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 namefirst, count(name) from (select s.namefirst , cb.name from student s join batch\_students bs join course\_batches cb where s.id=bs.studentid and bs.batchid=cb.id) e group by namefirst having count(name)>2; |
|  |
| 1. Display the student detail who have joined the same batch of the student ‘saleel’. |
| select \* from student s join batch\_students bs where s.id=bs.studentid and batchid in (select cb.id from student s join batch\_students bs join course\_batches cb where s.id=bs.studentid and bs.batchid=cb.id and namefirst='ram') ; |
|  |
| 1. Display all courses where least number of students have taken the admission. |
| select \* from ( select cs.name,count(cs.id) r1 from course cs join course\_batches cb join batch\_students bs join student s where cs.id=cb.courseid and cb.id=bs.batchid and bs.studentid=s.id group by cs.name) e order by r1 limit 2; |
|  |
| 1. Display student details who have not taken the admission. |
| select s.id,s.namefirst from student s where s.id not in (select s.id from course cs join course\_batches cb join batch\_students bs join student s where cs.id=cb.courseid and cb.id=bs.batchid and bs.studentid=s.id); |
|  |
| 1. Get all courses where no modules are defined in course\_modules table. |
| select c.name from course c where c.name not in(select c.name from course c join course\_modules cm join modules m where c.id= cm.courseid and cm.moduleid=m.id) ; |
|  |
| 1. Display course*\_batches* details where student has taken the admission. |
| select cs.name from course\_batches cs where cs.id in (select s.id from course cs join course\_batches cb join batch\_students bs join student s where cs.id=cb.courseid and cb.id=bs.batchid and bs.studentid=s.id); |
|  |
| 1. Display all students whose marks of ‘BE’ is more than ‘ULKA’ marks in ‘BE’. |
| select \* from (select s.namefirst,sq.marks r,sq.name from student s join student\_qualifications sq where s.id=sq.studentid and sq.name='be') e where r>67;  select s.namefirst,sq.marks,sq.name from student s join student\_qualifications sq where s.id=sq.studentid and marks>(select sq.marks from student s join student\_qualifications sq where s.id =sq.studentid and sq.name='be' and s.namefirst='ulka') and sq.name='be'; |
|  |
| 1. Display all students whose marks are more than ‘saleel’ marks in 10th std. |
| select s.namefirst,sq.marks,sq.name from student s join student\_qualifications sq where s.id=sq.studentid and marks>(select sq.marks from student s join student\_qualifications sq where s.id =sq.studentid and sq.name='10' and s.namefirst='ram') and sq.name='10'; |
|  |
| 1. Display students whose DOB is as same as ‘kaushal’ |
| select namefirst,dob from student where dob =(select dob from student where namefirst='kaushal'); |
|  |
| 1. Display all student details who have three or more phone numbers. |
| select \* from (select s.namefirst from student s join student\_phone sp where s.id =sp.studentid group by s.id having count(sp.number)>3 ) q; |
|  |
| 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 where s.id=sq.studentid and sq.name='be' and s.id=1) studentid1, (select sq.marks from student s join student\_qualifications sq where s.id=sq.studentid and sq.name='be' and s.id=7) studentid7 ; |
|  |
| 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 where s.id=sq.studentid and sq.name='be' and s.id=1) "studentid1" , (select sq.marks from student s join student\_qualifications sq where s.id=sq.studentid and sq.name='be' and s.id=7) "studentid7" , (select studentid7-studentid1) 'mark difference'; |
|  |
| 1. Display all student who are not joined any of the batch. |
| select s.\* from student s where s.id not in(select s.id from student s join batch\_students bs join course\_batches cb where s.id=bs.studentid and bs.batchid=cb.id); |
|  |
| 1. Display all course\_batches details who are starting on the same day as ‘Batch1’. |
| select cb.name from course\_batches cb where cb.starton =(select starton from course\_batches where 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 where s.id=sq.studentid and marks>(select sq.marks from student s join student\_qualifications sq where s.id =sq.studentid and sq.name='10' and s.namefirst='neel') and sq.name='10'; |
|  |
| 1. Get all student with their qualification details who have highest marks in ‘BE’. |
| select s.\* from student s join student\_qualifications sq where s.id=sq.studentid and marks =(select max(sq.marks) from student s join student\_qualifications sq where s.id = sq.studentid and sq.name='be'); |
|  |
| 1. Get all student with their qualification details who have second highest marks in ‘BE’. |
| select \* from (select s.\*, dense\_rank() over(order by sq.marks desc) r1,sq.marks from student s join student\_qualifications sq where s.id = sq.studentid and sq.name='be')e where r1=2 ; |
|  |
| 1. Display the student and student\_qualification details who have scored the maximum marks in ‘BE’ |
| select s.\* ,sq.\* from student s join student\_qualifications sq where s.id=sq.studentid and marks =(select max(sq.marks) from student s join student\_qualifications sq where s.id = sq.studentid and sq.name='be'); |
|  |
| 1. Display the student details who have scored the maximum marks in ‘BE’ |
| select s.\* from student s join student\_qualifications sq where s.id=sq.studentid and marks =(select max(sq.marks) from student s join student\_qualifications sq where s.id = sq.studentid and sq.name='be'); |
|  |
| 1. Display the student details who have scored the minimum marks in ‘10’ std. |
| select s.\* from student s join student\_qualifications sq where s.id=sq.studentid and marks =(select min(sq.marks) from student s join student\_qualifications sq where s.id = sq.studentid 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\*,sq.\* from student s join student\_qualifications sq where s.id=sq.studentid and marks>(select sq.marks from student s join student\_qualifications sq where s.id =sq.studentid and sq.name='be' and s.namefirst='rajan') and sq.name='be'; |
|  |
| 1. Display all student who have done ‘BE’ in the same year as of studentID 16. |
| select s.\*,sq.year,sq.name from student s join student\_qualifications sq where s.id=sq.studentid and sq.year=(select sq.year from student\_qualifications sq where sq.studentid=16 and name='be') and sq.name='be'; |
|  |
| 1. Display all odd records. |
| select \* from (select \* from student where id%2=1) e; |
|  |
| 1. Calculate the sum of marks student wise of their qualifications (i.e. 10th, 12th and BE marks) |
| select \* from ( select s.\*, sum( sq.marks) from student s join student\_qualifications sq where s.id=sq.studentid and name in('10','12','be') group by s.id)q; |
|  |
| 1. Display students’ details who are not having ***'Aadhaar'*** card. |
| select s.\*, group\_concat(sc.name) from student s join student\_cards sc where s.id =sc.studentid and s.id not in (select s.id from student join student\_cards sc where s.id=sc.studentid and sc.name='aadhaar' ) group by s.id; |
|  |