

DBMS-Lab Report-1

Name: Piyush Dubey

Date: 31-05-2021

U.S.N.: 1BM19CS221.

LAB-1:

```
CREATE TABLE PERSON(driver_id char(10), Name char(20), address char(40), PRIMARY KEY(driver_id));
CREATE TABLE CAR (Regno char(10), model char(10), year int, PRIMARY KEY(Regno));
CREATE TABLE ACCIDENT(report_number int, datex date, location char(40), PRIMARY
KEY(report_number));
CREATE TABLE OWNS(driver_id char(10), Regno char(10), PRIMARY KEY(driver_id, Regno), FOREIGN
KEY(driver_id) REFERENCES PERSON(driver_id), FOREIGN KEY(Regno) REFERENCES CAR(Regno));
CREATE TABLE PARTICIPATED (driver_id char(10), Regno char(10), report_number int, damage_amount
int, PRIMARY KEY(driver_id, Regno, report_number), FOREIGN KEY(driver_id) REFERENCES
PERSON(driver_id), FOREIGN KEY(Regno) REFERENCES CAR(Regno), FOREIGN KEY(report_number) REFERENCES
ACCIDENT(report_number));

INSERT INTO person VALUES('A10','James','Renukanagar');
INSERT INTO person VALUES('A14','Rishabh Pant','Srinagar');
INSERT INTO person VALUES('B33','David','Mumbai');
INSERT INTO person VALUES('B56','Tom ','Gopal Nagar');
INSERT INTO person VALUES('C14','Ronith ','Toy Town');
commit;
SELECT*FROM person;

INSERT INTO car VALUES('KA690','Nano',2006);
INSERT INTO car VALUES('KA466','Indica',2000);
INSERT INTO car VALUES('BR720','Sumo',2010);
INSERT INTO car VALUES('CK144','Alto',2014);
INSERT INTO car VALUES('RL221','Zing',2015);
commit;
SELECT*FROM car;

INSERT INTO accident VALUES(123,'2001-01-04','Delhi');
INSERT INTO accident VALUES(456,'2008-06-08','Kolkata');
INSERT INTO accident VALUES(789,'2004-04-10','Bangalore');
INSERT INTO accident VALUES(480,'2012-12-15','Jaipur');
INSERT INTO accident VALUES(921,'2003-08-20','Mumbai');
commit;
select*from accident;

INSERT INTO owns VALUES('A01','BR720');
INSERT INTO owns VALUES('A14','CK144');
```

```

INSERT INTO owns VALUES('B33','KA466');
INSERT INTO owns VALUES('B56','KA690');
INSERT INTO owns VALUES('C14','RL221');
commit;
SELECT*FROM owns;

INSERT INTO participated VALUES('A01','BR720',123,400);
INSERT INTO participated VALUES('A14','CK144',456,1000);
INSERT INTO participated VALUES('B33','KA466',480,20);
INSERT INTO participated VALUES('B56','KA690',789,90000);
INSERT INTO participated VALUES('C14','RL221',921,1500);
commit;
SELECT*from participated;

```

Query:

1.Update the damage amount for the car with a specific Regno in the accident with report number 12 to 25000.

```
UPDATE participated SET damage_amount=25000 WHERE report_number=123 AND Regno='BR720';
```

2. Add a new accident to the database.

```
INSERT INTO accident values(810, "2008-04-10", "Chandigardh");
```

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

```
SELECT COUNT(driver_id) FROM participated x,accident y WHERE x.report_number=y.report_number
and year(y.datex)=2008;
```

OR

```
SELECT COUNT(driver_id) FROM participated x,accident y WHERE x.report_number=y.report_number
and year(y.datex)=2008;
```

4.Find the number of accidents in which cars belonging to a specific model were involved.

```
SELECT COUNT(driver_id) FROM owns x, car y WHERE x.Regno=y.Regno and y.model='Nano';
```

OR

```
SELECT COUNT(report_number) FROM accident WHERE report_number IN(SELECT report_number FROM participated WHERE Regno IN (SELECT Regno FROM car WHERE model='Sumo'))
```

LAB-2

```

CREATE TABLE branch(branch_name char(30), branch_city char(30), assets real, PRIMARY KEY(branch_name));

CREATE TABLE bank_account(accno int, branch_name char(30), balance real, PRIMARY KEY(accno), FOREIGN
KEY(branch_name) REFERENCES branch(branch_name));

CREATE TABLE bank_customer(customer_name char(30), customer_street char(30), customer_city char(30));

