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)

```
SELECT ID
  FROM teaches a LEFT JOIN
    (SELECT DISTINCT course_id
      FROM course
      WHERE title
      LIKE "%programming%") b
  ON a.course_id = b.course_id
  WHERE b.course_id IS NOT NULL;
```

ID
22591

(13) 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.

```
SELECT ID, a.course_id, prereq_id
  FROM takes a LEFT JOIN prereq b
  ON a.course_id = b.course_id
  AND b.prereq_id NOT IN
  (SELECT course_id
    FROM takes
    WHERE ID = a.ID)
  WHERE b.course_id IS NOT NULL
  ORDER BY CAST(ID AS UNSIGNED);
```

Returns a table with 15637 rows (someone should really bring this up to the registrar...). Here are the first 10 (`LIMIT 10`)

ID	course_id	prereq_id
35	760	169
56	612	123
56	795	123
56	852	133
56	852	267
56	972	958
56	242	304
56	242	594
107	349	612
107	362	242

(14)

```
SELECT *
  FROM classroom c JOIN section b JOIN section a
  ON a.building = b.building
  AND b.building = c.building
  AND a.room_number = b.room_number
  AND b.room_number = c.room_number
  AND a.semester = b.semester
  AND a.year = b.year
  AND a.time_slot_id = b.time_slot_id
  AND (a.course_id, a.sec_id) <> (b.course_id, b.sec_id);
```

Returns an empty set (and thank goodness!)

(1)

```
CREATE TABLE actors(
  AID int,
  name varchar(255),
  PRIMARY KEY(AID)
);

CREATE TABLE movies(
  MID int,
  title varchar(255),
  PRIMARY KEY(MID)
);

CREATE TABLE actor_role(
  MID int,
  AID int,
  rolename varchar(255),
  FOREIGN KEY(MID) REFERENCES movies(MID),
  FOREIGN KEY(AID) REFERENCES actors(AID)
);
```

(2)

```
INSERT INTO actors
VALUES
(1, "Me"),
(2, "Charlie Chaplin"),
(3, "Benedict Cumberbatch");

INSERT INTO movies
VALUES
(1, "Spider Man"),
(2, "Spider Man 2"),
(3, "Polar Express");

INSERT INTO actor_role
VALUES
(1, 1, "Peter Parker"),
(1, 2, "Otto Octavius"),
(2, 3, "Conductor");
```

(3)

```
SELECT m.title, count(*)
  FROM actor_role r JOIN movies m JOIN actors a
  ON r.MID = m.MID
  AND a.AID = r.AID
  AND a.name = "Charlie Chaplin"
  GROUP BY m.MID;
```

(4)

```
SELECT name
  FROM actors b LEFT JOIN actor_role r
  ON b.AID = r.AID
  WHERE r.AID IS NULL;
```

(5)

I'm not entirely sure what you meant by don't use outer joins, I wasn't planning on using an outer join in the first place?

```
SELECT a.name, m.title
FROM actors a LEFT JOIN actor_role r ON a.AID = r.AID
LEFT JOIN movies m ON r.MID = m.MID;
```