

## **Q 1. Consider the following three tables.**

EMPLOYEE( empno, name, deptno,job,hiredate, sal , commission, 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.)

- 1) Get the name and city of the employee working for the accounting department?
- 2) Get the name, department name of all the employees whose pay is greater than 10000.
- 3) Get the name of the employee in ascending and descending order.
- 4) Update the city of the employee no.2 from Mumbai to Delhi.
- 5) Get the sum of the basic salary of the employees belongs to Delhi city.
- 6) Get the details of the highest income tax payee.
- 7) Which employee is the senior most?
- 8) Give the details of second highest salary employee (without use of ?<? operator).
- 9) Give the details of second highest salary employee (without use of max and limit operator).
- 10) Give the details of second highest salary employee (with the use of MINUS operator).
- 11) Give the details of all employees of 5th highest salary ( or nth highest salary).
- 12) How many clerks are there in the company?
- 13) Which department has exactly one employee as clerk?
- 14) Which department has the highest number of clerks? Show the deptno and count.
- 15) How many employees are there in each department?
- 16) List the lowest salary for different jobs used in a company and list them in descending order.
- 17) Which department average salary is the lowest among all? Show the deptno,average salary.
- 18) List the minimum, maximum and average salary for each job.
- 19) Compute the difference between maximum and minimum salary.
- 20) List the names of the employees whose name contains LA.
- 21) List the names of the employees whose joining date is between 2nd April,1981 and 8th Sept,1981.
- 22) How many different job titles exist in the employee table?

- 23) Compute the sum of all salaries of employee working under deptno=30.
- 24) For each salesman in the emp table retrieve the deptno and department name.
- 25) List the names of all the employees with their name of the manager.
- 26) List all employees who are working in department located at CHICAGO.
- 27) List all the employees who are working in same department as their managers.
- 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.
- 29) List all the department who have no employees
- 30) Delete the EC department.

Create table

DEPARMENT:

```
create table department1232(  
deptno int primary key auto_increment,  
dname varchar(100),  
manager varchar(100),  
loc varchar(100)  
);
```

EMPLOYEE:

```
create table employee1232(  

```

```
empno int primary key auto_increment,  
name varchar(100),  
deptno int,  
job varchar(100),  
hiredate date,  
sal int,  
commission int,  
dob date,  
city varchar(100),  
phone varchar(100),  
foreign key(deptno) references department1232 (deptno) ON DELETE CASCADE ON UPDATE  
CASCADE  
);
```

SALARY:

```
create table salary1232(  
empno int,  
basic int,  
hr int,  
da int,  
tax int,  
foreign key(empno) references employee1232(empno) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

**Values after insertion**

```
mysql> select * from department32;
```

deptno	dname	manager	loc
10	ACCOUNTING	Henry	NEW YORK
20	RESEARCH	John	DALLAS
30	SALES	Samules	NEVADA
40	OPERATIONS	Dev	BOSTON

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

```
mysql> select * from employee32;
```

empno	name	deptno	job	hiredate	sal	commission	dob	city	phone
33	jadavyadav	10	CLERK	1981-03-15	12500	100	1988-05-14	Delhi	9822121221
34	Rohitsharna	10	CLERK	1981-02-15	1200	100	1988-05-14	Delhi	9822121221
35	Milansar	20	SALESMAN	0000-00-00	5000	100	1988-05-14	Chennai	9822121221
36	Joseph	20	CLERK	1981-06-15	10000	100	1988-05-14	Haryana	9822121221
37	Root	30	MANAGER	1981-07-15	10002	100	1988-05-14	Chandigarh	9822121221
38	Drake	30	SALESMAN	1981-08-15	8211	100	1988-05-14	CHICAGO	9822121221
39	Jane	30	CLERK	1981-09-15	9000	100	1988-05-14	CHICAGO	9822121221
40	Niki	40	ASS.MANAGER	1981-10-15	4021	100	1988-05-14	NEW YORK	9822121221
41	James	40	PRO-MANAGER	1981-11-15	98002	100	1988-05-14	NEW YORK	9822121221
42	James	40	CLERK	1981-12-20	9122	100	1988-05-14	DALLAS	9822121221
43	Anish	40	SALEMAN	1991-12-20	8123	100	1988-05-14	DALLAS	9822121221
47	Rohan	20	SALESMAN	1981-05-15	8000	100	1988-05-14	Hyderabad	9822121221
48	Sureshsingh	30	TAX	1981-04-15	12000	100	1988-05-14	Banglore	9822121221
49	faizan	10	teacher	2007-03-15	1250000	100	2009-05-14	delhi	9897356806

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

```
mysql> select * from salary32;
```

empno	basic	hr	da	tax
49	10000	2000	300	500
34	12000	3000	600	600
45	10000	4000	100	900

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

#### Queries:

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

- SELECT e.name,e.city FROM ( SELECT deptno FROM department1232 WHERE dname="ACCOUNTING" ) d, employee1232 e WHERE d.deptno=e.deptno;

```
mysql> SELECT e.name,e.city FROM ( SELECT deptno FROM department32 WHERE dname="ACCOUNTING" ) d, employee32 e
-> WHERE d.deptno=e.deptno;
```

name	city
jadavyadav	Mumbai
Rohitsharna	Delhi
faizan	delhi

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

```
mysql> _
```

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

- SELECT e.name AS Name, d.dname AS "Department" FROM employee1232 e INNER JOIN department1232 d

WHERE e.sal > 10000 AND e.deptno=d.deptno;

```
mysql> SELECT e.name AS Name, d.dname AS "Department"
-> FROM employee32 e INNER JOIN department32 d
-> WHERE e.sal > 10000 AND e.deptno=d.deptno;
+-----+-----+
| Name | Department |
+-----+-----+
| jadavyadav | ACCOUNTING |
| faizan | ACCOUNTING |
| Root | SALES |
| Sureshsingh | SALES |
| James | OPERATIONS |
| farhan | EC |
| Himanshu | EC |
+-----+-----+
```

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

- (i) SELECT name  
FROM employee1232  
ORDER BY name;

