TU München, Fakultät für Informatik
Lehrstuhl III: Datenbanksysteme
Prof. Alfons Kemper, Ph.D.

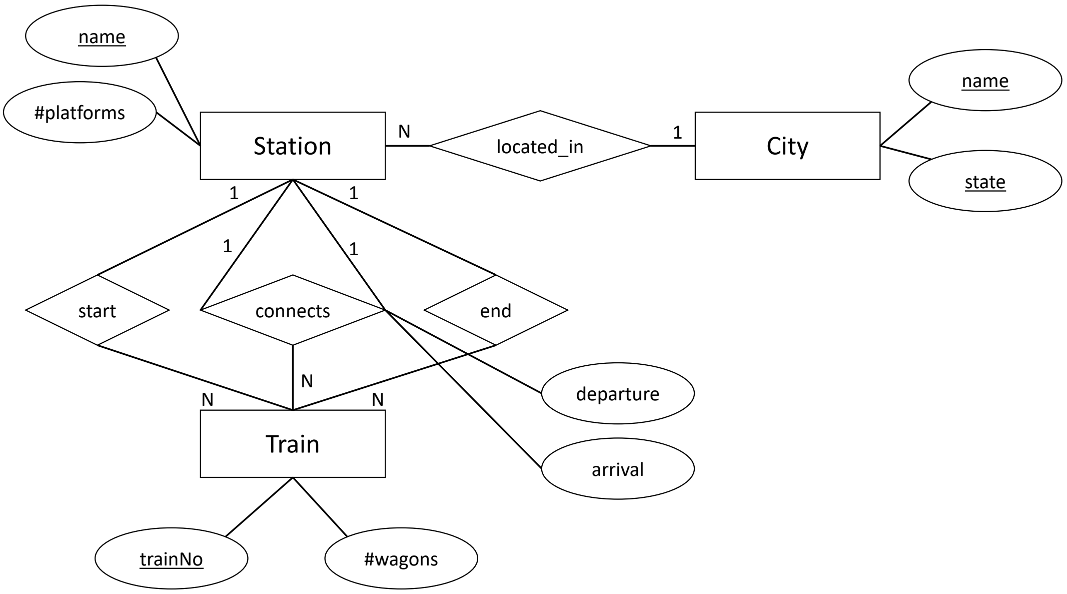
***Database System Concepts for Non-Computer Scientist* - WiSe 23/24**

Alice Rey (rey@in.tum.de)
http://db.in.tum.de/teaching/ws2324/DBSandere/?lang=en

Sheet 05

Exercise 1

Consider the entity relationship diagram from exercise sheet 3:



Refine the relational schema that you created in sheet 4 from the ER-Diagram. Underline keys and find appropriate data types. As a reminder, here is the un-refined schema:

$$\begin{aligned} \text{City} &: \{[\underline{\text{name : string}, \text{state : string}}]\} && (1)\\ \text{Station} &: \{[\underline{\text{name : string}}, \text{\#platforms : integer}]\} && (2)\\ \text{Train} &: \{[\underline{\text{trainNo : integer}}, \text{\#wagons : integer}]\} && (3)\end{aligned}$$

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

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

Solution:

During refinement, 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:

