VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT

on

DATABASE MANAGEMENT SYSTEM

Submitted by

NAME: NEHA BHASKAR KAMATH

USN: 1BM21CS113

in partial fulfilment 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

Oct 2022-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 SYSTEM**" carried out by **NEHA BHASKAR KAMATH(1BM21CS113)** , who is bonafide student of B. M. S. College of Engineering. It is in partial fulfilment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the year 2022-23.

The Lab report has been approved as it satisfies the academic requirements in respect of **Database Management System- (22CS3PCDBM)** work prescribed for the said degree.

Dr. Manjunath D R	Dr. Jyothi Nayak
Assistant professor	Professor and Head
Department of CSE	Department of CSE
BMSCE, Bengaluru	BMSCE, Bengaluru

TABLE OF CONTENTS

EXPERIMENT 1- INSURANCE DATABASE

WEEK 1

Question

Consider the Insurance database given below. The data types are specified.

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

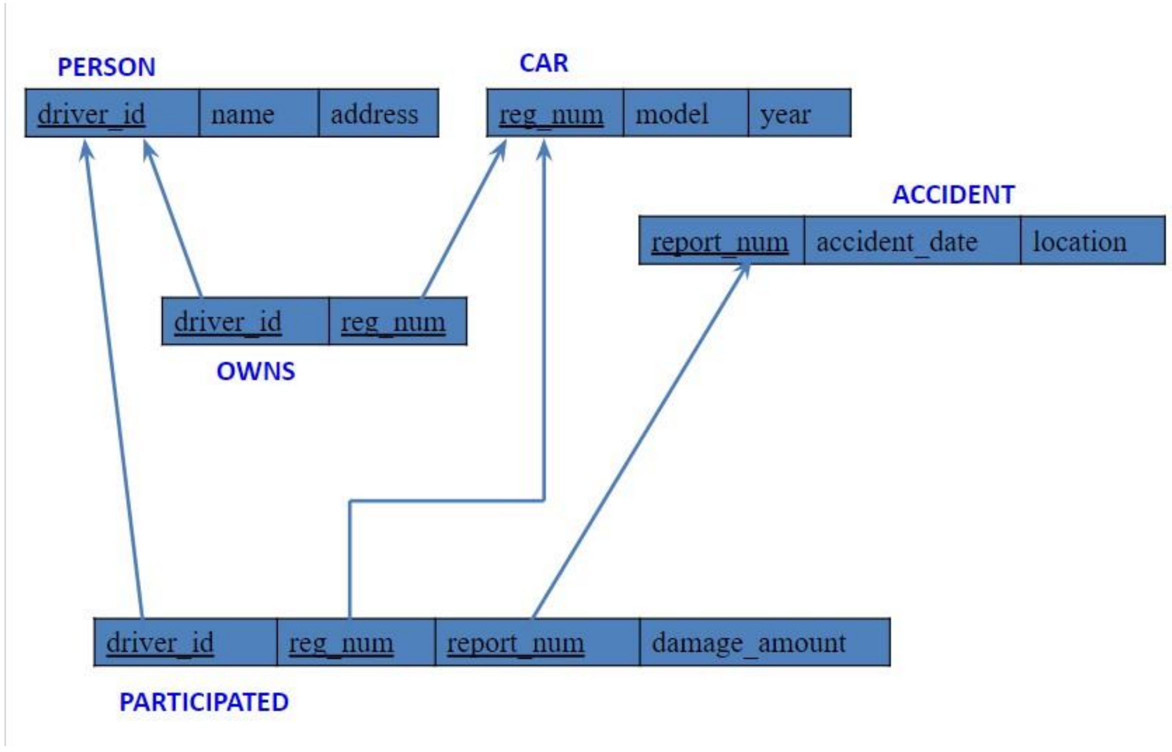
CAR (reg_num: String, model: String, year: int)

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)

Schema diagram



Creating database

create database 1BM21CS113_Insurance;

Using database

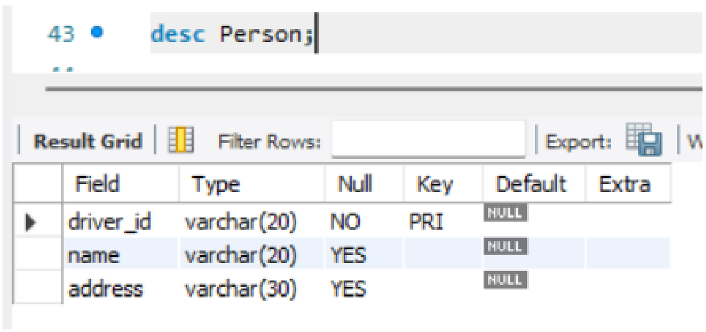
use 1BM21CS113_Insurance;

Creating tables

```
create table Person

(

driver_id varchar (20),

name varchar (20),

address varchar (30)

);


alter table Person

add primary key(driver_id);


create table Car

(

reg_num varchar (15) primary key,

model varchar (20),

year int

);


create table owns

(

driver_id varchar (20),

reg_num varchar (15),

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 varchar (20) primary key,
accident_date date,
location varchar (30)
);

create table participated
(
driver_id varchar (20),
reg_num varchar (15),
report_num varchar (20),
primary key (driver_id, reg_num, report_num),
damage_amount double,
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 the table

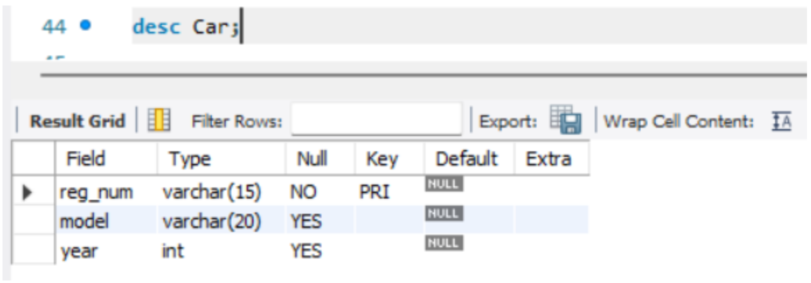
desc Person;



```
desc Person;
```

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

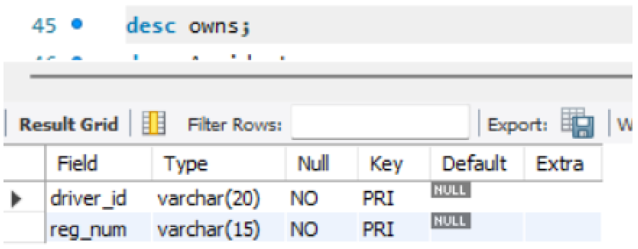
desc Car;



```
desc Car;
```

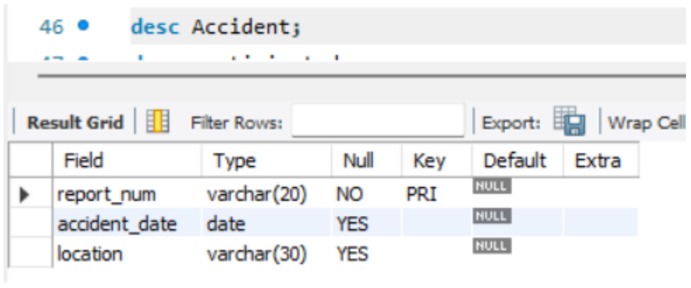
Field	Type	Null	Key	Default	Extra
reg_num	varchar(15)	NO	PRI	NULL	
model	varchar(20)	YES		NULL	
year	int	YES		NULL	

desc owns;


```
desc owns;
```

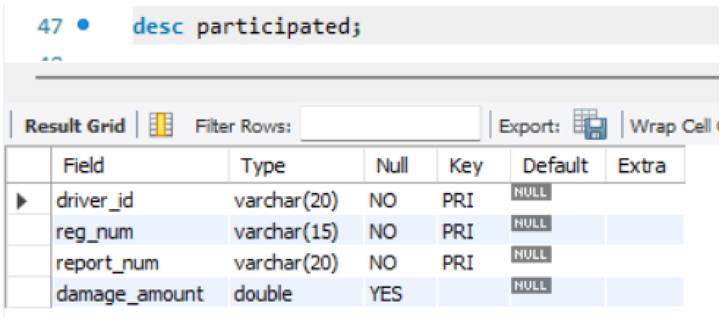
Field	Type	Null	Key	Default	Extra
driver_id	varchar(20)	NO	PRI	NULL	
reg_num	varchar(15)	NO	PRI	NULL	

desc Accident;


```
desc Accident;
```

Field	Type	Null	Key	Default	Extra
report_num	varchar(20)	NO	PRI	NULL	
accident_date	date	YES		NULL	
location	varchar(30)	YES		NULL	

desc participated;


```
desc participated;
```

Field	Type	Null	Key	Default	Extra
driver_id	varchar(20)	NO	PRI	NULL	
reg_num	varchar(15)	NO	PRI	NULL	
report_num	varchar(20)	NO	PRI	NULL	
damage_amount	double	YES		NULL	

Inserting values into the table

insert into Person values ('A01', 'Ramesh','RR nagar');

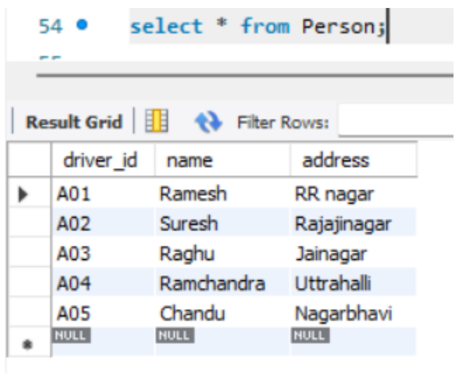
insert into Person values ('A02', 'Suresh', 'Rajajinagar');

insert into Person values ('A03', 'Raghu', 'Jainagar');

insert into Person values ('A04', 'Ramchandra', 'Uttrahalli');

insert into Person values ('A05','Chandu','Nagarbhavi');

select * from Person;



```
select * from Person;
```

driver_id	name	address
A01	Ramesh	RR nagar
A02	Suresh	Rajajinagar
A03	Raghu	Jainagar
A04	Ramchandra	Uttrahalli
A05	Chandu	Nagarbhavi
NULL	NULL	NULL

insert into Car values('KA04MB2345','Q1',2013);

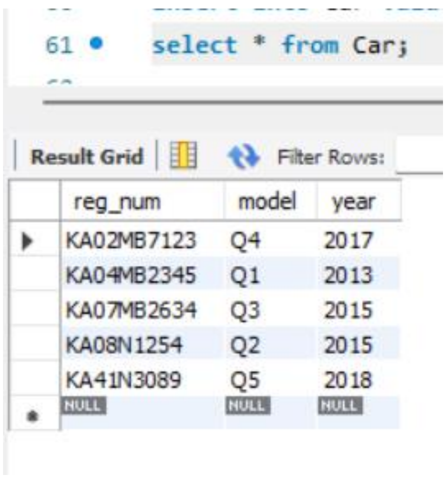
insert into Car values('KA08N1254','Q2',2015);

insert into Car values('KA07MB2634','Q3',2015);

insert into Car values('KA02MB7123','Q4',2017);

insert into Car values('KA41N3089','Q5',2018);

select * from Car;



```
select * from Car;
```

reg_num	model	year
KA02MB7123	Q4	2017
KA04MB2345	Q1	2013
KA07MB2634	Q3	2015
KA08N1254	Q2	2015
KA41N3089	Q5	2018
NULL	NULL	NULL

insert into owns values('A01','KA04MB2345');

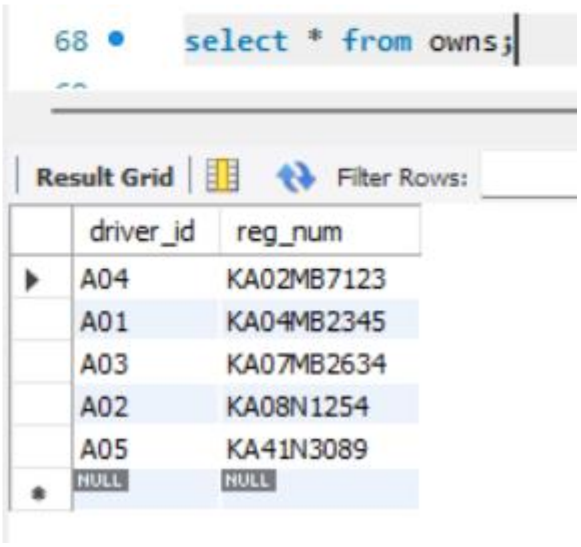
insert into owns values('A02','KA08N1254');

insert into owns values('A03','KA07MB2634');

insert into owns values('A04','KA02MB7123');

insert into owns values('A05','KA41N3089');

select * from owns;



```
select * from owns;
```

driver_id	reg_num
A04	KA02MB7123
A01	KA04MB2345
A03	KA07MB2634
A02	KA08N1254
A05	KA41N3089
NULL	NULL

insert into Accident values('01', '2022-01-24', 'Banglore');

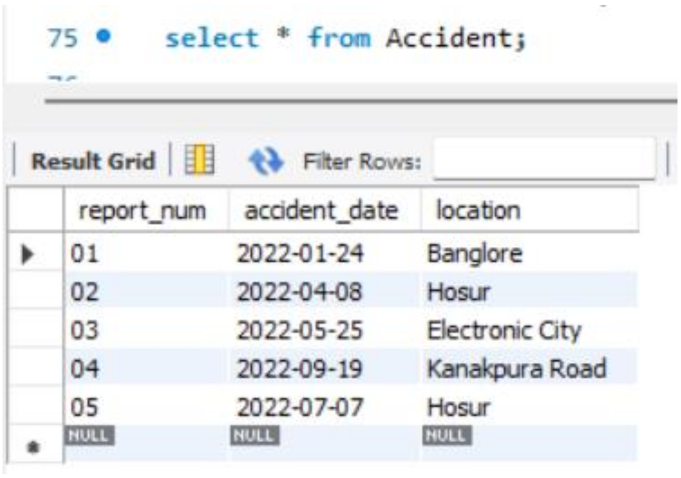
insert into Accident values('02', '2022-04-8', 'Hosur');

insert into Accident values('03', '2022-05-25', 'Electronic City');

insert into Accident values('04', '2022-09-19', 'Kanakpura Road');

insert into Accident values('05', '2022-07-07', 'Hosur');

select * from Accident;



```
select * from Accident;
```

report_num	accident_date	location
01	2022-01-24	Banglore
02	2022-04-08	Hosur
03	2022-05-25	Electronic City
04	2022-09-19	Kanakpura Road
05	2022-07-07	Hosur
NULL	NULL	NULL

insert into participated values('A01','KA04MB2345','01',25000);

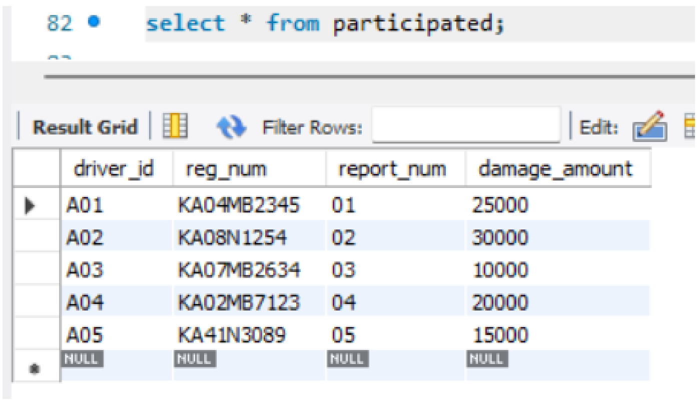
insert into participated values('A02','KA08N1254','02',30000);

insert into participated values('A03','KA07MB2634','03',10000);

insert into participated values('A04','KA02MB7123','04',20000);

insert into participated values('A05','KA41N3089','05',15000);

select * from participated;



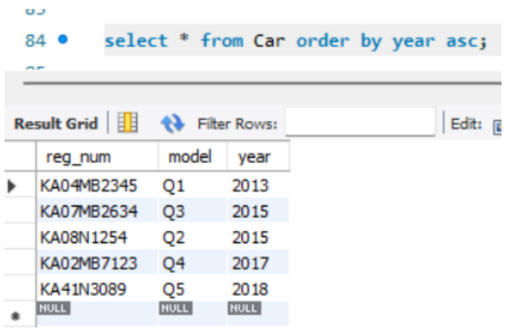
```
select * from participated;
```

driver_id	reg_num	report_num	damage_amount
A01	KA04MB2345	01	25000
A02	KA08N1254	02	30000
A03	KA07MB2634	03	10000
A04	KA02MB7123	04	20000
A05	KA41N3089	05	15000
NULL	NULL	NULL	NULL

Queries

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

select * from Car order by year asc;



```
select * from Car order by year asc;
```

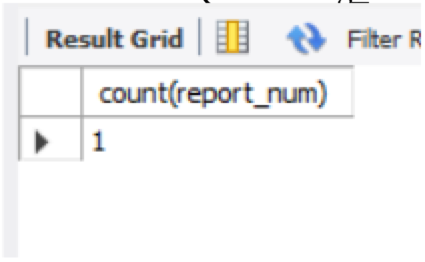
reg_num	model	year
KA04MB2345	Q1	2013
KA07MB2634	Q3	2015
KA08N1254	Q2	2015
KA02MB7123	Q4	2017
KA41N3089	Q5	2018
NULL	NULL	NULL

2.Find the number of accidents in which cars belonging to a specific model (example 'Q1') were involved.

select count(report_num)

from Car c, participated p

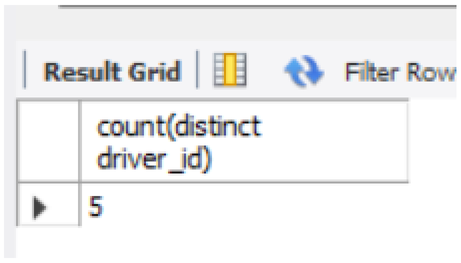
where c.model='Q1'and c.reg_num=p.reg_num;



count(report_num)
1

3. Find the total number of people who owned cars that involved in accidents in 2022.

