

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT
on

Database Management Systems (22CS3PCDBM)

Submitted by

**SAI KRUTHIN CR
(1BM22CS232)**

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-2023 to Feb-2024

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



CERTIFICATE

This is to certify that the Lab work entitled “Database Management Systems (22CS3PCDBM)” carried out by **SAI KRUTHIN CR (1BM21CS232)**, 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.

VIKRANTH B.M.

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	15-12-2024	Insurance Database	4
2	22-12-2024	More Queries on Insurance Database	11
3	29-12-2024	Bank Database	21
4	12-1-2024	More Queries on Bank Database	30
5	19-1-2024	Employee Database	41
6	26-1-2024	More Queries on Employee Database	51
7	2-2-2024	Supplier Database	62

Week 1:Insurance Database

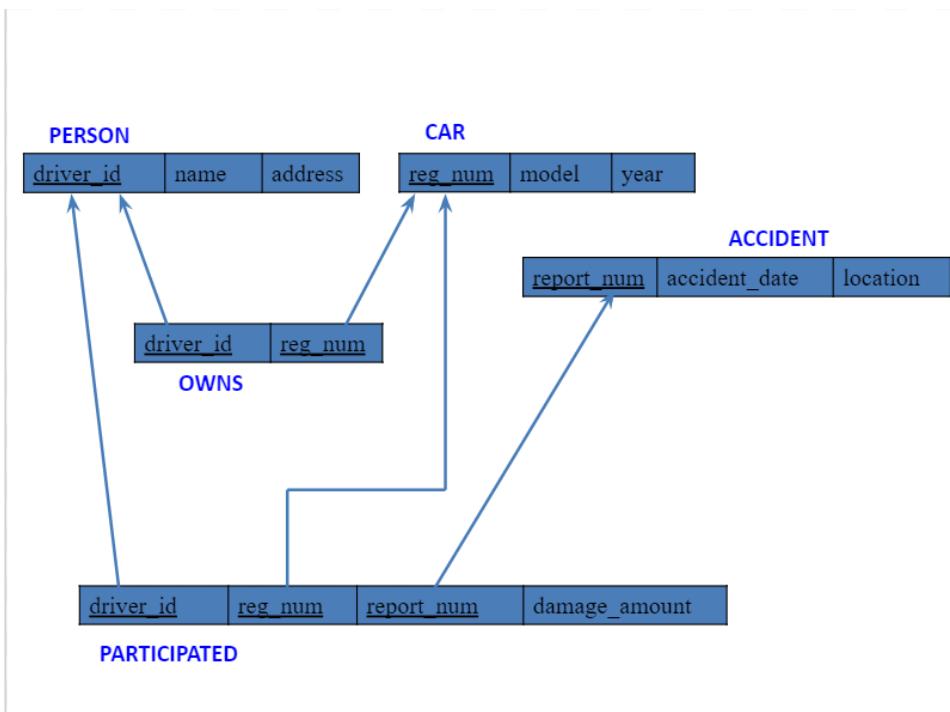
Question

- Create the above tables by properly specifying the primary keys and the foreign keys.
- Enter at least five tuples for each relation
- Display Accident date and location
- 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.
- Add a new accident to the database.

To Do

- Display Accident date and location
- Display driver id who did accident with damage amount greater than or equal to Rs.2500.

Schema Diagram



Create database

```
create database 1BM22CS232_insurance;  
use 1BM22CS232_insurance;
```

Create table

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

```
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 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)  
);
```

```
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),
foreign key(reg_num) references car(reg_num),
foreign key(report_num) references accident(report_num);

```

Structure of the table

desc person;

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

desc car;

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

desc owns;

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

desc accident;

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

desc participated;

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

Inserting Values to the table

```
insert into accident values(11,'2003-01-01','Mysore road' );
insert into accident values(12,'2004-02-02','South end circle' );
insert into accident values(13,'2003-01-21','Bull temple road' );
insert into accident values(14,'2008-02-17','Mysore road' );
insert into accident values(15,'2004-03-05','Kanakpura road' );
```

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

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

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

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

select *from person;

	driver_id	name	address
▶	A01	Richard	Srinivas nagar
	A02	Pradeep	Rajaji nagar
	A03	Smith	Ashok nagar
	A04	Venu	N R. Colony
	A05	Jhon	Hanumanth nagar
*	NULL	NULL	NULL

select *from car;

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

select *from owns;

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

select *from accident;

	report_num	accident_date	location
▶	11	2003-01-01	Mysore road
	12	2004-02-02	South end circle
	13	2003-01-21	Bull temple road
	14	2008-02-17	Mysore road
	15	2004-03-05	Kanakpura road
*	NULL	NULL	NULL

select *from participated;

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

Queries

Display Accident date and location

SQL> select accident_date,location from accident;

	accident_date	location
▶	2003-01-01	Mysore road
	2004-02-02	South end circle
	2003-01-21	Bull temple road
	2008-02-17	Mysore road
	2004-03-05	Kanakpura road

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.

SQL> update participated set damage_amount=25000
where reg_num='KA053408' and report_num=12;

select *from participated;

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

Add a new accident to the database.

SQL> insert into accident values(16,'2008-03-08','Domlur');

select *from accident;

	report_num	accident_date	location
11	2003-01-01	Mysore road	
12	2004-02-02	South end circle	
13	2003-01-21	Bull temple road	
14	2008-02-17	Mysore road	
15	2004-03-05	Kanakpura road	
16	2008-03-08	Domlur	
NULL	NULL	NULL	

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

SQL>select count(distinct driver_id)
from participated a, accident b
where a.report_num=b.report_num and b.accident_date like '%08%';

	count(distinct driver_id)
▶	1

Display Accident date and location.

SQL> select accident_date,location from accident;

accident_date	location
2003-01-01	Mysore road
2004-02-02	South end circle
2003-01-21	Bull temple road
2008-02-17	Mysore road
2004-03-05	Kanakpura road
2008-03-08	Domlur

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

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

driver_id
A02
A03

Week 2:More Queries Insurance Database

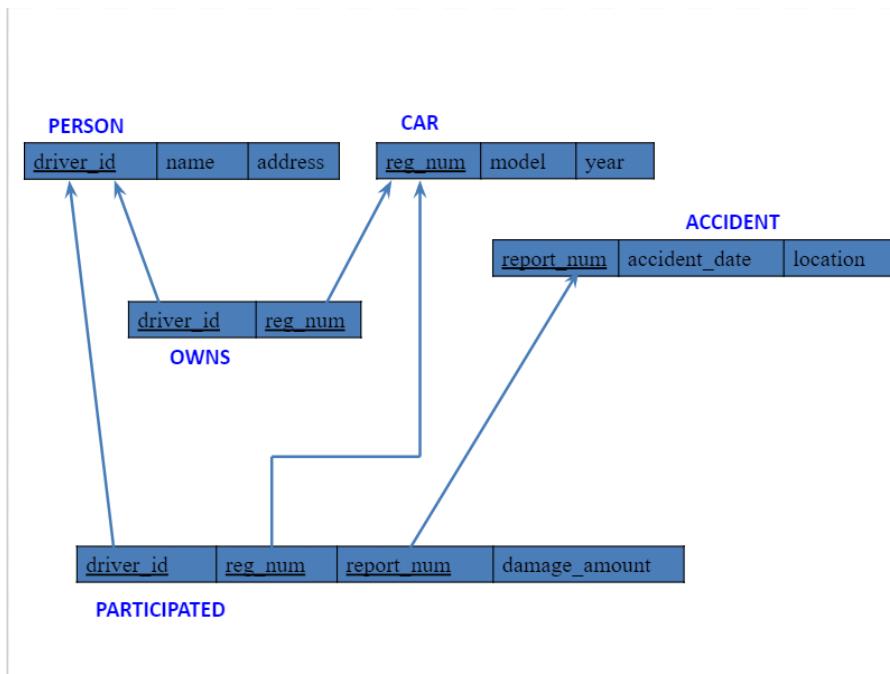
Question

- 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.

