Assignment

Sept23/ DBT/ 011

Database Technologies

Diploma in Advance Computing

September 2023

**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 s.namefirst,count(c.name) AdmissionBatch from student s join batch\_students b on(s.Id=b.studentid) join course\_batches c on(c.id=b.batchid) group by s.namefirst having count(c.name)>2; |
|  |
| 1. Display the student detail who have joined the same batch of the student ‘saleel’. |
| select \* from student s join batch\_students b on(s.Id=b.studentid) join course\_batches c on(c.id=b.batchid) where c.name in (select c1.name from course\_batches c1 join batch\_students b1 on (c1.Id=b1.studentid) join student s1 on (s1.id=b1.studentid) where s1.namefirst='saleel'); |
|  |
| 1. Display all courses where least number of students have taken the admission. |
| select c.name,count(\*) from course c join course\_modules cm on (c.id=cm.courseid) join course\_batches cb on (cb.courseid=cm.courseid) join batch\_students bs on (bs.batchid=cb.courseid) join student s on (bs.studentid=s.id) group by c.name order by count(\*) limit 1; |
|  |
| 1. Display student details who have not taken the admission. |
| select s.namefirst from student s where s.namefirst not in (select s.namefirst from student s join student\_qualifications sq on s.id=sq.studentid ); |
|  |
| 1. Get all courses where no modules are defined in course\_modules table. |
| select \* from course c where c.name not in (select c.name from course c join course\_modules cm on c.id=cm.courseid join modules m on m.id=cm.moduleid ); |
|  |
| 1. Display course*\_batches* details where student has taken the admission. |
| select distinct \* from course\_batches cb join batch\_students bs on cb.courseid=bs.batchid join student s on s.id=bs.studentid where cb.name in (select cb.name from course\_batches cb join batch\_students bs on cb.courseid=bs.batchid join student s on s.id=bs.studentid); |
|  |
| 1. Display all students whose marks of ‘BE’ is more than ‘ULKA’ marks in ‘BE’. |
| select \* from student s join student\_qualifications sq on(s.Id=sq.studentid) where sq.marks >all ( select sq.marks from student s join student\_qualifications sq on(s.Id=sq.studentid) where s.namefirst='ulka' and sq.name='be') and sq.name='be'; |
|  |
| 1. Display all students whose marks are more than ‘saleel’ marks in 10th std. |
| select \* from student s join batch\_students b on(s.Id=b.studentid) join student\_qualifications sq on(s.id=sq.studentid) join course\_batches c on(c.id=b.batchid) where sq.marks > all (select sq.marks from course\_batches c1 join batch\_students b1 on (c1.Id=b1.studentid) join student s1 on (s1.id=b1.studentid) join student\_qualifications sq on(s.id=sq.studentid) join where s1.namefirst='saleel'and sq.name=10 ); |
|  |
| 1. Display students whose DOB is as same as ‘kaushal’ |
| select \* from student s join student s1 on (s.id=s1.id) where s.dob =(select s.dob from student s where s.namefirst='kaushal') ; |
|  |
| 1. Display all student details who have three or more phone numbers. |
| select \* from student where id in(select studentid from student\_phone group by studentid having count(\*) > 3); |
|  |
| 1. Display marks for the studentID 1 and 7 who have done ‘BE’. (Note: the marks must be displayed side by side). |
| select (select sq.marks from student s join student\_qualifications sq on(s.Id=sq.studentid) where s.id=1 and sq.name='be' ) studentid1 ,(select sq.marks from student s join student\_qualifications sq on(s.Id=sq.studentid) where s.id=7 and sq.name='be') studentid2 from student s join student\_qualifications sq on(s.Id=sq.studentid) limit 1; |
|  |
| 1. 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) |
| select (select sq.marks from student s join student\_qualifications sq on(s.Id=sq.studentid) where s.id=1 and sq.name='be' ) studentid1 ,(select sq.marks from student s join student\_qualifications sq on(s.Id=sq.studentid) where s.id=7 and sq.name='be') studentid2,(select sq.marks from student s join student\_qualifications sq on(s.Id=sq.studentid) where s.id=7 and sq.name='be' ) -(select sq.marks from student s join student\_qualifications sq on(s.Id=sq.studentid) where s.id=1 and sq.name='be') diff from student s join student\_qualifications sq on(s.Id=sq.studentid) limit 1; |