```

```
CREATE TABLE depositor(customer_name char(30), accno int, FOREIGN KEY(accno) REFERENCES
bank_account(accno));
```

```
CREATE TABLE loan(loan_number int, branch_name char(30), amount real, PRIMARY KEY(loan_number), FOREIGN
KEY(branch_name) REFERENCES branch(branch_name));
```

```
INSERT INTO branch values('SBI_Basvangudi', 'Bangalore', 50000);
```

```
INSERT INTO branch values('BOI_Sakchit', 'Jamshedpur', 2000);
```

```
INSERT INTO branch values('SBI_Chamrajpet', 'Bangalore', 900000);
```

```
INSERT INTO branch values('HDFC_Tirupathur', 'Vellore', 6000);
```

```
INSERT INTO branch values('PNB_Banashankari', 'Bangalore', 10000);
```

```
Commit;
```

```
SELECT * FROM branch;
```

```
INSERT INTO loan values(1,'SBI_Basvangudi', 2000);
```

```
INSERT INTO loan values (3,'BOI_Sakchit', 100000);
```

```
INSERT INTO loan VALUES (2, 'SBI_Chamrajpet', 300);
```

```
INSERT INTO loan VALUES (4, 'HDFC_Tirupathur', 500);
```

```
INSERT INTO loan VALUES (5, 'PNB_Banashankari', 40000);
```

```
Commit;
```

```
SELETCT * FROM loan;
```

```
INSERT INTO bank_account VALUES(12, 'BOI_Sakchit', 1230);
```

```
INSERT INTO bank_account VALUES(23, 'HDFC_Tirupathur', 4220);
```

```
INSERT INTO bank_account VALUES(51, 'PNB_Banashankari', 9820);
```

```
INSERT INTO bank_account VALUES(109, 'SBI_Chamrajpet', 14000);
```

```
INSERT INTO bank_account VALUES(84, 'SBI_Basvangudi', 121);
```

```
INSERT INTO bank_account VALUES(123, 'SBI_Basvangudi', 13120);
```

```
INSERT INTO bank_account VALUES(13, 'SBI_Basvangudi', 2120);
```

```
INSERT INTO bank_account VALUES(902, 'SBI_Basvangudi', 13120);
```

```
INSERT INTO bank_account VALUES(832, 'SBI_Basvangudi', 13120);
```

```
INSERT INTO bank_account VALUES(119,"SBI_Chamrajpet",12412);

INSERT INTO bank_account VALUES(39,"HDFC_Tirupathur",1221);

INSERT INTO bank_account VALUES(190,"BOI_Sakchit",1212412);

Commit;

SELECT * FROM bank_account;
```

```
INSERT INTO bank_customer VALUES("Raju", "MG","Bangalore");

INSERT INTO bank_customer VALUES("Shashank","Jehangir","Ranchi");

INSERT INTO bank_customer VALUES("Abishek","Marine Drive","Delhi");

INSERT INTO bank_customer VALUES("Diwakar","14th Milestone","Jamshedpur");

INSERT INTO bank_customer VALUES("Sid","Teram","Vellore");

Commit;

Select * from bank_customer;
```

```
INSERT INTO depositor VALUES("Raju",12);

INSERT INTO depositor VALUES("Shashank",23);

INSERT INTO depositor VALUES("Abishek",51);

INSERT INTO depositor VALUES("Diwakar",109);

INSERT INTO depositor VALUES("Sid",84);

INSERT INTO depositor VALUES("Raju",123);

INSERT INTO depositor VALUES("Raju",832);

INSERT INTO depositor VALUES("Abishek",902);

INSERT INTO depositor VALUES("Abishek",39)

INSERT INTO depositor VALUES("Abishek",190)

INSERT INTO depositor VALUES("Abishek",119)
```

```
Commit;
```

```
Select * from depositor;
```

QUERY:

i. Create the above tables by properly specifying the primary keys and the foreign keys

ii. Enter at least five tuples for each relation.