To Do

- List the entire participated relation in descending order of damage amount.
- Find the average damage amount.
- Delete the tuple whose damage amount is below 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 the maximum damage amount

Schema Diagram



Create database

```
create database 1BM22CS232_insurance;
```

```
use 1BM22CS232_insurance;
```

Create table

```
create table person (
  driver_id varchar(10),
  name varchar(30),
```

```
address varchar(30),
primary key(driver_id)
);
desc person;

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 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)
);

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),
foreign key(reg_num) references car(reg_num),
foreign key(report_num) references accident(report_num)
);
```

Structure of the table

```
desc person;
```

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

Result 1	x
Detailed view	

```
desc car;
```

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

Result 2	x
Detailed view	

```
desc owns;
```

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

Result 4	x
Detailed view	

```
desc accident;
```

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

Result 3	x
Detailed view	

desc participated;

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

Result 5 ×

Inserting Values to the table

```
insert into accident values(11,'2003-01-01','Mysore road' );
insert into accident values(12,'2004-02-02','South end circle' );
insert into accident values(13,'2003-01-21','Bull temple road' );
insert into accident values(14,'2008-02-17','Mysore road' );
insert into accident values(15,'2004-03-05','Kanakpura road' );
```

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

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

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

```
insert into participated values('A01','KA052250',11,10000);
insert into participated values('A02','KA053408',12,50000);
insert into participated values('A03','KA095477',13,25000);
```

```
insert into participated values('A04','KA031181',14,3000);
insert into participated values('A05','KA041702',15,5000);
```

select *from person;

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

person 1 x car 2 owns 3 accident 4 participated 5

select *from car;

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

person 1 car 2 x owns 3 accident 4 participated 5

Output:

select *from owns;

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

person 1 car 2 owns 3 x accident 4 participated 5

Output:

select *from accident;

	report_num	accident_date	location
▶	11	2003-01-01	Mysore road
	12	2004-02-02	South end circle
	13	2003-01-21	Bull temple road
	14	2008-02-17	Mysore road
	15	2004-03-05	Kanakpura road
*	NULL	NULL	NULL

person 1 car 2 owns 3 accident 4 x participated 5

select *from participated;

Result Grid		Filter Rows:	Edi				
	driver_id	reg_num	report_num	damage_amount			
▶	A01	KA052250	11	10000			
	A02	KA053408	12	50000			
	A03	KA095477	13	25000			
	A04	KA031181	14	3000			
	A05	KA041702	15	5000			
*	NULL	NULL	NULL	NULL			

person 1 car 2 owns 3 accident 4 participated 5 ×

Output:

Queries

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

SQL> select *from car

order by year asc;

Result Grid | Filter Rows: Ed

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

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

```
SQL> 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

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

```
SQL> select count(distinct driver_id)  
from participated a, accident b
```

where a.report_num=b.report_num and b.accident_date like '2008%';

Result Grid | Filter Rows: Export: Wrap Cell Content

	count(distinct driver_id)
▶	1

TO DO

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

```
SQL> select *from participated  
order by damage_amount desc;
```

Result Grid				
	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

Find the average damage amount.

```
SQL> select avg(damage_amount)  
from participated;
```

The screenshot shows a database query result grid titled "Result Grid". It has one row with the header "avg(damage_amount)" and one data row containing the value "18600.0000". There are buttons for "Filter Rows" and "Export". Below the grid, it says "Result 14".

avg(damage_amount)
18600.0000

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

```
SQL> delete from participated  
where damage_amount < (select t.avg1 from (select avg(damage_amount) as avg1 from participated) t);  
select * from participated;
```

The screenshot shows a database query result grid titled "Result Grid". It has four columns: driver_id, reg_num, report_num, and damage_amount. There are three rows: A02 (50000), A03 (25000), and a blank row with NULL values. There are buttons for "Edit" and "Filter Rows". Below the grid, it says "participated 7".

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

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

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

```
from participated  
);
```

Result Grid | Filter Rows: []

name
▶ Pradeep

Result 12 ×

Result Grid | Filter Rows: []

avg(damage_amount)
▶ 37500.0000

Result 11 ×

Find the maximum damage amount

```
SQL>select max(damage_amount)  
from participated;
```

Result Grid | Filter Rows: []

max(damage_amount)
▶ 50000

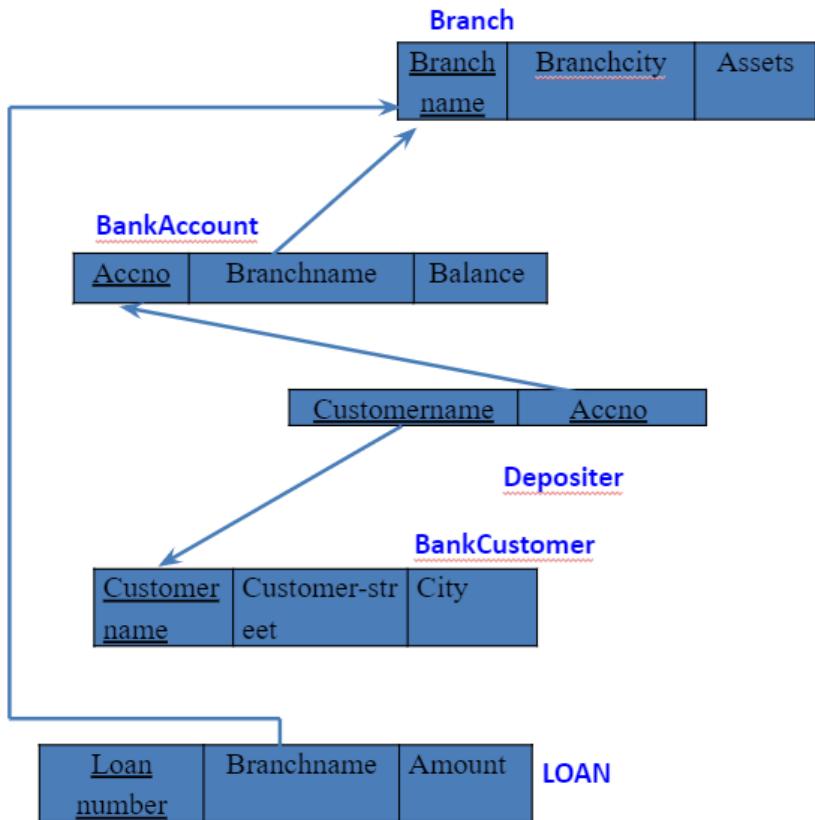
Result 13 ×

Week 3:Bank Database

Question

- Create the above tables by properly specifying the primary keys and the foreign keys.
- Enter at least five tuples for each relation.
- Display the branch name and assets from all branches in lakhs of rupees and rename the assets column to 'assets in lakhs'.
- Find all the customers who have at least two accounts at the same branch (ex. SBI_ResidencyRoad).
- Create a view which gives each branch the sum of the amount of all the loans at the branch.

Schema Diagram



Create database

```
create database 1BM22CS232_bank;
use 1BM22CS232_bank;
```

Create table

```
create table branch(
branchname varchar(20),
branchcity varchar(20),
assets int,
primary key(branchname)
);
```

```
create table bankcustomer(
```

```
customername varchar(20),  
customerstreet varchar(20),  
customercity varchar(20),  
primary key(customername)  
);
```

```
create table bankacc(  
accno int,  
branchname varchar(20),  
balance int,  
primary key(accno),  
foreign key(branchname) references branch(branchname)  
);
```

```
create table depositer(  
customername varchar(20),  
accno int,  
primary key(customername, accno),  
foreign key(customername) references bankcustomer(customername),  
foreign key(accno) references bankacc(accno)  
);
```

```
create table loan(  
loannumber int,  
branchname varchar(20),  
amount int,  
primary key(loannumber),  
foreign key(branchname) references branch(branchname)  
);
```

Structure of the table

```
desc branch;
```

	Field	Type	Null	Key	Default	Extra
▶	branchname	varchar(20)	NO	PRI	NULL	
	branchcity	varchar(20)	YES		NULL	
	assets	int	YES		NULL	

```
desc bankcustomer;
```

	Field	Type	Null	Key	Default	Extra
▶	customername	varchar(20)	NO	PRI	NULL	
	customerstreet	varchar(20)	YES		NULL	
	customercity	varchar(20)	YES		NULL	

desc bankacc;

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:						
	Field	Type	Null	Key	Default	Extra
▶	accno	int	NO	PRI	NULL	
	branchname	varchar(20)	YES	MUL	NULL	
	balance	int	YES		NULL	

Result 3 ×

desc depositer;

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:						
	Field	Type	Null	Key	Default	Extra
▶	customername	varchar(20)	NO	PRI	NULL	
	accno	int	NO	PRI	NULL	

Result 4 ×

desc loan;

	Field	Type	Null	Key	Default	Extra
▶	loannumber	int	NO	PRI	NULL	
	branchname	varchar(20)	YES	MUL	NULL	
	amount	int	YES		NULL	

Inserting Values to the table

