

# Database Technology

## Topic 3: Enhanced Entity-Relationship (EER) Modeling

Olaf Hartig

[olaf.hartig@liu.se](mailto:olaf.hartig@liu.se)

# Example

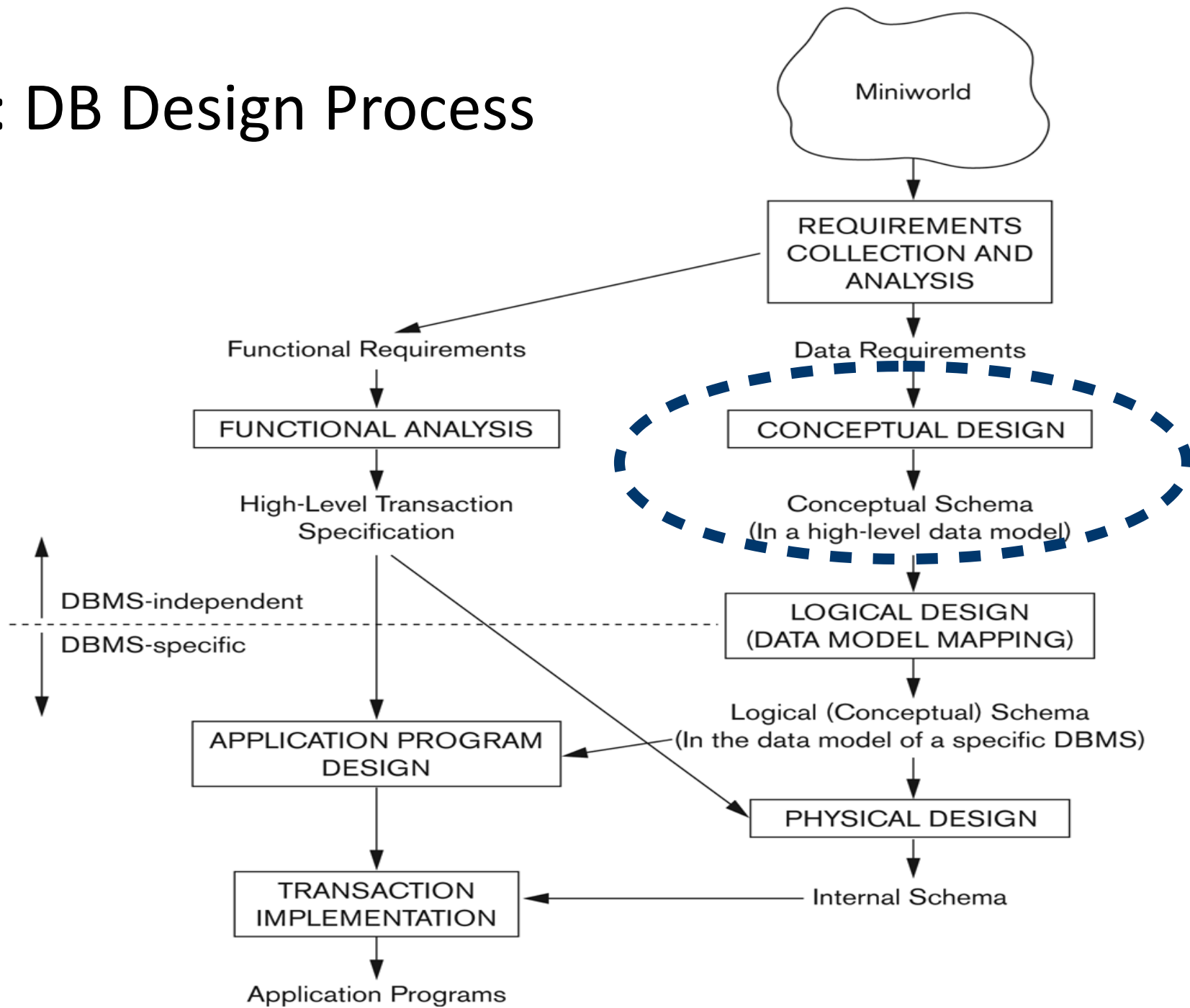
A taxi company needs to model their activities.

There are two types of **employees** in the company: **drivers** and **operators**. For drivers it is interesting to know the **date of issue** and **type** of the driving license, and the **date of issue** of the taxi driver's certificate. For all employees it is interesting to know their **personal number**, **address** and the available **phone numbers**.

The company owns a number of **cars**. For each car there is a need to know its **type**, **year of manufacturing**, **number of places** in the car and **date of the last service**.

The company wants to have a record of car **trips**. A taxi may be picked on a street or ordered through an **operator** who assigns the order to a certain **driver** and a **car**. **Departure** and **destination addresses** together with **times** should also be recorded.

