VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

Database Management Systems (23CS3PCDBM)

Submitted by

Bhavya J Makadia (1BM23CS064)

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
Sep-2024 to Jan-2025

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 **Bhavya J Makadia (1BM23CS064),** who is a 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 2024. 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.

Dr. Kayarvizhy N Assistant Professor Department of CSE, BMSCE	**Dr. Kavitha Sooda** Professor & HOD Department of CSE, BMSCE

Index

GitHub link:
https://github.com/Bhavya404/1BM23CS064_DBMS_3B

Insurance Database

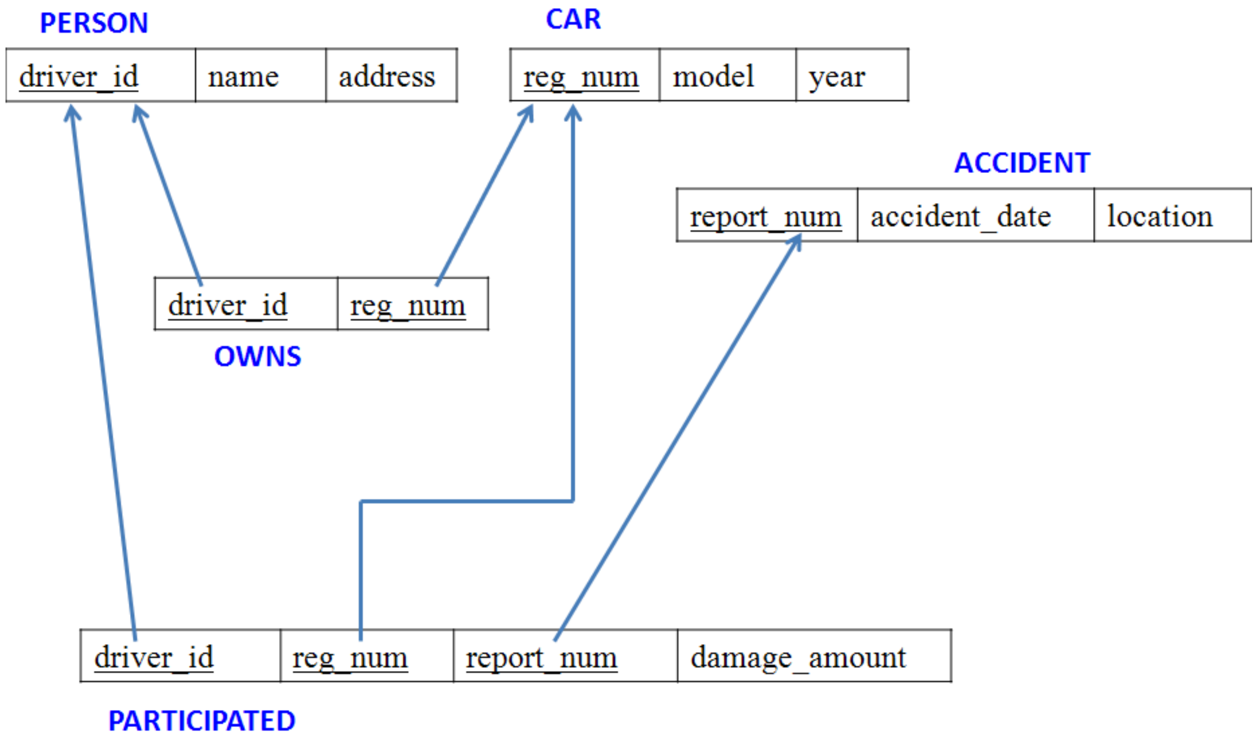
Question

(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
- Display Accident date and location
- Display driver id who did the accident damage greater than or equal to Rs.25000
- Add a new accident to the database.
- To Do
- Display Accident date and location
- Display driver id who did accident with damage amount greater than or equal to Rs.25000

Schema Diagram



Create database

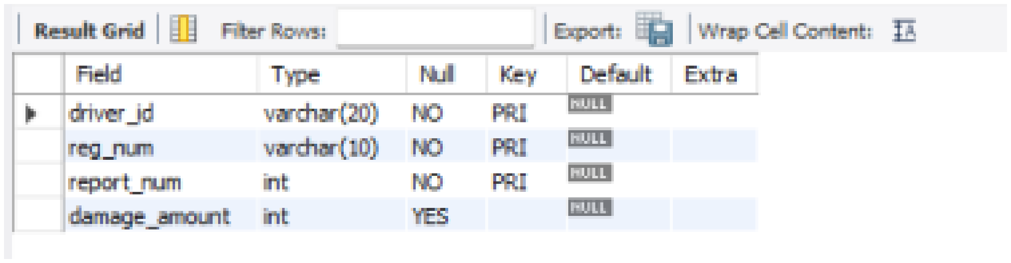
```
create database insurance_cs065;

use insurance_dhiksha;
```

Create table

```
create table insurance_dhiksha.person(

driver_id varchar(20),

name varchar(30),

address varchar(50),

PRIMARY KEY(driver_id)

);

create table insurance_dhiksha.car(

reg_num varchar(15),

model varchar(10),
year int,
PRIMARY KEY(reg_num)
);
create table insurance_dhiksha.owns(
driver_id varchar(20),
reg_num varchar(10),
PRIMARY KEY(driver_id, reg_num),
FOREIGN KEY(driver_id) REFERENCES person(driver_id),
FOREIGN KEY(reg_num) REFERENCES car(reg_num)
);
create table insurance_dhiksha.accident(
report_num int,
accident_date date,
location varchar(50),
PRIMARY KEY(report_num)
);
create table insurance_dhiksha.participated(
driver_id varchar(20),
reg_num varchar(10),
report_num int,
damage_amount 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)
);
```

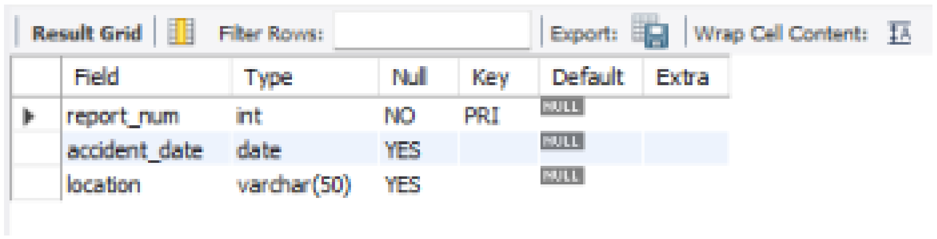
Structure of the table

desc person;



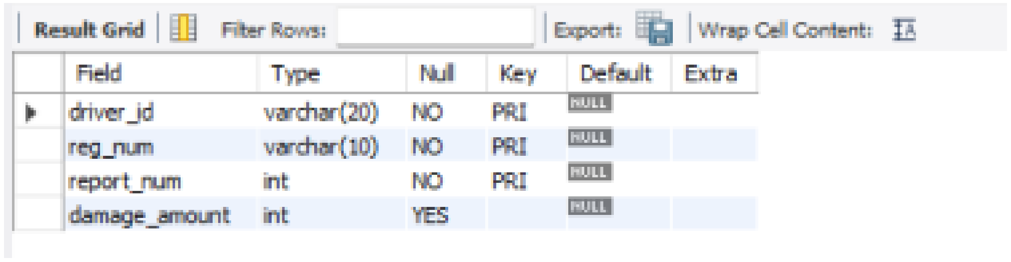
	Field	Type	Null	Key	Default	Extra
▶	driver_id	varchar(20)	NO	PRI	NULL	
	reg_num	varchar(10)	NO	PRI	NULL	
	report_num	int	NO	PRI	NULL	
	damage_amount	int	YES		NULL	

desc accident;



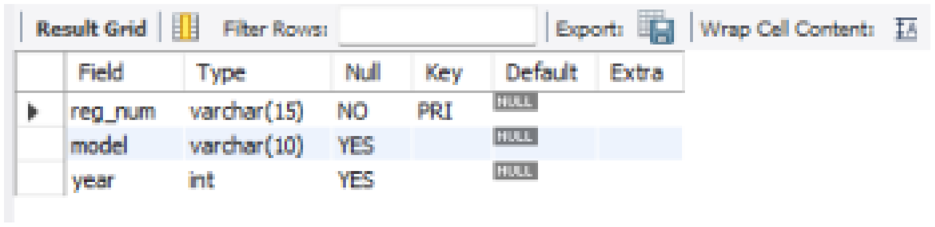
	Field	Type	Null	Key	Default	Extra
▶	report_num	int	NO	PRI	NULL	
	accident_date	date	YES		NULL	
	location	varchar(50)	YES		NULL	

desc participated;



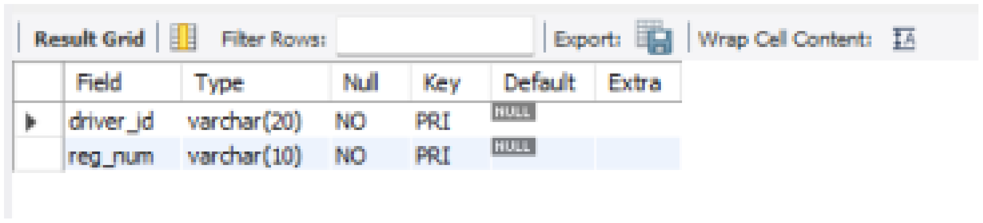
	Field	Type	Null	Key	Default	Extra
▶	driver_id	varchar(20)	NO	PRI	NULL	
	reg_num	varchar(10)	NO	PRI	NULL	
	report_num	int	NO	PRI	NULL	
	damage_amount	int	YES		NULL	

desc car;



	Field	Type	Null	Key	Default	Extra
▶	reg_num	varchar(15)	NO	PRI	NULL	
	model	varchar(10)	YES		NULL	
	year	int	YES		NULL	

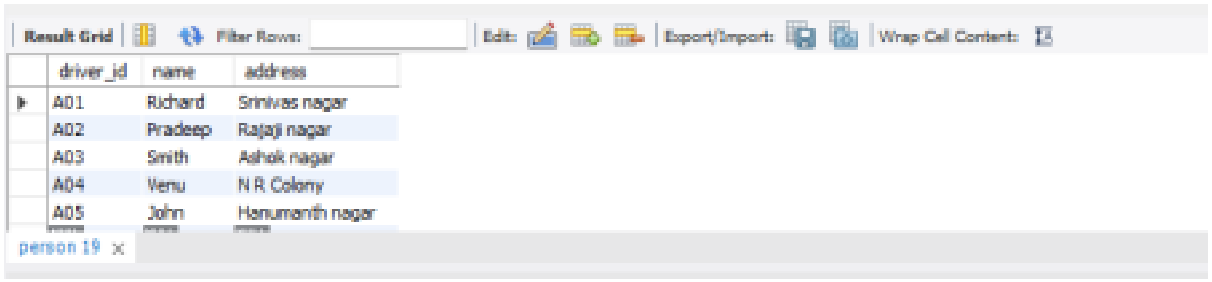
desc owns;



