Assignment

Aug24/ DBT/ 011

Database Technologies

Diploma in Advance Computing

August 2024

**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 studentID,count(\*) batch\_number from batch\_students group by studentID having count(\*) > 1; |
|  |
| 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 = 1)); |
|  |
| 1. Display all courses where least number of students have taken the admission. |
| select \* from course where ID IN (select courseID from course\_batches where ID IN (select batchID from (select rank() over(order by r1) rank1,batchID from (select batchID,count(\*) r1 from batch\_students group by batchID)T1) T2 where rank1=1)) ; |
|  |
| 1. Display student details who havenot taken the admission. |
| select \* from student where ID NOT IN (select studentID from (select studentID,count(\*) batch\_number from batch\_students group by studentID having count(\*) > 0) T1); |
|  |
| 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 group by courseID) ; |
|  |
| 1. Display course*\_batches*details where student has taken the admission. |
| select \* from course where ID IN (select ID from course where ID NOT IN (select courseID from course\_modules group by courseID)) ; |
|  |
| 1. Display all students whose marks of ‘BE’ is more than ‘ULKA’ marks in ‘BE’. |
| select \* from student\_qualifications where marks > (select marks from student\_qualifications where ID = (select ID from student where namefirst='ULKA')) and name ='BE' ; |
|  |
| 1. Display all students whose marks are more than ‘saleel’ marks in 10th std. |
| select \* from student where ID IN (select ID from student\_qualifications where marks > (select marks from student\_qualifications where ID = (select ID from student where namefirst='saleel'))); |
|  |
| 1. Display students whose DOB is as same as ‘kaushal’ |
| select \* 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 student where ID IN (select studentID from (select studentID,count(\*) R1 from student\_phone group by studentID having R1 >= 3) T1) ; |
|  |
| 1. Display marks for the studentID 1 and 7 who have done ‘BE’. (Note: the marks must be displayed side by side). |
|  |
|  |
| 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 \* from student where ID NOT IN (select distinct studentID batch\_number from batch\_students) ; |
|  |
| 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 12th marks. |
| select \* 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='Neel') and name=10) and name = 10) ; |
|  |
| 1. Get all student with their qualification details who have highest marks in ‘BE’. |
| select \* from student where ID IN (select studentId from (select rank() over(order by marks desc) R1 ,marks,studentID from student\_qualifications where name = 'BE') T1 where R1 = 1) ; |
|  |
| 1. Get all student with their qualification details who have second highest marks in ‘BE’. |
| select \* from student where ID IN (select studentId from (select dense\_rank() over(order by marks desc) R1 ,marks,studentID from student\_qualifications where name = 'BE') T1 where R1 = 2) ; |
|  |
| 1. Display the student and student\_qualification details who have scored the maximum marks in ‘BE’ |
| select std.namefirst name,std.namelast surname,std.emailID email,sq.name as Edu,sq.year passing,sq.university as university from student std join student\_qualifications sq on std.ID = sq.ID where sq.ID = (select ID from student\_qualifications where name='BE' order by marks desc limit 1) ; |
|  |
| 1. Display the student details who have scored the maximum marks in ‘BE’ |
| select \* from student where ID = (select ID from student\_qualifications where marks = (select max(marks) from student\_qualifications where name='BE') and name='BE' limit 1) ; |
|  |
| 1. Display the student details who have scored the minimum marks in ‘10’ std. |
| select \* from student where ID = (select ID from student\_qualifications where marks =(select min(marks) from student\_qualifications where name='10' and ID between 1 and 28) and name='10' limit 1 ); |
|  |
| 1. Display all student and student\_qualification details of those students who have scored marks more than ‘RAJAN’ in ‘BE’. |
| select \* 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='Rajan') and name='BE') and name='BE'); |
|  |
| 1. Display all student who have done ‘BE’ in the same year as of studentID 16. |
| select \* from student where ID IN (select studentID from student\_qualifications where year = (select year from student\_qualifications where studentID = 16 and name='BE') and name='BE') ; |
|  |
| 1. Display all odd records. |
| select \* from student where ID IN (select \* from (select row\_number()over() h1 from student) T1 where (h1%2)=1) ; |
|  |
| 1. Calculate the sum of marks student wise of their qualifications (i.e. 10th, 12th and BE marks) |
| select studentID,sum(h1) marks from (select studentID,sum(marks) h1 from student\_qualifications where name=10 or name=12 or name='BE' group by ID)T1 group by 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') ; |
|  |