```
mysql> SELECT name
-> FROM employee32
-> ORDER BY name;
+-----+
| name |
+-----+
| Anish |
| Drake |
| faizan |
| farhan |
| Himanshu |
| jadavyadav |
| James |
| James |
| Jane |
| Joseph |
| Milansar |
| Niki |
| Rohan |
| Rohitsharma |
| Root |
| Saleh |
| Sureshsingh |
+-----+
17 rows in set (0.00 sec)

mysql>
```

- (ii) SELECT name  
FROM employee1232  
ORDER BY name DESC;

```
mysql> SELECT name
-> FROM employee32
-> ORDER BY name DESC;
+-----+
| name |
+-----+
| Sureshsingh |
| Saleh |
| Root |
| Rohitsharma |
| Rohan |
| Niki |
| Milansar |
| Joseph |
| Jane |
| James |
| James |
| jadavyadav |
| Himanshu |
| farhan |
| faizan |
| Drake |
| Anish |
+-----+
17 rows in set (0.00 sec)

mysql> _
```

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

UPDATE employee1232 SET city="Delhi" WHERE empno=2;

```
mysql> UPDATE employee32 SET city="Delhi" WHERE empno=33;
Query OK, 1 row affected (0.08 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from employee32;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | name       | deptno | job       | hiredate | sal    | commission | dob       | city       | phone |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 33    | jadavyadav | 10     | CLERK     | 1981-03-15 | 12500 | 100        | 1988-05-14 | Delhi     | 98221 |
| 34    | Rohitsharma | 10     | CLERK     | 1981-02-15 | 1200  | 100        | 1988-05-14 | Delhi     | 98221 |
| 35    | Milansar   | 20     | SALESMAN  | 0000-00-00 | 5000  | 100        | 1988-05-14 | Chennai  | 98221 |
| 36    | Joseph     | 20     | CLERK     | 1981-06-15 | 10000 | 100        | 1988-05-14 | Haryana  | 98221 |
| 37    | Root       | 30     | MANAGER   | 1981-07-15 | 10002 | 100        | 1988-05-14 | Chandigarh | 98221 |
| 38    | Drake      | 30     | SALESMAN  | 1981-08-15 | 8211  | 100        | 1988-05-14 | CHICAGO   | 98221 |
| 39    | Jane       | 30     | CLERK     | 1981-09-15 | 9800  | 100        | 1988-05-14 | CHICAGO   | 98221 |
| 40    | Niki       | 40     | ASS-MANAGER | 1981-10-15 | 4021  | 100        | 1988-05-14 | NEW YORK  | 98221 |
| 41    | James      | 40     | PRO-MANAGER | 1981-11-15 | 98002 | 100        | 1988-05-14 | NEW YORK  | 98221 |
| 42    | James      | 40     | CLERK     | 1981-12-20 | 9122  | 100        | 1988-05-14 | DALLAS    | 98221 |
| 43    | Anish      | 40     | SALESMAN  | 1991-12-20 | 8123  | 100        | 1988-05-14 | DALLAS    | 98221 |
| 44    | Saleh      | 50     | ACC-HOLDER | 1991-02-20 | 2034  | 100        | 1988-05-14 | BOSTON    | 98221 |
| 45    | farhan     | 50     | ASSI-MANAGER | 1991-12-20 | 11239 | 100        | 1988-05-14 | BOSTON    | 98221 |
| 46    | Himanshu   | 50     | CLERK     | 2004-12-20 | 90000 | 100        | 1988-05-14 | Hyderabad | 98221 |
| 47    | Rohan      | 20     | SALESMAN  | 1981-05-15 | 8000  | 100        | 1988-05-14 | Hyderabad | 98221 |
| 48    | Sureshsingh | 30     | TAX       | 1981-04-15 | 12000 | 100        | 1988-05-14 | Bangalore | 98221 |
| 49    | faizan     | 10     | teacher   | 2007-03-15 | 1250000 | 100       | 2009-05-14 | delhi     | 98973 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
17 rows in set (0.00 sec)

mysql> _
```

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

- SELECT SUM(s.basic) AS "Sum Of Basic Salary of Employees from Delhi"

FROM employee1232 e INNER JOIN salary1232 s

WHERE e.empno=s.empno AND e.city="Delhi";

```
mysql> SELECT SUM(s.basic) AS "Sum Of Basic Salary of Employees from Delhi"
      -> FROM employee32 e INNER JOIN salary32 s
      -> WHERE e.empno=s.empno AND e.city="Delhi";
+-----+
| Sum Of Basic Salary of Employees from Delhi |
+-----+
|                                     22000 |
+-----+
1 row in set (0.00 sec)

mysql>
```

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

- SELECT \* FROM employee1232 e,(select empno FROM salary1232 s, (select MAX(tax) as max from salary1232) m where s.tax=m.max) a WHERE e.empno = a.empno ;

```
mysql> SELECT * FROM employee32 e,(select empno FROM salary32 s, (select MAX(tax) as max from salary32) m where s.tax=m.max) a WHERE e.empno = a.empno ;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | name  | deptno | job          | hiredate | sal  | commission | dob       | city    | phone  | empno |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 45    | farhan | 50     | ASSI-MANAGER | 1991-12-20 | 11239 | 100        | 1988-05-14 | BOSTON | 9822121221 | 45    |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

7) Which employee is the senior most?

- SELECT e.name AS name , e.hiredate as "Hire Date" FROM employee1232 e, (SELECT e.empno From employee1232 e, (Select MIN(hiredate) as hiredate FROM employee1232) h WHERE e.hiredate = h.hiredate) f WHERE e.empno=f.empno;

```
mysql> SELECT e.name AS name , e.hiredate as "Hire Date" FROM employee32 e, (SELECT e.empno From employee32 e, (Select MIN(hiredate) as hiredate FROM employee32)
RE e.empno=f.empno;
+-----+-----+
| name | Hire Date |
+-----+-----+
| Milansar | 0000-00-00 |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

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

Select \* from Employee1232 order by sal desc limit 1,1;

```
mysql> Select * from Employee32 order by sal desc limit 1,1;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | name | deptno | job          | hiredate | sal | commission | dob          | city      | phone      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 41 | James | 40 | PRO-MANAGER | 1981-11-15 | 98002 | 100 | 1988-05-14 | NEW YORK | 9822121221 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

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

select \* from Employee1232 e1 where 2=(select count(distinct sal) from Employee1232 where e1.sal<=sal);

```
mysql> select * from Employee32 e1 where 2=(select count(distinct sal) from Employee32 where e1.sal<=sal);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | name | deptno | job          | hiredate | sal | commission | dob          | city      | phone      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 41 | James | 40 | PRO-MANAGER | 1981-11-15 | 98002 | 100 | 1988-05-14 | NEW YORK | 9822121221 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

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

- MINUS not support in sql



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

select \* from Employee1232 e1 where 6 = (select count(\*) from Employee1232 where e1.sal <=sal);

```
mysql>
mysql> select * from Employee32 e1 where 6 = (select count(*) from Employee32 where e1.sal <=sal);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | name  | deptno | job          | hiredate | sal  | commission | dob       | city    | phone  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 45    | farhan | 50     | ASSI-MANAGER | 1991-12-20 | 11239 | 100        | 1988-05-14 | BOSTON | 9822121221 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

12) How many clerks are there in the company?

