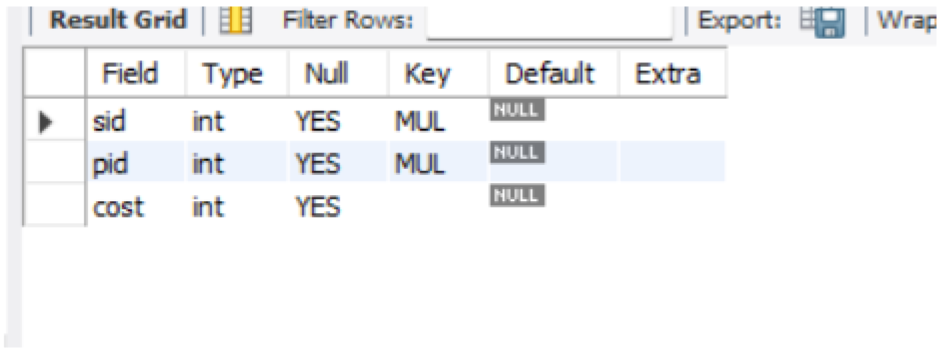
Structure of Tables



Field	Type	Null	Key	Default	Extra
sid	int	NO	PRI	NULL	
sname	varchar(50)	YES		NULL	
city	varchar(50)	YES		NULL	



Field	Type	Null	Key	Default	Extra
pid	int	NO	PRI	NULL	
pname	varchar(50)	YES		NULL	
colour	varchar(50)	YES		NULL	



Result Grid | Filter Rows: | Export: | Wrap

Field	Type	Null	Key	Default	Extra
sid	int	YES	MUL	NULL	
pid	int	YES	MUL	NULL	
cost	int	YES		NULL	

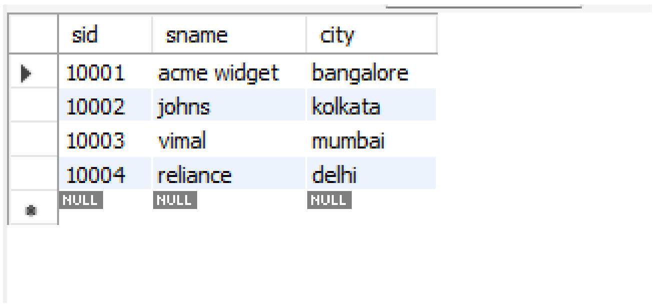
Inserting values

insert into supplier values(10001,'acme widget','bangalore'),(10002,'johns','kolkata'),(10003,'vimal','mumbai'),(10004,'reliance','delhi');

insert into parts values(20001,'book','red'),(20002,'pen','red'),(20003,'pencil','green'),(20004,'mobile','green'),(200 05,'charger','black');

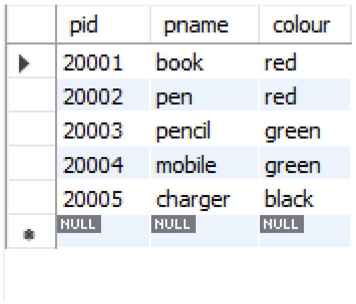
insert into catalog values(10001,20001,10),(10001,20002,10),(10001,20003,30),(10001,20004,10),(10001,20005, 10),(10002,20001,10),(10002,20002,20);
insert into catalog values(10003,20003,30),(10004,20003,40);

select * from supplier;



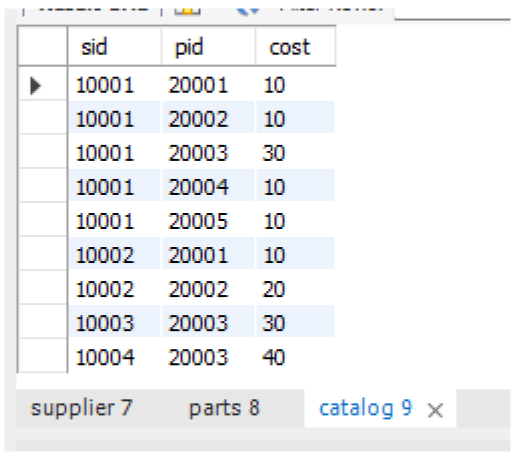
sid	sname	city
10001	acme widget	bangalore
10002	johns	kolkata
10003	vimal	mumbai
10004	reliance	delhi
NULL	NULL	NULL

select * from parts;



pid	pname	colour
20001	book	red
20002	pen	red
20003	pencil	green
20004	mobile	green
20005	charger	black
NULL	NULL	NULL

Select * from catalog;

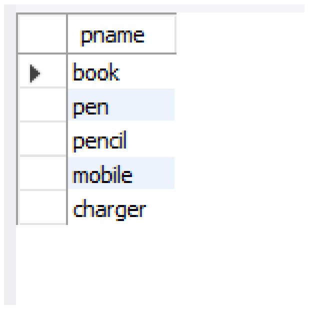


sid	pid	cost
10001	20001	10
10001	20002	10
10001	20003	30
10001	20004	10
10001	20005	10
10002	20001	10
10002	20002	20
10003	20003	30
10004	20003	40

Queries

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

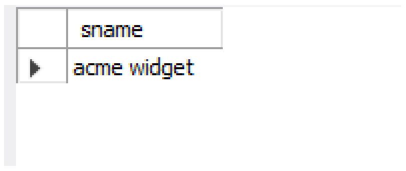
```
select pname from parts where pid in (select pid from catalog);
```



	pname
▶	book
	pen
	pencil
	mobile
	charger

2.Find the snames of suppliers who supply every part.

