**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.

****

LAB REPORT

on

**DATABASE MANAGEMENT SYSTEM**

Submitted by

**NAME: NEHA BHASKAR KAMATH**

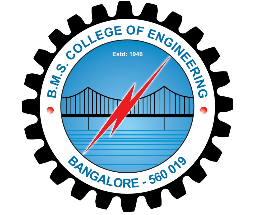
**USN: 1BM21CS113**

in partial fulfilment for the award of the degree of

**BACHELOR OF ENGINEERING**

in

**COMPUTER SCIENCE AND ENGINEERING**



**B.M.S. COLLEGE OF ENGINEERING**

**(Autonomous Institution under VTU)**

**BENGALURU-560019**

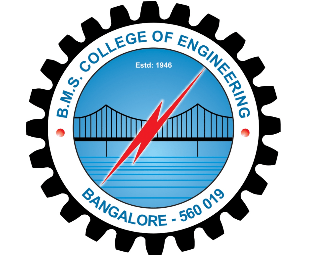
**Oct 2022-Feb 2023**

**B. M. S. College of Engineering,**

**Bull Temple Road, Bangalore 560019**

**(Affiliated To Visvesvaraya Technological University, Belgaum)**

**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled “**DATABASE MANAGEMENT SYSTEM**” carried out by **NEHA BHASKAR KAMATH(1BM21CS113)** , who is bonafide student of B. M. S. College of Engineering. It is in partial fulfilment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the year 2022-23.

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

Dr. Manjunath D R         Dr. Jyothi Nayak

Assistant professor Professor and Head

Department of CSE Department of CSE

BMSCE, Bengaluru BMSCE, Bengaluru

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No | Date | Title | Page No |
| 1 |  | Week 1: Insurance Database | 4-11 |
| 2 |  | Week 2: Queries on Insurance Database | 12-13 |
| 3 |  | Week 3: Bank Database | 14-22 |
| 4 |  | Week 4: Queries on Bank Database | 23- |
| 5 |  | Week 5: Employee Database |  |
| 6 |  | Week 6: Queries on Employee Database |  |
| 7 |  | Week 7: Supplier Database |  |
| 8 |  | Week 8: Airline Flight Database |  |

**EXPERIMENT 1- INSURANCE DATABASE**

**WEEK 1**

**Question**

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

PERSON (driver\_id: String, name: String, address: String)

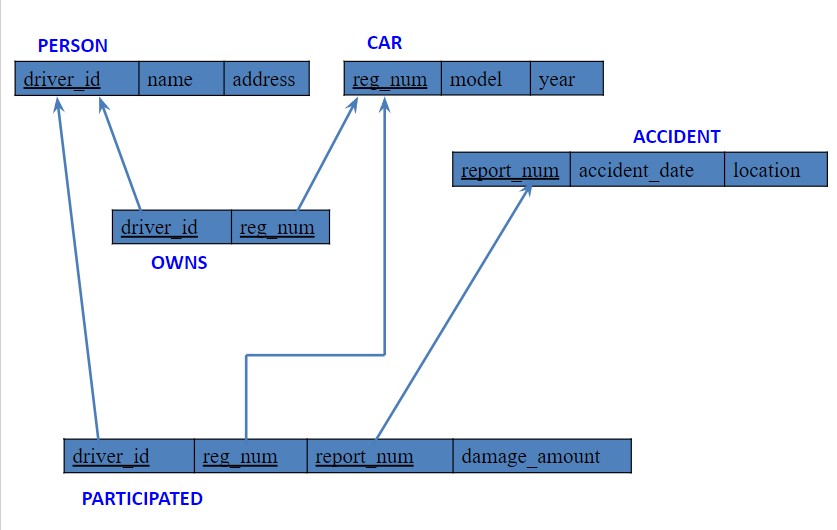
CAR (reg\_num: String, model: String, year: int)

ACCIDENT (report\_num: int, accident\_date: date, location: String)

OWNS (driver\_id: String, reg\_num: String)

PARTICIPATED (driver\_id: String,reg\_num: String, report\_num: int, damage\_amount: int)

**Schema diagram**

****

**Creating database**

create database 1BM21CS113\_Insurance;

**Using database**

use 1BM21CS113\_Insurance;

**Creating tables**

create table Person

(

driver\_id varchar (20),

name varchar (20),

address varchar (30)

);

alter table Person

add primary key(driver\_id);

create table Car

(

reg\_num varchar (15) primary key,

model varchar (20),

year int

);

create table owns

(

driver\_id varchar (20),

reg\_num varchar (15),

primary key (driver\_id, reg\_num),

foreign key(driver\_id) references Person(driver\_id),

foreign key(reg\_num) references Car(reg\_num)

);

create table Accident

(

report\_num varchar (20) primary key,

accident\_date date,

location varchar (30)

);

create table participated

(

driver\_id varchar (20),

reg\_num varchar (15),

report\_num varchar (20),

primary key (driver\_id, reg\_num, report\_num),

damage\_amount double,

foreign key(driver\_id) references Person(driver\_id),

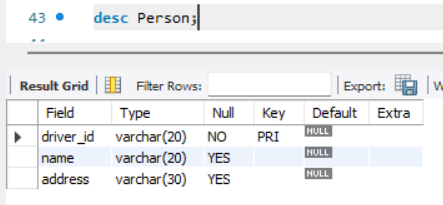
foreign key(reg\_num) references Car(reg\_num),

foreign key(report\_num) references Accident(report\_num)

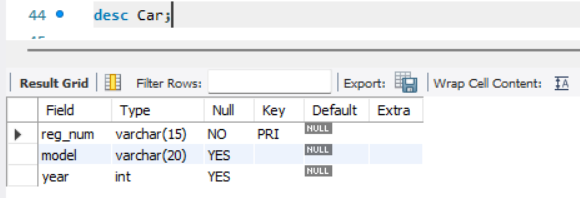
);

**Structure of the table**

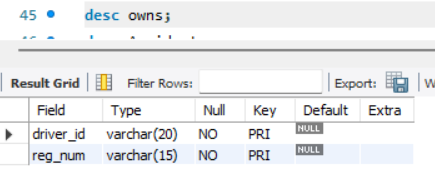
desc Person;

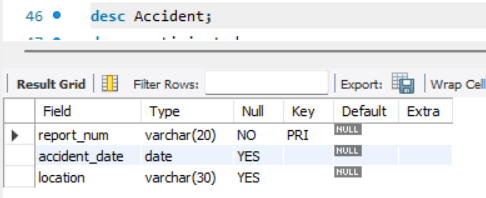


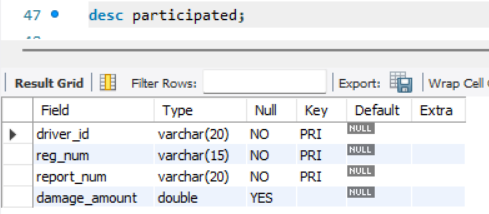
desc Car;



desc owns;



desc Accident;

desc participated;

**Inserting values into the table**

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

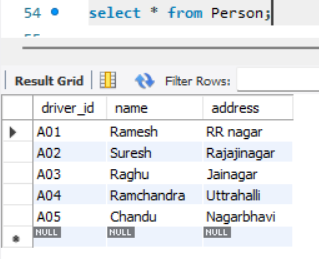
insert into Person values ('A02', 'Suresh', 'Rajajinagar');

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

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

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

select \* from Person;



