Database Systems, CSCI 4380-01
Homework # 1
Due Monday September 14, 2020 at 11:59:59 PM

Homework Statement. This homework is worth 5% of your total grade. If you choose to skip it, Midterm #1 will be worth 5% more. Remember, practice is extremely important to do well in this class. I recommend that not only you solve this homework, but also work on homeworks from past semesters. Link to those is already provided in Teams, which I am repeating here:

`http://cs.rpi.edu/~sibel/DBS_Past_Materials/`

This homework aims to test relational algebra first and foremost, and a bit of normalization theory.

Please do the parts in sequence. The questions get harder and build on your knowledge of relational algebra from previous parts. Each question is equal weight.

Database Description. Now that we have confused you all with so many sites for so many different purposes, I assume you would like to construct a website to help manage your remote learning lifestyle. Here is a starter database idea:

`Sites(sitename, username, password, bestbrowser)`
`Classes(classcode, classname, semester, year, credits)`
`Classmeetings(classcode, dayofweek, starttime, duration, sitename, url, note)`
`Exams(classcode, examname, examdate, starttime, sitename, url, note)`
`Teaches(classcode, instructorname, email, note)`
`Officehours(classcode, dayofweek, starttime, duration, sitename, username)`
`Resources(rid, classcode, resourcetype, sitename, username, url)`

where `resourcetype` is one of `'discussions','groups','hw','exercises','coursenotes','videos'`.

Keys for each relation are underlined.

`Sites` are sites you have account on such as Submitty, Piazza, Gradescope, Webex Teams, etc. You must store your username. You can also store your password in an encoded format!

`Classes` are classes you are currently taking or have taken in the past.

`ClassMeetings` are the specific meetings for that class, including lectures and study group meetings for each. For each day and start time, you store the duration of the meeting, the site that you use for the meeting, direct url and additional notes.

`Exams` stores for each class, the date, start time, site name, direct url and any additional notes. Examname are values like `exam1, exam2, final`.

`Teaches` stores all instructors for a course and the email that they use for that course. We also added a note so that you can note all weird habits and requests of each instructor.

`OfficeHours` for each class are stored for each day of the week (`'Monday','Tuesday',..`), start time, duration, site and username to go for them.

`Resources` is the most useful relation in the database. It stores where you go for each different part of a class (hw, exercise, discussions). You store which site to go to, which username for that site to use and the direct URL for it.

Note: All date fields are formatted as `mon-day-year`, e.g. `01-31-2020`. You can assume that you can check if a date value `X` comes after another value `Y` by checking whether `X > Y`. Direct URLs link to the specific URL for a specific site. For example, Submitty is a site but the direct URL for this course on Submitty is different than the direct URL for another course.

Question 1. Write the following queries using relational algebra. You may use any valid relational algebra expression, break into multiple steps as needed. However, please make sure that your answers are well-formatted and are easily readable. Also, pay attention to the attributes required in the output!

A Return `classname`, `examname` and `examdate` of all exams with `examdate` after November 1st of 2020 that the instructor named `Fogg` is giving.

```
T1(classcode, instructorname, email, note1) = Teaches
T2 = Classes * Exams * T1
T3 = select_{examdate > 09-01-2020 and instructornname = 'Fogg'}
Result = project_{classname, examname, examdate} T3
```

B Return `classcode, dayofweek and startime` of class meetings that conflict with an office hour for the same class (i.e. start at the same time on the same day).

```
T1(classcode1, dayofweek1, starttime1, duration1,
sitename1, username1) = Officehours
T2 = T1 join_{classcode1=classcode and starttime1=starttime and
dayofweek1=dayofweek} Classmeetings
Result = project_{classcode, dayofweek, starttime} T2
```

C Return `classcode, classname` of all classes that meet on mondays (`dayofweek`).

```
T1 = select_{dayofweek='Monday'} Classmeetings
Result = project_{classcode, classname} Classmeetings * Classes
```

D Return the username and best browser for all sites to be used for `hw` in a class taught in `Fall 2000` (`classes.semester`, `classes.year`).

```
T1 = Classes * Resources * Sites
T2 = Select_{semester='Fall' and year=2000} T1
Result = project_{username, bestbrowser} T2
```

E Find the `sitename` of all sites used in all classes for class meetings, exams, office hours or other resources.

```
T1 = project_{sitename} Classmeetings
T2 = project_{sitename} Exams
T3 = project_{sitename} Officehours
T4 = project_{sitename} Resources
Result = T1 ∪ T2 ∪ T3 ∪ T4
```

F Find the name of a pair of courses that have no sites in common for any activity, i.e. class meetings, exams, office hours or other resources.

```
T1(classcode1, dayofweek1, starttime1, duration1,
sitename1, username1) = Officehours
T2(classcode2, examname2, examdate2, starttime2, sitename2, url2, note2) = Exams
T3(classcode3, dayofweek3, starttime3, duration3,
sitename3, url3, note3) = Classmeetings
T4 = T1 join_{classcode1=classcode2} T2
T5 = T4 join_{classcode2=classcode3} T3
```

```
T6 = project_{classcode1, sitename1, sitename2, sitename3} T4
T7(classcode2, sitename4, sitename5, sitename6) = T6
Result = select_{classcode1<>classcode2
and sitename1<>sitename4 and sitename1<>sitename5 and sitename1<>sitename6 and
sitename2<>sitename4 and sitename2<>sitename5 and sitename2<>sitename6
and sitename3<>sitename4 and sitename3<>sitename5 and sitename3<>sitename6} T7
```

