**Name :   Musa Husseini**

WSU ID: 11658865

CptS 451 – Introduction to Database Systems
Homework-3 - Relational Algebra

1. Find the meetings which are hosted by the instructor of `CptS 321' (i.e., hosted by the instructor who teaches 'CptS 321'). Return the meeting title and the course_id for the meeting, and instructor's first and last names. Order the results by meeting course_id and meeting title.

   a. Relational algebra tree:
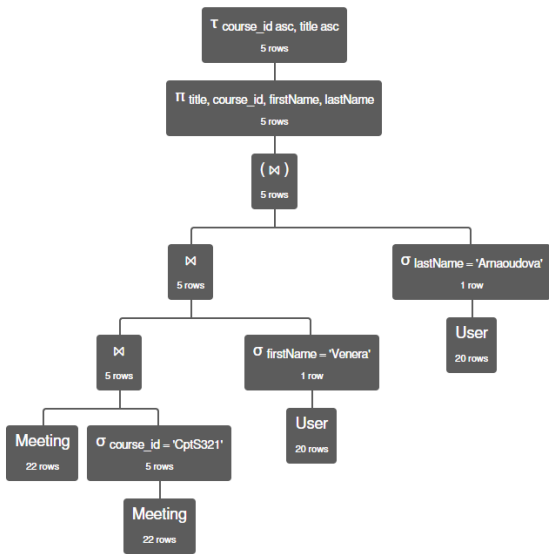
   b. Expression and Output:

$$\tau_{course\_id\ asc,\ title\ asc} (\pi_{title,\ course\_id,\ firstName,\ lastName} (((Meeting \bowtie (\sigma_{course\_id\ =\ 'CptS321'} Meeting)) \bowtie (\sigma_{firstName\ =\ 'Venera'} User)) \bowtie (\sigma_{lastName\ =\ 'Arnaoudova'} User)))$$

Meeting.title	Meeting.course_id	User.firstName	User.lastName
'Lecture1'	'CptS321'	'Venera'	'Arnaoudova'
'Lecture2'	'CptS321'	'Venera'	'Arnaoudova'
'Lecture3'	'CptS321'	'Venera'	'Arnaoudova'
'Lecture4'	'CptS321'	'Venera'	'Arnaoudova'
'Lecture5'	'CptS321'	'Venera'	'Arnaoudova'

2. Find the student users who did not post any messages. Return the student_id, email, firstname and lastname of those students.

    a. Relational algebra tree:
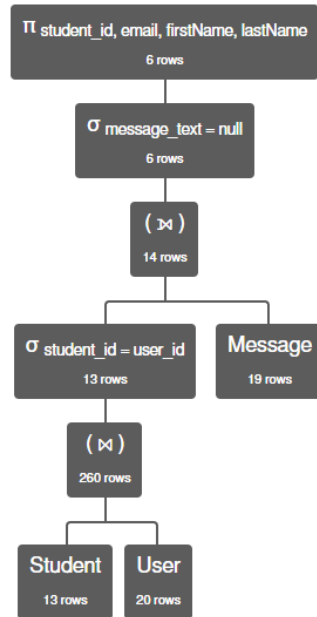
    b. Expression and Output:

$$\pi_{\text{student\_id, email, firstName, lastName}} \left(\sigma_{\text{message\_text = null}} \left(\sigma_{\text{student\_id = user\_id}} \left(\text{Student} \bowtie \text{User} \right) \bowtie \text{Message} \right) \right)$$

Student.student_id	User.email	User.firstName	User.lastName
15	'travis@wsu.edu'	'Travis'	'Person'
16	'min@wsu.edu'	'Min'	'Quin'
17	'kelly@wsu.edu'	'Kelly'	'Easton'
18	'amy@wsu.edu'	'Amy'	'Fan'
19	'ben@wsu.edu'	'Ben'	'Hill'
20	'connor@wsu.edu'	'Connor'	'Hall'