```
select count(distinct driver_id)
from participated p, accident a
where p.report_num=a.report_num and a.accident_date like '__22%';
```

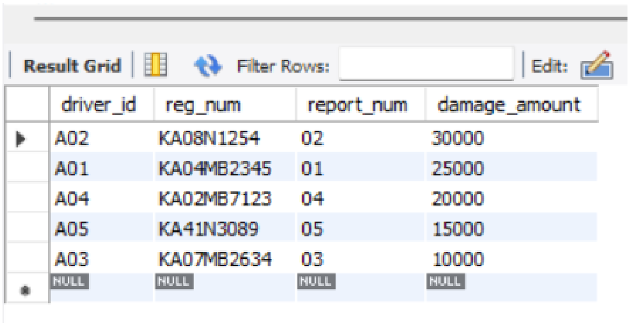


```
count(distinct
driver_id)
5
```

WEEK 2- Queries on Insurance database

1. List the entire participated relation in the descending order of damaged amount.

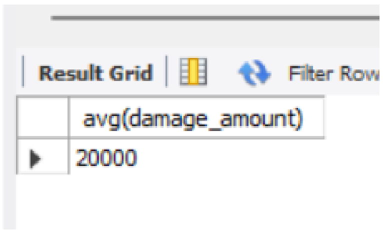
select * from participated order by damage_amount desc;



driver_id	reg_num	report_num	damage_amount
A02	KA08N1254	02	30000
A01	KA04MB2345	01	25000
A04	KA02MB7123	04	20000
A05	KA41N3089	05	15000
A03	KA07MB2634	03	10000
NULL	NULL	NULL	NULL

2. Find the average damaged amount.

select avg(damage_amount) from participated;



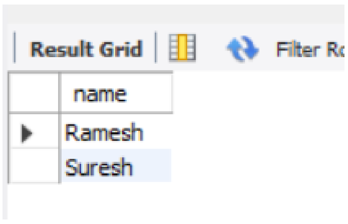
avg(damage_amount)
20000

3. List the names of drivers whose damage is greater than the average damage amount.

select p.name

from Person p, participated par

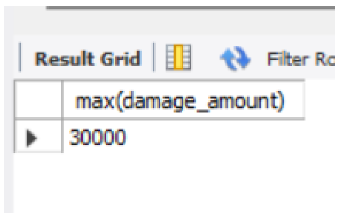
where p.driver_id=par.driver_id and par.damage_amount>(select avg(damage_amount) from participated);



name
Ramesh
Suresh

4. Find the maximum damage amount.

select max(damage_amount) from participated;



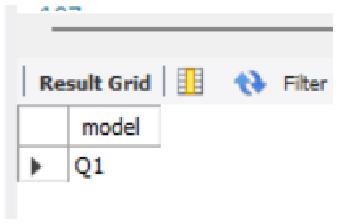
```
Result Grid | Filter Ro
max(damage_amount)
30000
```

5. Find the car model which participated in accident at location Banglore.

select c.model

from Accident a, participated p, Car c

where p.report_num=a.report_num and p.reg_num=c.reg_num and a.location='Banglore';



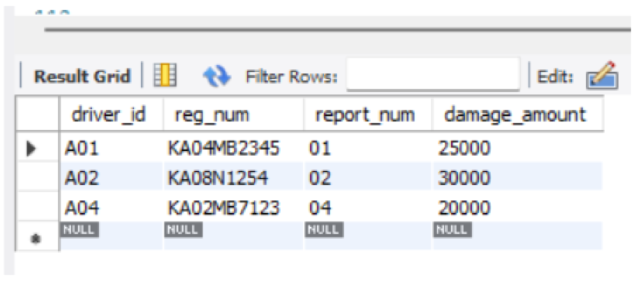
```
Result Grid | Filter
model
Q1
```

6. Delete the tuple whose damage amount is below the average damage amount.

delete from participated

where damage_amount<(select t.amt from (select avg(damage_amount) as amt from participated) t);

select * from participated;



```
Result Grid | Filter Rows:              | Edit:
driver_id  reg_num      report_num  damage_amount
A01        KA04MB2345   01          25000
A02        KA08N1254    02          30000
A04        KA02MB7123   04          20000
NULL       NULL         NULL        NULL
```

EXPERIMENT 2- BANK DATABASE

WEEK 3

Question

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

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

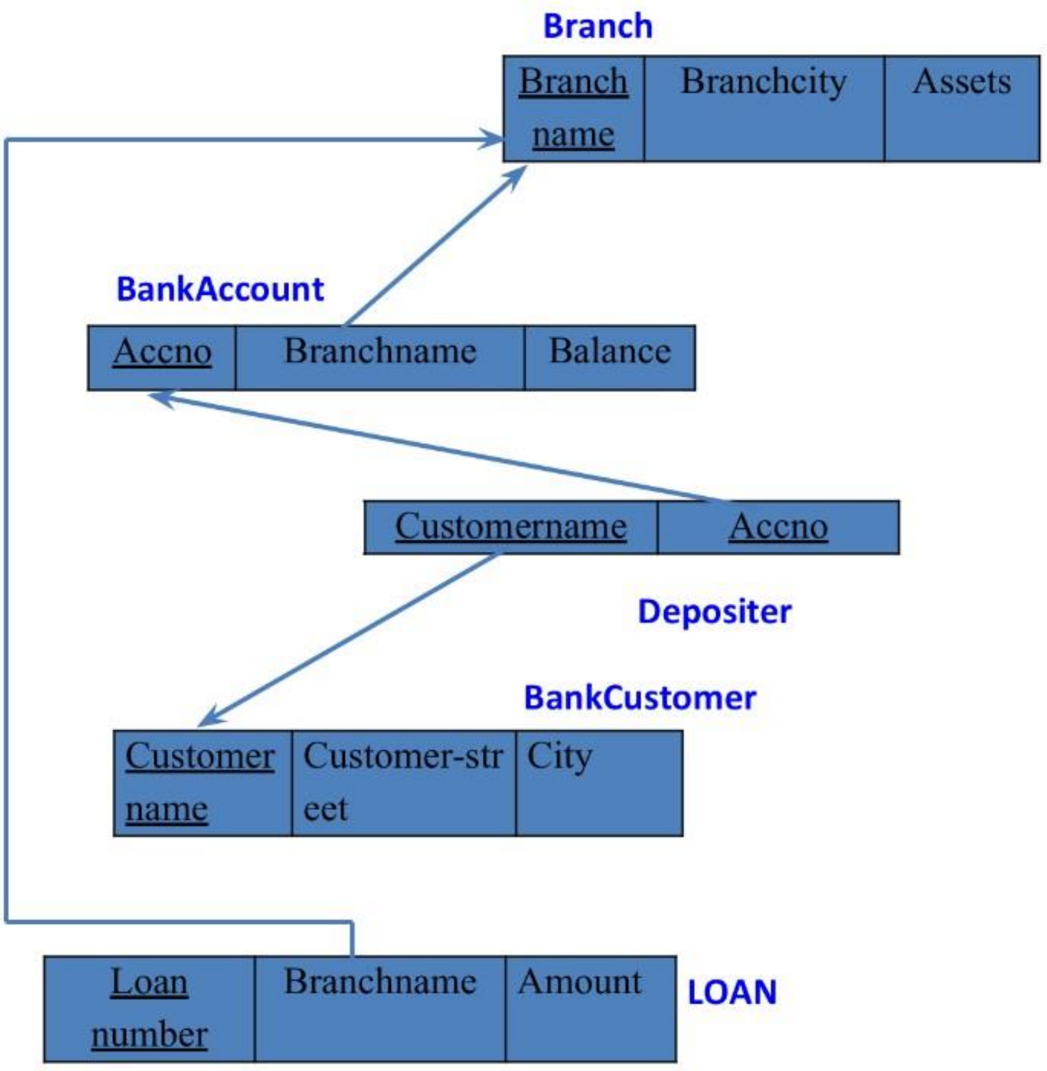
BankCustomer (customer-name: String, customer-street: String,

customer-city: String)

Depositor(customer-name: String, accno: int)

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

Schema diagram



Creating database

```
create database 1bm21cs113_bank;
```

Using database

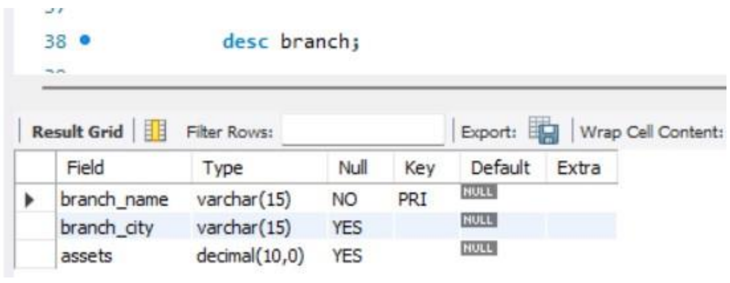
```
use 1bm21cs113_bank;
```

Creating tables

```
create table branch
(
  branch_name varchar(15) primary key,
  branch_city varchar(15),
  assets decimal
);

  create table BankAccount
  (
    accno int primary key,
    branch_name varchar(15) not null,
    foreign key(branch_name) references branch(branch_name),
    balance decimal
);

create table BankCustomer
(
customer_name varchar(25) primary key,
 customer_street varchar(20),
 city varchar(20)
);

create table Depositer
(
 customer_name varchar(25),
 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_no int primary key,
     branch_name varchar(15),
     foreign key(branch_name) references branch(branch_name),
     amount decimal
);
```

Structure of the tables

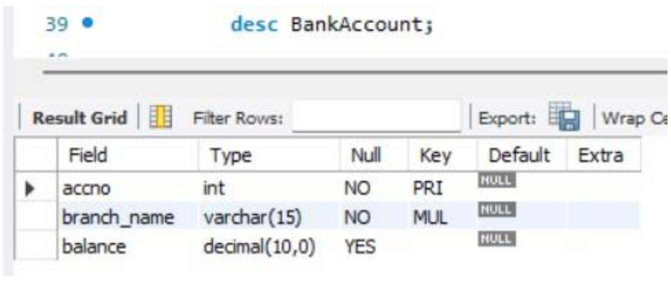
desc branch;



```
desc branch;
```

Field	Type	Null	Key	Default	Extra
branch_name	varchar(15)	NO	PRI	NULL	
branch_city	varchar(15)	YES		NULL	
assets	decimal(10,0)	YES		NULL	

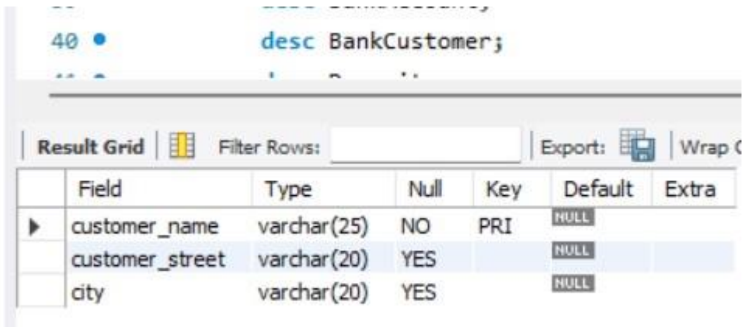
desc BankAccount;



```
desc BankAccount;
```

Field	Type	Null	Key	Default	Extra
accno	int	NO	PRI	NULL	
branch_name	varchar(15)	NO	MUL	NULL	
balance	decimal(10,0)	YES		NULL	

desc BankCustomer;

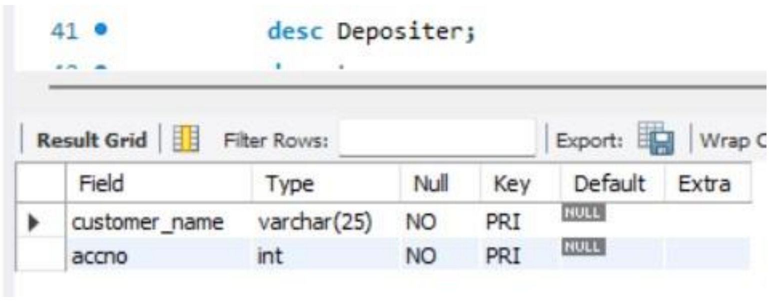


```
desc BankCustomer;
```

Result Grid | Filter Rows: | Export: | Wrap C

	Field	Type	Null	Key	Default	Extra
▶	customer_name	varchar(25)	NO	PRI	NULL	
	customer_street	varchar(20)	YES		NULL	
	city	varchar(20)	YES		NULL	

desc Depositer;

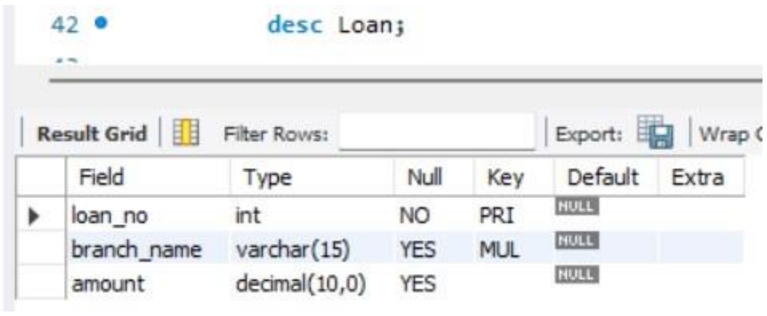


```
desc Depositer;
```

Result Grid | Filter Rows: | Export: | Wrap C

	Field	Type	Null	Key	Default	Extra
▶	customer_name	varchar(25)	NO	PRI	NULL	
	accno	int	NO	PRI	NULL	

desc Loan;