```
insert into branch values('SBI_Chamarajpet','bangalore',50000);
insert into branch values('SBI_Residencyroad','bangalore',10000);
insert into branch values('SBI_Shivajinagar','bombay',20000);
insert into branch values('SBI_Parliamentroad','delhi',10000);
insert into branch values('SBI_Jantarmantar','delhi',20000);
```

```
insert into bankcustomer values('Avinash','BullTempleRoad','bangalore');
insert into bankcustomer values('Dinesh','BannerghattaRoad','bangalore');
insert into bankcustomer values('Mohan','NationalCollegeRoad','bangalore');
insert into bankcustomer values('Nikhil','AkbarRoad','delhi');
insert into bankcustomer values('Ravi','PrithvirajRoad','delhi');

insert into bankacc values(1,'SBI_Chamarajpet',2000);
insert into bankacc values(2,'SBI_Residencyroad',5000);
insert into bankacc values(3,'SBI_Shivajinagar',6000);
insert into bankacc values(4,'SBI_Parliamentroad',9000);
insert into bankacc values(5,'SBI_Jantarmantar',8000);
insert into bankacc values(6,'SBI_Shivajinagar',4000);
insert into bankacc values(8,'SBI_Residencyroad',4000);
insert into bankacc values(9,'SBI_Parliamentroad',3000);
insert into bankacc values(10,'SBI_Residencyroad',5000);
insert into bankacc values(11,'SBI_Jantarmantar',2000);

insert into depositer values('Avinash',1);
insert into depositer values('Dinesh',2);
insert into depositer values('Nikhil',4);
insert into depositer values('Ravi',5);
insert into depositer values('Avinash',8);
insert into depositer values('Nikhil',9);
insert into depositer values('Dinesh',10);
insert into depositer values('Nikhil',11);

insert into loan values(1,'SBI_Chamarajpet',1000);
insert into loan values(2,'SBI_Residencyroad',2000);
insert into loan values(3,'SBI_Shivajinagar',3000);
insert into loan values(4,'SBI_Parliamentroad',4000);
insert into loan values(5,'SBI_Jantarmantar',5000);

select *from branch;
```

	branchname	branchcity	assets
▶	SBI_Chamarajpet	bangalore	50000
	SBI_Jantarmantar	delhi	20000
	SBI_Mantrimarg	delhi	200000
	SBI_Parliamentroad	delhi	10000
	SBI_Residencyroad	bangalore	10000
	SBI_Shivajinagar	bombay	20000
*	NULL	NULL	NULL

select *from bankacc;

	accno	branchname	balance
▶	1	SBI_Chamarajpet	2000
	2	SBI_Residencyroad	5000
	3	SBI_Shivajinagar	6000
	4	SBI_Parliamentroad	9000
	5	SBI_Jantarmantar	8000
	6	SBI_Shivajinagar	4000
	8	SBI_Residencyroad	4000
	9	SBI_Parliamentroad	3000
	10	SBI_Residencyroad	5000
	11	SBI_Jantarmantar	2000
	12	SBI_Mantrimarg	2000
	NULL	NULL	NULL

```
select *from bankcustomer;
```

	customername	customerstreet	customercity
▶	Avinash	BullTempleRoad	bangalore
	Dinesh	BannerghattaRoad	bangalore
	Mohan	NationalCollegeRoad	bangalore
	Nikhil	AkbarRoad	delhi
	Ravi	PrithvirajRoad	delhi
*	NULL	NULL	NULL

```
select *from depositer;
```

	customername	accno
▶	Avinash	1
	Dinesh	2
	Nikhil	4
	Ravi	5
	Avinash	8
	Nikhil	9
	Dinesh	10
	Nikhil	11
	Nikhil	12
*	NULL	NULL

```
select *from loan;
```

	loannumber	branchname	amount
▶	1	SBI_Chamarajpet	1000
	2	SBI_Residencyroad	2000
	3	SBI_Shivajinagar	3000
	4	SBI_Parlimentroad	4000
	5	SBI_Jantarmantar	5000
*	NULL	NULL	NULL

Queries

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

```
alter table branch
```

```

rename column assets to assets_in_lakhs;
select branchname,(assets_in_lakhs/100000)
from branch;

```

	branchname	assets_in_lakhs
▶	SBI_Chamarajpet	0.5000
	SBI_Jantarmantar	0.2000
	SBI_Parliamentroad	0.1000
	SBI_Residencyroad	0.1000
	SBI_Shivajinagar	0.2000

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

```

select b.branchname,d.customername
from bankacc b,depositor d
where d.accno=b.accno
group by b.branchname, d.customername
having count(d.customername)>1;

```

	branchname	customername
▶	SBI_Residencyroad	Dinesh
	SBI_Parliamentroad	Nikhil

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 branchname, SUM(amount)
from loan
group by branchname;
select *from sum_of_loan;

```

	branchname	SUM(amount)
▶	SBI_Chamarajpet	1000
	SBI_Jantarmantar	5000
	SBI_Parliamentroad	4000
	SBI_Residencyroad	2000
	SBI_Shivajinagar	3000

Update or add rupees 1000 to all balance for the customers who are residing in bangalore.

```
update bankacc
set balance=(balance+1000)
where accno in(
select accno
from depositer
where customername in (
select customername
from bankcustomer
where customercity='bangalore'));

select *from bankacc;
```

The screenshot shows the MySQL Workbench interface with a result grid titled 'bankacc 9'. The grid displays three columns: accno, branchname, and balance. The data is as follows:

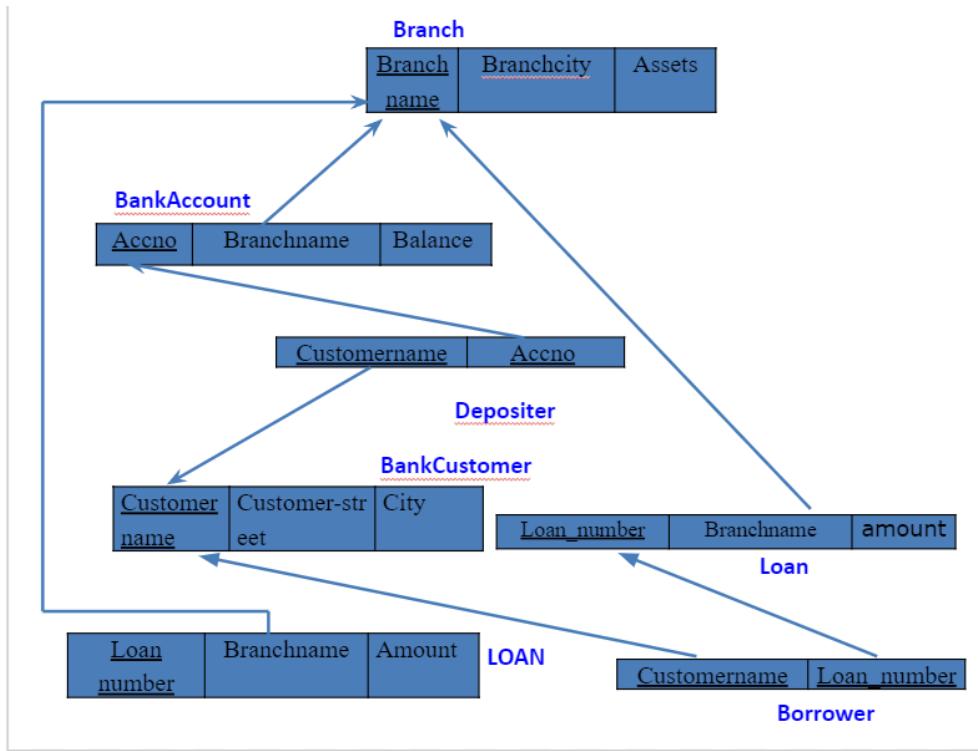
	accno	branchname	balance
▶	1	SBI_Chamarajpet	3000
	2	SBI_Residencyroad	6000
	3	SBI_Shivajinagar	6000
	4	SBI_Parlimentroad	9000
	5	SBI_Jantarmantar	8000
	6	SBI_Shivajinagar	4000
	8	SBI_Residencyroad	5000
	9	SBI_Parlimentroad	3000
	10	SBI_Residencyroad	6000
	11	SBI_Jantarmantar	2000
*	NULL	NULL	NULL

