VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Database Management Systems (23CS3PCDBM)

Submitted by

HARSHITHA H G (1BM23CS108)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Sep-2024 to Jan-2025

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "Database Management Systems (23CS3PCDBM)" carried out by **Harshitha H G(1BM23CS108),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a Database Management Systems (23CS3PCDBM) work prescribed for the said degree.

Pradeep Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

Insurance Database

Question

(Week 1)

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

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

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

- OWNS (driver_id: String, reg_num: String)

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

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

- Display Accident date and location

- Update the damage amount to 25000 for the car with a specific reg_num (example 'K A031181') 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.25000

Schema Diagram

Create database

create database insurances_421;
use insurances_421;

Create table

 create table person(
 driver_id varchar(3) **primary key,**
 name varchar(20) **not null**,
 address varchar(100)
 );

 create table car(
 reg_no char(8) **primary key,**
 model varchar(20),
 year int(4) **not null**
 );

 create table accident(
 report_no int(4) **primary key,**

accident_date date,
location varchar(100)
);

create table owns(
driver_id varchar(3),
reg_no char(8),
foreign key(driver_id) **references** person(driver_id),
foreign key(reg_no) **references** car(reg_no)
);

create table participated(
driver_id varchar(3),
reg_no char(8),
report_no int(4),
damage_amt int,
foreign **key**(driver_id) **references** person(driver_id),
foreign **key**(reg_no) **references** car(reg_no),
foreign **key** (report_no) **references** accident(report_no)
);

Person table :

	Field	Type	Null	Key	Default	Extra
▶	driver_id	varchar(3)	NO	PRI	NULL	
	name	varchar(20)	NO		NULL	
	address	varchar(100)	YES		NULL	

Car table :

	Field	Type	Null	Key	Default	Extra
▶	reg_no	char(8)	NO	PRI	NULL	
	model	varchar(20)	YES		NULL	
	year	int	NO		NULL	

Accident table :

Field	Type	Null	Key	Default	Extra
report_no	int	NO	PRI	NULL	
accident_date	date	YES		NULL	
location	varchar(100)	YES		NULL	

Owns table :

Field	Type	Null	Key	Default	Extra
driver_id	varchar(3)	YES	MUL	NULL	
reg_no	char(8)	YES	MUL	NULL	

Participated table :

Field	Type	Null	Key	Default	Extra
driver_id	varchar(3)	YES	MUL	NULL	
reg_no	char(8)	YES	MUL	NULL	
report_no	int	YES	MUL	NULL	
damage_amt	int	YES		NULL	

Inserting Values into the table

insert into person values("A01","kusuma", "tyagaraj nagar");
insert into person values("A02","salma", " M K nagar");
insert into person values("A03","jayshree", "Ashok nagar");
insert into person values("A04","chintu", "Naa Roai Colony");
insert into person values("A05","Johnn", "hanumanth nagar");
select * from person;

Inserting values into the table

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

insert into owns values("A01","KA052250");
insert into owns values("A02","KA031181");
insert into owns values("A03","KA095477");
insert into owns values("A04","KA053408");
insert into owns values("A05","KA041702");
select * from owns;

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

insert into accident values(11,'2001-01-01',"Mysore Road");
insert into accident values(12,'2002-02-02',"South end Circle");
insert into accident values(13,'2003-01-21',"Bull temple Road");
insert into accident values(14,'2004-12-17',"Mysore Road");
insert into accident values(15,'2005-03-05',"Kanakpura Road");
select * from accident;

report_num	accident_date	location
11	2001-01-01	Mysore Road
13	2003-01-21	Bull temple Road
14	2004-12-17	Mysore Road
15	2005-03-05	Kanakpura Road
NULL	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("A05","KA041702",15,5000);
select * from participated;

Queries

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

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

select * from participated;

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

select count(distinct driver_id) CNT from participated a, accident b **where** a.report_num=b.report_num and b.accident_date like '2001%';

	CNT
▶	1

- Add a new accident to the database.

insert into accident values(16,'2003-03-08',"Domlur");

select * from accident;

	report_num	accident_date	location
▶	11	2001-01-01	Mysore Road
	13	2003-01-21	Bull temple Road
	14	2004-12-17	Mysore Road
	15	2005-03-05	Kanakpura Road
*	NULL	NULL	NULL

TO DO:

● DISPLAY ACCIDENT DATE AND LOCATION

select accident_date **as** date, location **from** accident;

	date	location
▶	2001-01-03	Mysore Rd
	2002-02-04	SE Circle
	2021-01-03	Bull Temple Rd
	2017-02-08	Mysore Rd
	2004-03-05	KR Puram