3. Find the instructors who work in the "Machine Learning" field but are not teaching any courses. Return the instructor_id, first name, and lastname of those.
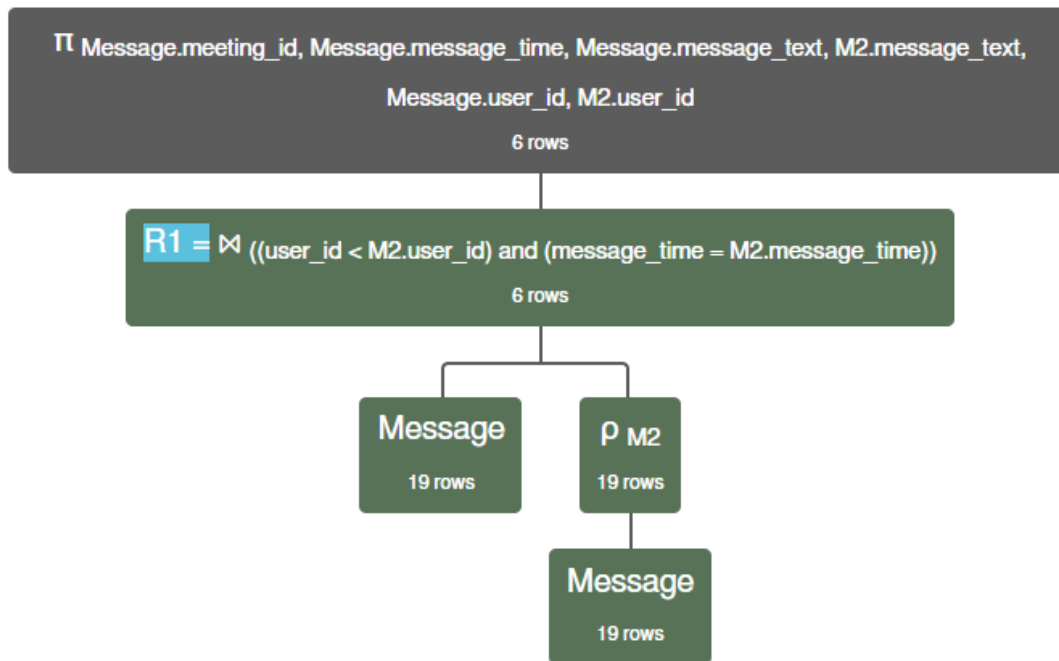
a. Relational algebra tree:
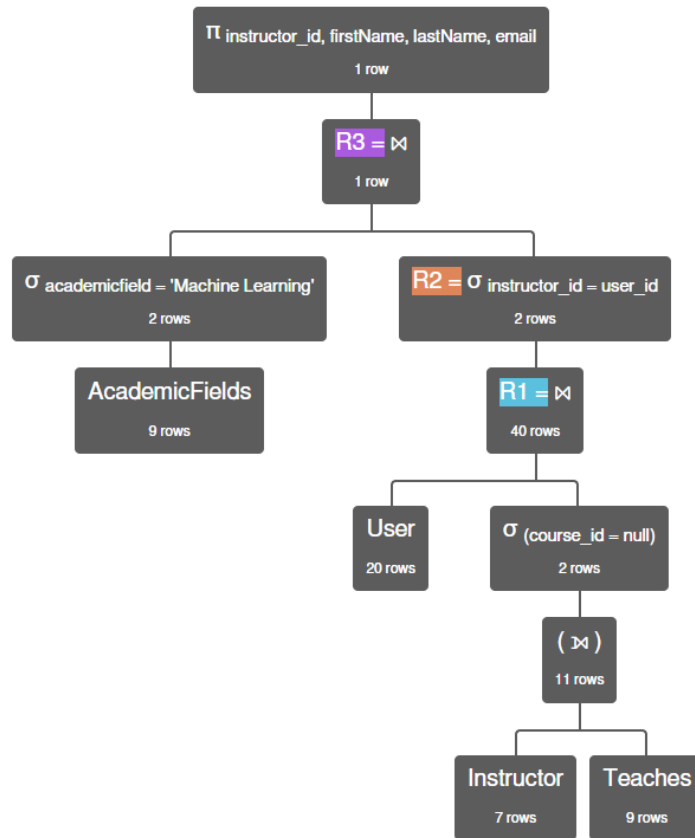
b. Expression and Output:

$\pi_{\text{instructor\_id, firstName, lastName, email}}$ (($\sigma_{\text{academicfield} = \text{'Machine Learning'}}$ AcademicFields) $\bowtie$ ($\sigma_{\text{instructor\_id} =}$ $_{\text{user\_id}}$ (User $\bowtie \sigma_{\text{(course\_id} = \text{null)}}$ (Instructor $\bowtie$ Teaches))))

