VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
On

DATABASE MANAGEMENT SYSTEMS

Submitted by

PALLE PADMAVATHI (1BM21CS125)

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
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 SYSTEMS"** carried out by **PALLE PADMAVATHI(1BM21CS125)**, 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 2022-23. The Lab report has been approved as it satisfies the academic requirements in respect of Database Management Systems Lab - **(22CS3PCDBM)**work prescribed for the said degree.

Dr SEEMA PATIL
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

Index

Course outcomes:

CO1	Apply the concepts of database management system for various applications.
CO2	Analyse database concepts for a given problem.
CO3	Design conceptual data model for database applications.
CO4	Demonstrate SQL commands to create, manipulate and query data in a database.

1.Insurance Database

Consider the Insurance database given below. The primary keys are underlined and the data types are specified.

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, damageamount: int) i. Create the above tables by properly specifying the primary keys and the foreign keys. ii. Enter at least five tuples for each relation.

iii. Demonstrate how you

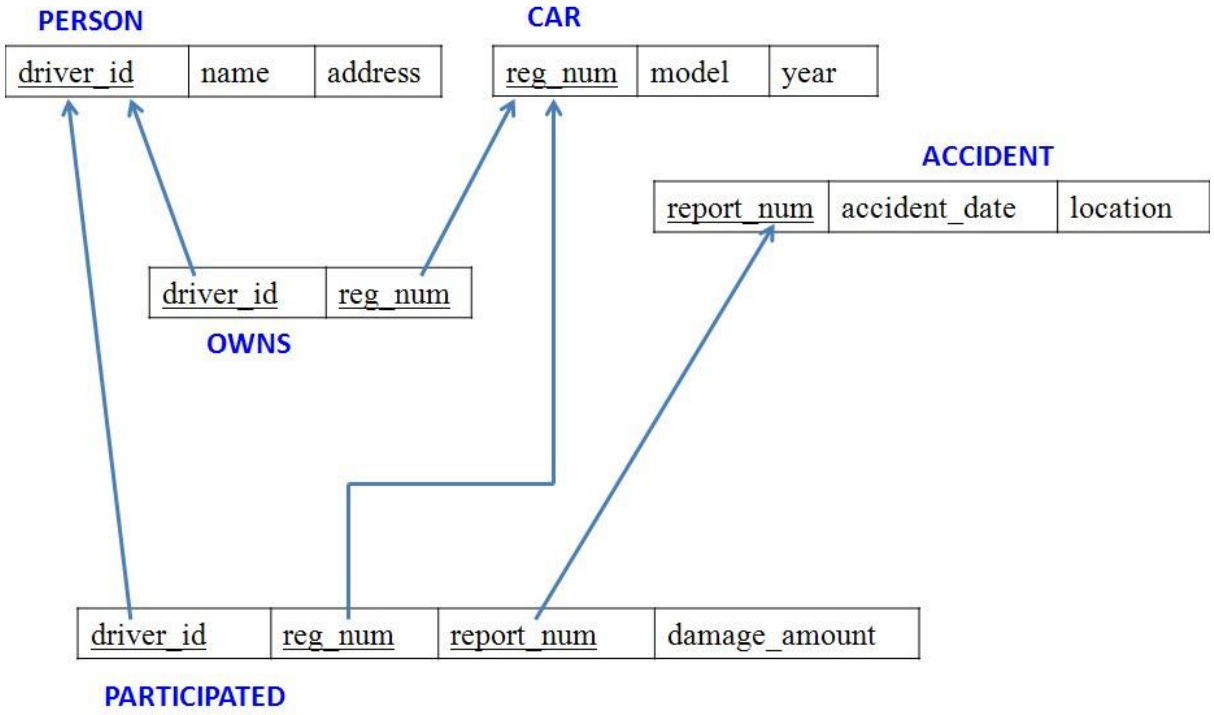
a. Update the damage amount for the car with a specific Regno in the accident with report number 12 to 25000.

b. Add a new accident to the database.

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

v. Find the number of accidents in which cars belonging to a specific model were involved.

Schema diagram:



Create database:

```
create database INSURANCE_1BM21CS083; use
INSURANCE_1BM21CS083;
```

Create table:

```
create table PERSON
(
    driver_id varchar(10),
d_Name varchar(30) not null,
address varchar(50),
   primary key(driver_id)
);
create table CAR
(
   reg_no varchar(10),
model varchar(20),
m_year int,     primary
key(reg_no)
);
create table ACCIDENT
```

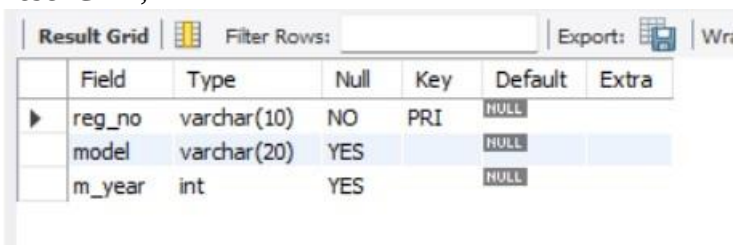
```
(
   report_no int,
accident_date date,     location
varchar(20),
   primary key(report_no)
);
create table OWNS
(
   driver_id varchar(10),     reg_no varchar(10),
primary key(driver_id,reg_no),     foreign key(driver_id)
references PERSON(driver_id),
   foreign key(reg_no) references CAR(reg_no)
);
create table PARTICIPATED
(
   driver_id varchar(10),
reg_no varchar(10),
report_no int,
damage_amount int,
   primary key(driver_id,reg_no,report_no),     foreign
key (driver_id) references PERSON(driver_id),
foreign key(reg_no) references CAR(reg_no),
   foreign key(report_no) references ACCIDENT(report_no)
);
```

desc PERSON;

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

Field	Type	Null	Key	Default	Extra
driver_id	varchar(10)	NO	PRI	NULL	
d_Name	varchar(30)	NO		NULL	
address	varchar(50)	YES		NULL	

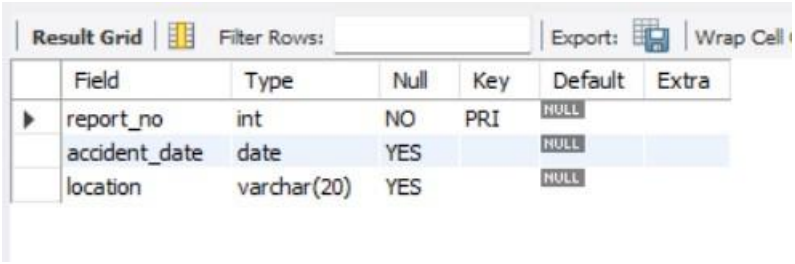
Desc CAR;



Result Grid | Filter Rows: | Export: | Wr

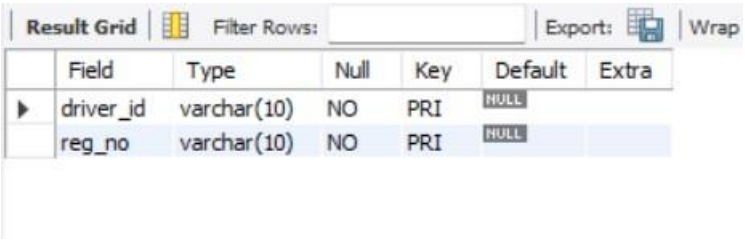
Field	Type	Null	Key	Default	Extra
reg_no	varchar(10)	NO	PRI	NULL	
model	varchar(20)	YES		NULL	
m_year	int	YES		NULL	

Desc ACCIDENT;



