

DBMS
LAB ASSIGNMENT
(2014)

SUBMITTED BY
SARAH MASUD
12CSS-57

Q1. Consider the following three tables.

EMPLOYEE (empno, name, deptno,job,hiredate, sal , comission, dob, city , phone)

DEPARTMENT (deptno, dname, manager,loc)

SALARY (eno, basic,HR,DA,tax).

Write equivalent SQL for the following query. (Use foreign key to join the tables.)

```
create table DEPARTMENT57(  
    deptno varchar(20) PRIMARY KEY,  
    dname varchar(20),  
    manager varchar(20)  
);
```

```
create table EMPLOYEE57(  
    empno varchar(20) PRIMARY KEY,  
    name varchar(20),  
    deptno varchar(20),  
    job varchar(20),  
    hiredate DATE,  
    sal int,  
    comission int,  
    dob DATE,  
    city varchar(20),  
    phone int,  
    FOREIGN KEY(deptno) references DEPARTMENT57(deptno)  
    ON UPDATE CASCADE ON DELETE CASCADE  
);
```

```
create table SALARY57(  
    eno varchar(20),  
    basic int,  
    HR int,  
    DA int,  
    tax int,  
    FOREIGN KEY(eno) references EMPLOYEE57(empno)  
    ON UPDATE CASCADE ON DELETE CASCADE  
);
```

```
insert into DEPARTMENT57 values('d3','accounting','Sohrab');
```

```
insert into EMPLOYEE57 values('e5','Umar','d3','clerk','2011-02-10',3000,3000,'1993-09-21','Delhi',888777);
```

```
insert into SALARY57 values('e5',200,300,100,50);
```

```
insert into DEPARTMENT57 values('d1','engin','Sarah'),('d2','Polytechnic','Sarib');
```

```
insert into EMPLOYEE57 values('e1','Sullu','d1','teacher','2010-02-12',20000,3000,'1993-02-16','Delhi',888888),('e2','Bushra','d2','teacher','2010-05-12',25000,3500,'1993-02-08','Delhi',666666),('e3','Sarah Khan','d1','Assistant','2009-01-10',50000,7000,'1990-07-02','Delhi',000000);
```

```
insert into SALARY57 values('e1',2000,3000,1000,500),('e2',5000,2000,1000,800),('e3',10000,2000,4000,900);
```

```
insert into DEPARTMENT57 values('d3','accounting','Sohrab');
```

```
insert into EMPLOYEE57 values('e4','Zeya','d3','clerk','2011-02-10',15000,3000,'1993-09-21','Delhi',888777);
```

```
insert into SALARY57 values('e4',200,300,100,50);
```

```
insert into DEPARTMENT57 values('d4','writer','Ashar');
```

```
insert into EMPLOYEE57 values('e6','Latika','d4','Editor','2011-02-10',3000,3000,'1993-09-
```

```

21','Paris',888777);
insert into SALARY57 values('e6',200,300,100,50);
insert into EMPLOYEE57 values('e8','Barun','d2','manager','1981-07-13',3000,3000,'1993-09-21','Paris',888777);
insert into EMPLOYEE57 values('e7','Dany','d4','Goalkeeper','1981-05-30',3000,3000,'1993-09-21','Paris',888777);
insert into EMPLOYEE57 values('e9','Nussy','d2','Saleman','1981-05-30',3000,3000,'1993-09-21','Chicago',888777);
insert into EMPLOYEE57 values('e10','Saif','d3','Saleman','1999-09-24',3000,3000,'1993-09-21','Chicago',888777);

insert into DEPARTMENT57 values('d5','Sales','Nikhat');
insert into DEPARTMENT57 values('d6','Booking','Nusrat');
insert into DEPARTMENT57 values('d7','Logistics','Masud');

```

```

mysql> desc DEPARTMENT57;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| deptno | varchar(20)   | NO   | PRI | NULL    |       |
| dname  | varchar(20)   | YES  |     | NULL    |       |
| manager | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> desc EMPLOYEE57;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| empno | varchar(20)   | NO   | PRI | NULL    |       |
| name  | varchar(20)   | YES  |     | NULL    |       |
| deptno | varchar(20)   | YES  | MUL | NULL    |       |
| job    | varchar(20)   | YES  |     | NULL    |       |
| hiredate | date         | YES  |     | NULL    |       |
| sal    | int(11)       | YES  |     | NULL    |       |
| comission | int(11)      | YES  |     | NULL    |       |
| dob    | date         | YES  |     | NULL    |       |
| city   | varchar(20)   | YES  |     | NULL    |       |
| phone  | int(11)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> desc SALARY57;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| eno    | varchar(20)   | YES  | MUL | NULL    |       |
| basic  | int(11)       | YES  |     | NULL    |       |
| HR     | int(11)       | YES  |     | NULL    |       |
| DA     | int(11)       | YES  |     | NULL    |       |
| tax    | int(11)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

1) Get the name and city of the employee working for the accounting de partment?

select name,city from EMPLOYEE57,DEPARTMENT57 where

EMPLOYEE57.deptno=DEPARTMENT57.deptno AND DEPARTMENT57.dname='accounting';

```
mysql> select name,city from EMPLOYEE57,DEPARTMENT57 where EMPLOYEE57.deptno=DEPARTMENT57.deptno AND DEPARTMENT57.dname='accounting'
```

name	city
Zeya	Delhi

```
1 row in set (0.00 sec)
```

2) Get the name, department name of all the employees whose pay is gre ater than 10000.

**select EMPLOYEE57.name,DEPARTMENT57.dname from EMPLOYEE57,DEPARTMENT57
where EMPLOYEE57.sal>10000 AND EMPLOYEE57.deptno=DEPARTMENT57.deptno;**

```
mysql> select EMPLOYEE57.name,DEPARTMENT57.dname from EMPLOYEE57,DEPARTMENT57 where EMPLOYEE57.sal>10000 AND  
-> EMPLOYEE57.deptno=DEPARTMENT57.deptno;
```

name	dname
Sullu	engin
Bushra	Polytechnic
Sarah Khan	engin
Zeya	accounting

```
4 rows in set (0.00 sec)
```

3) Get the name of the employee in ascending and descending order.

**select name from EMPLOYEE57 order by name;
select name from EMPLOYEE57 order by name DESC;**

```
mysql> select name from EMPLOYEE57 order by name;
```

name
Bushra
Sarah Khan
Sullu
Zeya

```
4 rows in set (0.00 sec)
```

```
mysql> select name from EMPLOYEE57 order by name DESC;
```

name
Zeya
Sullu
Sarah Khan
Bushra

```
4 rows in set (0.00 sec)
```

Update the city of the employee no. 2 from Mumbai to Delhi.

Update EMPLOYEE57 set city='Delhi' where empno=2;

```
mysql> select * from EMPLOYEE57;
```

empno	name	deptno	job	hiredate	sal	comission	dob	city	phone
e1	Sullu	d1	teacher	2010-02-12	20000	3000	1993-02-16	Delhi	888888
e2	Bushra	d2	teacher	2010-05-12	25000	3500	1993-02-08	Delhi	666666
e3	Sarah Khan	d1	Assistant	2009-01-10	50000	7000	1990-07-02	Delhi	0
e4	Zeya	d3	clerk	2011-02-10	15000	3000	1993-09-21	Delhi	888777

```
4 rows in set (0.00 sec)
```

5) Get the sum of the basic salary of the employees belongs to Delhi city.

```
select sum(sal) from EMPLOYEE where sal>10000 AND city='Delhi';
```

```
mysql> select * from EMPLOYEE57;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | name      | deptno | job       | hiredate | sal    | comission | dob       | city  | phone |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| e1    | Sullu    | d1     | teacher   | 2010-02-12 | 20000 | 3000    | 1993-02-16 | Delhi | 888888 |
| e2    | Bushra   | d2     | teacher   | 2010-05-12 | 25000 | 3500    | 1993-02-08 | Delhi | 666666 |
| e3    | Sarah Khan | d1     | Assistant | 2009-01-10 | 50000 | 7000    | 1990-07-02 | Delhi | 0      |
| e4    | Zeya     | d3     | clerk     | 2011-02-10 | 15000 | 3000    | 1993-09-21 | Delhi | 888777 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select sum(sal) from EMPLOYEE57 where sal>10000 AND city='Delhi';
+-----+
| sum(sal) |
+-----+
| 110000   |
+-----+
1 row in set (0.00 sec)
```

6) Get the details of the highest income tax payee.