- select count(\*) from Employee1232 where job='Clerk';

```
mysql> select count(*) from Employee32 where job='Clerk';
+-----+
| count(*) |
+-----+
| 6        |
+-----+
1 row in set (0.00 sec)

mysql>
```

13) Which department has exactly one employee as clerk?

- select d.dname from Employee1232 as e join Department1232 as d where e.deptno=d.deptno and e.job='Clerk' group by d.deptno having count(\*)=1;

```
mysql> select d.dname from Employee32 as e join Department32 as d where e.deptno=d.deptno and e.job='Clerk' group by d.deptno having count(*)=1;
+-----+
| dname |
+-----+
| RESEARCH |
| SALES   |
| OPERATIONS |
| EC      |
+-----+
4 rows in set (0.00 sec)

mysql>
```

14) Which department has the highest number of clerks? Show the deptno and count.

- Select d.dname,e.deptno,count(name) from Employee1232 e,Department1232 d where e.deptno=d.deptno && e.job='Clerk' group by e.deptno order by name desc limit 0,1;

```
mysql> Select d.dname,e.deptno,count(name) from Employee32 e,Department32 d where e.deptno=d.deptno && e.job='Clerk' group by e.deptno order by name desc limit 0,1;
+-----+-----+-----+
| dname | deptno | count(name) |
+-----+-----+-----+
| RESEARCH | 20 | 1 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

15) How many employees are there in each department?

- select d.dname,count(\*) No\_of\_employees from Employee1232 as e join Department1232 as d where e.deptno=d.deptno group by d.deptno ;

```
mysql> select d.dname,count(*) No_of_employees from Employee32 as e join Department32 as d where e.deptno=d.deptno group by d.deptno ;
+-----+-----+
| dname | No_of_employees |
+-----+-----+
| ACCOUNTING | 3 |
| RESEARCH | 3 |
| SALES | 4 |
| OPERATIONS | 4 |
| EC | 3 |
+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

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

- select job,min(sal) from Employee1232 group by job order by sal desc;

```
mysql> select job,min(sal) from Employee32 group by job order by sal desc;
```

job	min(sal)
teacher	1250000
PRO-MANAGER	98002
CLERK	1200
TAX	12000
ASSI-MANAGER	11239
MANAGER	10002
SALEMAN	8123
SALESMAN	5000
ASS.MANAGER	4021
ACC.HOLDER	2034

```
10 rows in set (0.00 sec)

mysql>
```

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

- select deptno,avg(sal) from Employee1232 group by deptno order by avg(sal) asc limit 1;

```
mysql> select deptno,avg(sal) from Employee32 group by deptno order by avg(sal) asc limit 1;
```

deptno	avg(sal)
20	7666.6667

```
1 row in set (0.00 sec)

mysql>
```

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

- select job,min(sal),max(sal),avg(sal) from Employee1232 group by job;

```
mysql> select job,min(sal),max(sal),avg(sal) from Employee32 group by job;
```

job	min(sal)	max(sal)	avg(sal)
ACC.HOLDER	2034	2034	2034.0000
ASS.MANAGER	4021	4021	4021.0000
ASSI-MANAGER	11239	11239	11239.0000
CLERK	1200	90000	22103.6667
MANAGER	10002	10002	10002.0000
PRO-MANAGER	98002	98002	98002.0000
SALEMAN	8123	8123	8123.0000
SALESMAN	5000	8211	7070.3333
TAX	12000	12000	12000.0000
teacher	1250000	1250000	1250000.0000

```
10 rows in set (0.00 sec)

mysql>
```

19) Compute the difference between maximum and minimum salary.

- select max(sal)-min(sal) from Employee1232;

```
mysql> select max(sal)-min(sal) from Employee32;
```

max(sal)-min(sal)
1248800

```
1 row in set (0.03 sec)

mysql> _
```

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

- select name from Employee1232 where name like '%LA%';

```
mysql> select name from Employee32 where name like '%LA%';
```

name
Milansar

```
1 row in set (0.00 sec)

mysql> _
```

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

- select name from Employee1232 where hiredate between '1981-04-02' and '1981-09-08';

```
mysql> select name from Employee32 where hiredate between '1981-04-02' and '1981-09-08';
+-----+
| name |
+-----+
| Joseph |
| Root |
| Drake |
| Rohan |
| Sureshsingh |
+-----+
5 rows in set (0.00 sec)

mysql> _
```

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

- select count(distinct job) from Employee1232;

```
mysql> select count(distinct job) from Employee32;
+-----+
| count(distinct job) |
+-----+
| 10 |
+-----+
1 row in set (0.00 sec)

mysql>
```

23) Compute the sum of all salaries of employee working under deptno=30.