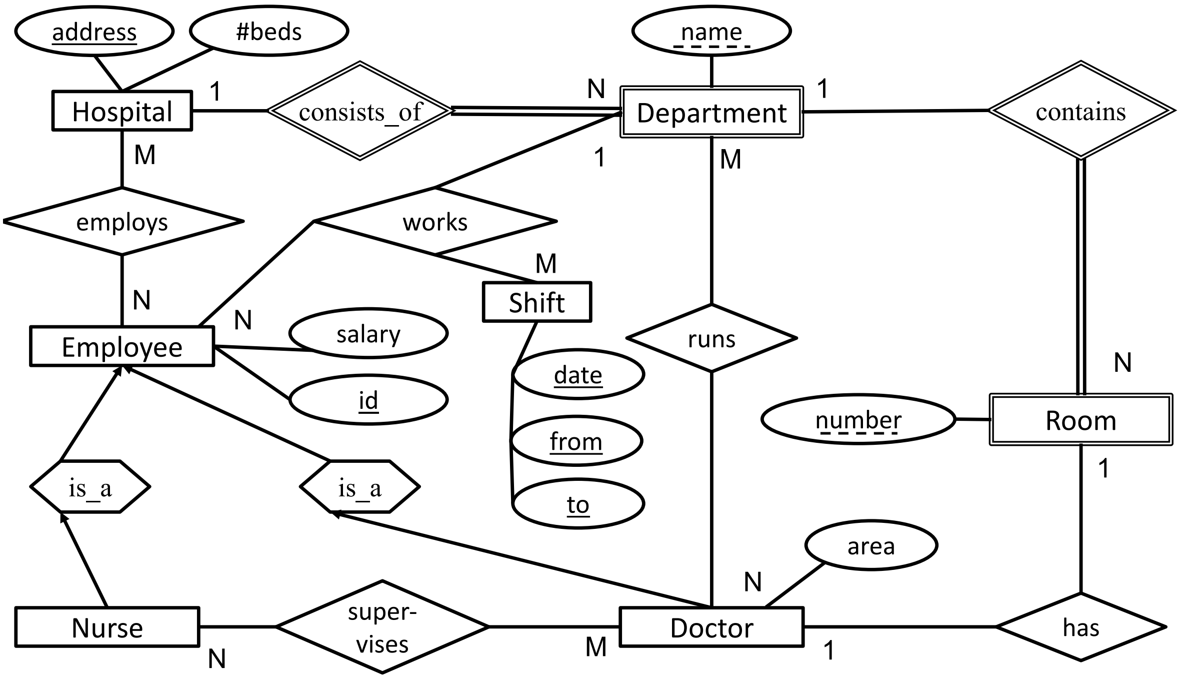
$$\begin{aligned}
\text{City} &: \{[\underline{\text{name} : \text{string}, \text{state} : \text{string}}]\}\\
\text{Station} &: \{[\underline{\text{name} : \text{string}}, \#\text{platforms} : \text{integer},\\
&\quad\ \text{cityName} : \text{string}, \text{state} : \text{string}]\}\\
\text{Train} &: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

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

For additional 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.

This is obviously a large example but practice is very helpful. However, if you want to save time, you could focus on the difficult parts: *employs*, *works*, *consists_of*, *Doctors + has*



Solution:

a) Create a relational schema

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

$$\text{Hospital} : \{[\underline{\text{address} : \text{string}}, \#\text{beds} : \text{int}]\} \tag{8}$$

$$\text{Department} : \{[\underline{\text{address} : \text{string}, \text{name} : \text{string}}]\} \tag{9}$$

$$\text{Room} : \{[\underline{\text{address} : \text{string}, \text{name} : \text{string}, \text{roomNo} : \text{int}}]\} \tag{10}$$

$$\text{Employee} : \{[\underline{\text{id} : \text{int}}, \text{salary} : \text{int}]\} \tag{11}$$

$$\text{Nurse} : \{[\underline{\text{id} : \text{int}}]\} \tag{12}$$

$$\text{Doctor} : \{[\underline{\text{id} : \text{int}}, \text{area} : \text{string}]\} \tag{13}$$

$$\text{Shift} : \{[\underline{\text{date} : \text{date}, \text{from} : \text{time}, \text{to} : \text{time}}\} \tag{14}$$

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

$$
\begin{aligned}
\text{consists\_of} &: \{[\underline{\text{address} : \text{string}, \text{departmentName} : \text{string}}]\} && (15)\\
\text{contains} &: \{[\underline{\text{address} : \text{string}, \text{departmentName} : \text{string},} && (16)\\
&\quad \underline{\text{roomNo} : \text{int}}]\}\\
\text{employs} &: \{[\underline{\text{address} : \text{string}, \text{id} : \text{int}}]\} && (17)\\
\text{supervises} &: \{[\underline{\text{nurseId} : \text{int}, \text{doctorId} : \text{int}}]\} && (18)\\
\text{doctor\_has} &: \{[\underline{\text{doctorId} : \text{int}}, \text{address} : \text{string}, \text{departmentName} : \text{string}, && (19)\\
&\quad \text{roomNo} : \text{int}]\}\\
\text{runs} &: \{[\underline{\text{doctorId} : \text{int}, \text{address} : \text{string}, \text{name} : \text{string}}]\} && (20)\\
\text{works} &: \{[\underline{\text{employeeId} : \text{int}, \text{date} : \text{date}, \text{from} : \text{time}, \text{to} : \text{time}}, && (21)\\
&\quad \text{address} : \text{string}, \text{name} : \text{string}]\}
\end{aligned}
$$

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:

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

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

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:

$$(15) \mapsto (9),\ (16) \mapsto (10)$$

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*:

$$(19) \mapsto (10)$$

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:

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

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

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