```
desc Loan;
```

Result Grid | Filter Rows: | Export: | Wrap C

	Field	Type	Null	Key	Default	Extra
▶	loan_no	int	NO	PRI	NULL	
	branch_name	varchar(15)	YES	MUL	NULL	
	amount	decimal(10,0)	YES		NULL	

Inserting values into the tables

insert into branch values("hdfc-Chamrajpet","Banglore",200000);

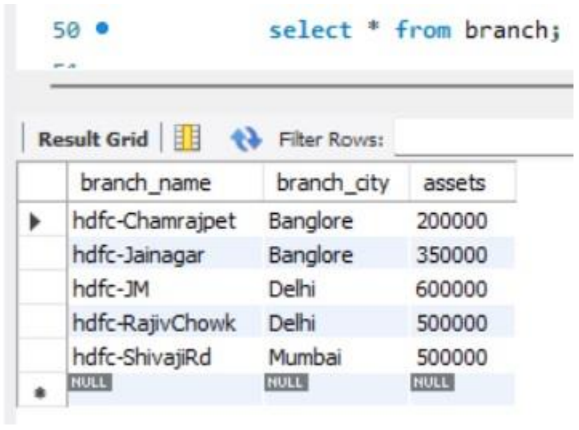
insert into branch values("hdfc-ShivajiRd","Mumbai",500000);

insert into branch values("hdfc-Jainagar","Banglore",350000);

insert into branch values("hdfc-RajivChowk","Delhi",500000);

insert into branch values("hdfc-JM","Delhi",600000);

select * from branch;



```
select * from branch;
```

	branch_name	branch_city	assets
▶	hdfc-Chamrajpet	Banglore	200000
	hdfc-Jainagar	Banglore	350000
	hdfc-JM	Delhi	600000
	hdfc-RajivChowk	Delhi	500000
	hdfc-ShivajiRd	Mumbai	500000
*	NULL	NULL	NULL

insert into BankAccount values(1,"hdfc-Chamrajpet",30000);

insert into BankAccount values(2,"hdfc-ShivajiRd",15000);

insert into BankAccount values(3,"hdfc-JM",20000);

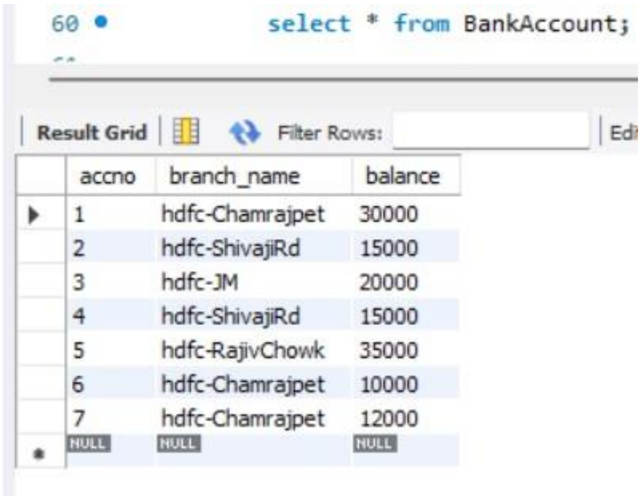
insert into BankAccount values(4,"hdfc-ShivajiRd",15000);

insert into BankAccount values(5,"hdfc-RajivChowk",35000);

insert into BankAccount values(6,"hdfc-Chamrajpet",10000);

insert into BankAccount values(7,"hdfc-Chamrajpet",12000);

select * from BankAccount;



```
select * from BankAccount;
```

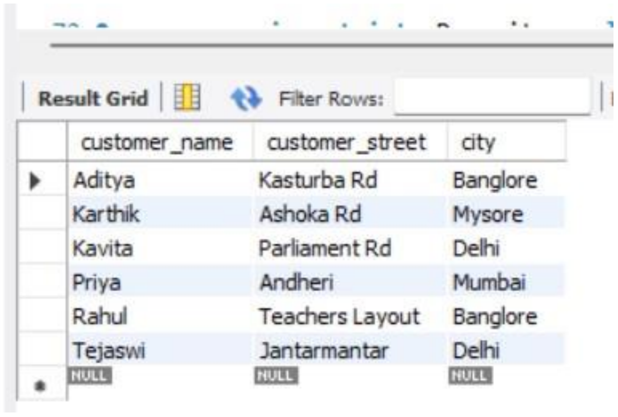
	accno	branch_name	balance
▶	1	hdfc-Chamrajpet	30000
	2	hdfc-ShivajiRd	15000
	3	hdfc-JM	20000
	4	hdfc-ShivajiRd	15000
	5	hdfc-RajivChowk	35000
	6	hdfc-Chamrajpet	10000
	7	hdfc-Chamrajpet	12000
*	NULL	NULL	NULL

insert into BankCustomer values("Rahul","Teachers Layout","Banglore");

insert into BankCustomer values("Aditya","Kasturba Rd","Banglore");

insert into BankCustomer values("Kavita","Parliament Rd","Delhi");

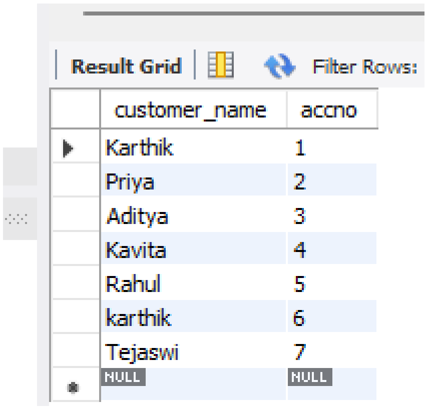
```
insert into BankCustomer values("Karthik","Ashoka Rd","Mysore");
insert into BankCustomer values("Tejaswi","Jantarmantar","Delhi");
insert into BankCustomer values("Priya","Andheri","Mumbai");

select * from BankCustomer;
```



Result Grid | Filter Rows:

customer_name	customer_street	city
Aditya	Kasturba Rd	Banglore
Karthik	Ashoka Rd	Mysore
Kavita	Parliament Rd	Delhi
Priya	Andheri	Mumbai
Rahul	Teachers Layout	Banglore
Tejaswi	Jantarmantar	Delhi
NULL	NULL	NULL

```
insert into Depositer values("Rahul",5);
insert into Depositer values("Aditya",3);
insert into Depositer values("Tejaswi",7);
insert into Depositer values("Kavita",4);
insert into Depositer values("karthik",6);
insert into Depositer values("Karthik",1);
insert into Depositer values("Priya",2);

select * from Depositer;
```



Result Grid | Filter Rows:

customer_name	accno
Karthik	1
Priya	2
Aditya	3
Kavita	4
Rahul	5
karthik	6
Tejaswi	7
NULL	NULL

insert into Loan values(1,"hdfc-Chamrajpet",9000);

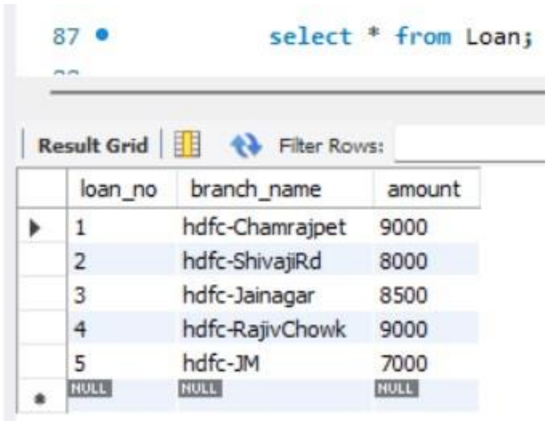
insert into Loan values(2,"hdfc-ShivajiRd",8000);

insert into Loan values(3,"hdfc-Jainagar",8500);

insert into Loan values(4,"hdfc-RajivChowk",9000);

insert into Loan values(5,"hdfc-JM",7000);

select * from Loan;



```
select * from Loan;
```

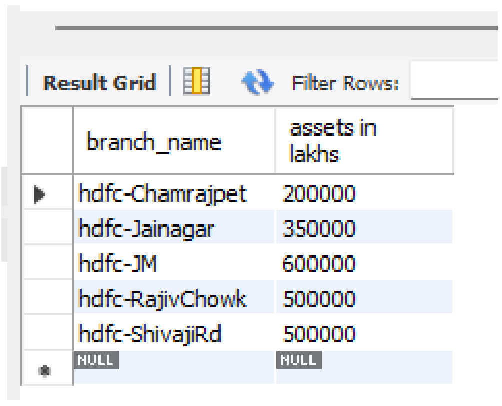
loan_no	branch_name	amount
1	hdfc-Chamrajpet	9000
2	hdfc-ShivajiRd	8000
3	hdfc-Jainagar	8500
4	hdfc-RajivChowk	9000
5	hdfc-JM	7000
NULL	NULL	NULL

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 branch_name, assets as "assets in lakhs"

from branch;



branch_name	assets in lakhs
hdfc-Chamrajpet	200000
hdfc-Jainagar	350000
hdfc-JM	600000
hdfc-RajivChowk	500000
hdfc-ShivajiRd	500000
NULL	NULL

2. Create a view which gives each branch the sum of the amount of all the loans at the branch.

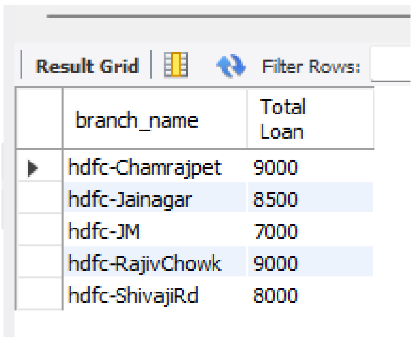
create view Total_Loan as

select branch_name, sum(amount) as "Total Loan"

from Loan

group by Branch_name;

select * from Total_Loan;



branch_name	Total Loan
hdfc-Chamrajpet	9000
hdfc-Jainagar	8500
hdfc-JM	7000
hdfc-RajivChowk	9000
hdfc-ShivajiRd	8000

3. Find all the customers who have atleast 2 accounts at the same branch(Ex: hdfc-Chamrajpet).

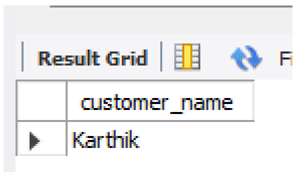
select d.customer_name

from Depositer d, BankAccount b

where d.accno=b.accno and b.branch_name="hdfc-Chamrajpet"

group by d.customer_name

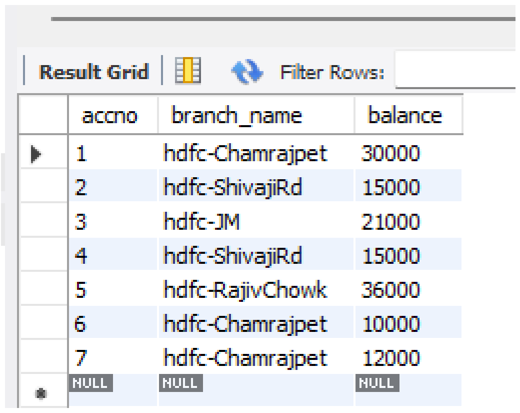
having count(d.accno)>=2;



customer_name
Karthik

4. Update the account balance by adding rupees 1000 for the customer residing in Banglore.

```
update BankAccount b
set b.balance=b.balance+1000
where accno in(select d.accno from Depositer d, BankCustomer b
        where d.customer_name=b.customer_name and b.city="Banglore");
select * from BankAccount;
```



accno	branch_name	balance
1	hdfc-Chamrajpet	30000
2	hdfc-ShivajiRd	15000
3	hdfc-JM	21000
4	hdfc-ShivajiRd	15000
5	hdfc-RajivChowk	36000
6	hdfc-Chamrajpet	10000
7	hdfc-Chamrajpet	12000
NULL	NULL	NULL

WEEK 4

Question

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

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

BankCustomer (customer-name: String, customer-street: String,

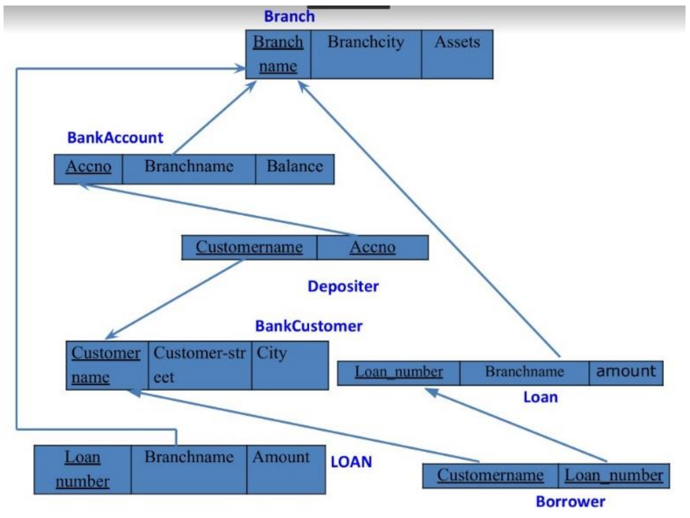
customer-city: String)

Depositer(customer-name: String, accno: int)

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

Borrower (customer-name: String, loan-number: int)

Relational Schema



Creating table 'Borrower'

create table Borrower

(

customer_name varchar(25),

loan_no int,

primary key(customer_name,loan_no),

foreign key(customer_name) references BankCustomer(customer_name),

foreign key(loan_no) references Loan(loan_no)

);

Inserting values into table ‘Borrower’

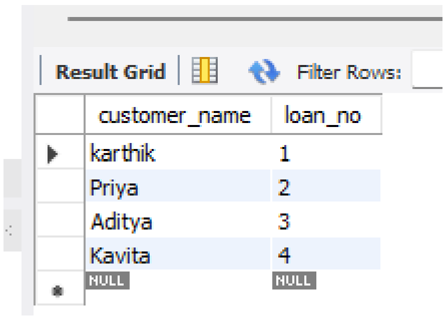
insert into Borrower values("karthik",1);

insert into Borrower values("Priya",2);

insert into Borrower values("Aditya",3);

insert into Borrower values("Kavita",4);

select * from Borrower;



customer_name	loan_no
karthik	1
Priya	2
Aditya	3
Kavita	4
NULL	NULL

Queries

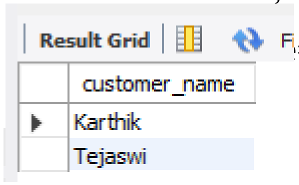
1. Find all the customers who have an account at all the branches located in a specific city (Ex: Banglore).

select distinct d.customer_name

from Depositer d

where d.accno in (select b.accno

from BankAccount b, branch br

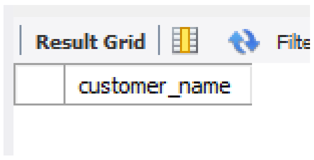


customer_name
Karthik
Tejaswi

=b.branch_name and br.branch_city="Banglore");

2.Find all customers who have a loan at the bank but do not have an account.

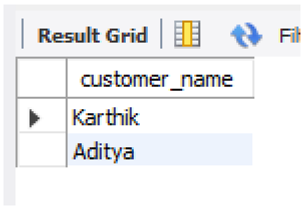
```
select customer_name
    from Borrower
    where customer_name not in(select customer_name
                                from depositer);
```



```
customer_name
```

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

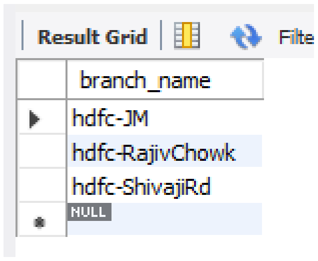
```
select distinct customer_name
    from depositer
    where customer_name in(select bo.customer_name
                            from Borrower bo, Loan l, branch br
                            where l.loan_no=bo.loan_no and l.branch_name=br.branch_name and br.branch_city="Banglore");
```



