VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

Database Management Systems (22CS3PCDBM)

Submitted by

SANJANA SHETTY (1BM22CS238)

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
December-2023 to Feb-2024

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 (23CS3PCDBM)” carried out by **SANJANA SHETTY (1BM22CS238),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a Database Management Systems (23CS3PCDBM) work prescribed for the said degree.

Name of Lab-incharge: Vikranth BM
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

Index

Insurance Database

Question

(Week 1)

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

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

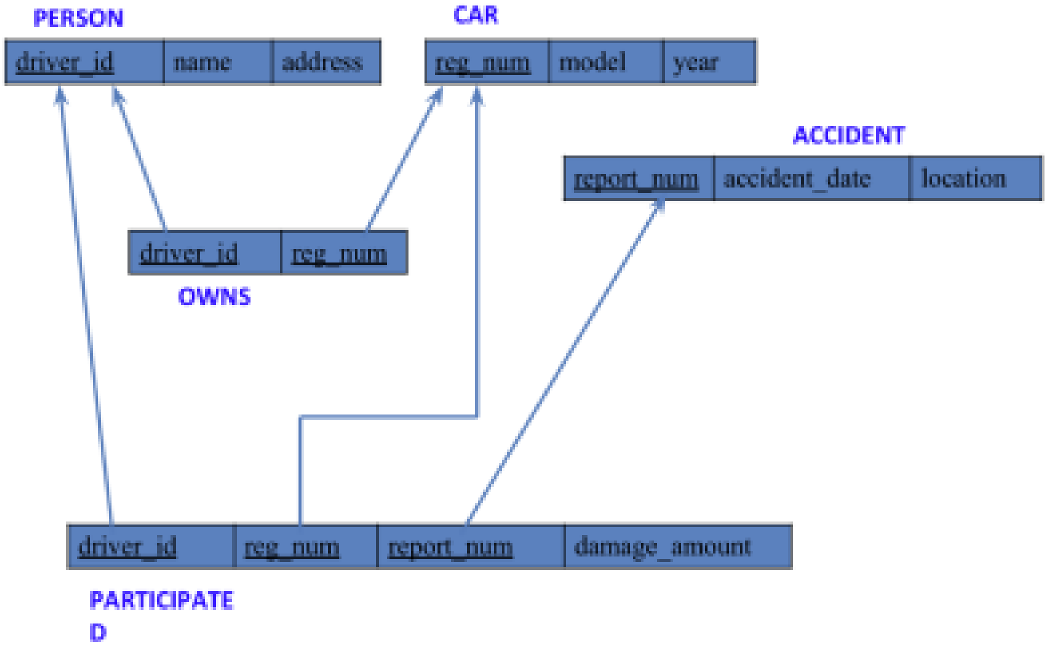
- ACCIDENT (report_num: int, accident_date: date, location: String)

- OWNS (driver_id: String, reg_num: String)

- PARTICIPATED (driver_id: String,reg_num: String, report_num: int, damage_amount: int)

- Create the above tables by properly specifying the primary keys and the foreign keys. - Enter at least five tuples for each relation

Schema Diagram



Create database

Create database insurance;
use insurance;

Create Tables

PERSON TABLE

```
create table person(
        driver_id varchar(5) PRIMARY KEY,
   driver_name varchar(15),
   address varchar(30)
);
```

Car Table

```
create table car(
        reg_num varchar(20) primary key,
   model varchar(10),
   year int(4)
);
```

Owns Table

```
create table owns(
          driver_id varchar(5),
   reg_num varchar(20),
   primary key(driver_id,reg_num),
   foreign key (driver_id) references person(driver_id),
   foreign key (reg_num) references car(reg_num)
);
```

Accident table

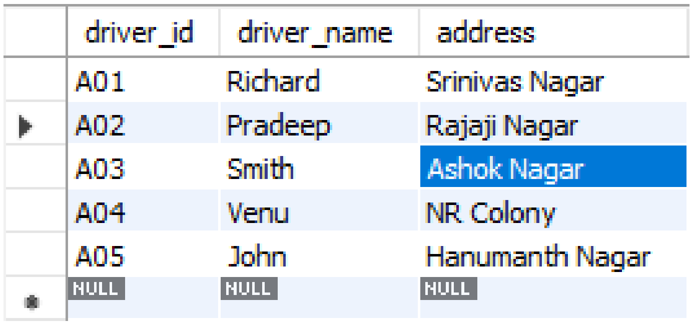
```
create table accident(
          report_num int primary key,
   accident_date date,
   location varchar(20)
);
```

Participate Table

```
create table participate(
          driver_id varchar(5),
          reg_num varchar(20),
          report_num int,
          damage_amt int,
          primary key(driver_id,reg_num,report_num),
          foreign key (driver_id) references person(driver_id),
           foreign key (reg_num) references car(reg_num),
          foreign key (report_num) references accident(report_num)
);
```

Inserting Values to the table

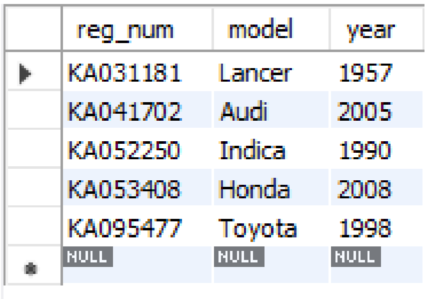
```
insert into person values('A01','Richard','Srinivas Nagar');
insert into person values('A02','Pradeep','Rajaji Nagar');
insert into person values('A03','Smith','Ashok Nagar');
insert into person values('A04','Venu','NR Colony');
insert into person values('A05','John','Hanumanth Nagar');

select*from person;
```



driver_id	driver_name	address
A01	Richard	Srinivas Nagar
A02	Pradeep	Rajaji Nagar
A03	Smith	Ashok Nagar
A04	Venu	NR Colony
A05	John	Hanumanth Nagar
NULL	NULL	NULL

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

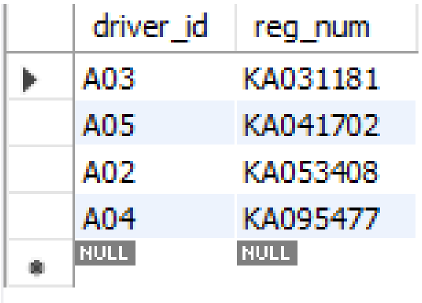
select *from car;