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

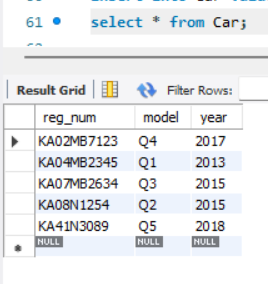
insert into Car values('KA08N1254','Q2',2015);

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

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

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

select \* from Car;



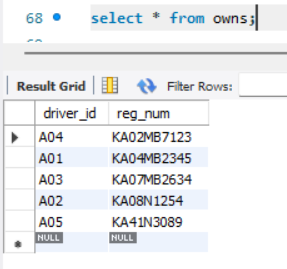
insert into owns values('A01','KA04MB2345');

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

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

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

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

select \* from owns;

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

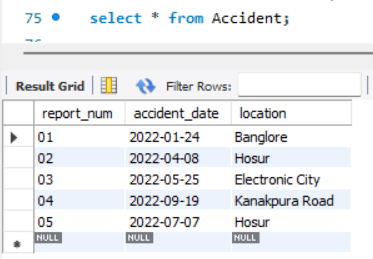
insert into Accident values('02', '2022-04-8', 'Hosur');

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

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

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

select \* from Accident;



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

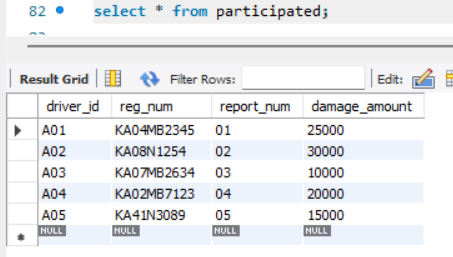
insert into participated values('A02','KA08N1254','02',30000);

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

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

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

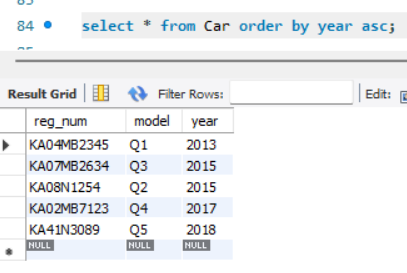
select \* from participated;



**Queries**

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

select \* from Car order by year asc;

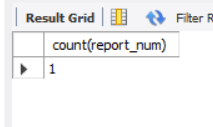


**2.Find the number of accidents in which cars belonging to a specific model (example 'Q1’)  were involved.**

select count(report\_num)

from Car c, participated p

where c.model='Q1'and c.reg\_num=p.reg\_num;

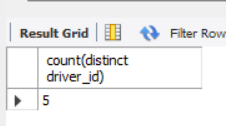


**3. Find the total number of people who owned cars that involved in accidents in 2022.**

select count(distinct driver\_id)

from participated p, accident a

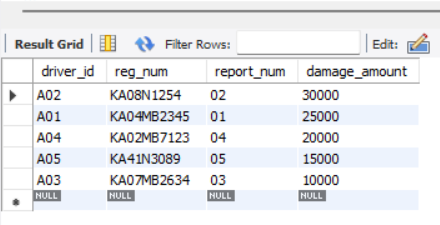
where p.report\_num=a.report\_num and a.accident\_date like '\_\_22%';



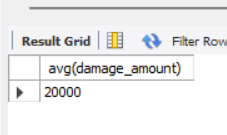
**WEEK 2- Queries on Insurance database**

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

select \* from participated order by damage\_amount desc;



**2. Find the average damaged amount.**

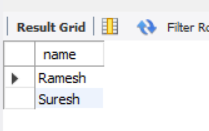
select avg(damage\_amount) from participated;

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

select p.name

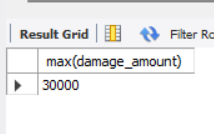
from Person p, participated par

where p.driver\_id=par.driver\_id and par.damage\_amount>(select avg(damage\_amount) from participated);



**4. Find the maximum damage amount.**

select max(damage\_amount) from participated;

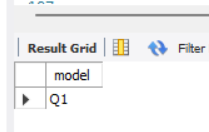


**5. Find the car model which participated in accident at location Banglore.**

select c.model

from Accident a, participated p, Car c

where p.report\_num=a.report\_num and p.reg\_num=c.reg\_num and a.location='Banglore';

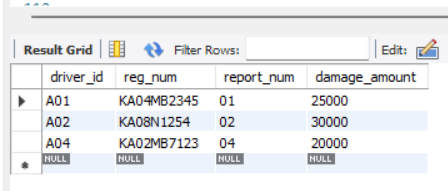


**6. Delete the tuple whose damage amount is below the average damage amount.**

delete from participated

where damage\_amount<(select t.amt from (select avg(damage\_amount) as amt from participated) t);

select \* from participated;



**EXPERIMENT 2- BANK DATABASE**

**WEEK 3**

**Question**

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

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

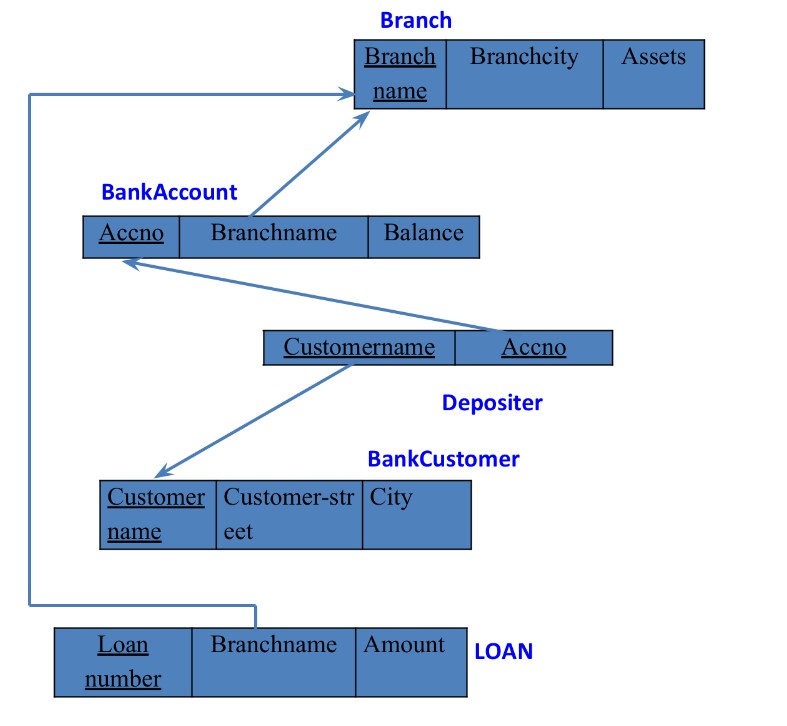
BankCustomer (customer-name: String, customer-street: String,

customer-city: String)

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

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

**Schema diagram**

****

**Creating database**

create database 1bm21cs113\_bank;

**Using database**

use 1bm21cs113\_bank;

**Creating tables**

create table branch

(

branch\_name varchar(15) primary key,

branch\_city varchar(15),

assets decimal

);

create table BankAccount

(

accno int primary key,

branch\_name varchar(15) not null,

foreign key(branch\_name) references branch(branch\_name),

balance decimal

);

create table BankCustomer

(

customer\_name varchar(25) primary key,

customer\_street varchar(20),

city varchar(20)

);

create table Depositer

(

customer\_name varchar(25),

accno int,

primary key(customer\_name,accno),

foreign key(customer\_name) references BankCustomer(customer\_name),

foreign key(accno) references BankAccount(accno)

