VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

DATABASE MANAGEMENT SYSTEMS

Submitted by

Dhanush H V
(1BM21CS052)

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 DHANUSH HV (1BM21CS052), 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.

Pallavi G B

Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

Index

Sl. No.	Date	Experiment Title	Page No.
1	1/11/22	Insurance Database	3-8
2	8/11/22	More Queries on Insurance Database	8-9
3	15/11/22	Bank Database	10-16
4	29/11/22	More Queries on Bank Database	16-19
5	06/12/22	Employee Database	20-24
6	13/12/22	More Queries on Employee Database	25-30
7	20/12/22	Supplier Database	31-36
8	27/12/22	Flight Database	37-43
9	17/01/23	NoSQL	44-48
10	24/01/23	NO SQL	48-51

Program 1: INSURANCE DATABASE

PERSON (driver-id #: String, name: String, address: String)

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

ACCIDENT (report-number: int, date: date, location: String)

OWNS (driver-id #: String, Regno: String)

PARTICIPATED (driver-id: String, Regno: String, report-number: int, damage-amount: int)

Schema Diagram

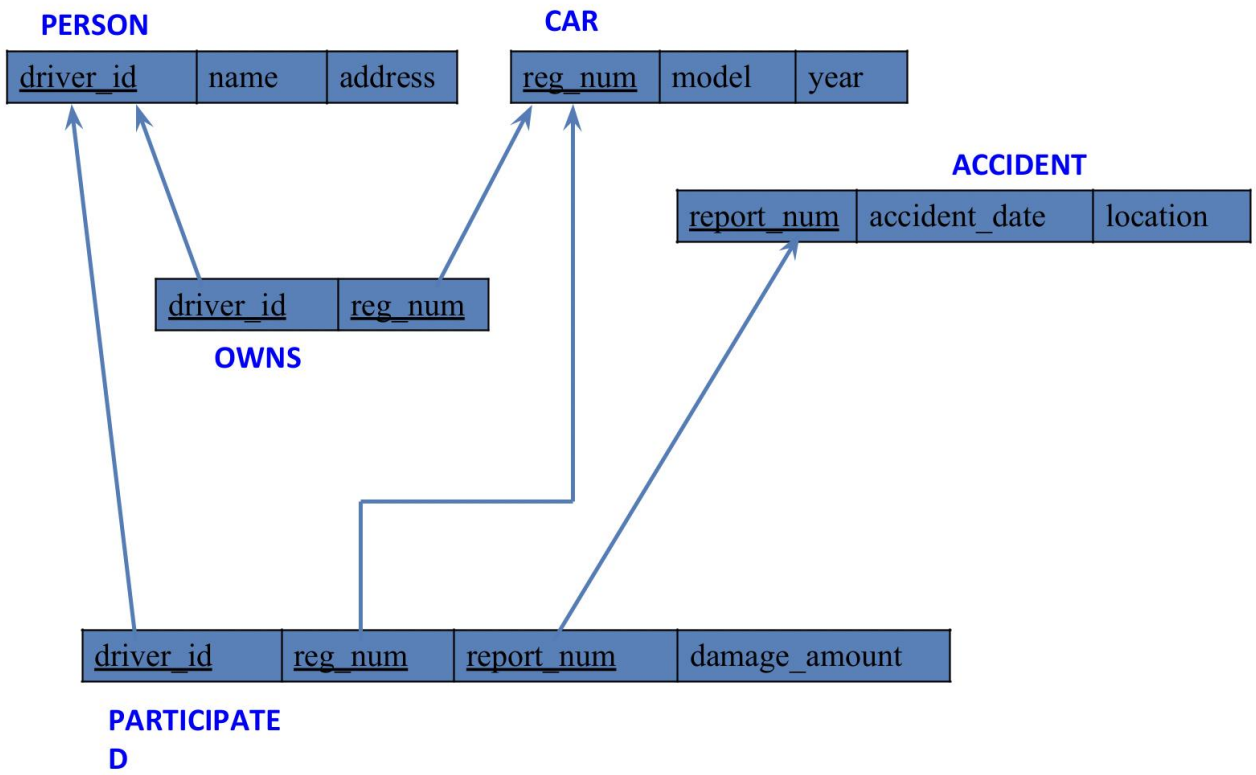
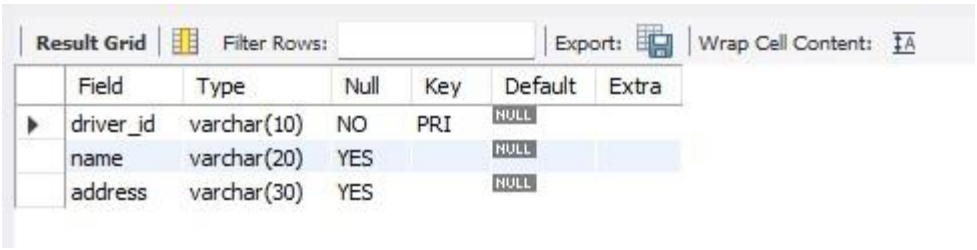


Table Creation:

```
create database 1bm21c052_insurance;
use 1bm21c052_insurance;

create table person (
driveri_d varchar(10),
name varchar(30),
address varchar(30), primary key(driver_id)
desc person);

create table car(
 reg_num varchar(10),
model varchar(10), vear int.
```

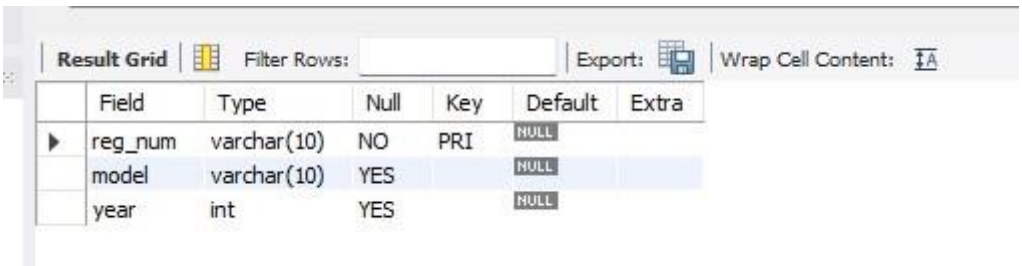
primary key(regnum)
create table accident
report_num int, accident_date date, location varchar(20), primary key(report_num);

create table owns
driver_id varchar(10),
reg_num varchar(10),
primary key(driver_id,reg_num),
foreign key(driver _id)references person(driver_id), foreign key(reg_num)references car(reg_num)
);
create table participated
driver_id varchar(10),
reg_num varchar(10),
reportnum int,
damageamount 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(reportn u m) references accident(reportnum)



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

Field	Type	Null	Key	Default	Extra
driver_id	varchar(10)	NO	PRI	NULL	
name	varchar(20)	YES		NULL	
address	varchar(30)	YES		NULL	

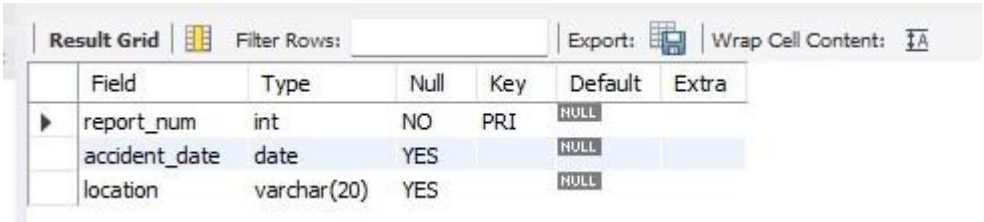


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

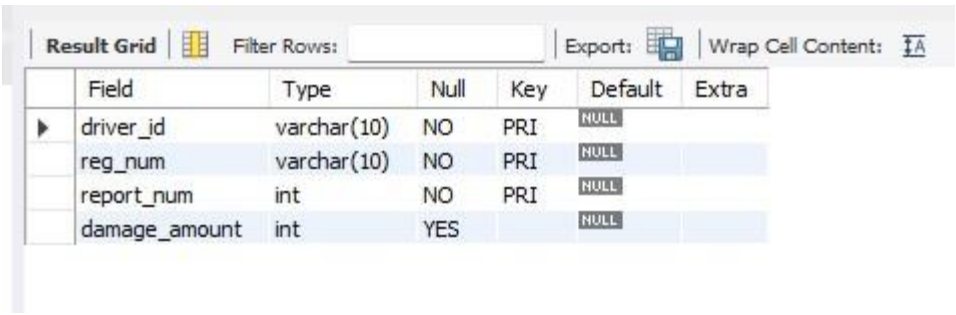
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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



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

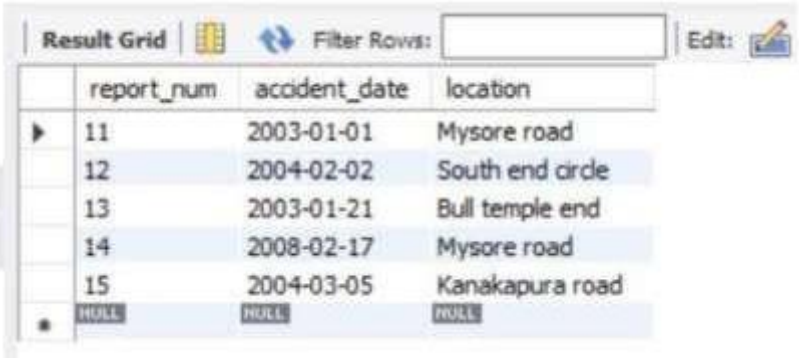
Field	Type	Null	Key	Default	Extra
report_num	int	NO	PRI	NULL	
accident_date	date	YES		NULL	
location	varchar(20)	YES		NULL	



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

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

```
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 road' );
insert into accident values(14,2008-02-17','Mysore road' ;)
insert into accident values(15,2004-03-05','Kanakpura road' );
insert into person values('A01''Richard','Srinivas nagar');
insert into person values(A02', Pradeep'Rajaji nagar');
```



Result Grid | Filter Rows: | Edit:

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
NULL	NULL	NULL

```
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');
```

```
47 •  insert into car values('KA052250','Indica',1990
```

Result Grid | Filter Rows: | Edit:

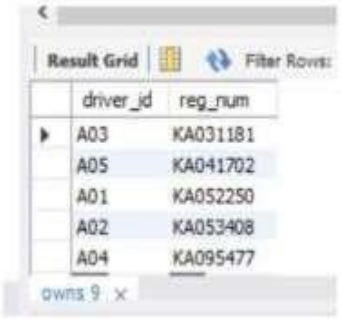
driver_id	name	address
A01	Richard	Srinivas nagar
A02	Pradeep	Rajaji nagar
A03	Smith	Ashok nagar
A04	Venu	N R colony
A05	Jhon	Hanumanth nagar
NULL	NULL	NULL

person 6 ×

Output

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

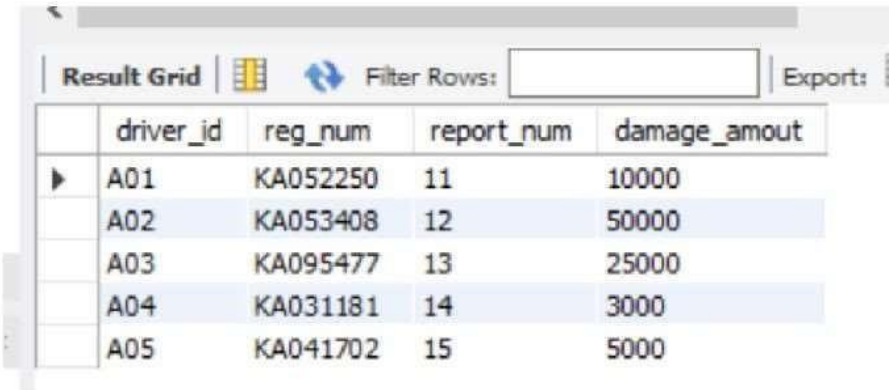
insert into owns values(A01', 'KA052250');
insert into owns values('A02','KA053408');
insert into owns values('A03','KA095477');
insert into owns values (A04', 'KA031181');
insert into owns values(A05','KA041702');



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

owns 9

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



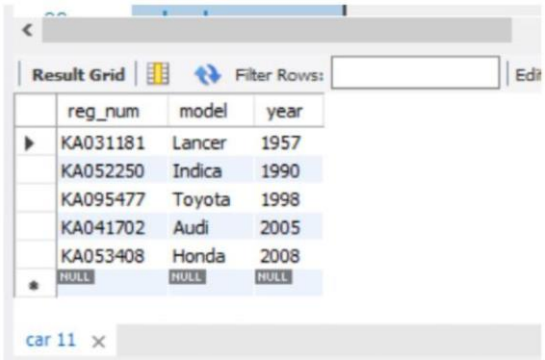
Result Grid | Filter Rows: | Export:

driver_id	reg_num	report_num	damage_amout
A01	KA052250	11	10000
A02	KA053408	12	50000
A03	KA095477	13	25000
A04	KA031181	14	3000
A05	KA041702	15	5000

Queries:

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

SQL> select *from car
order by year asc:



Result Grid | Filter Rows: | Edit

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

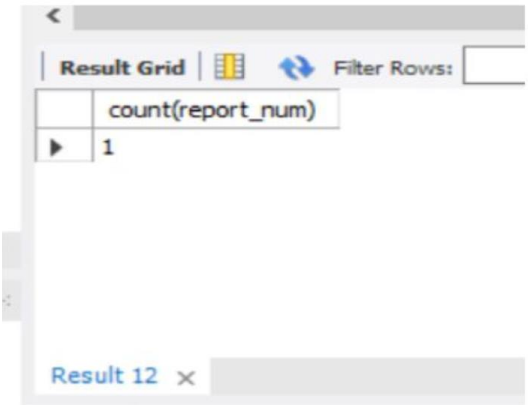
car 11

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

SQL> select count(reportnum)

from car c, participated p

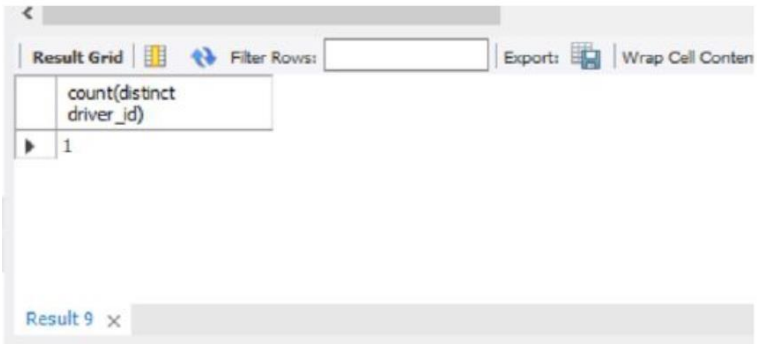
where c.reg_num=p.reg_num and c.model='Lancer";



3.Find the total number of people who owned cars that were involved in accidents in 2008. SQL> select count(distinct driver id)

from participated a, accident b

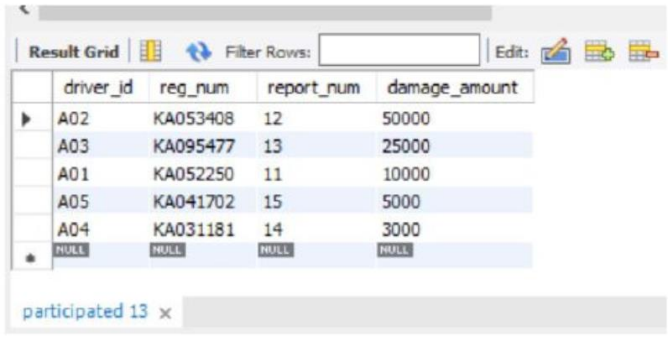
where a.reportnum=b.reportnum and b.accidentdate like '2008%"



TO DO

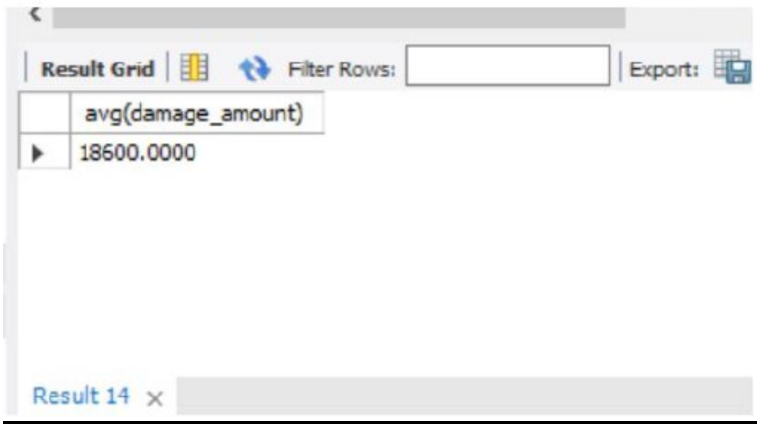
List the entire participated relation in descending order of damage amount. SQL> select *from participated

order by damage_amount desc;



Find the average damage amount.

