

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

Database Management Systems (22CS3PCDBM)

Submitted by

Tanisha Gotadke (1BM21CS229)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

October-2022 to Feb-2023

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled “Database Management Systems (22CS3PCDBM)” carried out by **Tanisha Gotadke (1BM21CS229)**, who is bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a Database Management Systems (22CS3PCDBM) work prescribed for the said degree.

Dr. Pallavi G B
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

Index

SL. No.	Date	Experiment Title	Page No.
1	09/11/2022	INSURANCE DATABSE	4
2	16/11/2022	MORE QUERIES ON INSURANCE DATABSE	9
3	23/11/2022	BANK DATABASE	13
4	30/11/2022	MORE QUERIES ON BANK DATABASE	21
5	06/12/2022	EMPLOYEE DATABASE	31
6	14/12/2022	MORE QUERIES ON EMPLOYEE DATABASE	39
7	21/12/2022	SUPPLIER DATABASE	48
8	28/12/2022	AIRLINE FLIGHT DATABASE	57
9	11/01/2023	NO SQL - STUDENT DATABASE	66

1. Insurance Database

Create database

```
show databases;
```

```
create database insurance;
```

```
use insurance;
```

Create table

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

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

```
create table person (driver_id varchar(10),name varchar(20), address varchar(30), primary  
key(driver_id));
```

```
create table owns(driver_id varchar(10),reg_num varchar(10),
```

```
primary key(driver_id, reg_num),
```

```
foreign key(driver_id) references person(driver_id)on delete cascade,
```

```
foreign key(reg_num) references car(reg_num)on delete cascade);
```

```
create table participated(driver_id varchar(10),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)on delete cascade,
```

```
foreign key(reg_num) references car(reg_num)on delete cascade,
```

```
foreign key(report_num) references accident(report_num)on delete cascade);
```

```
show tables;
```

Structure of the Table:

show tables;

desc

accident;

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

desc car;

Field	Type	Null	Key	Default	Extra
reg_no	char(10)	NO	PRI	NULL	
model	varchar(15)	YES		NULL	
year	year	YES		NULL	

desc owns;

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

desc participated;

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

desc person;

Field	Type	Null	Key	Default	Extra
driver_id	char(10)	NO	PRI	NULL	
name	varchar(20)	YES		NULL	
address	varchar(20)	YES		NULL	

Inserting Values to the table:

insert into accident values ('11', '2003-01-01','Mysore Road'), (12,'2004-02-02','South end Circle'), (13,'2003-01-21','Bull temple Road'), (14,'2008-02-17','Mysore Road'), (15,'2004-03-05','Kanakpura Road');select * from accident;

report_no	accident_date	location
11	2003-01-01	Mysore Road
12	2004-02-02	South end Circle
13	2003-01-21	Bull temple Road
14	2008-02-17	Mysore Roa
15	2004-03-05	Kanakpura Road
16	2004-03-05	Kanakpura Road
NULL	NULL	NULL

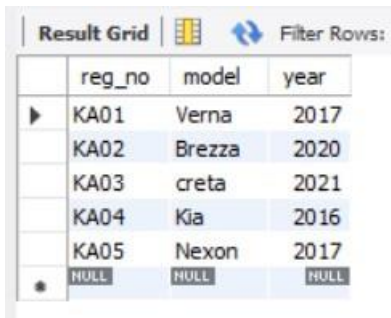
insert into person values ('001','Vignesh','Kundpura'), ('002','Sushanth','Hebri'), ('003','Subrahmanya','Bengaluru'), ('004','Kenny','Gulbarga'), ('005','Varun','Bhatkal');

driver_id	name	address
002	Sushanth	Hebri
003	Subrahmanya	Bengaluru
004	Kenny	Gulbarga
005	Varun	Bhatkal
NULL	NULL	NULL

select * from person;

insert into car values ('KA01','Verna','2017'), ('KA02','Brezza','2020'), ('KA03','creta','2021'),('KA04','Kia','2016'),('KA05','Nexon','2017');

select * from car;



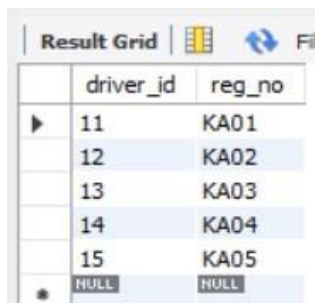
	reg_no	model	year
▶	KA01	Verna	2017
	KA02	Brezza	2020
	KA03	creta	2021
	KA04	Kia	2016
	KA05	Nexon	2017
*	NULL	NULL	NULL

set foreign_key_checks=0;

set global foreign_key_checks=0;

insert into owns values ('11','KA01'), (12,'KA02'), (13,'KA03'), (14,'KA04'), (15,'KA05');

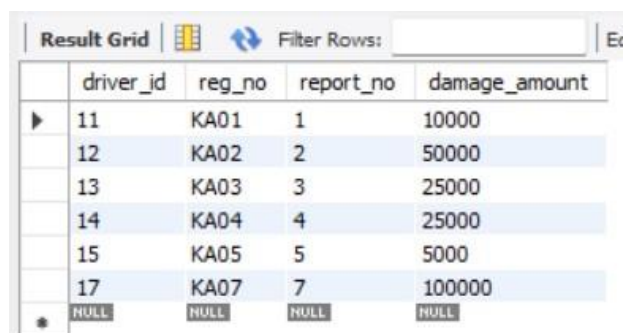
select * from Owns;



	driver_id	reg_no
▶	11	KA01
	12	KA02
	13	KA03
	14	KA04
	15	KA05
*	NULL	NULL

insert into participated values ('11','KA01','001','10000'), (12,'KA02','002','50000'),
(13,'KA03','003','25000'), (14,'KA04','004','3000'),
(15,'KA05','005','5000'),('17','KA07','007','100000');

select * from Participated;

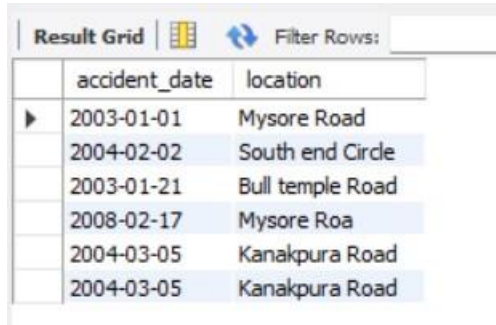


	driver_id	reg_no	report_no	damage_amount
▶	11	KA01	1	10000
	12	KA02	2	50000
	13	KA03	3	25000
	14	KA04	4	25000
	15	KA05	5	5000
	17	KA07	7	100000
*	NULL	NULL	NULL	NULL

QUERIES:

- **Display Accident date and location**

select accident_date,location from accident;



The screenshot shows a 'Result Grid' with a 'Filter Rows' button. The grid contains the following data:

	accident_date	location
▶	2003-01-01	Mysore Road
	2004-02-02	South end Circle
	2003-01-21	Bull temple Road
	2008-02-17	Mysore Roa
	2004-03-05	Kanakpura Road
	2004-03-05	Kanakpura Road

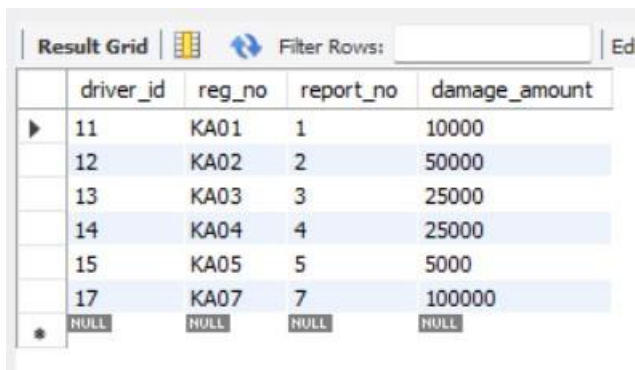
- **Update the damage amount to 25000 for the car with a specific reg_num (example 'K A053408') for which the accident report number was 12.**

update participated

set damage_amount='25000'

where reg_no='KA04'; select

* from participated;



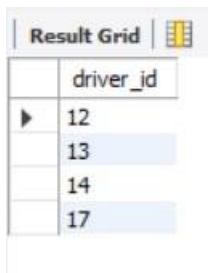
The screenshot shows a 'Result Grid' with a 'Filter Rows' button and an 'Edit' button. The grid contains the following data:

	driver_id	reg_no	report_no	damage_amount
▶	11	KA01	1	10000
	12	KA02	2	50000
	13	KA03	3	25000
	14	KA04	4	25000
	15	KA05	5	5000
	17	KA07	7	100000
*	NULL	NULL	NULL	NULL

- **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;