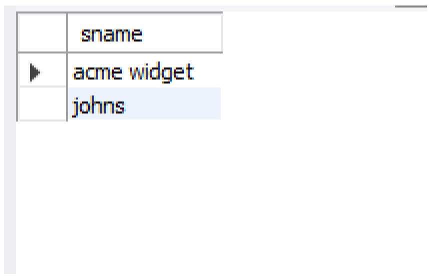
select sname from (select c.sname,count(distinct a.pid) as cnt from catalog a left join parts b on a.pid=b.pid left join supplier c on c.sid=a.sid group by 1) a where cnt=(select count(distinct a.pid) from catalog a left join parts b on a.pid=b.pid);



	sname
▶	acme widget

3.Find the snames of suppliers who supply every red part.

select sname from supplier where sid in(select sid from catalog where pid in(select pid from parts where colour='red'));

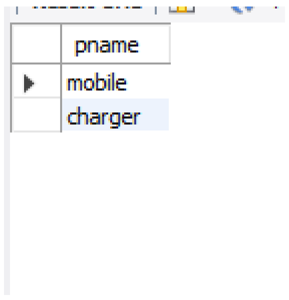


	sname
▶	acme widget
	johns

4. Find the pnames of parts supplied by Acme Widget Suppliers and by no one

else.

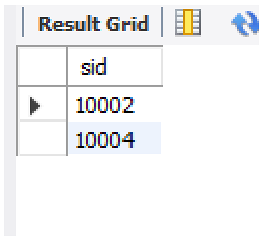
select pname from parts where pid in(select pid from catalog where sid in(select sid from supplier where sname='acme widget')) and pid not in(select pid from catalog where sid in(select sid from supplier where sname!='acme widget'));



	pname
▶	mobile
	charger

5.Find the sids of suppliers who charge more for some part than the average cost of that part

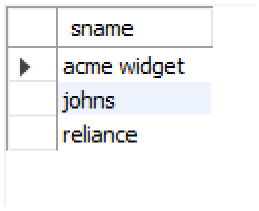
select c.sid from catalog c where c.cost >(select avg(cc.cost) from catalog cc where c.pid=cc.pid group by cc.pid);



	sid
▶	10002
	10004

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

```
select sname from supplier where sid in( select sid from catalog where cost
in( select max(cost) from catalog group by pid));
```



sname
acme widget
johns
reliance

8:Flight Database

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

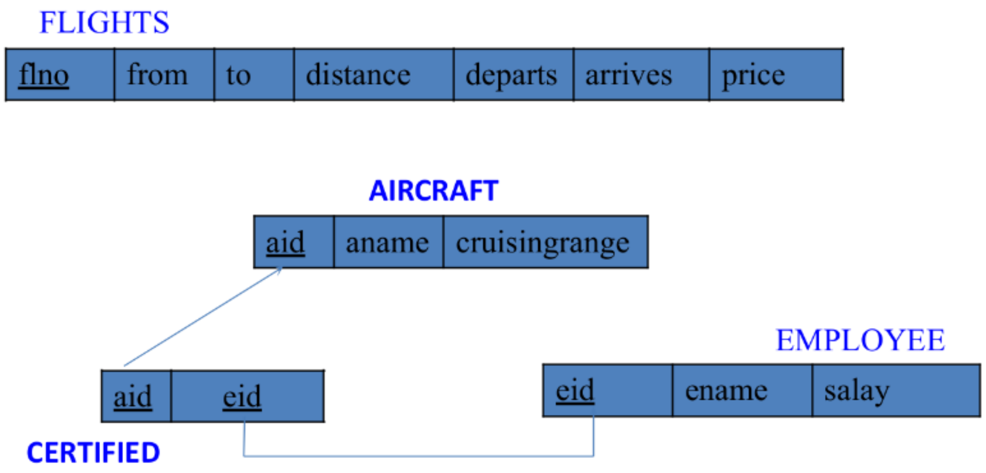
AIRCRAFT(aid: integer, aname: string, cruising_range: integer)

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.

Create database table and insert appropriate data



Creating Database

```
create database Airline;
use Airline;
```

Creating Tables

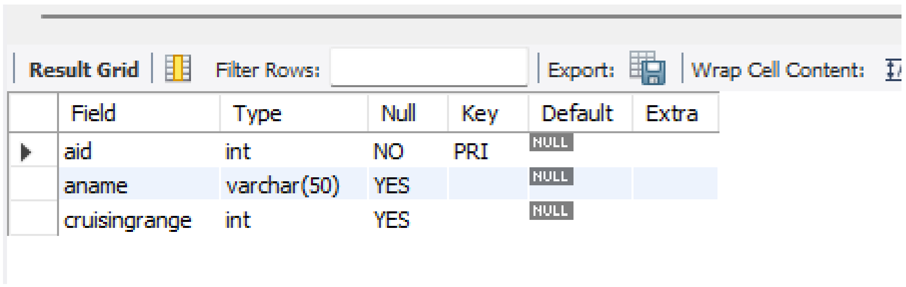
```
create table flights( flno int, ffrom varchar(50), tto varchar(50), distance int,
departs time, arrives time, price int, primary key(flno));
```

```
create table aircraft(
aid int, aname varchar(50), cruisingrange
int, primary key(aid));
```

