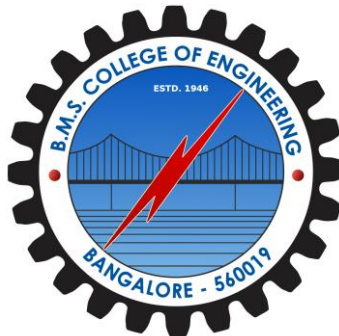


B.M.S. COLLEGE OF ENGINEERING

(AUTONOMOUS COLLEGE UNDER VTU)
BENGALURU-19



LAB TEST 1 RECORD(1-5)

NAME : TRISHUL M
USN : 1BM19CS173
**COURSE NAME : DATABASE MANAGEMENT
SYSTEMS**
COURSE TITLE : 19CS4PCDBM
SEM : 4
SECTION :D

PROGRAM 1- INSURANCE DATABASE

Consider the Insurance database given below. The primary keys are underlined and the data types are specified.

PERSON (driver-id #: String, name: String, address: String)

CAR (Regno: String, model: String, year: int)

ACCIDENT (report-number: int, date: date, location: String)

OWNS (driver-id #: String, Regno: String)

PARTICIPATED (driver-id: String, Regno: String, report-number: int, damage-amount: int)

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

```
create database Supplier;
```

```
use Supplier;
```

```
create table person (driver_id varchar(10),
```

```
name varchar(20),
```

```
address varchar(30),
```

```
primary key(driver_id));
```

```
desc person;
```

	Field	Type	Null	Key	Default	Extra
►	driver_id	varchar(10)	NO	PRI	NULL	
	name	varchar(20)	YES		NULL	
	address	varchar(30)	YES		NULL	

```
create table car(reg_num varchar(10),model
```

```
varchar(10),year int,primary key(reg_num));
```

```
desc car;
```

	Field	Type	Null	Key	Default	Extra
►	reg_num	varchar(10)	NO	PRI	NULL	
	model	varchar(10)	YES		NULL	
	year	int	YES		NULL	

```
create table accident(report_num int,accident_date
date,location varchar(20),primary key(report_num));
desc accident;
```

	Field	Type	Null	Key	Default	Extra
►	report_num	int	NO	PRI	NULL	
	accident_date	date	YES		NULL	
	location	varchar(20)	YES		NULL	

```
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));
desc owns;
```

	Field	Type	Null	Key	Default	Extra
►	driver_id	varchar(10)	NO	PRI	NULL	
	reg_num	varchar(10)	NO	PRI	NULL	

```
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));
desc participated;
```

	Field	Type	Null	Key	Default	Extra
►	driver_id	varchar(10)	NO	PRI	NULL	
	reg_num	varchar(10)	NO	PRI	NULL	
	report_num	int	NO	PRI	NULL	
	damage_amount	int	YES		NULL	

ii. Enter at least five tuples for each relation

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

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

insert into person values('A03','Smith','Ashoknagar');

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

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

select * from person;

	driver_id	name	address
►	A01	Richard	Srinivas Nagar
	A02	Pradeep	Rajajinagar
	A03	Smith	Ashoknagar
	A04	Venu	N.R.Colony
	A05	John	Hanumanth Nagar
*	NULL	NULL	NULL

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;

	reg_num	model	year
►	KA031181	Lancer	1957
	KA041702	Audi	2005
	KA052250	Indica	1990
	KA053408	Honda	2008
	KA095477	Toyota	1998
*	NULL	NULL	NULL

```

insert into accident values(11,'2003-01-01','Mysore Road');
insert into accident values(12,'2004-02-02','Southend Circle');
insert into accident values(13,'2003-01-21','Bulltemple Road');
insert into accident values(14,'2008-02-17','Mysore Road');
insert into accident values(15,'2005-03-04','Kanakpura Road');
select * from accident;

```

	report_num	accident_date	location
▶	11	2003-01-01	Mysore Road
	12	2004-02-02	Southend Circle
	13	2003-01-21	Bulltemple Road
	14	2008-02-17	Mysore Road
	15	2005-03-04	Kanakpura Road
•	NULL	NULL	NULL

```

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