SQL> select avg(damage_amount) from participated;

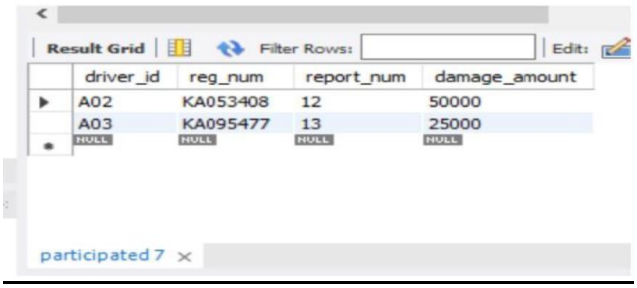


Delete the tuple whose damage amount is below the average damage amount.

SQL> delete from participated

where d a m a g eamount < (select t.avg1 from (select avg (d a m a g eamount) as avg1 from participated) t);

select *from participated;



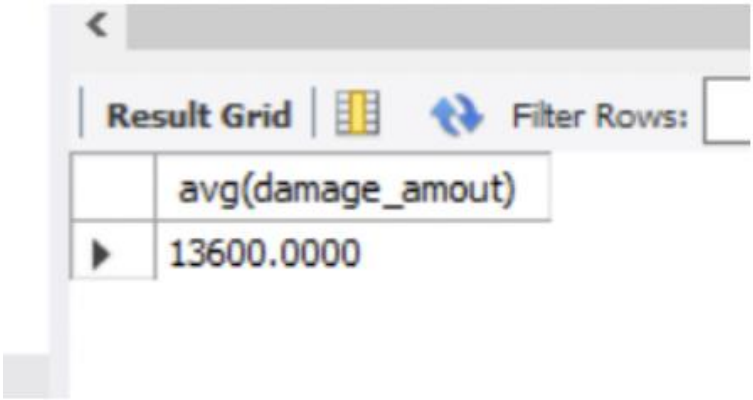
Week 2

More Queries on Insurance Database

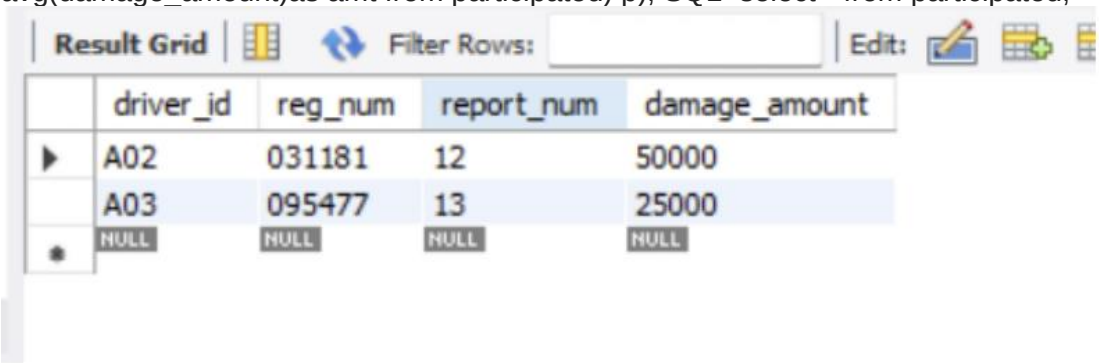
• LIST THE ENTIRE PARTICIPATED RELATION IN THE DESCENDING ORDER OF DAMAGE AMOUNT. SQL>select * from participated order by(damage_amout) desc;

driver_id	reg_num	report_num	damage_amount
A02	031181	12	50000
A03	095477	13	25000
A01	052250	11	10000
A05	041702	15	5000
A04	053408	14	3000
NULL	NULL	NULL	NULL

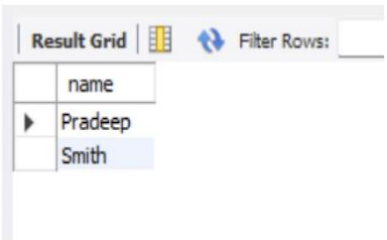
• FIND THE AVERAGE DAMAGE AMOUNT SQL>select avg(damage_amount) from participated;



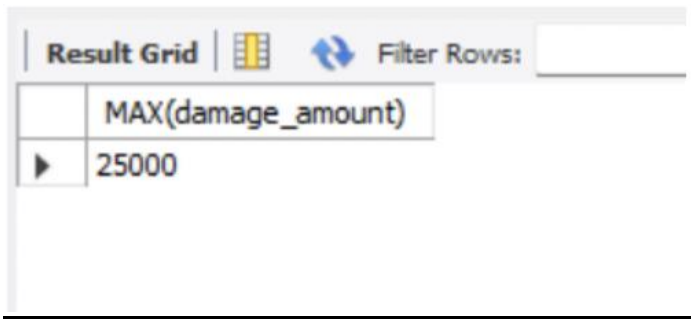
• DELETE THE TUPLE WHOSE DAMAGE AMOUNT IS BELOW THE AVERAGE DAMAGE AMOUNT SQL>delete from participated where damage_amount<(select p.amt from(select avg(damage_amount)as amt from participated) p); SQL>select * from participated;



• LIST THE NAME OF DRIVERS WHOSE DAMAGE IS GREATER THAN THE AVERAGE DAMAGE AMOUNT. SQL>select name from person,participated where person.driver_id=participated.driver_id and damage_amount>(select avg(damage_amount) from participated);



• FIND MAXIMUM DAMAGE AMOUNT. SQL>select * from participated;



Week 3

Program 2: BANKING ENTERPRISE DATABASE

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

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

DEPOSITOR (customer-name: String, customer-street: String, customer-city: String) LOAN (loan-number: int, branch-name: String, amount: real)

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

Schema Diagram

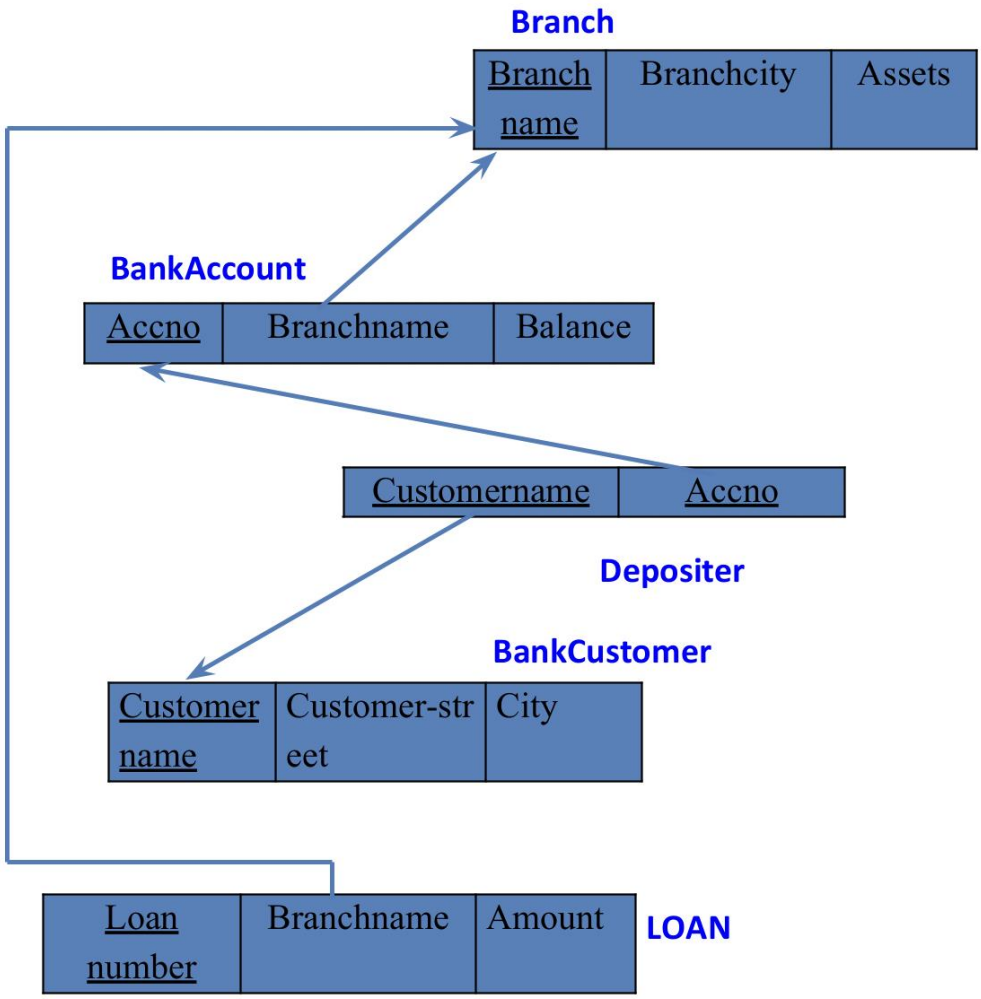
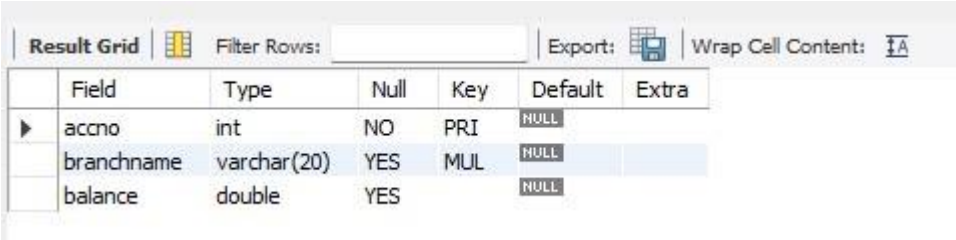


Table Creation:

```
create database 1bm21cs052_Bank1;
use 1bm21cs052_Bank1;

create table branch(
branch_name varchar(20),
branch_city varchar(10),
assets real,
PRIMARY KEY(branch_name)
);
create table bankCustomer(
customer_name varchar(20),
customer_street varchar(20),
customer_city varchar(15),
PRIMARY KEY(customer_name)
);

create table depositer(
customer_name varchar(20),
```

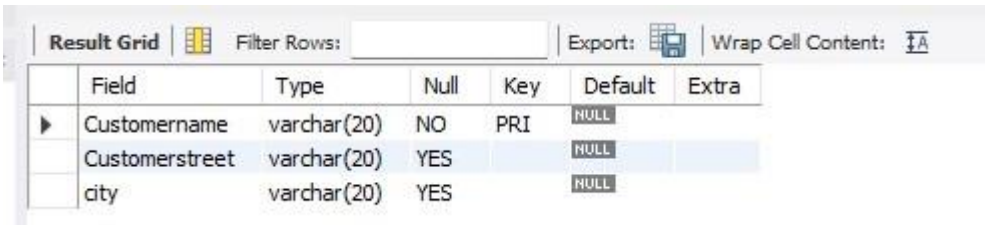
accno int,
FOREIGN KEY(customer_name) REFERENCES bankCustomer(customer_name)
ON UPDATE CASCADE ON DELETE CASCADE,
FOREIGN KEY(accno) REFERENCES bankAccount(accno)
ON UPDATE CASCADE ON DELETE CASCADE
);

create table loan(
loan_no int,
branch_name varchar(20),
amount real,
PRIMARY KEY(loan_no),
FOREIGN KEY(branch_name) REFERENCES branch(branch_name)
ON UPDATE CASCADE ON DELETE CASCADE
);



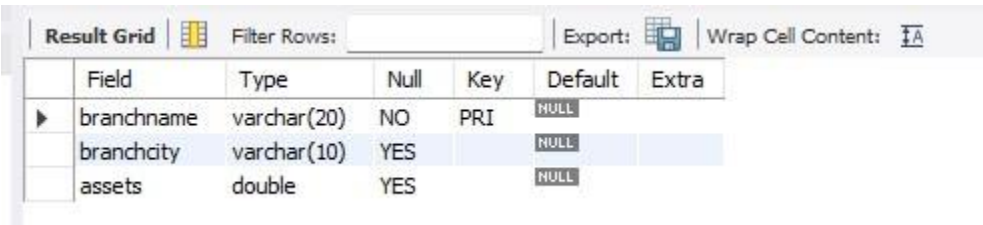
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
accno	int	NO	PRI	NULL	
branchname	varchar(20)	YES	MUL	NULL	
balance	double	YES		NULL	



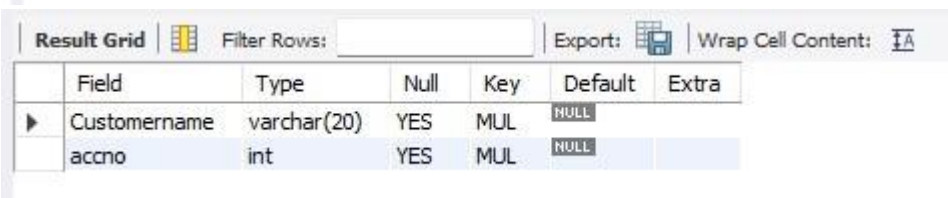
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Customername	varchar(20)	NO	PRI	NULL	
Customerstreet	varchar(20)	YES		NULL	
city	varchar(20)	YES		NULL	



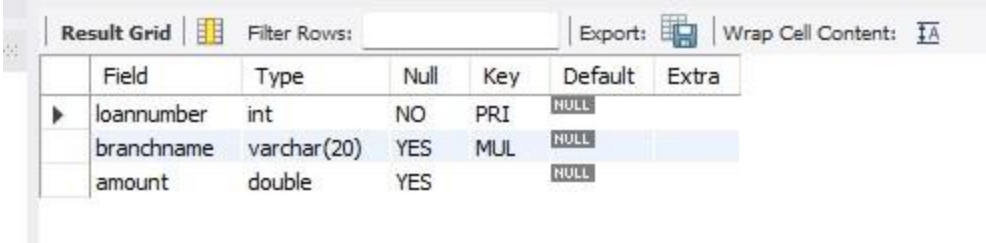
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
branchname	varchar(20)	NO	PRI	NULL	
branchcity	varchar(10)	YES		NULL	
assets	double	YES		NULL	



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

Field	Type	Null	Key	Default	Extra
Customername	varchar(20)	YES	MUL	NULL	
accno	int	YES	MUL	NULL	



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

Field	Type	Null	Key	Default	Extra
loannumber	int	NO	PRI	NULL	
branchname	varchar(20)	YES	MUL	NULL	
amount	double	YES		NULL	

(Inserting values in table)

```
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);

Queries:

(Displaying Tables)

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

• 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

• 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

• Create view which gives each branch the sum of the ammount of all the loans at the branch

create view sum_of_loan

as select branchname,sum(balance) from bankaccount

group by branchname;

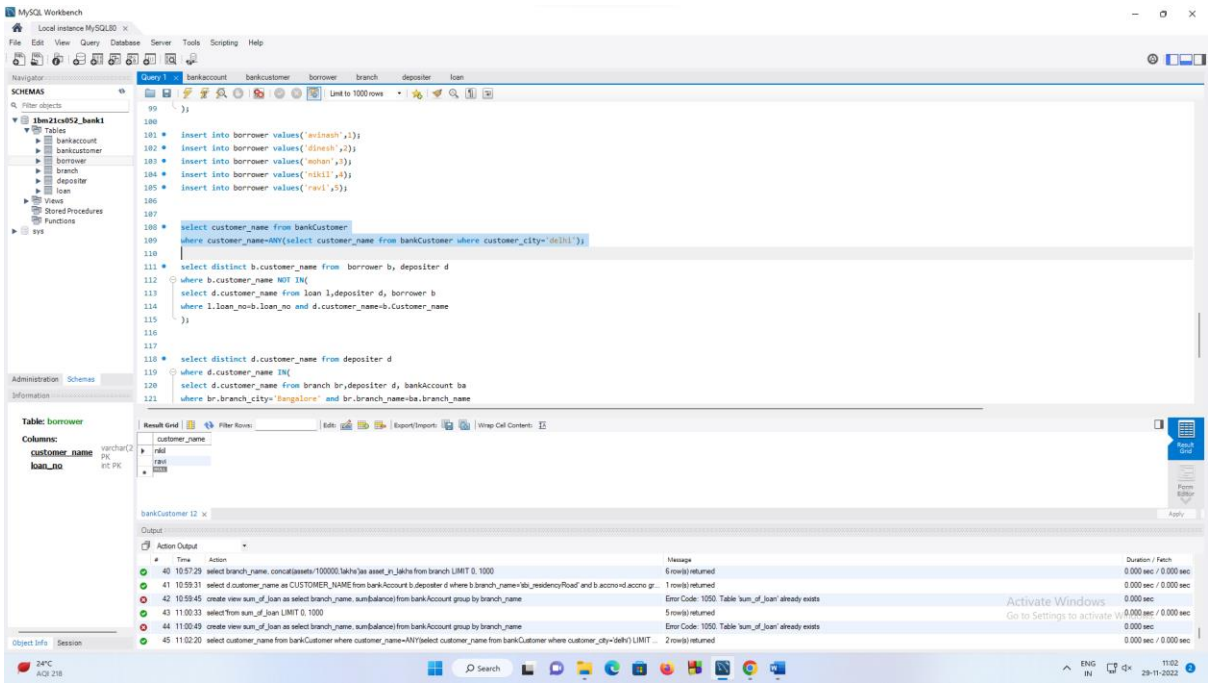
select * from sum_of_loan;

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

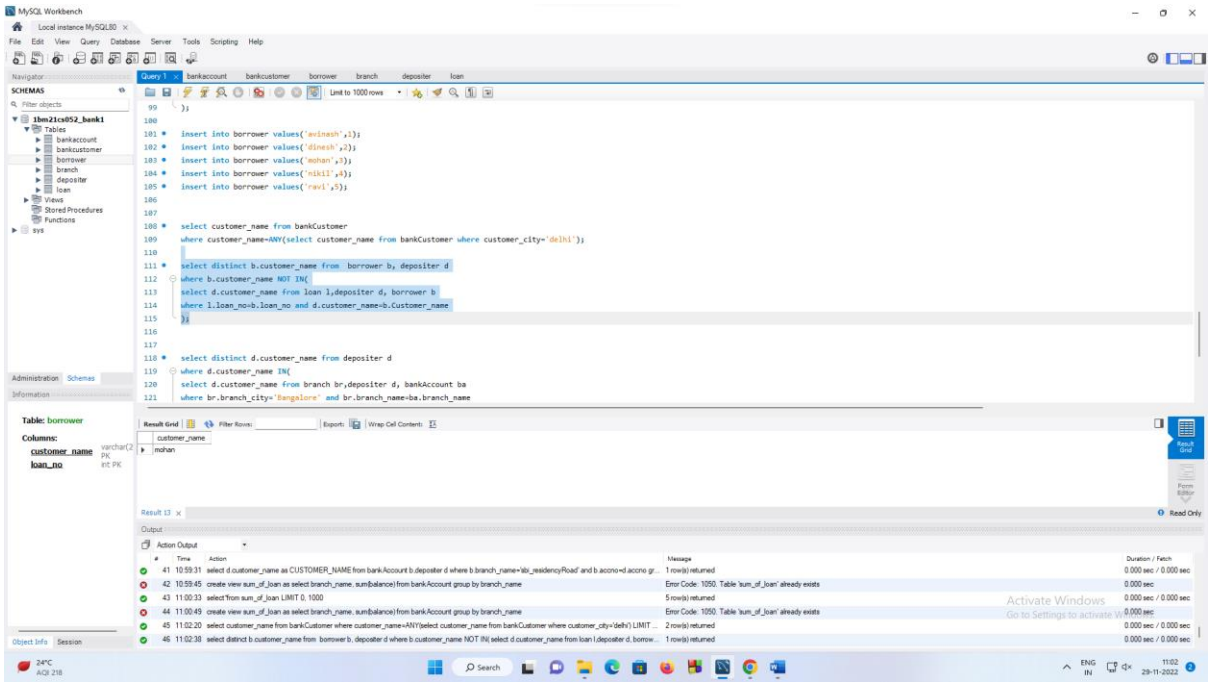
Week 4

More QUERIES ON BANK DATABASE