```
select * from EMPLOYEE57,SALARY57 where SALARY57.tax=(select MAX(SALARY57.tax)
from SALARY57) AND
EMPLOYEE57.empno=SALARY57.eno;
```

```
mysql> select * from EMPLOYEE57,SALARY57 where SALARY57.tax=(select MAX(SALARY57.tax) from SALARY57) AND
-> EMPLOYEE57.empno=SALARY57.eno;
```

empno	name	deptno	job	hiredate	sal	comission	dob	city	phone	eno	basic	HR	DA	tax
e3	Sarah Khan	d1	Assistant	2009-01-10	50000	7000	1990-07-02	Delhi	0	e3	10000	2000	4000	900

```
1 row in set (0.00 sec)
```

7) Which employee is the senior most?

```
select name from EMPLOYEE57 where dob=(select min(dob) from EMPLOYEE57);
```

```
mysql> select name,dob from EMPLOYEE57;
+-----+-----+
| name   | dob       |
+-----+-----+
| Sullu   | 1993-02-16 |
| Bushra  | 1993-02-08 |
| Sarah Khan | 1990-07-02 |
| Zeya    | 1993-09-21 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select name from EMPLOYEE57 where dob=(select min(dob) from EMPLOYEE57);
+-----+
| name   |
+-----+
| Sarah Khan |
+-----+
1 row in set (0.00 sec)
```

8) Give the details of second highest salary employee (without use of < operator).

```
select * from EMPLOYEE57 where sal=(select max(sal) from EMPLOYEE57 where sal NOT IN
(select max(sal) from EMPLOYEE57));
```

```
mysql> select * from EMPLOYEE57 where sal=(select max(sal) from EMPLOYEE57 where sal NOT IN (select max(sal) from EMPLOYEE57));
```

empno	name	deptno	job	hiredate	sal	comission	dob	city	phone
e2	Bushra	d2	teacher	2010-05-12	25000	3500	1993-02-08	Delhi	666666

1 row in set (0.00 sec)

9) Give the details of second highest salary employee (without use of max and limit operator).

```
SELECT * FROM employee57 AS e WHERE 2 = (
SELECT COUNT(DISTINCT sal) FROM employee57 WHERE e.sal <= sal
);
```

```
mysql> SELECT * FROM EMPLOYEE57 AS e WHERE 2 = ( SELECT COUNT(DISTINCT sal) FROM EMPLOYEE57 WHERE e.sal <= sal );
```

empno	name	deptno	job	hiredate	sal	comission	dob	city	phone
e2	Bushra	d2	teacher	2010-05-12	25000	3500	1993-02-08	Delhi	666666

1 row in set (0.00 sec)

10) Give the details of second highest salary employee (with the use of MINUS operator).

select distinct (sal) from EMPLOYEE57 order by sal desc limit 1,1;

```
mysql> select * from EMPLOYEE57 order by sal desc limit 1,1;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | name  | deptno | job      | hiredate | sal   | comission | dob       | city  | phone |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| e2    | Bushra | d2     | teacher  | 2010-05-12 | 25000 | 3500    | 1993-02-08 | Delhi | 666666 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

11) Give the details of all employees of 5th highest salary (or nth highest salary).

select * from EMPLOYEE57 order by sal DESC limit 4,1;

```
mysql> select * from EMPLOYEE57 order by sal DESC limit 4,1;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | name  | deptno | job      | hiredate | sal   | comission | dob       | city  | phone |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| e5    | Umar  | d3     | clerk    | 2011-02-10 | 3000  | 3000    | 1993-09-21 | Delhi | 888777 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

12) How many clerks are there in the company?

select count(job) from EMPLOYEE57 where job='clerk';

```
mysql> select * from EMPLOYEE57;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | name      | deptno | job        | hiredate | sal   | comission | dob       | city  | phone |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| e1    | Sullu     | d1     | teacher    | 2010-02-12 | 20000 | 3000    | 1993-02-16 | Delhi | 888888 |
| e2    | Bushra    | d2     | teacher    | 2010-05-12 | 25000 | 3500    | 1993-02-08 | Delhi | 666666 |
| e3    | Sarah Khan | d1     | Assistant  | 2009-01-10 | 50000 | 7000    | 1990-07-02 | Delhi | 0      |
| e4    | Zeya      | d3     | clerk      | 2011-02-10 | 15000 | 3000    | 1993-09-21 | Delhi | 888777 |
| e5    | Umar      | d3     | clerk      | 2011-02-10 | 3000  | 3000    | 1993-09-21 | Delhi | 888777 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select count(job) from EMPLOYEE57 where job='clerk';
+-----+
| count(job) |
+-----+
| 2          |
+-----+
1 row in set (0.00 sec)
```

15) How many employees are there in each department?

select dname,count(empno) from EMPLOYEE57,DEPARTMENT57 group by deptno where EMPLOYEE57.deptno=DEPARTMENT57.deptno;

```
mysql> select * from DEPARTMENT57;
+-----+-----+
| deptno | dname      | manager |
+-----+-----+
| d1      | engin      | Sarah   |
| d2      | Polytechnic | Sarib   |
| d3      | accounting | Sohrab   |
+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select * from EMPLOYEE57;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | name      | deptno | job      | hiredate | sal    | comission | dob      | city  | phone |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| e1     | Sullu     | d1      | teacher  | 2010-02-12 | 20000 | 3000     | 1993-02-16 | Delhi | 888888 |
| e2     | Bushra    | d2      | teacher  | 2010-05-12 | 25000 | 3500     | 1993-02-08 | Delhi | 666666 |
| e3     | Sarah Khan | d1      | Assistant | 2009-01-10 | 50000 | 7000     | 1990-07-02 | Delhi | 0      |
| e4     | Zeya      | d3      | clerk    | 2011-02-10 | 15000 | 3000     | 1993-09-21 | Delhi | 888777 |
| e5     | Umar      | d3      | clerk    | 2011-02-10 | 3000  | 3000     | 1993-09-21 | Delhi | 888777 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> select dname,count(empno) from EMPLOYEE57,DEPARTMENT57 where EMPLOYEE57.deptno=DEPARTMENT57.deptno group by DEPARTMENT57.deptno;
+-----+-----+
| dname      | count(empno) |
+-----+-----+
| engin      | 2            |
| Polytechnic | 1           |
| accounting  | 2            |
+-----+-----+
3 rows in set (0.00 sec)
```

16) List the lowest salary for different jobs used in a company and list them in descending order.

select job,min(sal) from EMPLOYEE57 group by job order by sal DESC;

```
mysql> select * from EMPLOYEE57;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | name      | deptno | job      | hiredate | sal    | comission | dob      | city  | phone |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| e1     | Sullu     | d1      | teacher  | 2010-02-12 | 20000 | 3000     | 1993-02-16 | Delhi | 888888 |
| e2     | Bushra    | d2      | teacher  | 2010-05-12 | 25000 | 3500     | 1993-02-08 | Delhi | 666666 |
| e3     | Sarah Khan | d1      | Assistant | 2009-01-10 | 50000 | 7000     | 1990-07-02 | Delhi | 0      |
| e4     | Zeya      | d3      | clerk    | 2011-02-10 | 15000 | 3000     | 1993-09-21 | Delhi | 888777 |
| e5     | Umar      | d3      | clerk    | 2011-02-10 | 3000  | 3000     | 1993-09-21 | Delhi | 888777 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select job,min(sal) from EMPLOYEE57 group by job order by sal DESC;
+-----+-----+
| job      | min(sal) |
+-----+-----+
| Assistant | 50000    |
| teacher   | 20000    |
| clerk     | 3000     |
+-----+-----+
3 rows in set (0.00 sec)
```

17) Which department average salary is the lowest among all? Show the deptno,average salary.

select deptno,avg(sal) from EMPLOYEE57 group by deptno order by avg(sal) limit 1;