```
customer_name
Karthik
Aditya
```

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

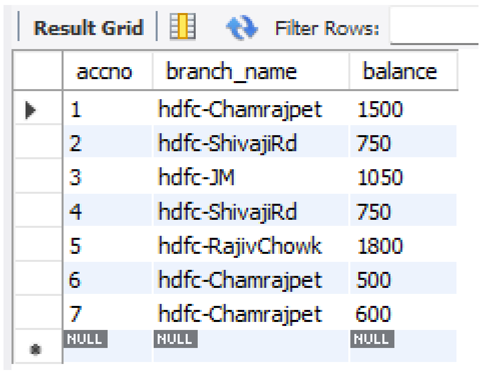
```
select branch_name
from branch
where assets >(select max(assets)
from branch
where branch_city="Banglore"
group by branch_city);
```



```
branch_name
hdfc-JM
hdfc-RajivChowk
hdfc-ShivajiRd
NULL
```

5. Update the Balance of all accounts by 5%.

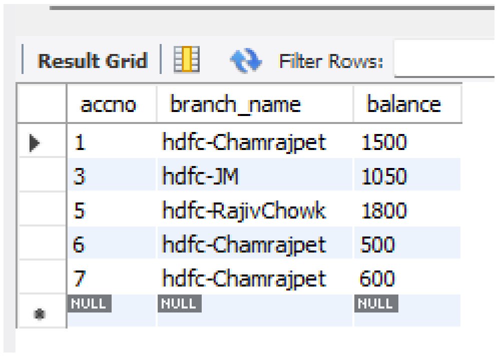
```
update BankAccount
set balance= (5/100)*balance;
select * from BankAccount;
```



accno	branch_name	balance
1	hdfc-Chamrajpet	1500
2	hdfc-ShivajiRd	750
3	hdfc-JM	1050
4	hdfc-ShivajiRd	750
5	hdfc-RajivChowk	1800
6	hdfc-Chamrajpet	500
7	hdfc-Chamrajpet	600
NULL	NULL	NULL

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

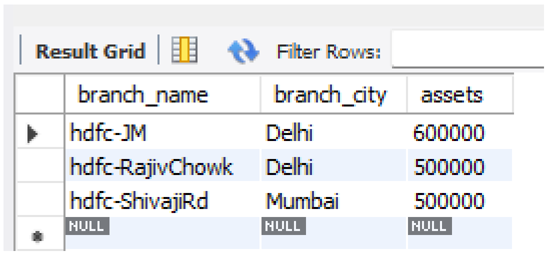
```
delete from Depositer
where  accno in(select accno from BankAccount where branch_name in(select branch_name from branch where branch_city="Mumbai"));
delete from BankAccount
where branch_name in(select branch_name from branch where branch_city="Mumbai");
```



accno	branch_name	balance
1	hdfc-Chamrajpet	1500
3	hdfc-JM	1050
5	hdfc-RajivChowk	1800
6	hdfc-Chamrajpet	500
7	hdfc-Chamrajpet	600
NULL	NULL	NULL

7. Demonstrate how you delete all branches located at Banglore.

```
delete from Depositer
  where  accno in(select accno from BankAccount where branch_name in(select branch_name
from branch where branch_city="Banglore"));

 delete from BankAccount
where branch_name in(select branch_name from branch where branch_city="Banglore");

delete from Loan
where branch_name in(select branch_name from branch where branch_city="Banglore");

delete from Borrower
where loan_no in(select loan_no from Loan);

delete from branch
where branch_city="Banglore";
select * from branch;
```

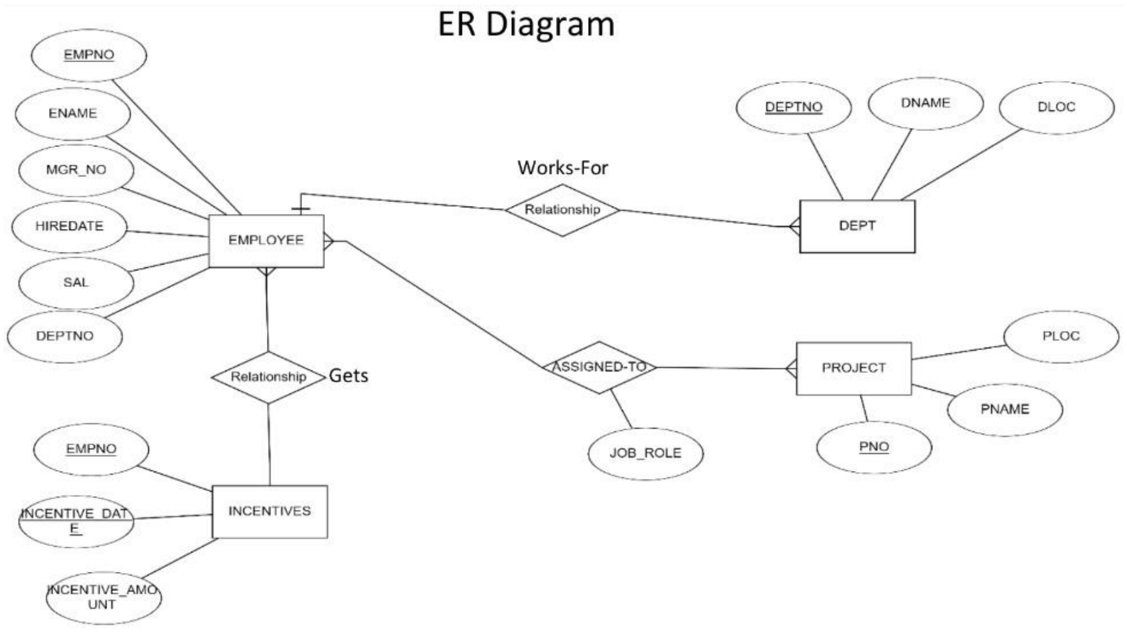


branch_name	branch_city	assets
hdfc-JM	Delhi	600000
hdfc-RajivChowk	Delhi	500000
hdfc-ShivajiRd	Mumbai	500000
NULL	NULL	NULL

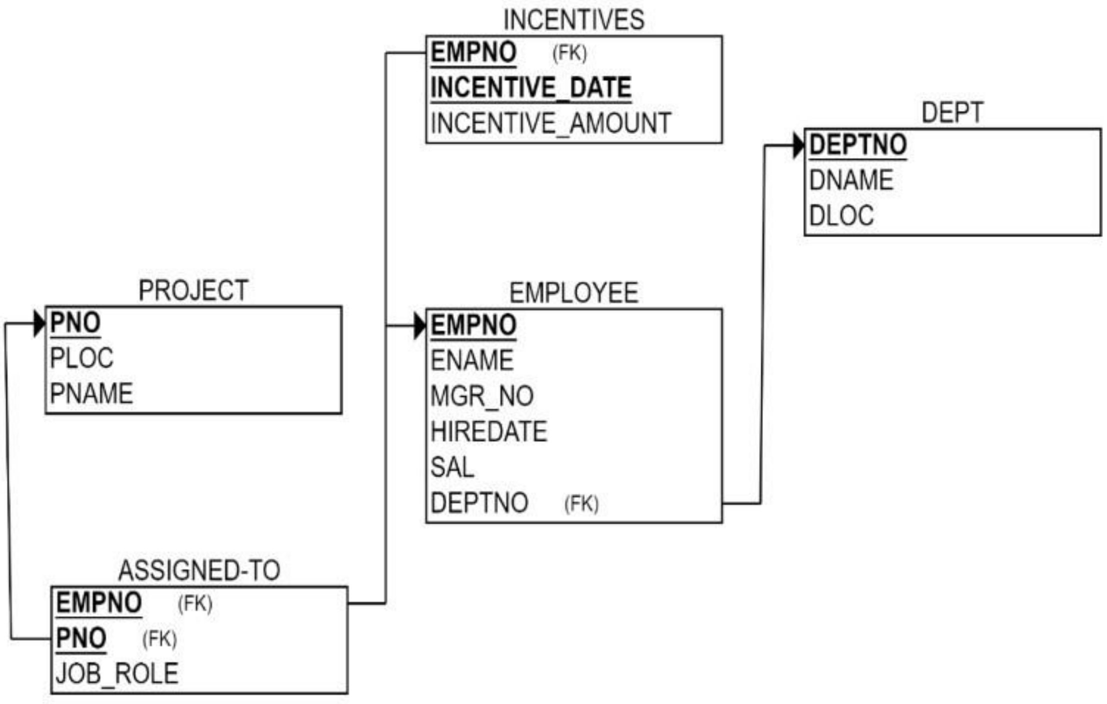
EXPERIMENT 3- EMPLOYEE DATABASE

WEEK 5

Question



Relational Schema



Creating database

```
Create database 1BM21CS113_Employee;
```

Using database

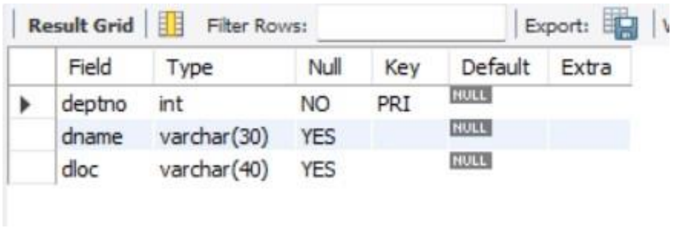
```
use 1BM21CS113_Employee;
```

Creating tables

```
create table Dept
(
deptno int primary key,
dname varchar(30),
dloc varchar(40)
);
create table Employee
(
empno int primary key,
ename varchar(40),
mgr_no int,
hiredate date,
sal decimal,
deptno int,
foreign key(deptno) references Dept(deptno) on delete cascade  on update cascade
);
create table Project
(
   pno int primary key,
   ploc varchar(40),
   pname varchar(40)
);
create table Assigned_to
(
   empno int,
pno int,
primary key(empno,pno),
foreign key(empno) references Employee(empno) on delete cascade  on update cascade,
foreign key(pno) references Project(pno) on delete cascade  on update cascade,
job_role varchar(30)
);
create table Incentives
(
   empno int,
   incentive_date date,
   primary key(empno,incentive_date),
   foreign key(empno) references Employee(empno) on delete cascade  on update cascade,
   incentive_amt decimal
);
```

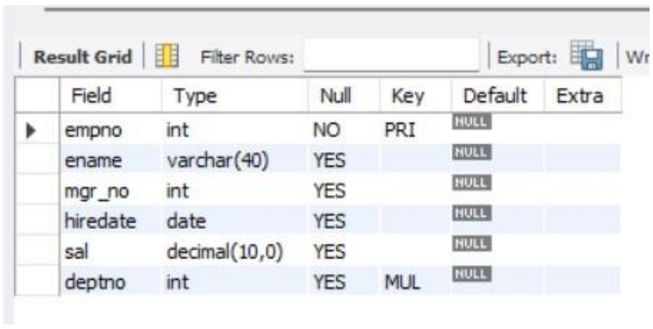
Structure of tables

desc Dept;



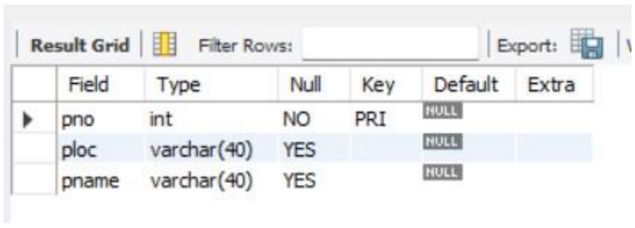
Field	Type	Null	Key	Default	Extra
deptno	int	NO	PRI	NULL	
dname	varchar(30)	YES		NULL	
dloc	varchar(40)	YES		NULL	

desc Employee;



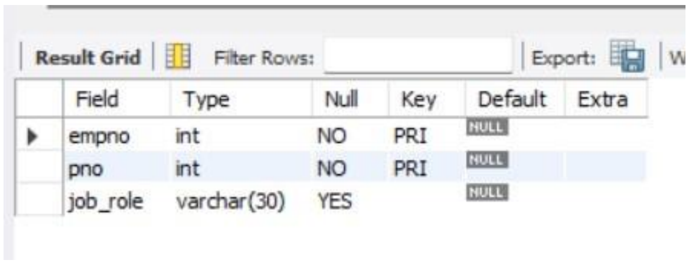
Field	Type	Null	Key	Default	Extra
empno	int	NO	PRI	NULL	
ename	varchar(40)	YES		NULL	
mgr_no	int	YES		NULL	
hiredate	date	YES		NULL	
sal	decimal(10,0)	YES		NULL	
deptno	int	YES	MUL	NULL	

desc Project;



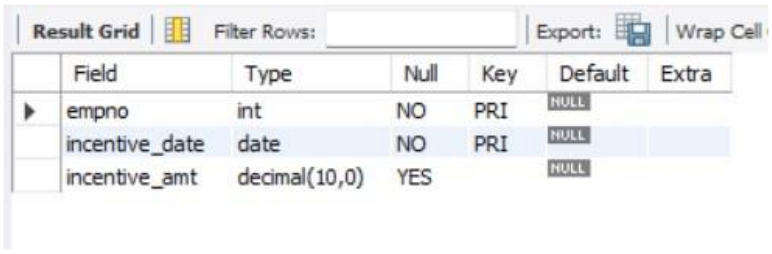
Field	Type	Null	Key	Default	Extra
pno	int	NO	PRI	NULL	
ploc	varchar(40)	YES		NULL	
pname	varchar(40)	YES		NULL	

desc Assigned_to;



Field	Type	Null	Key	Default	Extra
empno	int	NO	PRI	NULL	
pno	int	NO	PRI	NULL	
job_role	varchar(30)	YES		NULL	

desc Incentives;

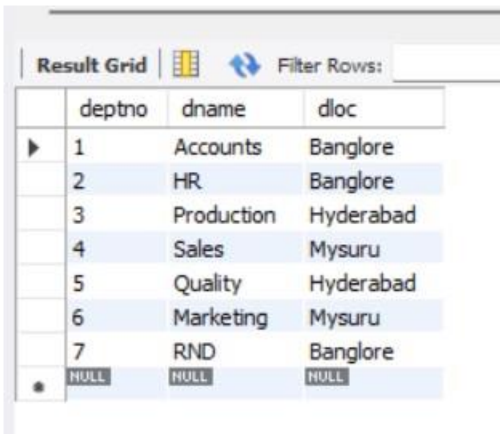


Field	Type	Null	Key	Default	Extra
empno	int	NO	PRI	NULL	
incentive_date	date	NO	PRI	NULL	
incentive_amt	decimal(10,0)	YES		NULL	

Inserting values into tables

