**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

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

****

**LAB REPORT**

**on**

**Database Management Systems (23CS3PCDBM)**

***Submitted by***

**Prathith Suresh Rao (1BM23CS247)**

***in partial fulfilment 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 **Prathith Suresh Rao (1BM23CS247),** who is Bonafede student of **B. M. S. College of Engineering.** It is in partial fulfilment 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.

|  |  |
| --- | --- |
| Lab faculty Incharge Name  Assistant Professor  Department of CSE, BMSCE | Dr. Joythi S Nayak  Professor HOD  Department of CSE, BMSCE |

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.**  **No.** | **Date** | **Experiment Title** | **Page No.** |
| 1 |  | Insurance Database |  |
| 2 |  | More Queries on Insurance Database |  |
| 3 |  | Bank Database |  |
| 4 |  | More Queries on Bank Database |  |
| 5 |  | Employee Database |  |
| 6 |  | More Queries on Employee Database |  |
| 7 |  | Supplier Database |  |
| 8 |  | NO SQL - Student Database |  |
| 9 |  | NO SQL - Customer Database |  |
| 10 |  | NO SQL – Restaurant Database |  |

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)

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

2. Display Accident date and location

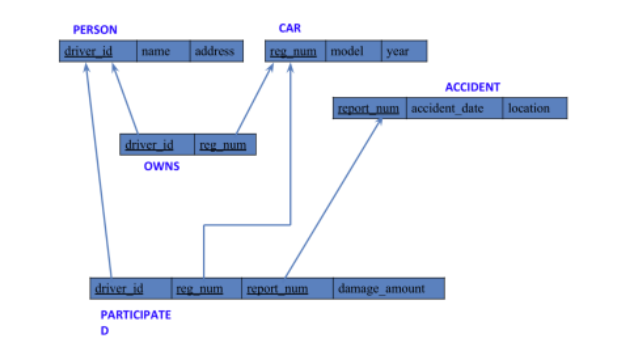
3. Update the damage amount to 25000 for the car with a specific reg\_num (example &#39;KA053408&#39; ) for which the accident report number was 12.

4. Add a new accident to the database.

5. 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 DATABASES:**

create database insurance;

use INSURANCE;

**CREATE TABLES :**

create table person

(Driver\_id varchar(10),

Name varchar(20),

Address varchar(30),

primary key(Driver\_id));

create table car

(reg\_num varchar(10),

Model varchar(10),

year int,

primary key(reg\_num));

create table accident

(Report\_num int,

Accident\_date date,

location varchar(20),

primary key(Report\_num));

create table owns

(Driver\_id varchar(10),

reg\_num varchar(10),

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

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

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

create table participated

(Driver\_id varchar(10),

reg\_num varchar(10),

Report\_num int,

Damage\_amount int,

primary key(Driver\_id, reg\_num, report\_num),

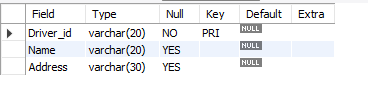
foreign key(Driver\_id) references person(driver\_id),

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

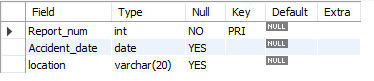
foreign key(report\_num) references accident(report\_num));

**STRUCTURE OF THE TABLE**

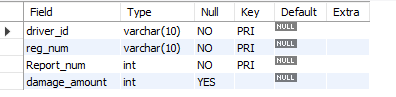
desc person;



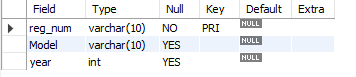
desc accident;

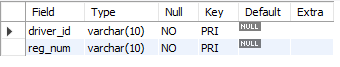


desc participated;



desc car;



desc owns;

**INSERTING VALUES TO THE TABLE**

insert into person values('A01','Richard','Srinivas Nagar');

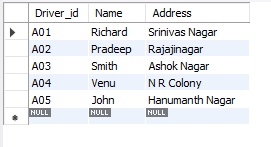
insert into person values('A02','Pradeep','Rajajinagar');

insert into person values('A03','Smith','Ashok Nagar');

insert into person values('A04','Venu','N R Colony');

insert into person values('A05','John','Hanumanth Nagar');

select \* from person;



insert into car values('KA052250','INDICA','1990');