```
create table borrower(
customer_name varchar(20),
loan_no int,
primary key(customer_name,loan_no),
foreign key(loan_no) references loan(loan_no)
);

insert into borrower values('avinash',1);
insert into borrower values('dinesh',2);
insert into borrower values('mohan',3);
insert into borrower values('nikil',4);
insert into borrower values('ravi',5);


select customer_name from bankCustomer
where customer_name=ANY(select customer_name from bankCustomer where
customer_city='delhi');
```

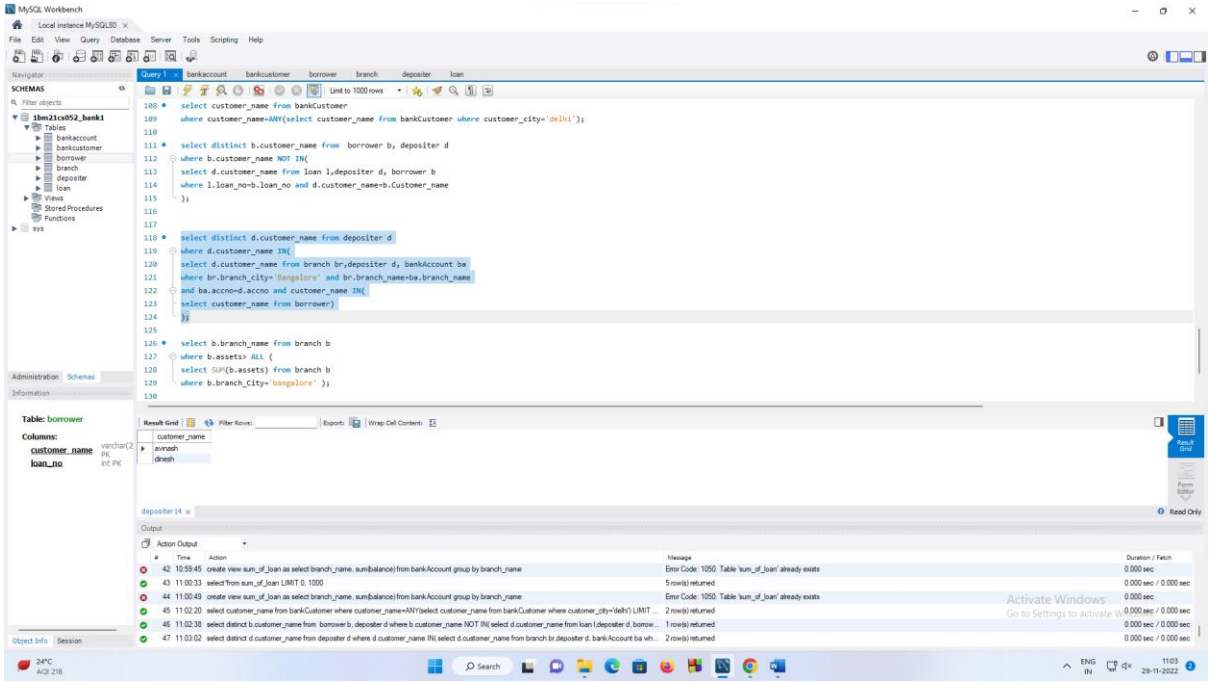


```
select distinct b.customer_name from  borrower b, depositer d
where b.customer_name NOT IN(
select d.customer_name from loan l,depositer d, borrower b
where l.loan_no=b.loan_no and d.customer_name=b.Customer_name
);
```

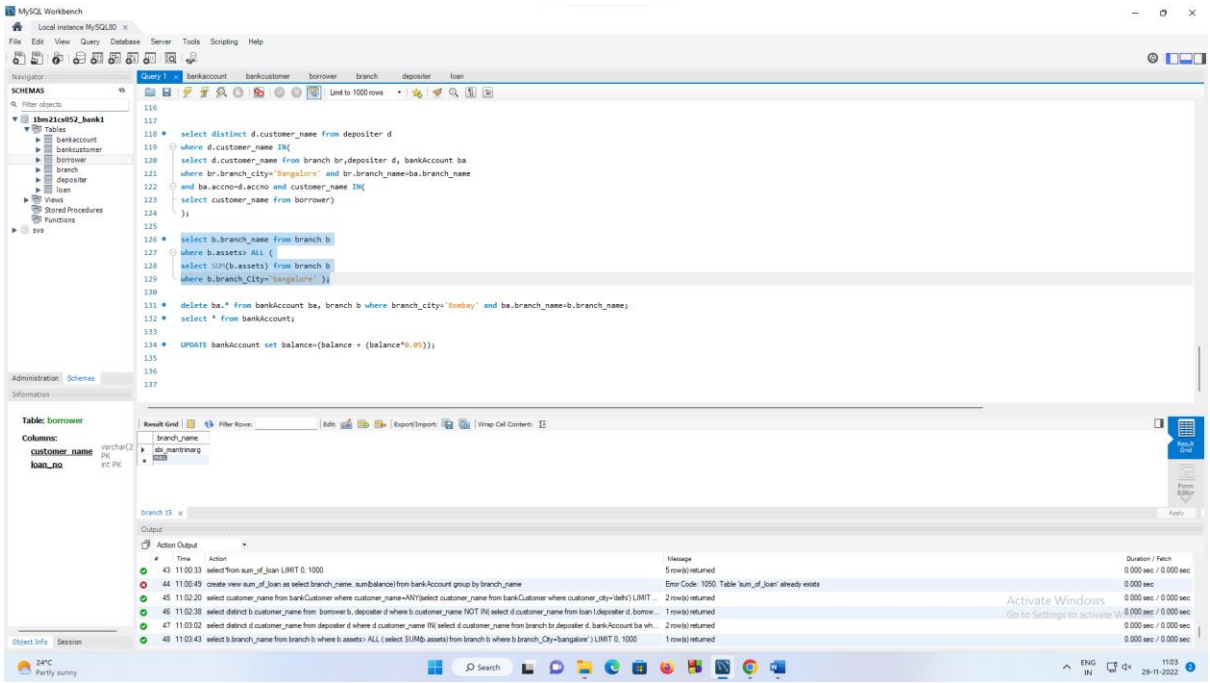


```
select distinct d.customer_name from depositer d
where d.customer_name IN(
select d.customer_name from branch br,depositer d, bankAccount ba
where br.branch_city='Bangalore' and br.branch_name=ba.branch_name
and ba.accno=d.accno and customer_name IN(
select customer_name from borrower)
```

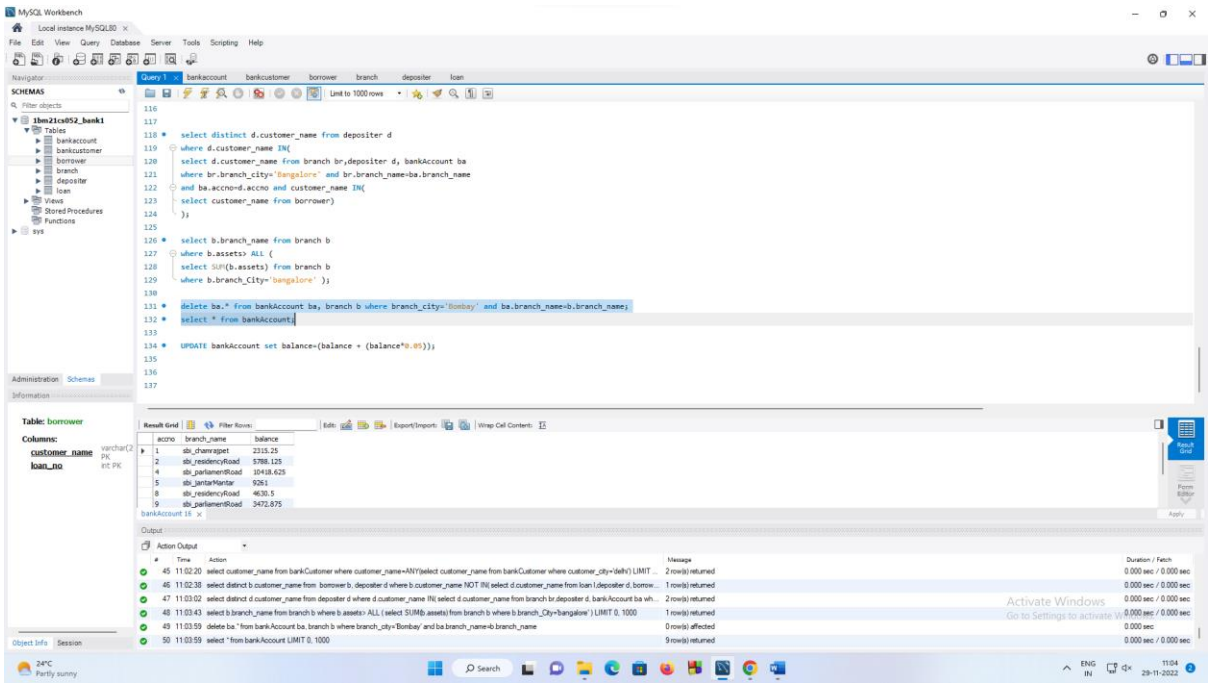
);



select b.branch_name from branch b
where b.assets> ALL (
select SUM(b.assets) from branch b
where b.branch_City='bangalore');

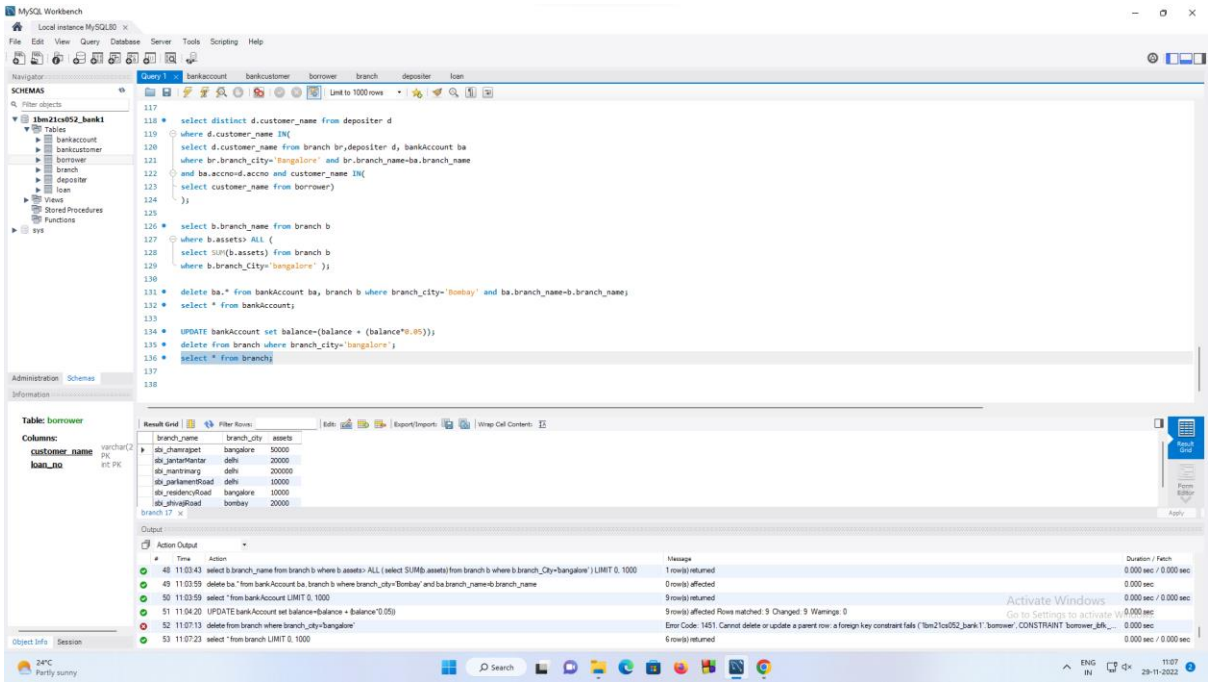


delete ba.* from bankAccount ba, branch b where branch_city='Bombay' and ba.branch_name=b.branch_name;
select * from bankAccount;



UPDATE bankAccount set balance=(balance + (balance*0.05));

delete from branch where branch_city='bangalore';
select * from branch;



Week 5

Program 3:EMPLOYEE DATABASE

Schema Diagram

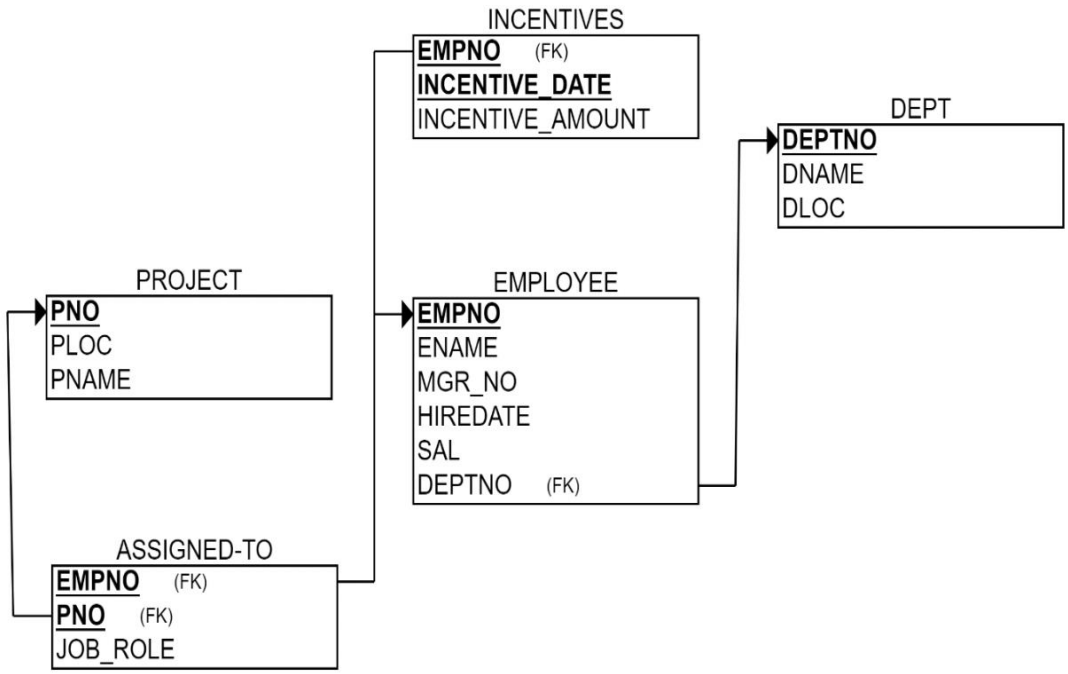
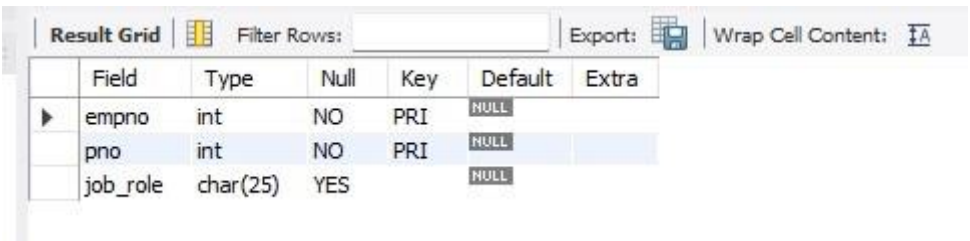


Table Creation:

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

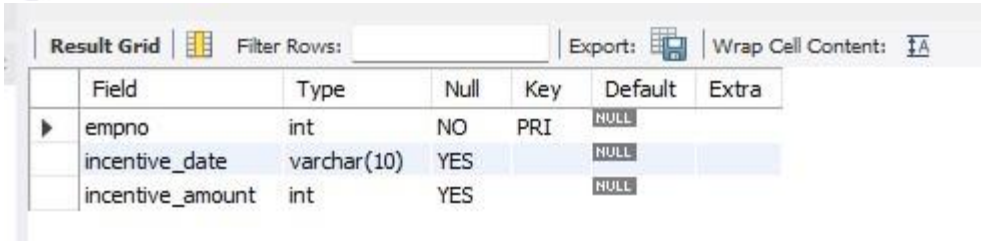


Field	Type	Null	Key	Default	Extra
empno	int	NO	PRI	NULL	
pno	int	NO	PRI	NULL	
job_role	char(25)	YES		NULL	

Field	Type	Null	Key	Default	Extra
deptno	int	NO	PRI	NULL	
dname	char(25)	YES		NULL	
dloc	char(25)	YES		NULL	

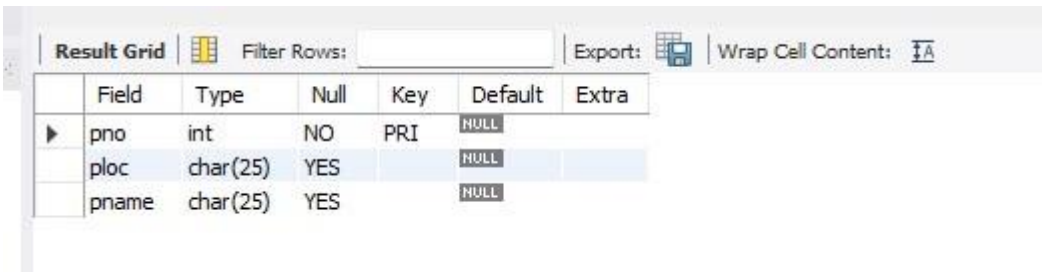
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
empno	int	NO	PRI	NULL	
ename	char(25)	YES		NULL	
mgrno	int	YES		NULL	
hiredate	varchar(10)	YES		NULL	
sal	int	YES		NULL	
deptno	int	NO	PRI	NULL	



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

Field	Type	Null	Key	Default	Extra
empno	int	NO	PRI	NULL	
incentive_date	varchar(10)	YES		NULL	
incentive_amount	int	YES		NULL	



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

Field	Type	Null	Key	Default	Extra
pno	int	NO	PRI	NULL	
ploc	char(25)	YES		NULL	
pname	char(25)	YES		NULL	

Values:

