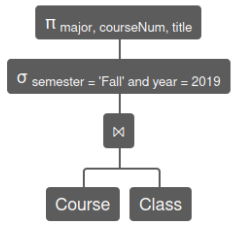
Ace Cassidy-HW3-CPTS451-Spring'20

Write the following queries in Relational Algebra

Q1

```
1 /*Find all courses offered in Fall 2019. Give the courseNum, major, title, and level of those courses*/
2
3 π major, courseNum, title (σ semester='Fall' ∧ year=2019 (Course ⋈ Class))
```

▶ execute query download history



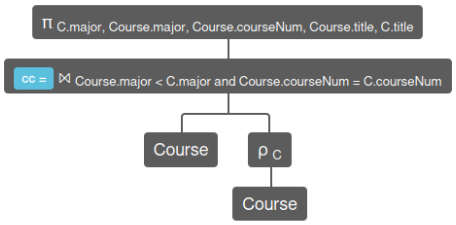
π major, courseNum, title (σ semester = 'Fall' and year = 2019 (Course ⋈ Class))

Course.major	Course.courseNum	Course.title
CptS	321	Object-Oriented Software Principles
CptS	322	Software Engineering Principles I
CptS	355	Programming Language Design
CptS	215	Data Analytics Systems and Algorithms
STAT	360	Probability and Statistics
CptS	581	Software Maintenance
CptS	415	Big Data

Q2

```
1 /*Find the pair of courses that have the same courseNum but are offered by different majors. Return the
  majors, courseNums , and titles of those courses*/
2
3 cc = Course ⋈ Course.major < C.major ∧ Course.courseNum = C.courseNum (ρ C (Course))
4 π C.major, Course.major, Course.courseNum, Course.title, C.title cc
```

execute query | download | history



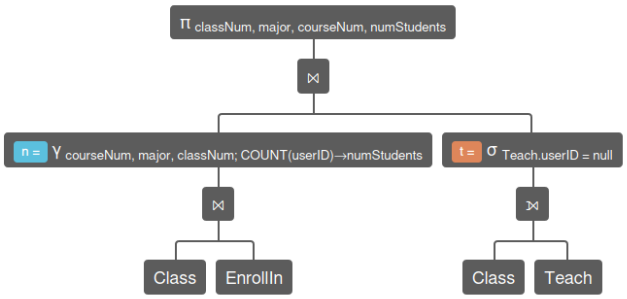
$\pi_{\text{C.major, Course.major, Course.courseNum, Course.title, C.title}}(\text{Course} \bowtie_{\text{Course.major < C.major and Course.courseNum = C.courseNum}} (\rho_{\text{C}}(\text{Course})))$

C.major	Course.major	Course.courseNum	Course.title	C.title
STAT	CptS	360	Systems Programming	Probability and Statistics
MATH	CptS	415	Big Data	Intermediate Differential Equations

Q3

```
1 /*Find the courses which don't have an instructor but have students enrolled init. Return the classNum,
  major, courseNum of those courses and the number of students enrolled in them(call this numStudents)*/
2
3 t = σ Teach.userID=null (Class ⋈ Teach)
4 n = γ courseNum, major, className; count(userID)->numStudents (Class ⋈ EnrollIn )
5 π classNum, major, courseNum, numStudents (n ⋈ t)
```

▶ execute query | download | history



π classNum, major, courseNum, numStudents ((γ courseNum, major, className; COUNT(userID)→numStudents (Class ⋈ EnrollIn)) ⋈ (σ Teach.userID = null (Class ⋈ Teach)))

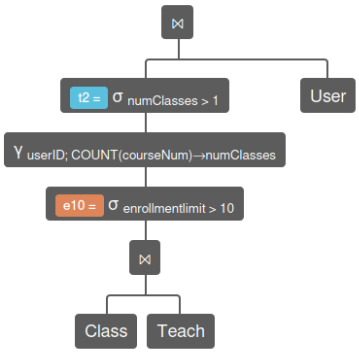
Class.className	Class.major	Class.courseNum	numStudents
2019S01	STAT	360	2
2020S01	CptS	582	3