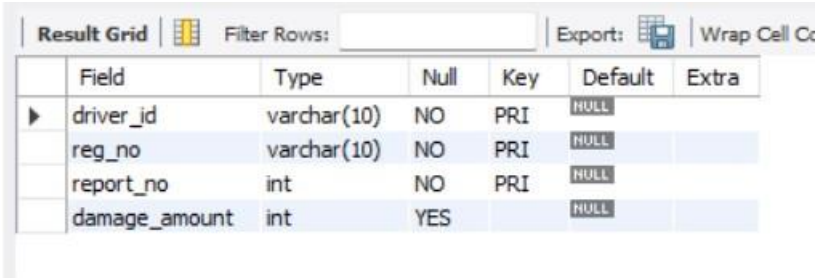
Field	Type	Null	Key	Default	Extra
report_no	int	NO	PRI	NULL	
accident_date	date	YES		NULL	
location	varchar(20)	YES		NULL	

Desc OWNS;



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

Desc PARTICIPATED;

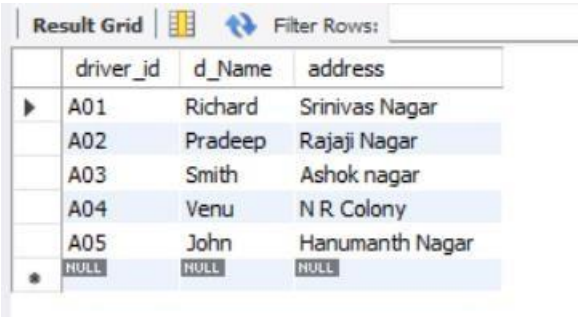


Field	Type	Null	Key	Default	Extra
driver_id	varchar(10)	NO	PRI	NULL	
reg_no	varchar(10)	NO	PRI	NULL	
report_no	int	NO	PRI	NULL	
damage_amount	int	YES		NULL	

Insert Values to the table

insert into PERSON
values('A01','Richard','Srinivas Nagar'),('A02','Pradeep','Rajaji Nagar'),('A03','Smith','Ashok nagar'),('A04','Venu','N R Colony'),('A05','John','Hanumanth Nagar');

SELECT *FROM PERSON;

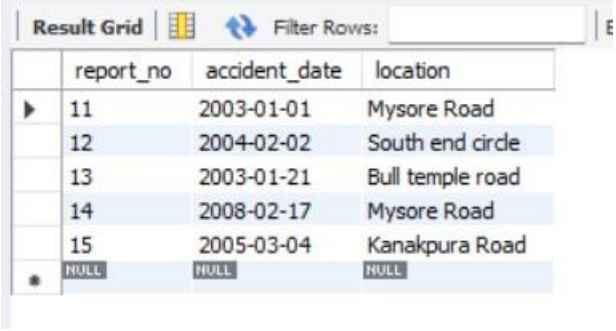


driver_id	d_Name	address
A01	Richard	Srinivas Nagar
A02	Pradeep	Rajaji Nagar
A03	Smith	Ashok nagar
A04	Venu	N R Colony
A05	John	Hanumanth Nagar
NULL	NULL	NULL

insert into CAR
values('KA052250','Indica',1990),('KA031181','Lancer',1957),('KA095477','Toyota',1998),('KA053408','Honda',2008),('KA041702','Audi',2005);
SELECT *FROM CAR;

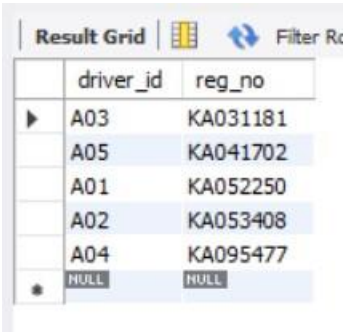
reg_no	model	m_year
KA031181	Lancer	1957
KA041702	Audi	2005
KA052250	Indica	1990
KA053408	Honda	2008
KA095477	Toyota	1998
NULL	NULL	NULL

insert into ACCIDENT
values(11,'2003-01-01','Mysore Road'),(12,'2004-02-02','South end circle'),(13,'2003-01-21','Bull temple road'),(14,'2008-02-17','Mysore Road'),(15,'2005-03-04','Kanakpura Road');
SELECT *FROM ACCIDENT;



report_no	accident_date	location
11	2003-01-01	Mysore Road
12	2004-02-02	South end circle
13	2003-01-21	Bull temple road
14	2008-02-17	Mysore Road
15	2005-03-04	Kanakpura Road
NULL	NULL	NULL

insert into OWNS
values('A01','KA052250'),('A02','KA053408'),('A03','KA031181'),('A04','KA095477'),('A05','KA041702');
SELECT *FROM OWNS;



driver_id	reg_no
A03	KA031181
A05	KA041702
A01	KA052250
A02	KA053408
A04	KA095477
NULL	NULL

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

Result Grid | Filter Rows: | Edit:

driver_id	reg_no	report_no	damage_amount
A01	KA052250	11	10000
A02	KA053408	12	50000
A03	KA095477	13	25000
A04	KA031181	14	3000
A05	KA041702	15	5000
NULL	NULL	NULL	NULL

Display Accident date and location SELECT
accident_date, location
from ACCIDENT;

Result Grid | Filter Rows:

accident_date	location
2003-01-01	Mysore Road
2004-02-02	South end circle
2003-01-21	Bull temple road
2008-02-17	Mysore Road
2005-03-04	Kanakpura Road

Display driver id who did the accident damage greater than or equal to Rs.25000

SELECT driver_id from
PARTICIPATED WHERE
damage_amount>=25000;

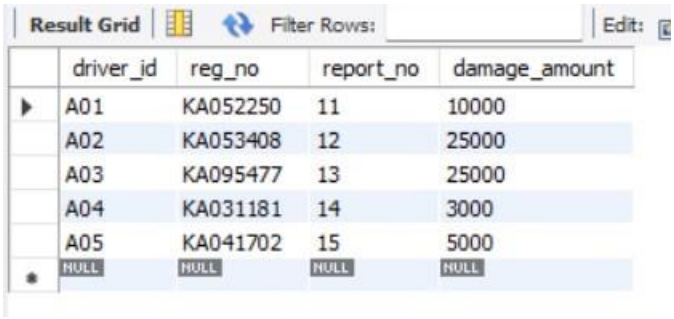
Result Grid |

driver_id
A02
A03

2.More Queries on Insurance Database

Update the damage amount for the car with a specific Regno in the accident with report number 12 to 25000.

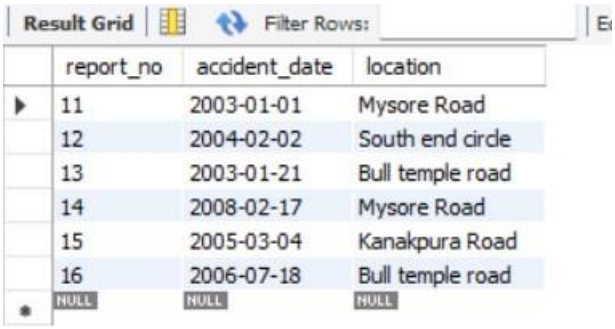
```
UPDATE PARTICIPATED SET
damage_amount=25000 where
report_no=12; select *FROM
PARTICIPATED;
```



driver_id	reg_no	report_no	damage_amount
A01	KA052250	11	10000
A02	KA053408	12	25000
A03	KA095477	13	25000
A04	KA031181	14	3000
A05	KA041702	15	5000
NULL	NULL	NULL	NULL

Add a new accident to the database.

```
insert into ACCIDENT
VALUES(16, '2006-07-18','Bull temple road');
select *from ACCIDENT;
```



report_no	accident_date	location
11	2003-01-01	Mysore Road
12	2004-02-02	South end circle
13	2003-01-21	Bull temple road
14	2008-02-17	Mysore Road
15	2005-03-04	Kanakpura Road
16	2006-07-18	Bull temple road
NULL	NULL	NULL

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

```
SELECT *
FROM CAR
ORDER BY m_year asc;
```

Result Grid | Filter Rows:

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

Find the number of accidents in which cars belonging to a specific model (‘Lancer’) were involved.

```
select count(report_no)
from CAR,PARTICIPATED
where CAR.reg_no=PARTICIPATED.reg_no AND model='Lancer';
```

Result Grid | Filte

count(report_no)
1

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

```
select count(reg_no)
from ACCIDENT,PARTICIPATED
WHERE ACCIDENT.report_no=PARTICIPATED.report_no and accident_date like
'2008%';
```

Result Grid | F

count(reg_no)
1

Find the average damage amount

```
select AVG(damage_amount)
from PARTICIPATED;
```

Result Grid | Filter Rows

AVG(damage_amount)
13600.0000

Delete the tuples from PARTICIPATED relation whose damage amount is below the average damage amount.

```
DELETE FROM PARTICIPATED
WHERE damage_amount<13600;
SELECT *FROM PARTICIPATED;
```

	driver_id	reg_no	report_no	damage_amount
▶	A02	KA053408	12	25000
	A03	KA095477	13	25000
*	NULL	NULL	NULL	NULL

3. Bank Database

Consider the following database for a banking enterprise.

