Name: SaiPraveen Marni

USN: 1BM19CS138

Dept: CSE

Section: C

Course Name: Database Management Systems

Lab_batch: C-3

PROGRAM 1: INSURANCE DATABASE

Consider the Insurance database given below. The data types are specified.

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)

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)Demonstrate how you

a.Update the damage amount to 2500 for the car with a specific reg-num(K

A37) for which the accident report number was 12.

b.Add a new accident to the database.

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

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

```
create database ins;
use ins;
create table person(
driver_id varchar(30) primary key,
nam varchar(30),
address varchar(30)
);
create table accident(
report_number int primary key,
dat date,
location varchar(30)
);
create table car(
regno varchar(30),
model varchar(30),
yea int
);
alter table car
add primary key(regno);

create table owns(
driver_id varchar(30),
regno varchar(30),
foreign key(driver_id) references person(driver_id) on delete set null on update cascade,
foreign key(regno) references car(regno) on delete set null on update cascade
);
```

```
create table participated(
driver_id varchar(30),
regno varchar(30),
report_number int,
dam int,
foreign key(driver_id) references person(driver_id) on update cascade on delete set null,
foreign key(regno) references car(regno) on update cascade on delete set null,
foreign key(report_number) references accident(report_number) on update cascade on delete set null
);
```

```
insert into person(driver_id,nam,address)
values('A01','Praveen','Karnataka'),('A02','Saquib','Jharkhand'),('A03','Vihari','Andhra Pradesh'),('A04','Karthik','Tamil Nadu'),('A05','Bumrah','Gujarat'),('A06','Hari' ,'Karnataka');
```

	driver_id	nam	address
Edit Copy Delete	A01	Praveen	Karnataka
Edit Copy Delete	A02	Saquib	Jharkhand
Edit Copy Delete	A03	Vihari	Andhra Pradesh
Edit Copy Delete	A04	Karthik	Tamil Nadu
Edit Copy Delete	A05	Bumrah	Gujarat
Edit Copy Delete	A06	Hari	Karnataka

```
insert into car(regno,model,yea)
values('KA37','MAHINDRA',2008),('KA35','SKODA',2008),('KA36','MARUTI',2009),('KA39','RENAULT',2010),('KA38','Honda',2012),('KA30','HYUNDAI',2014);
```

	regno	model	yea
Edit Copy Delete	KA30	HYUNDAI	2014
Edit Copy Delete	KA35	SKODA	2008
Edit Copy Delete	KA36	MARUTI	2009
Edit Copy Delete	KA37	MAHINDRA	2008
Edit Copy Delete	KA38	Honda	2012
Edit Copy Delete	KA39	RENAULT	2010

insert into accident(report_number,dat,location) values(12,'2008-10-11','Koppal'),(13,'2008-11-19','Hubli'),(14,'2008-08-11','Kolkata'),(15,'2008-08-08','Delhi'),(16,'2008-07-07','Bengaluru'),(17,'2008-06-05','Mumbai');

	report_number	dat	location
Edit Copy Delete	12	2008-10-11	Koppal
Edit Copy Delete	13	2008-11-19	Hubli
Edit Copy Delete	14	2008-08-11	Kolkata
Edit Copy Delete	15	2008-08-08	Delhi
Edit Copy Delete	16	2008-07-07	Bengaluru
Edit Copy Delete	17	2008-06-05	Mumbai

insert into owns(driver_id,regno)
values('A01','KA37'),('A02','KA35'),('A03','KA36'),('A0','KA39'),('A05','KA38'),('A06','KA30');

driver_id	regno
A01	KA37
A02	KA35
A03	KA36
A04	KA39
A05	KA38
A06	KA30

insert into participated(driver_id,regno,report_number,dam)
values('A01','KA37',12,2000),('A02','KA35',13,1500),('A03','KA36',14,2000),('A0','KA39',15,2500),('A05','KA38',16,3000),('A06','KA30',17,4000);

driver_id	regno	report_number	dam
A01	KA37	12	2000
A02	KA35	13	1500
A03	KA36	14	2000
A04	KA39	15	2500
A05	KA38	16	3000
A06	KA30	17	4000

```
select * from person;

select * from car;

select * from accident;

select * from owns;

select * from participated;
```

```
update participated

set dam=2500

where report_number=12 and regno='KA37';
```

driver_id	regno	report_number	dam
A01	KA37	12	2500
A02	KA35	13	1500
A03	KA36	14	2000
A04	KA39	15	2500
A05	KA38	16	3000
A06	KA30	17	4000
A09	KA31	12	2000

```
insert into accident(report_number,dat,location) values(18,'2008-11-19','Koppal');

insert into accident(report_number,dat,location) values(19,'2000-11-19','Koppal');
```

	report_number	dat	location
Edit Copy Delete	12	2008-10-11	Koppal
Edit Copy Delete	13	2008-11-19	Hubli
Edit Copy Delete	14	2008-08-11	Kolkata
Edit Copy Delete	15	2008-08-08	Delhi
Edit Copy Delete	16	2008-07-07	Bengaluru
Edit Copy Delete	17	2008-06-05	Mumbai
Edit Copy Delete	18	2008-11-19	Koppal
Edit Copy Delete	19	2000-11-19	Koppal

insert into person(driver_id,nam,address) values('A09','Hari K','Karnataka');

	driver_id	nam	address
Edit Copy Delete	A01	Praveen	Karnataka
Edit Copy Delete	A02	Saquib	Jharkhand
Edit Copy Delete	A03	Vihari	Andhra Pradesh
Edit Copy Delete	A04	Karthik	Tamil Nadu
Edit Copy Delete	A05	Bumrah	Gujarat
Edit Copy Delete	A06	Hari	Karnataka
Edit Copy Delete	A09	Hari K	Karnataka

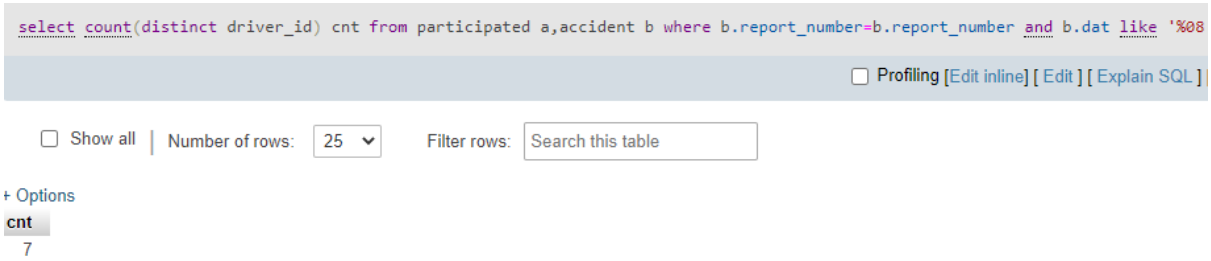
insert into car(regno,model,yea) values('KA31','MARUTI',2008);

	regno	model	yea
Edit Copy Delete	KA30	HYUNDAI	2014
Edit Copy Delete	KA31	MARUTI	2008
Edit Copy Delete	KA35	SKODA	2008
Edit Copy Delete	KA36	MARUTI	2009
Edit Copy Delete	KA37	MAHINDRA	2008
Edit Copy Delete	KA38	Honda	2012
Edit Copy Delete	KA39	RENAULT	2010

