Database Systems, CSCI 4380-01
Homework # 1
Due Friday February 10, 2023 at 11:59:59 PM EST

Homework Statement

This homework is worth 5% of your total grade. Remember, practice is extremely important to do well in this class. I recommend that not only you solve this homework, but also work on select homework assignments from past semesters that can be found at the following link:

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

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

You must show work for all parts of this homework in order to get credit. If you provide just the final answer but fail to show work, you will get a zero, even if the answer is correct.

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

We would like you to construct a website to help manage classes and resources. 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 `month-day-year`, e.g. `01-31-2021`. You can assume that you can check if a date value X comes after another value Y by checking whether $X > Y$. All time fields are formatted as `hour:minute` (using the 24-hour "military" clock), e.g. `22:56`. You can assume that you can check if a time 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, and break it 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`, `instructorname`, `examname`, `examdate`, and `starttime` of all exams given in `Spring 2023` (`Classes.semester`, `Classes.year`) with `startime` before 12:00 PM.

To find all exams given in Spring 2023 with starttime before 12:PM, we first natural-join Classes, Exams, and Teaches to get all relations we need and name it as T1. Then we perform a selection and projection on the resulting table to get matched results.

$T1$ = Classes * Exams * Teaches
$T2 = \sigma_{\text{semester = Spring and year=2023 and starttime<12:00}}(T1)$
$\pi_{\text{classname, instructorname, examname, examdate}}(T2)$

(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).

There are many attributes in common between Classmeetings and Officehours. Therefore we could first perform a projection on classcode, dayofweek, and starttime on both relations and natural join on the resulting tables to find the matched items.
$T1 = \pi_{\text{classcode, dayofweek, starttime}}(\text{Classmeetings})$
$T2 = \pi_{\text{classcode, dayofweek, starttime}}(\text{Officehours})$
$T1 * T2$

(c) Return `classcode, classname` of all classes that have office hours on Mondays and Thursdays (`dayofweek`).

We can first find the classcode by selecting on officehours relation based on the monday and thursday constraint. Then we can project the classcode and classname attributes of the Classes relation and then perform a natural join on these two relations.
$T1$(classcode1, dayofweek1) := Officehours
$T2 = T1 \bowtie_{\text{classcode = classcode1 dayofweek = 'Monday' and dayofweek1 = 'Thursday'}}$ (Officehours)
$T3 = \pi_{\text{classcode,classname}}$(Classes)
$\pi_{\text{classcode,classname}}(T1 * T2)$

(d) Find the `sitename, bestbrowser` of all sites to be used for `discussions` in all classes taught by the instructor named `Pitt`.

We first find the classcode of courses taught by pitt by selecting and projecting from the Teaches relation and name it T1. Then we find the sitename where resourcetype is 'discussion' and the classcode matches T1 from the Resources relation by performing a natural on T1 and name it T2. Then we perform a natural join of T2 with Sites to find the corresponding bestbrowser and do a projection.

$T1 = \pi_{\text{classcode}}(\sigma_{\text{instructorname = 'pitt'}}$(Teaches))
$T2 = \pi_{\text{sitename}}(\sigma_{\text{resourcetype='discussions'}}$(T1 * Resources))
$\pi_{\text{sitename,bestbrowser}}$(T2 * Sites)

(e) Find all days of week (`dayofweek`) which are used either for class meetings or office hours.

We can perform a projection on the dayofweek attribute on both the Classmeetings and Officehours attributes and then do a set Union.
$T1 = \pi_{\text{dayofweek}}$(Classmeetings)
$T2 = \pi_{\text{dayofweek}}$(Officehours)
$T1 \cup T2$

(f) Find the names of pairs of instructors that have no sites in common in any of their classes for any activity, i.e. class meetings, exams, office hours, or other resources.

We first find the projection of Classmeetings, Exams, Officehours, and resources relations natural join with Teaches on instructorname and sitename (T5). Then we create another relation called T6 equals to T5. Then we do a theta-join on T5 and T6 with the condition that instructorname on T5 and T6 does not equal to find the Cartesian product between T5 and T6 where instructorname are different in the pair and name it T7. We then perform another theta-join on T5 and T6 where instructorname are both different but sitename are the same and name it T8. We can find the instructor pairs with no sitename in common by subtracting T7 with T8 (considering an instructor could have multiple sites).
$T0 = \pi_{\text{classcode,instructorname}}$(Teaches)
$T1 = \pi_{\text{instructorname,sitename}}$(Classmeetings * T0)
$T2 = \pi_{\text{instructorname,sitename}}$(Exams * T0)
$T3 = \pi_{\text{instructorname,sitename}}$(Officehours * T0)
$T4 = \pi_{\text{instructorname,sitename}}$(Resources * T0)
$T5 = T1 \cup T2 \cup T3 \cup T4 \cup T5$
$T6$(instructorname1,sitename1) := $T5$
$T7 = T5 \bowtie_{\text{instructorname1<>instructorname}} T6$

$T8 = T5 \bowtie_{\text{instructorname1<>instructorname and sitename = sitename1}} T6$
$\pi_{\text{instructorname,instructorname1}}(T7 - T8)$

(g) Find the code and name of classes with no more than one exam.

