Washington State University

School of Electrical Engineering and Computer Science

CptS 451 – Introduction to Database Systems

Sakire Arslan Ay

Homework-3

Relational Algebra

Due Date: Monday, June 1st

Name: ______________________________

Student Number:______________________

Question:	Max points:	Score:
1	12.5	
2	12.5	
3	12.5	
4	12.5	
5	12.5	
6	12.5	
7	12.5	
8	12.5	
Total	100	

Consider the following schema.

```
CREATE TABLE UserTable (
    userID CHAR(8),
    firstName VARCHAR,
    lastName VARCHAR,
    email VARCHAR(50),
    PRIMARY KEY (userID)
);
CREATE TABLE Instructor(
    instructorID CHAR(8),
    title VARCHAR,
    PRIMARY KEY (instructorID),
    FOREIGN KEY (instructorID) REFERENCES UserTable(userID)
);
CREATE TABLE Student(
    studentID CHAR(8),
    major VARCHAR,
    PRIMARY KEY (studentID),
    FOREIGN KEY (studentID) REFERENCES UserTable(userID)
);
CREATE TABLE Course (
    major VARCHAR,
    courseNum CHAR(3),
    title VARCHAR,
    PRIMARY KEY (major,courseNum)
);
CREATE TABLE Prerequisite (
    major VARCHAR,
    courseNum CHAR(3),
    prereqMajor VARCHAR,
    prereqCourseNum CHAR(3),
    PRIMARY KEY (major,courseNum,prereqMajor,prereqCourseNum),
    FOREIGN KEY (major,courseNum) REFERENCES Course (major,courseNum),
    FOREIGN KEY (prereqMajor,prereqCourseNum) REFERENCES Course (major,courseNum)
);
CREATE TABLE Class (
    classID VARCHAR,
    major VARCHAR NOT NULL,
    courseNum CHAR(3) NOT NULL,
    semester VARCHAR(10),
    year CHAR(4),
    instructorID CHAR(8) NOT NULL,
    enrollmentlimit INTEGER,
    PRIMARY KEY (classID),
    FOREIGN KEY (major,courseNum) REFERENCES Course(major,courseNum),
    FOREIGN KEY (instructorID) REFERENCES Instructor(instructorID)
);
CREATE TABLE Enroll (
    studentID CHAR(8),
    classID VARCHAR,
    grade INTEGER,
    PRIMARY KEY (classID,studentID),
    FOREIGN KEY (classID) REFERENCES Class(classID),
    FOREIGN KEY (studentID) REFERENCES Student(studentID)
);
CREATE TABLE Assignment (
    classID VARCHAR,
    assignmentNo INTEGER,
    title VARCHAR,
    weight INTEGER,
    deadline DATE,
    PRIMARY KEY (classID, assignmentNo),
    FOREIGN KEY (classID) REFERENCES Class(classID)
);
CREATE TABLE Submit (
    studentID CHAR(8),
    classID VARCHAR,
    assignmentNo INTEGER,
    score INTEGER,
    submissionDate DATE,
    PRIMARY KEY (studentID, classID, assignmentNo),
    FOREIGN KEY (classID, assignmentNo) REFERENCES Assignment(classID, assignmentNo),
    FOREIGN KEY (studentID) REFERENCES Student(studentID)
);
CREATE TABLE Post (
    postID INTEGER,
    userID CHAR(8) NOT NULL,
    kind VARCHAR,
    timestamp DATE,
    content VARCHAR,
    popularity INTEGER,
    PRIMARY KEY (postID),
    FOREIGN KEY (userID) REFERENCES UserTable(userID)
);
CREATE TABLE PostAbout (
    postID INTEGER,
    classID VARCHAR,
    assignmentNo INTEGER,
    PRIMARY KEY (postID,classID,assignmentNo),
    FOREIGN KEY (postID) REFERENCES Post(postID),
    FOREIGN KEY (classID, assignmentNo) REFERENCES Assignment(classID, assignmentNo)
);
```

Based on the above database schema, write the following queries in relational algebra. Assume set semantics for all operations.