```
insert into Dept values(01,"Accounts","Banglore");
insert into Dept values(02,"HR","Banglore");
insert into Dept values(03,"Production","Hyderabad");
insert into Dept values(04,"Sales","Mysuru");
insert into Dept values(05,"Quality","Hyderabad");
insert into Dept values(06,"Marketing","Mysuru");
insert into Dept values(07,"RND","Banglore");
```

select * from Dept;



Result Grid | Filter Rows:

	deptno	dname	dloc
▶	1	Accounts	Banglore
	2	HR	Banglore
	3	Production	Hyderabad
	4	Sales	Mysuru
	5	Quality	Hyderabad
	6	Marketing	Mysuru
	7	RND	Banglore
●	NULL	NULL	NULL

insert into Employee values(10,"John","001","2017-04-12",50000,01);

insert into Employee values(11,"Bob","002","2018-05-11",40000,01);

insert into Employee values(12,"Kavita","003","2016-10-14",100000,02);

insert into Employee values(13,"Aliya","004","2015-04-16",55000,03);

insert into Employee values(14,"Ayan","005","2014-04-12",110000,02);

insert into Employee values(15,"Avinash","006","2014-09-20",60000,04);

insert into Employee values(16,"Aditi","007","2013-11-11",90000,07);

insert into Employee values(17,"Naina","008","2015-12-29",95000,07);

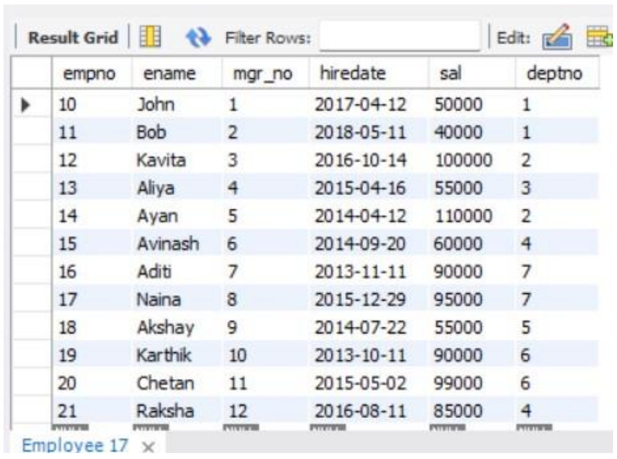
insert into Employee values(18,"Akshay","009","2014-07-22",55000,05);

insert into Employee values(19,"Karthik","010","2013-10-11",90000,06);

insert into Employee values(20,"Chetan","011","2015-05-02",99000,06);

insert into Employee values(21,"Raksha","012","2016-08-11",85000,04);

select * from Employee;



Result Grid | Filter Rows: | Edit:

	empno	ename	mgr_no	hiredate	sal	deptno
▶	10	John	1	2017-04-12	50000	1
	11	Bob	2	2018-05-11	40000	1
	12	Kavita	3	2016-10-14	100000	2
	13	Aliya	4	2015-04-16	55000	3
	14	Ayan	5	2014-04-12	110000	2
	15	Avinash	6	2014-09-20	60000	4
	16	Aditi	7	2013-11-11	90000	7
	17	Naina	8	2015-12-29	95000	7
	18	Akshay	9	2014-07-22	55000	5
	19	Karthik	10	2013-10-11	90000	6
	20	Chetan	11	2015-05-02	99000	6
	21	Raksha	12	2016-08-11	85000	4

Employee 17 ×

insert into Project values(1,"Banglore","Satellite Mapping");

insert into Project values(2,"Delhi","E-commerce");

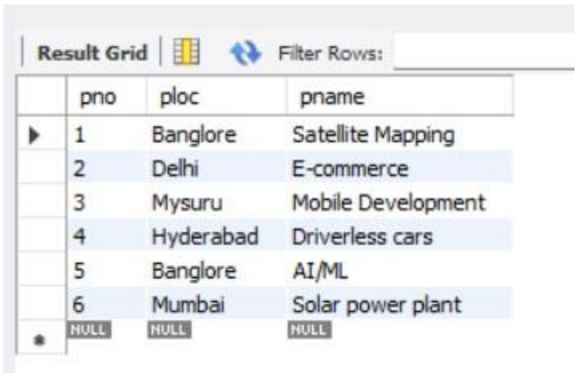
insert into Project values(3,"Mysuru","Mobile Development");

insert into Project values(4,"Hyderabad","Driverless cars");

insert into Project values(5,"Banglore","AI/ML");

insert into Project values(6,"Mumbai","Solar power plant");

select * from Project;



pno	ploc	pname
1	Banglore	Satellite Mapping
2	Delhi	E-commerce
3	Mysuru	Mobile Development
4	Hyderabad	Driverless cars
5	Banglore	AI/ML
6	Mumbai	Solar power plant
NULL	NULL	NULL

insert into Assigned_to values(10,1,"Manager");

insert into Assigned_to values(11,1,"Production head");

insert into Assigned_to values(12,5,"Financial Advisor");

insert into Assigned_to values(13,4,"Manager");

insert into Assigned_to values(14,5,"Manager");

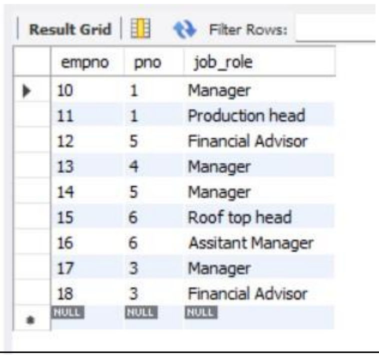
insert into Assigned_to values(15,6,"Roof top head");

insert into Assigned_to values(16,6,"Assitant Manager");

insert into Assigned_to values(17,3,"Manager");

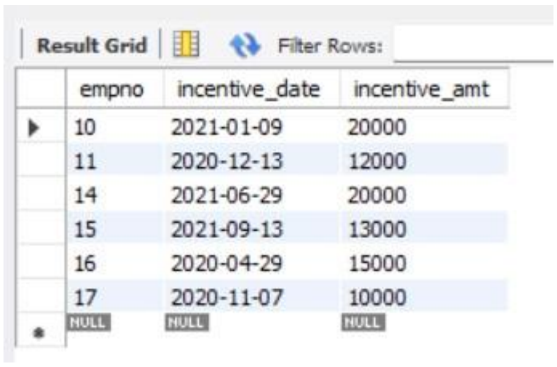
insert into Assigned_to values(18,3,"Financial Advisor");

select * from Assigned_to;



empno	pno	job_role
10	1	Manager
11	1	Production head
12	5	Financial Advisor
13	4	Manager
14	5	Manager
15	6	Roof top head
16	6	Assitant Manager
17	3	Manager
18	3	Financial Advisor
NULL	NULL	NULL

```
insert into Incentives values(11,"2020-12-13",12000);

insert into Incentives values(14,"2021-06-29",20000);

insert into Incentives values(15,"2021-09-13",13000);

insert into Incentives values(17,"2020-11-07",10000);

insert into Incentives values(10,"2021-01-09",20000);

insert into Incentives values(16,"2020-04-29",15000);


select * from Incentives;
```

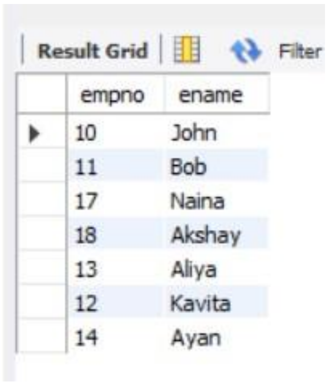


empno	incentive_date	incentive_amt
10	2021-01-09	20000
11	2020-12-13	12000
14	2021-06-29	20000
15	2021-09-13	13000
16	2020-04-29	15000
17	2020-11-07	10000
NULL	NULL	NULL

Queries

1. Retrieve the employee number and names who work on project located in Banglore, Hyderabad or Mysuru.

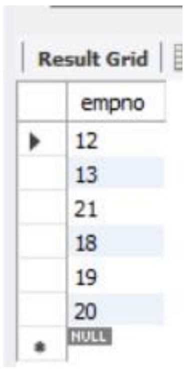
```
select e.empno, e.ename

from Employee e,Project p, Assigned_to a

where e.empno=a.empno and

    a.pno=p.pno and

    p.ploc in("Banglore","Hyderabad","Mysuru");
```



empno	ename
10	John
11	Bob
17	Naina
18	Akshay
13	Aliya
12	Kavita
14	Ayan

2. Get the employee numbers of all employees who did not receive incentives.

```
select empno
from Employee
where empno not in(select empno from Incentives);
```

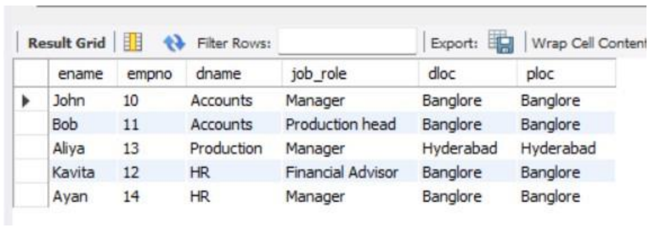


Result Grid

empno
12
13
21
18
19
20
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,e.empno,d.dname,a.job_role,d.dloc,p.ploc
from Employee e, Dept d, Assigned_to a,Project p
where e.deptno=d.deptno and
      a.empno=e.empno and
      a.pno=p.pno and
      p.ploc=d.dloc;
```

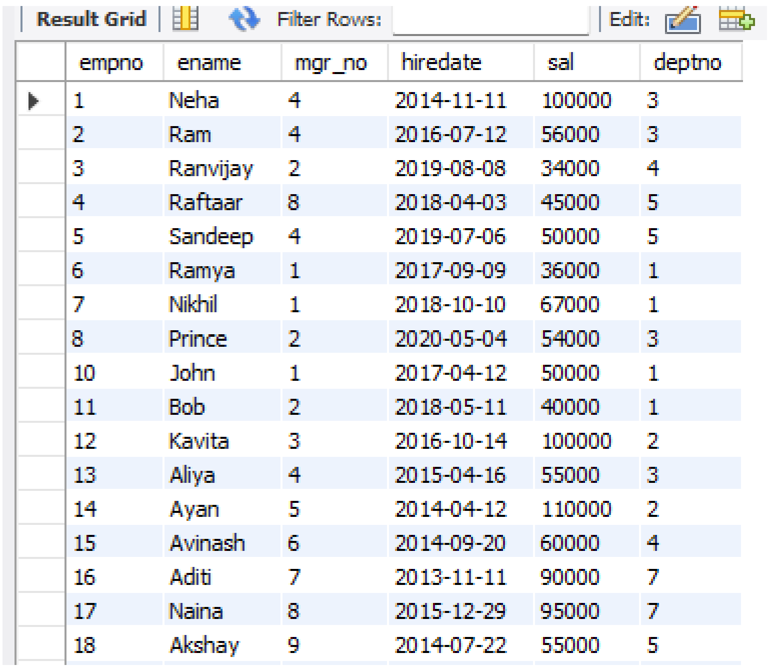


ename	empno	dname	job_role	dloc	ploc
John	10	Accounts	Manager	Banglore	Banglore
Bob	11	Accounts	Production head	Banglore	Banglore
Aliya	13	Production	Manager	Hyderabad	Hyderabad
Kavita	12	HR	Financial Advisor	Banglore	Banglore
Ayan	14	HR	Manager	Banglore	Banglore

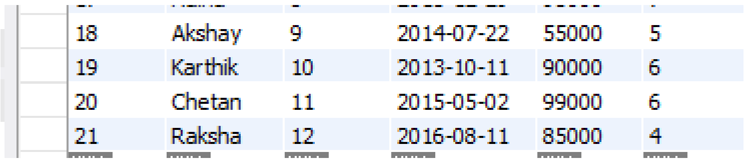
WEEK 6

Inserting more values in the table ‘Employee’

```
insert into Employee values(10,"John","001","2017-04-12",50000,01);
insert into Employee values(11,"Bob","002","2018-05-11",40000,01);
insert into Employee values(12,"Kavita","003","2016-10-14",100000,02);
insert into Employee values(13,"Aliya","004","2015-04-16",55000,03);
insert into Employee values(14,"Ayan","005","2014-04-12",110000,02);
insert into Employee values(15,"Avinash","006","2014-09-20",60000,04);
insert into Employee values(16,"Aditi","007","2013-11-11",90000,07);
insert into Employee values(17,"Naina","008","2015-12-29",95000,07);
insert into Employee values(18,"Akshay","009","2014-07-22",55000,05);
insert into Employee values(19,"Karthik","010","2013-10-11",90000,06);
insert into Employee values(20,"Chetan","011","2015-05-02",99000,06);
insert into Employee values(21,"Raksha","012","2016-08-11",85000,04);
insert into Employee values(006,"Ramya",001,"2017-09-09",36000,01);
insert into Employee values(007,"Nikhil",001,"2018-10-10",67000,01);
insert into Employee values(001,"Neha",004,"2014-11-11",100000,03);
insert into Employee values(002,"Ram",004,"2016-07-12",56000,03);
insert into Employee values(003,"Ranvijay",002,"2019-08-08",34000,04);
insert into Employee values(004,"Raftaar",001,"2018-04-03",45000,05);
insert into Employee Values(005,"Sandeep",004,"2019-07-06",50000,05);
insert into Employee values(008,"Prince",002,"2020-05-04",54000,03);

update Employee
set mgr_no=008 where ename="Raftaar";
select * from Employee;
```



empno	ename	mgr_no	hiredate	sal	deptno
1	Neha	4	2014-11-11	100000	3
2	Ram	4	2016-07-12	56000	3
3	Ranvijay	2	2019-08-08	34000	4
4	Raftaar	8	2018-04-03	45000	5
5	Sandeep	4	2019-07-06	50000	5
6	Ramya	1	2017-09-09	36000	1
7	Nikhil	1	2018-10-10	67000	1
8	Prince	2	2020-05-04	54000	3
10	John	1	2017-04-12	50000	1
11	Bob	2	2018-05-11	40000	1
12	Kavita	3	2016-10-14	100000	2
13	Aliya	4	2015-04-16	55000	3
14	Ayan	5	2014-04-12	110000	2
15	Avinash	6	2014-09-20	60000	4
16	Aditi	7	2013-11-11	90000	7
17	Naina	8	2015-12-29	95000	7
18	Akshay	9	2014-07-22	55000	5

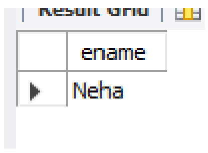


18	Akshay	9	2014-07-22	55000	5
19	Karthik	10	2013-10-11	90000	6
20	Chetan	11	2015-05-02	99000	6
21	Raksha	12	2016-08-11	85000	4

Queries

1. List the name of the managers with maximum number of employees.

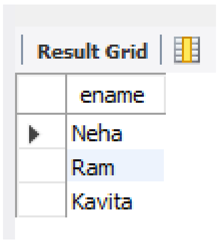
```
select m.ename
from Employee e, Employee m
where e.mgr_no=m.empno
group by m.ename
having count(*)=(select max(MyCount)
                from (select count(*) as MyCount
                      from Employee
                      group by mgr_no)a);
```

\ 

ename
Neha

2. Display the names of the managers whose salary is more than the average salary of his employee.

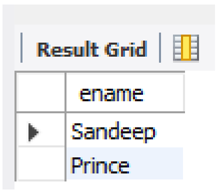
```
select m.ename
from  Employee m
where  m.sal>= (select avg(e.sal)
                                   from Employee e
             where m.empno=e.mgr_no
             group by e.mgr_no);
```



```
Result Grid
ename
Neha
Ram
Kavita
```

3. Display the names of the employees who are working in the same department his manager is working.

```
select e.ename
from Employee e,Employee m
where e.mgr_no=m.empno and
     e.deptno=m.deptno;
```



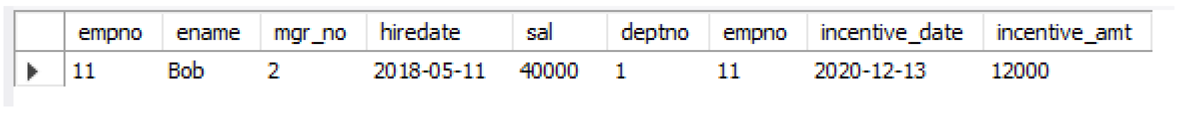
```
Result Grid
ename
Sandeep
Prince
```

4. Find the employee details who got second maximum incentive in 2020.

```
select * from
Employee e, Incentives i
where e.empno=i.empno and
1=(select count(distinct j.incentive_amt)
```

from Incentives j

where i.incentive_amt<j.incentive_amt and j.incentive_date like "2020%") and i.incentive_date like "2020%" ;



	empno	ename	mgr_no	hiredate	sal	deptno	empno	incentive_date	incentive_amt
▶	11	Bob	2	2018-05-11	40000	1	11	2020-12-13	12000