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

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

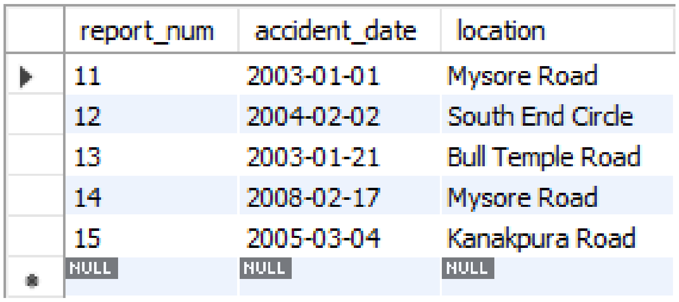
select *from owns;



driver_id	reg_num
A03	KA031181
A05	KA041702
A02	KA053408
A04	KA095477
NULL	NULL

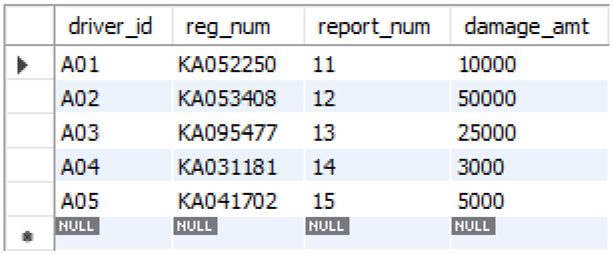
insert into accident values(11,'2003-01-01','Mysore Road');
insert into accident values(12,'2004-02-02','South End Circle');
insert into accident values(13,'2003-01-21','Bull Temple Road');
insert into accident values(14,'2008-02-17','Mysore Road');
insert into accident values(15,'2005-03-04','Kanakpura Road');

select * from accident;



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

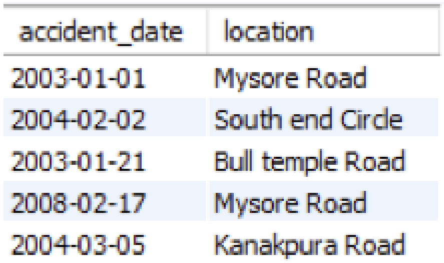


driver_id	reg_num	report_num	damage_amt
A01	KA052250	11	10000
A02	KA053408	12	50000
A03	KA095477	13	25000
A04	KA031181	14	3000
A05	KA041702	15	5000
NULL	NULL	NULL	NULL

Queries

1. **Display accident date and location**

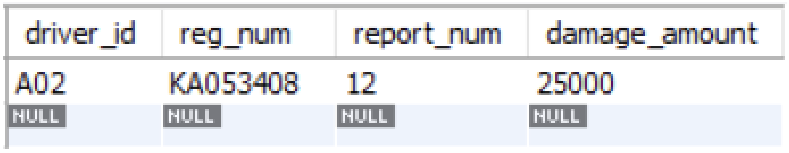
   select accident_date, location from accident;



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
2004-03-05	Kanakpura Road

2. **Update the damage amount to 25000 for the car with a specific reg-num (example 'KA053408') for which the accident report number was 12.**

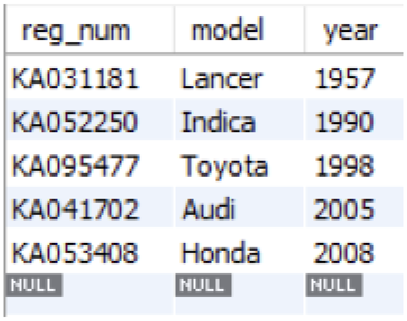
```
 update participated
set damage_amount=25000
where reg_num='KA053408' and report_num=12;
select * from participated where reg_num='KA053408' and report_num=12;
```



driver_id	reg_num	report_num	damage_amount
A02	KA053408	12	25000
NULL	NULL	NULL	NULL

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

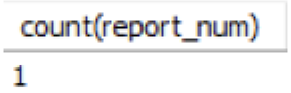
   select * from car order by year asc;



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

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

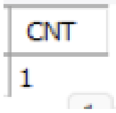
```
select count(report_num)
from car c, participated p
where c.reg_num=p.reg_num and c.model='Lancer';
```



count(report_num)
1

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

```
select count(distinct driver_id) CNT
from participated a, accident b
where a.report_num=b.report_num and b.accident_date like '__08%';
```



CNT
1

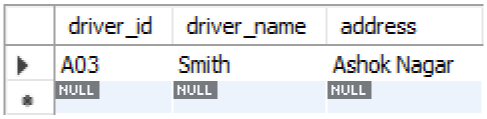
MORE QUERIES ON INSURANCE DATABASE

(Week 2)

Queries

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

- select * from person where address = 'Ashok Nagar' order by driver_name;



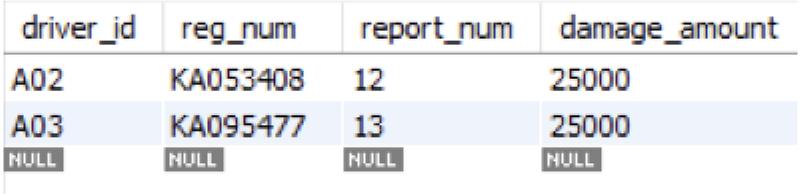
driver_id	driver_name	address
A03	Smith	Ashok Nagar
NULL	NULL	NULL

2. FIND THE AVERAGE DAMAGE AMOUNT

- select avg(damage_amount) as average;

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

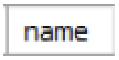
- delete from participated where damage_amount < (select p.damage_amount from(select AVG(damage_amount) as damage_amount FROM participated)p);
  select * from participated;



driver_id	reg_num	report_num	damage_amount
A02	KA053408	12	25000
A03	KA095477	13	25000
NULL	NULL	NULL	NULL

4. List the name of drivers whose damage amount is greater than average damage amount

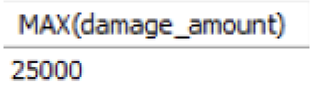
- select name from person p, participated part where p.driver_id=part.driver_id and damage_amount>(select AVG(damage_amount) FROM participated);



```
name
```

Find maximum damage amount.