```
insert into dept values(1,'CSE','Siddapur');

insert into dept values(2,'ISE','Sirsi');

insert into dept values(3,'ECE','Pune');

insert into dept values(4,'EEE','Hubli');

insert into dept values(5,'IM','Mumbai');

insert into dept values(6,'BT','Goa');
```

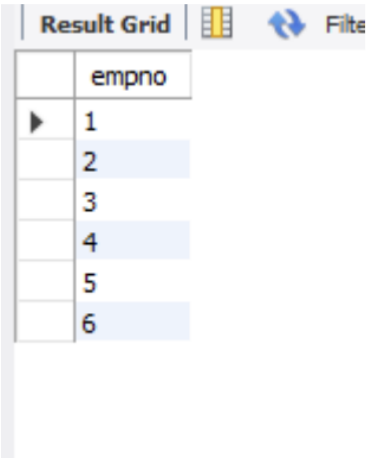
```
insert into emp values(1,'Akash',7,'2020-01-12',20000,4);
insert into emp values(2,'Bhuvan',8,'2020-03-14',40000,1);
insert into emp values(3,'Chris',9,'2021-02-25',35000,2);
insert into emp values(4,'Kiran',10,'2021-11-30',60000,6);
insert into emp values(5,'Pavan',11,'2022-3-10',52000,3);
insert into emp values(6,'Santosh',12,'2022-5-24',48000,5);
select *from emp;
```

```
insert into incentives values(1,'2021-01-12',00000);
insert into incentives values(2,'2021-12-12',8000);
insert into incentives values(3,'2022-06-09',6500);
insert into incentives values(4,'2022-11-29',0000);
insert into incentives values(5,'2023-05-30',9800);
insert into incentives values(6,'2023-08-17',7000);
select *from incentives;

insert into project values(1,'pro1','Bengaluru');
insert into project values(2,'pro2','Pune');
insert into project values(3,'pro3','Hyderabad');
insert into project values(4,'pro4','Kerala');
insert into project values(5,'pro5','Mysuru');
insert into project values(6,'pro6','Sirsi');
select *from project;

insert into assigned_to values(1,1,'Developer');

insert into assigned_to values(2,2,'Assistant');

insert into assigned_to values(3,3,'Co-assistant');

insert into assigned_to values(4,4,'Organiser');

insert into assigned_to values(5,5,'Manager');

insert into assigned_to values(6,6,'Supervisor');

select *from assigned_to;
```

Queries:

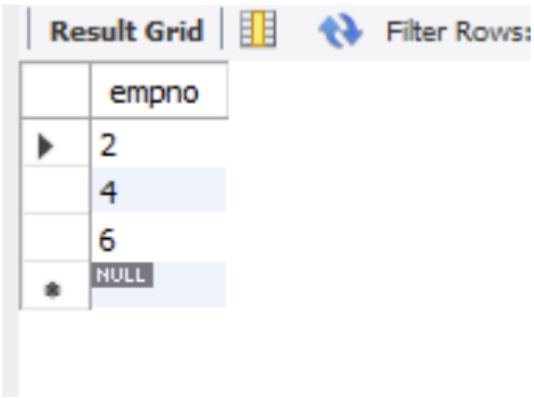
- Retrieve the employee numbers of all employees who work on project located in Banglore, Hyderbad, or Mysore

```
select emp_no from assigned_to a where p_no in (select p_no from project where p_loc in ('Bengaluru','Hyderabad','Mysuru'));
```



- Get Employee ID’s of those employees who didn’t receive incentives

select emp_no,e_name from emp where emp_no in (select emp_no from incentives where incentive_amount=0);



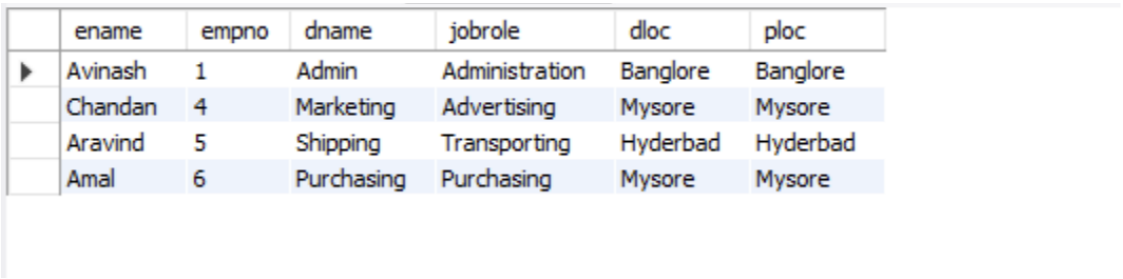
empno
2
4
6
NULL

• 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.

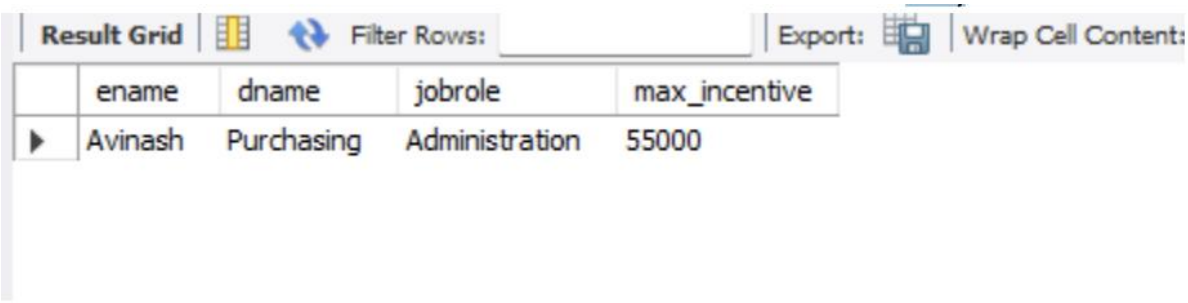
select e.e_name, e.emp_no,d.d_name,a.job_role,d.d_loc,p.p_loc from emp e,dept d,assigned_to a,project p where d.d_loc=p.p_loc and e.emp_no=a.emp_no and a.p_no=p.p_no;



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

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

select e.e_name,max(i.incentive_amount),d.d_name,a.job_role from emp e,dept d,assigned_to a,incentives i where e.emp_no in(select emp_no from incentives where incentive_amount= (select max(incentive_amount) from incentives where incentive_date between'2022-01-01'and '2022-12-31')) and d.dept_no=e.dept_no and a.emp_no=e.emp_no;



ename	dname	jobrole	max_incentive
Avinash	Purchasing	Administration	55000

More Queries on Employee Database

week 6

TO DO

1.Using Scheme diagram, Create tables by properly specifying the primary keys and the foreign keys.

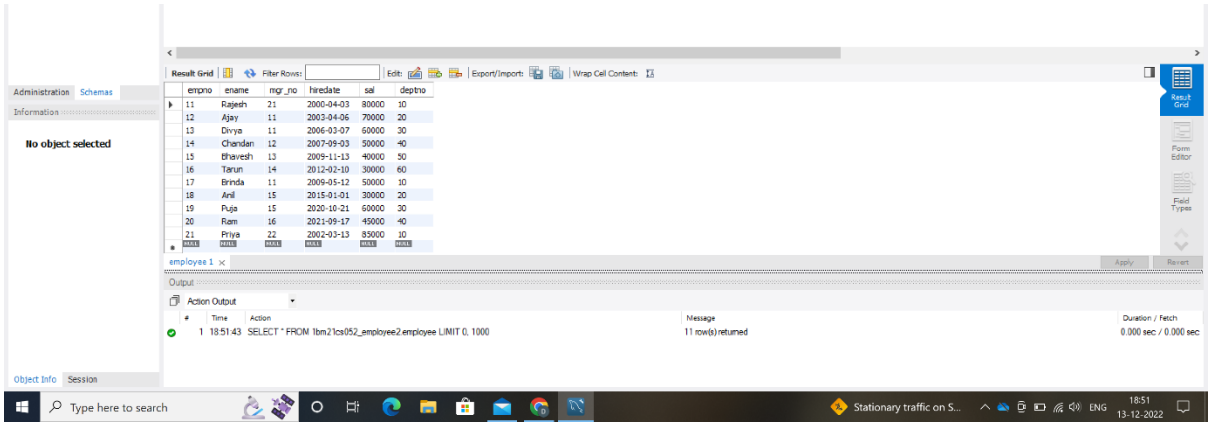
```
create database 1bm21cs052_employee2;
use 1bm21cs052_employee2;

create table dept(
deptno int,
dname varchar(20),
dloc varchar(20),
primary key(deptno)
);
create table employee(
empno int,
ename varchar(20),
mgr_no int,
hiredate date,
sal double,
deptno int,
primary key(empno),
foreign key (deptno) references dept(deptno)
on delete cascade
on update cascade
);
create table incentives(
empno int,
incentive_date date,
incentive_amount float,
primary key(empno,incentive_date),
foreign key (empno) references employee(empno)
on delete cascade
on update cascade
);
create table project(
pno int,
ploc varchar(20),
pname varchar(20),
primary key(pno)
);
create table assigned_to(
empno int,
pno int,
job_role varchar(20),
```

```
primary key(empno,pno),
foreign key (empno) references employee(empno),
foreign key (pno) references project(pno)
on delete cascade
on update cascade
);
```

2. Enter greater than five tuples for each table.

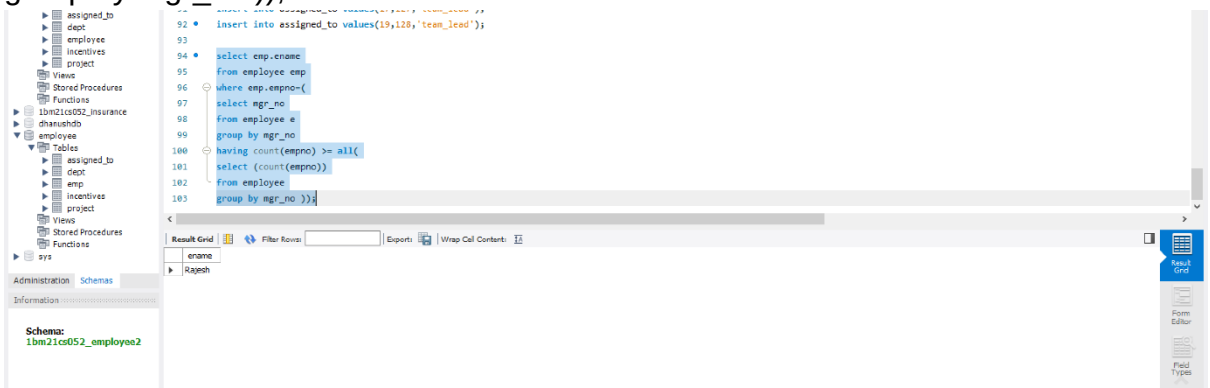
```
insert into dept values(10,'cse','bangalore');
insert into dept values(20,'ise','bangalore');
insert into dept values(30,'aiml','hyderabad');
insert into dept values(40,'ece','mysore');
insert into dept values(50,'eee','delhi');
insert into dept values(60,'iem','chennai');

insert into employee values(11,'Rajesh',21,'2000-04-03',80000,10);
insert into employee values(12,'Ajay',11,'2003-04-06',70000,20);
insert into employee values(13,'Divya',11,'2006-03-07',60000,30);
insert into employee values(14,'Chandan',12,'2007-09-03',50000,40);
insert into employee values(15,'Bhavesh',13,'2009-11-13',40000,50);
insert into employee values(16,'Tarun',14,'2012-02-10',30000,60);
insert into employee values(17,'Brinda',11,'2009-05-12',50000,10);
insert into employee values(18,'Anil',15,'2015-01-01',30000,20);
insert into employee values(19,'Puja',15,'2020-10-21',60000,30);
insert into employee values(20,'Ram',16,'2021-09-17',45000,40);
insert into employee values(21,'Priya',22,'2002-03-13',85000,10);

insert into incentives values(11,'2012-09-08',40000);
insert into incentives values(12,'2015-07-10',33000);
insert into incentives values(13,'2019-01-21',7000);
insert into incentives values(14,'2019-01-05',8000);
insert into incentives values(15,'2019-01-13',5000);
insert into incentives values(17,'2021-03-17',6000);
insert into incentives values(18,'2021-04-16',8000);
insert into incentives values(19,'2021-08-11',9000);

insert into project values(121,'bangalore','proj1');
insert into project values(122,'bangalore','proj2');
insert into project values(123,'mysore','proj3');
insert into project values(124,'hyderabad','proj4');
insert into project values(125,'delhi','proj5');
insert into project values(126,'mumbai','proj6');
insert into project values(127,'calicut','proj7');
insert into project values(128,'calicut','proj8');