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

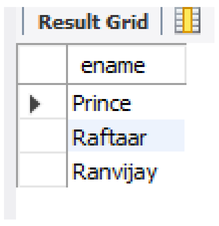
select g. ename

from Employee e, Employee m, Employee g

where e.mgr_no=m. empno and

m.mgr_no=g.empno

group by g.deptno;



	ename
▶	Prince
	Raftaar
	Ranvijay

EXPERIMENT 4- SUPPLIER DATABASE

WEEK 7

Question

Consider the following schema:

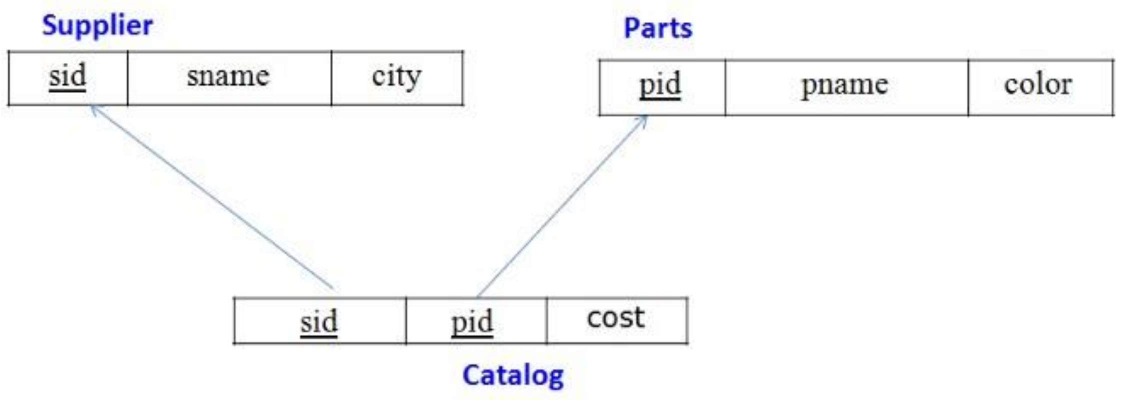
SUPPLIERS(sid: integer, sname: string, address: string)

PARTS(pid: integer, pname: string, color: string)

CATALOG(sid: integer, pid: integer, cost: real)

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

Relational Schema



Creating database

create database 1BM21CS113_Supplier;

Using database

use 1BM21CS113_Supplier;

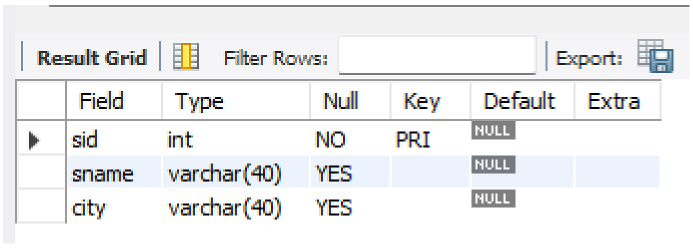
Creating tables

```
create table Supplier
(
sid int primary key,
sname varchar(40),
city varchar(40)
);
```

```
create table Parts
(
  pid int primary key,
  pname varchar(30),
  color varchar(30)
 );

create table Catalog
(
sid int,
pid int,
primary key(sid,pid),
foreign key(sid) references Supplier(sid) on delete cascade on update cascade,
foreign key(pid) references Parts(pid) on delete cascade on update cascade
);
```

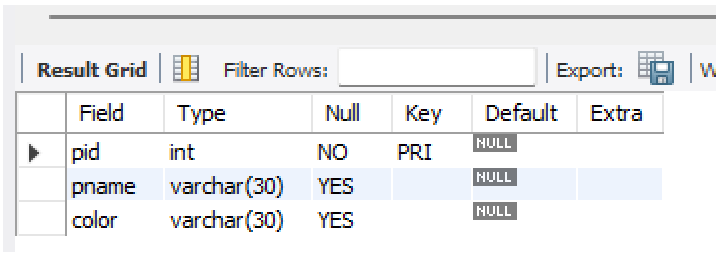
Structure of tables

desc Supplier;


Result Grid | Filter Rows: | Export:

Field	Type	Null	Key	Default	Extra
sid	int	NO	PRI	NULL	
sname	varchar(40)	YES		NULL	
city	varchar(40)	YES		NULL	

desc Parts;

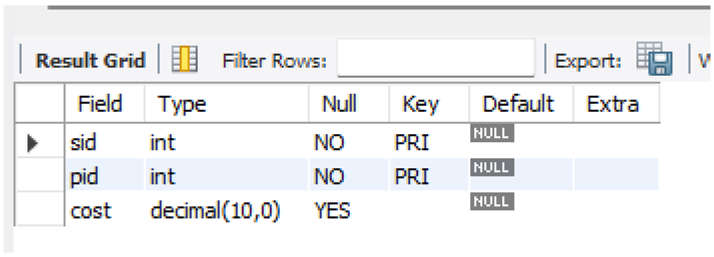

Result Grid | Filter Rows: | Export: | W

Field	Type	Null	Key	Default	Extra
pid	int	NO	PRI	NULL	
pname	varchar(30)	YES		NULL	
color	varchar(30)	YES		NULL	

Alter table Catalog

add column cost decimal;

desc Catalog;



Field	Type	Null	Key	Default	Extra
sid	int	NO	PRI	NULL	
pid	int	NO	PRI	NULL	
cost	decimal(10,0)	YES		NULL	

Inserting values into the tables

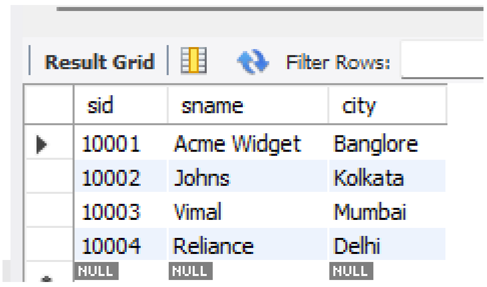
insert into Supplier values(10001,"Acme Widget","Banglore");

insert into Supplier values(10002,"Johns","Kolkata");

insert into Supplier values(10003,"Vimal","Mumbai");

insert into Supplier values(10004,"Reliance","Delhi");

select * from Supplier;



sid	sname	city
10001	Acme Widget	Banglore
10002	Johns	Kolkata
10003	Vimal	Mumbai
10004	Reliance	Delhi
NULL	NULL	NULL

insert into Parts values(2001,"Book","Red");

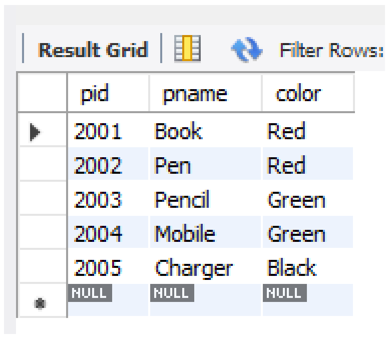
insert into Parts values(2002,"Pen","Red");

insert into Parts values(2003,"Pencil","Green");

insert into Parts values(2004,"Mobile","Green");

insert into Parts values(2005,"Charger","Black");

select * from Parts;



pid	pname	color
2001	Book	Red
2002	Pen	Red
2003	Pencil	Green
2004	Mobile	Green
2005	Charger	Black
NULL	NULL	NULL

insert into Catalog values(10001,2001,10);

insert into Catalog values(10001,2002,10);

insert into Catalog values(10001,2003,30);

insert into Catalog values(10001,2004,10);

insert into Catalog values(10001,2005,10);

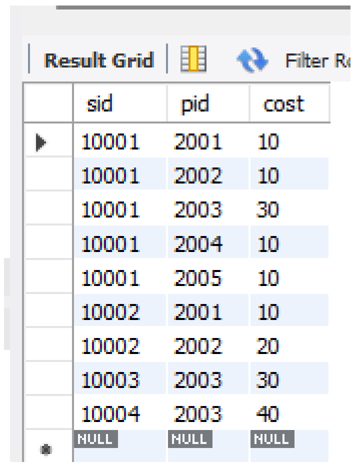
insert into Catalog values(10002,2001,10);

insert into Catalog values(10002,2002,20);

insert into Catalog values(10003,2003,30);

insert into Catalog values(10004,2003,40);

select * from Catalog;

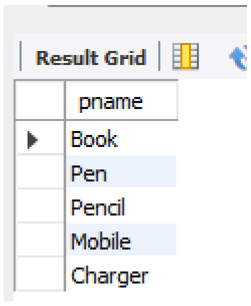


sid	pid	cost
10001	2001	10
10001	2002	10
10001	2003	30
10001	2004	10
10001	2005	10
10002	2001	10
10002	2002	20
10003	2003	30
10004	2003	40
NULL	NULL	NULL

Queries

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

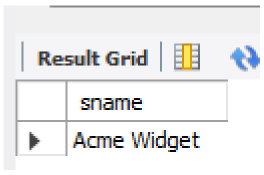
```
select  distinct p.pname
from Parts p, Supplier s, Catalog c
where s.sid=c.sid and p.pid=c.pid;
```



```
pname
Book
Pen
Pencil
Mobile
Charger
```

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

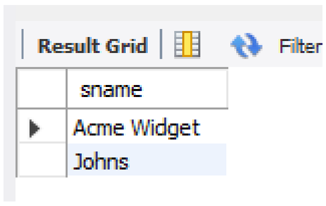
```
select distinct s.sname
from Supplier s
where s.sid in(select sid
               from Catalog
            group by sid
                having count(*)=(select count(*) from Parts));
```



```
sname
Acme Widget
```

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

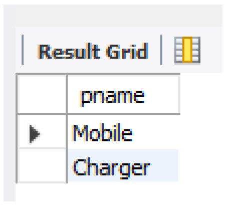
```
select distinct s.sname
from Parts p, Supplier s, Catalog c
where s.sid=c.sid and p.pid=c.pid and p.color="Red";
```



```
sname
Acme Widget
Johns
```

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

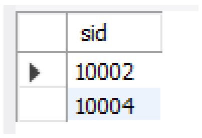
```
select distinct p.pname
from Parts p, Catalog c
where c.sid=(select sid from Supplier where sname="Acme Widget")
and p.pid=c.pid and c.pid not in(select c.pid
                                        from Catalog c
                                 where c.sid!=(select sid from Supplier where sname="Acme Widget"));
```



```
pname
Mobile
Charger
```

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

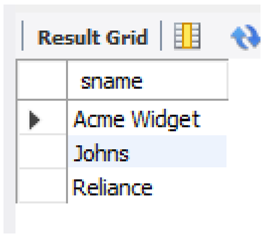
```
select c.sid
from  Catalog c
where c.cost>(select avg(cost) from catalog c1 where c.pid=c1.pid);
```



```
sid
10002
10004
```

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

```
select s.sname
from Supplier s
where s.sid in(select c.sid from Catalog c where c.cost=(select max(cost)
                                              from Catalog c1 where c.pid=c1.pid));
```



```
sname
Acme Widget
Johns
Reliance
```

EXPERIMENT 5- AIRLINE FLIGHT DATABASE

WEEK 8

Question

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

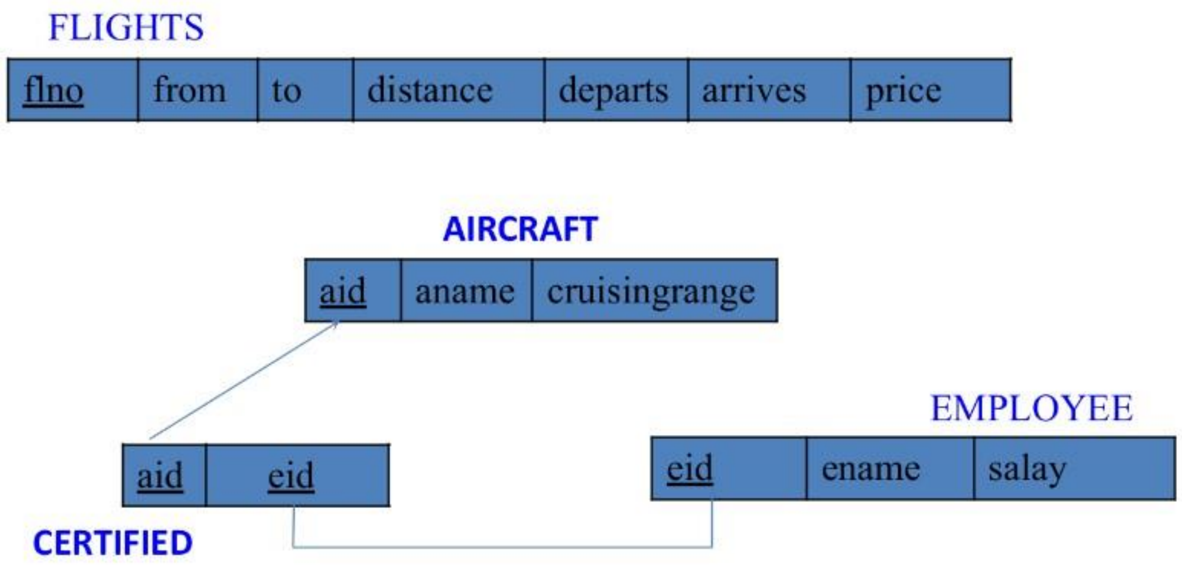
AIRCRAFT(aid: integer, aname: string, cruising_range: integer)

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.

Relational Schema



Creating database

create database 1bm21cs113_Week8_Flight;

Using database

use 1bm21cs113_Week8_FLight;

Creating tables

```
create table Flights
(
flno int primary key,

dep_airport varchar(40),

arr_airport varchar(40),

distance decimal,

departs time,

arrives time,

price decimal
);

create table Aircraft
(
aid int primary key,

aname varchar(40),

cruisingrange int
);

create table Employee
(
eid int primary key,

ename varchar(50),

salary decimal
);

create table certified
(
aid int,

eid int,

primary key(aid,eid),

foreign key(eid) references Employee(eid),

foreign key(aid) references Aircraft(aid)
);
```

Alter table certified

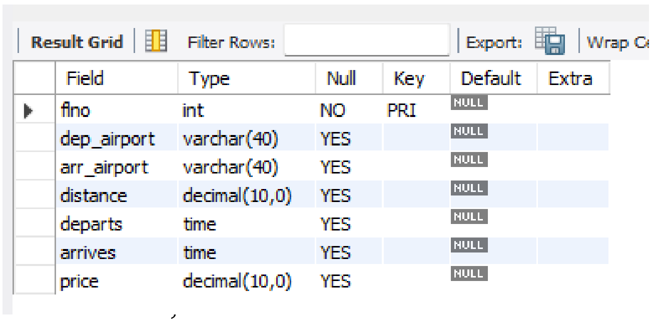
add foreign key(eid) references Employee(eid) on delete cascade on update cascade;

Alter table certified

add foreign key(aid) references Aircraft(aid) on delete cascade on update cascade;

Structure of tables

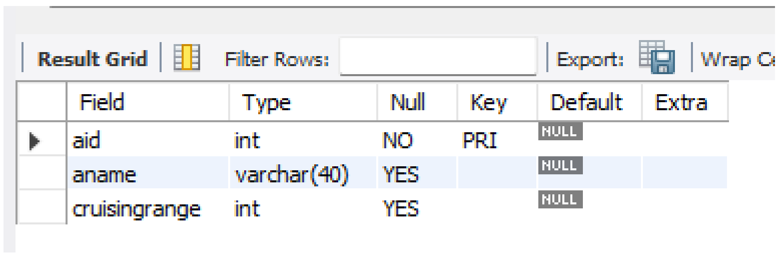
desc Flights;



Result Grid | Filter Rows: | Export: | Wrap C

Field	Type	Null	Key	Default	Extra
flno	int	NO	PRI	NULL	
dep_airport	varchar(40)	YES		NULL	
arr_airport	varchar(40)	YES		NULL	
distance	decimal(10,0)	YES		NULL	
departs	time	YES		NULL	
arrives	time	YES		NULL	
price	decimal(10,0)	YES		NULL	