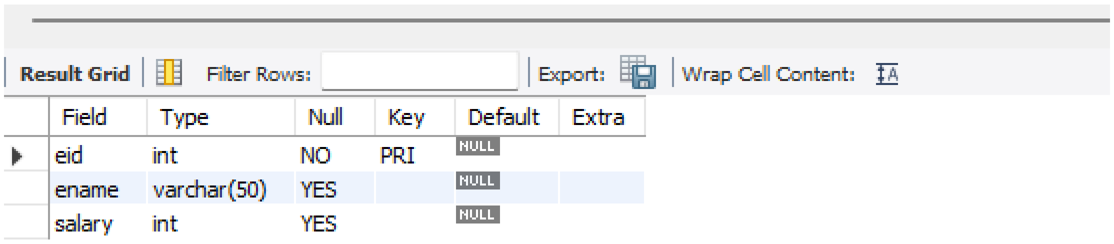
```
create table
certified( eid
int,aid int,
foreign key(aid) references
aircraft(aid) on update cascade
on delete cascade, foreign
key(eid) references
employee(eid) on update
cascade on delete cascade);
```

```
create table employee(
eid int, ename varchar(50), salary
int, primary key(eid));
```

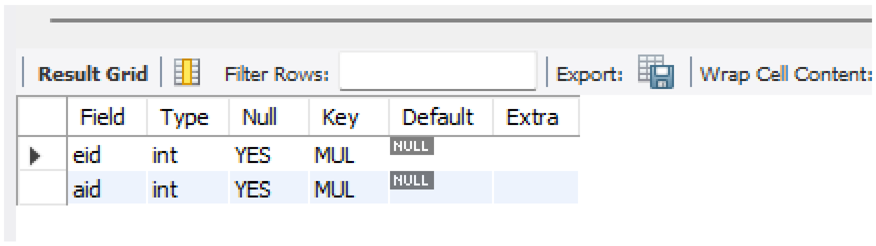
Structure of Tables



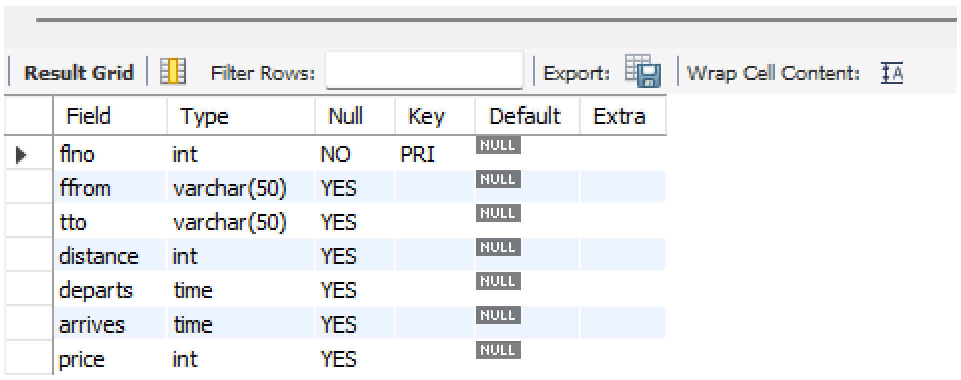
Field	Type	Null	Key	Default	Extra
aid	int	NO	PRI	NULL	
aname	varchar(50)	YES		NULL	
cruisingrange	int	YES		NULL	



Field	Type	Null	Key	Default	Extra
eid	int	NO	PRI	NULL	
ename	varchar(50)	YES		NULL	
salary	int	YES		NULL	



Field	Type	Null	Key	Default	Extra
eid	int	YES	MUL	NULL	
aid	int	YES	MUL	NULL	



Field	Type	Null	Key	Default	Extra
flno	int	NO	PRI	NULL	
ffrom	varchar(50)	YES		NULL	
tto	varchar(50)	YES		NULL	
distance	int	YES		NULL	
departs	time	YES		NULL	
arrives	time	YES		NULL	
price	int	YES		NULL	

Inserting Values

```
insert into employee values
(101,'Avinash',50000),
(102,'Lokesh',60000),
(103,'Rakesh',70000),
(104,'Santhosh',82000) ,
(105,'Tilak',5000);
```

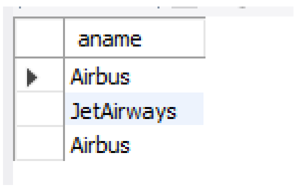
```
insert into aircraft values
(1,'Airbus',2000),
(2,'Boeing',700),
(3,'JetAirways',550),
(4,'Indigo',5000),
 (5,'Boeing',4500),
(6,'Airbus',2200);

insert into certified values
(101,2),
(101,4),
(101,5),
(101,6),
(102,1),
(102,3),
(102,5),
(103,2),
(103,3),
(103,5),
(103,6),
(10 4,6),
(104,1),
(104,3),
 (105,3);
```

```
insert into flights values
(1,'Banglore','New Delhi',500,'6:00','9:00',5000),
(2,'Banglore','Chennai',300,'7:00','8:30',3000),
(3,'Trivandrum','New
Delhi',800,'8:00','11:30',6000),(4,'Banglore','Frankfurt',10000,'6:00','23:30',50000
),
(5,'Kolkata','New Delhi',2400,'11:00','3:30',9000),
(6,'Banglore','Frankfurt',8000,'9:00','23:00',40000);
```