- select sum(sal) from Employee1232 where deptno=30;

```
mysql> select sum(sal) from Employee32 where deptno=30;
+-----+
| sum(sal) |
+-----+
| 40013 |
+-----+
1 row in set (0.00 sec)

mysql> _
```

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

- select e.name,e.deptno,d.dname from Employee1232 as e, Department1232 as d where e.deptno=d.deptno and e.job='salesman';

```
mysql> select e.name,e.deptno,d.dname from Employee32 as e, Department32 as d where e.deptno=d.deptno and e.job='salesman';
```

name	deptno	dname
Milansar	20	RESEARCH
Rohan	20	RESEARCH
Drake	30	SALES

```
3 rows in set (0.00 sec)

mysql>
```

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

- select e.name,d.manager from Employee1232 as e, Department1232 as d where e.deptno=d.deptno;

```
mysql> select e.name,d.manager from Employee32 as e, Department32 as d where e.deptno=d.deptno;
```

name	manager
jadavyadav	Henry
Rohitsharma	Henry
faizan	Henry
Milansar	John
Joseph	John
Rohan	John
Root	Samules
Drake	Samules
Jane	Samules
Sureshsingh	Samules
Niki	Dev
James	Dev
James	Dev
Anish	Dev
Saleh	Kev
farhan	Kev
Hinanshu	Kev

```
17 rows in set (0.00 sec)

mysql>
```

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

- select e.name from Employee1232 as e, Department1232 as d where e.deptno=d.deptno and d.loc='Chicago';

```
mysql> select e.name from Employee32 as e, Department32 as d where e.deptno=d.deptno and d.loc='Chicago';
+-----+
| name |
+-----+
| Saleh |
| farhan |
| Himanshu |
+-----+
3 rows in set (0.00 sec)

mysql>
```

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

- select e.name,d.dname from Employee1232 as e, Department1232 as d where e.deptno=d.deptno and e.job='Manager';

```
mysql> select e.name,d.dname from Employee32 as e, Department32 as d where e.deptno=d.deptno and e.job='Manager';
+-----+-----+
| name | dname |
+-----+-----+
| Root | SALES |
+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

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 Employee1232 where deptno=10 and sal>=any(select sal from Employee1232 where deptno=30);

```
mysql> select name from Employee32 where deptno=10 and sal>=any(select sal from Employee32 where deptno=30);
+-----+
| name |
+-----+
| jadavyadav |
| faizan |
+-----+
2 rows in set (0.00 sec)

mysql> _
```

29) List all the department who have no employees.

- select deptno, dname from Department1232 where deptno not in(select deptno from Employee1232);

```
mysql> select deptno, dname from Department32 where deptno not in(select deptno from Employee32);
Empty set (0.00 sec)

mysql>
```

30) Delete the EC department.

- Alter table Employee1232 drop foreign key employee1232\_ibfk\_1;
- Alter table Salary1232 drop foreign key salary1232\_ibfk\_1;
- Delete from Employee1232 where deptno=50;Delete from Department1232 where deptno=50;

```
mysql> select deptno, dname from Department32 where deptno not in(select deptno from Employee32);
Empty set (0.00 sec)

mysql> Alter table Employee32 drop foreign key employee32_ibfk_1;
Query OK, 17 rows affected (0.23 sec)
Records: 17 Duplicates: 0 Warnings: 0

mysql> Alter table Salary32 drop foreign key salary32_ibfk_1;
Query OK, 3 rows affected (0.25 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> Delete from Employee32 where deptno=50;Delete from Department32 where deptno=50;
Query OK, 3 rows affected (0.03 sec)

Query OK, 1 row affected (0.01 sec)

mysql> select * from department32;
+-----+-----+-----+-----+
| deptno | dname      | manager | loc      |
+-----+-----+-----+-----+
| 10     | ACCOUNTING | Henry   | NEW YORK |
| 20     | RESEARCH  | John    | DALLAS   |
| 30     | SALES     | Samuels | NEVADA    |
| 40     | OPERATIONS | Dev     | BOSTON    |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

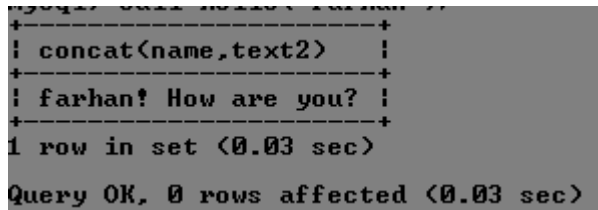


## **question-2**

Q 1. Write a function and a stored procedure to print Hello ! How are you?.

Store-Procedure:

```
create procedure hello(IN name varchar(10))  
  
begin  
  
declare text2 varchar(20) default "! How are you?";  
  
select concat(name,text2);  
  
end;  
  
/
```



The screenshot shows the output of the stored procedure. It displays a table with one row containing the concatenated string 'farhan! How are you?'. Below the table, it indicates '1 row in set (0.03 sec)' and 'Query OK, 0 rows affected (0.03 sec)'.

concat(name,text2)
farhan! How are you?

1 row in set (0.03 sec)  
Query OK, 0 rows affected (0.03 sec)

Function:

```
create function hello( name varchar(10))  
  
returns varchar(45)  
  
begin  
  
declare text2 varchar(20) default "! How are you?";  
  
declare m varchar(45);  
  
set m=concat(name,text2);  
  
return m;  
  
end
```

```
Query OK, 0 rows affected (0.00 sec)
mysql> select hello('farhan')/
+-----+
| hello('farhan') |
+-----+
| farhan! How are you? |
+-----+
1 row in set (0.05 sec)
mysql>
```

Q 2. Write a function and a stored procedure to count the number of employees in the table employee.

Store-Procedure:

```
create procedure numr()
begin
select count(*) from employee1232;
end
```

```
mysql> create procedure numr()
-> begin
-> select count(*) from employee32;
-> end
-> /
Query OK, 0 rows affected (0.02 sec)
mysql> call numr/
+-----+
| count(*) |
+-----+
|      14 |
+-----+
1 row in set (0.02 sec)
Query OK, 0 rows affected (0.02 sec)
mysql>
```

Function:

```
create function numr()
returns int
begin
declare x int;
select count (*) into x from employee1232;
return x;
```

end;

```
mysql> create function numr()
-> returns int
-> begin
-> declare x int;
-> select count(*) into x from employee32;
-> return x;
-> end;
-> /
Query OK, 0 rows affected (0.00 sec)

mysql> select numr();
+-----+
| numr() |
+-----+
|      14 |
+-----+
1 row in set (0.00 sec)
```

Q 3. Write a function and a stored procedure to calculate the factorial of the given number.

