Exercise for *Database System Concepts for Non-Computer Scientist* im WiSe 18/19

Alexander van Renen (renen@in.tum.de)
http://db.in.tum.de/teaching/ws1819/DBSandere/?lang=en

Sheet 04

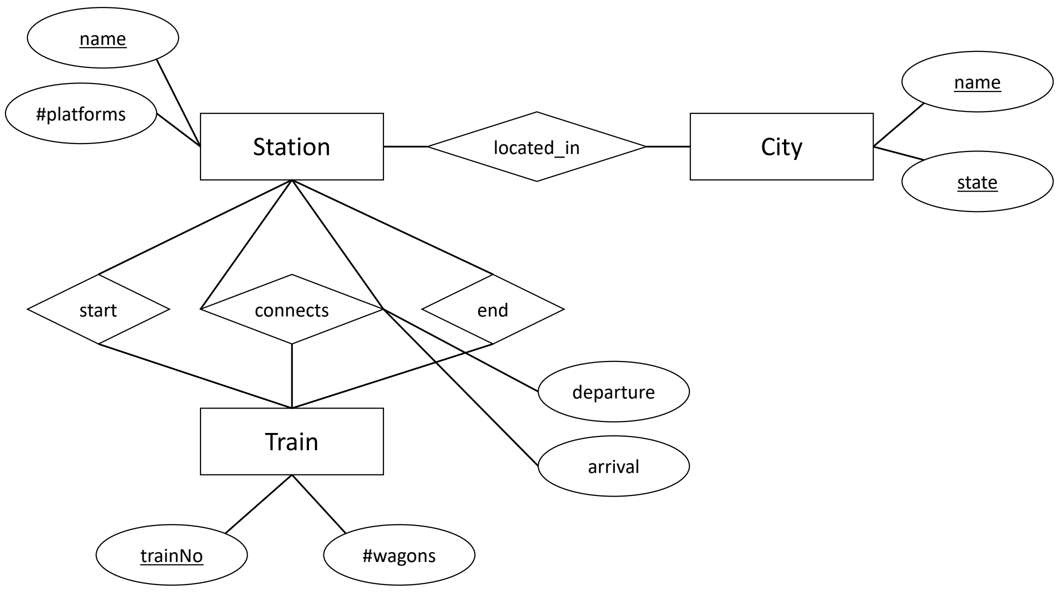
Exercise 1

Consider the entity relationship model of a train connection system (below). Note: `connects` models a the direct connection between two stations. For example, the train starting in Munich and ending in Hamburg passes through several stations. Each of these route-sections (e.g., Munich → Nürnberg or Nürnber → Würzburg) has an entry in the `connects` relation.

a) Add functionalities to the ER diagram.

b) Transform the ER diagram into a relational schema.

c) Refine the relational schema as far as possible by eliminating relations.



Solution:

a) Adding functionalities

Figure 1 shows the entity relationship model with functionalities.

b) Create a relational schema

The un-refined translation yields the following relations for the entities in the model:

$$\text{City} : \{[\underline{\text{name} : \text{string}, \text{state} : \text{string}}]\} \tag{1}$$

$$\text{Station} : \{[\underline{\text{name} : \text{string}}, \#\text{platforms} : \text{integer}]\} \tag{2}$$

$$\text{Train} : \{[\underline{\text{trainNo} : \text{integer}}, \#\text{wagons} : \text{integer}]\} \tag{3}$$

