**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JnanaSangama”, Belgaum -590014, Karnataka.**



### LAB REPORT

**on**

**Database Management Systems (22CS3PCDBM)**

***Submitted by***

### Nithin S (1BM21CS120)

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

***in***

### COMPUTER SCIENCE AND ENGINEERING



**B.M.S. COLLEGE OF ENGINEERING**

**(Autonomous Institution under VTU)**

### BENGALURU-560019

**October-2022 to Feb-2023**

### B. M. S. College of Engineering

### Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled “Database Management Systems (22CS3PCDBM)” carried out by **Nithin S (1BM21CS120),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a Database Management Systems (22CS3PCDBM) work prescribed for the said degree.

Dr Manjunath D R **Dr. Jyothi S Nayak**

Assistant Professor Professor and Head

Department of CSE Department of CSE

BMSCE, Bengaluru BMSCE, Bengaluru

`

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.**  **No.** | **Date** | **Experiment Title** | **Page No.** |
| **1** | 8/11/22 | Insurance Database | **1-7** |
| **2** | 15/11/22 | More Queries on Insurance Database | **8-16** |
| **3** | 22/11/22 | Bank Database | **17-24** |
| **4** | 29/11/22 | More Queries on Bank Database | **25-34** |
| **5** | 6/12/22 | Employee Database | **35-42** |
| **6** | 13/12/22 | More Queries on Employee Database | **43-51** |
| **7** | 25/12/22 | Supplier Database | **52-58** |
| **8** | 27/12/22 | Flight Database | **59-66** |
| **9** | 17/1/23 | NoSQL | **67-68** |

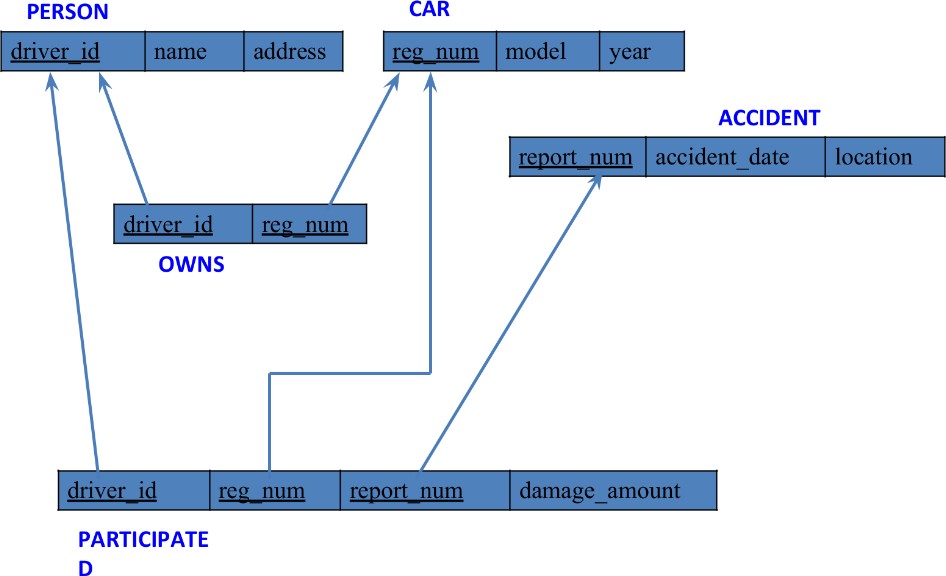
**WEEK 1**

**Insurance Database**

1.Create the above tables by properly specifying the primary keys and the foreign keys. 2.Enter at least five tuples for each relation

1. Display Accident date and location
2. 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.
3. Add a new accident to the database.
4. Display driver id who did accident with damage amount greater than or equal to Rs.25000

# Schema Diagram:



1

# CREATE DATABASE:

SQL> create database Nithin\_insurance;

use Nithin\_insurance;

# CREATE TABLE:

##### Accident Table

SQL> create table accident(

report\_num int,

accident\_date date,

location varchar(50),

primary key(report\_num)

);

##### Person Table

SQL>create table person(

driver\_id varchar(30) primary key,

person\_name varchar(40),

address varchar(60)

);

##### Car table

SQL>create table car(

reg\_num varchar(40),

model varchar(40),

buy\_year int,

primary key(reg\_num)

);

##### Participated Table

SQL>create table participated(

driver\_id varchar(30),

reg\_num varchar(40),

report\_num int,

damage\_amt 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)

);

##### Owns Table

SQL>create table owns(

driver\_id varchar(30),

reg\_num varchar(40),

primary key(driver\_id,reg\_num),

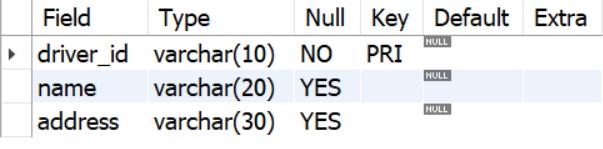
foreign key(driver\_id) references person(driver\_id),

foreign key(reg\_num) references car(reg\_num)

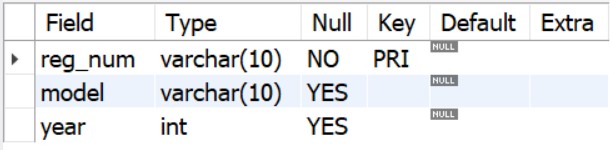
);

# Structure of the table:

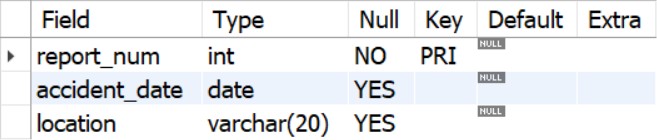
##### Person Table



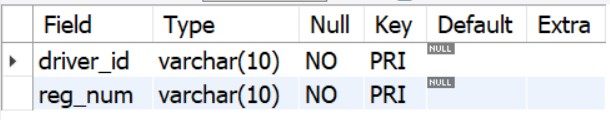
1. **Car Table**



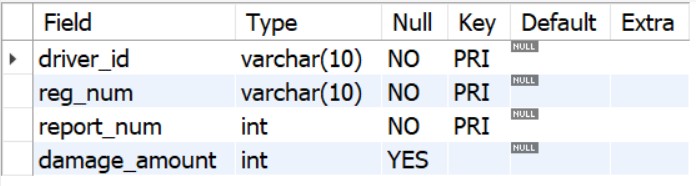
##### Accident Table



1. **Owns Table**



1. **Participated Table**

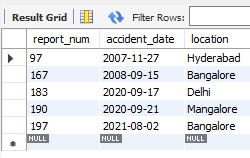


# INSERTING VALUES INTO TABLE:

##### Table Accident

into insert accident

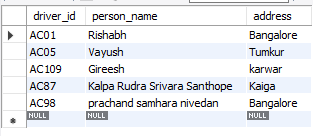
values(197,'2021-08-02','Bangalore'),(190,'2020-09-21','Mangalore'),(167,'2008-09-15','Bangalore'),(183,'2020-09-17','Delhi'),(097,'2007-11-27','Hyderabad');



##### Table Person

insert into person

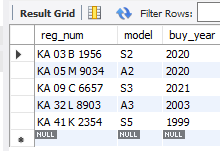
values('AC01','Rishabh','Bangalore'),('AC05','Vayush','Tumkur'),('AC109','Gireesh','karwar'),('AC98','prachand samhara nivedan','Bangalore'),('AC87','Kalpa Rudra Srivara Santhope','Kaiga');



##### Table Car

insert into car

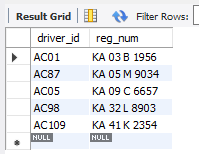
values('KA 03 B 1956','S2',2020),('KA 09 C 6657','S3',2021),('KA 41 K 2354','S5',1999),('KA 32 L 8903','A3',2003),('KA 05 M 9034','A2',2020);



##### Table Owns

insert into owns

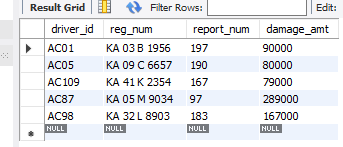
values('AC01','KA 03 B 1956'),('AC05','KA 09 C 6657'),('AC109','KA 41 K 2354'),('AC98','KA 32 L 8903'),('AC87','KA 05 M 9034');



##### Table Participated:

insert into participated

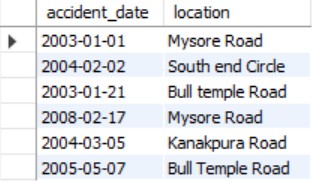
values('AC01','KA 03 B 1956',197,90000),('AC05','KA 09 C 6657',190,80000),('AC109','KA 41 K 2354',167,79000),('AC98','KA 32 L 8903',183,167000),('AC87','KA 05 M 9034',097,289000);