Q4

π σ ρ ← τ γ ∧ ∨ ¬ = ≠ ≥ ≤ ∩ ∪ ÷ - × ⋈ ⋉ ⋊ ▷ = -- /* {}

```
1 /*Find the instructors who teach at least 2 classes with enrollment limit greater than 10. Give the
  userIDs, first and lastnames, and emails of those instructors and the number of such classes they teach
  (callthis attribute numClasses)*/
2
3 e10 = σ enrollmentlimit > 10 (Class ⋈ Teach)
4 t2 = σ numClasses > 1 (γ userID; count(courseNum)->numClasses (e10))
5 t2 ⋈ User
```

▶ execute query download history



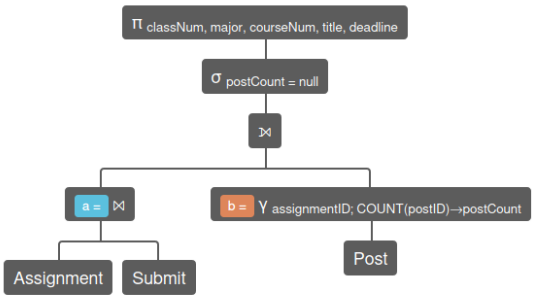
(σ numClasses > 1 (γ userID; COUNT(courseNum)→numClasses (σ enrollmentlimit > 10 (Class ⋈ Teach)))) ⋈ User

Teach.userID	numClasses	User.email	User.firstName	User.lastName
1	2	arslanay@wsu.edu	Sakire	ArslanAy
4	2	rahman@wsu.edu	Tazin	Rahman
11	2	arnaoudova@wsu.edu	Venera	Arnaoudova

Q5

```
1 /*Find the assignments which have submissions but don't have any posts about them. Return the classNum,
  major, and courseNum of the courses and the titles and deadlines of the assignments*/
2
3  a = (Assignment ⋈ Submit)
4  b = γ assignmentID; count(postID)->postCount (Post)
5 π classNum, major, courseNum, title, deadline (σ postCount=null (a ⋈ b))
```

execute query | download | history



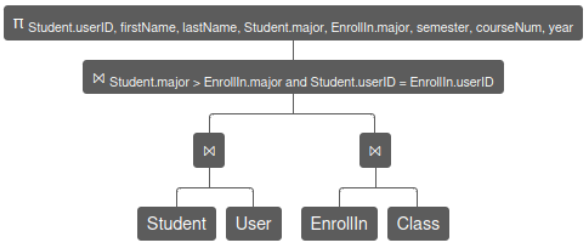
π classNum, major, courseNum, title, deadline (σ postCount = null ((Assignment ⋈ Submit) ⋈ (γ assignmentID; COUNT(postID)→postCount (Post))))

Assignment.classNum	Assignment.major	Assignment.courseNum	Assignment.title	Assignment.deadline
2019S01	CptS	451	Homework 5	2019-03-15

Q6

```
1 /*Find the students who are enrolled in a class offered by a major that is different than student's own
  major.  Return the userID,firstName, lastName, major of the student and the major, courseNum, semester,
  and year of the course student has taken.Order result by userID*/
2
3 π Student.userID , firstName, lastName, Student.major, EnrollIn.major, semester, courseNum, year
  ((Student ⋈ User ) ⋈ Student.major > EnrollIn.major ∧ Student.userID=EnrollIn.userID (EnrollIn ⋈ Class))
```

execute query | download | history



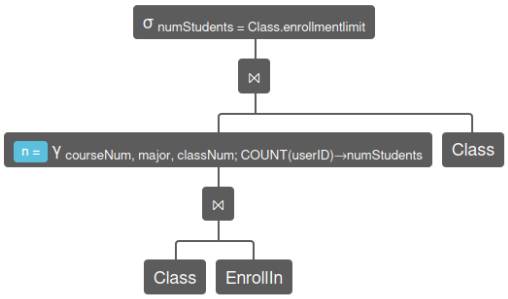
π Student.userID, firstName, lastName, Student.major, EnrollIn.major, semester, courseNum, year ((Student ⋈ User) ⋈ Student.major > EnrollIn.major and Student.userID = EnrollIn.userID (EnrollIn ⋈ Class))