);

create table Loan

(

loan\_no int primary key,

branch\_name varchar(15),

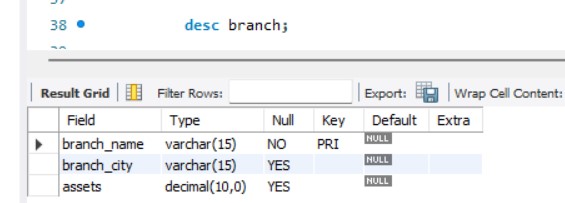
foreign key(branch\_name) references branch(branch\_name),

amount decimal

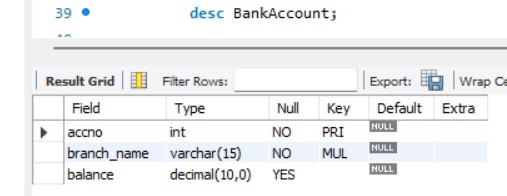
);

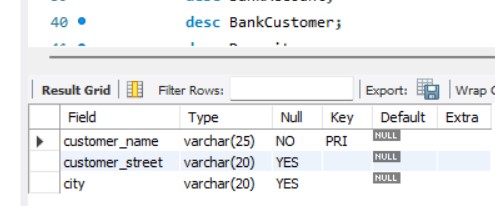
**Structure of the tables**

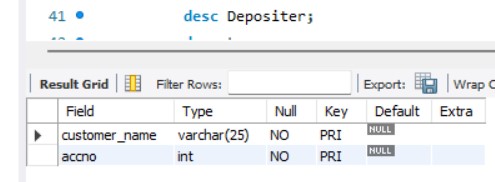
desc branch;



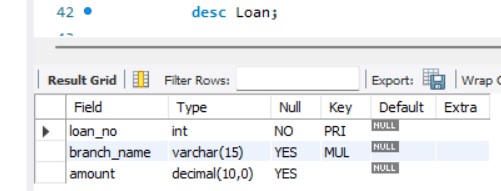
desc BankAccount;



desc BankCustomer;

desc Depositer;

desc Loan;



**Inserting values into the tables**

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

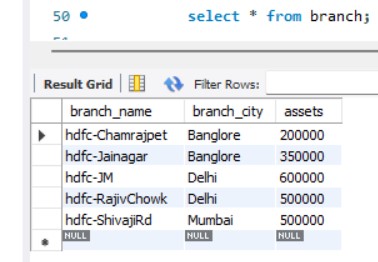
insert into branch values("hdfc-ShivajiRd","Mumbai",500000);

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

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

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

select \* from branch;



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

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

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

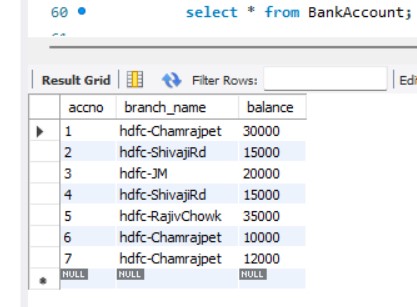
insert into BankAccount values(4,"hdfc-ShivajiRd",15000);

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

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

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

select \* from BankAccount;



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

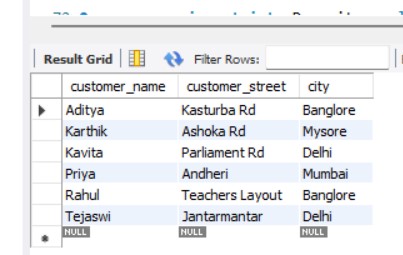
insert into BankCustomer values("Aditya","Kasturba Rd","Banglore");

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

insert into BankCustomer values("Karthik","Ashoka Rd","Mysore");

insert into BankCustomer values("Tejaswi","Jantarmantar","Delhi");

insert into BankCustomer values("Priya","Andheri","Mumbai");

select \* from BankCustomer;

insert into Depositer values("Rahul",5);

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

insert into Depositer values("Tejaswi",7);

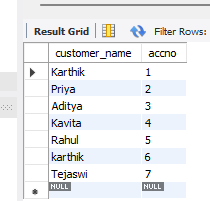
insert into Depositer values("Kavita",4);

insert into Depositer values("karthik",6);

insert into Depositer values("Karthik",1);

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

select \* from Depositer;



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

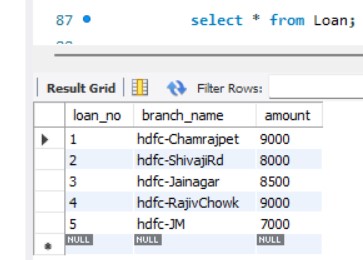
insert into Loan values(2,"hdfc-ShivajiRd",8000);

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

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

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

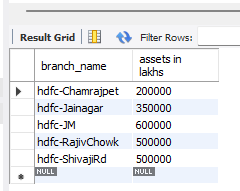
select \* from Loan;



**Queries**

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

select branch\_name, assets as "assets in lakhs"

from branch;

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

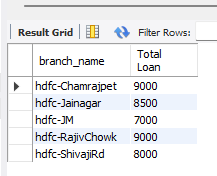
create view Total\_Loan as

select branch\_name, sum(amount) as "Total Loan"

from Loan

group by Branch\_name;

select \* from Total\_Loan;



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

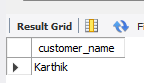
select d.customer\_name

from Depositer d, BankAccount b

where d.accno=b.accno and b.branch\_name="hdfc-Chamrajpet"

group by d.customer\_name

having count(d.accno)>=2;



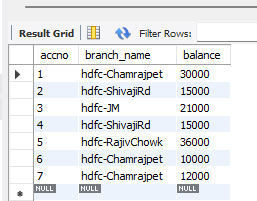
**4. Update the account balance by adding rupees 1000 for the customer residing in Banglore.**

update BankAccount b

set b.balance=b.balance+1000

where accno in(select d.accno from Depositer d, BankCustomer b

where d.customer\_name=b.customer\_name and b.city="Banglore");

select \* from BankAccount;

**WEEK 4**

**Question**

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

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

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

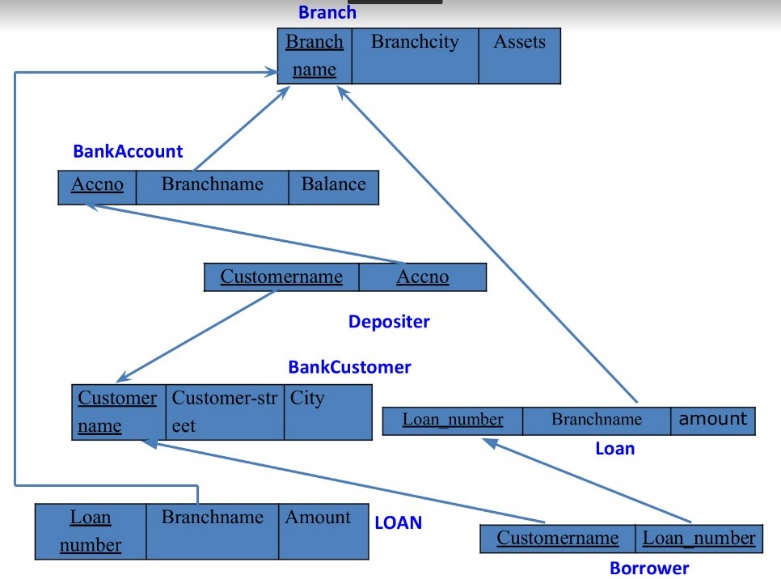
customer-city: String)