You will use the online RelaX relational algebra calculator tool to write the queries. To use the tool, simply go to the link:
https://dbis-uibk.github.io/relax/calc.htm?data=gist:dad302f1c596d5fd2d3dcad0b1d7c410
and type the relational algebra expressions in the textbox. A sample dataset for the above relations will be automatically loaded when you open the above link. In the tool, you can try the queries and check the output. The expected output for each relational algebra expression is also provided. (The RelaX tool uses set semantics, so you don’t need to remove duplicates)

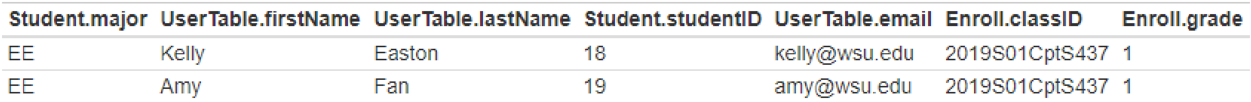
Important note: The sample outputs for the queries are provided to help you test your own solutions. If you include "**hard-coded**" conditions just to obtain the query output, you won't get any credit for your solution.

In your submission:
For each query, take a screenshot of the relational algebra expression, the tree, and the output and paste them in a MS Word file. Crop the unnecessary parts of the images to save space. An example is provided in Appendix-A. Make sure to include the parts marked with arrows.

Write the following queries in Relational Algebra:

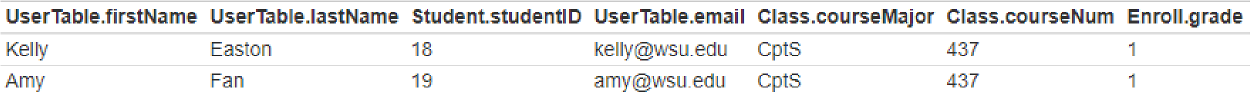
1. Find the students in 'EE' major who failed a class, i.e., earned a grade lower than 2 in a class that they are enrolled in.

   (a) Return the **major**, firstname, lastname, ID, email of the student; the **ID of the class** student failed; and the grade student earned in that class.



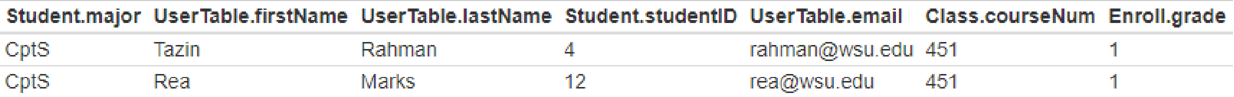
Student.major	UserTable.firstName	UserTable.lastName	Student.studentID	UserTable.email	Enroll.classID	Enroll.grade
EE	Kelly	Easton	18	kelly@wsu.edu	2019S01CptS437	1
EE	Amy	Fan	19	amy@wsu.edu	2019S01CptS437	1

   (b) Return the firstname, lastname, ID, email of the student; **the major and courseNum** of the class student failed; and the grade student earned in that class.



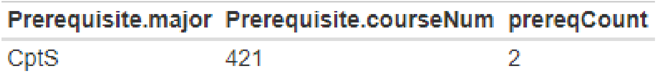
UserTable.firstName	UserTable.lastName	Student.studentID	UserTable.email	Class.courseMajor	Class.courseNum	Enroll.grade
Kelly	Easton	18	kelly@wsu.edu	CptS	437	1
Amy	Fan	19	amy@wsu.edu	CptS	437	1

   (c) Re-write your solution in part(b) to answer the following: "*Find all students who failed a class offered by their own majors.*" Return the major,firstname, lastname, ID, email of the student, the courseNum of the class student failed, and the grade student earned in that class.



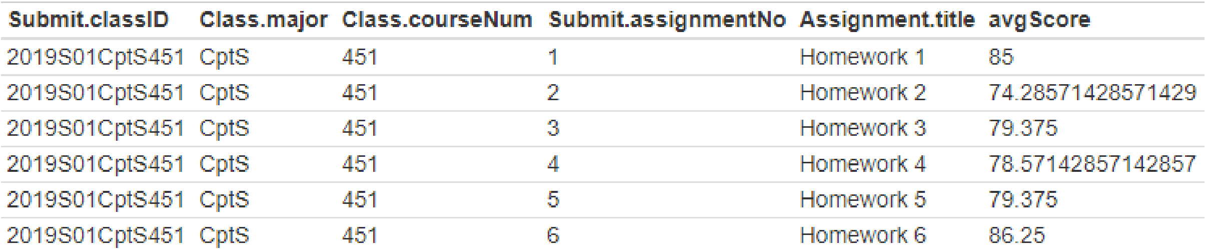
Student.major	UserTable.firstName	UserTable.lastName	Student.studentID	UserTable.email	Class.courseNum	Enroll.grade
CptS	Tazin	Rahman	4	rahman@wsu.edu	451	1
CptS	Rea	Marks	12	rea@wsu.edu	451	1

