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. |
| 1. SELECT \* FROM(SELECT studentID,count(batchID) count FROM batch\_students GROUP BY studentID HAVING count>2)e, student s WHERE s.ID=e.studentID; 2. SELECT \* FROM student s join (SELECT DISTINCT(bs.studentID),count(batchID) R1 FROM student s JOIN batch\_students bs ON s.ID=bs.studentID GROUP BY bs.studentID)e ON s.ID=e.studentID WHERE e.R1>2; |
|  |
| 1. Display the student detail who have joined the same batch of the student ‘saleel’. |
| SELECT s.\*,batchID FROM student s JOIN batch\_students ON s.ID=studentID WHERE batchID in (SELECT batchID FROM student s JOIN batch\_students ON s.ID=studentID WHERE namefirst="saleel"); |
|  |
| 1. Display all courses where least number of students have taken the admission. |
|  |
|  |
| 1. Display student details who have not taken the admission. |
| SELECT \* FROM student WHERE ID not in (SELECT DISTINCT(studentID) FROM student s JOIN batch\_students ON s.ID=studentID); |
|  |
| 1. Get all courses where no modules are defined in course\_modules table. |
| SELECT \* FROM course WHERE ID not in (SELECT DISTINCT(courseID) FROM course\_modules GROUP BY courseID); |
|  |
| 1. Display course*\_batches* details where student has taken the admission. |
| SELECT \* FROM course\_batches WHERE ID in(SELECT DISTINCT(batchID) FROM batch\_students GROUP BY batchID); |
|  |
| 1. Display all students whose marks of ‘BE’ is more than ‘ULKA’ marks in ‘BE’. |
| SELECT s.\*, sq.name, sq.marks FROM student s JOIN student\_qualifications sq ON s.ID=sq.studentID WHERE sq.marks > (SELECT marks FROM student s JOIN student\_qualifications sq ON s.ID=sq.studentID WHERE namefirst="ulka" AND name="BE") AND sq.name="BE"; |
|  |
| 1. Display all students whose marks are more than ‘saleel’ marks in 10th std. |
| SELECT s.\*, sq.name, sq.marks FROM student s JOIN student\_qualifications sq ON s.ID=sq.studentID WHERE sq.marks > (SELECT marks FROM student s JOIN student\_qualifications sq ON s.ID=sq.studentID WHERE namefirst="saleel" AND name="10") AND sq.name="10"; |
|  |
| 1. Display students whose DOB is as same as ‘kaushal’ |
| SELECT \* FROM student WHERE DOB like (SELECT DOB FROM student WHERE namefirst="kaushal"); |
|  |
| 1. Display all student details who have three or more phone numbers. |
| SELECT \* FROM student s JOIN (SELECT DISTINCT(studentID),count(number) count FROM student\_phone GROUP BY studentID HAVING count>=3)e ON s.ID=e.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 studentID=1 AND name="BE") AS "StudentID 1", (SELECT marks FROM student\_qualifications WHERE studentID=7 AND name="BE") 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 t1.marks "StudentID 1",t2.marks "studentID 7",t2.marks-t1.marks "Marks Difference" FROM (SELECT marks FROM student\_qualifications WHERE studentID=1 and name="BE")t1,(SELECT marks FROM student\_qualifications WHERE studentID=7 and name="BE")t2; |
|  |
| 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 GROUP BY studentID); |
|  |
| 1. Display all course\_batches details who are starting on the same day as ‘Batch1’. |
| SELECT \* FROM course\_batches WHERE starton like (SELECT starton FROM course\_batches WHERE name="Batch1"); |
|  |
| 1. Display all students whose 10th marks is more than student ‘Neel’s 10th marks. |
| SELECT \* FROM student\_qualifications WHERE marks > (SELECT marks FROM student\_qualifications WHERE studentID like (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 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"); |
|  |
| 1. Get all student with their qualification details who have second highest marks in ‘BE’. |
| 1. SELECT \* FROM (SELECT DENSE\_RANK() OVER (ORDER BY marks DESC) AS 'Rank', sq.\* FROM student\_qualifications sq HAVING name="BE")e WHERE e.Rank=2; 2. SELECT \* FROM student\_qualifications WHERE marks like (SELECT DISTINCT marks FROM student\_qualifications WHERE name="BE" GROUP BY marks ORDER BY marks DESC LIMIT 1,1) AND name="BE"; |
|  |
| 1. Display the student and student\_qualification details who have scored the maximum marks in ‘BE’ |
| SELECT \* 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"); |
|  |
| 1. Display the student details who have scored the maximum marks in ‘BE’ |
| SELECT \* 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"); |
|  |
| 1. Display the student details who have scored the minimum marks in ‘10’ std. |
| SELECT \* FROM student\_qualifications WHERE marks like (SELECT min(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 \* FROM student s JOIN (SELECT \* FROM student\_qualifications WHERE marks > (SELECT marks FROM student\_qualifications WHERE studentID like (SELECT ID FROM student WHERE namefirst="RAJAN") AND name="BE") AND name="BE")e ON s.ID=e.studentID; |
|  |
| 1. Display all student who have done ‘BE’ in the same year as of studentID 16. |
| SELECT \* FROM student\_qualifications WHERE name="BE" AND year like (SELECT year FROM student\_qualifications WHERE studentID=16 AND name="BE"); |
|  |
| 1. Display all odd records. |
| SELECT \* FROM student WHERE ID not in(SELECT ID FROM student WHERE (ID MOD 2) = 0); |
|  |
| 1. Calculate the sum of marks student wise of their qualifications (i.e. 10th, 12th and BE marks) |
| select studentid, sum(marks) from student\_qualifications where studentid in(select studentid from student\_qualifications where name in(10,12,'BE') group by studentid having count(studentid)=3)group by studentid; |
|  |
| 1. Display students’ details who are not having ***'Aadhaar'*** card. |
| SELECT \* FROM student\_cards WHERE studentID not in (SELECT studentID FROM student\_cards WHERE name="Aadhaar"); |
|  |