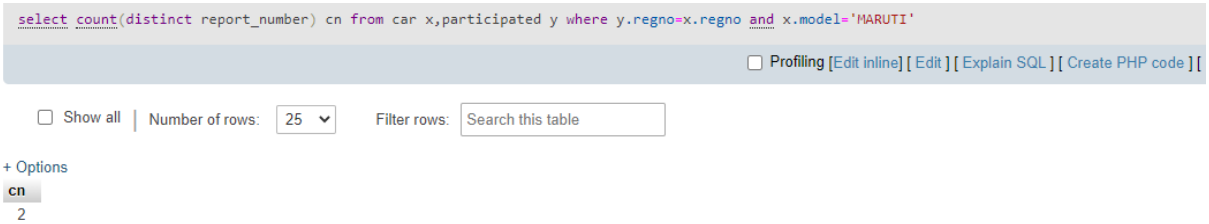
insert into participated(driver_id,regno,report_number,dam) values('A09','KA31',12,2000);

driver_id	regno	report_number	dam
A01	KA37	12	2500
A02	KA35	13	1500
A03	KA36	14	2000
A04	KA39	15	2500
A05	KA38	16	3000
A06	KA30	17	4000
A09	KA31	12	2000

select count(distinct driver_id) cnt from participated a,accident b where b.report_number=b.report_number and b.dat like '%08';



select count(distinct report_number) cn from car x,participated y where y.regno=x.regno and x.model='MARUTI';



select * from participated order by dam desc;

select avg(dam) av from participated;

SET SQL_SAFE_UPDATES=0;

delete from participated

where dam<=2500;

```
select nam from person p ,participated q where p.driver_id=q.driver_id and dam>2500;

select nam,dam from person p ,participated q where p.driver_id=q.driver_id and dam>2500;
```

PROGRAM 2: BANKING ENTERPRISE

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)

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 (ex.

SBI_ResidencyRoad).

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

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

```
create database bank;
use bank;

create table branch(
branch_name varchar(30) primary key,
branch_city varchar(30),
```

```
assests decimal(6,2)
);

drop table accounts;

create table accounts(
accno int primary key,
branch_name varchar(30),
balance decimal(7,2),
foreign key(branch_name) references branch(branch_name)
);

create table depositor(
customer_name varchar(30),
customer_street varchar(30),
customer_city varchar(30)
);

alter table depositor
add column accno int;

alter table depositor
add constraint fori
foreign key(accno) references accounts(accno);

alter table depositor
drop foreign key fori;

create table loan(
```

```
loan_number int primary key,
branch_name varchar(30),
amt decimal(7,2)
);

alter table loan
add constraint fk_1
foreign key(branch_name) references branch(branch_name);

create table borrower(
customer_name varchar(30),
loan_number int,
foreign key(loan_number) references loan(loan_number)
);

desc branch;
desc accounts;
desc depositor;
desc loan;
desc borrower;

truncate table depositor;

insert into branch(branch_name,branch_city,assests) values('SBH
RAICHUR','RAICHUR',1234.56),('SBI HOSPET','HOSPET',4567.89),('SBI
BANGLORE','BANGLORE RURAL',9687.25),('SBI GAGAG','GADAG',1000.25),('SBI
RAICHUR','RAICHUR',2000.24);
```

	branch_name	branch_city	assests
Edit Copy Delete	SBH RAICHUR	RAICHUR	1234.56
Edit Copy Delete	SBI BANGLORE	BANGLORE RURAL	9687.25
Edit Copy Delete	SBI GADAG	GADAG	1000.25
Edit Copy Delete	SBI HOSPET	HOSPET	4567.89
Edit Copy Delete	SBI RAICHUR	RAICHUR	2000.24

insert into accounts(accno,branch_name,balance) values(1234,'SBH RAICHUR ',40000.00),(4567,'SBI BANGLORE',50000.00),(2345,'SBI BANGLORE',60000.00),(1000,'SBI GADAG',70000.00),(1024,'SBI RAICHUR',10000.00);

	accno	branch_name	balance
Edit Copy Delete	1000	SBI GADAG	70000.00
Edit Copy Delete	1024	SBI RAICHUR	10000.00
Edit Copy Delete	1234	SBH RAICHUR	40000.00
Edit Copy Delete	2345	SBI BANGLORE	60000.00
Edit Copy Delete	4567	SBI BANGLORE	50000.00

insert into depositor(customer_name,customer_street,customer_city,accno) values('Hari','ALMERI','KOPPAL',1234),('Saquib','BTM LAYOUT,'BANGLORE',4567),('Praveen','SINDHNUR',' RAICHUR ',2345),('Subhas','CHAMRAJPET',’BIJAPUR',1000),('BalaJi','MUDIRABAD','KOLAR',1024);

customer_name	customer_street	customer_city	accno
Hari	ALMERI	KOPPAL	1234
Saquib	BTM LAYOUT	BANGLORE	4567
Praveen	SINDHNUR	RAICHUR	2345
Subhas	CHAMRAJPET	BIJAPUR	1000
BalaJi	MUDIRABAD	KOLAR	1024
Pramod	Vasavi Nagar	Sindhanur	1233
Pramod	Vasavi Nagar	RAICHUR	1232
Hari	GADAG ROAD	RAICHUR	1220
Hari	GADAG ROAD	RAICHUR	1221
Arya	MANNUR ROAD	RAICHUR	1220
Arya	MANNUR ROAD	RAICHUR	1220

insert into loan(loan_number,branch_name,amt) values(1,'SBI GADAG',10000.00),(2,'SBI RAICHUR ',15000.00),(3,'SBH RAICHUR,20000.00),(4,'SBI BANGLORE',25000.00),(5,'SBI BANGLORE',24000.00);

	loan_number	branch_name	amt
Edit Copy Delete	1	SBI GADAG	10000.00
Edit Copy Delete	2	SBI RAICHUR	15000.00
Edit Copy Delete	3	SBH RAICHUR	20000.00
Edit Copy Delete	4	SBI BANGLORE	25000.00
Edit Copy Delete	5	SBI BANGLORE	24000.00

insert into borrower(customer_name,loan_number)
values('Mayur',1),('Pooja',2),('Gagan',3),('Sufail,4),('Akash,5);

customer_name	loan_number
Mayur	1
Pooja	2
Gagan	3
Sufail	4
Akash	5

insert into branch(branch_name,branch_city,assests) values('SBI KOPPAL','KOPPAL',1234.56);

insert into accounts(accno,branch_name,balance) values(1233,'SBH KOPPAL',40000.00);

insert into accounts(accno,branch_name,balance) values(1232,'SBI KOPPAL',40000.00);

insert into depositor(customer_name,customer_street,customer_city,accno)
values('Pramod','Vasavi Nagar','Sindhanur',1233);

insert into depositor(customer_name,customer_street,customer_city,accno)
values('Pramod','Vasavi Nagar',' RAICHUR ',1232);

insert into accounts(accno,branch_name,balance) values(1220,'SBH KOPPAL',40000.00);

insert into depositor(customer_name,customer_street,customer_city,accno)
values('Hari','GADAG ROAD',' RAICHUR ',1220);

insert into accounts(accno,branch_name,balance) values(1221,'SBH KOPPAL',40000.00);