```
mysql> select * from EMPLOYEE57;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | name      | deptno | job       | hiredate | sal   | comission | dob       | city  | phone |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| e1    | Sullu     | d1     | teacher  | 2010-02-12 | 20000 | 3000    | 1993-02-16 | Delhi | 888888 |
| e2    | Bushra    | d2     | teacher  | 2010-05-12 | 25000 | 3500    | 1993-02-08 | Delhi | 666666 |
| e3    | Sarah Khan | d1     | Assistant | 2009-01-10 | 50000 | 7000    | 1990-07-02 | Delhi | 0      |
| e4    | Zeya      | d3     | clerk    | 2011-02-10 | 15000 | 3000    | 1993-09-21 | Delhi | 888777 |
| e5    | Umar      | d3     | clerk    | 2011-02-10 | 3000  | 3000    | 1993-09-21 | Delhi | 888777 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select deptno,avg(sal) from EMPLOYEE57 group by deptno order by avg(sal) limit 1;
+-----+-----+
| deptno | avg(sal) |
+-----+-----+
| d3     | 9000.0000 |
+-----+-----+
1 row in set (0.01 sec)
```

18) List the minimum, maximum and average salary for each job.

select job,min(sal),max(sal),avg(sal) from EMPLOYEE group by job;

```
mysql> select * from EMPLOYEE57;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | name      | deptno | job       | hiredate | sal   | comission | dob       | city  | phone |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| e1    | Sullu     | d1     | teacher  | 2010-02-12 | 20000 | 3000    | 1993-02-16 | Delhi | 888888 |
| e2    | Bushra    | d2     | teacher  | 2010-05-12 | 25000 | 3500    | 1993-02-08 | Delhi | 666666 |
| e3    | Sarah Khan | d1     | Assistant | 2009-01-10 | 50000 | 7000    | 1990-07-02 | Delhi | 0      |
| e4    | Zeya      | d3     | clerk    | 2011-02-10 | 15000 | 3000    | 1993-09-21 | Delhi | 888777 |
| e5    | Umar      | d3     | clerk    | 2011-02-10 | 3000  | 3000    | 1993-09-21 | Delhi | 888777 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select job,min(sal),max(sal),avg(sal) from EMPLOYEE57 group by job;
+-----+-----+-----+-----+
| job      | min(sal) | max(sal) | avg(sal) |
+-----+-----+-----+-----+
| Assistant | 50000    | 50000    | 50000.0000 |
| clerk     | 3000     | 15000    | 9000.0000  |
| teacher   | 20000    | 25000    | 22500.0000 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

19) Compute the difference between maximum and minimum salary.

select max(sal)-min(sal) from EMPLOYEE57;

```
mysql> select max(sal) from EMPLOYEE57;
+-----+
| max(sal) |
+-----+
|      50000 |
+-----+
1 row in set (0.00 sec)

mysql> select min(sal) from EMPLOYEE57;
+-----+
| min(sal) |
+-----+
|      3000 |
+-----+
1 row in set (0.00 sec)

mysql> select max(sal)-min(sal) from EMPLOYEE57;
+-----+
| max(sal)-min(sal) |
+-----+
|           47000 |
+-----+
1 row in set (0.00 sec)
```

20) List the names of the employees whose name contains LA.

select name from EMPLOYEE57 where name like '%La%';

```
mysql> select * from EMPLOYEE57;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | name      | deptno | job       | hiredate | sal    | comission | dob       | city  | phone |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| e1    | Sullu     | d1     | teacher   | 2010-02-12 | 20000 | 3000    | 1993-02-16 | Delhi | 888888 |
| e2    | Bushra    | d2     | teacher   | 2010-05-12 | 25000 | 3500    | 1993-02-08 | Delhi | 666666 |
| e3    | Sarah Khan | d1     | Assistant | 2009-01-10 | 50000 | 7000    | 1990-07-02 | Delhi | 0      |
| e4    | Zeya      | d3     | clerk     | 2011-02-10 | 15000 | 3000    | 1993-09-21 | Delhi | 888777 |
| e5    | Umar      | d3     | clerk     | 2011-02-10 | 3000  | 3000    | 1993-09-21 | Delhi | 888777 |
| e6    | Latika    | d4     | Editor    | 2011-02-10 | 3000  | 3000    | 1993-09-21 | Paris | 888777 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select name from EMPLOYEE57 where name like '%la%';
+-----+
| name |
+-----+
| Latika |
+-----+
1 row in set (0.00 sec)
```

21) List the names of the employees whose joining date is between 2nd April,1981 and 8 th Sept,1981.

select name from EMPLOYEE57 where hiredate>'1981-04-02' and hiredate<'1981-09-08'

```
mysql> select * from EMPLOYEE57;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | name      | deptno | job       | hiredate | sal   | comission | dob       | city  | phone |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| e1    | Sullu     | d1     | teacher   | 2010-02-12 | 20000 | 3000    | 1993-02-16 | Delhi | 888888 |
| e2    | Bushra    | d2     | teacher   | 2010-05-12 | 25000 | 3500    | 1993-02-08 | Delhi | 666666 |
| e3    | Sarah Khan | d1     | Assistant | 2009-01-10 | 50000 | 7000    | 1990-07-02 | Delhi | 0      |
| e4    | Zeya      | d3     | clerk     | 2011-02-10 | 15000 | 3000    | 1993-09-21 | Delhi | 888777 |
| e5    | Umar      | d3     | clerk     | 2011-02-10 | 3000  | 3000    | 1993-09-21 | Delhi | 888777 |
| e6    | Latika    | d4     | Editor    | 2011-02-10 | 3000  | 3000    | 1993-09-21 | Paris | 888777 |
| e7    | Dany      | d4     | Goalkeeper | 1981-05-30 | 3000  | 3000    | 1993-09-21 | Paris | 888777 |
| e8    | Barun     | d2     | manager   | 1981-07-13 | 3000  | 3000    | 1993-09-21 | Paris | 888777 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> select name from EMPLOYEE57 where hiredate>'1981-04-02' and hiredate<'1981-09-08';
+-----+
| name |
+-----+
| Dany |
| Barun |
+-----+
2 rows in set (0.01 sec)
```

22) How many different job titles exist in the employee table?

select job from EMPLOYEE57 group by job;

select count(DISTINCT job) EMPLOYEE57 group by job;

```
mysql> select job from EMPLOYEE57 group by job;
+-----+
| job       |
+-----+
| Assistant |
| clerk     |
| Editor    |
| Goalkeeper |
| manager   |
| teacher   |
+-----+
6 rows in set (0.00 sec)

mysql> select count(DISTINCT job) from EMPLOYEE57;
+-----+
| count(DISTINCT job) |
+-----+
| 6 |
+-----+
1 row in set (0.00 sec)
```

23) Compute the sum of all salaries of employee working under deptno=3.

select sum(sal) from EMPLOYEE57 where deptno=d3;

```
mysql> select sum(sal) from EMPLOYEE57 where deptno='d3';
+-----+
| sum(sal) |
+-----+
|      18000 |
+-----+
1 row in set (0.00 sec)
```

24) For each salesman in the emp table retrieve the deptno and department name.

```
select EMPLOYEE57.name,DEPARTMENT57.deptno,DEPARTMENT57.dname from
EMPLOYEE57,DEPARTMENT57 where EMPLOYEE57.job='Saleman' and
EMPLOYEE57.deptno=DEPARTMENT57.deptno;
```

```
mysql> select EMPLOYEE57.name,DEPARTMENT57.deptno,DEPARTMENT57.dname from EMPLOYEE57,DEPARTMENT57 where EMPLOYEE57.job='Saleman' and EMPLOYEE57.
deptno=DEPARTMENT57.deptno;
+-----+-----+-----+
| name | deptno | dname      |
+-----+-----+-----+
| Nussy | d2     | Polytechnic |
| Saif  | d3     | accounting  |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

25) List the names of all the employees with their name of the manager .

