VISVESVARAYA TECHNOLOGICAL UNIVERSITY
"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Database Management Systems (22CS3PCDBM)

Submitted by

DIKSHYA ARYAL (1BM21CS058)

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 **Dikshya Aryal (1BM21CS058),** 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 2023. 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. Nandhini Vineeth

Assistant professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak

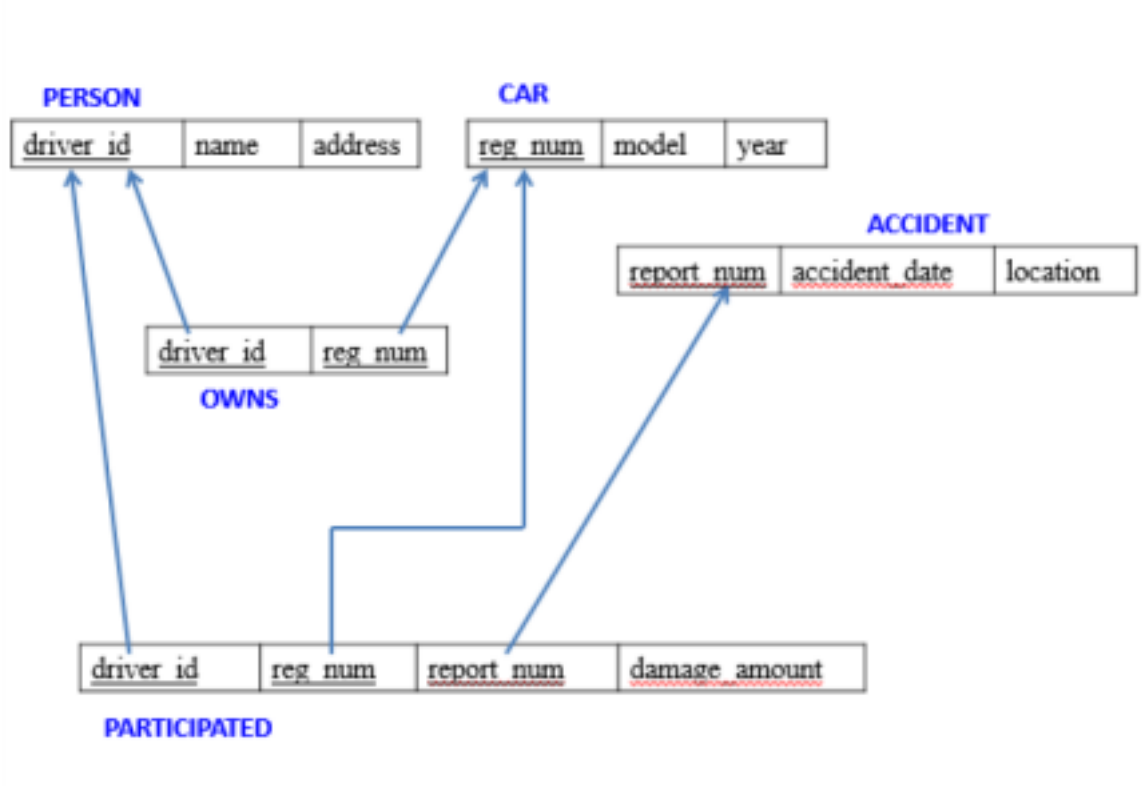
Professor and Head
Department of CSE
BMSCE, Bengaluru

Index

Insurance database

Questions and schema diagram

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 Accident date and location
4. 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.
5. Add a new accident to the database.



Create database and table with structure of tables

```
create database 058_insurance;

use  058_insurance;

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

Result Grid | Filter Rows: | Export: | Wra

Field	Type	Null	Key	Default	Extra
driver_id	varchar(10)	NO	PRI	NULL	
driver_name	varchar(20)	YES		NULL	
address	varchar(30)	YES		NULL	

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

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

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

```
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),
foreign key (reg_num) references car(reg_num));
desc OWNS;
```

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

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

```
create table Accident(
report_num varchar(30),
accident_date varchar(20),
location varchar(20),
primary key (report_num));
desc Accident;
```

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

Field	Type	Null	Key	Default	Extra
report_num	varchar(30)	NO	PRI	NULL	
accident_date	varchar(20)	YES		NULL	
location	varchar(20)	YES		NULL	

```
 create table participated(
driver_id varchar(10),
reg_num varchar(10),
report_num varchar(30),
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));
desc  participated;
```

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

Field	Type	Null	Key	Default	Extra
driver_id	varchar(10)	NO	PRI	NULL	
reg_num	varchar(10)	NO	PRI	NULL	
report_num	varchar(30)	NO	PRI	NULL	
damage_amount	int	YES		NULL	

<u>Inserting values into the tables</u>

```
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","BullTempleRoad");

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:

report_num	accident_date	location
11	2003-01-01	Mysore R...
12	2004-02-02	South end...
13	2003-01-21	Bull Templ...
14	2008-02-17	Mysore R...
15	2004-03-05	Kanakpur...
NULL	NULL	NULL

insert into person values ('A01','Richaed','Srinivas nagar');

insert into person values('A02','Pradeep','Rajaji nagar');

insert into person values('A03','Smith','Aahok nagar'),

('A04','venu','N R Colony'),

('A05','Jhon','Hanumanth Nagar');

select * from person;