insert into depositor(customer_name,customer_street,customer_city,accno) values('Hari','GADAG ROAD',' RAICHUR ',1221);

insert into accounts(accno,branch_name,balance) values(1120,'SBH KOPPAL',40000.00);

insert into depositor(customer_name,customer_street,customer_city,accno) values('Arya','MANNUR ROAD',' RAICHUR ',1220);

insert into accounts(accno,branch_name,balance) values(1128,'SBH KOPPAL',40000.00);

insert into depositor(customer_name,customer_street,customer_city,accno) values('Arya','MANNUR ROAD',' RAICHUR ',1220);

customer_name	customer_street	customer_city	accno
Hari	ALMERI	KOPPAL	1234
Saquib	BTM LAYOUT	BANGLORE	4567
Praveen	SINDHNUR	RAICHUR	2345
Subhas	CHAMRAJPET	BIJAPUR	1000
BalaJi	MUDIRABAD	KOLAR	1024
Pramod	Vasavi Nagar	Sindhanur	1233
Pramod	Vasavi Nagar	RAICHUR	1232
Hari	GADAG ROAD	RAICHUR	1220
Hari	GADAG ROAD	RAICHUR	1221
Arya	MANNUR ROAD	RAICHUR	1220
Arya	MANNUR ROAD	RAICHUR	1220

select * from branch;

select * from accounts;

select * from depositor;

select * from loan;

select * from borrower;

select c.customer_name,a.branch_name from depositor c join accounts a on c.accno=a.accno group by c.customer_name having count(*)>1;

create table joined as(

select br.branch_city,acn.branch_name from branch br left join accounts acn on br.branch_name=acn.branch_name

);

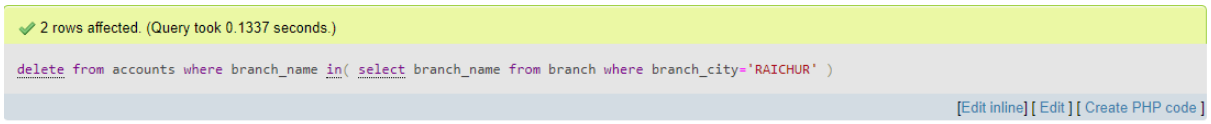
```
create table final as(

select bran.branch_name,anc.accno from branch  bran join accounts anc on
bran.branch_name=anc.branch_name

);


select * from final;

select * from  joined;


select dpt.customer_name from depositor dpt left join joined jn on
dpt.branch_name=jn.branch_name where jn.branch_city='RAICHUR';
```

branch_city	branch_name
RAICHUR	SBH RAICHUR
RAICHUR	SBH RAICHUR
RAICHUR	SBH RAICHUR
RAICHUR	SBH RAICHUR
RAICHUR	SBH RAICHUR
BANGLORE RURAL	*NULL*
GADAG	*NULL*
HOSPET	*NULL*
RAICHUR	SBI RAICHUR

```
create table f_join as(

select fl.branch_name,fl.accno,bc.branch_city from final fl join branch bc on
fl.branch_name=bc.branch_name

);


select * from f_join;


select distinct customer_name,branch_city from depositor f,f_join fn where
f.accno=fn.accno and fn.branch_city='RAICHUR';
```

branch_name	accno	branch_city
SBH RAICHUR	1120	RAICHUR
SBH RAICHUR	1128	RAICHUR
SBH RAICHUR	1220	RAICHUR
SBH RAICHUR	1221	RAICHUR
SBH RAICHUR	1233	RAICHUR
SBI RAICHUR	1232	RAICHUR

```
SET SQL_SAFE_UPDATES=0;

delete from accounts

where branch_name in(

select branch_name from branch where branch_city='RAICHUR'

);
```



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.

Write the following queries in SQL:

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.

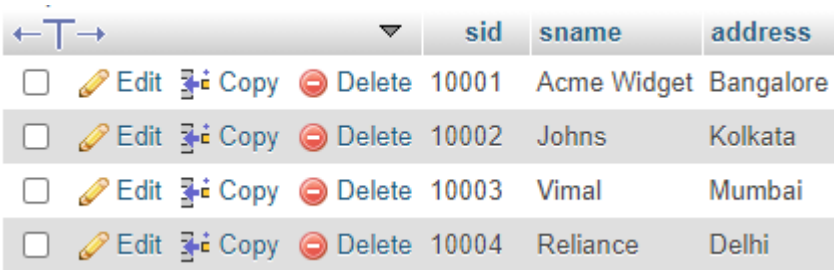
iv. Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.

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.

```
create database supplier;
use supplier;
create table suppliers(
sid int primary key,
sname varchar(30),
address varchar(30)
);

create table parts(
pid int primary key,
pname varchar(30),
```

```
color varchar(30)
);

create table catalog(
sid int,
pid int,
cost int,
constraint fk_1
 foreign key(sid) references suppliers(sid) on update cascade on delete set null,
constraint fk_2
 foreign key(pid) references parts(pid) on update cascade on delete set null
);

select * from suppliers;
select * from parts;
select * from catalog;

insert into suppliers values (10001,'Acme
Widget','Bangalore'),(10002,'Johns','Kolkata'),(10003,'Vimal','Mumbai'),(10004,'Reliance','D
elhi');
```



	sid	sname	address
Edit Copy Delete	10001	Acme Widget	Bangalore
Edit Copy Delete	10002	Johns	Kolkata
Edit Copy Delete	10003	Vimal	Mumbai
Edit Copy Delete	10004	Reliance	Delhi

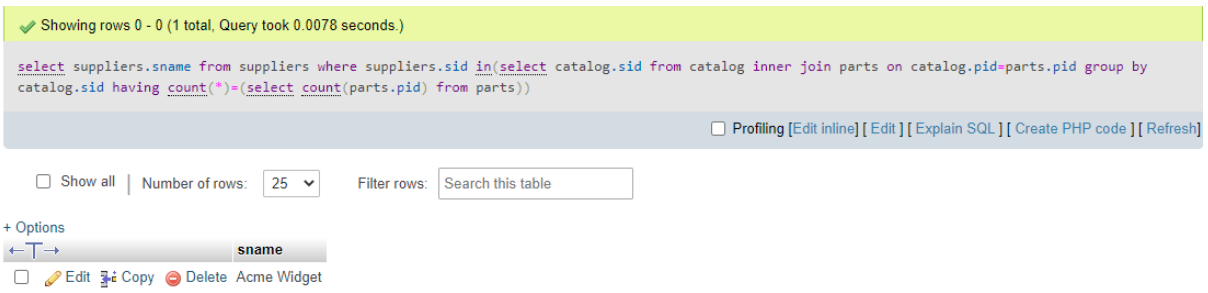
```
insert into parts values
(20001,'Book','Red'),(20002,'Pen','Red'),(20003,'Pencil','Green'),(20004,'Mobile','Green'),(20
005,'Charger','Black');
```

	pid	pname	color
Edit Copy Delete	20001	Book	Red
Edit Copy Delete	20002	Pen	Red
Edit Copy Delete	20003	Pencil	Green
Edit Copy Delete	20004	Mobile	Green
Edit Copy Delete	20005	Charger	Black