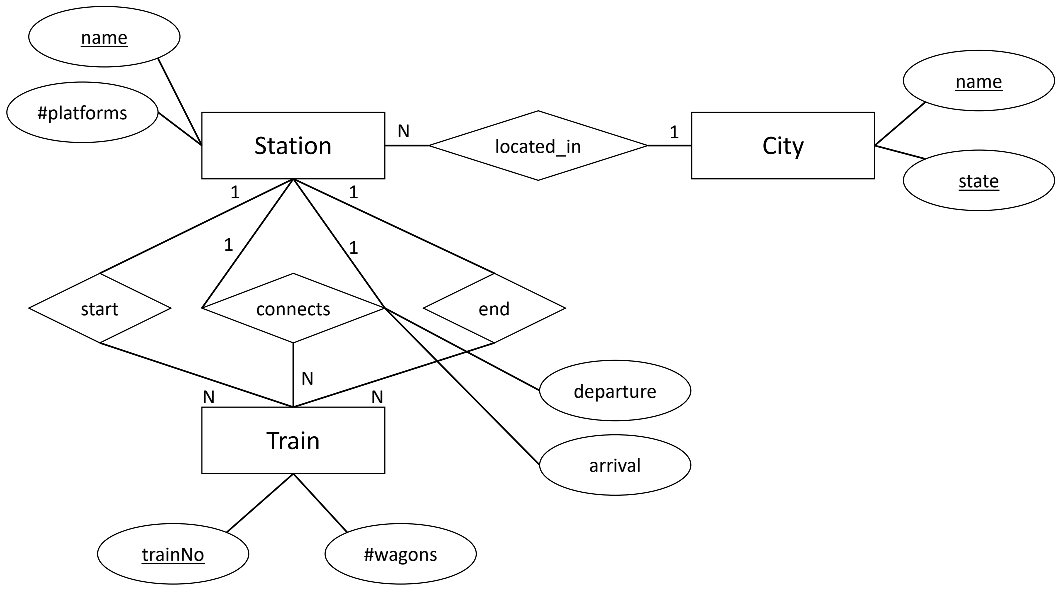


Figure 1: ER-Model for train connection with functionalities.

For the relationships in the model, we create the following relations:

$$\begin{aligned}
\text{located\_in} &: \{[\underline{\text{stationName} : \text{string}}, \text{cityName} : \text{string}, \text{cityState} : \text{string}]\} && (4)\\
\text{start} &: \{[\underline{\text{trainNo} : \text{integer}}, \text{stationName} : \text{string}]\} && (5)\\
\text{end} &: \{[\underline{\text{trainNo} : \text{integer}}, \text{stationName} : \text{string}]\} && (6)\\
\text{connects} &: \{[\underline{\text{fromStationName} : \text{string}}, \text{toStationName} : \text{string}, && (7)\\
&\quad \underline{\text{trainNo} : \text{integer}}, \text{departure} : \text{date}, \text{arrival} : \text{date}]\}
\end{aligned}$$

c) Refine the relational schema

Next, we refine the relational schema by combining relations.

In this step we merge relations for binary relationships into relations for entities, if the relations have the same key and it was a 1:N, N:1 or 1:1 relationship in the ER-model. Note: A binary 1:N relationship can be merged into the entity with the N next to it.

Doing so we can merge the (4) relation into (2). (5) gets merged into (3). And same for the *end* relation, which also gets merged into *train*.

$$(4) \mapsto (2),\ (5) \mapsto (3),\ (6) \mapsto (3)$$

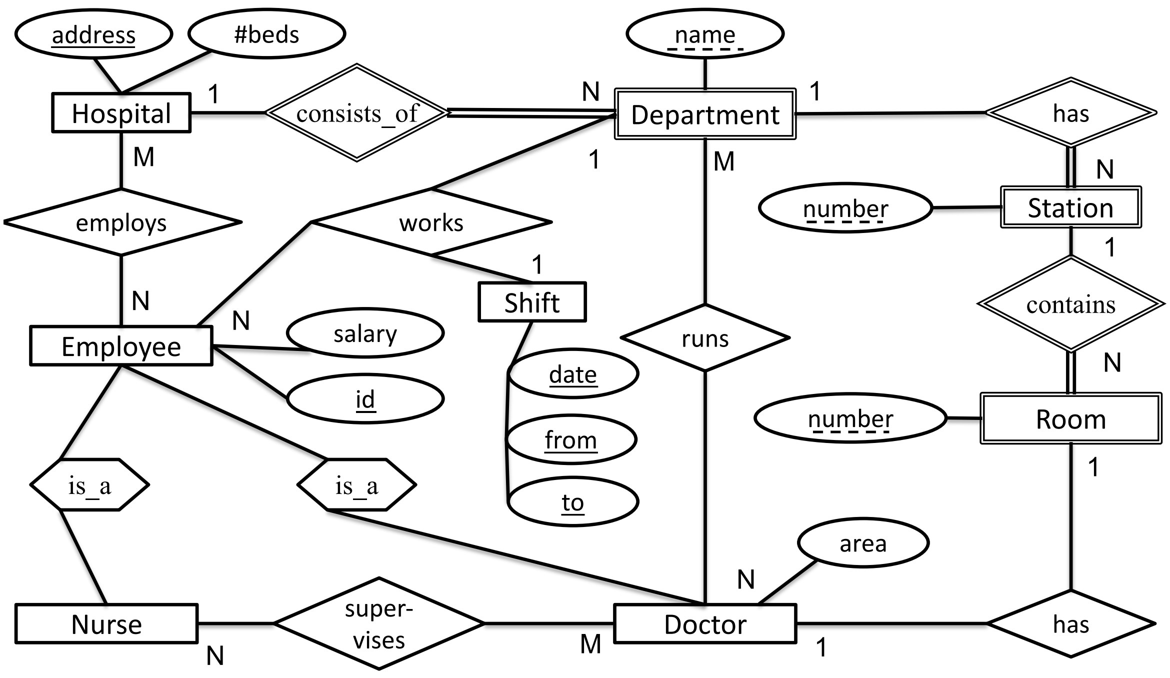
Thus, we end up with the following schema:

$$\begin{aligned}
\text{City} &: \{[\underline{\text{name} : \text{string}, \text{state} : \text{string}}]\}\\
\text{Station} &: \{[\underline{\text{name} : \text{string}}, \#\text{platforms} : \text{integer},\\
&\quad \text{cityName} : \text{string}, \text{state} : \text{string}]\}\\
\text{Train} &: \{[\underline{\text{trainNo} : \text{integer}}, \#\text{wagons} : \text{integer},\\
&\quad \text{startStationName} : \text{string}, \text{endStationName} : \text{string}]\}\\
\text{connects} &: \{[\underline{\text{fromStationName} : \text{string}}, \text{toStationName} : \text{string},\\
&\quad \underline{\text{trainNo} : \text{integer}}, \text{departure} : \text{date}, \text{arrival} : \text{date}]\}
\end{aligned}$$

In our model the train number is uniquely identifying a connection between two cities (possibly involving serveral stations). An ICE starting in Munich (*startStationName*) and going to Berlin (*endStationName*) has a unique train number. When the train returns it has a different train number. Therefore, in the *connects* relation, the (*trainNo*, *fromStationName*)-pair and the (*trainNo*, *toStationName*)-pair are both valid keys (as they are both uniquely identifying a tuple in the relation).

Exercise 2

Now, if you want more practice, consider the hospital example, again. This time take the entity relationship diagram and transform it into a relational schema. Then, optimize it by eliminating relations.



Solution:

a) Create a relational schema

The un-refined translation yields the following relations for the entities in the model:

$$\text{Hospital} : \{[\underline{\text{address : string}}, \#\text{beds : int}]\} \tag{8}$$
$$\text{Department} : \{[\underline{\text{address : string}, \text{name : string}}]\} \tag{9}$$
$$\text{Station} : \{[\underline{\text{address : string}, \text{name : string}, \text{stationNo : int}}]\} \tag{10}$$
$$\text{Room} : \{[\underline{\text{address : string}, \text{name : string}, \text{stationNo : int}, \text{roomNo : int}}]\} \tag{11}$$
$$\text{Employee} : \{[\underline{\text{id : int}}, \text{salary : int}]\} \tag{12}$$
$$\text{Nurse} : \{[\underline{\text{id : int}}]\} \tag{13}$$
$$\text{Doctor} : \{[\underline{\text{id : int}}, \text{area : string}]\} \tag{14}$$
$$\text{Shift} : \{[\underline{\text{date : date}}, \text{from : time}, \text{to : time}]\} \tag{15}$$