```

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

```

insert into participated values('A01','KA052250',11,10000);
insert into participated values('A02','KA053408',12,50000);
insert into participated values('A03','KA095477',13,25000);
insert into participated values('A04','KA031181',14,3000);
insert into participated values('A05','KA041702',15,5000);
select * from participated;

```

	driver_id	reg_num	report_num	damage_amount
▶	A01	KA052250	11	10000
	A02	KA053408	12	50000
	A03	KA095477	13	25000
	A04	KA031181	14	3000
	A05	KA041702	15	5000
*	NULL	NULL	NULL	NULL

iii. Demonstrate how you

a. 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_num=12;
select * from participated;

	driver_id	reg_num	report_num	damage_amount
▶	A01	KA052250	11	10000
	A02	KA053408	12	25000
	A03	KA095477	13	25000
	A04	KA031181	14	3000
	A05	KA041702	15	5000

b. Add a new accident to the database.

insert into person values('A06','Jospeh','Shanti Nagar');

insert into car values('KA012370','Honda', 2008);

insert into accident values(16,'2008-01-01','MG Road');

insert into owns values('A06','KA012370');

insert into participated values('A06','KA012370',16,15000);

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

select count(*) from accident where year(accident_date)=2008;

	count(*)
▶	2

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

select count(*) as no_of_acc from participated where reg_num in(select reg_num from car where model='Lancer');

	no_of_acc
▶	2

PROGRAM 2-BANKING ENTERPRISE DATABASE

Consider the following database for a banking enterprise.

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

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

BankCustomer (customer-name: String, customer-street: String, customer-city: String)

Depositer(customer-name: String, accno: int)

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

- Create the above tables by properly specifying the primary keys and the foreign keys.
- Enter at least five tuples for each relation.
- Find all the customers who have at least two accounts at the *Main* branch (ex. SBI_ResidencyRoad).
- Find all the customers who have an account at *all* the branches located in a specific city (Ex. Delhi).
- Demonstrate how you delete all account tuples at every branch located in a specific city (Ex. Bombay).

INTRODUCTION: This database is developed for supporting banking facilities. Details of the branch along with the accounts and loans handled by them are recorded. Also details of the depositors of the corresponding branches are maintained.

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

```
create database Banking;  
use Banking;
```

```
create table Branch(branch_name varchar(30),branch_city varchar(30),assets real, primary  
key(branch_name));  
desc Branch;
```

	Field	Type	Null	Key	Default	Extra
►	branch_name	varchar(30)	NO	PRI	HULL	
	branch_city	varchar(30)	YES		HULL	
	assests	double	YES		HULL	

```
create      table      BankCustomer(customer_name      varchar(30),customer_street  
varchar(30),customer_city varchar(30), primary key(customer_name));  
desc BankCustomer;
```

	Field	Type	Null	Key	Default	Extra
►	customer_name	varchar(30)	NO	PRI	NULL	
	customer_street	varchar(30)	YES		NULL	
	customer_city	varchar(30)	YES		NULL	

```

create table BankAccount(
  accno int,
  branch_name varchar(20),
  balance real,
  primary key(accno),
  foreign key(branch_name) references Branch(branch_name)
);
desc BankAccount;

```

	Field	Type	Null	Key	Default	Extra
►	accno	int	NO	PRI	NULL	
	branch_name	varchar(20)	YES	MUL	NULL	
	balance	double	YES		NULL	

```

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)
);
desc Depositer;

```

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

```

create table Loan(
  loan_number int,
  branch_name varchar(20),
  Amount real,
  primary key(loan_number),
  foreign key(branch_name) references Branch(branch_name)
);
desc Loan;