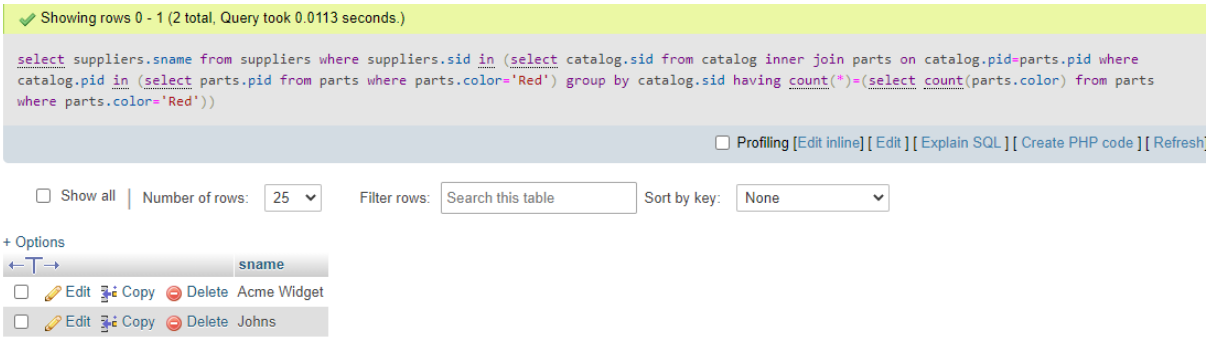
insert into catalog values(10001,20001,10),(10001,20002,10),(10001,20003,30),(10001,20004,10),(10001,20005,10),(10002,20001,10),(10002,20002,20),(10003,20003,30),(10004,20003,40);

sid	pid	cost
10001	20001	10
10002	20001	10
10002	20002	20
10004	20003	40
10001	20004	10
10001	20005	10

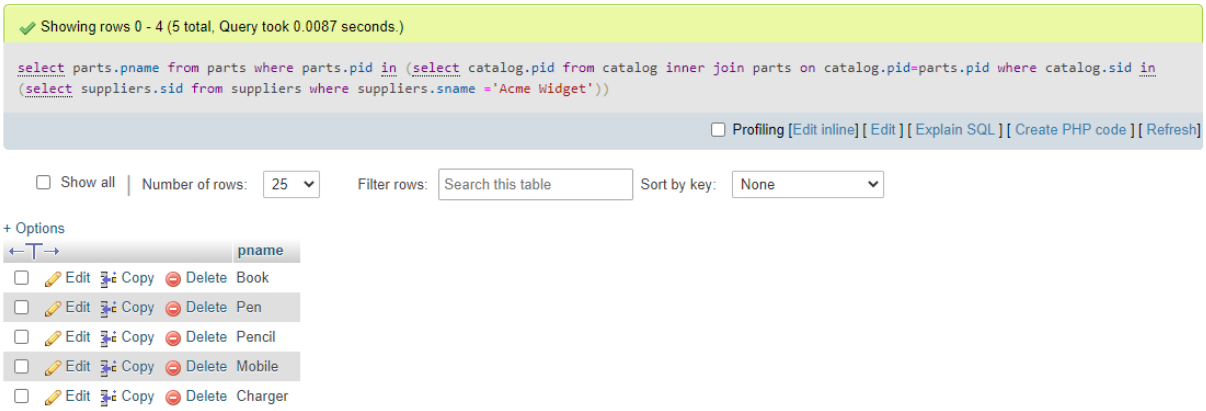
select suppliers.sname from suppliers where suppliers.sid in(select catalog.sid from catalog inner join parts on catalog.pid=parts.pid group by catalog.sid having count(*)=(select count(parts.pid) from parts)); #2



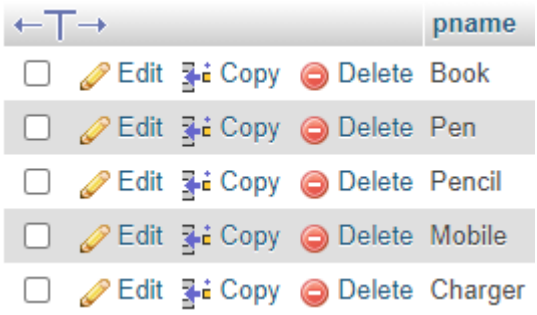
select suppliers.sname from suppliers where suppliers.sid in (select catalog.sid from catalog inner join parts on catalog.pid=parts.pid where catalog.pid in (select parts.pid from parts where parts.color='Red') group by catalog.sid having count(*)=(select count(parts.color) from parts where parts.color='Red')); #3



select parts.pname from parts where parts.pid in (select catalog.pid from catalog inner join parts on catalog.pid=parts.pid where catalog.sid in (select suppliers.sid from suppliers where suppliers.sname ='Acme Widget')); #4



select pname from parts where parts.pid in (select catalog.pid from catalog group by catalog.pid); #1



select catalog.pid,avg(catalog.cost) from catalog group by catalog.pid;

select * from catalog ou where ou.cost >= (select avg(inn.cost) from catalog inn where inn.pid=ou.pid) order by pid; #5

Showing rows 0 - 5 (6 total, Query took 0.0075 seconds.)

```
select * from catalog ou where ou.cost >= ( select avg(inn.cost) from catalog inn where inn.pid=ou.pid) order by pid
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

sid	pid	cost
10001	20001	10
10002	20001	10
10002	20002	20
10004	20003	40
10001	20004	10
10001	20005	10

create table ref as(select * from catalog ou where ou.cost = (select max(inn.cost) from catalog inn where inn.pid=ou.pid) order by pid); #ref Table For Query 6

select ref.sid,suppliers.sname,ref.pid,ref.cost from suppliers inner join ref on suppliers.sid=ref.sid order by ref.pid;#6

Showing rows 0 - 5 (6 total, Query took 0.0146 seconds.) [pid: **20001...** - **20005...**]

```
select ref.sid,suppliers.sname,ref.pid,ref.cost from suppliers inner join ref on suppliers.sid=ref.sid order by ref.pid
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

sid	sname	pid	cost
10002	Johns	20001	10
10001	Acme Widget	20001	10
10002	Johns	20002	20
10004	Reliance	20003	40
10001	Acme Widget	20004	10
10001	Acme Widget	20005	10

PROGRAM 4: STUDENT_FACULTY DATABASE

Consider the following database for student enrolment for course:

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

CLASS (name: string, meets at: time, room: string, fid: integer)

ENROLLED (snum: integer, cname: string)

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

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class such that the student is enrolled in the class. Level is a two character code with 4 different values (example: Junior: JR etc) Write the following queries in SQL. No duplicates should be printed in any of the answers.