The screenshot shows a 'Result Grid' with a 'Filter Rows' button. The grid contains the following data:

	driver_id
▶	12
	13
	14
	17

2. More Queries on Insurance Database

Question

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

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

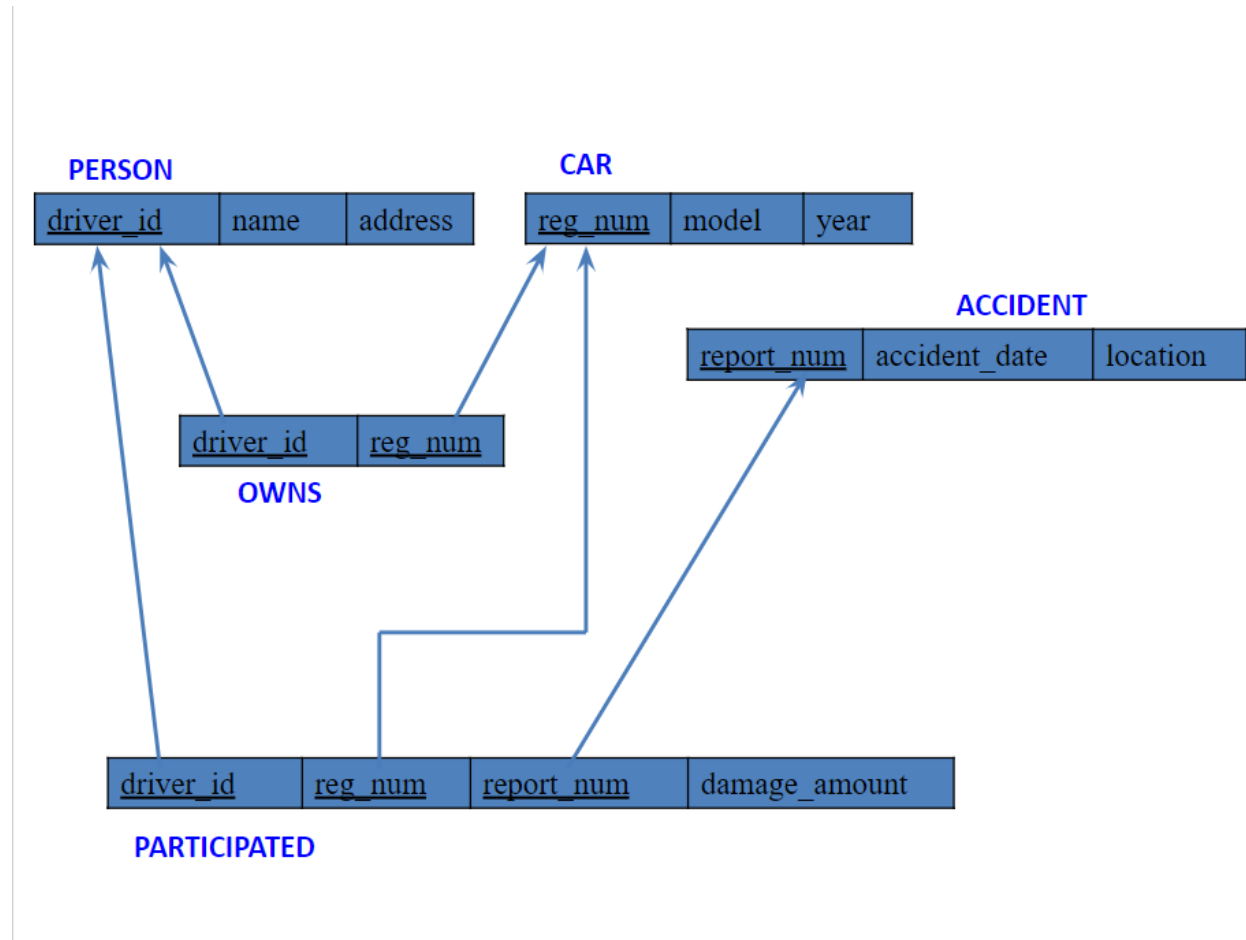
ACCIDENT (report_num: int, accident_date: date, location: String)

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 as done in previous week's lab and Enter at least five tuples for each relationEnter at least five tuples for each relation
- Enter at least five tuples for each relation
- Display the entire CAR relation in the ascending order of manufacturing year.
- Find the number of accidents in which cars belonging to a specific model (example 'Lancer') were involved.
- Find the total number of people who owned cars that involved in accidents in 2008.
- LIST THE ENTIRE PARTICIPATED RELATION IN THE DESCENDING ORDER OF DAMAGE AMOUNT.
- FIND THE AVERAGE DAMAGE AMOUNT
- DELETE THE TUPLE WHOSE DAMAGE AMOUNT IS BELOW THE AVERAGE DAMAGE AMOUNT
- LIST THE NAME OF DRIVERS WHOSE DAMAGE IS GREATER THAN THE AVERAGE DAMAGE AMOUNT.
- FIND MAXIMUM DAMAGE AMOUNT.

Schema Diagram



Queries

Q.>select * from car order by year asc;

KA01AB1234	BMW	2010
KA01AB1314	PORSCHE	2010
KA01AB1516	JAGUAR	2010
KA01AB5678	MERCEDES	2010
KA01BA5678	AUDI	2010

Q.> select count(*) from participated p, car c where c.reg_num=p.reg_num and c.model='BMW';

1

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

1

Q.>select * from participated order by damage_amount desc;

IND11123KA KA01BA5678 13 200000

IND09876KA KA01AB1516 15 175000

IND67891KA KA01AB5678 12 150000

IND13145KA KA01AB1314 14 120000

IND12345KA KA01AB1234 11 100000

Q.> select avg(damage_amount) average from participated;

149000.0000

Q.> delete from participated where damage_amount<(select t.damount from (select
avg(damage_amount) as damount from participated) t);

IND09876KA KA01AB1516 15 175000

IND11123KA KA01BA5678 13 200000

IND67891KA KA01AB5678 12 150000

Q.> select p.name from person p, participated pa where pa.driver_id=p.driver_id and
pa.damage_amount>(select avg(damage_amount) average from participated);

jake

zayd

krsna

Q.>select max(damage_amount) from participated;

200000

Q.>select count(*) from accident a, participated p where a.report_num=p.report_num and
a.accident_date like '2004%';

2

3.Bank Database

Question

1. Create the above tables by properly specifying the primary keys and the foreign keys.
2. Enter at least five tuples for each relation.
3. Display the branch name and assets from all branches in lakhs of rupees and rename the assets column to 'assets in lakhs'.
4. Find all the customers who have at least two accounts at the same branch (ex. SBI_ResidencyRoad).
5. CREATE A VIEW WHICH GIVES EACH BRANCH THE SUM OF THE AMOUNT OF ALL THE LOANS AT THE BRANCH.

Schema Diagram

Branch

<u>Branch name</u>	<u>Branchcity</u>	Assets
------------------------	-------------------	--------

BankAccount

<u>Accno</u>	Branchname	Balance
--------------	------------	---------

<u>Customername</u>	<u>Accno</u>
---------------------	--------------

Depositer

BankCustomer

<u>Customer name</u>	Customer-str eet	City
--------------------------	---------------------	------

<u>Loan number</u>	Branchname	Amount
------------------------	------------	--------

LOAN

Create database

```
create database bankdatabase;
```

```
use bankdatabase;
```

Create table

```
create table branch(branch_name varchar(25), branch_city varchar(25), assets float);
```

```
alter table branch modify branch_name varchar(25) primary key;
```

```
create table bankAccount(
```

```
accno int,
```

```
branch_name varchar(20),  
balance real,  
PRIMARY KEY(accno),  
FOREIGN KEY(branch_name) REFERENCES branch(branch_name)  
ON UPDATE CASCADE ON DELETE CASCADE  
);
```

```
create table bankCustomer(  
customer_name varchar(20),  
customer_street varchar(20),  
customer_city varchar(20),  
PRIMARY KEY(customer_name)  
);
```

```
create table depositer(  
customer_name varchar(20),  
accno int,  
FOREIGN KEY(customer_name)REFERENCES bankCustomer(customer_name)  
ON UPDATE CASCADE ON DELETE CASCADE,  
FOREIGN KEY(accno) REFERENCES bankAccount(accno)  
ON UPDATE CASCADE ON DELETE CASCADE  
);
```

```
create table loan(  
loan_no int,  
branch_name varchar(20),  
amount real,
```

```
PRIMARY KEY(loan_no),  
FOREIGN KEY(branch_name) REFERENCES branch(branch_name)  
ON UPDATE CASCADE ON DELETE CASCADE  
);
```

Structure of the table

```
[mysql> show tables;  
+-----+  
| Tables_in_bankdatabase |  
+-----+  
| bankAccount  
| bankCustomer  
| branch  
| depositer  
| loan  
+-----+  
5 rows in set (0.00 sec)  
  
mysql> █
```

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','mumbai',20000);  
insert into branch values('sbi_parliamentRoad','delhi',10000);  
insert into branch values('sbi_jantarmanatar','delhi',20000);  
  
insert into bankAccount values(2,'sbi_residencyRoad',5000);
```


insert into bankAccount values(3,'sbi_shivajiRoad',6000);

insert into bankAccount values(4,'sbi_parliamentRoad',9000);

insert into bankAccount values(5,'sbi_jantarmentar',8000);

insert into bankAccount values(6,'sbi_shivajiRoad',4000);

insert into bankAccount values(1,'sbi_chamrajpet',2000);

insert into bankCustomer values('Avinash','banerghatta_road','bangalore');

insert into bankCustomer values('Dinesh','nationalCollege_road','bangalore');

insert into bankCustomer values('Nikil','akbar_road','delhi');

insert into bankCustomer values('Ravi','prithviraj_road','delhi');

insert into bankCustomer values('Arjun','bull_temple_road','bangalore');

insert into depositer values('Avinash',1);

insert into depositer values('Dinesh',2);

insert into depositer values('Nikil',3);

insert into depositer values('Ravi',4);

insert into depositer values('Arjun',5);

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_parliamentroad',4000);

insert into loan values(5,'sbi_jantarmentar',5000);

Query OK, 1 row affected (0.00 sec)

[mysql> 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	mumbai	20000

5 rows in set (0.01 sec)

[mysql> select * from bankaccount;

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

6 rows in set (0.00 sec)

[mysql> select * from bankCustomer;

customer_name	customer_street	customer_city
Arjun	bull_temple_road	bangalore
Avinash	banerghatta_road	bangalore
Dinesh	nationalCollege_road	bangalore
Nikil	akbar_road	delhi
Ravi	prithviraj_road	delhi

5 rows in set (0.00 sec)

[mysql> select * from depositer;

customer_name	accno
Avinash	1
Dinesh	2
Nikil	3
Ravi	4
Arjun	5

5 rows in set (0.00 sec)

[mysql> select * from loan;

loan_no	branch_name	amount
1	sbi_chamrajpet	1000
2	sbi_residencyroad	2000
3	sbi_shivajiroad	3000
4	sbi_parlimentroad	4000
5	sbi_jantarmantar	5000

5 rows in set (0.00 sec)

Queries

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

>select distinct(customer_name) from depositer d, bankaccount b where (d.accno=b.accno and b.branch_name='sbi_residencyRoad') having(count(*)>1);

