Assignment 1 CIS3530

Due October 21st, 2021 11:55pm, although you will be allowed to submit until October 23rd without penalty

Learning Goals

By the end of this assignment, you should:

1. Have a good understanding of the relational model
2. Be able to read and understand a given relational schema and its structure
3. Have mastered the techniques for writing relational algebra (RA) queries and basic SQL queries

Submission Instructions

- You must use the same notation of Relational Algebra that we have used in class and in the slides (∏, ∞, ÷, σ , ×, ∩ , ∪, - , ∞_{left} . .)
- Your assignment must be typed - handwritten assignments will not be marked. You may use any text editor of your choice to type in the answers. Submit your assignment on moodle.
- Each question has its own submission instruction. Please read them carefully and submit accordingly.
- To get full marks, each SQL query that you submit for questions 2 and 3 must compile and run correctly on postgres on linux.socs.uoguelph.ca (you may refer to tutorial 1 posted on your course website for instructions on using postgres on our school server).

Later in the course, towards weeks 5 and 6, you will learn about designing and developing your own schema, given a set of requirements. For this assignment, a relational schema is provided to you for each question.

Schema for Questions 1 and 2:

Relations

- **Student (sID, firstName, lastName, email, cgpa)**
  A tuple in this relation represents a student – a student is identified by their sID. Other information stored about a student is their firstName, lastName, email address and their cgpa. An example student tuple is <0411111, 'Fay', 'Simpson', 'fsimpson@uoguelph.ca', 1.4>
- **Course (cNum, name, dept, credit)**
  A tuple in this relation represents a course. Note that a course is identified by its number and the department to which it belongs. An example course tuple is <3530, 'Intro to Databases', 'CIS', 3>

- **Offering (<u>oID,</u> cNum, dept, year, term, instructor)**
  A tuple in this relation stores information on a course offering, which is identified by oID. An example offering tuple is <1, 3530, 'CIS', 2017, 'F', 'Ritu'>
- **Took (sid, oid, grade)**
  A tuple in this relation stores a student's grade in a course offering. An example tuple of took is < 0991326, 1, 79>

<u>Integrity Constraints</u>

- Primary Key of relation
  - Student is sID
  - Course is (cNum, dept)
  - Offering is oID
  - Took is (sID, oID)
- Foreign Key
  - Offering has a foreign key (cNum, dept) that references relation Course
  - Took has 2 foreign keys
    - Foreign key (sID) references relation Student
    - Foreign key (oID) references relation Offering

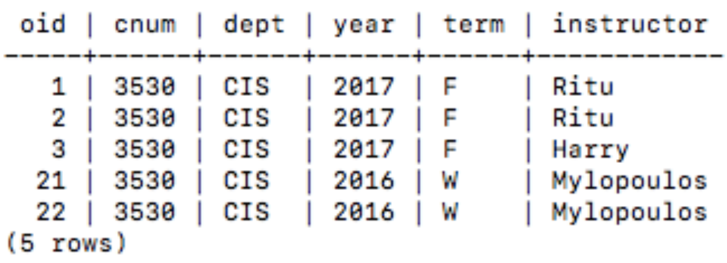
Question 1: (18 marks) You may use an online relational algebra calculator to test your queries, such as RelaX (available under a Creative Commons Attribution-ShareAlike 4.0 International License). You can temporarily load the COURSE database to this calculator by using the following link:

https://dbis-uibk.github.io/relax/calc/gist/37755354db39941cfab752fb52d571be

Write relational algebra expressions to answer the following. **Expected results** for the given instance are shown for your convenience. Submit a file called A1Q1_lastname_firstname.txt that includes all the RA queries for questions 1a to f (other file extensions such as docx or rtf will also be allowed).

Tip: once you compile and run your RA query on the RelaX editor, copy and paste it onto Word (or notepad or textEdit).