```
select EMPLOYEE57.name,DEPARTMENT57.manager,EMPLOYEE57.deptno from
EMPLOYEE57,DEPARTMENT57 where EMPLOYEE57.deptno=DEPARTMENT57.deptno;
```

```
mysql> select EMPLOYEE57.name,DEPARTMENT57.manager,EMPLOYEE57.deptno from EMPLOYEE57,DEPARTMENT57 where EMPLOYEE57.deptno=DEPARTMENT57.deptno;
Workspace Switcher +-----+
| name      | manager | deptno |
+-----+-----+-----+
| Sullu     | Sarah   | d1     |
| Sarah Khan | Sarah   | d1     |
| Bushra    | Sarib   | d2     |
| Barun     | Sarib   | d2     |
| Nussy     | Sarib   | d2     |
| Saif      | Sohrab  | d3     |
| Zeya      | Sohrab  | d3     |
| Umar      | Sohrab  | d3     |
| Latika    | Ashar   | d4     |
| Dany      | Ashar   | d4     |
+-----+-----+-----+
10 rows in set (0.00 sec)
```

26) List all employees who are working in department located at CHICAGO.

```
select name from EMPLOYEE57 where city='Chicago';
```

```
mysql> select name from EMPLOYEE57 where city='Chicago';
+-----+
| name |
+-----+
| Saif |
| Nussy |
+-----+
2 rows in set (0.00 sec)
```

27) List all the employees who are working in same department as their managers.

```
select manager,name from (EMPLOYEE57 inner join DEPARTMENT57 on  
EMPLOYEE57.deptno=DEPARTMENT.deptno) group by manager,empno;
```

```
mysql> select manager,name from (EMPLOYEE57 inner join DEPARTMENT57 on EMPLOYEE57.deptno=DEPARTMENT57.deptno) group by manager,empno;  
+-----+  
| manager | name |  
+-----+  
| Ashar   | Latika |  
| Ashar   | Dany  |  
| Sarah   | Sullu  |  
| Sarah   | Sarah Khan |  
| Sarib   | Bushra |  
| Sarib   | Barun  |  
| Sarib   | Nussy  |  
| Sohrab  | Saif   |  
| Sohrab  | Zeya   |  
| Sohrab  | Umar   |  
+-----+  
10 rows in set (0.00 sec)
```

28) Retrieve all the employees who are working in deptno=10 and who earn salary atleast as much as any employee working in deptno=30.

```
select name from EMPLOYEE57 where deptno=1 and sal>=(select min(sal) from EMPLOYEE57  
where deptno=3);
```

```
mysql> select name from EMPLOYEE57 where deptno=1 and sal>=(select min(sal) from  
EMPLOYEE57 where deptno=3);  
Empty set (0.00 sec)
```

29) List all the department who have no employees

```
select deptno,dname from DEPARTMENT57 where deptno NOT IN(select deptno from EMPLOYEE57);
```

```
mysql> select deptno from DEPARTMENT57 where deptno NOT IN(select deptno from EMPLOYEE57);  
+-----+  
| deptno |  
+-----+  
| d5     |  
| d6     |  
| d7     |  
+-----+  
3 rows in set (0.00 sec)
```

30) Delete the department d1;

```
delete from DEPARTMENT57 where deptno='d1';
```

```
mysql> select * from EMPLOYEES7;
```

empno	name	deptno	job	hiredate	sal	comission	dob	city	phone
e1	Sullu	d1	teacher	2010-02-12	20000	3000	1993-02-16	Delhi	888888
e10	Saif	d3	Saleman	1999-09-24	3000	3000	1993-09-21	Chicago	888777
e2	Bushra	d2	teacher	2010-05-12	25000	3500	1993-02-08	Delhi	666666
e3	Sarah Khan	d1	Assistant	2009-01-10	50000	7000	1990-07-02	Delhi	0
e4	Zeya	d3	clerk	2011-02-10	15000	3000	1993-09-21	Delhi	888777
e5	Umar	d3	clerk	2011-02-10	3000	3000	1993-09-21	Delhi	888777
e6	Latika	d4	Editor	2011-02-10	3000	3000	1993-09-21	Paris	888777
e7	Dany	d4	Goalkeeper	1981-05-30	3000	3000	1993-09-21	Paris	888777
e8	Barun	d2	manager	1981-07-13	3000	3000	1993-09-21	Paris	888777
e9	Nussy	d2	Saleman	1981-05-30	3000	3000	1993-09-21	Chicago	888777

```
10 rows in set (0.00 sec)
```

```
mysql> delete from DEPARTMENT57 where deptno='d1';
```

```
Query OK, 1 row affected (0.12 sec)
```

```
mysql> select * from EMPLOYEES7;
```

empno	name	deptno	job	hiredate	sal	comission	dob	city	phone
e10	Saif	d3	Saleman	1999-09-24	3000	3000	1993-09-21	Chicago	888777
e2	Bushra	d2	teacher	2010-05-12	25000	3500	1993-02-08	Delhi	666666
e4	Zeya	d3	clerk	2011-02-10	15000	3000	1993-09-21	Delhi	888777
e5	Umar	d3	clerk	2011-02-10	3000	3000	1993-09-21	Delhi	888777
e6	Latika	d4	Editor	2011-02-10	3000	3000	1993-09-21	Paris	888777
e7	Dany	d4	Goalkeeper	1981-05-30	3000	3000	1993-09-21	Paris	888777
e8	Barun	d2	manager	1981-07-13	3000	3000	1993-09-21	Paris	888777
e9	Nussy	d2	Saleman	1981-05-30	3000	3000	1993-09-21	Chicago	888777

```
8 rows in set (0.00 sec)
```

Q2.

- i. Write a function and a stored procedure to print Hello ! How are you?.

```
delimiter /
create function helloworld()
returns varchar(30)
begin
    return 'Hello! How are you?';
end /
```

```
mysql> create function helloworld()
-> returns varchar(30)
-> begin
-> return 'Hello! How are you?';
-> end /
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> select helloworld();
+-----+
| helloworld() |
+-----+
| Hello! How are you? |
+-----+
1 row in set (0.00 sec)
```

```
delimiter /
create procedure sayhello()
begin
    select 'Hello ! How are you?';
end;
/
delimiter ;
call sayhello();
```

```
mysql> delimiter /
mysql> create procedure sayhello()
-> begin
-> select 'Hello ! How are you?';
-> end;
-> /
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> call sayhello();
+-----+
| Hello ! How are you? |
+-----+
| Hello ! How are you? |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

ii. Write a function and a stored procedure to count the number of employees in the table employee.

```
delimiter /
create function countemployee1()
returns int
begin
    declare total int;
    select count(*) from EMPLOYEE57 into total;
    return total;
end /
```

```
select countemployee1()/
```

```
create procedure count2()
begin
    select count(empno) from EMPLOYEE57;
end /
```

```
call count2()/
```

```
mysql> delimiter /
mysql> create function countemployee1()
    -> returns int
    -> begin
    -> declare total int;
    -> select count(*) from EMPLOYEE57 into total;
    -> return total;
    -> end /
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> select countemployee1()/
```

```
+-----+
| countemployee1() |
+-----+
|                8 |
+-----+
1 row in set (0.00 sec)
```

```
mysql>
```

```
mysql> create procedure count2()
    -> begin
    -> select count(empno) from EMPLOYEE57;
    -> end /
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> call count2() /
```

```
+-----+
| count(empno) |
+-----+
|                8 |
+-----+
1 row in set (0.00 sec)
```


iii. Write a function and a stored procedure to calculate the factorial of the given number.

```
create function factorial(f int)
returns int
begin
    declare res int default 1;
    myloop:loop
    if f<=0 then
        leave myloop;
    else
        set res=res*f;
        set f=f-1;
    end if;
    iterate myloop;
end loop;
return res;
end/

create procedure factorial(in f int)
begin
    declare res int default 1;
    myloop:loop
    if f<=0 then
        leave myloop;
    else
        set res=res*f;
        set f=f-1;
    end if;
    iterate myloop;
end loop;
select res;
end/
```

```
mysql> create function factorial(f int)
-> returns int
-> begin
-> declare res int default 1;
-> myloop:loop
-> if f<=0 then
-> leave myloop;
-> else
-> set res=res*f;
-> set f=f-1;
-> end if;
-> iterate myloop;
-> end loop;
-> return res;
-> end/
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> select factorial(5)
-> /
```

```
+-----+
| factorial(5) |
+-----+
|          120 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select factorial(0)/
```

```
+-----+
| factorial(0) |
+-----+
|           1 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select factorial(1)/
```

```
+-----+
| factorial(1) |
+-----+
|           1 |
+-----+
```

iv) Write a function and a stored procedure to calculate the average of three numbers.

delimiter #

```
create function average(a float,b float,c float)
returns float
begin
    declare answer float;
    set answer= ((a+b+c)/3);
    return answer;

end;
#
create procedure average(a float,b float,c float)
begin
    declare answer float;
    set answer= (a+b+c)/3;
    select answer;

end;
#
```

```
mysql> delimiter #
mysql> create function average(a float,b float,c float)
-> returns float
-> begin
-> declare answer float;
-> set answer= ((a+b+c)/3);
-> return answer;
-> end;
-> #
Query OK, 0 rows affected (0.00 sec)

mysql> select average(3.5,2,5.5)#
+-----+
| average(3.5,2,5.5) |
+-----+
| 3.6666667461395264 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> create procedure average(a float,b float,c float)
-> begin
-> declare answer float;
-> set answer= (a+b+c)/3;
-> select answer;
-> end;
-> #
Query OK, 0 rows affected (0.00 sec)

mysql> call average(3.5,2,5.5);
-> #
+-----+
| answer |
+-----+
| 3.66667 |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

- v. Write a function and stored procedure to find fibonacci series and its sum.