	Field	Type	Null	Key	Default	Extra
▶	driver_id	varchar(20)	NO	PRI	NULL	
	reg_num	varchar(10)	NO	PRI	NULL	

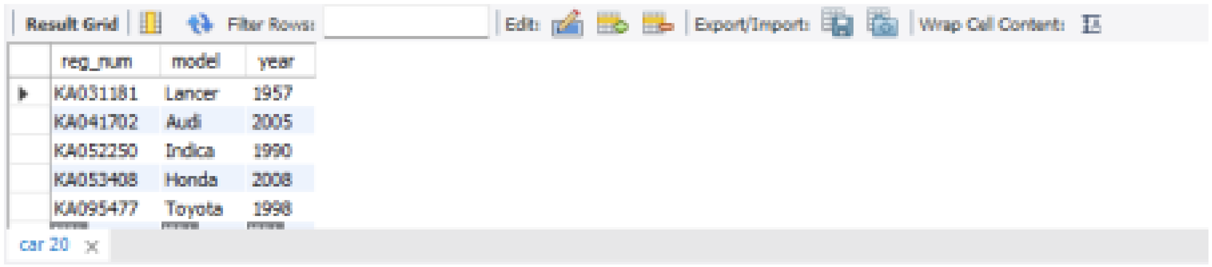
Inserting Values to the table

```
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", "N R Colony");

insert into person values("A05","John", "Hanumanth nagar");

select * from person;
```



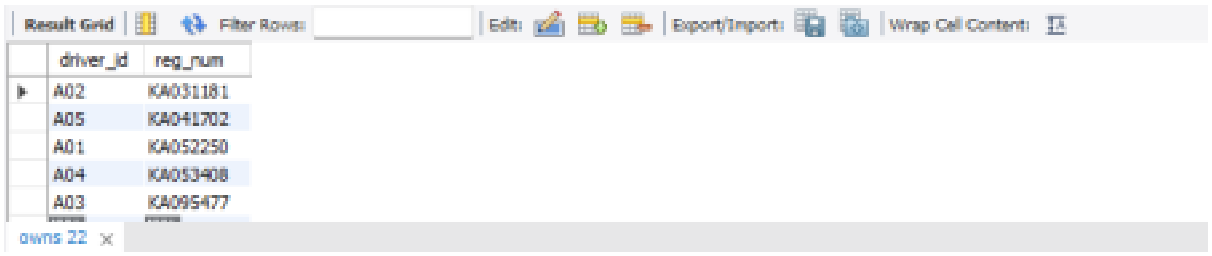
driver_id	name	address
A01	Richard	Srinivas nagar
A02	Pradeep	Rajaji nagar
A03	Smith	Ashok nagar
A04	Venu	N R Colony
A05	John	Hanumanth nagar

```
insert into car values("KA052250","Indica", "1990");
insert into car values("KA031181","Lancer", "1957");

insert into car values("KA095477","Toyota", "1998");

insert into car values("KA053408","Honda", "2008");

insert into car values("KA041702","Audi", "2005");

select * from car;
```



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

```
insert into owns values("A01","KA052250");

insert into owns values("A02","KA031181");

insert into owns values("A03","KA095477");

insert into owns values("A04","KA053408");

insert into owns values("A05","KA041702");

select * from owns;
```

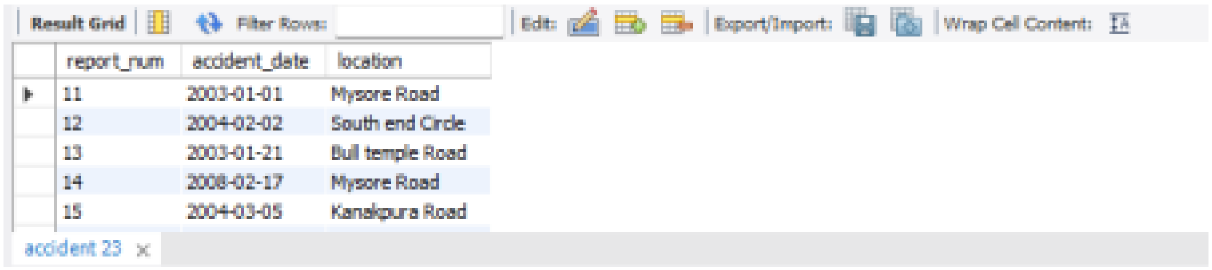


Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

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

owns 22

```
insert into accident values(11,'2003-01-01',"Mysore Road");
insert into accident values(12,'2004-02-02',"South end Circle");

insert into accident values(13,'2003-01-21',"Bull temple Road");

insert into accident values(14,'2008-02-17',"Mysore Road");

insert into accident values(15,'2004-03-05',"Kanakpura Road");

select * from accident;
```

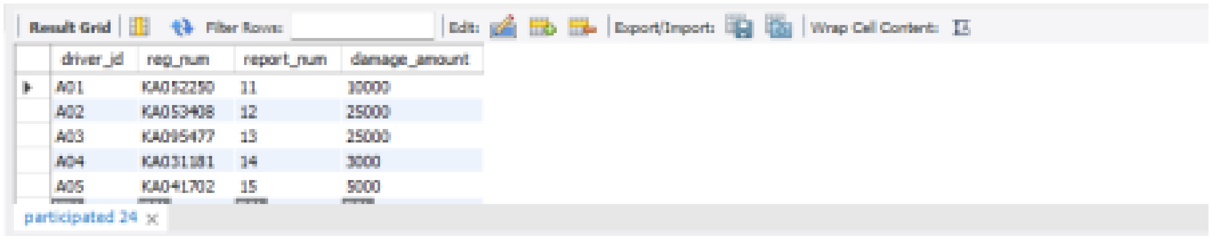


Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

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	2004-03-05	Kanakpura Road

accident 23

```
insert into participated values("A01","KA052250",11,10000);
insert into participated values("A02","KA053408",12,50000);

insert into participated values("A03","KA095477",13,25000);

insert into participated values("A04","KA031181",14,3000);

insert into participated values("A05","KA041702",15,5000);

select * from participated;
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

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

participated 24

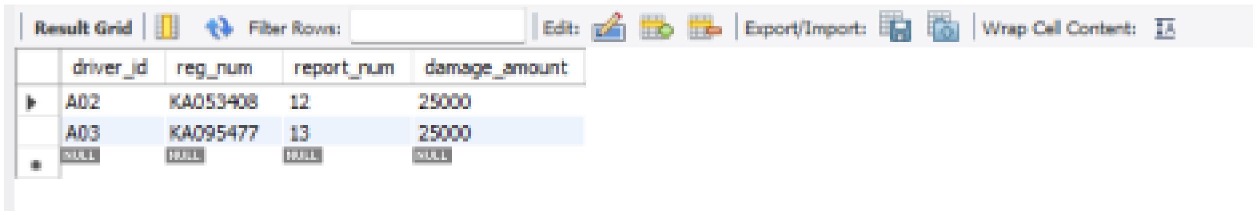
Queries

- **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

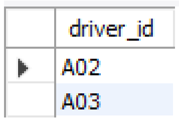
  where reg_num='KA053408' and report_num=12;



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

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

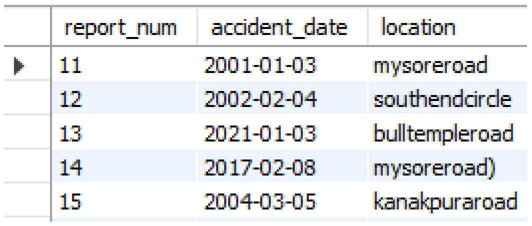
  select count(distinct driver_id) CNT
  from participated a, accident b
  where a.report_num=b.report_num and b.accident_date like '2008%';



driver_id
A02
A03

- **Add a new accident to the database.**

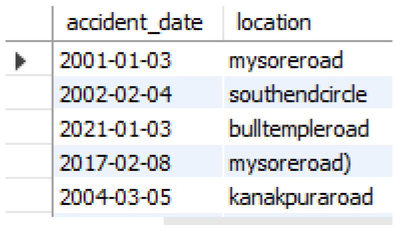
  insert into accident values(16,'2008-03-08',"Domlur"); select * from accident;



report_num	accident_date	location
11	2001-01-03	mysoreroad
12	2002-02-04	southendcircle
13	2021-01-03	bulltempleroad
14	2017-02-08	mysoreroad)
15	2004-03-05	kanakpuraroad

- **Display Accident date and location**

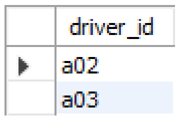
  select accident_date ,location from accident;



accident_date	location
2001-01-03	mysoreroad
2002-02-04	southendcircle
2021-01-03	bulltempleroad
2017-02-08	mysoreroad)
2004-03-05	kanakpuraroad

- **Display driver id who did accident with damage amount greater than or equal to Rs.25000**

```
select driver_id from participated
where damage_amount>=25000;
```



driver_id
a02
a03

Bank Database

Question:
(Week 3&4)

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

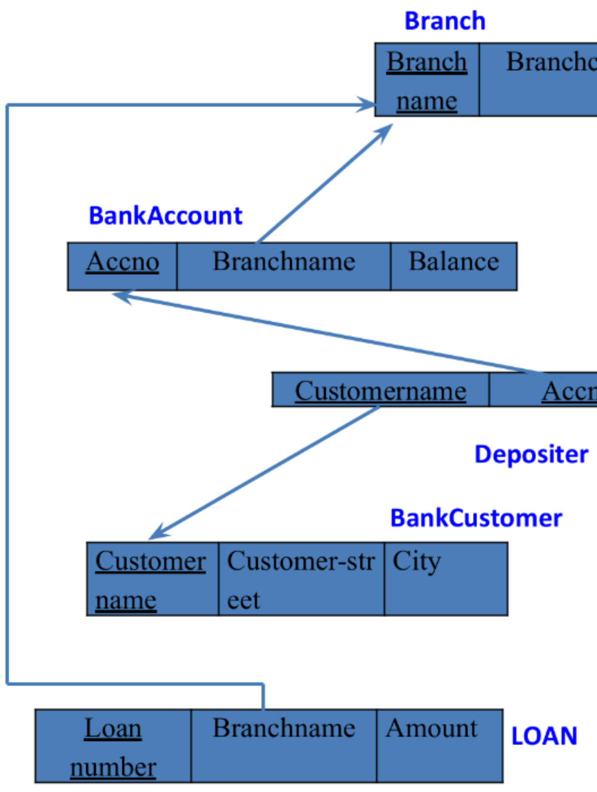
Enter at least five tuples for each relation.

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.

Schema Diagram



CREATE DATABASE

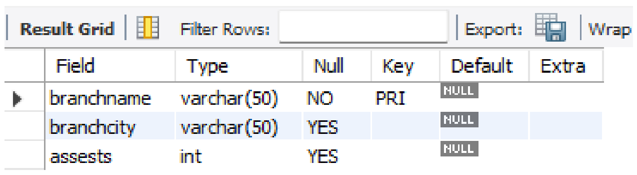
```
create database bhavya_j_064;
use bhavya_j_064;
```

CREATE TABLES