```
mysql> create view loan_sum as
  -> select branch_name, sum(amount)
  -> from loan
  -> group by branch_name;
Query OK, 0 rows affected (0.01 sec)

[mysql> select * from loan_sum;
+-----+-----+
| branch_name | sum(amount) |
+-----+-----+
| sbi_chamrajpet | 1000 |
| sbi_jantarmantar | 5000 |
| sbi_parlimentroad | 4000 |
| sbi_residencyroad | 2000 |
| sbi_shivajiroad | 3000 |
+-----+-----+
5 rows in set (0.01 sec)

mysql>
```

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

```
>create view loan_sum as
select branch_name, sum(amount)
from loan
group by branch_name;
```

Output:-

sbi_chamrajpet	11000
Sbi_jantarmantar	5000
sbi_parlimentRoad	4000
sbi_residencyRoad	2000
Sbi_shivajiroad	3000

Q. Display branch name and assets from all branches in lakhs of rupees and rename assets column to assets in Lakhs

```
>select branch_name,branch_city,assets_in_lakhs/100000 as assets_in_Lakhs from Branch;
```

```
[mysql> select branch_name,branch_city,assets/100000 as assets_in_Lakhs from Branch;
+-----+-----+-----+
| branch_name | branch_city | assets_in_Lakhs |
+-----+-----+-----+
| sbi_chamrajpet | bangalore | 0.5 |
| sbi_jantarmantar | delhi | 0.2 |
| sbi_parlimentRoad | delhi | 0.1 |
| sbi_residencyRoad | bangalore | 0.1 |
| sbi_shivajiRoad | mumbai | 0.2 |
+-----+-----+-----+
5 rows in set (0.00 sec)

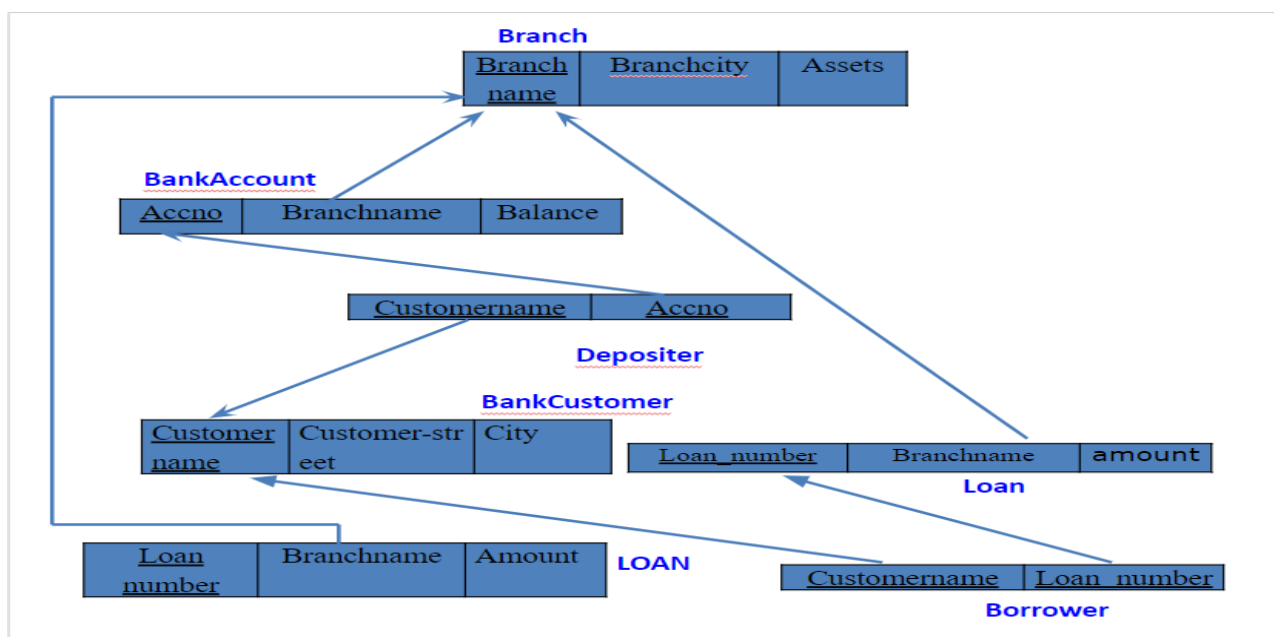
mysql> █
```

4. More Queries on Bank Database

Question

1. Find all the customers who have an account at all the branches located in a specific city (Ex. Delhi).
2. Find all customers who have a loan at the bank but do not have an account.
3. Find all customers who have both an account and a loan at the Bangalore branch
4. Find the names of all branches that have greater assets than all branches located in Bangalore.
5. Demonstrate how you delete all account tuples at every branch located in a specific city (Ex. Bombay).
6. Update the Balance of all accounts by 5%

Schema Diagram



Create database

```
create database bankdetails;
```

```
use bankdetails;
```

Create table

```
create table Branch (branch_name varchar(30), branch_city  
varchar(15), assets real, primary key(branch_name));
```

```
create table BankAccount (accno int, branch_name varchar(30),  
balance float, primary key(accno), foreign key(branch_name)  
references Branch(branch_name) on delete cascade on update  
cascade);
```

```
create table BankCustomer (customer_name varchar(20),  
customer_street varchar(20), customer_city varchar(10), primary  
key(customer_name));
```

```
create table depositor(customer_name varchar(20), accno int, primary  
key(customer_name, accno), foreign key (customer_name) references  
BankCustomer(customer_name) on delete cascade on update cascade,  
foreign key(accno) references BankAccount(accno) on delete cascade  
on update cascade);
```

```
create table Borrower(customer_name varchar(20), loan_number int ,  
primary key(customer_name));
```

```
loan_number int,  
branch_name varchar(30),  
amount float, primary key(loan_number), foreign key(branch_name)  
references Branch(branch_name) on delete cascade on update  
cascade);  
  
create table Borrower(  
  
SET FOREIGN_KEY_CHECKS=0;  
  
SET GLOBAL FOREIGN_KEY_CHECKS=0;
```

Structure of the table

```
[mysql> show tables;  
+-----+  
| Tables_in_bankdatabase |  
+-----+  
| bankAccount  
| bankCustomer  
| branch  
| depositer  
| loan  
+-----+  
5 rows in set (0.00 sec)  
  
mysql> █
```

Inserting Values to the table

```
insert into BankAccount values(1, 'SBI-Chamrajpet', 2000);
insert into BankAccount values(2, 'SBI-Residency Road', 5000);
insert into BankAccount values(3, 'SBI-Shivaji Road', 6000);
insert into BankAccount values(4, 'SBI-Parliament Road', 9000);
insert into BankAccount values(5, 'SBI-JantarMantar', 8000);
insert into BankAccount values(6, 'SBI-Shivaji Road', 4000);
insert into BankAccount values(8, 'SBI-Residency Road', 4000);
insert into BankAccount values(9, 'SBI-Parliament Road', 3000);
insert into BankAccount values(10, 'SBI-Residency Road', 5000);
insert into BankAccount values(11, 'SBI-JantarMantar', 2000);
select * from BankAccount;

SET FOREIGN_KEY_CHECKS=0;

SET GLOBAL FOREIGN_KEY_CHECKS=0;

insert into BankCustomer values('Avinash', 'Bull-Temple
Road','Bangalore');

insert into BankCustomer values('Dinesh', 'Bannerghatta-
Road','Bangalore');

insert into BankCustomer values('Mohan', 'NationalCollege-
Road','Bangalore');

insert into BankCustomer values('Nikhil', 'Akbar-Road','Delhi');
```



```
insert into BankCustomer values('Ravi', 'Prithviraj-Road','Delhi');  
  
select * from BankCustomer;  
  
desc BankCustomer;  
  
drop table BankCustomer;  
  
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 ('Nikhil', 9);  
  
insert into depositor values ('Dinesh', 10);  
  
insert into depositor values ('Nikil', 11);  
  
select * from depositor;  
  
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-ParliamentRoad','Delhi', 10000);  
  
insert into Branch values('SBI-JantarMantar','Delhi', 20000);  
  
insert into Branch values('SBI-MantriMarg','Delhi',20000);
```

```
select * from Branch;  
  
insert into loan values(1, 'SBI-Chamrajpet',1000);  
insert into loan values(2, 'ResidencyRoad',2000);  
insert into loan values(3, 'SBI-ShivajiRoad',3000);  
insert into loan values(4, 'SBI-ParliamentRoad',4000);  
insert into loan values(5, 'SBI-JantarMantar',5000);
```

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

```
Query OK, 1 row affected (0.00 sec)

mysql> 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 | mumbai | 20000 |
+-----+-----+-----+
5 rows in set (0.01 sec)

mysql> select * from bankaccount;
+-----+-----+-----+
| 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 |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from bankCustomer;
+-----+-----+-----+
| customer_name | customer_street | customer_city |
+-----+-----+-----+
| Arjun | bull_temple_road | bangalore |
| Avinash | banerghatta_road | bangalore |
| Dinesh | nationalCollege_road | bangalore |
| Nikil | akbar_road | delhi |
| Ravi | prithviraj_road | delhi |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from depositer;
+-----+-----+
| customer_name | accno |
+-----+-----+
| Avinash | 1 |
| Dinesh | 2 |
| Nikil | 3 |
| Ravi | 4 |
| Arjun | 5 |
+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from loan;
+-----+-----+-----+
| loan_no | branch_name | amount |
+-----+-----+-----+
| 1 | sbi_chamrajpet | 1000 |
| 2 | sbi_residencyroad | 2000 |
| 3 | sbi_shivajiroad | 3000 |
| 4 | sbi_parlimentroad | 4000 |
| 5 | sbi_jantarmantar | 5000 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Queries