# QUERIES:

#### Display Accident date and Location

SQL> SELECT accident\_date, location from accident



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

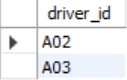
SQL> update participated set damage\_amount=25000 where reg\_num='KA053408' and report\_num=12;

#### Add a new accident to the database.

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

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

SQL> Select driver\_id from participated Where damage\_amount>=25000;



**WEEK 2**

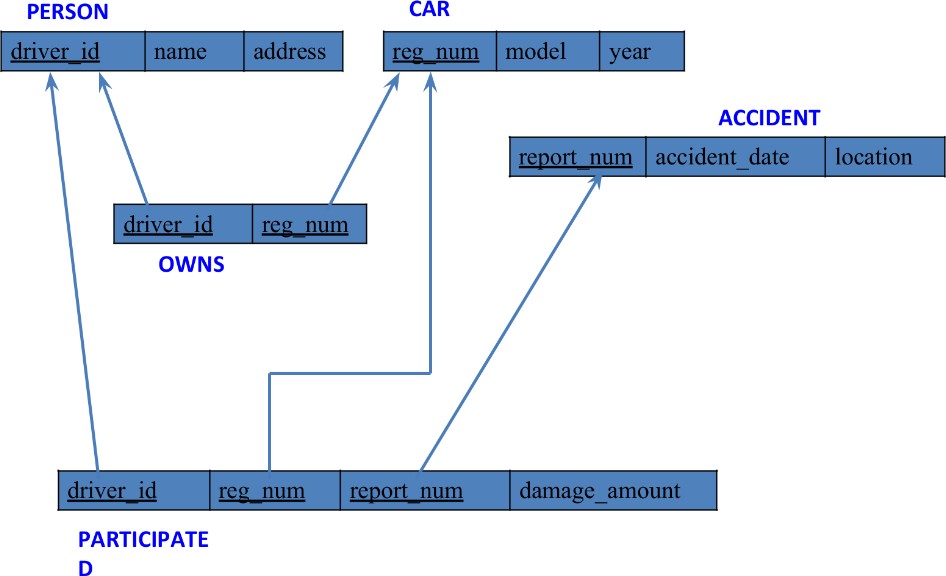
**Insurance Database**

1. 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
2. Enter at least five tuples for each relation
3. Display the entire CAR relation in the ascending order of manufacturing year.
4. Find the number of accidents in which cars belonging to a specific model (example 'Lancer') were involved.
5. Find the total number of people who owned cars that involved in accidents in 2008.

##### To Do List:

1. List the entire participated relation in the descending order of damage amount.
2. Find the average damage amount
3. Delete the tuple whose damage amount is below
4. The average damage amount
5. List the name of drivers whose damage is greater than the average damage amount.
6. Find maximum damage amount.

**Schema Diagram:**



**CREATE DATABASE:**

SQL> create database INSURANCE;

# CREATE TABLE:

##### Accident Table

SQL> create table accident(report\_num int, accident\_date date, location varchar(20),primary key(report\_num));

##### Person Table

SQL>create table person (driver\_id varchar(10), name varchar(20), address varchar(30), primary key(driver\_id));

##### Car table

SQL>create table car(reg\_num varchar(10),model varchar(10),year int, primary key(reg\_num));

##### Participated Table

SQL>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));

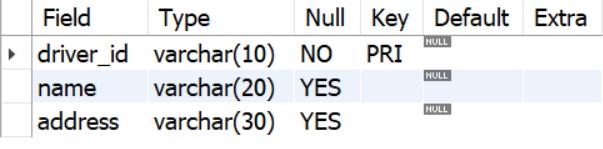
##### Owns Table

SQL>create table owns(driver\_id varchar(10),reg\_num varchar(10), primary key(driver\_id, reg\_num),

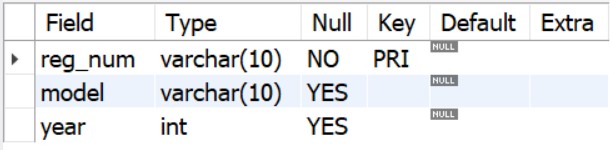
foreign key(driver\_id) references person(driver\_id), foreign key(reg\_num) references car(reg\_num));

# Structure of the table:

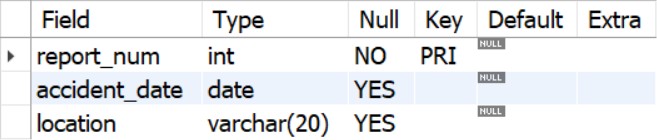
##### Person Table



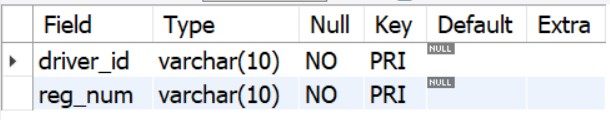
1. **Car Table**



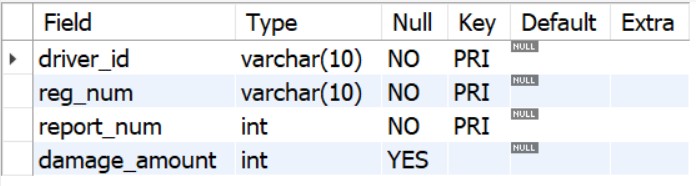
##### Accident Table



1. **Owns Table**



1. **Participated Table**

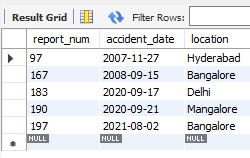


# INSERTING VALUES INTO TABLE:

##### Table Accident

insert into accident

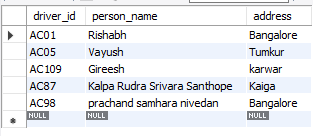
values(197,'2021-08-02','Bangalore'),(190,'2020-09-21','Mangalore'),(167,'2008-09-15','Bangalore'),(183,'2020-09-17','Delhi'),(097,'2007-11-27','Hyderabad');



##### Table Person

insert into person

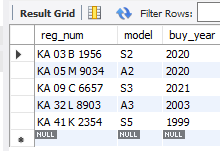
values('AC01','Rishabh','Bangalore'),('AC05','Vayush','Tumkur'),('AC109','Gireesh','karwar'),('AC98','prachand samhara nivedan','Bangalore'),('AC87','Kalpa Rudra Srivara Santhope','Kaiga');



##### Table Car

insert into car

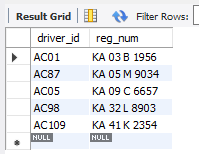
values('KA 03 B 1956','S2',2020),('KA 09 C 6657','S3',2021),('KA 41 K 2354','S5',1999),('KA 32 L 8903','A3',2003),('KA 05 M 9034','A2',2020);



##### Table Owns

insert into owns

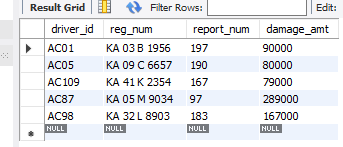
values('AC01','KA 03 B 1956'),('AC05','KA 09 C 6657'),('AC109','KA 41 K 2354'),('AC98','KA 32 L 8903'),('AC87','KA 05 M 9034');



##### Table Participated:

insert into participated

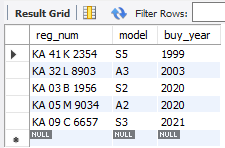
values('AC01','KA 03 B 1956',197,90000),('AC05','KA 09 C 6657',190,80000),('AC109','KA 41 K 2354',167,79000),('AC98','KA 32 L 8903',183,167000),('AC87','KA 05 M 9034',097,289000);



# QUERIES:

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

SQL> select \* from car order by buy\_year asc;



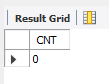
#### Find the number of accidents in which cars belonging to a specific model (example 'Lancer') were involved.

