

## 1 Views

Suppose we wish to display each student ID and a CGPA. We can break this into steps by first creating a **VIEW** . You may assume that each letter grades translates to a GPA value as shown by the table **grade\_points**.

The CGPA is defined as

$$\frac{\sum_i (credit_i * points_i)}{\sum_i credits_i}$$

Which tables do we need columns from?

**course, takes, grade\_points**

We simplify the problem by creating a view with which columns?

ID, course\_id, grade, credits

The **VIEW** statement is then:

```
CREATE VIEW student_marks AS (SELECT id, takes.course_id, grade, credits FROM takes
NATURAL JOIN course);
```

and the final **SELECT** statement is then:

```
SELECT id, sum(credits*points)/sum(credits) AS gpa FROM student_marks NATURAL JOIN
grade_points GROUP BY id;
```

grade	points
A	4.0
A+	4.0
A-	3.7
B	3.0
B+	3.3
B-	2.7
C	2.0
C+	2.3
C-	1.7
D	1.0
D+	1.3
D-	0.7
F	0.0

## 2 Outer Joins

Let's try some **OUTER JOINS**.

- Find all students and the courses they have taken - include students who have not taken any courses yet.

```
SELECT * FROM student NATURAL LEFT OUTER JOIN takes;
```

Notice the difference with **SELECT \* FROM student NATURAL JOIN takes;**.

Student Snow is missing.

Can we rewrite the **LEFT OUTER JOIN** using an equivalent **RIGHT OUTER JOIN**?

yes. switch relations order so **SELECT \* FROM takes NATURAL RIGHT OUTER JOIN student;**

What if we would like to find all students who have *not* take a course?

```
SELECT * FROM student NATURAL LEFT OUTER JOIN takes WHERE course_id IS NULL;
```

- Display a list of all students in the Comp. Sci. department, along with the course sections, if any, that they have taken in Spring 2009; all course sections from Spring 2009 must be displayed, even if no student from the Comp. Sci. department has taken the course section.

Let's first think of this in words. We want to select all the students in Comp. Sci. and natural full outer join with all the spring 2009 courses. In MySQL this is the union of the left outer join with the right outer join.

```
SELECT * FROM
(SELECT * FROM student WHERE dept_name='Comp. Sci.') A
NATURAL FULL OUTER JOIN
(SELECT * FROM takes WHERE semester='Spring' AND year=2009) B;
```

which is in MySQL:

```
(select A.*, B.* from (select * from student where dept_name='Comp. Sci.') A
LEFT OUTER JOIN
(select * from takes where semester='Spring' and year=2009) B ON A.id=B.id)

UNION

(select A.*, B.* from (select * from student where dept_name='Comp. Sci.') A
RIGHT OUTER JOIN
(select * from takes where semester='Spring' and year=2009) B ON A.id=B.id);
```

- Another JOIN example. Select all student names and their advisor names include those students who do not have advisors.

```
SELECT student.name AS 'student name', instructor.name AS 'instructor name' FROM
(student LEFT JOIN advisor ON student.id = s_id) LEFT JOIN instructor ON advisor.i_id
= instructor.id;
```

- What if we'd like to see the instructors who don't have students to advise?

```
(SELECT student.name AS 'student name', instructor.name AS 'Instructor Name'
FROM (student LEFT JOIN advisor ON student.id = s_id)
LEFT JOIN instructor ON advisor.i_id = instructor.id)
UNION
(SELECT student.name AS 'student name', instructor.name AS 'Instructor Name'
FROM (student JOIN advisor ON student.id = s_id)
RIGHT JOIN instructor ON advisor.i_id = instructor.id);
```

### 3 Formatting Output

We've already seen how we can use the **AS** command to rename a column into something nicer.

We've also seen how to use basic arithmetic to calculate values to put into columns. We can also use the **CONCAT** function to merge column names together. For example, if we wanted to display the day and time of all time slots we could use the following:

```
SELECT time_slot_id, CONCAT(day, ': ', start_hr, ':', start_min, '-', end_hr, ':', end_min) AS
```

Many other string functions are available see

<http://www.tutorialspoint.com/mysql/mysql-string-functions.htm>.

There are also many mathematical functions available for computations which can be seen at <http://www.tutorialspoint.com/mysql/mysql-numeric-functions.htm>.