G Find the code and name of courses with a single instructor in the database.

```
T1(classcode1, instructorname1, email1, note1) = Teaches
T2 = Teaches join_{classcode=classcode1 and instructorname<>instructorname1} T1
T3 = project_{classcode} Classes - project_{classcode} T2
Result = project_{classcode, classname} T3 * Classes
```

H Return the course code of all courses in `Fall 2000` (`classes.semester`, `classes.year`) with office hours that are not on `Monday, Wednesday`. Return also the start time and duration for each office hour.

```
T1 = Classes * Officehours
T2 = select_{semester='Fall' and year=2000 and dayofweek<>'Monday'
and dayofweek<>'Wednesday' } T1
Result = project_{classcode, starttime, duration} T2
```

Question 2. For the following relations, (a) find and list the keys, (b) check whether they satisfy BCNF, discuss why or why not, (c) check whether they satisfy 3NF, discuss why or why not.

To show that a relation is not in BCNF or 3NF, you only need to show a violation. To show that they are in BCNF or 3NF, check each functional dependency and discuss why it is ok.

A $R1(A, B, C, D, E, F)$, $\mathcal{F} = \{AC \to DE, BD \to F\}$

- A $ABC^+ = \{A, B, C, D, E, F\}$
  keys: $[ABC]$
- B This does not satisfy BCNF because in relation $AC \to DE$, AC is neither a superkey nor is the relation trivial
- C This does not satisfy 3NF because in relation $AC \to DE$, AC is neither a superkey nor is the relation trivial, and DE is not prime attribute

B $R2(A, B, C, D, E, F)$, $\mathcal{F} = \{ABC \to DEF, AB \to A, BCD \to AEF\}$

- A $ABC^+ = \{A, B, C, D, E, F\}$
  $BCD^+ = \{B, C, D, A, E, F\}$
  keys: $[ABC, BCD]$
- B This satisfies BCNF:
  $ABC \to DEF$: ABC is a superkey
  $AB \to A$: This functional dependency is trivial, given that A and any other attribute(s), would imply A(itself)
  $BCD \to AEF$: BCD is a superkey
- C This satisfies 3NF, these functional dependencies are all valid under BCNF

C $R3(A, B, C, D, E, F)$, $\mathcal{F} = \{ABC \to DE, BC \to AF\}$

A $BC^+ = \{B, C, A, F, D, E\}$
keys: $[BC]$

B This satisfies BCNF:
$ABC \rightarrow DE$: ABC is a superkey
$BC \rightarrow AF$: BC is a superkey

C This satisfies 3NF, these functional dependencies are all valid under BCNF

D $R4(A, B, C, D, E, F)$, $\mathcal{F} = \{ABC \rightarrow DEF, BD \rightarrow A\}$

A $ABC^+ = \{A, B, C, D, E, F\}$
$BCD^+ = \{B, C, D, A, E, F\}$
keys: $[ABC, BCD]$

B This satisfies BCNF:
$ABC \rightarrow DEF$: ABC is a superkey
$BD \rightarrow A$: BD is not a superkey nor is the functional dependency trivial

C This satisfies 3NF: $ABC \rightarrow DEF$: ABC is a superkey
$BD \rightarrow A$: on the right hand side, A is a prime attribute.

SUBMISSION INSTRUCTIONS. Submit a PDF document or a Text document for this homework using Submitty. No other format and no hand written homeworks please. No late submissions will be allowed.

If the Submitty for homework submissions is not immediately available, we will announce it on Teams when it becomes available.

Help with relational algebra formatting. While in class I have been using a text version of relational algebra, which I have allowed for many years for students who do not want to figure out the Greek symbols. However, many past solutions use the more standard version with Greek symbols. You can use either one in your solutions, but do not mix and match. Use one consistently.

I present you with the full syntax here in both ways (as well as the Latex symbols for it). Note that for the standard version, I simply use the Math mode in Latex.

Operation	Text Version	Standard Version
Set Union	R union S	$R \cup S$
Set Intersection	R intersect S	$R \cap S$
Set Difference	R - S	$R - S$
Rename	T(A,B,C) = R	$\rho_{T(A,B,C)}(R)$
Select	select_{C} (R)	$\sigma_C(R)$
Project	project_{A1,..,An} (R)	$\pi_{A1,..,An}(R)$
Cartesian product	R x S	$R \times S$
Theta-Join	R join_{C} S	$R \bowtie_C S$
Natural Join	R * S	$R * S$

As an additional help, I format one of the queries we did in class in the standard format below for two equivalent solutions.

Query: Find name of Marvel heroes who have starred in a movie

Solution 1 (text format):

T1(hid1, mid1) = HeroInMovie
Result = project_{hero} (select_{hid=hid1} (MarvelHeroes x T1))

Solution 1 (in standard format):

$$\begin{aligned} T1(hid1, mid1) &= HeroInMovie \\ Result &= \pi_{hero}(\sigma_{hid=hid1}\ (MarvelHeroes \times T1)) \end{aligned}$$

Solution 2 (text format):

T1(hid1, mid1) = HeroInMovie
Result = project_{hero} (MarvelHeroes join_{hid=hid1} T1))

Solution 2 (in standard format):

$$\begin{aligned} T1(hid1, mid1) &= HeroInMovie \\ Result = \pi_{hero}(MarvelHeroes &\bowtie_{hid=hid1} T1)) \end{aligned}$$