VISVESVARAYA TECHNOLOGICAL UNIVERSITY “JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

DATABASE MANAGEMENT SYSTEMS

Submitted by

AISHWARYA A GHATAPANADI

(1BM21CS011)

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 AISHWARYA A G (1BM21CS011), 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.

**UMADEVI V Dr. Jyothi S Nayak** Professor and Head

Department of CSE Department of CSE

BMSCE, Bengaluru BMSCE, Bengaluru

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Date** | **Experiment Title** | **Page No.** |
| **1** | 1/11/22 | Insurance Database | **3-6** |
| **2** | 8/11/22 | More Queries on Insurance Database | **7-8** |
| **3** | 15/11/22 | Bank Database | **9-13** |
| **4** | 29/11/22 | More Queries on Bank Database | **14-14** |
| **5** | 06/12/22 | Employee Database | **15-18** |
| **6** | 13/12/22 | More Queries on Employee Database | **19-21** |
| **7** | 20/12/22 | Supplier Database | **22-27** |
| **8** | 27/12/22 | Flight Database | **28-31** |
| **9** | 10/1/23 | NoSQL | **32-** |

**LAB 1: INSURANCE DATABASE**

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

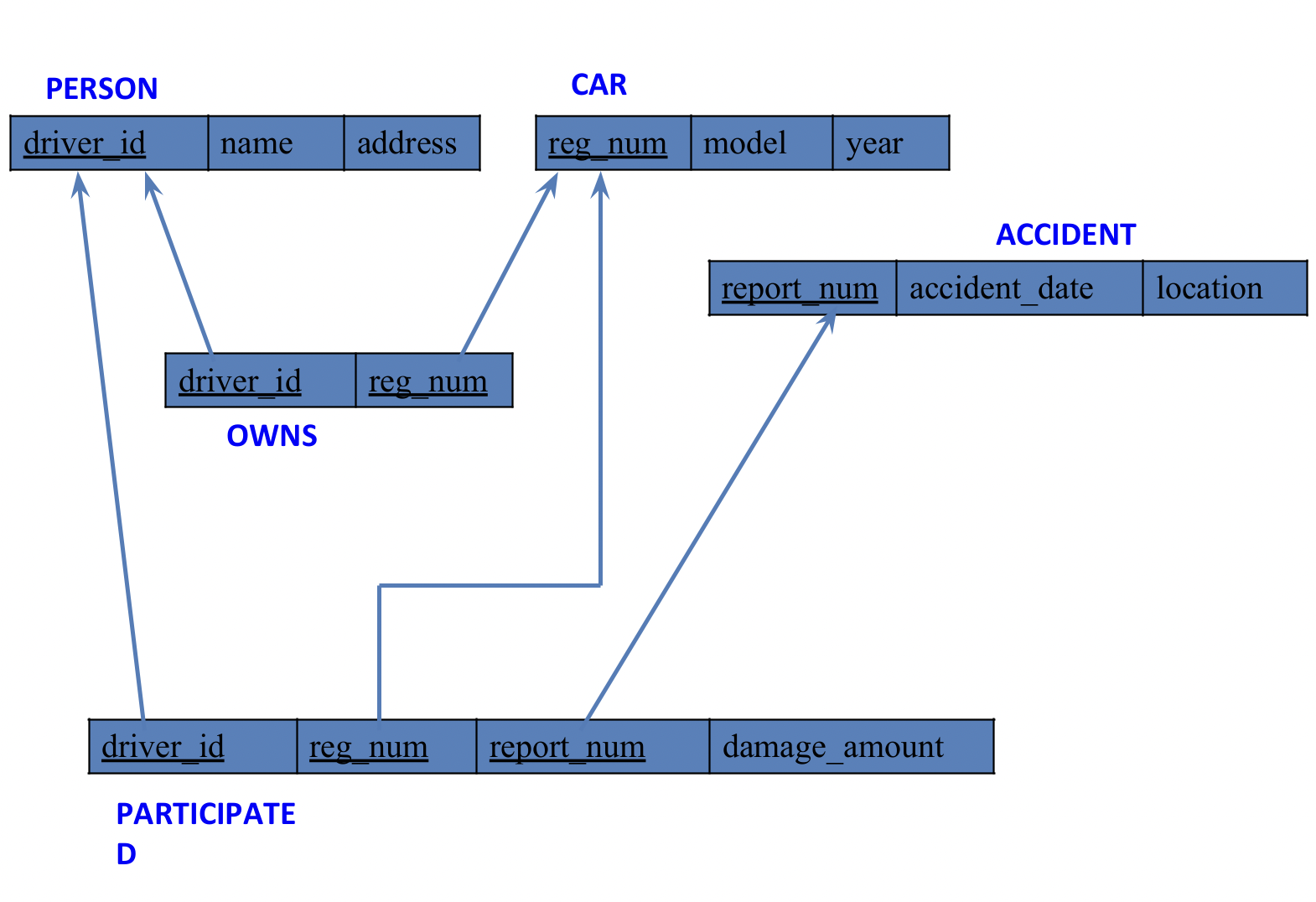
CAR (Regno: String, model: String, year: int)

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 insurance;

use insurance;

create table person (

driverid varchar(10),

name varchar(30),

address varchar(30), primary key(driver\_id));

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));

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');

insert into person values(A03', 'Smith','Ashok nagar');

insert into person values('A04','Venu','N R Colony');

insert into person values('A05','Jhon','Hanumanth nagar');

insert into car values(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');

insert into participated values(A01', 'KA052250',11,10000); insert

into participated

values ('A02', KA053408',12,50000); insert

into participated

values('03'KA095477',13,25000); insert

into participated

values(A04', KA031181',14,3000); insert

into participated

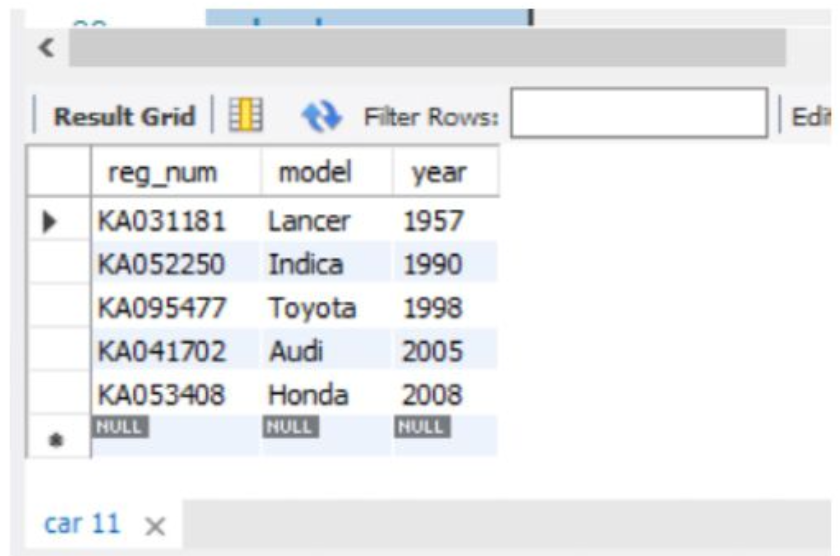
values('05',KA041702,15,5000);

Queries:

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

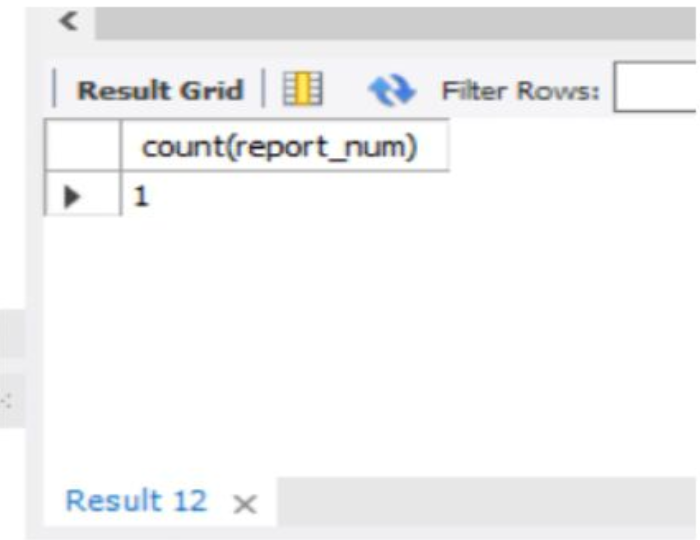
SQL> select \*from car

order by vear asc:



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

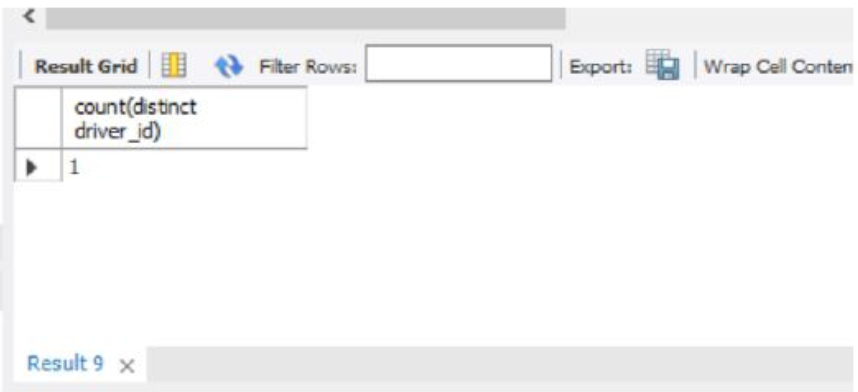
SQL> select count(reportnum ) from car c, participated p where c.reg\_num=p.reg\_num and c.model='Lancer";



3.Find the total n u m b e rof 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%"

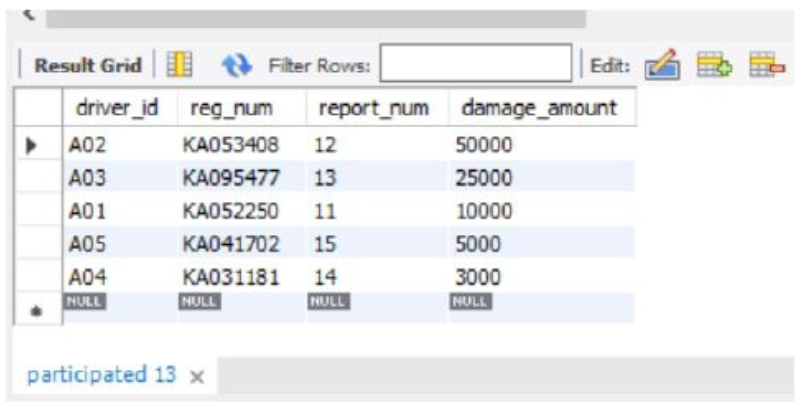


TO DO

List the entire participated relation in descending order of damage amount.

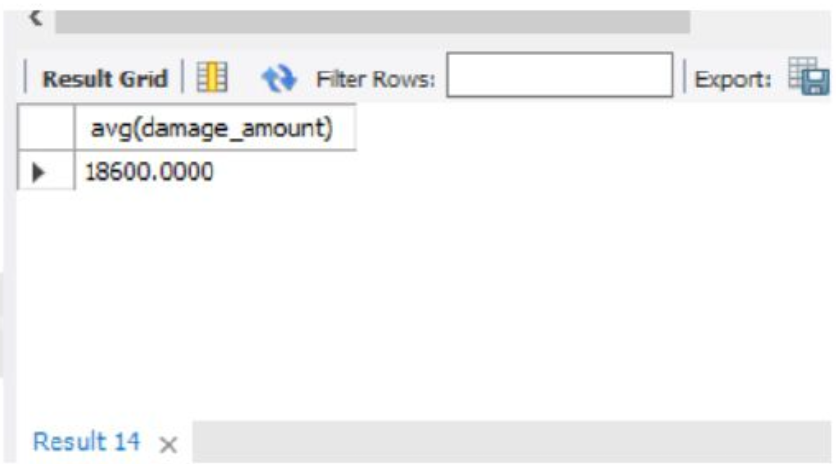
SQL> select \*from participated

order by damageamount desc;



Find the average damage amount.

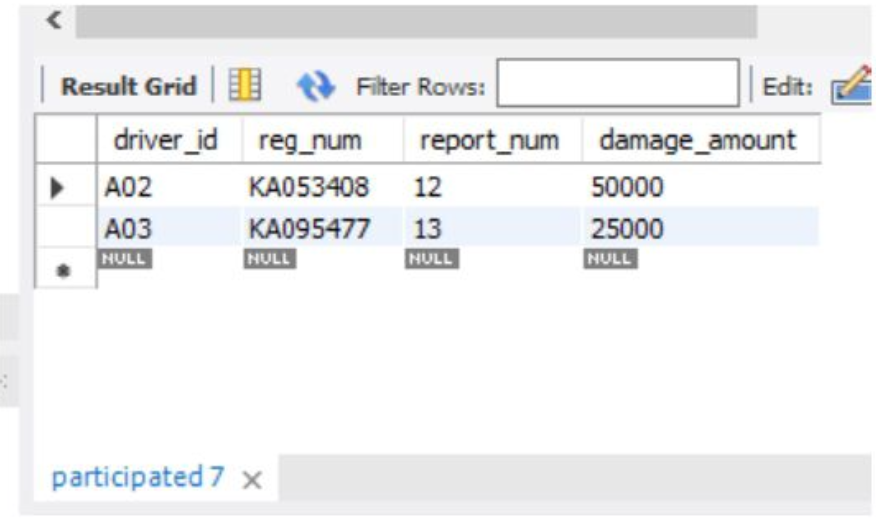
SQL> select avg(damage\_amount) from participated;



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

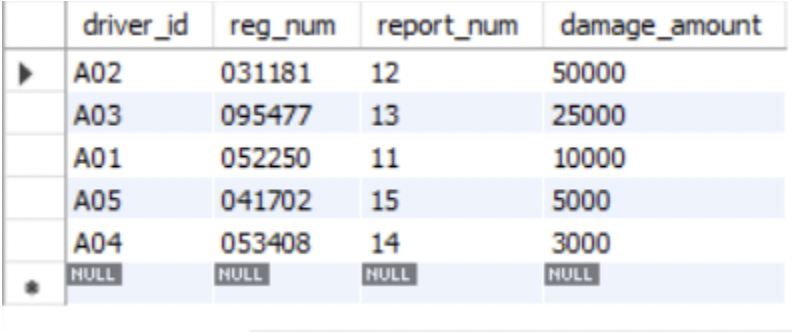
SQL> delete from participated where damage\_amount < (select t.avg1 from (select avg ( damageamount) as avg1 from participated) t);

select \*from participated;



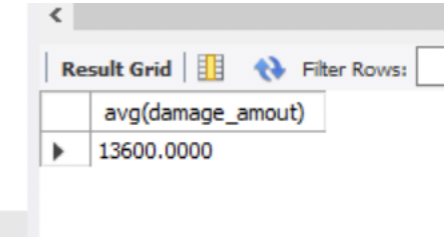
**LAB 2:More Queries on Insurance Database**

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

SQL>select \* from participated order by(damage\_amount) desc;