Week 4:More Queries on Bank Database

Question

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

Schema Diagram



Create database

```
create database 1BM22CS232_bank2;
use 1BM22CS232_bank2;
```

Create table

```
create table branch(
branchname varchar(20),
branchcity varchar(20),
assets int,
primary key(branchname)
);
```

```
create table bankcustomer(
customername varchar(20),
customerstreet varchar(20),
customercity varchar(20),
primary key(customername)
);
```

```
create table bankacc(
```

```
accno int,  
branchname varchar(20),  
balance int,  
primary key(accno),  
foreign key(branchname) references branch(branchname)  
on delete cascade  
on update cascade  
);
```

```
create table depositer(  
customername varchar(20),  
accno int,  
primary key(customername, accno),  
foreign key(customername) references bankcustomer(customername),  
foreign key(accno) references bankacc(accno)  
on delete cascade  
on update cascade  
);
```

```
create table loan(  
loannumber int,  
branchname varchar(20),  
amount int,  
primary key(loannumber),  
foreign key(branchname) references branch(branchname)  
on delete cascade  
on update cascade  
);
```

```
create table borrower(  
customername varchar(20),  
loannumber int,  
primary key(loannumber, customername),  
foreign key (customername) references bankcustomer(customername),  
foreign key (loannumber) references loan(loannumber)  
on delete cascade  
on update cascade
```

);

Structure of the table

desc branch;

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:						
	Field	Type	Null	Key	Default	Extra
▶	branchname	varchar(20)	NO	PRI	NULL	
	branchcity	varchar(20)	YES		NULL	
	assets	int	YES		NULL	

desc bankcustomer;

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:						
	Field	Type	Null	Key	Default	Extra
▶	customername	varchar(20)	NO	PRI	NULL	
	customerstreet	varchar(20)	YES		NULL	
	customercity	varchar(20)	YES		NULL	

desc bankacc;

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:						
	Field	Type	Null	Key	Default	Extra
▶	accno	int	NO	PRI	NULL	
	branchname	varchar(20)	YES	MUL	NULL	
	balance	int	YES		NULL	

desc depositer;

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:						
	Field	Type	Null	Key	Default	Extra
▶	customername	varchar(20)	NO	PRI	NULL	
	accno	int	NO	PRI	NULL	

```
desc loan;
```

Result Grid						
	Field	Type	Null	Key	Default	Extra
▶	loannumber	int	NO	PRI	NULL	
	branchname	varchar(20)	YES	MUL	NULL	
	amount	int	YES		NULL	

Result 5 ×

```
desc borrower;
```

Result Grid						
	Field	Type	Null	Key	Default	Extra
▶	customername	varchar(20)	NO	PRI	NULL	
	loannumber	int	NO	PRI	NULL	

Result 11 ×

Inserting Values to the table

```
insert into branch values('SBI_Chamarajpet','bangalore',50000);
insert into branch values('SBI_Residencyroad','bangalore',10000);
insert into branch values('SBI_Shivajinagar','bombay',20000);
insert into branch values('SBI_Parlimentroad','delhi',10000);
insert into branch values('SBI_Jantarmantar','delhi',20000);
insert into branch values('SBI_Mantrimarg','delhi',200000);
```

```
insert into bankcustomer values('Avinash','BullTempleRoad','bangalore');
insert into bankcustomer values('Dinesh','BannerghattaRoad','bangalore');
insert into bankcustomer values('Mohan','NationalCollegeRoad','bangalore');
insert into bankcustomer values('Nikhil','AkbarRoad','delhi');
insert into bankcustomer values('Ravi','PrithvirajRoad','delhi');
```

```
insert into bankacc values(1,'SBI_Chamarajpet',2000);
```

```
insert into bankacc values(2,'SBI_Residencyroad',5000);
insert into bankacc values(3,'SBI_Shivajinagar',6000);
insert into bankacc values(4,'SBI_Parliamentroad',9000);
insert into bankacc values(5,'SBI_Jantarmantar',8000);
insert into bankacc values(6,'SBI_Shivajinagar',4000);
insert into bankacc values(8,'SBI_Residencyroad',4000);
insert into bankacc values(9,'SBI_Parliamentroad',3000);
insert into bankacc values(10,'SBI_Residencyroad',5000);
insert into bankacc values(11,'SBI_Jantarmantar',2000);
insert into bankacc values(12,'SBI_Mantrimarg',2000);

insert into depositer values('Avinash',1);
insert into depositer values('Dinesh',2);
insert into depositer values('Nikhil',4);
insert into depositer values('Ravi',5);
insert into depositer values('Avinash',8);
insert into depositer values('Nikhil',9);
insert into depositer values('Dinesh',10);
insert into depositer values('Nikhil',11);
insert into depositer values('Nikhil',12);

insert into loan values(1,'SBI_Chamarajpet',1000);
insert into loan values(2,'SBI_Residencyroad',2000);
insert into loan values(3,'SBI_Shivajinagar',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('Nikhil',4);
insert into borrower values('Ravi',5);

select *from branch;
```

	branchname	branchcity	assets
▶	SBI_Chamarajpet	bangalore	50000
	SBI_Jantarmantar	delhi	20000
	SBI_Mantrimarg	delhi	200000
	SBI_Parliamentroad	delhi	10000
	SBI_Residencyroad	bangalore	10000
	SBI_Shivajinagar	bombay	20000
*	NULL	NULL	NULL

select *from bankacc;

	accno	branchname	balance
▶	1	SBI_Chamarajpet	2000
	2	SBI_Residencyroad	5000
	3	SBI_Shivajinagar	6000
	4	SBI_Parliamentroad	9000
	5	SBI_Jantarmantar	8000
	6	SBI_Shivajinagar	4000
	8	SBI_Residencyroad	4000
	9	SBI_Parliamentroad	3000
	10	SBI_Residencyroad	5000
	11	SBI_Jantarmantar	2000
	12	SBI_Mantrimarg	2000
*	NULL	NULL	NULL

select *from bankcustomer;

	customername	customerstreet	customercity
▶	Avinash	BullTempleRoad	bangalore
	Dinesh	BannerghattaRoad	bangalore
	Mohan	NationalCollegeRoad	bangalore
	Nikhil	AkbarRoad	delhi
	Ravi	PrithvirajRoad	delhi
*	NULL	NULL	NULL

select *from depositer;

	customername	accno
▶	Avinash	1
	Dinesh	2
	Nikhil	4
	Ravi	5
	Avinash	8
	Nikhil	9
	Dinesh	10
	Nikhil	11
	Nikhil	12
*	NULL	NULL

select *from loan;

	loannumber	branchname	amount
▶	1	SBI_Chamarajpet	1000
	2	SBI_Residencyroad	2000
	3	SBI_Shivajinagar	3000
	4	SBI_Parliamentroad	4000
	5	SBI_Jantarmantar	5000
*	NULL	NULL	NULL

select *from borrower;

	customername	loannumber
▶	Avinash	1
	Dinesh	2
	Mohan	3
	Nikhil	4
	Ravi	5
*	NULL	NULL

Queries

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

```
select distinct customername, count(distinct ba.branchname)
from depositer d, bankacc ba, branch b
where d.accno=ba.accno and ba.branchname=b.branchname and b.branchcity='delhi'
group by customername
having count(distinct b.branchname) = (select count(distinct x.branchname)
from branch x inner join bankacc y
on x.branchname=y.branchname
where branchcity='delhi');
```

	customername	count(distinct ba.branchname)
▶	Nikhil	3

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

```
select customername
from borrower
where customername not in(
select customername
from depositer
);
```

	customername
▶	Mohan

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

```
select customername  
from borrower  
where customername in(  
select customername  
from depositer  
where accno= any  
(  
select accno  
from bankacc ba inner join branch b  
on ba.branchname=b.branchname  
where b.branchcity='bangalore' )  
);
```

The screenshot shows a database interface with a 'Result Grid' tab selected. The grid displays a single column named 'customername'. Two rows are visible: 'Avinash' and 'Dinesh'. The row for 'Dinesh' is highlighted with a blue background.

