

An SQL Query walks into a bar. In one corner of the bar are two tables. The Query walks up to the tables and asks them: Do you mind if I join you?

## Basi di Dati – 2019 Assignment 1 (Phase 2)

### Deadline

This phase has a strict deadline **Sunday 10 November 2019 at 23:59**. Any assignment delivered after that time will not be considered.

### Task Description & Marking

Roberto Speranza, the minister of Health, received 3 different ER diagrams from three different companies on how to design the database that the ministry needs. Since this is a public sector task, he is not allowed to say publicly the names of these companies, so let's call them RED, GREEN and BLUE. He would like to ask your help in evaluating these ER diagrams. He will send them to you by email. Once you see them, you need to fill online an evaluation report, where you give points for every correct modeling, and no points for wrong modelings. The parts that you need to check, are those described in the next page. You will be getting one point (mark) if something is wrong and you recognize that is wrong, or if it is correct and you say that it is correct. You get 0 points if something is modeled correctly and you say that is wrong, or if it is modeled wrongly and you say it is correct.

### Delivery

3 emails will soon be sent to you with the RED GREEN and BLUE ER diagrams. If you do not receive that email, it is because you did not deliver the phase 1 of the assignment or because the mail went to spam. Check your spam folder. For the delivery the only thing that you need to do is to fill up these forms:

For RED: <https://forms.gle/o2HLSdT6jDH1AE6V8>

For GREEN: <https://forms.gle/jD4B7bbC7q4rX4R29>

For BLUE: <https://forms.gle/X7PSqEDyALHFXDzv8>

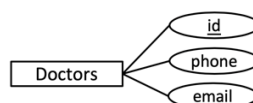
You are able to edit/modify your responses as many times as you want before the deadline. And keep In mind that the form is accessible only with your @studenti.unitn.it account .. NO GMAIL.COM accounts.

### Guidelines

Below is a number of guidelines for the evaluation. Note that the examples provided are only one way to implement the described constraint. Other ways of modeling the specification may also be possible. So, make sure that you check well the ER you see to understand if it correctly models the requirement. Is there something on the ER that reveals the name of the author? (email or name?) If yes, mark this point as incorrect. Is also the standard notation used? (Meaning the one that was mentioned in the classroom?). If any different not-standard symbols are used, mark this point as incorrect

1. *The database should contain information about doctors. Each doctor is characterized by unique id, and we need to keep also her/his phone number and email, in order to be able to contact them when we need them.*

Check whether there is an entity for doctors with these three attributes (id, phone and email). It is ok if the names are different, e.g., telephone instead of phone. Check also whether the id is underlined (indicating that it is a key). Here is an example of modeling it.



2. *Each doctor works for a hospital.*

There should be a many-to-one and total-participation relationship between an entity Doctors and Hospitals. For example:



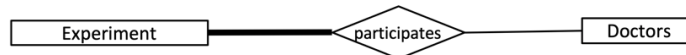
3. *The hospital for which a doctor works is the one that pays him/her for his/her services. Salaries are not fixed but is the result of any negotiation that the doctor has done with the hospital.*

Since the salary is for the doctor and by the hospital, it means that there should be an attribute for the salary on the relationship worksFor that was mentioned in point 2 above. (Attention: If the salary is an entity connected to the worksFor, it allows a doctor to work for a hospital and have different salaries which is not correct, unless a special care has been taken in the ER diagram. )



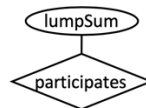
4. *Doctors participate in lab experiments. A lab experiment is done with a number of doctors that get together to experiment with something. No lab experiment can be done without doctors.*

There must be a total participation relationship (thick line, no arrow) from an entity set "LabExperiment" to "Doctors".



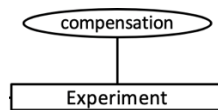
5. *The doctors get a lump-sum funding when they participate in a lab experiment. (This is done in order to top-up the salary they already get with some more money. This means that this funding is extra and is different from their basic salary.) Different lab experiments have different lump-sum that they offer to their participants.*

Since the lump-sum is for the participation in the lab experiment (and not for the lab experiment or for the doctor), it should be an attribute on the "Participates" relationship.



