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 namefirst,name from student s join course\_batches cb join batch\_students bs where s.ID=bs.studentID and cb.id=bs.batchID) e group by namefirst having count(name)>2; |
|  |
| 1. Display the student detail who have joined the same batch of the student ‘saleel’. |
| select namefirst from student s join batch\_students bs where bs.studentid=s.id and batchID in(select batchID from student s join course\_batches cb join batch\_students bs where s.id=bs.studentID and cb.id=bs.batchID and namefirst="saleel") ; |
|  |
| 1. Display all courses where least number of students have taken the admission. |
| select distinct course.name from course join(select \* from course\_batches join (Select batchid from (Select batchid,count(\*) count from batch\_students group by batchid) a join (select min(b.count) ct from (Select batchid,count(\*) count from batch\_students group by batchid) b) c where a.count=c.ct) t1 where course\_batches.id=t1.batchid) t2 where course.id=t2.courseid; |
|  |
| 1. Display student details who have not taken the admission. |
| select namefirst from student s where s.id not in (select distinct s.id from student s join course\_batches cb join course c join batch\_students bs where s.id=bs.studentid and bs.batchid=cb.id and cb.courseid=c.id order by s.id); |
|  |
| 1. Get all courses where no modules are defined in course\_modules table. |
| select name from course where 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 cb.\* from student s join batch\_students bs join course\_batches cb where bs.studentID=s.id and bs.batchID=cb.id; |
|  |
| 1. Display all students whose marks of ‘BE’ is more than ‘ULKA’ marks in ‘BE’. |
|  |
|  |
| 1. Display all students whose marks are more than ‘saleel’ marks in 10th std. |
| select namefirst, marks from student s join student\_qualifications sq where s.id=sq.studentid and name='10' and marks>(select marks from student s join student\_qualifications sq where s.id=sq.studentid and namefirst='saleel' and sq.name='10'); |
|  |
| 1. Display students whose DOB is as same as ‘kaushal’ |
| select namefirst, dob from student s where dob=(select dob from student where namefirst='kaushal'); |
|  |
| 1. Display all student details who have three or more phone numbers. |
| select namefirst, count(p.number) from student s join student\_phone p where s.id=p.studentid group by namefirst having count(p.number)>3;  select q1.\* from(select namefirst, count(p.number) 'pp' from student s join student\_phone p where s.id=p.studentid group by namefirst)q1 having pp>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 marks from student s join student\_qualifications sq where s.id=studentid and studentid=1 and name='BE')’StudentID 1’, (select marks from student s join student\_qualifications sq where s.id=studentid and studentid=7 and name='BE')’StudentID 7’; |
|  |
| 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) |
|  |
|  |
| 1. Display all student who are not joined any of the batch. |
| select s.id,namefirst from student s where s.id not in(select s.id from student s join course\_batches cb join batch\_students bs 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 \* from course\_batches where 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 namefirst, marks from student s join student\_qualifications sq where s.id=sq.studentID and sq.name=10 and marks > (select marks from student s join student\_qualifications sq where s.id=sq.studentID and sq.name=10 and s.namefirst="neel"); |
|  |
| 1. Get all student with their qualification details who have highest marks in ‘BE’. |
| select namefirst, sq.\* from student s join student\_qualifications sq where s.id=sq.studentid and name='be' and marks=(select max(marks) from student\_qualifications sq where name='BE'); |
|  |
| 1. Get all student with their qualification details who have second highest marks in ‘BE’. |
| select namefirst, sq.\* from student s join student\_qualifications sq where s.id=sq.studentid and name='be' and marks<(select max(marks) from student\_qualifications sq where name='be')order by marks desc limit 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 name='be' and marks=(select max(marks) from student\_qualifications sq where 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 name='be' and marks=(select max(marks) from student\_qualifications sq where 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(marks) from student\_qualifications where name='10'); |
|  |
| 1. Display all student and student\_qualification details of those students who have scored marks more than ‘RAJAN’ in ‘BE’. |
| select namefirst, sq.\* from student s join student\_qualifications sq where s.id=sq.studentid and name='BE' and marks>(select marks from student s join student\_qualifications sq where s.id=sq.studentid and name='BE' and namefirst='Rajan');  Formatted:  select s.id, namefirst, name, college, university, marks, year from student s join student\_qualifications sq where s.id=sq.studentid and name='BE' and marks>(select marks from student s join student\_qualifications sq where s.id=sq.studentid and name='BE' and namefirst='Rajan')order by studentID; |
|  |
| 1. Display all student who have done ‘BE’ in the same year as of studentID 16. |
| select namefirst, sq.\* from student s join student\_qualifications sq where s.id=sq.studentid and year=(select year from student s join student\_qualifications sq where s.id=sq.studentid and name='BE' and s.id=16)order by s.id; |
|  |
| 1. Display all odd records. |
| select \* from (select row\_number() over() 'RN', namefirst, namelast, dob, emailid from student)q1 where q1.rn%2=1; |
|  |
| 1. Calculate the sum of marks student wise of their qualifications (i.e. 10th, 12th and BE marks) |
| select id, n, m from(select studentID 'id', group\_concat(name) 'n', sum(marks) 'm' from student\_qualifications where name in (10,12,'BE') group by studentID)q1;  select ID, NAME, SUMMATION from (select studentID 'ID', group\_concat(name) 'NAME', sum(marks) 'SUMMATION' from student\_qualifications group by studentID)q1;  SINGLE: select studentID, group\_concat(name), sum(marks) from student\_qualifications group by studentID; |
|  |
| 1. Display students’ details who are not having ***'Aadhaar'*** card. |
| select studentID, namefirst, group\_concat(name) from student s join student\_cards c where s.id=c.studentID and name not in ('Aadhaar')group by namefirst, studentID order by studentID; |
|  |