

Assignment # 1, Basi di Dati 2015/2016

Instructor: Prof. Yannis Velegrakis

The assignment consists of 3 parts. Here you will find only Part A and Part B. The Part C will be communicated to you after the submission of Parts A and B. To submit Part A and B you need to provide a pdf document containing your answers. The pdf **should NOT** contain your name or matricola number. Any indication that can reveal your identity will automatically give you 0 marks. When somebody reads your answers, he/she should not be able to figure out who you are.

The answers should all be typed in a computer (no photos of handwritten solutions are allowed). Latex or MS Word or any other editor is fine. For designing the ER part, you can use whatever you like. Power point or Open Office is fine. One proposal is the yEd editor that is freely available for Windows, OSX and Linux operating system. The web site is: <http://www.yworks.com/en/downloads.html#yEd> Note that you **cannot** use any notation that is not among those that you were taught in class or that you can find in the text book. Using a difference notation gives you 0 marks automatically.

The deadline for the submission is **Thu Oct 22 at 12.00 noon**. The submission system will stop automatically accepting submissions on that day and time.

The assignment can be done in groups of 2 persons. Groups of 1 person may be allowed if there is a serious problem and only after a communication with the instructor. You need to send an email to the instructor explaining the reason or the problem that does not allow you to form a group of 2.

To submit the assignment, go to page:

<https://easychair.org/conferences/?conf=db15unitn>

If this is the first time that you use easychair you will need to create an account first. To do so, in the login page and click on the "create an account" link and follow the instructions. In the registration page that will follow, as a user name put your unitn email. As first and last name, put your real name. For the rest of the fields you can put whatever you like. After you have created your account you can log in.

When you log in, to submit your assignment click on the "New Submission" button at the top left. When you click "New Submission", you will see a screen where you will have to fill up the names of the two members of your group. The important fields are your name and your email. Remember that the email has to be your unitn email because the marks are automatically registered based on your email. In the rest (like in the address and country) you can put whatever you like. There is also a field called title. There you should write: "Assignment 1, PART A and B"

Finally, at the bottom you have an option to upload a pdf. This is where you upload your assignment. You can upload the assignment as many times as you want. Every time you upload a document it overwrites the previous one. After the deadline you will not be able to upload the document any more.

Note that the UNITN code of conduct strictly forbids any copying (and considers it cheating). If cheating is suspected, the assignments will both get 0 marks and the issue will be reported to the related authorities within the university.

Part A - Alitalia

Luca Cordero di Montezemolo (the CEO of Alitalia) has realized that the system Alitalia uses for recording flights and other related information is very old and needs to be redesigned. He has heard about the excellent quality of the students of the University of Trento that take the Database course. So this morning he called the rector asking for help. The rector told him to provide some more information about how they airports work and he gave the following:

- Airports have an airport code and a name. Furthermore, for each airport we store the length of its runway, the city where the airport resides, and the province this city belongs to.
- For each airplane type we need to store its name, a maximum number of seats and of the company that designs the airplane type.
- An airplane type can only land in certain airports.
- For each airplane we need to store its id and its total number of seats. Furthermore, for each airplane we need to store its type.
- For each flight we have a flight number and a weekday in which the flight occurs. Each flight has a number of legs (or flight segments) which are non-stop portions of the flight. Legs are distinguished by their leg number.
- Each leg has a departing airport and an arriving airport. Furthermore, for each leg there is a scheduled time of departure and a scheduled time of arrival (with respect to the corresponding airport).
- A leg instance is a particular occurrence of a leg on a particular date.
- For each leg instance we record the date of occurrence, the airplane used for the leg instance, and the number of seats that are available.
- For each leg instance we need to store the actual departure and arrival times.
- In each airport there are a number of Alitalia employees working. For each employee we record his SIN and name.
- There are two types of employees, namely, technicians and pilots. Technicians are capable of repairing specific airplane types. Pilots are assigned to specific airplanes.

What Mr. Cordero would like from you is to create:

1. **An ER diagram for the Alitalia database.** The ER should indicate the various attributes of each entity and relationship set. It should also indicate every key and referential integrity constraint (i.e., foreign keys) that exists. If some constraints cannot be captured, it should be explained why.
2. **A translation of the above ER diagram into a relational schema.** If some constraints of the ER diagram cannot be captured in the relational schema, it should be explained why. As a reminder, the relational schema can be described with statements of the form:

Person(codiceFiscale, nome: unique, eta, citta: not NULL)

citta is a Foreign Key to table Locations

All keys and foreign keys, unique and not null constraints should be indicated as in the above example.