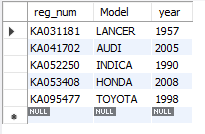
insert into car values('KA031181', 'LANCER','1957');

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 accident values('11','2003-01-01','Mysore Road');

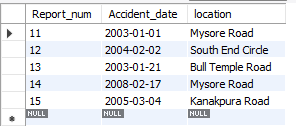
insert into accident values('12', '2004-02-02','South End Circle');

insert into accident values('13','2003-01-21','Bull Temple Road');

insert into accident values('14', '2008-02-17','Mysore Road');

insert into accident values('15','2005-03-04','Kanakpura Road');

select \* from accident;



insert into owns values('A01','KA052250');

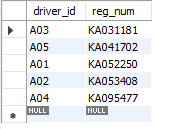
insert into owns values('A02','KA053408');

insert into owns values('A03','KA031181');

insert into owns values('A04','KA095477');

insert into owns values('A05','KA041702');

select \* from owns;



insert into participated values('A01','KA052250','11','10000');

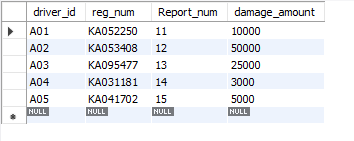
insert into participated values('A02','KA053408','12','50000');

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**

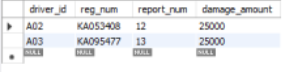
-Update the damage amount to 25000 for the car with a specific reg-num

for which the accident report number was 12.

update participated

set damage\_amount=25000

where reg\_num=’KA053408’ and report\_num=12;



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

select count(distinct driver\_id) CNT

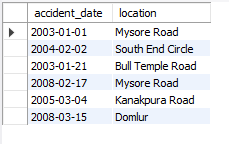
from participated a, accident b

where a.report\_num=b.report\_num and b.accident\_date like ‘2008%’;

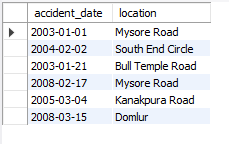


- Display Accident date and location

select accident\_date, location from accident;



- Add a new accident to the database.

insert into accident values ('16','2008-03-15','Domlur');

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

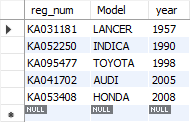
select driver\_id from participated where damage\_amount >=25000;



WEEK 2 (additional queries)

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

select \*from car order by year asc;



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

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



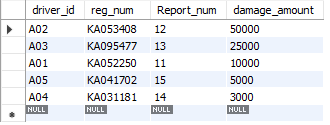
**3)Find the total number of people who owned cars that were involved in accidents in 2008.**

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



**4)LIST THE ENTIRE PARTICIPATED RELATION IN THE DESCENDING ORDER OF DAMAGE AMOUNT.**

select \*from participated order by damage\_amount desc;



**5)FIND THE AVERAGE DAMAGE AMOUNT**

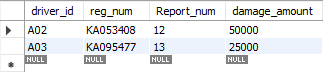
select avg(damage\_amount) from participated;



**6)DELETE THE TUPLE WHOSE DAMAGE AMOUNT IS BELOW THE AVERAGE DAMAGE AMOUNT**

delete from participated where damage\_amount<18600;

select \*from participated;



**7)LIST THE NAME OF DRIVERS WHOSE DAMAGE IS GREATER THAN THE AVERAGE DAMAGE AMOUNT.**

select name from person a, participated b WHERE a.driver\_id = b.driver\_id and damage\_amount > (select avg(damage\_amount) from participated);