# Recall: DB Design Process

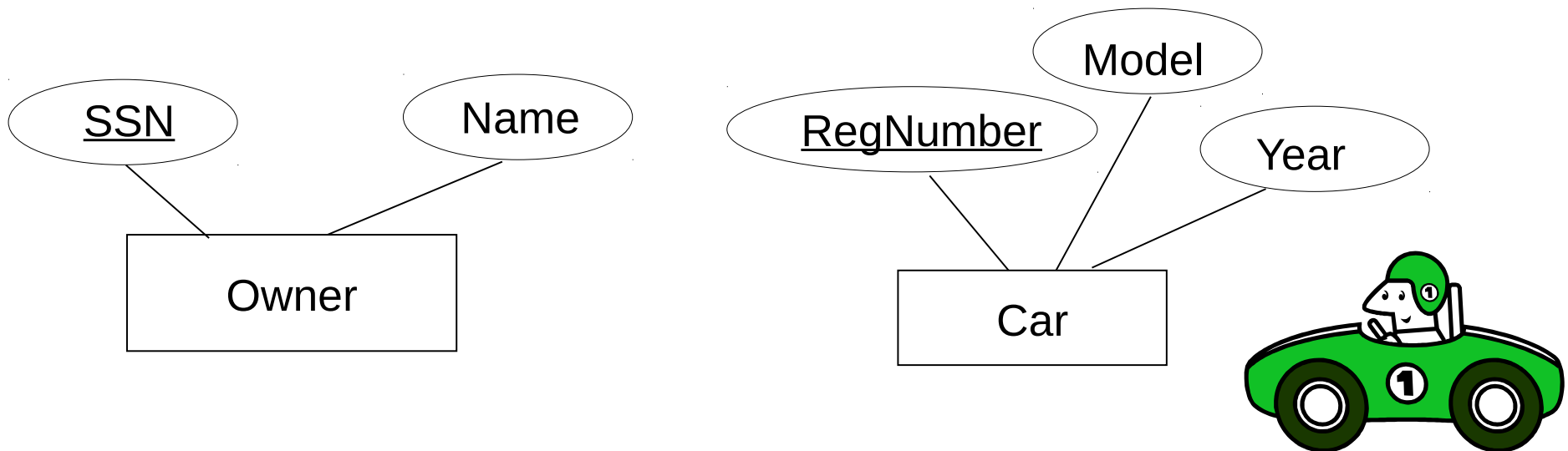


# Entity-Relationship (ER) Model

- High-level conceptual data model
  - An overview of the database
  - Easy to discuss with non-database experts
  - Easy to translate to data model of DBMS
- ER diagram
  - Diagrammatic notation associated with the ER model

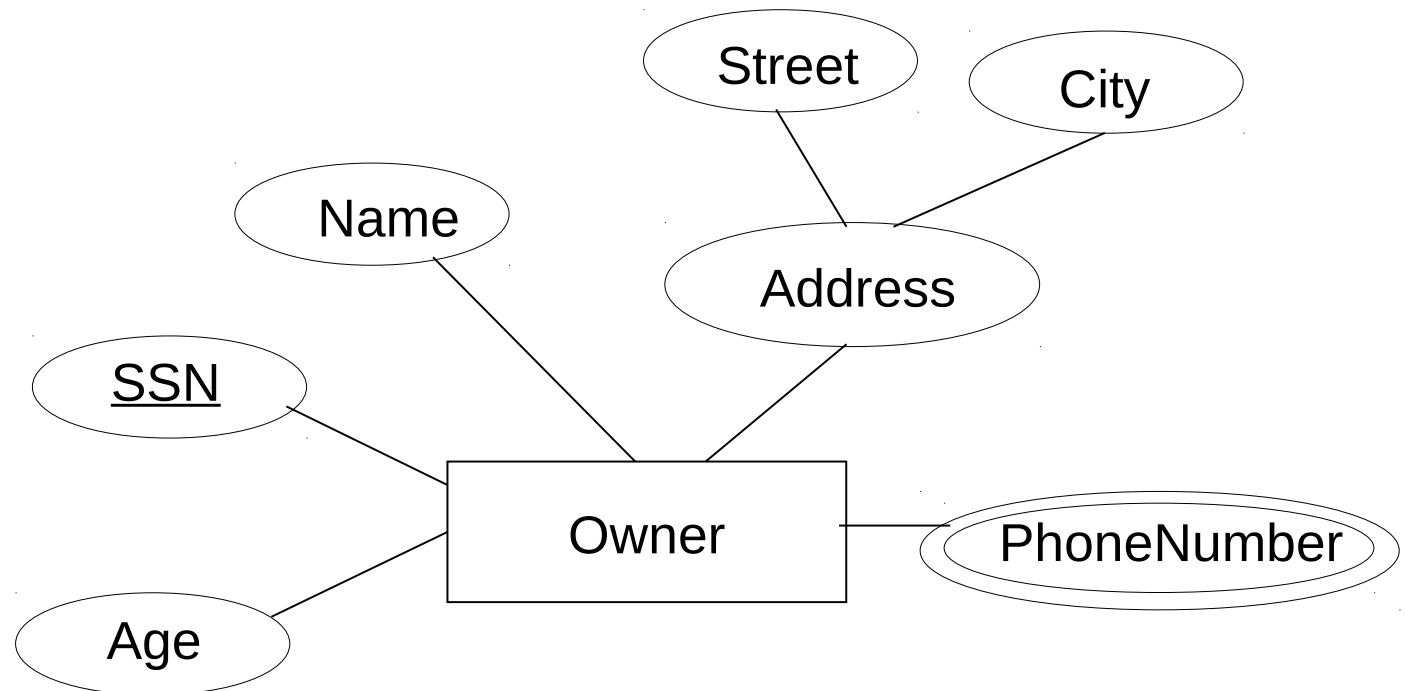
# Entities and Entity Types

- **Entity**: a "thing" in the real world with an independent existence
- **Attributes**: Properties that describe an entity
- **Entity type**: A collection of entities that have the same set of attributes



