Exercise 1: Introduction to databases and ER-modelling

Task 1

a) Data is known facts that can be recorded and that has an implicit meaning. Database is a collection of related data, for example several data records with the same structure. Database system is the whole package, from how the database is defined (meta-data), the actual database storage, the DBMS (database management system) software to how users/programs can use this database system (by queries for example).
    i) The relation between them is that the data is stored in the database which is contained within the database system (in other words the database is the thing in the database system that all the data is stored in).
b) A database is used for storing related, structured data with well defined formats (there is a lot of meta data describing the actual data.). The database also has defined methods for inserting, updating and retrieving data. The file system is used for unstructured data that doesn't have to be related and it is more general. Databases is built upon file systems, as it uses the file system to store the data.
c) What is it? The database? One advantage is the the database is centralized, meaning that each user of the data doesn't need to have the data locally. This saves storage space and makes data updates easy, meaning that data only has to be updated at one location. Database management systems also gives authorization control, meaning that it is possible to restrict user access to different parts of the database. Persistent storage for objects is also an advantage, meaning that objects can be stored in the database and persist even though the program it originated from has been terminated. Databases also has functionality to efficiently execute queries and updates, which includes indexing and caching. It other words, it has to be fast, which is what defines a good database and a reason to store your data in a database. The list goes on and on and contains some these things:
    - Backup and recovery
    - Multiple user interfaces (both text/terminal base and some GUIs)
    - Representing complex relationships among data (cars belongs to persons)