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)

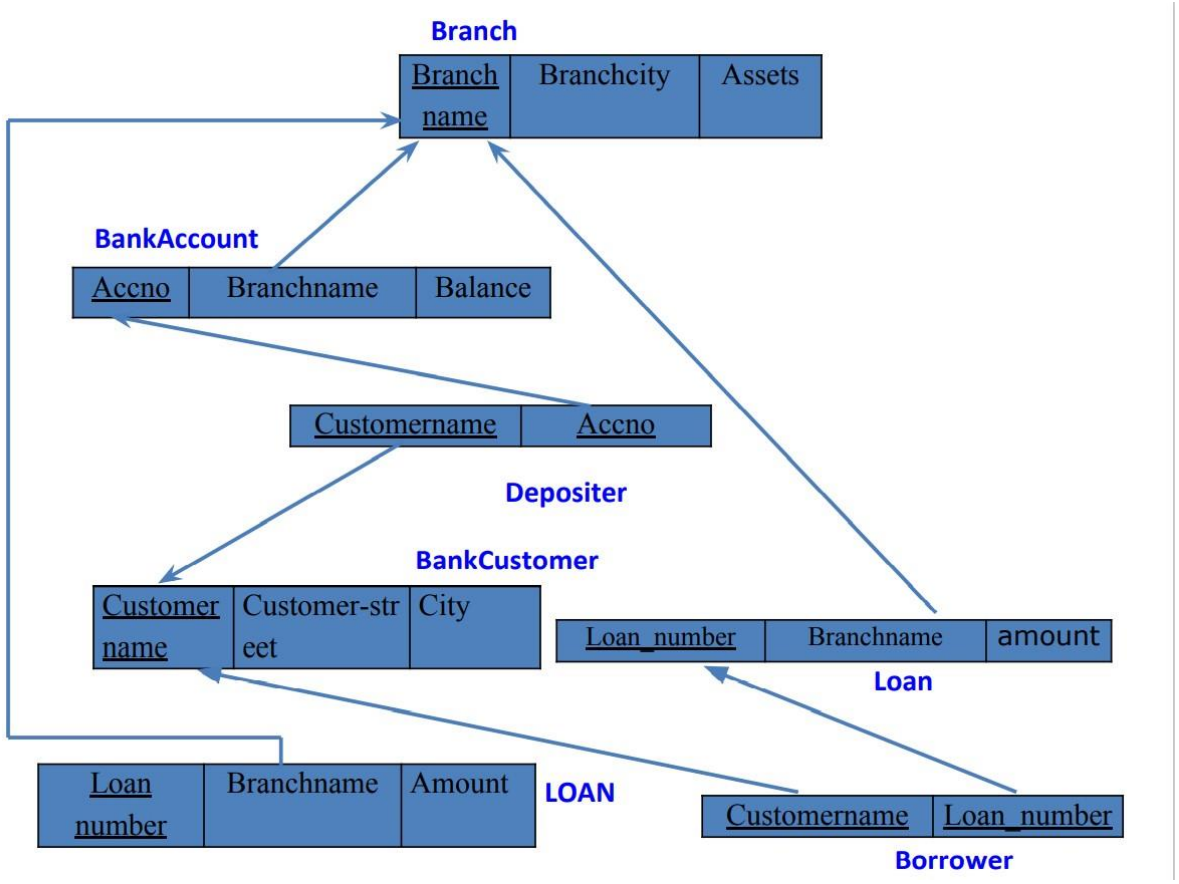
LOAN (loan-number: int, branch-name: String, amount: real)

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

i. Create the above tables by properly specifying the primary keys and the foreign keys. ii. Enter at least five tuples for each relation. iii. Find all the customers who have at least two accounts at the Main branch.

iv. Find all the customers who have an account at all the branches located in a specific city.

v. Demonstrate how you delete all account tuples at every branch located in a specific city.




```
create database BANK_1BM21CS083;
USE BANK_1BM21CS083;


create table Branch
(
   Branch_name varchar(30),
   Branch_city varchar(20),
   Assets real,
   primary key(Branch_name)
);
create table Bank_account
(
   accno int,
   Branch_name
varchar(30),     Balance real,
primary key(accno),
   foreign key(Branch_name) references Branch(Branch_name)
);
create  table Bank_customer
(
```

```
  Customer_name varchar(20),
Customer_str varchar(30),
city varchar(20),
   primary key(Customer_name)
);
create table Depositor
(
   Customer_name varchar(20),
accno int,
   primary key(Customer_name,accno),
   foreign key(Customer_name) references Bank_customer(Customer_name),
foreign key(accno) references Bank_account(accno)
);
create  table Loan
(
   loanno int,
   Branch_name varchar(30),    amount real,    primary
key(loanno),    foreign key(Branch_name) references
Branch(Branch_name)
);
create table Borrower
(
   Customer_name varchar(20),
loanno int,
   primary key(Customer_name,loanno),
   foreign key(Customer_name) references Bank_customer(Customer_name),
foreign key(loanno) references Loan(loanno)
);
```

desc Branch;

Field	Type	Null	Key	Default	Extra
Branch_name	varchar(30)	NO	PRI	NULL	
Branch_city	varchar(20)	YES		NULL	
Assets	double	YES		NULL	

desc Bank_account;

Field	Type	Null	Key	Default	Extra
accno	int	NO	PRI	NULL	
Branch_name	varchar(30)	YES	MUL	NULL	
Balance	double	YES		NULL	

desc Bank_customer;

Field	Type	Null	Key	Default	Extra
Customer_name	varchar(20)	NO	PRI	NULL	
Customer_str	varchar(30)	YES		NULL	
city	varchar(20)	YES		NULL	

desc Depositor;

	Field	Type	Null	Key	Default	Extra
▶	Customer_name	varchar(20)	NO	PRI	NULL	
	accno	int	NO	PRI	NULL	

desc Loan;

	Field	Type	Null	Key	Default	Extra
▶	loanno	int	NO	PRI	NULL	
	Branch_name	varchar(30)	YES	MUL	NULL	
	amount	double	YES		NULL	

desc Borrower;

	Field	Type	Null	Key	Default	Extra
▶	Customer_name	varchar(20)	NO	PRI	NULL	
	loanno	int	NO	PRI	NULL	

INSERT INTO Branch
values('SBI_Chamrajpet','Bangalore',50000),
('SBI_ResidencyRoad','Bangalore',10000),
('SBI_ShivajiRoad','Bombay',20000),

('SBI_ParliamentRoad','Delhi',10000),
('SBI_Jantarmantar','Delhi',20000); select
* from Branch;

	Branch_name	Branch_city	Assets
▶	SBI_Chamrajpet	Bangalore	50000
	SBI_Jantarmantar	Delhi	20000
	SBI_ParliamentRoad	Delhi	10000
	SBI_ResidencyRoad	Bangalore	10000
	SBI_ShivajiRoad	Bombay	20000
*	NULL	NULL	NULL