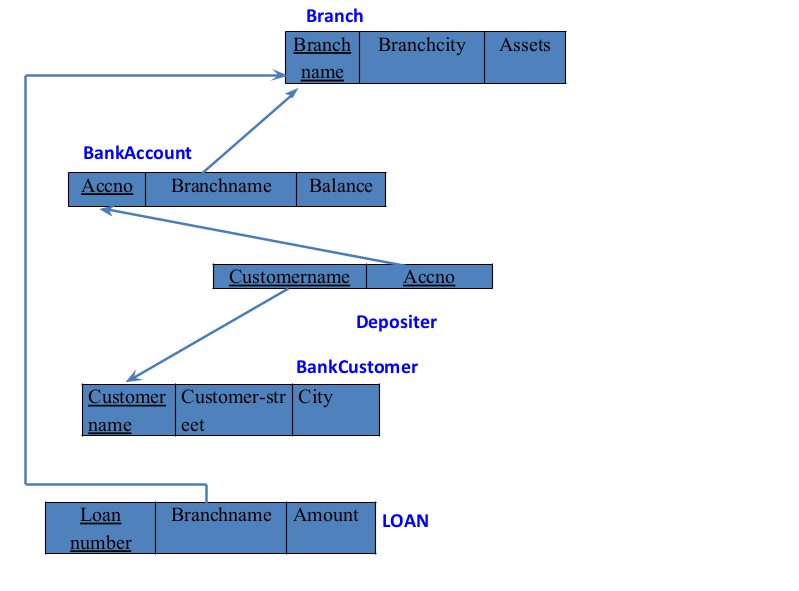


**8)FIND MAXIMUM DAMAGE AMOUNT.**

select max(damage\_amount) from participated;



**Bank database**

****

**Questions**

**(week3)**

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)

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.

**Create Database:**

**create database bank\_247;**

**use bank\_247;**

**Create table:**

**CREATE TABLE branch (**

**branch\_name VARCHAR(30),**

**branch\_city VARCHAR(25),**

**assets INT,**

**PRIMARY KEY (branch\_name)**

**);**

**CREATE TABLE bankaccount (**

**accno INT,**

**branch\_name VARCHAR(30),**

**balance INT,**

**PRIMARY KEY (accno),**

**FOREIGN KEY (branch\_name)**

**REFERENCES branch (branch\_name)**

**);**

**CREATE TABLE bankcustomer (**

**customername VARCHAR(20),**

**customer\_street VARCHAR(30),**

**customer\_city VARCHAR(35),**

**PRIMARY KEY (customername)**

**);**

**CREATE TABLE depositer (**

**customername VARCHAR(20),**

**accno INT,**

**PRIMARY KEY (customername , accno),**

**FOREIGN KEY (accno)**

**REFERENCES bankaccount (accno),**

**FOREIGN KEY (customername)**

**REFERENCES bankcustomer (customername)**

**);**

**CREATE TABLE loan (**

**loan\_number INT,**

**branch\_name VARCHAR(30),**

**amount INT,**

**PRIMARY KEY (loan\_number),**

**FOREIGN KEY (branch\_name)**

**REFERENCES branch (branch\_name)**

**);**

**CREATE TABLE borrower (**

**loan\_number INT,**

**customername VARCHAR(20),**

**PRIMARY KEY (loan\_number),**

**FOREIGN KEY (loan\_number)**

**REFERENCES loan (loan\_number),**

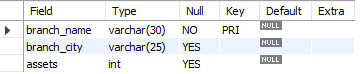
**FOREIGN KEY (customername)**

**REFERENCES bankcustomer (customername)**

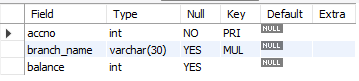
**);**

**Structure of the table:**

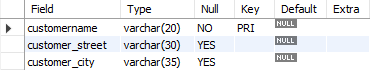
**desc branch;**

****

**desc bankaccount;**

****

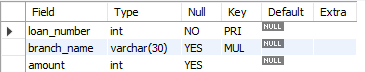
**desc bankcustomer;**

****

**desc depositer;**

****

**desc loan;**

****

**desc borrower;**

****

**Inserting values to the table:**

**insert into branch values("SBI\_Chamrajpet","Bangalore",50000);**

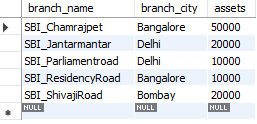
**insert into branch values("SBI\_ResidencyRoad","Bangalore",10000);**