- select MAX(damage_amount) from participated;



```
MAX(damage_amount)
25000
```

BANK DATABASE

Question

(Week 3)

- 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)

- Create the above tables by properly specifying the primary keys and the foreign

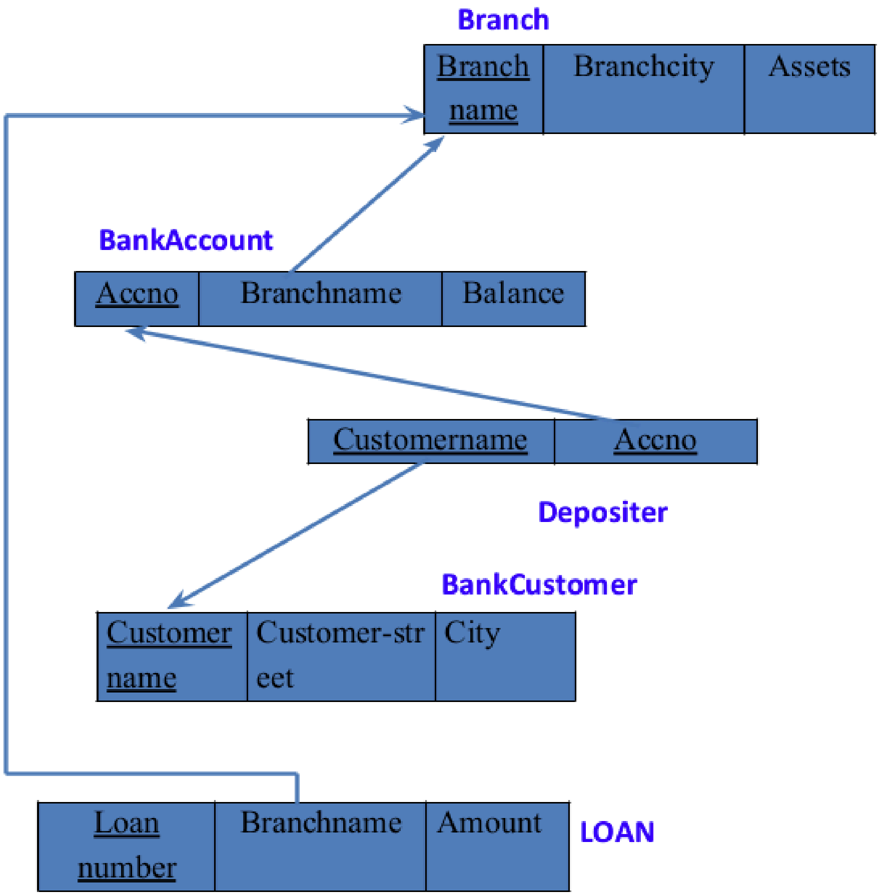
keys. - Enter at least five tuples for each relation.

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

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

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

Scheme Diagram



Create database

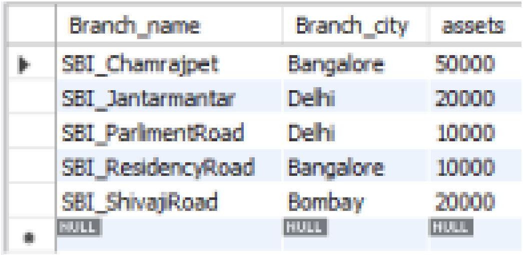
```
create database bank;

use bank;
```

Create table

```
create table branch(

            branch_name varchar(30) primary key,

    branch_city varchar(20),

    assets real

);


create table bankacc(
         accno int primary key,
   branch_name varchar(30),
   balance int,
   foreign key(branch_name) references branch(branch_name)
);

create table bankcustomer(
         customer_name varchar(20) primary key,
   customer_street varchar(30),
   city varchar(20)
);

create table depositor(
         customer_name varchar(20),
   accno int,
   primary key(customer_name,accno),
   foreign key(customer_name) references bankcustomer(customer_name),
   foreign key(accno) references bankacc(accno)
);

create table loan(
         loan_no int primary key,
   branch_name varchar(30),
   amt int,
   foreign key(branch_name) references bankacc(branch_name) );
```

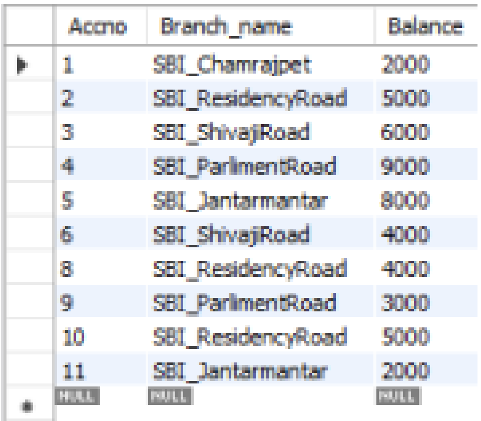
Inserting Values to the table

```
insert into branch values("SBI_Chamrajpet","Bangalore",50000);
insert into branch values("SBI_ResidencyRoad","Bangalore",10000);

insert into branch values("SBI_ShivajiRoad","Bombay",20000);

insert into branch values("SBI_ParlimentRoad","Delhi",10000);

insert into branch values("SBI_Jantarmantar","Delhi",20000);

select * from branch;
```



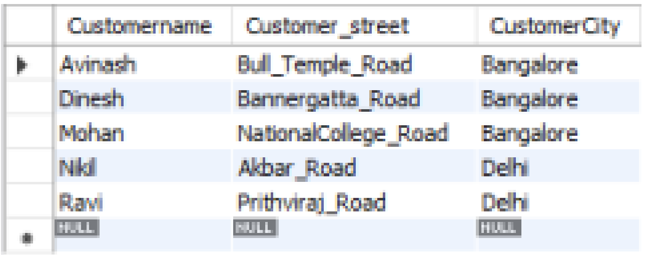
Branch_name	Branch_city	assets
SBI_Chamrajpet	Bangalore	50000
SBI_Jantarmantar	Delhi	20000
SBI_ParlimentRoad	Delhi	10000
SBI_ResidencyRoad	Bangalore	10000
SBI_ShivajiRoad	Bombay	20000
NULL	NULL	NULL

```
insert into bankacc values(1,"SBI_Chamrajpet",2000);

insert into bankacc values(2,"SBI_ResidencyRoad",5000);

insert into bankacc values(3,"SBI_ShivajiRoad",6000);

insert into bankacc values(4,"SBI_ParlimentRoad",9000);

insert into bankacc values(5,"SBI_Jantarmantar",8000);

insert into bankacc values(6,"SBI_ShivajiRoad",4000);

insert into bankacc values(8,"SBI_ResidencyRoad",4000);

insert into bankacc values(9,"SBI_ParlimentRoad",3000);

insert into bankacc values(10,"SBI_ResidencyRoad",5000);

insert into bankacc values(11,"SBI_Jantarmantar",2000);

select * from bankacc;
```