• FIND THE AVERAGE DAMAGE AMOUNT

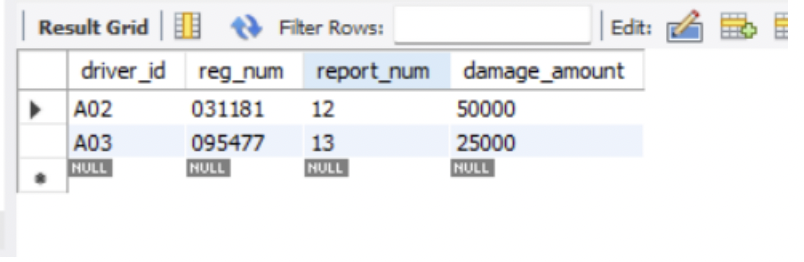
SQL>select avg(damage\_amount) from participated;



• DELETE THE TUPLE WHOSE DAMAGE AMOUNT IS BELOW THE AVERAGE DAMAGE AMOUNT

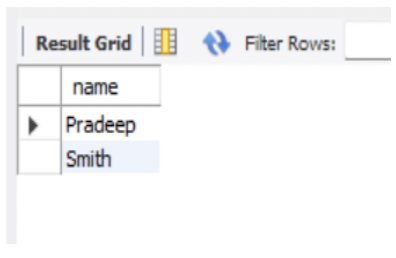
SQL>delete from participated where damage\_amount<( select p.amt from(select avg(damage\_amount)as amt from participated) p);

SQL>select \* from participated;



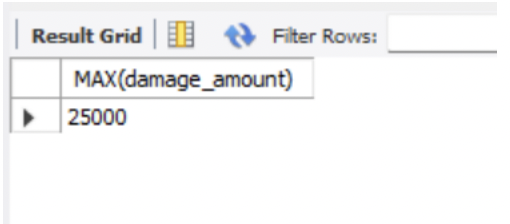
• LIST THE NAME OF DRIVERS WHOSE DAMAGE IS GREATER THAN THE AVERAGE DAMAGE AMOUNT.

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 max(damage\_amount) from participated;



**LAB 3: BANKING ENTERPRISE DATABASE**

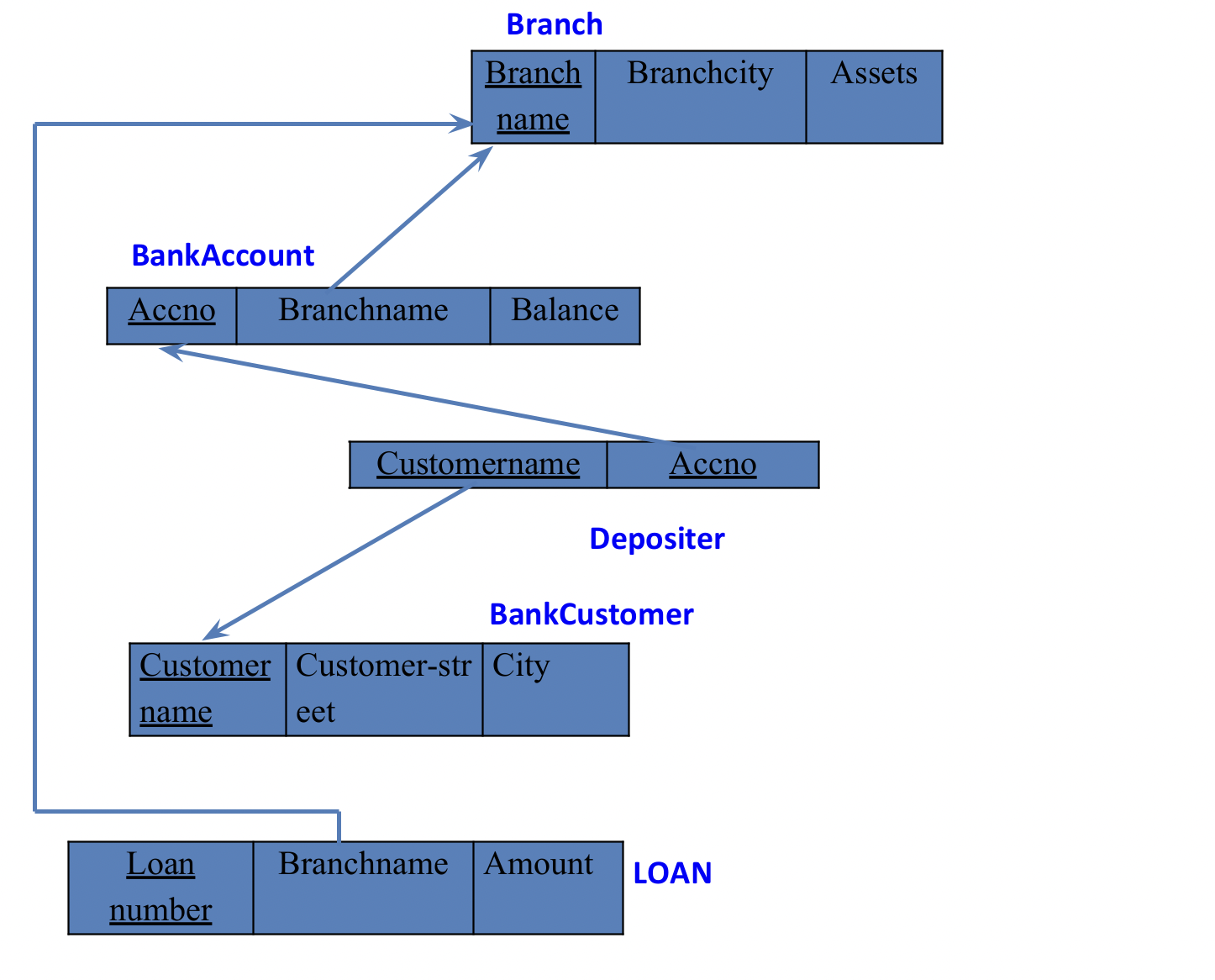
BRANCH (branch-name: String, branch-city: String, assets: real)

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 bank;

use bank;

create table branch

( branchname varchar(50), branchcity varchar(50), assets int,

primary key(branchname) );

create table bankaccount

( accno int, branchname varchar(50), balance int, primary key(accno),

foreign key(branchname) references branch(branchname));

create table bankcustomer

(customername varchar(50), customerstreet varchar(50), city varchar(50),

primary key(customername));

create table depositer

( customername varchar(50), accno int,

primary key(accno),

foreign key(accno) references bankaccount(accno),

foreign key(customername) references bankcustomer(customername));

create table loan

( loannumber int, branchname varchar(50), amount int,foreign key(branchname) references branch(branchname));

**(Inserting values in table)**

insert into branch values("SBI\_Chamrajpet","Banglore",50000); ("SBI\_ResidencyRoad","Banglore",10000);("SBI\_ShivajiRoad","Bombay",20000);("SBI\_ParlimentRoad","Delhi",10000),("SBI\_Jantarmantar","Delhi",20000);

insert into bankaccount values(1,"SBI\_Chamrajpet",2000); (2,"SBI\_ResidencyRoad",5000);(3,"SBI\_ShivajiRoad",6000);(4,"SBI\_ParlimentRoad",9000),(5,"SBI\_Jantarmantar",8000);(6,"SBI\_ShivajiRoad",4000); (8,"SBI\_ResidencyRoad",4000);(9,"SBI\_ParlimentRoad",3000);(10,"SBI\_ResidencyRoad",5000); (11,"SBI\_Jantarmantar",2000);

insert into bankcustomer values("Avinash","Bull\_Temple\_Road","Banglore"); ("Dinesh","Bannergatta\_Road","Banglore");("Mohan","NationalCollege\_Road","Banglore");("Nikil","Akbar\_Road","Delhi");("Ravi","Prithviraj\_Road","Delhi");

