assignment

aug24/ dbt/ 007

database technologies

diploma in advance computing

august 2024

**dml commands: select data with where, group by, having, order by and limit clause.**

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. list all student. |
| **select \* from student;** |
|  |
| 1. list namefirst, namelast of all student. |
| **select namefirst, namelast from student;** |
|  |
| 1. display student information of the student whose student *id* is 10. |
| **select \* from student where id = 10;** |
|  |
| 1. list of various faculties available from faculty table. |
| **select \* from faculty;** |
|  |
| 1. list all student having ‘a’ as second letter in their namefirst. |
| **select \* from student where substring(namefirst, 2, 1) = 'a';** |
|  |
| 1. list all student having letter ‘a’ in their namefirst. |
| **select \* from student where namefirst like '%a%' order by namefirst asc;** |
|  |
| 1. display the details of the student whose dob is '1986-12-14'. |
| **select \* from student where dob = '1986-12-14';** |
|  |
| 1. list all student having ‘r’ as first letter in their namefirst. |
| **select \* from student where substring(namefirst, 1, 1) = 'r';** |
|  |
| 1. display the *namefirst, lastname* from student relation with customized column headings. |
| **select namefirst as 'first name', namelast as 'last name' from student;** |
|  |
| 1. display all students in ascending order of their dob. |
| **select \* from student order by dob;** |
|  |
| 1. display two records of student whose name starts with the letter ‘s’. |
| **select \* from student where substring(namefirst, 1, 1) = 's' limit 2;** |
|  |
| 1. display the student detail whose dob is ‘1986-12-14’. |
| **select \* from student where dob = '1986-12-14';** |
|  |
| 1. display all modules whose module duration is 1 (use modules table). |
| **select \* from Modules where duration = 1;** |
|  |
| 1. display all batches whose sitting capacity is 80 students (use course\_batches table). |
| **select \* from course\_batches where capacity = 80;** |
|  |
| 1. display all student qualification who have done’ be’ and secured marks more than 70. (use student\_qualifications table). |
| **select \* from student\_qualifications where name = 'BE' and marks >= 70;** |
|  |
| 1. display all student qualification who have done’ be’ and graduated in the year 2017. (use student\_qualifications table). |
| **select \* from student\_qualifications where name = 'BE' and year = 2017;** |
|  |
| 1. display all student qualification who have done’ be’ and graduated in the year 2017 and scored marks more than 80. (use student\_qualifications table). |
| **select \* from student\_qualifications where name = 'BE' and year = 2017 and marks >= 80 ;** |
|  |
| 1. display faculty qualification who have done ‘be’ from ‘harvard university’(use faculty\_qualifications table) |
| **select \* from faculty\_qualifications where name = 'BE' and university = "harvard university";** |
|  |
| 1. display all courses whose course duration is 6 months.(use course table) |
| **select \* from course where duration = 6;** |
|  |
| 1. display module details whose module duration is between 1 and 2, arrange the data in ascending order of module duration. (use module table) |
| **select \* from modules where duration between 1 and 2 ORDER BY duration ASC;** |
|  |
| 1. display all student with their voting rights, if the student is below 1980 then print the message “\*the student can vote” else print “the student cannot vote”. |
| **select \*, if(year(dob) < 1980, "Student can vote", "Student cannot vote") as voting from student;** |
|  |
| 1. display all distinct universities from student\_qualifications table. |
| **select distinct university from student\_qualifications;** |
|  |
| 1. display the second highest marks scored by any student in ‘be’. |
| **select max(marks) as second\_highest\_marks from student\_qualifications where marks < (select max(marks) from student\_qualifications where name = 'be') and name = 'be';** |
|  |
| 1. display the second lowest marks scored by any student in ‘be’. |
| **select min(marks) as second\_lowest\_marks from student\_qualifications where marks > (select min(marks) from student\_qualifications where name = 'be') and name = 'be';** |
|  |
| 1. display last 7 student. |
| **select \* from student limit 21, 7;** |
|  |