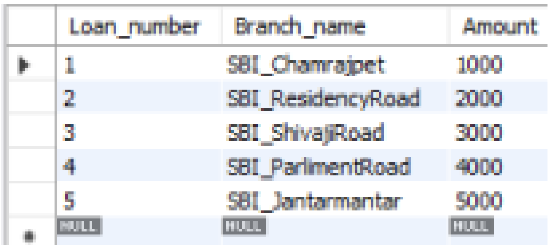
Accno	Branch_name	Balance
1	SBI_Chamrajpet	2000
2	SBI_ResidencyRoad	5000
3	SBI_ShivajiRoad	6000
4	SBI_ParlimentRoad	9000
5	SBI_Jantarmantar	8000
6	SBI_ShivajiRoad	4000
8	SBI_ResidencyRoad	4000
9	SBI_ParlimentRoad	3000
10	SBI_ResidencyRoad	5000
11	SBI_Jantarmantar	2000
NULL	NULL	NULL

```
insert into bankcustomer values("Avinash","Bull_Temple_Road","Bangalore");
insert into bankcustomer values("Dinesh","Bannergatta_Road","Bangalore");
insert into bankcustomer values("Mohan","NationalCollege_Road","Bangalore");
insert into bankcustomer values("Nikil","Akbar_Road","Delhi");
insert into bankcustomer values("Ravi","Prithviraj_Road","Delhi");
select * from bankcustomer;
```



Customername	Customer_street	CustomerCity
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 loan values(1,"SBI_Chamrajpet",1000);
insert into loan values(2,"SBI_ResidencyRoad",2000);
insert into loan values(3,"SBI_ShivajiRoad",3000);
insert into loan values(4,"SBI_ParlimentRoad",4000);
insert into loan values(5,"SBI_Jantarmantar",5000);
select * from loan;
```



Loan_number	Branch_name	Amount
1	SBI_Chamrajpet	1000
2	SBI_ResidencyRoad	2000
3	SBI_ShivajiRoad	3000
4	SBI_ParlimentRoad	4000
5	SBI_Jantarmantar	5000
NULL	NULL	NULL

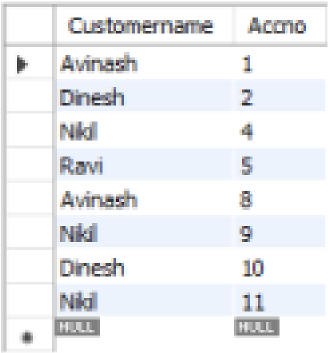
```
insert into depositor values("Avinash",1);
insert into depositor values("Dinesh",2);
insert into depositor values("Nikil",4);
insert into depositor values("Ravi",5);
insert into depositor values("Avinash",8);
```

insert into depositor values("Nikil",9);

insert into depositor values("Dinesh",10);

insert into depositor values("Nikil",11);

select* from depositor;

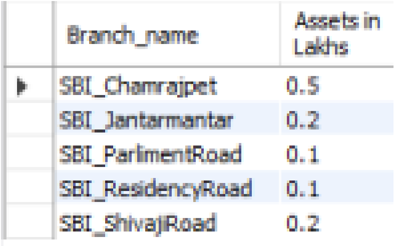


Customername	Accno
Avinash	1
Dinesh	2
Nikil	4
Ravi	5
Avinash	8
Nikil	9
Dinesh	10
Nikil	11
NULL	NULL

Queries

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

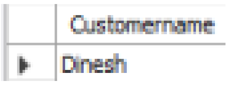
  select branch_name, assets/100000 assets_in_lakhs from branch;



Branch_name	Assets in Lakhs
SBI_Chamrajpet	0.5
SBI_Jantarmantar	0.2
SBI_ParlimentRoad	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).**

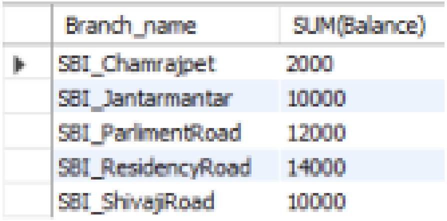
```
 select d.customer_name from depositor d, bankacc b where
b.branch_name='SBI_ResidencyRoad' and d.accno=b.accno group by
d.customername having count(d.accno)>=2;
```



	Customername
▶	Dinesh

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

```
create view branch_loans
as select branch_name, SUM(amt)
from bankacc
group by branch_name;
select * from branch_loans;
```



	Branch_name	SUM(Balance)
▶	SBI_Chamrajpet	2000
	SBI_Jantarmantar	10000
	SBI_ParlimentRoad	12000
	SBI_ResidencyRoad	14000
	SBI_ShivajiRoad	10000

More Queries on Bank Database

(Week 4)

- 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)

- Find all the customers who have an account at all the branches

- located in a specific city (Ex. Delhi).

- Find all customers who have a loan at the bank but do not have an account. - Find all customers who have both an account and a loan at the Bangalore branch

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

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

- Update the Balance of all accounts by 5%

Create database

```
create database bank;

use bank;
```

Create table

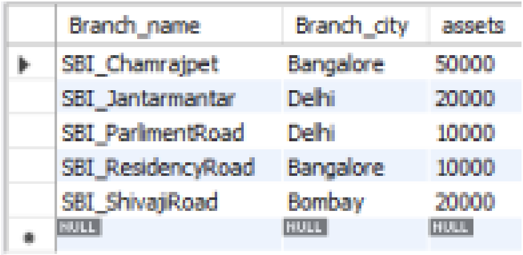
```
create table branch(

          branch_name varchar(30) primary key,

   branch_city varchar(20),

   assets real

);
```

```
create table bankacc(
          accno int primary key,
   branch_name varchar(30),
   balance int,
   foreign key(branch_name) references branch(branch_name)
);

create table bankcustomer(
          customer_name varchar(20) primary key,
   customer_street varchar(30),
   city varchar(20)
);

create table depositor(
          customer_name varchar(20),
   accno int,
   primary key(customer_name,accno),
   foreign key(customer_name) references bankcustomer(customer_name),
   foreign key(accno) references bankacc(accno)
);

create table loan(
          loan_no int primary key,
   branch_name varchar(30),
   amt int,
   foreign key(branch_name) references bankacc(branch_name) );
```

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

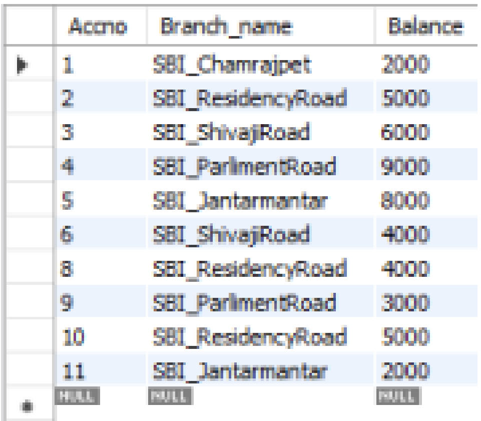
Inserting Values to the table