insert into assigned_to values(11,121,'manager');
```

```
insert into assigned_to values(12,122,'team_lead');
insert into assigned_to values(13,123,'analyst');
insert into assigned_to values(14,124,'team_lead');
insert into assigned_to values(15,125,'manager');
insert into assigned_to values(16,126,'programmer');
insert into assigned_to values(17,127,'team_lead');
insert into assigned_to values(19,128,'team_lead');
```



3. List the name of the managers with the maximum employees

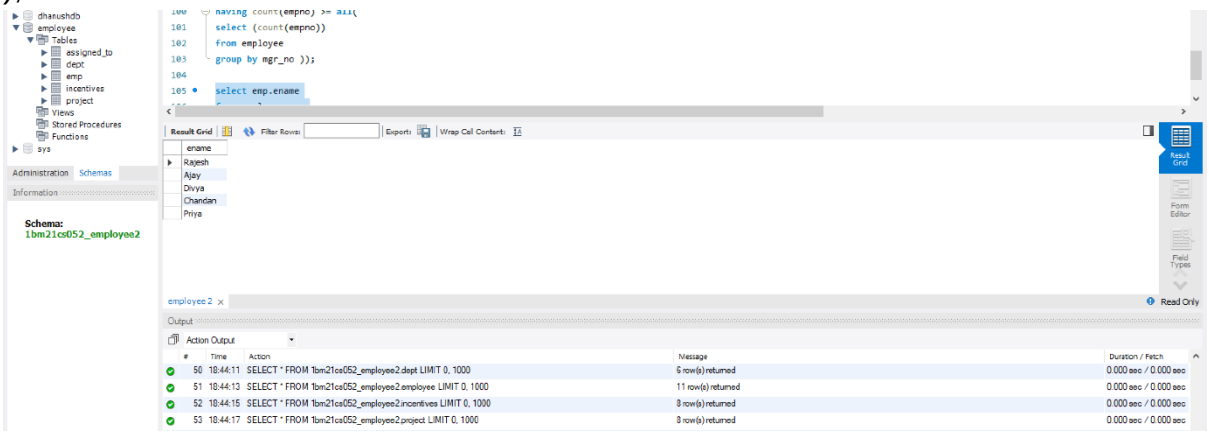
```
select emp.ename
from employee emp
where emp.empno=(
select mgr_no
from employee e
group by mgr_no
having count(empno) >= all(
select (count(empno))
from employee
group by mgr_no ));
```



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

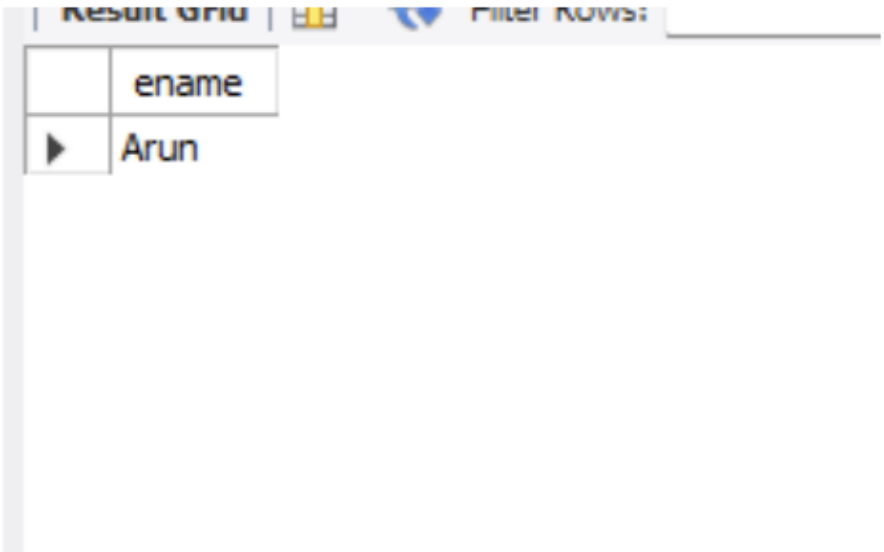
```
select emp.ename
from employee emp
where emp.sal > any (
select avg(e.sal)
```

from employee e
where emp.empno=e.mgr_no
);



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

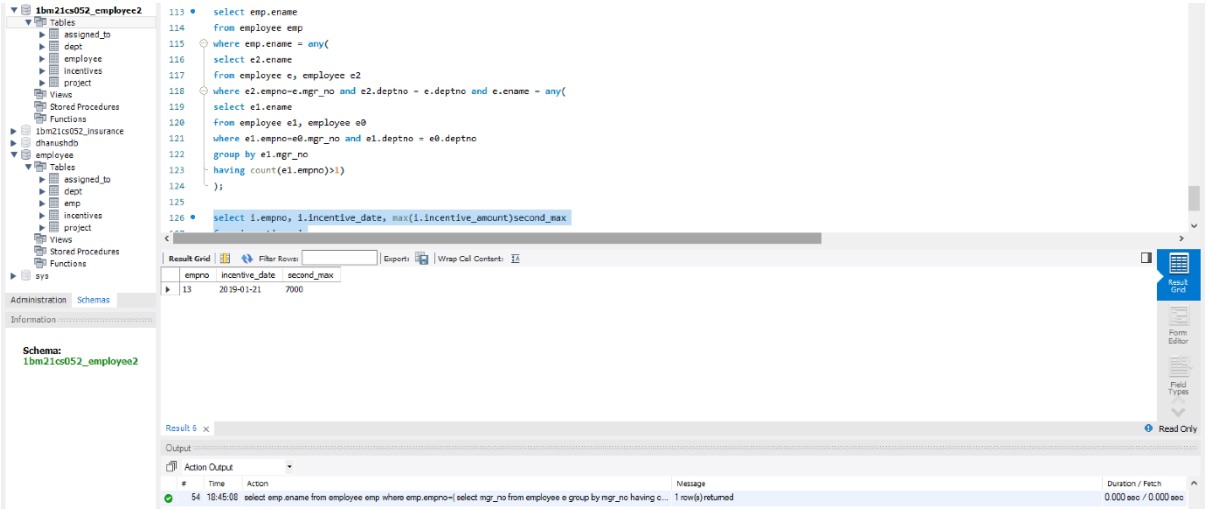
```
select emp.ename
from employee emp
where emp.ename = any(
select e2.ename
from employee e, employee e2
where e2.empno=e.mgr_no and e2.deptno = e.deptno and e.ename = any(
select e1.ename
from employee e1, employee e0
where e1.empno=e0.mgr_no and e1.deptno = e0.deptno
group by e1.mgr_no
having count(e1.empno)>1)
);
```



6. Find the employee details who got second maximum incentive in January 2019.

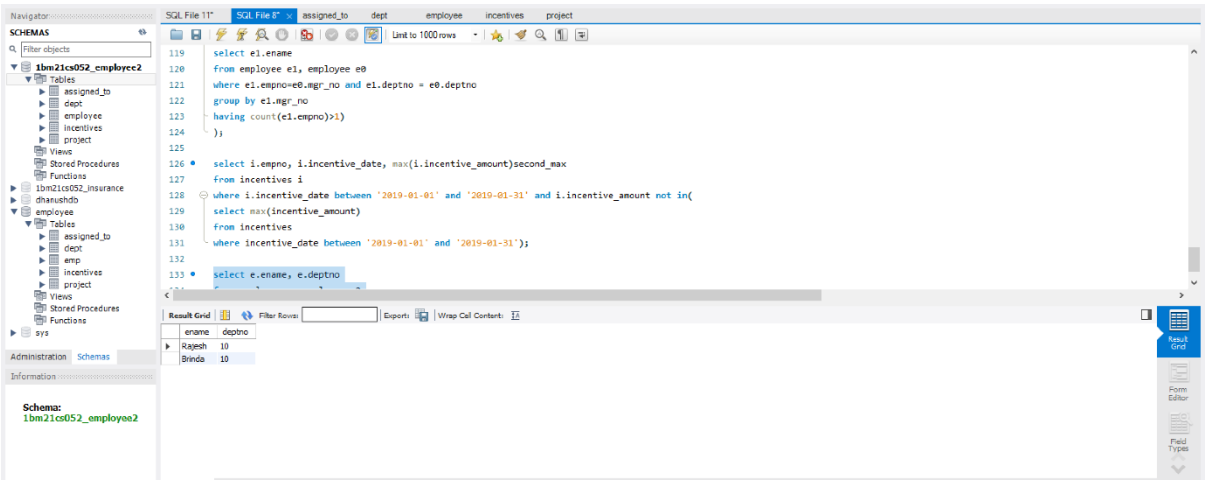
```
select i.empno, i.incentive_date, max(i.incentive_amount)second_max
```

from incentives i
where i.incentive_date between '2019-01-01' and '2019-01-31' and i.incentive_amount not in(
select max(incentive_amount)
from incentives
where incentive_date between '2019-01-01' and '2019-01-31');



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

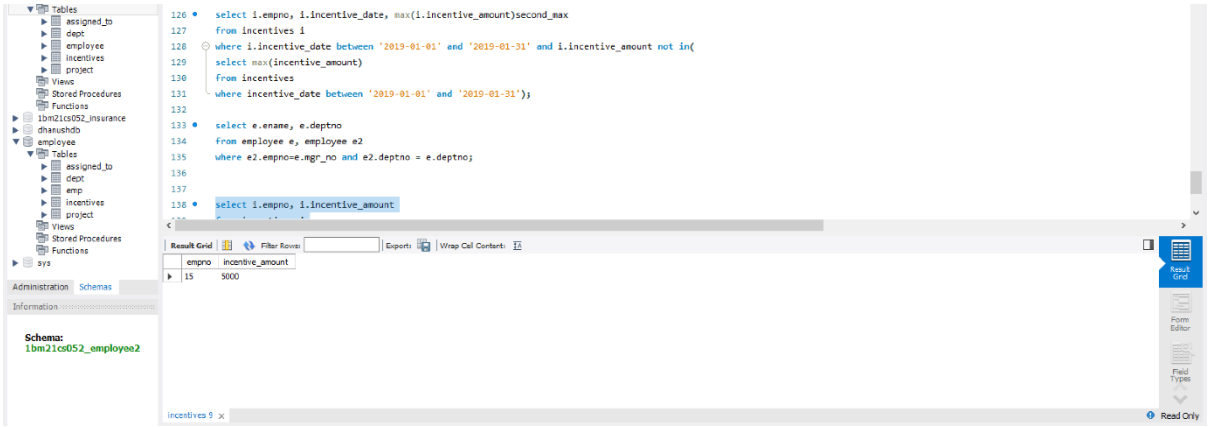
select e.ename, e.deptno
from employee e, employee e2
where e2.empno=e.mgr_no and e2.deptno = e.deptno;



Spot query-Find the employee details who got third maximum incentive in January 2019

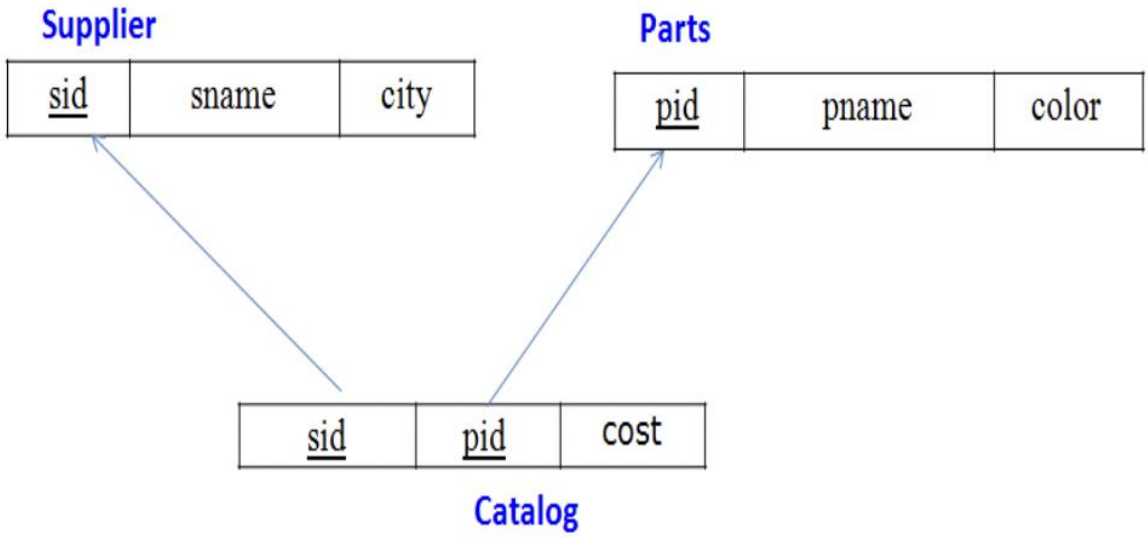
select i.empno, i.incentive_amount
from incentives i
where 3 = (
select count(*)
from incentives j
where incentive_date between '2019-01-01' and '2019-01-31' and i.incentive_amount <=

j.incentive_amount)
and incentive_date between '2019-01-01' and '2019-01-31';



WEEK 7
Program 5: SUPLLIER

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));
```

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

Field	Type	Null	Key	Default	Extra
sid	int	NO	PRI	NULL	
sname	char(25)	YES		NULL	
city	char(25)	YES		NULL	

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

Field	Type	Null	Key	Default	Extra
pid	int	NO	PRI	NULL	
pname	char(15)	YES		NULL	
color	char(15)	YES		NULL	

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

Field	Type	Null	Key	Default	Extra
sid	int	NO	PRI	NULL	
pid	int	NO	PRI	NULL	
cost	int	YES		NULL	

Values:

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

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

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

Queries:

```
select * from supplier;
select * from parts;
```

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

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

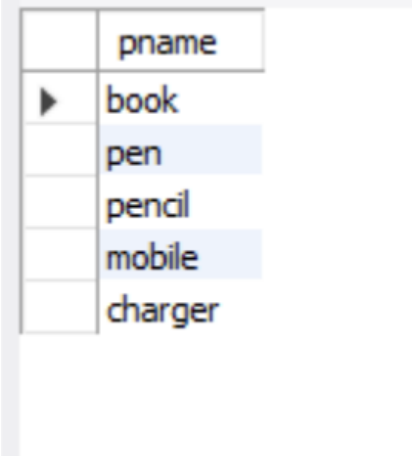
select * from catalog;

	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

supplier 7 parts 8 catalog 9 ×

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

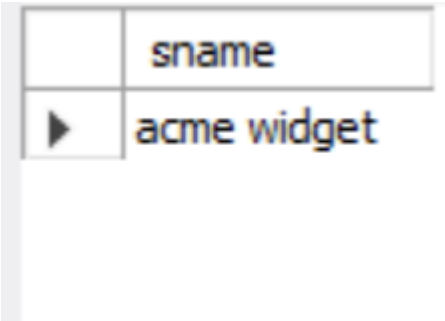
select pname from parts where pid in (select pid from catalog)



	pname
▶	book
	pen
	pencil
	mobile
	charger

● Find the snames of suppliers who supply every part.

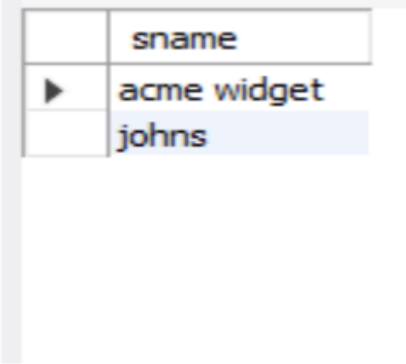
```
select sname from

(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

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

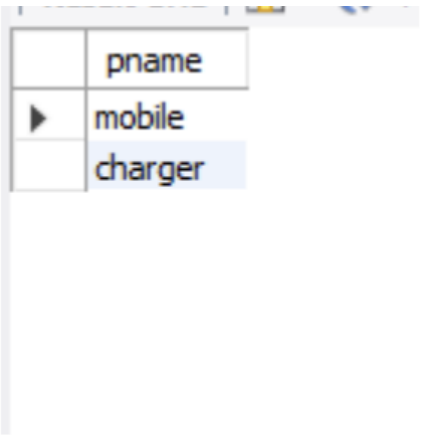
```
select sname from supplier
where sid in( select sid from catalog where pid in( select pid from parts where
colour='red'));
```



	sname
▶	acme widget
	johns

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

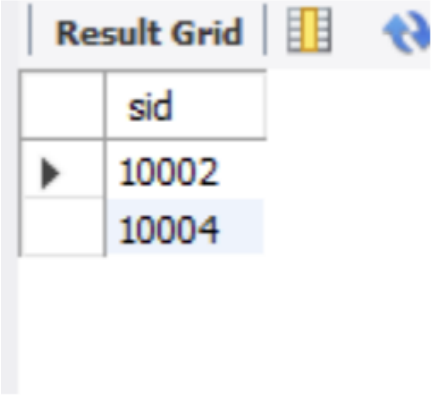
```
select pname from parts

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

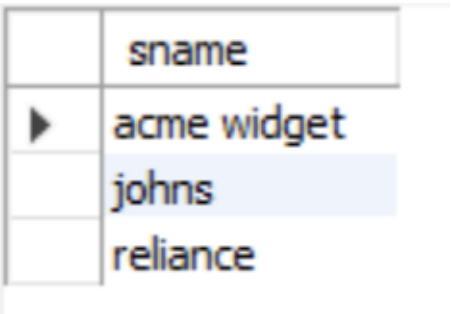
● Find the sids of suppliers who charge more for some part than the average cost of that part.

```
 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);
```



