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',15,25
00),('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');

←T→				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');

←T→				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);

←T→	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 driver_id) cnt from participated a,accident b where b.report_number=b.report_number and b.dat like '%08
```

Profiling [Edit inline] [Edit] [Explain SQL]

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

+ Options

cnt
7

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

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

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

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

+ Options

cn
2

```
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'
);
```

✔ 2 rows affected. (Query took 0.1337 seconds.)

```
delete from accounts where branch_name in( select branch_name from branch where branch_city='RAICHUR' )
```

[Edit inline] [Edit] [Create PHP code]

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','Delhi');
```

				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'),(20005,'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

Showing rows 0 - 0 (1 total, Query took 0.0078 seconds.)

```
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))
```

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

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

+ Options

	sname
Edit Copy Delete	Acme Widget

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

Showing rows 0 - 1 (2 total, Query took 0.0113 seconds.)

```
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'))
```

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

←T→	sname
Edit Copy Delete	Acme Widget
Edit Copy Delete	Johns

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

Showing rows 0 - 4 (5 total, Query took 0.0087 seconds.)

```
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'))
```

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

←T→	pname
Edit Copy Delete	Book
Edit Copy Delete	Pen
Edit Copy Delete	Pencil
Edit Copy Delete	Mobile
Edit Copy Delete	Charger

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

←T→	pname
Edit Copy Delete	Book
Edit Copy Delete	Pen
Edit Copy Delete	Pencil
Edit Copy Delete	Mobile
Edit Copy Delete	Charger

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

Showing rows 0 - 0 (1 total, Query took 0.0098 seconds.)

```
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'
```

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

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

+ Options

	sname
Edit Copy Delete	Tom

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

Showing rows 0 - 2 (3 total, Query took 0.0513 seconds.)

```
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'
```

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

	cname
Edit Copy Delete	class10
Edit Copy Delete	class5
Edit Copy Delete	class6

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

Showing rows 0 - 0 (1 total, Query took 0.0103 seconds.)

```
#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)
```

[Edit inline] [Edit] [Create PHP code]

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

+ Options

	sname
Edit Copy Delete	Rahul

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));

Showing rows 0 - 0 (1 total, Query took 0.0129 seconds.)

```
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 refer.c from refer))
```

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

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

+ Options

	fname
Edit Copy Delete	Shiva

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

Showing rows 0 - 2 (3 total, Query took 0.0066 seconds.)

```
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))
```

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

	fname
Edit Copy Delete	Harish
Edit Copy Delete	MV
Edit Copy Delete	Shiva

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

Showing rows 0 - 0 (1 total, Query took 0.0352 seconds.)

```
select student.sname from student where student.snum not in (select distinct(enrolled.snum) from enrolled)
```

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

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

+ Options

	sname
Edit Copy Delete	Rana

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

Showing rows 0 - 4 (5 total, Query took 0.0148 seconds.) [age: 17... - 25...]

```
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
```

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

	age	lvl
Edit Copy Delete	17	JR
Edit Copy Delete	19	SR
Edit Copy Delete	20	JR
Edit Copy Delete	21	FR
Edit Copy Delete	25	FR

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

Showing rows 0 - 2 (3 total, Query took 0.0118 seconds.)

```
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 ))
```

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

	A_NAME
Edit Copy Delete	AIR ASIA
Edit Copy Delete	GO AIR
Edit Copy Delete	SPICE JET

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

Showing rows 0 - 0 (1 total, Query took 0.0137 seconds.)

```
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)
```

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

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

+ Options

E_ID	MAX_CRUSING_RANGE
4	12000

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

Showing rows 0 - 0 (1 total, Query took 0.0048 seconds.)

```
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')
```

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

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

+ Options

	E_NAME
Edit Copy Delete	SAAKSHI

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

Showing rows 0 - 2 (3 total, Query took 0.0057 seconds.)

```
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
```

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

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

+ Options

A_ID	AVERAGE_SALARY
10	81500.0000
20	81000.0000
40	60000.0000

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

Showing rows 0 - 5 (6 total, Query took 0.0039 seconds.)

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

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

←T→	A_ID
Edit Copy Delete	10
Edit Copy Delete	20
Edit Copy Delete	30
Edit Copy Delete	40
Edit Copy Delete	50
Edit Copy Delete	60

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

←T→	E_NAME	SALARY
Edit Copy Delete	AYMAN	70000

PROGRAM 6: ORDER PROCESSING DATABASE

Consider the following relations for an Order Processing database application in a company.

CUSTOMER (CUST #: int, cname: String, city: String) ORDER (order #: int, odate: date, cust #: int, ord-Amt: int)

ITEM (item #: int, unit-price: int)

ORDER-ITEM (order #: int, item #: int, qty: int)

WAREHOUSE (warehouse #: int, city: String)

SHIPMENT (order #: int, warehouse #: int, ship-date: date)

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

ii. Enter at least five tuples for each relation.

iii. Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column is the total numbers of orders by the customer and the last column is the average order amount for that customer.

iv. List the order# for orders that were shipped from all warehouses that the company has in a specific city.