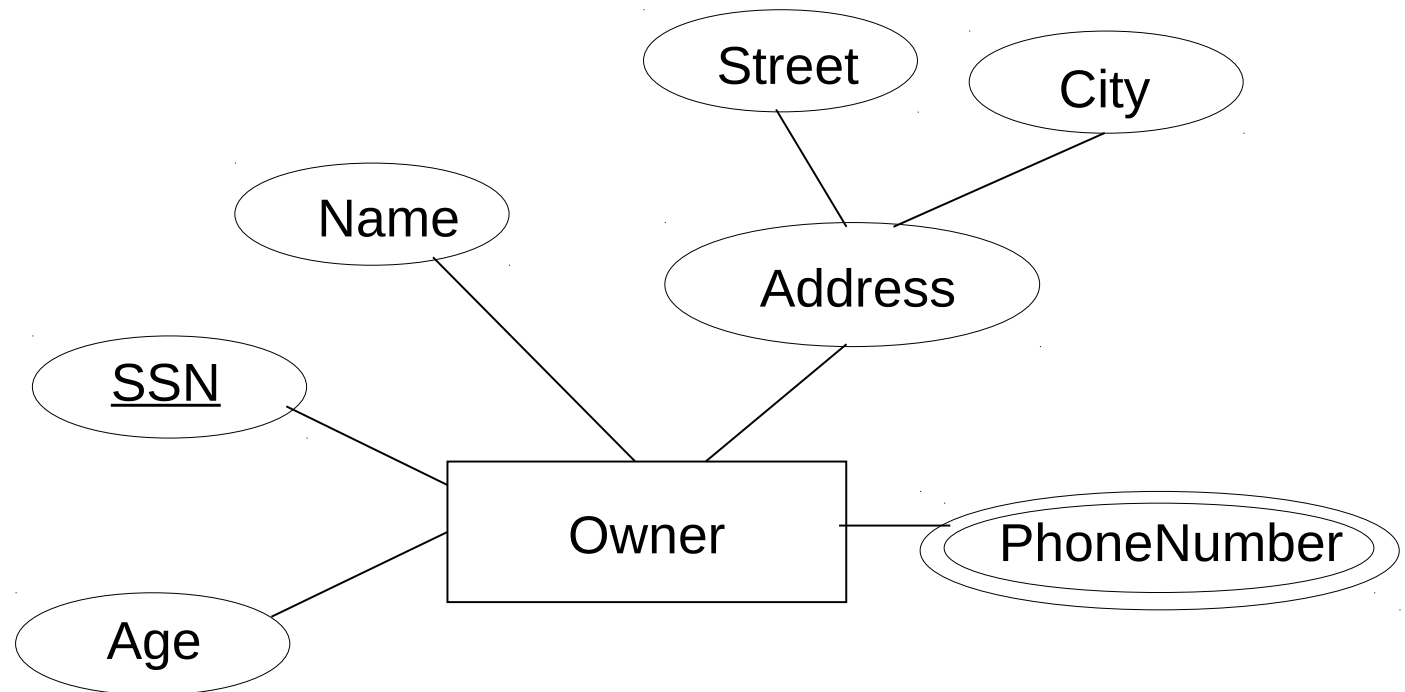
# Attributes

- **simple** versus **composite**
- **single-valued** versus **multivalued**
- **stored** versus **derived**