Store-Procedure:

```
create procedure fac(in N int)

begin
declare f int default 1;
myloop:loop
if N=0 then
leave myloop;
else
set f=f*N;
set N=N-1;
end if;
end loop;
select f;
end;
```

```

mysql> create procedure fac(in N int)
-> begin
-> declare f int default 1;
-> myloop:loop
-> if N=0 then
-> leave myloop;
-> else
-> set f=f*N;
-> set N=N-1;
-> end if;
-> end loop;
-> select f;
-> end;
-> /
Query OK, 0 rows affected (0.05 sec)

mysql> call fac(5)/
+-----+
| f      |
+-----+
| 120    |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>

```

Function:

create function fac(N int)

returns int

begin

declare f int default 1;

myloop:loop

if N=0 then

leave myloop;

else

```

set f=f*N;

set N=N-1;

end if;

end loop;

return f;

end;

```

```

mysql> create function fac(N int)
-> returns int
-> begin
-> declare f int default 1;
-> myloop:loop
-> if N=0 then
-> leave myloop;
-> else
-> set f=f*N;
-> set N=N-1;
-> end if;
-> end loop;
-> return f;
-> end;
-> /
Query OK, 0 rows affected (0.00 sec)

mysql> select fac(5)/
+-----+
| fac(5) |
+-----+
|    120 |
+-----+
1 row in set (0.00 sec)

mysql>

```

Q 4. Write a function and a stored procedure to calculate the average of three numbers.

Store-Procedure:

```

create procedure ave(in a real,in b real,in c real)

begin

declare av real;

set av=(a+b+c)/3;

```

select av;

end ;

```
mysql> create procedure ave(in a real,in b real,in c real)
-> begin
-> declare av real;
-> set av=(a+b+c)/3;
-> select av;
-> end ;
-> #
Query OK, 0 rows affected (0.05 sec)

mysql> call ave(5,9,7)#
+-----+
| av    |
+-----+
|      7 |
+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)
```

Function:

create function ave(a real,b real,c real)

returns real

begin

declare av real;

set av=(a+b+c)/3;

return av;

end ;

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

mysql> select ave(7,5,1)#
+-----+
| ave(7,5,1) |
+-----+
| 4.333333333333333 |
+-----+
1 row in set (0.00 sec)
```

Q5. Write a function and stored procedure to find fibonacci series and its sum.

Store-procedure:

```
create procedure fib(num INT)
begin
declare sum INT default 0;
declare a INT default 0;
declare b INT default 1;
declare c INT default 0;
declare s varchar(100);
SET num=num-2;
set sum = sum + a;
select concat(a, ' ') into s;
myloop:loop
if num < 0 then
leave myloop;
else
set c=a+b;
set a=b;
set b=c;
set num=num-1;
select concat(s, ' ',a) into s;
set sum = sum + a;
end if;
end loop;
select s,sum;
end?
```

```

mysql> create procedure fib(num INT)
-> begin
-> declare sum INT default 0;
-> declare a INT default 0;
-> declare b INT default 1;
-> declare c INT default 0;
-> declare s varchar(100);
-> SET num=num-2;
-> set sum = sum + a;
-> select concat(a,' ') into s;
-> myloop:loop
-> if num < 0 then
-> leave myloop;
-> else
-> set c=a+b;
-> set a=b;
-> set b=c;
-> set num=num-1;
-> select concat(s,' ',a) into s;
-> set sum = sum + a;
-> end if;
-> end loop;
-> select s,sum;
-> end;
-> /
Query OK, 0 rows affected (0.00 sec)

mysql> select fib(10)/
ERROR 1305 (42000): FUNCTION farhan.fib does not exist
mysql> call fib(10)/
+-----+-----+
| s | sum |
+-----+-----+
| 0 1 1 2 3 5 8 13 21 34 | 88 |
+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

```

Function:

create function fib(num INT)

returns varchar(100)

begin

declare sum INT default 0;

declare a INT default 0;

declare b INT default 1;

declare c INT default 0;

declare s varchar(100);

SET num=num-2;



```

set sum = sum + a;

select concat(a,' ') into s;

myloop:loop

if num < 0 then

leave myloop;

else

set c=a+b;

set a=b;

set b=c;

set num=num-1;

set s=concat(s,' ',a) ;

set sum = sum + a;

end if;

end loop;

return concat(s," :: sum= ",sum);

end;

```

```

mysql> create function fib(num INT)
-> returns varchar(100)
-> begin
-> declare sum INT default 0;
-> declare a INT default 0;
-> declare b INT default 1;
-> declare c INT default 0;
-> declare s varchar(100);
-> SET num=num-2;
-> set sum = sum + a;
-> select concat(a,' ') into s;
-> myloop:loop
-> if num < 0 then
-> leave myloop;
-> else
-> set c=a+b;
-> set a=b;
-> set b=c;
-> set num=num-1;
-> set s=concat(s,' ',a) ;
-> set sum = sum + a;
-> end if;
-> end loop;
-> return concat(s," :: sum= ",sum);
-> end;
-> /

```

Query OK, 0 rows affected (0.00 sec)

```
mysql> select fib(10)/
```

```

+-----+
| fib(10) |
+-----+
| 0  1  1  2  3  5  8  13  21  34  :: sum= 88 |
+-----+
1 row in set (0.00 sec)

```

## **question-3**

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 one record per student-class pair such that the student is enrolled in the class.

Write the SQL queries. No duplicates should be printed.(use foreign key )

1. Find the names of all Juniors (level = JR) who are enrolled in a class taught by I. Teach.
2. Find the age of the oldest student who is either a History major or enrolled in a course taught by I. Teach.
3. Find the names of all classes that either meet in room BA1080 or have 2 or more students enrolled.
4. Find the names of all students who are enrolled in two classes that meet at the same time.
5. Find the names of faculty members who teach in every room in which some class is taught.
6. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.
7. For each level, print the level and the average age of students for that level.
8. For all levels except JR, print the level and the average age of students for that level.
9. For each faculty member that has taught classes only in room R128, print the faculty member's name and the total number of classes she or he has taught.
10. Find the names of students enrolled in the maximum number of classes.

TABLE-CREATE:

1)STUDENT1232: create table student1232(

snum int primary key,

```

sname varchar(30),
major varchar(30),
level varchar(30),
age int
);
2)CLASS1232: create table class1232(
name varchar(20) primary key,
meets_at time,
room varchar(20),
fid int,
foreign key(fid) references faculty1232(fid)
ON UPDATE CASCADE ON DELETE CASCADE
);
3)ENROLLED1232: create table enrolled1232(
snum int,
cname varchar(20),
foreign key(snum) references student1232(snum)
ON UPDATE CASCADE ON DELETE CASCADE,
foreign key(cname) references class1232(name)
ON UPDATE CASCADE ON DELETE CASCADE
);
4)FACULTY1232: create table faculty1232(
fid int primary key,
fname varchar(30),
deptid int
);

```

**After insertion the values:**