customername
Avinash
Dinesh

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

```
select branchname  
from branch  
where assets>  
(select sum(assets)  
from branch  
where branchcity='bangalore');
```

	branchname
▶	SBI_Mantrimarg
*	NULL

branch 51 ×

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

```
delete from bankacc
where branchname=any
(select branchname
from branch
where branchcity='bombay');
select * from bankacc;
```

	accno	branchname	balance
▶	1	SBI_Chamarajpet	2000
	2	SBI_Residencyroad	5000
	4	SBI_Parliamentroad	9000
	5	SBI_Jantarmantar	8000
	8	SBI_Residencyroad	4000
	9	SBI_Parliamentroad	3000
	10	SBI_Residencyroad	5000
	11	SBI_Jantarmantar	2000
	12	SBI_Mantrimarg	2000
*	NULL	NULL	NULL

bankacc 17 ×

Output

6.Update the Balance of all accounts by 5%

```
update bankacc
set balance=(balance+0.05*balance);
select *from bankacc;
```

Result Grid | Filter Rows: | Edit:

	accno	branchname	balance
▶	1	SBI_Chamarajpet	2100
	2	SBI_Residencyroad	5250
	3	SBI_Shivajinagar	6300
	4	SBI_Parliamentroad	9450
	5	SBI_Jantarmantar	8400
	6	SBI_Shivajinagar	4200
	8	SBI_Residencyroad	4200
	9	SBI_Parliamentroad	3150
	10	SBI_Residencyroad	5250
	11	SBI_Jantarmantar	2100
	12	SBI_Mantrimarg	2100
*	NULL	NULL	NULL

bankacc 15 ×

Spot query

Q) Demonstrate how you delete all the branches located in bangalore.

delete from branch

where branchcity='bangalore';

Result Grid | Filter Rows: | Edit:

	branchname	branchcity	assets
▶	SBI_Jantarmantar	delhi	20000
	SBI_Mantrimarg	delhi	200000
	SBI_Parliamentroad	delhi	10000
	SBI_Shivajinagar	bombay	20000
*	NULL	NULL	NULL

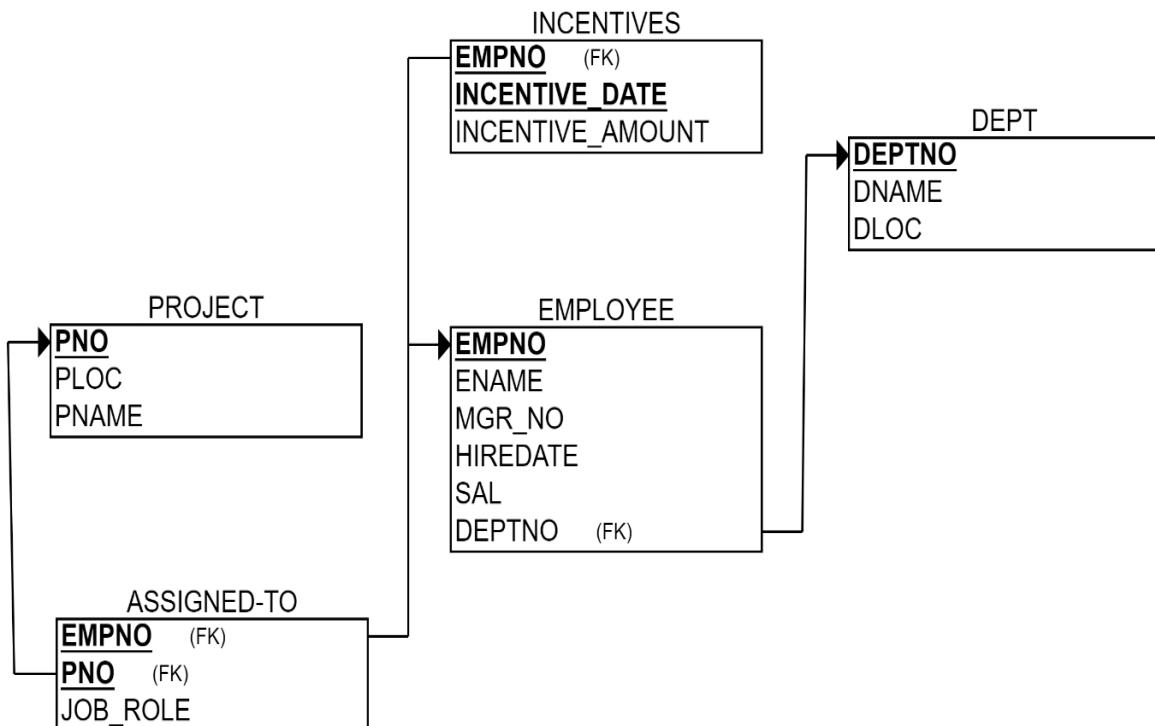
branch 18 ×

Week 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 1BM22CS232_employee;
use 1BM22CS232_employee;
```

Create table

```
create table dept(
deptno int,
```

```
    dname varchar(20),  
    dloc varchar(20),  
    primary key(deptno)  
);
```

```
create table employee(  
    empno int,  
    ename varchar(20),  
    mgr_no int,  
    hiredate date,  
    sal double,  
    deptno int,  
    primary key(empno),  
    foreign key (deptno) references dept(deptno)  
    on delete cascade  
    on update cascade  
);
```

```
create table incentives(  
    empno int,  
    incentive_date date,  
    incentive_amount float,  
    primary key(empno,incentive_date),  
    foreign key (empno) references employee(empno)  
    on delete cascade  
    on update cascade  
);
```

```
create table project(  
    pno int,  
    ploc varchar(20),  
    pname varchar(20),  
    primary key(pno)  
);
```

```
create table assigned_to(  
    empno int,  
    pno int,  
    job_role varchar(20),  
    primary key(empno,pno),
```

foreign key (empno) references employee(empno),
 foreign key (pno) references project(pno)
 on delete cascade
 on update cascade
);

Structure of the table

desc dept;

	Field	Type	Null	Key	Default	Extra
▶	deptno	int	NO	PRI	NULL	
	dname	varchar(20)	YES		NULL	
	dloc	varchar(20)	YES		NULL	

Result 7 ×

desc employee;

	Field	Type	Null	Key	Default	Extra
▶	empno	int	NO	PRI	NULL	
	ename	varchar(20)	YES		NULL	
	mgr_no	int	YES		NULL	
	hiredate	date	YES		NULL	
	sal	double	YES		NULL	
	deptno	int	YES	MUL	NULL	

Result 2 ×

desc incentives;

	Field	Type	Null	Key	Default	Extra
▶	empno	int	NO	PRI	NULL	
	incentive_date	date	NO	PRI	NULL	
	incentive_amount	float	YES		NULL	

Result 5 ×

desc project;

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

	Field	Type	Null	Key	Default	Extra
▶	pno	int	NO	PRI	NULL	
	ploc	varchar(20)	YES		NULL	
	pname	varchar(20)	YES		NULL	

Result 3 ×

Output

desc assigned_to;

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

	Field	Type	Null	Key	Default	Extra
▶	empno	int	NO	PRI	NULL	
	pno	int	NO	PRI	NULL	
	job_role	varchar(20)	YES		NULL	

Result 4 ×

Inserting Values to the table

```
insert into dept values(10,'cse','bangalore');
insert into dept values(20,'ise','bangalore');
insert into dept values(30,'aiml','hyderabad');
insert into dept values(40,'ece','mysore');
insert into dept values(50,'eee','delhi');
insert into dept values(60,'iem','chennai');
```

```
insert into employee values(11,'Rajesh',21,'2000-04-03',80000,10);
insert into employee values(12,'Ajay',11,'2003-04-06',70000,20);
insert into employee values(13,'Divya',11,'2006-03-07',60000,30);
insert into employee values(14,'Chandan',12,'2007-09-03',50000,40);
insert into employee values(15,'Bhavesh',13,'2009-11-13',40000,50);
insert into employee values(16,'Tarun',14,'2012-02-10',30000,60);
insert into employee values(17,'Brinda',14,'2009-05-12',50000,10);
insert into employee values(18,'Anil',15,'2015-01-01',30000,20);
insert into employee values(19,'Puja',15,'2020-10-21',60000,30);
insert into employee values(20,'Ram',16,'2021-09-17',45000,40);
```