SQL> select count(report\_num) from car c,participated p where c.reg\_num=p.reg\_num and c.model='S2';

https://lh4.googleusercontent.com/VQ0-v-jjd8eraTo51iPe1Bo74NEcDjqxXcLTmnZIRHj3UNAOxoZkgHJix80cUHk9FgEOhpmY_IrsQQV-NJB2mPE0agUYaOEhr84K7cNylQ3mg54CPJcDHz4LrCUmB2-Szmn4OA1al9-7z-VT-hMiQTuZD6nLpV-lQu5b0wnaeKdCoAF4DwhLu4sscqNZyNxTFlFHJN4

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

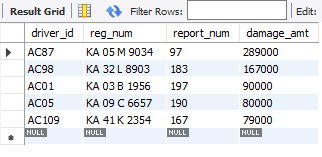
SQL> select count(distinct driver\_id) CNT from participated a,accident b where a.report\_num=b.report\_num and b.accident\_date like '\_\_08';



**TO DO LIST:**

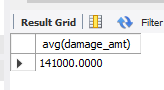
#### List the entire participated relation in the descending order of damage amount.

SQL> select \* from participated order by damage\_amt desc;



#### Find the average damage amount

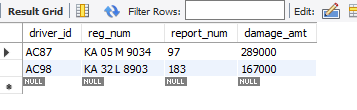
SQL> select avg(damage\_amt) from participated;



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

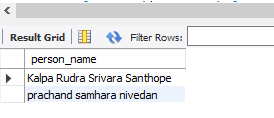
SQL> delete from participated where

Damage\_amt<(select t.amt from (select avg(damage\_amt) as amt from participated) t);



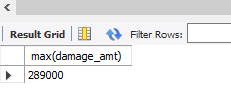
#### List the name of drivers whose damage is greater than the average damage amount.

SQL> select p.person\_name from person p,participated pa where p.driver\_id=pa.driver\_id and pa.damage\_amt>(select avg(damage\_amt) from participated);



#### Find maximum damage amount.

SQL> select max(damage\_amt) from participated;

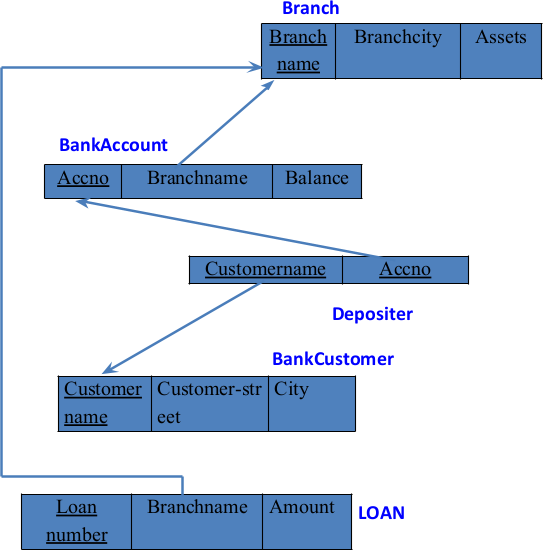


**WEEK 3**

**Bank Database**

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

# SCHEMA DIAGRAM:



**CREATE COMMAND:**

##### Table Branch

create table Branch (branch\_name varchar(30), branch\_city varchar(20), assets real, primary key(branch\_name));

##### Table BankAccount

create table BankAccount (accno int, branch\_name varchar(30), balance real, primary key (accno),

foreign key (branch\_name) references Branch(branch\_name));

##### Table BankCustomer

create table BankCustomer (customer\_name varchar(20), customer\_street varchar(20),customer\_city varchar(20),primary key(customer\_name));

##### Table Depositer

create table Depositer(customer\_name varchar(20), accno int, primary key(customer\_name, accno),

foreign key (customer\_name) references BankCustomer(customer\_name), foreign key (accno) references BankAccount(accno));

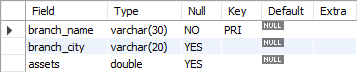
##### Table Loan

create table LOAN (loan\_number int, branch\_name varchar(30), amount real, primary key(loan\_number),

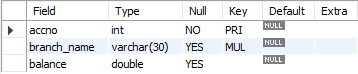
foreign key (branch\_name) references Branch(branch\_name));

# Structure of the table:

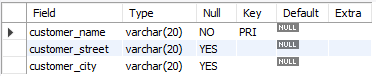
##### Branch Table



1. **BankAccount Table**



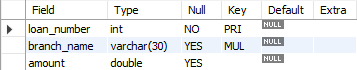
##### Bank Customer Table



1. **Depositer Table**



1. **Loan Table**

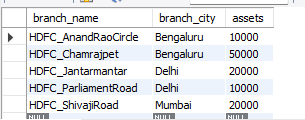


# INSERT COMMAND:

##### Table Branch

insert into branch

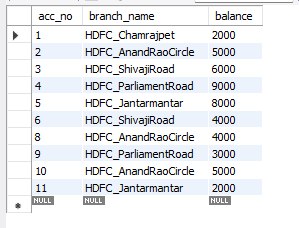
values('HDFC\_Chamrajpet','Bengaluru',50000),('HDFC\_AnandRaoCircle','Bengaluru',10000),('HDFC\_ShivajiRoad','Mumbai',20000),('HDFC\_ParliamentRoad','Delhi',10000),('HDFC\_Jantarmantar','Delhi',20000);



##### Table BankAccount

insert into bank\_account

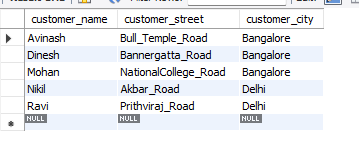
values(1,'HDFC\_Chamrajpet',2000),(2,'HDFC\_AnandRaoCircle',5000),(3,'HDFC\_ShivajiRoad',6000),(4,'HDFC\_ParliamentRoad',9000),(5,'HDFC\_Jantarmantar',8000),(6,'HDFC\_ShivajiRoad',4000),(8,'HDFC\_AnandRaoCircle',4000),(9,'HDFC\_ParliamentRoad',3000),(10,'HDFC\_AnandRaoCircle',5000),(11,'HDFC\_Jantarmantar',2000);



##### Table BankCustomer

insert into bank\_customer

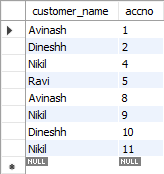
values('Avinash','Bull\_Temple\_Road','Bangalore'),('Dinesh','Bannergatta\_Road','Bangalore'),('Mohan','NationalCollege\_Road','Bangalore'),('Nikil','Akbar\_Road','Delhi'),('Ravi','Prithviraj\_Road','Delhi');



##### Table Depositer

insert into depositer

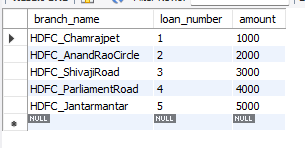
values('Avinash',1),('Dinesh',2),('Nikil',4),('Ravi',5),('Avinash',8),('Nikil',9),('Dinesh',10),('Nikil',11);



##### Table Loan

insert into loan

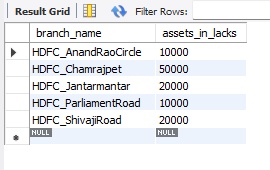
values('HDFC\_Chamrajpet',1,1000),('HDFC\_AnandRaoCircle',2,2000),('HDFC\_ShivajiRoad',3,3000),('HDFC\_ParliamentRoad',4,4000),('HDFC\_Jantarmantar',5,5000);



# SQL Queries:

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

SQL > select branch\_name,assets as assets\_in\_lacks from branch;



1. Find all the customers who have at least two accounts at the same branch (ex.SBI\_ResidencyRoad).

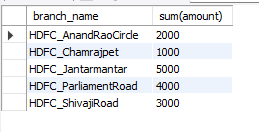
SQL> select d.customer\_name,count(d.acc\_no) from bank\_account b,depositer d where b.acc\_no=d.acc\_no and branch\_name='HDFC\_AnandRaoCircle' group by customer\_name having count(d.acc\_no)>=2;

https://lh5.googleusercontent.com/T9i2FOYbOU23Uc6j2TeoV9g-wNhvNyCba5PvuJleBvB1ktnVYvA0j7719TsnmWuSDUcDQNVMbz6pyEntuVhxk2vQzy_8-bMJmBXqj9qW02NrWREmFKIrII9kLQF5Vv_OooNG2pvl2qWladJneLa1CchcoocHdUxxBZKzSJRPhxyjY80jdHByApeXPX353NwC8MdxV3M

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

SQL> create view loan\_sum as

select l.branch\_name,sum(amount) from loan l group by l.branch\_name;