v. Demonstrate how you delete item# 10 from the ITEM table and make that field null in the ORDER_ITEM table.

```
create database orderpro;

use orderpro;


CREATE TABLE CUSTOMERS(

   CUSTNO INT,

   CNAME VARCHAR2(50),

   CITY VARCHAR2(50),

   PRIMARY KEY(CUSTNO)

   );
```

```
CREATE TABLE ORDERS(
   ORDERNO INT,
   ODATE DATE,
   CUSTNO INT,
   ORD_AMT INT,
   PRIMARY KEY(ORDERNO),
   FOREIGN KEY(CUSTNO) REFERENCES CUSTOMERS(CUSTNO)
   );

CREATE TABLE ITEM(
   ITEMNO INT,
   UNIT_PRICE INT,
   PRIMARY KEY(ITEMNO)
   );

CREATE TABLE ORDER_ITEM(
   ORDERNO INT,
   ITEMNO INT,
   QTY INT,
   PRIMARY KEY(ORDERNO, ITEMNO),
   FOREIGN KEY(ORDERNO) REFERENCES ORDERS(ORDERNO),
   FOREIGN KEY(ITEMNO) REFERENCES ITEM(ITEMNO)
   );

CREATE TABLE WAREHOUSE(
   WAREHOUSENO INT,
   CITY VARCHAR2(50),
   PRIMARY KEY(WAREHOUSENO)
   );
```

```
CREATE TABLE SHIPMENT(
   ORDERNO INT,
   WAREHOUSENO INT,
   SHIP_DATE DATE,
   PRIMARY KEY(ORDERNO, WAREHOUSENO),
   FOREIGN KEY(ORDERNO) REFERENCES ORDERS(ORDERNO),
   FOREIGN KEY(WAREHOUSENO) REFERENCES WAREHOUSE(WAREHOUSENO)
   );

INSERT INTO CUSTOMERS VALUES(CUSTNO, 'CNAME', 'CITY');
INSERT INTO ORDERS VALUES(ORDERNO, 'ODATE', CUSTNO, ORD_AMT);
INSERT INTO ITEM VALUES(ITEMNO, UNIT_PRICE);
INSERT INTO ORDER_ITEM VALUES(ORDERNO, ITEMNO, QTY);
INSERT INTO WAREHOUSE VALUES(WAREHOUSENO, 'CITY');
INSERT INTO SHIPMENT VALUES(ORDERNO, WAREHOUSENO, 'SHIP_DATE');

SELECT * FROM CUSTOMERS;
SELECT * FROM ORDERS;
SELECT * FROM ITEM;
SELECT * FROM ORDER_ITEM;
SELECT * FROM WAREHOUSE;
SELECT * FROM SHIPMENT;

INSERT INTO customers (custno, `cname`, `city`) VALUES
(111, 'hari', 'bangalore'),(112, 'siddu', 'bangalore'),(113, 'praveen', 'bangalore'),(114, 'yash', 'bangalore'),(115, 'anil', 'bangalore'),(116, 'piyush', 'bangalore');
```

	CUSTNO	CNAME	CITY
Edit Copy Delete	0	CNAME	CITY
Edit Copy Delete	111	hari	bangalore
Edit Copy Delete	112	siddu	bangalore
Edit Copy Delete	113	praveen	bangalore
Edit Copy Delete	114	yash	bangalore
Edit Copy Delete	115	anil	bangalore
Edit Copy Delete	116	piyush	bangalore

INSERT INTO item (itemno, unit_price) VALUES

(11, 500),(12, 600),(14, 300),(15, 200),(16, 1000);

	ITEMNO	UNIT_PRICE
Edit Copy Delete	0	NULL
Edit Copy Delete	11	500
Edit Copy Delete	12	600
Edit Copy Delete	14	300
Edit Copy Delete	15	200
Edit Copy Delete	16	1000

INSERT INTO orders (orderno, `odate`, custno, ord_amt) VALUES

(2001, '2020-06-11', 111, 1200),(2002, '2020-05-12', 112, 1300),

(2003, '2020-08-13', 113, 1100),(2004, '2020-11-14', 114, 1400),

(2005, '2020-06-15', 115, 1500),(2006, '2020-12-16', 116, 1100);

	ORDERNO	ODATE	CUSTNO	ORD_AMT
Edit Copy Delete	0	0000-00-00	NULL	NULL
Edit Copy Delete	2001	2020-06-11	111	1200
Edit Copy Delete	2002	2020-05-12	112	1300
Edit Copy Delete	2003	2020-08-13	113	1100
Edit Copy Delete	2004	2020-11-14	114	1400
Edit Copy Delete	2005	2020-06-15	115	1500
Edit Copy Delete	2006	2020-12-16	116	1100

INSERT INTO order_item (orderno, itemno, qty) VALUES

(2001, 11, 2),(2002, 16, 1),(2003, 15, 5),(2004, 14, 3),(2006, 12, 1);

←T→	ORDERNO	ITEMNO	QTY
Edit Copy Delete	0	0	NULL
Edit Copy Delete	2001	11	2
Edit Copy Delete	2002	16	1
Edit Copy Delete	2003	15	5
Edit Copy Delete	2004	14	3
Edit Copy Delete	2006	12	1

INSERT INTO shipment (orderno, warehouseno, `ship_date`) VALUES

(2001, 31, '2020-06-11'),(2002, 32, '2020-05-12'),

(2003, 33, '2020-08-13'),(2004, 34, '2020-11-14'),

(2005, 35, '2020-06-15');

←T→	ORDERNO	WAREHOUSENO	SHIP_DATE
Edit Copy Delete	0	0	0000-00-00
Edit Copy Delete	2001	31	2020-06-11
Edit Copy Delete	2002	32	2020-05-12
Edit Copy Delete	2003	33	2020-08-13
Edit Copy Delete	2004	34	2020-11-14
Edit Copy Delete	2005	35	2020-06-15

INSERT INTO warehouse (warehouseno, `city`) VALUES

(31, 'ranchi'),(32, 'bangalore'),(33, 'mumbai'),(34, 'delhi'),(35, 'punjab');

←T→	WAREHOUSENO	CITY
Edit Copy Delete	0	CITY
Edit Copy Delete	31	ranchi
Edit Copy Delete	32	bangalore
Edit Copy Delete	33	mumbai
Edit Copy Delete	34	delhi
Edit Copy Delete	35	punjab

select c.custno,count(*) as No_of_orders,Avg_order_amt from customers c,orders o where o.custno=c.custno group by c.custno; #1

Showing rows 0 - 5 (6 total, Query took 0.1154 seconds.)

```
select c.custno,count(*) as No_of_orders, ord_amt from customers c,orders o where o.custno=c.custno group by c.custno
```

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

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

+ Options

custno	No_of_orders	ord_amt
111	1	1200
112	1	1300
113	1	1100
114	1	1400
115	1	1500
116	1	1100

select s.orderno from shipment s,warehouse w where s.warehouseno=w.warehouseno and w.city='banglore' group by orderno having count(*)=(select count(*) from warehouse where city='banglore') and not(count(*)=0); #2