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

select d.customername from branch b, depositer d,
bankaccount ba

where b.branchcity='Delhi' and

d.accno=ba.accno and

b.branchname=ba.branchname

group by d.customername

having count(customername)>1;

	customername
▶	Niki

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

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

	customername
▶	Mohan

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

Bangalore branch

```
select distinct d.customername from depositer d
where d.customername in(
  select d.customername from branch br,depositer d, bankaccount
  ba where br.branchcity="Bangalore" and
```

```
br.branchname=ba.branchname and ba.accno=d.accno and  
d.customername in(  
select customername from borrower));
```

	customername
▶	Avinash
	Dinesh

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

```
select branch_name from branch where assets_in_lakhs>(select sum(assets_in_lakhs) from branch  
where branch_city='Bangalore');
```

```
select branch_name from branch where assets_in_lakhs>(select sum(assets_in_lakhs) from  
branch where branch_city='Bangalore');
```

Result Grid	
	branch_name
*	NULL

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

```
delete from bank_account where branch_name in (select branch_name from branch where  
branch_city='Bombay');
```

Result Grid			
Filter Rows:			
	accno	branch_name	balance
▶	1	SBI_CHAMRAJPET	2100
	2	SBI_RESIDENCYROAD	5250
	4	SBI_PARLIMENTROAD	9450
	5	SBI_JANTARMANTAR	8400
	8	SBI_RESIDENCYROAD	4200
	9	SBI_PARLIMENTROAD	3150
	10	SBI_RESIDENCYROAD	5250
	11	SBI_JANTARMANTAR	2100
*	NULL	NULL	NULL

6) Update the Balance of all accounts by 5%

Update bank_account

Set balance = balance * 1.05

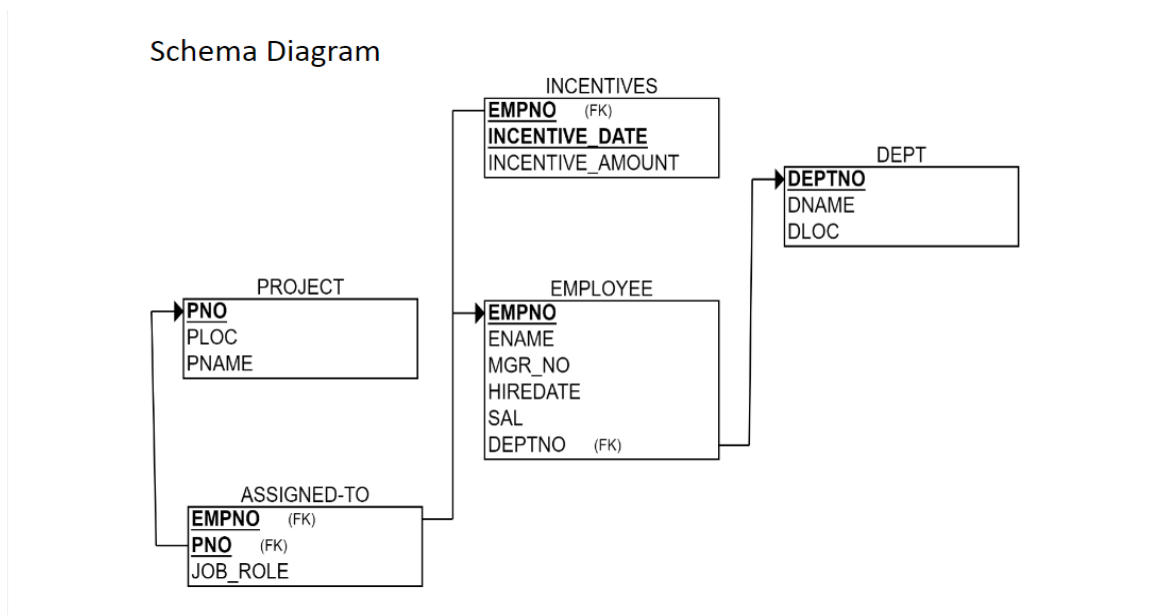
Result Grid			
Filter Rows:			
	accno	branch_name	balance
▶	1	SBI_CHAMRAJPET	2205
	2	SBI_RESIDENCYROAD	5513
	4	SBI_PARLIMENTROAD	9923
	5	SBI_JANTARMANTAR	8820
	8	SBI_RESIDENCYROAD	4410
	9	SBI_PARLIMENTROAD	3308
	10	SBI_RESIDENCYROAD	5513
	11	SBI_JANTARMANTAR	2205
*	NULL	NULL	NULL

5.Employee Database

Question

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 employees;  
use employees;
```

Create table

```
create table dept(  
d_no int,  
d_name varchar (10),  
d_loc varchar (30),  
primary key(d_no));
```

```
create table project(  
p_no int,  
p_loc varchar(20),  
p_name varchar(15),  
PRIMARY KEY(p_no));
```

```
create table employee(  
emp_no int,  
emp_name varchar(10),  
mgr_no int,  
hiredate date,  
sal real,  
d_no int,  
primary key(emp_no),  
foreign key(d_no) references dept(d_no)  
on update cascade on delete cascade);
```

```
create table incentives(  
emp_no int,  
incentive_date date,  
incentive_amt real,  
primary key(incentive_date),  
foreign key(emp_no) references employee(emp_no)  
on update cascade on delete cascade);
```

```
create table assigned(  

```



```
emp_no int,  
p_no int,  
job_role varchar(10),  
foreign key(emp_no) references employee(emp_no)  
on update cascade on delete cascade,  
foreign key(p_no) references project(p_no)  
on update cascade on delete cascade);
```

Structure of the table

```
[mysql> show tables;  
+-----+  
| Tables_in_employee |  
+-----+  
| assigned            |  
| dept                |  
| employee            |  
| incentives          |  
| project             |  
+-----+  
5 rows in set (0.00 sec)
```

Inserting Values to the table

```
insert into dept values(100,'IT','mysore');  
insert into dept values(200,'Marketing','patna');  
insert into dept values(300,'HR','delhi');  
insert into dept values(400,'finance','panaji');  
insert into dept values(500,'logistics','bangalore');  
insert into dept values(600,'accounts','ahmedabad');  
insert into dept values(700,'design','hydrebad');  
  
insert into project values(10,'mysore','alpha');  
insert into project values(20,'patna','beta');
```

```
insert into project values(30,'delhi','gama');
insert into project values(40,'panaji','delta');
insert into project values(50,'bangalore','omega');
insert into project values(60,'ahmedabad','sin');
```

```
insert into employee values(01,'charlie',11,'2001-01-01',10000,500);
insert into employee values(02,'michel',22,'2002-02-02',15000,100);
insert into employee values(03,'suzan',33,'2003-03-03',20000,300);
insert into employee values(04,'velvet',44,'2004-04-04',25000,200);
insert into employee values(05,'june',55,'2005-05-05',30000,500);
insert into employee values(06,'july',66,'2006-06-06',35000,500);
insert into employee values(07,'autumn',77,'2007-07-07',40000,400);
insert into employee values(08,'daisy',88,'2008-08-08',45000,100);
insert into employee values(09,'april',99,'2009-09-09',50000,200);
insert into employee values(10,'spring',111,'2010-10-10',55000,300);
```

```
insert into incentives values(01,'2021-05-11',1000);
insert into incentives values(03,'2022-04-20',1500);
insert into incentives values(05,'2021-09-10',2000);
insert into incentives values(07,'2022-06-21',2500);
insert into incentives values(09,'2021-04-17',3000);
insert into incentives values(10,'2022-10-19',3500);
```

```
insert into assigned values(01,20,'architect');
insert into assigned values(02,10,'peon');
insert into assigned values(03,30,'supervisor');
insert into assigned values(04,60,'assistant');
insert into assigned values(05,50,'mason');
```

insert into assigned values(06,50,'manager');

insert into assigned values(07,40,'medic');

insert into assigned values(08,60,'mechanic');

insert into assigned values(09,30,'engineer');

insert into assigned values(10,30,'maid');

```
mysql> select * from assigned;
+-----+-----+-----+
| emp_no | p_no | job_role |
+-----+-----+-----+
|      1 |   20 | architect |
|      2 |   10 | peon      |
|      3 |   30 | supervisor |
|      4 |   60 | assistant |
|      5 |   50 | mason      |
|      6 |   50 | manager    |
|      7 |   40 | medic      |
|      8 |   60 | mechanic   |
|      9 |   30 | engineer   |
|     10 |   30 | maid       |
+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> █
```

```
mysql> select * from dept;
```

d_no	d_name	d_loc
100	IT	mysore
200	Marketing	patna
300	HR	delhi
400	finance	panaji
500	logistics	bangalore
600	accounts	ahmedabad
700	design	hydrebad

```
7 rows in set (0.00 sec)
```



```
mysql> select * from project;
```

p_no	p_loc	p_name
10	mysore	alpha
20	patna	beta
30	delhi	gama
40	panaji	delta
50	bangalore	omega
60	ahmedabad	sin

```
6 rows in set (0.00 sec)
```



```
mysql> select * from employee;
```

emp_no	emp_name	hiredate	sal	d_no
1	charlie	2001-01-01	10000	500
2	micheel	2002-02-02	15000	100
3	suzan	2003-03-03	20000	300
4	velvet	2004-04-04	25000	200
5	june	2005-05-05	30000	500
6	july	2006-06-06	35000	500
7	automn	2007-07-07	40000	400
8	daisy	2008-08-08	45000	100
9	april	2009-09-09	50000	200
10	spring	2010-10-10	55000	300

