**School Management System**

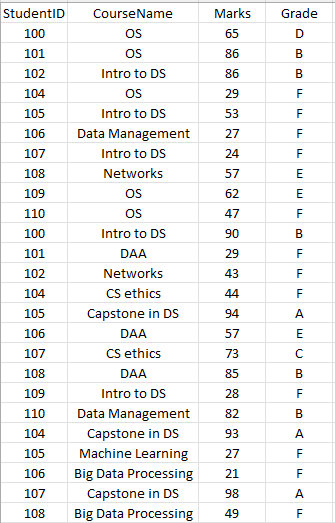
**QUERIES :**

**Query 1 : To Retrieve grade and marks obtained by a student in each course.**

Select StudentID,CourseName, Marks, Grade from marks

left join grades on marks.Marks>= grades.MinMarks and marks.Marks<=MaxMarks left join enrollment using(EnrollmentID)

left join student using(StudentID) left join courses using(CourseID);



**Query 2 : Total number of students for each course who gave exam and results were graded**

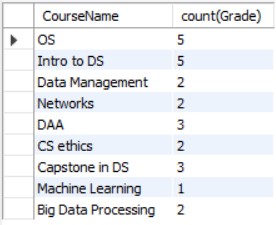
select CourseName , count(Grade)

from (select StudentID,CourseName, Marks, Grade from marks

left join grades on marks.Marks>= grades.MinMarks and marks.Marks<=MaxMarks left join enrollment using(EnrollmentID)

left join student using(StudentID)

left join courses using(CourseID)) as graded group by CourseName;



**Query 3 : Number of Students who passed the exams for each course**

select CourseName , count(Grade)

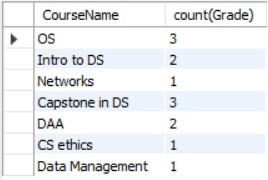
from (select StudentID,CourseName, Marks, Grade from marks

left join grades on marks.Marks>= grades.MinMarks and marks.Marks<=MaxMarks left join enrollment using(EnrollmentID)

left join student using(StudentID)

left join courses using(CourseID)) as graded where Grade !="F"

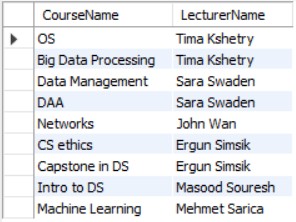
group by CourseName;



**Query 4 :To view lecturer for each course**

select distinct( CourseName), LecturerName from courses

join enrollment using (CourseID) join lecturers using (LecturerID);



**Query 5 : How many subjects a professor is teaching**

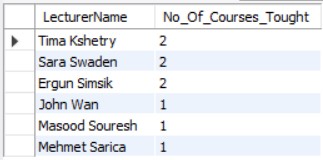
select LecturerName, count(CourseName) No\_Of\_Courses\_Tought from

(select distinct( CourseName), LecturerName from courses

join enrollment using (CourseID)

join lecturers using (LecturerID)) as CoursesTought group by LecturerName

order by No\_Of\_Courses\_Tought desc;



**Query 6 : Parent contact details for each student**

select StudentID , ParentName, PhoneNumber from parent

join student using (studentID) Order By StudentID;



**Query 7 : Number of courses each student is enrolled in**

select student.StudentID, ifnull( No\_Of\_Courses,0) No\_OF\_Courses from student

left join

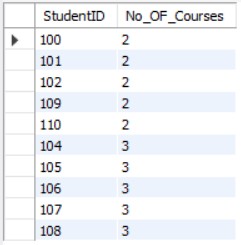
(select student.StudentID, concat(LastName," ",FirstName) as Name, count(EnrollmentID)

No\_Of\_Courses from student

join enrollment using(StudentID)

group by student.StudentID) as No\_Of using(StudentID) where No\_OF\_Courses >0

order by No\_OF\_Courses;

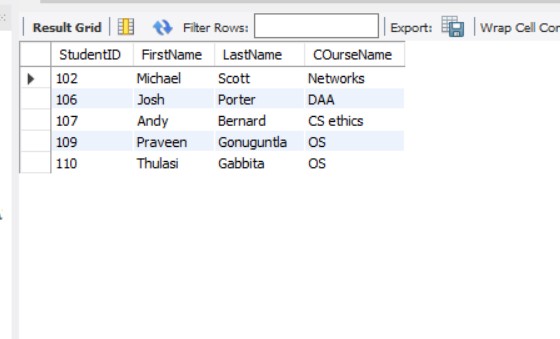


**Query 8 : Number of DS major students enrolled in CS courses**

select student.StudentID,FirstName,LastName,CourseName from enrollment join student on student.StudentID = enrollment.StudentID

join courses using(CourseID)

where enrollment.courseid like 'CS%' and student.major like 'DS%';