Queries

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

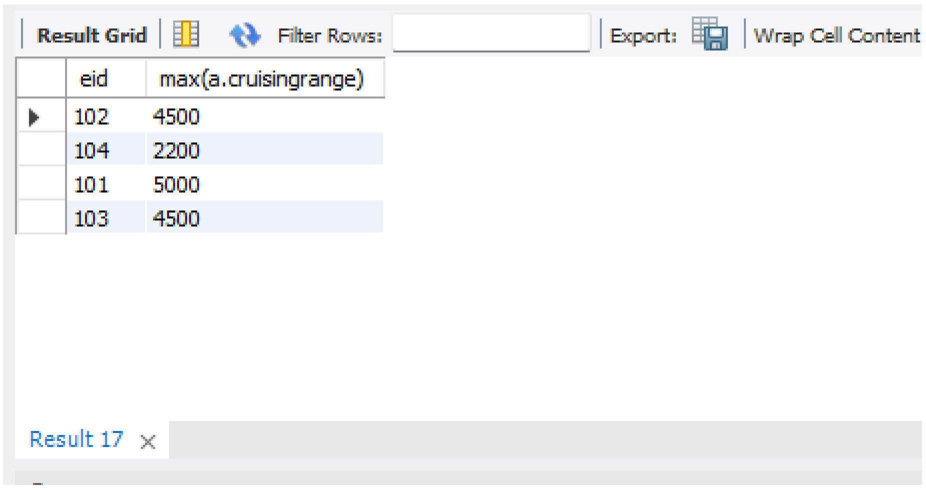
select a.aname from aircraft a where a.aid in(select c.aid from certified c where c.eid in(select e.eid from employee e where salary>80000));



	aname
▶	Airbus
	JetAirways
	Airbus

2.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.

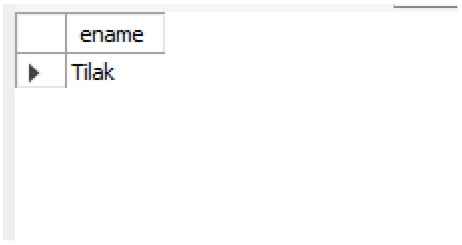
select c.eid,max(a.cruisingrange) from certified c,aircraft a where c.aid=a.aid group by c.eid having count(*)>=3;



	eid	max(a.cruisingrange)
▶	102	4500
	104	2200
	101	5000
	103	4500

3.Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.

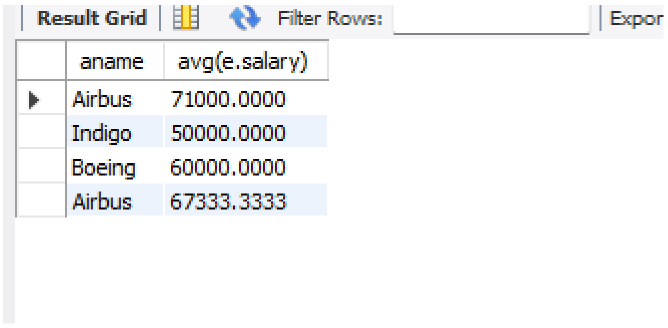
select e.ename from employee e where e.salary<(select min(f.price) from flights f where f.ffrom='Banglore' and f.tto='Frankfurt');



	ename
▶	Tilak

4.For all aircraft with cruising range over 1000 Kms, find the name of the aircraft and the Average salary of all pilots certified for this aircraft.

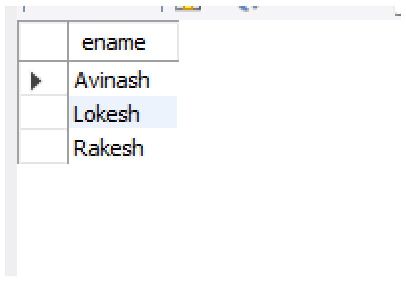
select a.aname,avg(e.salary) from aircraft a,employee e,certified c where a.aid=c.aid and e.eid=c.eid and a.cruisingrange>1000 group by c.aid;



	aname	avg(e.salary)
▶	Airbus	71000.0000
	Indigo	50000.0000
	Boeing	60000.0000
	Airbus	67333.3333

5.Find the names of pilots certified for some Boeing aircraft.