INSERT INTO Bank_account
values(1,'SBI_Chamrajpet',2000),
(2,'SBI_ResidencyRoad',5000),
(3,'SBI_ShivajiRoad',6000),
(4,'SBI_ParliamentRoad',9000),
(5,'SBI_Jantarmantar',8000),
(6,'SBI_ShivajiRoad',4000),
(8,'SBI_ResidencyRoad',4000),
(9,'SBI_ParliamentRoad',3000),
(10,'SBI_ResidencyRoad',5000),
(11,'SBI_Jantarmantar',2000);
select * from
Bank_account;

accno	Branch_name	Balance
1	SBI_Chamrajpet	2000
2	SBI_ResidencyRoad	5000
3	SBI_ShivajiRoad	6000
4	SBI_ParliamentRoad	9000
5	SBI_Jantarmantar	8000
6	SBI_ShivajiRoad	4000
8	SBI_ResidencyRoad	4000
9	SBI_ParliamentRoad	3000
10	SBI_ResidencyRoad	5000
11	SBI_Jantarmantar	2000
NULL	NULL	NULL

```
INSERT INTO Bank_customer
values('Avinash','Bull_Temple_Road','Bangalore'),
        ('Dinesh','Bannergatta_Road','Bangalore'),
      ('Mohan','NationalCollege_Road','Bangalore'),
      ('Nikil','Akbar_Road','Delhi'),
('Ravi','Prithviraj_Road','Delhi');
select * from Bank_customer;
```

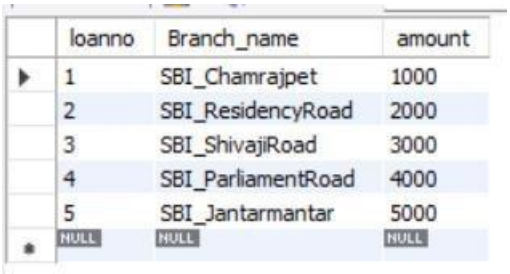
Customer_name	Customer_str	city
Avinash	Bull_Temple_Road	Bangalore
Dinesh	Bannergatta_Road	Bangalore
Mohan	NationalCollege_Road	Bangalore
Nikil	Akbar_Road	Delhi
Ravi	Prithviraj_Road	Delhi
NULL	NULL	NULL

```
insert into Depositor
values('Avinash',1),
        ('Dinesh',2),
      ('Nikil',4),
      ('Ravi',5),
      ('Avinash',8),
      ('Nikil',9),

('Dinesh',10),
('Nikil',11); select
* from Depositor;
```

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

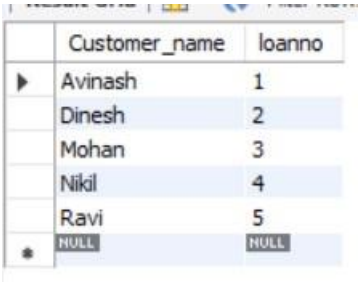
```
insert into Loan
values(1,'SBI_Chamrajpet',1000),
```

```
                (2,'SBI_ResidencyRoad',2000),
        (3,'SBI_ShivajiRoad',3000),

(4,'SBI_ParliamentRoad',4000),
(5,'SBI_Jantarmantar',5000); select
* from Loan;
```



loanno	Branch_name	amount
1	SBI_Chamrajpet	1000
2	SBI_ResidencyRoad	2000
3	SBI_ShivajiRoad	3000
4	SBI_ParliamentRoad	4000
5	SBI_Jantarmantar	5000
NULL	NULL	NULL

```
insert into Borrower
values('Avinash',1),
          ('Dinesh',2),
        ('Mohan',3),
        ('Nikil',4),
        ('Ravi',5);
Select * from
Borrower;
```



Customer_name	loanno
Avinash	1
Dinesh	2
Mohan	3
Nikil	4
Ravi	5
NULL	NULL

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/100000 as Assets_in_lakhs
from Branch;
```

Branch_name	Assets_in_lakhs
SBI_Chamrajpet	0.5
SBI_Jantarmantar	0.2
SBI_ParliamentRoad	0.1
SBI_ResidencyRoad	0.1
SBI_ShivajiRoad	0.2

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

```
select Customer_name from
Depositor D,Bank_account B
where D.accno=B.accno and Branch_name='SBI_ResidencyRoad' group
by Customer_name
having count(*)>=2;
```

	Customer_name
▶	Dinesh

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

```
create view Total_amount_of_loan
as select Branch_name,sum(amount)
from Loan group by Branch_name;
select *
from Total_amount_of_loan;
```

	Branch_name	sum(amount)
▶	SBI_Chamrajpet	1000
	SBI_Jantarmantar	5000
	SBI_ParliamentRoad	4000
	SBI_ResidencyRoad	2000
	SBI_ShivajiRoad	3000

4. More Queries on Bank Database:

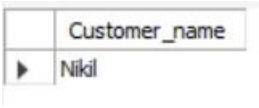
Inserting some more values

```
insert into Branch
values('SBI_MantriMarg','Delhi',200000);
insert into Bank_account
values(12,'SBI_Mantrimarg',2000);
insert into Depositor
values('Nikil',12);
```

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

```
select distinct
Customer_name from
Depositor where accno
in(select accno
```

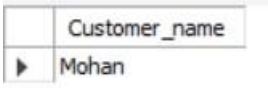
```
                                from Bank_account
            where Branch_name in(select Branch_name
        from Branch                                where Branch_city='Delhi'))
group by Customer_name
having count(Customer_name)>(select count(Branch_name)
                                                                from Branch

                    where Branch_city='Delhi');
```



	Customer_name
▶	Nikil

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

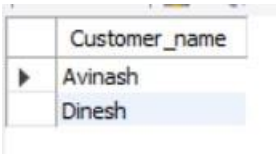
```
select Customer_name
from Borrower
where Customer_name not in(select Customer_name
                                from Depositor);
```



	Customer_name
▶	Mohan

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

```
select Customer_name
from Borrower b
where b.loanno=any(select l.loanno
                from loan l
                        where l.Branch_name=any(select Branch_name
                                                from Branch
                                where Branch_city='Bangalore')) and Customer_name in
(select Customer_name
                        from Depositor
                              where Customer_name =any(select Customer_name
                                                from Bank_customer
                                                        where city='Bangalore'));
```



	Customer_name
▶	Avinash
	Dinesh

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

```
select Branch_name from
Branch  where Assets
>(select Assets
```

```
                from Branch
            where Branch_city='Bangalore'
group by Branch_city);
```

Branch_name
SBI_MantriMarg
NULL

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

```
delete from Bank_account where
Branch_name in(select Branch_name
                    from Branch
                        where Branch_city='Bombay');
select * from Bank_account;
```

accno	Branch_name	Balance
1	SBI_Chamrajpet	2000
2	SBI_ResidencyRoad	5000
4	SBI_ParliamentRoad	9000
5	SBI_Jantarmantar	8000
8	SBI_ResidencyRoad	4000
9	SBI_ParliamentRoad	3000
10	SBI_ResidencyRoad	5000
11	SBI_Jantarmantar	2000
12	SBI_Mantrimarg	2000
NULL	NULL	NULL

Update the Balance of all accounts by 5%