```
create table branch (
branchname varchar(50),
branchcity varchar(50),
assests int ,
primary key (branchname));

create table bankcustomer(
customername varchar(50),
customer_street varchar(50),
city varchar(50),
primary key(customername));

create table bankaccount (
accno int,
branchname varchar(50),
balance int,
primary key (accno),
foreign key (branchname) references branch (branchname));

create table depositer(
customername varchar(50),
accno int,
primary key (customername, accno),
foreign key (customername) references bankcustomer(customername),
foreign key (accno) references bankaccount(accno));

create table loan(
loannumber int,
branchname varchar(50),
amount int,
primary key (loannumber),
foreign key (branchname) references branch (branchname));
```

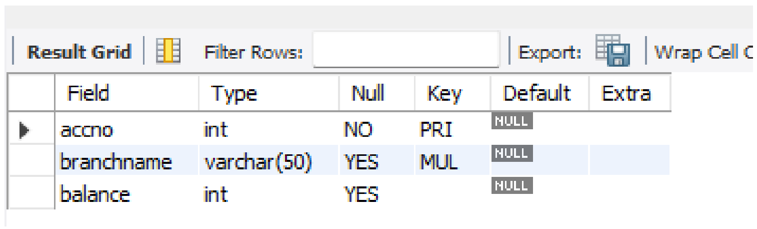
STRUCTURE OF TABLE

desc branch;



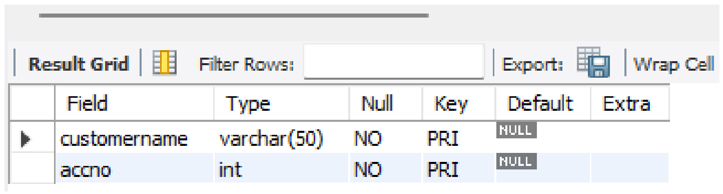
Field	Type	Null	Key	Default	Extra
branchname	varchar(50)	NO	PRI	NULL	
branchcity	varchar(50)	YES		NULL	
assests	int	YES		NULL	

desc bankaccount;



Field	Type	Null	Key	Default	Extra
accno	int	NO	PRI	NULL	
branchname	varchar(50)	YES	MUL	NULL	
balance	int	YES		NULL	

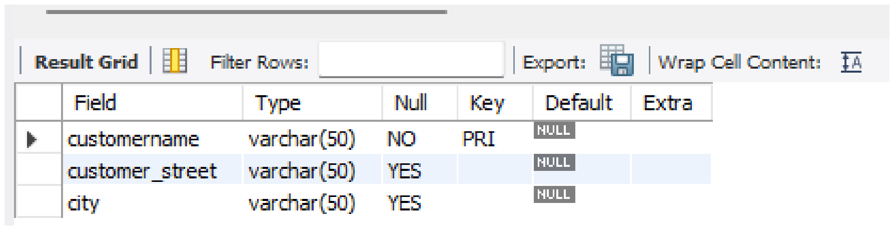
desc depositer;



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

	Field	Type	Null	Key	Default	Extra
▶	customername	varchar(50)	NO	PRI	NULL	
	accno	int	NO	PRI	NULL	

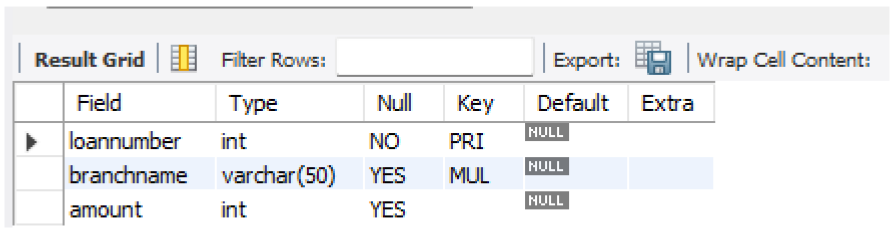
desc bankcustomer;



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

	Field	Type	Null	Key	Default	Extra
▶	customername	varchar(50)	NO	PRI	NULL	
	customer_street	varchar(50)	YES		NULL	
	city	varchar(50)	YES		NULL	

desc loan;



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

	Field	Type	Null	Key	Default	Extra
▶	loannumber	int	NO	PRI	NULL	
	branchname	varchar(50)	YES	MUL	NULL	
	amount	int	YES		NULL	

INSERTING VALUES INTO THE TABLE

insert into branch

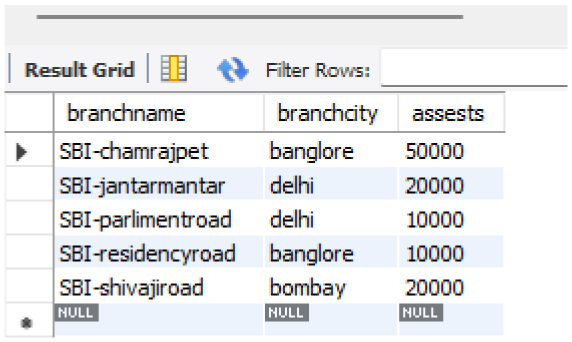
values('SBI-chamrajpet','banglore', 50000),

('SBI-residencyroad','banglore',10000),

('SBI-shivajiroad','bombay',20000),

('SBI-parlimentroad','delhi',10000),

('SBI-jantarmantar','delhi',20000);



branchname	branchcity	assests
SBI-chamrajpet	banglore	50000
SBI-jantarmantar	delhi	20000
SBI-parlimentroad	delhi	10000
SBI-residencyroad	banglore	10000
SBI-shivajiroad	bombay	20000
NULL	NULL	NULL

insert into bankcustomer

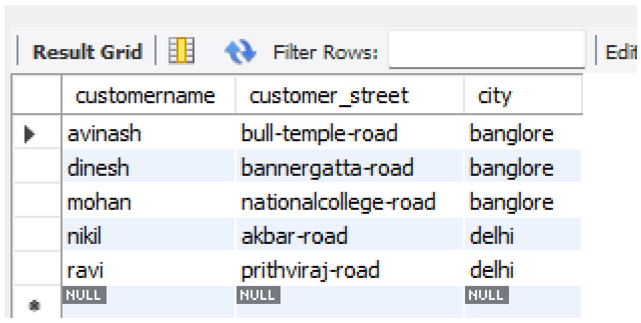
values('avinash','bull-temple-road','banglore'),

('dinesh','bannergatta-road','banglore'),

('mohan','nationalcollege-road','banglore'),

('nikil','akbar-road','delhi'),

('ravi','prithviraj-road','delhi');



customername	customer_street	city
avinash	bull-temple-road	banglore
dinesh	bannergatta-road	banglore
mohan	nationalcollege-road	banglore
nikil	akbar-road	delhi
ravi	prithviraj-road	delhi
NULL	NULL	NULL

insert into bankaccount

values(1,'SBI-chamrajpet',2000),

(2,'SBI-residencyroad',5000),

(3,'SBI-shivajiroad',6000),

(4,'SBI-parlimentroad',9000),

(5,'SBI-jantarmantar',8000),

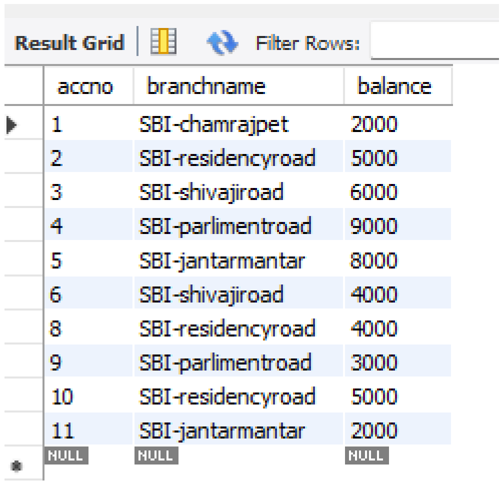
(6,'SBI-shivajiroad',4000),

(8,'SBI-residencyroad',4000),

(9,'SBI-parlimentroad',3000),

(10,'SBI-residencyroad',5000),

(11,'SBI-jantarmantar',2000);



accno	branchname	balance
1	SBI-chamrajpet	2000
2	SBI-residencyroad	5000
3	SBI-shivajiroad	6000
4	SBI-parlimentroad	9000
5	SBI-jantarmantar	8000
6	SBI-shivajiroad	4000
8	SBI-residencyroad	4000
9	SBI-parlimentroad	3000
10	SBI-residencyroad	5000
11	SBI-jantarmantar	2000
NULL	NULL	NULL

insert into depositer

values('avinash',1),

('dinesh',2),

('nikil',4),

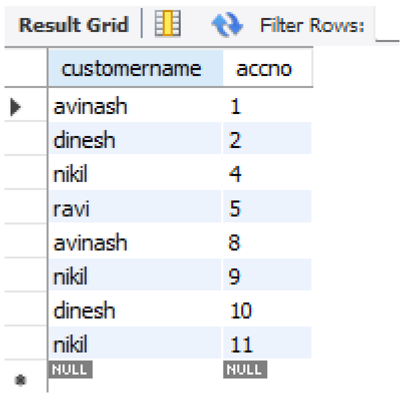
('ravi',5),

('avinash',8),

('nikil',9),

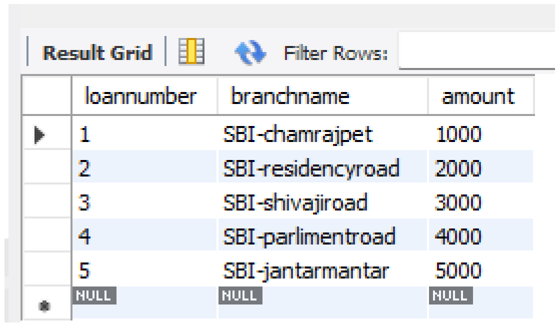
('dinesh',10),

('nikil',11);



customername	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-parlimentroad',4000),
(5,'SBI-jantarmantar',5000);
```