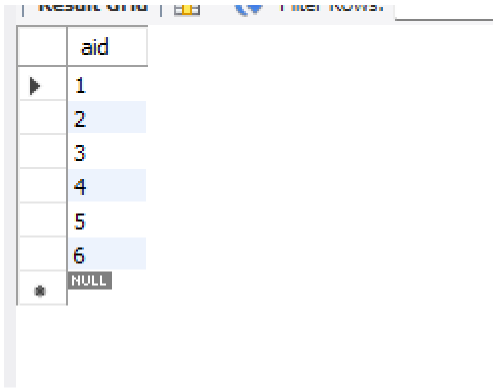
select e.ename from employee e where e.eid in (select c.eid from certified c where c.aid in (select a.aid from aircraft a where a.aname='Boeing'));



```
ename
Avinash
Lokesh
Rakesh
```

6.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 distance from flights where ffrom='Banglore' and tto='New Delhi');



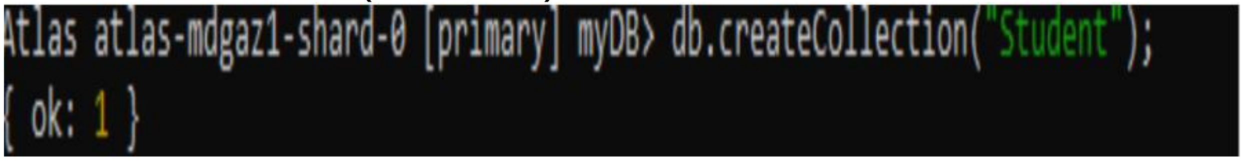
	aid
	1
	2
	3
	4
	5
	6
	NULL

LAB 9: NOSQL

Perform the following DB operations using MongoDB.

1. Create a database “Student” with the following attributes Rollno, Age, ContactNo, Email-Id.

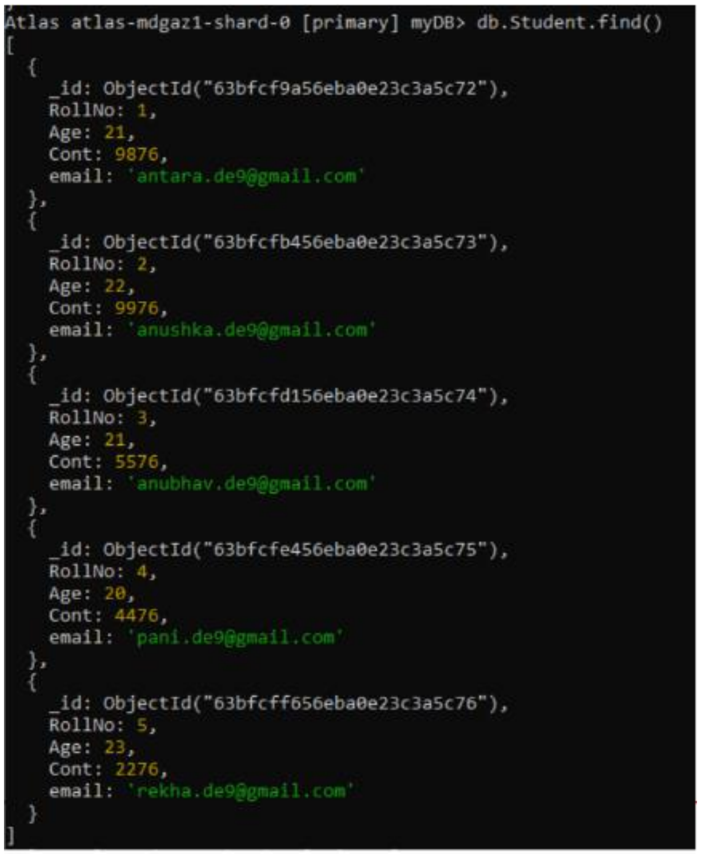
db.createCollection("Student");



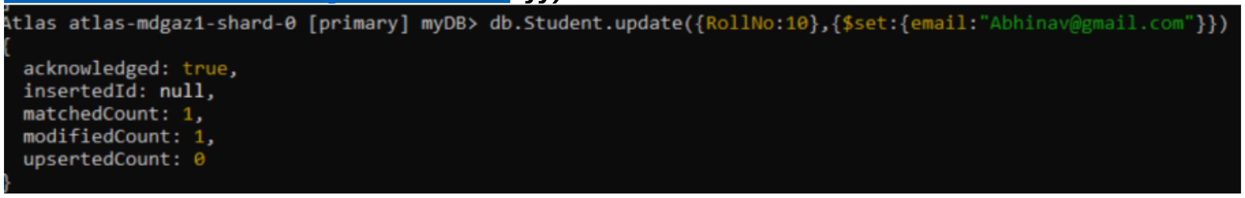
```
Atlas atlas-mdgaz1-shard-0 [primary] myDB> db.createCollection("Student");
{ ok: 1 }
```

2. Insert appropriate values

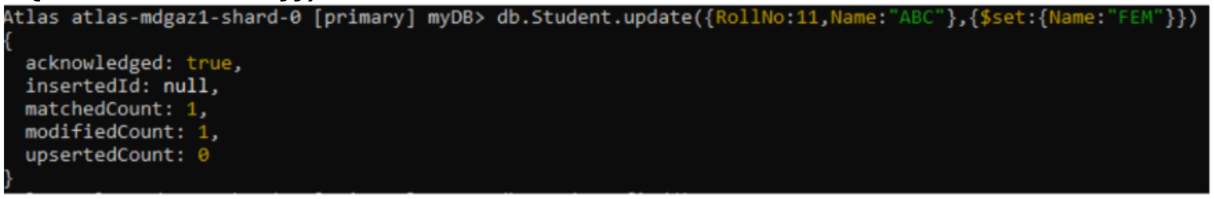
```
db.Student.insert({RollNo:1,Age:21,Cont:9876,email:"antara.de9@gmail.com"});
db.Student.insert({RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"});
db.Student.insert({RollNo:3,Age:21,Cont:5576,email:"anubhav.de9@gmail.com"});
db.Student.insert({RollNo:4,Age:20,Cont:4476,email:"pani.de9@gmail.com"});
db.Student.insert({RollNo:10,Age:23,Cont:2276,email:"rekha.de9@gmail.com"});
db.Student.find()
```


```
Atlas atlas-mdgaz1-shard-0 [primary] myDB> db.Student.find()
[
  {
    _id: ObjectId("63bfcf9a56eba0e23c3a5c72"),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId("63bfcfb456eba0e23c3a5c73"),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de9@gmail.com'
  },
  {
    _id: ObjectId("63bfcfd156eba0e23c3a5c74"),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
  },
  {
    _id: ObjectId("63bfcfe456eba0e23c3a5c75"),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId("63bfcff656eba0e23c3a5c76"),
    RollNo: 5,
    Age: 23,
    Cont: 2276,
    email: 'rekha.de9@gmail.com'
  }
]
```

