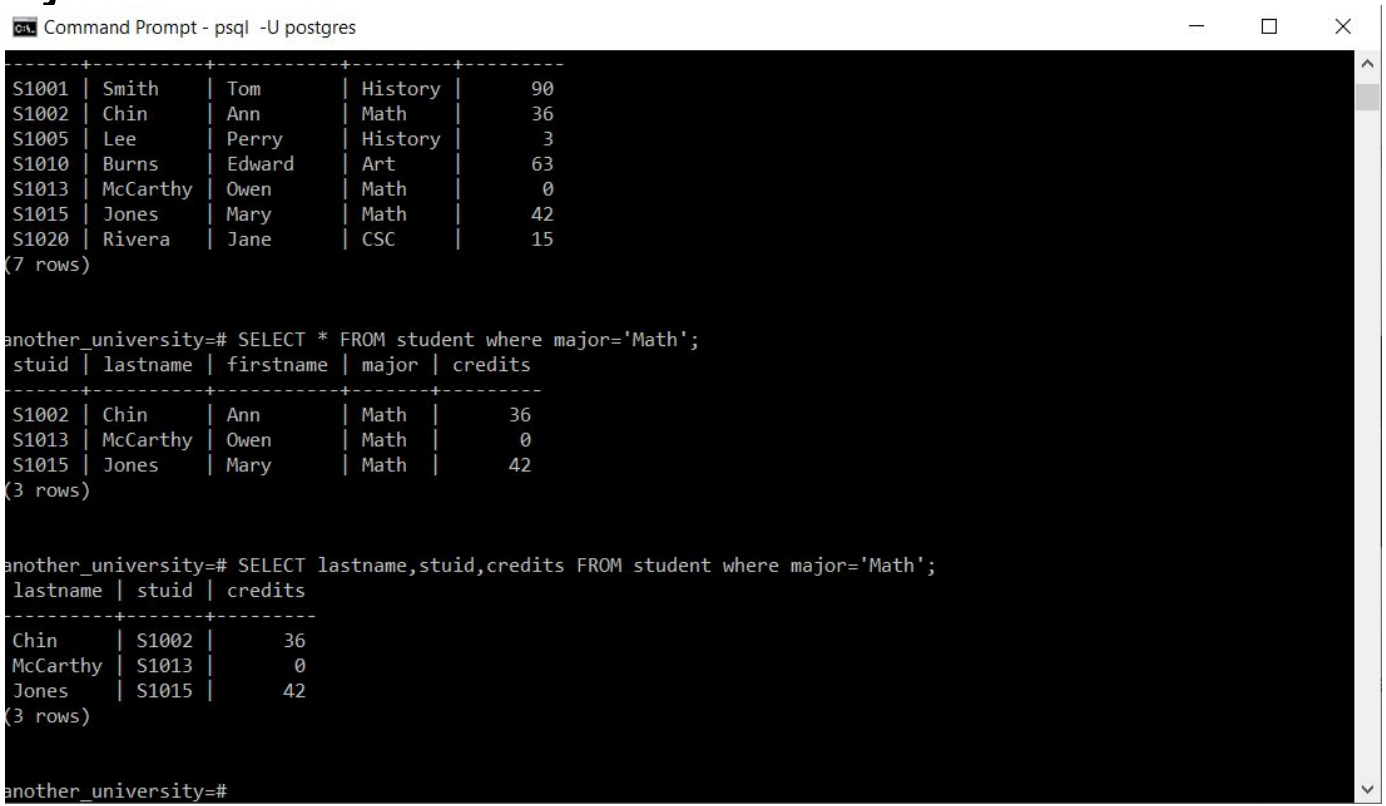
SQL 5 write-up

Asher Kirshtein

1. “Get (only) the last name, id and number of credits of all Math majors.”