```
insert into incentives values(11,'2002-09-08',40000);
insert into incentives values(12,'2005-07-10',33000);
```

```
insert into incentives values(13,'2008-01-21',7000);
insert into incentives values(14,'2014-08-05',8000);
insert into incentives values(15,'2017-09-13',5000);
insert into incentives values(17,'2021-03-17',6000);
insert into incentives values(18,'2021-04-16',8000);
insert into incentives values(19,'2021-08-11',9000);
```

```
insert into project values(121,'bangalore','proj1');
insert into project values(122,'bangalore','proj2');
insert into project values(123,'mysore','proj3');
insert into project values(124,'hyderabad','proj4');
insert into project values(125,'delhi','proj5');
insert into project values(126,'mumbai','proj6');
insert into project values(127,'calicut','proj7');
insert into project values(128,'calicut','proj8');
```

```
insert into assigned_to values(11,121,'manager');
insert into assigned_to values(12,122,'team_lead');
insert into assigned_to values(13,123,'analyst');
insert into assigned_to values(14,124,'team_lead');
insert into assigned_to values(15,125,'manager');
insert into assigned_to values(16,126,'programmer');
insert into assigned_to values(17,127,'team_lead');
insert into assigned_to values(19,128,'team_lead');
```

```
select *from dept;
```

	deptno	dname	dloc
►	10	cse	bangalore
	20	ise	bangalore
	30	aiml	hyderabad
	40	ece	mysore
	50	eee	delhi
*	60	iem	chennai
*	NULL	NULL	NULL

```
select *from employee;
```

The screenshot shows a database query results window with the following details:

- Result Grid:** The main area displays a grid of data from the 'employee' table.
- Columns:** The columns are labeled: empno, ename, mgr_no, hiredate, sal, and deptno.
- Data:** The grid contains 20 rows of data, starting with empno 11 and ending with empno 20. The last row is marked with an asterisk (*) and shows all columns as NULL.
- Toolbar:** At the top, there are buttons for Result Grid, Filter Rows, Edit, and Export/Import.
- Title Bar:** The title bar reads "employee 61 X".

	empno	ename	mgr_no	hiredate	sal	deptno
▶	11	Rajesh	21	2000-04-03	80000	10
	12	Ajay	11	2003-04-06	70000	20
	13	Divya	11	2006-03-07	60000	30
	14	Chandan	12	2007-09-03	50000	40
	15	Bhavesh	13	2009-11-13	40000	50
	16	Tarun	14	2012-02-10	30000	60
	17	Brinda	14	2009-05-12	50000	10
	18	Anil	15	2015-01-01	30000	20
	19	Puja	15	2020-10-21	60000	30
	20	Ram	16	2021-09-17	45000	40
*	HULL	HULL	HULL	HULL	HULL	HULL

```
select *from incentives;
```

Result Grid | Filter Rows: | Edit:

	empno	incentive_date	incentive_amount
▶	11	2002-09-08	40000
	12	2005-07-10	33000
	13	2008-01-21	7000
	14	2014-08-05	8000
	15	2017-09-13	5000
	17	2021-03-17	6000
	18	2021-04-16	8000
	19	2021-08-11	9000
*	NULL	NULL	NULL

incentives 62 ×

select *from project;

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

	pno	ploc	pname
▶	121	bangalore	proj1
	122	bangalore	proj2
	123	mysore	proj3
	124	hyderabad	proj4
	125	delhi	proj5
	126	mumbai	proj6
	127	calicut	proj7
	128	calicut	proj8
*	NULL	NULL	NULL

project 63 ×

Output

select *from assigned_to;

Result Grid | Filter Rows: | Edit:

	empno	pno	job_role
▶	11	121	manager
	12	122	team_lead
	13	123	analyst
	14	124	team_lead
	15	125	manager
	16	126	programmer
	17	127	team_lead
	19	128	team_lead
*	HULL	HULL	HULL

assigned_to64 ×

Output ::::::::::::

Queries

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

```
select a.empno
from assigned_to a, project p
where p.pno=a.pno and p.ploc in(
select ploc
from project
where ploc='bangalore' or ploc='hyderabad' or ploc='mysore'
);
```

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

	empno
▶	11
	12
	13
	14

Result 56 ×

Output ::::::::::::

Get Employee IDs of those employees who didn't receive incentives.

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

The screenshot shows the Oracle SQL Developer interface with the 'Result Grid' tab selected. The grid displays a single column named 'empno' with three rows: 20, 16, and NULL. The 'employee 57' tab is active, and the 'Output' tab is visible below it.

empno
20
16
NULL

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

```
select e.ename, d.dname, a.job_role  
from employee e, dept d, assigned_to a  
where e.deptno=d.deptno and a.empno=e.empno and e.empno in (  
select empno  
from incentives  
where incentive_amount = (select max(incentive_AMOUNT) from incentives where  
incentive_date between '2021-01-01'and '2021-12-31') );
```

The screenshot shows the Oracle SQL Developer interface with the 'Result Grid' tab selected. The grid displays six columns: ename, empno, dname, job_role, dloc, and ploc. It contains three rows for Rajesh, Ajay, and Bhavesh, all of whom work in the bangalore department and have their project location set to bangalore.

ename	empno	dname	job_role	dloc	ploc
Rajesh	11	cse	manager	bangalore	bangalore
Ajay	12	ise	team_lead	bangalore	bangalore
Bhavesh	15	eee	manager	delhi	delhi

Spot query

Find the employee name, dept name, job role of an employee who received maximum incentive in the year 2021.

```
select e.ename, d.dname, a.job_role  
from employee e, dept d, assigned_to a  
where e.deptno=d.deptno and a.empno=e.empno and e.empno in (  
select empno  
from incentives  
where incentive_amount = (select max(incentive_AMOUNT) from incentives where  
incentive_date between '2021-01-01'and '2021-12-31')  
);
```

The screenshot shows a database query results window. At the top, there are buttons for 'Result Grid' (highlighted in blue), 'Filter Rows:', 'Export:' (with icons for CSV and PDF), and 'Wrap Cell Content'. Below this is a table with three columns: 'ename', 'dname', and 'job_role'. A single row of data is shown: 'Puja' in 'ename', 'aiml' in 'dname', and 'team_lead' in 'job_role'. The table has a light gray background with white borders between rows and columns. At the bottom left, there is a 'Result 59' button with a close icon, and at the bottom right, there is an 'Output' field containing several dots.

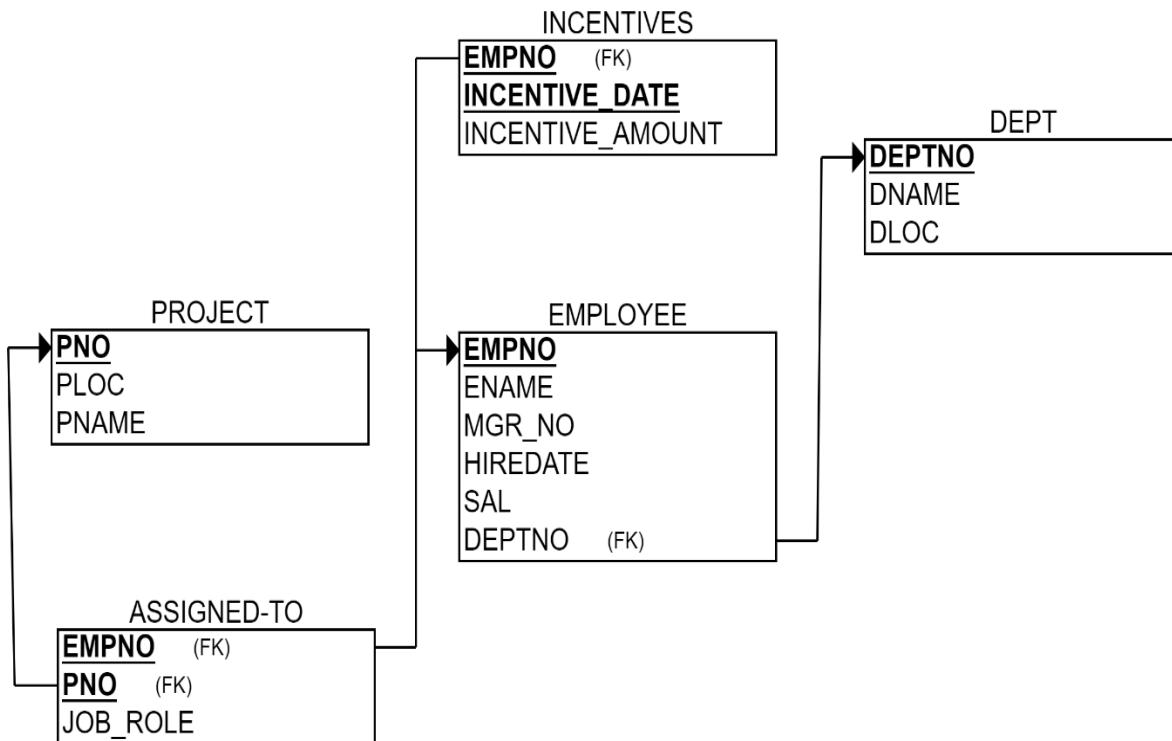
	ename	dname	job_role
▶	Puja	aiml	team_lead

Week 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



Create database