Result Grid | Filter Rows:

driver_id	driver_name	address
A01	Richaed	Srinivas nagar
A02	Pradeep	Rajaji nagar
A03	Smith	Aahok nagar
A04	venu	N R Colony
A05	Jhon	Hanumanth Nagar
NULL	NULL	NULL

insert into car values

('KA052250','Indica',1990),

('KA031181','Lancer',1998),

('KA095477','Toyota',1998),

('KA053408','Honda',2008),

('KA041702','Audi',2005);

select * from car;

Result Grid | Filter Rows:

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

insert into OWNS values('A01','KA052250'),

('A02','KA031181'),

('A03','KA095477'),

('A04','KA053408'),

('A05','KA041702');

select * from OWNS;

Result Grid | Filter Rows: | Edit:

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

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("A01",'KA041702',15,5000);

select * from participated;

Result Grid | Filter Rows: | Edit:

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

Queries

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

select *from car

order by year asc;

Result Grid | Filter Rows: | Edit

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

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

select count(report_num) from car c, participated p

where c.reg_num=p.reg_num and c.model='Lancer";

Result Grid | Filter Rows:

count(report_num)
1

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

select count(distinct driver id) from participated a, accident b

where a.reportnum=b.reportn u m and b.accidentdate like '2008%"

Result Grid | Filter Rows:

count(distinct driver_id)
1

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

select *from participated order by damageamount desc;

Result Grid | Filter Rows: | Edit:

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

5. Find the average damage amount.

select avg(damage_amount) from participated;

Result Grid | Filter Rows: | Export:

avg(damage_amount)
18500.0000

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

delete from participated

where damage amount < (select t.avg1 from (select avg (damage amount) as avg1

from participated) t);

select *from participated;

Result Grid | Filter Rows: | Edit:

driver_id	reg_num	report_num	damage_amount
A02	KA053408	12	50000
A03	KA095477	13	25000

More queries on insurance database

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

Select * from participated order by (damage_amout) desc;

driver_id	reg_num	report_num	damage_amount
A02	031181	12	50000
A03	095477	13	25000
A01	052250	11	10000
A05	041702	15	5000
A04	053408	14	3000
NULL	NULL	NULL	NULL

2. Find the average damage amount

select avg(damage_amount) from participated;

Result Grid | Filter Rows:

avg(damage_amout)
13600.0000

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

Delete from participated where damage_amount<(select p.amt from(select avg(damage_amount)as amt from participated) p);

select * from participated;

Result Grid | Filter Rows: | Edit:

driver_id	reg_num	report_num	damage_amount
A02	031181	12	50000
A03	095477	13	25000
NULL	NULL	NULL	NULL

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

```
select name from person,participated
where person.driver_id=participated.driver_id and damage_amount>(select
avg(damage_amount) from participated);
```

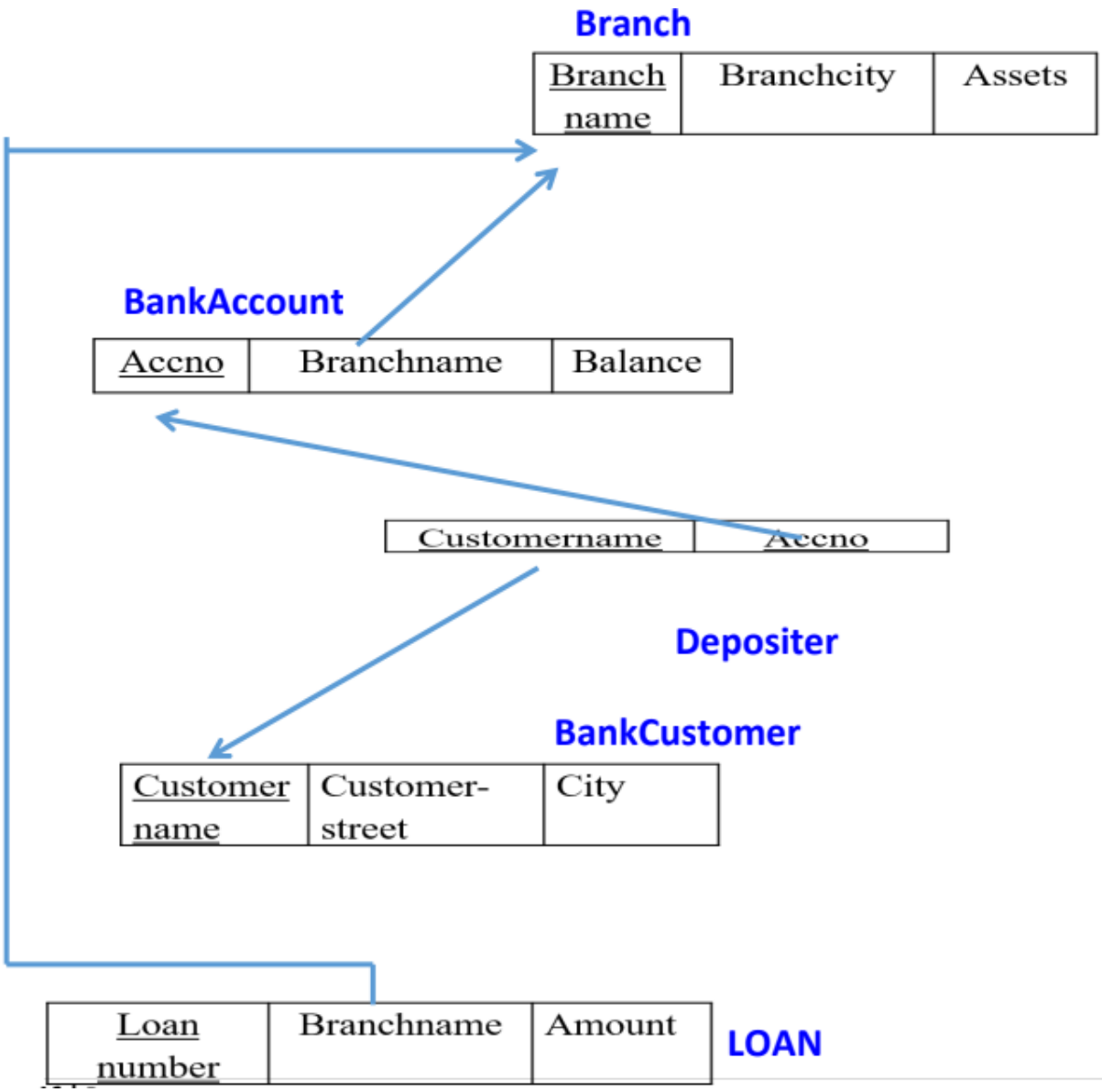
Result Grid | Filter Rows:

name
Pradeep
Smith

Bank Database

Question And Schema Diagram

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



Create Database and table with structures of table

```
create database 1bm21cs058_bankDb;
use 1bm21cs058_bankDb;

create table branch( branch_name varchar(20), branch_city
varchar(10), assets real, PRIMARY KEY(branch_name) );

create table bankCustomer( customer_name varchar(20),
customer_street varchar(20), customer_city varchar(15),
PRIMARY KEY(customer_name) );

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 );

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 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 );
```

Desc branch;

Field	Type	Null	Key	Default	Extra
branch_name	varchar(20)	NO	PRI	NULL	
branch_city	varchar(10)	YES		NULL	
assets	double	YES		NULL	

Desc bankCustomer;

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

Field	Type	Null	Key	Default	Extra
customer_name	varchar(20)	NO	PRI	NULL	
customer_street	varchar(20)	YES		NULL	
customer_city	varchar(15)	YES		NULL	

Desc loan;

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

Field	Type	Null	Key	Default	Extra
loan_no	int	NO	PRI	NULL	
branch_name	varchar(20)	YES	MUL	NULL	
amount	double	YES		NULL	

Desc bankAccount;

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

Field	Type	Null	Key	Default	Extra
accno	int	NO	PRI	NULL	
branch_name	varchar(20)	YES	MUL	NULL	
balance	double	YES		NULL	

Desc depositer;

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

Field	Type	Null	Key	Default	Extra
customer_name	varchar(20)	YES	MUL	NULL	
accno	int	YES	MUL	NULL	

<u>Inserting values to the table</u>

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);

select * from branch;

Result Grid | Filter Rows: | Edit:

branch_name	branch_city	assets
sbi_chamrajpet	bangalore	50000
sbi_jantarMantar	delhi	20000
sbi_parliamentRoad	delhi	10000
sbi_residencyRoad	bangalore	10000
sbi_shivajiRoad	bombay	20000
NULL	NULL	NULL

```
insert into bankAccount values(1,'sbi_chamrajpet',2000);
insert  into bankAccount values(2,'sbi_residencyRoad',5000);
insert  into bankAccount values(3,'sbi_shivajiRoad',6000);
insert into  bankAccountvalues(4,'sbi_parliamentRoad',9000);
insert into  bankAccount values(5,'sbi_jantarMantar',8000);
insert into  bankAccount values(6,'sbi_shivajiRoad',4000);
insert into  bankAccount values(8,'sbi_residencyRoad',4000);
insert into  bankAccountvalues(9,'sbi_parliamentRoad',3000);
insert into  bankAccountvalues(10,'sbi_residencyRoad',5000);
insert into  bankAccount values(11,'sbi_jantarMantar',2000);
select * from  bankAccount;
```

Result Grid | Filter Rows: | Edit:

accno	branch_name	balance
1	sbi_chamrajpet	2000
2	sbi_residencyRoad	5000
3	sbi_shivajiRoad	6000
4	sbi_parliamentRoad	9000
5	sbi_jantarMantar	8000
6	sbi_shivajiRoad	4000
8	sbi_residencyRoad	4000
9	sbi_parliamentRoad	3000
10	sbi_residencyRoad	5000
11	sbi_jantarMantar	2000
NULL	NULL	NULL

bankAccount 7

```
insert into bankCustomervalues('avinash','bull_temple_road','bangalore');
insert into bankCustomervalues('dinesh','bannergatta_road','bangalore');
insert into bankCustomervalues('mohan','nationalCollege_road','bangalore');
insert into bankCustomer values('nikil','akbar_road','delhi'); insert into
bankCustomervalues('ravi','prithviraj_road','delhi');
select * from bankCustomer;
```

Result Grid | Filter Rows: | Edit:

customer_name	customer_street	customer_city
avinash	bull_temple_road	bangalore
dinesh	bannergatta_road	bangalore
mohan	nationalCollege_road	bangalore
nikil	akbar_road	delhi
ravi	prithviraj_road	delhi
NULL	NULL	NULL

```
insert into depositer values('avinash',1);
insert into depositer values('dinesh',2);
insert into depositer values('nikil',4);
insert  into depositer values('ravi',5);
 insert into  depositer values('avinash',8);
 insert into  depositer values('nikil',9);
insert into  depositer values('dinesh',10);
insert into  depositer values('nikil',11);
select * from depositer;
```

Result Grid | Filter Rows:

customer_name	accno
avinash	1
dinesh	2
nikil	4
ravi	5
avinash	8
nikil	9
dinesh	10
nikil	11

```
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_jantarMantar',5000);
select * from  loan;
```