```
create function fiboo(n int)
returns varchar(200)
begin
    declare a int default 0;
    declare b int default 1;
    declare c int default 0;
    declare sum int default 1;
    declare result varchar(200) default '';
    set result=concat(a,' ',b);
    set n=n-2;
    while n>0 do
        set c=a+b;
        set sum=sum+c;
        set result=concat(result,' ',c);
        set n=n-1;
        set a=b;
        set b=c;
    end while;
    set result=concat(result,' ||Sum==',sum);
    return result;
end #
```

```
create procedure fibbo(in n int)
begin
    declare a int default 0;
    declare b int default 1;
    declare c int default 0;
    declare sum int default 1;
    declare result varchar(200) default '';
    set result=concat(a,' ',b);
    set n=n-2;
    while n>0 do
        set c=a+b;
        set sum=sum+c;
        set result=concat(result,' ',c);
        set n=n-1;
        set a=b;
        set b=c;
    end while;
    result=concat(result,' ||Sum==',sum);
    select result;
end #
```

```
mysql> call fibbo(7)#
+-----+
| result                                     |
+-----+
| 0 1 1 2 3 5 8 ||Sum==20 |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> call fibbo(13)#
+-----+
| result                                     |
+-----+
| 0 1 1 2 3 5 8 13 21 34 55 89 144 ||Sum==376 |
+-----+
1 row in set (0.00 sec)
```

Q3. Consider the following relations

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);

Enrolled has on record per student class pair such that the student is enrolled in the class.

```
create table Faculty57
```

```
(  
  fid int PRIMARY KEY,  
  fname varchar(20),  
  deptid int  
);
```

```
create table Class57
```

```
(  
  name varchar(20) PRIMARY KEY,  
  meets_at timestamp,  
  room varchar(20),  
  fid int,  
  FOREIGN KEY(fid) references Faculty57(fid)  
  ON UPDATE CASCADE ON DELETE CASCADE  
);
```

```
create table Student57
```

```
(  
  snum int PRIMARY KEY,  
  sname varchar(20),  
  major varchar(20),  
  level varchar(20),  
  age int  
);
```

```
create table Enrolled57
```

```
(  
  snum int,  
  cname varchar(20),  
  PRIMARY KEY(snum,cname),  
  FOREIGN KEY(snum) references Student57(snum)  
  ON UPDATE CASCADE ON DELETE CASCADE,  
  FOREIGN KEY(cname) references Class57(name)  
  ON UPDATE CASCADE ON DELETE CASCADE  
);
```

```
insert into Faculty57 values(01,'Farheen','01'),(02,'Arif',02),(03,'Sohrab',03),(04,'Yasir',01),  
(05,'I.Teach',02);
```

```
insert into Class57 values('Computer','10-00-00','R128',01),('Networking','02-30-00','R128',01),  
( 'Acting','10-00-00','BA1080',05),('Writing','12-30-00','R128',05),('History','06-30-00','R678',02);
```

```
insert into Student57 values(01,'Bushra','Computer','SR',20),(02,'Maria','Acting','JR',21),  
(03,'Sarib','Writing','SR',22),(04,'Sarah','History','JR',21),(05,'Sultana','Networking','JR',23);
```

```
insert into Enrolled57 values(01,'Computer'),(02,'Acting'),(03,'Writing'),(04,'History'),(05,'Networking'),
```

```
insert into Class57 values('Maths','10-00-00','R300',03);
```

```
insert into Enrolled57 values(01,'Maths');
```

```
mysql> select * from Faculty57;
+-----+-----+-----+
| fid | fname | deptid |
+-----+-----+-----+
| 1 | Farheen | 1 |
| 2 | Arif | 2 |
| 3 | Sohrab | 3 |
| 4 | Yasir | 1 |
| 5 | I.Teach | 2 |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from Student57;
+-----+-----+-----+-----+-----+
| snum | sname | major | level | age |
+-----+-----+-----+-----+-----+
| 1 | Bushra | Computer | SR | 20 |
| 2 | Maria | Acting | JR | 21 |
| 3 | Sarib | Writing | SR | 22 |
| 4 | Sarah | History | JR | 21 |
| 5 | Sultana | Networking | JR | 23 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from Class57;
+-----+-----+-----+-----+
| name | meets_at | room | fid |
+-----+-----+-----+-----+
| Acting | 0000-00-00 00:00:00 | BA1080 | 5 |
| Computer | 0000-00-00 00:00:00 | R128 | 1 |
| History | 0000-00-00 00:00:00 | R678 | 2 |
| Networking | 0000-00-00 00:00:00 | R128 | 1 |
| Writing | 0000-00-00 00:00:00 | R128 | 5 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from Enrolled57;
+-----+-----+
| snum | cname |
+-----+-----+
| 2 | Acting |
| 1 | Computer |
+-----+-----+
```

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

select Student57.sname from Student57,Enrolled57,Class57,Faculty57 where Student57.snum=Enrolled57.snum AND Enrolled57.cname=Class57.name and Class57.fid=Faculty57.fid and Student57.level='JR' and Faculty57.fname='I.Teach';

```
Database changed
mysql> select Student57.sname from Student57,Enrolled57,Class57,Faculty57 where Student57.snum=Enrolled57.snum AND Enrolled57.cname=Class57.name
and Class57.fid=Faculty57.fid and Student57.level='JR' and Faculty57.fname='I.Teach';
+-----+
| sname |
+-----+
| Maria |
+-----+
1 row in set (0.00 sec)
```

2. Find the age of the oldest student who is either a History major or enrolled in a course taught by I. Teach.

select max(Student57.age) from Student57,Enrolled57,Class57,Faculty57 where Student57.snum=Enrolled57.snum AND Enrolled57.cname=Class57.name and Class57.fid=Faculty57.fid and (Student57.major='History' OR Faculty57.fname='I.Teach');

```
mysql> select max(Student57.age) from Student57,Enrolled57,Class57,Faculty57 where Student57.snum=Enrolled57.snum AND Enrolled57.cname=Class57.name and Class57.fid=Faculty57.fid and (Student57.major='History' OR Faculty57.fname='I.Teach');
+-----+
| max(Student57.age) |
+-----+
|          22       |
+-----+
1 row in set (0.00 sec)
```

3. Find the names of all classes that either meet in room BA1080 or have 2 or more students enrolled.

select Class57.name from ((Enrolled57 inner join Student57 on Enrolled57.snum=Student57.snum) inner join Class57 on Class57.name=Enrolled57.cname) group by Class57.name having count(Class57.name) >=2 UNION select Class57.name from Class57 where Class57.room='BA1080';

```
mysql> select Class57.name from ((Enrolled57 inner join Student57 on Enrolled57.snum=Student57.snum) inner join Class57 on Class57.name=Enrolled57.cname) group by Class57.name having count(Class57.name) >=2 UNION select Class57.name from Class57 where Class57.room='BA1080';
+-----+
| name |
+-----+
| Acting |
+-----+
1 row in set (0.00 sec)
```

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

select Student57.sname from ((Enrolled57 inner join Student57 on Enrolled57.snum=Student57.snum) inner join Class57 on Class57.name=Enrolled57.cname) group by Class57.meets_at,Student57.snum having count(Enrolled57.cname) >= 2;