```
10 rows in set (0.00 sec)
```



```
mysql> select * from incentives;
```

emp_no	incentive_date	incentive_amt
9	2021-04-17	3000
1	2021-05-11	1000
5	2021-09-10	2000
3	2022-04-20	1500
7	2022-06-21	2500
10	2022-10-19	3500

```
6 rows in set (0.01 sec)
```

Queries

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

```
select emp_no  
from assigned a, project p  
where a.p_no = p.p_no and p.p_loc in ('mysore', 'hyderabad', 'bangalore');
```

Result Grid	
	emp_no
▶	2
	5
	6

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

```
select distinct(e.emp_no)  
FROM employee e, incentives i  
where e.emp_no not in (select emp_no from incentives);
```

Result Grid	
	emp_no
▶	2
	8
	4
	6

Q. 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.emp_name,e.emp_no,d.d_name,a.job_role,d.d_loc,p.p_loc
from employee e, dept d, assigned a, project p
where d.d_no=e.d_no
and e.emp_no=a.emp_no
and a.p_no=p.p_no
and p.p_loc=d.d_loc;
```

emp_name	emp_no	d_name	job_role	d_loc	p_loc
▶ michel	2	IT	peon	mysore	mysore
suzan	3	HR	supervisor	delhi	delhi
spring	10	HR	maid	delhi	delhi
spring	10	HR	maid	delhi	delhi
autumn	7	finance	medic	panaji	panaji
june	5	logistics	mason	bangalore	bangalore
july	6	logistics	manager	bangalore	bangalore

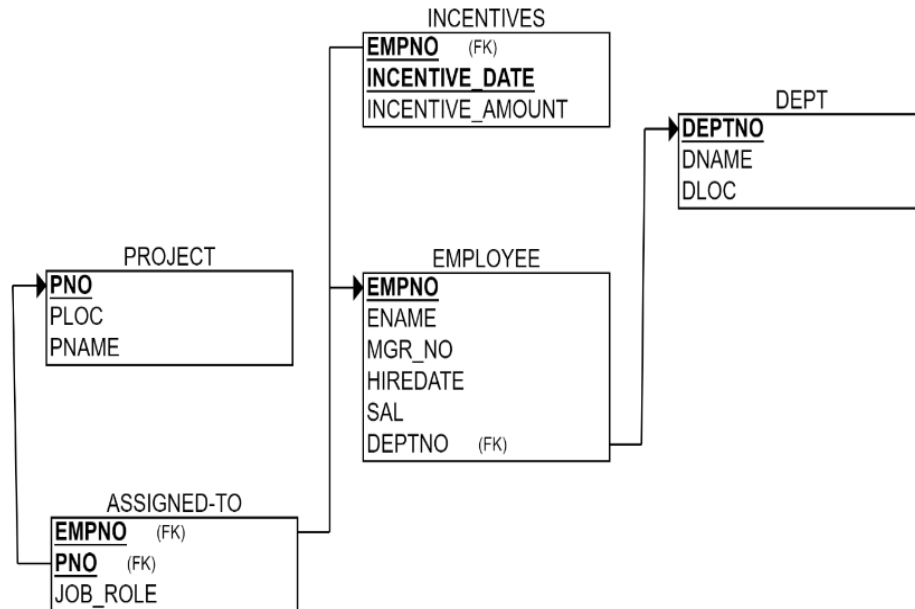
6. More Queries on Employee Database

Question

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. List the name of the managers with the maximum employees
4. Display those managers name whose salary is more than average salary of his employee.
5. Find the name of the second top level managers of each department.
6. Find the employee details who got second maximum incentive in January 2019.
7. Display those employees who are working in the same department where his manager is working.

Schema Diagram

Schema Diagram



Create database

```
create database employee;
use employee;
```

Create table

```
create table project (p_no int, p_name varchar(20), p_loc varchar(20), primary
key(p_no));
create table dept (dept_no int, d_name varchar(20), d_loc varchar(20), primary
key(dept_no));
create table employee (emp_no int, e_name varchar(20), mgr_no int, hire_date
date, sal int, dept_no int, primary key(emp_no), foreign key(dept_no) references
dept(dept_no) on delete cascade);
create table incentives (emp_no int, incentive_date date, incentive_amt int, primary
key(emp_no,incentive_date), foreign key(emp_no) references employee(emp_no) on
delete cascade);
create table assigned_to (emp_no int, p_no int, job_role varchar(20), primary
key(emp_no,p_no), foreign key (emp_no) references employee(emp_no) on delete
cascade, foreign key(p_no) references project(p_no) on delete cascade);
show tables;
```

Structure of the table


```
mysql> show tables;
+-----+
| Tables_in_employeee |
+-----+
| assigned_to          |
| dept                 |
| employee             |
| incentives           |
| project              |
+-----+
5 rows in set (0.00 sec)
```

Inserting Values to the table

```
insert into dept values(100,'IT','mysore');
```

```
insert into dept values(200,'Marketing','patna');
```

```
insert into dept values(300,'HR','delhi');
```

```
insert into dept values(400,'finance','panaji');
```

```
insert into dept values(500,'logistics','bangalore');
```

```
insert into dept values(600,'accounts','ahmedabad');
```

```
insert into dept values(700,'design','hydrebad');
```

```
insert into project values(10,'mysore','alpha');
```

```
insert into project values(20,'patna','beta');
```

```
insert into project values(30,'delhi','gama');
```

```
insert into project values(40,'panaji','delta');
```

```
insert into project values(50,'bangalore','omega');
```

```
insert into project values(60,'ahmedabad','sin');
```

```
insert into employee values(01,'charlie',11,'2001-01-01',10000,500);
insert into employee values(02,'michel',22,'2002-02-02',15000,100);
insert into employee values(03,'suzan',33,'2003-03-03',20000,300);
insert into employee values(04,'velvet',44,'2004-04-04',25000,200);
insert into employee values(05,'june',55,'2005-05-05',30000,500);
insert into employee values(06,'july',66,'2006-06-06',35000,500);
insert into employee values(07,'automn',77,'2007-07-07',40000,400);
insert into employee values(08,'daisy',88,'2008-08-08',45000,100);
insert into employee values(09,'april',99,'2009-09-09',50000,200);
insert into employee values(10,'spring',111,'2010-10-10',55000,300);
```

```
insert into incentives values(01,'2021-05-11',1000);
insert into incentives values(03,'2022-04-20',1500);
insert into incentives values(05,'2021-09-10',2000);
insert into incentives values(07,'2022-06-21',2500);
insert into incentives values(09,'2021-04-17',3000);
insert into incentives values(10,'2022-10-19',3500);
```

```
insert into assigned values(01,20,'architect');
insert into assigned values(02,10,'peon');
insert into assigned values(03,30,'supervisor');
insert into assigned values(04,60,'assistant');
insert into assigned values(05,50,'mason');
insert into assigned values(06,50,'manager');
insert into assigned values(07,40,'medic');
insert into assigned values(08,60,'mechanic');
```

```
insert into assigned values(09,30,'engineer');
```

```
insert into assigned values(10,30,'maid');
```

Query OK, 0 rows affected (0.01 sec)

mysql> SET GLOBAL FOREIGN_KEY_CHECKS=0;

Query OK, 0 rows affected (0.00 sec)

mysql> select * from assigned_to;

emp_no	p_no	job_role
7	1	Leader
10	2	Architect
11	2	Manager
12	3	Site Engineering
12	5	Developer
13	7	Finanace Manager
15	4	Director
15	6	CEO
17	5	Entertainer

9 rows in set (0.00 sec)

mysql> select * from employee;

emp_no	e_name	mgr_no	hire_date	sal	dept_no
7	Velvet	107	2007-07-07	70000	3
10	John	110	2010-10-10	30000	2
11	Michel	102	2015-01-12	55000	6
12	Suzan	101	2016-11-21	35000	1
13	Harry	104	2015-07-25	20000	4
15	Charlie	102	2016-11-21	45000	6
17	Simon	107	2012-07-07	60000	3

7 rows in set (0.00 sec)

mysql> select * from project;

p_no	p_name	p_loc
1	World cup	Qatar
2	Hostel Rooms	Rajasthan
3	Railway Station	Gujarat
4	Kennys company	Mumbai
5	Theme Park	Bangalore
6	national Airport	Rajasthan
7	Varun Gaming	Kerala
8	HS briyani	Hyderbad

8 rows in set (0.00 sec)

mysql> select * from dept;

dept_no	d_name	d_loc
1	R & D	Bangalore
2	Architecture	Mumbai
3	Sports	Kerala
4	Finance	Gujarat
5	Engineering	Bangalore
6	Business Management	Rajasthan

6 rows in set (0.00 sec)

```
[mysql> select * from incentives;
```

emp_no	incentive_date	incentive_amt
7	2019-01-21	2000
11	2020-10-15	8000
13	2021-11-05	5000
15	2015-11-15	10000
17	2021-11-05	9000

```
5 rows in set (0.00 sec)

mysql>
```

Queries

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

```
create view manager_count (mgr_no, count) as select mgr_no, count(mgr_no)
from employee group by mgr_no;
```

```
select employee.emp_name from employee where mgr_no = (select mgr_no from
manager_count where count = (select max(count) from manager_count));
```

Result Grid	
	emp_name
▶	daisy
	april
	spring

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

```
select emp_name from employee e1 where sal > (select avg(sal) from employee e2 where e1.mgr_no = e2.mgr_no);
```

	emp_name
▶	spring

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

```
select emp_name from employee e where d_no in (select d_no from employee e1 where e1.mgr_no=e.emp_no);
```

	emp_name
	charlie
	micheel
	suzan

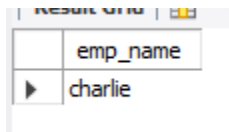
Q. Find the employee details who got second maximum incentive in January 2019.

```
select * from employee where emp_no = (select emp_no from incentives i where i.incentive_amt < (select max(incentive_amt) from incentives having incentive_date like '2021%') order by incentive_amt desc limit 1);
```

	emp_no	emp_name	mgr_no	hiredate	sal	d_no
▶	9	april	99	2009-09-09	50000	200
*	NULL	NULL	NULL	NULL	NULL	NULL

