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

Exercise for *Database System Concepts for Non-Computer Scientist* im WiSe 19/20

Alexander van Renen (renen@in.tum.de)
http://db.in.tum.de/teaching/ws1920/DBSandere/?lang=en

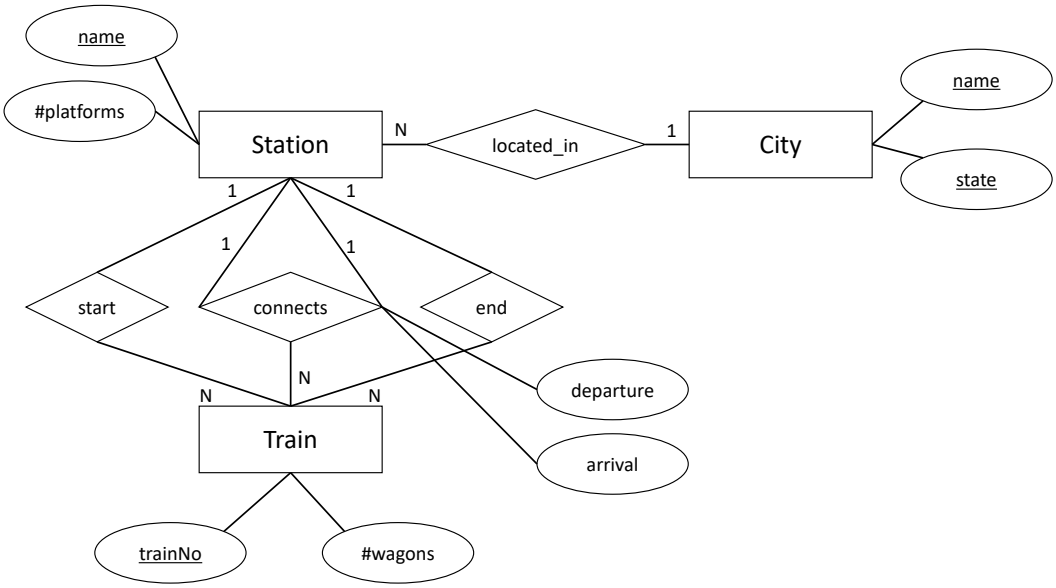
Sheet 05

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 Nürnberg → Würzburg) has an entry in the `connects` relation.

c) Refine the relation schema as far as possible.

d) Create SQL DDL statements to create the respective tables from the refined relational schema.



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

$$\text{City} : \{[\underline{\text{name} : \text{string}}, \text{state} : \text{string}]\} \quad (1)$$

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

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

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

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

Solution:

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).

c) Create a database schema

```
create table City (
    name varchar(30),
    state varchar(30),
    primary key(name, state)
);

create table Station (
    name varchar(30) primary key,
    platforms int,
    cityName varchar(30),
```

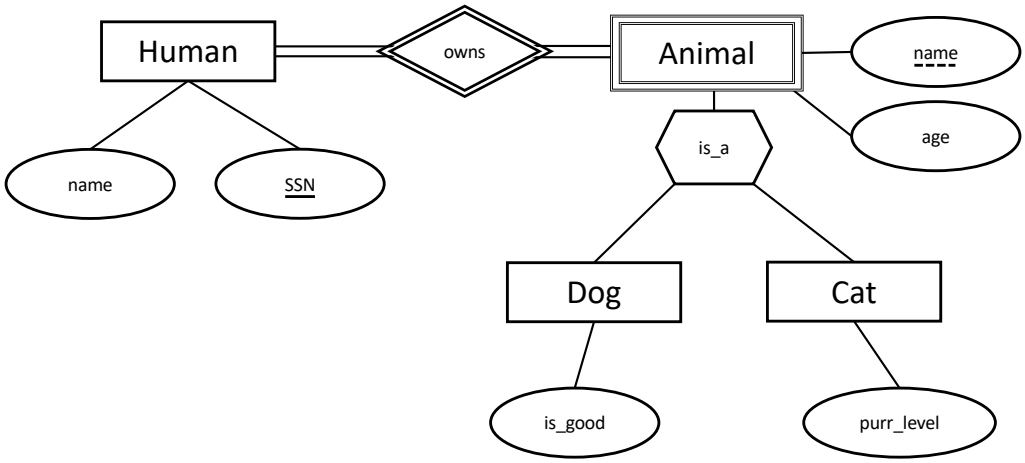
```
    cityState varchar(30),
    foreign key(cityName, cityState) references City
);

create table Train (
    trainNo int primary key,
    wagons int,
    startStationName varchar(30) references Station,
    endStationName varchar(30) references Station
);

create table connects (
   fromStationName varchar(30) references Station,
   toStationName varchar(30) references Station,
   trainNo int,
   departure date,
   arrival date,
   primary key(fromStationName, trainNo),
   unique(toStationName, trainNo)
);
```

Exercise 2

Look at the following ER-diagram. Think about different ways of how to transform these into a database schema.



Lösung:

Option one (inclusive):

```
create table Human(name varchar(30),
                   SSN char(9) primary key);

create table Dog(SSN char(9),
                 name varchar(30),
                 age int,
                 is_good char(1),
```

```
                    primary key(SSN, name));

create table Cat(SSN char(9),
                    name varchar(30),
                    age int,
                    purr_level int,
                    primary key(SSN, name));
```

Option two (inherited):

```
create table Human(name varchar(30),
                      SSN char(9) primary key);

create table Animal(SSN char(9) references Human,
                       name varchar(30),
                       age int,
                       primary key(SSN, name));

create table Dog(SSN char(9),
                   name varchar(30),
                   is_good char(1),
                   primary key(SSN, name),
                   foreign key(SSN, name) references Animal);

create table Cat(SSN char(9),
                   name varchar(30),
                   purr_level int,
                   primary key(SSN, name),
                   foreign key(SSN, name) references Animal);
```