Assignment

Feb19/ DBT/ 007

Database Technologies

Diploma in Advance Computing

February 2019

**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*** 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 namefirst like ‘\_A%’; |
|  |
| 1. List all student having letter ‘A’ in their namefirst. |
| select \* from student where namefirst like ‘%A%’; |
|  |
| 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 namefirst like ‘R%’; |
|  |
| 1. Display the *namefirst, lastname* from student relation with Customized column headings. |
| select namefirst as “Frist Name”, lastname 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 namefirst like '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 capicity is 80 students (use course\_batches table). |
| select \* from course\_batches where capicity =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 university = 'Harvard University' and name='BE'; |
|  |
| 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 ; |
|  |
| 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 student.\*, if(year(DOB)< 1980, '\*The student can vote', 'The student cannot vote') Election from student; |
|  |
| 1. Display all distinct universities from student\_qualifications table. |
| select distinct university from student\_qualifications; |
|  |
| 1. Calculate the sum of marks student wise of their qualifications (ie 10th, 12th and BE marks) |
| select sum(marks) from student\_qualifications group by studentID; |
|  |
| 1. Display the second highest marks scored by any student in ‘BE’. |
| select marks from student\_qualifications where name='BE' group by marks order by marks desc limit 1, 1; |
|  |
| 1. Display the second lowest marks scored by any student in ‘BE’. |
| select marks from student\_qualifications where name='BE' group by marks order by marks limit 1, 1; |
|  |
| 1. Display last 7 student. |
| Set @x := 0;  select \* from (select @x:=@x+1 R1, student.\* from student, (Select @x:=0) T1) T2 order by R1 desc limit 7; |
|  |
|  |
|  |
|  |
|  |
|  |
|  |