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

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

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

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

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

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

vii. 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)

```
create database student_faculty;

use student_faculty;

create table student(

snum int primary key,

sname varchar(30),
```

```
major varchar(30),
lvl varchar(30),
age int
);

create table faculty(
fid int primary key,
fname varchar(30),
dept_id int
);

create table class(
cname varchar(30) primary key,
meets_at varchar(30),
room varchar(30),
fid int,
constraint fk_1
foreign key(fid) references faculty(fid)
);

create table enrolled(
snum int ,
cname varchar(30),
constraint fk_2
foreign key(snum) references student(snum),
constraint fk_3
foreign key(cname) references class(cname)
);
```

insert into student values
(1,'jhon','CS','SR',19),(2,'Smith','CS','JR',17),(3,'Jacob','CV','FR',20),(4,'Tom','CS','FR',25),(5,'Rahul','CS','JR',20),(6,'RANa','CS','FR',21);

	snum	sname	major	lvl	age
Edit Copy Delete	1	Jhon	CS	SR	19
Edit Copy Delete	2	Smith	CS	JR	17
Edit Copy Delete	3	Jacob	CV	FR	20
Edit Copy Delete	4	Tom	CS	FR	25
Edit Copy Delete	5	Rahul	CS	JR	20
Edit Copy Delete	6	Rana	CS	FR	21

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

	fid	fname	dept_id
Edit Copy Delete	11	Harish	1000
Edit Copy Delete	12	MV	1000
Edit Copy Delete	13	Mira	1001
Edit Copy Delete	14	Shiva	1002
Edit Copy Delete	15	Nupur	1000

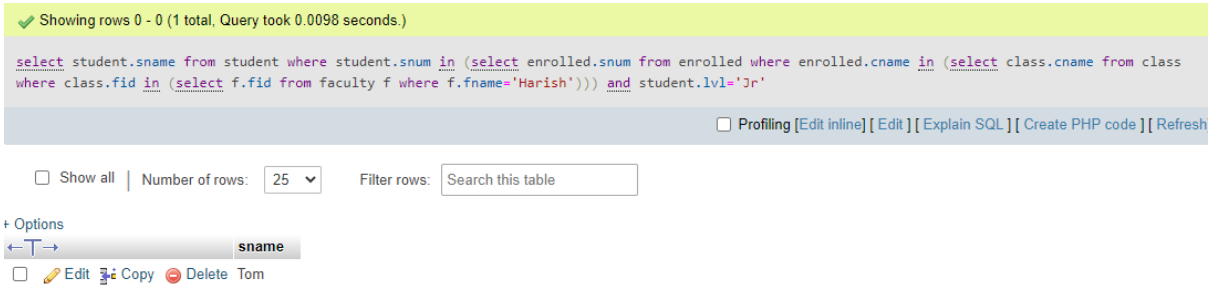
insert into class values ('class1','12/11/15 10:15:16','R1',14),('class10','12/11/15 10:15:16','R128',14),('class2','12/11/15 10:15:20','R2',12),('class3','12/11/15 10:15:25','R3',11),('class4','12/11/15 20:15:20','R4',14),('class5','12/11/15 20:15:20','R3',15),('class6','12/11/15 13:20:20','R128',14),('class7','12/11/15 10:10:10','R3',14);

	cname	meets_at	room	fid
Edit Copy Delete	class1	12/11/15 10:15:16	R1	14
Edit Copy Delete	class10	12/11/15 10:15:16	R128	14
Edit Copy Delete	class2	12/11/15 10:15:20	R2	12
Edit Copy Delete	class3	12/11/15 10:15:25	R3	11
Edit Copy Delete	class4	12/11/15 20:15:20	R4	14
Edit Copy Delete	class5	12/11/15 20:15:20	R3	15
Edit Copy Delete	class6	12/11/15 13:20:20	R128	14
Edit Copy Delete	class7	12/11/15 10:10:10	R3	14

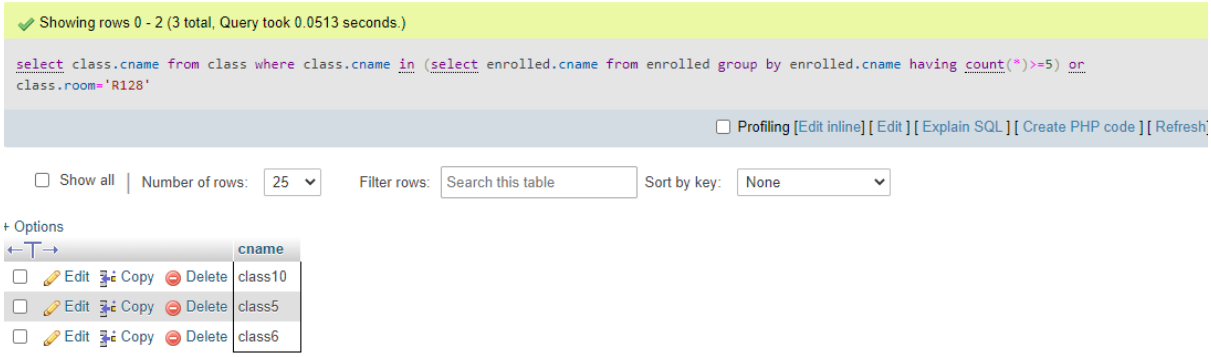
insert into enrolled values
(1,'class1'),(2,'class1'),(3,'class3'),(4,'class3'),(5,'class4'),(1,'class5'),(2,'class5'),(3,'class5'),(4,'class5'),(5,'class5');

snum	cname
1	class1
2	class1
3	class3
4	class3
5	class4
1	class5
2	class5
3	class5
4	class5
5	class5

select student.sname from student where student.snum in (select enrolled.snum from enrolled where enrolled.cname in (select class.cname from class where class.fid in (select f.fid from faculty f where f.fname='Harish'))) and student.lvl='Jr'; #1



select class.cname from class where class.cname in (select enrolled.cname from enrolled group by enrolled.cname having count(*)>=5) or class.room='R128'; #2



create table ref as(

select ou.cname,ou.meets_at from class ou where exists (select inn.cname from class inn where inn.meets_at=ou.meets_at having count(*)>1)

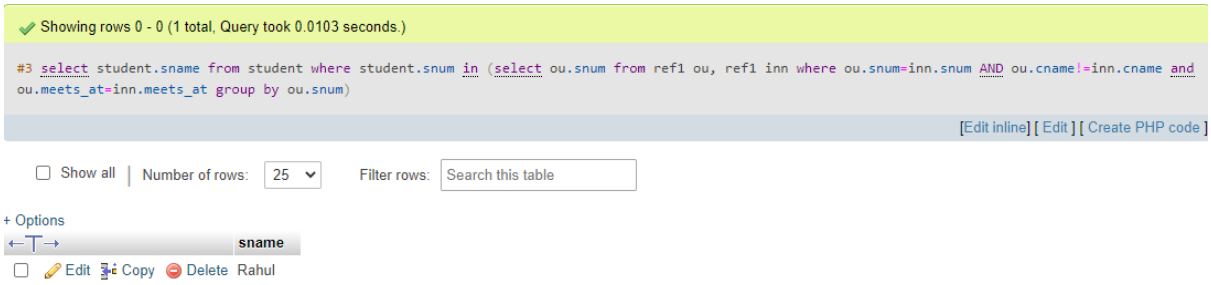
); #3

create table ref1 as(

select enrolled.snum,enrolled.cname,ref.meets_at from enrolled inner join ref on enrolled.cname=ref.cname

); #3

select student.sname from student where student.snum in (select ou.snum from ref1 ou, ref1 inn where ou.snum=inn.snum AND ou.cname!=inn.cname and ou.meets_at=inn.meets_at group by ou.snum); #3