```
insert into branch values("SBI_Chamrajpet","Bangalore",50000);
insert into branch values("SBI_ResidencyRoad","Bangalore",10000);
insert into branch values("SBI_ShivajiRoad","Bombay",20000);
insert into branch values("SBI_ParlimentRoad","Delhi",10000);
insert into branch values("SBI_Jantarmantar","Delhi",20000);
select * from branch;
```



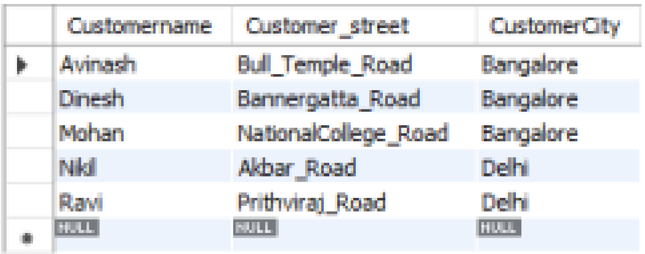
Branch_name	Branch_city	assets
SBI_Chamrajpet	Bangalore	50000
SBI_Jantarmantar	Delhi	20000
SBI_ParlimentRoad	Delhi	10000
SBI_ResidencyRoad	Bangalore	10000
SBI_ShivajiRoad	Bombay	20000
NULL	NULL	NULL

```
insert into bankacc values(1,"SBI_Chamrajpet",2000);
insert into bankacc values(2,"SBI_ResidencyRoad",5000);
insert into bankacc values(3,"SBI_ShivajiRoad",6000);
insert into bankacc values(4,"SBI_ParlimentRoad",9000);
insert into bankacc values(5,"SBI_Jantarmantar",8000);
insert into bankacc values(6,"SBI_ShivajiRoad",4000);
insert into bankacc values(8,"SBI_ResidencyRoad",4000);
insert into bankacc values(9,"SBI_ParlimentRoad",3000);
insert into bankacc values(10,"SBI_ResidencyRoad",5000);
insert into bankacc values(11,"SBI_Jantarmantar",2000);
select * from bankacc;
```



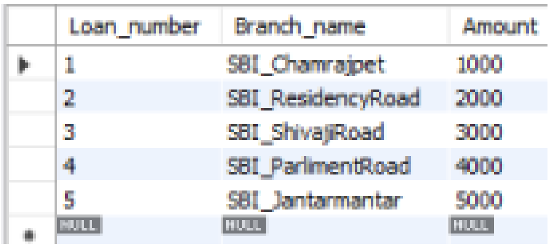
	Accno	Branch_name	Balance
▶	1	SBI_Chamrajpet	2000
	2	SBI_ResidencyRoad	5000
	3	SBI_ShivajiRoad	6000
	4	SBI_ParlimentRoad	9000
	5	SBI_Jantarmantar	8000
	6	SBI_ShivajiRoad	4000
	8	SBI_ResidencyRoad	4000
	9	SBI_ParlimentRoad	3000
	10	SBI_ResidencyRoad	5000
	11	SBI_Jantarmantar	2000
*	NULL	NULL	NULL

```
insert into bankcustomer values("Avinash","Bull_Temple_Road","Bangalore");
insert into bankcustomer values("Dinesh","Bannergatta_Road","Bangalore");
insert into bankcustomer values("Mohan","NationalCollege_Road","Bangalore");
insert into bankcustomer values("Nikil","Akbar_Road","Delhi");
insert into bankcustomer values("Ravi","Prithviraj_Road","Delhi");
select * from bankcustomer;
```



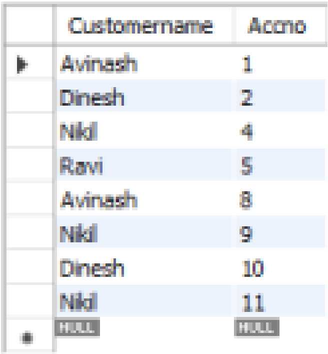
	Customername	Customer_street	CustomerCity
▶	Avinash	Bull_Temple_Road	Bangalore
	Dinesh	Bannergatta_Road	Bangalore
	Mohan	NationalCollege_Road	Bangalore
	Nikil	Akbar_Road	Delhi
	Ravi	Prithviraj_Road	Delhi
*	NULL	NULL	NULL