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

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

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

**Relational Schema**



**Creating table ‘Borrower’**

create table Borrower

(

customer\_name varchar(25),

loan\_no int,

primary key(customer\_name,loan\_no),

foreign key(customer\_name) references BankCustomer(customer\_name),

foreign key(loan\_no) references Loan(loan\_no)

);

**Inserting values into table ‘Borrower’**

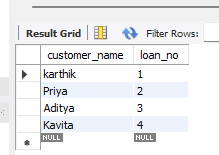
insert into Borrower values("karthik",1);

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

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

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

select \* from Borrower;



**Queries**

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

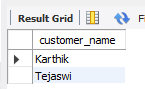
select distinct d.customer\_name

from Depositer d

where d.accno in (select b.accno

from BankAccount b, branch br

where br.branch\_name=b.branch\_name and br.branch\_city="Banglore");



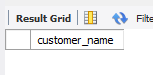
**2.Find all customers who have a loan at the bank but do not have an account.**

select customer\_name

from Borrower

where customer\_name not in(select customer\_name

from depositer);



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

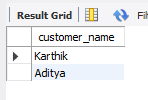
select distinct customer\_name

from depositer

where customer\_name in(select bo.customer\_name

from Borrower bo, Loan l, branch br

where l.loan\_no=bo.loan\_no and l.branch\_name=br.branch\_name and br.branch\_city="Banglore");



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

select branch\_name

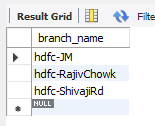
from branch

where assets >(select max(assets)

from branch

where branch\_city="Banglore"

group by branch\_city);

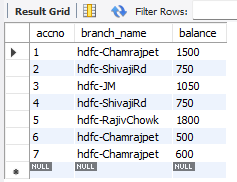


**5.** **Update the Balance of all accounts by 5%.**

update BankAccount

set balance= (5/100)\*balance;

select \* from BankAccount;



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

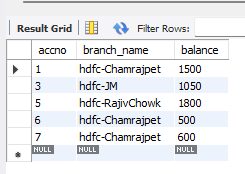
delete from Depositer

where accno in(select accno from BankAccount where branch\_name in(select branch\_name from branch where branch\_city="Mumbai"));

delete from BankAccount

where branch\_name in(select branch\_name from branch where branch\_city="Mumbai");

select \* from BankAccount;



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

delete from Depositer

where accno in(select accno from BankAccount where branch\_name in(select branch\_name from branch where branch\_city="Banglore"));

delete from BankAccount

where branch\_name in(select branch\_name from branch where branch\_city="Banglore");

delete from Loan

where branch\_name in(select branch\_name from branch where branch\_city="Banglore");

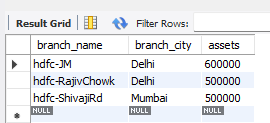
delete from Borrower

where loan\_no in(select loan\_no from Loan);

delete from branch

where branch\_city="Banglore";

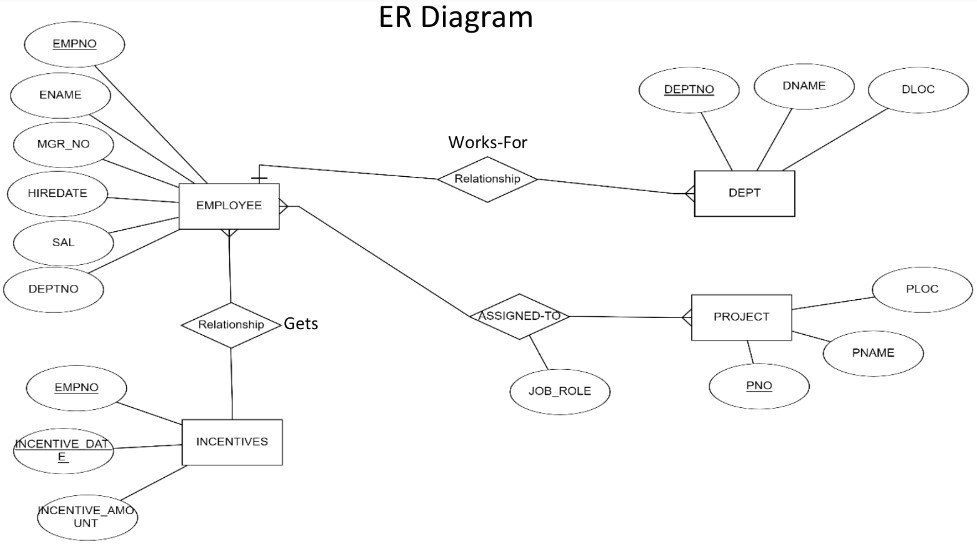
select \* from branch;



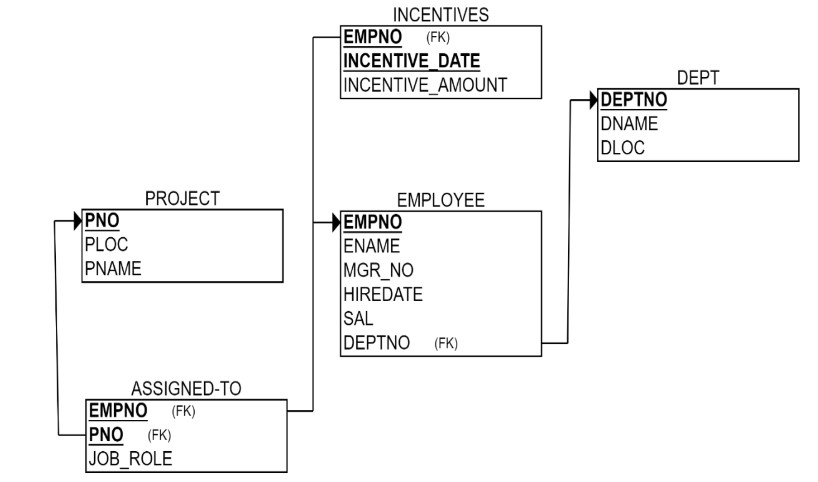
**EXPERIMENT 3- EMPLOYEE DATABASE**

**WEEK 5**

**Question**

****

**Relational Schema**

****

**Creating database**

Create database 1BM21CS113\_Employee;

**Using database**

use 1BM21CS113\_Employee;

**Creating tables**

create table Dept

(

deptno int primary key,

dname varchar(30),

dloc varchar(40)

);

create table Employee

(

empno int primary key,

ename varchar(40),

mgr\_no int,

hiredate date,

sal decimal,

deptno int,

foreign key(deptno) references Dept(deptno) on delete cascade on update cascade

);

create table Project

(

pno int primary key,

ploc varchar(40),

pname varchar(40)

);

create table Assigned\_to

(

empno int,

pno int,

primary key(empno,pno),

foreign key(empno) references Employee(empno) on delete cascade on update cascade,

foreign key(pno) references Project(pno) on delete cascade on update cascade,

job\_role varchar(30)