|  |
| 1. Display all student who are not joined any of the batch. |
| select \* from student s where s.id not in (select s.id from student s join batch\_students b on(s.Id=b.studentid) join course\_batches c on(c.id=b.batchid)); |
|  |
| 1. Display all course\_batches details who are starting on the same day as ‘Batch1’. |
| select \* from course c join course\_batches cb on(c.id=cb.courseid) where cb.starton in (select cb.starton from course\_batches cb where cb.name='batch1'); |
|  |
| 1. Display all students whose 10th marks is more than student ‘Neel’s 10th marks. |
| select s.namefirst,sq.marks,sq.name from student s join student\_qualifications sq on(s.id=sq.studentid) where sq.marks >all (select sq.marks from student s join student\_qualifications sq on (s.id=sq.studentid) where s.namefirst='neel' and sq.name='10') and sq.name=10; |
|  |
| 1. Get all student with their qualification details who have highest marks in ‘BE’. |
| select s.namefirst,sq.marks,sq.name from student s join student\_qualifications sq on(s.id=sq.studentid) where sq.marks >all(select sq.marks from student\_qualifications sq where sq.name in('10','12') and sq.studentid=s.id) and sq.name='be' ; |
|  |
| 1. Get all student with their qualification details who have second highest marks in ‘BE’. |
| select s.namefirst,sq.marks,sq.name from student s join student\_qualifications sq on(s.id=sq.studentid) where sq.marks between (select min( sq.marks) from student\_qualifications sq where sq.name in('10','12') and s.id = sq.studentid ) and (select max( sq.marks) from student\_qualifications sq where sq.name in('10','12') and s.id = sq.studentid ) and sq.name='be' ; |
|  |
| 1. Display the student and student\_qualification details who have scored the maximum marks in ‘BE’ |
| select s.namefirst,sq.marks,sq.name from student s join student\_qualifications sq on(s.id=sq.studentid) where sq.marks >all(select sq.marks from student\_qualifications sq where sq.name in('10','12') and sq.studentid=s.id) and sq.name='be' ; |
|  |
| 1. Display the student details who have scored the maximum marks in ‘BE’ |
| select s.namefirst,sq.marks,sq.name from student s join student\_qualifications sq on(s.id=sq.studentid) where sq.marks >all(select sq.marks from student\_qualifications sq where sq.name in('10','12') and sq.studentid=s.id) and sq.name='be' ; |
|  |
| 1. Display the student details who have scored the minimum marks in ‘10’ std. |
| select s.namefirst,sq.name,sq.marks from student s join student\_qualifications sq on(s.id=sq.studentid) where sq.marks <all (select min(sq.marks) from student\_qualifications sq where sq.name ='12' and sq.name='10' )and sq.name='10' ; |
|  |
| 1. Display all student and student\_qualification details of those students who have scored marks more than ‘RAJAN’ in ‘BE’. |
| select s.namefirst,sq.marks,sq.name from student s join student\_qualifications sq on(s.id=sq.studentid) where sq.marks >all (select sq.marks from student s join student\_qualifications sq on (s.id=sq.studentid) where s.namefirst='rajan' and sq.name='be') ; |
|  |
| 1. Display all student who have done ‘BE’ in the same year as of studentID 16. |
| select \* from student s join student\_qualifications sq on(s.Id=sq.studentid) where sq.year = (select sq.year from student\_qualifications sq join student s on(s.id=sq.studentid) where sq.name='be' and s.id=16) and sq.name='be'; |
|  |
| 1. Display all odd records. |
|  |
|  |
| 1. Calculate the sum of marks student wise of their qualifications (i.e. 10th, 12th and BE marks) |
| select s.namefirst,group\_concat(sq.name),group\_concat(sq.marks),sum(sq.marks) from student s join student\_qualifications sq on(s.id=sq.studentid) group by (s.namefirst) ; |
|  |
| 1. Display students’ details who are not having ***'Aadhaar'*** card. |
| select \* from student s join student\_cards sc on(s.Id=sc.studentid) where sc.name not in (select sc.name from student\_cards sc join student s on(s.id=sc.studentid) where sc.name='aadhaar'); |
|  |