```

	Field	Type	Null	Key	Default	Extra
►	loan_number	int	NO	PRI	NULL	
	branch_name	varchar(20)	YES	MUL	NULL	
	Amount	double	YES		NULL	

ii. Enter at least five tuples for each relation.

```
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_ParlimentRoad','Delhi',10000);
insert into Branch values('SBI_Jantarmanatar','Delhi',20000);
select *from Branch;
```

	branch_name	branch_city	assests
►	SBI_Chamrajpet	Bangalore	50000
	SBI_Jantarmanatar	Delhi	20000
	SBI_ParlimentRoad	Delhi	10000
	SBI_ResidencyRoad	Bangalore	10000
	SBI_ShivajiRoad	Bombay	20000
•	NULL	NULL	NULL

```
insert into Loan values(2,'SBI_ResidencyRoad',2000);
insert into Loan values(1,'SBI_Chamrajpet',1000);
insert into Loan values(3,'SBI_ShivajiRoad',3000);
insert into Loan values(4,'SBI_ParlimentRoad',4000);
insert into Loan values(5,'SBI_Jantarmanatar',3000);
select *from Loan;
```

	loan_number	branch_name	Amount
►	1	SBI_Chamrajpet	10000
	2	SBI_ResidencyRoad	20000
	3	SBI_ShivajiRoad	30000
	4	SBI_ParlimentRoad	40000
	5	SBI_Jantarmanatar	30000
•	NULL	NULL	NULL

```
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_ParlimentRoad',9000);
insert into BankAccount values(5,'SBI_Jantarmanatar',8000);
insert into BankAccount values(6, 'SBI_ShivajiRoad', 4000);
insert into BankAccount values(8, 'SBI_ResidencyRoad', 4000);
insert into BankAccount values(9, 'SBI_ParlimentRoad', 3000);
insert into BankAccount values(10, 'SBI_ResidencyRoad', 5000);
insert into BankAccount values(11, 'SBI_Jantarmanatar', 2000);
select *from BankAccount;
```

	accno	branch_name	balance
▶	1	SBI_Chamrajpet	2000
	2	SBI_ResidencyRoad	5000
	3	SBI_ShivajiRoad	6000
	4	SBI_ParlimentRoad	9000
	5	SBI_Jantarmantar	8000
	6	SBI_ShivajiRoad	4000
	8	SBI_ResidencyRoad	4000
	9	SBI_ParlimentRoad	3000
	10	SBI_ResidencyRoad	5000
	11	SBI_Jantarmantar	2000
★	NULL	NULL	NULL

```

insert into BankCustomer values ('Avinash', 'Bull_Temple_Road', 'Bangalore');
insert into BankCustomer values ('Dinesh', 'Bannergatta_Road', 'Bangalore');
insert into BankCustomer values ('Mohan', 'National_College_Road', 'Bangalore');
insert into BankCustomer values ('Nikhil', 'Akbar_Road', 'Delhi');
insert into BankCustomer values ('Ravi', 'Prithviraj_Road', 'Delhi');
select *from BankCustomer;

```

	customer_name	customer_street	customer_city
▶	Avinash	Bull_Temple_Road	Bangalore
	Dinesh	Bannergatta_Road	Bangalore
	Mohan	National_College_Road	Bangalore
	Nikhil	Akbar_Road	Delhi
	Ravi	Prithviraj_Road	Delhi
★	NULL	NULL	NULL

```

insert into Depositer values('Avinash', 1);
insert into Depositer values('Dinesh', 2);
insert into Depositer values('Nikhil', 4);
insert into Depositer values('Ravi', 5);
insert into Depositer values('Avinash', 8);
insert into Depositer values('Nikhil', 9);
insert into Depositer values('Dinesh', 10);
insert into Depositer values('Nikhil', 11);
select *from Depositer;

