The aim of the second set of exercises is obtain a better understanding in the techniques that can be used to obtain a good database design. To this aim, you will need to design two databases, from the conceptual design to the physical design. This exercise is done in **groups of 3**. The final deadline for the answers to all exercises is **March 27, 23:55**.
Creating a conceptual design for a database is similar to writing a text: an important aim of the conceptual design is to communicate which information needs to be stored in the database, and which constraints can be imposed on the data. The success of a diagram can hence be measured by the clarity of the design in the eyes of people that have to study the diagram. To evaluate the quality of one of your designs, we will rely on peer-reviewing. To support this peer-reviewing, this exercise has several intermediate deadlines.
Please read these instructions fully and carefully before starting; take note of the intermediate deadlines.
Consider the use of software such as Dia (`http://dia-installer.de`) or Diagrams.Net (`https://www.diagrams.net/`) for creating diagrams; this software is available for all common operating systems.

1 Web Application

The context of this exercise is the following.

> You are creating a new smartphone application that allows people to share different kinds of vehicles: cars, trucks, bicycles (including cargo bikes) and electric scooters. Each of these vehicles has a certain capacity, and belongs to a customer of the app; when the vehicle is not moving, its location at that moment in time is stored, using longitude and latitude coordinates. Customers of the app can make a request for the use of a specific vehicle, for a certain moment in time; if the owner accepts the request, the vehicle is allocated to the customer for the time requested.
>
> The owner can specify a pricing scheme based on the time and duration of the rent.
>
> For customers we store information such as their username, their password, their full name, their address and credit card.
>
> Some aspects of the app need to be clarified: should coordinates be included in a request, and in an accepted trip? Should other customers be allowed to express their interest in joining another customer, to support car pooling? Should customers have friends in the app, allowing them to quickly track whether the vehicle of a friend is available? When a time is indicated, should this involve both a starting and an ending time? How is pricing scheme defined exactly? ...
>
> You are tasked with the creation of a first proposal for all the data that could be collected in this application. This proposal should fill in all the details of the data that will be collected in this application.

The tasks in this exercise are the following.

1. Create a **conceptual** design for your database by creating a diagram for your database using ER Chen notation. Take care that the following requirements are met:

- Your diagram will reflect all the requirements listed above, and will provide additional relations as you believe to be necessary in your application.
- Your diagram will have to contain at least 5 entities.
- Your diagram should reflect your choices in as much detail as possible: hence, make sure that cardinality constraints, participation constraints, and primary keys are indicated.

Note that your diagram should be **conceptual**. A conceptual diagram **does not** exactly represent all the tables that will be stored in a relational database; it **does not** represent a phyical design of the database. The most important goal of a conceptual diagram is to make clear what is conceptually going to be stored in the database.

2. Once you have finished this diagram, you are required to create a second diagram in ER/Crow's foot notation or UML notation. Also here, you should make sure that the diagram indicates all constraints correctly, and is **conceptual**.

3. Submit both your diagrams on Moodle for review, in PDF form; the deadline for this phase is **March 16, 23:55**. You are only allowed to send the diagrams; hence, the diagrams should be understandable without additional information describing them. You are allowed to add limited textual annotations in the diagram, for instance, labels that indicate the meaning of an edge in the diagram. Each group will receive the diagrams of three other groups, as well as instructions for reviewing these diagrams. Each member of your group is expected to review the diagram of one other group. You have to complete these reviews by **March 23, 23:55**, and will also receive the reviews of your design at this same moment. 20% percent of the grade for this exercise will be determined by the grades that you receive from other students!

4. Once you have created the conceptual diagram, you are required to create a **physical design** of the database. Create the schema for your proposed database using statements in SQL. These statements should be executable in SQLite. Ensure the following:

   - All primary key and foreign key constraints are properly defined;
   - The types of all columns are sensible, including whether or not attributes are NULLable;
   - The database is in third normal form;
   - The design of your database reflects the conceptual design.

   The most important design criterion when creating the physical design is that the quality of the data is ensured; hence, you are not required to take into account considerations related to the performance of the database. You may start with this schema before the reviews are finished; however, the final schema should reflect the choices in your final diagram. Submit all your schemas, your updated conceptual diagrams, and a short description of your choices and the way you processed the other's students comments by **March 27, 23:55**. The latter document should have at most 2 pages.

This exercise determines 60% of your final grade.

2 European Soccer

On Moodle you will find a database for European Soccer, also available from `kaggle.com`: `https://www.kaggle.com/hugomathien/soccer`. We will study the quality of this database.