);

create table Incentives

(

empno int,

incentive\_date date,

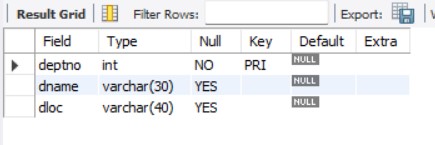
primary key(empno,incentive\_date),

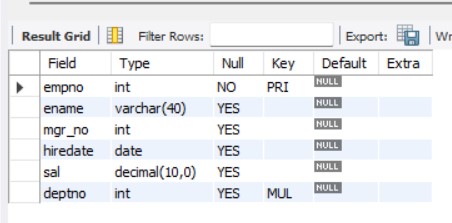
foreign key(empno) references Employee(empno) on delete cascade on update cascade,

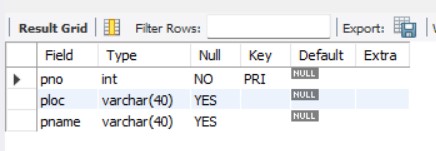
incentive\_amt decimal

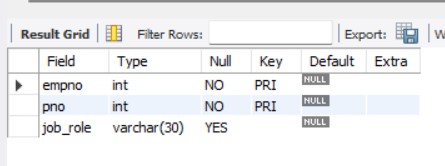
);

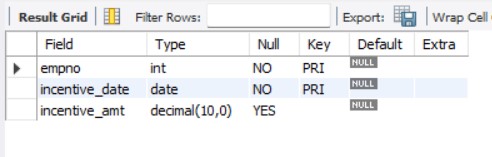
**Structure of tables**

desc Dept;

desc Employee;

desc Project;

desc Assigned\_to;

desc Incentives;

**Inserting values into tables**

insert into Dept values(01,"Accounts","Banglore");

insert into Dept values(02,"HR","Banglore");

insert into Dept values(03,"Production","Hyderabad");

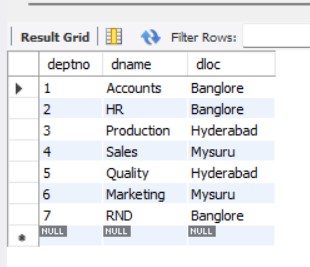
insert into Dept values(04,"Sales","Mysuru");

insert into Dept values(05,"Quality","Hyderabad");

insert into Dept values(06,"Marketing","Mysuru");

insert into Dept values(07,"RND","Banglore");

select \* from Dept;



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

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

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

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

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

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

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

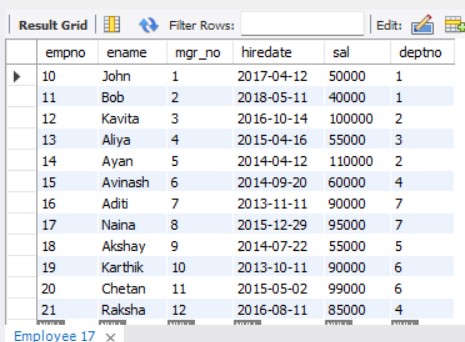
insert into Employee values(17,"Naina","008","2015-12-29",95000,07);

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

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

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

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

select \* from Employee;

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

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

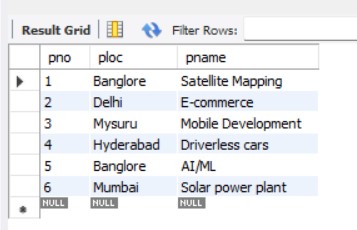
insert into Project values(3,"Mysuru","Mobile Development");

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

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

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

select \* from Project;



insert into Assigned\_to values(10,1,"Manager");

insert into Assigned\_to values(11,1,"Production head");

insert into Assigned\_to values(12,5,"Financial Advisor");

insert into Assigned\_to values(13,4,"Manager");

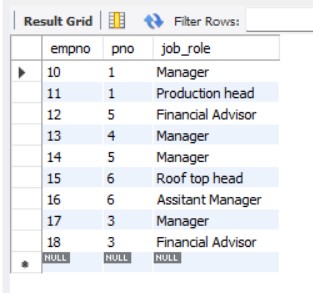
insert into Assigned\_to values(14,5,"Manager");

insert into Assigned\_to values(15,6,"Roof top head");

insert into Assigned\_to values(16,6,"Assitant Manager");

insert into Assigned\_to values(17,3,"Manager");

insert into Assigned\_to values(18,3,"Financial Advisor");

select \* from Assigned\_to;

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

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

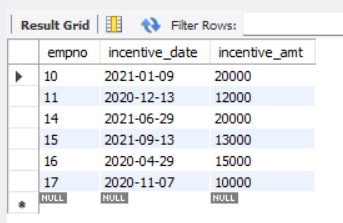
insert into Incentives values(15,"2021-09-13",13000);

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

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

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

select \* from Incentives;



**Queries**

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

select e.empno, e.ename

from Employee e,Project p, Assigned\_to a

where e.empno=a.empno and

a.pno=p.pno and

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

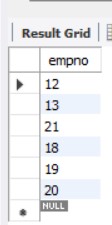


**2. Get the employee numbers of all employees who did not receive incentives.**

select empno

from Employee

where empno not in(select empno from Incentives);



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

select e.ename,e.empno,d.dname,a.job\_role,d.dloc,p.ploc

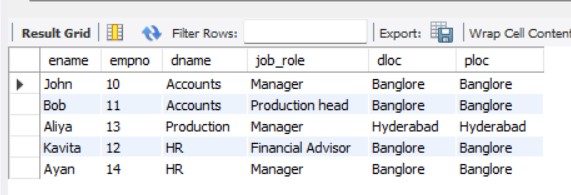
from Employee e, Dept d, Assigned\_to a,Project p

where e.deptno=d.deptno and

a.empno=e.empno and

a.pno=p.pno and

p.ploc=d.dloc;



**WEEK 6**

**Inserting more values in the table ‘Employee’**

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

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

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

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

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

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

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

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

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

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

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

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

insert into Employee values(006,"Ramya",001,"2017-09-09",36000,01);

insert into Employee values(007,"Nikhil",001,"2018-10-10",67000,01);

insert into Employee values(001,"Neha",004,"2014-11-11",100000,03);

insert into Employee values(002,"Ram",004,"2016-07-12",56000,03);

insert into Employee values(003,"Ranvijay",002,"2019-08-08",34000,04);

insert into Employee values(004,"Raftaar",001,"2018-04-03",45000,05);

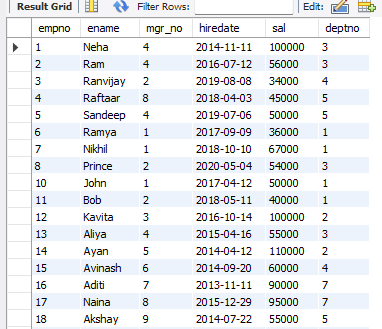
insert into Employee Values(005,"Sandeep",004,"2019-07-06",50000,05);

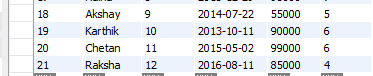
insert into Employee values(008,"Prince",002,"2020-05-04",54000,03);

update Employee

set mgr\_no=008 where ename="Raftaar";

select \* from Employee;





**Queries**

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

select m.ename

from Employee e, Employee m

where e.mgr\_no=m.empno

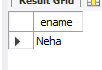
group by m.ename

having count(\*)=(select max(MyCount)

from (select count(\*) as MyCount

from Employee

group by mgr\_no)a);



\

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

select m.ename

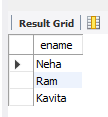
from Employee m

where m.sal>= (select avg(e.sal)

from Employee e

where m.empno=e.mgr\_no

group by e.mgr\_no);