● 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));
```



WEEK 8

Program 6 week 8:FLIGHT

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

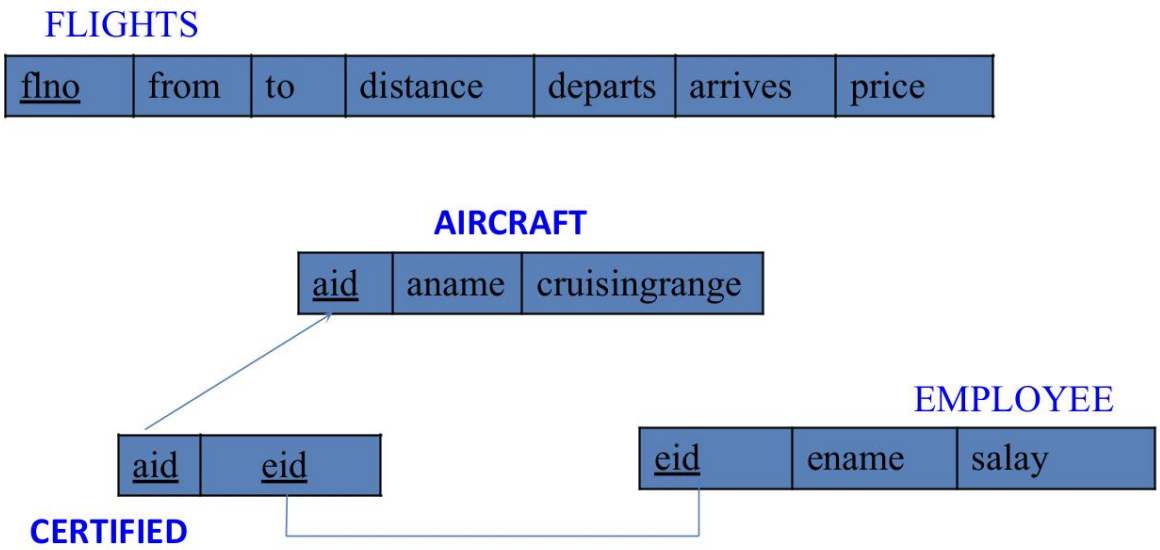
AIRCRAFT(aid: integer, aname: string, cruising_range: integer)

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.

SCHEMADIAGRAM



DBMS WEEK 8 create database Airline;

use Airline;

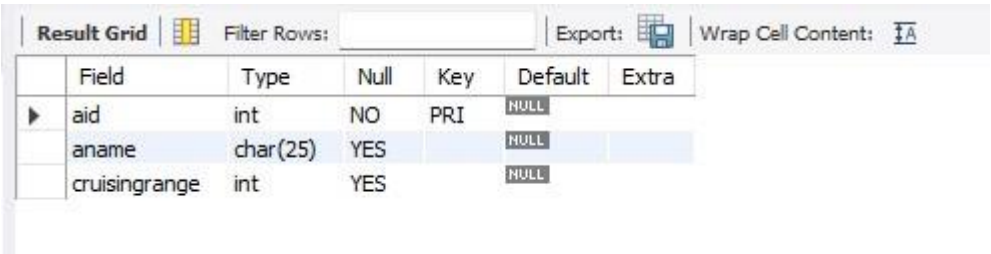
create table flights(

flno int, ffrom varchar(50), tto varchar(50),

distance int, departs time,

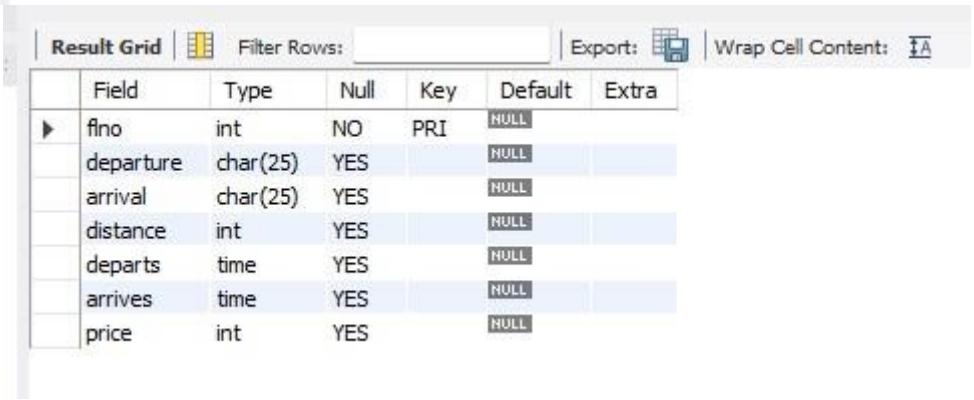
arrives time, price int, primary key(flno));

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

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

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);
```



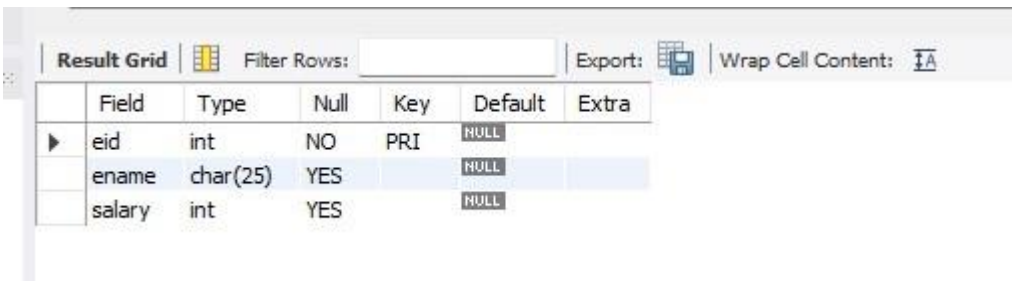
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
aid	int	NO	PRI	NULL	
aname	char(25)	YES		NULL	
cruisingrange	int	YES		NULL	



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

Field	Type	Null	Key	Default	Extra
flno	int	NO	PRI	NULL	
departure	char(25)	YES		NULL	
arrival	char(25)	YES		NULL	
distance	int	YES		NULL	
departs	time	YES		NULL	
arrives	time	YES		NULL	
price	int	YES		NULL	



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

Field	Type	Null	Key	Default	Extra
eid	int	NO	PRI	NULL	
ename	char(25)	YES		NULL	
salary	int	YES		NULL	

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

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

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

```
select * from employee;
```

Result Grid | Filter Rows: | Edit:

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

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

```
select * from aircraft;
```

Result Grid | Filter Rows: | Edit:

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

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),(104,6), (104,1),(104,3),(105,3);

Result Grid | Filter Rows:

eid	aid
101	2
101	4
101	5
101	6
102	1
102	3
102	5
103	2
103	3
103	5
103	6
104	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);
select * from flights;

Result Grid | Filter Rows: | Edit: | Export/Import:

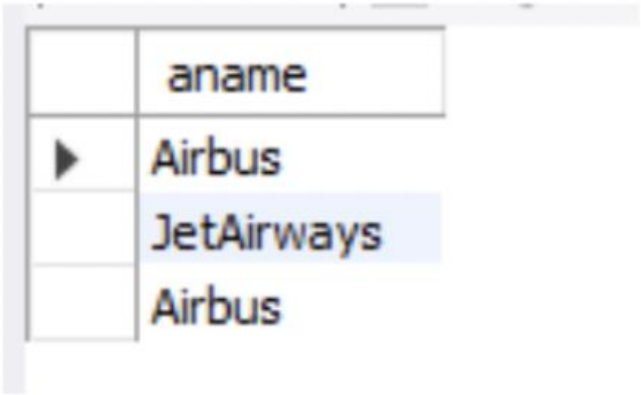
fino	from_place	to_place	distance	departs	arrives	price
1	Bengaluru	New Delhi	500	06:00:00	09:00:00	5000
2	Bengaluru	Chennai	300	07:00:00	08:30:00	3000
3	Trivandrum	New Delhi	800	08:00:00	11:30:00	6000
4	Bengaluru	Frankfurt	1000	06:00:00	23:30:00	50000
5	Kolkata	New Delhi	2400	11:00:00	03:30:00	9000
6	Bengaluru	Frankfurt	8000	09:00:00	23:00:00	40000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

flights 16 ×

Queries:

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

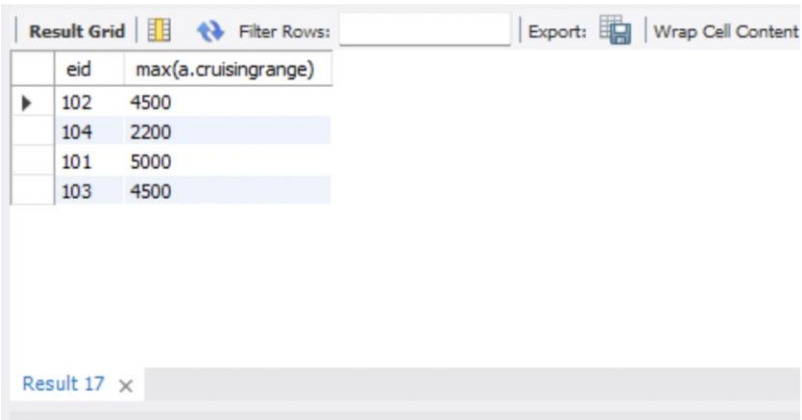
select a.aname from aircraft a,employee s, certified c where s.salary >80000 and a.aid=c.aid and c.eid=s.eid;



aname
Airbus
JetAirways
Airbus

ii. 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(cruisingrange) FROM certified c,aircraft a WHERE c.aid=a.aid



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

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

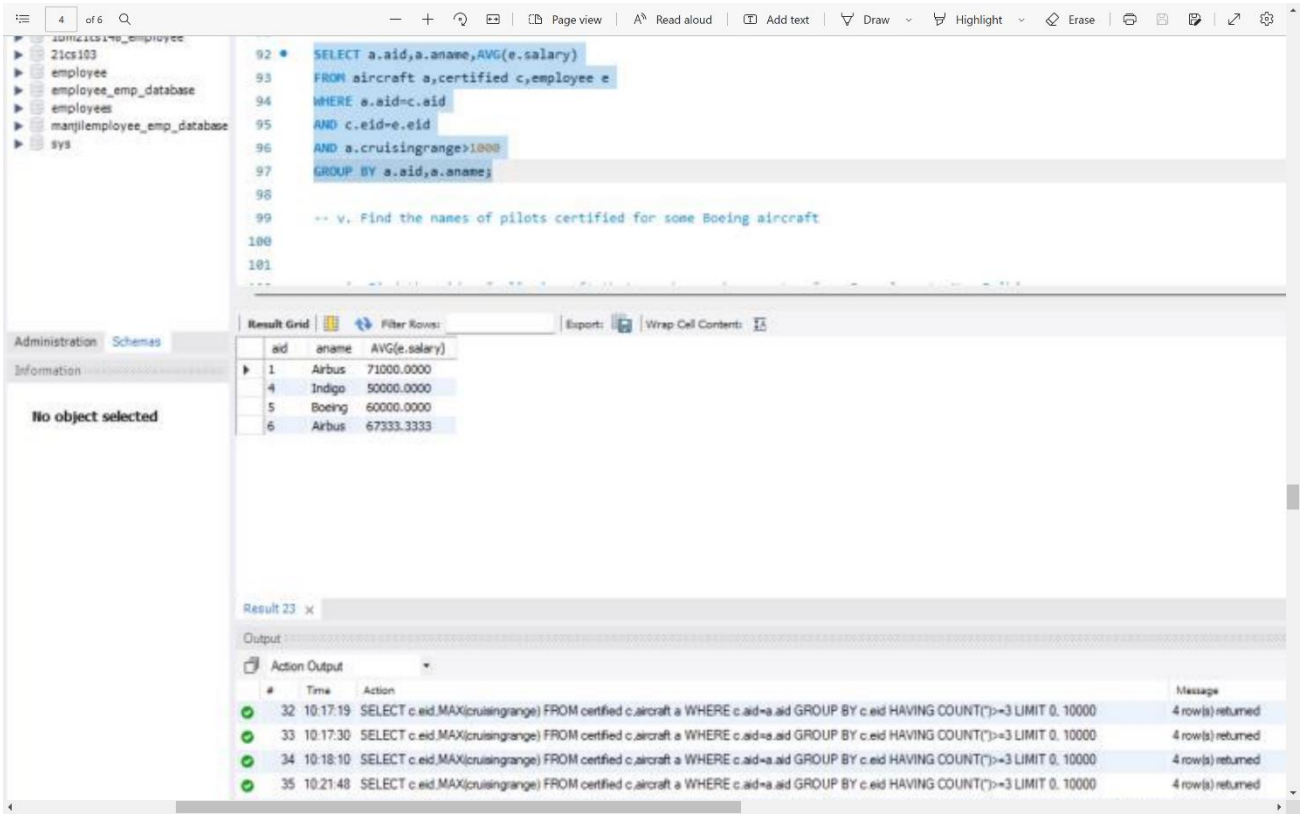
Result 17

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

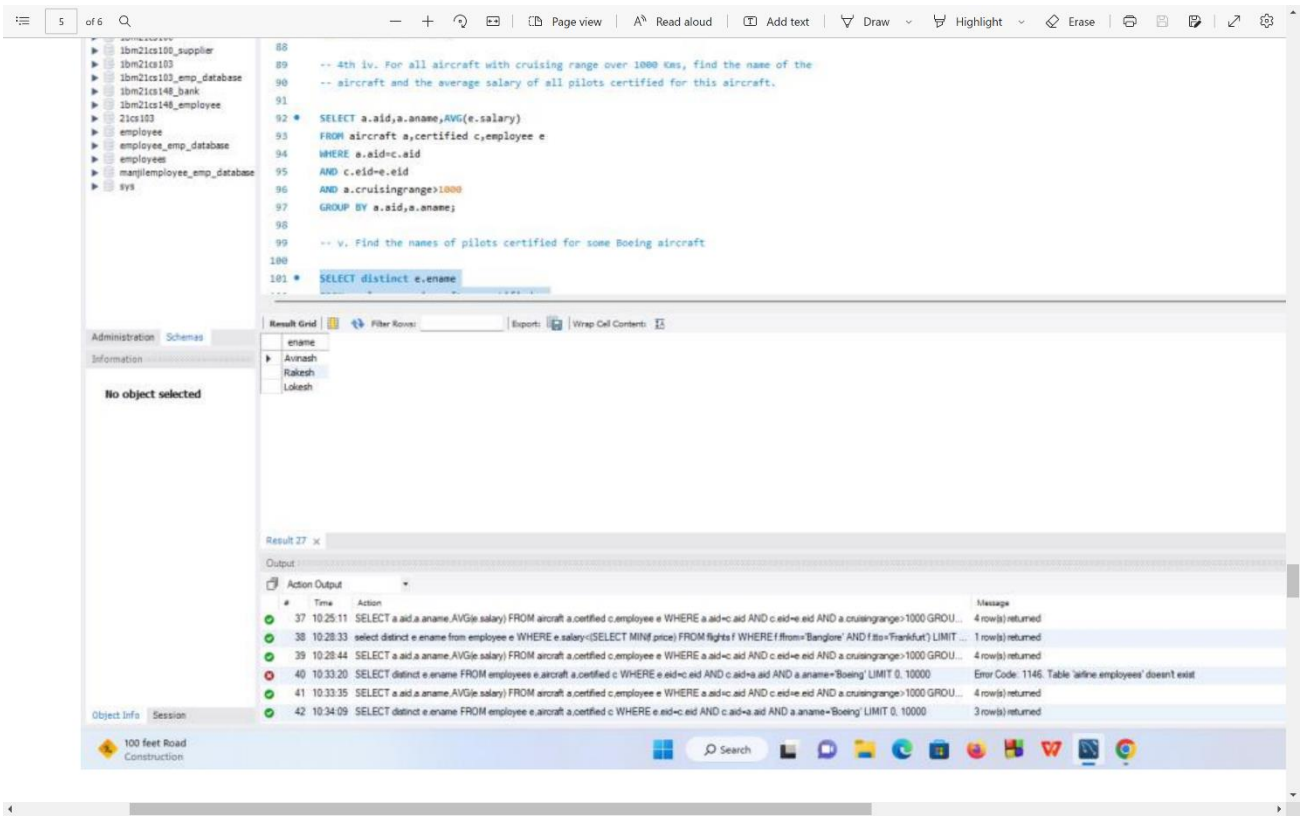
select e.ename from employee e where e.salary<(select min(f.price) from flights f where f.ffrom='Banglore' and f.tto='Frankfurt');

ename
Tilak

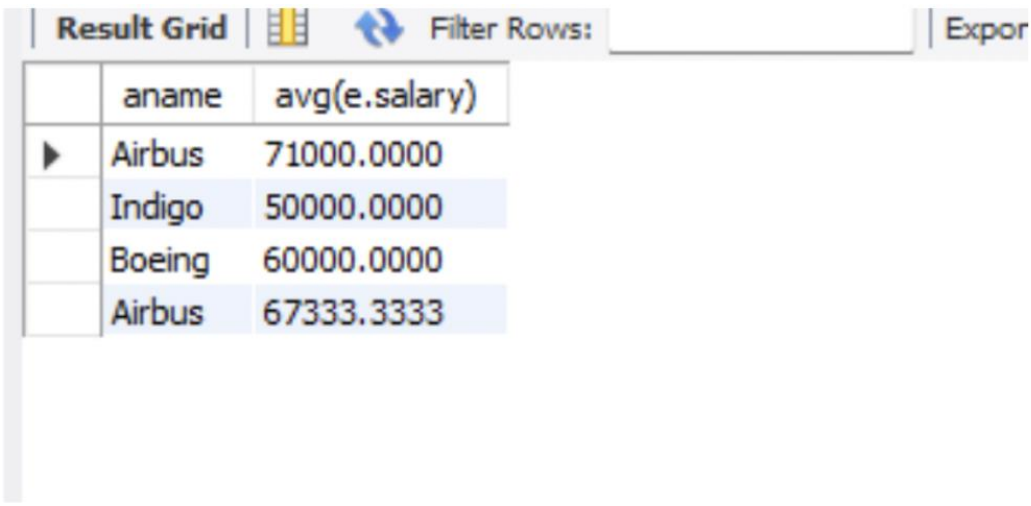
iii. For all aircraft with cruising range over 1000 Kms, find the name of the aircraft and the Average salary of all pilots certified for this aircraft. select a.aname,avg(e.salary) from aircraft a,employee e,certified c where a.aid=c.aid and e.eid=c.eid and a.cruisingrange>1000 group by c.aid;



iv. 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'));



vi. 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');



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

Program 7 week 9: NOSQL

Perform the following DB operations using MongoDB.