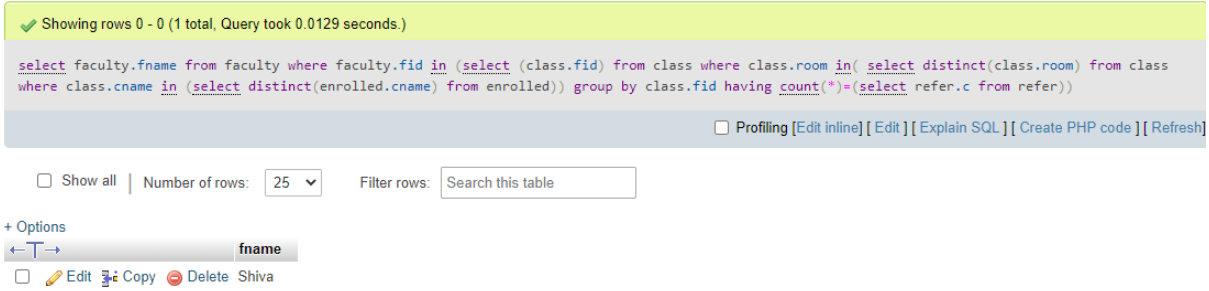
create table ref1 as(

select count(distinct(class.room))as c from class where class.cname in (select distinct(enrolled.cname) from enrolled)

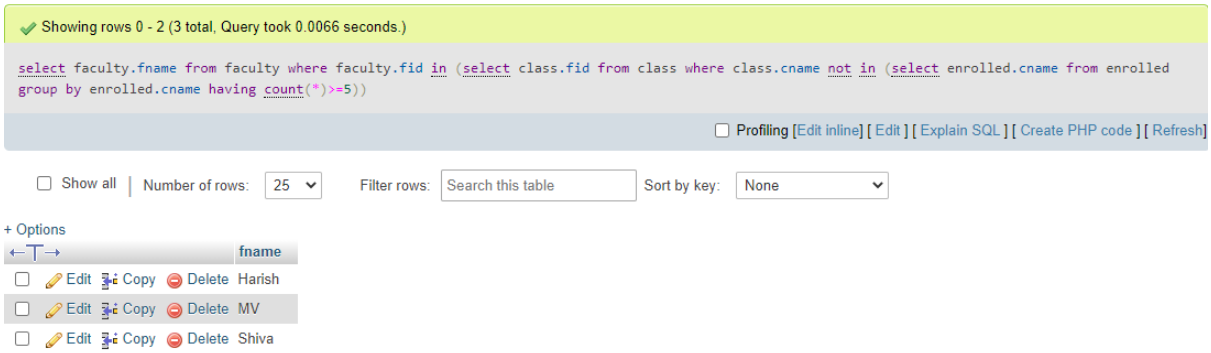
); #4

select faculty.fname from faculty where faculty.fid in (select (class.fid) from class where class.room in(select distinct(class.room) from class where class.cname in (select distinct(enrolled.cname) from enrolled)) group by class.fid having count(*)=(select ref1.c from ref1)); #4

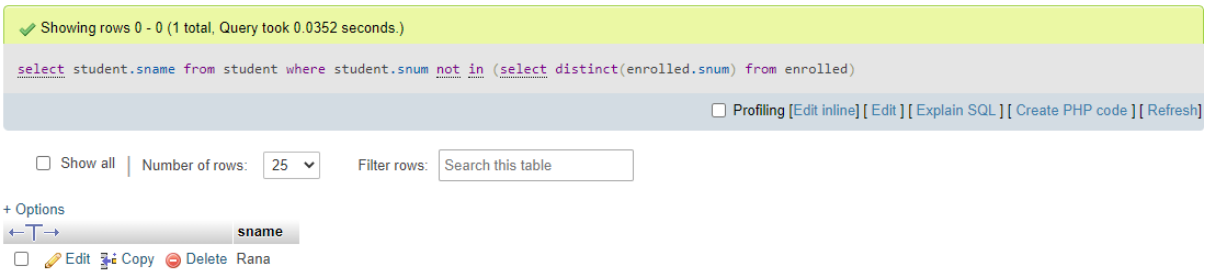
select distinct f.fname from faculty f where not exists ((select c.room from class c) MINUS (select c1.room from class c1 where c1.fid=f.fid));



select faculty.fname from faculty where faculty.fid in (select class.fid from class where class.cname not in (select enrolled.cname from enrolled group by enrolled.cname having count(*)>=5)); #5



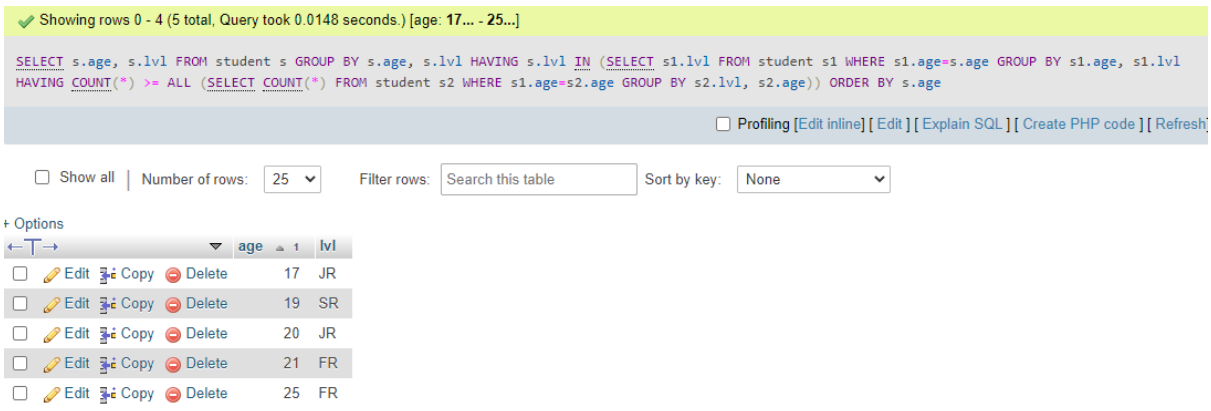
select student.sname from student where student.snum not in (select distinct(enrolled.snum) from enrolled); #6



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.age, s1.lvl HAVING COUNT(*) >= ALL (SELECT COUNT(*) FROM

student s2 WHERE s1.age=s2.age GROUP BY s2.lvl, s2.age)) ORDER BY s.age; #7



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)

EMPLOYEE (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. Write each of the following queries in SQL.

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

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

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

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

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

vi. 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.