```
mysql> select * from student32;
```

snum	sname	major	level	age
102	ILMA	CS	JR	21
103	SHAHBAZ	ELEC	JR	22
104	FARHAN	Mech	SR	25
105	FAIZ RAB	civ	SR	40
132	FAIZAN khan	CS	JR	20

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

```
mysql> select * from enrolled32;
```

snum	cname
132	DBMS
103	signal and system
104	Electronics
104	OPERATING SYS
132	ARCHITECTURE

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

```
mysql> select * from calss32;
```

ERROR 1146 (42S02): Table 'farhan.calss32' doesn't exist

```
mysql> select * from class32;
```

name	meets_at	room	fid
ARCHITECTURE	09:00:00	324	2
Automatat theory	09:50:00	324	2
DATAstructure	09:00:00	BA1080	1
DBMS	09:00:00	324	1
Electronics	09:00:00	325	3
Microprocessor	09:00:00	R128	3
OPERATING sys	09:50:00	307	2
signal and system	09:50:00	324	1

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

```
mysql> select * from faculty32;
```

fid	fname	deptid
1	computer	123
2	civil	191
3	electronics	423

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

```
mysql>
```

## QUERIES

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

```
select sname from student1232 as s, enrolled1232 as e where s.snum = e.snum AND
e.cname="DBMS" AND s.level="JR";
```

```
mysql> select sname from student32 as s,enrolled32 as e where s.snum = e.snum AND e.cname="DBMS" AND s.level="JR";
+-----+
| sname |
+-----+
| FAIZAN khan |
+-----+
1 row in set (0.02 sec)

mysql>
```

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(age) from student1232 as s,enrolled1232 as e where (s.snum = e.snum) AND
(s.major="history" OR e.cname="DBMS");
```

```
mysql> select max(age) from student32 as s,enrolled32 as e where (s.snum = e.snum) AND (s.major="history" OR e.cname="DBMS");
+-----+
| max(age) |
+-----+
|      20 |
+-----+
1 row in set (0.00 sec)

mysql>
```

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

```
select name from class1232 as c where c.room='BA1080';
```

```
mysql> select name from class32 as c where c.room='BA1080';
+-----+
| name |
+-----+
| DATAstructure |
+-----+
1 row in set (0.00 sec)

mysql>
```

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

```
select Student1232.sname from ((Enrolled1232 inner join student1232 on
Enrolled1232.snum=Student1232.snum) inner join Class1232 on
Class1232.name=Enrolled1232.cname) group by Class1232.meets_at,Student1232.snum having
count(Enrolled1232.cname) >= 2;
```

```
mysql> select Student32.sname from (<Enrolled32 inner join student32 on Enrolled32.snum=Student32.snum) inner join Class32 on Class32.name=Enrolled32.cname) group by Cl
ving count(Enrolled32.cname) >= 2;
+-----+
| sname |
+-----+
| FAIZAN khan |
+-----+
1 row in set (0.01 sec)

mysql>
```

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

```
select Faculty1232.fname from (Class1232 inner join Faculty1232 on Class1232.fid=Faculty1232.fid)
group by Faculty1232.fid having count(DISTINCT Class1232.room)=(select count(DISTINCT
Class1232.room) from Class1232);
```

```
mysql> select Faculty32.fname from (Class32 inner join Faculty32 on Class32.fid=Faculty32.fid) group by Faculty32.fid having count(DISTINCT Class32.room)=(select count
s32);
Empty set (0.00 sec)
```

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

```
select Faculty1232.fname from(Class1232 right join Faculty1232 on Class1232.fid=Faculty1232.fid)
group by Faculty1232.fname having count(DISTINCT Class1232.name) < 5;
```

```
mysql> select Faculty32.fname from(Class32 right join Faculty32 on Class32.fid=Faculty32.fid) group by Faculty32.fname having count(DISTINCT Class32.name) < 5;
+-----+
| fname |
+-----+
| civil |
| computer |
| electronics |
+-----+
3 rows in set (0.00 sec)
```

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

```
select Student1232.level, avg(age) as AVG_age from Student1232 group by Student1232.level;
```

```
mysql> select Student32.level, avg(age) as AVG_age from Student32 group by Student32.level;
+-----+-----+
| level | AVG_age |
+-----+-----+
| JR    | 21.0000 |
| SR    | 32.5000 |
+-----+-----+
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 Student1232.level, avg(Student1232.age) as AVG_NOT_JR from Student1232 where
Student1232.level != 'JR' group by Student1232.level;
```

```
mysql> select Student32.level, avg(Student32.age) as AVG_NOT_JR from Student32 where Student32.level != 'JR' group by Student32.level;
+-----+-----+
| level | AVG_NOT_JR |
+-----+-----+
| SR    | 32.5000 |
+-----+-----+
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 Faculty1232.fname, count(Class1232.name) as COUNT from (Class1232 inner join Faculty1232
on Class1232.fid=Faculty1232.fid) where Class1232.room='R128' group by Class1232.fid;
```

```
mysql> select Faculty32.fname, count(Class32.name) as COUNT from (Class32 inner join Faculty32 on Class32.fid=Faculty32.fid) where Class32.room='R128' group by Class32.fid;
+-----+-----+
| fname      | COUNT |
+-----+-----+
| electronics | 1     |
+-----+-----+
1 row in set (0.01 sec)
```

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

```
select Student1232.sname, Class1232.name from ((Enrolled1232 inner join Student1232 on
Enrolled1232.snum=Student1232.snum) inner join Class1232 on
Class1232.name=Enrolled1232.cname) group by Student1232.sname having count(Class1232.name)
>= ALL(select count(*) from ((Enrolled1232 inner join Student1232 on
```

Enrolled1232.snum=Student1232.snum) inner join Class1232 on  
Class1232.name=Enrolled1232.cname) group by Student1232.sname);

```
mysql> select Student32.sname,Class32.name from (<Enrolled32 inner join Student32 on Enrolled32.snum=Student32.snum> inner join Class32 on Class32.name=Enrolled32.cname)
g count(Class32.name) >= ALL(select count(*) from(<Enrolled32 inner join Student32 on Enrolled32.snum=Student32.snum> inner join Class32 on Class32.name=Enrolled32.cname)
+-----+-----+
| sname | name |
+-----+-----+
| FAIZAN khan | DBMS |
| FARHAN | Electronics |
+-----+-----+
2 rows in set (0.00 sec)
```

## **QUESTION-4**

Write equivalent SQL for the following query.

