Homework Due 2022-04-22 by 23:59 New York Time

Contents

1 General Instructions

1. You need to follow carefully the instructions for the assignment as written below.

   It is advisable to print out this document and check off various points as they are addressed. It is easy to miss something when switching between the assignment and the solution on a single screen, especially on a laptop with a relatively small screen.

   If you do not have access to a printer, at least review your solution before the submission to make sure that you followed the instructions and that you did all that you were requested to do and only what you were requested to do.

2. If you want to refer to a specific line in this document, refer to the small numbers in the left margin.

3. If you have questions concerning this homework

   - for questions 1 and 2 email Chloe Yuan, `mailto:yy3754@nyu.edu`,
   - for questions 3 and 4 email Shreya Gupta, `mailto:sg6606@nyu.edu`,

   in the way specified in the course description. Note, however, that you should not ask for help in producing your submission. If you need help in understanding the material required, contact Zvi Kedem, `mailto:zk1@nyu.edu` *in the way specified in the course description.*

   To be sure that you get an answer to your question before the submission deadline, *do not delay your question to the date on which the assignment is due.*

   If you still have unresolved questions, email Zvi Kedem, `mailto:zk1@nyu.edu`, including all relevant correspondence with the assistant(s) listed above, *in the way specified in the course description.*

4. Submit your homework in an electronic form by uploading it to Brightspace by the due date and time. Use only permitted software and format. E.g., if you are asked for a relational database specification using SQL Power Architect than that's what you must submit.

   Do not package the files you need to submit in an archive unless specifically asked to do that.

5. If you submit a scanned, handwritten assignment when permitted, it has to be written neatly, that is, it should be neatly divided into lines just as a typeset document, etc. You may submit a handwritten assignment only when that is explicitly allowed. And, unless stated otherwise, you must submit such a handwritten assignment as a file in PDF format only.

6. It is important that you follow the directions precisely. Also, please *check* that you submitted what you intended to submit, as you are responsible for making sure of that. The best way to do is to download what you submitted to check that.

   And the best way to manage your work is to dedicate a folder/directory to each assignment.

7. Until the deadline you may resubmit your homework as many times as you like and you may want to submit it relatively frequently in case something happens to your partial work on your machine. If you submit your homework after the deadline, it may not be noticed or evaluated.

8. Do not email your submission to any of the assistants. If you did not submit your solution on time, please email Zvi Kedem, `mailto:zk1@nyu.edu`, *in the way specified in the course description* with an explanation of what has happened, and if you have a solution (possibly partial), email the solution also.

   If you do need to submit the solution by email, and *only* if you need to submit by email because you are late or for other reasons, please follow the format as described next. Assuming that you are submitting your solution to Homework due 2034-02-15 and your Net ID is abc123, all the files of your homework should be emailed as a zip file named `20340215abc123.zip`. Of course you need to specify the correct date and the correct Net ID.

   Do not communicate with any of the graders concerning a late submission.

9. Be sure to follow the academic integrity rules listed in the posted syllabus. The department, the GSAS, and NYU treat academic integrity very seriously and we are required to report all possible violations.

10. Under some circumstance, we may be able to extend a deadline on request, but generally only on a one-by-one case. All such requests need to be addressed to Zvi Kedem, `mailto:zk1@nyu.edu` *in the way specified in the course description*, as soon as possible and preferably before the deadline, and with a reason for such a request.

2 Homework

Reminder: If you are not officially registered in the class and the class does not show on Albert for you, do not submit any assignments.

Please read and follow carefully the instructions in Section 1.

2.1 Description

This is an assignment dealing with logical design of relational databases.

The assignment is not very time-consuming, if you know the material, but it is rather long because it

- Reviews some important class material
- Elaborates on some material more than was done in class

2.2 Submission format

You need to submit a typeset solution, but you may simplify your typesetting by writing, e.g., (AB)+ instead of AB^+ and A -> B instead of $A \rightarrow B$, and similar. Just make sure that what you submit is clear and unambiguous. Of course, you do not need to use tables to list the Old and the New sets of FDs as is done in Example 1 below, but clearly state which set is which, as applicable to the question and the answer, and number the FDs; similarly to what we had in the class slides.

After using a text editor of your choice, save your document in the PDF format and submit as stated in Section 3.4.

3 Assignments

3.1 Introduction

1. Unless specified otherwise, you have to show *all* of your work. We cannot look into your mind and determine your thinking process unless it is written out. You may refer to the numbers of the FDs as you refer to the original FDs or as you number FDs in the new sets you get to simplify your explanation.

2. You have to follow the procedures we covered in class and not any other procedures for what's asked for. You must use only the algorithmic techniques and not the ad-hoc approach that was described using the small university example in the class, which may or may not work.

3.2 Review of the procedure

For your convenience a summary is provided. It may or may not be helpful for any specific question.

Input: A relational schema (informally, a relation or table) R and a set of FDs α.