● DISPLAY DRIVER ID WHO DID ACCIDENT WITH DAMAGE AMOUNT

GREATER THAN OR EQUAL TO RS.25000

Select participated.driver_id as driver_id from accident,participated **where** accident.report_no = participated.report_no and participated.damage_amt >= 25000;

	driver_id
▶	A02
	A03

More Queries on Insurance Database

Question

(Week 2)

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

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

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

- OWNS (driver_id: String, reg_num: String)

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

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

- Display Accident date and location

- Update the damage amount to 25000 for the car with a specific reg_num (example 'K A031181') 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.25000

Schema Diagram

Queries

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

select * from car **order by** year asc;

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

select model, **count(model) from** participated, car **where** participated.reg_no = car.reg_no **group by** model;

	model	count(model)
▶	Lancer	1
	Audi	1
	Indica	1
	Honda	1
	Toyota	1

TO DO:

● **FIND THE AVERAGE DAMAGE AMOUNT**

   select avg(damage_amout) **as** average **from** participated;

	average
▶	18600.0000

● **DELETE THE TUPLE WHOSE DAMAGE AMOUNT IS BELOW THE AVERAGE DAMAGE AMOUNT**

delete from participated **where** damage_amt < (select * from (select **avg**(damage_amount) **from** participated) **as** average);

	average
▶	18600.0000

● **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 participated.damage_amount > (select **avg**(damage_amount) from participated);

	name
▶	Pradeep
	Smith

● **FIND MAXIMUM DAMAGE AMOUNT.**

select max(damage_amount) **from** participated;

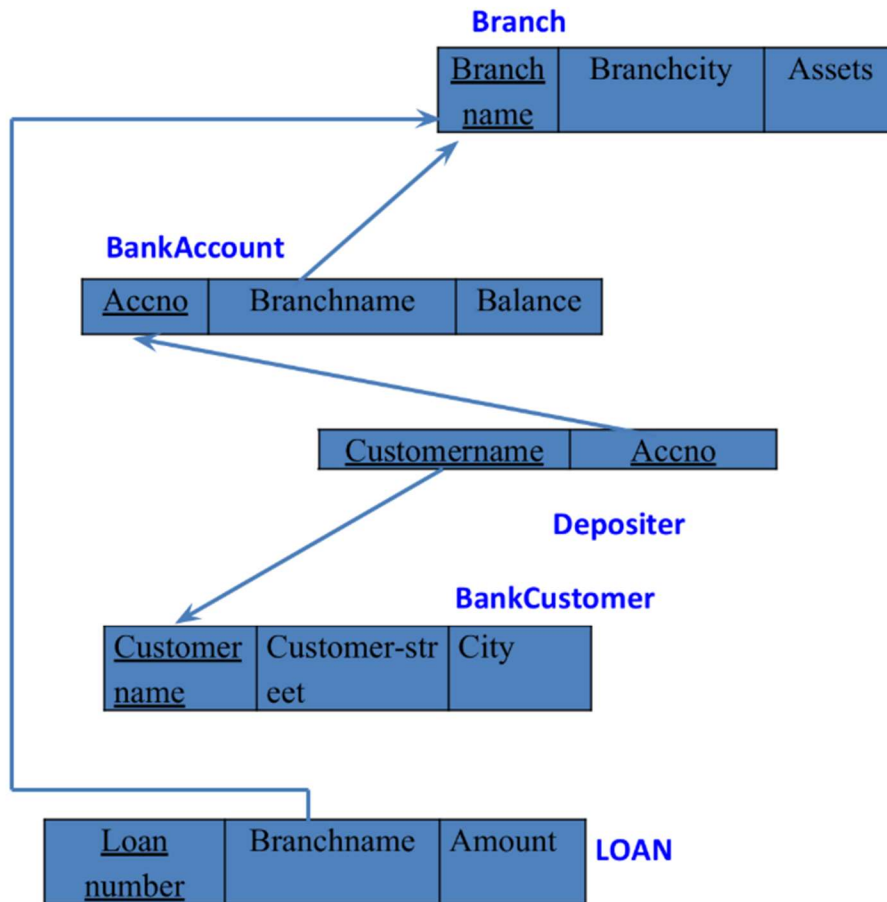|	max(damage_amt)
▶	50000

Bank Database

Question

(Week 3)

- Branch (branch-name: String, branch-city: String, assets: real)

- BankAccount(accno: int, branch-name: String, balance: real)

- BankCustomer (customer-name: String, customer-street: String, customer-city: String) - Depositer(customer-name: String, accno: int)

- LOAN (loan-number: int, branch-name: String, amount: real)

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

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

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

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

Schema Diagram

Create database

  create database bank_421;
  use bank_421;

Create table
  create table Branch(
  branchname varchar(20),
  city varchar(20),
  assets varchar(20),
  primary key(branchname));