Result Grid | Filter Rows:

loan_no	branch_name	amount
1	sbi_chamrajpet	1000
2	sbi_residencyRoad	2000
3	sbi_shivajiRoad	3000
4	sbi_parliamentRoad	4000
5	sbi_jantarMantar	5000
NULL	NULL	NULL

Queries

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

   select branch_name, concat(assets/100000,'lakhs') as assesst_in_lakhs from branch;

Result Grid | Filter Rows:

branch_name	assesst_in_lakhs
sbi_chamrajpet	0.5lakhs
sbi_jantarMantar	0.2lakhs
sbi_parliamentRoad	0.1lakhs
sbi_residencyRoad	0.1lakhs
sbi_shivajiRoad	0.2lakhs

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

   select d.customer_name as CUSTOMER_NAME from bankAccount b,depositor d where b.branch_name='sbi_residencyRoad' and b.accno=d.accno group by d.customer_name having count(d.accno)>=2;

Result Grid | Filter Rows:

CUSTOMER_NAME
dinesh

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

```
create view sum_of_loan as select branch_name,sum(balance) from  bankAccount group
 by branch_name;
 select * from sum_of_loan;
```

Result Grid | Filter Rows:

branch_name	sum(balance)
sbi_chamrajpet	2000
sbi_jantarMantar	10000
sbi_parliamentRoad	12000
sbi_residencyRoad	14000
sbi_shivajiRoad	10000

More queries in bank database

insert into bankaccount values(12,"SBI_MatriMarg",2000); insert into branch values("SBI_MatriMarg","Delhi",200000); insert into depositer values("Nikil",12); create table borrower(customername varchar(50), loannumber int, foreign key(customername) references bankcustomer(customername), foreign key(loannumber) references loan(loannumber)); insert into borrower values("Avinash",1),("Dinesh",2),("Mohan",3),("Nikil",4),("Ravi",5);

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
▶	Nikil

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);

3.Find all customers who have both an account and a loan at the 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="Banglore" 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 b.branchname from branch b where b.assets> all (select sum(b.assets) from branch b where b.branchcity='Banglore');

branchname
SBI_MantriMarg
NULL

5.Demonstrate how you delete all account tuples at every branch located in a specific city (Ex. Bombay). update bankaccount set balance=(balance+(balance*0.05));
select * from bankaccount;

accno	branchname	balance
1	SBI_Chamrajpet	2100
2	SBI_ResidencyRoad	5250
3	SBI_ShivajiRoad	6300
4	SBI_ParlimentRoad	9450
5	SBI_Jantarmantar	8400
6	SBI_ShivajiRoad	4200
8	SBI_ResidencyRoad	4200
9	SBI_ParlimentRoad	3150
10	SBI_ResidencyRoad	5250
11	SBI_Jantarmantar	2100
12	SBI_MatriMarg	2100
NULL	NULL	NULL

6.Update the Balance of all accounts by 5%

delete b.*,ba.* from branch b, bankaccount ba,loan l Where b.branchcity="Banglore" and b.branchname=ba.branchname And l.branchname=ba.branchname;
select * from branch; select * from bankaccount; select * from loan;

branchname	branchcity	assets
SBI_Jantarmantar	Delhi	20000
SBI_MatriMarg	Delhi	200000
SBI_ParlimentRoad	Delhi	10000
SBI_ShivajiRoad	Bombay	20000
NULL	NULL	NULL

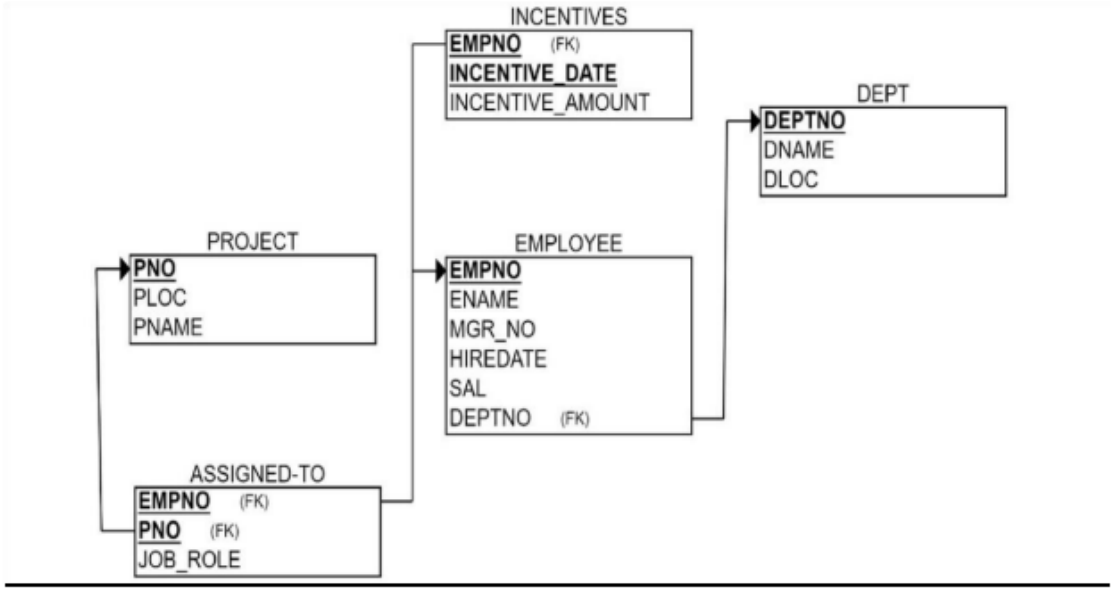
accno	branchname	balance
4	SBI_ParlimentRoad	9450
5	SBI_Jantarmantar	8400
9	SBI_ParlimentRoad	3150
11	SBI_Jantarmantar	2100
12	SBI_MatriMarg	2100
NULL	NULL	NULL

loannumber	branchname	amount
3	SBI_ShivajiRoad	3000
4	SBI_ParlimentRoad	4000
5	SBI_Jantarmantar	5000
NULL	NULL	NULL

Employee Database

Question and Schema Diagram

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 IDs 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.