```

	customer_name	accno
▶	Avinash	1
	Dinesh	2
	Nikhil	4
	Ravi	5
	Avinash	8
	Nikhil	9
	Dinesh	10
	Nikhil	11
★	NULL	NULL

iii. Find all the customers who have at least two accounts at the *Main* branch (ex. SBI_ResidencyRoad).

```
select c.customer_name
from BankCustomer c
where exists(
select d.customer_name
from Depositer d, BankAccount ba
where
d.accno=ba.accno and
c.customer_name=d.customer_name and
ba.branch_name='SBI_ResidencyRoad'
group by d.customer_name
having count(d.customer_name)>=2
);
```

	customer_name
▶	Dinesh
•	NULL

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

```
select distinct d.customer_name from Depositer d where exists( select * from BankAccount ba
where ba.accno=d.accno and exists (select * from Branch b where b.branch_name =
ba.branch_name and b.branch_city='Delhi'));
```

	customer_name
▶	Ravi
	Nikhil

v. 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');
select *from BankAccount;
```

	accno	branch_name	balance
▶	1	SBI_Chamrajpet	2000
	2	SBI_ResidencyRoad	5000
	4	SBI_ParliamentRoad	9000
	5	SBI_Jantarantar	8000
	8	SBI_ResidencyRoad	4000
	9	SBI_ParliamentRoad	3000
	10	SBI_ResidencyRoad	5000
	11	SBI_Jantarantar	2000
•	NULL	NULL	NULL

PROGRAM 3- SUPPLIER DATABASE

Consider the following schema:

SUPPLIERS(sid: integer, sname: string, address: string)

PARTS(pid: integer, pname: string, color: string)

CATALOG(sid: integer, pid: integer, cost: real)

The Catalog relation lists the prices charged for parts by Suppliers.

```
create database Supplier;
```

```
use Supplier;
```

```
create table catalog(sid int,pid int,cost int);
```

```
desc catalog;
```

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

```
create table supplier(sid int,sname varchar(20),city varchar(15),primary key (sid));
```

```
desc supplier;
```

	Field	Type	Null	Key	Default	Extra
►	sid	int	NO	PRI	NULL	
	sname	varchar(20)	YES		NULL	
	city	varchar(15)	YES		NULL	

```
create table parts(pid int,pname varchar(15),color varchar(10),primary key (pid));
```

```
desc parts;
```

	Field	Type	Null	Key	Default	Extra
►	pid	int	NO	PRI	NULL	
	pname	varchar(15)	YES		NULL	
	color	varchar(10)	YES		NULL	

insert into supplier values(10001, 'Acme Widget','Bengaluru');

insert into supplier values(10002,'Johns','Kolkata');

insert into supplier values(10003, 'Vimal','Mumbai');

insert into supplier values(10004, 'Reliance','Delhi');

insert into supplier values(10005, 'Mahindra','Mumbai');

select * from supplier;

	sid	sname	city
►	10001	Acme Widget	Bengaluru
	10002	Johns	Kolkata
	10003	Vimal	Mumbai
	10004	Reliance	Delhi
•	NULL	NULL	NULL

insert into parts values(20001, 'Book','Red');

insert into parts values(20002, 'Pen','Red');

insert into parts values(20003, 'Pencil','Green');

insert into parts values(20004, 'Mobile','Green');

insert into parts values(20005, 'Charger','Black');

select * from parts;

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

insert into catalog values(10001, '20001','10');

```

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

```

	sid	pid	cost
▶	10001	20001	10
	10001	20002	10
	10001	20003	30
	10001	20004	10
	10001	20005	10
	10002	20001	10
	10002	20002	20
	10003	20003	30
	10004	20003	40
*	NULL	NULL	NULL

1.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;
```

	pname
▶	Book
	Pen
	Pencil
	Mobile
	Charger

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

```
select s.sname from supplier s
```

```
where not exists (select p.pid from parts p where not exists
```

(select c.sid from catalog c where c.sid = s.sid and c.pid = p.pid));

	sname
▶	Acme Widget

3. 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' and

(not exists (select c.sid from catalog c where c.sid = s.sid and c.pid = p.pid))));

	sname
▶	Acme Widget
	Johns

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

select p.pname from parts p , catalog c, supplier s

where p.pid = c.pid and c.sid = s.sid and s.sname = 'Acme Widget'

and not exists

(select * from catalog c1, supplier s1 where

p.pid = c1.pid and c1.sid = s1.sid and s1.sname <> 'Acme Widget');

	pname
▶	Mobile
	Charger

5. 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);

	sid
▶	10002
	10004

6.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);

	pid	sname
▶	20001	Acme Widget
	20004	Acme Widget
	20005	Acme Widget
	20001	Johns
	20002	Johns
	20003	Reliance