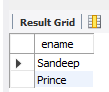
**3. Display the names of the employees who are working in the same department his manager is working.**

select e.ename

from Employee e,Employee m

where e.mgr\_no=m.empno and

e.deptno=m.deptno;



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

select \* from

Employee e, Incentives i

where e.empno=i.empno and

1=(select count(distinct j.incentive\_amt)

from Incentives j

 where i.incentive\_amt<j.incentive\_amt and j.incentive\_date like "2020%") and i.incentive\_date like "2020%" ;

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

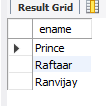
select g. ename

from Employee e, Employee m, Employee g

where e.mgr\_no=m. empno and

m.mgr\_no=g.empno

group by g.deptno;



**EXPERIMENT 4- SUPPLIER DATABASE**

**WEEK 7**

**Question**

Consider the following schema:

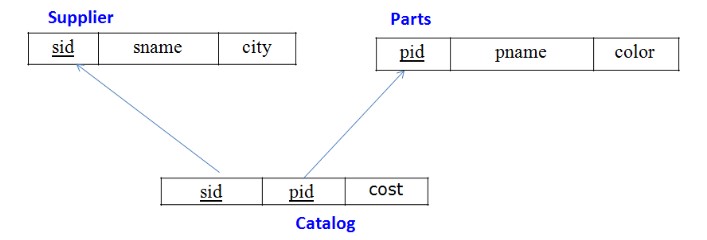
SUPPLIERS(sid: integer, sname: string, address: string)

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

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

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

**Relational Schema**

****

**Creating database**

create database 1BM21CS113\_Supplier;

**Using database**

use 1BM21CS113\_Supplier;

**Creating tables**

create table Supplier

(

sid int primary key,

sname varchar(40),

city varchar(40)

);

create table Parts

(

pid int primary key,

pname varchar(30),

color varchar(30)

);

create table Catalog

(

sid int,

pid int,

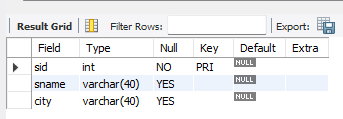
primary key(sid,pid),

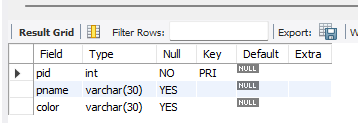
foreign key(sid) references Supplier(sid) on delete cascade on update cascade,

foreign key(pid) references Parts(pid) on delete cascade on update cascade

);

**Structure of tables**

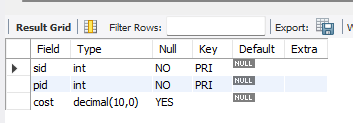
desc Supplier;

desc Parts;

Alter table Catalog

add column cost decimal;

desc Catalog;



**Inserting values into the tables**

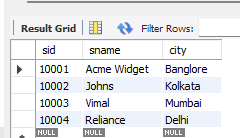
insert into Supplier values(10001,"Acme Widget","Banglore");

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

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

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

select \* from Supplier;



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

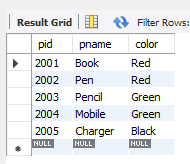
insert into Parts values(2002,"Pen","Red");

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

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

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

select \* from Parts;



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

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

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

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

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

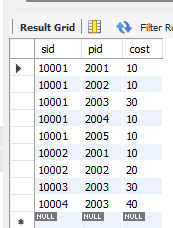
insert into Catalog values(10002,2001,10);

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

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

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

select \* from Catalog;



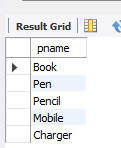
**Queries**

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

select distinct p.pname

from Parts p, Supplier s, Catalog c

where s.sid=c.sid and p.pid=c.pid;



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

select distinct s.sname

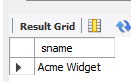
from Supplier s

where s.sid in(select sid

from Catalog

group by sid

having count(\*)=(select count(\*) from Parts));

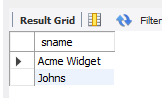


**3. Find the snames of suppliers who supply every red part.**

select distinct s.sname

from Parts p, Supplier s, Catalog c

where s.sid=c.sid and p.pid=c.pid and p.color="Red";



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

select distinct p.pname

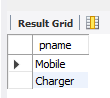
from Parts p, Catalog c

where c.sid=(select sid from Supplier where sname="Acme Widget")

and p.pid=c.pid and c.pid not in(select c.pid

from Catalog c

where c.sid!=(select sid from Supplier where sname="Acme Widget"));



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

select c.sid

from Catalog c

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

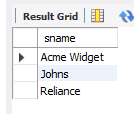


**6. For each part, find the sname of the supplier who charges the most for that part.**

select s.sname

from Supplier s

where s.sid in(select c.sid from Catalog c where c.cost=(select max(cost)

 from Catalog c1 where c.pid=c1.pid));

**EXPERIMENT 5- AIRLINE FLIGHT DATABASE**

**WEEK 8**

**Question**

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

price: integer)

AIRCRAFT(aid: integer, aname: string, cruising\_range: integer)

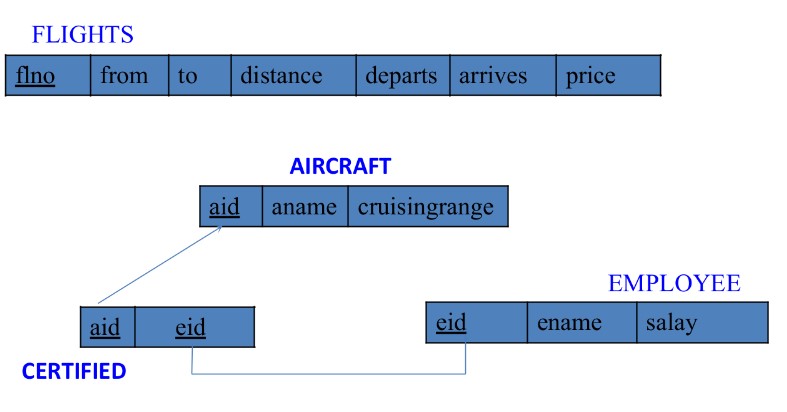
CERTIFIED(eid: integer, aid: integer)

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

Note that the Employees relation describes pilots and other kinds of employees as well; Every

pilot is certified for some aircraft, and only pilots are certified to fly.

**Relational Schema**

****

**Creating database**

create database 1bm21cs113\_Week8\_Flight;

**Using database**

use 1bm21cs113\_Week8\_FLight;

**Creating tables**

create table Flights

(

flno int primary key,

dep\_airport varchar(40),

arr\_airport varchar(40),

distance decimal,

departs time,

arrives time,

price decimal

);

create table Aircraft

(

aid int primary key,

aname varchar(40),

cruisingrange int

);

create table Employee

(

eid int primary key,

ename varchar(50),

salary decimal

);

create table certified

(

aid int,

eid int,

primary key(aid,eid),

foreign key(eid) references Employee(eid),

foreign key(aid) references Aircraft(aid)

);

Alter table certified

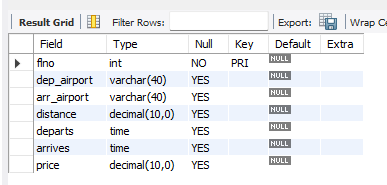
add foreign key(eid) references Employee(eid) on delete cascade on update cascade;