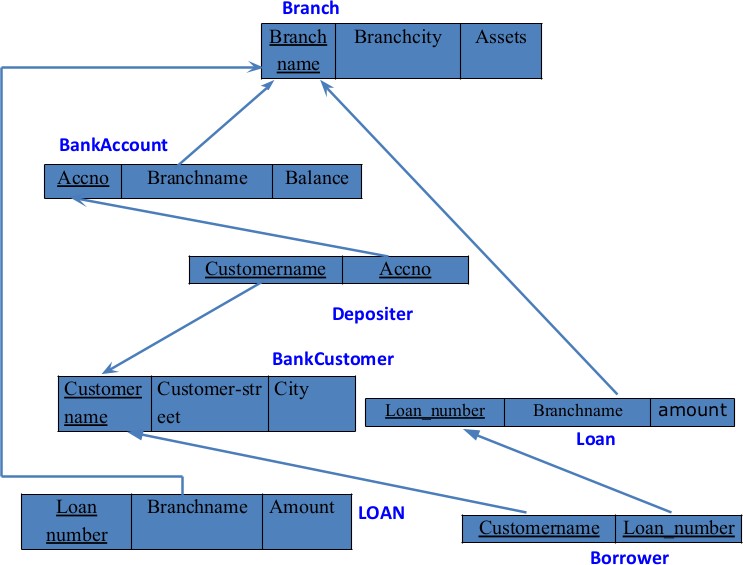


**WEEK 4**

**Bank Database**

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

**SCHEMA DIAGRAM:**



**CREATE COMMAND:**

1. **Table Branch**

create table Branch (branch\_name varchar(30), branch\_city varchar(20), assets real, primary key(branch\_name));

1. **Table BankAccount**

create table BankAccount (accno int, branch\_name varchar(30), balance real, primary key (accno),

foreign key (branch\_name) references Branch(branch\_name));

1. **Table BankCustomer**

create table BankCustomer (customer\_name varchar(20), customer\_street varchar(20),customer\_city varchar(20),primary key(customer\_name));

1. **Table Depositer**

create table Depositer(customer\_name varchar(20), accno int, primary key(customer\_name, accno),

foreign key (customer\_name) references BankCustomer(customer\_name), foreign key (accno) references BankAccount(accno));

1. **Table Loan**

create table LOAN (loan\_number int, branch\_name varchar(30), amount real, primary key(loan\_number),

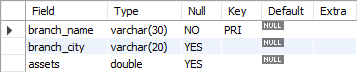
foreign key (branch\_name) references Branch(branch\_name));

1. **Table Borrower**

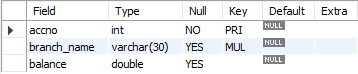
create table Borrower (customer\_name varchar(20), loan\_number int, foreign key (customer\_name) references bankcustomer(customer\_name), foreign key (loan\_number) references LOAN(loan\_number));

# Structure of the table:

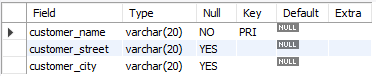
##### Branch Table



1. **BankAccount Table**



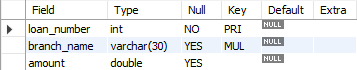
##### Bank Customer Table



1. **Depositer Table**



##### Loan Table



1. **Borrower Table**

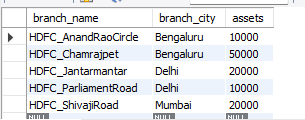


# INSERT COMMAND:

##### Table Branch

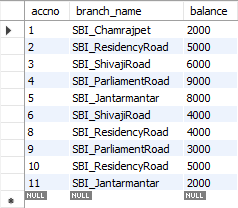
insert into branch

values('HDFC\_Chamrajpet','Bengaluru',50000),('HDFC\_AnandRaoCircle','Bengaluru',10000),('HDFC\_ShivajiRoad','Mumbai',20000),('HDFC\_ParliamentRoad','Delhi',10000),('HDFC\_Jantarmantar','Delhi',20000);



##### Table BankAccount

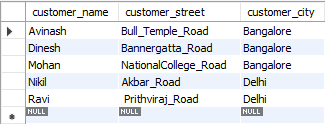
insert into bank\_account

values(1,'HDFC\_Chamrajpet',2000),(2,'HDFC\_AnandRaoCircle',5000),(3,'HDFC\_ShivajiRoad',6000),(4,'HDFC\_ParliamentRoad',9000),(5,'HDFC\_Jantarmantar',8000),(6,'HDFC\_ShivajiRoad',4000),(8,'HDFC\_AnandRaoCircle',4000),(9,'HDFC\_ParliamentRoad',3000),(10,'HDFC\_AnandRaoCircle',5000),(11,'HDFC\_Jantarmantar',2000

##### Table BankCustomer

insert into bank\_customer

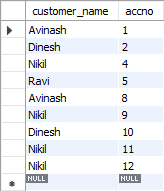
values('Avinash','Bull\_Temple\_Road','Bangalore'),('Dinesh','Bannergatta\_Road','Bangalore'),('Mohan','NationalCollege\_Road','Bangalore'),('Nikil','Akbar\_Road','Delhi'),('Ravi','Prithviraj\_Road','Delhi');



##### Table Depositer

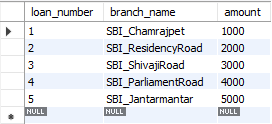
insert into depositer

values('Avinash',1),('Dinesh',2),('Nikil',4),('Ravi',5),('Avinash',8),('Nikil',9),('Dinesh',10),('Nikil',11);



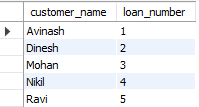
##### Table Loan

insert into loan

values('HDFC\_Chamrajpet',1,1000),('HDFC\_AnandRaoCircle',2,2000),('HDFC\_ShivajiRoad',3,3000),('HDFC\_ParliamentRoad',4,4000),('HDFC\_Jantarmantar',5,5000);

##### Table Borrower

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



# SQL Queries:

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

SQL> select customer\_name from bank\_account ba, depositer d

where branch\_name in (select branch\_name from branch

where branch\_city = "Delhi") and ba.acc\_no = d.acc\_no

group by customer\_name

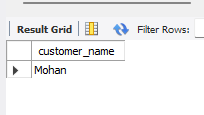
    having count(distinct branch\_name) = (select count(branch\_name) from branch

where branch\_city = "Delhi");



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

SQL> select customer\_name from borrower where customer\_name not in (select customer\_name from depositer);



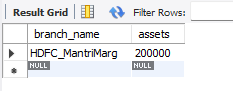
1. Find all customers who have both an account and a loan at the Bangalore branch.

SQL> select distinct customer\_name from depositer where customer\_name in (select b.customer\_name from borrower b,branch bh,loan l where l.loan\_number=b.loan\_number and l.branch\_name=bh.branch\_name and bh.branch\_city='Bengaluru');



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

SQL>select br.branch\_name,br.assets from branch br group by br.branch\_name having br.assets>all(select b.assets from branch b where b.branch\_city='Bengaluru');



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

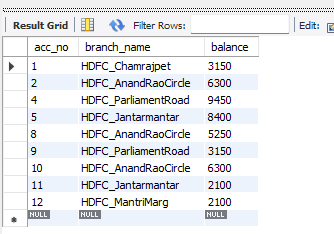
SQL> delete from depositer

where acc\_no in (select acc\_no from bank\_account where branch\_name in (select branch\_name from branch where branch\_city='Mumbai'));

delete from bank\_account

where branch\_name in (select branch\_name from branch where branch\_city='Mumbai');

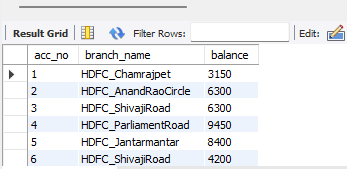
select \* from bank\_account;



1. Update the Balance of all accounts by 5%

SQL> update bank\_account set balance=balance+0.05\*balance;

select \* from bank\_account;

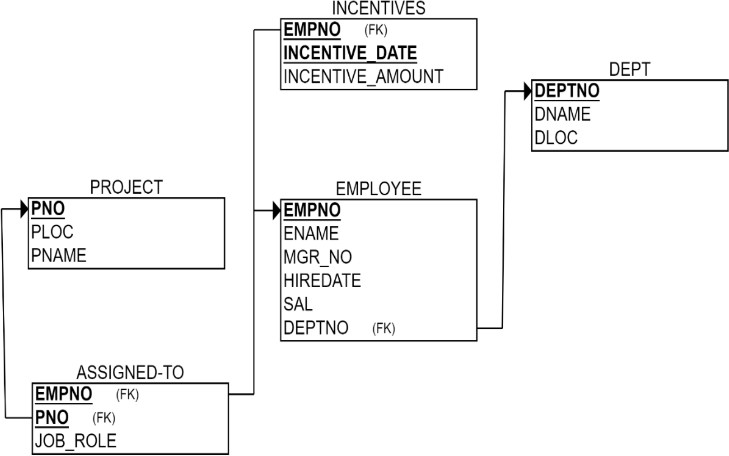


**Week 5**

**Employee Database**

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 COMMAND:**

##### Table Project

create table Project (Pno int, Ploc Varchar(30), Pname varchar(20), primary key(Pno));