```
update Bank_account set
Balance=Balance+Balance*0.05;
select * from Bank_account;
```

accno	Branch_name	Balance
1	SBI_Chamrajpet	2100
2	SBI_ResidencyRoad	5250
4	SBI_ParliamentRoad	9450
5	SBI_Jantarmantar	8400
8	SBI_ResidencyRoad	4200
9	SBI_ParliamentRoad	3150
10	SBI_ResidencyRoad	5250
11	SBI_Jantarmantar	2100
12	SBI_Mantrimarg	2100
NULL	NULL	NULL

5. Employee database

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

2. Enter greater than five tuples for each table.

3. Retrieve the employee numbers of all employees who work on project located in Bengaluru, Hyderabad, or Mysuru

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

5. 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.

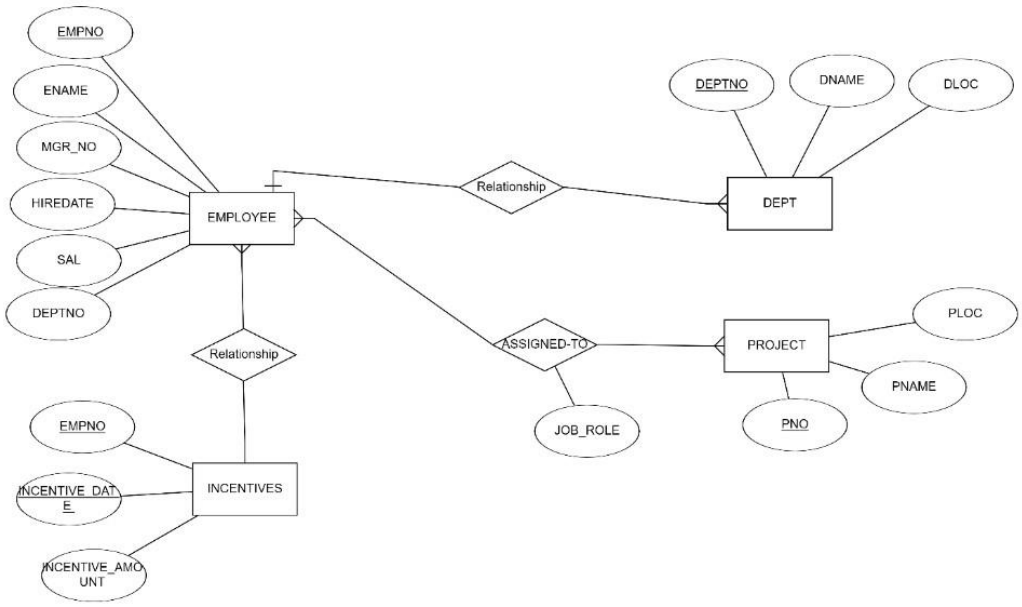
6. List the name of the managers with the maximum employees

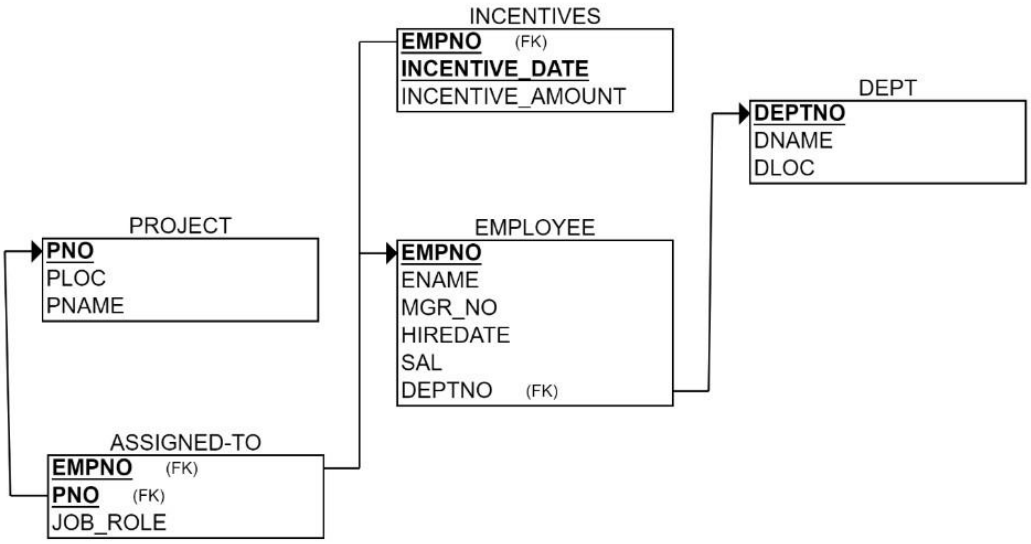
7. Display those managers name whose salary is more than average salary of his employee.

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

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

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






```
create database
EMPLOYEE083; use
EMPLOYEE083; CREATE
TABLE DEPT
(
   DEPTNO INT,
   DNAME VARCHAR(30),
DLOC VARCHAR(30),
   primary key(DEPTNO)
);
create table EMPLOYEE
(
   EMPNO INT,
   ENAME VARCHAR(30),
   MGRNO INT,
   HIREDATE DATE,
   SAL INT,    DEPTNO
INT,    primary
key(EMPNO),
   foreign key(DEPTNO) references DEPT(DEPTNO)
);
create table PROJECT
(
   PNO INT,
   PLOC VARCHAR(30),
PNAME VARCHAR(30),
primary key(PNO)
);
create table ASSIGNEDTO
```

```
(
   EMPNO INT,
   PNO INT,
   JOBROLE varchar(30),
primary key(EMPNO,PNO),
   foreign key(EMPNO) references EMPLOYEE(EMPNO),
foreign key(PNO) references PROJECT(PNO)
);
create table INCENTIVES
(
   EMPNO INT,
   INCENTIVE_DATE DATE,
INCENTIVE_AMT INT,
   primary key(EMPNO,INCENTIVE_DATE),     foreign
key(EMPNO) references EMPLOYEE(EMPNO)
);
```

desc DEPT;

Field	Type	Null	Key	Default	Extra
DEPTNO	int	NO	PRI	NULL	
DNAME	varchar(30)	YES		NULL	
DLOC	varchar(30)	YES		NULL	

desc EMPLOYEE;

Field	Type	Null	Key	Default	Extra
EMPNO	int	NO	PRI	NULL	
ENAME	varchar(30)	YES		NULL	
MGRNO	int	YES		NULL	
HIREDATE	date	YES		NULL	
SAL	int	YES		NULL	
DEPTNO	int	YES	MUL	NULL	

desc PROJECT;

Field	Type	Null	Key	Default	Extra
PNO	int	NO	PRI	NULL	
PLOC	varchar(30)	YES		NULL	
PNAME	varchar(30)	YES		NULL	

desc ASSIGNEDTO;

Field	Type	Null	Key	Default	Extra
EMPNO	int	NO	PRI	NULL	
PNO	int	NO	PRI	NULL	
JOBROLE	varchar(30)	YES		NULL	

desc INCENTIVES;

Field	Type	Null	Key	Default	Extra
EMPNO	int	NO	PRI	NULL	
INCENTIVE_DATE	date	NO	PRI	NULL	
INCENTIVE_AMT	int	YES		NULL	

insert into DEPT
values(1,"Sales","Bangalore"),(2,"Marketing","Mysore"),(3,"Resources","Hyderabad"),(4,"Producion","Bangalore"),(5,"Information","Mumbai"),(6,"Security","Delhi");
select * from DEPT;

DEPTNO	DNAME	DLOC
1	Sales	Bangalore
2	Marketing	Mysore
3	Resources	Hyderabad
4	Producion	Bangalore
5	Information	Mumbai
6	Security	Delhi
NULL	NULL	NULL

insert into EMPLOYEE
values(11,"Mohan",12,"2000-06-18",60000,1), (12,"Avinash",12,"2000-02-17",80000,1),
(13,"Nikil",12,"2000-05-05",80000,3),
(14,"Ravi",12,"2000-10-20",80000,3),
(15,"Ram",12,"2000-06-06",70000,2),
(16,"Siri",12,"2000-05-20",60000,2),
(17,"Seeta",12,"2000-06-16",70000,4),
(18,"Mohana",12,"2000-10-10",50000,4),
(19,"sam",12,"2000-12-11",50000,4),
(20,"Keerthi",12,"2000-01-01",90000,5);
Select * from EMPLOYEE;

EMPNO	ENAME	MGRNO	HIREDATE	SAL	DEPTNO
11	Mohan	12	2000-06-18	60000	1
12	Avinash	12	2000-02-17	80000	1
13	Nikil	12	2000-05-05	80000	3
14	Ravi	12	2000-10-20	80000	3
15	Ram	12	2000-06-06	70000	2
16	Siri	12	2000-05-20	60000	2
17	Seeta	12	2000-06-16	70000	4
18	Mohana	12	2000-10-10	50000	4
19	sam	12	2000-12-11	50000	4
20	Keerthi	12	2000-01-01	90000	5
NULL	NULL	NULL	NULL	NULL	NULL

insert into PROJECT
VALUES(1,"Bangalore","Sales report"),
(2,"Bangalore","Production report"),
(3,"Mumbai","Survey report"),
(4,"Mysore","Infographics"),

(5,"Delhi","Data security"),
(6,"Mysore","Advertisements");
Select * from PROJECT;

PNO	PLOC	PNAME
1	Bangalore	Sales report
2	Bangalore	Production report
3	Mumbai	Survey report
4	Mysore	Infographics
5	Delhi	Data security
6	Mysore	Advertisements
NULL	NULL	NULL

insert into ASSIGNEDTO
values(11,1,"Associate"),
(15,4,"Assistant"),
(16,4,"Assistant"),
(18,2,"Assistant"),
(19,2,"Associate"),
(20,5,"Manager");
Select * from ASSIGNEDTO;

EMPNO	PNO	JOBROLE
11	1	Associate
15	4	Assistant
16	4	Assistant
18	2	Assistant
19	2	Associate
20	5	Manager
NULL	NULL	NULL

insert into INCENTIVES values(20,"2005-06-09",5000),
(15,"2005-05-10",3000),
(12,"2005-01-01",4000),
(17,"2005-02-01",4000),
(18,"2005-03-02",1000);
Select * from INCENTIVES;

EMPNO	INCENTIVE_DATE	INCENTIVE_AMT
12	2005-01-01	4000
15	2005-05-10	3000
17	2005-02-01	4000
18	2005-03-02	1000
20	2005-06-09	5000
NULL	NULL	NULL

Retrieve the employee numbers of all employees who work on project located in Bengaluru, Hyderabad, or Mysuru
SELECT EMPNO
FROM EMPLOYEE
WHERE EMPNO IN(SELECT EMPNO