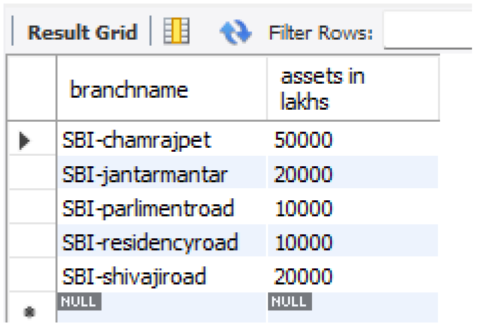


loannumber	branchname	amount
1	SBI-chamrajpet	1000
2	SBI-residencyroad	2000
3	SBI-shivajiroad	3000
4	SBI-parlimentroad	4000
5	SBI-jantarmantar	5000
NULL	NULL	NULL

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 branchname,assests as 'assets in lakhs'
from  branch;
```

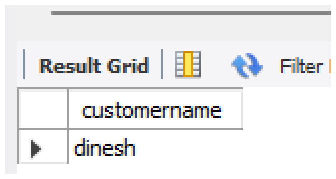


branchname	assets in lakhs
SBI-chamrajpet	50000
SBI-jantarmantar	20000
SBI-parlimentroad	10000
SBI-residencyroad	10000
SBI-shivajiroad	20000
NULL	NULL

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

```
select d.customername
from bankaccount b, depositer d
```

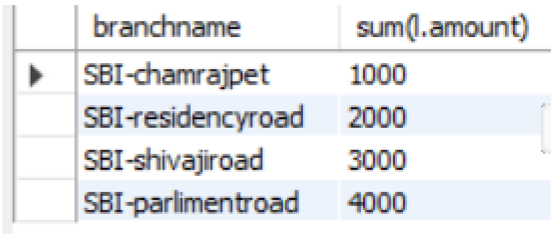
```
where b.accno=d.accno and branchname='SBI-residencyroad'
group by customername
having count(*)>=2;
```



customername
dinesh

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

```
create view loan_info as
select b.branchname, sum(l.amount)
from branch b , loan l
where b.branchname=l.branchname
group by l.branchname;
select * from loan_info;
```

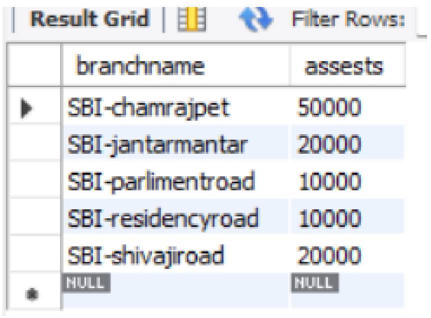


branchname	sum(l.amount)
SBI-chamrajpet	1000
SBI-residencyroad	2000
SBI-shivajiroad	3000
SBI-parlimentroad	4000

Week - 04 - Additional queries

4. Retrieve all branches and their respective total assets

```
select branchname, assests
from branch;
```



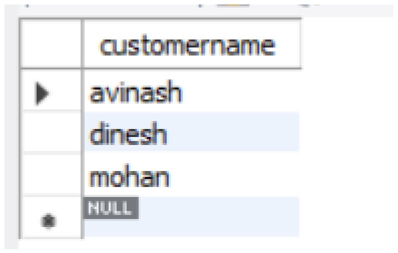
branchname	assests
SBI-chamrajpet	50000
SBI-jantarmantar	20000
SBI-parlimentroad	10000
SBI-residencyroad	10000
SBI-shivajiroad	20000
NULL	NULL

5. **List all customers who live in a particular city**

select customername

from bankcustomer

where city='banglore';

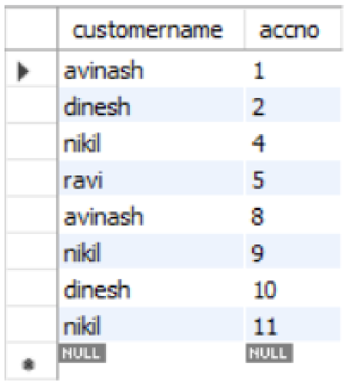


	customername
▶	avinash
	dinesh
	mohan
*	NULL

6. **List all customers with their account numbers**

select customername ,accno

from depositer ;



	customername	accno
▶	avinash	1
	dinesh	2
	nikil	4
	ravi	5
	avinash	8
	nikil	9
	dinesh	10
	nikil	11
*	NULL	NULL

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

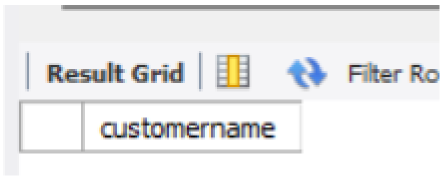
select c.customername

from bankcustomer c, depositer d, bankaccount a, branch b

where c.customername=d.customername and d.accno=a.accno and a.branchname=b.branchname and b.branchname=all(select b.branchname

from branch b

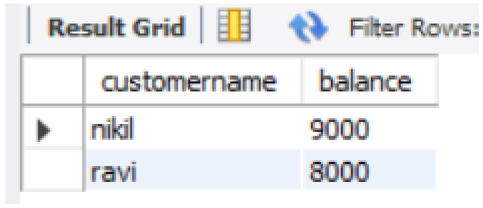
where b.branchcity='delhi');


Result Grid | Filter Ro

	customername

8. Find all customers who have accounts with a balance greater than a specified amount (5000)

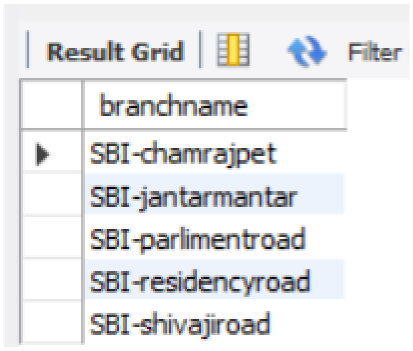
```
select c.customername, b.balance
from bankcustomer c, bankaccount b, depositer d
where d.accno=b.accno and c.customername=d.customername and  b.balance>5000;
```


Result Grid | Filter Rows:

	customername	balance
▶	nikil	9000
	ravi	8000

9. List all branch who have both a loan and an account

```
select distinct(b.branchname)
from branch b, bankaccount a, loan l
where b. branchname=a.branchname and b.branchname=l.branchname;
```


Result Grid | Filter

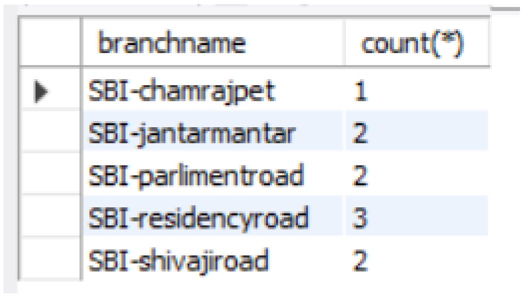
	branchname
▶	SBI-chamrajpet
	SBI-jantarmantar
	SBI-parlimentroad
	SBI-residencyroad
	SBI-shivajiroad

10. Get the number of accounts held at each branch

```
select branchname , count(*)
```

from bankaccount

group by branchname;



branchname	count(*)
SBI-chamrajpet	1
SBI-jantarmantar	2
SBI-parlimentroad	2
SBI-residencyroad	3
SBI-shivajiroad	2

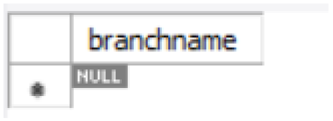
11. Find all branches that have no loans issued

select b.branchname

from branch b

where b.branchname not in(select branchname

from loan);



branchname
NULL

12. Retrieve the branch with the smallest total loan amount

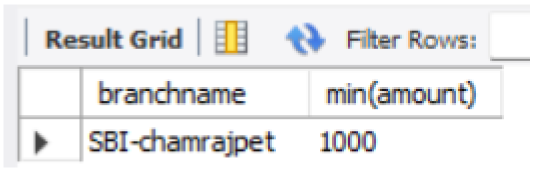
select branchname ,min(amount)

from loan

group by branchname

order by min(amount)

limit 1;



branchname	min(amount)
SBI-chamrajpet	1000

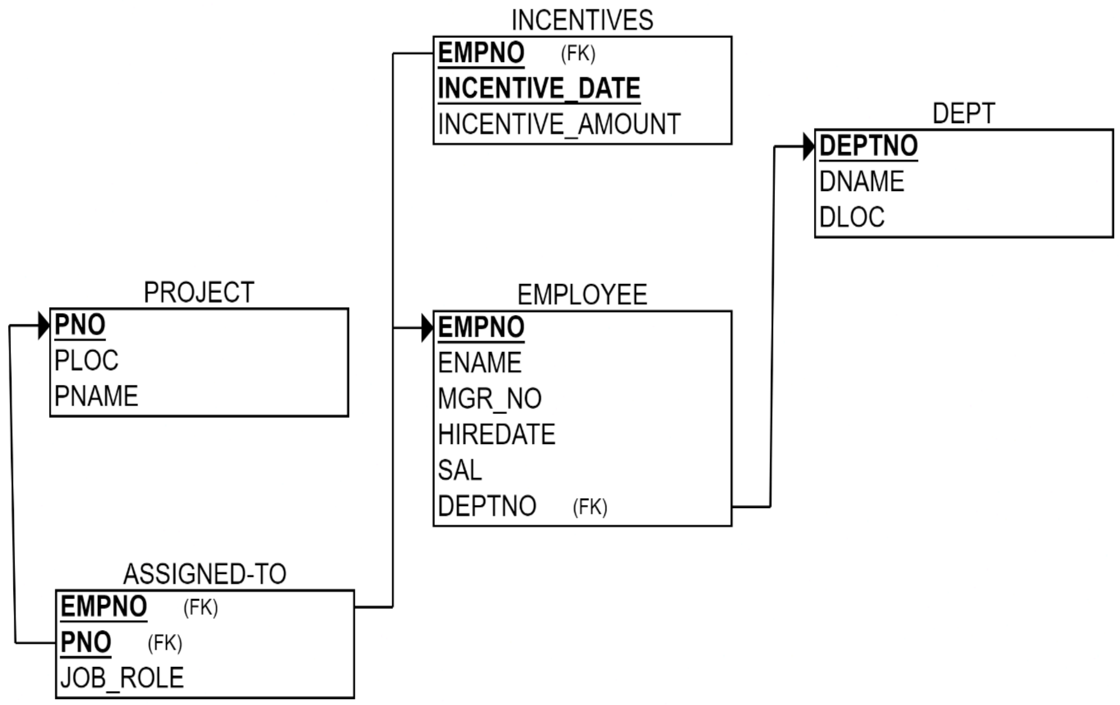
Employee Database

Question:
(Week 5&6)

Incentives (empno, incentive_date,incentive_amount)
project (pno,ploc,pname)
employee(empno,ename,mgr_no,hiredate,sal,deptno)
dept(deptno,dname,dloc)
assigned-to(empno,pno,job_role)

- List all employees along with their project details (if assigned)
- Find all employees who received incentives, along with the total incentive amount
- Retrieve the project names and locations of projects with employees assigned as 'Manager'
- List departments along with the number of employees in each department
- Find employees who have not been assigned to any project
- List all employees along with their department names and location
- Retrieve the details of employees who work under a specific manager (e.g., manager with empno = 101)
- List all projects that have employees assigned and the number of employees on each project:
- Find employees with the same manager and list their department details
- List the total number of incentives given to each employee and the sum of incentives for each:
- Retrieve all employees who have the role of 'Developer' on any project:
- Display the department-wise average salary of employees:

Schema Diagram:



Create database:

create database emp_cs065;
use emp_cs065;

Create tables:

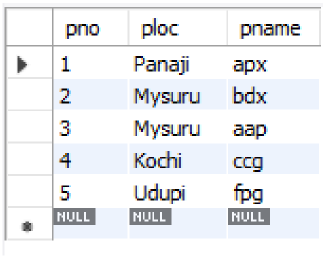
create database emp_cs065;
use emp_cs065;
create table project(
pno int,
ploc varchar(50),
pname varchar(50),
primary key (pno));

create table dept(
deptno int primary key,
dname varchar(50),
dloc varchar(50));