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0013 seconds.)

```
select s.orderno from shipment s,warehouse w where s.warehouseno=w.warehouseno and w.city='banglore' group by orderno having count(*)=(select
count(*) from warehouse where city='banglore') and not(count(*)=0)
```

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

orderno

delete from item where itemno=15; #3

1 row affected. (Query took 0.0992 seconds.)

```
delete from order_item where itemno=15
```

[Edit inline] [Edit] [Create PHP code]

PROGRAM 7: BOOK DEALER DATABASE

The following tables are maintained by a book dealer:

AUTHOR(author-id: int, name: String, city: String, country: String)
PUBLISHER(publisher-id: int, name: String, city: String, country: String)
CATALOG (book-id: int, title: String, author-id: int, publisher-id: int, category-id: int, year: int, price: int)

CATEGORY(category-id: int, description: String)

ORDER-DETAILS(order-no: int, book-id: int, quantity: 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. Give the details of the authors who have 2 or more books in the catalog and the price of the books in the catalog and the year of publication is after 2000.

iv. Find the author of the book which has maximum sales.

v. Demonstrate how you increase the price of books published by a specific publisher by 10%.

```
CREATE DATABASE BOOK;

USE BOOK;


CREATE TABLE AUTHORS(

   AUTHOR_ID INT,

   A_NAME VARCHAR(10),

   CITY VARCHAR(10),

   COUNTRY VARCHAR(10),

   PRIMARY KEY(AUTHOR_ID)

   );


CREATE TABLE PUBLISHERS(
```

```
    PUBLISHER_ID INT,
    P_NAME VARCHAR(10),
    CITY VARCHAR(20),
    COUNTRY VARCHAR(10),
    PRIMARY KEY(PUBLISHER_ID)
    );

CREATE TABLE CATEGORY(
    CATEGORY_ID INT,
    DESCRIPTIONS VARCHAR(10),
    PRIMARY KEY(CATEGORY_ID)
    );

CREATE TABLE CATALOG(
    BOOK_ID INT,
    TITLE VARCHAR(10),
    AUTHOR_ID INT,
    PUBLISHER_ID INT,
    CATEGORY_ID INT,
    P_YEAR INT,
    PRICE INT,
    PRIMARY KEY(BOOK_ID),
    FOREIGN KEY(AUTHOR_ID) REFERENCES AUTHORS(AUTHOR_ID),
    FOREIGN KEY(PUBLISHER_ID) REFERENCES PUBLISHERS(PUBLISHER_ID),
    FOREIGN KEY(CATEGORY_ID) REFERENCES CATEGORY(CATEGORY_ID)
    );

CREATE TABLE ORDER_DETAILS(
    ORDER_NO INT,
```

```
BOOK_ID INT,
QTY INT,
PRIMARY KEY(ORDER_NO),
FOREIGN KEY (BOOK_ID) REFERENCES CATALOG(BOOK_ID)
);
```

```
INSERT INTO AUTHORS VALUES(10,'SAM','BLORE','INDIA');
INSERT INTO AUTHORS VALUES(20,'SHAAN','COORG','INDIA');
INSERT INTO AUTHORS VALUES(30,'SARA','CA','USA');
INSERT INTO AUTHORS VALUES(40,'AYMAN','SPAIN','BARCELONA');
INSERT INTO AUTHORS VALUES(50,'TANYA','DELHI','INDIA');
```

	AUTHOR_ID	A_NAME	CITY	COUNTRY
Edit Copy Delete	10	SAM	BLORE	INDIA
Edit Copy Delete	20	SHAAN	COORG	INDIA
Edit Copy Delete	30	SARA	CA	USA
Edit Copy Delete	40	AYMAN	SPAIN	BARCELONA
Edit Copy Delete	50	TANYA	DELHI	INDIA

```
SELECT * FROM AUTHORS;
```

```
INSERT INTO PUBLISHERS VALUES(1,'PHI','NY','USA');
INSERT INTO PUBLISHERS VALUES(2,'EEE','LA','USA');
INSERT INTO PUBLISHERS VALUES(3,'SWAPNA','','INDIA');
INSERT INTO PUBLISHERS VALUES(4,'MGH','NY','USA');
INSERT INTO PUBLISHERS VALUES(5,'PEARSON','DELHI','INDIA');
```

	PUBLISHER_ID	P_NAME	CITY	COUNTRY
Edit Copy Delete	1	PHI	NY	USA
Edit Copy Delete	2	EEE	LA	USA
Edit Copy Delete	3	SWAPNA		INDIA
Edit Copy Delete	4	MGH	NY	USA
Edit Copy Delete	5	PEARSON	DELHI	INDIA

```
SELECT * FROM PUBLISHERS;
```

INSERT INTO CATEGORY VALUES(11,'CS');

INSERT INTO CATEGORY VALUES(22,'EC');

INSERT INTO CATEGORY VALUES(33,'OS');

INSERT INTO CATEGORY VALUES(44,'C++');

INSERT INTO CATEGORY VALUES(55,'DBMS');

	CATEGORY_ID	DESCRIPTIONS
Edit Copy Delete	11	CS
Edit Copy Delete	22	EC
Edit Copy Delete	33	OS
Edit Copy Delete	44	C++
Edit Copy Delete	55	DBMS

SELECT * FROM CATEGORY;

INSERT INTO CATALOG VALUES(111,'BASICS',50,2,11,2003,200);

INSERT INTO CATALOG VALUES(222,'PROG',40,5,44,1999,500);

INSERT INTO CATALOG VALUES(333,'MP',50,1,11,2009,900);

INSERT INTO CATALOG VALUES(444,'CIRCUITS',50,2,22,1997,300);

INSERT INTO CATALOG VALUES(555,'DATA',30,3,55,2005,600);

	BOOK_ID	TITLE	AUTHOR_ID	PUBLISHER_ID	CATEGORY_ID	P_YEAR	PRICE
Edit Copy Delete	111	BASICS	50	2	11	2003	200
Edit Copy Delete	222	PROG	40	5	44	1999	500
Edit Copy Delete	333	MP	50	1	11	2009	900
Edit Copy Delete	444	CIRCUITS	50	2	22	1997	300
Edit Copy Delete	555	DATA	30	3	55	2005	600