PROGRAM 4- STUDENT FACULTY DATABASE

Consider the following database for student enrollment for course :

STUDENT(snum: integer, sname:string, major: string, lvl: string, age: integer)

CLASS(cname: string, meetsat: time, room: string, fid: integer)

ENROLLED(snum: integer, cname:string)

FACULTY(fid: integer, fname:string, deptid: integer)

create database Students;

use Students;

create table student(snum int, sname varchar(10), major varchar(20), lvl varchar(20), age int,primary key (snum));

desc student;

	Field	Type	Null	Key	Default	Extra
►	snum	int	NO	PRI	NULL	
	sname	varchar(10)	YES		NULL	
	major	varchar(20)	YES		NULL	
	lvl	varchar(20)	YES		NULL	
	age	int	YES		NULL	

create table faculty(fid int, fname varchar(20), deptid int,primary key(fid));

desc faculty;

	Field	Type	Null	Key	Default	Extra
►	fid	int	NO	PRI	NULL	
	fname	varchar(20)	YES		NULL	
	deptid	int	YES		NULL	

create table class(cname varchar(20), meetsat timestamp, room varchar(10), fid int,primary key (cname),foreign key(fid) references faculty(fid));

desc class;

	Field	Type	Null	Key	Default	Extra
►	cname	varchar(20)	NO	PRI	NULL	
	meetsat	timestamp	YES		NULL	
	room	varchar(10)	YES		NULL	
	fid	int	YES	MUL	NULL	

```

create table enrolled(snum int, cname varchar(20),primary key(snum,cname),
foreign key(snum) references student(snum),
foreign key(cname) references class(cname));
desc enrolled;

```

	Field	Type	Null	Key	Default	Extra
►	snum	int	NO	PRI	NULL	
	cname	varchar(20)	NO	PRI	NULL	

```

insert into student values(1, 'jhon', 'CS', 'Sr', 19);
insert into student values(2, 'Smith', 'CS', 'Jr', 20);
insert into student values(3 , 'Jacob', 'CV', 'Sr', 20);
insert into student values(4, 'Tom ', 'CS', 'Jr', 20);
insert into student values(5, 'Rahul', 'CS', 'Jr', 20);
insert into student values(6, 'Rita', 'CS', 'Sr', 21);
select * from student;

```

	snum	sname	major	lvl	age
►	1	jhon	CS	Sr	19
	2	Smith	CS	Jr	20
	3	Jacob	CV	Sr	20
	4	Tom	CS	Jr	20
	5	Rahul	CS	Jr	20
	6	Rita	CS	Sr	21
*	NULL	NULL	NULL	NULL	NULL

```

insert into faculty values(11, 'Harish', 1000);
insert into faculty values(12, 'MV', 1000);
insert into faculty values(13 , 'Mira', 1001);
insert into faculty values(14, 'Shiva', 1002);
insert into faculty values(15, 'Nupur', 1000);
select * from faculty;

```

	fid	fname	deptid
▶	11	Harish	1000
	12	MV	1000
	13	Mira	1001
	14	Shiva	1002
	15	Nupur	1000
★	NULL	NULL	NULL

insert into class values('class1', '12/11/15 10:15:16', 'R1', 14);

insert into class values('class10', '12/11/15 10:15:16', 'R128', 14);

insert into class values('class2', '12/11/15 10:15:20', 'R2', 12);

insert into class values('class3', '12/11/15 10:15:25', 'R3', 12);

insert into class values('class4', '12/11/15 20:15:20', 'R4', 14);

insert into class values('class5', '12/11/15 20:15:20', 'R3', 15);

insert into class values('class6', '12/11/15 13:20:20', 'R2', 14);

insert into class values('class7', '12/11/15 10:10:10', 'R3', 14);

select * from class;