Create database and table with structures of table

create database employee;
use employee;

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

Result Grid | Filter Rows: | Export: | Wrap

Field	Type	Null	Key	Default	Extra
deptno	int	NO	PRI	NULL	
dname	varchar(50)	YES		NULL	
dloc	varchar(50)	YES		NULL	

create table employee
(empno int, ename varchar(50), mgrno int, hiredate date, sal int, deptno int, primary key(empno), foreign key(deptno) references dept(deptno) on update cascade on delete cascade);
desc employee;

Result Grid | Filter Rows: | Export: | Wrap

Field	Type	Null	Key	Default	Extra
empno	int	NO	PRI	NULL	
ename	varchar(50)	YES		NULL	
mgrno	int	YES		NULL	
hiredate	date	YES		NULL	
sal	int	YES		NULL	
deptno	int	YES	MUL	NULL	

create table incentive
(empno int, incentivedate date, incentiveamount int, primary key(incentivedate),
foreign key(empno) references employee(empno)
on update cascade on delete cascade);
desc incentive;

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

Field	Type	Null	Key	Default	Extra
empno	int	YES	MUL	NULL	
incentivedate	date	NO	PRI	NULL	
incentiveamount	int	YES		NULL	

```
create table project
(pno int, ploc varchar(50), pname varchar(50), primary
key(pno));
desc project;
```

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

Field	Type	Null	Key	Default	Extra
pno	int	NO	PRI	NULL	
ploc	varchar(50)	YES		NULL	
pname	varchar(50)	YES		NULL	

```
create table assignedto
( empno int,pno int, jobrole varchar(50),
 foreign  key(empno) references employee(empno),
 foreign  key(pno) references project(pno)
on update  cascade on delete cascade);
desc assignedto;
```

Result Grid | Filter Rows: | Export:

Field	Type	Null	Key	Default	Extra
empno	int	YES	MUL	NULL	
pno	int	YES	MUL	NULL	
jobrole	varchar(50)	YES		NULL	

Inserting values to the table

insert into dept values(1,"Admin","Banglore"),
(2,"Sales","Bangolre"),
(3,"Finance","Hyderbad"),
(4,"Marketing","Mysore"),
(5,"Shipping","Hyderbad"),
(6,"Purchasing","Mysore");

Result Grid | Filter Rows:

deptno	dname	dloc
1	Admin	Banglore
2	Sales	Bangolre
3	Finance	Hyderbad
4	Marketing	Mysore
5	Shipping	Hyderbad
6	Purchasing	Mysore
NULL	NULL	NULL

insert into employee values(1,"Avinash",3,"2000-02-14",25000,1),
(2,"Balaji",3,"1999-05-11",31000,3),
(3,"Dinesh",NULL,"1992-01-26",46000,2),
(4,"Chandan",3,"2001-05-21",28000,4),
(5,"Aravind",2,"1998-09-22",17000,5),
(6,"Amal",3,"2003-02-14",25000,6);

Result Grid | Filter Rows: | Edit:

empno	ename	mgrno	hiredate	sal	deptno
1	Avinash	3	2000-02-14	25000	1
2	Balaji	3	1999-05-11	31000	3
3	Dinesh	NULL	1992-01-26	46000	2
4	Chandan	3	2001-05-21	28000	4
5	Aravind	2	1998-09-22	17000	5
6	Amal	3	2003-02-14	25000	6
NULL	NULL	NULL	NULL	NULL	NULL

insert into incentive values(1,"2005-03-23",5000),
(3,"2001-08-23",50000),
(5,"2011-04-02",1500);

Result Grid | Filter Rows:

empno	incentivedate	incentiveamount
3	2001-08-23	50000
1	2005-03-23	5000
5	2011-04-02	1500
NULL	NULL	NULL

insert into project values(11,"Banglore","Documentation"),
(12,"Banglore","Selling"),
(13,"Hyderbad","Accounting"),
(14,"Mysore","Advertising"),
(15,"Hyderbad","Transportation"),
(16,"Mysore","Purchasing"),
(17,"Hubli","Presentatiom");

Result Grid | Filter Rows:

pno	ploc	pname
11	Banglore	Documentation
12	Banglore	Selling
13	Hyderbad	Accounting
14	Mysore	Advertising
15	Hyderbad	Transportation
16	Mysore	Purchasing
17	Hubli	Presentatiom
NULL	NULL	NULL

insert into assignedto values(1,11,"Administration");
insert into assignedto values (2,12,"Salesman");
insert into assignedto values (3,13,"Accounts"); insert into assignedto values (4,14,"Advertising"); insert into assignedto values (5,15,"Transporting"); insert into assigned to values(6,16,"Purchasing”);

Result Grid | Filter Rows:

empno	pno	jobrole
1	11	Administration
2	12	Salesman
3	13	Accounts
4	14	Advertising
5	15	Transporting
6	16	purchasing

Queries

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

select empno from assignedto e where e.pno=any(select p.pno from project p where ploc="Banglore" or ploc="Hyderbad" or ploc="Mysore");

