

**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: Wednesday, February 26<sup>th</sup>

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. The ER diagram for the same schema is provided in the **Appendix-B**.

```
CREATE TABLE User (
    userID INTEGER,
    email VARCHAR(50),
    firstName VARCHAR,
    lastName VARCHAR,
    PRIMARY KEY (userID)
);

CREATE TABLE Instructor(
    userID INTEGER,
    title VARCHAR,
    PRIMARY KEY (userID)
    FOREIGN KEY userID REFERENCES User(userID)
);

CREATE TABLE Student(
    userID INTEGER,
    major VARCHAR,
    PRIMARY KEY (userID)
    FOREIGN KEY userID REFERENCES User(userID)
);

CREATE TABLE Course (
    major VARCHAR,
    courseNum CHAR(3),
    title VARCHAR,
    level VARCHAR,
    PRIMARY KEY (major,courseNum)
);

CREATE TABLE Class (
    classNum CHAR(7)
    major VARCHAR,
    courseNum CHAR(3),
    semester VARCHAR,
    year INTEGER,
    enrollmentlimit INTEGER,
    PRIMARY KEY (major,courseNum,classNum),
    FOREIGN KEY (major,courseNum) REFERENCES Course(major,courseNum)
);

CREATE TABLE Teach (
    major VARCHAR,
    courseNum CHAR(3),
    classNum CHAR(7),
    userID INTEGER,
    PRIMARY KEY (major,courseNum,classNum,userID),
    FOREIGN KEY (major,courseNum,classNum) REFERENCES Class(major,courseNum,classNum),
    FOREIGN KEY (userID) REFERENCES Instructor(userID)
);
```

```

CREATE TABLE EnrollIn (
    major VARCHAR,
    courseNum CHAR(3),
    classNum CHAR(7),
    userID INTEGER,
    grade CHAR(2),
    PRIMARY KEY (major,courseNum,classNum,userID),
    FOREIGN KEY (major,courseNum,classNum) REFERENCES Class(major,courseNum,classNum),
    FOREIGN KEY (userID) REFERENCES Student(userID)
);

CREATE TABLE Assignment(
    assignmentID INTEGER,
    major VARCHAR NOT NULL,
    courseNum CHAR(3) NOT NULL,
    classNum CHAR(7) NOT NULL,
    title VARCHAR,
    deadline DATETIME,
    PRIMARY KEY (assignmentID)
    FOREIGN KEY (major,courseNum,classNum) REFERENCES Class(major,courseNum,classNum)
);

CREATE TABLE Submit (
    userID INTEGER,
    assignmentID INTEGER,
    PRIMARY KEY (assignmentID,userID),
    FOREIGN KEY (assignmentID) REFERENCES Assignment(assignmentID),
    FOREIGN KEY (userID) REFERENCES Student(userID)
);

CREATE TABLE Post (
    postID INTEGER,
    userID INTEGER, NOT NULL,
    assignmentID INTEGER,
    kind VARCHAR,
    timestamp DATETIME,
    content VARCHAR,
    popularity INTEGER,
    PRIMARY KEY (postID),
    FOREIGN KEY (userID) REFERENCES User(userID)
);

CREATE TABLE Like (
    postID INTEGER,
    userID INTEGER,
    PRIMARY KEY (postID,userID),
    FOREIGN KEY (postID) REFERENCES Post(postID),
    FOREIGN KEY (userID) REFERENCES User(userID)
);

```

---

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:9b8dfcbdfa1521ab2ea86475ef99b143>

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-B. Make sure to include the parts marked with arrows.

**Write the following queries in Relational Algebra:**

- Find all courses offered in Fall 2019. Give the courseNum, major, title, and level of those courses.

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

- 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.

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

- Find the courses which don't have an instructor but have students enrolled in it. Return the classNum, major, courseNum of those courses and the number of students enrolled in them (call this numStudents).

EnrollIn.classNum	EnrollIn.major	EnrollIn.courseNum	numStudents
2020S01	CptS	582	3
2019S01	STAT	360	2

- Find the instructors who teach at least 2 classes with enrollment limit greater than 10. Give the userIDs, first and last names, and emails of those instructors and the number of such classes they teach (call this attribute numClasses).

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

5. 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.

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

6. 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.

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

7. 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 classes and the number of students enrolled (call this numStudents).

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

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

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

### **Submission Instructions:**

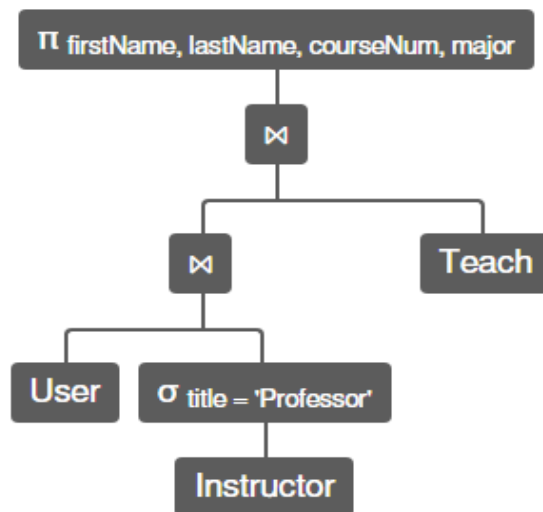
HW3 will be submitted on Blackboard (HW3 dropbox under "Homeworks"). HW3 should be submitted in .pdf format (name your file "HW3.pdf"). Please don't zip your file. Email submissions will not be accepted.

## APPENDIX-A

### Example output

- ⇒ a. Relational Algebra Expression:
- $$\pi_{\text{firstName, lastName, courseNum, major}} (\text{User} \bowtie (\sigma_{\text{title} = \text{'Professor'}} \text{Instructor}) \bowtie \text{Teach})$$

- ⇒ b. Relational algebra tree:



- ⇒ c. Expression and Output:

$\pi_{\text{firstName, lastName, courseNum, major}} (\text{User} \bowtie (\sigma_{\text{title} = \text{'Professor'}} \text{Instructor}) \bowtie \text{Teach})$

User.firstName	User.lastName	Teach.courseNum	Teach.major
Sakire	ArslanAy	451	CptS
Sakire	ArslanAy	322	CptS
Sakire	ArslanAy	355	CptS
Diane	Cook	437	CptS
Venera	Arnaoudova	321	CptS
Venera	Arnaoudova	581	CptS

APPENDIX-B

ER diagram of the DB schema