2. Find the courses that have two or more prerequisite courses.



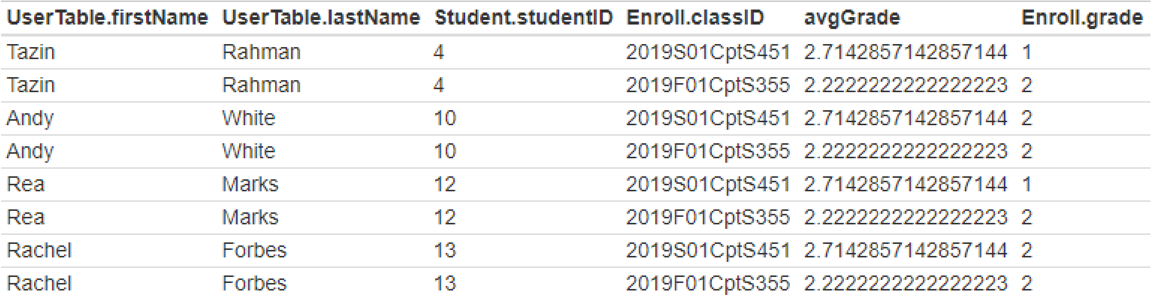
Prerequisite.major	Prerequisite.courseNum	prereqCount
CptS	421	2

3. Find the average of the student scores for each homework assigned in CptS 451 in Spring 2019 semester. Return the classID, course major, courseNum, assignmentNo, assignment title, and average score for each assignment.



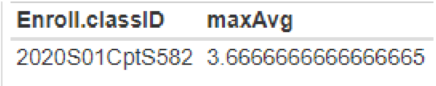
Submit.classID	Class.major	Class.courseNum	Submit.assignmentNo	Assignment.title	avgScore
2019S01CptS451	CptS	451	1	Homework 1	85
2019S01CptS451	CptS	451	2	Homework 2	74.28571428571429
2019S01CptS451	CptS	451	3	Homework 3	79.375
2019S01CptS451	CptS	451	4	Homework 4	78.57142857142857
2019S01CptS451	CptS	451	5	Homework 5	79.375
2019S01CptS451	CptS	451	6	Homework 6	86.25

4. Find the CptS major students whose grade in some class was less than the average grade in that class. Return the name (first and last) and ID of the student; the classID and average grade in the course; and the grade student earned in that class.



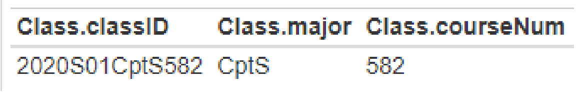
UserTable.firstName	UserTable.lastName	Student.studentID	Enroll.classID	avgGrade	Enroll.grade
Tazin	Rahman	4	2019S01CptS451	2.7142857142857144	1
Tazin	Rahman	4	2019F01CptS355	2.2222222222222223	2
Andy	White	10	2019S01CptS451	2.7142857142857144	2
Andy	White	10	2019F01CptS355	2.2222222222222223	2
Rea	Marks	12	2019S01CptS451	2.7142857142857144	1
Rea	Marks	12	2019F01CptS355	2.2222222222222223	2
Rachel	Forbes	13	2019S01CptS451	2.7142857142857144	2
Rachel	Forbes	13	2019F01CptS355	2.2222222222222223	2

5. Find the course with the maximum average score. Return classID and the average grade in that course.



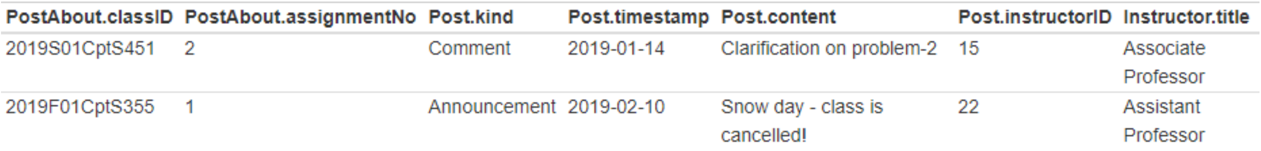
Enroll.classID	maxAvg
2020S01CptS582	3.6666666666666665

6. Find the classes that don't have any assignments, but there are students enrolled in them. Return the classID, major, and courseNum for those courses.