Output: A set of relational schemas (informally, relations or tables) such that the decompostion (a formal term for the set) satisfies the conditions

1. It is lossless join
2. It preserves dependencies
3. The resulting tables are in 3NF.

Procedure:

1. Find a minimal cover for α, say ω
2. Produce a relation from each FD in ω by combining the attributes from both the LHS and the RHS
3. One by one, remove a relation that is a subset (proper or not) of another relation
4. If (at least) one of the relations contains a key of R, you are done
5. Otherwise, add a relation whose attributes form a key for R

Example 1. Assume that I am supposed to do that for

1. Relational Schema R = ABCDEF, with the following set of FDs
2. α:
   1. AA → ABB
   2. AB → C
   3. B → C
   4. E → F
   5. F → E

I will start with the given set as **Old**, and I will attempt to simplify it by producing a "candidate" set **New**. In general, the two sets are not equivalent. I can replace **Old** by **New** if and only if I can prove equivalence.

Let's start. Below we have a table with the current **Old** and the proposed **New**.

Old	**New**
1. AA → ABB	**1. A → B**
2. AB → C	2. AB → C
3. B → C	3. B → C
4. E → F	4. E → F
5. F → E	5. F → E

New was obtained by removing the "defective" parts of **Old** (which appeared only in 1.), which produces a trivially equivalent **New**, so there is nothing to do for proving the equivalence. **New** now becomes **Old**.

Old	**New**
1. A → B	1. A → B
2. AB → C	**2. A → C**
3. B → C	3. B → C
4. E → F	4. E → F
5. F → E	5. F → E

We attempt to simplify 2. in **Old** getting 2. in **New**. **New** could only be stronger, so to prove equivalence we need to prove 2. in **New** from all the FDs in **Old**.

We compute A^+ = ABC, and as it contains C, we proved equivalence. **New** becomes **Old**.

Old	New
1. A $\rightarrow$ B	**1. A $\rightarrow$ BC**
2. A $\rightarrow$ C	2. B $\rightarrow$ C
3. B $\rightarrow$ C	3. E $\rightarrow$ F
4. E $\rightarrow$ F	4. F $\rightarrow$ E
5. F $\rightarrow$ E	

We attempt to simplify 1. and 2. in **Old** getting 1. in **New**. This is an application of the union rule, which always produces an equivalent set. So there is nothing to prove/check and **New** becomes **Old**.

Old	New
1. A $\rightarrow$ BC	**1. A $\rightarrow$ C**
2. B $\rightarrow$ C	2. B $\rightarrow$ C
3. E $\rightarrow$ F	3. E $\rightarrow$ F
4. F $\rightarrow$ E	4. F $\rightarrow$ E

We will try something that does not work, just to have an example. You do *not* need to try transformations that you believe do not work (if your belief is correct).

We attempt to simplify 1. in **Old** getting 1. in **New**. **New** could only be weaker, so to prove equivalence we need to prove 1. in **Old** from all the FDs in **New**.

We compute $\mathrm{A}^+ = \mathrm{AC}$. As BC is not in AC, we proved non-equivalence. So we continue with **Old** (and do not replace it by **New**).

Old	New
1. A $\rightarrow$ BC	**1. A $\rightarrow$ B**
2. B $\rightarrow$ C	2. B $\rightarrow$ C
3. E $\rightarrow$ F	3. E $\rightarrow$ F
4. F $\rightarrow$ E	4. F $\rightarrow$ E

We attempt to simplify 1. in **Old** getting 1. in **New**. **New** could only be weaker, so to prove equivalence we need to prove 1. in **Old** from all the FDs in **New**.

We compute $\mathrm{A}^+ = \mathrm{ABC}$. As BC is in ABC, we proved equivalence.

We are done, but in general we need to make sure that no simplifications are possible. Here this is trivial because

1. There are no defective parts
2. The union rule cannot be applied
3. Both sides of all the FDs consist of one attribute only, so no simplification is possible (generally, this step will require more work because even in a minimal cover there may be several attributes in some sides of some FDs, and we need to check whether some of them should be removed).

Our minimal cover ω is

1. A $\rightarrow$ B
2. B $\rightarrow$ C
3. E $\rightarrow$ F
4. F $\rightarrow$ E

We get the following tables/relations

1. AB
2. BC
3. EF
4. FE

We remove FE because it is a subset of EF and we get

1. AB
2. BC
3. EF

We now proceed to get a global key for R. Perhaps one of the 3 tables already includes a global key for R. We compute

1. $AB^+ = ABC$
2. $BC^+ = BC$
3. $EF^+ = EF$

but we need R = ABCDEF.

We examine all the attributes of R accounting for all the FDs that are satisfied. It may be simplest to use the minimal cover, so we will do that. But you can use any equivalent set of FDs, including the initial set given to us.

We classify the attributes based on where they appear in the FDs:

1. On both sides: B, E, F
2. On left side only: A
3. On right side only: C
4. Nowhere: D

AD appear in every key. C does not appear in any key. B, E, and F may appear in a key. We start with ABDEF, which must contain a key. We cannot remove AD but may try to remove B, E, and F.

We attempt to remove B. We compute $ADEF^+ = ABCDEF$. We can remove B and we continue with ADEF.