```
create database 1BM22CS232_employee2;
use 1BM22CS232_employee2;
```

Create table

```
create table dept(
deptno int,
dname varchar(20),
dloc varchar(20),
primary key(deptno)
);
create table employee(
empno int,
ename varchar(20),
mgr_no int,
hiredate date,
sal double,
deptno int,
primary key(empno),
foreign key (deptno) references dept(deptno)
on delete cascade
on update cascade
);
create table incentives(
empno int,
incentive_date date,
incentive_amount float,
primary key(empno,incentive_date),
foreign key (empno) references employee(empno)
on delete cascade
on update cascade
);
create table project(
pno int,
ploc varchar(20),
pname varchar(20),
primary key(pno)
);
create table assigned_to(
empno int,
pno int,
job_role varchar(20),
primary key(empno,pno),
foreign key (empno) references employee(empno),
```

```

foreign key (pno) references project(pno)
on delete cascade
on update cascade
);

```

Structure of the table

```
desc dept;
```

	Field	Type	Null	Key	Default	Extra
▶	deptno	int	NO	PRI	NULL	
	dname	varchar(20)	YES		NULL	
	dloc	varchar(20)	YES		NULL	

Result 7 ×

```
desc employee;
```

	Field	Type	Null	Key	Default	Extra
▶	empno	int	NO	PRI	NULL	
	ename	varchar(20)	YES		NULL	
	mgr_no	int	YES		NULL	
	hiredate	date	YES		NULL	
	sal	double	YES		NULL	
	deptno	int	YES	MUL	NULL	

Result 2 ×

```
desc incentives;
```

	Field	Type	Null	Key	Default	Extra
▶	empno	int	NO	PRI	NULL	
	incentive_date	date	NO	PRI	NULL	
	incentive_amount	float	YES		NULL	

Result 5 ×

desc project;

	Field	Type	Null	Key	Default	Extra
▶	pno	int	NO	PRI	NULL	
	ploc	varchar(20)	YES		NULL	
	pname	varchar(20)	YES		NULL	

Result 3 ×

Output:

desc assigned_to;

	Field	Type	Null	Key	Default	Extra
▶	empno	int	NO	PRI	NULL	
	pno	int	NO	PRI	NULL	
	job_role	varchar(20)	YES		NULL	

Result 4 ×

Inserting Values to the table

```
insert into dept values(10,'cse','bangalore');
insert into dept values(20,'ise','bangalore');
insert into dept values(30,'aiml','hyderabad');
insert into dept values(40,'ece','mysore');
insert into dept values(50,'eee','delhi');
insert into dept values(60,'iem','chennai');
```

```
insert into employee values(11,'Rajesh',21,'2000-04-03',80000,10);
insert into employee values(12,'Ajay',11,'2003-04-06',70000,20);
insert into employee values(13,'Divya',11,'2006-03-07',60000,30);
insert into employee values(14,'Chandan',12,'2007-09-03',50000,40);
insert into employee values(15,'Bhavesh',13,'2009-11-13',40000,50);
insert into employee values(16,'Tarun',14,'2012-02-10',30000,60);
insert into employee values(17,'Brinda',11,'2009-05-12',50000,10);
insert into employee values(18,'Anil',15,'2015-01-01',30000,20);
insert into employee values(19,'Puja',15,'2020-10-21',60000,30);
```

```
insert into employee values(20,'Ram',16,'2021-09-17',45000,40);  
insert into employee values(21,'Priya',22,'2002-03-13',85000,10);
```

```
insert into incentives values(11,'2012-09-08',40000);  
insert into incentives values(12,'2015-07-10',33000);  
insert into incentives values(13,'2019-01-21',7000);  
insert into incentives values(14,'2019-01-05',8000);  
insert into incentives values(15,'2019-01-13',5000);  
insert into incentives values(17,'2021-03-17',6000);  
insert into incentives values(18,'2021-04-16',8000);  
insert into incentives values(19,'2021-08-11',9000);
```

```
insert into project values(121,'bangalore','proj1');  
insert into project values(122,'bangalore','proj2');  
insert into project values(123,'mysore','proj3');  
insert into project values(124,'hyderabad','proj4');  
insert into project values(125,'delhi','proj5');  
insert into project values(126,'mumbai','proj6');  
insert into project values(127,'calicut','proj7');  
insert into project values(128,'calicut','proj8');
```

```
insert into assigned_to values(11,121,'manager');  
insert into assigned_to values(12,122,'team_lead');  
insert into assigned_to values(13,123,'analyst');  
insert into assigned_to values(14,124,'team_lead');  
insert into assigned_to values(15,125,'manager');  
insert into assigned_to values(16,126,'programmer');  
insert into assigned_to values(17,127,'team_lead');  
insert into assigned_to values(19,128,'team_lead');
```

```
select *from dept;
```

Result Grid | Filter Rows: | Edit:

	deptno	dname	dloc
▶	10	cse	bangalore
	20	ise	bangalore
	30	aiml	hyderabad
	40	ece	mysore
	50	eee	delhi
	60	iem	chennai
*	NULL	NULL	NULL

dept 6 ×

select *from employee;

Result Grid | Filter Rows: | Edit: Export/

	empno	ename	mgr_no	hiredate	sal	deptno
▶	11	Rajesh	21	2000-04-03	80000	10
	12	Ajay	11	2003-04-06	70000	20
	13	Divya	11	2006-03-07	60000	30
	14	Chandan	12	2007-09-03	50000	40
	15	Bhavesh	13	2009-11-13	40000	50
	16	Tarun	14	2012-02-10	30000	60
	17	Brinda	11	2009-05-12	50000	10
	18	Anil	15	2015-01-01	30000	20
	19	Puja	15	2020-10-21	60000	30
	20	Ram	16	2021-09-17	45000	40
	21	Priya	22	2002-03-13	85000	10
*	NULL	NULL	NULL	NULL	NULL	NULL

employee 7 ×

select *from incentives;

Result Grid | Filter Rows: | Edit:

	empno	incentive_date	incentive_amount
▶	11	2012-09-08	40000
	12	2015-07-10	33000
	13	2019-01-21	7000
	14	2019-01-05	8000
	15	2019-01-13	5000
	17	2021-03-17	6000
	18	2021-04-16	8000
	19	2021-08-11	9000
*	NULL	NULL	NULL

incentives 8 ×

```
select *from project;
```

	pno	ploc	pname
▶	121	bangalore	proj1
	122	bangalore	proj2
	123	mysore	proj3
	124	hyderabad	proj4
	125	delhi	proj5
	126	mumbai	proj6
	127	calicut	proj7
	128	calicut	proj8
*	HULL	HULL	HULL

project 9 ×

```
select *from assigned_to;
```

	empno	pno	job_role
▶	11	121	manager
	12	122	team_lead
	13	123	analyst
	14	124	team_lead
	15	125	manager
	16	126	programmer
	17	127	team_lead
	19	128	team_lead
*	HULL	HULL	HULL

assigned_to10 ×

Queries

List the name of the managers with the maximum employees

```
select emp.ename  
from employee emp  
where emp.empno=(  
select mgr_no  
from employee e  
group by mgr_no  
having count(empno) >= all(  
select (count(empno))  
from employee  
group by mgr_no ));
```

The screenshot shows a database query results window. At the top, there is a toolbar with a 'Result Grid' button, a refresh icon, and a 'Filter Rows:' input field. Below the toolbar is a table with a single row. The table has a header row with a single column labeled 'ename'. The data row contains a single cell with the value 'Rajesh'. At the bottom of the window, there is a status bar with the text 'employee 38' and a close button.

ename
Rajesh

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

