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. |
| **SELECT \* FROM STUDENT;** |
|  |
| 1. List namefirst, namelast of all students. |
| **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 students having ‘A’ as second letter in their namefirst. |
| **SELECT \* FROM STUDENT WHERE NAMEFIRST LIKE "\_A%";** |
| 1. List all students 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 students 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 FIRST\_NAME, NAMELAST LAST\_NAME FROM STUDENT; |
|  |
| 1. Display all students in ascending order of their DOB. |
| SELECT \* FROM STUDENT ORDER BY DOB ASC; |
|  |
| 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 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 IN(1,2) 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 NAMEFIRST, NAMELAST,DOB,IF(YEAR(DOB)<1980,"THE STUDENT CAN VOTE", "THE STUDENT CANNOT VOTE") "VOTING\_RIGHTS" 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 MARKS FROM (SELECT MARKS, DENSE\_RANK()OVER( ORDER BY MARKS DESC) AS RNK FROM STUDENT\_QUALIFICATIONS WHERE NAME='BE') AS RNK1 WHERE RNK=2 LIMIT 1; |
|  |
| 1. Display the second lowest marks scored by any student in ‘BE’. |
| SELECT MARKS FROM (SELECT MARKS, DENSE\_RANK()OVER(ORDER BY MARKS) AS RANK1 FROM STUDENT\_QUALIFICATIONS WHERE NAME="BE")AS RNK WHERE RANK1=2; |
|  |
| 1. Display last 7 students. |
| SELECT \* FROM STUDENT ORDER BY ID DESC LIMIT 7; |
|  |