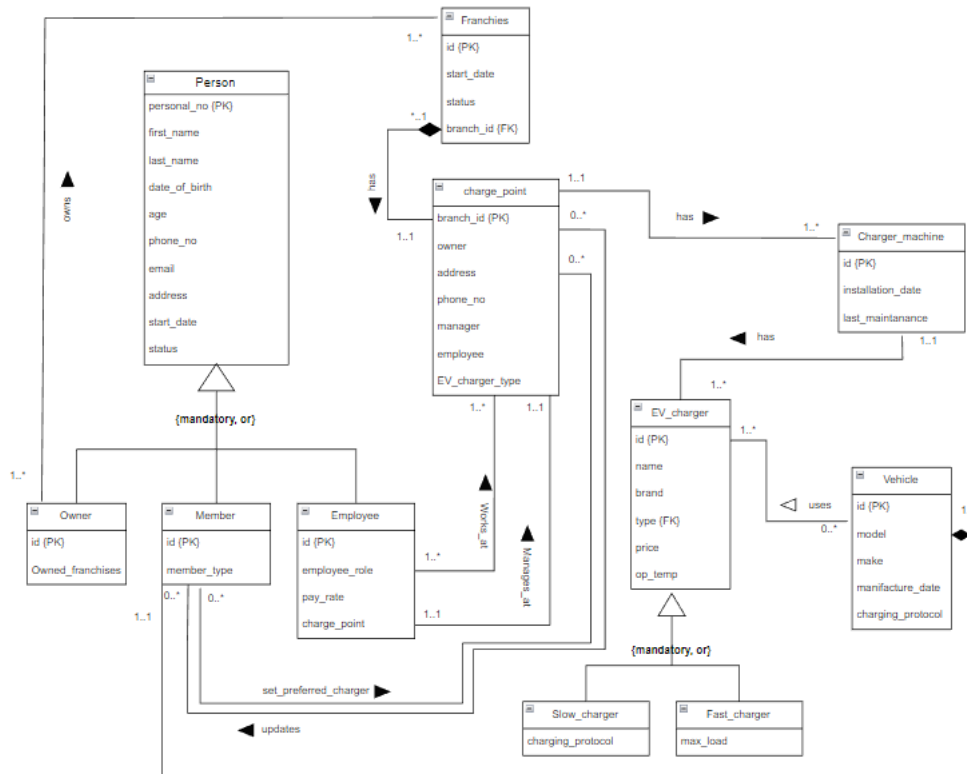


# Assignment 2

## UML diagram



## How I interpreted the assignment

The relations I could identify are pictured in the UML diagram above. There are in general two things I interpreted from the assignment text that could be done in more efficient ways than what was indicated (I guess this is a part of the task).

Firstly, I found that I could use specialization for the EV\_charger relation, where I could separate the slow\_charger and fast\_charger out from the general charger. These entities become children of the EV\_charger relation.

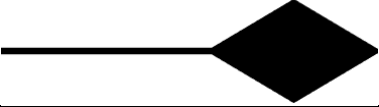



The Person relation is a generalization of what is common in Owner, Member and Employee. This was to get smaller and more efficient tables for the database to look up in. If I did not do the generalization it would have been very long and big tables to look up in each time. We don't want that due to bad performance.

Since the assignment does not ask for any foreign keys, I have not added them in the UML diagram.

See the table on the next page in this document to see what the arrows and icons mean.

## Table icons and arrows

Here is the meaning of each type of icon and arrow I used and a short description of how I have understood their function.

	<b>Composition</b> The lifecycle of the part is dependent of the lifecycle of the whole.
	<b>Relationship type: strong – strong</b> Entity <i>must</i> have the type of relation connected. E. g. a car must have runs_on -> *type of fuel* because all cars runs on some kind of fuel, regardless of what type it is, BUT it <i>might</i> have a owner_history.
	<b>Relationship type: strong – weak</b> Entity exists only in relation to another relation. Meaning here that the vehicle cant function as a vehicle without its charger.
	<b>Specialization</b> Takes a big table and makes it smaller. Must have constraints such as done in the UML diagram above (“{mandatory, or}”). The constraints I used means one of them must be.
1..*	<b>Structural constraints</b> The numbers like this means how many there can be of one entity type in relation to another. E. g. in my diagram I have used this one between the owner and the franchise relation. This means that one owner can own many franchises.