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


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


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


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


create table Borrower(
customername varchar(20),
loannumber int,
Primary key(customername,loannumber),
foreign key(loannumber)references loan_402(loannumber),
foreign key(customername) references bankcustomer_402(customername));
```

Structure of the table

desc Branch;

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

desc Bankaccount;

Field	Type	Null	Key	Default	Extra
accno	int	NO	PRI	NULL	
branchname	varchar(20)	YES	MUL	NULL	
balance	varchar(20)	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 depositer;

Field	Type	Null	Key	Default	Extra
customername	varchar(20)	NO	PRI	NULL	
accno	int	NO	PRI	NULL	

desc loan;

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

Desc Borrower;

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

Inserting the values

insert into Branch values('SBI_Chamrajpete', 'Bangalore', 50000);
insert into Branch values('SBI_Residency_road', 'Bangalore',10000);
insert into Branch values('SBI_Shivaji_road', 'Bombay', 20000);
insert into Branch values('SBI_Parliament_road','Delhi', 10000);
insert into Branch values('SBI_Jantarmantar', 'Delhi',20000);
insert into Branch values('SBI_Mantrimarg','Delhi',150000);
select * from Branch;

branchname	city	assets
SBI_Chamrajpete	Bangalore	50000
SBI_Jantarmantar	Delhi	20000
SBI_Mantrimarg	Delhi	150000
SBI_Parliament_road	Delhi	10000
SBI_Residency_road	Bangalore	10000
SBI_Shivaji_road	Bombay	20000
NULL	NULL	NULL

insert into Bankaccount values(1, 'SBI_Chamrajpete',2000);

insert into Bankaccount values(2,'SBI_Residency_road', 5000);

insert into Bankaccount values(3,'SBI_Shivaji_road', 6000);

insert into Bankaccount values(4, 'SBI_Parliament_road', 9000);

insert into Bankaccount values(5, 'SBI_Jantarmantar', 8000);

insert into Bankaccount values(6, 'SBI_Shivaji_road', 4000);

insert into Bankaccount values(8, 'SBI_Residency_road', 4000);

insert into Bankaccount values(9, 'SBI_Parliament_road', 3000);

insert into Bankaccount values(10, 'SBI_Residency_road', 5000);

insert into Bankaccount values(11, 'SBI_Jantarmantar', 2000);

insert into Bankaccount values(12, 'SBI_Mantrimarg',2000);

select * from Bankaccount;

accno	branchname	balance
1	SBI_Chamrajpete	2000
2	SBI_Residency_road	5000
3	SBI_Shivaji_road	6000
4	SBI_Parliament_road	9000
5	SBI_Jantarmantar	8000
6	SBI_Shivaji_road	4000
8	SBI_Residency_road	4000
9	SBI_Parliament_road	3000
10	SBI_Residency_road	5000
11	SBI_Jantarmantar	2000
12	SBI_Mantrimarg	2000
NULL	NULL	NULL

insert into bankcustomer values('Avinash','Bulltemple_road','Bangalore');

insert into bankcustomer values('Dinesh', 'Bannerghatta_road','Bangalore');

insert into bankcustomer values('Mohan', 'National_college','Bangalore');