```
        FROM ASSIGNEDTO
    WHERE PNO IN(SELECT PNO
            FROM PROJECT
        WHERE PLOC IN("Bangalore","Hyderabad","Mysore")));
```

EMPNO
11
15
16
18
19
NULL

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

```
SELECT EMPNO
FROM EMPLOYEE
WHERE EMPNO NOT IN (SELECT EMPNO FROM INCENTIVES);
```

EMPNO
11
16
13
14
19
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.ENAME,E.EMPNO,D.DNAME,A.JOBROLE,D.DLOC,P.PLOC
FROM EMPLOYEE E,DEPT D,ASSIGNEDTO A,PROJECT P
WHERE E.EMPNO=A.EMPNO AND D.DEPTNO=E.DEPTNO AND A.PNO=P.PNO AND
D.DLOC=P.PLOC;
```

ENAME	EMPNO	DNAME	JOBROLE	DLOC	PLOC
Mohan	11	Sales	Associate	Bangalore	Bangalore
Ram	15	Marketing	Assistant	Mysore	Mysore
Siri	16	Marketing	Assistant	Mysore	Mysore
Mohana	18	Producion	Assistant	Bangalore	Bangalore
sam	19	Producion	Associate	Bangalore	Bangalore

6. More Queries on Employee Database :

List the name of the managers with the maximum employees.

```
SELECT m.ENAME, count(*)
FROM EMPLOYEE e,EMPLOYEE m
WHERE e.MGRNO = m.EMPNO
GROUP BY m.ENAME
HAVING count(*) =(SELECT MAX(mycount)
from (SELECT COUNT(*) mycount
FROM EMPLOYEE
GROUP BY MGRNO) a);
```

	ENAME	count(*)
▶	Avinash	10

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

```
SELECT *
FROM EMPLOYEE m
WHERE m.EMPNO IN
(SELECT MGRNO
FROM EMPLOYEE)
AND m.SAL >(SELECT avg(e.SAL)
        FROM EMPLOYEE e
        WHERE e.MGRNO = m.EMPNO );
```

	EMPNO	ENAME	MGRNO	HIREDATE	SAL	DEPTNO
▶	12	Avinash	12	2000-02-17	80000	1
*	NULL	NULL	NULL	NULL	NULL	NULL

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

```
select distinct m.MGRNO, M.ENAME
FROM EMPLOYEE E,EMPLOYEE M
WHERE E.MGRNO=M.MGRNO AND E.DEPTNO=M.DEPTNO AND E.EMPNO IN (
select distinct

M.MGRNO
                                                    FROM EMPLOYEE
E,EMPLOYEE M
                                        WHERE E.MGRNO=M.MGRNO AND E.DEPTNO
=M.DEPTNO);
```

	MGRNO	ENAME
▶	12	Mohan
	12	Avinash

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

```
SELECT *
FROM EMPLOYEE E,INCENTIVES I
WHERE E.EMPNO=I.EMPNO AND 1=(SELECT COUNT(*)
                             FROM INCENTIVES J
                                  WHERE I.INCENTIVE_AMT<J.INCENTIVE_AMT
AND INCENTIVE_DATE LIKE '2005-01-%');
```

EMPNO	ENAME	MGRNO	HIREDATE	SAL	DEPTNO	EMPNO	INCENTIVE_DATE	INCENTIVE_AMT
15	Ram	12	2000-06-06	70000	2	15	2005-05-10	3000
18	Mohana	12	2000-10-10	50000	4	18	2005-03-02	1000

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

```
SELECT *
FROM EMP E
WHERE E.DEPTNO = (SELECT E1.DEPTNO
                  FROM EMP E1
                  WHERE E1.EMPNO=E.MGR_NO);
```

EMPNO	ENAME	MGRNO	HIREDATE	SAL	DEPTNO
11	Mohan	12	2000-06-18	60000	1
12	Avinash	12	2000-02-17	80000	1
NULL	NULL	NULL	NULL	NULL	NULL

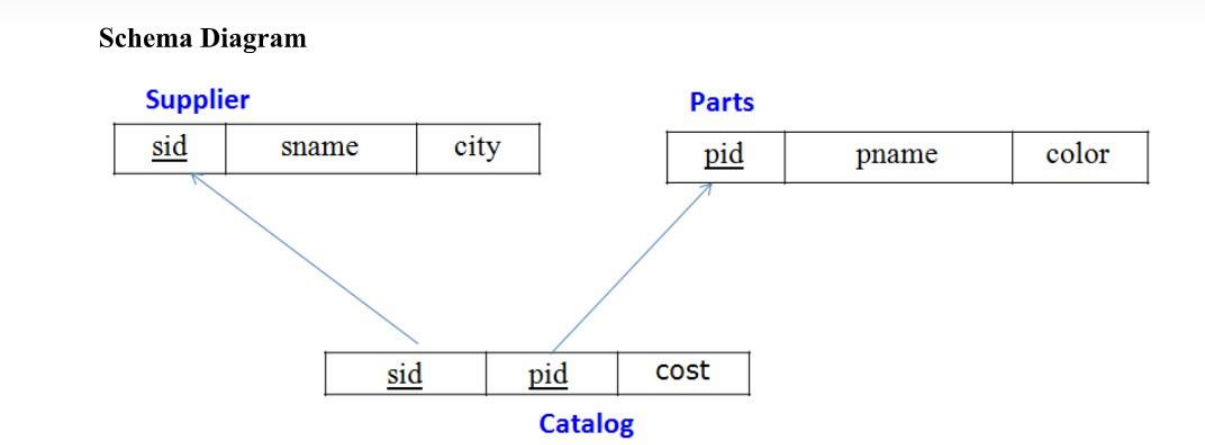
Write a SQL query to find those employees whose net pay are higher than or equal to the salary of any other employee in the company.

```
SELECT distinct e.ENAME
FROM EMPLOYEE e,INCENTIVES i
WHERE (SELECT max(SAL+INCENTIVE_AMT)
FROM EMPLOYEE,INCENTIVES) >= ANY
(SELECT SAL
FROM EMPLOYEE e1 where
e.DEPTNO=e1.DEPTNO);
```