Alter table certified

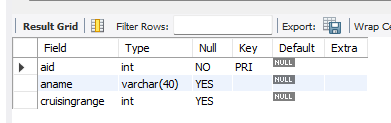
add foreign key(aid) references Aircraft(aid) on delete cascade on update cascade;

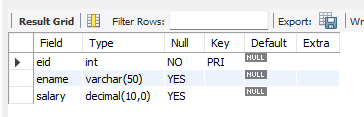
**Structure of tables**

desc Flights;

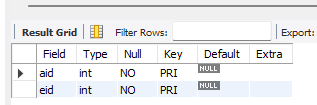


desc Aircraft;

desc Aircraft;

desc Employee;

desc certified;



**Inserting values into the tables**

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

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

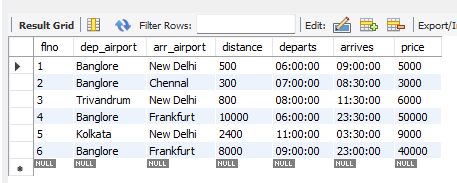
insert into Flights values(3,"Trivandrum","New Delhi",800,"8:00","11:30",6000);

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

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

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

select \* from Flights;



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

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

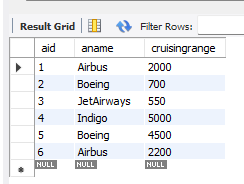
insert into Aircraft values(3,"JetAirways",550);

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

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

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

select \* from Aircraft;



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

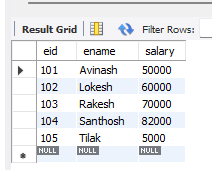
insert into Employee values(102,"Lokesh",60000);

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

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

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

select \* from Employee;

  
  
  
  
  
  
  
  
  
  
  
insert into certified values(2,101);

insert into certified values(4,101);

insert into certified values(5,101);

insert into certified values(6,101);

insert into certified values(1,102);

insert into certified values(3,102);

insert into certified values(5,102);

insert into certified values(2,103);

insert into certified values(3,103);

insert into certified values(5,103);

insert into certified values(6,103);

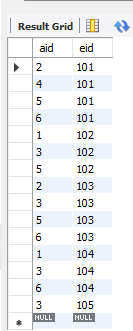
insert into certified values(6,104);

insert into certified values(1,104);

insert into certified values(3,104);

insert into certified values(3,105);

select \* from certified;



**Queries**

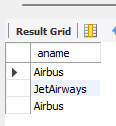
**1. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.**

select a.aname

from Aircraft a, certified c, Employee e

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

and e.salary>80000;



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

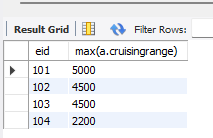
select c.eid, max(a.cruisingrange)

from Aircraft a, certified c, Employee e

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

group by c.eid

having count(c.aid)>=3;



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

**Bengaluru to Frankfurt.**

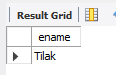
select e.ename

from Employee e

where e.salary< all(select min(price)

from Flight

where dep\_airport="Banglore" and arr\_airport="Frankfurt");

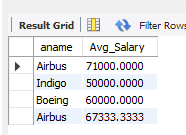
  
  
  
  
  
  
**4. For all aircraft with cruising range over 1000 Kms, find the name of the aircraft and the average salary of all pilots certified for this aircraft.**

select a.aname, avg(e.salary) as Avg\_Salary

from Aircraft a, certified c, Employee e

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

group by c.aid;

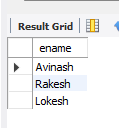


**5. Find the names of pilots certified for some Boeing aircraft.**

select distinct e.ename

from Aircraft a, certified c, Employee e

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



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

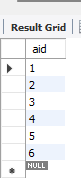
select distinct a.aid

from Aircraft a

where a.cruisingrange >any(select distance

from Flights

where dep\_airport="Banglore" and arr\_airport="New Delhi");



**EXPERIMENT 6- STUDENT DATABASE USING NoSQL**

**WEEK 9**

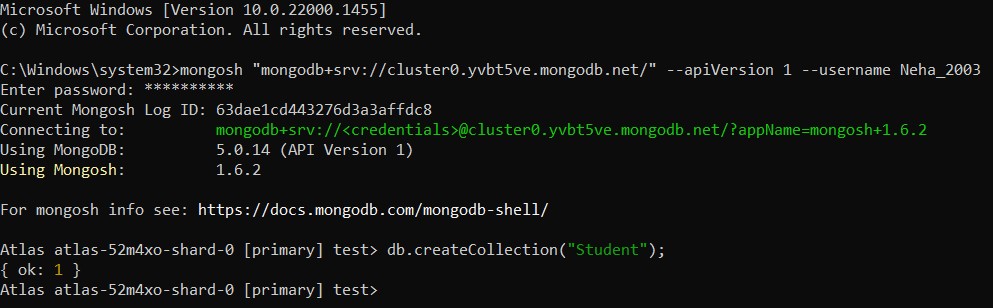
**Question**

Perform the database operations using MongoDB

**Queries**

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

db. createCollection(“Student”);

****

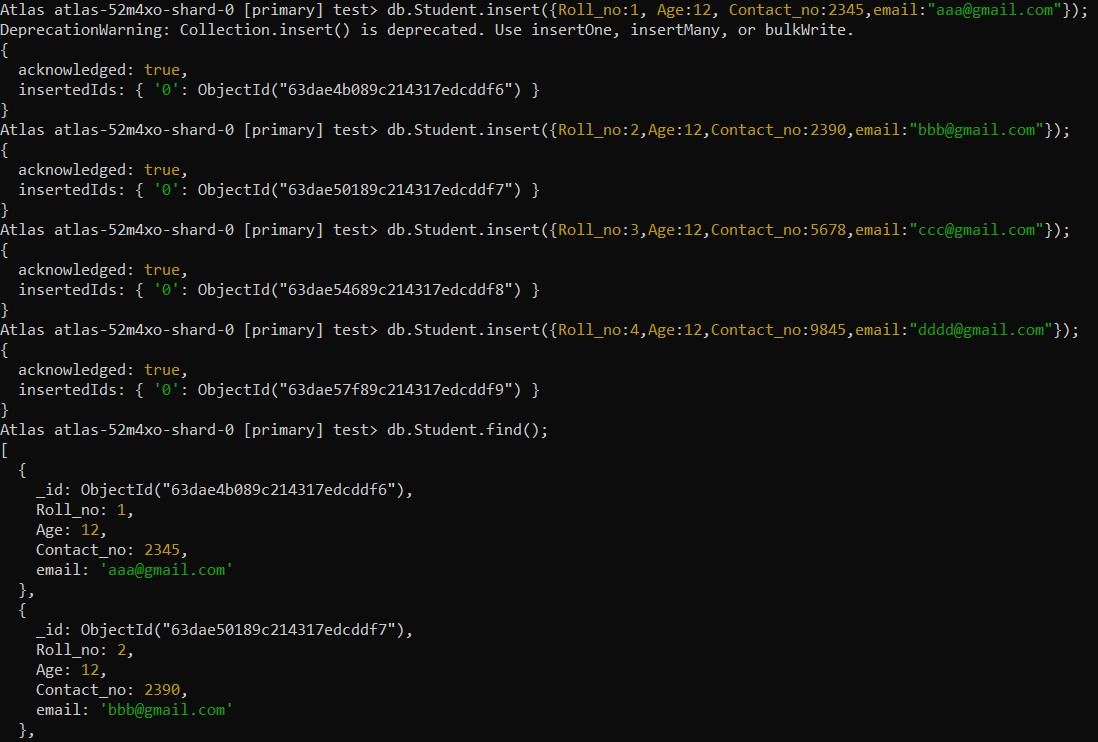
**2. Insert appropriate values.**

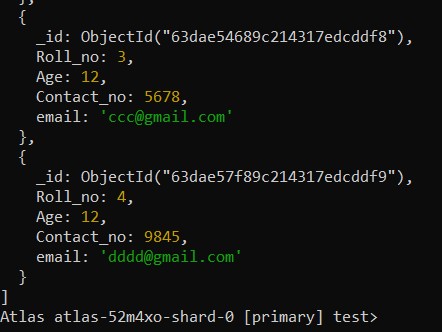
db.Student.insert({Roll\_no:1,Age:12,Contact\_no:2345,email:”aaa@gmail”});