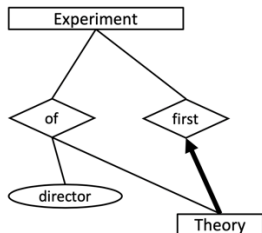
6. *A Doctor who participates in lab experiment receives also some compensation that is flat (meaning it is the same for everyone participating in the experiment and is not related to compensations other lab experiments give to their participants).*

Since this is fixed per lab experiment, and independent of who participates to the lab experiment, then it should be an attribute of the lab experiment entity.



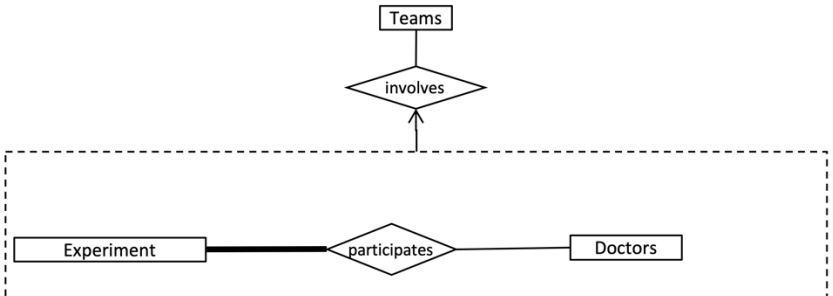
7. *The goal of lab experiments is to prove theories. In particular, for a theory to be proven a set of lab experiments need to be performed. This means that every theory has a set of lab experiments (no theory exists without at least one lab experiment). It is possible for a lab experiment to be serving more than one theory. For every lab experiment that is done for a specific theory, there is an appointed director. The directors are doctors and are decided during the previous lab experiment. Only the first lab experiment among those of a theory does not have a director because people do not know each other yet. But for the other subsequent lab experiments there is always a doctor serving as director.*

There should be an entity called theory and a lab experiment associate to it through a relationship. Since a theory cannot exist with at least one lab experiment, we need a total participation (thick line) on the side of the Theory entity and a normal line on the side of the lab experiment. Furthermore, since a lab experiment in a theory has one director, we place the director as an attribute on that relationship. The challenging part is that the first lab experiment of a theory does not have a director. For this reason, we need to separate the first lab experiment from the rest. To do that, we create two relationships. One as before and one for the first lab experiment. Since every theory has always a first lab experiment, the line on the theory side should be a thick arrow for the first lab experiment while for the other lab experiments it should be plain line (no thick no arrow).



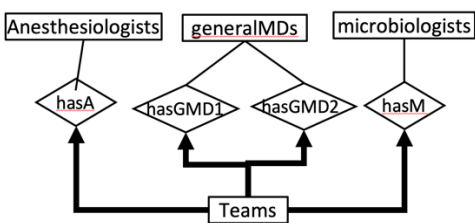
8. When a doctor participates in a lab experiment, he/she has the option to bring in a training team.

Since the team is optional and is related to the involvement of a team to a participation, it should be related to the relationship “Participates” with a many to one relationship, but since this is not possible to have relationship on relationship we need to do an aggregation. Note that we cannot connect the team to the “Participates” directly because if we do that it will be mandatory that for every participation there must be a team.

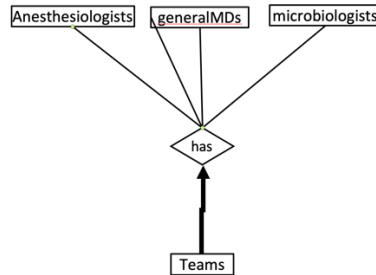


9. The trainee team consists of exactly 4 trainees. In particular, two are general MD, one anesthesiologist, and one microbiologist.