	cname	meetsat	room	fid
▶	class1	2012-11-15 10:15:16	R1	14
	class10	2012-11-15 10:15:16	R128	14
	class2	2012-11-15 10:15:20	R2	12
	class3	2012-11-15 10:15:25	R3	11
	class4	2012-11-15 20:15:20	R4	14
	class5	2012-11-15 20:15:20	R3	15
	class6	2012-11-15 13:20:20	R2	14
	class7	2012-11-15 10:10:10	R3	14
★	NULL	NULL	NULL	NULL

insert into enrolled values(1, 'class1');

insert into enrolled values(2, 'class1');

insert into enrolled values(3, 'class3');

insert into enrolled values(4, 'class3');

```

insert into enrolled values(5, 'class4');
insert into enrolled values(1, 'class5');
insert into enrolled values(2, 'class5');
insert into enrolled values(3, 'class5');
insert into enrolled values(4, 'class5');
insert into enrolled values(5, 'class5');
select * from enrolled;

```

	snum	cname
▶	1	class1
	2	class1
	3	class3
	4	class3
	5	class4
	1	class5
	2	class5
	3	class5
	4	class5
	5	class5
*	NULL	NULL

i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by Harish

```
SELECT DISTINCT S.sname
```

```
FROM student S, class C, enrolled E, faculty F
```

```
WHERE S.snum = E.snum AND E.cname = C.cname AND C.fid = F.fid AND
```

```
F.fname = 'Harish' AND S.lvl = 'Jr';
```

	sname
▶	Tom

ii. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.

```
SELECT C.cname
```

```
FROM class C WHERE C.room = 'R128'
```

```
OR C.cname IN (SELECT E.cname FROM enrolled E
```

GROUP BY E.cname HAVING COUNT(*) >= 5);

	cname
▶	class10
	class5
•	NULL

iii. Find the names of all students who are enrolled in two classes that meet at the 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.cname <> E2.cname

AND E1.cname = C1.cname

AND E2.cname = C2.cname AND C1.meetsat = C2.meetsat);

	sname
▶	Rahul

iv. 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)
);

	fname	fid
▶	Shiva	14
•	NULL	NULL

v. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

```

SELECT DISTINCT F.fname
FROM faculty F
WHERE 5 > (SELECT COUNT(E.snum)
FROM class C, enrolled E
WHERE C.cname = E.cname
AND C.fid = F.fid);

```

	fname
▶	Harish
	MV
	Mira
	Shiva

vi. Find the names of students who are not enrolled in any class.

```

SELECT DISTINCT S.sname
FROM student S
WHERE S.snum NOT IN (SELECT E.snum
FROM enrolled E );

```

	sname
▶	Rita

vii. For each age value that appears in Students, find the level value that appears most often.

```

SELECT S.age, S.lvl
FROM Student S
GROUP BY S.age, S.lvl
HAVING S.lvl IN (SELECT S1.lvl FROM Student S1
WHERE S1.age = S.age
GROUP BY S1.lvl, S1.age
HAVING COUNT(*) >= ALL (SELECT COUNT(*)
FROM Student S2

```

WHERE s1.age = S2.age

GROUP BY S2.lvl, S2.age));

	age	lvl
▶	19	Sr
	20	Jr
	21	Sr

PROGRAM 5-AIRLINE FLIGHT DATABASE

Consider the following database that keeps track of airline flight information:

FLIGHTS(flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price: integer)

AIRCRAFT(aid: integer, aname: string, cruisingrange: integer)

CERTIFIED(eid: integer, aid: integer)

EMPLOYEES(eid: integer, ename: string, salary: integer)

Note that the Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly.

```
create database flight;
```

```
use flight;
```

```
create table flights(
```

```
    flno int,
```

```
    fromplace varchar(15),
```

```
    toplace varchar(15),
```

```
    distance int,
```

```
    departs datetime,
```

```

arrives datetime,

price int,

primary key (flno));

```

desc flights;