##### Table Dept

create table Dept (Deptno int, Dname Varchar(30), Dloc varchar(30), primary key(Deptno));

##### Table Employee

create table Employee (Empno int, Ename varchar(30), Mgr\_No int, Hiredate date, sal int, Deptno int,

primary key (Empno),

foreign key (Deptno) references Dept(Deptno));

##### Table Incentives

create table Incentives (Empno int, Incentive\_Date date, Incentive\_Amount int, primary key (Empno,Incentive\_Date),

foreign key (Empno) references Employee(Empno));

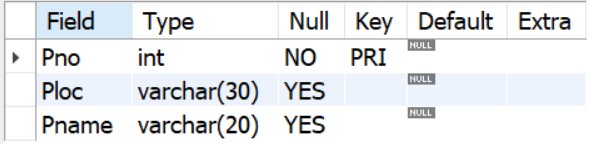
##### Table Assigned\_to

create table Assigned\_To (Empno int, Pno int, Job\_Role varchar(20), primary key (Empno,Pno),

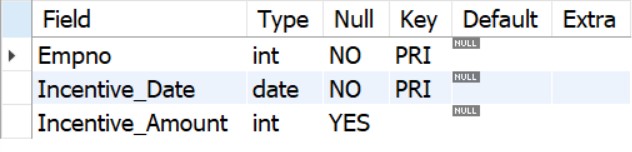
foreign key (Empno) references Employee(Empno), foreign key (Pno) references Project(Pno));

# Structure of the table:

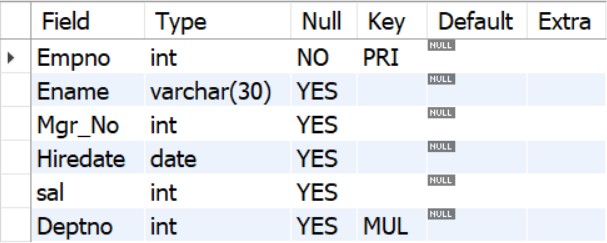
##### Project Table



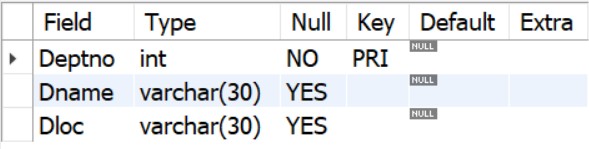
1. **Incentives Table**



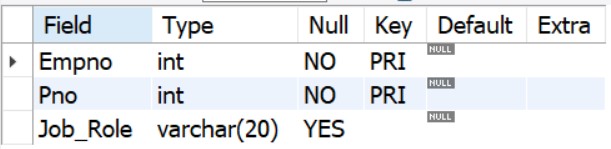
##### Employee Table



1. **Dept Table**



1. **Assigned\_to Table**

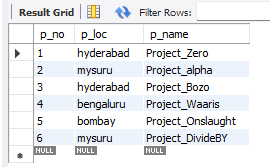


# INSERT COMMAND:

##### Table Project

insert into project

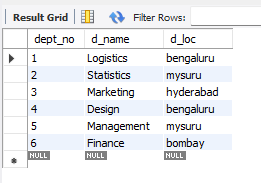
values(01,'hyderabad','Project\_Zero'),(02,'mysuru','Project\_alpha'),(03,'hyderabad','Project\_Bozo'),(04,'bengal uru','Project\_Waaris'),(05,'bombay','Project\_Onslaught'),(06,'mysuru','Project\_DivideBY');



##### Table Dept

insert into department

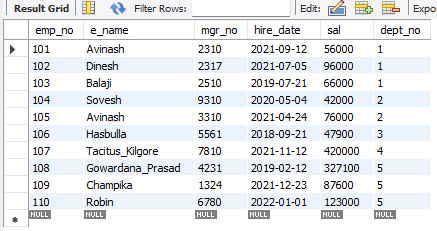
values(1,'Logistics','bengaluru'),(2,'Statistics','mysuru'),(3,'Marketing','hyderabad'),(4,'Design','bengaluru'),(5,' Management','mysuru'),(6,'Finance','bombay');



##### Table Employee

insert into employee

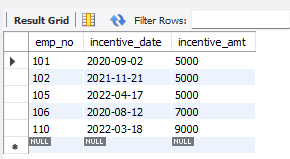
values(101,'Avinash',2310,'2021-09-12',56000,1),(102,'Dinesh',2317,'2021-07- 05',96000,1),(103,'Balaji',2510,'2019-07-21',66000,1),(104,'Sovesh',9310,'2020-05- 04',42000,2),(105,'Avinash',3310,'2021-04-24',76000,2),(106,'Hasbulla',5561,'2018-09- 21',47900,3),(107,'Tacitus\_Kilgore',7810,'2021-11-12',420000,4),(108,'Gowardana\_Prasad',4231,'2019-02- 12',327100,5),(109,'Champika',1324,'2021-12-23',87600,5),(110,'Robin',6780,'2022-01-01',123000,5);



##### Table Incentives

insert into incentives

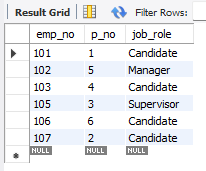
values(101,'2020-09-02',5000),(106,'2020-08-12',7000),(102,'2021-11-21',5000),(105,'2022-04- 17',5000),(110,'2022-03-18',9000);



##### Table Assigned\_To

insert into assigned\_to

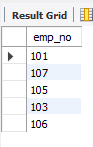
values(101,01,'Candidate'),(103,04,'Candidate'),(102,05,'Manager'),(105,03,'Supervisor'),(107,02,'Ca ndidate'),( 106,06,'Candidate');



# SQL QUERIES:

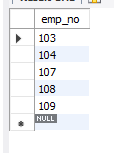
1. Retrieve the employee numbers of all employees who work on project located in Bengaluru, Hyderabad, or Mysuru

SQL> select E.emp\_no from employee E,project P,assigned\_to A where E.emp\_no=A.emp\_no and P.p\_no=A.p\_no and p\_loc in ('bengaluru','mysuru','hyderabad');



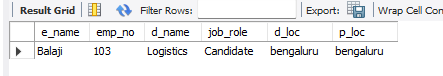
1. Get Employee ID’s of those employees who didn’t receive incentives

SQL>select emp\_no from employee where emp\_no not in (select emp\_no from incentives);



1. 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.

SQL>select E.e\_name,E.emp\_no,D.d\_name,A.job\_role,D.d\_loc,P.p\_loc from employee E,department D,Assigned\_to A,project P where E.emp\_no=A.emp\_no and E.dept\_no=D.dept\_no and A.p\_no=P.p\_no and P.p\_loc=D.d\_loc;

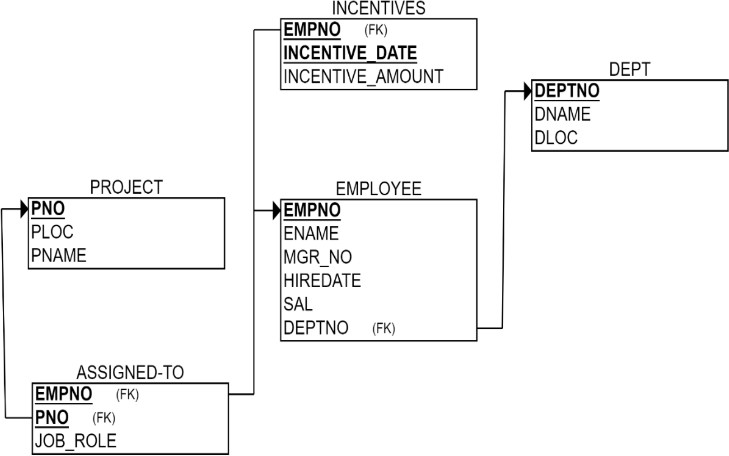


**Week 6**

**Employee Database**

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 COMMAND:**

##### Table Project

create table Project (Pno int, Ploc Varchar(30), Pname varchar(20), primary key(Pno));

##### Table Dept

create table Dept (Deptno int, Dname Varchar(30), Dloc varchar(30), primary key(Deptno));

##### Table Employee

create table Employee (Empno int, Ename varchar(30), Mgr\_No int, Hiredate date, sal int, Deptno int,

primary key (Empno),

foreign key (Deptno) references Dept(Deptno));

