

U.S.N: 1BM19CS231.**LAB-1:**

```
CREATE TABLE PERSON(driver_id char(10), Name char(20), address char(40), PRIMARY KEY(driver_id));

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;
```

_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", "Chandigarh");
```

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;
```

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';
```

LAB-2

CREATE TABLE

`bankaccount`

```
(  
    `accno` int(11) NOT NULL,  
    `branchname` char(30) DEFAULT NULL,  
    `balance` double DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
--  
-- Dumping data for table `bankaccount`  
--  
INSERT INTO `bankaccount` (`accno`, `branchname`, `balance`) VALUES  
(1, 'SBI_Chamrajpet', 2000),  
(2, 'SBI_ResidencyRd', 5000),  
(4, 'SBI_ParliamentRd', 6000),  
(5, 'SBI_Jantarmentar', 8000),  
(8, 'SBI_ResidencyRd', 4000),  
(9, 'SBI_ParliamentRd', 3000),  
(10, 'SBI_ResidencyRd', 5000),  
(11, 'SBI_Jantarmentar', 2000);  
-----  
--  
-- Table structure for table `bankcostumer`  
--  
CREATE TABLE `bankcostumer` (  
    `costumername` char(30) NOT NULL,
```

```

    `costumerstreet` char(30) DEFAULT NULL,
    `costumercity` char(30) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
-- Dumping data for table `bankcostumer`
--
INSERT INTO `bankcostumer` (`costumername`, `costumerstreet`, `costumercity`) VALUES
('Avinash', 'Bull_Temple_road', 'Banglore'),
('Dinesh', 'Prithviraj_Road', 'Delhi'),
('Mohan', 'National_College_Road', 'Banglore'),
('Nikhil', 'Abc_road', 'Delhi'),
('Ravi', 'Temple_road', 'Delhi');
-----
--
-- Table structure for table `branch`
--
CREATE TABLE `branch` (
  `branchname` char(30) NOT NULL,
  `branchcity` char(30) DEFAULT NULL,
  `assets` double DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
-- Dumping data for table `branch`
--
INSERT INTO `branch` (`branchname`, `branchcity`, `assets`) VALUES
('SBI_Chamrajpet', 'Banglore', 50000),
('SBI_Jantarmantar', 'Banglore', 50000),
('SBI_ParliamentRd', 'Banglore', 10000),
('SBI_ResidencyRd', 'Banglore', 10000),
('SBI_ShivajiRd', 'Delhi', 20000);
-----
--
-- Table structure for table `depositor`
--
CREATE TABLE `depositor` (
  `costumername` char(30) NOT NULL,
  `accno` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
-- Dumping data for table `depositor`
--
INSERT INTO `depositor` (`costumername`, `accno`) VALUES
('Avinash', 1),
('Avinash', 8),
('Dinesh', 2),
('Dinesh', 10),
('Nikhil', 4),
('Nikhil', 9),
('Nikhil', 11),

```

```

('Ravi', 2),
('Ravi', 5);

-- -----
--
-- Table structure for table `loan`
--
CREATE TABLE `loan` (
  `loannumber` int(11) NOT NULL,
  `branchname` char(30) DEFAULT NULL,
  `amount` double DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
-- Dumping data for table `loan`
--
INSERT INTO `loan` (`loannumber`, `branchname`, `amount`) VALUES
(1, 'SBI_Chamrajpet', 1000),
(2, 'SBI_ResidencyRd', 2000),
(3, 'SBI_ShivajiRd', 3000),
(4, 'SBI_ParliamentRd', 4000),
(5, 'SBI_Jantarmantar', 5000);
--
-- Indexes for dumped tables
--
--
-- Indexes for table `bankaccount`
--
ALTER TABLE `bankaccount`
  ADD PRIMARY KEY (`accno`),
  ADD KEY `branchname` (`branchname`);
--
-- Indexes for table `bankcostumer`
--
ALTER TABLE `bankcostumer`
  ADD PRIMARY KEY (`costumername`);
--
-- Indexes for table `branch`
--
ALTER TABLE `branch`
  ADD PRIMARY KEY (`branchname`);
--
-- Indexes for table `depositor`
--
ALTER TABLE `depositor`
  ADD PRIMARY KEY (`costumername`,`accno`),
  ADD KEY `accno` (`accno`);
--
-- Indexes for table `loan`
--
ALTER TABLE `loan`

```