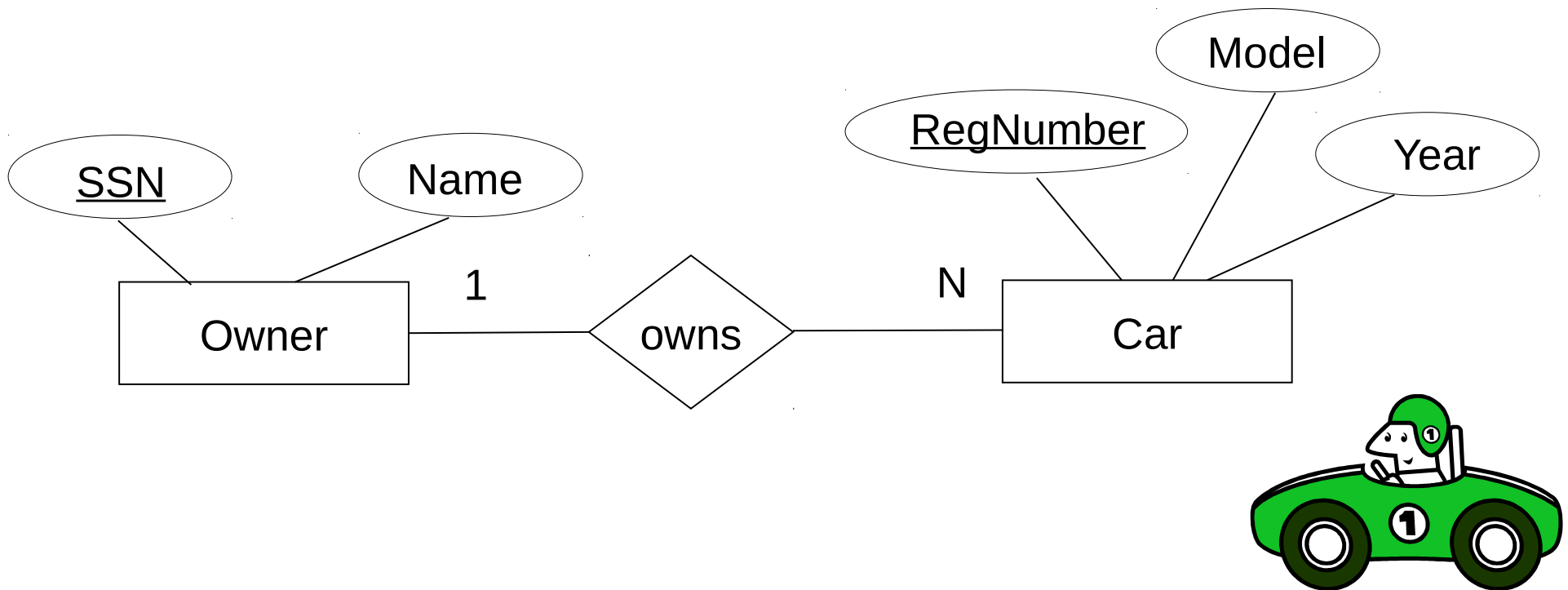
# Constraints on Attributes

- Value sets (domains) of attributes
- Key attributes



# Relationship Types

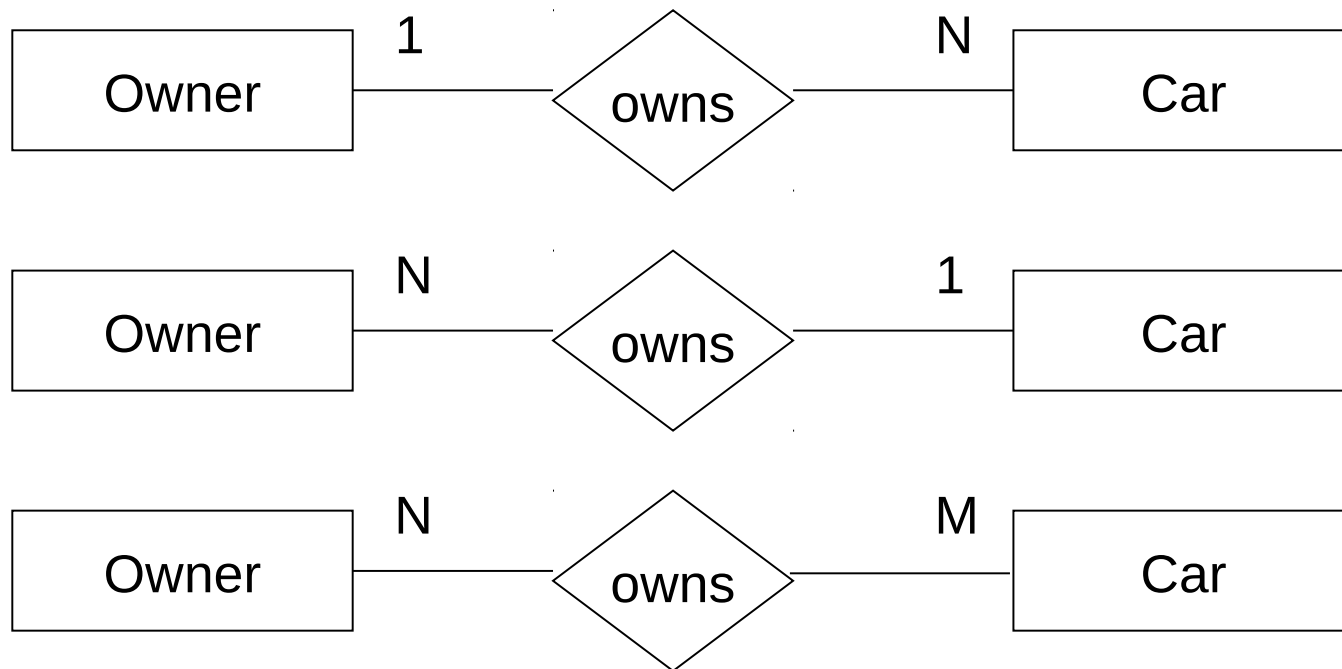
- Relationship type: Association among entity types





# Constraints on Relationship Types

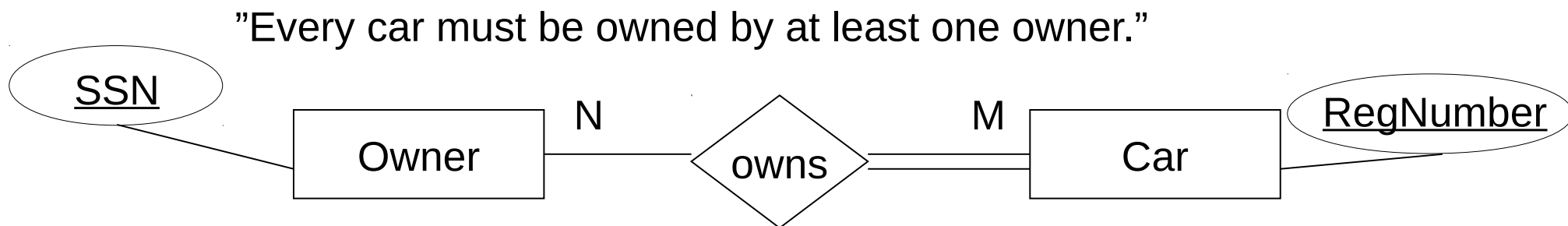
- Cardinality ratio: *Maximum* number of relationships an entity can participate in
- Possible cardinality ratio:  $1:1$ ,  $1:N$ ,  $N:1$ , and  $N:M$