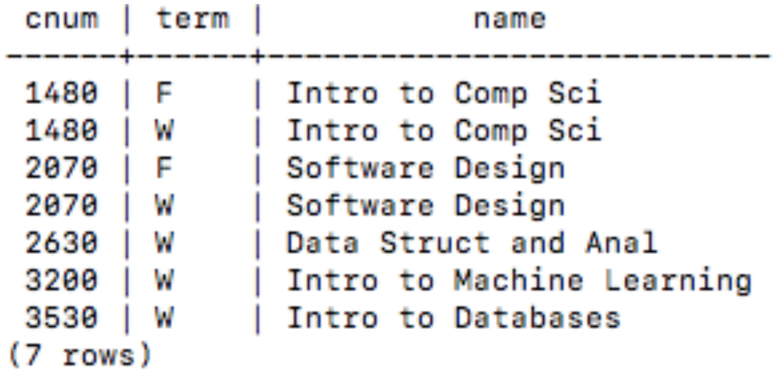
a. Find all offerings of 3530
   Expected Output:



```
 oid | cnum | dept | year | term | instructor
-----+------+------+------+------+------------
   1 | 3530 | CIS  | 2017 | F    | Ritu
   2 | 3530 | CIS  | 2017 | F    | Ritu
   3 | 3530 | CIS  | 2017 | F    | Harry
  21 | 3530 | CIS  | 2016 | W    | Mylopoulos
  22 | 3530 | CIS  | 2016 | W    | Mylopoulos
(5 rows)
```

b. Find all CIS courses offered in 2016

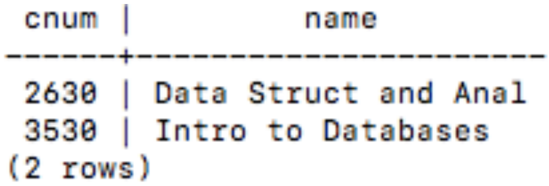
Expected Output:



```
 cnum | term |            name
------+------+---------------------------
 1480 | F    | Intro to Comp Sci
 1480 | W    | Intro to Comp Sci
 2070 | F    | Software Design
 2070 | W    | Software Design
 2630 | W    | Data Struct and Anal
 3200 | W    | Intro to Machine Learning
 3530 | W    | Intro to Databases
(7 rows)
```

c. Find all courses offered by Ritu

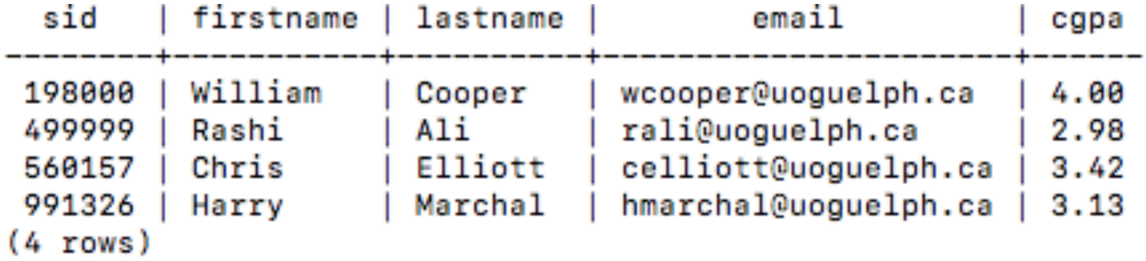
Expected Output:



```
 cnum |         name
------+----------------------
 2630 | Data Struct and Anal
 3530 | Intro to Databases
(2 rows)
```

d. Find all students registered in 3530 in fall 2017.