iii. Find all the customers who have at least two accounts at the Main branch.

```
SELECT customer_name FROM depositor d, bank_account a where d.accno=a.accno AND  
a.branch_name="SBI_Basvangudi" GROUP BY d.customer_name HAVING COUNT(d.customer_name)>=2;
```

iv. Find all the customers who have an account at all the branches located in a specific city.

```
SELECT d.customer_name FROM bank_account a,branch b,depositor d WHERE  
b.branch_name=a.branch_name AND a.accno=d.accno AND b.branch_city='Bangalore' GROUP BY  
d.customer_name HAVING COUNT(distinct b.branch_name)=(SELECT COUNT(branch_name) FROM branch  
WHERE branch_city='Bangalore');
```

OR

```
SELECT d.customer_name FROM depositor d, bank_account b, branch c WHERE b.accno=d.accno AND c.  
branch_city='Bangalore' GROUP BY d.customer_name HAVING COUNT(d.accno)=(SELECT COUNT(branch_na  
me) FROM branch WHERE branch_city='Bangalore')
```

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

```
DELETE FROM bank_account WHERE branch_name IN(SELECT branch_name FROM branch WHERE  
branch_city='Jamshedpur');
```

LAB-3:

```
CREATE TABLE parts(pid int, pname char(30), color char(30), PRIMARY KEY(pid));  
CREATE TABLE catalog(sid int, pid int, cost int, foreign key(sid) references  
supplier(sid), foreign key(pid) references parts(pid));
```

```
INSERT INTO supplier VALUES((110,'Ananad pvt ltd','Jamshedpur'),  
                             (123,'Shrinivas hardware','Bangalore'),  
                             (785,'Raghav industries','Vellore'),  
                             (345,'Bablu enterpries','Delhi'),  
                             (876,'Ramnihal motoparts','Bombay'));
```

```
INSERT INTO parts VALUES(01,'Screw','red'),  
                          (02,'Bearing','red'),  
                          (03,'Nut','grey'),  
                          (10,'Bolt','red'),  
                          (27,'panner','grey'),  
                          (32,'rubbergrip',red),
```

```
(89,'rubbergrip','grey');
INSERT INTO catalog
VALUES(110,01,23),(110,02,45),(110,03,86),(110,10,73),(110,27,93),(110,32,20),(110,89,120);
INSERT INTO catalog VALUES(123,01,43),(123,02,18),(123,10,33),(123,32,32);
INSERT INTO catalog VALUES(785,01,13),(785,32,62);
INSERT INTO catalog VALUES(345,01,53),(345,27,45),(345,32,82),(345,89,109);
INSERT INTO catalog VALUES(876,02,25),(876,03,72),(876,32,23);
```

Queries:

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

```
SELECT DISTINCT P.pname FROM parts P, catalog C WHERE P.pid = C.pid;
```

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

```
SELECT S.sname FROM supplier S WHERE NOT EXISTS ((SELECT P.pid FROM parts P)
EXCEPT(SELECT C.pid FROM catalog C WHERE C.sid = S.sid ));
```

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

```
SELECT S.sname FROM supplier S WHERE NOT EXISTS((SELECT P.pid FROM parts P WHERE
P.color='red') EXCEPT (SELECT C.pid FROM catalog C, parts P WHERE C.sid = S.sid AND C.pid
= P.pid AND P.color = 'red'));
```

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

```
SELECT DISTINCT pname FROM catalog x, parts y, supplier z WHERE(z.sname='Ananad pvt ltd'
AND z.sid=x.sid);
```

v. 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 DISTINCT c.sid FROM catalog c WHERE c.cost>(SELECT AVG(c1.cost) FROM catalog c1
WHERE c1.pid=c.pid);
```

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

```
SELECT P.pid, S.sname FROM parts P, supplier S, catalog C WHERE C.pid = P.pid AND C.sid =
S.sid AND C.cost = (SELECT MAX(C1.cost) FROM catalog C1 WHERE C1.pid = P.pid);
```

vii. Find the sids of suppliers who supply only red parts

```
SELECT DISTINCT c.sid FROM catalog c WHERE NOT EXISTS(SELECT * FROM parts p WHERE
p.pid=c.pid and p.color<>'red');
```

LAB-4:

```
CREATE database studentfacultydb;
```

```
use studentfacultydb;
```

```
CREATE TABLE student(snum int, sname char(30), major char(30), level char(30), age int, PRIMARY
```

```
KEY(snum));

CREATE TABLE faculty(fid int, fname char(30), deptid int, PRIMARY KEY(fid));

