Assignment

DBT/ 007

Database Technologies

Diploma in Advance Computing

**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 students. |
| **mysql> select CONCAT(namefirst," ",namelast) AS Full\_Name from student;** |
|  |
| 1. List namefirst, namelast of all students. |
| **mysql> select namefirst,namelast from student;** |
|  |
| 1. Display student information of the student whose student*ID* is 10. |
| **mysql> select \* from student where id=10;** |
|  |
| 1. List of various faculties available from faculty table. |
| mysql> select namefirst, namelast from faculty; |
|  |
| 1. List all students having ‘A’ as second letter in their namefirst. |
| **mysql> select \* from student where namefirst like "\_a%";** |
|  |
| 1. List all students having letter ‘A’ in their namefirst. |
| **mysql> select \* from student where namefirst like "%a%";** |
|  |
| 1. Display the details of the student whose DOB is'1986-12-14'. |
| mysql> select \* from student where dob='1986-12-14'; |
|  |
| 1. List all students having ‘R’ as first letter in their namefirst. |
| **mysql> select \* from student where namefirst like "R%";** |
|  |
| 1. Display the *namefirst, lastname* from student relation with Customized column headings. |
| mysql> select namefirst FIRST\_NAME, namelast LAST\_NAME from student; |
|  |
| 1. Display all students in ascending order of their DOB. |
| mysql> select namefirst FIRST\_NAME, namelast LAST\_NAME,dob BIRTHDAY from student order by dob ; |
|  |
| 1. Display two records of student whose name starts with the letter ‘S’. |
| mysql> select namefirst,emailid from student where namefirst like "S%"; |
|  |
| 1. Display the student detail whose DOB is ‘1986-12-14’. |
| mysql> select \* from student where dob='1986-12-14'; |
|  |
| 1. Display all modules whose module duration is 1 (use modules table). |
| mysql> select \* from modules where duration=1; |
|  |
| 1. Display all batches whose sitting capacity is 80 students (use course\_batches table). |
| mysql> 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). |
| mysql> select \* from student\_qualifications where name='BE'; |
|  |
| 1. Display all student qualification who have done’ BE’ and graduated in the year 2017. (use student\_qualifications table). |
| mysql> 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). |
| mysql> select \* from student\_qualifications where name='BE’ and marks>80 and year=2017; |
|  |
| 1. Display faculty qualification who have done ‘BE’ from ‘Harvard University’(use faculty\_qualifications table) |
| mysql> select \* from student\_qualifications where university='Harvard University'; |
|  |
| 1. Display all courses whose course duration is 6 months.(use course table) |
| mysql> 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) |
| mysql> 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”. |
| mysql> select namefirst, dob, case when date\_format(dob,'%Y')<=1980 then "Student can Vote." when date\_format(dob,'%Y') NOT IN(1980) then "Student cannot Vote." end AS VOTE from student; |
|  |
| 1. Display all distinct universities from student\_qualifications table. |
| mysql> select distinct university from student\_qualifications; |
|  |
| 1. Display the second highest marks scored by any student in ‘BE’. |
| mysql> select studentID, name,marks, RANKS from(SELECT studentID, name, marks, DENSE\_RANK() OVER (ORDER BY marks DESC) AS RANKS FROM student\_qualifications WHERE name = 'BE')RANKED where RANKS=2; |
|  |
| 1. Display the second lowest marks scored by any student in ‘BE’. |
| mysql> select studentID, name,marks, RANKS from(SELECT studentID, name, marks, DENSE\_RANK() OVER (ORDER BY marks) AS RANKS FROM student\_qualifications WHERE name = 'BE')RANKED where RANKS=2; |
|  |
| 1. Display last 7 students. |
| mysql> select \* from student\_qualifications order by ID desc limit 7; |
|  |