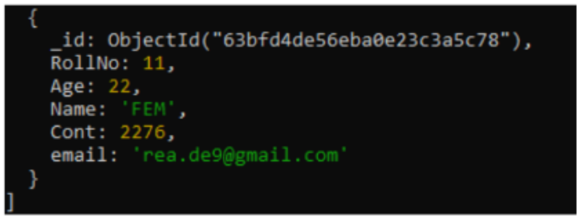
3. Write query to update Email-Id of a student with rollno 10.
db.Student.update({RollNo:10},{$set:{
email:"Abhinav@gmail.com"}})


```
Atlas atlas-mdgaz1-shard-0 [primary] myDB> db.Student.update({RollNo:10},{$set:{email:"Abhinav@gmail.com"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

4. Replace the student name from “ABC” to “FEM” of rollno 11.
db.Student.insert({RollNo:11,Age:22,Name:
"ABC",Cont:2276,email:"rea.de9@gmail.com"});
db.Student.update({RollNo:11,Name:"ABC"},{$se
t:{Name:"FEM"}})

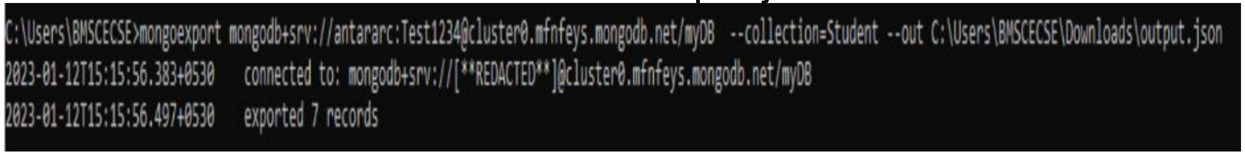


```
Atlas atlas-mdgaz1-shard-0 [primary] myDB> db.Student.update({RollNo:11,Name:"ABC"},{$set:{Name:"FEM"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```



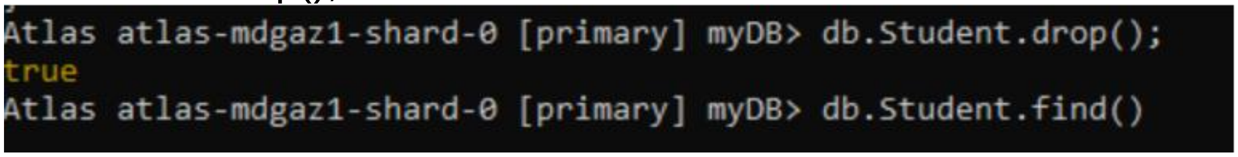
```
  {
    _id: ObjectId("63bfd4de56eba0e23c3a5c78"),
    RollNo: 11,
    Age: 22,
    Name: 'FEM',
    Cont: 2276,
    email: 'rea.de9@gmail.com'
  }
]
```

5. Export the created table into local file system
mongoexport
mongodb.net/myDB --collection=Student --out
C:\Users\BMSCECSE\Downloads\output.json



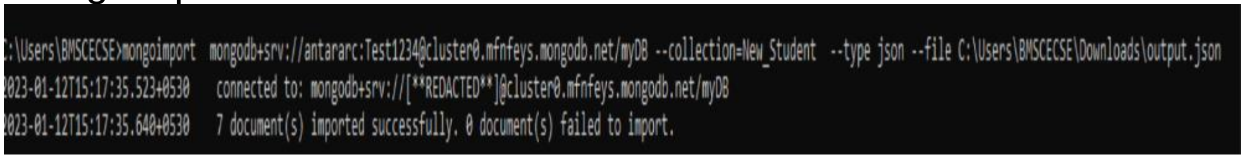
```
C:\Users\BMSCECSE>mongoexport mongodb+srv://antararc:Test1234@cluster0.mfnfeys.mongodb.net/myDB  --collection=Student --out C:\Users\BMSCECSE\Downloads\output.json
2023-01-12T15:15:56.383+0530    connected to: mongodb+srv://[**REDACTED**]@cluster0.mfnfeys.mongodb.net/myDB
2023-01-12T15:15:56.497+0530    exported 7 records
```

6. Drop the table
db.Student.drop();



```
Atlas atlas-mdgaz1-shard-0 [primary] myDB> db.Student.drop();
true
Atlas atlas-mdgaz1-shard-0 [primary] myDB> db.Student.find()
```