CREATE TABLE enrolled(snum int, name char(30), FOREIGN KEY(snum) REFERENCES student(snum));

CREATE TABLE class(name char(30), meet time, room char(30), fid int, FOREIGN KEY(fid) REFERENCES
faculty(fid));

INSERT INTO student (snum,sname,major,level,age) VALUES (1,'jhon','CS','Sr',19),

(2,'smith','CS','Jr',20),

(3,'jacob','CV','Sr',20),

(4,'tom','CS','Jr',20),

(5,'sid','CS','Jr',20),

(6,'harry','CS','Sr',21);

INSERT INTO faculty (fid,fname, deptid) VALUES

(11,'Harshith',1000),

(12,'Mohan',1000),

(13,'Kumar',1001),

(14,'Shobha',1002),

(15,'Shan',1000);

INSERT INTO class (name,meet,room,fid) VALUES

('class1','12:00:00','room1',14),

('class10','05:00:00','room128',14),

('class2','08:00:00','room2',12),

('class3','07:00:00','room3',11),

('class4','18:00:00','room4',14),

('class5','20:00:00','room3',15),

('class6','08:00:00','room2',14),

('class7','19:00:00','room3',14);
```

```

INSERT INTO enrolled (snum,name) VALUES (1,'class1'),

(2,'class1'),

(3,'class3'),

(4,'class3'),

(3,'class3'),

(5,'class4'),

(1,'class5'),

(2,'class5'),

(3,'class5'),

(4,'class5'),

(5,'class5'),

(6,'class5');

```

Query 1: Find the names of all juniors (level=Jr) who are enrolled for class taught by professor Harshith.

```

SELECT DISTINCT s.sname FROM student s,class c,faculty f,enrolled e WHERE s.snum=e.snum AND
e.name=c.name AND s.level='jr' AND f.fname='Harshith' AND f.fid=c.fid;

```

Query 2: Find the names of all classes that either meet in room128 or have 5 or more students enrolled.

```

SELECT DISTINCT name FROM class WHERE room='room128' OR name IN (SELECT e.name FROM enrolled e
GROUP BY e.name HAVING COUNT(*)>=5);

```

Query3: Find the names of all students who are enrolled in two classes that meet at same time.

```

SELECT DISTINCT s.sname FROM student s WHERE s.snum IN (SELECT e1.snum FROM enrolled e1,enrolled
e2,class c1,class c2 WHERE e1.snum=e2.snum AND e1.name<>e2.name AND e1.name=c1.name AND
e2.name=c2.name AND c1.meet=c2.meet);

```

Query 4: Find the names of faculty members who teach in every room in which some class is taught.

```

SELECT f.fname,f.fid FROM faculty f WHERE f.fid in (SELECT fid FROM class GROUP BY fid HAVING
COUNT(*)=(SELECT COUNT(DISTINCT room) FROM class));

```

Query 5: Find the names of the faculty members for whome the combined enrollment of the classes that they teach is less then five.

```

SELECT DISTINCT f.fname FROM faculty f WHERE f.fid IN ( SELECT c.fid FROM class c, enrolled e
WHERE c.name = e.name GROUP BY c.name HAVING COUNT(c.name)< 5 );

```

Query 6: Find the names of students who are not enrolled in any class.

```

SELECT sname FROM student where snum not in(SELECT snum from enrolled);

```

Query 7: For each age value that appears in Students, find the level value that appears most often. For example, if

there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18,FR)

```
select s.age, s.level from student s group by s.age,s.level having s.level in (select s1.level
from student s1 where s1.age = s.age group by s1.level,s1.age having count(*) >= all (select
count(*) from student s2 where s1.age = s2.age group by s2.level,s2.age))
```

LAB-5:

```
CREATE database airlineflightdb;
CREATE TABLE flights(flno int, frm char(255), too char(255), distance int, departs time, arrives
time, price integer);
```

```
CREATE TABLE flights(flno int, frm char(255), distance int, departs time, arrives time, price
integer, PRIMARY KEY(flno));
CREATE TABLE aircraft(aid int, anmae char(255), cruisingrange int,PRIMARY KEY(aid));
CREATE TABLE employee(eid int, ename char(255), salary int, PRIMARY KEY(eid));
CREATE TABLE certified(eid int, aid int, FOREIGN KEY (eid) REFERENCES employee(eid), FOREIGN KEY
(aid) REFERENCES aircraft(aid));
```