“Get the class number of all classes in which students are enrolled.” Eliminate duplicates (if any).

```
another_university=# SELECT DISTINCT classnumber FROM enroll where grade IS NOT NULL;
 classnumber
-------------
 HST205A
 MTH101B
 ART103A
 CSC201A
 MTH103C
(5 rows)
```

2. “Get all information about CSC Faculty.”

```
another_university=# SELECT * FROM facuty;
ERROR:  relation "facuty" does not exist
LINE 1: SELECT * FROM facuty;
                      ^
another_university=# SELECT * FROM faculty;
 facid |  name  | department |    rank
-------+--------+------------+------------
 F101  | Adams  | Art        | Professor
 F105  | Tanaka | CSC        | Instructor
 F110  | Byrne  | Math       | Assistant
 F115  | Smith  | History    | Associate
 F221  | Smith  | CSC        | Professor
(5 rows)


another_university=# SELECT CSC FROM faculty;
ERROR:  column "csc" does not exist
LINE 1: SELECT CSC FROM faculty;
               ^
another_university=# SELECT * FROM faculty where department='CSC';
 facid |  name  | department |    rank
-------+--------+------------+------------
 F105  | Tanaka | CSC        | Instructor
 F221  | Smith  | CSC        | Professor
(2 rows)


another_university=#
```

“Get (only) the names and ids of all Faculty members”. Order the result-set by faculty name, in ascending alphabetical order. The columns should be named FacultyName and FacultyNumber.

```
another_university=# SELECT name AS FacultyName,facid AS FacultyNumber FROM faculty ORDER BY NAME;
 facultyname | facultynumber
-------------+---------------
 Adams       | F101
 Byrne       | F110
 Smith       | F115
 Smith       | F221
 Tanaka      | F105
(5 rows)
```

“Get the last name and first name of all math majors who have at least 30 credits but no more than 59 credits.”

```
another_university=# SELECT lastname,firstname FROM student WHERE credits BETWEEN 30 AND 59;
 lastname | firstname
----------+-----------
 Chin     | Ann
 Jones    | Mary
(2 rows)
```

“Find student id and grade of all students taking any class taught by the Faculty member whose faculty Id is F110.” Order the result-set by ascending student id.

```
another_university=# SELECT stuid,grade FROM enroll where classnumber LIKE 'MTH%' ORDER BY stuid ASC;
 stuid | grade
-------+-------
 S1002 | B
 S1010 |
 S1020 | A
(3 rows)
```

“Find the class numbers, last name, first name, and majors of all students enrolled in the classes taught by Faculty member F110”

```
another_university=# SELECT enroll.classnumber, student.lastname,student.firstname, student.major FROM enroll JOIN stude
nt ON student.stuid = enroll.stuid JOIN class ON class.classnumber = enroll.classnumber JOIN faculty ON faculty.facid =
class.facid WHERE faculty.facid = 'F110';
 classnumber | lastname | firstname | major
-------------+----------+-----------+-------
 MTH103C     | Chin     | Ann       | Math
 MTH103C     | Burns    | Edward    | Art
 MTH101B     | Rivera   | Jane      | CSC
(3 rows)
```

“Find all classes that meet in the same room, with their schedules and room numbers.”. Each tuple in the result-set should include the class number, schedule of each of the pair of classes that meet in the same room, along with the room. Order the result-set by the class number of the first of the classes in the pair, in ascending order.

```
another_university=# SELECT c1.classnumber, c1.schedule, c1.room, c2.classnumber, c2.schedule
another_university-# FROM class c1
another_university-# JOIN class c2
another_university-# ON c1.room = c2.room
another_university-# WHERE c1.classnumber < c2.classnumber
another_university-# ORDER BY c1.classnumber ASC;
 classnumber | schedule | room | classnumber | schedule
-------------+----------+------+-------------+----------
 ART103A     | MWF9     | H221 | HST205A     | MWF11
 CSC201A     | uThF10   | M110 | CSC203A     | MThF12
 MTH101B     | MTuTh9   | H225 | MTH103C     | MWF11
(3 rows)
```