```
create table employee(
empno int primary key,
empname varchar(50),
mgr_no int,
hiredate date,
sal int,
deptno int,
foreign key (deptno) references dept (deptno));

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

create table assigned_to(
empno int,
pno int,
job_role varchar (50),
primary key (empno, pno),
foreign key (empno) references employee(empno),
foreign key (pno) references project (pno));
```

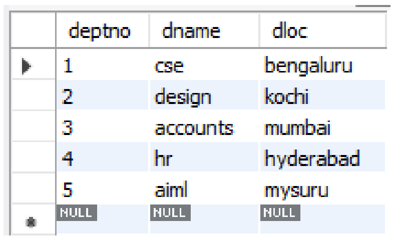
Inserting values:

```
select * from employee;
select * from project;
select * from assigned_to;
select * from incentives;
select * from dept;

insert into project
values(1,'Panaji','apx'),
(2,'Mysuru','bdx'),
(3,'Mysuru','aap'),
(4,'Kochi','ccg'),
(5,'Udupi','fpg');
```



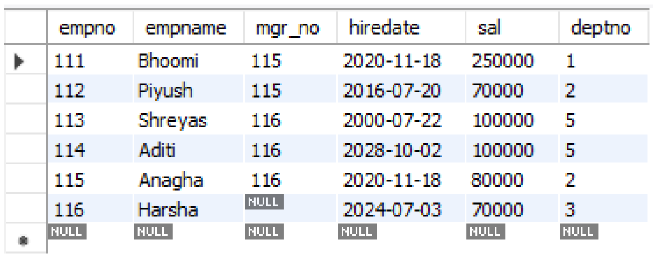
pno	ploc	pname
1	Panaji	apx
2	Mysuru	bdx
3	Mysuru	aap
4	Kochi	ccg
5	Udupi	fpg
NULL	NULL	NULL

```
insert into dept
values(1,'cse','bengaluru'),
(2,'design','kochi'),
(3,'accounts','mumbai'),
(4,'hr','hyderabad'),
(5,'aiml','mysuru');
```



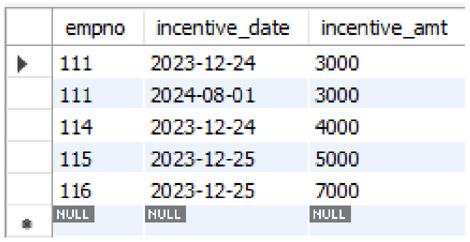
deptno	dname	dloc
1	cse	bengaluru
2	design	kochi
3	accounts	mumbai
4	hr	hyderabad
5	aiml	mysuru
NULL	NULL	NULL

```
insert into employee
values (111,'Bhoomi',115,'2020-11-18',250000,1),
(112,'Piyush',115,'2016-07-20',70000,02),
(113,'Shreyas',116,'2000-07-22',100000,05),
(114,'Aditi',116,'2028-10-02',100000,05),
(115,'Anagha',116,'2020-11-18',80000,02),
(116,'Harsha',NULL,'2024-07-03',70000,03);
```



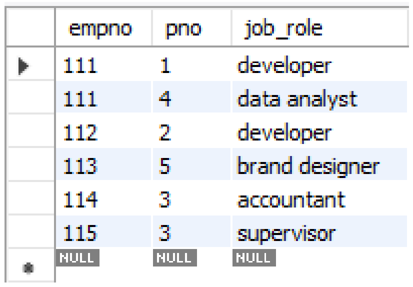
empno	empname	mgr_no	hiredate	sal	deptno
111	Bhoomi	115	2020-11-18	250000	1
112	Piyush	115	2016-07-20	70000	2
113	Shreyas	116	2000-07-22	100000	5
114	Aditi	116	2028-10-02	100000	5
115	Anagha	116	2020-11-18	80000	2
116	Harsha	NULL	2024-07-03	70000	3
NULL	NULL	NULL	NULL	NULL	NULL

```
insert into incentives
values(111,'2023-12-24',3000),
(114,'2023-12-24',4000),
(115,'2023-12-25',5000),
(116,'2023-12-25',7000),
(111,'2024-08-01',3000);
```



empno	incentive_date	incentive_amt
111	2023-12-24	3000
111	2024-08-01	3000
114	2023-12-24	4000
115	2023-12-25	5000
116	2023-12-25	7000
NULL	NULL	NULL

```
insert into assigned_to
values(111,1,'developer'),
(111,4,'data analyst'),
(112,2,'developer'),
(114, 3,'accountant'),
(113,5,'brand designer'),
(115,3,'supervisor'),
(112,3,'manager');
```

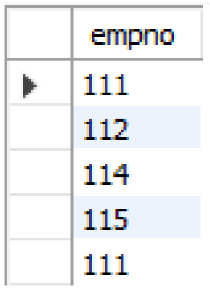


empno	pno	job_role
111	1	developer
111	4	data analyst
112	2	developer
113	5	brand designer
114	3	accountant
115	3	supervisor
NULL	NULL	NULL

Queries

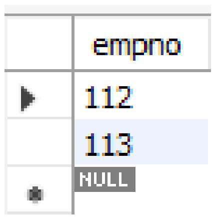
List all employees along with their project details (if assigned)

```
select e.empno
from employee e, assigned_to a
where e.empno=a.empno and a.pno in(select pno
                            from project
  where ploc in ('Panaji' ,'Kochi','Mysuru'));
```



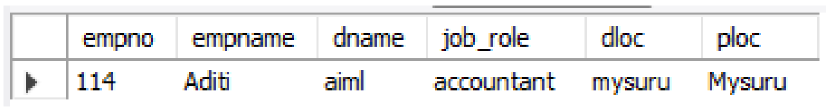
empno
111
112
114
115
111

```
select empno
from employee
where not exists(select 1
              from incentives
              where empno=employee.empno);
```



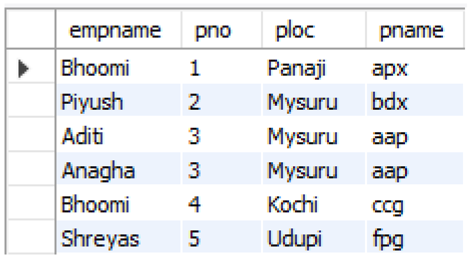
empno
112
113
NULL

select e.empno, e.empname, d.dname, a.job_role, d.dloc ,p.ploc
from employee e, project p, assigned_to a, dept d
where e.empno=a.empno and p.pno=a.pno and e.deptno=d.deptno and d.dloc=p.ploc;



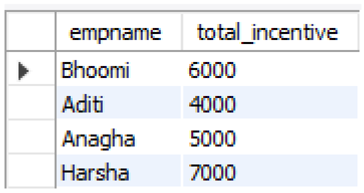
empno	empname	dname	job_role	dloc	ploc
114	Aditi	aiml	accountant	mysuru	Mysuru

select e.empname, p.*
from employee e, project p, assigned_to a
where a.empno = e.empno and a.pno = p.pno;



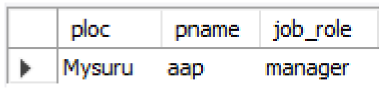
empname	pno	ploc	pname
Bhoomi	1	Panaji	apx
Piyush	2	Mysuru	bdx
Aditi	3	Mysuru	aap
Anagha	3	Mysuru	aap
Bhoomi	4	Kochi	ccg
Shreyas	5	Udupi	fpg

select e.empname, sum(i.incentive_amt) as total_incentive
from employee e, incentives i
where e.empno = i.empno
group by e.empname;



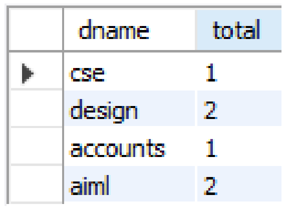
empname	total_incentive
Bhoomi	6000
Aditi	4000
Anagha	5000
Harsha	7000

select p.ploc, p.pname, a.job_role
from project p, assigned_to a
where p.pno = a.pno and a.job_role = "manager";



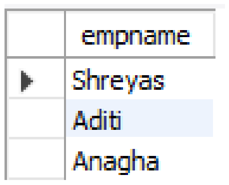
ploc	pname	job_role
Mysuru	aap	manager

```
select d.dname, count(e.empno) as total
from dept d, employee e
where d.deptno = e.deptno
group by d.dname;
```



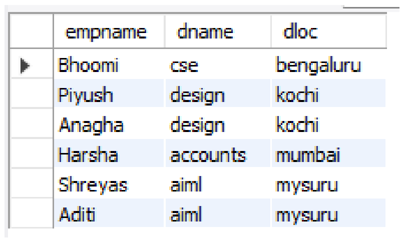
dname	total
cse	1
design	2
accounts	1
aiml	2