Result Grid

empno
1
2
3
4
5
6

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

```
select e.empno from employee e
where e.empno not in
(select i.empno from incentive i);
```

Result Grid

empno
2
4
6
NULL

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

select e.ename ename, e.empno empno, d.dname dname, a.jobrole jobrole, d.dloc dloc,p.ploc ploc

from project p, dept d, employee e, assignedto a

where e.empno=a.empno and p.pno=a.pno and e.deptno=d.deptno andp.ploc=d.dloc;

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

ename	empno	dname	jobrole	dloc	ploc
Avinash	1	Admin	Administration	Banglore	Banglore
Chandan	4	Marketing	Advertising	Mysore	Mysore
Aravind	5	Shipping	Transporting	Hyderbad	Hyderbad
Amal	6	Purchasing	purchasing	Mysore	Mysore

More Queries on Employee Database

1. List the name of the managers with the maximum employees select e.ename from employee e,Employee f where e.empno = f.mgr_no group by e.empno having count(*)=(select max(mycount) from (select count(*) mycount fromEmployee group by mgr_no) a);

Result Grid | Filter Rows:

ename
Avinash

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

select * from employee m where m.empno in (select mgr_no from employee) and m.salary>(select avg(n.salary) from Employee n where n.mgr_no=m.empno)

Result Grid | Filter Rows: | Edit:

empno	ename	mgr_no	hiredate	salary	deptno
1000	Avinash	1002	2010	200000	1
1005	Rahul	1000	2021	75000	6
NULL	NULL	NULL	NULL	NULL	NULL

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

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

Result Grid | Filter Rows:

ename
Avinash
Chandan
Rahul

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

select * from Employee where empno= (select iii.empno from incentives iii where iii.incentives_amount=(select max(ii.incentives_amount) from incentives ii where ii.incentives_amount<(select max(i.incentives_amount) from incentives i where i.incentives_date between "2019-01-01" and "2019-12-31") and incentives_date between "2019-01-01" and "2019-12-31"));

Result Grid | Filter Rows: | Export:

empno	ename	mgr_no	hiredate	salary	deptno
1003	Dinesh	1000	2019-02-01	50000	4

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

select e.ename from Employee e where e.Deptno=(select Deptno from Employee where e.mgr_no=empno);

Result Grid | Filter Rows:

ename
Balaji

Supplier Database

Question and Schema Diagram

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.

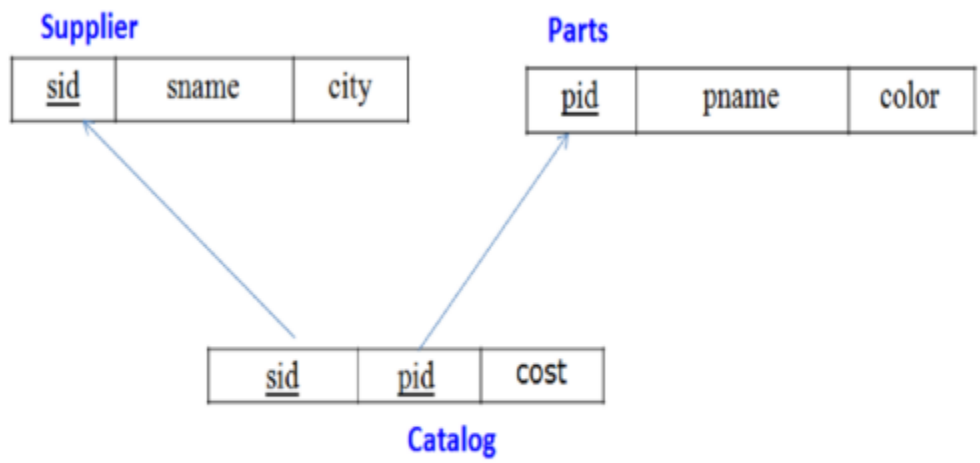
4. Find the snames of suppliers who supply every part.

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.



Create database and table with structures of table

create database supplier__cs058;

use supplier__cs058;

create table supplier(

sid varchar(20),

sname varchar(20),

city varchar(20),

primary key(sid)

);

Des supplier;

Result Grid | Filter Rows: | Export:

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

create table parts(pid

varchar(20), pname

varchar(20), color

varchar(20), primary

key(pid)

);

Desc parts;

Result Grid | Filter Rows: | Export:

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

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 supplier(sid)

);

Desc catlog;

Result Grid | Filter Rows: | Export:

Field	Type	Null	Key	Default	Extra
sid	varchar(20)	NO	PRI	NULL	
pid	varchar(20)	NO	PRI	NULL	
cost	varchar(20)	YES		NULL	

<u>Insert values in the table</u>

insert into supplier values(10001,'acme widget','bangalore'); insert into supplier values(10002,'johns','kolkata'); insert into supplier values(10003,'vimal','mumbai'); insert into supplier values(10004,'reliance','delhi');

Result Grid | Filter Rows:

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');

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');

Result Grid | Filter Rows:

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

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);

Result Grid | Filter Rows:

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

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

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

Result Grid | Filter Rows:

pname
book
pen
pencil
mobile
charger

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)

Result Grid | Filter Rows:

sname
acme widget

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 supllier c on c.sid=a.sid )a
where color='red';
```

Result Grid | Filter Rows:

sname
acme widget
johns

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
supllier 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 supllier C on B.sid=C.sid where lower(c.sname)<>'acme widget');
```

Result Grid | Filter Rows:

pname
mobile
charger

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

```
select  pid,sname from
Where (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;
```

Result Grid | Filter Rows:

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

Flight Database

Questions and Schema diagram

1.Create database table and insert appropriate data.