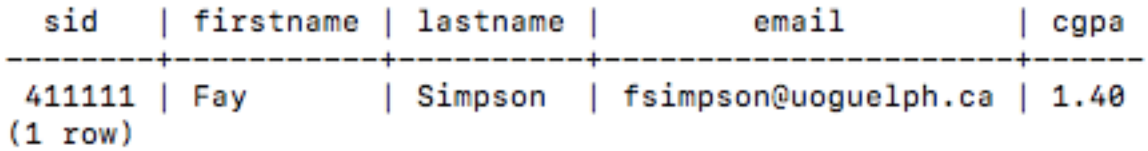
Expected Output:



```
  sid   | firstname | lastname |          email          | cgpa
--------+-----------+----------+-------------------------+------
 198000 | William   | Cooper   | wcooper@uoguelph.ca     | 4.00
 499999 | Rashi     | Ali      | rali@uoguelph.ca        | 2.98
 560157 | Chris     | Elliott  | celliott@uoguelph.ca    | 3.42
 991326 | Harry     | Marchal  | hmarchal@uoguelph.ca    | 3.13
(4 rows)
```

e. Find all students who do not take course CIS3530.

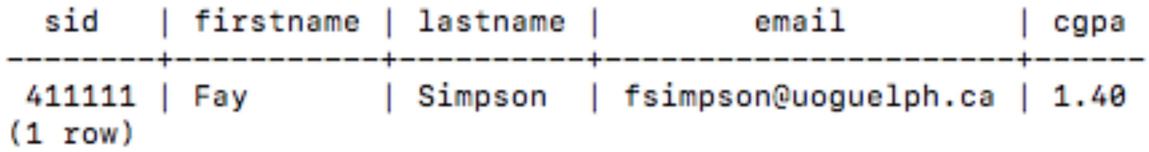
Expected Output:



```
  sid   | firstname | lastname |        email          | cgpa
--------+-----------+----------+-----------------------+------
 411111 | Fay       | Simpson  | fsimpson@uoguelph.ca | 1.40
(1 row)
```

f. Find students who take all 'HIS' courses offered

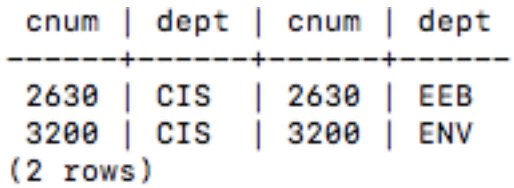
Expected Output:



```
  sid   | firstname | lastname |        email          | cgpa
--------+-----------+----------+-----------------------+------
 411111 | Fay       | Simpson  | fsimpson@uoguelph.ca | 1.40
(1 row)
```

g. Find all courses that have the same number but are offered by different departments.

Expected Output:



```
 cnum | dept | cnum | dept
------+------+------+------
 2630 | CIS  | 2630 | EEB
 3200 | CIS  | 3200 | ENV
(2 rows)
```

h. Write a query that displays all courses and the term and year they are offered, if offered. The courses that are not offered must display NULL values for term and year.

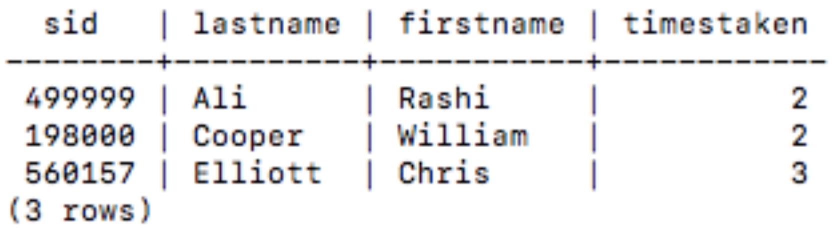
i. Develop a query for this database, write its description / question in English (as given in questions 1a to 1h), give its solution and expected output. The query must involve at least 2 tables and at least 3 operators. You will be marked based on its originality and the operators used.

Question 2 (SQL): (24 marks) You must run the attached scripts given in the 2 files coursesDDL.sql and coursesData.sql to create the tables on postgres required for this question. There are 12 queries to write for this question (a-l). Submit a file called A1Q2_lastname_firstname.sql that includes all the SQL queries for questions 1a to f.

a-h. Write SQL queries for question 1a-h.