```
select empname
from employee
where not exists(select 1
              from assigned_to
              where empno=employee.empno);
```



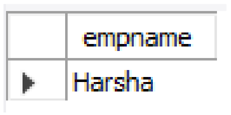
empname
Shreyas
Aditi
Anagha

```
select e.empname, d.dname, d.dloc
from employee e, dept d
where e.deptno = d.deptno;
```



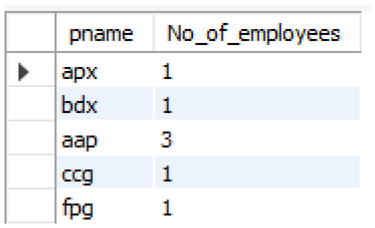
empname	dname	dloc
Bhoomi	cse	bengaluru
Piyush	design	kochi
Anagha	design	kochi
Harsha	accounts	mumbai
Shreyas	aiml	mysuru
Aditi	aiml	mysuru

```
select e.empname
from employee e
where mgr_no = 116;
```



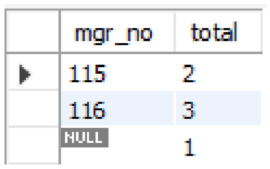
empname
Harsha

```
select p.pname, count(a.empno) as No_of_employees
from project p, assigned_to a
where a.pno = p.pno
group by p.pname;
```



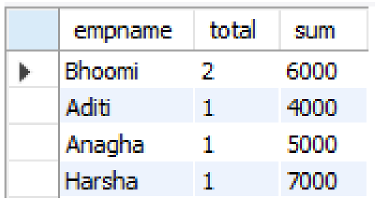
pname	No_of_employees
apx	1
bdx	1
aap	3
ccg	1
fpg	1

```
select e.mgr_no, count(e.empno) as total
from employee e
group by e.mgr_no;
```



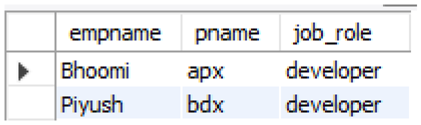
mgr_no	total
115	2
116	3
NULL	1

```
select e.empname, count(i.empno) as total, sum(i.incentive_amt) as sum
from employee e, incentives i
where e.empno = i.empno
group by e.empname;
```



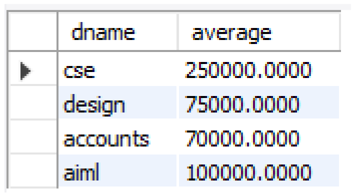
empname	total	sum
Bhoomi	2	6000
Aditi	1	4000
Anagha	1	5000
Harsha	1	7000

```
select e.empname, p.pname, a.job_role
from employee e, project p, assigned_to a
where e.empno = a.empno and p.pno = a.pno and a.job_role = "developer";
```



empname	pname	job_role
Bhoomi	apx	developer
Piyush	bdx	developer

```
select d.dname, avg(e.sal) as average
from employee e, dept d
where e.deptno = d.deptno
group by d.dname;
```



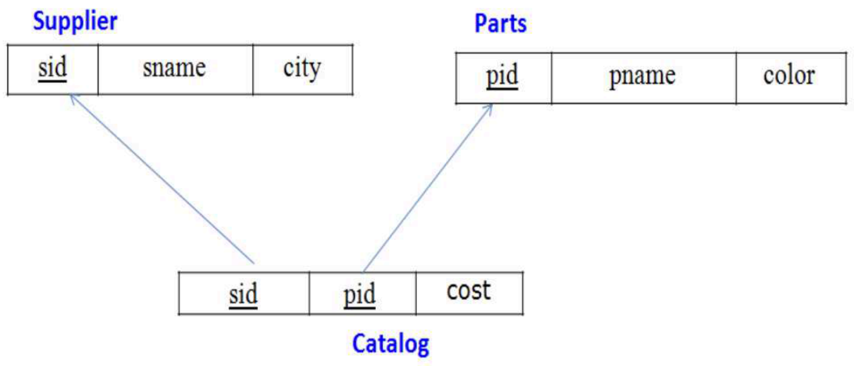
	dname	average
▶	cse	250000.0000
	design	75000.0000
	accounts	70000.0000
	aiml	100000.0000

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:



Create Database:

```
create database supp;
 use supp;
```

Create Tables:

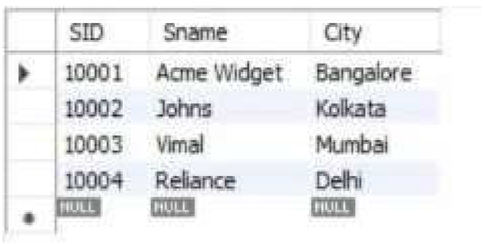
```
create table Supplier(
 s_id int primary key,
s_name varchar(30),
city varchar(20));
        create table Parts( p_id int primary key, p_name varchar(30), color varchar(30));
        create table Catalog( s_id int,
        p_id int, cost float,
        foreign key(s_id) references Supplier(s_id), foreign key(p_id) references Parts(p_id));
```

Structure of the Table:

```
desc Supplier;
desc Parts;
desc Catalog;
```

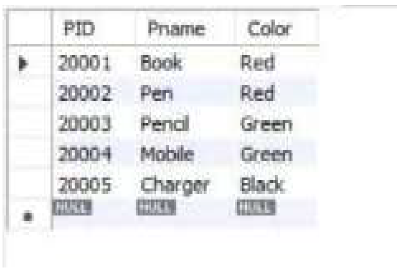
Inserting Values to the tables:

```
insert into Supplier values
(10001, 'Acme_Widget', 'Bangalore'),
(10002, 'Johns', 'Kolkata'),
(10003, 'Vimal', 'Mumbai'),
(10004, 'Reliance', 'Delhi'); select * from Supplier;
```



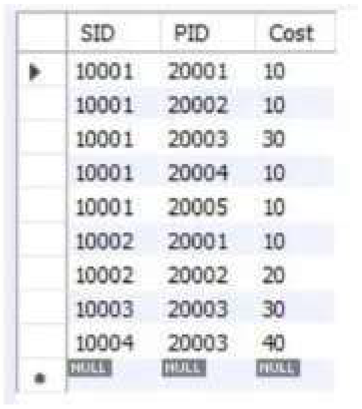
SID	Sname	City
10001	Acme Widget	Bangalore
10002	Johns	Kolkata
10003	Vimal	Mumbai
10004	Reliance	Delhi
NULL	NULL	NULL

```
 insert into Parts values (20001, 'Book', 'Red'),
(20002, 'Pen', 'Red'),
(20003, 'Pencil', 'Green'),
(20004, 'Mobile', 'Green'),
(20005, 'Charger', 'Black');
```



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);
```

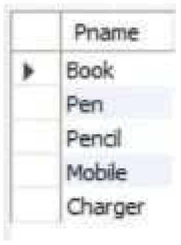


SID	PID	Cost
10001	20001	10
10001	20002	10
10001	20003	30
10001	20004	10
10001	20005	10
10002	20001	10
10002	20002	20
10003	20003	30
10004	20003	40
NULL	NULL	NULL

Queries:

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

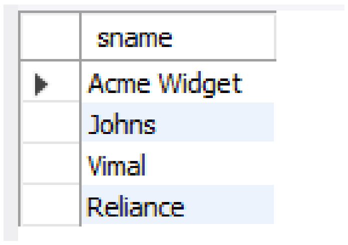
```
select distinct p.p_name
from Supplier s, Catalog c, Parts p where s.s_id = c.s_id and
p.p_id = c.p_id and c.s_id is not null;
```



Pname
Book
Pen
Pencil
Mobile
Charger

Find the snames of suppliers who supply every part.

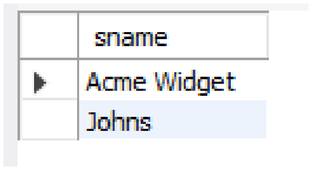
```
select distinct s_name
from Supplier s, Catalog c, Parts p where s.s_id = c.s_id
group by s.s_id, s.s_name
having count(distinct c.p_id)=(select count(*) from Parts p);
```



sname
Acme Widget
Johns
Vimal
Reliance

Find the snames of suppliers who supply every red part.

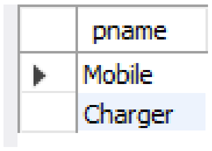
```
select distinct s_name
from Supplier s, Catalog c, Parts p where s.s_id = c.s_id and
c.p_id in (select p_id from Parts p where p.color = 'Red')
```



sname
Acme Widget
Johns

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

```
select distinct p_name from Supplier s, Parts p, Catalog c where p.p_id in (select c.p_id from
Catalog c, Supplier s where
s.s_id = c.s_id and s.s_name = 'Acme_Widget') and
p.p_id not in (select c.p_id from Catalog c, Supplier s where s.s_id = c.s_id and s.s_name !=
'Acme_Widget');
```

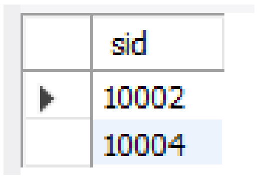


pname
Mobile
Charger

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)

```
create view Average(p_id, Average_Product_Cost) as select c.p_id, avg(cost)
from Catalog c group by c.p_id;
 select c.s_id from Catalog c, Average a where c.p_id = a.p_id and
```

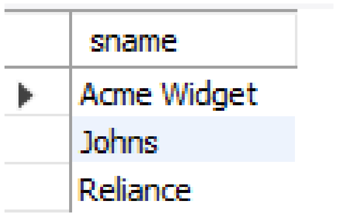
c.cost>(a.Average_Product_Cost)
group by c.p_id, c.s_id;



sid
10002
10004

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

select distinct s.s_name, c.cost, c.p_id from Catalog c, Supplier s where s.s_id = c.s_id and c.cost in (select max(cost) from Catalog c group by c.p_id);



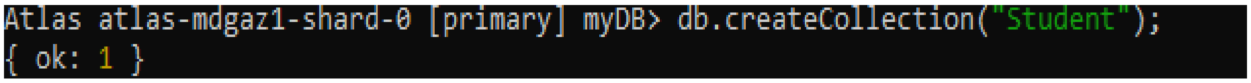
sname
Acme Widget
Johns
Reliance

NoSQL - 1 - Student Database

Question:
(Week 8)

1. Create a database "Student" with the following attributesRollno, Age, ContactNo, Email-Id.

db.createCollection("Student");