insert into bankcustomer values('Nikhil', 'Akbar_road', 'Delhi');

insert into bankcustomer values('Ravi', 'Prithviraj_road', 'Delhi');

select * from bankcustomer;

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);

select * from depositer;

insert into loan values(1, 'SBI_Chamrajpete',1000);

insert into loan values(2, 'SBI_Residency_road', 2000);

insert into loan values(3, 'SBI_Shivaji_road', 3000);

insert into loan values(4, 'SBI_Parliament_road', 4000);

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

select * from loan;

loannumber	branchname	amount
1	SBI_Chamrajpete	1000
2	SBI_Residency_road	2000
3	SBI_Shivaji_road	3000
4	SBI_Parliament_road	4000
5	SBI_Jantarmantar	5000
NULL	NULL	NULL

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 Borrower;

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

More Queries on Bank Database

Question

(Week 4)

- Branch (branch-name: String, branch-city: String, assets: real)

- BankAccount(accno: int, branch-name: String, balance: real)

- BankCustomer (customer-name: String, customer-street: String, customer-city: String) - Depositer(customer-name: String, accno: int)

- LOAN (loan-number: int, branch-name: String, amount: real)

- 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

Branch

Branch name	Branchcity	Assets

BankAccount

Accno	Branchname	Balance

Customername	Accno

Depositer

BankCustomer

Customer name	Customer-street	City

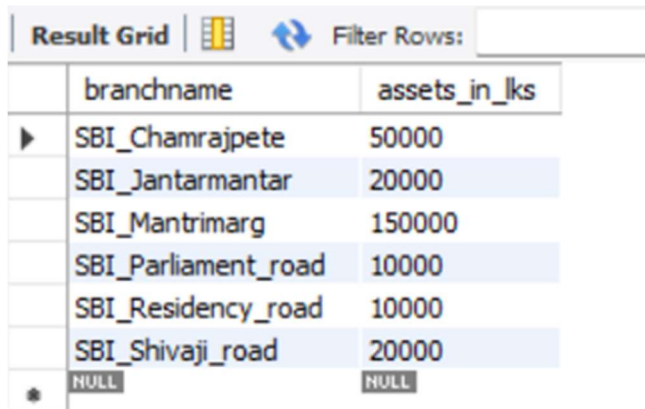| Loan number	Branchname	Amount

LOAN

Queries

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

alter table Branch **rename column** assets **to** assets_in_lks;
select branchname, assets_in_lks **from** Branch;
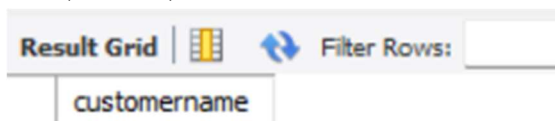
	branchname	assets_in_lks
▶	SBI_Chamrajpete	50000
	SBI_Jantarmantar	20000
	SBI_Mantrimarg	150000
	SBI_Parliament_road	10000
	SBI_Residency_road	10000
	SBI_Shivaji_road	20000
*	NULL	NULL

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

select d.customername **from** depositer d, Bankaccount b **where**
b.branchname=**'ResideRoad' and** d.accno=b.accno **group by** d.customername **having**
count(d.accno)>=2;

customername

● Create a view which gives each branch the sum of the amount of all the loans at the branch.
create view br as select branchname, **sum**(amount) from loan

group by branchname;

select * from br;

	branchname	sum(amount)
▶	SBI_Chamrajpete	1000
	SBI_Jantarmantar	5000
	SBI_Parliament_road	4000
	SBI_Residency_road	2000
	SBI_Shivaji_road	3000

Result Grid | Filter Rows:

Employee Database

Question

(Week 5)

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

2. Enter greater than five tuples for each table.

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

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

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

ER Diagram
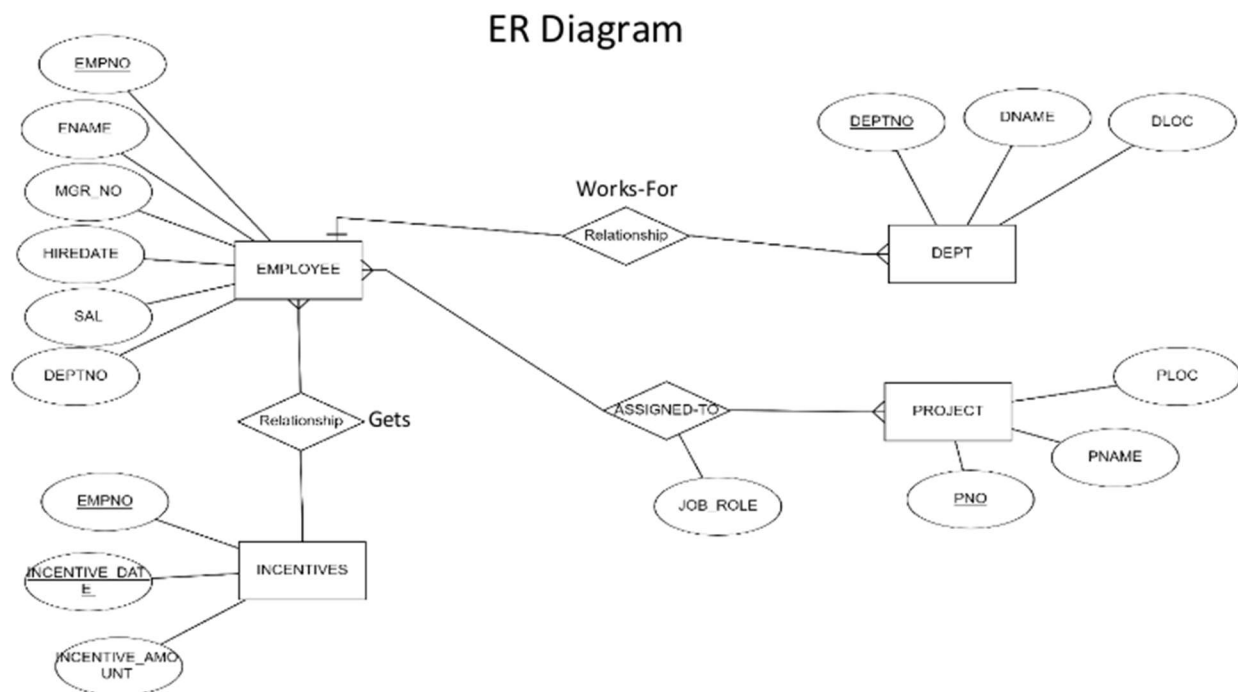
Create database

