Homework 4

(20 points)

Overview

Objectives: To get hands on practice with relational algebra
Submission File: hw4.yaml
What to Submit: Submit the .yaml file you downloaded above, with your answers to Gradescope. Validate your syntax using the online RelaX web application. You are only allowed a **MAXIMUM OF 10 SUBMISSIONS** to the autograder.
Due Date: Check Gradescope
Links: RelaX calculator, RelaX help

Example yaml structure (Either syntax can be used)

```
- question: 1
  answer: |
     π col2 (σ col1 = 2 (table_name))

- question: 2
  answer: |
    T1 = pi col1, col2 (table_name)
    tau col1, col2 (T1)
  ...
```

Supported Relational Algebra syntax

Classical Notation	Alternative Notation
π	pi
σ	sigma
ρ	rho
τ	tau
γ	gamma
∩	intersect
∪	union

Classical Notation	Alternative Notation
-	\
×	x
⋈	join, inner join, natural join
⟕	left join, left outer join
⟖	right join, right outer join
⟗	full outer join
←	<-
→	->
=	=

Instructions

- The columns should be projected in the same order in which they are mentioned in the question.
  E.g. If the question asks you to get the student name and course title, you should do `pi name, title (...)`
- The sorting should be done in ascending order of all the columns that you selected above and in the same order
  E.g. The selection above would be sorted using `tau name, title (pi name, title (...))`
- Do not rename columns unless explicitly asked to do so

Part 1

Student

sid	name	dept
001	Alice	CS
002	Bob	EE
003	Carol	CS
004	David	PHYS

Course

cid	title	dept	credits
CS425	Databases	CS	3
CS595	Database Security	CS	3
EE591	Microcomputers	EE	4
EE401	VLSI Design	EE	3
PHYS571	Radiation Physics	PHYS	3

Enroll

cid	sid	grade	gradepoint
CS425	001	A	4.0
CS595	001	B	3.0
CS595	002	A	4.0
EE401	001	A	4.0
EE401	002	B	3.0
EE401	004	A	4.0
PHYS571	002	C	2.0
PHYS571	004	A	4.0

Prereq

cid	pid
CS595	CS425
EE591	EE401
...	...

Info

- Attributes shown with gray background form the keys of a relation (e.g. `cid`).
- The attributes `cid` and `sid` of relation **Enroll** are foreign keys to relations **Course** and **Student**, respectively.
- The attributes `cid` and `pid` of relation **Prereq** are foreign keys to relation **Course**'s attribute `cid`

1. (1 Point) Write a relational algebra expression that returns the titles of all the courses with more than 3 credits from the `EE` department.
2. (2 Points) Write a relational algebra expression that returns for each student, their name and the title of courses they have taken from their major, i.e., where the course's dept is equal to the student's dept.
3. (2 Points) Write a relational algebra expression that returns the title and dept of courses that do not have any prerequisites.
4. (3 Points) Write a relational algebra expression that returns for the two students with `sid` 001 and 002, the `cid` of courses that only one of them has taken.
5. (4 Points) Write a relational algebra expression that returns for **EVERY** student, their `sid`, `name` and the number of failed credits as `failedcredits`. The failed credits for a student is the sum of credits of courses in which they have scored a `gradepoint` of less than 2.0.

Part 2

person_living

name
John
Alex
Mike
Chris
Emily
Kate

parent_child

parent_name	child_name
John	Alex
Alex	Mike
Chris	Emily
Emily	Kate

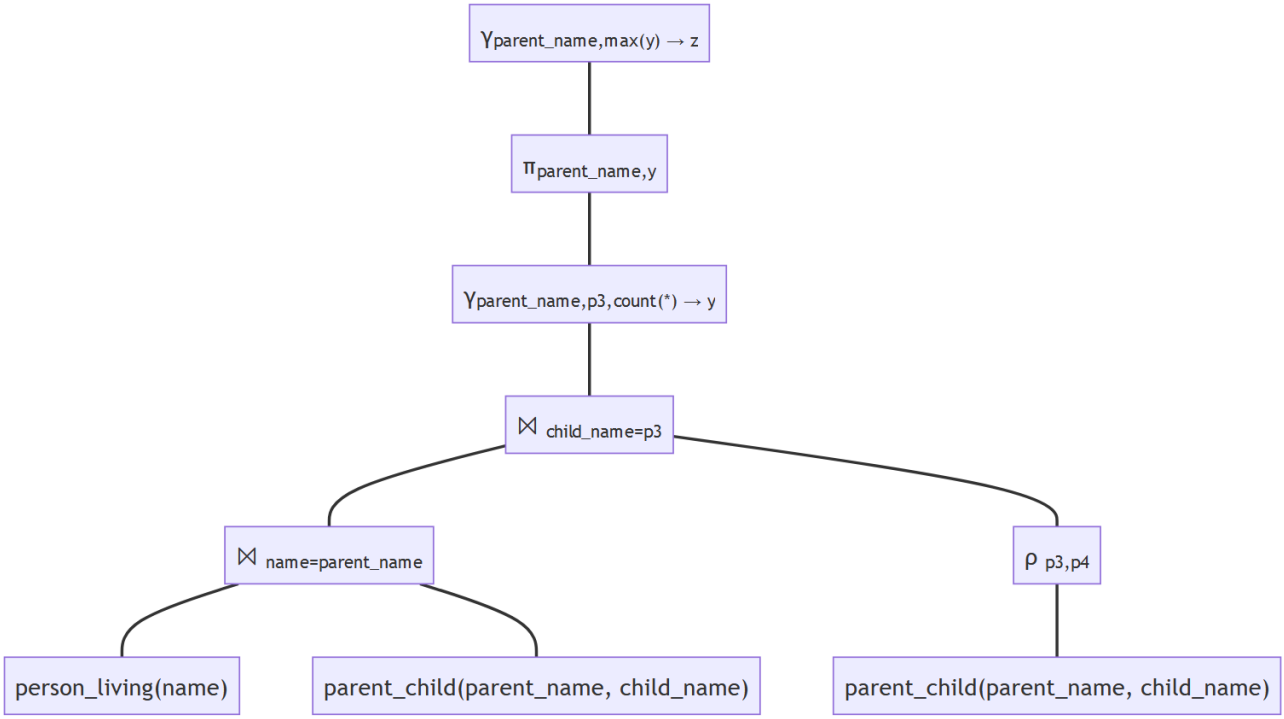
female

name
Alex
Emily
Kate

male

name
John
Mike
Chris

6. (2 points) Write a relational algebra expression to find out parent names who have at least one male child but no female child.
7. (3 points) Write the equivalent SQL query to the following relational algebra query plan



8. (3 points) Write the relational algebra expression that represents the following SQL query:

```
SELECT pl.name
FROM   person_living AS pl, male AS m
WHERE  pl.name = m.name AND
       NOT EXISTS (SELECT *
                   FROM parent_child AS pc, female AS f
                   WHERE pc.parent_name=f.name AND pc.child_name=pl.name)
ORDER BY pl.name;
```