Homework Due 2021-12-04 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 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 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.*

   Note, however, that you should not ask Shreya Gupta, `mailto:sg6606@nyu.edu`, 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.*

4. Submit your homework in an electronic form by uploading it to NYU Classes 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.

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 that the system imposes you can 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.

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. **Note:** Due to the unusual circumstances, we will be more able to extend deadlines, 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*, 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.

2.2 Submission format

You may submit a handwritten solution as described in Item 5 of Section 1.

If you submit a typeset solution, 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.

3 Assignments

3.1 Description

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 university example, 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) R and a set of FDs α.

Output: A set of relational schemas (informally, relations) 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 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 $\rightarrow$ ABB
   2. AB $\rightarrow$ C
   3. B $\rightarrow$ C
   4. E $\rightarrow$ F
   5. F $\rightarrow$ 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 $\rightarrow$ ABB	**1. A $\rightarrow$ B**
2. AB $\rightarrow$ C	2. AB $\rightarrow$ C
3. B $\rightarrow$ C	3. B $\rightarrow$ C
4. E $\rightarrow$ F	4. E $\rightarrow$ F
5. F $\rightarrow$ E	5. F $\rightarrow$ 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 $\rightarrow$ B	1. A $\rightarrow$ B
2. AB $\rightarrow$ C	**2. A $\rightarrow$ C**
3. B $\rightarrow$ C	3. B $\rightarrow$ C
4. E $\rightarrow$ F	4. E $\rightarrow$ F
5. F $\rightarrow$ E	5. F $\rightarrow$ 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 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 → BC	**1. A → C**
2. B → C	2. B → C
3. E → F	3. E → F
4. F → E	4. F → 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 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 → BC	**1. A → B**
2. B → C	2. B → C
3. E → F	3. E → F
4. F → E	4. F → 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 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 → B
2. B → C
3. E → F
4. F → 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. $\mathrm{AB}^+ = \mathrm{ABC}$

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 = ABCDE satisfying the set of FDs α:
   1. $AC \to E$
   2. $AC \to CB$
   3. $E \to DE$
   4. $D \to B$

   (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 for your larger case

   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 = ABCDEF satisfying the set of FDs α:
   1. $AB \to C$
   2. $A \to E$

3. DE → F
4. E → EF
5. F → B

(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 for your larger case

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 = ABCDEFGH satisfying the set of FDs:

1. AB → CD
2. C → EF
3. E → C
4. F → G

(a) 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.

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

3.4 What to submit

Please produce your solution and upload it as file `logicalDesign08.pdf`.