# Constraints on Relationship Types

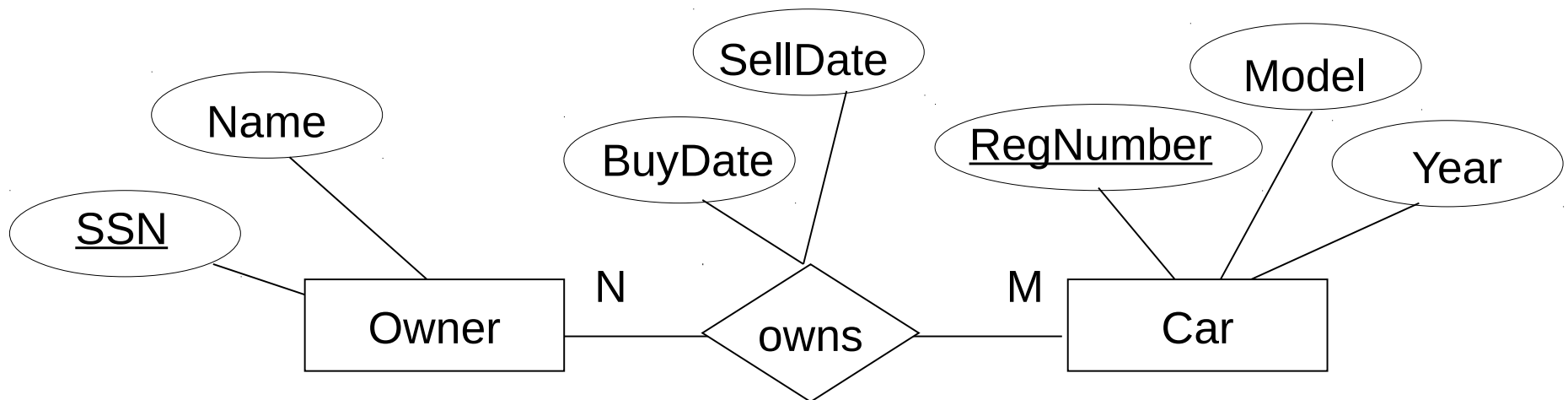
- Participation constraint

- **Total participation:** Every entity participates in *at least* one relationship with another entity



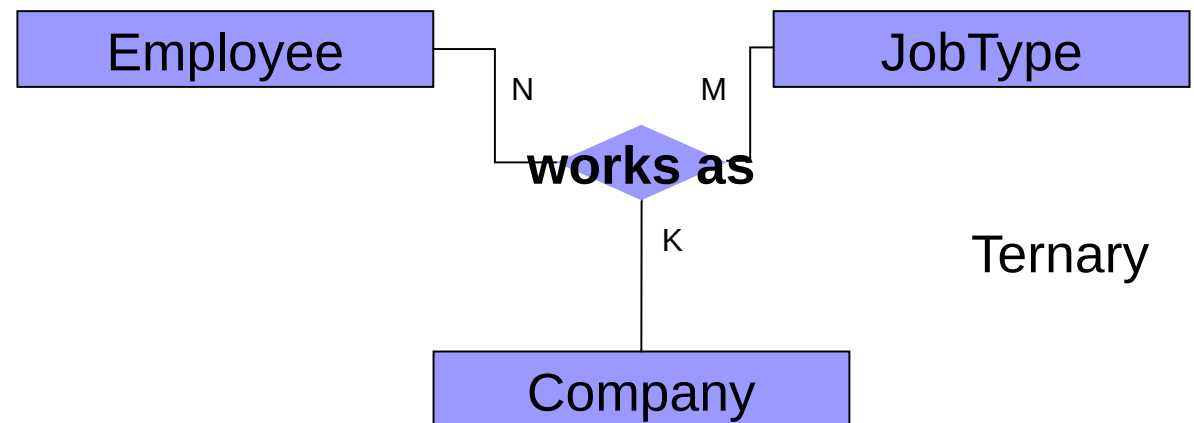
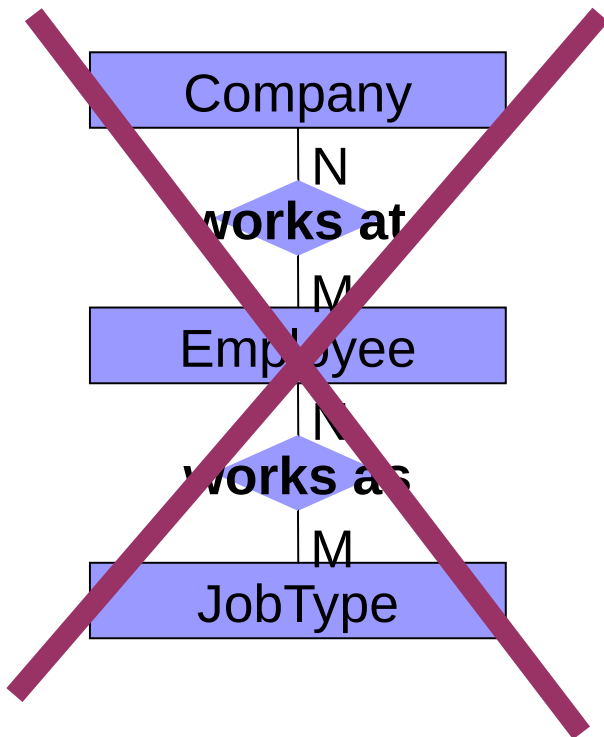
# Attributes of Relationship Types

”Store information on who owned which car and during which period of time”