```
mysql> select * from Enrolled57 order by snum;
+-----+-----+
| snum | cname |
+-----+-----+
| 1    | Computer |
| 1    | Maths    |
| 2    | Acting   |
| 3    | Writing  |
| 4    | History  |
| 5    | Networking |
+-----+-----+
6 rows in set (0.00 sec)

mysql> select Student57.sname from ((Enrolled57 inner join Student57 on Enrolled57.snum=Student57.snum) inner join Class57 on Class57.name=Enrolled57.cname) group by Class57.meets_at,Student57.snum having count(Enrolled57.cname) >= 2;
+-----+
| sname |
+-----+
| Bushra |
+-----+
1 row in set (0.00 sec)
```

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

```
select Faculty57.fname from (Class57 inner join Faculty57 on Class57.fid=Faculty57.fid) group by Faculty57.fid having count(DISTINCT Class57.room)=(select count(DISTINCT Class57.room) from Class57);
```

```
mysql> select Faculty57.fname from (Class57 inner join Faculty57 on Class57.fid=Faculty57.fid) group by Faculty57.fid having count(DISTINCT Class57.room)=(select count(DISTINCT Class57.room) from Class57);
+-----+
| fname |
+-----+
| I.Teach |
+-----+
1 row in set (0.00 sec)
```

7. For each level, print the level and the average age of students for that level.

```
select Student57.level,avg(age) from Student57 group by Student57.level;
```

```
mysql>
mysql> select Student57.level,avg(age) from Student57 group by Student57.level;
+-----+-----+
| level | avg(age) |
+-----+-----+
| JR    | 21.6667 |
| SR    | 21.0000 |
+-----+-----+
2 rows in set (0.00 sec)
```

8. For all levels except JR, print the level and the average age of students for that level.

```
select level ,avg(age) from Student57 where level != 'JR' group by level;
```

```
mysql> select level ,avg(age) from Student57 where level != 'JR' group by level;
+-----+-----+
| level | avg(age) |
+-----+-----+
| SR    | 21.0000 |
+-----+-----+
1 row in set (0.00 sec)
```

9. For each faculty member that has taught classes only in room R128, print the faculty members name and the total number of classes she or he has taught.

select Faculty57.fname,count(Class57.name) as COUNT from (Class57 inner join Faculty57 on Class57.fid=Faculty57.fid) where Class57.room='R128' group by Class57.fid;

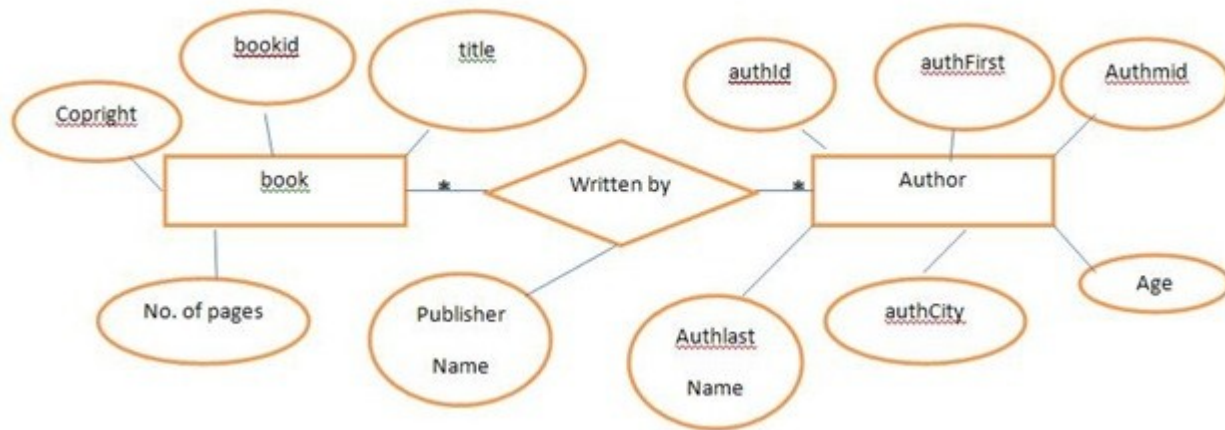
```
mysql> select Faculty57.fname,count(Class57.name) as COUNT from (Class57 inner join Faculty57 on Class57.fid=Faculty57.fid) where Class57.room='R128' group by Class57.fid;
+-----+
| fname | COUNT |
+-----+
| Farheen | 2 |
| Arif | 1 |
| Sohrab | 1 |
| I.Teach | 1 |
+-----+
4 rows in set (0.00 sec)
```

10. Find the names of students enrolled in the maximum number of classes.

SELECT sname FROM student57 AS s WHERE snum IN (
SELECT snum FROM enrolled57 GROUP BY snum
HAVING count(*) >= all (
SELECT count(*) FROM enrolled57 GROUP BY snum));

```
mysql> SELECT sname FROM Student57 WHERE snum IN ( SELECT snum FROM Enrolled57 GROUP BY snum HAVING count(*) >= all ( SELECT count(*) FROM Enrolled57 GROUP BY snum ) );
+-----+
| sname |
+-----+
| Bushra |
+-----+
1 row in set (0.00 sec)
```

Q4. Write equivalent SQL for the following query.



```
create table book57(
    bookid INT,
    title varchar(20),
    No_of_pages int,
    copyright varchar(20),
    PRIMARY KEY(bookid)
);
```

```
create table Author57(
    authid int,
    authFirst varchar(20),
    Authmid varchar(20),
    Age int,
    authCity varchar(20),
    AuthlastName varchar(20),
    PRIMARY KEY(authid)
);
```

```
create table WrittenBy57(
    Publisher_Name varchar(20),
    bookid int,
    authid int,
    PRIMARY KEY(bookid,authid),
    FOREIGN KEY(bookid) references book57(bookid)
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(authid) references Author57(authid)
    ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
insert into book57 values(1,'Introduction to Java',400,'applet'),(2,'Mathematics at stake',599,'cal'),
(3,'Reading power',453,'booked')
```

```
insert into Author57 values(1,'Jenny','Lyi',20,'London','Dave'),(2,'Sarib','Shaa',80,'Kanpur','Shahbaz'),
(3,'Jenny','Lyi',20,'London','Dave'),(4,'Arheen','Pilo',90,'Mumbai','Pillai');
```

```
insert into WrittenBy57 values ('Wiley',1,1),('digit',2,1),('Wiley',1,2),('Prinkson',3,4),('Outstanding',2,3);
```

```
mysql> select * from WrittenBy57;
```

```
+-----+-----+-----+
| Publisher_Name | bookid | authid |
+-----+-----+-----+
| Wiley          | 1      | 1      |
| Wiley          | 1      | 2      |
| digit          | 2      | 1      |
|                | 2      | 3      |
| PETERSON      | 3      | 4      |
+-----+-----+-----+
```

```
5 rows in set (0.00 sec)
```

```
mysql> select * from Author57;
```

```
+-----+-----+-----+-----+-----+-----+
| authid | authFirst | Authmid | Age | authCity | AuthlastName |
+-----+-----+-----+-----+-----+-----+
| 1      | Jenny    | Lyi     | 20  | London   | Dave         |
| 2      | Sarib    | Shaa    | 80  | Kanpur   | Shahbaz      |
| 3      | Jenny    | Lyi     | 20  | London   | Dave         |
| 4      | Arheen   | Pilo    | 90  | Mumbai   | Pillai       |
+-----+-----+-----+-----+-----+-----+
```

```
4 rows in set (0.00 sec)
```

```
mysql> select * from book57;
```

```
+-----+-----+-----+-----+
| bookid | title                | No_of_pages | copyright |
+-----+-----+-----+-----+
| 1      | Introduction to Java | 400         | applet    |
| 2      | Mathematics at stake | 599         | cal       |
| 3      | Reading power        | 453         | booked    |
+-----+-----+-----+-----+
```

```
3 rows in set (0.00 sec)
```

1. Get the title,author name,publisher name for author whose city contain total no of o=2?

select book57.title,concat(Author57.authFirst,' ',Author57.Authmid,'.',AuthlastName) as Author_Name,WrittenBy57.Publisher_Name from book57,Author57,WrittenBy57 where Author57.authid=WrittenBy57.authid and book57.bookid=WrittenBy57.bookid and Author57.authcity like '%o%o%';