	Field	Type	Null	Key	Default	Extra
►	flno	int	NO	PRI	NULL	
	fromplace	varchar(15)	YES		NULL	
	toplace	varchar(15)	YES		NULL	
	distance	int	YES		NULL	
	departs	datetime	YES		NULL	
	arrives	datetime	YES		NULL	
	price	int	YES		NULL	

```

create table aircraft(

```

```

    aid int,

    aname varchar(15),

    cruisingrange int,

    primary key (aid));

```

desc aircraft;

	Field	Type	Null	Key	Default	Extra
►	aid	int	NO	PRI	NULL	
	aname	varchar(15)	YES		NULL	
	cruisingrange	int	YES		NULL	

```

create table employees (

```

```

    eid int,

    ename varchar(15),

    salary int,

    primary key (eid));

```

desc employees;

	Field	Type	Null	Key	Default	Extra
►	eid	int	NO	PRI	HULL	
	ename	varchar(15)	YES		HULL	
	salary	int	YES		HULL	

```
create table certified (
    eid int,
    aid int,
    foreign key (eid) references employees(eid),
    foreign key (aid) references aircraft(aid));
desc certified;
```

	Field	Type	Null	Key	Default	Extra
►	eid	int	YES	MUL	HULL	
	aid	int	YES	MUL	HULL	

```
insert into flights values(101, 'Bangalore', 'Delhi', 2500, '2005-05-13 07:15:31', '2005-05-13 18:15:31', 5000);
```

```
insert into flights values(102, 'Bangalore', 'Lucknow', 3000, '2013-05-05 07:15:31', '2013-05-05 11:15:31', 6000);
```

```
insert into flights values(103, 'Lucknow', 'Delhi', 500, '2013-05-05 12:15:31', '2013-05-05 17:15:31', 3000);
```

```
insert into flights values(107, 'Bangalore', 'Frankfurt', 8000, '2013-05-05 07:15:31', '2013-05-05 22:15:31', 60000);
```

```
insert into flights values(104, 'Bangalore', 'Frankfurt', 8500, '2013-05-05 07:15:31', '2013-05-05 23:15:31', 75000);
```

```
insert into flights values(105, 'Kolkata', 'Delhi', 3400, '2013-05-05 07:15:31', '2013-05-05 09:15:31', 7000);
```

```
insert into flights values(106, 'Bangalore', 'Kolkata', 1000, '2013-05-05 01:15:30', '2013-05-05 09:20:30', 10000);
```

```
insert into flights values(108, 'Lucknow', 'Kolkata', 1000, '2013-05-05 11:30:30', '2013-05-05 15:20:30', 10000);
```

select * from flights;

	fno	fromplace	toplace	distance	departs	arrives	price
▶	101	Bangalore	Delhi	2500	2005-05-13 07:15:31	2005-05-13 18:15:31	5000
	102	Bangalore	Lucknow	3000	2013-05-05 07:15:31	2013-05-05 11:15:31	6000
	103	Lucknow	Delhi	500	2013-05-05 12:15:31	2013-05-05 17:15:31	3000
	104	Bangalore	Frankfurt	8500	2013-05-05 07:15:31	2013-05-05 23:15:31	75000
	105	Kolkata	Delhi	3400	2013-05-05 07:15:31	2013-05-05 09:15:31	7000
	106	Bangalore	Kolkata	1000	2013-05-05 01:15:30	2013-05-05 09:20:30	10000
	107	Bangalore	Frankfurt	8000	2013-05-05 07:15:31	2013-05-05 22:15:31	60000
	108	Lucknow	Kolkata	1000	2013-05-05 11:30:30	2013-05-05 15:20:30	10000
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

insert into aircraft values(101, '747', 3000);

insert into aircraft values(102, 'Boeing', 900);

insert into aircraft values(103, '647', 800);

insert into aircraft values(104, 'Dreamliner', 10000);

insert into aircraft values(105, 'Boeing', 3500);

insert into aircraft values(106, '707', 1500);

insert into aircraft values(107, 'Dream', 120000);

insert into aircraft values(108, '707', 760);

insert into aircraft values(109, '747', 1000);

select * from aircraft;

	aid	aname	cruisingrange
▶	101	747	3000
	102	Boeing	900
	103	647	800
	104	Dreamliner	10000
	105	Boeing	3500
	106	707	1500
	107	Dream	120000
	108	707	760
	109	747	1000
•	NULL	NULL	NULL

insert into employees values(701, 'A', 50000);

```

insert into employees values(702, 'B', 100000);
insert into employees values(703, 'C', 150000);
insert into employees values(704, 'D', 90000);
insert into employees values(705, 'E', 40000);
insert into employees values(706, 'F', 60000);
insert into employees values(707, 'G', 90000);
select * from employees;