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

```
select emp_name from employee where emp_no in((select distinct mgr_no from employee));
```



	emp_name
▶	charlie

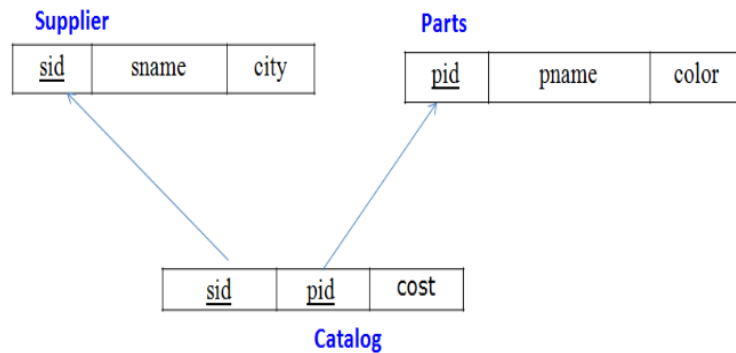
7. Supplier Database

Question

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

Schema Diagram



Create database

```
create database supplier;
```

```
use supplier;
```

Create table

```
create table supllier(
```

```
sid varchar(20),
```

```
sname varchar(20),
```

```
city varchar(20),
```

```
primary key(sid)
```

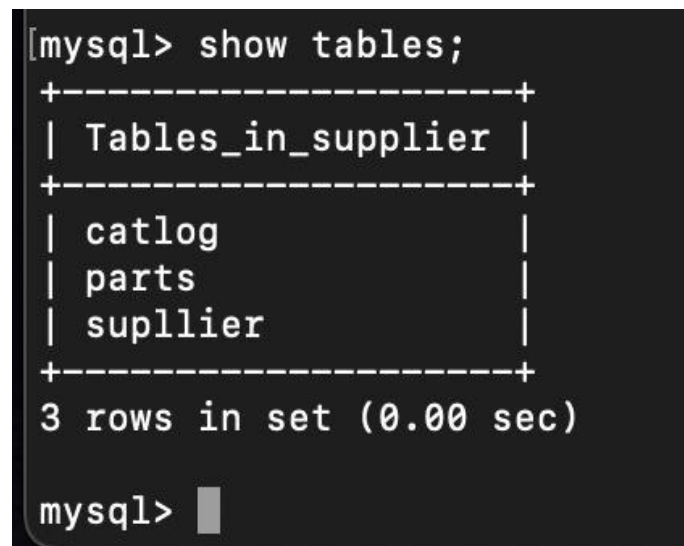
```
);
```

```
DESC supllier;
```

```
create table parts(
```

```
pid varchar(20),
pname varchar(20),
color varchar(20),
primary key(pid)
);
create table catlog(
sid varchar(20),
pid varchar(20),
cost varchar(20),
primary key(sid,pid),
foreign key(pid)references parts(pid),
foreign key(sid)references supllier(sid)
);
```

Structure of the table



```
[mysql> show tables;
+-----+
| Tables_in_supplier |
+-----+
| catlog              |
| parts               |
| supllier            |
+-----+
3 rows in set (0.00 sec)

mysql> █
```

Inserting Values to the table

```
insert into supllier values(10001,'acme widget','bangalore');
```

```
insert into supllier values(10002,'johns','kolkata');
```

```
insert into supllier values(10003,'vimal','mumbai');
```

```
insert into supllier values(10004,'reliance','delhi');
```

```
insert into parts values(20001,'book','red');
```

```
insert into parts values(20002,'pen','red');
```

```
insert into parts values(20003,'pencil','green');
```

```
insert into parts values(20004,'mobile','green');
```

```
insert into parts values(20005,'charger','black');
```

```
insert into catlog values(10001,20001,10);
```

```
insert into catlog values(10001,20002,10);
```

```
insert into catlog values(10001,20003,30);
```

```
insert into catlog values(10001,20004,10);
```

```
insert into catlog values(10001,20005,10);
```

```
insert into catlog values(10002,20001,10);
```

```
insert into catlog values(10002,20002,20);
```

```
insert into catlog values(10003,20003,30);
```

```
insert into catlog values(10004,20003,40);
```

```
mysql> select * from suplllier; select * from parts; select * from catlog;  
[ERROR 1146 (42S02): Table 'supplier.suplllier' doesn't exist
```

pid	pname	color
20001	book	red
20002	pen	red
20003	pencil	green
20004	mobile	green
20005	charger	black

```
[5 rows in set (0.00 sec)]
```

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

```
9 rows in set (0.00 sec)
```

```
mysql> select * from supplier;
```

sid	sname	city
10001	acme widget	bangalore
10002	johns	kolkata
10003	vimal	mumbai
10004	reliance	delhi

```
4 rows in set (0.00 sec)
```

Queries

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

```
>select pname from parts where pid IN (select pid from catlog);
```

```
mysql> select pname from parts where pid IN (select pid from catlog);
+-----+
| pname |
+-----+
| book  |
| pen   |
| pencil|
| mobile|
| charger|
+-----+
5 rows in set (0.00 sec)
```

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

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

```
mysql> select sname from
-> (select c.sname,count(distinct a.pid) as cnt from catlog a
-> left join parts b on a.pid=b.pid
-> left join supplier c on c.sid=a.sid group by 1) a
-> where cnt=(select count(distinct a.pid) from catlog a
-> left join parts b on a.pid=b.pid);
+-----+
| sname |
+-----+
| acme widget |
+-----+
1 row in set (0.01 sec)
```

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

```
>select distinct sname from(select c.sname,b.pname,b.color from catlog a
left join parts b on a.pid=b.pid left join supplier c on c.sid=a.sid )a
```

where color='red';

```
mysql> select distinct sname from (select c.sname, b.pname, b.color from catlog a
-> left join parts b on a.pid=b.pid left join supplier c on c.sid=a.sid )a
-> where color='red';
+-----+
| sname |
+-----+
| acme widget |
| johns   |
+-----+
2 rows in set (0.00 sec)
```

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

>select A.pname from parts A

left join catlog B on A.pid=B.pid

left join supplier C on B.sid=C.sid where lower(c.sname)='acme widget'

and a.pname not in (select A.pname from parts A

left join catlog B on A.pid=B.pid

left join supplier C on B.sid=C.sid where lower(c.sname)<>'acme widget');

```
mysql> select A.pname from parts A
-> left join catlog B on A.pid=B.pid
-> left join supplier C on B.sid=C.sid where lower(c.sname)='acme widget'
-> and a.pname not in (select A.pname from parts A
-> left join catlog B on A.pid=B.pid
-> left join supplier C on B.sid=C.sid where lower(c.sname)<>'acme widget');
+-----+
| pname |
+-----+
| mobile |
| charger |
+-----+
2 rows in set (0.00 sec)

mysql> █
```

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

```

>select a.sid from
(select A.pid,C.sid,cost from parts A
left join catlog B on A.pid=B.pid
left join supllier C on B.sid=C.sid )A
left join
(select A.pid,avg(cost) as cost from parts A
left join catlog B on A.pid=B.pid
left join supllier C on B.sid=C.sid where cost is not null group by 1 )B on A.pid=B.pid
where a.cost>b.cost

```

```

mysql> select a.sid from
-> (select A.pid,C.sid,cost from parts A
-> left join catlog B on A.pid=B.pid
-> left join supllier C on B.sid=C.sid )A
-> left join
-> (select A.pid,avg(cost) as cost from parts A
-> left join catlog B on A.pid=B.pid
-> left join supllier C on B.sid=C.sid where cost is not null group by 1 )B on A.pid=B.pid
-> where a.cost>b.cost
-> ;
+-----+
| sid   |
+-----+
| 10002 |
| 10004 |
+-----+
2 rows in set (0.00 sec)

```

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

```

>select pid,sname from
(select A.pid,C.sname,cost,rank() over(partition by pid order by cost desc) as rnk from parts x A
left join catlog B on A.pid=B.pid left join supllier C on B.sid=C.sid)A where rnk=1 and cost is not
null order by sname;

```

```
mysql> select pid,sname from
-> (select A.pid,C.sname,cost,rank() over(partition by pid order by cost desc) as rnk from parts A
-> left join catlog B on A.pid=B.pid
-> left join supllier C on B.sid=C.sid)A where rnk=1 and cost is not null order by sname;
```

pid	sname
20001	acme widget
20004	acme widget
20005	acme widget
20001	johns
20002	johns
20003	reliance

```
6 rows in set (0.01 sec)

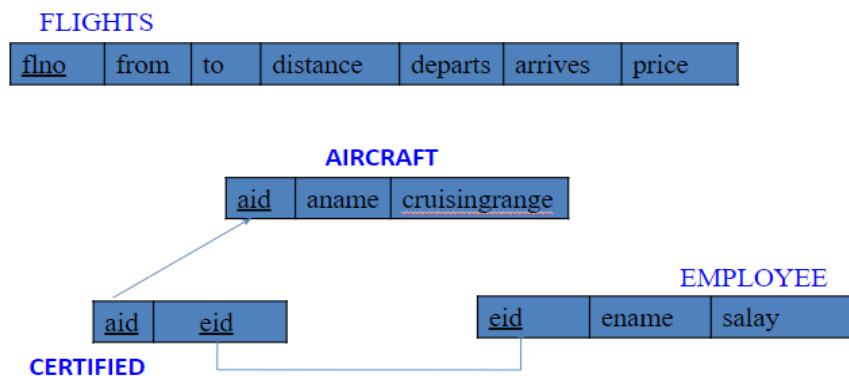
mysql>
```


8.Flight Database

Questions

- i. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.
- ii. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.
- iii. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.
- iv. For all aircraft with cruising range over 1000 Kms, find the name of the aircraft and the Average salary of all pilots certified for this aircraft.
- v. Find the names of pilots certified for some Boeing aircraft.
- vi. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.

