Homework Due 2020-09-18 by 22:55 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.

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 Priyanka Gumpena, `mailto:pg1828@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 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. In addition, there is a one-hour automatic extension, which you can use without any penalty, though the system may mark it as a late submission—don't worry about this. But do not count on it as it is only there in case you have communication problems and did not succeed in

9. 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.

10. **Be sure to follow the academic integrity rules listed in the course description (the syllabus) posted on NYU Classes**; you will find the link to it in the Syllabus section. The department and the GSAS treat academic integrity very seriously and we are required to report all possible violations.

11. **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

There are a number of small, mostly trivial, assignments (though the description is long).

2.2 Assignments

1. You need to have a CIMS computer account. If you do not have one yet please follow `https://cims.nyu.edu/webapps/content/systems/userservices/accounts/obtain` to request an account. Your request should generally be automatically granted.

2. Read the syllabus for which the link is provided in the `Syllabus` section of the class site. Decide whether you fully understand it. If you do not, state what needs to be clarified for full understanding.

3. In this assignment you will make sure that you are able to produce an ER diagram an associated text file when required. In this assignment, reproduce

   - The diagram `ER01.pdf` and the file `ER01.txt`, which is a a part of this archive, but replacing the Metadata in `ER01.txt` with your information

   The diagram has the following shapes

   - rectangles
   - diamonds
   - ellipses
   - triangles

   The diagram has the following types of lines

   - regular (relatively thin)
   - bold (relatively thick)
   - dashed
   - with and without arrows

   The diagram has the following types (formats) of text

   - regular
   - underlined
   - double-underlined

   Your drawing page should be of letter size in the landscape orientation ($11'' \times 8.5''$) and the diagram should look like the one in this assignment. Don't be fanatical about this. It is enough to make sure that the various shapes are recognizable as the ones given and placed roughly in the same locations, that the lines are recognizable as the ones given, that the shapes are connected as in the diagram, and the text in the shapes is formatted as given. If your software cannot conveniently produce underlined or double-underlined text, you can presumably place short line or lines under the text, as needed.

   You may not submit a hand-drawn diagram but have to use the software of your choice. Some options

   - draw.io `https://www.draw.io/`, which is free and does not require any installation.. Use the `Basic` template. The *native format* is `xml`
   - PowerPoint. The *native format* is `pptx` or `ppt`

     Microsoft Office 365 is free for NYU-affiliated students who use an NYU email account. If you would like to get it, go to `https://www.microsoft.com/en-us/education/products/office`.

- LibreOffice Draw. The *native format* is `odg`

but you can use whatever you like. However, you will have to use the same software for an assignment in which you will need to create an ER diagram.

4. (a) **Introduction**

   Our client, who does not know anything about ER (Entity-relationship) diagrams or relational database management systems asked us to build a system for maintaining its data. It provided the description below of the data to be managed, without using the terms we use in modeling. The bold font does not have a significant meaning. It's there to make following the description easier.

   Below is the description, *which on purpose and regrettably*, was written so it is difficult to understand what is going on. It is written in this way for us to practice understanding such descriptions so as to convert them into clear, actionable specifications.

   Try to understand and build "a mental picture" of the application. (You do not know what ER diagrams are after the first class.) If anything is unclear in the description, wait until the class in which we will work on this application.

   To reiterate: you are not asked to produce any drawing or implementation for this application—just think about it. Anything unclear: just wait for the class and do not spend too much time on it.

   (b) **The application**

   We maintain information about **Horses**. We know the **Name** of the **Horse** and it uniquely identifies the **Horse**. We also, sometimes, know the **Name** of its **Mother**. The client used to assign **Horses** for transportation but now they are just retired and are enjoying themselves.

   We maintain information about **Persons**. For each **Person** we may need to maintain the properties **ID#**, **SSN**, **Name**, **DOB**. **Children**. **Name** consists of two parts: **FN** and **LN**. A **Person** may have 0 or more **Children**. All we need to know about the **Children** of a **Person** is their first names. We do not necessarily know the value of **FN** for everybody, but we know the values of all the other attributes. No two **Persons** can have the same value of **SSN**. No two **Persons** can have the same value of **ID#**. Even if a **Child** has more than 1 **Person** as a parent, the **Child** is assigned to only the older parent. We also need to know the **Age** of a **Person**, which is computed in the obvious way from the **DOB**.

   We maintain information about **Automobiles**. An **Automobile** is a description of a type of an **Automobile** and not a specific physical object. An example would be an entry for a model of Honda. Such an entry would store **Model** with the value of "Honda CR-V", **Year** with the value of "2018", and **Weight** with the value of "3358". All these properties of an entry are always known and for each pair of values of **Model** and **Year** there is exactly one value of **Weight**.

   We maintain information about which **Persons** like which **Automobiles**.

We maintain information about **Cars**. In contrast with **Automobiles**, these are physical objects. A **Car** has two attributes **VIN** and **Color**. **VIN** is like a Social Security Number for **Cars**.

Each **Car** is associated with exactly 1 **Automobile**. Through this association we can find out for each **Car** what **Model Year** it is and what is its **Weight**.

A **Person** has at least 2 **Cars** but no 2 **Persons** can have the same **Car**. We keep information about that, but we may also keep information about the **Date** on which a **Car** was acquired by a **Person** who currently has it.

Some of the **Persons** are **Students** and some are **Professors**. A **Student** has an attribute **GPA**. A **Professor** has an attribute **Salary**, which is always known.

We maintain the following information about **Courses**: **C#**, **Title**, and **Description**. **C#** identifies a specific **Course** and is always known. **Title** is also always known.

We need to maintain required prerequisites. For example **C#** 101 might be a prerequisite for **C#** 102.

We maintain information about **Books**. For each **Book**, **Title** and **Author** are always known and together identify that **Book**.

We maintain information about required **Books**. A requirement of a **Book** specifies which **Professor** required it for which **Course**. There are no restrictions on requirements.

Courses are listings in a university catalog. When an offering of a **Course** is taught a **Section** is generated for the **Course**. For example our **Section** is 001 in Spring of 2020 for the **C#** CSCI-GA 2433 or DS-GA 2433. A **Section** has natural attributes **Year**, **Semester**, and **Sec#**, which are always known. It also has attribute **MaxSize**, which is sometimes known. No **Course** is listed in a catalog until at least one **Section** has been offered (which could be also a **Section** being offered now).

We keep information about which **Students** took which **Sections**. For each enrollment the grade may be known.

We keep information which **Professors** taught which **Sections**. Such a teaching assignment may be monitored by at most 1 **Professor**.

3 What to submit

Please upload 4 files, named *exactly* as specified and in the format *exactly* as specified.

(a) `text.txt` in the text format

In this file you will place your responses to Item 1, Item 2, and Item 4 of Section 2.2 by placing appropriate text after the item label as listed below (so your first item will labeled "i."):

i. State that you have a CIMS account *or* state that you have requested such an account. If neither holds, explain why.

ii. State that you have read and understood the syllabus *or* state that you have read the syllabus but that you have some questions and list them in the file. If neither holds, explain why.

iii. State that you have read the application and have considered what it could mean. If you have not done so, explain why.

(b) A file with a diagram in the native format as produced by your software. The file should be called `diagram.ext`, where `ext` is the *native* extension of the file produced by the software as described in Item 3 of Section 2.2. For example, for draw.io it would be `diagram.xml`.

(c) A file with a PDF version of the diagram. The file should be called `diagramPDF.pdf`.

(d) A file with your version of ER01.txt, and it should also be called `ER01.txt`.