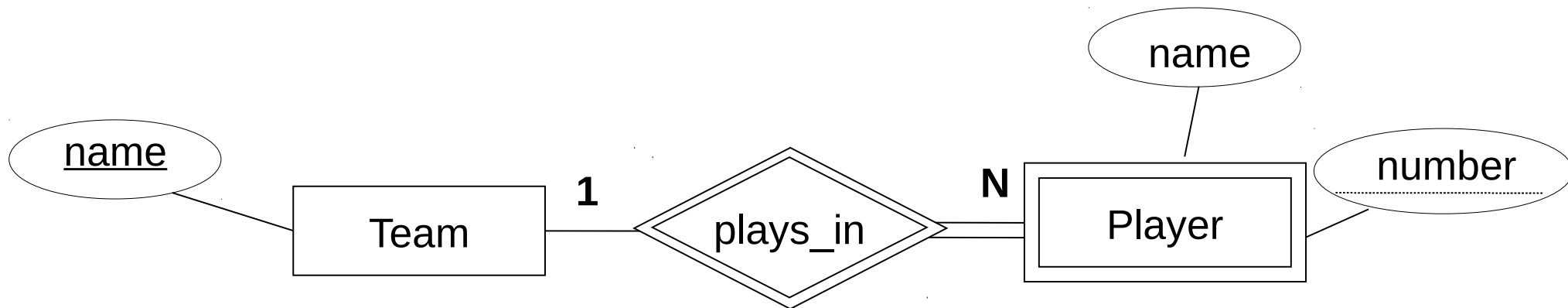
# *n*-ary Relationships

- Example: A person works as an engineer at one company and as a gym instructor at another company



# Weak Entity Types

- Identified by their relationship to a specific entity from another entity type
- Do not have key attributes of their own
  - Only partial key
  - The identifying entity has the rest of the key



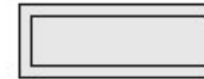
# ER Notation

## Symbol

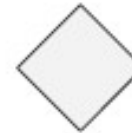
## Meaning



Entity



Weak Entity



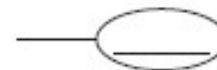
Relationship



Identifying Relationship



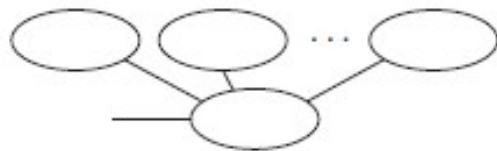
Attribute



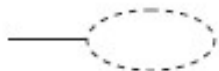
Key Attribute



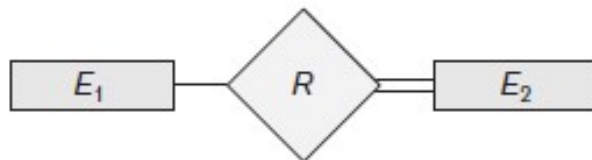
Multivalued Attribute



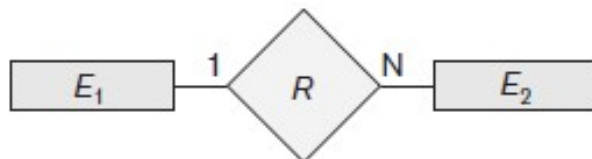
Composite Attribute



Derived Attribute



Total Participation of  $E_2$  in  $R$



Cardinality Ratio 1: N for  $E_1:E_2$  in  $R$

# Enhanced Entity-Relationship (EER) Model

# Enhanced ER (EER) Model

## ■ Why more?

- ☐ To support more complex data requirements
- ☐ Example: Only some employees can use a company car, only managers have to write a monthly report, but all employees have assigned personal number, salary account and a place in the office.

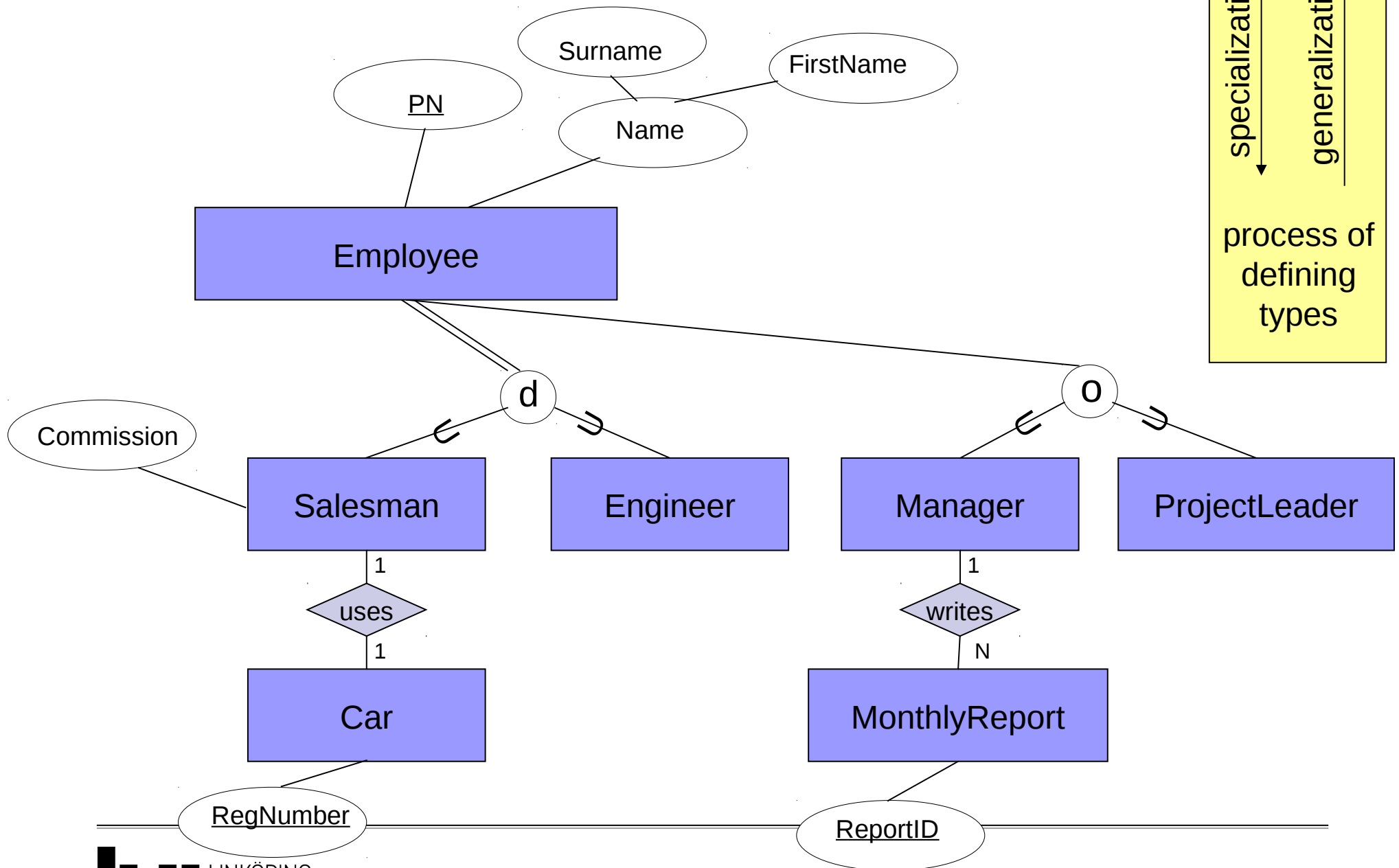
## ■ What more?