##### Table Incentives

create table Incentives (Empno int, Incentive\_Date date, Incentive\_Amount int, primary key (Empno,Incentive\_Date),

foreign key (Empno) references Employee(Empno));

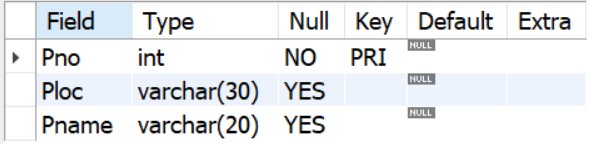
**5.Table Assigned\_to**

create table Assigned\_To (Empno int, Pno int, Job\_Role varchar(20), primary key (Empno,Pno),

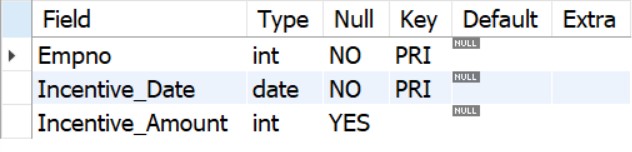
foreign key (Empno) references Employee(Empno), foreign key (Pno) references Project(Pno));

# Structure of the table:

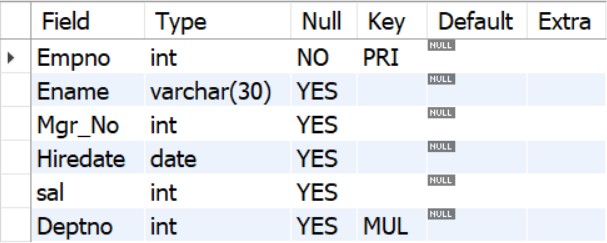
##### Project Table



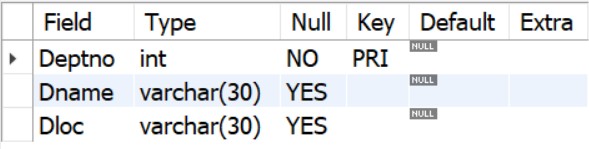
1. **Incentives Table**



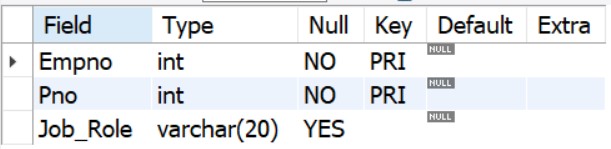
##### Employee Table



1. **Dept Table**



1. **Assigned\_to Table**

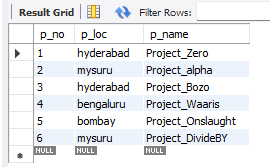


# INSERT COMMAND:

##### Table Project

insert into project

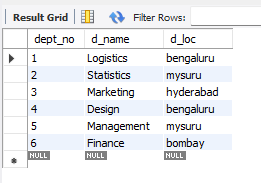
values(01,'hyderabad','Project\_Zero'),(02,'mysuru','Project\_alpha'),(03,'hyderabad','Project\_Bozo'),(0 4,'bengaluru','Project\_Waaris'),(05,'bombay','Project\_Onslaught'),(06,'mysuru','Project\_DivideBY');



##### Table Dept

insert into department

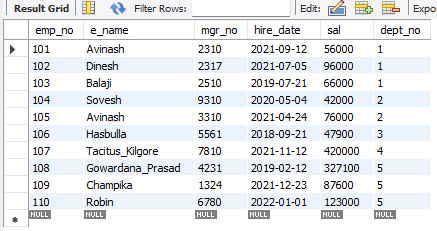
values(1,'Logistics','bengaluru'),(2,'Statistics','mysuru'),(3,'Marketing','hyderabad'),(4,'Design','bengal uru'),(5,'Management','mysuru'),(6,'Finance','bombay');



##### Table Employee

insert into employee

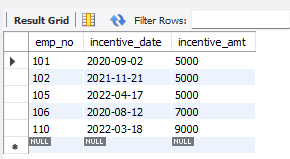
values(101,'Avinash',2310,'2021-09-12',56000,1),(102,'Dinesh',2317,'2021-07- 05',96000,1),(103,'Balaji',2510,'2019-07-21',66000,1),(104,'Sovesh',9310,'2020-05- 04',42000,2),(105,'Avinash',3310,'2021-04-24',76000,2),(106,'Hasbulla',5561,'2018-09- 21',47900,3),(107,'Tacitus\_Kilgore',7810,'2021-11- 12',420000,4),(108,'Gowardana\_Prasad',4231,'2019-02- 12',327100,5),(109,'Champika',1324,'2021-12-23',87600,5),(110,'Robin',6780,'2022-01- 01',123000,5);



##### Table Incentives

insert into incentives

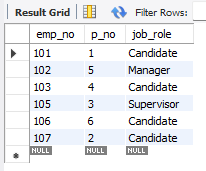
values(101,'2020-09-02',5000),(106,'2020-08-12',7000),(102,'2021-11-21',5000),(105,'2022-04- 17',5000),(110,'2022-03-18',9000);



##### Table Assigned\_To

insert into assigned\_to

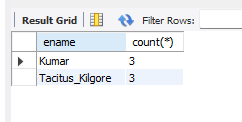
values(101,01,'Candidate'),(103,04,'Candidate'),(102,05,'Manager'),(105,03,'Supervisor'),(107,02,'Ca ndidate'),(106,06,'Candidate');



# SQL QUERIES:

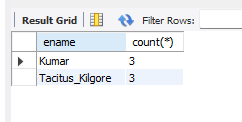
1. List the name of the managers with the maximum employees.

SQL>select m.ename,count(\*) from employee e,employee m where e.mgr\_no=m.emp\_no group by m.ename having count(\*)=(select max(mycount) from (select count(\*) mycount from employee group by mgr\_no) a);



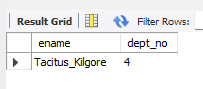
1. Display those managers name whose salary is more than average salary of his employee.

SQL> select E.ename from employee E where E.emp\_no in (select P.mgr\_no from employee P where sal>(select avg(sal) from employee G where G.mgr\_no=E.emp\_no));



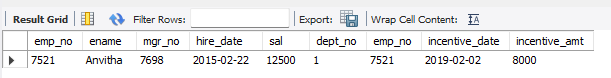
1. Find the name of the second top level managers of each department.

SQL> select G.ename,G.dept\_no from employee E,employee M,employee G where E.mgr\_no=M.emp\_no and M.mgr\_no=G.emp\_no group by G.dept\_no;



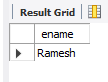
1. Find the employee details who got second maximum incentive in January 2019.

SQL> select \* from employee E,incentives I where E.emp\_no=I.emp\_no and 2=(select count(\*) from incentives J where I.incentive\_amt<=J.incentive\_amt ) and incentive\_date like '2019-02%';



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

SQL> select E.ename from employee E,employee M where E.mgr\_no=M.emp\_no and M.dept\_no=E.dept\_no;

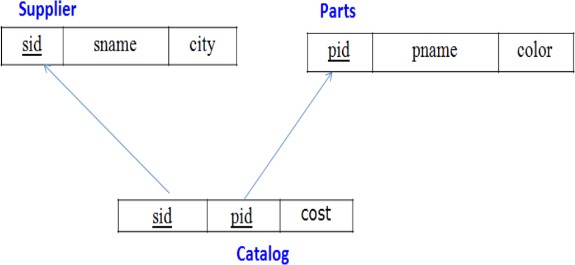


**Week 7**

**Supplier Database**

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 COMMAND:

##### Table Supplier

Create table supplier(sid int, sname varchar(30),city varchar(20), Primary key(sid));

##### Table Parts

Create table Parts(pid int, pname varchar(30),color varchar(20), Primary key(pid));

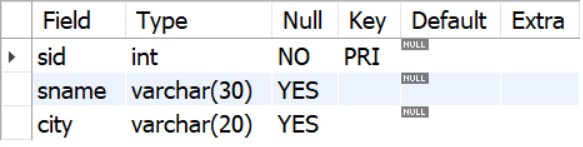
##### Table Catalog

Create table Catalog(sid int, pid int,cost int, Primary key(pid,sid),

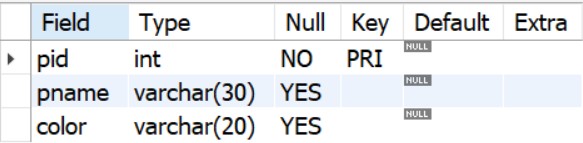
Foreign Key(sid) references supplier(sid), Foreign Key(pid) references parts(pid) );