ENAME
Mohan
Avinash
Nikil
Ravi
Ram
Siri
Seeta
Mohana
sam
Keerthi

7. SUPPLIER database:

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.
Write the following queries in SQL:
i) Find the pnames of parts for which there is some supplier.
ii) Find the snames of suppliers who supply every part. iii) Find the snames of suppliers who supply every red part. iv) Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.
v) 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). vi) For each part, find the sname of the supplier who charges the most for that part.




```
create database SUPPLIERS_1BM21CS083;
use SUPPLIERS_1BM21CS083;
create table SUPPLIERS
(
sid int primary key,
sname varchar(20),
city varchar(20)
);
create table PARTS
(
pid int primary key,
pname varchar(20),
color varchar(10)
);
create table CATALOG(
sid int,
```

```
pid int,
foreign key(sid) references SUPPLIERS(sid),
foreign key(pid) references PARTS(pid),
cost float(6),
primary key(sid, pid)
 );
```

desc SUPPLIERS;

	Field	Type	Null	Key	Default	Extra
▶	sid	int	NO	PRI	NULL	
	sname	varchar(20)	YES		NULL	
	city	varchar(20)	YES		NULL	

Desc PARTS;

	Field	Type	Null	Key	Default	Extra
▶	pid	int	NO	PRI	NULL	
	pname	varchar(20)	YES		NULL	
	color	varchar(10)	YES		NULL	

desc catalog;

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

insert into suppliers values(10001, ' Acme Widget','Bangalore'); insert into suppliers values(10002, ' Johns','Kolkata'); insert into suppliers values(10003, ' Vimal','Mumbai'); insert into suppliers values(10004, ' Reliance','Delhi');
insert into suppliers values(10005, ' Mahindra','Mumbai');
SELECT * FROM SUPPLIERS;

	sid	sname	city
▶	10001	Acme Widget	Bangalore
	10002	Johns	Kolkata
	10003	Vimal	Mumbai
	10004	Reliance	Delhi
	10005	Mahindra	Mumbai
*	NULL	NULL	NULL

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(20005,
'Charger','Black'); select * from PARTS;

	pid	pname	color
▶	20001	Book	Red
	20002	Pen	Red
	20003	Pencil	Green
	20004	Mobile	Green
	20005	Charger	Black
*	NULL	NULL	NULL

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;

	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
*	NULL	NULL	NULL

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

SELECT distinct pname
from PARTS
where pid in (select pid
from CATALOG)
order by pname;

	pname
▶	Book
	Charger
	Mobile
	Pen
	Pencil

Find the snames of suppliers who supply every part.

```
select sname from
SUPPLIERS s
WHERE ((select count(p.pid)
                from PARTS p)=(select count(c.pid)
                from CATALOG c
                   where c.sid=s.sid));
```

	sname
▶	Acme Widget

Find the snames of suppliers who supply every red part.

```
select sname
from SUPPLIERS
where sid in(select sid
                from CATALOG
                WHERE pid in (select pid
                                from PARTS
                                where color="Red"));
```

	sname
▶	Acme Widget
	Johns

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

```
select p.pname
from  SUPPLIERS s,PARTS p,CATALOG c
where c.pid=p.pid and s.sid=c.sid and s.sname="Acme Widget" and not exists(select *
        from CATALOG c1,SUPPLIERS s1
            where p.pid=c1.pid and c1.sid=s1.sid and s1.sname<>"Acme Widget");
```

	pname
▶	Mobile
	Charger

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 distinct c.sid
from CATALOG C
WHERE C.cost>( select avg(cost)
                from CATALOG c1
            where c1.pid=c.pid);
```

	sid
▶	10002
	10004

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

```
select distinct s.sname
from SUPPLIERS s,PARTS p,CATALOG c
where c.pid=p.pid and s.sid=c.sid and c.cost=(select max(c1.cost)
        from CATALOG c1                    where c1.pid=p.pid);
```

	sname
▶	Acme Widget
	Johns
	Reliance

8. Flight Database:

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

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

CERTIFIED(eid: integer, aid: integer)

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

Note that the Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly.

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

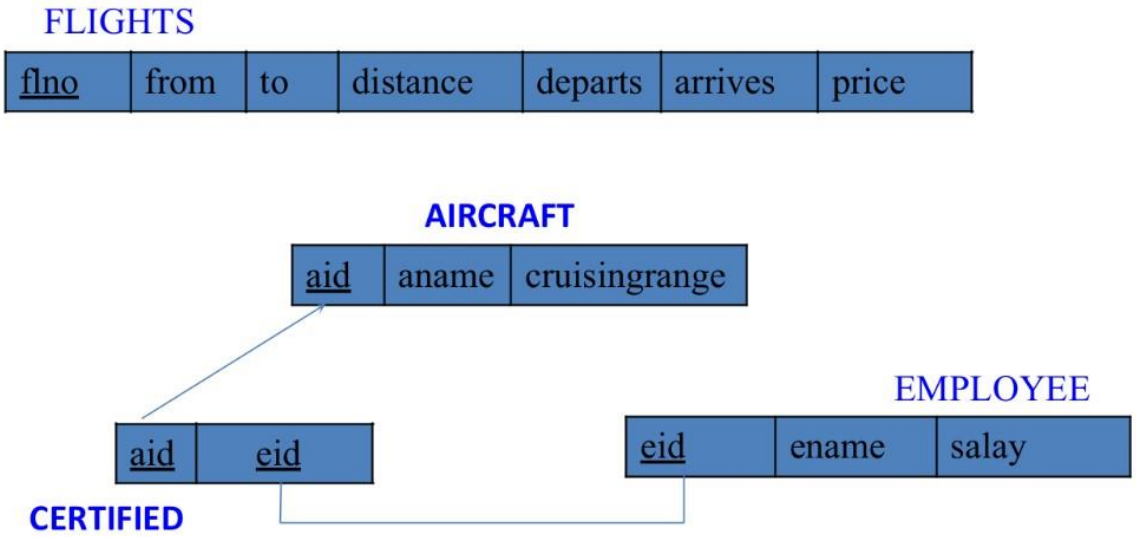
ii. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.

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

iv. 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.

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

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




```
create database AIRLINE_FLIGHT; USE
AIRLINE_FLIGHT;
create table AIRCRAFT
(
   aid int,
   aname varchar(30),
cruisingrange int,
   primary key(aid)
);
create table EMPLOYEE
(
   eid int,
   ename varchar(30),
salary int,
   primary key(eid)
);
create table CERTIFIED
(    aid int,    eid int,    primary key(aid,eid),
foreign key (aid) references AIRCRAFT(aid),
   foreign key (eid) references EMPLOYEE(eid)
);
create table FLIGHTS
(
   flno int,    lfrom
varchar(30),    lto
varchar(30),
distance int,
```

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

desc AIRCRAFT;

Field	Type	Null	Key	Default	Extra
aid	int	NO	PRI	NULL	
aname	varchar(30)	YES		NULL	
cruisingrange	int	YES		NULL	

DESC EMPLOYEE;

Field	Type	Null	Key	Default	Extra
eid	int	NO	PRI	NULL	
ename	varchar(30)	YES		NULL	
salary	int	YES		NULL	

DESC CERTIFIED;

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

DESC FLIGHTS;

Field	Type	Null	Key	Default	Extra
flno	int	NO	PRI	NULL	
lfrom	varchar(30)	YES		NULL	
lto	varchar(30)	YES		NULL	
distance	int	YES		NULL	
depart	time	YES		NULL	
arrives	time	YES		NULL	
price	int	YES		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;

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

insert into EMPLOYEE

VALUES(101,'Avinash',50000),(102,'Lokesh',60000),(103,'Rakesh',70000),(104,'Santhosh',8
2000),(105,'Tilak',5000);
SELECT * FROM EMPLOYEE;

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

insert into CERTIFIED
values(2,101),(4,101),(5,101),(6,101),(1,102),(3,102),(5,102),(2,103),(3,103),(5,103),(6,103),
(6,104),(1,104),(3,104),(3,105);
SELECT * FROM CERTIFIED;

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

insert into FLIGHTS
VALUES(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',10000,'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);
SELECT * FROM FLIGHTS;

flno	lfrom	lto	distance	depart	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	10000	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

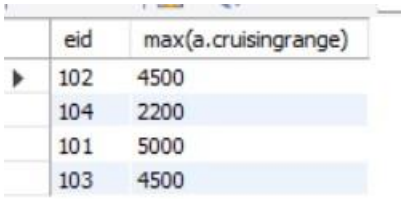
Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.