AcademicFields.instructor_id	User.firstName	User.lastName	User.email
3	'Carl'	'Hauser'	'hauser@wsu.edu'

4. Find the pair of messages that are posted at the same meeting and at the same time but by different users. Return the common meeting_id and message_time of the two messages as well as the message_text and user_id of each message.

    a. Relational algebra tree:

Π Message.meeting_id, Message.message_time, Message.message_text, M2.message_text,
Message.user_id, M2.user_id
6 rows

R1 = ⋈ ((user_id < M2.user_id) and (message_time = M2.message_time))
6 rows

Message
19 rows

ρ M2
19 rows

Message
19 rows

b. Expression and Output:

$$\Pi_{\text{Message.meeting\_id, Message.message\_time, Message.message\_text, M2.message\_text, Message.user\_id, M2.user\_id}} (\text{Message} \bowtie_{((\text{user\_id} < \text{M2.user\_id}) \text{ and } (\text{message\_time} = \text{M2.message\_time}))} (\rho_{\text{M2}} (\text{Message})))$$

Message.meeting_id	Message.message_time	Message.message_text	M2.message_text	Message.user_id	M2.user_id
1	'2022-01-13 01:27pm'	'Clarification on problem-2'	'Problem-3 clarification'	5	14
1	'2022-01-13 01:30pm'	'Strong vs. weak entity'	'Do all strong entities have a primary key!'	8	9
4	'2022-01-20 01:40pm'	'All HWs should be submitted on time!'	'Is it possible to submit HW1 a little bit late?'	1	10
5	'2022-01-22 01:10pm'	'Do you have any questions?'	'Should we include the query output in our solution. '	1	11
5	'2022-01-22 01:10pm'	'HW3 is due today!'	'Should we include the query output in our solution. '	1	11
12	'2022-01-15 10:20am'	'Can we handwrite the solution?'	'Is ML hard?'	12	13

5. Find the users who are mentioned 2 or more times in the messages of CptS451 meetings. ('CptS451' is the id of the course associated with the meetings; assume we consider all meetings of 'CptS451'.) Return the user_id, email, first and lastname of the user mentioned and the number of times they are mentioned.
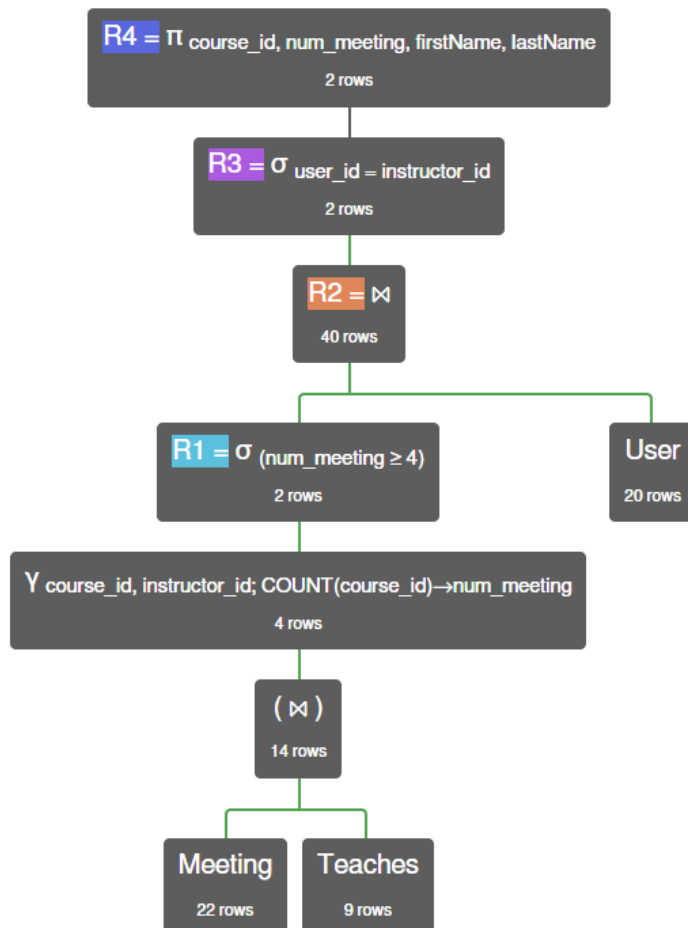
   a. Relational algebra tree:
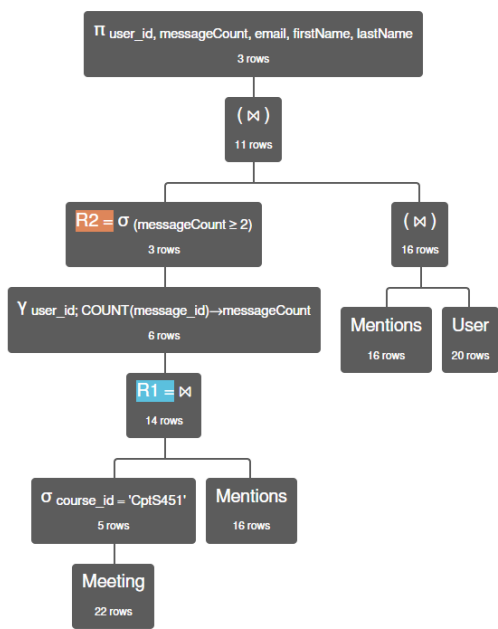
   b. Expression and Output:

π user_id, messageCount, email, firstName, lastName $((\sigma$ (messageCount ≥ 2) $(Y$ user_id; COUNT(message_id)→messageCount $((\sigma$ course_id = 'CptS451' Meeting $) \bowtie$ Mentions $))) \bowtie ($ Mentions $\bowtie$ User $))$

Mentions.user_id	messageCount	User.email	User.firstName	User.lastName
1	6	'arslanay@wsu.edu'	'Sakire'	'ArslanAy'
9	3	'noel@wsu.edu'	'Noel'	'Sam'
10	2	'andy@wsu.edu'	'Andy'	'White'

6. Find the instructors who hosted more than 3 meetings that are associated with the same course. Return the course_id, number of meetings, and the first and last name of the instructor.

a. Relational algebra tree:

R4 = π course_id, num_meeting, firstName, lastName
2 rows

R3 = σ user_id = instructor_id
2 rows

R2 = $\bowtie$
40 rows

R1 = σ (num_meeting ≥ 4)
2 rows

User
20 rows

γ course_id, instructor_id; COUNT(course_id)→num_meeting
4 rows

($\bowtie$)
14 rows

Meeting
22 rows

Teaches
9 rows

b. Expression and Output:

π course_id, num_meeting, firstName, lastName (σ user_id = instructor_id (σ (num_meeting ≥ 4) (γ course_id, instructor_id; COUNT(course_id)→num_meeting (Meeting $\bowtie$ Teaches)) $\bowtie$ User))

Meeting.course_id	num_meeting	User.firstName	User.lastName
'CptS451'	4	'Sakire'	'ArslanAy'
'CptS321'	4	'Venera'	'Arnaoudova'

7. Find the 'CptS451' students who didn't attend a meeting of the 'CptS451' course, but they watched that meeting's recording. Return the student_id, meeting's title, and recording_number of the recording student watched.

a. Relational algebra tree:
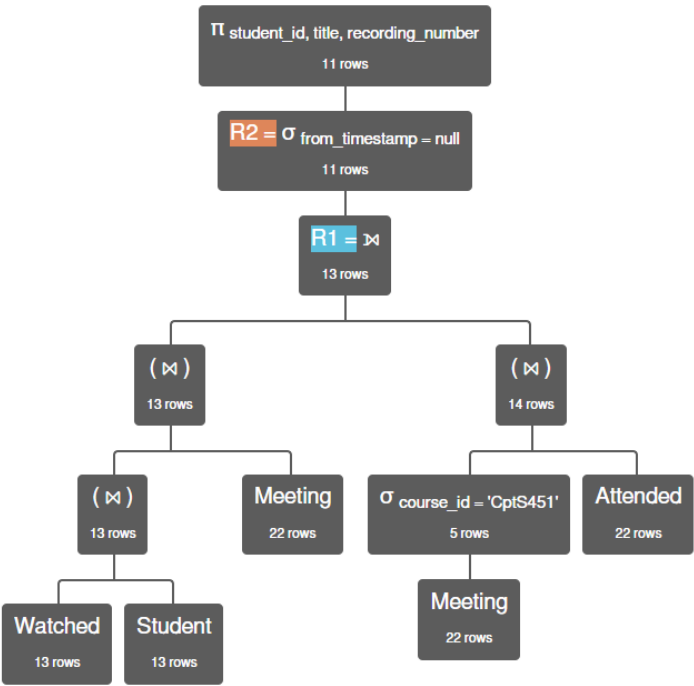
b. Expression and Output:

$\pi_{student\_id,\ title,\ recording\_number}$ ($\sigma_{from\_timestamp\ =\ null}$ (((Watched ⋈ Student) ⋈ Meeting) ⋈ ($\sigma_{course\_id\ =\ 'CptS451'}$ Meeting ⋈ Attended)))

Watched.student_id	Meeting.title	Watched.recording_number
9	'Lecture1'	1
10	'Lecture1'	1
12	'Lecture1'	1
13	'Lecture1'	1
17	'Lecture1'	1
18	'Lecture1'	1
13	'Lecture2'	1
14	'Lecture2'	1
16	'Lecture2'	1
15	'Lecture4'	1

8. Find the meetings for which the number of messages posted by instructors is greater than the number of messages posted by students. Return the meeting_id, number of messages by instructors, and the number of messages by students for those meetings.
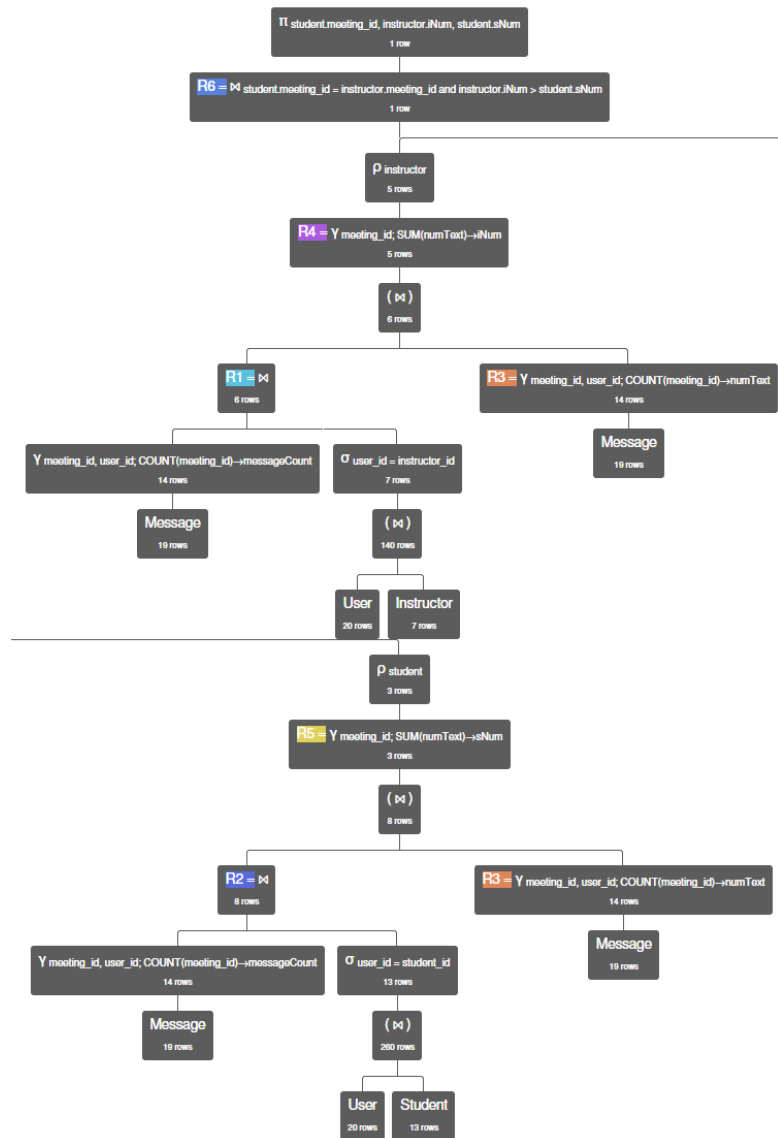
   a. Relational algebra tree:

   b. Expression and Output:

π student.meeting_id, instructor.iNum, student.sNum (ρ instructor γ meeting_id; SUM(numText)→iNum (((γ meeting_id, user_id; COUNT(meeting_id)→messageCount Message) ⋈ (σ user_id = instructor_id (User ⋈ Instructor))) ⋈ γ meeting_id, user_id; COUNT(meeting_id)→numText Message) ⋈ student.meeting_id = instructor.meeting_id and instructor.iNum > student.sNum ρ student γ meeting_id; SUM(numText)→sNum (((γ meeting_id, user_id; COUNT(meeting_id)→messageCount Message) ⋈ (σ user_id = student_id (User ⋈ Student))) ⋈ γ meeting_id, user_id; COUNT(meeting_id)→numText Message))

student.meeting_id	instructor.iNum	student.sNum
4	4	2

9. Find the longest meeting(s) (i.e., the meetings with max duration). Return the meeting_id, title and duration of those meeting(s).

    a. Relational algebra tree:
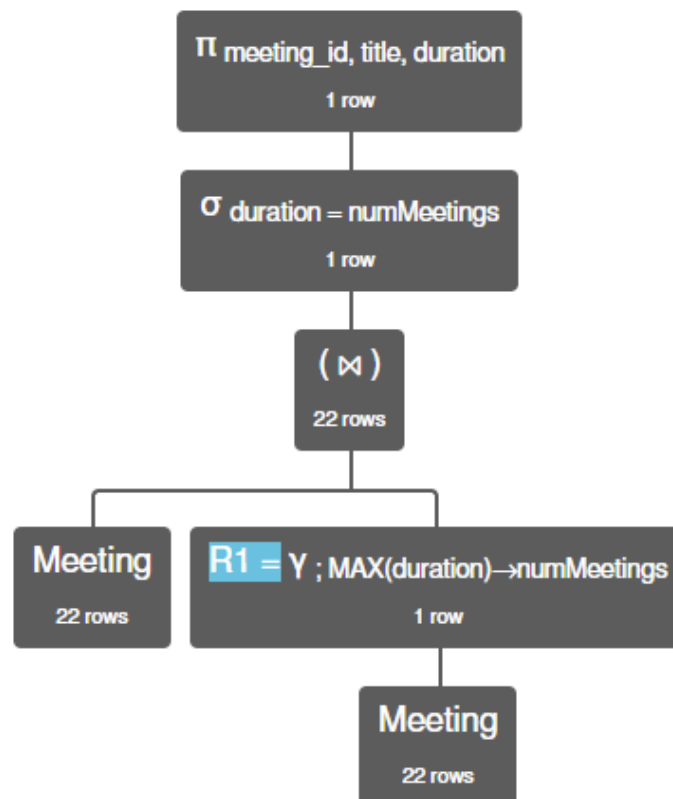
    b. Expression and Output:

Π meeting_id, title, duration $($ σ duration = numMeetings $($ Meeting $\bowtie$ γ ; MAX(duration)→numMeetings $($ Meeting $)$ $)$ $)$

Meeting.meeting_id	Meeting.title	Meeting.duration
20	'Curriculum meeting'	115

10. Find the meeting(s) with the most number of attendees. Return the meeting title and the number of attendees for those meeting(s).

　　a.　Relational algebra tree:

　　b.　Expression and Output:

π line2.title, line2.max_attended $((\rho$ line1 γ title; MAX(max_attended1)→max_attended π title, max_attended1 $($ Meeting ⋈ γ meeting_id; MAX(attended)→max_attended1 γ meeting_id; COUNT(student_id)→attended $($ Meeting ⋈ Attended $)))$ ⋈ line1.max_attended < line2.max_attended $(\rho$ line2 γ title; MAX(max_attended1)→max_attended π title, max_attended1 $($ Meeting ⋈ γ meeting_id; MAX(attended)→max_attended1 γ meeting_id; COUNT(student_id)→attended $($ Meeting ⋈ Attended $)))$

line2.title	line2.max_attended
'Lecture1'	3
'Lecture2'	3
'Lecture3'	3
'Lecture5'	3