**insert into branch values("SBI\_ShivajiRoad","Bombay",20000);**

**insert into branch values("SBI\_Parliamentroad","Delhi",10000);**

**insert into branch values("SBI\_Jantarmantar","Delhi",20000);**

**select \* from branch;**

****

**insert into bankaccount values(1,"SBI\_Chamrajpet",2000);**

**insert into bankaccount values(2,"SBI\_ResidencyRoad",5000);**

**insert into bankaccount values(3,"SBI\_ShivajiRoad",6000);**

**insert into bankaccount values(4,"SBI\_Parliamentroad",9000);**

**insert into bankaccount values(5,"SBI\_Jantarmantar",8000);**

**insert into bankaccount values(6,"SBI\_ShivajiRoad",4000);**

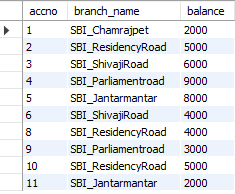
**insert into bankaccount values(8,"SBI\_ResidencyRoad",4000);**

**insert into bankaccount values(9,"SBI\_Parliamentroad",3000);**

**insert into bankaccount values(10,"SBI\_ResidencyRoad",5000);**

**insert into bankaccount values(11,"SBI\_Jantarmantar",2000);**

**select \* from bankaccount;**

****

**insert into bankcustomer values("Avinash","BUll\_temple\_Road","Bangalore");**

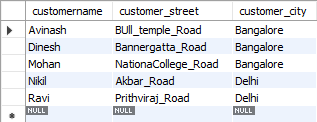
**insert into bankcustomer values("Dinesh","Bannergatta\_Road","Bangalore");**

**insert into bankcustomer values("Mohan","NationaCollege\_Road","Bangalore");**

**insert into bankcustomer values("Nikil","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("Nikil",4);**

**insert into depositer values("Ravi",5);**

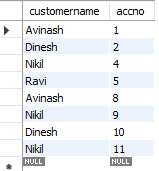
**insert into depositer values("Avinash",8);**

**insert into depositer values("Nikil",9);**

**insert into depositer values("Dinesh",10);**

**insert into depositer values("Nikil",11);**

**select \* from depositer;**

****

**insert into loan values(1,"SBI\_Chamrajpet",1000);**

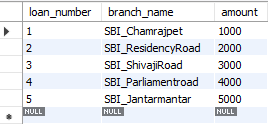
**insert into loan values(2,"SBI\_ResidencyRoad",2000);**

**insert into loan values(3,"SBI\_ShivajiRoad",3000);**

**insert into loan values(4,"SBI\_Parliamentroad",4000);**

**insert into loan values(5,"SBI\_Jantarmantar",5000);**

**select \* from loan;**

****

**insert into borrower values(1,"Mohan");**

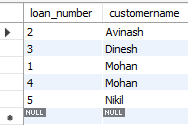
**insert into borrower values(2,"Avinash");**

**insert into borrower values(3,"Dinesh");**

**insert into borrower values(4,"Mohan");**

**insert into borrower values(5,"Nikil");**

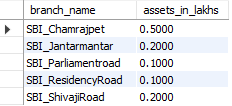
**select \* from borrower;**

****

**Queries:**

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

select branch\_name,(assets/100000) as assets\_in\_lakhs from branch;

****

**iv. 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.branch\_name = 'SBI\_ResidencyRoad'

AND d.accno = b.accno

GROUP BY d.customername

HAVING COUNT(d.accno) >= 2;



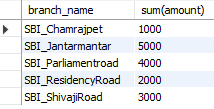
**v. Create a view which gives each branch the sum of the amount of all the loans at the branch.**

create view sum\_of\_loan

as select branch\_name,sum(amount) from loan

group by branch\_name;

select \* from sum\_of\_loan;



Additional Queries on Bank Database

**Questions**

**(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)

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

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

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

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

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

vi. Update the Balance of all accounts by 5%

**Queries:**

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

select distinct s.customername from depositer s

where not exists ((select branch\_name from branch where branch\_city="Delhi") except (select r.branch\_name from depositer t, bankaccount r

where t.accno=r.accno and

s.customername=t.customername));



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

select distinct customername from borrower

where customername not in (select customername from depositer);



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

SELECT DISTINCT

b.customername

FROM

borrower b,

loan l,

depositer d,

branch br

WHERE

b.loan\_number = l.loan\_number

AND l.branch\_name = br.branch\_name

AND br.branch\_city = 'Bangalore'