```
insert into loan values(1,"SBI_Chamrajpet",1000);
insert into loan values(2,"SBI_ResidencyRoad",2000);
insert into loan values(3,"SBI_ShivajiRoad",3000);
insert into loan values(4,"SBI_ParlimentRoad",4000);
insert into loan values(5,"SBI_Jantarmantar",5000);
select * from loan;
```



Loan_number	Branch_name	Amount
1	SBI_Chamrajpet	1000
2	SBI_ResidencyRoad	2000
3	SBI_ShivajiRoad	3000
4	SBI_ParlimentRoad	4000
5	SBI_Jantarmantar	5000
NULL	NULL	NULL

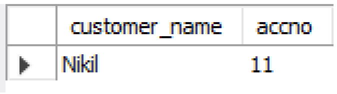
```
insert into depositor values("Avinash",1);
insert into depositor values("Dinesh",2);
insert into depositor values("Nikil",4);
insert into depositor values("Ravi",5);
insert into depositor values("Avinash",8);
insert into depositor values("Nikil",9);
insert into depositor values("Dinesh",10);
insert into depositor values("Nikil",11);
select* from depositor;
```



Customername	Accno
Avinash	1
Dinesh	2
Nikil	4
Ravi	5
Avinash	8
Nikil	9
Dinesh	10
Nikil	11
NULL	NULL

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

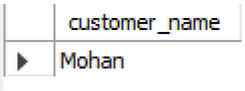
```
select d.customer_name,d.accno from bankcustomer c,depositor d, bankaccount ba, branch b
where b.branch_city="Delhi" and d.accno=ba.accno and b.branch_name=ba.branch_name
group by d.customer_name
having count(distinct b.branch_name)= (select count(distinct b.branch_name) from branch b
where b.branch_city="Delhi");
```



customer_name	accno
Nikil	11

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

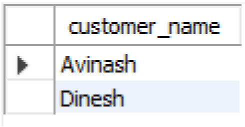
select distinct b.customer_name from Borrower b, depositor d
where b.customer_name NOT IN(select d.customer_name from loan l,depositor d, Borrower b
where l.loan_no=b.loan_no and d.customer_name=b.customer_name);



customer_name
Mohan

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

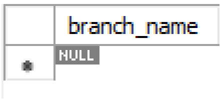
select distinct d.customer_name from depositor d where d.customer_name IN(
select d.customer_name from branch br,depositor d, bankaccount ba
where br.branch_city='Bangalore' and br.branch_name=ba.branch_name and ba.accno=d.accno
and customer_name IN(select customer_name from Borrower));



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 b where b.assets > ALL(select SUM(b.assets) from branch b
where b.branch_city = 'Bangalore')



branch_name
NULL

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

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

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

delete ba.* from bankaccount ba, branch b where branch_city=”Bombay” and
ba.branch_name=b.branch_name;

Employee Database

Question

(Week 5)

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

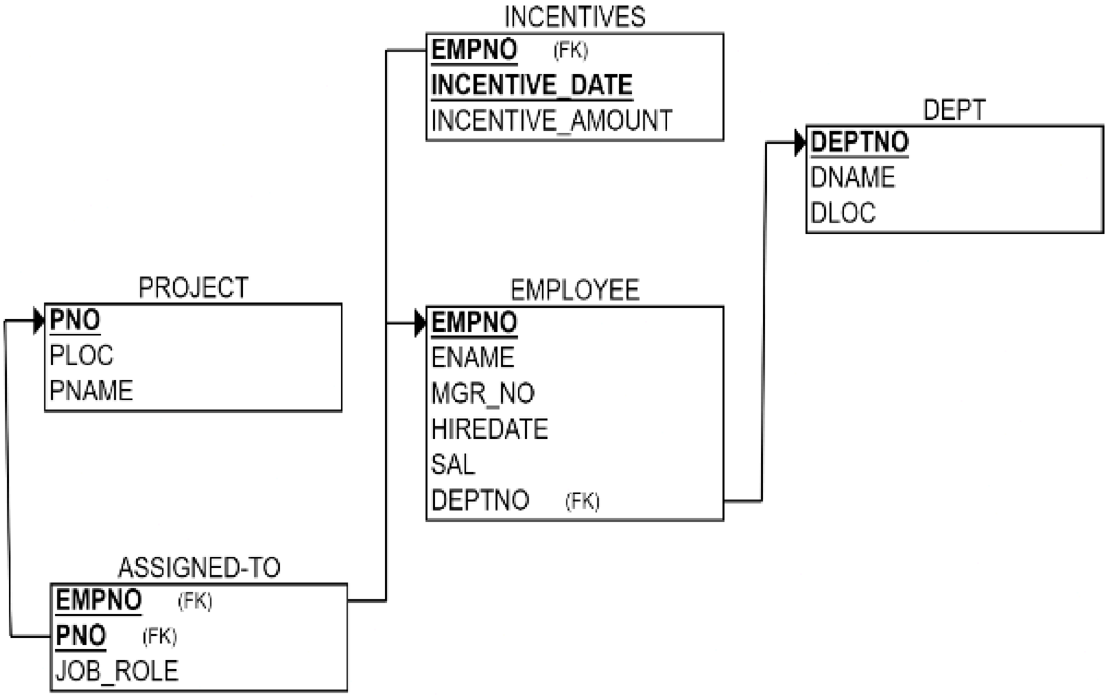
2. Enter greater than five tuples for each table.

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.

Schema Diagram



Create database

```
create database employee;

use employee;
```

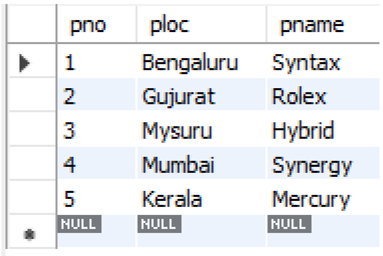
Create Tables

```
create table project(
          pno int primary key,
   ploc varchar(40),
   pname varchar(40)
);


create table dept(
          deptno int primary key,
   dname varchar(40),
   dloc varchar(40)
);
```

```
create table employee(
        empno int primary key,
   ename varchar(30),
   mgr_no int,
   hiredate date,
   sal int,
   deptno int,
   foreign key(deptno) references dept(deptno)
);

create table incentives(
        empno int,
   incentive_date date primary key,
   incentive_amt int,
   foreign key(empno) references employee (empno)
);

create table assigned_to(
        empno int,
   pno int,
   job_role varchar(40),
   foreign key(empno) references employee (empno),
   foreign key(pno) references project (pno)
);
```

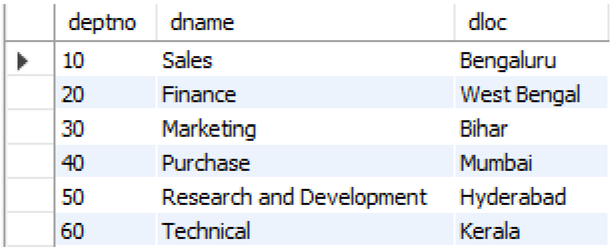
Inserting Values to the table

```
insert into project values(1,'Bengaluru','Syntax');
insert into project values(2,'Gujurat','Rolex');
insert into project values(3,'Mysuru','Hybrid');
insert into project values(4,'Mumbai','Synergy');
insert into project values(5,'Kerala','Mercury');
select *from project;
```



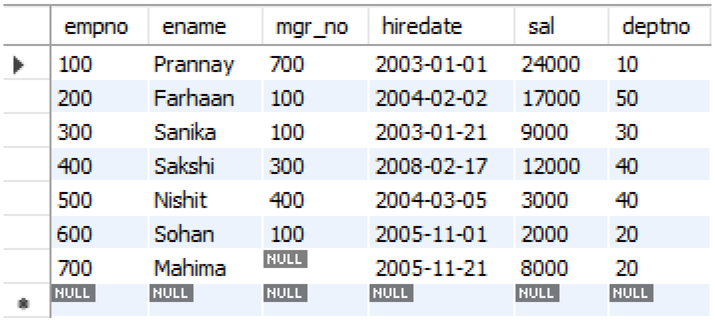
pno	ploc	pname
1	Bengaluru	Syntax
2	Gujurat	Rolex
3	Mysuru	Hybrid
4	Mumbai	Synergy
5	Kerala	Mercury
NULL	NULL	NULL