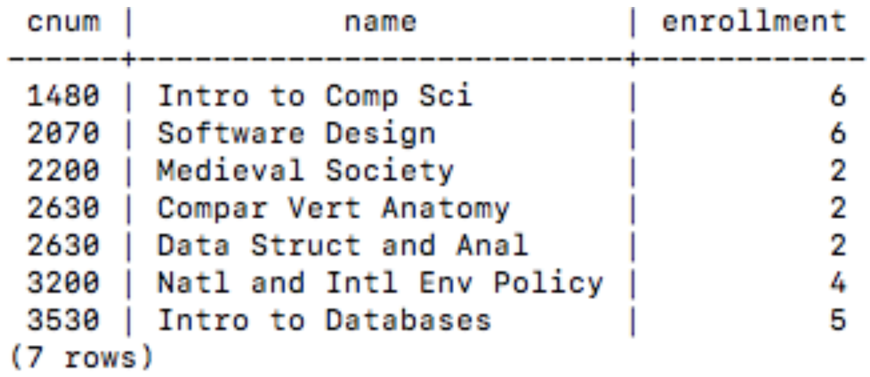
i. Find all students who have taken CIS3530 multiple times.

Expected Output:



```
  sid   | lastname | firstname | timestaken
--------+----------+-----------+------------
 499999 | Ali      | Rashi     |          2
 198000 | Cooper   | William   |          2
 560157 | Elliott  | Chris     |          3
(3 rows)
```

j. Find all courses for the term F 2017 and the current enrollment



```
 cnum |           name           | enrollment
------+--------------------------+------------
 1480 | Intro to Comp Sci        |          6
 2070 | Software Design          |          6
 2200 | Medieval Society         |          2
 2630 | Compar Vert Anatomy      |          2
 2630 | Data Struct and Anal     |          2
 3200 | Natl and Intl Env Policy |          4
 3530 | Intro to Databases       |          5
(7 rows)
```

k. Write a query to update table OFFERING such that course 3530 is always offered by ‘Harry’.

l. Write a query that deletes all courses taught by ‘Ritu’.

Question 3 (SQL): 12 marks: Attached is a zip file called setup_A1Q3 that includes one sql file called A1Q3_create.sql and 4 csv files. Run the sql file to create 4 tables required for this question. Then, on psql prompt, run \copy command on each of the csv files to populate the tables (e.g. \copy males FROM males.csv delimiter ‘,’ csv). Make sure that the csv files are stored in the folder from where you launch or start postgres. The expected output for some of the questions is given for convenience.

Schema for Question 3

Relations

Males (name)
A tuple in this relation stores names of all males in this database.

Females (name)
A tuple in this relation stores names of all females in this database.

parent_of (parent, child)
A tuple in this relation stores names of all parents in this database and their children. For

example, if ‘David’ has 2 kids, ‘Paul’ and ‘Sue’, then this relation will consist of 2 tuples: <’David’, ‘Paul’> and <’David’, ‘Sue’>

married_to (husband, wife)
A tuple in this relation stores names of husband and their wife. An example tuple is <’John’, ‘Sue’>

Note that there are no foreign key constraints defined in this database. Primary keys are shown as underlined.

Write SQL Queries for the following. **Submit a file called** A1Q3_lastname_firstname.sql that includes all the SQL queries for questions 1a to f.

a. **all_grand_parents**

Expected Output:

```
Grand Parents
---------------
 Jeff
 David
 John
 Sue
(4 rows)
```

b. **sister_of**

Expected Output:

```
sister     | name
-----------+-------
 Angela  | John
 Sue       | Paul
 Mary     | Simon
(3 rows)
```

c. aunty_or_uncle_of

Expected Output:

```
Aunt Or Uncle | name
------------------+-------
 Angela           | Mary
 Angela           | Simon
 John               | Beth
 Paul                | Mary
 Paul                | Simon
 Simon            | Bill
(6 rows)
```

d. neice_of

Expected Output:

```
Aunt Or Uncle | niece
---------------+-------
 John          | Beth
 Paul          | Mary
 Angela        | Mary
(3 rows)
```

e. cousin_of

f. daughter_in_law_of