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 \* from student where id in (select studentid from batch\_students group by studentid having count(\*) > 2); |
|  |
| 1. Display the student detail who have joined the same batch of the student ‘saleel’. |
| select \* from student where id in(select studentid from batch\_students where batchid in(select batchid from batch\_students where studentid= (select id from student where namefirst='saleel'))); |
|  |
| 1. Display all courses where least number of students have taken the admission. |
| select min(f.noOfStudents) m from (select c.id srno, c.name course\_name, count(distinct studentid) noOfstudents from course c, course\_batches cb, batch\_students bs where c.id = cb.courseid and cb.id =bs.batchid group by c.name, c.id) f, course c where c.name =f.course\_name and f.srno = c.id; |
|  |
| 1. Display student details who have not taken the admission. |
| select \* from student where id not in (select distinct c.id from course\_batches cb, batch\_students bs, course c,student s where cb.id = bs.batchid and courseid = c.id and s.id = bs.studentid); |
|  |
| 1. Get all courses where no modules are defined in course\_modules table. |
| select \* from course c where c.id not in (select distinct cm.courseid from modules m , course\_modules cm where m.id = cm.moduleid order by cm.courseid); |
|  |
| 1. Display course*\_batches* details where student has taken the admission. |
| select \* from course\_batches cb where cb.id in (select bs.batchid from batch\_students bs); |
|  |
| 1. Display all students whose marks of ‘BE’ is more than ‘ULKA’ marks in ‘BE’. |
| select row\_number() over() 'SR No', namefirst, namelast, marks from student s, student\_qualifications sq where s.id = sq.studentid and sq.name = 'BE' having sq.marks > (select marks from student s, student\_qualifications sq where namefirst='ulka' and name='BE' and sq.studentid = s.id); |
|  |
| 1. Display all students whose marks are more than ‘saleel’ marks in 10th std. |
| select row\_number() over() 'SR No', namefirst, namelast, marks from student s, student\_qualifications sq where s.id = sq.studentid and sq.name = '10' having sq.marks > (select marks from student s, student\_qualifications sq where namefirst='saleel' and name = '10' and s.id = sq.studentid); |
|  |
| 1. Display students whose DOB is as same as ‘kaushal’ |
| select \* from student where dob in (select DOB from student where namefirst='kaushal'); |
|  |
| 1. Display all student details who have three or more phone numbers. |
| select distinct namefirst, namelast, emailid, dob, sa.address from student s, student\_qualifications sq, student\_phone sp, student\_address sa where s.id in (select distinct studentid from student\_phone group by studentid having count(number) > 2) and s.id = sq.studentid and s.id = sp.studentid and s.id = sa.studentid; |
|  |
| 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 name='BE' and studentid = 1) 'StudentId 1', (select marks from student\_qualifications where name='BE' and studentid = 7) '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 name='BE' and studentid = 1) 'StudentId 1', (select marks from student\_qualifications where name='BE' and studentid = 7) 'StudentId 7', abs((select marks from student\_qualifications where name='BE' and studentid = 1)- (select marks from student\_qualifications where name='BE' and studentid = 7)) 'Marks Difference'; |
|  |
| 1. Display all student who are not joined any of the batch. |
| select \* from student where id not in (select distinct studentid from batch\_students); |
|  |
| 1. Display all course\_batches details who are starting on the same day as ‘Batch1’. |
| select \* from course\_batches where starton in (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, namelast, emailid, marks from student s, student\_qualifications sq where s.id = sq.studentid and marks > (select marks from student s, student\_qualifications sq where namefirst='neel' and s.id = sq.studentid and name='10') and name = '10' order by marks; |
|  |
| 1. Get all student with their qualification details who have highest marks in ‘BE’. |
| select s.id, namefirst, namelast, emailid, dob, college, university from student s, student\_qualifications sq where s.id in (select studentid from student\_qualifications where marks = (select max(marks) from student\_qualifications where name='BE') and name='BE') and s.id = sq.studentid and sq.name='BE'; |
|  |
| 1. Get all student with their qualification details who have second highest marks in ‘BE’. |
|  |
|  |
| 1. Display the student and student\_qualification details who have scored the maximum marks in ‘BE’ |
| select s.id, namefirst, namelast, emailid, dob, college, university from student s, student\_qualifications sq where s.id in (select studentid from student\_qualifications where marks = (select max(marks) from student\_qualifications where name='BE') and name='BE') and 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, student\_qualifications sq where s.id in (select studentid from student\_qualifications where marks = (select max(marks) from student\_qualifications where name='BE') and name='BE') and s.id = sq.studentid and sq.name='BE'; |
|  |
| 1. Display the student details who have scored the minimum marks in ‘10’ std. |
| select \* from student s, student\_qualifications sq where marks = (select min(marks) from student\_qualifications where name='10') and s.id=sq.studentid and 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, student\_qualifications sq where marks > (select marks from student\_qualifications sq, student s where s.id = sq.studentid and name='BE' and namefirst='rajan') and name='BE' and s.id = sq.studentid; |
|  |
| 1. Display all student who have done ‘BE’ in the same year as of studentID 16. |
| select s.\* from student s, student\_qualifications sq where s.id = sq.studentid and year = (select year from student\_qualifications where name='BE' and studentid = 16) and 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 student s, (select studentid,sum(marks) from student\_qualifications group by studentid) t where s.id = t.studentid; |
|  |
| 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'); |
|  |