```
INSERT INTO employees (eid,ename,salary) VALUES
    (1,'Ajay',30000),
    (2,'Ajith',85000),
    (3,'Arnab',50000),
    (4,'Harry',45000),
    (5,'Ron',90000),
    (6,'Josh',75000),
    (7,'Ram',100000);
(8,"Rishabh",12000);
```

```
INSERT INTO aircraft
(aid,anmae,cruisingrange) values
(123,'Airbus',1000),
(302,'Boeing',5000),
(306,'Jet01',5000),
(378,'Airbus380',8000),
(456,'Aircraft',500),
(789,'Aircraft02',800),
(951,'Aircraft03',1000);
```

```
INSERT INTO certified
(eid,aid) VALUES
(1,123),
(2,123),
(1,302),
(5,302),
```

```
(7,302),
(1,306),
(2,306),
(1,378),
(2,378),
(4,378),
(6,456),
(3,456),
(5,789),
(6,789),
(3,951),
(1,951),
(1,789);
```

```
INSERT INTO flights
(flno,frm,too,distance,departs,arrives,price) VALUES
(1,'Bangalore','Mangalore',360,'10:45:00','12:00:00',10000),
(2,'Bangalore','Delhi',5000,'12:15:00','04:30:00',25000),
(3,'Bangalore','Mumbai',3500,'02:15:00','05:25:00',30000),
(4,'Delhi','Mumbai',4500,'10:15:00','12:05:00',35000),
(5,'Delhi','Frankfurt',18000,'07:15:00','05:30:00',90000),
(6,'Bangalore','Frankfurt',19500,'10:00:00','07:45:00',95000),
(7,'Bangalore','Frankfurt',17000,'12:00:00','06:30:00',99000);
```

Query 1:

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

```
SELECT DISTINCT a.anmae FROM
aircraft a,certified c,employee e
WHERE a.aid=c.aid
AND c.eid=e.eid
AND NOT EXISTS (
SELECT * FROM employee e1 WHERE e1.eid=e.eid AND e1.salary<80000);
```

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

```
SELECT DISTINCT a.anmae FROM

WHERE a.aid=c.aid

AND c.eid=e.eid

AND NOT EXISTS (

SELECT * FROM employee e1 WHERE e1.eid=e.eid AND e1.salary<80000);
```

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

```
SELECT c.eid, MAX(cruisingrange) FROM certified c, aircraft a WHERE c.aid=a.aid GROUP BY c.eid HAVING COUNT(*)>3;
```

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

```
SELECT DISTINCT e.ename FROM employee e WHERE e.salary<(SELECT MIN(f.price) FROM flights f WHERE f.frm='Bangalore' AND f.too='Frankfurt');
```

4. For all aircrafts with cruising range over 1000 kms, find the name of the aircraft and the average salary of all pilots certified for this aircraft.

```
SELECT a.aid, a.anmae, AVG(e.salary) FROM aircraft a, certified c, employee e WHERE a.aid=c.aid AND c.eid=e.eid AND a.cruisingrange>1000 GROUP BY a.aid, a.anmae;
```

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

```
SELECT distinct e.ename FROM employee e, aircraft a, certified c WHERE e.eid=c.eid AND c.aid=a.aid AND a.anmae='Boeing';
```

6. Find the aid's of all aircraft that can be used on routes from Bangalore to Delhi.

```
SELECT a.aid FROM aircraft a WHERE a.cruisingrange>(SELECT MIN(f.distance) FROM flights f WHERE f.frm='Bangalore' AND f.too='Delhi');
```

7. A customer wants to travel from Madison to New York with no more than two changes of flight. List the choice of departure times from Madison if the customer wants to arrive in New York by 6 p.m.

```
SELECT DISTINCT f1.* from flights f1, flights f2 where (f1.frm = "Bangalore" and f1.too = "Frankfurt" AND hour(f1.arrives)<6) or (f1.frm = "Bangalore" and f2.too = "Frankfurt" and f1.too=f2.frm and hour(f2.arrives)<6);
```

8. Print the name and salary of every non-pilot whose salary is more than the average salary for pilots.

```
(SELECT ename FROM employee WHERE salary>(SELECT avg(salary) FROM employee) EXCEPT (SELECT DISTINCT e2.ename from employee e2, certified c WHERE c.eid=e2.eid));
```