SELECT * FROM CATALOG;

INSERT INTO ORDER_DETAILS VALUES(123,222,100);

INSERT INTO ORDER_DETAILS VALUES(231,111,150);

INSERT INTO ORDER_DETAILS VALUES(143,333,90);

INSERT INTO ORDER_DETAILS VALUES(156,555,200);

INSERT INTO ORDER_DETAILS VALUES(218,444,70);

←T→	ORDER_NO	BOOK_ID	QTY
Edit Copy Delete	123	222	100
Edit Copy Delete	143	333	90
Edit Copy Delete	156	555	200
Edit Copy Delete	218	444	70
Edit Copy Delete	231	111	150

SELECT * FROM ORDER_DETAILS;

SELECT A.A_NAME,C.TITLE,C.PRICE FROM AUTHORS A,CATALOG C WHERE C.AUTHOR_ID=A.AUTHOR_ID AND C.P_YEAR>=2000 AND A.A_NAME=(SELECT A.A_NAME FROM AUTHORS A,CATALOG C WHERE A.AUTHOR_ID=C.AUTHOR_ID GROUP BY C.AUTHOR_ID HAVING COUNT(*)>=2); #1

Showing rows 0 - 1 (2 total, Query took 0.0095 seconds.)

```
SELECT A.A_NAME,C.TITLE,C.PRICE FROM AUTHORS A,CATALOG C WHERE C.AUTHOR_ID=A.AUTHOR_ID AND C.P_YEAR>=2000 AND A.A_NAME=(SELECT A.A_NAME FROM AUTHORS A,CATALOG C WHERE A.AUTHOR_ID=C.AUTHOR_ID GROUP BY C.AUTHOR_ID HAVING COUNT(*)>=2)
```

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

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

+ Options

A_NAME	TITLE	PRICE
TANYA	BASICS	200
TANYA	MP	900

SELECT A.A_NAME FROM AUTHORS A,CATALOG C,ORDER_DETAILS O WHERE O.BOOK_ID=C.BOOK_ID AND A.AUTHOR_ID=C.AUTHOR_ID AND O.BOOK_ID=(SELECT BOOK_ID FROM ORDER_DETAILS WHERE QTY=(SELECT MAX(QTY) FROM ORDER_DETAILS));
#2

Showing rows 0 - 0 (1 total, Query took 0.0147 seconds.)

```
SELECT A.A_NAME FROM AUTHORS A,CATALOG C,ORDER_DETAILS O WHERE O.BOOK_ID=C.BOOK_ID AND A.AUTHOR_ID=C.AUTHOR_ID AND O.BOOK_ID=(SELECT BOOK_ID FROM ORDER_DETAILS WHERE QTY=(SELECT MAX(QTY) FROM ORDER_DETAILS))
```

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

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

+ Options

←T→	A_NAME
Edit Copy Delete	SARA

UPDATE CATALOG SET PRICE=1.10*PRICE;

SELECT * FROM CATALOG;

←T→	BOOK_ID	TITLE	AUTHOR_ID	PUBLISHER_ID	CATEGORY_ID	P_YEAR	PRICE
Edit Copy Delete	111	BASICS	50	2	11	2003	220
Edit Copy Delete	222	PROG	40	5	44	1999	550
Edit Copy Delete	333	MP	50	1	11	2009	990
Edit Copy Delete	444	CIRCUITS	50	2	22	1997	330
Edit Copy Delete	555	DATA	30	3	55	2005	660

PROGRAM 08: STUDENT ENROLLMENT DATABASE

Consider the following database of student enrollment in courses and books adopted for each course.

STUDENT (regno: String, name: String, major: String, bdate: date)

COURSE (course #: int, cname: String, dept: String)

ENROLL (regno: String, cname: String, sem: int, marks: int)

BOOK_ADOPTION (course #: int, sem: int, book-ISBN: int)

TEXT(book-ISBN:int, book-title:String, publisher:String, author:String)

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 add a new text book to the database and make this book be adopted by some department.

iv. Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.

v. List any department that has all its adopted books published by a specific publisher.

```
CREATE DATABASE STUDENTENROLL;

USE STUDENTENROLL;


CREATE TABLE STUDENT(

   REGNO VARCHAR(20),

   NAME VARCHAR2(20),

   MAJOR VARCHAR2(5),

   BDATE DATE,
```

```
    PRIMARY KEY(REGNO)
    );

CREATE TABLE COURSE(
    COURSE_ID INT,
    CNAME VARCHAR(20),
    DEPT VARCHAR(4),
    PRIMARY KEY(COURSE_ID)
    );

CREATE TABLE TEXT(
    ISBN INT,
    BOOK_TITLE VARCHAR(50),
    PUBLISHER VARCHAR(20),
    AUTHOR VARCHAR(20),
    PRIMARY KEY(ISBN)
    );

CREATE TABLE ADOPTION(
    COURSE_ID INT,
    SEM INT,
    ISBN INT,
    PRIMARY KEY(COURSE_ID, ISBN),
    FOREIGN KEY(COURSE_ID) REFERENCES COURSE(COURSE_ID),
    FOREIGN KEY(ISBN) REFERENCES TEXT(ISBN)
    );

CREATE TABLE ENROLL(
    REGNO VARCHAR(20),
```

```
    COURSE_ID INT,
    SEM INT,
    MARKS INT,
    PRIMARY KEY(REGNO, COURSE_ID),
    FOREIGN KEY(REGNO) REFERENCES STUDENT(REGNO),
    FOREIGN KEY(COURSE_ID) REFERENCES COURSE(COURSE_ID)
    );

INSERT INTO STUDENT VALUES('REGNO', 'NAME', 'MAJOR', 'DATE');
INSERT INTO COURSE VALUES(COURSE_ID, 'CNAME', 'DEPT');
INSERT INTO TEXT VALUES(ISBN, 'BOOK_TITLE', 'PUBLISHER', 'AUTHOR');
INSERT INTO ADOPTION VALUES(COURSE_ID, SEM, ISBN);
INSERT INTO ENROLL VALUES('REGNO', COURSE_ID, SEM, MARKS);

SELECT * FROM STUDENT;
SELECT * FROM COURSE;
SELECT * FROM TEXT;
SELECT * FROM ADOPTION;
SELECT * FROM ENROLL;

INSERT INTO STUDENT (REGNO, NAME, MAJOR, DATE) VALUES
    ('1PE11CS002','SAM','SR','19930924'),
    ('1PE11CS003','CAMERON','SR','19931127'),
    ('1PE11CS004','DANNY','SR','19930413'),
    ('1PE11CS005','ETHER','JR','19940824');
```