Schema Diagram



Create database

```
create database Airline;  
use Airline;
```

Create table

```
create table flights(  
  flno int,  
  ffrom varchar(50),  
  tto varchar(50),  
  distance int,  
  departs time,  
  arrives time,  
  price int,  
  primary key(flno));  
create table aircraft(  
  aid int,  
  aname varchar(50),  
  cruisingrange int,  
  primary key(aid));  
create table employee(  
  eid int,  
  ename varchar(50),  
  salary int,  
  primary key(eid));  
create table certified(  
  eid int, aid int,  
  foreign key(aid) references aircraft(aid)  
  on update cascade on delete cascade,
```

foreign key(eid) references employee(eid)
on update cascade on delete cascade);

Structure of the table

```
[mysql> show tables;
+-----+
| Tables_in_airline |
+-----+
| aircraft          |
| certified         |
| employee          |
| flights           |
+-----+
4 rows in set (0.01 sec)
```

Inserting Values to the table

insert into employee values

(101,'Avinash',50000),
(102,'Lokesh',60000),
(103,'Rakesh',70000),
(104,'Santhosh',82000),
(105,'Tilak',5000);

insert into aircraft values

(1,'Airbus',2000),
(2,'Boeing',700),
(3,'JetAirways',550),
(4,'Indigo',5000),

```
(5,'Boeing',4500),  
(6,'Airbus',2200);
```

insert into certified values

```
(101,2),(101,4),(101,5),  
(101,6),(102,1),(102,3),  
(102,5),(103,2),(103,3),  
(103,5),(103,6),(104,6),  
(104,1),(104,3),(105,3);
```

insert into flights values

```
(1,'Banglore','New Delhi',500,'6:00','9:00',5000),  
(2,'Banglore','Chennai',300,'7:00','8:30',3000),  
(3,'Trivandrum','New Delhi',800,'8:00','11:30',6000),  
(4,'Banglore','Frankfurt',10000,'6:00','23:30',50000),  
(5,'Kolkata','New Delhi',2400,'11:00','3:30',9000),  
(6,'Banglore','Frankfurt',8000,'9:00','23:00',40000);
```

```

mysql> select * from aircraft;
+-----+-----+-----+
| aid | aname      | cruisingrange |
+-----+-----+-----+
| 1   | Airbus     | 2000          |
| 2   | Boeing     | 700           |
| 3   | JetAirways | 550           |
| 4   | Indigo      | 5000          |
| 5   | Boeing     | 4500          |
| 6   | Airbus     | 2200          |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from certified;
+-----+-----+
| eid | aid |
+-----+-----+
| 101 | 2   |
| 101 | 4   |
| 101 | 5   |
| 101 | 6   |
| 102 | 1   |
| 102 | 3   |
| 102 | 5   |
| 103 | 2   |
| 103 | 3   |
| 103 | 5   |
| 103 | 6   |
| 104 | 6   |
| 104 | 1   |
| 104 | 3   |
| 105 | 3   |
+-----+-----+
15 rows in set (0.00 sec)

mysql> select * from employee;
+-----+-----+-----+
| eid | ename      | salary |
+-----+-----+-----+
| 101 | Avinash    | 50000  |
| 102 | Lokesh     | 60000  |
| 103 | Rakesh     | 70000  |
| 104 | Santhosh   | 82000  |
| 105 | Tilak      | 5000   |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from flights;
+-----+-----+-----+-----+-----+-----+-----+
| flno | ffrom      | tto      | distance | departs | arrives | price |
+-----+-----+-----+-----+-----+-----+-----+
| 1   | Bangalore  | New Delhi | 500      | 06:00:00 | 09:00:00 | 5000  |
| 2   | Bangalore  | Chennai  | 300      | 07:00:00 | 08:30:00 | 3000  |
| 3   | Trivandrum | New Delhi | 800      | 08:00:00 | 11:30:00 | 6000  |
| 4   | Bangalore  | Frankfurt | 10000    | 06:00:00 | 23:30:00 | 50000 |
| 5   | Kolkata    | New Delhi | 2400     | 11:00:00 | 03:30:00 | 9000  |
| 6   | Bangalore  | Frankfurt | 8000     | 09:00:00 | 23:00:00 | 40000 |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

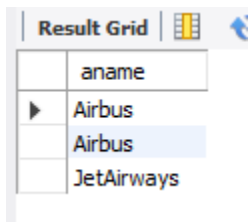
mysql>

```

Queries

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

```
SELECT a.aname
FROM aircraft a,certified c,employee e
WHERE a.aid=c.aid
AND c.eid=e.eid AND NOT EXISTS
(SELECT * FROM employee e1 WHERE e1.eid=e.eid AND e1.salary<80000);
```

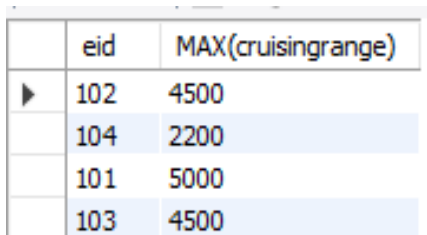


The screenshot shows a 'Result Grid' window with a table containing aircraft names. The table has one column labeled 'aname' and four rows of data: 'Airbus', 'Airbus', and 'JetAirways'. The second 'Airbus' row is highlighted in blue.

aname
Airbus
Airbus
JetAirways

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

```
SELECT c.eid,MAX(cruisingrange)
FROM certified c,aircraft a
WHERE c.aid=a.aid
GROUP BY c.eid
HAVING COUNT(*)>2;
```

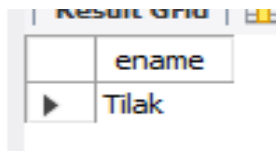


The screenshot shows a table with two columns: 'eid' and 'MAX(cruisingrange)'. There are four rows of data: (102, 4500), (104, 2200), (101, 5000), and (103, 4500). The first row is highlighted in blue.

eid	MAX(cruisingrange)
102	4500
104	2200
101	5000
103	4500

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

```
SELECT DISTINCT e.ename
FROM employee e
WHERE e.salary<
(SELECT MIN(f.price)
FROM flights f
WHERE f.ffrom='Bangalore'
AND f.tto='Frankfurt');
```



A screenshot of a database result grid. The grid has two columns: 'ename' and a value 'Tilak'. The 'ename' column is highlighted with a mouse cursor. The grid is titled 'Result Grid'.

	ename
▶	Tilak

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

```
select a.aid, a.aname, avg(e.salary)
from aircraft a, employee e, certified c
where a.aid=c.aid and c.eid=e.eid
and a.cruisingrange>1000
group by a.aid, a.aname;
```

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

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

```
SELECT distinct e.ename
FROM employee e,aircraft a,certified c
WHERE e.eid=c.eid
AND c.aid=a.aid
AND a.aname='Boeing';
```

	ename
▶	Avinash
	Rakesh
	Lokesh

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

```
SELECT a.aid
FROM aircraft a
WHERE a.cruisingrange>
(SELECT MIN(f.distance)
FROM flights f
WHERE f.ffrom='Bangalore'
AND f.tto='New Delhi');
```


Result Grid	
	aid
▶	1
	2
	3
	4
	5
	6
*	NULL

NOSQL LAB1 and LAB2

LAB-1 Perform the following DB operations using MongoDB.

1. Create a database “Student” with the following attributes Rollno, Age, ContactNo, Email-Id.

```
use myDB;
```

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

Output:

A screenshot of a MongoDB terminal window. The title bar shows the connection string: 'mongosh mongodb+srv://<credentials>@dbmslab.nutzzfd.mongodb.net/myFirstDatabase'. The terminal output shows the following commands and responses:
Atlas atlas-hwtuk-shard-0 [primary] myDB> use myDB
already on db myDB
Atlas atlas-hwtuk-shard-0 [primary] myDB> db.createCollection("Student");
{ ok: 1 }
Atlas atlas-hwtuk-shard-0 [primary] myDB> |

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" });
```

Output:

```
mongosh mongodb+srv://<credentials>@dbmslab.nu1zzfd.mongodb.net/myFirstDatabase
Atlas atlas-hvwtuk-shard-0 [primary] myDB> db.Student.find();
[
  {
    _id: ObjectId("63c6c1d58fc43cee59d05c7c"),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId("63c6c1e68fc43cee59d05c7d"),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de9@gmail.com'
  },
  {
    _id: ObjectId("63c6c1f68fc43cee59d05c7e"),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
  },
  {
    _id: ObjectId("63c6c2048fc43cee59d05c7f"),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId("63c6c22b8fc43cee59d05c80"),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'rekha.de9@gmail.com'
  }
]
Atlas atlas-hvwtuk-shard-0 [primary] myDB> 
```

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

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

Output:

```
mongosh mongodb+srv://<credentials>@dbmslab.nu1zzfd.mongodb.net/myFirstDatabase
Atlas atlas-hvwtuk-shard-0 [primary] myDB> db.Student.update({RollNo:10},{ $set:{email:"Abhinav@gmail.com"}});
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-hvwtuk-shard-0 [primary] myDB> db.Student.find({RollNo:10});
[
  {
    _id: ObjectId("63c6c22b8fc43cee59d05c80"),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'Abhinav@gmail.com'
  }
]
Atlas atlas-hvwtuk-shard-0 [primary] myDB> 
```

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},{ $set:{Name:"FEM"}});
```

