Sub-queries with joins.

USE ***student_phone, student_address, faculty_phone, faculty_address, batch_students, course_batches, student_qualifications, faculty_qualifications, course_modules, modules, faculty, student, course, student_cards, and student_order*** relation to solve the following queries.

1. Display all student who have taken admission in more than 2 batches.

```
select distinct s.id, namefirst,namelast from student s join batch_students bs on s.id=bs.studentid where
s.id in(select studentid from batch_students group by studentid having count(batchid)>2);
```

2. Display the student detail who have joined the same batch of the student ‘saleel’.

```
select student.* from student join(select studentid from batch_students where batchid=(select id from
student where student.namefirst = 'saleel')) as saleel_batch on student.id = saleel_batch.studentid;
```

3. Display all courses where least number of students have taken the admission.

```
select c.* from batch_students bs, course_batches cb, course c where bs.batchid=cb.id and cb.courseid=c.id
and bs.studentid in(select studentid from batch_students group by studentid having
count(studentid)=(select min(r1) from (select studentid, count(*) R1 from batch_students group by
studentid)x));
```

4. Display student details who havenot taken the admission.

```
select s.id,s.namefirst from student s where not exists (select true from batch_students bs where
s.id=bs.studentid);
```

5. Get all courses where no modules are defined in course_modules table.

```
select c.id,c.name from course c where not exists (select true from course_modules cm where
c.id=cm.courseid);
```

6. Display course_*batches* details where student has taken the admission.

```
select *  from course_batches cb where exists (select true from batch_students bs where cb.id=bs.batchid);
```

7. Display all students whose marks of ‘BE’ is more than ‘ULKA’ marks in ‘BE’.

```
select s.id,s.namefirst,s.namelast,q.marks,q.name from student s join student_qualifications q on
s.id=q.studentid where q.marks>(select q.marks from student s join student_qualifications q on
s.id=q.studentid where s.namefirst="ulka" and q.name="BE") and s.namefirst<>"ulka" and q.name="BE";
```

8. Display all students whose marks are more than ‘saleel’ marks in 10th std.

```
select s.id,s.namefirst,s.namelast,q.marks,q.name from student s join student_qualifications q on
s.id=q.studentid where q.marks>(select q.marks from student s join student_qualifications q on
s.id=q.studentid where s.namefirst="saleel" and q.name=10) and s.namefirst<>"saleel" and q.name=10;
```

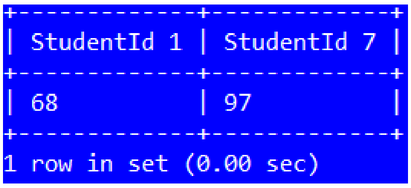
9. Display students whose DOB is as same as ‘kaushal’

```
select namefirst,namelast,DOB from student s where s.DOB=(select DOB from student where
namefirst="kaushal");
```

10. Display all student details who have three or more phone numbers.

```
select distinct s.id, namefirst,namelast,p.number from student s join student_phone p on s.id=p.studentid
where s.id in(select studentid from student_phone group by studentid having count(number)>=3)order by
id;
```

11. Display marks for the studentID 1 and 7 who have done ‘BE’. (Note: the marks must be displayed side by side).

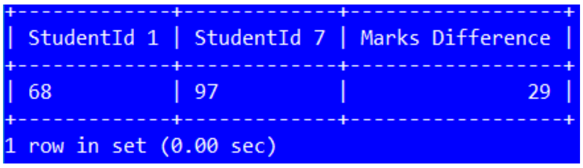


```
+-------------+-------------+
| StudentId 1 | StudentId 7 |
+-------------+-------------+
| 68          | 97          |
+-------------+-------------+
1 row in set (0.00 sec)
```

```
select(select q.marks from student s join student_qualifications q on s.id=q.studentid where q.name="BE"
and s.id=7 )r1,(select q.marks from student s join student_qualifications q on s.id=q.studentid where
q.name="BE" and s.id=1) r2;
```

12. Display marks for the studentID 1 and 7 who have done ‘BE’ also fine out the difference of marks between them.

(Note: the marks and difference between the marks must be displayed side by side)



```
+-------------+-------------+------------------+
| StudentId 1 | StudentId 7 | Marks Difference |
+-------------+-------------+------------------+
| 68          | 97          |               29 |
+-------------+-------------+------------------+
1 row in set (0.00 sec)
```

```
select abs((select q.marks from student s join student_qualifications q on s.id=q.studentid where
q.name="BE" and s.id=7 )-(select q.marks from student s join student_qualifications q on s.id=q.studentid
where q.name="BE" and s.id=1)) differ ;
```