1. Use SQLite's functionality to determine the schemas of the relations (relation names and their attributes); furthermore, determine the primary keys and foreign keys in this database. Note that SQLite uses 'dot commands' for this purpose; see `http://www.w3resource.com/sqlite/sqlite-dot-commands.php`. Create an ER diagram, using Crow's foot notation, that closely reflects the physical design of this database. You may abbreviate numbered attributes using ... notation.

2. This database violates the first normal form. Propose an improved schema (giving relation names, their attribute names, primary keys, and foreign keys). You may use SQL to denote this schema, but this is not required.

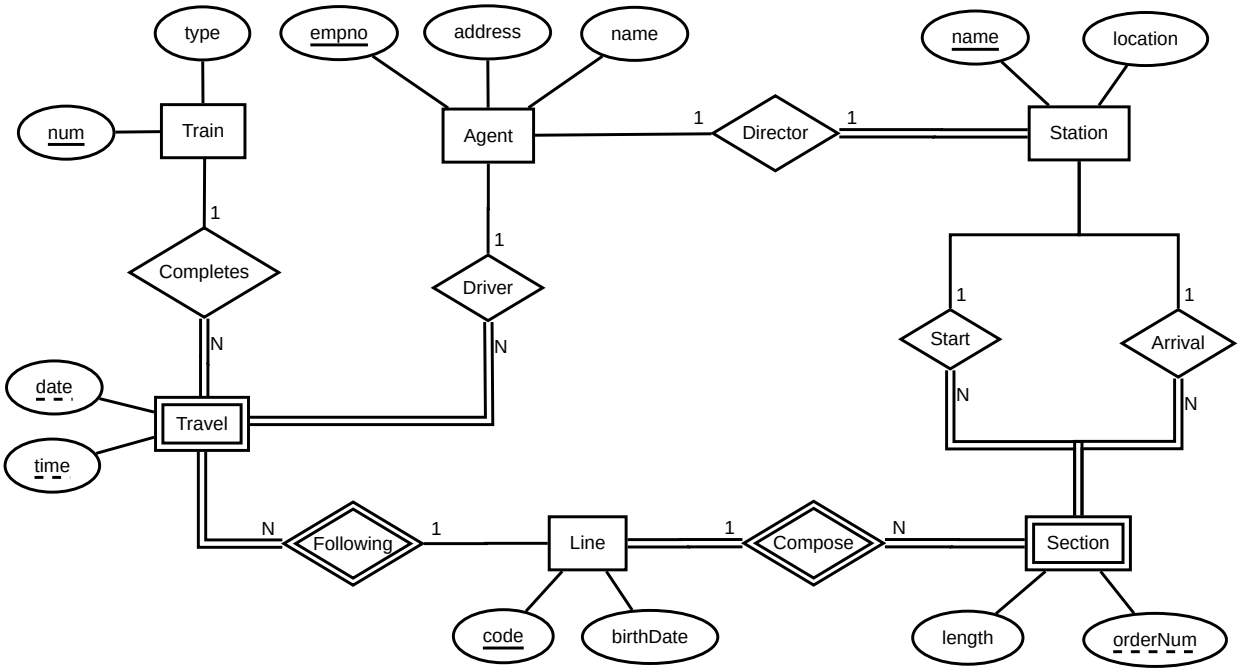
This exercise determines 20% of your final grade.

3 Trains

The following excerpt is extracted from

> J.L. Hainaut, Bases de Données, Concepts utilisation et développement, seconde édition, Dunod, 2012, Chapter 11, page 326.
>
> "We consider train travels. A travel is done by a train (for which a unique train number is known) at a given date and given time within that day, following a known train line (identified by a code, and characterized by a "birth" date, i.e. when the line has been put in service). The train driver is a so-called Agent. The latter has a unique employee number, a name, and an address. A train line is composed of consecutive sections, each of a given length. A line section starts at a station and arrives to another. A station is identified by a name and has a location. Each station is directed by an agent, yet not all agents are responsible for one."

The following entity-relationship diagram shows a conceptual model for the domain at hand (using the notation of Elmasri's textbook).



Convert this ER diagram into a relational schema (relation names, attributes, primary keys, foreign keys), using the approach discussed in Elmasri's textbook and the lectures. Provide the necessary statements for creating this database in SQLite. Take into account these considerations:

- The database should satisfy **all** constraints specified in the conceptual schema: the database should only allow to store data such as represented in the schema;
- All relations created should be in 3rd Normal Form;
- While satisfying the above constraints, the database should have a minimal number of relations; consider representing relationships by adding attributes to relations, where possible;
- The database should have a minimal number of attributes that are NULLable;
- Some constraints can only be enforced by adding SQL constraints to the database; you are expected to add at least one such constraint, but don't add more constraints than necessary;
- Also specify in your schema what happens when a tuple is deleted from a relation.

This exercise determines 20% of your final grade.