			REGNO	NAME	MAJOR	BDATE
Edit	Copy	Delete	&REGNO	&NAME	&MAJO	0000-00-00
Edit	Copy	Delete	1PE11CS002	SAM	SR	1993-09-24
Edit	Copy	Delete	1PE11CS003	CAMERON	SR	1993-11-27
Edit	Copy	Delete	1PE11CS004	DANNY	SR	1993-04-13
Edit	Copy	Delete	1PE11CS005	ETHER	JR	1994-08-24

INSERT INTO COURSE (COURSE_ID, CNAME, DEPT) VALUES

(111,'OS','CSE'), (112,'EC','CSE'), (113,'SS','ISE'),

(114,'DBMS','CSE'), (115,'SIGNALS','ECE');

	COURSE_ID	CNAME	DEPT
Edit Copy Delete	111	OS	CSE
Edit Copy Delete	112	EC	CSE
Edit Copy Delete	113	SS	ISE
Edit Copy Delete	114	DBMS	CSE
Edit Copy Delete	115	SIGNALS	ECE

INSERT INTO TEXT (ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES

(10,'DATABASE SYSTEMS','PEARSON','SCHIELD'),

(900,'OPERATING SYS','PEARSON','LELAND'),

(901,'CIRCUITS','HALL INDIA','BOB'),

(902,'SYSTEM SOFTWARE','PETERSON','JACOB'),

(903,'SCHEDULING','PEARSON','PATIL'),

(904,'DATABASE SYSTEMS','PEARSON','JACOB'),

(905,'DATABASE MANAGER','PEARSON','BOB'),

(906,'SIGNALS','HALL INDIA','SUMIT');

	ISBN	BOOK_TITLE	PUBLISHER	AUTHOR
Edit Copy Delete	10	DATABASE SYSTEMS	PEARSON	SCHIELD
Edit Copy Delete	900	OPERATING SYS	PEARSON	LELAND
Edit Copy Delete	901	CIRCUITS	HALL INDIA	BOB
Edit Copy Delete	902	SYSTEM SOFTWARE	PETERSON	JACOB
Edit Copy Delete	903	SCHEDULING	PEARSON	PATIL
Edit Copy Delete	904	DATABASE SYSTEMS	PEARSON	JACOB
Edit Copy Delete	905	DATABASE MANAGER	PEARSON	BOB
Edit Copy Delete	906	SIGNALS	HALL INDIA	SUMIT

INSERT INTO ENROLL (REGNO, COURSE_ID, SEM, MARKS) VALUES

('1PE11CS002',114,5,100), ('1PE11CS003',113,5,100),

('1PE11CS004',111,5,100), ('1PE11CS005',112,3,100);

	REGNO	COURSE_ID	SEM	MARKS
Edit Copy Delete	1PE11CS002	114	5	100
Edit Copy Delete	1PE11CS003	113	5	100
Edit Copy Delete	1PE11CS004	111	5	100
Edit Copy Delete	1PE11CS005	112	3	100

INSERT INTO ADOPTION (COURSE_ID, SEM, ISBN) VALUES

(111,5,900), (111,5,903), (111,5,904), (112,3,901),

(113,3,10), (114,5,905), (113,5,902), (115,3,906);

	COURSE_ID	SEM	ISBN
Edit Copy Delete	111	5	900
Edit Copy Delete	111	5	903
Edit Copy Delete	111	5	904
Edit Copy Delete	112	3	901
Edit Copy Delete	113	3	10
Edit Copy Delete	113	5	902
Edit Copy Delete	114	5	905
Edit Copy Delete	115	3	906

INSERT INTO TEXT VALUES('4242', 'PYTHON PROG', 'PHI', 'DAVE R');

INSERT INTO ADOPTION VALUES(40, 4, 4242); #1

	ISBN	BOOK_TITLE	PUBLISHER	AUTHOR
Edit Copy Delete	10	DATABASE SYSTEMS	PEARSON	SCHIELD
Edit Copy Delete	900	OPERATING SYS	PEARSON	LELAND
Edit Copy Delete	901	CIRCUITS	HALL INDIA	BOB
Edit Copy Delete	902	SYSTEM SOFTWARE	PETERSON	JACOB
Edit Copy Delete	903	SCHEDULING	PEARSON	PATIL
Edit Copy Delete	904	DATABASE SYSTEMS	PEARSON	JACOB
Edit Copy Delete	905	DATABASE MANAGER	PEARSON	BOB
Edit Copy Delete	906	SIGNALS	HALL INDIA	SUMIT
Edit Copy Delete	4242	PYTHON PROG	PHI	DAVE R

SELECT C.CNAME, T.ISBN, T.BOOK_TITLE

FROM COURSE C, TEXT T

ORDER BY C.CNAME; #2

CNAME ▲ 1	ISBN	BOOK_TITLE
DBMS	901	CIRCUITS
DBMS	906	SIGNALS
DBMS	903	SCHEDULING
DBMS	900	OPERATING SYS
DBMS	905	DATABASE MANAGER
DBMS	902	SYSTEM SOFTWARE
DBMS	10	DATABASE SYSTEMS
DBMS	4242	PYTHON PROG
DBMS	904	DATABASE SYSTEMS
EC	903	SCHEDULING
EC	900	OPERATING SYS
EC	905	DATABASE MANAGER
EC	902	SYSTEM SOFTWARE
EC	10	DATABASE SYSTEMS
EC	4242	PYTHON PROG
EC	904	DATABASE SYSTEMS
EC	901	CIRCUITS
EC	906	SIGNALS
OS	900	OPERATING SYS
OS	905	DATABASE MANAGER
OS	902	SYSTEM SOFTWARE
OS	4242	PYTHON PROG
OS	10	DATABASE SYSTEMS
OS	904	DATABASE SYSTEMS