Student.userID	User.firstName	User.lastName	Student.major	EnrollIn.major	Class.semester	EnrollIn.courseNum	Class.year
14	Bob	Zeng	DA	CptS	Spring	451	2019
14	Bob	Zeng	DA	CptS	Fall	355	2019
14	Bob	Zeng	DA	CptS	Spring	437	2019
15	Jacob	Lee	DA	CptS	Spring	451	2019
15	Jacob	Lee	DA	CptS	Fall	355	2019
15	Jacob	Lee	DA	CptS	Spring	437	2019
16	Travis	Person	DA	CptS	Spring	451	2019
16	Travis	Person	DA	CptS	Spring	437	2019
17	Min	Quin	DA	CptS	Spring	437	2019
18	Kelly	Easton	EE	CptS	Spring	582	2020
18	Kelly	Easton	EE	CptS	Spring	437	2019
19	Amy	Fan	EE	CptS	Spring	582	2020
19	Amy	Fan	EE	CptS	Spring	437	2019
20	Ben	Hill	EE	CptS	Spring	451	2019
20	Ben	Hill	EE	CptS	Spring	582	2020
20	Ben	Hill	EE	CptS	Spring	437	2019

Q7

```
1 /*Find the classes that are full(i.e., class enrollment is equal to the enrollment limit). Return
  classNumber, major,courseNum, semester, year, enrollmentlimit of the classesand the number of students
  enrolled(call this numStudents)*/
2
3 n = γ courseNum, major, classNum; count(userID)->numStudents (Class ⋈ EnrollIn)
4 σ numStudents=Class.enrollmentlimit (n ⋈  Class)
```

execute query | download | history



σ numStudents = Class.enrollmentlimit ((γ courseNum, major, classNum; COUNT(userID)→numStudents (Class ⋈ EnrollIn)) ⋈ Class)

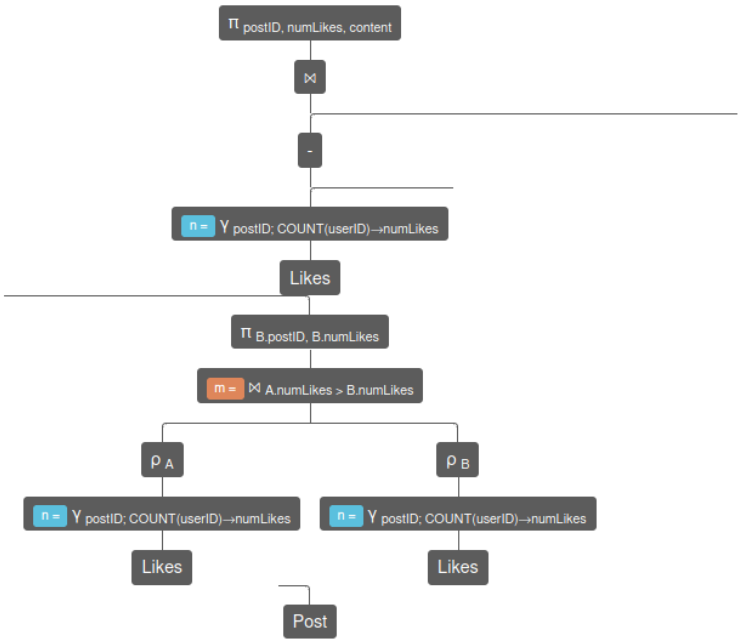
Class.courseNum	Class.major	Class.className	numStudents	Class.semester	Class.year	Class.enrollmentlimit
451	CptS	2019S01	11	Spring	2019	11

Q8

```
/*Find the post with the most number of likes.Return the postID, number of likes(call this numLikes) and
the content of the post*/

n = (γ postID; count(userID)->numLikes (Likes))
m = (ρ A n) ⋈ A.numLikes>B.numLikes (ρ B n)
π postID, numLikes, content ((n - π B.postID, B.numLikes (m)) ⋈ Post)
```

execute query | download | history



π postID, numLikes, content (((γ postID; COUNT(userID)→numLikes (Likes)) - π B.postID, B.numLikes ((ρ A (γ postID; COUNT(userID)→numLikes (Likes))) ⋈ A.numLikes > B.numLikes (ρ B (γ postID; COUNT(userID)→numLikes (Likes))))) ⋈ Post)

Likes.postID	numLikes	Post.content
11	10	Snow day - class is cancelled!