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

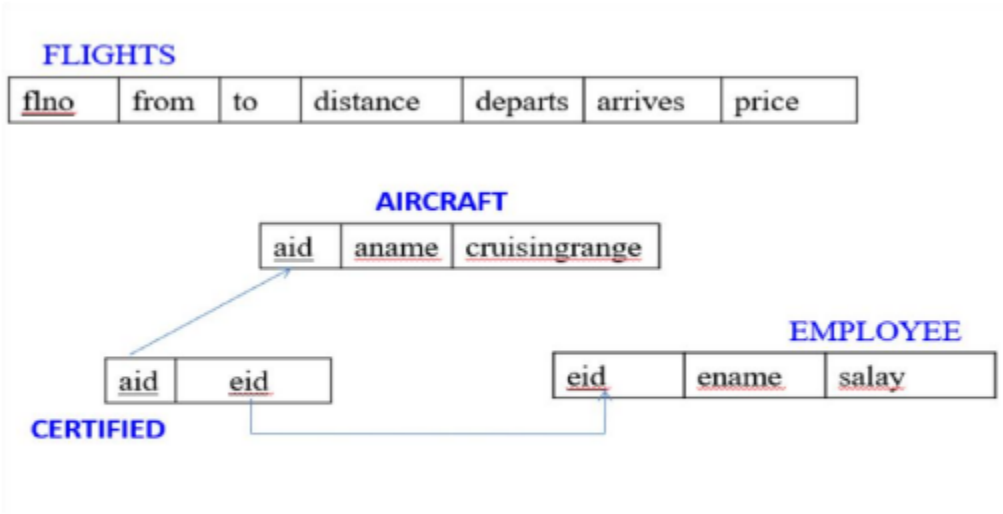
3.For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.

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

5.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.

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

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



Create database and table with structures of tables

create database airline_flight;

use airline_flight ;

create table FLIGHTS(flno int, departs_from varchar(50), arrives_to varchar(30), distance int, departs time, arrives time, price int, primary key (flno));

Desc FLIGHTS;

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

Field	Type	Null	Key	Default	Extra
flno	int	NO	PRI	NULL	
departs_from	varchar(50)	YES		NULL	
arrives_to	varchar(30)	YES		NULL	
distance	int	YES		NULL	
departs	time	YES		NULL	
arrives	time	YES		NULL	
price	int	YES		NULL	

create table Aircraft (aid int, aname varchar(50), cruising_range int, primary key (aid));

desc Aircraft;

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

Field	Type	Null	Key	Default	Extra
aid	int	NO	PRI	NULL	
aname	varchar(50)	YES		NULL	
cruising_range	int	YES		NULL	

create table EMPLOYEE (eID int, ename varchar(20), salary real,

primary key (eID));

desc EMPLOYEE;

Result Grid | Filter Rows: | Export:

Field	Type	Null	Key	Default	Extra
eID	int	NO	PRI	NULL	
ename	varchar(20)	YES		NULL	
salary	double	YES		NULL	

create table certified (aid int, eID varchar(50), primary key (aid, eID), foreign key (aid) REFERENCES Aircraft (aid), foreign key (eID) REFERENCES EMPLOYEE (eID));

Desc certified;

Insert into table values

INSERT INTO FLIGHTS VALUES (1,'Bengaluru','New Delhi',500,'6:00','9:00',50000);

INSERT INTO FLIGHTS VALUES (2,'Bengaluru','chennai',300,'7:00','8:30',3000);

INSERT INTO FLIGHTS VALUES (3,'trivandrum','New Delhi',800,'8:00','11:30',6000);

INSERT INTO FLIGHTS VALUES (4,'Bengaluru','Frankfurt',10000,'6:00','23:30',50000);

INSERT INTO FLIGHTS VALUES (5,'kolkata','New Delhi',2400,'11:00','3:30',9000);

INSERT INTO FLIGHTS VALUES (6,'Bengaluru','Frankfurt',8000,'9:00','23:00',40000);

Result Grid | Filter Rows: | Edit: | Export/Import:

fno	departs_from	arrives_to	distance	departs	arrives	price
1	Bengaluru	New Delhi	500	06:00:00	09:00:00	50000
2	Bengaluru	chennai	300	07:00:00	08:30:00	3000
3	trivandrum	New Delhi	800	08:00:00	11:30:00	6000
4	Bengaluru	Frankfurt	10000	06:00:00	23:30:00	50000
5	kolkata	New Delhi	2400	11:00:00	03:30:00	9000
6	Bengaluru	Frankfurt	8000	09:00:00	23:00:00	40000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

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

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

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

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

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

Result Grid | Filter Rows:

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

insert into certified values(101,2);

insert into certified values(101,4);

insert into certified values(101,5);

insert into certified values(101,6);

insert into certified values(102,1);

insert into certified values(102,3);

insert into certified values(102,5);

insert into certified values(103,2);

insert into certified values(103,3);

insert into certified values(103,5);

insert into certified values(103,6);

insert into certified values(104,6);

insert into certified values(104,1);

insert into certified values(104,3);

insert into certified values(105,3);

insert into EMPLOYEE values(101,'Avinash',50000);

insert into EMPLOYEE values(102,'Lokesh',60000);
insert into EMPLOYEE values(103,'Rakesh',70000);
insert into EMPLOYEE values(104,'Santhosh',82000);
insert into EMPLOYEE values(105,'Tilak',5000);

Result Grid | Filter Rows:

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

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);

Result Grid | Filter Rows

aname
Airbus
JetAirways
Airbus

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

SELECT e.ename FROM EMPLOYEE e WHERE e.salary< (SELECT MIN(f.price) FROM FLIGHTS f WHERE f.departs_from='Bengaluru' AND f.arrives_to='Frankfurt');

Result Grid

ename
Tilak

4. 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.