```
select emp.ename  
from employee emp  
where emp.sal > any (  
select avg(e.sal)  
from employee e  
where emp.empno=e.mgr_no  
);
```

Result Grid				Filter Rows:
ename				
►	Rajesh			
	Ajay			
	Divya			
	Chandan			
	Priya			

employee 39

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

```
select emp.ename
from employee emp
where emp.ename = any(
select e2.ename
from employee e, employee e2
where e2.empno=e.mgr_no and e2.deptno = e.deptno and e.ename = any(
select e1.ename
from employee e1, employee e0
where e1.empno=e0.mgr_no and e1.deptno = e0.deptno
group by e1.mgr_no
having count(e1.empno)>1)
);
```

Result Grid				Filter Rows:
ename				
►	Priya			

employee 40

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

```
select i.empno, i.incentive_date, max(i.incentive_amount)second_max
from incentives i
where i.incentive_date between '2019-01-01' and '2019-01-31' and i.incentive_amount not in(
```

```

select max(incentive_amount)
from incentives
where incentive_date between '2019-01-01' and '2019-01-31';

```

	empno	incentive_date	second_max
▶	13	2019-01-21	7000

Result 41 ×

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

```

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

```

	ename	deptno
▶	Rajesh	10
	Brinda	10
	Manas	10
	Prem	10

Result 42 ×

Output ::::::::::::::::::::

Spot query-Find the employee details who got third maximum incentive in January 2019.

```

select i.empno, i.incentive_amount
from incentives i
where 3 = (
    select count(*)
    from incentives j
    where incentive_date between '2019-01-01' and '2019-01-31' and i.incentive_amount <=
        j.incentive_amount)
and incentive_date between '2019-01-01' and '2019-01-31';

```

Result Grid | Filter Rows: Export:

	empno	incentive_amount
▶	15	5000

incentives 1 ×

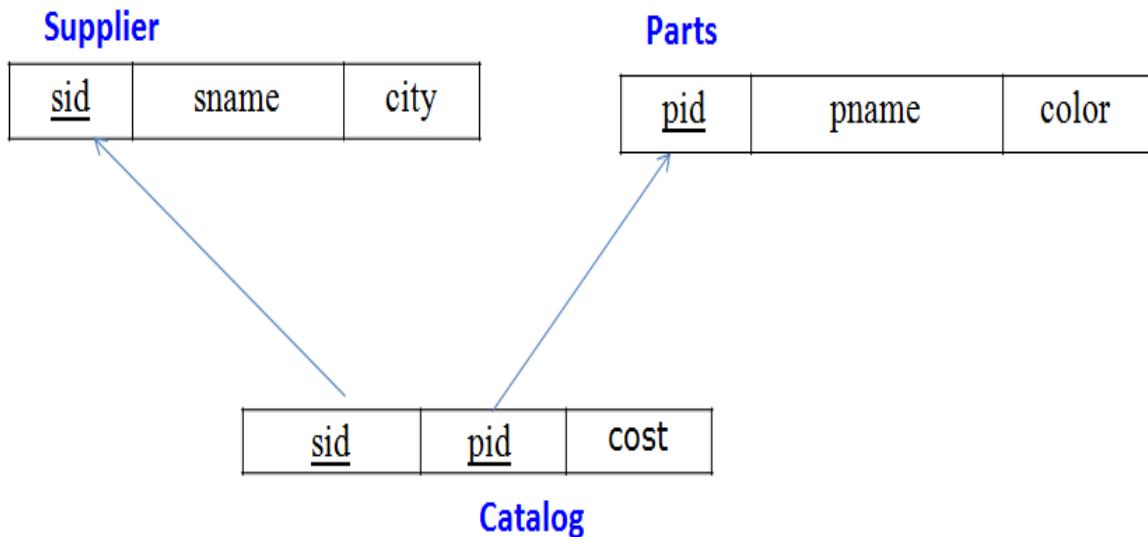
Output ::::::::::::

Week 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



Create database

```
create database 1BM22CS232_supplier;
use 1BM22CS232_supplier;
```

Create table

```
create table supplier(
    sid int,
    sname varchar(20),
```

```
city varchar(20),  
primary key(sid)  
);
```

```
create table parts(  
pid int,  
pname varchar(20),  
color varchar(10),  
primary key(pid)  
);
```

```
create table catalog(  
sid int,  
pid int,  
cost int,  
primary key(sid,pid),  
foreign key (sid) references supplier(sid),  
foreign key (pid) references parts(pid)  
on update cascade  
on delete cascade  
);
```

Structure of the table

```
desc supplier;
```

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

Result 1 ×

desc parts;

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

desc catalog;

	Field	Type	Null	Key	Default	Extra
▶	sid	int	YES	MUL	NULL	
	pid	int	YES	MUL	NULL	
	cost	int	YES		NULL	

Inserting Values to the table

```
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');
```

```
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 catalog values (10001,20001,10);
insert into catalog values (10001,20002,10);
```

```
insert into catalog values (10001,20003,30);
insert into catalog values (10001,20004,10);
insert into catalog values (10001,20005,10);
insert into catalog values (10002,20001,10);
insert into catalog values (10002,20002,20);
insert into catalog values (10003,20003,30);
insert into catalog values (10004,20003,40);
```

```
select * from supplier;
```

	sid	sname	city
▶	10001	Acme Widget	bangalore
	10002	Johns	kolkata
	10003	Vimal	mumbai
	10004	Reliance	delhi
*	NULL	NULL	NULL

supplier 29 x parts 30 catalog 31
Output :::::::::::::::::::::

```
select * from parts;
```

	pid	pname	color
▶	20001	Book	red
	20002	Pen	red
	20003	Pencil	green
	20004	Mobile	green
	20005	Charger	black
*	NULL	NULL	NULL

parts 30 x
Output :::::::::::::::::::::

```
select * from catalog;
```

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

catalog 32 ×

Output :.....

Queries

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

```
select p.pname
from parts p
where exists(
    select c1.pid
    from catalog c1
    where c1.pid = p.pid
);
```

Result Grid | Filter Rows:

	pname
▶	Book
	Pen
	Pencil
	Mobile
	Charger

parts 33 ×

Output :.....

Find the snames of suppliers who supply every part.

```
select s.sname  
from supplier s  
where s.sid in  
(select c.sid  
from catalog c  
group by c.sid  
having count(c.pid)=  
select count(p.pid)  
from parts p));
```

Result Grid	
	sname
▶	Acme Widget

supplier 34 ×

Output ::::::::::::::::::::

Find the snames of suppliers who supply every red part.

```
select s.sname  
from supplier s  
where s.sid in(  
select c.sid  
from catalog c, parts po  
where po.color='red' and c.pid=po.pid  
group by c.sid  
having count(c.pid)=  
select count(p.pid)  
from parts p  
where p.color='red'));
```

Result Grid | Filter Rows

	sname
▶	Acme Widget
	Johns

supplier 35 ×

Output ::::::::::::::::::::

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

```
select p.pname
from parts p, supplier s
where s.sname = 'Acme Widget' and p.pid not in(
select c0.pid
from parts p0, catalog c0, supplier s0
where (s0.sname != 'Acme Widget') and c0.pid = p0.pid and s0.sid=c0.sid);
```

Result Grid | Filter Rows

	pname
▶	Mobile
	Charger

Result 36 ×

Output ::::::::::::::::::::

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 s.sname, s.sid
from catalog c1, supplier s
where c1.sid = s.sid and c1.cost > any(
select avg(c.cost)
from catalog c
where c.pid = c1.pid
group by c.pid);
```

Result Grid | Filter Rows:

	sname	sid
▶	Johns	10002
	Reliance	10004

Result 37 ×

Output ::::::::::::::::::::

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

```
select c.pid, s.sname
from supplier s, catalog c
where c.sid = s.sid and c.cost in(
select max(c1.cost)
from catalog c1
where c1.pid = c.pid
group by c1.pid
);
```

Result Grid | Filter Rows:

	pid	sname
▶	20001	Acme Widget
	20004	Acme Widget
	20005	Acme Widget
	20001	Johns
	20002	Johns
	20003	Reliance

Result 38 ×

Output ::::::::::::::::::::