- ☐ Specialization / generalization
- ☐ Subtype / supertype
- ☐ Union subtypes
- ☐ Attribute and relationship inheritance



# Subtype / Supertype

specialization  
↓  
generalization  
↑  
process of defining types



# Constraints on Subtypes

- **Disjointness constraint**
  - Specifies that the subclasses of the specialization must be disjoint
  - Otherwise "overlapping"
- **Completeness constraint (or totalness constraint)**
  - Specifies that every superclass entity must be in a subclass
  - Otherwise "partial"
- Disjointness and completeness are *independent* constraints
  - i.e., four cases are possible
    - disjoint and total
    - disjoint and partial
    - overlapping and total
    - overlapping and partial

# Example

# Example

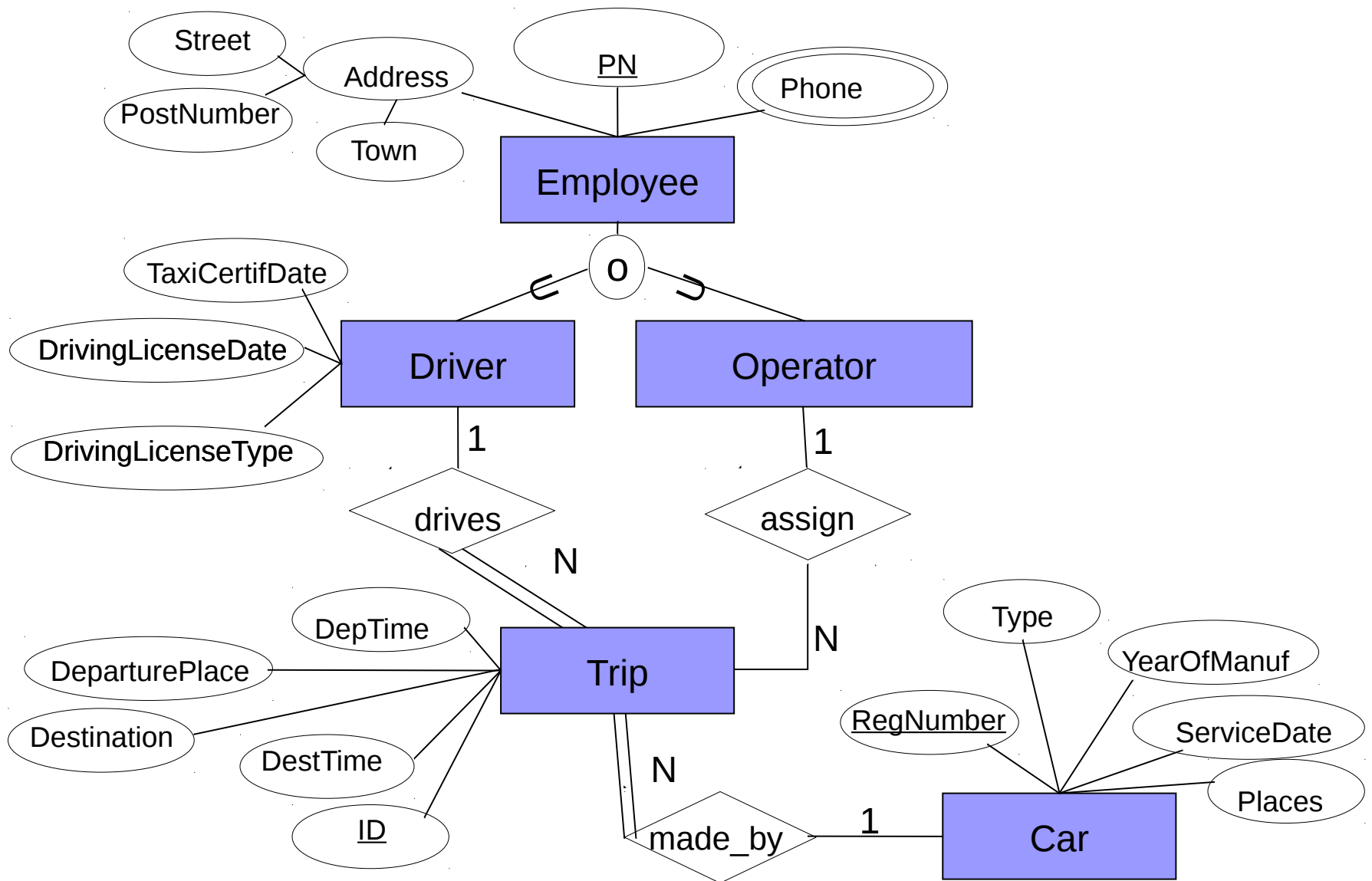
A taxi company needs to model their activities.