SELECTc.eID,MAX(cruising_range)
FROM certified c,Aircraft a
WHERE c.aid=a.aid GROUP BY
c.eid HAVING COUNT(*)>2;

Result Grid Filter Rows:

eID	MAX(cruising_range)
102	4500
104	2200
101	5000
103	4500

5. 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,certified c,EMPLOYEE e WHERE a.aid=c.aid AND c.eID=e.eID AND a.cruising_range>1000 GROUP BY a.aid,a.aname;

Result Grid Filter Rows:

aid	aname	AVG(e.salary)
1	Airbus	71000
4	Indigo	50000
5	Boeing	60000
6	Airbus	67333.33333333333

6. 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';

Result Grid | Filter

ename
Avinash
Rakesh
Lokesh

7. 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.cruising_range> (SELECT MIN(f.distance) FROM FLIGHTS f WHERE f.departs_from='Bengaluru' AND f.arrives_to='new delhi');

Result Grid | Fil

aid
1
2
3
4
5
6
NULL

NO SQL

Questions

1. Create a collection by name Customers with the following attributes. (Cust_id, Acc_Bal, Acc_Type).
2. Insert at least 5 values into the table.
3. Write a query to display those records whose total account balance is greater than 1200 of account type 'Z' for each customer_id.
4. Determine Minimum and Maximum account balance for each

customer_id. 5. Export the created collection into local file system.

6. Drop the table.
7. Import a given csv dataset from local file system into mongodb collection.

Create table and insert appropriate value

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

2. Insert appropriate values db.createCollect("Student");
db.Student.insert({rollno:1,age:21,cont:9876,email:"antara.de@gmail.com"});
db.Student.insert({rollno:2,age:22,cont:9976,email:"anushka.de@gmail.com"})
;
db.Student.insert({rollno:3,age:21,cont:5576,email:"anubhav.de@gmail.com"})
; db.Student.insert({rollno:4,age:20,cont:4476,email:"pani.de@gmail.com"});
db.Student.insert({rollno:5,age:23,cont:2276,email:"rekha.de@gmail.com"});

```
Connecting to:          mongodb+srv://<credentials>@cluster1.f6zdnam.mongodb.net/myfirstDatabase?appName=mongosh+1.6.2
Using MongoDB:          5.0.14 (API Version 1)
Using Mongosh:          1.6.2

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Atlas atlas-hu2siy-shard-0 [primary] myFirstDatabase> db.createCollection("Student");
{ ok: 1 }
Atlas atlas-hu2siy-shard-0 [primary] myFirstDatabase> db.Student.insert({RollNo:1,Age:21,Cont:9876,email:"antara.de9@gmail.com"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63cfe9fdd71db8b41f31818c") }
}
Atlas atlas-hu2siy-shard-0 [primary] myFirstDatabase> db.Student.insert({RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63cfea00d71db8b41f31818d") }
}
Atlas atlas-hu2siy-shard-0 [primary] myFirstDatabase> db.Student.insert({RollNo:3,Age:21,Cont:5576,email:"anubhav.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63cfea13d71db8b41f31818e") }
}
Atlas atlas-hu2siy-shard-0 [primary] myFirstDatabase> db.Student.insert({RollNo:4,Age:20,Cont:4476,email:"pani.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63cfea1ed71db8b41f31818f") }
}
Atlas atlas-hu2siy-shard-0 [primary] myFirstDatabase> db.Student.insert({RollNo:10,Age:23,Cont:2276,email:"rekha.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63cfea25d71db8b41f318190") }
}
Atlas atlas-hu2siy-shard-0 [primary] myFirstDatabase>
```

Queries

3.Write query to update Email-Id of a student with roll no 10

db.Student.update({rollno:10},{$set:{email:"abhinav@gmail.com"}})

```
Atlas atlas-hu2siy-shard-0 [primary] myFirstDatabase> 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
}
```

4. Replace the student name from “ABC” to “FEM” of roll no

db.Student.update({rollno:11,name:"ABC"},{$set:{name:"FEM"}})

```
Atlas atlas-hu2siy-shard-0 [primary] myFirstDatabase> db.Student.insert({RollNo:11,Age:22,Name:
... "ABC",Cont:2276,email:"rea.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63cfeaa6d71db8b41f318191") }
}
```

```
Atlas atlas-hu2siy-shard-0 [primary] myFirstDatabase> db.Student.update({RollNo:11,Name:"ABC"},{$set:{Name:"FEM"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

5. Export the created table into local file system mongoexport mongodb+srv://dikshya:<password>@cluster0.xbmgopf.mongodb.net/Lab_9 -collection=Student -- out C:\Users\dikshya\Desktop\export\output.json

6. Drop the table db.Student.drop();

```
Atlas atlas-hu2siy-shard-0 [primary] myFirstDatabase> db.Student.drop();
true
Atlas atlas-hu2siy-shard-0 [primary] myFirstDatabase>
```

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

mongoimportmongodb+srv://dikshya:< password>@cluster0.xbmgopf.mong odb.net/Lab_9 --collection=new_Student – type json –file C:\Users\dikshya \Desktop\export\output.json