

## ASSIGNMENT 2: SQL

**Q1. Find the number of instructors who have never taught any course. If the result of your query is empty, add the appropriate data (and include corresponding insert statements) to ensure the result is not empty.**

| count(name) |
|-------------|
| 3           |

```
mysql> select count(name) from instructor where id not in (select ID from teaches);
+-----+
| count(name) |
+-----+
|          3 |
+-----+
1 row in set (0.12 sec)
```

**Q2. Find the total capacity of every building in the university.**

| building | sum(capacity) |
|----------|---------------|
| Packard  | 500           |
| Painter  | 10            |
| Taylor   | 70            |
| Watson   | 80            |

```
mysql> select building, sum(capacity)
-> from classroom group by building;
+-----+-----+
| building | sum(capacity) |
+-----+-----+
| Packard  |          500 |
| Painter  |           10 |
| Taylor   |           70 |
| Watson   |           80 |
+-----+-----+
4 rows in set (0.03 sec)
```

**Q3. Find all departments that have at least one instructor and list the names of the departments along with the number of instructors; order the result in descending order of number of instructors.**

| dept_name  | Total_instructors |
|------------|-------------------|
| Comp. Sci. | 3                 |
| Finance    | 2                 |
| History    | 2                 |
| Physics    | 2                 |
| Biology    | 1                 |
| Elec. Eng. | 1                 |
| Music      | 1                 |

```
mysql> select dept_name,count(instructor.ID) as Total_instructors
-> from department
-> inner join instructor
-> using(dept_name)
-> group by department.dept_name
-> Having count(instructor.ID) >= 1
-> order by count(instructor.ID) desc;
```

| dept_name  | Total_instructors |
|------------|-------------------|
| Comp. Sci. | 3                 |
| Finance    | 2                 |
| History    | 2                 |
| Physics    | 2                 |
| Biology    | 1                 |
| Elec. Eng. | 1                 |
| Music      | 1                 |

7 rows in set (0.03 sec)

**Q4.** For each student, compute the total credits they have successfully completed, i.e. total credits of courses they have taken, for which they have a non-null grade other than 'F'. Do NOT use the tot\_credits attribute of student.

| ID    | name     | sum(credits) |
|-------|----------|--------------|
| 00128 | Zhang    | 7            |
| 12345 | Shankar  | 14           |
| 19991 | Brandt   | 3            |
| 23121 | Chavez   | 3            |
| 44553 | Peltier  | 4            |
| 45678 | Levy     | 7            |
| 54321 | Williams | 8            |
| 55739 | Sanchez  | 3            |
| 76543 | Brown    | 7            |
| 76653 | Aoi      | 3            |
| 98765 | Bourikas | 7            |
| 98988 | Tanaka   | 4            |

```
mysql> select student.ID , student.name, sum(credits)
-> from student
-> left join takes
-> using(ID)
-> inner join course
-> on takes.course_id = course.course_id
-> where takes.grade <> 'F'
-> group by student.ID;
```

| ID    | name     | sum(credits) |
|-------|----------|--------------|
| 00128 | Zhang    | 7            |
| 12345 | Shankar  | 14           |
| 19991 | Brandt   | 3            |
| 23121 | Chavez   | 3            |
| 44553 | Peltier  | 4            |
| 45678 | Levy     | 7            |
| 54321 | Williams | 8            |
| 55739 | Sanchez  | 3            |
| 76543 | Brown    | 7            |
| 76653 | Aoi      | 3            |
| 98765 | Bourikas | 7            |
| 98988 | Tanaka   | 4            |

12 rows in set (0.04 sec)

## Section B

**Q1. Find the id and title of all courses which do not require any prerequisites.**

| course_id | title                      |
|-----------|----------------------------|
| BIO-101   | Intro. to Biology          |
| CS-101    | Intro. to Computer Science |
| FIN-201   | Investment Banking         |
| HIS-351   | World History              |
| MU-199    | Music Video Production     |
| PHY-101   | Physical Principles        |

```
mysql> select course_id, title from course
-> where course_id not in (select course_id from prereq);
+-----+-----+
| course_id | title |
+-----+-----+
| BIO-101   | Intro. to Biology |
| CS-101    | Intro. to Computer Science |
| FIN-201   | Investment Banking |
| HIS-351   | World History |
| MU-199    | Music Video Production |
| PHY-101   | Physical Principles |
+-----+-----+
6 rows in set (0.01 sec)
```

**Q2. Find the names of students who have not taken any biology dept. courses.**

| name     |
|----------|
| Zhang    |
| Shankar  |
| Levy     |
| Williams |
| Brown    |
| Bourikas |
| Aoi      |
| Chavez   |
| Brandt   |
| Sanchez  |
| Peltier  |

```
mysql> select student.name
-> from student
-> left join takes
-> using(ID)
-> inner join course
-> on takes.course_id = course.course_id
-> where course.dept_name <> "Biology"
-> group by student.ID;
+-----+
| name |
+-----+
| Zhang |
| Shankar |
| Levy |
| Williams |
| Brown |
| Bourikas |
| Aoi |
| Chavez |
| Brandt |
| Sanchez |
| Peltier |
+-----+
11 rows in set (0.01 sec)
```