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(batchid) > 2); |
|  |
| 1. Display the student detail who have joined the same batch of the student ‘saleel’. |
| select \* from student where id in (select bs.studentid from batch\_students bs where bs.batchid in (select bs.batchid from batch\_students bs, student s where s.id = bs.studentid and s.namefirst = 'saleel')); |
|  |
| 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 batch\_students group by batchid having count(studentid) =(select min(r) from (select batchid r1, count(studentid) r from batch\_students group by batchid having count(studentid))e))); |
|  |
| 1. Display student details who have not taken the admission. |
| select s.namefirst from student s where s.namefirst not in (select s.namefirst from student s join student\_qualifications sq on s.id=sq.studentid ); |
|  |
| 1. Get all courses where no modules are defined in course\_modules table. |
| select \* from course c where c.name not in (select c.name from course c join course\_modules cm on c.id=cm.courseid join modules m on m.id=cm.moduleid ); |
|  |
| 1. Display course*\_batches* details where student has taken the admission. |
| select distinct \* from course\_batches cb join batch\_students bs on cb.courseid=bs.batchid join student s on s.id=bs.studentid where cb.name in (select cb.name from course\_batches cb join batch\_students bs on cb.courseid=bs.batchid join student s on s.id=bs.studentid); |
|  |
| 1. Display all students whose marks of ‘BE’ is more than ‘ULKA’ marks in ‘BE’. |
| select \* from student s where s.id in(select sq.studentid from student\_qualifications sq where sq.marks > (select sq.marks from student\_qualifications sq where sq.studentid = (select s.id from student s where s.namefirst = 'ulka') and sq.name = 'BE') and sq.name = 'BE'); |
|  |
| 1. Display all students whose marks are more than ‘saleel’ marks in 10th std. |
| Select \* from student s where s.id in (Select sq.studentid from student\_qualifications sq where sq.marks > (Select sq.marks from student\_qualifications where sq.studentid = (select s.id from student s where s.namefirst = ‘saleel’) and sq.name = 10) and sq.name =10); |
|  |
| 1. Display students whose DOB is as same as ‘kaushal’ |
| select \* from student where DOB = (select s.DOB from student s where namefirst = "kaushal"); |
|  |
| 1. Display all student details who have three or more phone numbers. |
| select \* from student where id = (select sp.studentid from student\_phone sp group by sp.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 studentid =1 and name = 'BE') studentID\_1, (Select marks from student\_qualifications where studentid =7 and name = 'BE') 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 studentid =1 and name = 'BE') studentID\_1,  (Select marks from student\_qualifications where studentid =7 and name = 'BE') studentID\_7,  (Select abs(studentID\_7- studentID\_1)) Marks\_Difference; |
|  |
| 1. Display all student who are not joined any of the batch. |
| select \* from student where id not in (select bs.studentid from batch\_students bs group by bs.studentid); |
|  |
| 1. Display all course\_batches details who are starting on the same day as ‘Batch1’. |
| select \* from student where id not in (select bs.studentid from batch\_students bs group by bs.studentid); |
|  |
| 1. Display all students whose 10th marks is more than student ‘Neel’s 10th marks. |
| Select \* from student s where s.id in (Select sq.studentid from student\_qualifications sq where sq.marks > (Select sq.marks from student\_qualifications sq where sq.studentid = (select s.id from student s where s.namefirst = "neel") and sq.name = 10) and sq.name =10); |
|  |
| 1. Get all student with their qualification details who have highest marks in ‘BE’. |
| select s.\* , sq.\* from student s , student\_qualifications sq where s.id in(select sq.studentid from student\_qualifications sq where sq.marks = (select max(sq.marks) from student\_qualifications sq where sq.name = 'BE') and sq.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’. |
| select s.\*, sq.\* from student s, student\_qualifications sq where sq.studentid in (select sq.studentid from student\_qualifications sq where sq.marks = (select max(sq.marks) from student\_qualifications sq where sq.marks < (select max(sq.marks) from student\_qualifications sq where sq.name="BE") and sq.name = 'BE') and sq.name = 'BE') and sq.studentid = s.id and sq.name = 'BE'; |
|  |
| 1. Display the student and student\_qualification details who have scored the maximum marks in ‘BE’ |
| select s.\* , sq.\* from student s , student\_qualifications sq where s.id in(select sq.studentid from student\_qualifications sq where sq.marks = (select max(sq.marks) from student\_qualifications sq where sq.name = 'BE') and sq.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.\* , sq.\* from student s , student\_qualifications sq where s.id in(select sq.studentid from student\_qualifications sq where sq.marks = (select max(sq.marks) from student\_qualifications sq where sq.name = 'BE') and sq.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 s.\* , sq.\* from student s , student\_qualifications sq where s.id in(select sq.studentid from student\_qualifications sq where sq.marks = (select min(sq.marks) from student\_qualifications sq where sq.name = 10) and sq.name = 10) and s.id = sq.studentid and sq.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\_qualifications where studentid 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')and name = 'BE'; |
|  |
| 1. Display all student who have done ‘BE’ in the same year as of studentID 16. |
| select \* from student\_qualifications where studentid in (select studentid from student\_qualifications where year = (select year from student\_qualifications where studentid = 16 and name = 'BE') and name = 'BE') and name = 'BE'; |
|  |
| 1. Display all odd records. |
| select s.\* from student s where s.id%2 =1; |
|  |
| 1. Calculate the sum of marks student wise of their qualifications (i.e. 10th, 12th and BE marks) |
| select sq.studentid , sum(sq.marks) total\_marks from student\_qualifications sq where sq.studentid in(select distinct studentid from student\_qualifications) group by sq.studentid; |
|  |
| 1. Display students’ details who are not having *'Aadhaar'* card. |
| select \* from student where id in (select sc.studentid from student\_cards sc where name != "Aadhaar"); |
|  |