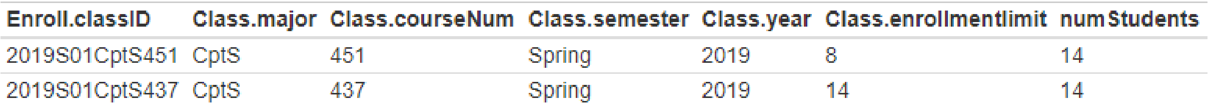
Class.classID	Class.major	Class.courseNum
2020S01CptS582	CptS	582

7. Find the instructors who posted a "post" about an assignment of a class that she/he doesn't teach. Return the classID and assignmentNo of the assignment; kind, timestamp, content of the post; and ID and title of the instructor.



PostAbout.classID	PostAbout.assignmentNo	Post.kind	Post.timestamp	Post.content	Post.instructorID	Instructor.title
2019S01CptS451	2	Comment	2019-01-14	Clarification on problem-2	15	Associate Professor
2019F01CptS355	1	Announcement	2019-02-10	Snow day - class is cancelled!	22	Assistant Professor

8. Find the classes that are full (i.e., enrollment reached the enrollment limit of the class). Return classID, major, courseNum, semester, year, enrollmentlimit of the class and the number of students enrolled.



Enroll.classID	Class.major	Class.courseNum	Class.semester	Class.year	Class.enrollmentlimit	numStudents
2019S01CptS451	CptS	451	Spring	2019	8	14
2019S01CptS437	CptS	437	Spring	2019	14	14

<u>Submission Instructions:</u>
HW3 will be submitted on Blackboard (HW3 dropbox under “Homeworks”). HW3 should be submitted in .pdf format (name your file “HW3.pdf”). Please don’t zip your file.
Email submissions will not be accepted.

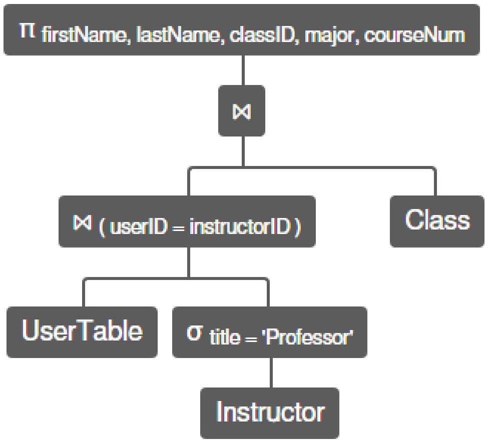
APPENDIX-A

Example output

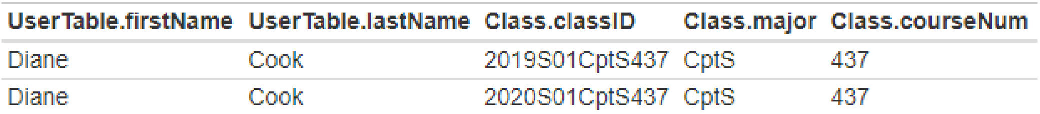
⇨ a. Relational Algebra Expression:

```
π firstName,lastName,classID,major,courseNum
((UserTable ⨝ (userID = instructorID) (σ title='Professor' Instructor))
  ⨝ Class)
```

⇨ b. Relational algebra tree:


π firstName, lastName, classID, major, courseNum
⨝
⨝ (userID = instructorID)
Class
UserTable
σ title = 'Professor'
Instructor

⇨ c. Expression and Output:



UserTable.firstName	UserTable.lastName	Class.classID	Class.major	Class.courseNum
Diane	Cook	2019S01CptS437	CptS	437
Diane	Cook	2020S01CptS437	CptS	437