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

2. Insert appropriate values

3. Write query to update Email-Id of a student with rollno 10.

4. Replace the student name from “ABC” to “FEM” of rollno 11.

5. Export the created table into local file system

6. Drop the table

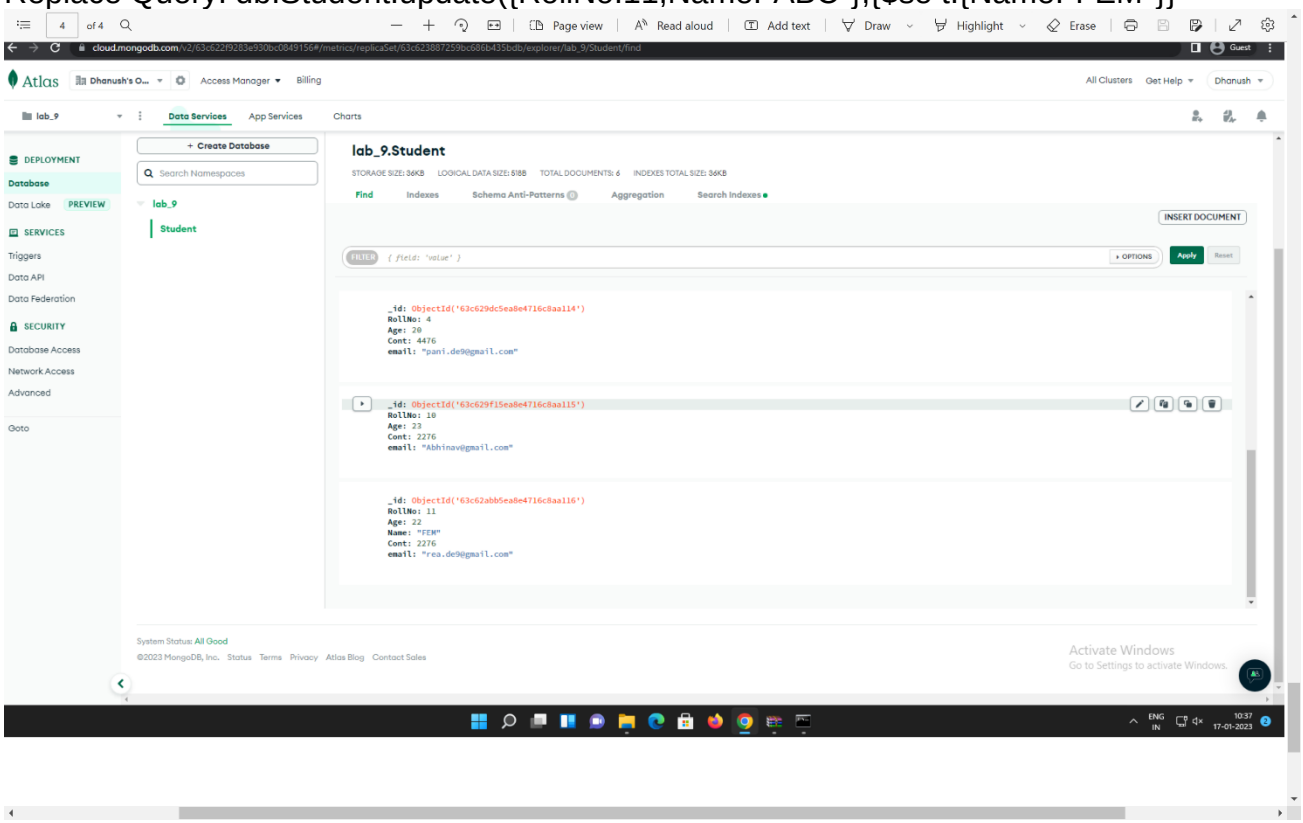
7. Import a given csv dataset from local file system into mongodb collection.

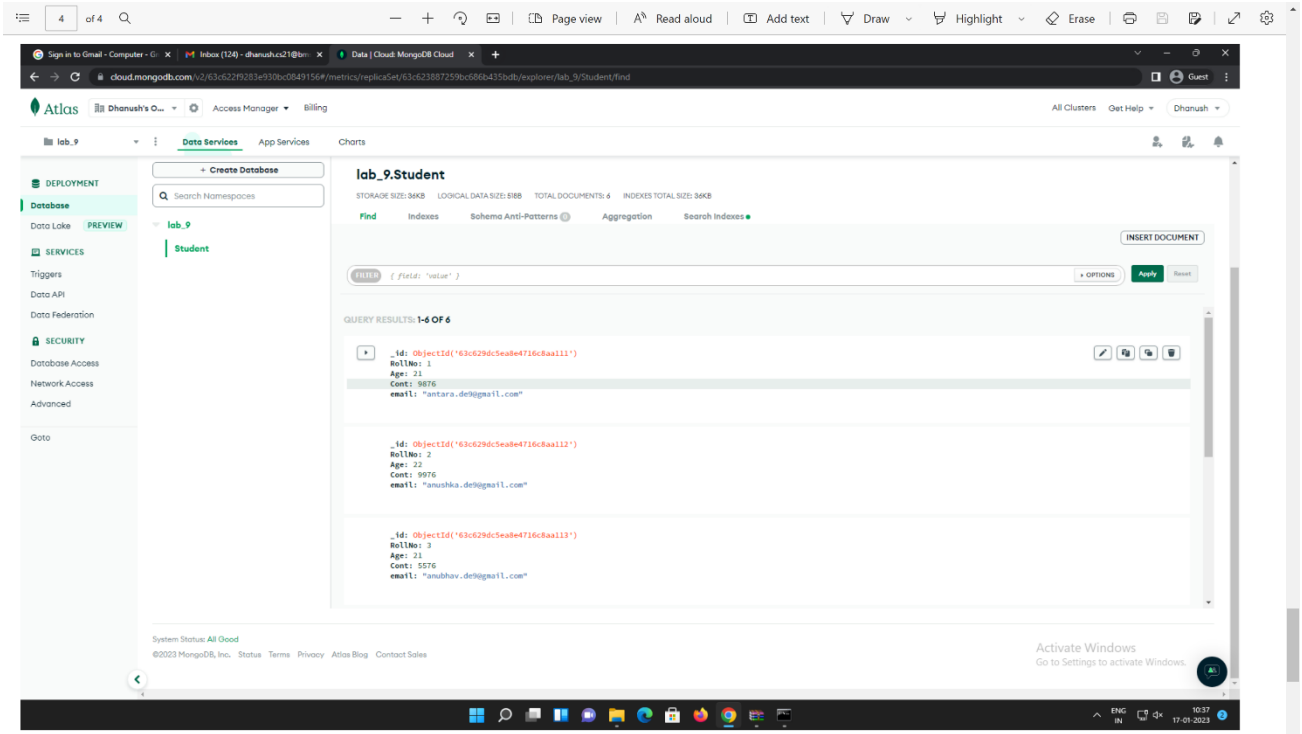
Create database

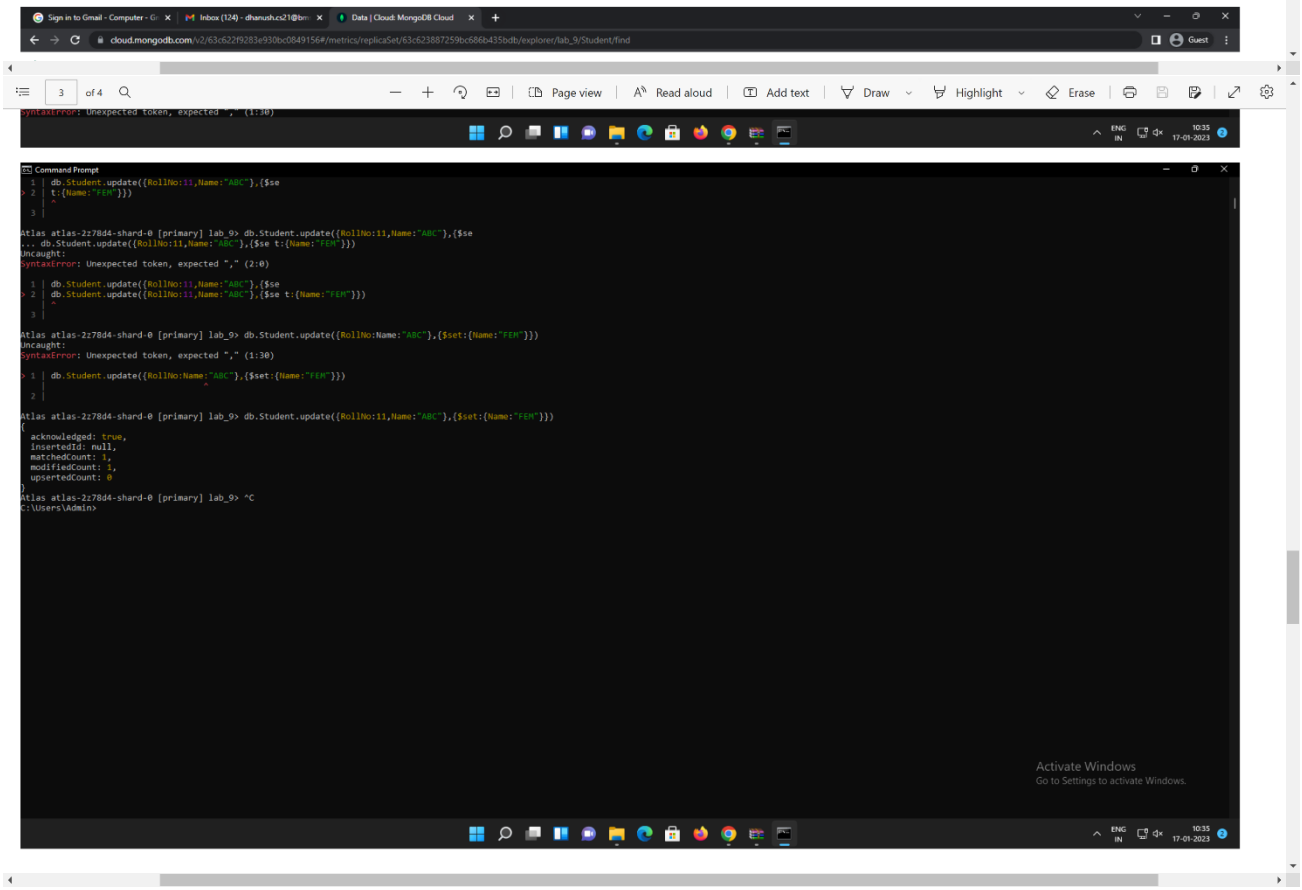
```
db.createCollection("Student");

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()

db.Student.update({RollNo:10},{$set:{ email:"Abhinav@gmail.com"}})
```

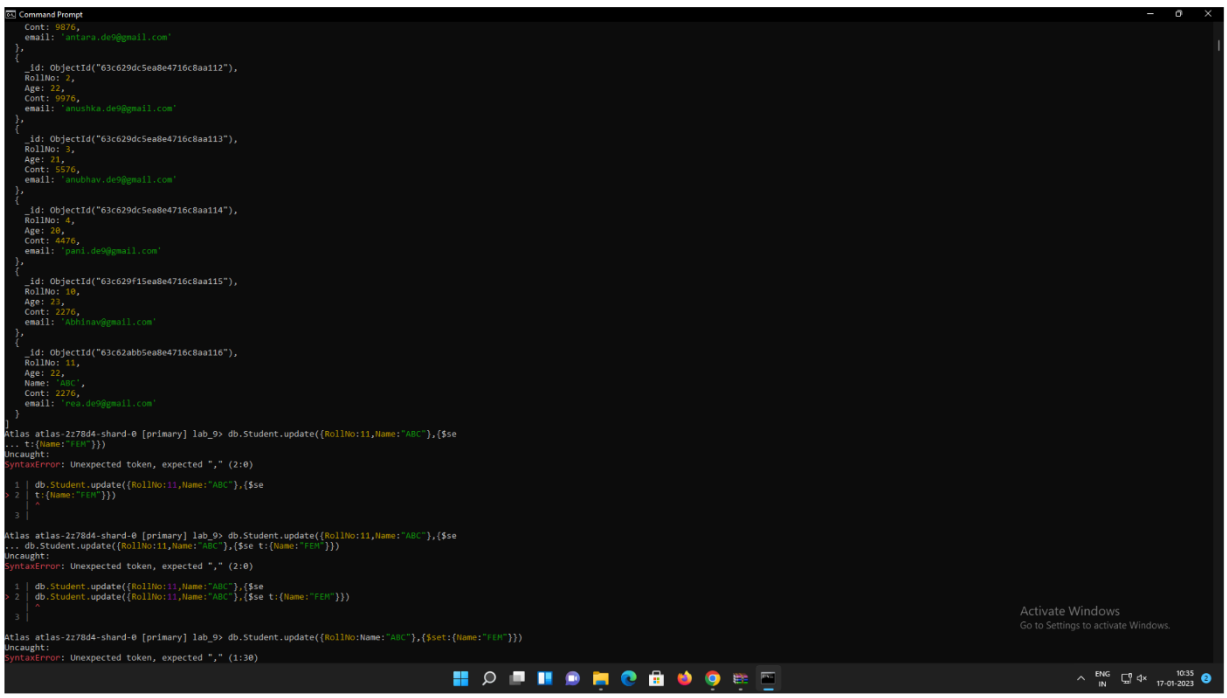
Insert appropriate data: db.Student.insert({RollNo:11,Age:22,Name: "ABC",Cont:2276,email:"rea.de9@gmail.com"});

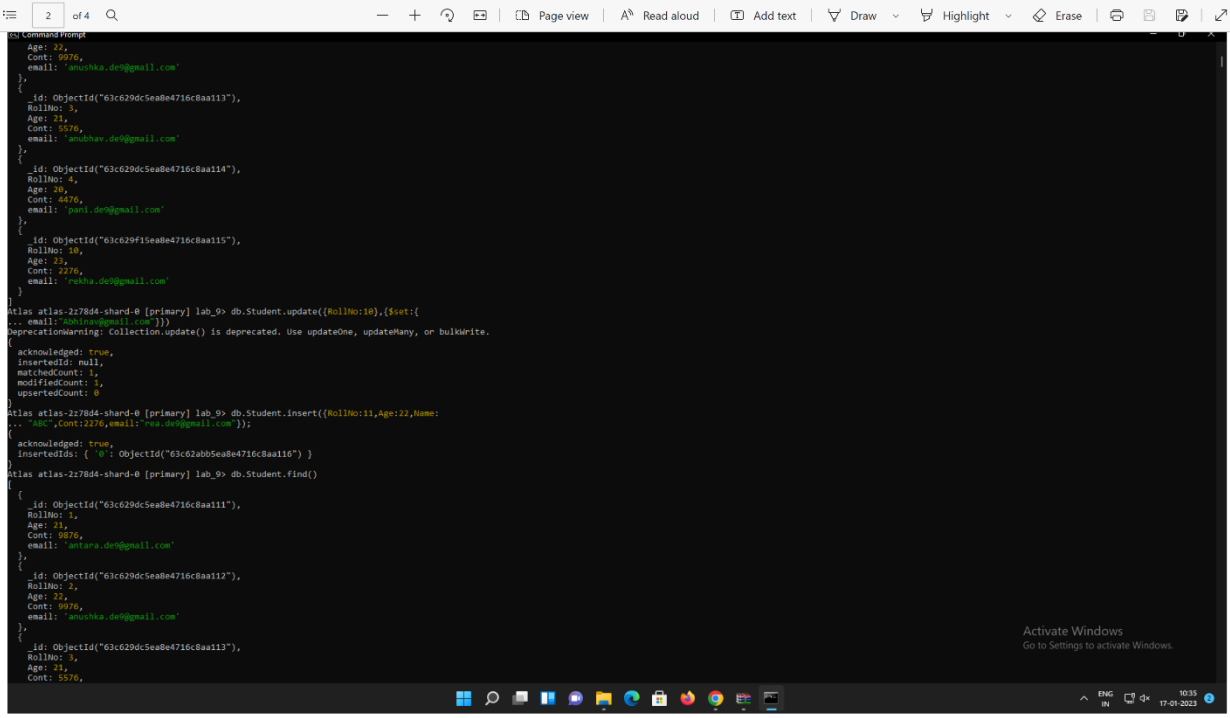
Replace Query: db.Student.update({RollNo:11,Name:"ABC"},{$se t:{Name:"FEM"}}