```
select aname
from AIRCRAFT
where aid in(select aid
                from CERTIFIED
                WHERE eid in(select eid
        from EMPLOYEE
                    WHERE salary>80000));
```

aname
Airbus
Jetairways
Airbus

For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.

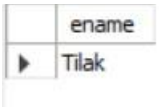
```
select c.eid,max(a.cruisingrange)
from AIRCRAFT a,CERTIFIED
c where a.aid=c.aid group by c.eid
having count(c.aid)>=3;
```



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

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

```
select ename
from EMPLOYEE
where salary<(select min(price)
                from FLIGHTS
                WHERE lfrom='Bengaluru' and lto='Frankfurt');
```



ename
Tilak

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.aid,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 a.aname,a.aid;
```

	aid	aname	avg(e.salary)
▶	1	Airbus	71000.0000
	4	Indigo	50000.0000
	5	Boeing	60000.0000
	6	Airbus	67333.3333

Find the names of pilots certified for some Boeing aircraft.

```
select ename
from EMPLOYEE
WHERE eid in (select eid
                from CERTIFIED
                WHERE aid in (select aid
                                from AIRCRAFT
                                WHERE aname='Boeing'));
```

	ename
▶	Avinash
	Lokesh
	Rakesh

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

```
select aid
from AIRCRAFT
WHERE EXISTS(select *
                from FLIGHTS
                where lfrom='Bengaluru' and lto='New Delhi');
```

	aid
▶	1
	2
	3
	4
	5
	6
*	NULL

9. NoSQL:

Perform the following DB operations using MongoDB.

1. Create a database “Student” with the following attributesRollno, Age, ContactNo, EmailId.

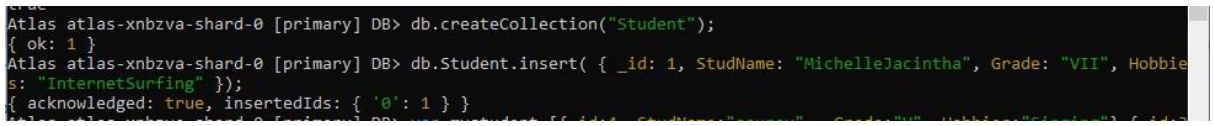
2. Insert appropriate values use DB;

Confirm the existence of your database db;

To create a collection by the name “Student”.

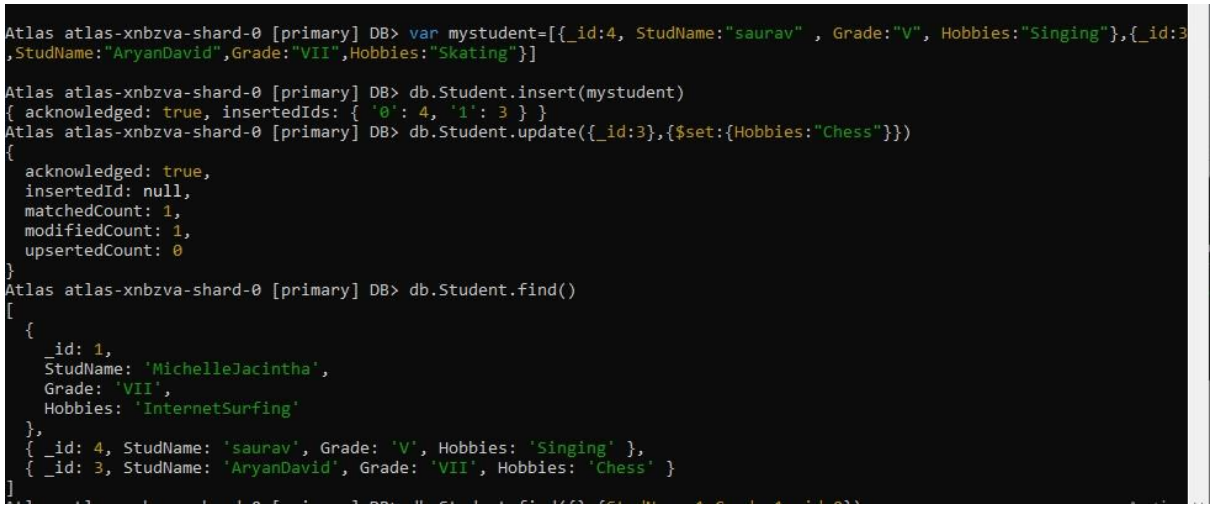
db.createCollection(“Student”); db.Student.insert(

{_id:1,
StudName:"MichelleJacintha",
Grade:"VII",
Hobbies:"InternetSurfing"});



var mystudent =[{_id:4, StudName:"saurav", Grade:"V", Hobbies:"Dance"},{_id:5, StudName:"kumar", Grade:"VI", Hobbies:"Singing"}]

db.Student.insert(mystudent) db.Student.find()



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

```
]
Atlas atlas-xnbzva-shard-0 [primary] DB> db.Student.find().limit(2)
[
  {
    _id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  { _id: 3, StudName: 'AryanDavid', Grade: 'VII', Hobbies: 'Chess' }
]
Atlas atlas-xnbzva-shard-0 [primary] DB>
```

db.Students.find({},{StudName:1,Grade:1,_id:0});

To find those documents where the Grade is set to 'VII'.

db.Students.find({Grade:{$eq:'VII'}})

To find documents from the Students collection where the StudName begins with "M".

db.Students.find({StudName:/^M/});

```
Atlas atlas-xnbzva-shard-0 [primary] DB> db.Student.find({},{StudName:1,Grade:1,_id:0});
[
  { StudName: 'MichelleJacintha', Grade: 'VII' },
  { StudName: 'saurav', Grade: 'V' },
  { StudName: 'AryanDavid', Grade: 'VII' }
]
Atlas atlas-xnbzva-shard-0 [primary] DB> db.Student.find({Grade:{$eq:'VII'}})
[
  {
    _id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  { _id: 3, StudName: 'AryanDavid', Grade: 'VII', Hobbies: 'Chess' }
]
Atlas atlas-xnbzva-shard-0 [primary] DB> db.Student.find({StudName:/^M/});
[
  {
    _id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  }
]
```

To sort the documents from the Students collection in the descending order of StudName.

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

db.Student.remove({"StudName":"saurav"})

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

```
Atlas atlas-xnbzva-shard-0 [primary] DB> db.Student.find().sort({StudName:-1})
[
  { _id: 4, StudName: 'saurav', Grade: 'V', Hobbies: 'Singing' },
  {
    _id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  { _id: 3, StudName: 'AryanDavid', Grade: 'VII', Hobbies: 'Chess' }
]
Atlas atlas-xnbzva-shard-0 [primary] DB> db.Student.remove({"StudName":"saurav"})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-xnbzva-shard-0 [primary] DB> db.Student.find()
[
  {
    _id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  { _id: 3, StudName: 'AryanDavid', Grade: 'VII', Hobbies: 'Chess' }
]
```

To find documents from the Students collection where the StudName has an “e” in any position.

db.Students.find({StudName:/e/});

To sort the documents from the Students collection in the ascending order of StudName.

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

```
Atlas atlas-xnbzva-shard-0 [primary] DB> db.Student.find({StudName:/e/});
[
  {
    _id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  }
]
Atlas atlas-xnbzva-shard-0 [primary] DB> db.Student.count();
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
3
Atlas atlas-xnbzva-shard-0 [primary] DB> db.Student.find().sort({StudName:1})
[
  { _id: 3, StudName: 'AryanDavid', Grade: 'VII', Hobbies: 'Chess' },
  {
    _id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  { _id: 4, StudName: 'saurav', Grade: 'V', Hobbies: 'Singing' }
]
```