Since we are asking for exactly 4, it is hard to impose it with a simple relationship. So we create an entity called Team in which we explicitly state the members it must have, one by one, with a total participation and many to one relationship each one. And we create also an entity for every type of member.



An alternative that is also correct is the following:



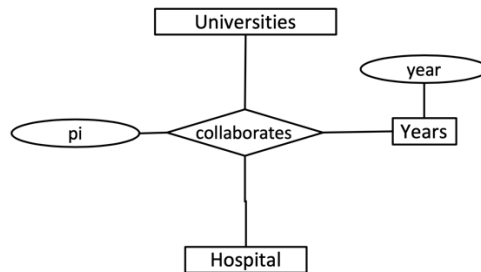
10. *All the trainees are students in some university. It is also possible to be students in more than one university.*

We need to associate the trainees to a group. To do so, we need to put them all under the same category. For this reason, we create an entity Students that has the three types we had before as IsA. Note that before (in point 8) we did not actually need to have the isA. Now we need it. So, we associate the Student with a new entity Universities. And since the trainees are all students in some university, the relationship has to be total participation.



11. *The universities collaborate with hospitals. Such collaboration has a duration of one year. This year needs to be recorded for every possible collaboration, even if the year has passed. For each such collaboration, there is a Principal Investigator that oversees the collaboration to ensure that everything goes well.*

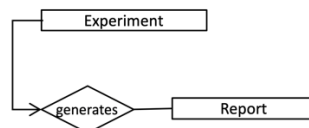
We create a relationship between the Universities and the Hospitals. Since we need to keep also the year of each contract we need to make the year an entity and connect it also to the relationship. If we had the year as an attribute, then the database will not be able to keep track of the different years of the contracts but only the latest year. Furthermore, one may assume that a University has at least one contract with the government. This is ok and making the line between the consultancy group and the relationship being thick is not an error. A director (supervisor) can simply be an attribute. No reason to make it an entity



The above modeling does not allow a consultancy group and a government to establish more than one “collaborates” in the same year. This is something that was not explicitly stated in the requirements, so a solution that has the contract as an entity that has many to one and total participation (meaning thick arrow) relationship to Universities, and Hospitals and the Year and supervisor as an attribute, is also accepted as a solution.

12. *Sometimes, at the end of a lab experiment a report is generated.*

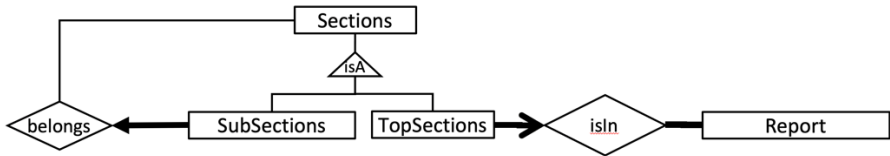
We create a Report entity and associate it through a relationship with the lab experiment. Since one Report is created for a lab experiment, we need an arrow from the side of the lab experiment. And since it says “sometimes”, then it should not be thick. So only many to one.



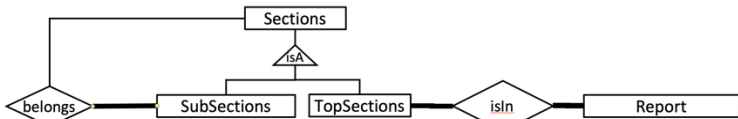
13. *The report consists of sections. These sections have a number code used to refer to it. The sections can be nested, meaning that they can have inside them other smaller sections (subsections) that in turn also may have inside them other subsections.... etc.*

We can see that there are two kinds of sections. Those that can be at the top level, and those that can be nested. So we create two entity sets. One that represents the top level sections and one that represents those that are not top-level. Any of the two kinds (top level or not

top level) can have other sections as subsections inside it. But only the non-top level sections have a parent section (where they are nested). That parent section can be either top-level section, or another non-top level. So we create a new entity called section, that has the top level and the non-top-level as isA subclasses. Then, only for the non-top level sections, we create a relationship that indicates the parent section. Then we need to also associate the report with the sections. That is a many to many relationship. The important part here is that the relationship is with the top-level section and not with the generic superclass entity Sections. Last but not least, there should be an attribute called “no” that records the section number id. So, all in one, the ER may look like this:



The above modeling does not allow subsections to be part of more than one sections (for instance a section repeated in different documents as is. The text does not specify that situation. So if we want to allow this to happen, we can simply make the model to be instead of a thick arrow simply a thick line. Like this:



The important part is that it is not possible to have a section (or subsection) not belonging to another bigger section or report (This is because the text does not say if anyone is ever creating any sections without a report. It only says that reportssals are created and these are having sections and subsections. So why was a section ever created? The text does not say anything about having sections that are not part of a report or of another subsection, which is why the links between “Subsections” and “belongs”, or between “TopSections” and “contains” have to be thick.

If we want to allow a Legislation proposal to consist of sections that are part of bigger sections, then we simply need to NOT create the entity set “TopSections” and instead associate the “contains” to the “Sections” entity.

The Sections entity should also have an attribute called id (underlined).

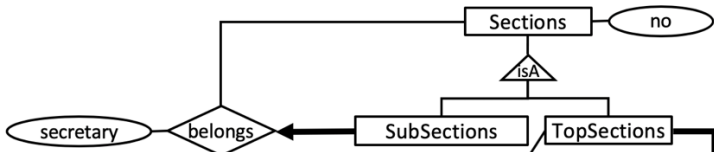
#### 14. The top-level sections of a report have a color associated to them ...

That is simply an attribute to the entity of the top level sections only.



#### 15. The ministry has also a set of secretaries that are checking for typos and expressions of the language. Every time a section is nested into another section, a secretary is appointed to be responsible for reading the two sections and making sure that the flow is good. We do not keep info of secretaries if they are not appointed as responsible to some section/subsection nesting

That means simply that we have a secretary attribute on the non-top sections, i.e..

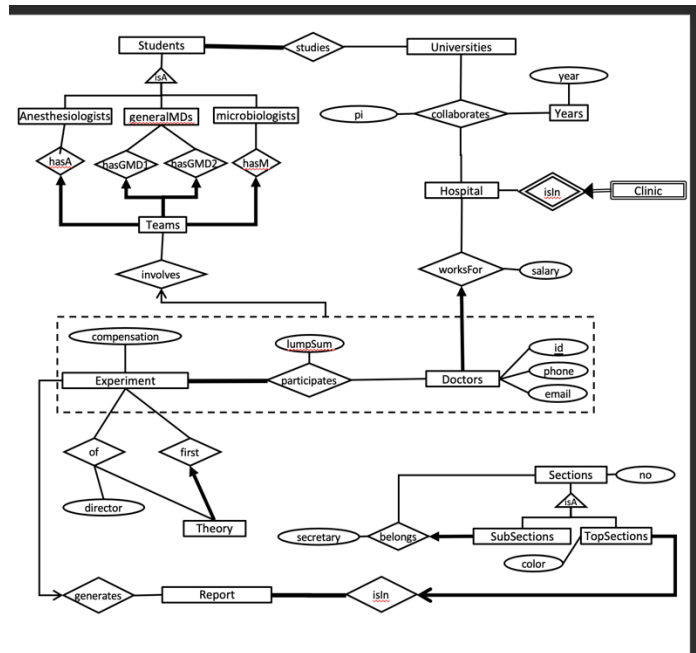


#### 16. Hospitals have also clinics. Each clinic is part of one specific hospital. We need to keep the information about what clinics each hospital has, but only for the hospitals we have in our database

That is basically saying to create a weak entity Clinic on hospitals.



Here is the overall ER:



Let's repeat here that in the actual ER all the entities will have attributes associated to them, and some of these attributes will be keys, exactly as we did on the entity Doctors. The reason that here we have no attributes is because the letter from the client does not specify them (so we have to ask for them) and also in this way the ER diagram has become easier to read.