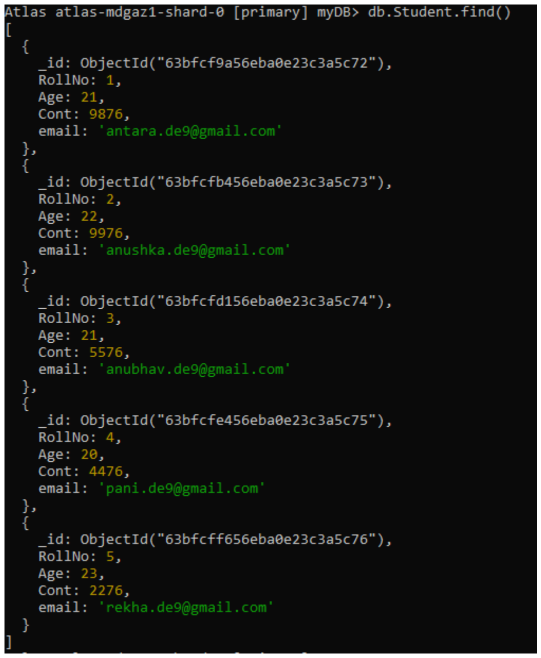


```
Atlas atlas-mdgaz1-shard-0 [primary] myDB> db.createCollection("Student");
{ ok: 1 }
```

2. Insert appropriate values

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

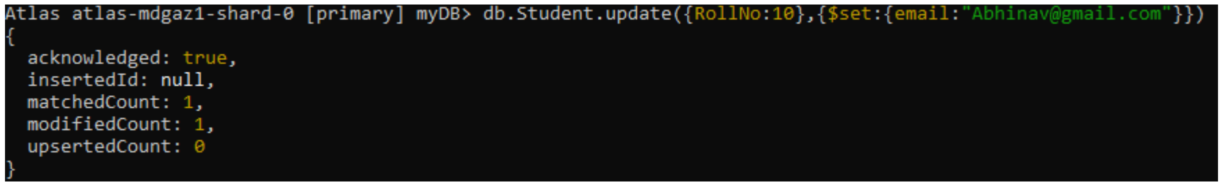
```
db.Student.insert({RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"});
db.Student.insert({RollNo:3,Age:21,Cont:5576,email:"anubhav.de9@gmail.com"});
db.Student.insert({RollNo:4,Age:20,Cont:4476,email:"pani.de9@gmail.com"});
db.Student.insert({RollNo:10,Age:23,Cont:2276,email:"rekha.de9@gmail.com"});
```



```
Atlas atlas-mdgaz1-shard-0 [primary] myDB> db.Student.find()
[
  {
    _id: ObjectId("63bfcf9a56eba0e23c3a5c72"),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId("63bfcfb456eba0e23c3a5c73"),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de9@gmail.com'
  },
  {
    _id: ObjectId("63bfcfd156eba0e23c3a5c74"),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
  },
  {
    _id: ObjectId("63bfcfe456eba0e23c3a5c75"),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId("63bfcff656eba0e23c3a5c76"),
    RollNo: 5,
    Age: 23,
    Cont: 2276,
    email: 'rekha.de9@gmail.com'
  }
]
```

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

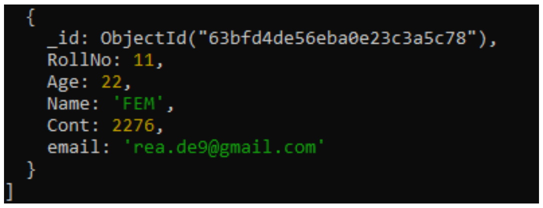
```
db.Student.update({RollNo:10},{$set:{email:"Abhinav@gmail.com"}})
```



```
Atlas atlas-mdgaz1-shard-0 [primary] myDB> db.Student.update({RollNo:10},{$set:{email:"Abhinav@gmail.com"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

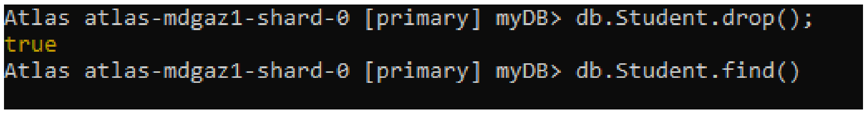
4. Replace the student name from "ABC" to "FEM" of rollno 11.

```
db.Student.insert({RollNo:11,Age:22,Name:
"ABC",Cont:2276,email:"rea.de9@gmail.com"});
db.Student.update({RollNo:11,Name:"ABC"},{$set:{Name:"FEM"}});
```


```
  {
    _id: ObjectId("63bfd4de56eba0e23c3a5c78"),
    RollNo: 11,
    Age: 22,
    Name: 'FEM',
    Cont: 2276,
    email: 'rea.de9@gmail.com'
  }
]
```

5. Drop the table

db.Student.drop();

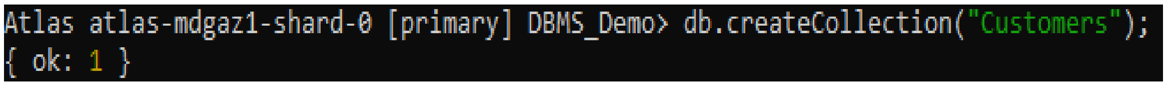

```
Atlas atlas-mdgaz1-shard-0 [primary] myDB> db.Student.drop();
true
Atlas atlas-mdgaz1-shard-0 [primary] myDB> db.Student.find()
```

NoSQL - 2 - Customer Database

Question:
(Week 9)

1. Create a collection by name Customers with the following attributes: Cust_id, Acc_Bal, Acc_Type

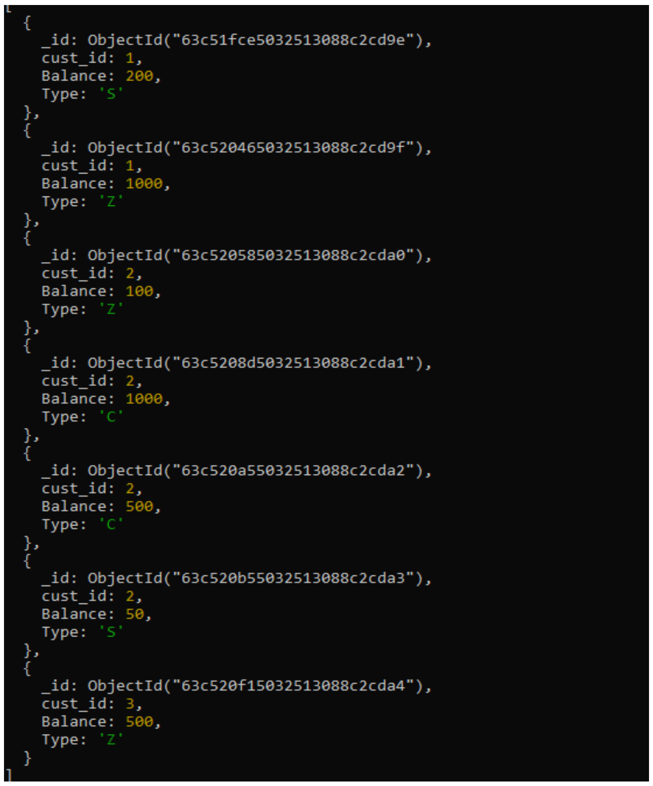
db.createCollection("Customers");


```
Atlas atlas-mdgaz1-shard-0 [primary] DBMS_Demo> db.createCollection("Customers");
{ ok: 1 }
```

2. Insert at least 5 values into the table

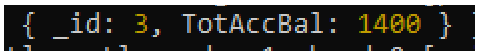
db.Customers.insert({cust_id:1,Balance:200, Type:"S"});
db.Customers.insert({cust_id:1,Balance:1000, Type:"Z"})
db.Customers.insert({cust_id:2,Balance:100, Type:"Z"});
db.Customers.insert({cust_id:2,Balance:1000, Type:"C"});
db.Customers.insert({cust_id:2,Balance:500, Type:"C"});
db.Customers.insert({cust_id:2,Balance:50, Type:"S"});

db.Customers.insert({cust_id:3,Balance:500, Type:"Z"});


```
[
  {
    _id: ObjectId("63c51fce5032513088c2cd9e"),
    cust_id: 1,
    Balance: 200,
    Type: 'S'
  },
  {
    _id: ObjectId("63c520465032513088c2cd9f"),
    cust_id: 1,
    Balance: 1000,
    Type: 'Z'
  },
  {
    _id: ObjectId("63c520585032513088c2cda0"),
    cust_id: 2,
    Balance: 100,
    Type: 'Z'
  },
  {
    _id: ObjectId("63c5208d5032513088c2cda1"),
    cust_id: 2,
    Balance: 1000,
    Type: 'C'
  },
  {
    _id: ObjectId("63c520a55032513088c2cda2"),
    cust_id: 2,
    Balance: 500,
    Type: 'C'
  },
  {
    _id: ObjectId("63c520b55032513088c2cda3"),
    cust_id: 2,
    Balance: 50,
    Type: 'S'
  },
  {
    _id: ObjectId("63c520f15032513088c2cda4"),
    cust_id: 3,
    Balance: 500,
    Type: 'Z'
  }
]
```

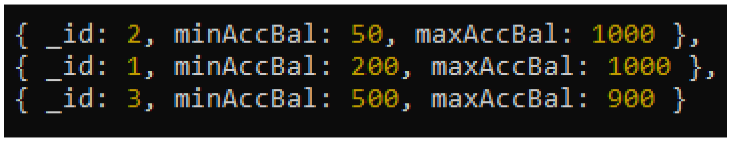
3. Write a query to display those records whose total account balance is greater than 1200 of account type 'Z' for each customer_id.

db.Customers.aggregate ({$match:{Type:"Z"}},
{$group : { _id : "$cust_id",
TotAccBal :{$sum:"$Balance"} } }, {$match:{TotAccBal:{$gt:1200}}});


```
{ _id: 3, TotAccBal: 1400 }
```

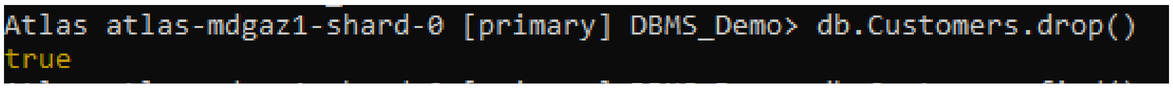
4. Determine Minimum and Maximum account balance for each customer_id.

db.Customers.aggregate (
{$group : { _id : "$cust_id",
minAccBal :{$min:"$Balance"}, maxAccBal :{$max:"$Balance"} }});