```
mysql> select * from Author57;
+-----+-----+-----+-----+-----+-----+
| authid | authFirst | Authmid | Age | authCity | AuthlastName |
+-----+-----+-----+-----+-----+-----+
| 1 | Jenny | Lyi | 20 | London | Dave |
| 2 | Sarib | Shaa | 80 | Kanpur | Shahbaz |
| 4 | Arheen | Pilo | 90 | Mumbai | Pillai |
| 5 | Rommi | Trew | 34 | Delhi | Upny |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from book57;
+-----+-----+-----+-----+
| bookid | title | No_of_pages | copyright |
+-----+-----+-----+-----+
| 2 | Mathematics at stake | 599 | cal |
| 3 | Reading power | 453 | booked |
| 21 | Blue moon | 450 | points |
| 45 | Hello C | 567 | CC |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select book57.title,concat(Author57.authFirst,' ',Author57.Authmid,'.',AuthlastName) as Author_Name,WrittenBy57.Publisher_Name from book57,Author57,WrittenBy57 where Author57.authid=WrittenBy57.authid and book57.bookid=WrittenBy57.bookid and Author57.authcity like '%o%o%';
+-----+-----+-----+
| title | Author_Name | Publisher_Name |
+-----+-----+-----+
| Mathematics at stake | Jenny Lyi.Dave | digit |
| Hello C | Jenny Lyi.Dave | Wiley |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

2. Give the details of the book which is written by at least two authors.

select book57.bookid,title,No_of_pages,copyright,Publisher_Name from ((book57 inner join WrittenBy57 on book57.bookid=WrittenBy57.bookid) inner join Author57 on WrittenBy57.authid=Author57.authid) group by Author57.authid having count(*) >= 2;

```
mysql> select book57.bookid,title,No_of_pages,copyright,Publisher_Name from ((book57 inner join WrittenBy57 on book57.bookid=WrittenBy57.bookid) inner join Author57 on WrittenBy57.authid=Author57.authid) group by Author57.authid having count(*) >= 2;
+-----+-----+-----+-----+-----+
| bookid | title | No_of_pages | copyright | Publisher_Name |
+-----+-----+-----+-----+-----+
| 1 | Introduction to Java | 400 | applet | Wiley |
+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

3. Write a stored procedure (SP Name : insertIntoAuth) to insert the Author information.

delimiter #

create procedure insertIntoAuth(in aid int,in afn varchar(20),in amn varchar(20),in ag int,in ac varchar(20),in aln varchar(20))

begin

insert into Author57(authid,authFirst,Authmid,Age,authCity,AuthlastName)

values(aid,afn,amn,ag,ac,aln);

end#

delimiter ;

```
mysql> delimiter #
mysql> create procedure insertIntoAuth(in aid int,in afn varchar(20),in amn varchar(20),in ag int,in ac varchar(20),in aln varchar(20))
-> begin
-> insert into Author57(authid,authFirst,Authmid,Age,authCity,AuthlastName) values(aid,afn,amn,ag,ac,aln);
-> end#
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> call insertIntoAuth(5,'Rommi','Trew',34,'Delhi','Upny');
ERROR 1318 (42000): Incorrect number of arguments for PROCEDURE dbms_project2014.insertIntoAuth; expected 6, got 5
mysql> call insertIntoAuth(5,'Rommi','Trew',34,'Delhi','Upny');
Query OK, 1 row affected (0.08 sec)

mysql> call insertIntoAuth(6,'Zenu','Yuew',19,'Xnag','Hun');
Query OK, 1 row affected (0.06 sec)

mysql> select * from Author57;
+-----+-----+-----+-----+-----+-----+
| authid | authFirst | Authmid | Age | authCity | AuthlastName |
+-----+-----+-----+-----+-----+-----+
| 1 | Jenny | Lyi | 20 | London | Dave |
| 2 | Sarib | Shaa | 80 | Kanpur | Shahbaz |
| 3 | Jenny | Lyi | 20 | London | Dave |
| 4 | Arheen | Pilo | 90 | Mumbai | Pillai |
| 5 | Rommi | Trew | 34 | Delhi | Upny |
| 6 | Zenu | Yuew | 19 | Xnag | Hun |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

4. Write a stored procedure (SP Name : insertBookInfo) to insert the book information such as bookid, title, no. of pages, copyright, authorId, Publisher Name. (Use two stored procedure and call it from one stored procedure i.e nested SP). (SP Name : insertBook, insertWBy).

delimiter \$\$

create procedure insertWBy1(in pub varchar(20),in bid int,in aid int)

begin

insert into WrittenBy57 values(pub,bid,aid);

end\$\$

create procedure insertBook1(in bk int,in tit varchar(20),in nop int,in cpr varchar(20))

begin

insert into book57 values(bk,tit,nop,cpr);

end\$\$

create procedure insertBookInfo1(in bkid int,in title varchar(20),in page int,in cp varchar(20),in aid int,in pb varchar(20))

begin

```

        call insertBook1(bkid,title,page,cp);
        call insertWBy1(pb,bkid,aid);
end$$
delimiter ;

call insertBookInfo1(21,'Blue moon',450,'points',5,'Wely');

```

```

mysql> drop procedure insertBookInfo1;
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter $$
mysql> create procedure insertWBy(in pub varchar(20),in bid int,in aid int)
-> begin
-> insert into WrittenBy57 values(pub,bid,aid);
-> end $$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> create procedure insertBook(in bkid int,in title varchar(20),in page int,in cp varchar(20))
-> begin
-> insert into book57 values(bkid,tite,page,cp);
-> end $$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> create procedure insertBookInfo(in bkid,in title varchar(20),in page int , in cp varchar(20),in aid int,in pb varchar(20))
-> begin
-> call insertBook(bkid,title,page,cp);
-> call insertWby(pb,bkid,aid);
-> end $$
50000 1001 (10000) You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right

```

5. Write a stored procedure to delete the Author information using its AuthID. (Note: If Author book(in Book Table) exists for AuthID, then it should display message as You cant delete Author because total no book exist in BookTable. First delete all the books written by him).

```

delimiter $
CREATE PROCEDURE deleteAuthor (IN pauthid INT)
BEGIN
    DECLARE val INT DEFAULT 0;
    SELECT COUNT(*) FROM WrittenBy57 WHERE authid=pauthid INTO val;
    IF val>0 THEN
        SELECT 'You cannot delete author because books exist in the book74 table for
the given author id. First delete all the books written by him.';
    ELSE
        DELETE FROM Author57 WHERE authid = pauthid;
        SELECT 'Author deleted';
    END IF;
END $$
delimiter ;

```

```
mysql> CREATE PROCEDURE deleteAuthor (IN pauthid INT)
-> BEGIN
-> DECLARE val INT DEFAULT 0;
-> SELECT COUNT(*) FROM WrittenBy57 WHERE authid=pauthid INTO val;
-> IF val>0 THEN
-> SELECT 'You cannot delete author because books exist in the book74 table for
'> the given author id. First delete all the books written by him.';
-> ELSE
-> DELETE FROM Author57 WHERE authid = pauthid;
-> SELECT 'Author deleted';
-> END IF;
-> END $$
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> delimiter ;
```

```
mysql> select *from WrittenBy57;
```

```
+-----+-----+-----+
| Publisher_Name | bookid | authid |
+-----+-----+-----+
| Wiley          | 1      | 1      |
| Wiley          | 1      | 2      |
| digit          | 2      | 1      |
| Outstanding    | 2      | 3      |
| Prinkson       | 3      | 4      |
| Wiley          | 45     | 1      |
+-----+-----+-----+
```

6 rows in set (0.00 sec)

```
mysql> call deleteAuthor(1);
```

```
+-----+-----+-----+
| You cannot delete author because books exist in the book74 table for
the given author id. First delete all the books written by him. |
+-----+-----+-----+
| You cannot delete author because books exist in the book74 table for
the given author id. First delete all the books written by him. |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> call deleteAuthor(6);
```