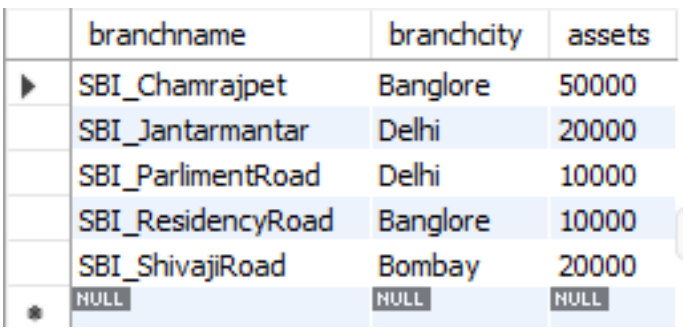
insert into depositer values("Avinash",1),("Dinesh",2); ("Nikil",4);("Ravi",5); ("Avinash",8); ("Nikil",9);("Dinesh",10); ("Nikil",11);

insert into loan values(1,"SBI\_Chamrajpet",1000);(2,"SBI\_ResidencyRoad",2000); (3,"SBI\_ShivajiRoad",3000); (4,"SBI\_ParlimentRoad",4000); (5,"SBI\_Jantarmantar",5000);

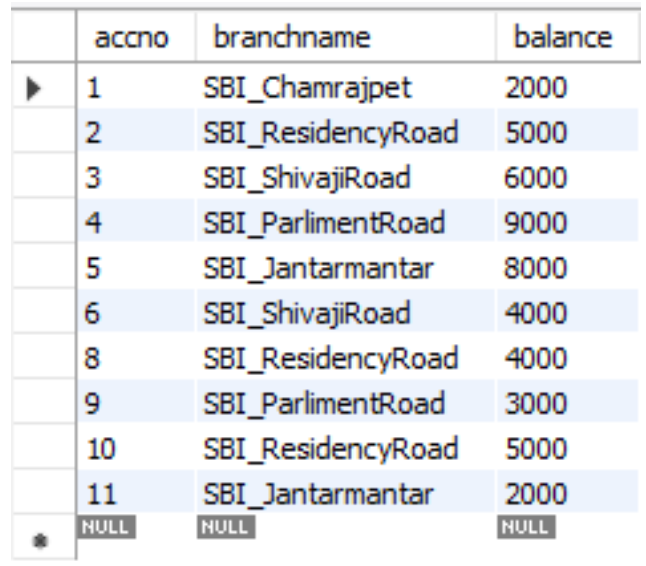
Queries:

**(Displaying Tables)**

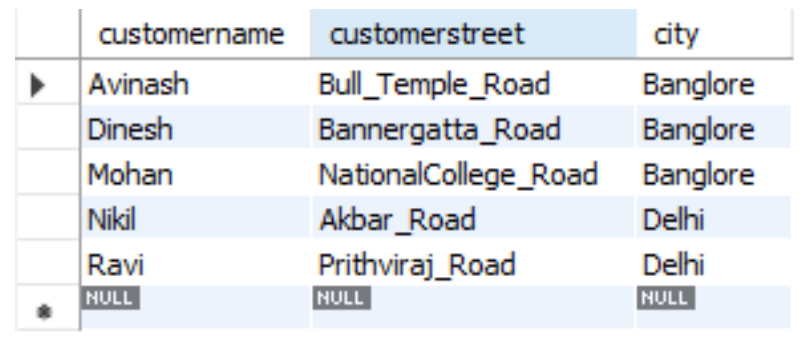
select \* from branch;



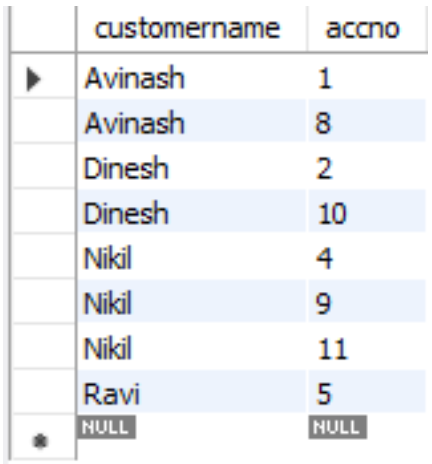
select \* from bankaccount;



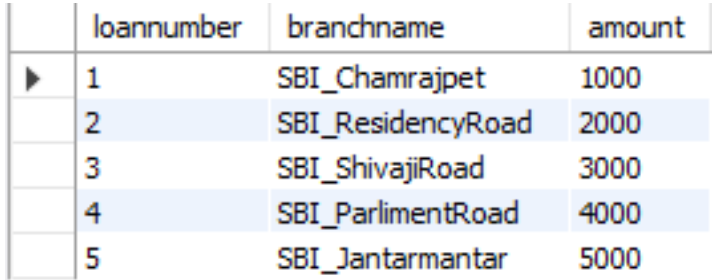
select \* from bankcustomer;



select \* from depositer;

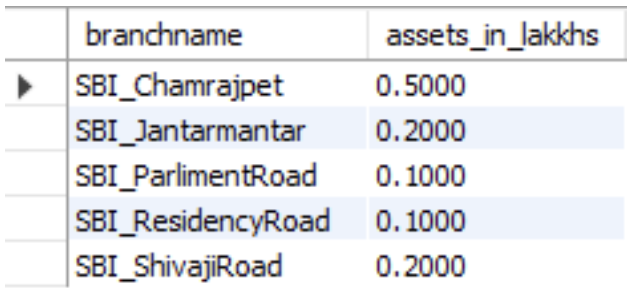


select \* from loan;



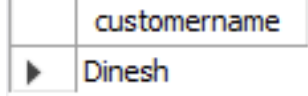
• Display the branch name and assets from all branches in lakhs of rupees and rename the assets column to 'assets in lakhs'.

select branchname,assets/100000 as assets\_in\_lakkhs from branch;

****

• **Find all the customers who have at least two accounts at the same branch (ex. SBI\_ResidencyRoad).**

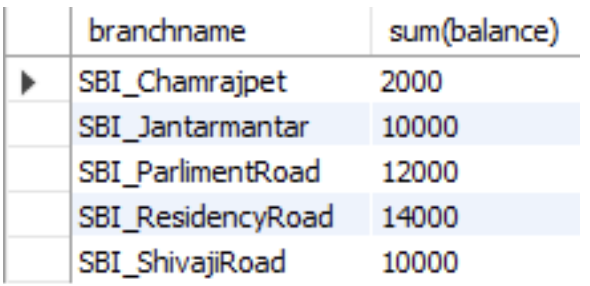
select d.customername from bankaccount b, depositer d where b.branchname="SBI\_ResidencyRoad" and b.accno=d.accno group by d.customername having count(d.accno)>=2;



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

create view sum\_of\_loan as select branchname,sum(balance) from bankaccount group by branchname;

select \* from sum\_of\_loan;

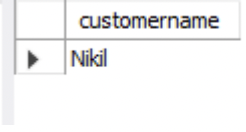


**LAB 4:More Queries on Bank Database**

insert into bankaccount values(12,"SBI\_MatriMarg",2000); insert into branch values("SBI\_MatriMarg","Delhi",200000); insert into depositer values("Nikil",12); create table borrower(customername varchar(50), loannumber int, foreign key(customername) references bankcustomer(customername), foreign key(loannumber) references loan(loannumber)); insert into borrower values("Avinash",1),("Dinesh",2),("Mohan",3),("Nikil",4),("Ravi",5);

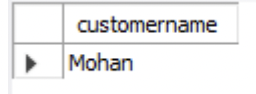
• Find all the customers who have an account at all the branches located in a specific city (Ex. Delhi).