1. Get the title,author name,publisher name for author whose city contain total no of a=2?
2. Give the details of the book which is written by at least two authors.
3. Write a stored procedure (SP Name : insertIntoAuth) to insert the Author information.
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).
5. Write a stored procedure to delete the Author information using its AuthID. (Notr: 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).



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

NOTE: Book information should be deleted from both Book and Book\_writtenBy\_Author table.

Book ID Already exist when we insert Book with same id.

Author of given ID does not exist when we enter wrong AuthID in insertBookInfo, insertBook stored procedure.

Author ID already exist when we insert duplicate AuthID in insertIntoAuth stored procedure.

Age Should be greater than 18 and less than 60' if age is invalid (Age data type should be DATE ). Use function to validate the age in stored procedure. Function Name : AgeValidate .

Table-content

```
mysql> select * from author1232#
```

authid	authfirst	authlast	authmid	authcity	age
1	mohd	farhan	khan	patna	21
2	mohd	faizan	khan	bareilly	19
3	peter	galvin		usa	119
4	thomas	gregne		usa	25

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

```
mysql> select * from book1232#
```

bookid	copyright	title	pages
1	EEE	OPERATING SYSTEM	400

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

```
mysql> select * from written_by1232#
```

bookid	authid
1	3
1	4

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

## Queries:

- Get the title,author name,publisher name for author whose city contain total no of a=2?

select title,authfirst,authmid,authlast from book1232 as b,author1232 as a,written\_by1232 as w  
where b.bookid=w.bookid and a.authid=w.authid and a.authcity like '%a%a%';

```
mysql> select title,authfirst,authmid,authlast from book1232 as b,author1232 as a,written_by1232 as w where b.bookid=w.bookid a
-> #
+-----+-----+-----+-----+
| title | authfirst | authmid | authlast |
+-----+-----+-----+-----+
| maths | mohd      | khan    | farhan   |
+-----+-----+-----+-----+
row in set (0.02 sec)
```

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

```
select * from book1232 where bookid in(select bookid from written_by1232
as w,author1232 as a where a.authid=w.authid group by w.bookid having count(*)>=2
);
```

```
mysql> select * from book1232 where bookid in(select bookid from written_by1232
-> as w,author1232 as a where a.authid=w.authid group by w.bookid having count(*)>=2
-> );
-> #
+-----+-----+-----+-----+
| bookid | copyright | title           | pages |
+-----+-----+-----+-----+
| 1      | EEE       | OPERATING SYSTEM | 400   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

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

```
create procedure insertIntoAuth(in authid varchar(25),in authfirst varchar(25),in authmiddle
varchar(25),in authlast varchar(25),in age date,in authcity varchar(25))
begin
declare x int;
declare m varchar(45);
select AgeVAlidate(age) into x;
set m=concat("age not valid");
```

```

if x=1
then
insert into author1232 values(authid,authfirst,authlast,authmiddle,authcity,age);
else
select m;
end if;
end;

```

```

mysql> call insertIntoAuth(6,'sd','gh','dfg','1990-8-9','bareilly');
-> #
Query OK, 1 row affected (0.06 sec)

mysql> select * from author1232;
-> #

```

authid	authfirst	authlast	authmid	authcity	age
1	mohd	farhan	khan	ahmadabad	1985-5-5
2	mohd	faizan	khan	bareilly	1997-8-12
3	peter	galvin		usa	1950-5-6
4	thomas	gregne		usa	1940-9-5
6	sd	dfg	gh	bareilly	1990-08-09

```

5 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).**

```

create procedure insertBook(in bookid varchar(45),in title varchar(45),in no_pages int,in
copyright varchar(25))
begin
insert into Book1232 values(bookid,title,no_pages,copyright);
end;

```

```

create procedure insertWBy(in bookid varchar(45),in publisher_name varchar(45),in authid
varchar(25))
begin
insert into writtenBy1232 values(bookid,Publisher_Name,authid);
end ;

```

```

create procedure insertBookinfo(in bid varchar(45),in title varchar(45),in no_pages int,in
copyright varchar(25),in publisher_name varchar(45),in aid varchar(25))
begin
declare i int;
declare j int;
declare m varchar(45);
declare n varchar(45);

```

```

set m=concat("book id already exist");
set n=concat("author id must exist");
select count(*) INTO I from Book1232 as b where b.bookid=bid;
select count(*) from author1232 as a where a.authid=aid into j;
if i=0
then
if j>0
then
call insertbook(bid,title,no_pages,copyright);
call insertWBy(bid,publisher_name,aid);
end if;
else
if i>0
then
select m;
end if;
if j=0
then
select n;
end if;
end if;
end ;

```

```

mysql> create procedure insertBook(in bookid varchar(45),in title varchar(45),in no_pages int,in copyright varchar(25))
-> begin
-> insert into Book values(bookid,title,no_pages,copyright);
-> end;
->
-> create procedure insertWBy(in bookid varchar(45),in publisher_name varchar(45),in authid varchar(25))
-> begin
-> insert into writtenBy values(bookid,Publisher_Name,authid);
-> end ;
->
-> create procedure insertBookinfo(in bid varchar(45),in title varchar(45),in no_pages int,in copyright varchar(25),in p
-> begin
-> declare i int;
-> declare j int;
-> declare m varchar(45);
-> declare n varchar(45);
-> set m=concat("book id already exist");
-> set n=concat("author id must exist");
-> select count(*) INTO I from Book as b where b.bookid=bid;
-> select count(*) from auth as a where a.authid=authid into j;
-> if i=0
-> then
-> if j>0
-> then
-> call insertbook(bookid,title,no_pages,copyright);
-> call insertWBy(bookid,publisher_name,authid);
-> end if;
-> else
-> if i>0
-> then
-> select m;
-> end if;
-> if j=0
-> then
-> select n;
-> end if;
-> end if;
-> end if;
-> end ;
-> #
query OK, 0 rows affected (0.01 sec)
query OK, 0 rows affected (0.01 sec)
query OK, 0 rows affected (0.01 sec)