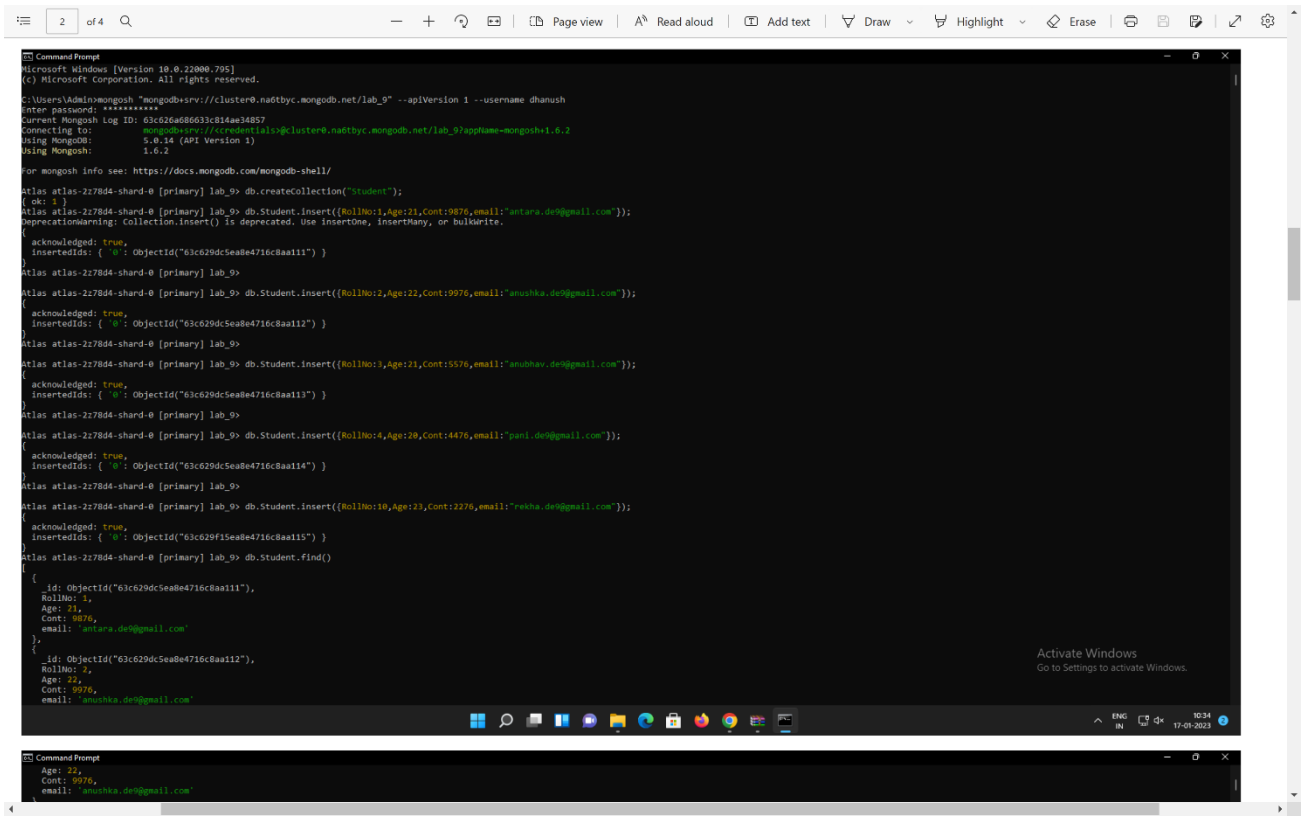



lab_9.Student
STORAGE SIZE: 36KB
LOGICAL DATA SIZE: 518B
TOTAL DOCUMENTS: 6
INDEXES TOTAL SIZE: 36KB
QUERY RESULTS: 1-6 OF 6
_id: ObjectId('63c629dc5ea8e4716c8aa111')
RollNo: 1
Age: 21
Cont: 9876
email: "antara.de9@gmail.com"
_id: ObjectId('63c629dc5ea8e4716c8aa112')
RollNo: 2
Age: 22
Cont: 9976
email: "anushka.de9@gmail.com"
_id: ObjectId('63c629dc5ea8e4716c8aa113')
RollNo: 3
Age: 21
Cont: 5576
email: "anubhav.de9@gmail.com"
Activate Windows


Command Prompt
db.Student.update({RollNo:11,Name:"ABC"},{$se
t:{Name:"FEM"}})
SyntaxError: Unexpected token, expected "," (2:0)
Atlas atlas-2z78d4-shard-0 [primary] lab_9> db.Student.update({RollNo:Name:"ABC"},{$set:{Name:"FEM"}})
Uncaught:
SyntaxError: Unexpected token, expected "," (1:30)
Atlas atlas-2z78d4-shard-0 [primary] lab_9> db.Student.update({RollNo:11,Name:"ABC"},{$set:{Name:"FEM"}})
{
acknowledged: true,
insertedId: null,
matchedCount: 1,
modifiedCount: 1,
upsertedCount: 0
}
Atlas atlas-2z78d4-shard-0 [primary] lab_9> ^C
C:\Users\Admin>
Activate Windows


Command Prompt
Activate Windows
Go to Settings to activate Windows.


Command Prompt
Activate Windows
Go to Settings to activate Windows.



Week 10 :NOSQL

1. Create a collection by name Customers with the following attributes.

Cust_id, Acc_Bal, Acc_Type

db.createCollection(“Student”);

2.DROP COLLECTION
db.Student.drop();

3. Insert at least 5 values into the table

```
db.Student.insert(
{_id:1,
StudName:"MichelleJacintha",
Grade:"VII",
Hobbies:"InternetSurfing"});

BULK INSERT//
var mystudent =
 [ {_id:4, StudName:"saurav", Grade:"V", Hobbies:"Dance"}, {_id:5, StudName:"kumar",
Grade:"VI", Hobbies:"Singing"}]
 db.Student.insert(mystudent)
```

4.UPDATE

```
db.student.update({StudName:"Aryan  David"},{$set:{StudName:"Saurav"}});
```

5. View the entire table

```
db.Student.find()
db.Student.find({StudName:"Aryan David"});
db.Students.find({},{StudName:1,Grade:1,_id:0});
db.Students.find({Grade:{$eq:'VII'}})
```

6.COUNT

```
db.Student.count();
```

7.SORT

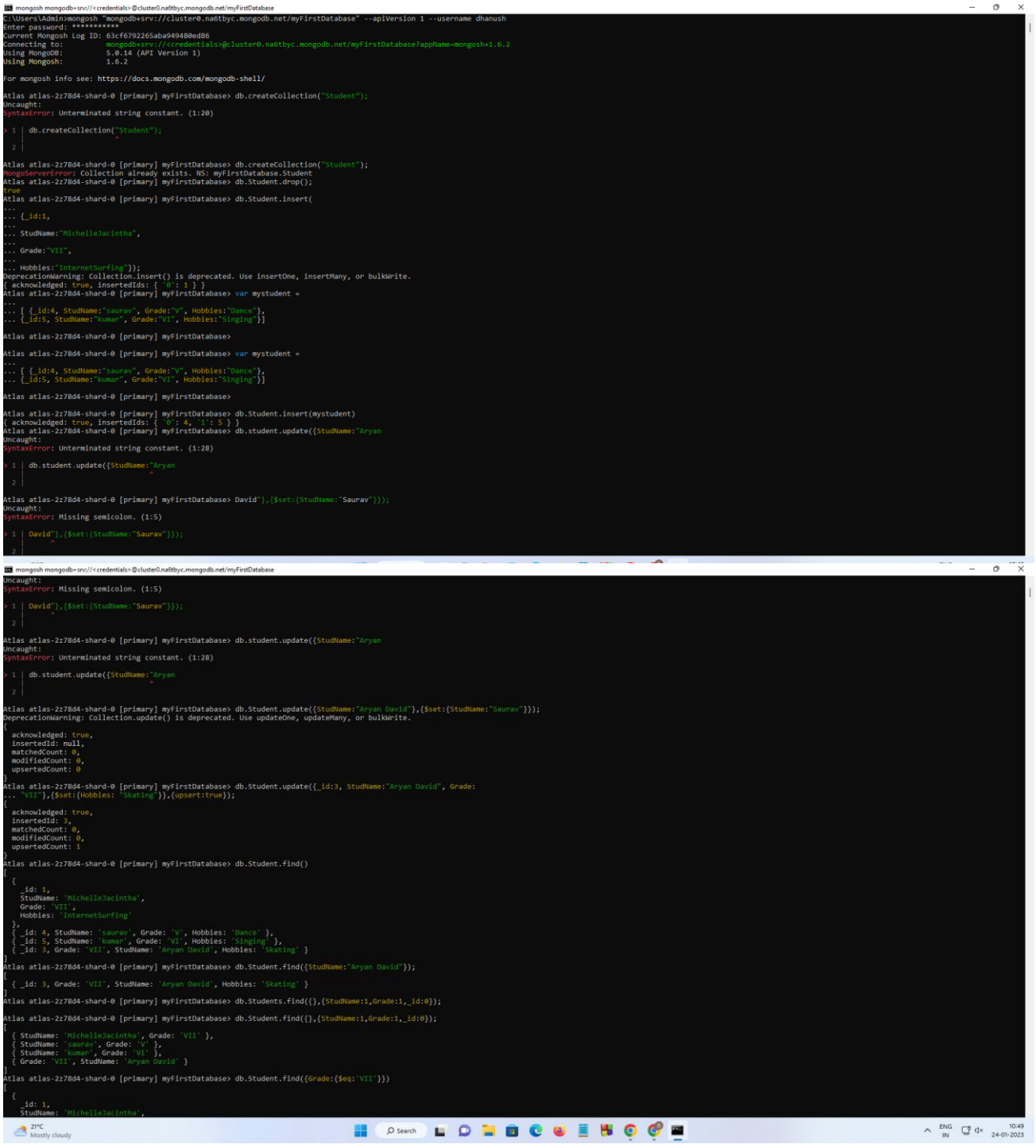
```
db.Students.find().sort({StudName:1})
```

8.REMOVE

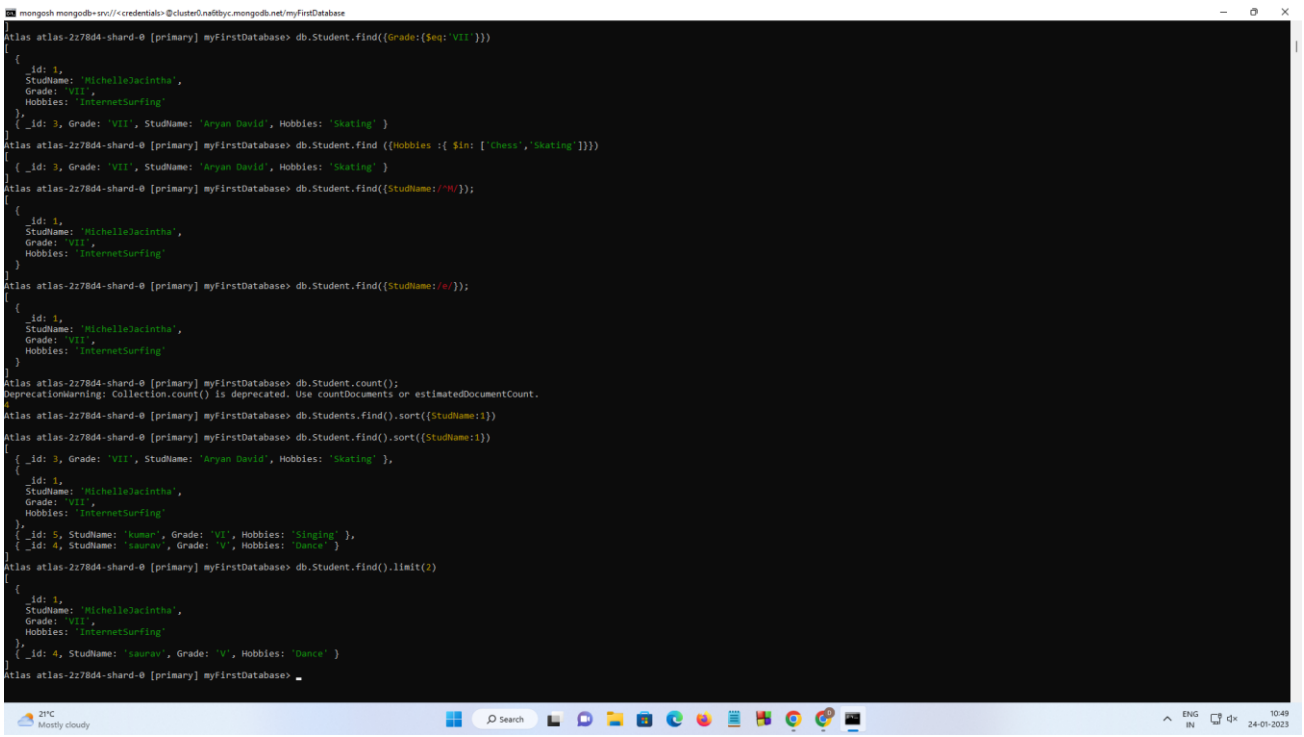
```
db.Student.remove({"StudName":"saurav"})
db.student.find({"StudName":"saurav"});
```

9.DISPLAY

```
db.Student.find().limit(2)
```


mongosh mongodb+srv://<credentials>@cluster0.na6tbyc.mongodb.net/myFirstDatabase
C:\Users\Admin>mongosh "mongodb+srv://cluster0.na6tbyc.mongodb.net/myFirstDatabase" --apiVersion 1 --username dhanush
Enter password: **********
Current Mongosh Log ID: 63cf6792265aba949480ed86
Connecting to: mongodb+srv://<credentials>@cluster0.na6tbyc.mongodb.net/myFirstDatabase?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-2z78d4-shard-0 [primary] myFirstDatabase> db.createCollection("Student");
Uncaught:
SyntaxError: Unterminated string constant. (1:20)
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.createCollection("Student");
MongoServerError: Collection already exists. NS: myFirstDatabase.Student
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.drop();
true
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.insert(
... {_id:1,
... StudName:"MichelleJacintha",
... Grade:"VII",
... Hobbies:"InternetSurfing"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{ acknowledged: true, insertedIds: { '0': 1 } }
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> var mystudent =
... [{_id:4, StudName:"saurav", Grade:"V", Hobbies:"Dance"},
... {_id:5, StudName:"kumar", Grade:"VI", Hobbies:"Singing"}]
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase>
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> var mystudent =
... [{_id:4, StudName:"saurav", Grade:"V", Hobbies:"Dance"},
... {_id:5, StudName:"kumar", Grade:"VI", Hobbies:"Singing"}]
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase>
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.insert(mystudent)
{ acknowledged: true, insertedIds: { '0': 4, '1': 5 } }
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.student.update({StudName:"Aryan
Uncaught:
SyntaxError: Unterminated string constant. (1:28)
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> David"},{$set:{StudName:"Saurav"}});
Uncaught:
SyntaxError: Missing semicolon. (1:5)

mongosh mongodb+srv://<credentials>@cluster0.na6tbyc.mongodb.net/myFirstDatabase
Uncaught:
SyntaxError: Missing semicolon. (1:5)
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.student.update({StudName:"Aryan
Uncaught:
SyntaxError: Unterminated string constant. (1:28)
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.update({StudName:"Aryan David"},{$set:{StudName:"Saurav"}});
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
acknowledged: true,
insertedId: null,
matchedCount: 0,
modifiedCount: 0,
upsertedCount: 0
}
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.update({_id:3, StudName:"Aryan David", Grade:
... "VII"},{$set:{Hobbies: "Skating"}},{upsert:true});
{
acknowledged: true,
insertedId: 3,
matchedCount: 0,
modifiedCount: 0,
upsertedCount: 1
}
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find()
[
{
_id: 1,
StudName: 'MichelleJacintha',
Grade: 'VII',
Hobbies: 'InternetSurfing'
},
{ _id: 4, StudName: 'saurav', Grade: 'V', Hobbies: 'Dance' },
{ _id: 5, StudName: 'kumar', Grade: 'VI', Hobbies: 'Singing' },
{ _id: 3, Grade: 'VII', StudName: 'Aryan David', Hobbies: 'Skating' }
]
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find({StudName:"Aryan David"});
[
{ _id: 3, Grade: 'VII', StudName: 'Aryan David', Hobbies: 'Skating' }
]
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Students.find({},{StudName:1,Grade:1,_id:0});
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find({},{StudName:1,Grade:1,_id:0});
[
{ StudName: 'MichelleJacintha', Grade: 'VII' },
{ StudName: 'saurav', Grade: 'V' },
{ StudName: 'kumar', Grade: 'VI' },
{ Grade: 'VII', StudName: 'Aryan David' }
]
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find({Grade:{$eq:'VII'}})
[
{
_id: 1,
StudName: 'MichelleJacintha',
21°C
Mostly cloudy
Search
ENG
IN
10:49
24-01-2023


mongosh mongodb+srv://<credentials>@cluster0.nafitbyc.mongodb.net/myFirstDatabase
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find({Grade:{$eq:'VII'}})
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find ({Hobbies :{ $in: ['Chess','Skating']}})
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find({StudName:/^M/});
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find({StudName:/e/});
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.count();
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
4
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Students.find().sort({StudName:1})
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find().sort({StudName:1})
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase> db.Student.find().limit(2)
Atlas atlas-2z78d4-shard-0 [primary] myFirstDatabase>
21°C
Mostly cloudy
Search
ENG
IN
10:49
24-01-2023