db.Student.insert({Roll\_no:2,Age:12,Contact\_no:2390,email:”bbb@gmail”});

db.Student.insert({Roll\_no:3,Age:12,Contact\_no:5678,email:”ccc@gmail”});

db.Student.insert({Roll\_no:4,Age:12,Contact\_no:9845,email:”dddd@gmail”});

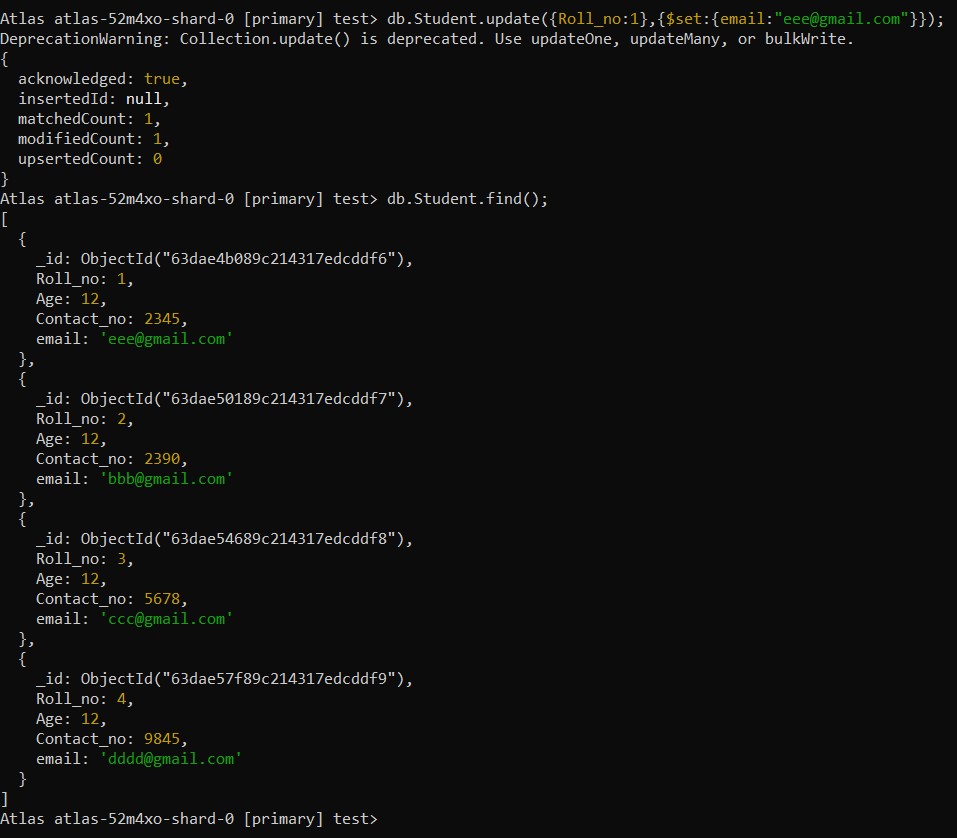




**3. Write query to update email if of a student with Roll\_no 1.**

db.Student.update({Roll\_no:1},{$set:{email:”eee@gmail”}});

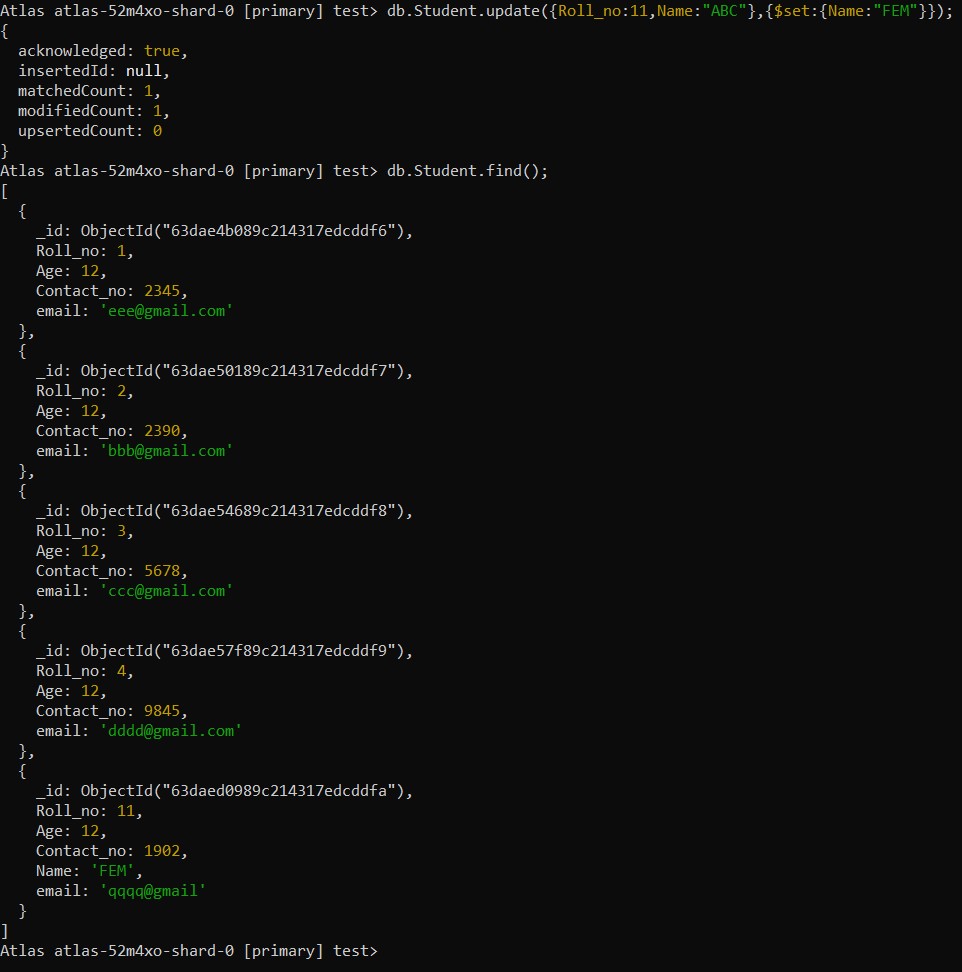
db.Student.find();



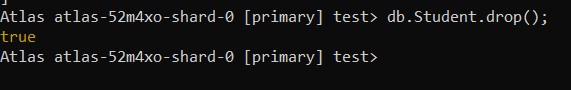
**4. Replace the student name from “ABC” to “FEM” of Roll\_no 6.**

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

db.Student.find();



**5. Drop the collection Student.**

Db.Student.drop();