```
{ _id: 2, minAccBal: 50, maxAccBal: 1000 },
{ _id: 1, minAccBal: 200, maxAccBal: 1000 },
{ _id: 3, minAccBal: 500, maxAccBal: 900 }
```

5. Drop the table

db.Customers.drop()



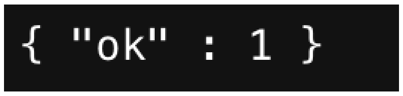
```
Atlas atlas-mdgaz1-shard-0 [primary] DBMS_Demo> db.Customers.drop()
true
```

NoSQL - 3 - Restaurant Database

Question:
(Week 10)

1. Write a MongoDB query to display all the documents in the collection restaurants.

db.createCollection("restaurants");

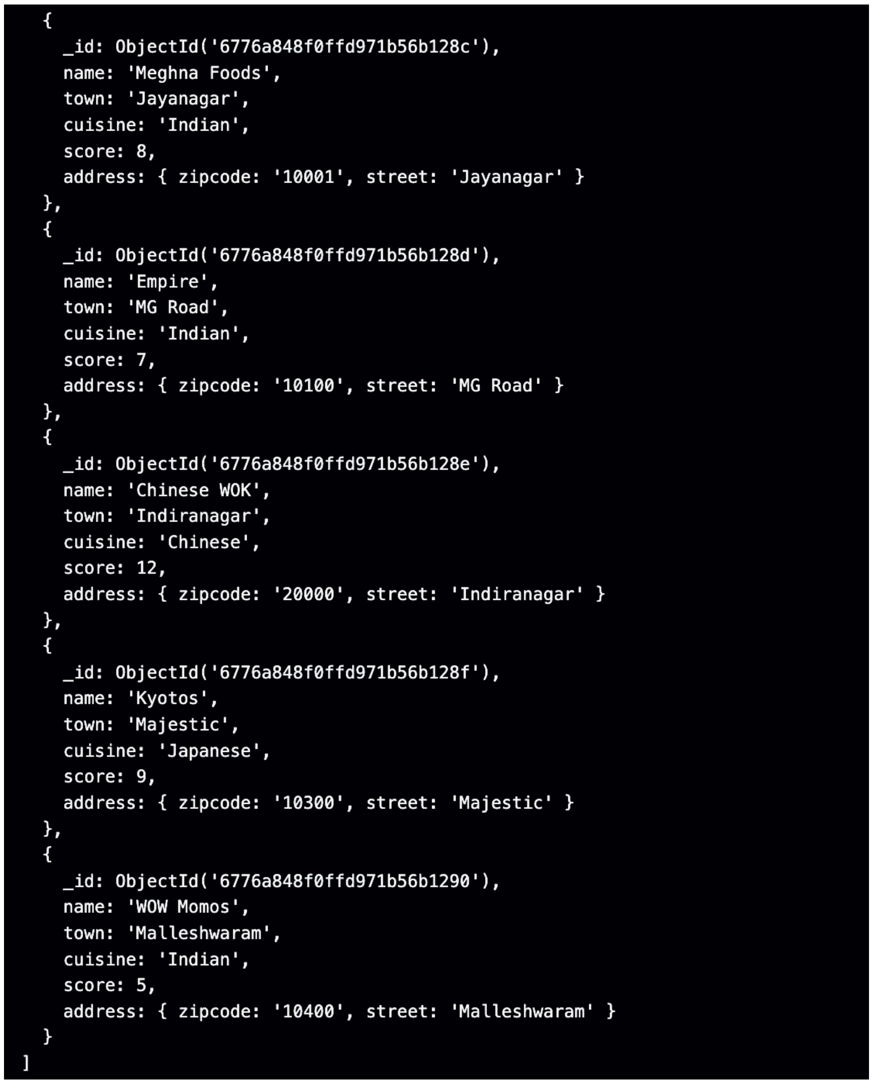


```
{ "ok" : 1 }
```

2. Write a MongoDB query to arrange the name of the restaurants in descending order along with all the columns.

db.restaurants.insertMany([
{ name: "Meghna Foods", town: "Jayanagar", cuisine: "Indian", score: 8, address: { zipcode: "10001", street: "Jayanagar“
} },
{ name: "Empire", town: "MG Road", cuisine: "Indian", score: 7, address: { zipcode: "10100", street: "MG Road" } },
{ name: "Chinese WOK", town: "Indiranagar", cuisine: "Chinese", score: 12, address:

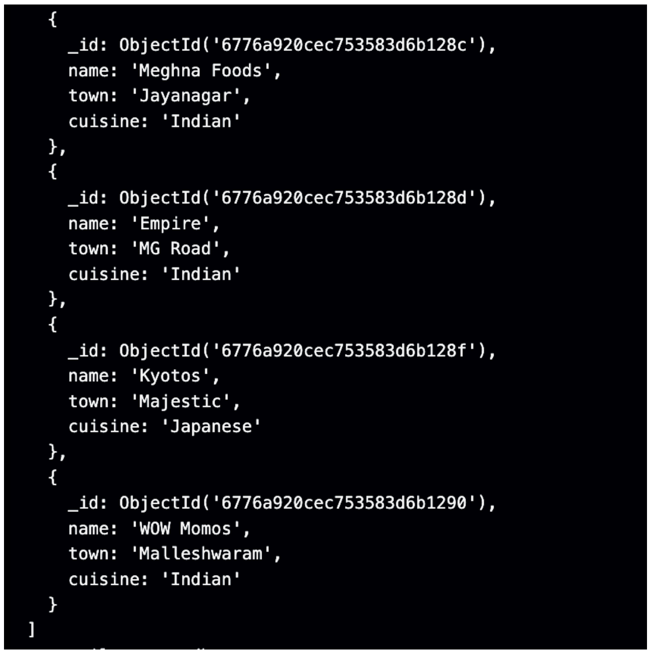
{ zipcode: "20000", street: "Indiranagar" } },
{ name: "Kyotos", town: "Majestic", cuisine: "Japanese", score: 9, address: { zipcode: "10300", street: "Majestic" } },
{ name: "WOW Momos", town: "Malleshwaram", cuisine: "Indian", score: 5, address: { zipcode: "10400", street: "Malleshwaram" }
}])
db.restaurants.find({})



```
  {
    _id: ObjectId('6776a848f0ffd971b56b128c'),
    name: 'Meghna Foods',
    town: 'Jayanagar',
    cuisine: 'Indian',
    score: 8,
    address: { zipcode: '10001', street: 'Jayanagar' }
  },
  {
    _id: ObjectId('6776a848f0ffd971b56b128d'),
    name: 'Empire',
    town: 'MG Road',
    cuisine: 'Indian',
    score: 7,
    address: { zipcode: '10100', street: 'MG Road' }
  },
  {
    _id: ObjectId('6776a848f0ffd971b56b128e'),
    name: 'Chinese WOK',
    town: 'Indiranagar',
    cuisine: 'Chinese',
    score: 12,
    address: { zipcode: '20000', street: 'Indiranagar' }
  },
  {
    _id: ObjectId('6776a848f0ffd971b56b128f'),
    name: 'Kyotos',
    town: 'Majestic',
    cuisine: 'Japanese',
    score: 9,
    address: { zipcode: '10300', street: 'Majestic' }
  },
  {
    _id: ObjectId('6776a848f0ffd971b56b1290'),
    name: 'WOW Momos',
    town: 'Malleshwaram',
    cuisine: 'Indian',
    score: 5,
    address: { zipcode: '10400', street: 'Malleshwaram' }
  }
]
```

3. Write a MongoDB query to find the restaurant Id, name, town and cuisine for those restaurants which achieved a score which is not more than 10.

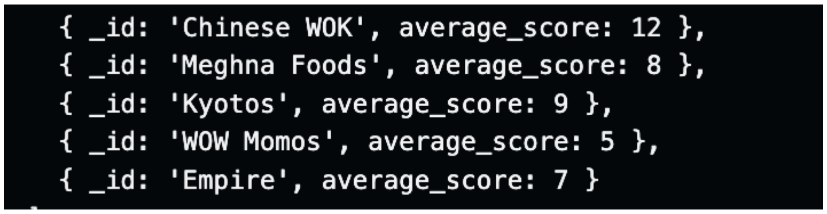
db.restaurants.find({ "score": { $lte: 10 } }, { _id: 1, name: 1, town: 1, cuisine: 1 })



```
  {
    _id: ObjectId('6776a920cec753583d6b128c'),
    name: 'Meghna Foods',
    town: 'Jayanagar',
    cuisine: 'Indian'
  },
  {
    _id: ObjectId('6776a920cec753583d6b128d'),
    name: 'Empire',
    town: 'MG Road',
    cuisine: 'Indian'
  },
  {
    _id: ObjectId('6776a920cec753583d6b128f'),
    name: 'Kyotos',
    town: 'Majestic',
    cuisine: 'Japanese'
  },
  {
    _id: ObjectId('6776a920cec753583d6b1290'),
    name: 'WOW Momos',
    town: 'Malleshwaram',
    cuisine: 'Indian'
  }
]
```

4. Write a MongoDB query to find the average score for each restaurant.

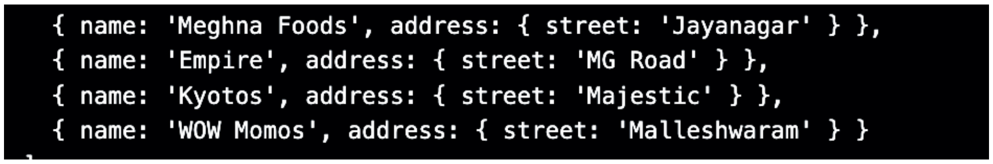
db.restaurants.aggregate([{ $group: { _id: "$name", average_score: { $avg: "$score" } } }])



```
  { _id: 'Chinese WOK', average_score: 12 },
  { _id: 'Meghna Foods', average_score: 8 },
  { _id: 'Kyotos', average_score: 9 },
  { _id: 'WOW Momos', average_score: 5 },
  { _id: 'Empire', average_score: 7 }
```

5. Write a MongoDB query to find the name and address of the restaurants that have a zipcode that starts with 10

db.restaurants.find({ "address.zipcode": /^10/ }, { name: 1, "address.street": 1, _id: 0 })



```
  { name: 'Meghna Foods', address: { street: 'Jayanagar' } },
  { name: 'Empire', address: { street: 'MG Road' } },
  { name: 'Kyotos', address: { street: 'Majestic' } },
  { name: 'WOW Momos', address: { street: 'Malleshwaram' } }
```