To find the code and name of classes with no more than one exam, we find the code of classes with more than one exam and find the set difference of all the class code with the class code we found and theta join it with the classes relation to project the classcode and classname.

$T1(\text{classcode1,examname1}) := \pi_{\text{classcode,examname}}(\text{Exams})$
$T2 = \pi_{\text{classcode}}(T1 \bowtie_{\text{classcode=classcode1 and examname<>examname1}} \text{Exams})$
$T3 = \pi_{\text{classcode}}(\text{Classes}) - T2$
$\pi_{\text{classcode,classname}}(T3 * Classes)$

(h) Return the class code of all courses in `Spring 2023` (`Classes.semester`, `Classes.year`) that have their exam date and start time (`Exams.examdate`, `Exams.starttime`) conflicting with an exam in another course (i.e., start at the same time on the same date). Return also the course name, exam date, start time, and the name of the instructor for each course with such a conflict.

We first find the classcode of all courses in spring 2023. Then we do find the corresponding exam date and time and do a thetajoin with it self to find the conflicting courses. We use the classcode to find the corresponding instructorname.

$T1 = \sigma_{\text{semester = 'Spring' and year = 2023}}(\text{Classes})$
$T2 = \pi_{\text{classcode,classname,examdate,starttime}} T1 * Exams$
$T3(\text{classcode1,examdate1,starttime1}) := T2$
$T4 = T2 \bowtie_{\text{classcode<>classcode1 and examdate=examdate1 and starttime=starttime1}} T3$
$\pi_{\text{classcode,classname,examdate,starttime,instructorname}}(T4 \bowtie_{\text{classcode = classcode1}} \text{Teaches})$

Question 2

For the following relations, find and list all the keys.

1. $R1(A, B, C, D, E, F)$, $\mathcal{F} = \{AE \to F, BD \to EA\}$

   There is no B, C, and D on the right-hand side, so they must be the key. We know $BD \to EA$ which gives us A and E. $AE \to F$ gives us F. No other functional dependencies other than BD gives us EA so $\{B, C, D\}$ is the key.

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

   C is not in the righthand side of any functional dependencies, so C must be in the keys. $C \to B$ gives us B. To deduce more values, we can either add A or D to the keys, and we will discuss them case by case.

   If we add A to the keys, we would have $ABC \to DEF$. Thus we can deduce all the values A,B,C,D,E,and F from key $\{A, C\}$

   If we add D to the keys, we would have $BCD \to EBF$. Immediately we notice that we cannot deduce A from this functional dependency and thus $\{C, D\}$ is not a key.

   If there are any other keys than $\{A, C\}$, it would not be able to use B for deduction, but every functional dependency has B on the left-hand side. Therefore the key is $\{A, C\}$.

3. $R3(A, B, C, D, E, F)$, $\mathcal{F} = \{ABCF \to E, C \to A\}$

   We do not have B,C,D,F on the right-hand side of any functional dependencies, so they must be in the keys. Since we have $C \to A$, we can deduce A, B, C, D, F. Since $ABCF \to E$, we can deduce all values in this relation. Therefore the key is $\{B, C, D, F\}$ as it is the minimal set required.

4. $R4(A, B, C, D, E, F)$, $\mathcal{F} = \{ABC \to DEF, BD \to C, AF \to DE, E \to FA\}$

   We immediately find we can deduce all the attributes in this relation from ABC as $ABC \to DEF$. Since any combination of A,B, and C does not cover all the sets, $\{A, B, C\}$ is a key. We can get $\{A, B, C\}$ from $\{A, B, D\}$ so $\{A, B, D\}$ is a key. We also notice $\{A, B, F\}$ is a key since $AF \to DE$ and $BD \to C$. Since B is not on the right-hand side of any functional dependencies, we know B is in the set of keys. If the key has only two elements, we can add D or E to deduce more attributes in this relation.

   If we add D in to the key set, we can deduce $BD \to C$. Thus we have B, C, D and can not deduce further.

   If we add E in to the key set, we can deduce $E \to FA$. Thus we have B, E, F, A and can further deduce D, E as $AF \to DE$. Then we have B, E, F, A, D. Then we can deduce C as $BD \to C$. Therefore we can deduce all the values if B, E is the key.

   From the discussion we had above, we know the keys are $\{A, B, C\}, \{A, B, D\}, \{A, B, F\}$, and $\{B, E\}$.

5. $R5(A, B, C, D, E, F)$, $\mathcal{F} = \{AB \to C, BC \to D, CD \to E, DE \to F, EF \to C\}$

   A and B are not on the right-hand side of any functional dependencies, so A and B must be in the keys. We can deduce C as $AB \to C$. Then we can deduce D as $BC \to D$. Then we can deduce E as $CD \to E$. Then we can deduce F as $DE \to F$. Since A and B are required to be in the key and we can deduce all the values from $\{A, B\}$, the key is $\{A, B\}$

Submission instructions

Submit a computer-generated PDF document or a text document for this homework using Submitty. No other format and no hand written homework please.

Help with relational algebra formatting

In class I have been using a text version of relational algebra. 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 names 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):

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