6. Write a stored procedure to delete the Book using AuthID.

```
delimiter $$;
```

```
CREATE PROCEDURE deleteBook(IN iauthid INT)
```

```
BEGIN
```

```
    DECLARE bid INT;
```

```
    SELECT bookid FROM WrittenBy57 WHERE authid = iauthid INTO bid;
```

```
    DELETE FROM book57 WHERE bookid = bid;
```

```
    SELECT 'Book deleted';
```

```
END $$
```

```
delimiter ;
```

```
mysql> delimiter $
mysql> CREATE PROCEDURE deleteBook(IN iauthid INT)
  -> BEGIN
  -> DECLARE bid INT;
  -> SELECT bookid FROM WrittenBy57 WHERE authid = iauthid INTO bid;
  -> DELETE FROM book57 WHERE bookid = bid;
  -> SELECT 'Book deleted';
  -> END $$
```

Query OK, 0 rows affected (0.00 sec)

ERROR:

No query specified

```
mysql> delimiter ;
mysql> call deletebook(1);
ERROR 1172 (42000): Result consisted of more than one row
mysql> call deletebook(2);
```

```
+-----+
```

```
| Book deleted |
```

```
+-----+
```

```
| Book deleted |
```

```
+-----+
```

1 row in set (0.42 sec)

Query OK, 0 rows affected (0.42 sec)

Q5. Create function that validate the age of employee. Function accept the dob of employee and return 1 if age is lies between 18 and 60 else re turn 0

```
delimiter #
create function agevalid(age date)
returns int
begin
    declare ans int;
    select timestampdiff(year,age,now()) into ans;
    if ans>=20 AND ans<=60 then
        return 1;
    else
        return 0;
    end if;
end #
```

```
select agevalid('19940802')#
select agevalid('20001007')#
select agevalid('19240102')#
```



```
mysql> delimiter #
mysql> create function agevalid(age date)
    -> returns int
    -> begin
    -> declare ans int;
    -> select timestampdiff(year,age,now()) into ans;
    -> if ans>=20 AND ans<=60 then
    -> return 1;
    -> else
    -> return 0;
    -> end if;
    -> end #
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select agevalid('19940802')#
+-----+
| agevalid('19940802') |
+-----+
|                      1 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select agevalid('20001007')#
+-----+
| agevalid('20001007') |
+-----+
|                      0 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select agevalid('19240102')#
+-----+
| agevalid('19240102') |
+-----+
|                      0 |
+-----+
1 row in set (0.00 sec)
```

Q6. Consider a following table of a database :

Book(bid, bname,authname)

1. Create triggers which create a log of every Insert ,Delete and Upda te operation on the book table record.

It should also hold the username who was operating at that time and t ime and type of operation.

NOTE: log table attributes are user, operation, pbid, pbname, pauthname, nbid, nbname, nauthname and timeofop

```
create table Book57(
bid varchar(20),
bname varchar(20),
authname varchar(20)
);
```

```
create table log57(
    user varchar(20),
    opertion varchar(20),
    pbid varchar(20),
    pbname varchar(20),
    pauthname varchar(20),
    nbid varchar(20),
    nbname varchar(20),
    nauthname varchar(20),
    timeofop date
);
```

delimiter #

create trigger intrig AFTER INSERT on Book57 FOR EACH ROW

begin

insert into log57 value(user(),'INSERT',NULL,',',new.bid,new.bname,new.authname,now());

end#

delimiter ;

insert into Book57 values(123,'To kill a mocking bird','Harper Lee');

insert into Book57 values(124,'Room on the roof','Ruskin Bond');

insert into Book57 values(125,'Paths of Glory','Jeffery Archer');

insert into Book57 values(126,'Inferno','Dan Brown');

```
mysql> insert into Book57 values(123,'To kill a mocking bird','Harper Lee');
Query OK, 1 row affected, 1 warning (0.06 sec)
```

```
mysql> insert into Book57 values(124,'Room on the roof','Ruskin Bond');
Query OK, 1 row affected (0.10 sec)
```

```
mysql> insert into Book57 values(125,'Paths of Glory','Jeffery Archer');
Query OK, 1 row affected (0.09 sec)
```

```
mysql> insert into Book57 values(126,'Inferno','Dan Brown');
Query OK, 1 row affected (0.11 sec)
```

```
mysql> select * from log57;
```

user	opertion	pbid	pbname	pauthname	nbid	nbname	nauthname	timeofop
root@localhost	INSERT	NULL			123	To kill a mocking bi	Harper Lee	2014-11-19
root@localhost	INSERT	NULL			124	Room on the roof	Ruskin Bond	2014-11-19
root@localhost	INSERT	NULL			125	Paths of Glory	Jeffery Archer	2014-11-19
root@localhost	INSERT	NULL			126	Inferno	Dan Brown	2014-11-19

```
4 rows in set (0.00 sec)
```

```

delimiter #
create trigger uptrig AFTER UPDATE on Book57 FOR EACH ROW
begin
    insert into log57
value(user(), 'UPDATE', old.bid, old.bname, old.authrname, new.bid, new.bname, new.authrname, now());
end#
delimiter ;

update Book57 set bname='Grandfather's Zoo' where authrname='Ruskin Bond';
update Book57 set bname='Be careful what you wish for' where authrname='Jeffery Archer';

```

```

mysql> update Book57 set bname='Be careful what you wish for' where authrname='Jeffery Archer';
Query OK, 1 row affected, 1 warning (0.09 sec)
Rows matched: 1  Changed: 1  Warnings: 1

mysql> select * from log57;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| user          | operation | pbid | pbname          | pauthname | nbid | nbname          | nauthname | timeofop |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| root@localhost | INSERT   | NULL |                |           | 123 | To kill a mocking bi | Harper Lee | 2014-11-19 |
| root@localhost | INSERT   | NULL |                |           | 124 | Room on the roof    | Ruskin Bond | 2014-11-19 |
| root@localhost | INSERT   | NULL |                |           | 125 | Paths of Glory      | Jeffery Archer | 2014-11-19 |
| root@localhost | INSERT   | NULL |                |           | 126 | Inferno             | Dan Brown | 2014-11-19 |
| root@localhost | UPDATE   | 124 | Room on the roof | Ruskin Bond | 124 | Grandfather's Zoo   | Ruskin Bond | 2014-11-19 |
| root@localhost | UPDATE   | 125 | Paths of Glory   | Jeffery Archer | 125 | Be careful what you | Jeffery Archer | 2014-11-19 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

```

delimiter #
create trigger deltrig AFTER DELETE on Book57 FOR EACH ROW
begin
    insert into log57 value(user(), 'DELETE', old.bid, old.bname, old.authrname, NULL, '', '', now());
end#
delimiter ;

delete from Book57 where bid=123;
delete from Book57 where bname='Inferno';

```



```
mysql> create trigger deltrig AFTER DELETE on Book57 FOR EACH ROW
-> begin
-> insert into log57 value(user(),'DELETE',old.bid,old.bname,old.authrname,NULL,'','now());
-> end#
```

Query OK, 0 rows affected (0.18 sec)

```
mysql> delimiter ;
```

```
mysql> delete from Book57 where bid=123;
```

Query OK, 1 row affected (0.06 sec)

```
mysql> delete from Book57 where bname='Inferno';
```

Query OK, 1 row affected (0.08 sec)

```
mysql> select * from log57;
```

user	opertion	pbid	pbname	pauthname	nbid	nbname	nauthname	timeofop
root@localhost	INSERT	NULL			123	To kill a mocking bi	Harper Lee	2014-11-19
root@localhost	INSERT	NULL			124	Room on the roof	Ruskin Bond	2014-11-19
root@localhost	INSERT	NULL			125	Paths of Glory	Jeffery Archer	2014-11-19
root@localhost	INSERT	NULL			126	Inferno	Dan Brown	2014-11-19
root@localhost	UPDATE	124	Room on the roof	Ruskin Bond	124	Grandfather's Zoo	Ruskin Bond	2014-11-19
root@localhost	UPDATE	125	Paths of Glory	Jeffery Archer	125	Be careful what you	Jeffery Archer	2014-11-19
root@localhost	DELETE	123	To kill a mocking bi	Harper Lee	NULL			2014-11-19
root@localhost	DELETE	126	Inferno	Dan Brown	NULL			2014-11-19

8 rows in set (0.00 sec)

-----END-----