```
create database employee_421;
use employee_421;
```

Create tables

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

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

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

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

 create table incentives (
 empno int,
 incentive_date date,
 incentive_amount int,
 foreign key (empno) references employee(empno));
```

Structure of the table

Department table :

  desc dept;

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

Employee Table :

  desc employee;

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

Project table :

  desc project;

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

Assigned_to table :

desc assigned_to;

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

Incentive table:

desc incentives;

Field	Type	Null	Key	Default	Extra
empno	int	YES	MUL	NULL	
incentive_date	date	YES		NULL	
incentive_amount	int	YES		NULL	

Inserting the values to the tables

insert into dept values(10,'sales','bengaluru');
insert into dept values(20,'marketing','hyderbad');
insert into dept values(30,'finance','mysuru');
insert into dept values(40,'HR','bengaluru');
insert into dept values(50,'IT','hyderbad');
select * from dept;

depno	dname	dloc
10	sales	bengaluru
20	marketing	hyderbad
30	finance	mysuru
40	HR	bengaluru
50	IT	hyderbad
NULL	NULL	NULL

insert into employee values(1,'alice',2,'2022-01-01',55000,10);
insert into employee values(2,'bob',3,'2023-04-05',68000,20);
insert into employee values(3,'charlie',1,'2025-06-11',23000,10);
insert into employee values(4,'david',2,'2022-03-01',55800,20);
insert into employee values(5,'emily',null,'2022-04-11',67800,10);
select * from employee;

empno	ename	mgrno	hiredate	sal	depno
1	alice	2	2022-01-01	55000	10
2	bob	3	2023-04-05	68000	20
3	charlie	1	2025-06-11	23000	10
4	david	2	2022-03-01	55800	20
5	emily	NULL	2022-04-11	67800	10
NULL	NULL	NULL	NULL	NULL	NULL

insert into project values(1,'e-learning','bengaluru');
insert into project values(2,'hostel management','hyderbad');
insert into project values(3,'hotel management','bengaluru');
insert into project values(4,'face recognition','chennai');
insert into project values(5,'face emotion recognition','mysuru');
select * from project;

empno	ename	mgrno	hiredate	sal	depno
1	alice	2	2022-01-01	55000	10
2	bob	3	2023-04-05	68000	20
3	charlie	1	2025-06-11	23000	10
4	david	2	2022-03-01	55800	20
5	emily	NULL	2022-04-11	67800	10
NULL	NULL	NULL	NULL	NULL	NULL

insert into assignment values(1,1,'manager');
insert into assignment values(2,2,'team lead');
insert into assignment values(3,1,'analyst');
insert into assignment values(4,2,'developer');
insert into assignment values(5,3,'tester');
select * from assignment;

empno	pno	jobrole
1	1	manager
2	2	team lead
3	1	analyst
4	2	developer
5	3	tester
NULL	NULL	NULL

insert into incentives values(1,'2022-11-10',5000);
insert into incentives values(2,'2023-12-10',5000);
insert into incentives values(3,'2024-12-10',5000);
insert into incentives values(4,'2022-10-10',5000);
select * from incentives;

empno	incentive_date	incentive_amount
1	2022-11-10	5000
2	2023-12-10	5000
3	2024-12-10	5000
4	2022-10-10	5000
NULL	NULL	NULL

Queries

3 Retrieve the employee numbers of all employees who work on project located in Bengaluru, Hyderabad, or Mysuru.
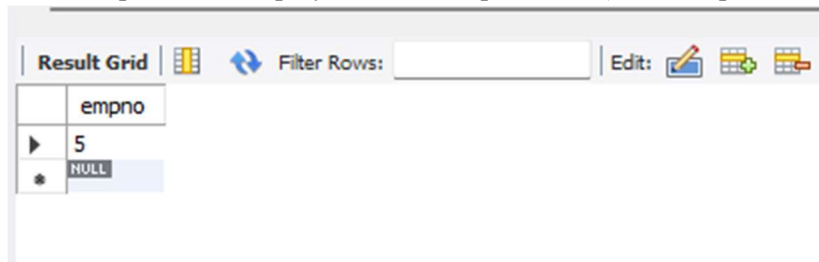
select empno **from** assignment **where** pno in(select pno **from** project **where** ploc in('bengaluru','mysuru'));