```
insert into dept values(10,'Sales','Bengaluru');
insert into dept values(20,'Finance','West Bengal');
insert into dept values(30,'Marketing','Bihar');
insert into dept values(40,'Purchase','Mumbai');
insert into dept values(50,'Research and Development','Hyderabad');
insert into dept values(60,'Technical','Kerala');
select * from dept;
```



deptno	dname	dloc
10	Sales	Bengaluru
20	Finance	West Bengal
30	Marketing	Bihar
40	Purchase	Mumbai
50	Research and Development	Hyderabad
60	Technical	Kerala

```
insert into employee values(100,'Prannay',700,'2003-01-01',24000,10);
insert into employee values(200,'Farhaan',100,'2004-02-02',17000,50);
insert into employee values(300,'Sanika',100,'2003-01-21',9000,30);
insert into employee values(400,'Sakshi',300,'2008-02-17',12000,40);
insert into employee values(500,'Nishit',400,'2004-03-05',3000,40);
insert into employee values(600,'Sohan',100,'2005-11-01',2000,20);
insert into employee values(700,'Mahima',NULL,'2005-11-21',8000,20);
select *from employee;
```



empno	ename	mgr_no	hiredate	sal	deptno
100	Prannay	700	2003-01-01	24000	10
200	Farhaan	100	2004-02-02	17000	50
300	Sanika	100	2003-01-21	9000	30
400	Sakshi	300	2008-02-17	12000	40
500	Nishit	400	2004-03-05	3000	40
600	Sohan	100	2005-11-01	2000	20
700	Mahima	NULL	2005-11-21	8000	20
NULL	NULL	NULL	NULL	NULL	NULL

insert into incentives values(100,'2019-02-17',6000);

insert into incentives values(200,'2019-05-21',7000);

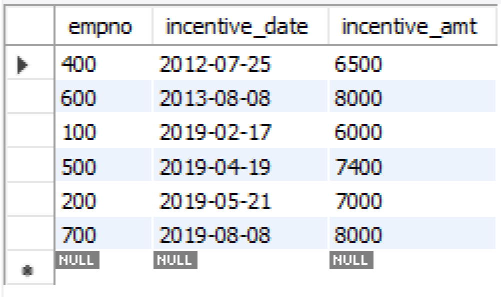
insert into incentives values(400,'2012-07-25',6500);

insert into incentives values(500,'2019-04-19',7400);

insert into incentives values(600,'2013-08-08',8000);

insert into incentives values(700,'2019-08-08',8000);

select *from incentives;



empno	incentive_date	incentive_amt
400	2012-07-25	6500
600	2013-08-08	8000
100	2019-02-17	6000
500	2019-04-19	7400
200	2019-05-21	7000
700	2019-08-08	8000
NULL	NULL	NULL

insert into assigned_to values(100,1,'Project Manager');

insert into assigned_to values(200,1,'Resource Manager');

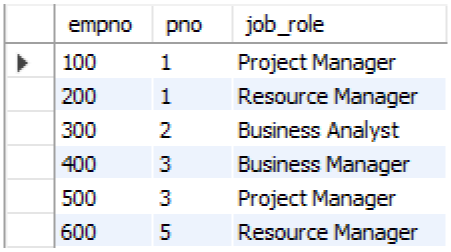
insert into assigned_to values(300,2,'Business Analyst');

insert into assigned_to values(400,3,'Business Manager');

insert into assigned_to values(500,3,'Project Manager');

insert into assigned_to values(600,5,'Resource Manager');

select *from assigned_to;

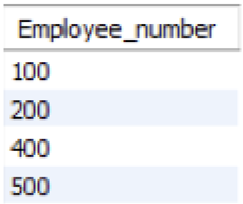


	empno	pno	job_role
▶	100	1	Project Manager
	200	1	Resource Manager
	300	2	Business Analyst
	400	3	Business Manager
	500	3	Project Manager
	600	5	Resource Manager

Queries

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

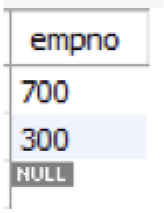
- select a.empno Employee_number from project p, assigned_to a where p.pno=a.pno and p.ploc in("Hyderabad","Bengaluru","Mysuru");



Employee_number
100
200
400
500

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

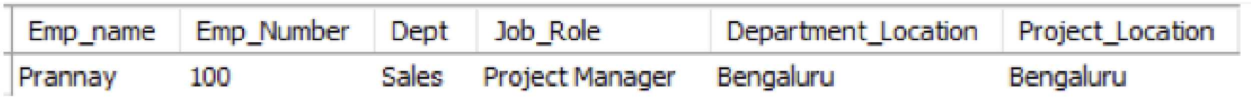
- select e.empno from employee e where e.empno NOT IN (select i.empno from incentives i);



empno
700
300
NULL

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

select e.ename Emp_name, e.empno Emp_Number, d.dname Dept, a.job_role Job_Role, d.dloc Department_Location, p.ploc Project_Location from project p, dept d, employee e, assigned_to a
where e.empno=a.empno and p.pno=a.pno and e.deptno=d.deptno and p.ploc=d.dloc;



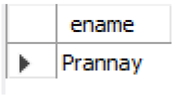
Emp_name	Emp_Number	Dept	Job_Role	Department_Location	Project_Location
Prannay	100	Sales	Project Manager	Bengaluru	Bengaluru

(Week 6)

Queries

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

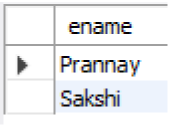
```
select e1.ename

from employee e1, employee e2

where e1.empno=e2.mgr_no group by e1.ename

having count(e1.mgr_no)=(select count(e1.ename)