We attempt to remove E. We compute $ADF^+ = ABCDEF$. We can remove E and we continue with ADF.

We attempt to remove F. We compute $AD^+ = ABCD$. We cannot remove F and we continue with ADF.

Nothing else can be done. To remind: AD have to be in every key and we cannot remove F from ADF (because we tried to remove it and did not succeed).

As nothing else can be done, ADF is a global key. (ADE is another global key, but we are only looking for one.)

Our final decomposition is

1. AB
2. BC
3. EF
4. ADF

3.3 Assignments

1. You are given a relational schema R = ABCDEFG satisfying the set of FDs α:
   1. $AB \rightarrow ADF$
   2. $C \rightarrow D$
   3. $CD \rightarrow E$
   4. $E \rightarrow FG$

   (a) Compute ω, a minimal cover for α. You do not need to prove that what you claim to be a minimal cover is in fact a minimal cover.

   Review Example 1.

   Do not skip any steps in your solution. That means that:

   i. Whenever your want to show an equivalence of two sets of FDs, you need to *explicitly* compute the closure of an appropriate set of attributes using an appropriate set of FDs.

   ii. You do *not* need to prove that you have actually obtained a minimal cover. (You will be asked for such a proof in another problem.)

2. You are given a relational schema R = ABCDEFGHI satisfying the set of FDs α:

   1. A $\to$ ADE
   2. B $\to$ BC
   3. B $\to$ I
   4. C $\to$ FG
   5. D $\to$ EG
   6. F $\to$ AG
   7. BG $\to$ HI

   (a) Compute ω, a minimal cover for α. You do not need to prove that what you claim to be a minimal cover is in fact a minimal cover.

   Review Example 1.

   Do not skip any steps in your solution. That means that:

   i. Whenever your want to show an equivalence of two sets of FDs, you need to *explicitly* compute the closure of an appropriate set of attributes using an appropriate set of FDs.

   ii. You do *not* need to prove that you have actually obtained a minimal cover. (You will be asked for such a proof in another problem.)

3. You are given a relational schema R = ABCDEF satisfying the set of FDs:

   1. A $\to$ D
   2. AB $\to$ C
   3. D $\to$ AE

   (a) Find all the keys of R using the procedure with all the heuristics in slides 7:130–134.

   Review Example 1.

   Two points:

   - The procedure there is described in the context of a specific example, but it can be applied to other cases, including this problem.
   - Make sure you find all the keys and not just one of them (if there are more keys that just one in this problem)

   Important comment to clarify the general situation: In general, it is not feasible to find all the keys of a relation as there may be so many of them that you may not have the computer time/space to write all of them down.

   Consider the example of R with the 12 attributes A1, A2, A3, AR, A5, A6, B1, B2, B3, BR, B5, B6 and FDs: A1 $\to$ A2, A2 $\to$ A3, A3 $\to$ A4, A4 $\to$ A5, A5 $\to$ A6, A6 $\to$ A1, B1 $\to$ B2, B2 $\to$ B3, B3 $\to$ B4, B4 $\to$ B5, B5 $\to$ B6, B6 $\to$ B1.

   Then there are 36 keys: all the sets Ai Bj for i, j = 1,...,6.

   (b) Prove that the given set of FDs is already a minimal cover. To do that show that no simplification is possible. Attempt all simplifications.

   To show that a simplification is not possible, you will need to *explicitly* compute the closures of the relevant sets of attributes using the FDs in a relevant set of FDs. Do not use any shortcuts or heuristics.

(c) Produce a decomposition satisfying the conditions for **Output** in Section 3.2.

4. You are given a relational schema R = ABC satisfying the set of FDs:
   1. AC → B
   2. B → C

   (a) Show that R is not in BCNF.

   (b) Show that it is in 3NF. Follow the definitions in slide 7:213 to produce your proof.

   Important hint: Do not rely on the informal/heuristic discussion in slides through 7:104. Although the ideas/procedures presented there are very frequently used by the practitioners, not all of them are sufficiently correct (if there is a meaning to "sufficiently correct").

   Let's elaborate on the pitfalls of using the informal discussion. Note that there are two primary keys possible AC and AB. Let's consider them in turn.

   - Choose AC as the primary key. B → C is not problematic as it is a transitive dependency

     So R is in 3NF.

   - Choose AB as the primary key. B → C is problematic as it is a partial dependency

     So R is not even in 2NF—or so it seems.

   But if you look at the definitions in slide 7:213, you will see that R is indeed in 3NF

   Comment. Practitioners frequently seem to assume that non-primary keys can be ignored, and therefore if AC is chosen as the primary key and the possibility of choosing AB as the primary key is ignored, then they are not bothered about B → C.

   It indeed, is not a problem for the 3NF condition in this case, but it is worth understanding why.

   It is also worth knowing that attributes that appear in *any* key are referred to as *prime attributes*.

3.4 What to submit

Please upload 1 file, named *exactly* as specified and in the format *exactly* as specified.

1. `logicalDesign08.pdf`, the file with your solution.