

Homework 2

Ethan Meltzer
I have adhered to the Honor Code on this assignment.

(1)

```
SELECT name
FROM instructor
WHERE dept_name LIKE "%Biology%";
```

name
Quieroz
Valtchev

(2)

```
SELECT title
FROM course
WHERE credits = 3
AND dept_name = "Comp. Sci.";
```

title
International Finance
Computability Theory
Japanese

(3)

```
SELECT course.course_id, course.title
FROM course JOIN takes
ON course.course_id = takes.course_id
AND takes.ID = 12345;
```

This query returns empty (meaning: student with ID 12345 either does not exist or has registered in zero courses)

For an existing student (randomly selected ID 74639)

course_id	title
105	Image Processing
158	Elastic Structures
169	Marine Mammals
237	Surfing
274	Corporate Law
349	Networking
366	Computational Biology
408	Bankruptcy
445	Biostatistics
599	Mechanics
692	Cat Herding
760	How to Groom your Cat
808	Organic Chemistry
959	Bacteriology
960	Tort Law
962	Animal Behavior

(4)

```
SELECT SUM(course.credits)
FROM course JOIN takes
ON course.course_id = takes.course_id
AND takes.ID = 74639;
```

sum(course.credits)
53

(5)

```
SELECT takes.ID, sum(course.credits)
FROM course JOIN takes
ON course.course_id = takes.course_id
GROUP BY takes.ID
ORDER BY CAST(takes.ID AS UNSIGNED);
```

Returns a table with 2000 rows. Here are the first 10 (LIMIT 10):

ID	sum(course.credits)
35	35
56	56
107	86
108	56
123	44
163	52
259	47
282	43
288	55
336	54

(6)

```
SELECT DISTINCT student.name
FROM student JOIN (takes, course)
ON student.ID = takes.ID
AND course.course_id = takes.course_id
AND course.dept_name = "Comp. Sci.";
```

Returns a table with 866 rows. Here are the first 10 (LIMIT 10):

name
Colin
Mediratta
Shabuno
Jr
Saito
Yamashita
Rakoj
Mendelzon
Stone
Sin

(7)

```
SELECT instructor.ID
FROM instructor LEFT JOIN teaches
ON instructor.ID = teaches.ID
WHERE teaches.ID IS NULL
ORDER BY CAST(instructor.ID AS UNSIGNED);
```

ID
4034
16807
31955
35579
37687
50885
52647
57180
58558
59795
63395
64871
72553
74426
78699
79653
95030
96895
97302

(8)

```
SELECT instructor.ID, instructor.name
FROM instructor LEFT JOIN teaches
ON instructor.ID = teaches.ID
WHERE teaches.ID IS NULL
ORDER BY CAST(instructor.ID AS UNSIGNED);
```

ID	name
4034	Murata
16807	Yazdi
31955	Moreira
35579	Soisalon-Soininen
37687	Arias
50885	Konstantinides
52647	Bancilhon
57180	Hau
58558	Dusserre
59795	Desyl
63395	McKinnon
64871	Gutierrez
72553	Yin
74426	Kenje
78699	Pingr
79653	Levine
95030	Arinb
96895	Mird
97302	Bertolino

(9)

```
SELECT [MAX/MIN](t1)
FROM (SELECT COUNT(takes.ID) AS t1, section.course_id, section.sec_id,
section.semester, section.year
FROM takes JOIN section
ON section.course_id = takes.course_id
AND section.sec_id = takes.sec_id
AND section.semester = takes.semester
AND section.year = takes.year
GROUP BY section.course_id, section.sec_id, section.semester,
section.year) AS T;
```

MAX(t1)
338

MIN(t1)
264

(10)

```
WITH enrollment AS
(SELECT COUNT(takes.ID) AS enrolled, section.course_id, section.sec_id,
section.semester, section.year
FROM takes JOIN section
ON section.course_id = takes.course_id
AND section.sec_id = takes.sec_id
AND section.semester = takes.semester
AND section.year = takes.year
GROUP BY section.course_id, section.sec_id, section.semester, section.year),
max_enr AS (SELECT MAX(enrolled) as max FROM enrollment)
SELECT enrollment.* FROM enrollment JOIN max_enr
ON enrollment.enrolled = max;
```

enrolled	course_id	sec_id	semester	year
338	192	1	Fall	2002
338	362	1	Fall	2005

(11)

```
SELECT DISTINCT title FROM course WHERE title LIKE "%programming%";
```

title
Game Programming
C Programming
RPG Programming
FOCAL Pogramming
Assembly Language Programming

(12)

Show a list of all students who are taking/have taken a course that they did not have the prerequisite classes for, as well as the corresponding course information.

```
WITH t AS (SELECT takes.*, prereq_id
FROM takes JOIN prereq ON takes.course_id = prereq.course_id)
```

```
SELECT * FROM t A JOIN t B ON A.ID = B.ID AND A.course_id NOT IN (SELECT
prereq_id FROM B);
```