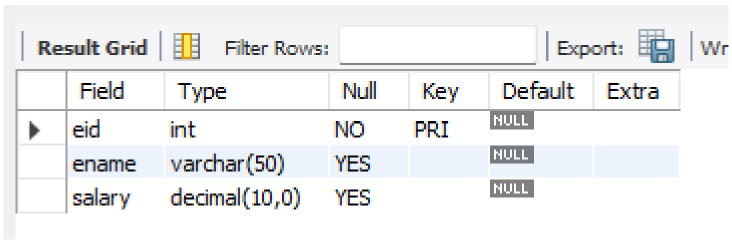
desc Aircraft;



Result Grid | Filter Rows: | Export: | Wrap C

Field	Type	Null	Key	Default	Extra
aid	int	NO	PRI	NULL	
aname	varchar(40)	YES		NULL	
cruisingrange	int	YES		NULL	

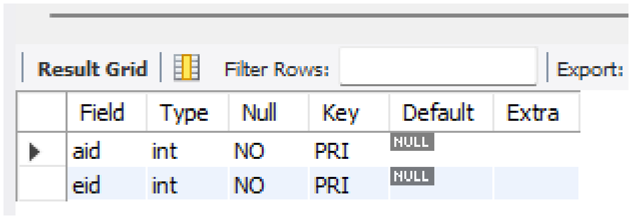
desc Employee;



Result Grid | Filter Rows: | Export: | Wr

Field	Type	Null	Key	Default	Extra
eid	int	NO	PRI	NULL	
ename	varchar(50)	YES		NULL	
salary	decimal(10,0)	YES		NULL	

desc certified;



Result Grid | Filter Rows: | Export:

Field	Type	Null	Key	Default	Extra
aid	int	NO	PRI	NULL	
eid	int	NO	PRI	NULL	

Inserting values into the tables

insert into Flights values(1,"Banglore","New Delhi",500,"6:00","9:00",5000);

insert into Flights values(2,"Banglore","Chennal",300,"7:00","8:30",3000);

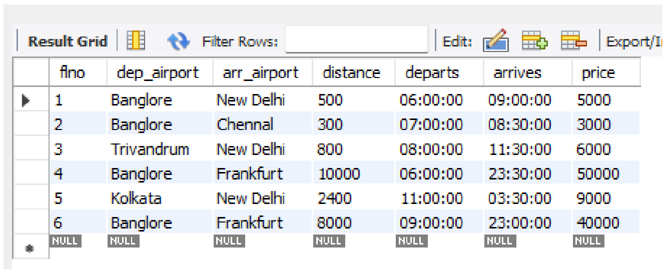
insert into Flights values(3,"Trivandrum","New Delhi",800,"8:00","11:30",6000);

insert into Flights values(4,"Banglore","Frankfurt",10000,"6:00","23:30",50000);

insert into Flights values(5,"Kolkata","New Delhi",2400,"11:00","3:30",9000);

insert into Flights values(6,"Banglore","Frankfurt",8000,"9:00","23:00",40000);

select * from Flights;



flno	dep_airport	arr_airport	distance	departs	arrives	price
1	Banglore	New Delhi	500	06:00:00	09:00:00	5000
2	Banglore	Chennal	300	07:00:00	08:30:00	3000
3	Trivandrum	New Delhi	800	08:00:00	11:30:00	6000
4	Banglore	Frankfurt	10000	06:00:00	23:30:00	50000
5	Kolkata	New Delhi	2400	11:00:00	03:30:00	9000
6	Banglore	Frankfurt	8000	09:00:00	23:00:00	40000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

insert into Aircraft values(1,"Airbus",2000);

insert into Aircraft values(2,"Boeing",700);

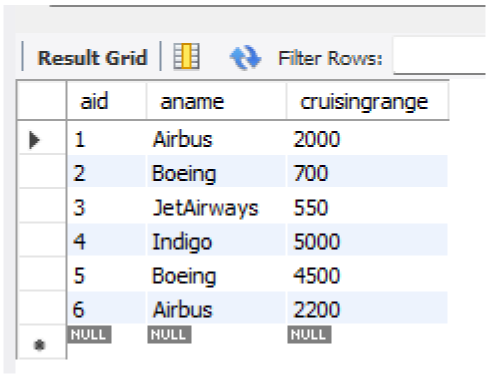
insert into Aircraft values(3,"JetAirways",550);

insert into Aircraft values(4,"Indigo",5000);

insert into Aircraft values(5,"Boeing",4500);

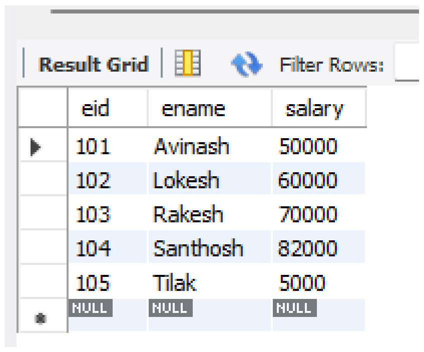
insert into Aircraft values(6,"Airbus",2200);

select * from Aircraft;



aid	aname	cruisingrange
1	Airbus	2000
2	Boeing	700
3	JetAirways	550
4	Indigo	5000
5	Boeing	4500
6	Airbus	2200
NULL	NULL	NULL

```
insert into Employee values(101,"Avinash",50000);

insert into Employee values(102,"Lokesh",60000);

insert into Employee values(103,"Rakesh",70000);

insert into Employee values(104,"Santhosh",82000);

insert into Employee  values(105,"Tilak",5000);

select * from Employee;
```

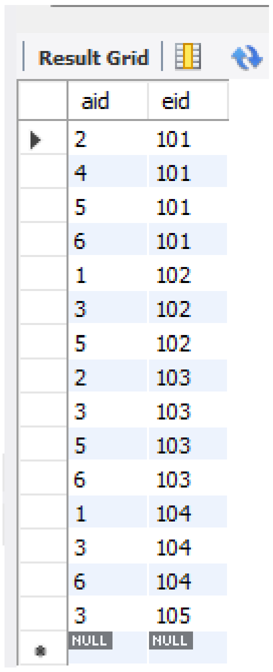


eid	ename	salary
101	Avinash	50000
102	Lokesh	60000
103	Rakesh	70000
104	Santhosh	82000
105	Tilak	5000
NULL	NULL	NULL

```
insert into certified values(2,101);

insert into certified values(4,101);

insert into certified values(5,101);

insert into certified values(6,101);

insert into certified values(1,102);

insert into certified values(3,102);

insert into certified values(5,102);

insert into certified values(2,103);

insert into certified values(3,103);

insert into certified values(5,103);

insert into certified values(6,103);

insert into certified values(6,104);

insert into certified values(1,104);

insert into certified values(3,104);
```

insert into certified values(3,105);

select * from certified;



Result Grid

aid	eid
2	101
4	101
5	101
6	101
1	102
3	102
5	102
2	103
3	103
5	103
6	103
1	104
3	104
6	104
3	105
NULL	NULL

Queries

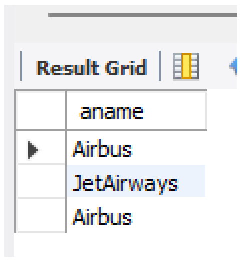
1. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.

select a.aname

from Aircraft a, certified c, Employee e

where a.aid=c.aid and c.eid=e.eid

and e.salary>80000;

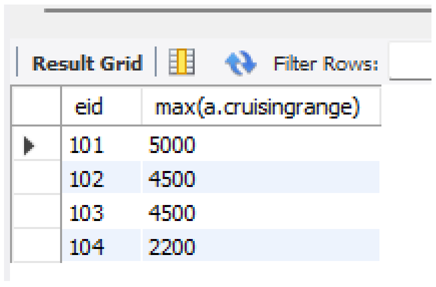


Result Grid

aname
Airbus
JetAirways
Airbus

2. For each pilot who is certified for greater than equal to three aircrafts, find the eid and the maximum cruising range of the aircraft for which she or he is certified.

```
 select c.eid, max(a.cruisingrange)

from Aircraft a, certified c, Employee e

where a.aid=c.aid and c.eid=e.eid

group by c.eid

having count(c.aid)>=3;
```

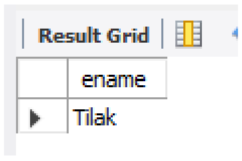


Result Grid | Filter Rows:

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

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< all(select min(price)

from Flight

where dep_airport="Banglore" and arr_airport="Frankfurt");
```

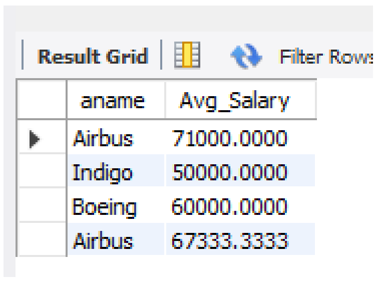


Result Grid

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) as Avg_Salary

from Aircraft a, certified c, Employee e

where a.aid=c.aid and c.eid=e.eid and a.cruisingrange>1000

group by c.aid;
```



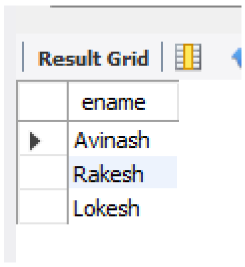
aname	Avg_Salary
Airbus	71000.0000
Indigo	50000.0000
Boeing	60000.0000
Airbus	67333.3333

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

select distinct e.ename

from Aircraft a, certified c, Employee e

where a.aid=c.aid and c.eid=e.eid and a.aname="Boeing";



ename
Avinash
Rakesh
Lokesh

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

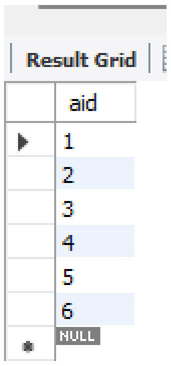
select distinct a.aid

from Aircraft a

where a.cruisingrange >any(select distance

from Flights

where dep_airport="Banglore" and arr_airport="New Delhi");



aid
1
2
3
4
5
6
NULL

EXPERIMENT 6- STUDENT DATABASE USING NoSQL

WEEK 9

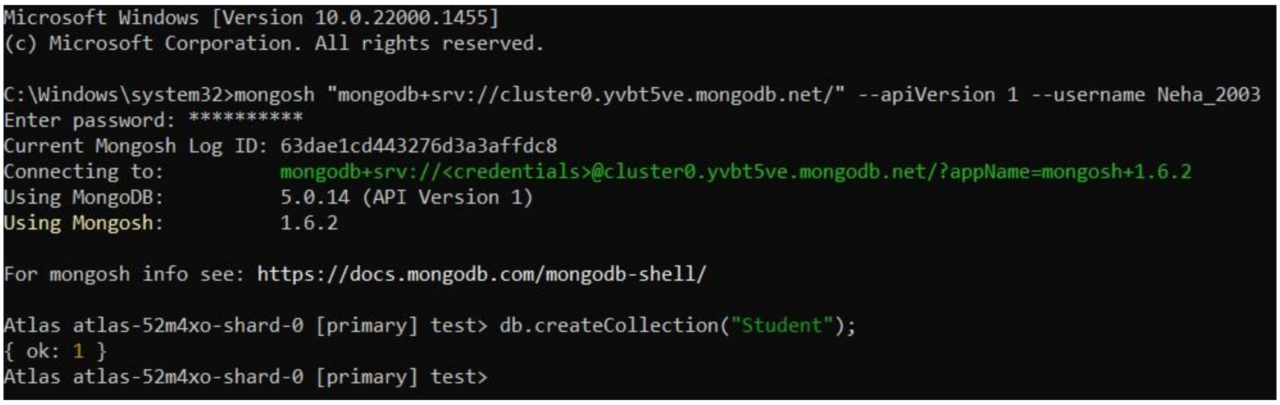
Question

Perform the database operations using MongoDB

Queries

1. Create a database Student with the following attributes Rollno, Age, Contact No, Email-id.

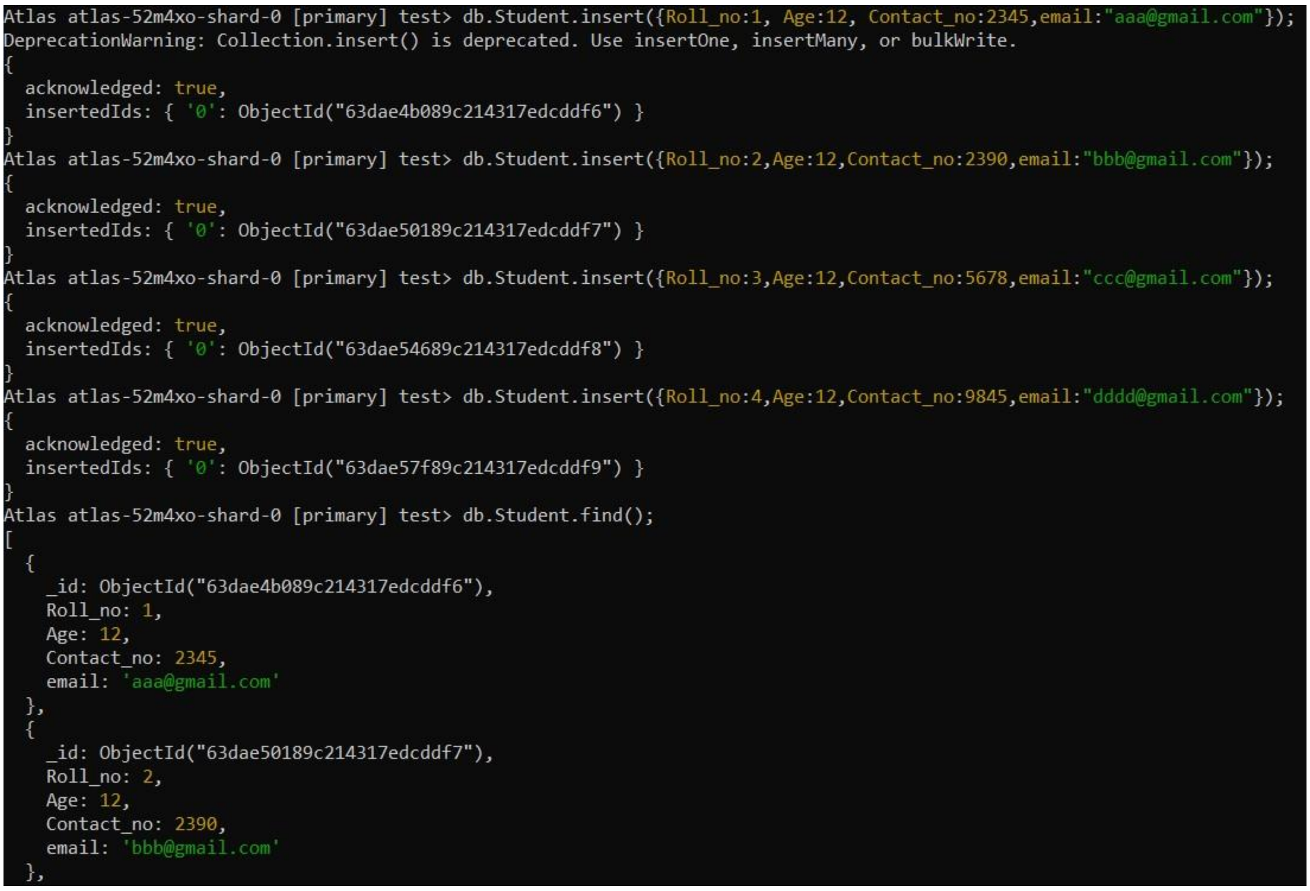
db. createCollection("Student");