AND b.customername IN (SELECT

customername

FROM

depositer);



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

SELECT

branch\_name

FROM

branch

WHERE

assets > ALL (SELECT

assets

FROM

branch

WHERE

branch\_city = 'Bangalore');



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

DELETE FROM bankaccount

WHERE

branch\_name IN (SELECT

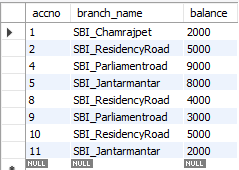
branch\_name

FROM

branch

WHERE

branch\_city = "Bombay");



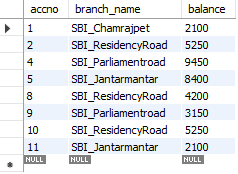
* **Update the Balance of all accounts by 5%**

UPDATE bankaccount

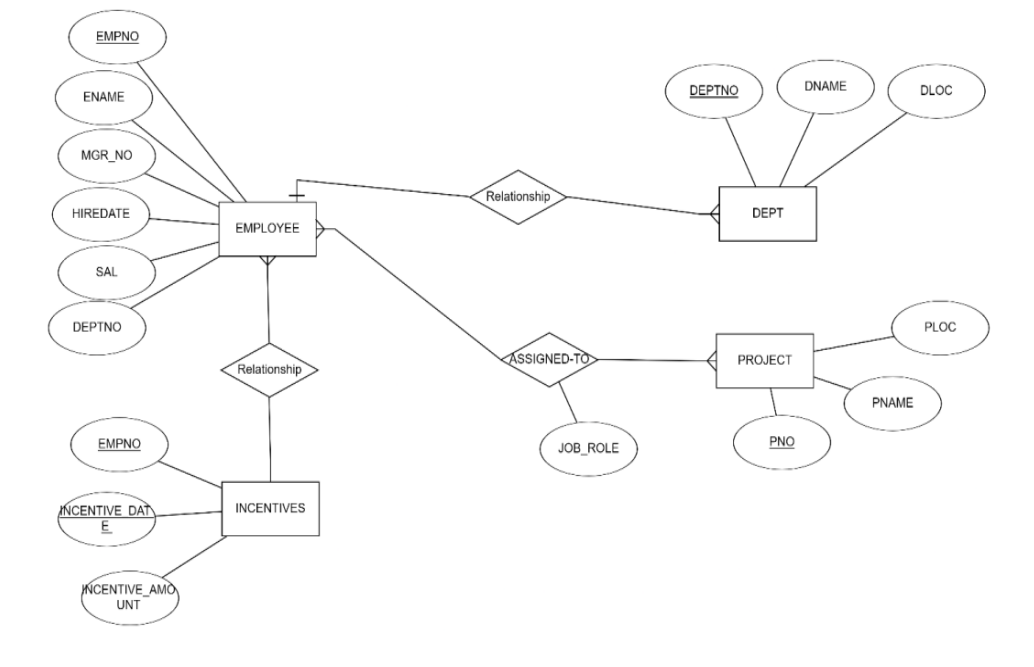
SET

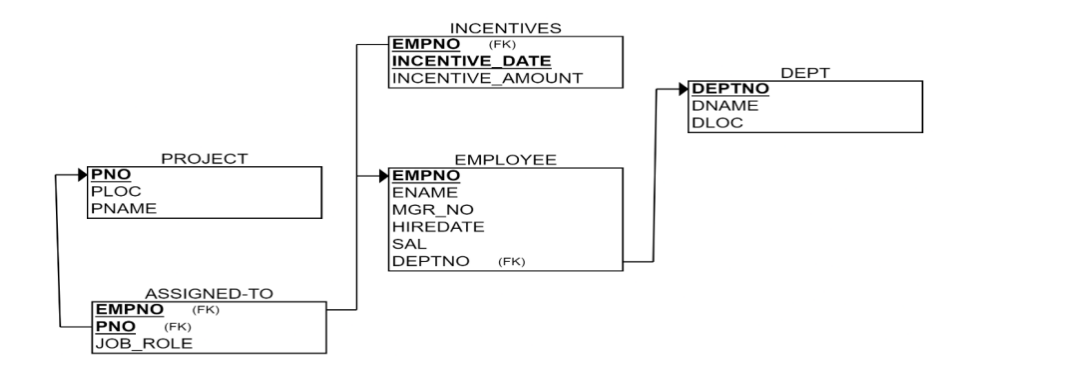
balance = balance \* 1.05;

select \* from bankaccount;