# Structure of the table:

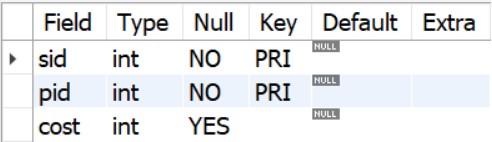
##### Supplier Table



1. **Parts Table**



1. **Catalog Table**



# INSERT COMMAND:

##### Table Supplier

insert into Supplier

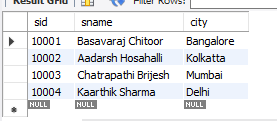
values('10001','Basavaraj Chitoor','Bangalore'),

('10002','Aadarsh Hosahalli','Kolkatta'),

     ('10003','Chatrapathi Brijesh','Mumbai'),

    ('10004','Kaarthik Sharma','Delhi');

select \* from Supplier;



##### Table Parts

insert into parts

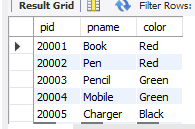
values('20001','Book','Red'),

('20002','Pen','Red'),

     ('20003','Pencil','Green'),

      ('20004','Mobile','Green'),

       ('20005','Charger','Black');



##### Table Catalog

insert into Catalog

values ('10001','20001','10'),

        ('10001','20002','10'),

        ('10001','20003','30'),

        ('10001','20004','10'),

       ('10001','20005','10'),

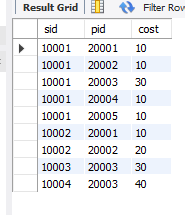
       ('10002','20001','10'),

       ('10002','20002','20'),

        ('10003','20003','30'),

        ('10004','20003','40');

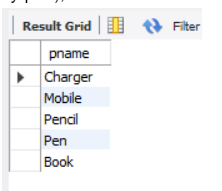
select \* from Catalog;



# QUERIES:

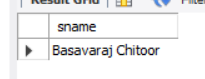
* 1. Find the pnames of parts for which there is some supplier.

SQL> select distinct P.pname from parts P,Catalog C where C.sid=some(select sid from Catalog group by pid );



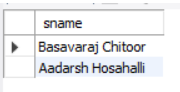
* 1. Find the snames of suppliers who supply every part.

SQL> select S.sname from Supplier S where S.sid in (select C.sid from Catalog C group by C.sid having count(distinct C.pid)=(select count(\*) from parts));



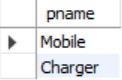
* 1. Find the snames of suppliers who supply every red part.

SQL> select distinct(sname) from Supplier S where not exists(select p.pid,p.color from parts p where not exists(select \* from Catalog C where C.sid=S.sid and p.pid=C.pid) and p.color='Red');



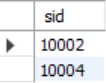
* 1. Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.

SQL> select P.pname from parts P where P.pid in(select C.pid from Catalog C where C.sid=10001 and C.pid not in(select ca.pid from Catalog ca where ca.sid!=10001));



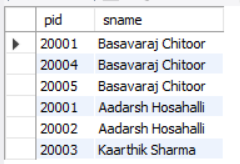
* 1. Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).

SQL> select S.sid from Supplier S where S.sid in (select C.sid from Catalog C where C.cost>(select avg(C1.cost) from Catalog C1 where C1.pid=C.pid group by pid));



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

SQL> select c.pid,s.sname from Supplier s,Catalog c where c.sid=s.sid and c.cost=(select max(c1.cost) from Catalog c1 where c.pid=c1.pid group by c1.pid);

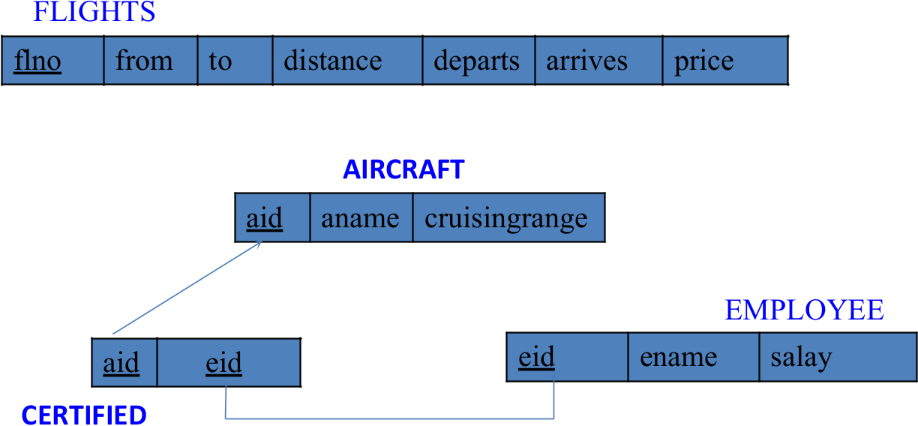


**Week 8**

**Airline Database**

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

**SCHEMA DIAGRAM:**



**CREATE DATABASE**

create database Airline\_Flight;

### CREATE TABLE

##### Flights Table

SQL>Create table Flights(flno int, fr\_om varchar(20), t\_o varchar(20),distance int, departs time, arrives time(0),price int(0), Primary Key (flno));

##### Aircraft Table

SQL>create table aircraft(aid int, aname varchar(20),Cruising\_range int, primary key (aid));

##### Employees Table

SQL>Create table employees( eid int, ename varchar(20), salary int, Primary key (eid));

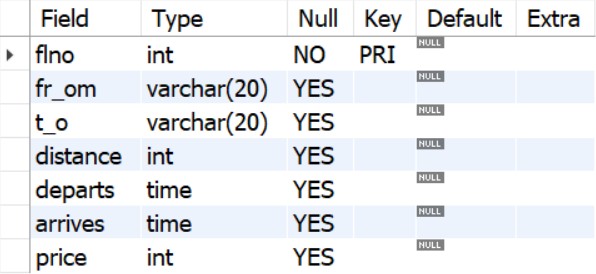
##### Certified Table

SQL> create table certified( eid int, aid int, primary key (eid,aid), foreign key(eid) references employees (eid),

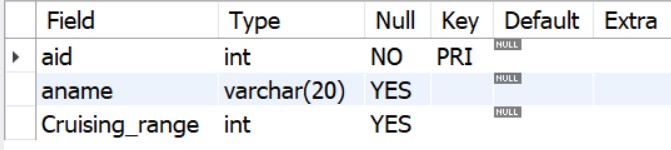
foreign key(aid) references aircraft(aid));

### STRUCTURE OF TABLE:

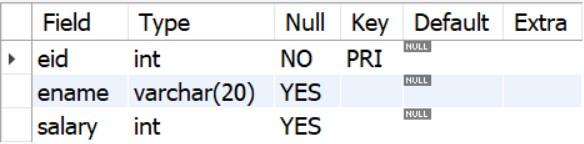
##### Flights Table



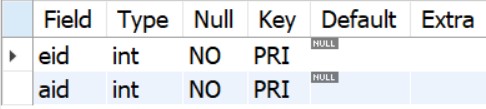
1. **Aircraft Table**



##### Employees Table



1. **Certified Table**

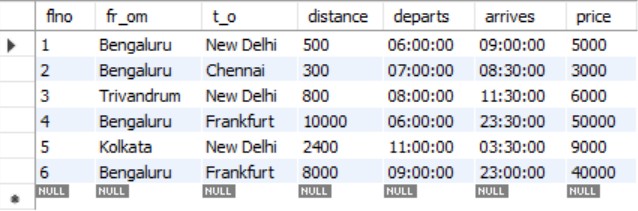


### INSERTING VALUES INTO TABLES:

##### Flights Table

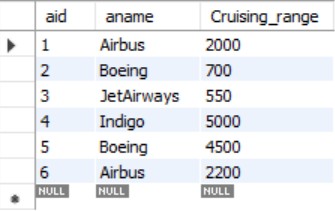
insert into flights

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



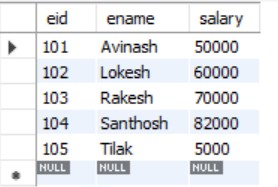
##### Aircraft Table

insert into Aircraft Values (1,'Airbus',2000); insert into Aircraft Values (2,'Boeing',700); insert into Aircraft Values (3,'JetAirways',550); insert into Aircraft Values (4,'Indigo',5000); insert into Aircraft Values (5,'Boeing',4500); insert into Aircraft Values (6,'Airbus',2200);



##### Employees Table

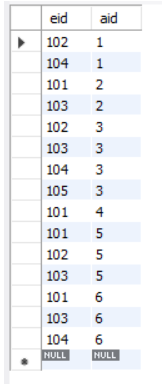
insert into employees values(101,'Avinash',50000); insert into employees values(102,'Lokesh',60000); insert into employees values(103,'Rakesh',70000); insert into employees values(104,'Santhosh',82000); insert into employees values(105,'Tilak',5000);



##### Certified Table

insert into certified

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



**QUERIES:**

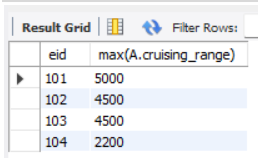
#### Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.