There are two types of **employees** in the company: **drivers** and **operators**. For drivers it is interesting to know the **date of issue** and **type** of the driving license, and the **date of issue** of the taxi driver's certificate. For all employees it is interesting to know their **personal number**, **address** and the available **phone numbers**.

The company owns a number of **cars**. For each car there is a need to know its **type**, **year of manufacturing**, **number of places** in the car and **date of the last service**.

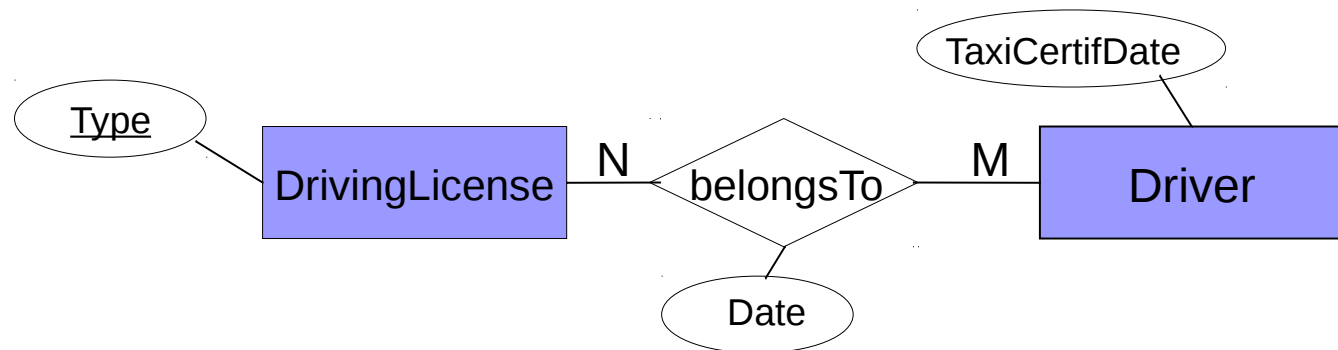
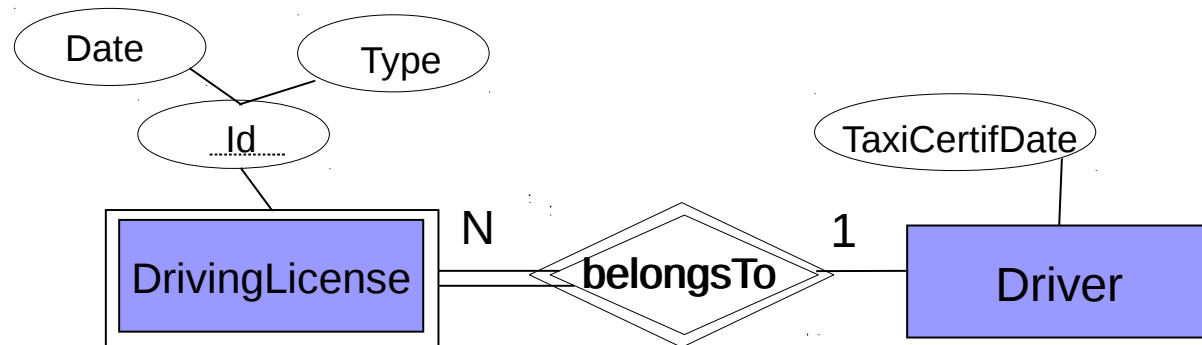
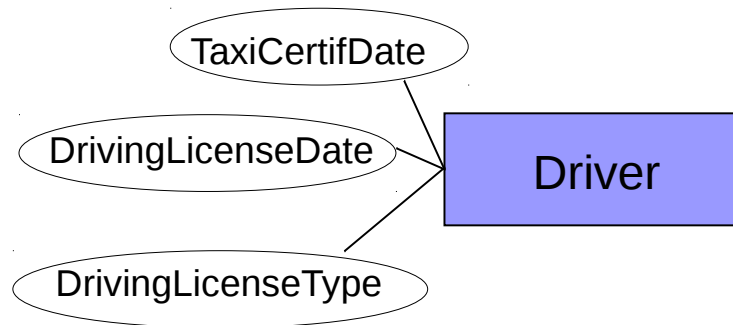
The company wants to have a record of car **trips**. A taxi may be picked on a street or ordered through an **operator** who assigns the order to a certain **driver** and a **car**. **Departure** and **destination addresses** together with **times** should also be recorded.

# Example



# Example (cont'd)

A driver may have many driving licenses (types)



# Summary

# Summary

- Entity-relationship (ER) model: a graphical way to model the world
- Main concepts:
  - Entity type
  - Relationship type
  - Attributes
- Different types of constraints
- Enhanced ER model