For the relationships in the model, we create the following relations:

$$\text{consists\_of} : \{[\underline{\text{address : string}, \text{departmentName : string}}]\} \tag{16}$$
$$\text{department\_has} : \{[\underline{\text{address : string}, \text{departmentName : string}, \text{stationNo : int}}]\} \tag{17}$$
$$\text{contains} : \{[\underline{\text{address : string}, \text{departmentName : string}, \text{stationNo : int}, \text{roomNo : int}}]\} \tag{18}$$
$$\text{employs} : \{[\underline{\text{address : string}, \text{id : int}}]\} \tag{19}$$
$$\text{supervises} : \{[\underline{\text{nurseId : int}, \text{doctorId : int}}]\} \tag{20}$$
$$\text{doctor\_has} : \{[\underline{\text{doctorId : int}}, \text{address : string}, \text{departmentName : string}, \text{stationNo : int}, \text{roomNo : int}]\} \tag{21}$$
$$\text{runs} : \{[\underline{\text{doctorId : int}, \text{address : string}, \text{name : string}}]\} \tag{22}$$
$$\text{works} : \{[\underline{\text{employeeId : int}, \text{address : string}, \text{name : string}}, \text{date : date}, \text{from : time}, \text{to : time}]\} \tag{23}$$

There are several alternative translation options:

1. The *is_a* relationship could have also been translated by merging the attributes of the *Employee* into the *Nurse* and *Doctor* relation:

$$\begin{aligned} \text{Nurse} &: \{[\underline{\text{id : int}}, \text{salary : int}]\} \\ \text{Doctor} &: \{[\underline{\text{id : int}}, \text{area : string}, \text{salary : int}]\} \end{aligned}$$

2. In the 1:1 relation *has* between *Doctor* and *Room* we could have also chosen the key of the *Room* as a key.

3. In the ternary relation *works* we could have also chosen (*employeeId*, *date*, *from*, *to*) as a key.

b) Refine the relational schema

Next, we refine the relational schema by combining relations.

All binary relations with 1:1, 1:N, N:1 can be refined in the following way:

First, we can eliminate all relations that originate from weak relationships in the ER-model. In this case we do not have to add additional keys to the entity we merge them into because they already have this key because they are weak entities:

$$(16) \mapsto (9),\ (17) \mapsto (10),\ (18) \mapsto (11)$$

Next, we take care of the *has* relation between *Doctor* and *Room*. This is a 1:1 relation and can therefore be merged into *Doctor* or *Room*. We choose to merge it into room, as this requires us to only add one attribute to *Room* instead of four to *Doctor*:

$$(21) \mapsto (11)$$

Now, there is no binary relation left with a 1:1, 1:N or N:1 functionality. Therefore, we are done and end up with the following relational schema:

$$
\begin{aligned}
\text{Hospital} &: \{[\underline{\text{address} : \text{string}}, \#\text{beds} : \text{int}]\} \\
\text{Department} &: \{[\underline{\text{address} : \text{string}, \text{name} : \text{string}}]\} \\
\text{Station} &: \{[\underline{\text{address} : \text{string}, \text{name} : \text{string}, \text{stationNo} : \text{int}}]\} \\
\text{Room} &: \{[\underline{\text{address} : \text{string}, \text{name} : \text{string}, \text{stationNo} : \text{int}, \text{roomNo} : \text{int}}, \text{doctorId} : \text{int}]\} \\
\text{Employee} &: \{[\underline{\text{id} : \text{int}}, \text{salary} : \text{int}]\} \\
\text{Nurse} &: \{[\underline{\text{id} : \text{int}}]\} \\
\text{Doctor} &: \{[\underline{\text{id} : \text{int}}, \text{area} : \text{string}]\} \\
\text{Shift} &: \{[\underline{\text{date} : \text{date}}, \text{from} : \text{time}, \text{to} : \text{time}]\}
\end{aligned}
$$

For the relationships in the model, we create the following relations:

$$
\begin{aligned}
\text{employs} &: \{[\underline{\text{address} : \text{string}, \text{id} : \text{int}}]\} \\
\text{supervises} &: \{[\underline{\text{nurseId} : \text{int}, \text{doctorId} : \text{int}}]\} \\
\text{runs} &: \{[\underline{\text{doctorId} : \text{int}, \text{address} : \text{string}, \text{name} : \text{string}}]\} \\
\text{works} &: \{[\underline{\text{employeeId} : \text{int}, \text{address} : \text{string}, \text{name} : \text{string},} \\
&\quad\ \text{date} : \text{date}, \text{from} : \text{time}, \text{to} : \text{time}]\}
\end{aligned}
$$