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 ID, Namefirst, Namelast from student s where s.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 s.id StudentID, Namefirst, Namelast, DOB, EmailID from student s where id in(select studentid from batch\_students where batchid in (select batchid from batch\_students where studentid in (select id from student where namefirst = 'saleel' ) ) )order by s.id; |
|  |
| 1. Display all courses where least number of students have taken the admission. |
| **select name CourseName from course where id in (select courseid from course\_batches cb join batch\_students bs where cb.id in (select bs.batchid from batch\_students) group by courseid having count(courseid) = (select min (c) from (select count(courseid) c from course\_batches cb join batch\_students bs where cb.id in(select bs.batchid from batch\_students) group by courseid) c ) );** |
|  |
| 1. Display student details who have not taken the admission. |
| select s.id StudentID, Namefirst, Namelast, DOB, EmailID from student s join batch\_students bs where s.id not in (select bs.studentid from batch\_students bs) group by s.id order by s.id; |
|  |
| 1. Get all courses where no modules are defined in course\_modules table. |
| select c.name CourseName from course c where name not in(select name from course where c.id in(select courseid from course\_modules group by courseid ) ); |
|  |
| 1. Display course*\_batches* details where student has taken the admission. |
| **BATCH DETAILS WITH STUDENTID :**  select s.id StudentID, bs.batchid BatchID, name BatchNumber, cb.courseid CourseID, StartOn, EndsOn, Capacity from student s join batch\_students bs join course\_batches cb where s.id in (select bs.studentid from batch\_students) and bs.batchid in ( select cb.id from course\_batches) order by s.id;  **FOR ONLY DISTINCT BATCH DETAILS :**  select ID, CourseID, StartOn, EndsOn, Capacity from course\_batches where id in (select batchid from batch\_students where studentid in (select id from student ) ); |
|  |
| 1. Display all students whose marks of ‘BE’ is more than ‘ULKA’ marks in ‘BE’. |
| select ID, Namefirst, Namelast, DOB, EmailID from student where id in (select studentid from student\_qualifications where marks > (select marks from student\_qualifications where studentid = (select id from student where namefirst = 'ulka') and name = 'BE' ) and name = 'BE' ); |
|  |
| 1. Display all students whose marks are more than ‘saleel’ marks in 10th std. |
| select ID, Namefirst, Namelast, DOB, EmailID from student where id in (select studentid from student\_qualifications where marks > (select marks from student\_qualifications where studentid = (select id from student where namefirst = 'saleel') and name = 10 ) and name = 10 ); |
|  |
| 1. Display students whose DOB is as same as ‘kaushal’ |
| select ID, Namefirst, Namelast, DOB, EmailID 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 ID, Namefirst, Namelast, DOB, EmailID from student where id in (select studentid from student\_phone group by studentid having count(number)>=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 name = 'BE' and studentid = 1) as 'StudentId 1', (select marks from student\_qualifications where name = 'BE' and studentid = 7) as '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) as 'StudentId\_1', (select marks from student\_qualifications where name = 'BE' and studentid = 7) as 'StudentId\_7', (select abs(StudentId\_1 - StudentId\_7)) as 'Marks Difference'; |
|  |
| 1. Display all student who are not joined any of the batch. |
| select ID, Namefirst, Namelast, DOB, EmailID from student where id not in(select studentid from batch\_students order by studentid); |
|  |
| 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 s.id StudentID, Namefirst, Namelast, DOB, EmailID, Name, Marks from student s join student\_qualifications sq on s.id = sq.studentid and marks > (select marks from student\_qualifications where name = '10' and studentid = (select id from student where namefirst = 'neel') and name = '10')and name = '10' ; |
|  |
| 1. Get all student with their qualification details who have highest marks in ‘BE’. |
| select s.id ID, Namefirst, Namelast, DOB, EmailID, name Degree, College, University, sq.marks Marks, Year from student s join student\_qualifications sq on s.id = sq.studentid and marks = (select max(marks) from student\_qualifications where marks in (select marks from student\_qualifications where name = 'BE' )) and name = 'BE'; |
|  |
| 1. Get all student with their qualification details who have second highest marks in ‘BE’. |
| sselect s.id ID, Namefirst, Namelast, name Degree, College, University, sq.marks Marks, Year from student s join student\_qualifications sq on s.id = sq.studentid where marks = (select max(marks) from student\_qualifications where marks < (select max(marks) from student\_qualifications where name = 'BE' ) and name = 'BE') and name = 'BE' ; |
|  |
| 1. Display the student and student\_qualification details who have scored the maximum marks in ‘BE’ |
| select s.id ID, Namefirst, Namelast, DOB, EmailID, name Degree, College, University, sq.marks Marks, Year from student s join student\_qualifications sq on s.id = sq.studentid and marks = (select max(marks) from student\_qualifications where marks in (select marks from student\_qualifications where name = 'BE' )) and name = 'BE' ; |
|  |
| 1. Display the student details who have scored the maximum marks in ‘BE’ |
| select s.id ID, Namefirst, Namelast, EmailID, DOB, sq.marks Marks from student s join student\_qualifications sq on s.id = sq.studentid and marks = (select max(marks) from student\_qualifications where marks in (select marks from student\_qualifications where name = 'BE')) and name = 'BE' ; |
|  |
| 1. Display the student details who have scored the minimum marks in ‘10’ std. |
| select s.id ID, Namefirst, Namelast, EmailID, DOB, sq.marks Marks from student s join student\_qualifications sq on s.id = sq.studentid and marks = (select min(marks) from student\_qualifications where marks in (select marks from student\_qualifications where name = 10)) 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.id StudentID, Namefirst, Namelast, Name, University, Year, Marks from student s join student\_qualifications sq on s.id = sq.studentid and marks > (select marks from student\_qualifications where name = 'be' and studentid = (select id from student where namefirst = 'Rajan') and name = BE) and name = BE; |
|  |
| 1. Display all student who have done ‘BE’ in the same year as of studentID 16. |
| select ID, Namefirst, Namelast from student where id in (select studentid from student\_qualifications where year in (select year from student\_qualifications where studentid = 16 and name = 'BE') and name='BE'); |
|  |
| 1. Display all odd records. |
| select \* from (select \* from student where (id%2)=1)s; |
|  |
| 1. Calculate the sum of marks student wise of their qualifications (i.e. 10th, 12th and BE marks) |
| select s.id ID, s.namefirst Namefirst, group\_concat(name) Name, group\_concat(marks) Marks, sum(marks) MarksSum from student\_qualifications sq join student s where sq.studentid  in(select id from student) and s.id=sq.studentid group by studentid; |
|  |
| 1. Display students’ details who are not having ***'Aadhaar'*** card. |
| select ID, Namefirst, Namelast, DOB, EmailID from student where id not in (select studentid from student\_cards where name = 'Aadhaar'); |
|  |

**HELP :**

**Question 3 tricked me for a long time (4-5 hours in total). If you are smart, congaratulations you will save your time.**

**Question number 7, 8, 15, 21 are similar.**

**Question number 16, 18, 19 are same.**

**Question no. 17, 18, 19, 20 are based on min/max .**

**I took 11 hours in total.**

**NOTE :**

**Orange colors are for me.. I am not satisfied with that output.**

**TODO : I want to get all student who individualy have his best score in ‘BE’.**