SQL> select A.aname from aircraft A,certified C,employees E where A.aid=C.aid and C.eid=E.eid and E.salary>80000;



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

SQL> select E.eid,max(A.cruising\_range) from employees E,aircraft A,certified C where A.aid=C.aid and C.eid=E.eid group by E.eid having count(C.aid)>=3;



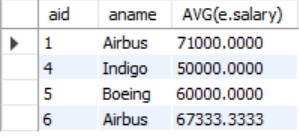
#### Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.

SQL> select ename from employees E where E.salary<(select min(price) from flights F where \_from='Bengaluru' and \_to='Frankfurt');



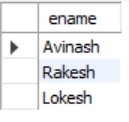
#### For all aircraft with cruising range over 1000 Kms, find the name of the aircraft and the Average salary of all pilots certified for this aircraft.

SQL>select A.aid,A.aname,avg(E.salary) from aircraft A,employees E,certified C where A.aid=C.aid and C.eid=E.eid and A.cruising\_range>1000 group by A.aid;



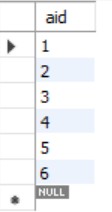
* 1. Find the names of pilots certified for some Boeing aircraft.

SQL> select distinct E.ename from employees E,certified C where C.eid=E.eid and E.eid in(select eid from certified where aid=some(select aid from aircraft where aname='Boeing'));



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

SQL> select A.aid from aircraft A,flights F where F.distance<A.cruising\_range and F.\_from='Bengaluru' and F.\_to='New Delhi';

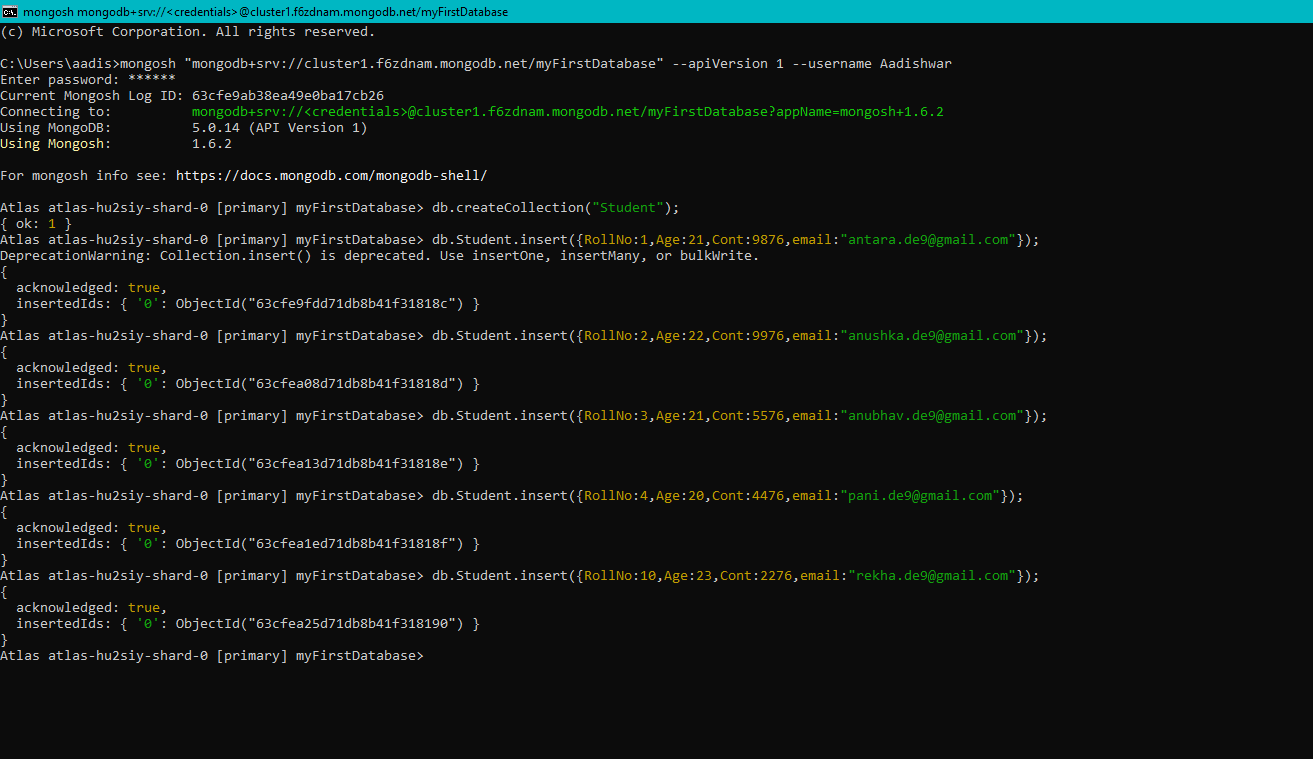


**Week 9**

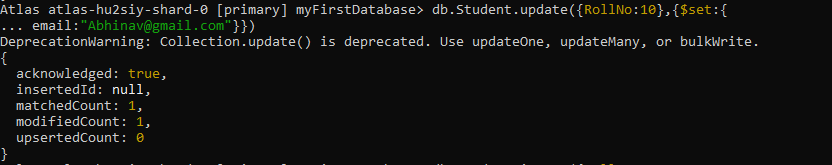
**NOSQL**

Perform the following DB operations using MongoDB.

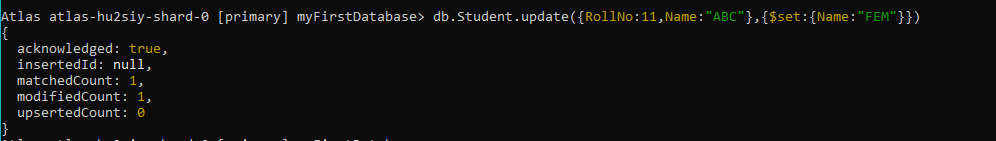
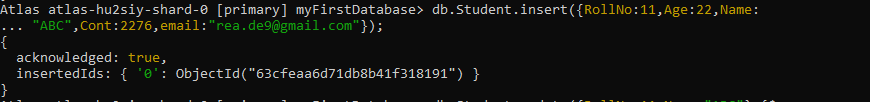
1. Create a database “Student” with the following attributes Rollno, Age, ContactNo, Email-Id.
2. Insert appropriate values



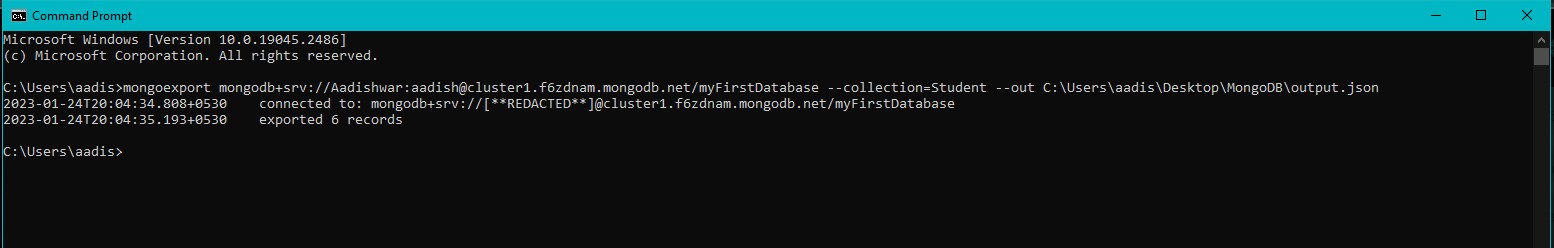
1. Write query to update Email-Id of a student with roll no



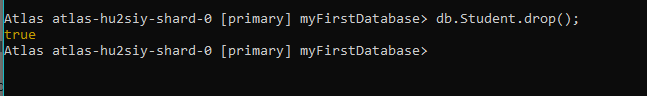
1. Replace the student name from “ABC” to “FEM” of roll no



1. Export the created table into local file system



1. Drop the table



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