empno
1
3
5

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

select empno **from** employee **where** empno not in(select empno from incentives);
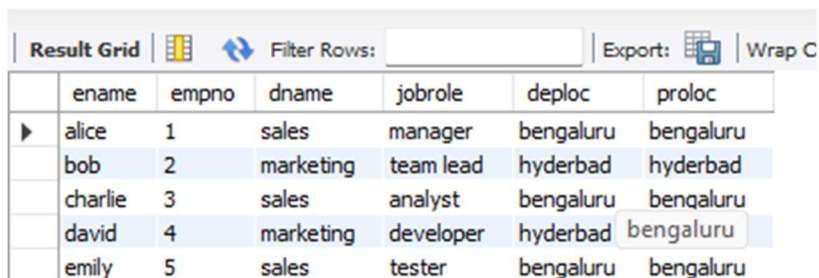
● **Write a SQL query to find the employees name, number, dept, job_role, department location and project location who are working for a project location same as his/her department location.**

select e.ename,e.empno, d.dname, a.jobrole, d.dloc as deploc, p.ploc as proloc **from** employee e **join** dept d on e.depno = d.depno join assignment a **on** e.empno = a.empno **join** project p on a.pno=p.pno **where** d.dloc = p.ploc;

ename	empno	dname	jobrole	deploc	proloc
alice	1	sales	manager	bengaluru	bengaluru
bob	2	marketing	team lead	hyderbad	hyderbad
charlie	3	sales	analyst	bengaluru	bengaluru
david	4	marketing	developer	hyderbad	bengaluru
emily	5	sales	tester	bengaluru	bengaluru

More Queries on Employee Database

Question

(Week 6)

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.

ER Diagram
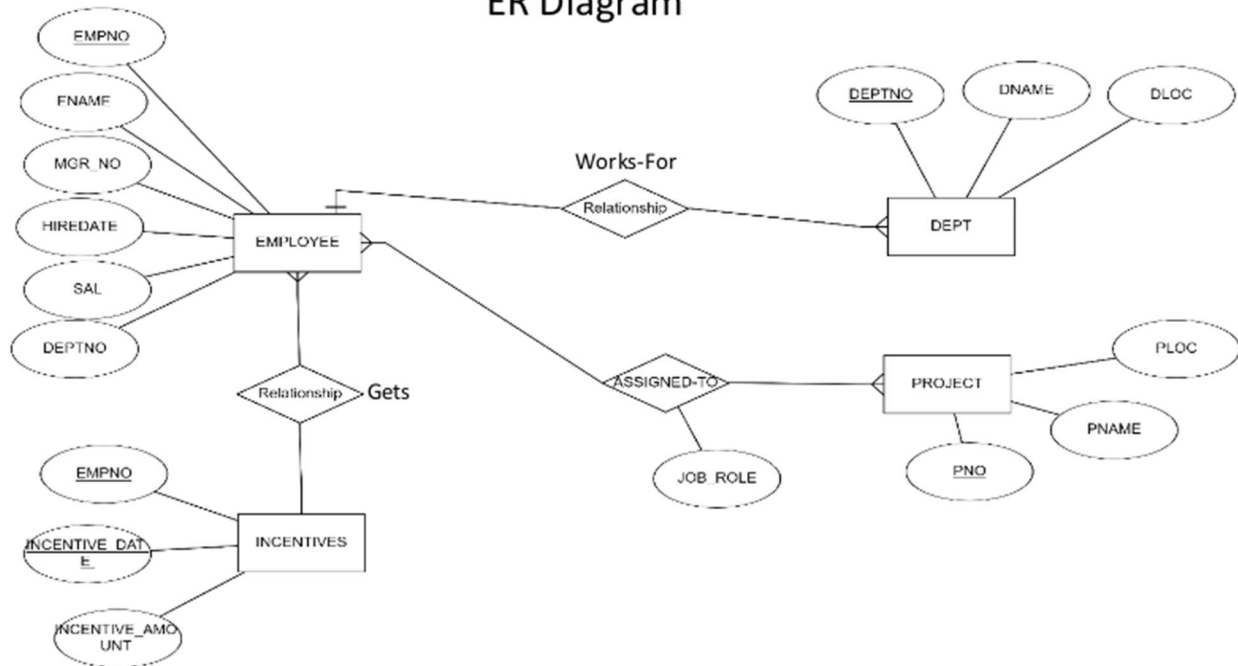
● **List the name of the managers with the maximum employees**

   select mgrno as manager_id, **count(**empno) as employeecount **from** employee **group by** mgrno **order by** employeecount **desc** limit 1;

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

select m.ename as managername,m.sal **as** managersalary,emp_avg.avg_employee_salary **from** employee m **join** (select mgrno,avg(sal)as avg_employee_salary from employee **group by** mgrno) as emp_avg **on** m.empno=emp_avg.mgrno;

managername	managersalary	avg_employee_salary
alice	55000	23000.0000
bob	68000	55400.0000
charlie	23000	68000.0000

