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 Student\_ID FROM Students GROUP BY Student\_ID HAVING COUNT(DISTINCT Batch\_ID) > 2;** |
|  |
| 1. Display the student detail who have joined the same batch of the student ‘saleel’. |
| SELECT \* FROM Students WHERE Batch\_ID IN (SELECT Batch\_ID FROM Students WHERE Student\_Name = 'saleel'); |
|  |
| 1. Display all courses where least number of students have taken the admission. |
| SELECT Course\_ID FROM Courses WHERE Course\_ID NOT IN (SELECT Course\_ID FROM Course\_Students GROUP BY Course\_ID); |
|  |
| 1. Display student details who havenot taken the admission. |
| SELECT \* FROM Students WHERE Student\_ID NOT IN (SELECT Student\_ID FROM Admissions); |
|  |
| 1. Get all courses where no modules are defined in course\_modules table. |
| SELECT Course\_ID FROM Courses WHERE Course\_ID NOT IN (SELECT Course\_ID FROM Course\_Modules); |
|  |
| 1. Display course*\_batches*details where student has taken the admission. |
| SELECT \* FROM Course\_Batches WHERE Batch\_ID IN (SELECT Batch\_ID FROM Admissions); |
|  |
| 1. Display all students whose marks of ‘BE’ is more than ‘ULKA’ marks in ‘BE’. |
| SELECT \* FROM Students WHERE BE\_Marks > (SELECT BE\_Marks FROM Students WHERE Student\_Name = 'ULKA'); |
|  |
| 1. Display all students whose marks are more than ‘saleel’ marks in 10th std. |
| SELECT \* FROM Students WHERE Tenth\_Marks > (SELECT Tenth\_Marks FROM Students WHERE Student\_Name = 'saleel'); |
|  |
| 1. Display students whose DOB is as same as ‘kaushal’ |
| SELECT \* FROM Students WHERE DOB = (SELECT DOB FROM Students WHERE Student\_Name = 'kaushal'); |
|  |
| 1. Display all student details who have three or more phone numbers. |
| SELECT \* FROM Students WHERE Student\_ID IN (SELECT Student\_ID FROM Phone\_Numbers GROUP BY Student\_ID HAVING COUNT(\*) >= 3); |
|  |
| 1. Display marks for the studentID 1 and 7 who have done ‘BE’. (Note: the marks must be displayed side by side). |
| SELECT BE\_Marks AS Marks\_1, (SELECT BE\_Marks FROM Students WHERE Student\_ID = 7) AS Marks\_2 FROM Students WHERE Student\_ID = 1; |
|  |
| 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 BE\_Marks AS Marks\_1, (SELECT BE\_Marks FROM Students WHERE Student\_ID = 7) AS Marks\_2, |
|  |
| 1. Display all student who are not joined any of the batch. |
| ABS((SELECT BE\_Marks FROM Students WHERE Student\_ID = 1) - (SELECT BE\_Marks FROM Students WHERE Student\_ID = 7)) AS Marks\_Difference  FROM Students WHERE Student\_ID IN (1, 7); |
|  |
| 1. Display all course\_batches details who are starting on the same day as ‘Batch1’. |
| SELECT \* FROM Students WHERE Student\_ID NOT IN (SELECT Student\_ID FROM Admissions); |
|  |
| 1. Display all students whose 10th marks is more than student ‘Neel’s 10th marks. |
| SELECT \* FROM Students WHERE Tenth\_Marks > (SELECT Tenth\_Marks FROM Students WHERE Student\_Name = 'Neel'); |
|  |
| 1. Get all student with their qualification details who have highest marks in ‘BE’. |
| SELECT \* FROM Students JOIN Qualifications ON Students.Student\_ID = Qualifications.Student\_ID WHERE BE\_Marks = (SELECT MAX(BE\_Marks) FROM Students); |
|  |
| 1. Get all student with their qualification details who have second highest marks in ‘BE’. |
| SELECT \* FROM Students JOIN Qualifications ON Students.Student\_ID = Qualifications.Student\_ID WHERE BE\_Marks = (SELECT MAX(BE\_Marks) FROM Students WHERE BE\_Marks < (SELECT MAX(BE\_Marks) FROM Students)); |
|  |
| 1. Display the student and student\_qualification details who have scored the maximum marks in ‘BE’ |
| SELECT Students.\*, Qualifications.\* FROM Students JOIN Qualifications ON Students.Student\_ID = Qualifications.Student\_ID WHERE BE\_Marks = (SELECT MAX(BE\_Marks) FROM Students); |
|  |
| 1. Display the student details who have scored the maximum marks in ‘BE’ |
| SELECT \* FROM Students WHERE BE\_Marks = (SELECT MAX(BE\_Marks) FROM Students); |
|  |
| 1. Display the student details who have scored the minimum marks in ‘10’ std. |
| SELECT \* FROM Students WHERE Tenth\_Marks = (SELECT MIN(Tenth\_Marks) FROM Students); |
|  |
| 1. Display all student and student\_qualification details of those students who have scored marks more than ‘RAJAN’ in ‘BE’. |
| SELECT Students.\*, Qualifications.\* FROM Students JOIN Qualifications ON Students.Student\_ID = Qualifications.Student\_ID WHERE BE\_Marks > (SELECT BE\_Marks FROM Students WHERE Student\_Name = 'RAJAN'); |
|  |
| 1. Display all student who have done ‘BE’ in the same year as of studentID 16. |
| SELECT \* FROM Students WHERE Year\_of\_BE = (SELECT Year\_of\_BE FROM Students WHERE Student\_ID = 16); |
|  |
| 1. Display all odd records. |
| SELECT \* FROM Students WHERE MOD(Student\_ID, 2) = 1; |
|  |
| 1. Calculate the sum of marks student wise of their qualifications (i.e. 10th, 12th and BE marks) |
| SELECT Student\_ID, SUM(Marks) AS Total\_Marks FROM (SELECT Student\_ID, Tenth\_Marks AS Marks FROM Students UNION ALL SELECT Student\_ID, Twelfth\_Marks AS Marks FROM Students UNION ALL SELECT Student\_ID, BE\_Marks AS Marks FROM Students) AS All\_Marks GROUP BY Student\_ID; |
|  |
| 1. Display students’ details who are not having ***'Aadhaar'*** card. |
| SELECT \* FROM Students WHERE Aadhaar\_Card IS NULL; |
|  |