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



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

SQL> select distinct b.customername from borrower b, depositer d where b.Customername not in( select d.customername from loan l,depositer d, borrower b where l.loannumber=b.loannumber and d.customername=b.customername );



**LAB 5:EMPLOYEE DATABASE**

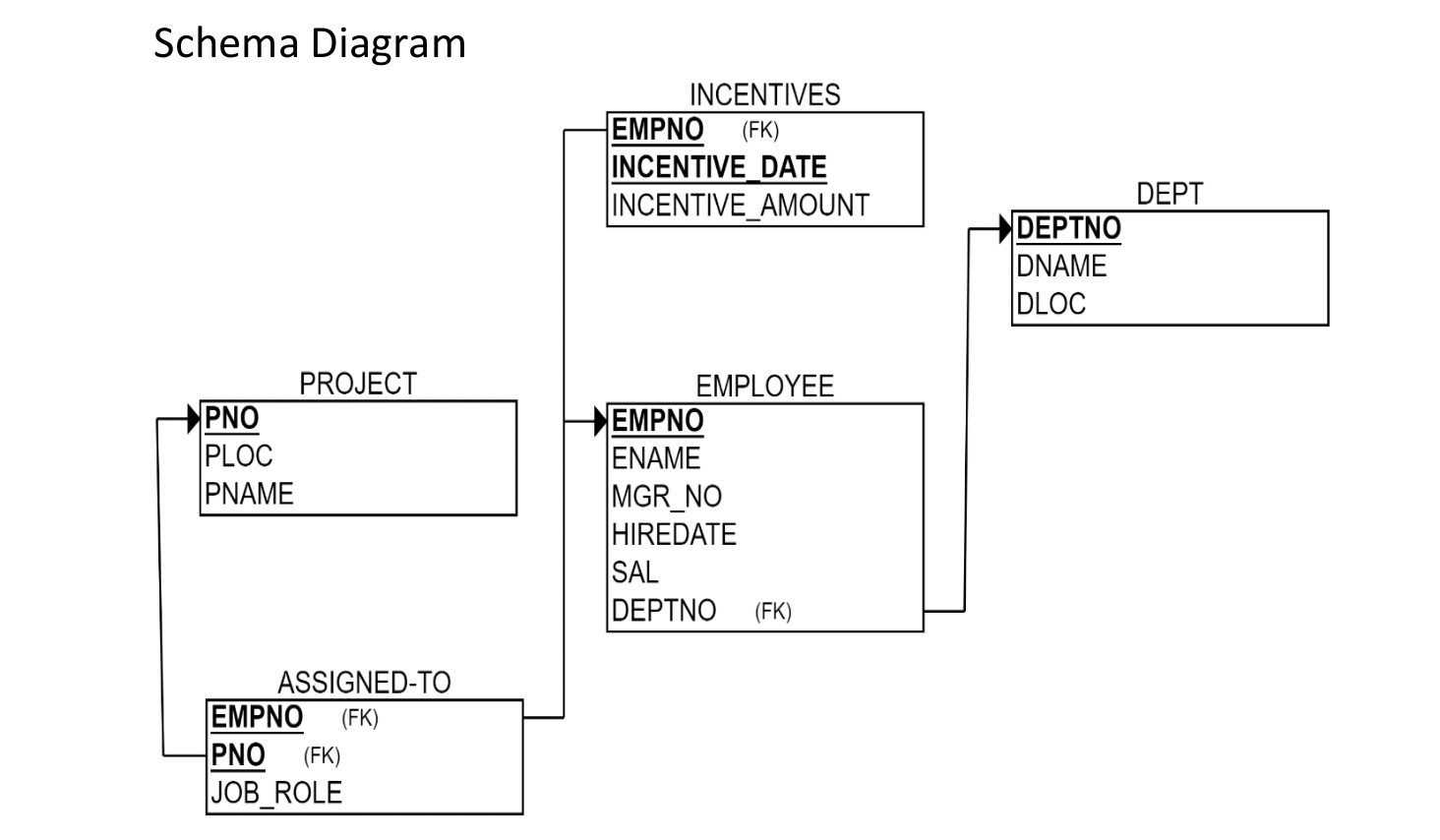


Table Creation:

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);

Values:

insert into dept values((1,"Admin","Banglore"), (2,"Sales","Bangolre"), (3,"Finance","Hyderbad"),(4,"Marketing","Mysore"), (5,"Shipping","Hyderbad"); (6,"Purchasing","Mysore"));

insert into employee values(1,"Avinash",3,"2000-02-14",25000,1), (2,"Balaji",3,"1999-05-11",31000,3), (3,"Dinesh",NULL,"1992-01-26",46000,2),(4,"Chandan",3,"2001-05-21",28000,4), (5,"Aravind",2,"1998-09-22",17000,5);(6,"Amal",3,"2003-02-14",25000,6);

insert into incentive values(1,"2005-03-23",5000), (3,"2001-08-23",50000),(5,"2011-04-02",1500);(3,"2001-08-23",50000), (5,"2011-04-02",1500);

(11,"Banglore","Documentation"),(12,"Banglore","Selling"),(13,"Hyderbad","Accounting"),(14,"Mysore","Advertising"),(15,"Hyderbad","Transportation");

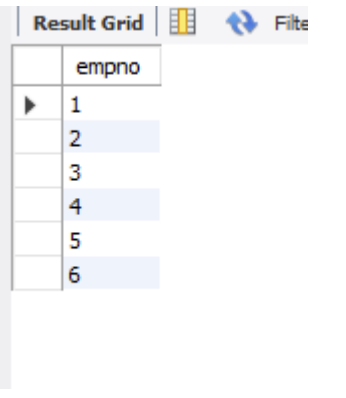
insert into project values(16,"Mysore","Purchasing"); (17,"Hubli","Presentatiom");\

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

Queries:

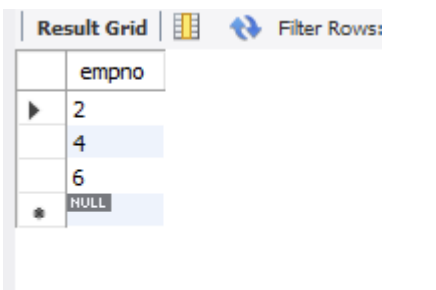
• Retrieve the employee numbers of all employees who work on project located in Banglore, Hyderbad, or Mysore

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



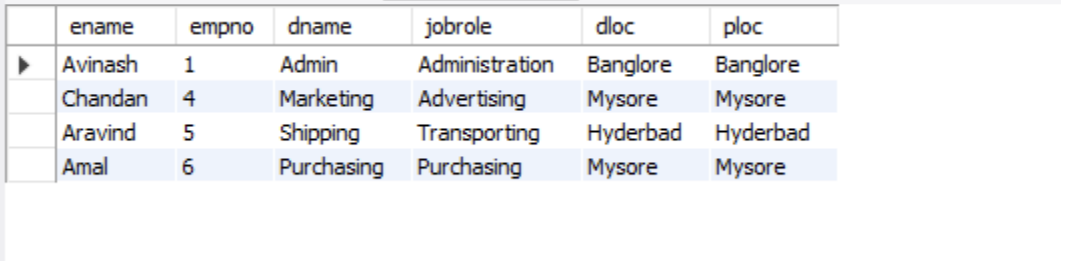
• Get Employee ID’s of those employees who didn’t receive incentives

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



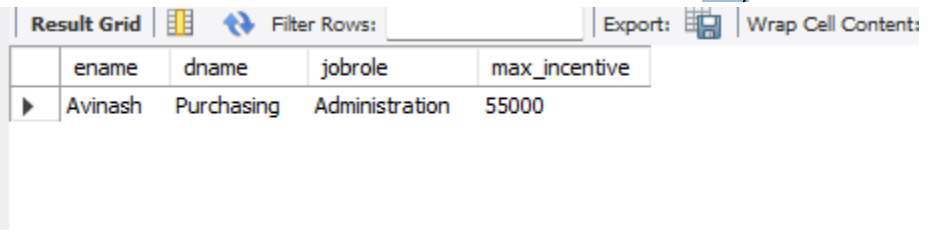
• Write a SQL query to find the employees name, number, dept, job\_role, department location and project location who are working for a project location same as his/her department location.