3. **The statements needed to create the relations described in the relational schema alongside their constraints into a relational database.**

Part B - IMF

Yesterday at 3.00 in the night someone slipped an envelop under the door of your apartment. You opened it and inside you found the following letter:

Agenzia Informazioni e Sicurezza Esterna

Dear student,

Your country needs you. We have become very big as an organization and we need urgently to create a database to keep track of the spies we know. Your objectives, should you choose to accept the mission of course, would be to analyze our requirements and

1. Create an ER diagram for the database we need to build. The ER should indicate the various attributes of each entity and relationship set; each key and referential integrity constraint (i.e., foreign keys) should also be indicated. If some constraints cannot be captured, it should be explained why.
2. Create a translation of the above ER diagram into a relational schema. If some constraints of the ER diagram cannot be captured in the relational schema, it should be explained why. As a reminder, the relational schema can be described with statements of the form:

Person(codiceFiscale, nome: unique, eta, citta: not NULL)

citta is a Foreign Key to table Locations

All keys and foreign keys, unique and not null constraints should be indicated as in the above example.

3. Produce the statements needed to create the tables described in the relational schema alongside their constraints into a relational database.

The next page contains the list of the requirements as it has been collected by many brave agents that worked hard before you. Treat them with respect. These agents almost gave their life to collect these requirements.

For this mission you need to team up with another student. It is your responsibility to create your team by choosing the right partner for this mission. For this, you have at your disposal ~150 students that take the database course. Choose wisely.

Any document you can produce is highly classified. Cannot be discussed with anyone else apart from the other member of your team. You cannot let anyone know what you are doing. This includes relatives, husbands, wives, children, boyfriends, girlfriends, colleagues, or other professors. If it is discovered that you have talked to someone or that the information has been leaked in any form, we will shut down your operation, close the file and mark your mission as FAIL.

Furthermore, if during the process you get captured and hold hostage by members of other teams that will try by force to extract information of how you were planning to implement this task, if you starve due to lack of time for eating, or if you get killed by working too much for this mission, the agency will officially refuse any knowledge of your actions.

You have been trained extremely well all these weeks in ER and Relational. This isn't mission difficult, it's mission impossible. "Difficult" should be a walk in the park for you. Good luck.



Requirements for a Spy Database

We collect information from many different sources through many agents (spies). Spies work for secret agencies. If a spy works for one of our secret agency, it is a normal spy. However, there are spies from enemy secret agencies that are not ours but work with us as well. Those are the double spies). In the database we would like to keep information about:

Secret Agencies

The database must store information about each known secret agency, such as, its name, the country whose interests it serves, the location of its headquarters, and the country (or countries) in which it operates. It will also be useful to keep track of countries that share a common border since conflicts usually arise between such countries. For our own agencies, we also know that each of them has a number of (sub) divisions. For example, there might be one division responsible for foreign affairs, one for internal affairs, and the like.

Spies

Each secret agency employs a number of spies. For each spy we must be able to record her name, nationality (since an agency might employ foreign nationals) a list of nicknames that she uses and most importantly the list of things in which the agent is specialized. For example, a specific agent might be specialized in computer hacking, while another is an expert in explosives and/or firearms. Thus, your database must be able to also store the type of skills in which an agent can be specialized. It must also store a skill level, that is, the degree to which a spy possesses a skill. Since we know more about our own spies than the ones belonging to other agencies, we expect that the above information is more accurate for the former set of spies. Spies (from the same or different agencies) can develop bonds. A bond between two agents is developed when one of them makes a favor to the other.

Missions

We keep information on the missions initiated by only our own agencies. Each mission has a code name, some duration (expressed in number of days), a primary and probably a secondary objective. One division of the agency that initiated a mission is responsible for supervising it. In more detail, a mission is separated into a number of legs and each leg can take place in a different country. For each mission we should be able to record whether the mission is completed or not and if so what was the outcome of the mission. Our spies can work independently or in groups and they can participate in missions. For each completed mission that a spy participates in she gets a grade (performance).