**Employee database**





**Questions(week 5)**

**i. Using Scheme diagram, create tables by properly specifying the primary keys and the foreign keys.**

**ii. Enter greater than five tuples for each table.**

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

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

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

**Create database**

create database emp;

use emp;

**Create tables**

CREATE TABLE dept (

deptno DECIMAL(2 , 0 ) PRIMARY KEY,

dname VARCHAR(14) DEFAULT NULL,

loc VARCHAR(13) DEFAULT NULL

);

CREATE TABLE emp (

empno DECIMAL(4 , 0 ) PRIMARY KEY,

ename VARCHAR(10) DEFAULT NULL,

mgr\_no DECIMAL(4 , 0 ) DEFAULT NULL,

hiredate DATE DEFAULT NULL,

sal DECIMAL(7 , 2 ) DEFAULT NULL,

deptno DECIMAL(2 , 0 ) REFERENCES dept (deptno)

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE incentives (

empno DECIMAL(4 , 0 ) REFERENCES emp (empno)

ON DELETE CASCADE ON UPDATE CASCADE,

incentive\_date DATE,

incentive\_amount DECIMAL(10 , 2 ),

PRIMARY KEY (empno , incentive\_date)

);

CREATE TABLE project (

pno INT PRIMARY KEY,

pname VARCHAR(30) NOT NULL,

ploc VARCHAR(30)

);

CREATE TABLE assigned\_to (

empno DECIMAL(4 , 0 ) REFERENCES emp (empno)

ON DELETE CASCADE ON UPDATE CASCADE,

pno INT REFERENCES project (pno)

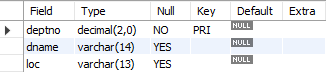
ON DELETE CASCADE ON UPDATE CASCADE,

job\_role VARCHAR(30),

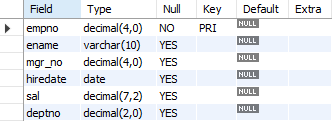
PRIMARY KEY (empno , pno));

**Structure of table**

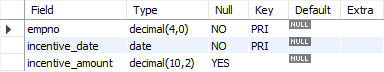
desc dept;



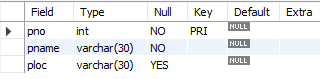
desc emp;



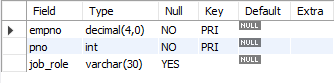
desc incentives;



desc project;



desc assigned\_to;



**Inserting values into tables**

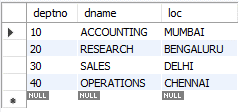
INSERT INTO dept VALUES (10, 'ACCOUNTING', 'MUMBAI');

INSERT INTO dept VALUES (20, 'RESEARCH', 'BENGALURU');

INSERT INTO dept VALUES (30, 'SALES', 'DELHI');

INSERT INTO dept VALUES (40, 'OPERATIONS', 'CHENNAI');

select \* from dept;



INSERT INTO emp VALUES (7369, 'Adarsh', 7902, '2012-12-17', '80000.00', '20');

INSERT INTO emp VALUES (7499, 'Shruthi', 7698, '2013-02-20', '16000.00', '30');

INSERT INTO emp VALUES (7521, 'Anvitha', 7698, '2015-02-22', '12500.00', '30');

INSERT INTO emp VALUES (7566, 'Tanvir', 7839, '2008-04-02', '29750.00', '20');

INSERT INTO emp VALUES (7654, 'Ramesh', 7698, '2014-09-28', '12500.00', '30');

INSERT INTO emp VALUES (7698, 'Kumar', 7839, '2015-05-01', '28500.00', '30');

INSERT INTO emp VALUES (7782, 'CLARK', 7839, '2017-06-09', '24500.00', '10');

INSERT INTO emp VALUES (7788, 'SCOTT', 7566, '2010-12-09', '30000.00', '20');