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



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

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



**LAB 6:More Queries on Employee Database**

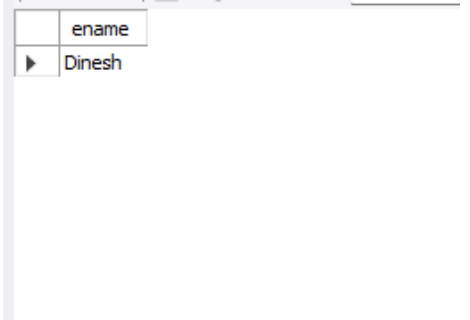
Queries:

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

SQL>select e.ename from employee e,employee f

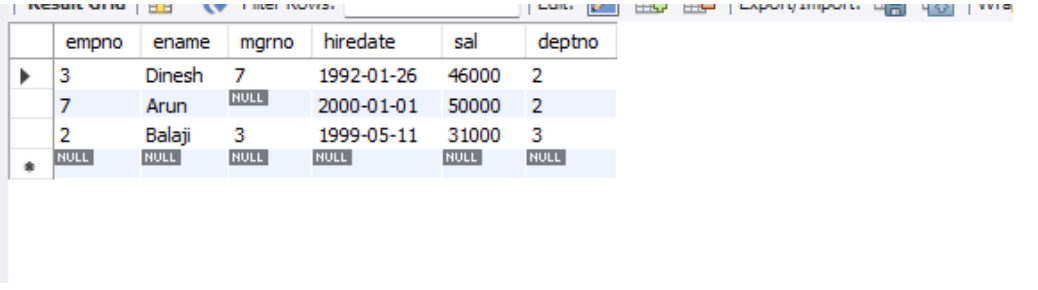
where e.empno=f.mgrno

group by e.empno having count(\*)=(select max(mycount)

From (select count(\*) mycount from employee group by mgrno) a);

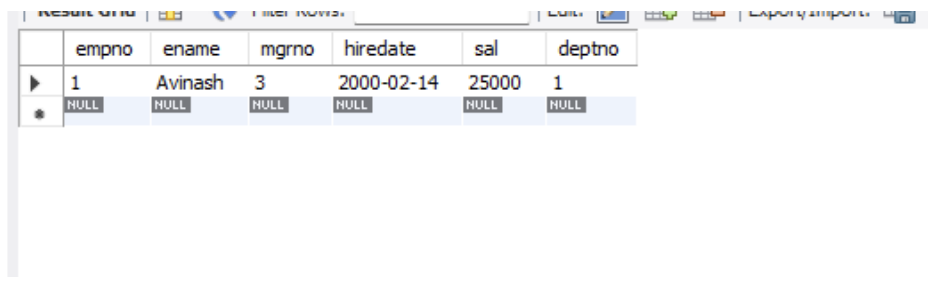
--Display those managers name whose salary is more than average salary of his employee.

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



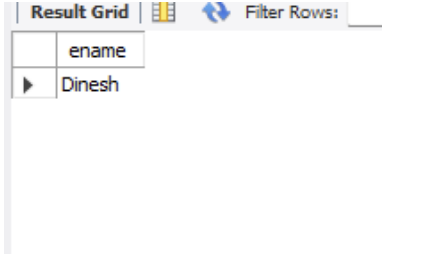
--Find the employee details who got second maximum incentive in 2005.

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

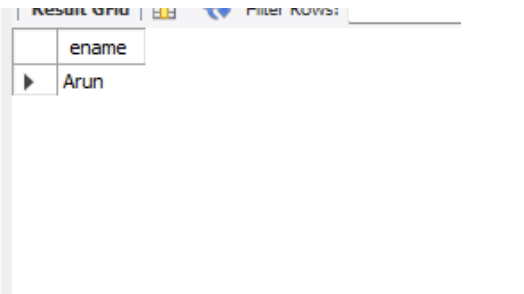


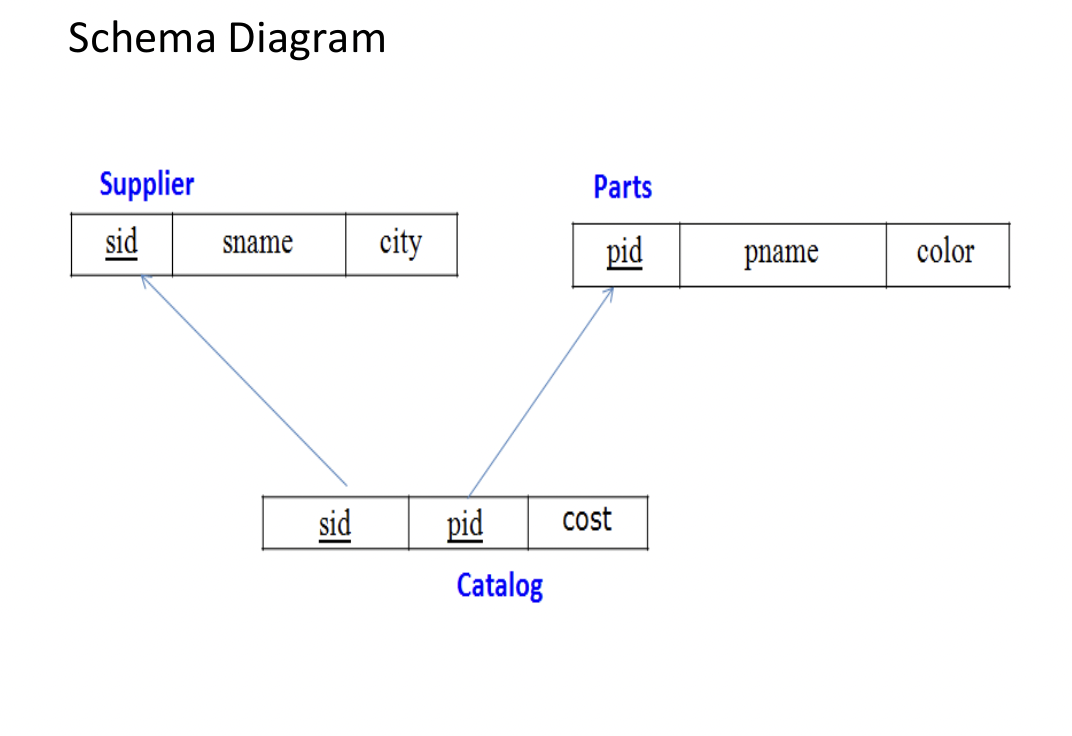
--Display those employees who are working in the same department where his manager is working.

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



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

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

**LAB 7: SUPLLIER** 

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));

Values:

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

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

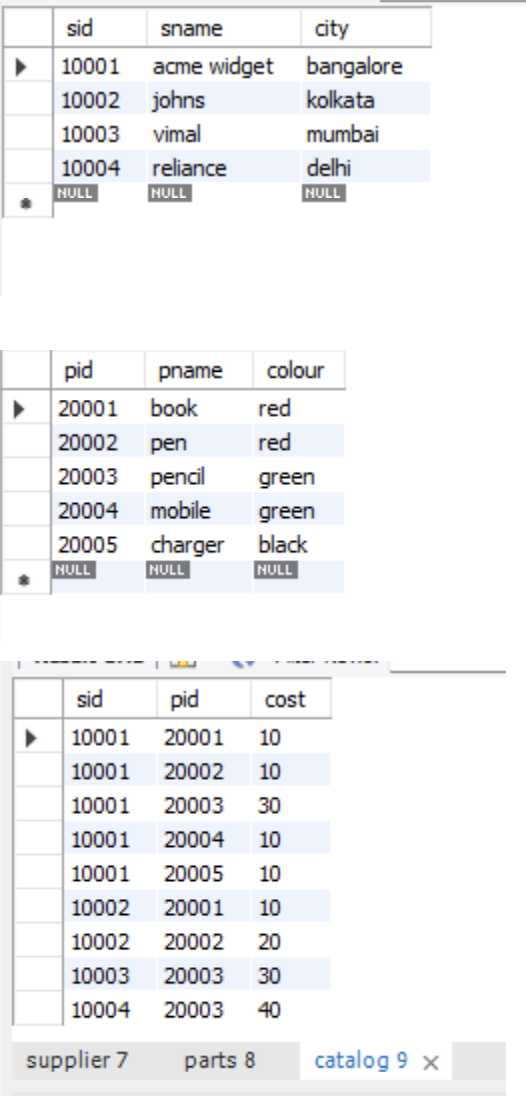
insert into catalog values(10001,20001,10),(10001,20002,10),(10001,20003,30),(10001,20004,10),(10001,20005, 10),(10002,20001,10),(10002,20002,20);

insert into catalog values(10003,20003,30),(10004,20003,40);

select \* from supplier;