```

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).

```
create procedure del_auth(in aid int)
begin
Declare cnt int default 0;
Declare at int default 0;
set at=aid;
select count(*) into cnt from written_by1232 where authid=aid;
if cnt=0 then
delete from Author1232 where authid=aid;
else
select "You cant delete Author because total
no book exist in BookTable. First delete all the books written by him" as deletion;
end if;
end ;
```

```
mysql>
mysql> create procedure del_auth(in aid int)
-> begin
-> Declare cnt int default 0;
-> Declare at int default 0;
-> set at=aid;
-> select count(*) into cnt from written_by1232 where authid=aid;
-> if cnt=0 then
-> delete from Author1232 where authid=aid;
-> else
-> select "You cant delete Author because total
-> no book exist in BookTable. First delete all the books written by him" as deletion;
-> end if;
-> end ;
-> #
Query OK, 0 rows affected (0.00 sec)
mysql>
```

## QUESTION-5

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 return 0

```
create function AgeValidate(age date)

returns int

begin

declare set_age int;

declare message int;

set set_age=datediff(curdate(),age)/365;

if set_age>18 and set_age<60

then

set message=1;

else

set message=0;

end if;

return message;

end;
```

```
mysql> create function AgeValidate<age date>
-> returns int
-> begin
-> declare set_age int;
-> declare message int;
-> set set_age=datediff(curdate(),age)/365;
-> if set_age>18 and set_age<60
-> then
-> set message=1;
-> else
-> set message=0;
-> end if;
-> return message;
-> end;
-> #
ERROR 1304 (42000): FUNCTION AgeValidate already exists
mysql> select AgeValidate('1993-4-1')#
+-----+
| AgeValidate('1993-4-1') |
+-----+
| 1 |
+-----+
1 row in set (0.00 sec)

mysql>
```

```
mysql> create trigger mytrigger after insert on Book32 for each row
-> begin
-> insert into log32 value ('root','insert','','','',new.bid,new.bname,new.authrname,now());
-> end;
-> #
Query OK, 0 rows affected (0.14 sec)

mysql> create trigger mytrigger1 after update on Book32 for each row
-> begin
-> insert into log32 value ('root','update',old.bid,old.bname,old.authrname,new.bid,new.bname,new.authrname,now());
-> end ;
-> #
Query OK, 0 rows affected (0.16 sec)

mysql> create trigger mytrigger3 after delete on Book32 for each row
-> begin
-> insert into log32 value ('root','delete',old.bid,old.bname,old.authrname','','','',now());
-> end ;
-> #
Query OK, 0 rows affected (0.58 sec)

mysql>
```

## After insert

```
mysql> insert into book32 values(1,'LET US C','Y KANETKAR')#
Query OK, 1 row affected (0.42 sec)

mysql> select * from log32#
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| user | operation | pbid | pbname | pauthrname | nbid | nbname | nauthrname | timeofop |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| root | insert   | 0    |      |             | 1    | LET US C | Y KANETKAR | 2014-11-20 10:29:06 |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

## After delete

```
mysql> delete from book32 where bid=1#
Query OK, 1 row affected (0.06 sec)

mysql> select * from log32#
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| user | operation | pbid | pbname | pauthrname | nbid | nbname | nauthrname | timeofop |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| root | insert   | 0    |      |             | 1    | LET US C | Y KANETKAR | 2014-11-20 10:29:06 |
| root | delete   | 1    | LET US C | Y KANETKAR | 0    |      |      | 2014-11-20 10:31:16 |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

## After update

```
mysql> update book32 set bname='POINTER IN C'where bid=1#
Query OK, 1 row affected (0.06 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from log32#
```

user	operation	pbid	pname	pauthname	nbid	nname	nauthname	timeofop
root	insert	0			1	LET US C	Y KANETKAR	2014-11-20 10:29:06
root	delete	1	LET US C	Y KANETKAR	0			2014-11-20 10:31:16
root	insert	0			1	LET US C	Y KANETKAR	2014-11-20 10:32:26
root	update	1	LET US C	Y KANETKAR	1	POINTER IN C	Y KANETKAR	2014-11-20 10:33:16

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

```
mysql>
```

## **CURSOR**

```
create procedure mycursor()
```

```
begin
```

```
declare f int default 1;
```

```
declare u varchar(50) default " ";
```

```
declare o varchar(50) default " ";
```

```
declare pid varchar(50) default " ";
```

```
declare pname varchar(50) default " ";
```

```
declare pauthname varchar(50) default " ";
```

```
declare nid varchar(50) default " ";
```

```
declare nname varchar(50);
```

```
declare nauthname varchar(50) default " ";
```

```
declare time varchar(50);
```

```
declare cn cursor for select * from log1232 where timeofop=(select max(timeofop) from log1232);
```

```
declare continue handler for not found set f=0;
```

```
open cn;
```

```
fetch cn into u,o,pid,pname,pauthname,nid,nname,nauthname,time;
```



```

select pid,nname;

if o ='insert' then

delete from book1232 where bid=nid;

delete from log1232 where nbid=nid or pbid=nid;

end if;

if o ='delete' then

insert into book1232

close cn;

end;

```

```

mysql> create procedure mycursor()
-> begin
-> declare f int default 1;
-> declare u varchar(50) default " ";
-> declare o varchar(50) default " ";
-> declare pid varchar(50) default " ";
-> declare pname varchar(50) default " ";
-> declare pauthrname varchar(50) default " ";
->
-> declare nid varchar(50) default " ";
-> declare nname varchar(50);
-> declare nauthrname varchar(50) default " ";
->
-> declare time varchar(50);
-> declare cn cursor for select * from log32 where timeofop=(select max(timeofop) from log32);
-> declare continue handler for not found set f=0;
-> open cn;
-> fetch cn into u,o,pid,pname,pauthrname,nid,nname,nauthrname,time;
-> select pid,nname;
-> if o ='insert' then
-> delete from book32 where bid=nid;
-> delete from log32 where nbid=nid or pbid=nid;
-> end if;
-> if o ='delete' then
-> insert into book32
-> close cn;
-> end;
-> #

```

/