SELECT C.CNAME

FROM COURSE C, ADOPTION A, TEXT T

WHERE T.ISBN = A.ISBN AND C.COURSE_ID = A.COURSE_ID ; #3

←T→				CNAME
☐	Edit	Copy	Delete	OS
☐	Edit	Copy	Delete	OS
☐	Edit	Copy	Delete	OS
☐	Edit	Copy	Delete	EC
☐	Edit	Copy	Delete	SS
☐	Edit	Copy	Delete	SS
☐	Edit	Copy	Delete	DBMS
☐	Edit	Copy	Delete	SIGNALS

PROGRAM 9: MOVIE DATABASE

Consider the schema for Movie Database:

ACTOR(Act_id, Act_Name, Act_Gender)

DIRECTOR(Dir_id, Dir_Name, Dir_Phone)

MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)

MOVIE_CAST(Act_id, Mov_id, Role)

RATING(Mov_id, Rev_Stars)

Write SQL queries to

i. List the titles of all movies directed by 'Speed'.

ii. Find the movie names where one or more actors acted in two or more movies.

iii. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).

iv. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.

v. Update rating of all movies directed by 'Steven Spielberg' to 5

```
CREATE DATABASE MOVIE;
USE MOVIE;

CREATE TABLE ACTOR(
   ACT_ID INT,
   ACT_NAME VARCHAR(10),
   ACT_GENDER VARCHAR(10),
   PRIMARY KEY(ACT_ID)
   );

CREATE TABLE DIRECTOR(
```

```
    DIR_ID INT,
    DIR_NAME VARCHAR(10),
    DIR_PHONE INT,
    PRIMARY KEY(DIR_ID)
    );

CREATE TABLE MOVIES(
    MOV_ID INT,
    MOV_TITLE VARCHAR(10),
    MOV_YEAR VARCHAR(10),
    MOV_LANG VARCHAR(10),
    DIR_ID INT,PRIMARY KEY(MOV_ID),
    FOREIGN KEY(DIR_ID) REFERENCES DIRECTOR(DIR_ID)
    );

CREATE TABLE MOVIE_CAST(
    ACT_ID INT,MOV_ID INT,
    ROLE_PLAYED VARCHAR(10),
    FOREIGN KEY(ACT_ID) REFERENCES ACTOR(ACT_ID),
    FOREIGN KEY(MOV_ID) REFERENCES MOVIES(MOV_ID)
    );

CREATE TABLE RATING(
    MOV_ID INT,REV_STARS FLOAT,
    FOREIGN KEY(MOV_ID) REFERENCES MOVIES(MOV_ID)
    );

INSERT INTO ACTOR VALUES(2,'Reeves','MALE');
INSERT INTO ACTOR VALUES(1,'Ice T','MALE');
```

INSERT INTO ACTOR VALUES(5,'Rai','FEMALE');

INSERT INTO ACTOR VALUES(3,'Khan','MALE');

INSERT INTO ACTOR VALUES(4,'Pitt','MALE');

←T→	ACT_ID	ACT_NAME	ACT_GENDER
Edit Copy Delete	1	Ice T	MALE
Edit Copy Delete	2	Reeves	MALE
Edit Copy Delete	3	Khan	MALE
Edit Copy Delete	4	Pitt	MALE
Edit Copy Delete	5	Rai	FEMALE

SELECT * FROM ACTOR;

INSERT INTO DIRECTOR VALUES(30,'Clint',247382);

INSERT INTO DIRECTOR VALUES(20,'Steven',241829);

INSERT INTO DIRECTOR VALUES(10,'Speed',241342);

INSERT INTO DIRECTOR VALUES(50,'Kevin',241456);

INSERT INTO DIRECTOR VALUES(40,'Downey',241666);

←T→	DIR_ID	DIR_NAME	DIR_PHONE
Edit Copy Delete	10	Speed	241342
Edit Copy Delete	20	Steven	241829
Edit Copy Delete	30	Clint	247382
Edit Copy Delete	40	Downey	241666
Edit Copy Delete	50	Kevin	241456

SELECT * FROM DIRECTOR;

INSERT INTO MOVIES VALUES(44,'GUILTY',2020,'HINDI',10);

INSERT INTO MOVIES VALUES(33,'1917',1999,'ENGLISH',10);

INSERT INTO MOVIES VALUES(11,'NOTEBOOK',2018,'HINDI',20);

INSERT INTO MOVIES VALUES(22,'AVENGERS',2007,'ENGLISH',20);

INSERT INTO MOVIES VALUES(55,'WAR',2018,'HINDI',50);

←T→	MOV_ID	MOV_TITLE	MOV_YEAR	MOV_LANG	DIR_ID
Edit Copy Delete	11	NOTEBOOK	2018	HINDI	20
Edit Copy Delete	22	AVENGERS	2007	ENGLISH	20
Edit Copy Delete	33	1917	1999	ENGLISH	10
Edit Copy Delete	44	GUILTY	2020	HINDI	10
Edit Copy Delete	55	WAR	2018	HINDI	50

SELECT * FROM MOVIES;

INSERT INTO MOVIE_CAST VALUES(5,55,'VILLAIN');

INSERT INTO MOVIE_CAST VALUES(1,55,'MAIN LEAD');

INSERT INTO MOVIE_CAST VALUES(5,22,'SISTER');

INSERT INTO MOVIE_CAST VALUES(4,22,'MAIN LEAD');

INSERT INTO MOVIE_CAST VALUES(4,55,'BROTHER');

INSERT INTO MOVIE_CAST VALUES(2,44,'VILLAIN');

ACT_ID	MOV_ID	ROLE_PLAYED
5	55	VILLAIN
1	55	MAIN LEAD
5	22	SISTER
4	22	MAIN LEAD
4	55	BROTHER
2	44	VILLAIN

SELECT * FROM MOVIE_CAST;

INSERT INTO RATING VALUES(11,4);

INSERT INTO RATING VALUES(55,3.5);

INSERT INTO RATING VALUES(22,2.5);