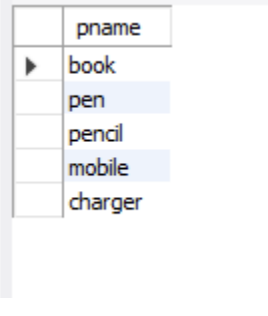
select \* from parts;

select \* from catalog;

select \* from supplier; select \* from parts;select \* from catalog;

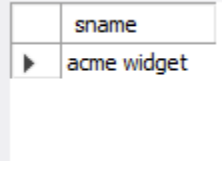
● Find the pnames of parts for which there is some supplier.

SQL>select pname from parts where pid in (select pid from catalog);



● Find the snames of suppliers who supply every part.

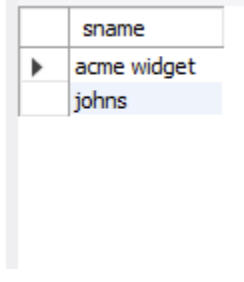
SQL>select sname from (select c.sname,count(distinct a.pid) as cnt from catalog a left join parts b on a.pid=b.pid left join supplier c on c.sid=a.sid group by 1) a where cnt=(select count(distinct a.pid) from catalog a left join parts b on a.pid=b.pid);



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

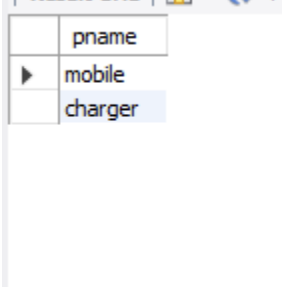
SQL>select sname from supplier

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



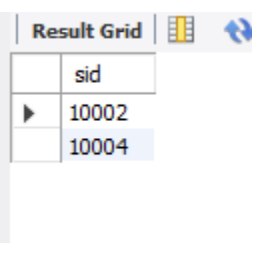
● Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.

SQL>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'));



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

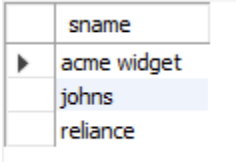
SQL>select c.sid from catalog c where c.cost >(select avg(cc.cost) from catalog cc where c.pid=cc.pid group by cc.pid);



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

SQL>select sname from supplier where sid in( select sid from catalog

where cost in( select max(cost) from catalog group by pid));



**LAB 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)

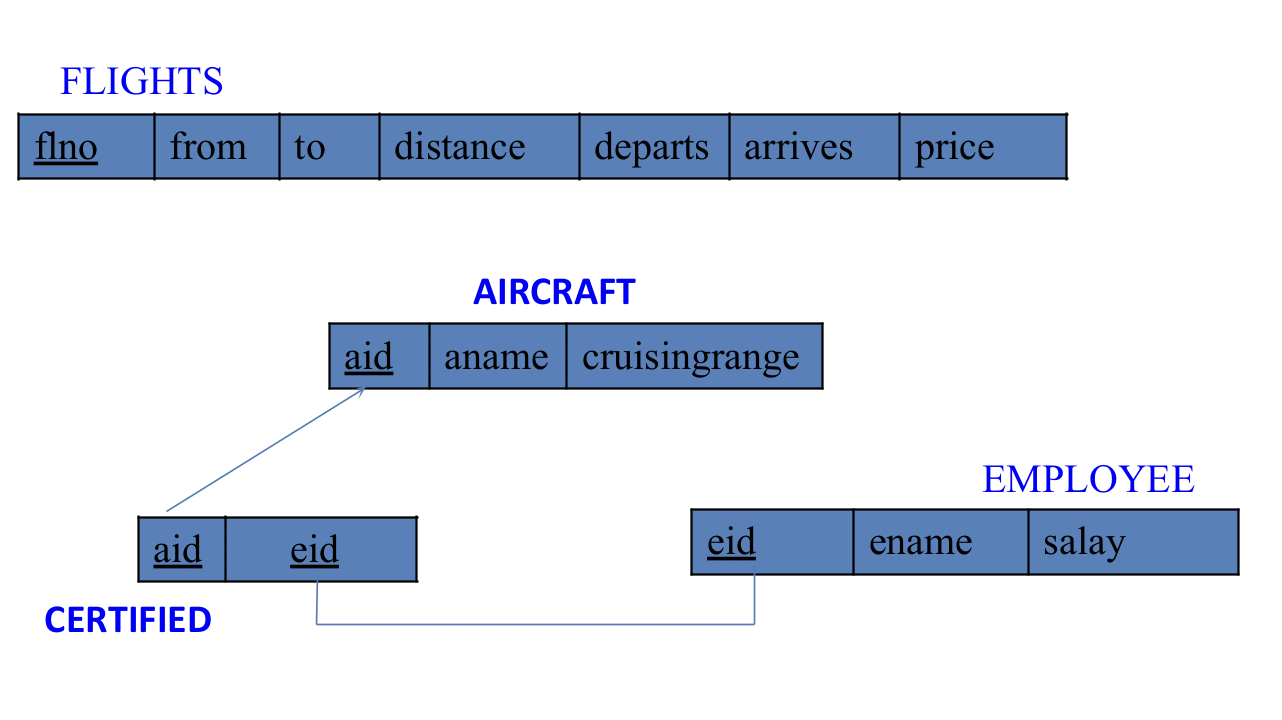
CERTIFIED(eid: integer, aid: integer)

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**



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 certified(

eid int,aid int,

foreign key(aid) references aircraft(aid)

on update cascade on delete cascade,

foreign key(eid) references employee(eid)

on update cascade on delete cascade);

create table employee(

eid int, ename varchar(50), salary int,

primary key(eid));

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

(105,'Tilak',5000);

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

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

insert into flights values

(1,'Banglore','New Delhi',500,'6:00','9:00',5000), (2,'Banglore','Chennai',300,'7:00','8:30',3000),

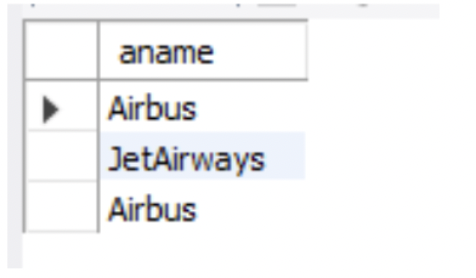
(3,'Trivandrum','New Delhi',800,'8:00','11:30',6000), (4,'Banglore','Frankfurt',10000,'6:00','23:30',50000),

(5,'Kolkata','New Delhi',2400,'11:00','3:30',9000), (6,'Banglore','Frankfurt',8000,'9:00','23:00',40000);

Queries:

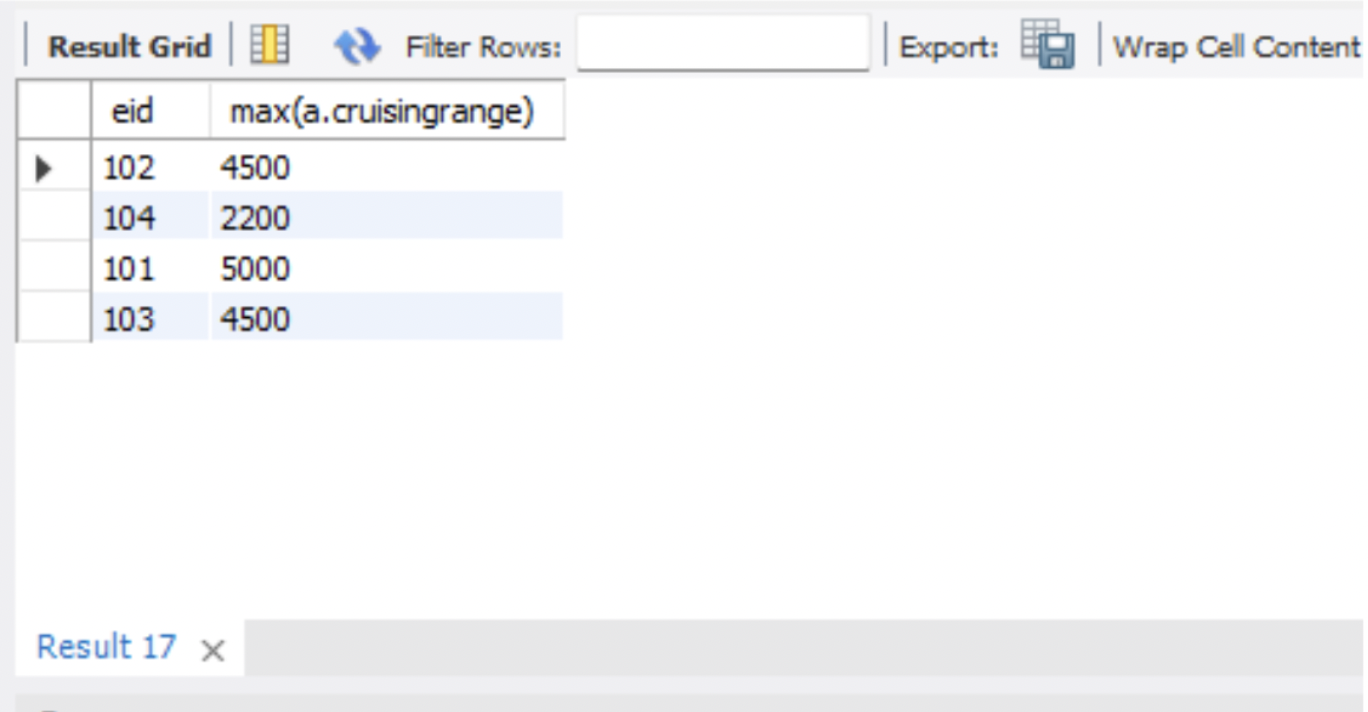
* Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.

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



* 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.

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

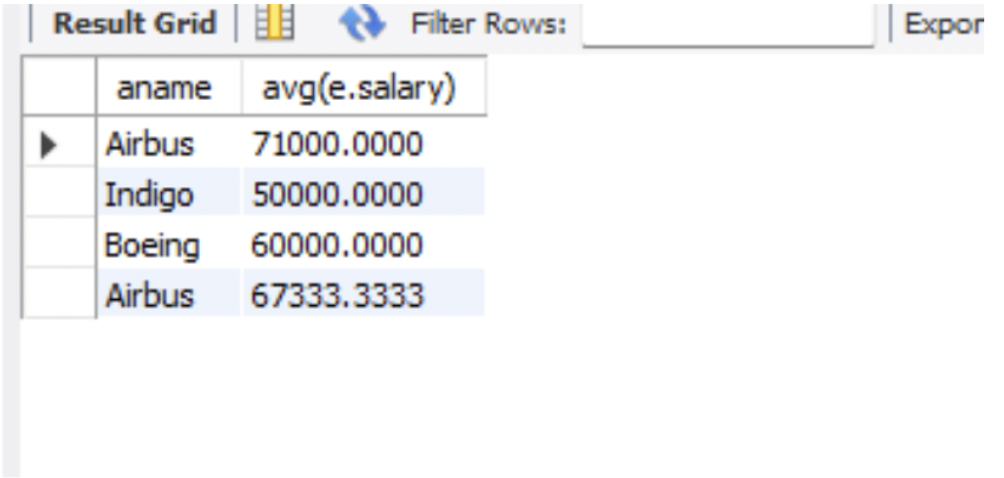


* Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.

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

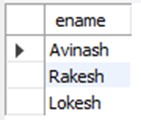
* For all aircraft with cruising range over 1000 Kms, find the name of the aircraft and the Average salary of all pilots certified for this aircraft.

SQL>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;

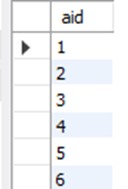


* Find the names of pilots certified for some Boeing aircraft.

SQL>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'));



* Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi. SQL>select a.aid from aircraft a where a.cruisingrange>(select distance from flights where ffrom='Banglore' and tto='New Delhi');



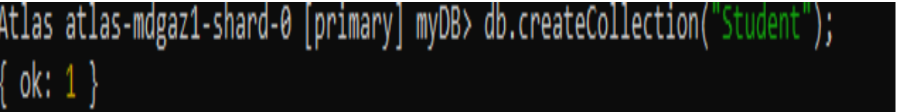
**LAB 9: NOSQL**

Perform the following DB operations using MongoDB.

1. Create a database “Student” with the following attributes

Rollno, Age, ContactNo, Email-Id.

db.createCollection("Student");



2. Insert appropriate values

db.Student.insert({RollNo:1,Age:21,Cont:9876,email:"antara.de9@gmail.com"});

db.Student.insert({RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"});

db.Student.insert({RollNo:3,Age:21,Cont:5576,email:"anubhav.de9@gmail.com"});

db.Student.insert({RollNo:4,Age:20,Cont:4476,email:"pani.de9@gmail.com"});

db.Student.insert({RollNo:10,Age:23,Cont:2276,email:"rekha.de9@gmail.com"});

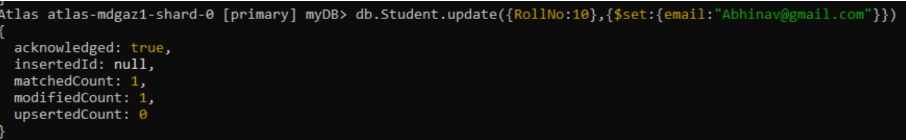
db.Student.find()



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

db.Student.update({RollNo:10},{$set:{

email:"Abhinav@gmail.com"}})



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

db.Student.insert({RollNo:11,Age:22,Name:

"ABC",Cont:2276,email:"rea.de9@gmail.com"});

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

t:{Name:"FEM"}})



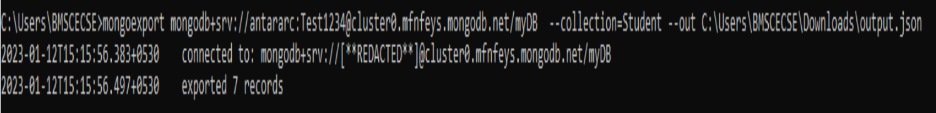
5. Export the created table into local file system

mongoexport

mongodb+srv://AISH:aish143@cluster0.mfnfeys.

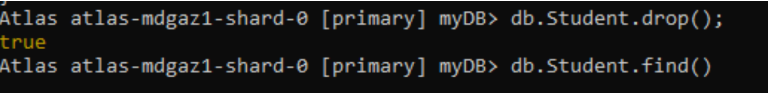
mongodb.net/myDB --collection=Student --out

C:\Users\BMSCECSE\Downloads\output.json



6. Drop the table

db.Student.drop();



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

mongodb collection.

mongoimport

mongodb+srv://AISH:aish143@cluster0.mfnfeys.m

ongodb.net/myDB --collection=New\_Student --type json

--file C:\Users\BMSCECSE\Downloads\output.json