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

select ename as secondtopmanager **from**(select m.empno,d.depno,row_number() over(partition by d.depno **order by** m.sal desc) as rank1 **from** employee m **join** dept d on m.depno=d.depno **where** m.mgrno is null) as rankedmanagers **where** rank1=2;

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

select e.empno,e.ename,e.sal, e.depno from employee e join incentives i on e.empno=i.empno where i.incentive_date between '2022-11-10' and '2024-12-10' order by i.incentive_amount desc limit 1 offset 1;

empno	ename	sal	depno
2	bob	68000	20

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

select e.ename as employeename **from** employee e **join** employee m **on** e.mgrno=m.mgrno **where** e.depno=m.depno;

employeename
alice
bob
charlie
david

# Supplier Database

Question

(Week 7)

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

2. Insert appropriate records in each table.

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

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

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

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

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

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

Schema Diagram

Supplier

sid	sname	city

Parts

pid	pname	color

sid	pid	cost

Catalog
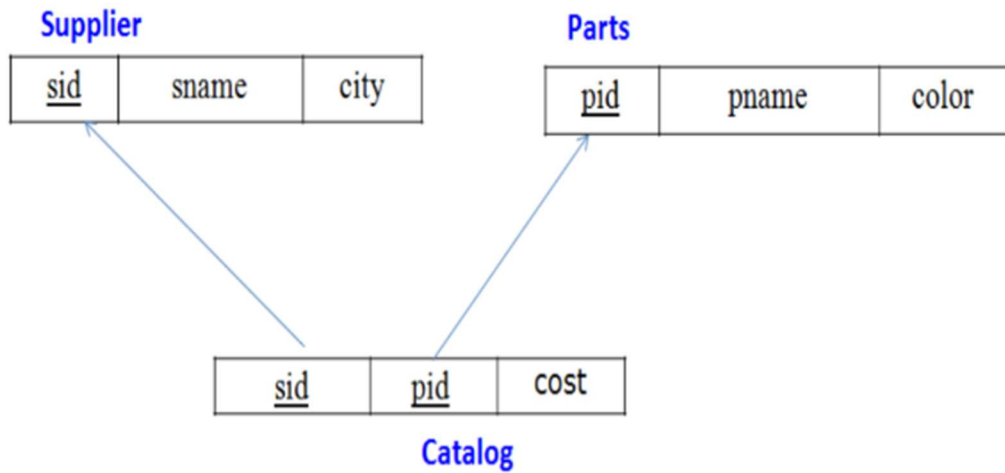
Create Database

Create database supplier_421;
Use supplier_421;

Create tables

create table supplier(
sid int **primary key,**
sname varchar(20),
city varchar(30)
);
desc supplier;

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

desc parts;
create table catalog(

sid int, pid int,
cost int,
foreign key(sid) **references** supplier(sid),
foreign key(pid) **references** parts(pid)
);
desc catalog;

Supplier Table

Field	Type	Null	Key	Default	Extra
sid	int	NO	PRI	NULL	
sname	varchar(20)	YES		NULL	
city	varchar(30)	YES		NULL	

Parts Table

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

Catalog table

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

Inserting the values

insert into supplier values
(10001, "acne", "Bangalore"),

(10002, "johns", "Kolkata"),
(10003, "vimal", "Mumbai"),
(10004, "reliance", "Delhi");
select * from supplier;

Supplier Table

sid	sname	city
10001	acne	Bangalore
10002	johns	Kolkata
10003	vimal	Mumbai
10004	reliance	Delhi
NULL	NULL	NULL

insert into parts values
(20001,"Book","Red"),
(20002,"Pen","Red"),
(20003,"Pencil","Green"),
(20004,"Mobile","Green"),
(20005,"Charger","Black");
select * from parts;

Parts Table

pid	pname	color
20001	Book	Red
20002	Pen	Red
20003	Pencil	Green
20004	Mobile	Green
20005	Charger	Black
NULL	NULL	NULL

insert into catalog values
(10001,20001,10),
(10001,20002,10),
(10001,20003,30),

(10001,20004,10),
(10001,20005,10),
(10002,20001,10),
(10002,20002,20),
(10003,20003,30),
(10004,20003,40);
select * from catalog;