INSERT INTO emp VALUES (7839, 'KING', NULL, '2009-11-17', '90000', '10');

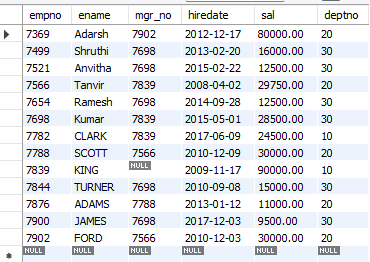
INSERT INTO emp VALUES (7844, 'TURNER', 7698, '2010-09-08', '15000.00', '30');

INSERT INTO emp VALUES (7876, 'ADAMS', 7788, '2013-01-12', '11000.00', '20');

INSERT INTO emp VALUES (7900, 'JAMES', 7698, '2017-12-03', '9500.00', '30');

INSERT INTO emp VALUES (7902, 'FORD', 7566, '2010-12-03', '30000.00', '20');

select \* from emp;



INSERT INTO incentives VALUES(7499, '2019-02-01', 5000.00);

INSERT INTO incentives VALUES(7521, '2019-03-01', 2500.00);

INSERT INTO incentives VALUES(7566, '2022-02-01', 5070.00);

INSERT INTO incentives VALUES(7654, '2020-02-01', 2000.00);

INSERT INTO incentives VALUES(7654, '2022-04-01', 879.00);

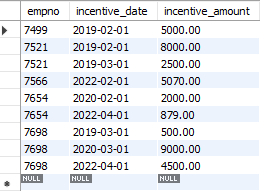
INSERT INTO incentives VALUES(7521, '2019-02-01', 8000.00);

INSERT INTO incentives VALUES(7698, '2019-03-01', 500.00);

INSERT INTO incentives VALUES(7698, '2020-03-01', 9000.00);

INSERT INTO incentives VALUES(7698, '2022-04-01', 4500.00);

select \* from incentives;



INSERT INTO project VALUES(101, 'AI Project', 'BENGALURU');

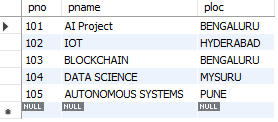
INSERT INTO project VALUES(102, 'IOT', 'HYDERABAD');

INSERT INTO project VALUES(103, 'BLOCKCHAIN', 'BENGALURU');

INSERT INTO project VALUES(104, 'DATA SCIENCE', 'MYSURU');

INSERT INTO project VALUES(105, 'AUTONOMOUS SYSTEMS', 'PUNE');

select \* from project;



INSERT INTO assigned\_to VALUES(7499, 101, 'Software Engineer');

INSERT INTO assigned\_to VALUES(7521, 101, 'Software Architect');

INSERT INTO assigned\_to VALUES(7566, 101, 'Project Manager');

INSERT INTO assigned\_to VALUES(7654, 102, 'Sales');

INSERT INTO assigned\_to VALUES(7521, 102, 'Software Engineer');

INSERT INTO assigned\_to VALUES(7499, 102, 'Software Engineer');

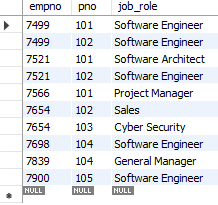
INSERT INTO assigned\_to VALUES(7654, 103, 'Cyber Security');

INSERT INTO assigned\_to VALUES(7698, 104, 'Software Engineer');

INSERT INTO assigned\_to VALUES(7900, 105, 'Software Engineer');

INSERT INTO assigned\_to VALUES(7839, 104, 'General Manager');

select \* from assigned\_to;



**Queries:**

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

SELECT

e.empno

FROM

emp e,

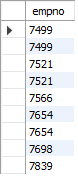
project p,

assigned\_to a

WHERE

e.empno = a.empno AND a.pno = p.pno

AND p.ploc IN ('BENGALURU' , 'HYDERABAD', 'MYSURU');



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

SELECT

empno

FROM

emp

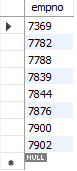
WHERE

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.empno , e.ename,d.dname,d.loc,a.job\_role,p.ploc

from emp e,dept d,assigned\_to a,project p

where e.deptno=d.deptno and

e.empno=a.empno and

a.pno=p.pno and d.loc=p.ploc;