```

	eid	ename	salary
▶	701	A	50000
	702	B	100000
	703	C	150000
	704	D	90000
	705	E	40000
	706	F	60000
	707	G	90000
*	NULL	NULL	NULL

```

insert into certified values(701, 101);
insert into certified values(701, 102);
insert into certified values(701, 106);
insert into certified values(701, 105);
insert into certified values(702, 104);
insert into certified values(703, 104);
insert into certified values(704, 104);
insert into certified values(702, 107);
insert into certified values(703, 107);
insert into certified values(704, 107);
insert into certified values(702, 101);
insert into certified values(702, 108);
insert into certified values(701, 109);
select * from certified;

```

	eid	aid
▶	701	101
	701	102
	701	106
	701	105
	702	104
	703	104
	704	104
	702	107
	703	107
	704	107
	702	101
	702	108
	701	109

i. 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 where a.aid in (

select c.aid from certified c, employees e where

c.eid = e.eid and not exists(

select * from employees e1 where e1.eid=e.eid and e1.salary<80000

));

	aname
▶	747
	Dreamliner
	Dream
	707

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

select max(a.cruisingrange), c.eid from certified c, aircraft a

where c.aid = a.aid group by c.eid having count(c.aid)>3;

	max(a.cruisingrange)	eid
▶	3500	701
	120000	702

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

```
select ename from employees where salary < (
select min(price) from flights where fromplace='Bengaluru' and toplace='Frankfurt');
```

	ename
▶	A
	E

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

```
select avg(e.salary), c.aid from certified c, employees e where c.aid in(
select aid from aircraft where cruisingrange>1000) and e.eid = c.eid group by c.aid;
```

	avg(e.salary)	aid
▶	75000.0000	101
	113333.3333	104
	50000.0000	105
	50000.0000	106
	113333.3333	107

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

```
select ename from employees where eid in(
select eid from certified where aid in(
select aid from aircraft where aname = 'Boeing'));
```

	ename
▶	A

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

```
select aname from aircraft where cruisingrange > any
(select distance from flights where fromplace='Bengaluru' and toplace='Delhi');
```

	aname
▶	747
	Dreamliner
	Boeing
	Dream

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

```
select F.flno, F.departs
```

```
from flights F
```

```
Where F.flno in ( ( select F0.flno
```

```
from flights F0
```

```
where F0.fromplace = 'Bangalore' and F0.toplace = 'Kolkata'
```

```
and extract(hour from F0.arrives) < 18 )
```

```
union
```

```
( select F0.flno
```

```
from flights F0, flights F1
```

```
where F0.fromplace = 'Bangalore' and F0.toplace <> 'Kolkata'
```

```
and F0.toplace = F1.fromplace and F1.toplace = 'Kolkata'
```

```
and F1.departs > F0.arrives
```

```
and extract(hour from F1.arrives) < 18)
```

```
union
```

```
( select F0.flno
```

```
from flights F0, flights F1, flights F2
```

```
where F0.fromplace = 'Bangalore'
```

```
and F0.toplace = F1.fromplace
```

```
and F1.toplace = F2.fromplace
```

```
and F2.toplace = 'Kolkata'
```

```
and F0.toplace <> 'Kolkata'  
and F1.toplace <> 'Kolkata'  
and F1.departs > F0.arrives  
and F2.departs > F1.arrives  
and extract(hour from F2.arrives) < 18));
```

	fno	departs
►	102	2013-05-05 07:15:31
	106	2013-05-05 01:15:30