13. Display all student who are not joined any of the batch.

```
select s.id,s.namefirst from student s where not exists (select true from batch_students bs where
s.id=bs.studentid);
```

14. Display all course_batches details who are starting on the same day as 'Batch1'.

```
select cb.* from course_batches cb join( select starton from course_batches where id = 1) as batch1_date
on cb.starton = batch1_date.starton;
```

15. Display all students whose 10th marks is more than student 'Neel's 10th marks.

```
select s.* from student s join student_qualifications sq on s.id = sq. studentid join(select sq1.marks as
neel_10th from student_qualifications sq1 join student s1 on sq1.studentid = s1.id where s1.namefirst =
'neel' and sq1.name = 1
0) as neel_marks on sq.name = 10 and sq.marks > neel_marks.neel_10th where s.id = sq.studentid;
```

16. Get all student with their qualification details who have highest marks in 'BE'.

```
select s.*,sq.name,sq.college,sq.university,sq.marks,sq.year from student s join student_qualifications sq
on s.id = sq.studentid join(select max(marks) as highest_be_marks from student_qualifications where
name = 'be') as max_be_marks on sq.name = 'be' and sq.marks = max_be_marks.highest_be_marks where
s.id = sq.studentid;
```

17. Get all student with their qualification details who have second highest marks in 'BE'.

```
select s.namefirst,s.namelast,q.name,q.college,q.university,q.marks,q.year from student s join
student_qualifications q on s.id=q.studentid where q.marks=(select max(q.marks) from student s join
student_qualifications q on s.id=q.studentid where q.marks < (select max(marks) from student s join
student_qualifications q on s.id=q.studentid where q.name="BE") and q.name="BE") and name="BE";
```

18. Display the student and student_qualification details who have scored the maximum marks in 'BE'

```
select s.namefirst,s.namelast,q.name,q.college,q.university,q.marks,q.year from student s join
student_qualifications q on s.id=q.studentid where q.marks in(select max(marks) from
student_qualifications where name="BE") and q.name="BE";
```

19. Display the student details who have scored the maximum marks in 'BE'

```
select s.namefirst,s.namelast,q.name,q.college,q.university,q.marks,q.year from student s join
student_qualifications q on s.id=q.studentid where q.marks=(select max(q.marks) from student s join
student_qualifications q on s.id=q.studentid where q.marks < (select max(marks) from student s join
student_qualifications q on s.id=q.studentid where q.name="BE") and q.name="BE") and name="BE";
```

20. Display the student details who have scored the minimum marks in ‘10’ std.

```
select s.namefirst,s.namelast,q.name,q.college,q.university,q.marks,q.year from student s join
student_qualifications q on s.id=q.studentid where q.marks in(select min(marks) from
student_qualifications where name="10") and q.name="10";
```

21. Display all student and student_qualification details of those students who have scored marks more than ‘RAJAN’ in ‘BE’.

```
select s.*,sq.name,sq.college,sq.university,sq.marks,sq.year from student s join student_qualifications sq
on s.id = sq.studentid join (select sq1.marks as rajan_be_marks from student_qualifications sq1 join
student s1 on sq1.studentid = s1.id where s1.namefirst = 'rajan' and sq1.name ='be') as rajan_marks on
sq.name = 'be' and sq.marks > rajan_marks.rajan_be_marks where s.id = sq.studentid;
```

22. Display all student who have done ‘BE’ in the same year as of studentID 16.

```
select s.* from student s join (select sq.year from student_qualifications sq where sq.studentid = 16 and
sq.name='be') as passing_year on exists ( select 1 from student_qualifications sq where sq.studentid = s.id
and sq.name = 'be' and sq.year = passing_year.year) where s.id != 16;
```

23. Display all odd records.

24. Calculate the sum of marks student wise of their qualifications (i.e. 10th, 12th and BE marks)

```
select s.id,sum(marks) as total_marks from student s left join(select studentid,sum(marks) as marks from
student_qualifications where name in('10','12','be') group by studentid) as sq on s.id = sq.studentid group
by s.id ;
```

25. Display students’ details who are not having *’Aadhaar’* card.

```
select student.* from student left join( select studentid from student_cards where name = 'Aadhaar') as
aadhar_st
ud on student.id = aadhar_stud.studentid where aadhar_stud.studentid is null;
```