from employee e1, employee e2 where e1.empno=e2.mgr_no

group by e1.ename order by count(e1.ename) desc limit 1);
```



ename
Prannay

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

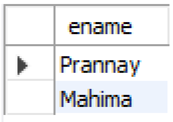
```
select m.ename from employee m
where m.empno in
   (select mgr_no from employee)
    and m.sal>(select avg(n.sal) from employee n
    where n.mgr_no=m.empno);
```



ename
Prannay
Sakshi

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

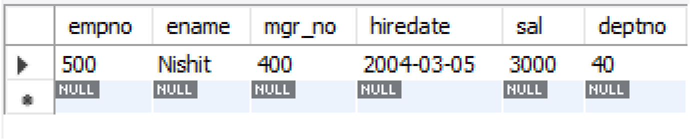
```
select ename from employee where empno in(select distinct mgr_no
from employee where empno in
          (select distinct mgr_no from employee where empno in
                         (select distinct mgr_no from employee)));
```



ename
Prannay
Mahima

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

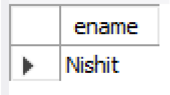
```
select * from employee where empno=
    (select i.empno from incentives i
    where i.incentive_amount= (select max(n.incentive_amount) from incentives n
    where n.incentive_amount<(select max(inc.incentive_amount) from incentives inc
    where inc.incentive_date between '2019-01-01' and '2019-12-31') and
    incentive_date between '2019-01-01' and '2019-12-31'));
```



empno	ename	mgr_no	hiredate	sal	deptno
500	Nishit	400	2004-03-05	3000	40
NULL	NULL	NULL	NULL	NULL	NULL

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

select e2.ename
from employee e1, employee e2
where e1.empno=e2.mgr_no and e1.deptno=e2.deptno;



ename
Nishit

Supplier Database

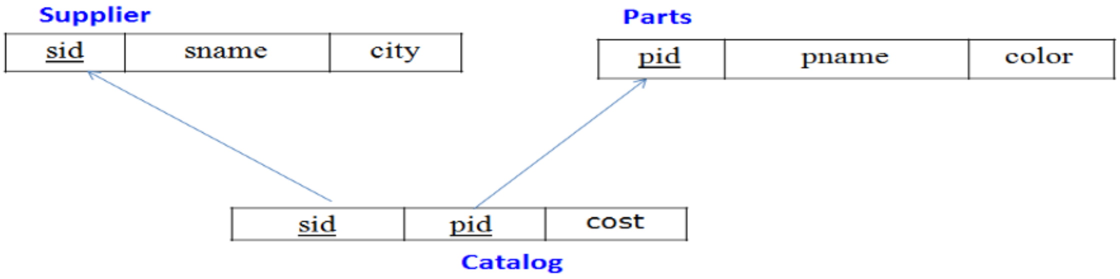
Question

(Week 7)

1. Using Scheme diagram, create tables by properly specifying the primary keys and the foreign keys.
2. Insert appropriate records in each table.
3. Find the pnames of parts for which there is some supplier.
4. Find the snames of suppliers who supply every part.
5. Find the snames of suppliers who supply every red part.
6. Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.
7. 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).

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

Schema Diagram

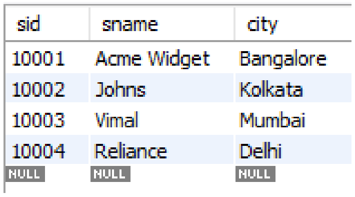


Creating database and table:

```
create database supplier_141;
use supplier_141;

create table Supplier
        (sid int primary key,
sname varchar(35),
   city varchar(35));

create table parts
        (pid int primary key,
pname varchar(35),
   color varchar(35));

create table catalog
        (sid int,
pid int,
   cost float,
   primary key(sid,pid),
   foreign key(sid) references Supplier(sid),
   foreign key(pid) references parts(pid));
```

Inserting values to tables:

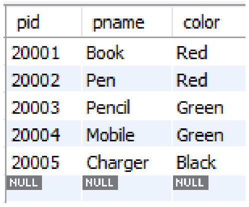
```
insert into Supplier values (10001,"Acme Widget","Bangalore"),
```



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

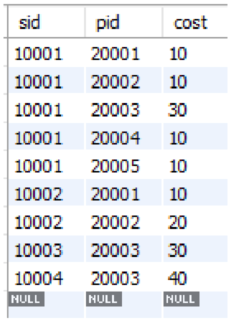
```
(10002,"Johns","Kolkata"),
(10003,"Vimal","Mumbai"),
(10004,"Reliance","Delhi");
Select * from  Supplier;
```

```
insert into parts values
(20001,"Book","Red"),
(20002,"Pen","Red"),
(20003,"Pencil","Green"),
(20004,"Mobile","Green"),
(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),
(10001,20002,10),
(10001,20003,30),
(10001,20004,10),
(10001,20005,10),
(10002,20001,10),
(10002,20002,20),
(10003,20003,30),
(10004,20003,40);
Select * from  catalog;
```

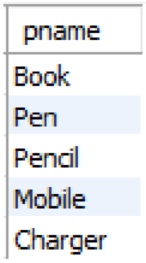


sid	pid	cost
10001	20001	10
10001	20002	10
10001	20003	30
10001	20004	10
10001	20005	10
10002	20001	10
10002	20002	20
10003	20003	30
10004	20003	40
NULL	NULL	NULL

Queries

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

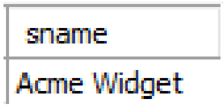
```
select distinct pname from parts p,catalog c where p.pid=c.pid;
```



pname
Book
Pen
Pencil
Mobile
Charger

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

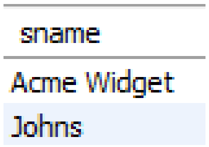
```
select sname from Supplier where sid in(select sid from catalog c group by sid having
count(pid)=(select count(pid) from parts));
```



sname
Acme Widget

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

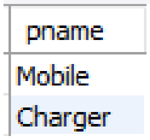
```
select distinct sname from Supplier s,catalog c where s.sid=c.sid and pid in(select pid from
parts where color="red");
```



sname
Acme Widget
Johns

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

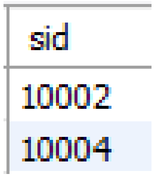
```
select pname from parts p,supplier s where pid in(select pid from catalog group by pid
having count(pid)=1) and s.sname="Acme Widget";
```



pname
Mobile
Charger

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).

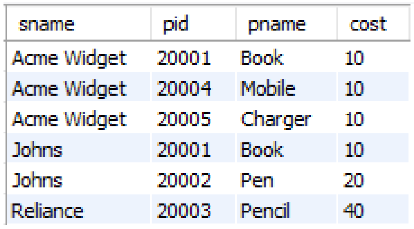
```
create view c as select c.pid,p.pname,avg(cost) as co from catalog c,parts p where
c.pid=p.pid group by c.pid;
select ca.sid from catalog ca,c where ca.pid=c.pid and ca.cost>c.co and c.pid=ca.pid;
```



sid
10002
10004

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

```
select sname,co.pid,pname,cost from Supplier s,parts po,catalog co where co.pid=po.pid
and s.sid=co.sid and co.cost =(select max(cost) from catalog where pid=po.pid) ;
```



sname	pid	pname	cost
Acme Widget	20001	Book	10
Acme Widget	20004	Mobile	10
Acme Widget	20005	Charger	10
Johns	20001	Book	10
Johns	20002	Pen	20
Reliance	20003	Pencil	40