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

```
CREATE DATABASE AIRLINE;

USE AIRLINE;

CREATE TABLE FLIGHTS(

 FL_ID INT,FROM_PLACE VARCHAR(20),
```

```
    TO_PLACE VARCHAR(20),
    DISTANCE INT,
    DEPARTS TIME,
    ARRIVES TIME,
    PRICE INT,
    PRIMARY KEY(FL_ID));

CREATE TABLE AIRCRAFT(
    A_ID INT,
    A_NAME VARCHAR(10),
    CRUISING_RANGE INT,
    PRIMARY KEY(A_ID));

CREATE TABLE EMPLOYEE(
    E_ID INT,
    E_NAME VARCHAR(10),
    SALARY INT,
    PRIMARY KEY(E_ID));

CREATE TABLE CERTIFIED(
    E_ID INT,
    A_ID INT,
    FOREIGN KEY(E_ID) REFERENCES EMPLOYEE(E_ID),
    FOREIGN KEY(A_ID) REFERENCES AIRCRAFT(A_ID));

INSERT INTO FLIGHTS VALUES(111,'BENGALURU','FRANKFURT',1000,'09:30','16:00',10000);
INSERT INTO FLIGHTS VALUES(222,'MANDISON','BENGALURU',1020,'01:30','7:00',9000);
INSERT INTO FLIGHTS VALUES(333,'BENGALURU','FRANKFURT',1000,'01:40','12:00',9500);
INSERT INTO FLIGHTS VALUES(555,'NEW DELHI','NEW YORK',5000,'13:30','17:00',15000);
```

INSERT INTO FLIGHTS VALUES(444,'BENGALURU','NEW DELHI',550,'08:00','13:00',5000);

INSERT INTO FLIGHTS VALUES(666,'MANDISION','NEW YORK',12000,'16:30','19:00',20000);

	FL_ID	FROM_PLACE	TO_PLACE	DISTANCE	DEPARTS	ARRIVES	PRICE
Edit Copy Delete	111	BENGALURU	FRANKFURT	1000	09:30:00	16:00:00	10000
Edit Copy Delete	222	MANDISON	BENGALURU	1020	01:30:00	07:00:00	9000
Edit Copy Delete	333	BENGALURU	FRANKFURT	1000	01:40:00	12:00:00	9500
Edit Copy Delete	444	BENGALURU	NEW DELHI	550	08:00:00	13:00:00	5000
Edit Copy Delete	555	NEW DELHI	NEW YORK	5000	13:30:00	17:00:00	15000
Edit Copy Delete	666	MANDISION	NEW YORK	12000	16:30:00	19:00:00	20000

SELECT * FROM FLIGHTS;

INSERT INTO AIRCRAFT VALUES(10,'AIR ASIA',12000);

INSERT INTO AIRCRAFT VALUES(20,'GO AIR',2000);

INSERT INTO AIRCRAFT VALUES(30,'AIR ASIA',600);

INSERT INTO AIRCRAFT VALUES(40,'INDIGO',5000);

INSERT INTO AIRCRAFT VALUES(50,'SPICE JET',900);

INSERT INTO AIRCRAFT VALUES(60,'SPICE JET',12500);

	A_ID	A_NAME	CRUISING_RANGE
Edit Copy Delete	10	AIR ASIA	12000
Edit Copy Delete	20	GO AIR	2000
Edit Copy Delete	30	AIR ASIA	600
Edit Copy Delete	40	INDIGO	5000
Edit Copy Delete	50	SPICE JET	900
Edit Copy Delete	60	SPICE JET	12500

SELECT * FROM AIRCRAFT;

INSERT INTO EMPLOYEE VALUES(3,'SHAAN',60000);

INSERT INTO EMPLOYEE VALUES(2,'TANYA',15000);

INSERT INTO EMPLOYEE VALUES(4,'RAJATH',81000);

INSERT INTO EMPLOYEE VALUES(1,'SAM',82000);

INSERT INTO EMPLOYEE VALUES(6,'AYMAN',70000);

INSERT INTO EMPLOYEE VALUES(5,'SAAKSHI',9000);

	E_ID	E_NAME	SALARY
Edit Copy Delete	1	SAM	82000
Edit Copy Delete	2	TANYA	15000
Edit Copy Delete	3	SHAAN	60000
Edit Copy Delete	4	RAJATH	81000
Edit Copy Delete	5	SAAKSHI	9000
Edit Copy Delete	6	AYMAN	70000

SELECT * FROM EMPLOYEE;

INSERT INTO CERTIFIED VALUES(1,10);

INSERT INTO CERTIFIED VALUES(4,10);

INSERT INTO CERTIFIED VALUES(3,40);

INSERT INTO CERTIFIED VALUES(5,30);

INSERT INTO CERTIFIED VALUES(5,50);

INSERT INTO CERTIFIED VALUES(4,50);

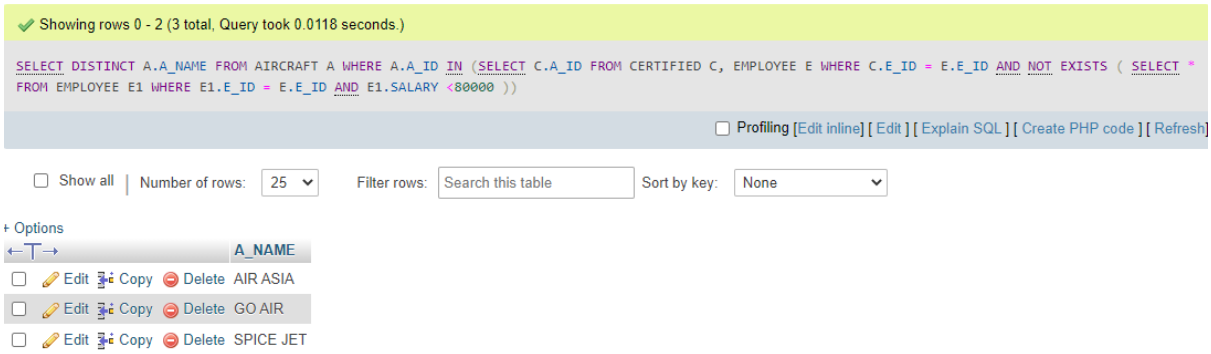
INSERT INTO CERTIFIED VALUES(4,20);

INSERT INTO CERTIFIED VALUES(4,30);

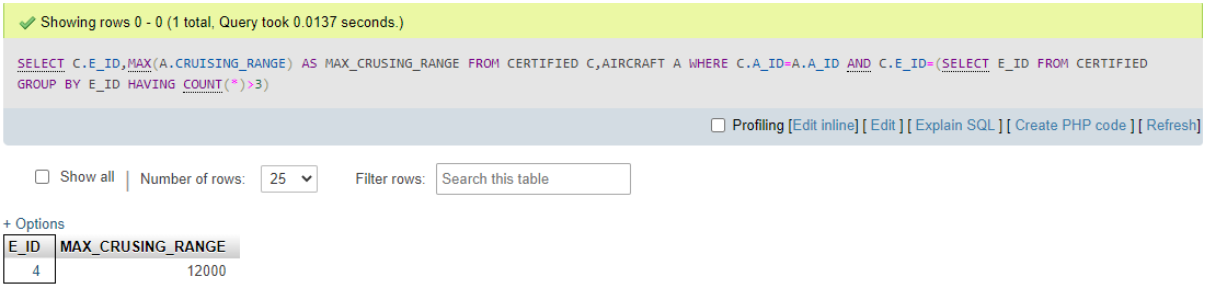
E_ID	A_ID
1	10
4	10
3	40
5	30
5	50
4	50
4	20
4	30

SELECT * FROM CERTIFIED;

