You work for a university and we need your team to run some queries off of the structure put in place.

Here is what the database has so far:

students

--------

id

name

email

ssn

courses

-------

id

name

description

enrollment

----------

id

student\_id

course\_id

semester (e.g. "Fall 2021")

1) How would I get a list of students (name, email, etc...) enrolled in a particular class (by name) for a particular semester?

**Answer:**

Select s.name, s.email, c.name, e.semester

From enrollment e

Inner join courses c on c. id = e.course\_id

Inner join students s on s.id = e.student\_id

Where e. semester = ‘Fall 2021’;

2) How about a list of students and the number of classes they are enrolled in?

E.g.

Name | Number of Classes

------------------------

Joe | 5

Sal | 2

**Answer:**

Select s.name, count(e.course\_id) as [Number of Classes]

From enrollment e

Inner join students s on s.id = e.student\_id

Group by s.name;

3) Now we need to be able to email all of the students that haven’t yet enrolled in any classes this semester to remind them that the deadline is coming up, how would we get a list of students that have not enrolled in any classes in a particular semester?

**Answer:**

Select s.name, s.email

From enrollment e

right join students s on s.id = e.student\_id and e. semester = ‘Fall 2021’

Where e.id is null;

4) Now for the bonus round! A new table called “professors” is created that has basically the same columns as the students table, but for tracking professors rather than students. Now we want a single query that would return all students and professors in the database with the name, email, and an is\_student flag that would indicate if the person is a student or a professor.

E.g.

Name | is student?

------------------

Joe | 1

Mary | 0

**Answer:**

Select \*, 1 as is\_student

From students

Union all

Select \*, 0

From professors;

Python alphabet filter

class LetterFilter:

    def \_\_init\_\_(self, s):

        self.s = s

    def filter\_vowels(self):

        string  =  self.s

        for token in string:

            if token in ["a","e","i","o","u"]:

                string =  string.replace(token, "")

        return string

    def filter\_consonants(self):

        string  =  self.s

        for token in string:

            if token not in ["a","e","i","o","u"]:

                string =  string.replace(token, "")

        return string

        # Return a string

s = input()

f = LetterFilter(s)

print(f.filter\_vowels())

print(f.filter\_consonants())