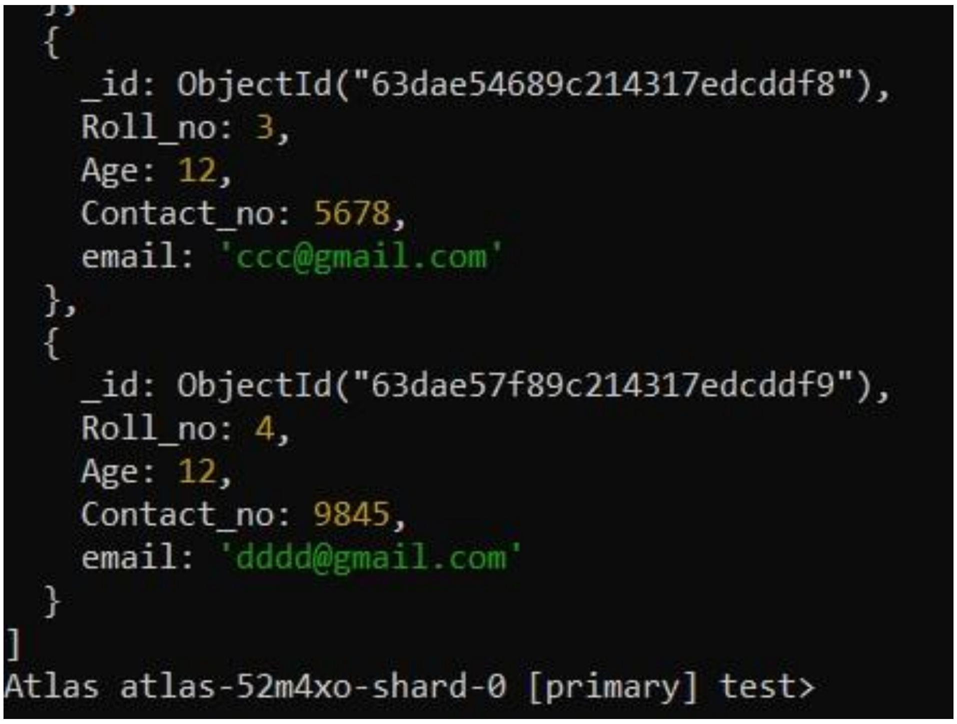
```
Microsoft Windows [Version 10.0.22000.1455]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>mongosh "mongodb+srv://cluster0.yvbt5ve.mongodb.net/" --apiVersion 1 --username Neha_2003
Enter password: **********
Current Mongosh Log ID: 63dae1cd443276d3a3affdc8
Connecting to:          mongodb+srv://<credentials>@cluster0.yvbt5ve.mongodb.net/?appName=mongosh+1.6.2
Using MongoDB:          5.0.14 (API Version 1)
Using Mongosh:          1.6.2

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Atlas atlas-52m4xo-shard-0 [primary] test> db.createCollection("Student");
{ ok: 1 }
Atlas atlas-52m4xo-shard-0 [primary] test>
```

2. Insert appropriate values.

```
db.Student.insert({Roll_no:1,Age:12,Contact_no:2345,email:"aaa@gmail"});
db.Student.insert({Roll_no:2,Age:12,Contact_no:2390,email:"bbb@gmail"});
db.Student.insert({Roll_no:3,Age:12,Contact_no:5678,email:"ccc@gmail"});
db.Student.insert({Roll_no:4,Age:12,Contact_no:9845,email:"dddd@gmail"});
```



```
Atlas atlas-52m4xo-shard-0 [primary] test> db.Student.insert({Roll_no:1, Age:12, Contact_no:2345,email:"aaa@gmail.com"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63dae4b089c214317edcddf6") }
}
Atlas atlas-52m4xo-shard-0 [primary] test> db.Student.insert({Roll_no:2,Age:12,Contact_no:2390,email:"bbb@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63dae50189c214317edcddf7") }
}
Atlas atlas-52m4xo-shard-0 [primary] test> db.Student.insert({Roll_no:3,Age:12,Contact_no:5678,email:"ccc@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63dae54689c214317edcddf8") }
}
Atlas atlas-52m4xo-shard-0 [primary] test> db.Student.insert({Roll_no:4,Age:12,Contact_no:9845,email:"dddd@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63dae57f89c214317edcddf9") }
}
Atlas atlas-52m4xo-shard-0 [primary] test> db.Student.find();
[
  {
    _id: ObjectId("63dae4b089c214317edcddf6"),
    Roll_no: 1,
    Age: 12,
    Contact_no: 2345,
    email: 'aaa@gmail.com'
  },
  {
    _id: ObjectId("63dae50189c214317edcddf7"),
    Roll_no: 2,
    Age: 12,
    Contact_no: 2390,
    email: 'bbb@gmail.com'
  },
```

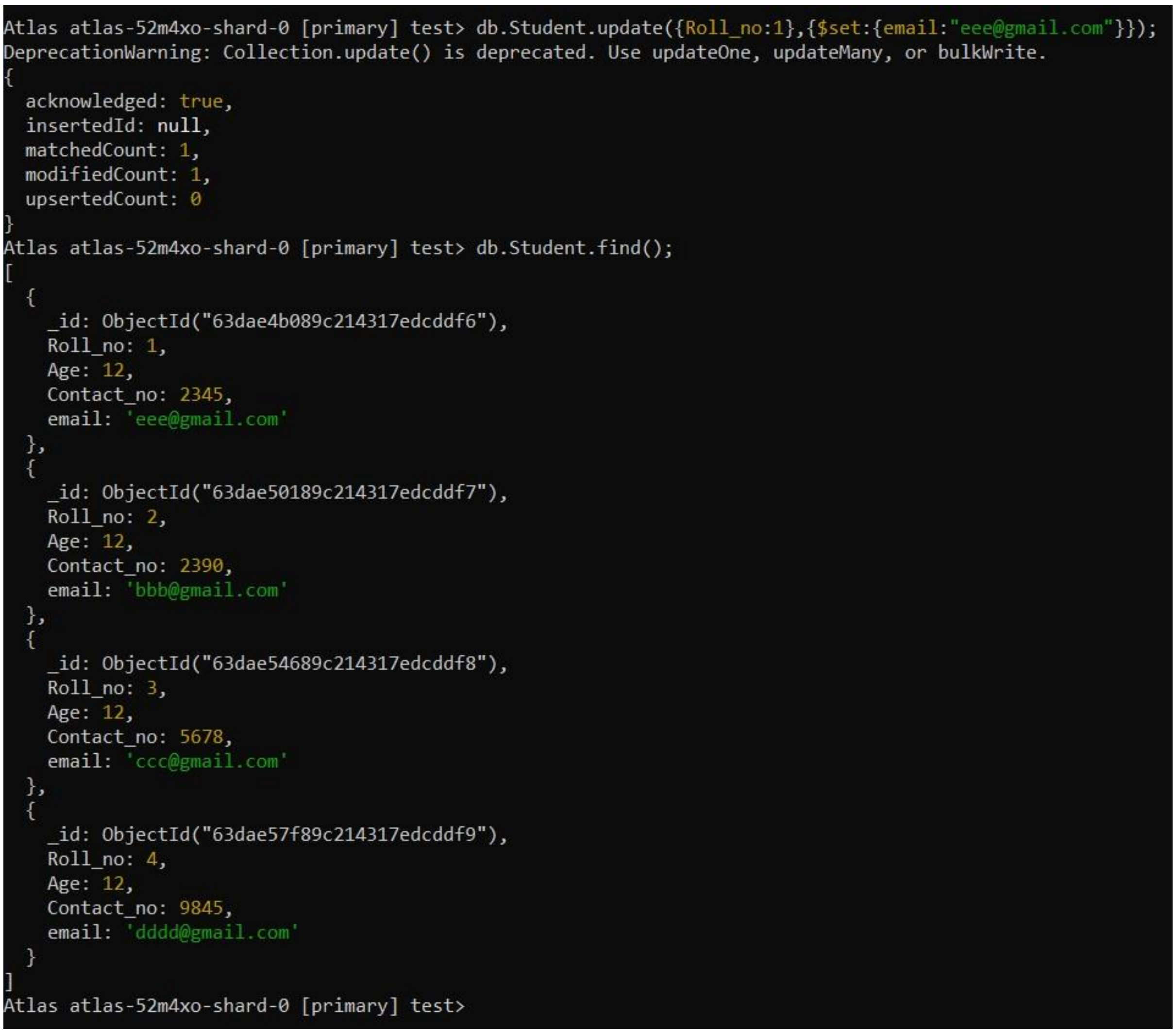


```
  {
    _id: ObjectId("63dae54689c214317edcddf8"),
    Roll_no: 3,
    Age: 12,
    Contact_no: 5678,
    email: 'ccc@gmail.com'
  },
  {
    _id: ObjectId("63dae57f89c214317edcddf9"),
    Roll_no: 4,
    Age: 12,
    Contact_no: 9845,
    email: 'dddd@gmail.com'
  }
]
Atlas atlas-52m4xo-shard-0 [primary] test>
```

3. Write query to update email if of a student with Roll_no 1.

db.Student.update({Roll_no:1},{$set:{email:"eee@gmail"}});

db.Student.find();

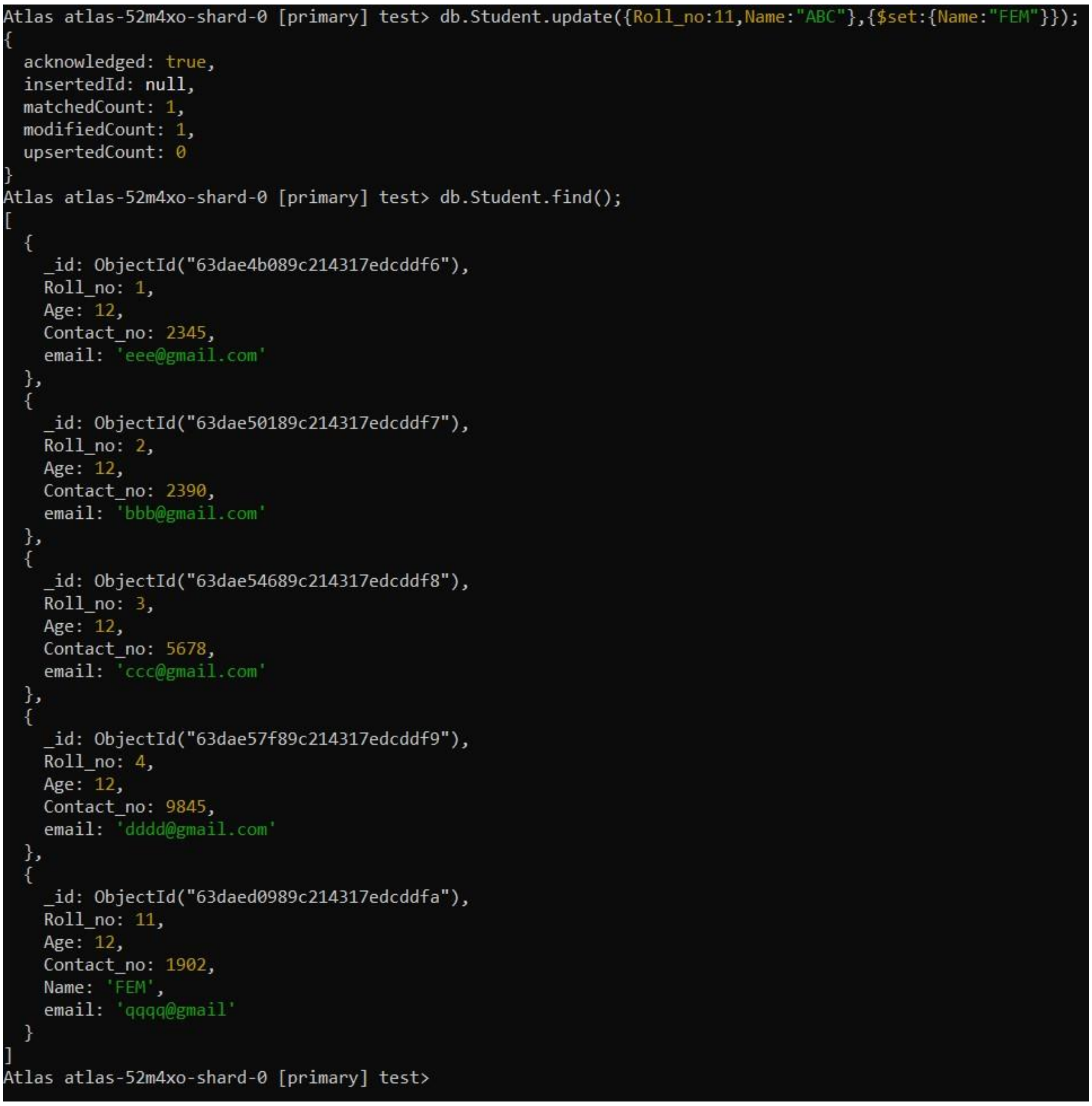


```
Atlas atlas-52m4xo-shard-0 [primary] test> db.Student.update({Roll_no:1},{$set:{email:"eee@gmail.com"}});
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-52m4xo-shard-0 [primary] test> db.Student.find();
[
  {
    _id: ObjectId("63dae4b089c214317edcddf6"),
    Roll_no: 1,
    Age: 12,
    Contact_no: 2345,
    email: 'eee@gmail.com'
  },
  {
    _id: ObjectId("63dae50189c214317edcddf7"),
    Roll_no: 2,
    Age: 12,
    Contact_no: 2390,
    email: 'bbb@gmail.com'
  },
  {
    _id: ObjectId("63dae54689c214317edcddf8"),
    Roll_no: 3,
    Age: 12,
    Contact_no: 5678,
    email: 'ccc@gmail.com'
  },
  {
    _id: ObjectId("63dae57f89c214317edcddf9"),
    Roll_no: 4,
    Age: 12,
    Contact_no: 9845,
    email: 'dddd@gmail.com'
  }
]
Atlas atlas-52m4xo-shard-0 [primary] test>
```

4. Replace the student name from “ABC” to “FEM” of Roll_no 6.

db.Student.update({Roll_no:6,Name:”ABC”},{$set:{Name:”FEM”}});

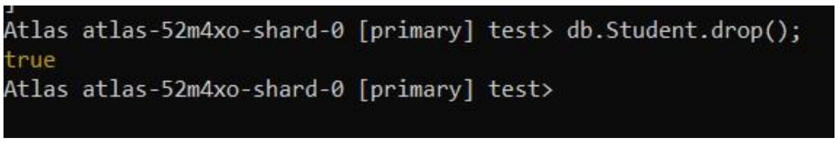
db.Student.find();



```
Atlas atlas-52m4xo-shard-0 [primary] test> db.Student.update({Roll_no:11,Name:"ABC"},{$set:{Name:"FEM"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-52m4xo-shard-0 [primary] test> db.Student.find();
[
  {
    _id: ObjectId("63dae4b089c214317edcddf6"),
    Roll_no: 1,
    Age: 12,
    Contact_no: 2345,
    email: 'eee@gmail.com'
  },
  {
    _id: ObjectId("63dae50189c214317edcddf7"),
    Roll_no: 2,
    Age: 12,
    Contact_no: 2390,
    email: 'bbb@gmail.com'
  },
  {
    _id: ObjectId("63dae54689c214317edcddf8"),
    Roll_no: 3,
    Age: 12,
    Contact_no: 5678,
    email: 'ccc@gmail.com'
  },
  {
    _id: ObjectId("63dae57f89c214317edcddf9"),
    Roll_no: 4,
    Age: 12,
    Contact_no: 9845,
    email: 'dddd@gmail.com'
  },
  {
    _id: ObjectId("63daed0989c214317edcddfa"),
    Roll_no: 11,
    Age: 12,
    Contact_no: 1902,
    Name: 'FEM',
    email: 'qqqq@gmail'
  }
]
Atlas atlas-52m4xo-shard-0 [primary] test>
```

5. Drop the collection Student.

Db.Student.drop();



```
Atlas atlas-52m4xo-shard-0 [primary] test> db.Student.drop();
true
Atlas atlas-52m4xo-shard-0 [primary] test>
```