```

        ADD PRIMARY KEY (`loannumber`),
        ADD KEY `branchname` (`branchname`);
--
-- Constraints for dumped tables
--
--
-- Constraints for table `bankaccount`
--
ALTER TABLE `bankaccount`
  ADD CONSTRAINT `bankaccount_ibfk_1` FOREIGN KEY (`branchname`) REFERENCES `branch`
  (`branchname`);
--
-- Constraints for table `depositor`
--
ALTER TABLE `depositor`
  ADD CONSTRAINT `depositor_ibfk_1` FOREIGN KEY (`costumername`) REFERENCES
  `bankcostumer` (`costumername`),
  ADD CONSTRAINT `depositor_ibfk_2` FOREIGN KEY (`accno`) REFERENCES `bankaccount`
  (`accno`);
--
-- Constraints for table `loan`
--
ALTER TABLE `loan`
  ADD CONSTRAINT `loan_ibfk_1` FOREIGN KEY (`branchname`) REFERENCES `branch`
  (`branchname`);
COMMIT;

```

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.
- iv. Find all the customers who have an account at all the branches located in a specific city.
- v. Demonstrate how you delete all account tuples at every branch located in a specific city.

Solutions:

Query 3) SELECT costumername

FROM depositor d,bankaccount a

WHERE d.accno=a.accno

AND a.branchname='SBI_ResidencyRd'

GROUP BY d.costumername

HAVING COUNT(d.costumername)>=2

Query 4) SELECT d.costumername

```

FROM bankaccount a,branch b,depositor d

WHERE b.branchname=a.branchname AND

a.accno=d.accno AND

b.branchcity='Banglore'

GROUP BY d.costumername

HAVING COUNT(distinct b.branchname)=(

SELECT COUNT(branchname)

FROM branch

WHERE branchcity='Delhi')

```

Query 5) DELETE FROM bankaccount WHERE branchname IN(SELECT branchname FROM branch WHERE branchcity='Delhi');

LAB-3

```

INSERT INTO
supplier
VALUES((110,'Anan
ad 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),(11
0,89,120);
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.
- ii. Find the snames of suppliers who supply every part.

- iii. Find the snames of suppliers who supply every red part.
- iv. Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.
- 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).
- vi. For each part, find the sname of the supplier who charges the most for that part.
- vii. Find the sids of suppliers who supply only red parts.

SOLUTIONS:

- 1) SELECT DISTINCT P.pname FROM parts P, catalog C WHERE P.pid = C.pid;
- 2) 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));
- 3) 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'));
- 4) SELECT DISTINCT pname FROM catalog x, parts y, supplier z WHERE (z.sname='Ananad pvt ltd' AND z.sid=x.sid);
- 5) SELECT DISTINCT c.sid FROM catalog c WHERE c.cost > (SELECT AVG(c1.cost) FROM catalog c1 WHERE c1.pid=c.pid);
- 6) 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);
- 7) 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 whom the combined enrollment of the classes that they teach is less than 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 TABLE flight(  
    no INT,  
    frm VARCHAR(20),  
    too VARCHAR(20),  
    distance INT,  
    departs VARCHAR(20),  
    arrives VARCHAR(20),  
    price REAL,  
    PRIMARY KEY (no) );
```

```
CREATE TABLE aircraft(  
    aid INT,  
    aname VARCHAR(20),  
    cruisingrange INT,  
    PRIMARY KEY (aid) );
```

```
CREATE TABLE employees(  
    eid INT,  
    ename VARCHAR(20),  
    salary INT,  
    PRIMARY KEY (eid) );
```

```
CREATE TABLE certified(  
    eid INT,  
    aid INT,  
    PRIMARY KEY (eid,aid),  
    FOREIGN KEY (eid) REFERENCES employees (eid),  
    FOREIGN KEY (aid) REFERENCES aircraft (aid) );
```

```
INSERT INTO flight (no,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);
```

```
INSERT INTO aircraft (aid,aname,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 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);
```

```
INSERT INTO certified (edit,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);
```

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

```
SELECT DISTINCT a.aname
FROM aircraft a,certified c,employees e
WHERE a.aid=c.aid
AND c.eid=e.eid
AND NOT EXISTS
(SELECT *
FROM employees e1
```

```
WHERE e1.eid=e.eid  
AND e1.salary<80000);
```

2. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange 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 employees e  
WHERE e.salary<  
(SELECT MIN(f.price)  
FROM flight f  
WHERE f.frm='Bangalore'  
AND f.too='Frankfurt');
```

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

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

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

```
SELECT distinct e.ename  
FROM employees e, aircraft a, certified c  
WHERE e.eid=c.eid  
AND c.aid=a.aid  
AND a.aname='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 flight f  
WHERE f.frm='Bangalore'  
AND f.too='Delhi');
```