INSERT INTO RATING VALUES(11,3.5);

INSERT INTO RATING VALUES(11,4);

MOV_ID	REV_STARS
11	4
55	3.5
22	2.5
11	3.5
11	4

SELECT * FROM RATING;

SELECT D.DIR_NAME,M.MOV_TITLE FROM DIRECTOR D,MOVIES M WHERE D.DIR_ID=M.DIR_ID AND D.DIR_NAME='Speed'; #1

Showing rows 0 - 1 (2 total, Query took 0.0843 seconds.)

```
SELECT D.DIR_NAME,M.MOV_TITLE FROM DIRECTOR D,MOVIES M WHERE D.DIR_ID=M.DIR_ID AND D.DIR_NAME='Speed'
```

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

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

+ Options

DIR_NAME	MOV_TITLE
Speed	1917
Speed	GUILTY

SELECT M.MOV_TITLE FROM MOVIES M,MOVIE_CAST MC WHERE M.MOV_ID=MC.MOV_ID AND MC.ACT_ID IN(SELECT ACT_ID FROM MOVIE_CAST GROUP BY ACT_ID HAVING COUNT(*)>1) GROUP BY M.MOV_TITLE HAVING COUNT(*)>1; #2

Showing rows 0 - 1 (2 total, Query took 0.0753 seconds.)

```
SELECT M.MOV_TITLE FROM MOVIES M,MOVIE_CAST MC WHERE M.MOV_ID=MC.MOV_ID AND MC.ACT_ID IN(SELECT ACT_ID FROM MOVIE_CAST GROUP BY ACT_ID HAVING
COUNT(*)>1) GROUP BY M.MOV_TITLE HAVING COUNT(*)>1
```

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

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

+ Options

←T→	MOV_TITLE
Edit Copy Delete	AVENGERS
Edit Copy Delete	WAR

SELECT ACT_NAME FROM (MOVIES NATURAL JOIN MOVIE_CAST) NATURAL JOIN ACTOR WHERE MOV_YEAR>2015 OR MOV_YEAR<2000; #3

Showing rows 0 - 3 (4 total, Query took 0.0920 seconds.)

```
SELECT ACT_NAME FROM (MOVIES NATURAL JOIN MOVIE_CAST) NATURAL JOIN ACTOR WHERE MOV_YEAR>2015 OR MOV_YEAR<2000
```

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

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

+ Options

ACT_NAME
Rai
Ice T
Pitt
Reeves

SELECT M.MOV_TITLE,MAX(R.REV_STARS) AS MAX_RATING FROM MOVIES M,RATING R WHERE M.MOV_ID=R.MOV_ID GROUP BY R.MOV_ID ORDER BY M.MOV_TITLE; #4

Showing rows 0 - 2 (3 total, Query took 0.0796 seconds.) [MOV_TITLE: **AVENGERS... - WAR...**]

```
SELECT M.MOV_TITLE,MAX(R.REV_STARS) AS MAX_RATING FROM MOVIES M,RATING R WHERE M.MOV_ID=R.MOV_ID GROUP BY R.MOV_ID ORDER BY M.MOV_TITLE
```

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

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

+ Options

←T→	MOV_TITLE	MAX_RATING
Edit Copy Delete	AVENGERS	2.5
Edit Copy Delete	NOTEBOOK	4
Edit Copy Delete	WAR	3.5

UPDATE RATING SET REV_STARS=5 WHERE MOV_ID IN (SELECT M.MOV_ID FROM MOVIES M,DIRECTOR D WHERE M.DIR_ID=D.DIR_ID AND D.DIR_NAME='Steven');

SELECT * FROM RATING; #5

MOV_ID	REV_STARS
11	5
55	3.5
22	5
11	5
11	5

PROGRAM 10: COLLEGE DATABASE

Consider the schema for College Database:

STUDENT(USN, SName, Address, Phone, Gender)

SEMSEC(SSID, Sem, Sec) CLASS(USN, SSID)

SUBJECT(Subcode, Title, Sem, Credits)

MARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)

Write SQL queries to

i. List all the student details studying in fourth semester 'C' section.

ii. Compute the total number of male and female students in each semester and in each section.

iii. Create a view of Test1 marks of student USN '22' in all subjects.

iv. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.

v. Categorize students based on the following criterion: If FinalIA = 17 to 20 then CAT = 'Outstanding' If FinalIA = 12 to 16 then CAT = 'Average' If FinalIA< 12 then CAT = 'Weak' Give these details only for 8th semester A, B, and C section students.

```
CREATE DATABASE COLLEGE;
USE COLLEGE;

CREATE TABLE STUDENT(
   USN INT,S_NAME VARCHAR(10),
   ADDRESS VARCHAR(20),
   PHONE INT,
   GENDER VARCHAR(10),
   PRIMARY KEY(USN)
   );
```

```
CREATE TABLE SEM_SEC(
   SSID INT,
   SEM INT,
   SEC VARCHAR(5),
   PRIMARY KEY(SSID)
   );

CREATE TABLE CLASS(
   USN INT,
   SSID INT,
   FOREIGN KEY(USN) REFERENCES STUDENT(USN),
   FOREIGN KEY(SSID) REFERENCES SEM_SEC(SSID)
   );

CREATE TABLE SUBJECTS(
   SUBCODE INT,
   TITLE VARCHAR(20),
   SEM INT,
   CREDITS INT,
   PRIMARY KEY(SUBCODE)
   );

CREATE TABLE MARKS(
   USN INT,
   SUBCODE INT,
   SSID INT,
   TEST1 INT,
   TEST2 INT,
```

```
TEST3 INT,
FOREIGN KEY(USN) REFERENCES STUDENT(USN),
FOREIGN KEY(SSID) REFERENCES SEM_SEC(SSID),
FOREIGN KEY(SUBCODE) REFERENCES SUBJECTS(SUBCODE)
);
```

```
INSERT INTO STUDENT VALUES(44,'DEVANG','RR NAGAR',231213,'MALE');
INSERT INTO STUDENT VALUES(22,'AYMAN','INDIRANAGAR',231232,'MALE');
INSERT INTO STUDENT VALUES(11,'SAM','INDIRANAGAR',231432,'FEMALE');
INSERT INTO STUDENT VALUES(55,'SHAAN','MG ROAD',212543,'MALE');
INSERT INTO STUDENT VALUES(33,'TANYA','WHITE FIELD',231443,'FEMALE');
```