7. Import a given csv dataset from local file system into mongodb collection.
mongoimport

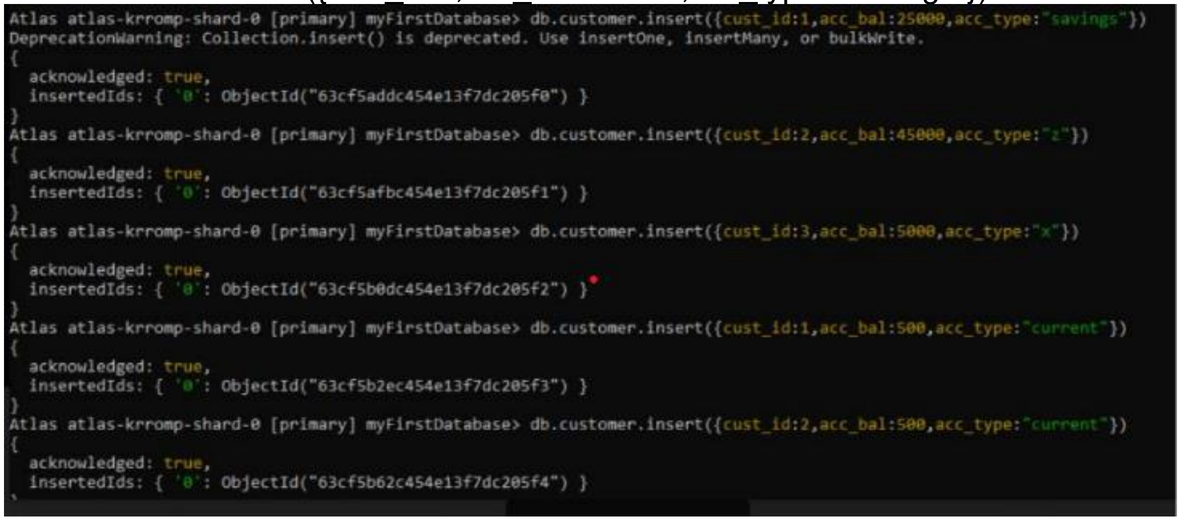


```
:\Users\BMSCECSE>mongoimport  mongodb+srv://antararc:Test1234@cluster0.mfnfeys.mongodb.net/myDB --collection=New_Student  --type json --file C:\Users\BMSCECSE\Downloads\output.json
023-01-12T15:17:35.523+0530    connected to: mongodb+srv://[**REDACTED**]@cluster0.mfnfeys.mongodb.net/myDB
023-01-12T15:17:35.640+0530    7 document(s) imported successfully. 0 document(s) failed to import.
```

LAB 10:NO SQL

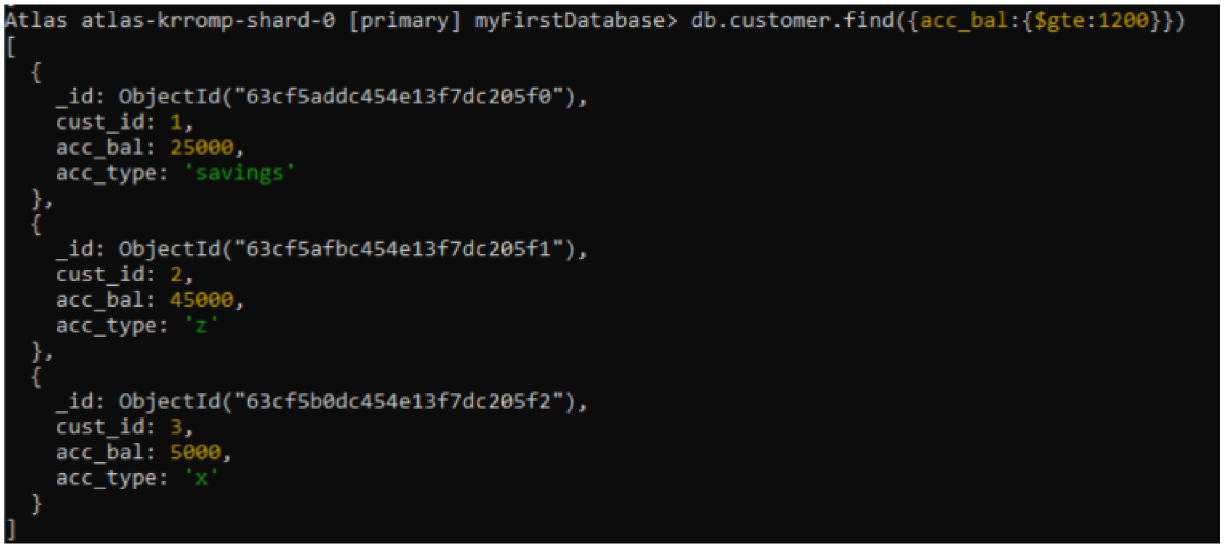
1. Create a collection by name Customers with the following attributes.
Cust_id, Acc_Bal, Acc_Type
>>db.createCollection("customer")

2. Insert at least 5 values into the table
>>db.customer.insert({cust_id:1,acc_bal=12000,acc_type="savings"})
>>db.customer.insert({cust_id:2,acc_bal=500,acc_type="z"})
>>db.customer.insert({cust_id:3,acc_bal=12000,acc_type="z"})
>>db.customer.insert({cust_id:1,acc_bal=12000,acc_type="current"})
>>db.customer.insert({cust_id:2,acc_bal=12000,acc_type="x"})
>>db.customer.insert({cust_id:3,acc_bal=12000,acc_type="savings"})