Output:

```
mongosh mongodb+srv://<credentials>@dbmslab.nu1zzfd.mongodb.net/myFirstDatabase
Atlas atlas-hwtuk-shard-0 [primary] myDB> db.Student.insert({RollNo:11,Age:22,Name:"ABC",Cont:2276,email:"rea.de9@gmail.com"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63c6c6e6490337e41bcfc476") }
}
Atlas atlas-hwtuk-shard-0 [primary] myDB> db.Student.find({RollNo:11});
[
  {
    _id: ObjectId("63c6c6e6490337e41bcfc476"),
    RollNo: 11,
    Age: 22,
    Name: 'ABC',
    Cont: 2276,
    email: 'rea.de9@gmail.com'
  }
]
Atlas atlas-hwtuk-shard-0 [primary] myDB> db.Student.update({RollNo:11},{ $set:{Name:"FEM"}});
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-hwtuk-shard-0 [primary] myDB> db.Student.find({RollNo:11});
[
  {
    _id: ObjectId("63c6c6e6490337e41bcfc476"),
    RollNo: 11,
    Age: 22,
    Name: 'FEM',
    Cont: 2276,
    email: 'rea.de9@gmail.com'
  }
]
Atlas atlas-hwtuk-shard-0 [primary] myDB>
```

5. Export the created table into local file system

Mongoexport mongodb+srv://:Tanisha#511821@dbmslab.nu1zzfd.mongodb.net/myDB --
collection=Student --out /home/Tanishagotadke/Documents/output.json

Output:

```
suryabhat@pop-os: ~
suryabhat@pop-os:~$ mongoexport mongodb+srv://surya4206:Surya#511821@dbmslab.nu1zzfd.mongodb.net/myDB --collection=Student --out /home/suryabhat/Documents/output.json
2023-01-18T12:32:22.844+0530      connected to: mongodb+srv://[**REDACTED**]@dbmslab.nu1zzfd.mongodb.net/myDB
2023-01-18T12:32:23.106+0530      exported 6 records
suryabhat@pop-os:~$
```

6. Drop the table

```
db.Student.drop();
```

Output:

```
mongosh mongodb+srv://<credentials>@dbmslab.nu1zzfd.mongodb.net/myFirstDatabase
Atlas atlas-hwtuk-shard-0 [primary] myDB> db.Student.drop();
true
Atlas atlas-hwtuk-shard-0 [primary] myDB> 
```

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

mongoimport

mongodb+srv://Tanisha4206:Tanisha#511821@dbmslab.nu1zzfd.mongodb.net/myDB --
collection=New_Student --type json --file /home/Tanishagotadke/Documents/output.json

Output:

```
suryabhat@pop-os: ~$ mongoimport mongodb+srv://surya4206:Surya#511821@dbmslab.nu1zzfd.mongodb.net/myDB --collection=New_Student --type json --file /home/suryabhat/Documents/output.json
2023-01-18T12:48:16.403+0530 connected to: mongodb+srv://[REDACTED]@dbmslab.nu1zzfd.mongodb.net/myDB
2023-01-18T12:48:16.505+0530 6 document(s) imported successfully. 0 document(s) failed to import.
suryabhat@pop-os: ~$ 
```

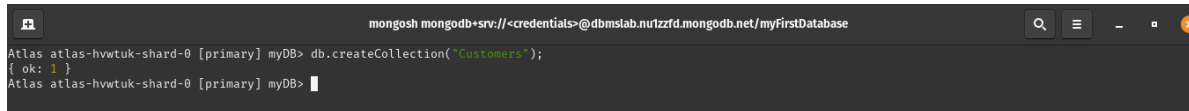
LAB2: Perform the following DB operations using MongoDB.

1. Create a collection by name Customers with the following attributes.

Cust_id, Acc_Bal, Acc_Type

```
db.createCollection("Customers");
```

Output:

A screenshot of a MongoDB terminal window. The title bar shows the connection string: 'mongosh mongodb+srv://<credentials>@dbmslab.nutzzfd.mongodb.net/myFirstDatabase'. The terminal output shows the command 'db.createCollection("Customers");' being executed, followed by the response '{ ok: 1 }'. The prompt 'Atlas atlas-hwtuk-shard-0 [primary] myDB>' is visible at the end of the line.

```
mongosh mongodb+srv://<credentials>@dbmslab.nutzzfd.mongodb.net/myFirstDatabase
Atlas atlas-hwtuk-shard-0 [primary] myDB> db.createCollection("Customers");
{ ok: 1 }
Atlas atlas-hwtuk-shard-0 [primary] myDB>
```

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"});
```

Output:

```
mongosh mongodb+srv://<credentials>@dbmslab.nur1zfd.mongodb.net/myFirstDatabase

{
  "_id": ObjectId("63c6cc0ead00846306e7c311"),
  "cust_id": 1,
  "Balance": 200,
  "Type": "S"
},
{
  "_id": ObjectId("63c6cc1ead00846306e7c312"),
  "cust_id": 1,
  "Balance": 1000,
  "Type": "Z"
},
{
  "_id": ObjectId("63c6cc31ad00846306e7c313"),
  "cust_id": 2,
  "Balance": 100,
  "Type": "Z"
},
{
  "_id": ObjectId("63c6cc3ead00846306e7c314"),
  "cust_id": 2,
  "Balance": 1000,
  "Type": "C"
},
{
  "_id": ObjectId("63c6cc4aad00846306e7c315"),
  "cust_id": 2,
  "Balance": 500,
  "Type": "C"
},
{
  "_id": ObjectId("63c6cc5fad00846306e7c316"),
  "cust_id": 2,
  "Balance": 50,
  "Type": "S"
},
{
  "_id": ObjectId("63c6cc6cad00846306e7c317"),
  "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}}});
```

Output:

```
mongosh mongodb+srv://<credentials>@dbmslab.nur1zfd.mongodb.net/myFirstDatabase

Atlas atlas-hvwtuk-shard-0 [primary] myDB> db.Customers.aggregate({$match:{Type:"Z"}},{ $group:{_id:"$cust_id",TotAccBal:{$sum:"$Balance"}}},{ $match:{TotAccBal:{$gt:1200}}});
[ { _id: 3, TotAccBal: 1500 } ]
Atlas atlas-hvwtuk-shard-0 [primary] myDB> █
```

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

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

Output:

```
mongosh mongodb+srv://<credentials>@dbmslab.nu1zzfd.mongodb.net/myFirstDatabase
Atlas atlas-hvwtuk-shard-0 [primary] myDB> db.Customers.aggregate({$group:{_id:"$cust_id",minAccBal:{$min:"$Balance"},maxAccBal:{$max:"$Balance"}}});
[
  { _id: 1, minAccBal: 200, maxAccBal: 1000 },
  { _id: 2, minAccBal: 50, maxAccBal: 1000 },
  { _id: 3, minAccBal: 500, maxAccBal: 1000 }
]
Atlas atlas-hvwtuk-shard-0 [primary] myDB> |
```

5. Export the created collection into local file system

mongoexport

mongodb+srv://Tanisha4206:Tanisha#511821@dbmslab.nu1zzfd.mongodb.net/myDB --collection=Customers --out /home/Tanishagotadke/Documents/output1.json

Output:

```
suryabhat@pop-os: ~$ mongoexport mongodb+srv://surya4206:Surya#511821@dbmslab.nu1zzfd.mongodb.net/myDB --collection=Customers --out /home/suryabhat/Documents/output1.json
2023-01-18T12:34:22.994+0530 connected to: mongodb+srv://[**REDACTED**]@dbmslab.nu1zzfd.mongodb.net/myDB
2023-01-18T12:34:23.184+0530 exported 8 records
suryabhat@pop-os: ~$ |
```

6. Drop the table

db.Customers.drop();

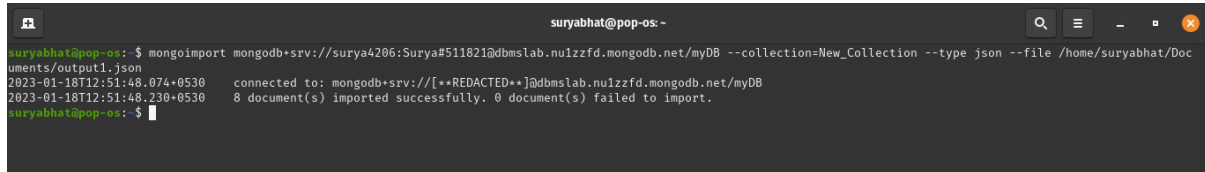
Output:

```
mongosh mongodb+srv://<credentials>@dbmslab.nu1zzfd.mongodb.net/myFirstDatabase
Atlas atlas-hvwtuk-shard-0 [primary] myDB> db.Customers.drop();
true
Atlas atlas-hvwtuk-shard-0 [primary] myDB> |
```

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


```
mongoimport
mongodb+srv://Tanisha4206:Tanisha#511821@dbmslab.nu1zzfd.mongodb.net/myDB --
collection=New_Collection --type json --file /home/Tanishagotadke/Documents/output1.json
```

Output:

A terminal window titled 'suryabhat@pop-os: ~' showing the execution of the mongoimport command. The command is: mongoimport mongodb+srv://surya4206:Surya#511821@dbmslab.nu1zzfd.mongodb.net/myDB --collection=New_Collection --type json --file /home/suryabhat/Documents/output1.json. The output shows the connection to the MongoDB server and the successful import of 8 documents.

```
suryabhat@pop-os: ~$ mongoimport mongodb+srv://surya4206:Surya#511821@dbmslab.nu1zzfd.mongodb.net/myDB --collection=New_Collection --type json --file /home/suryabhat/Documents/output1.json
2023-01-18T12:51:48.074+0530   connected to: mongodb+srv://[**REDACTED**]@dbmslab.nu1zzfd.mongodb.net/myDB
2023-01-18T12:51:48.230+0530   8 document(s) imported successfully. 0 document(s) failed to import.
suryabhat@pop-os: ~$
```