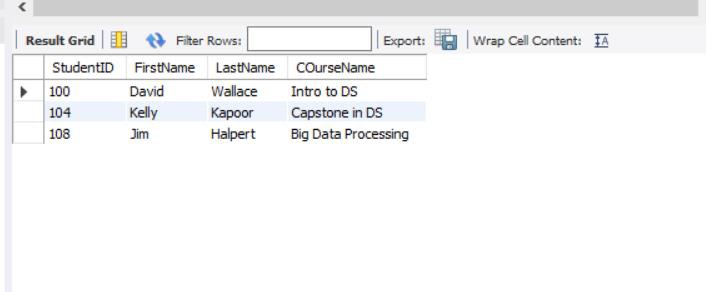


**Query 9 : Number of CS major students enrolled in DS courses**

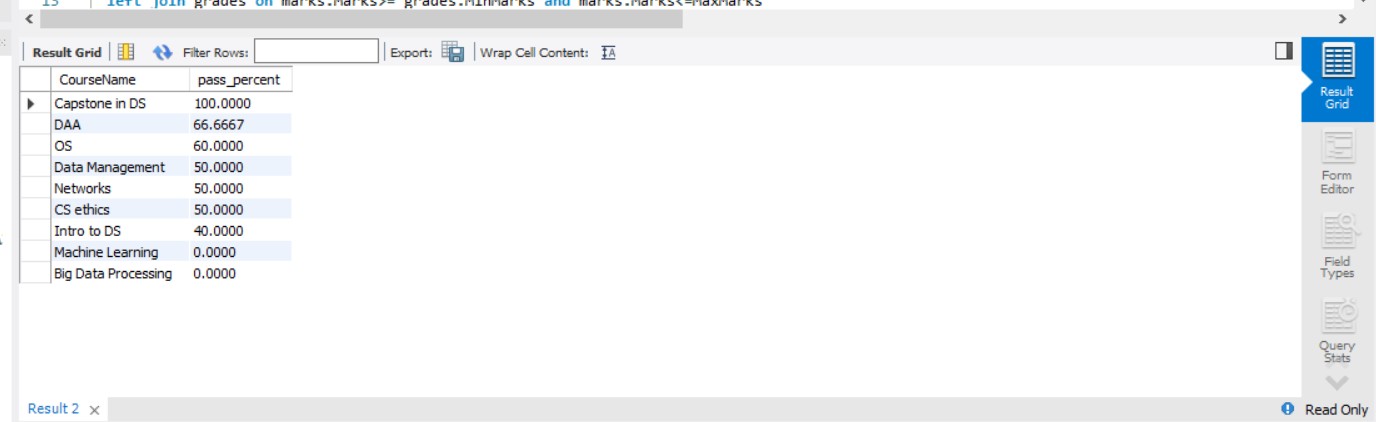
select student.StudentID,FirstName,LastName,COurseName from enrollment join student on student.StudentID = enrollment.StudentID

join courses using(CourseID)

where enrollment.courseid like 'DA%' and student.major like 'CS%'**;**



**Query 10 : Percentage of graded students who were passed for each course**



select CourseName ,ifnull((passed/students)\*100,0) as pass\_percent from (select CourseName , count(Grade) as students from

(select StudentID,CourseName, Marks, Grade from marks

left join grades on marks.Marks>= grades.MinMarks and marks.Marks<=MaxMarks left join enrollment using(EnrollmentID)

left join student using(StudentID)

left join courses using(CourseID)) as graded group by CourseName) as Total\_Students left join(

select CourseName , count(Grade) as passed

from (select StudentID,CourseName, Marks, Grade from marks

left join grades on marks.Marks>= grades.MinMarks and marks.Marks<=MaxMarks left join enrollment using(EnrollmentID)

left join student using(StudentID)

left join courses using(CourseID)) as graded where Grade !="F"

group by CourseName) as passed using(CourseName) order by pass\_percent desc;

**Query 11 : Students who are not registered for any course**

select StudentID, FirstName,LastName from( select \* from

student

left join enrollment using(StudentID)) as registred where EnrollmentID is null;

