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 q1.namefirst, q1.studentID, count(\*) from(select namefirst, bs.studentID, bs.batchID from student s join batch\_students bs where bs.studentID=s.id order by bs.studentID)q1 group by q1.namefirst, q1.studentid; |
|  |
| 1. Display the student detail who have joined the same batch of the student ‘saleel’. |
| select \* from (select s.\* from student s join batch\_students bs where s.id=bs.studentid and bs.batchid in (1,6) group by s.id order by s.id)b; |
|  |
| 1. Display all courses where least number of students have taken the admission. |
| select z as id, x as course\_name, student\_count from (select c.id as z, c.name as x, count(bs.studentid) 'student\_count' from course c join course\_batches cb join batch\_students bs where c.id=cb.courseid and cb.id=bs.batchid group by c.name,c.id) r group by z, x having student\_count = (select min(student\_count) from (select c.id as z, c.name as x, count(bs.studentid) 'student\_count' from course c join course\_batches cb join batch\_students bs where c.id=cb.courseid and cb.id=bs.batchid group by c.name,c.id)p); |
|  |
| 1. Display student details who have not taken the admission. |
| SELECT \* from student where id not in (select studentid from batch\_students); |
|  |
| 1. Get all courses where no modules are defined in course\_modules table. |
| select \* from course where id not in (select courseid from course\_modules); |
|  |
| 1. Display course*\_batches* details where student has taken the admission. |
| select A as Studentid, B as firstname, K as batchname, l as courseid, p as startson, q as endson, u as capacity from (select s.id 'A', s.namefirst 'B',cb.name 'K', cb.courseid 'l', cb.starton 'q', cb.endson 'p', cb.capacity 'u' from course\_batches cb join batch\_students bs join student s where s.id=bs.studentid and bs.batchid=cb.id order by s.id) j; |
|  |
| 1. Display all students whose marks of ‘BE’ is more than ‘ULKA’ marks in ‘BE’. |
| select s.\*, sq.marks from student s join student\_qualifications sq where s.id=sq.studentid and marks > (select marks from student\_qualifications where studentid=3 and name = 'BE') and sq.name='BE'; |
|  |
| 1. Display all students whose marks are more than ‘saleel’ marks in 10th std. |
| select s.\*, sq.marks, sq.name from student s join student\_qualifications sq where s.id=sq.studentid and marks > (select marks from student\_qualifications where studentid=1 and name = '10') and sq.name='10’; |
|  |
| 1. Display students whose DOB is as same as ‘kaushal’ |
| select \* from student where date(dob) = (select dob from student where namefirst='kaushal'); |
|  |
| 1. Display all student details who have three or more phone numbers. |
| select a as studentid,b as firstname,c as lastname, d as DOB, e as email\_id, f as 'No. of phones' from (select s.id 'a', s.namefirst 'b', s.namelast 'c', s.dob 'd', s.emailid 'e', count(distinct number) 'f' from student s join student\_phone p where s.id = p.studentid group by s.id) v having f >=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\_qualifications where studentid = 1 and name='BE') 'StudentID 1',(select marks from student\_qualifications where 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) |
| select (select marks from student\_qualifications where studentid = 1 and name='BE') 'StudentID 1',(select marks from student\_qualifications where studentid = 7 and name='BE') 'StudentID 7', abs((select marks from student\_qualifications where studentid = 1 and name='BE') - (select marks from student\_qualifications where studentid = 7 and name='BE')) as Difference; |
|  |
| 1. Display all student who are not joined any of the batch. |
| SELECT \* from student where id not in (select studentid from batch\_students); |
|  |
| 1. Display all course\_batches details who are starting on the same day as ‘Batch1’. |
| select \* from (select \* from course\_batches where date(starton)= (select starton from course\_batches where name='Batch1'))v; |
|  |
| 1. Display all students whose 10th marks is more than student ‘Neel’s 10th marks. |
| select s.\*, sq.name, sq.marks from student s join student\_qualifications sq where s.id= sq.studentid and marks > (select sq.marks from student\_qualifications sq join student s where s.id=sq.studentid and s.namefirst='neel' and sq.name=10) having sq.name=10; |
|  |
| 1. Get all student with their qualification details who have highest marks in ‘BE’. |
| select \* from (select s.\* ,sq.name, sq.college, sq.university, sq.marks 'K',sq.year from student s join student\_qualifications sq where s.id=sq.studentid and name = 'BE' and sq.marks = (select max(marks) from student\_qualifications where name=’BE’)) v ; |
|  |
| 1. Get all student with their qualification details who have second highest marks in ‘BE’. |
| select s.\*, sq.name, sq.marks as 'k' from student s join student\_qualifications sq on s.id = sq.studentid where sq.name = 'BE' and sq.marks = (select max(marks) from student\_qualifications where name = 'BE' and marks < (select max(marks) from student\_qualifications where name = 'BE')); |
|  |
| 1. Display the student and student\_qualification details who have scored the maximum marks in ‘BE’ |
| select \* from (select s.\* ,sq.name, sq.marks 'K' from student s join student\_qualifications sq where s.id=sq.studentid and name = 'BE' and sq.marks = (SELECT MAX(marks) FROM student\_qualifications WHERE name = 'BE')) v; |
|  |
| 1. Display the student details who have scored the maximum marks in ‘BE’ |
| select \* from (select s.\* ,sq.name, sq.marks 'K' from student s join student\_qualifications sq where s.id=sq.studentid and name = 'BE' and sq.marks = (SELECT MAX(marks) FROM student\_qualifications WHERE name = 'BE')) v; |
|  |
| 1. Display the student details who have scored the minimum marks in ‘10’ std. |
| select \* from (select s.\* ,sq.name, sq.marks 'K' from student s join student\_qualifications sq where s.id=sq.studentid and name = '10' and sq.marks = (SELECT MAX(marks) FROM student\_qualifications WHERE name = '10')) v; |
|  |
| 1. Display all student and student\_qualification details of those students who have scored marks more than ‘RAJAN’ in ‘BE’. |
| select s.\*, sq.name, sq.marks from student s join student\_qualifications sq where s.id= sq.studentid and marks > (select sq.marks from student\_qualifications sq join student s where s.id=sq.studentid and s.namefirst='RAJAN' and sq.name='BE') having sq.name='BE'; |
|  |
| 1. Display all student who have done ‘BE’ in the same year as of studentID 16. |
| select s.\*, sq.name, sq.marks from student s join student\_qualifications sq where s.id= sq.studentid and date(sq.year) = (select year from student\_qualifications where studentid=16 and name = 'BE') having sq.name='BE'; |
|  |
| 1. Display all odd records. |
| select \* from (select \* from student where id%2 != 0) v; |
|  |
| 1. Calculate the sum of marks student wise of their qualifications (i.e. 10th, 12th and BE marks) |
| select \* from (select sq.studentid,s.namefirst,s.namelast, group\_concat(sq.name),sum(sq.marks) from student\_qualifications sq join student s where s.id=sq.studentid group by sq.studentid) v; |
|  |
| 1. Display students’ details who are not having ***'Aadhaar'*** card. |
| select \* from student where id not in (select studentid from student\_cards where name='Aadhaar'); |
|  |