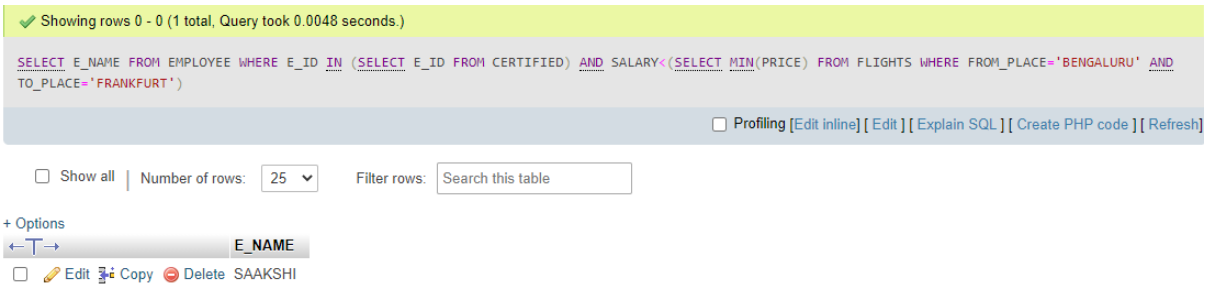
SELECT DISTINCT A.A_NAME FROM AIRCRAFT A WHERE A.A_ID IN (SELECT C.A_ID FROM CERTIFIED C, EMPLOYEE E WHERE C.E_ID = E.E_ID AND NOT EXISTS (SELECT * FROM EMPLOYEE E1 WHERE E1.E_ID = E.E_ID AND E1.SALARY <80000)); #1



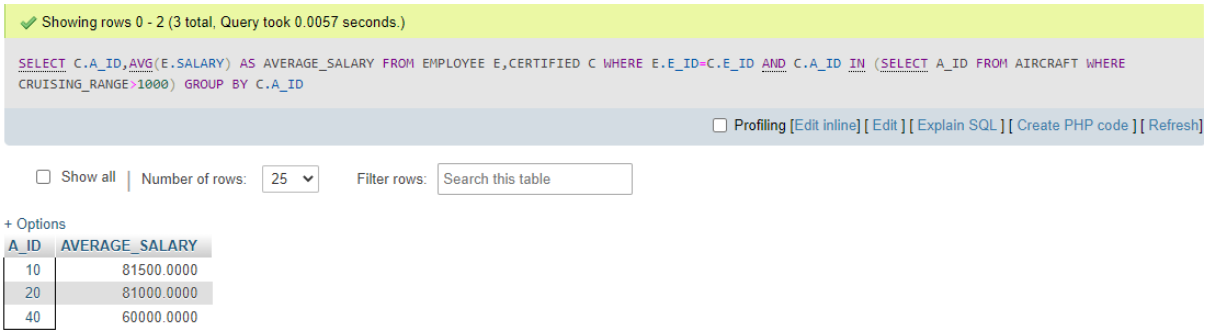
SELECT C.E_ID,MAX(A.CRUISING_RANGE) AS MAX_CRUSING_RANGE FROM CERTIFIED C,AIRCRAFT A WHERE C.A_ID=A.A_ID AND C.E_ID=(SELECT E_ID FROM CERTIFIED GROUP BY E_ID HAVING COUNT(*)>3); #2



SELECT E_NAME FROM EMPLOYEE WHERE E_ID IN (SELECT E_ID FROM CERTIFIED) AND SALARY<(SELECT MIN(PRICE) FROM FLIGHTS WHERE FROM_PLACE='BENGALURU' AND TO_PLACE='FRANKFURT'); #3



SELECT C.A_ID,AVG(E.SALARY) AS AVERAGE_SALARY FROM EMPLOYEE E,CERTIFIED C WHERE E.E_ID=C.E_ID AND C.A_ID IN (SELECT A_ID FROM AIRCRAFT WHERE CRUISING_RANGE>1000) GROUP BY C.A_ID; #4



SELECT A_ID FROM AIRCRAFT WHERE CRUISING_RANGE>(SELECT DISTANCE FROM FLIGHTS WHERE FROM_PLACE='BENGALURU' AND TO_PLACE='NEW DELHI'); #5

SELECT F.DEPARTS FROM FLIGHTS F WHERE F.FL_ID IN ((SELECT F0.FL_ID FROM FLIGHTS F0 WHERE F0.FROM_PLACE = 'MANDISION' AND F0.TO_PLACE = 'NEW YORK'

AND EXTRACT(HOUR FROM F0.ARRIVES) < 18) UNION (SELECT F0.FL_ID FROM FLIGHTS F0, FLIGHTS F1 WHERE F0.FROM_PLACE = 'MANDISION' AND F0.TO_PLACE <> 'NEW YORK'

AND F0.TO_PLACE = F1.FROM_PLACE AND F1.TO_PLACE = 'NEW YORK' AND F1.DEPARTS > F0.ARRIVES AND EXTRACT(HOUR FROM F1.ARRIVES) < 18)

UNION(SELECT F0.FL_ID FROM FLIGHTS F0, FLIGHTS F1, FLIGHTS F2 WHERE F0.FROM_PLACE = 'MANDISION' AND F0.TO_PLACE = F1.FROM_PLACE AND F1.TO_PLACE = F2.FROM_PLACE

AND F2.TO_PLACE = 'NEW YORK' AND F0.TO_PLACE <> 'NEW YORK' AND F1.TO_PLACE <> 'NEW YORK' AND F1.DEPARTS > F0.ARRIVES AND F2.DEPARTS > F1.ARRIVES

AND EXTRACT(HOUR FROM F2.ARRIVES) < 18)); #6

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0511 seconds.)

```
SELECT F.DEPARTS FROM FLIGHTS F WHERE F.FL_ID IN ( ( SELECT F0.FL_ID FROM FLIGHTS F0 WHERE F0.FROM_PLACE = 'MANDISION' AND F0.TO_PLACE = 'NEW
YORK' AND EXTRACT(HOUR FROM F0.ARRIVES) < 18 ) UNION ( SELECT F0.FL_ID FROM FLIGHTS F0, FLIGHTS F1 WHERE F0.FROM_PLACE = 'MANDISION' AND
F0.TO_PLACE <> 'NEW YORK' AND F0.TO_PLACE = F1.FROM_PLACE AND F1.TO_PLACE = 'NEW YORK' AND F1.DEPARTS > F0.ARRIVES AND EXTRACT(HOUR FROM
F1.ARRIVES) < 18 ) UNION( SELECT F0.FL_ID FROM FLIGHTS F0, FLIGHTS F1, FLIGHTS F2 WHERE F0.FROM_PLACE = 'MANDISION' AND F0.TO_PLACE =
F1.FROM_PLACE AND F1.TO_PLACE = F2.FROM_PLACE AND F2.TO_PLACE = 'NEW YORK' AND F0.TO_PLACE <> 'NEW YORK' AND F1.TO_PLACE <> 'NEW YORK' AND
F1.DEPARTS > F0.ARRIVES AND F2.DEPARTS > F1.ARRIVES AND EXTRACT(HOUR FROM F2.ARRIVES) < 18 ))
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

DEPARTS

SELECT E_NAME,SALARY FROM EMPLOYEE WHERE E_ID NOT IN(SELECT E_ID FROM CERTIFIED) AND SALARY>(SELECT AVG(SALARY) FROM EMPLOYEE WHERE E_ID IN (SELECT E_ID FROM CERTIFIED)); #7

Showing rows 0 - 0 (1 total, Query took 0.0064 seconds.)

```
SELECT E_NAME,SALARY FROM EMPLOYEE WHERE E_ID NOT IN(SELECT E_ID FROM CERTIFIED) AND SALARY>(SELECT AVG(SALARY) FROM EMPLOYEE WHERE E_ID IN (SELECT
E_ID FROM CERTIFIED))
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

+ Options

	E_NAME	SALARY
Edit Copy Delete	AYMAN	70000