Catalog Table

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

Queries

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

  select pname **from** parts **where** pid **in** (**select** pid **from** catalog);

pname
Book
Pen
Pencil
Mobile
Charger

● **Find the snames of suppliers who supply every part.**

  select sname **from** supplier **where sid in**
  (**select sid from** catalog **group by sid having count(distinct** pid) = (**select count(distinct**
  pid) **from** parts));

sname
▶ acne

● **Find the snames of suppliers who supply every red part.**

select distinct sname **from** supplier, parts, catalog
where supplier.sid = catalog.sid **and** parts.pid = catalog.pid **and** parts.color="Red";

sname
▶ acne
johns

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

select pname **from** parts **where** pid **not in**
(**select** pid **from** catalog **where** sid **in** (**select** sid **from** supplier **where** sname != "acne"));

pname
▶ Mobile
Charger

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

select sid **from** catalog a **where** a.cost > (**select avg**(b.cost) **from** catalog b **where** a.pid = b.pid **group by** b.pid);

sid
▶ 10002
10004

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

select pid, sname **from** catalog a, supplier **where** a.cost = (select max(b.cost) **from**

catalog b **where** a.pid = b.pid group by b.pid) and
supplier.sid = a.sid;

pid	sname
20001	acne
20004	acne
20005	acne
20001	johns
20002	johns
20003	reliance

NoSQL Lab 1

Question

(Week 8)

Perform the following DB operations using MongoDB.

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

2. Insert appropriate values

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

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

5. Export the created table into local file system

6. Drop the table

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

mongodb collection.

Create database

db.createCollection("Student"**)**;

Show dbs;

```
Atlas atlas-ilms3w-shard-0 [primary] test> db.createCollection("Student");
{ ok: 1 }
Atlas atlas-ilms3w-shard-0 [primary] test> show dbs;
sample_mflix  121.20 MiB
test           72.00 KiB
admin         328.00 KiB
local           4.90 GiB
Atlas atlas-ilms3w-shard-0 [primary] test> show dbs;
sample_mflix  121.20 MiB
test           72.00 KiB
admin         328.00 KiB
local           4.90 GiB
Atlas atlas-ilms3w-shard-0 [primary] test> |
```

Create table & Inserting Values to the table

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

db.Student.insert({RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"});

db.Student.insert({RollNo:3,Age:21,Cont:5576,email:"anubhav.de9@gmail.com"});

db.Student.insert({RollNo:4,Age:20,Cont:4476,email:"pani.de9@gmail.com"});

db.Student.insert({RollNo:10,Age:23,Cont:2276,email:"rekha.de9@gmail.com"});

```
Atlas atlas-ilms3w-shard-0 [primary] test> 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("674bfdd272c929d70ce00e40") }
}
Atlas atlas-ilms3w-shard-0 [primary] test>

Atlas atlas-ilms3w-shard-0 [primary] test> db.Student.insert({RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"});

{
  acknowledged: true,
  insertedIds: { '0': ObjectId("674bfdd272c929d70ce00e41") }
}
Atlas atlas-ilms3w-shard-0 [primary] test>

Atlas atlas-ilms3w-shard-0 [primary] test> db.Student.insert({RollNo:3,Age:21,Cont:5576,email:"anubhav.de9@gmail.com"});

{
  acknowledged: true,
  insertedIds: { '0': ObjectId("674bfdd272c929d70ce00e42") }
}
Atlas atlas-ilms3w-shard-0 [primary] test>

Atlas atlas-ilms3w-shard-0 [primary] test> db.Student.insert({RollNo:4,Age:20,Cont:4476,email:"pani.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("674bfdd372c929d70ce00e43") }
}
Atlas atlas-ilms3w-shard-0 [primary] test>

Atlas atlas-ilms3w-shard-0 [primary] test> db.Student.insert({RollNo:10,Age:23,Cont:2276,email:"rekha.de9@gmail.com"});
{
  acknowledged: true,
Atlas atlas-ilms3w-shard-0 [primary] test> |
```

Structure of the table

db.Student.**find()**;

```
    _id: ObjectId("674bfdd272c929d70ce00e40"),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId("674bfdd272c929d70ce00e41"),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de9@gmail.com'
  },
  {
    _id: ObjectId("674bfdd272c929d70ce00e42"),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
  },
  {
    _id: ObjectId("674bfdd372c929d70ce00e43"),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId("674bfdd372c929d70ce00e44"),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'rekha.de9@gmail.com'
  }
]
```

Queries

● **Write a query to update the Email-Id of a student with rollno 5.**

db.Student.update({rollno:5},{$set:{email:"abhinav@gmail.com"}});

```
]
Atlas atlas-ilms3w-shard-0 [primary] test> 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: 0,
  upsertedCount: 0
}
Atlas atlas-ilms3w-shard-0 [primary] test> |
```

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

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

```
Atlas atlas-ilms3w-shard-0 [primary] test> db.Student.insert({RollNo:11,Age:22,Name:"ABC",Cont:2276,email:"rea.de9@gmail
.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("674c001b72c929d70ce00e45") }
}
Atlas atlas-ilms3w-shard-0 [primary] test> db.Student.update({RollNo:11,Name:"ABC"},{$set:{Name:"FEM"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```