	USN	S_NAME	ADDRESS	PHONE	GENDER
Edit Copy Delete	11	SAM	INDIRANAGAR	231432	FEMALE
Edit Copy Delete	22	AYMAN	INDIRANAGAR	231232	MALE
Edit Copy Delete	33	TANYA	WHITE FIELD	231443	FEMALE
Edit Copy Delete	44	DEVANG	RR NAGAR	231213	MALE
Edit Copy Delete	55	SHAAN	MG ROAD	212543	MALE

```
SELECT * FROM STUDENT;
```

```
INSERT INTO SEM_SEC VALUES(2,6,'B');
INSERT INTO SEM_SEC VALUES(4,4,'C');
INSERT INTO SEM_SEC VALUES(5,4,'B');
INSERT INTO SEM_SEC VALUES(3,4,'A');
INSERT INTO SEM_SEC VALUES(1,2,'B');
```

	SSID	SEM	SEC
Edit Copy Delete	1	2	B
Edit Copy Delete	2	6	B
Edit Copy Delete	3	4	A
Edit Copy Delete	4	4	C
Edit Copy Delete	5	4	B

```
SELECT * FROM SEM_SEC;
```

INSERT INTO CLASS VALUES(33,5);

INSERT INTO CLASS VALUES(11,4);

INSERT INTO CLASS VALUES(55,2);

INSERT INTO CLASS VALUES(22,4);

INSERT INTO CLASS VALUES(44,4);

USN	SSID
33	5
11	4
55	2
22	4
44	4

SELECT * FROM CLASS;

INSERT INTO SUBJECTS VALUES(20,'DBMS',2,4);

INSERT INTO SUBJECTS VALUES(10,'MP',4,4);

INSERT INTO SUBJECTS VALUES(40,'ADA',1,4);

INSERT INTO SUBJECTS VALUES(30,'LD',5,3);

INSERT INTO SUBJECTS VALUES(50,'COA',3,3);

←T→	SUBCODE	TITLE	SEM	CREDITS
Edit Copy Delete	10	MP	4	4
Edit Copy Delete	20	DBMS	2	4
Edit Copy Delete	30	LD	5	3
Edit Copy Delete	40	ADA	1	4
Edit Copy Delete	50	COA	3	3

SELECT * FROM SUBJECTS;

INSERT INTO MARKS VALUES(33,10,5,19,19,20);

INSERT INTO MARKS VALUES(22,50,4,16,15,12);

INSERT INTO MARKS VALUES(55,30,2,19,19,19);

INSERT INTO MARKS VALUES(22,40,4,12,18,16);

INSERT INTO MARKS VALUES(44,10,4,10,12,11);

INSERT INTO MARKS VALUES(11,20,4,15,14,13);

USN	SUBCODE	SSID	TEST1	TEST2	TEST3
33	10	5	19	19	20
22	50	4	16	15	12
55	30	2	19	19	19
22	40	4	12	18	16
44	10	4	10	12	11
11	20	4	15	14	13

SELECT * FROM MARKS;

SELECT * FROM STUDENT S WHERE S.USN IN (SELECT C.USN FROM CLASS C,SEM_SEC S WHERE S.SSID=C.SSID AND S.SEM=4 AND S.SEC='C'); #1

←T→	USN	S_NAME	ADDRESS	PHONE	GENDER
Edit Copy Delete	11	SAM	INDIRANAGAR	231432	FEMALE
Edit Copy Delete	22	AYMAN	INDIRANAGAR	231232	MALE
Edit Copy Delete	44	DEVANG	RR NAGAR	231213	MALE

SELECT S.GENDER,SS.SEM,SS.SEC,COUNT(*) FROM STUDENT S,SEM_SEC SS,CLASS C WHERE C.USN=S.USN AND C.SSID=SS.SSID GROUP BY SS.SSID; #2

GENDER	SEM	SEC	COUNT(*)
MALE	6	B	1
FEMALE	4	C	3
FEMALE	4	B	1

CREATE VIEW USN_22(USN,SUB,MARKS) AS SELECT M.USN,S.TITLE,M.TEST1 FROM MARKS M,SUBJECTS S WHERE M.SUBCODE=S.SUBCODE AND M.USN=22;

SELECT * FROM USN_22; #3

USN	SUB	MARKS
22	COA	16
22	ADA	12

ALTER TABLE MARKS ADD COLUMN FINAL_ALL FLOAT;

UPDATE MARKS SET FINAL_ALL=((TEST1+TEST2+TEST3)-LEAST(TEST1,TEST2,TEST3))/2;

SELECT * FROM MARKS; #4

USN	SUBCODE	SSID	TEST1	TEST2	TEST3	FINAL_ALL
33	10	5	19	19	20	19.5
22	50	4	16	15	12	15.5
55	30	2	19	19	19	19
22	40	4	12	18	16	17
44	10	4	10	12	11	11.5
11	20	4	15	14	13	14.5

```
ALTER TABLE MARKS ADD COLUMN CATEGORY VARCHAR(20);

UPDATE MARKS SET CATEGORY=

                    CASE

                        WHEN FINAL_ALL>=17 AND FINAL_ALL<=20 THEN
'OUTSTANDING'

        WHEN FINAL_ALL>=12 AND FINAL_ALL<17 THEN 'AVERAGE'

        WHEN FINAL_ALL<12 THEN 'WEAK'

                    END;
```

SELECT * FROM MARKS; #5

USN	SUBCODE	SSID	TEST1	TEST2	TEST3	FINAL_ALL	CATEGORY
33	10	5	19	19	20	19.5	OUTSTANDING
22	50	4	16	15	12	15.5	AVERAGE
55	30	2	19	19	19	19	OUTSTANDING
22	40	4	12	18	16	17	OUTSTANDING
44	10	4	10	12	11	11.5	WEAK
11	20	4	15	14	13	14.5	AVERAGE