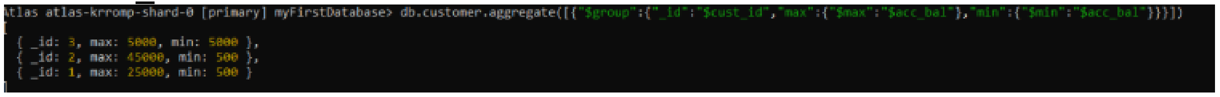
```
Atlas atlas-krromp-shard-0 [primary] myFirstDatabase> db.customer.insert({cust_id:1,acc_bal:25000,acc_type:"savings"})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63cf5addc454e13f7dc205f0") }
}
Atlas atlas-krromp-shard-0 [primary] myFirstDatabase> db.customer.insert({cust_id:2,acc_bal:45000,acc_type:"z"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63cf5afbc454e13f7dc205f1") }
}
Atlas atlas-krromp-shard-0 [primary] myFirstDatabase> db.customer.insert({cust_id:3,acc_bal:5000,acc_type:"x"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63cf5b0dc454e13f7dc205f2") }
}
Atlas atlas-krromp-shard-0 [primary] myFirstDatabase> db.customer.insert({cust_id:1,acc_bal:500,acc_type:"current"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63cf5b2ec454e13f7dc205f3") }
}
Atlas atlas-krromp-shard-0 [primary] myFirstDatabase> db.customer.insert({cust_id:2,acc_bal:500,acc_type:"current"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63cf5b62c454e13f7dc205f4") }
```

3. Write a query to display those records whose total account balance
is greater than 1200 of account type 'Z' for each customer_id.
>>db.customer.find({acc_bal:{$gte:1200}})



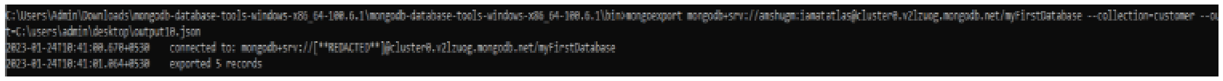
```
Atlas atlas-krromp-shard-0 [primary] myFirstDatabase> db.customer.find({acc_bal:{$gte:1200}})
[
  {
    _id: ObjectId("63cf5addc454e13f7dc205f0"),
    cust_id: 1,
    acc_bal: 25000,
    acc_type: 'savings'
  },
  {
    _id: ObjectId("63cf5afbc454e13f7dc205f1"),
    cust_id: 2,
    acc_bal: 45000,
    acc_type: 'z'
  },
  {
    _id: ObjectId("63cf5b0dc454e13f7dc205f2"),
    cust_id: 3,
    acc_bal: 5000,
    acc_type: 'x'
  }
]
```

4. Determine Minimum and Maximum account balance for each customer_id.



```
tlas atlas-krromp-shard-0 [primary] myFirstDatabase> db.customer.aggregate([{"$group":{"_id":"$cust_id","max":{"$max":"$acc_bal"},"min":{"$min":"$acc_bal"}}}])
[
  { _id: 3, max: 5000, min: 5000 },
  { _id: 2, max: 45000, min: 500 },
  { _id: 1, max: 25000, min: 500 }
]
```

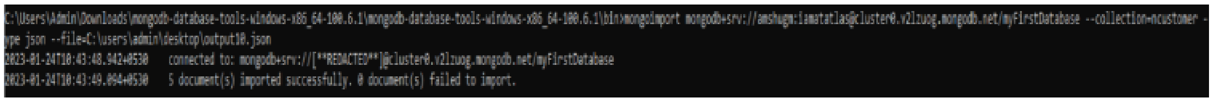
5. Export the created collection into local file system



```
C:\Users\Admin\Downloads\mongodb-database-tools-windows-x86_64-100.6.1\mongodb-database-tools-windows-x86_64-100.6.1\bin>mongoexport mongodb+srv://amshugm:iamatatlas@cluster0.v2lzuog.mongodb.net/myFirstDatabase --collection=customer --ou
t=C:\users\admin\desktop\output10.json
2023-01-24T10:41:00.670+0530    connected to: mongodb+srv://[**REDACTED**]@cluster0.v2lzuog.mongodb.net/myFirstDatabase
2023-01-24T10:41:01.064+0530    exported 5 records
```

6. Drop the table
>>db.customer.drop()

7. Import a given csv dataset from local file system into mongodb collection.



```
C:\Users\Admin\Downloads\mongodb-database-tools-windows-x86_64-100.6.1\mongodb-database-tools-windows-x86_64-100.6.1\bin>mongoimport mongodb+srv://amshugm:iamatatlas@cluster0.v2lzuog.mongodb.net/myFirstDatabase --collection=ncustomer -
ype json --file=C:\users\admin\desktop\output10.json
2023-01-24T10:43:48.942+0530    connected to: mongodb+srv://[**REDACTED**]@cluster0.v2lzuog.mongodb.net/myFirstDatabase
2023-01-24T10:43:49.094+0530    5 document(s) imported successfully. 0 document(s) failed to import.
```