Database System Concepts for Non-Computer Scientist - WiSe 24/25

Alice Rey (rey@in.tum.de)
http://db.in.tum.de/teaching/ws2425/DBSandere/?lang=en

Sheet 01

Exercise 1

You are designing a web application for the university (i.e., TUMOnline). Early on, you decide to use a databases management system (DBMS) to store data on the server-side. One of your colleges is sceptical about using a DBMS and would prefer to implement the data management themselves. Convince them of your decision! Find solid arguments to debunk the following claims of your college:

a) "The setup and maintenance of a DBMS is laborious. Creating our own data format is straight-forward and more flexible."

b) "We do not need multi user synchronization (i.e., concurrency control)."

c) "It is an unnecessary effort to make every developer learn a new query language (SQL), just to extract data from the DBMS."

d) "Having redundancy is useful, why should we avoid it, as in the DBMS"

Solution:

Some possible arguments (non exhaustive):

a) Custom data formats are usually not flexible. As they, by their very definition, are custom to a company. Therefore, they do not play well with other infrastructure (archiving data, distribution among servers and data centers, recovery). For example, depending on which file system is used, there is a limit on the file size (2GB for the older FAT32 file system). Lastly, even the effort for setting up a complex product like IBM DB2[1] is neglectable compared to re-implementing (a subset of) its features again. For an easy setup an embedded DBMS, like sqlite[2], could be used.

b) Concurrency control is inherently necessary when developing a system that is going to be used by multiple users. It is neither easy to develop nor can it be patched-in later on. A DBMS, in general, implements a sound concurrency protocol which allows multiple users to interact with the same data without noticing "unintended" side effects.

c) A custom data format and the API to use it have to be learned as well (assuming there is even an API to access data). SQL is a well known and documented language (you can find help for most problems on the web). Using SQL even reduces the effort when switching to another database system.

d) Redundancy causes anomalies or leads to inconsistencies, for example when updating data.

[1] `http://www-01.ibm.com/software/data/db2/`
[2] `http://www.sqlite.org/`

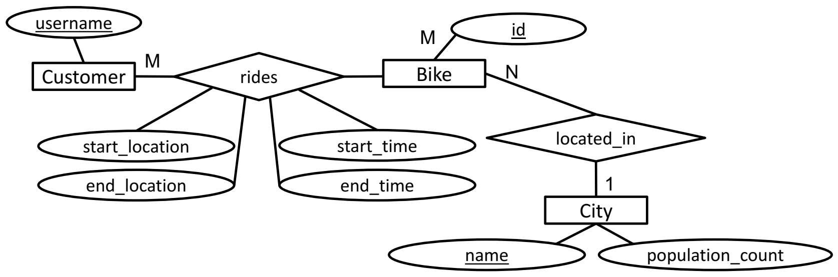
Exercise 2

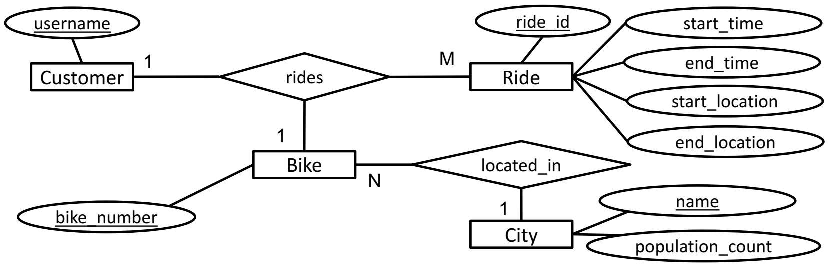
iBike is a new free-floating bike sharing app in Munich. Customers are able to rent and return bikes wherever they find them within the city limit. The company wants you to design an Entity Relationship (ER) Diagram for their business.

- Model an ER Diagram for the data schema for the iBike app. Some things that need to be included are *Customers*, *Bikes*, and *Rides*. For analysis purposes, we want to keep track of start and end time as well as pick-up and drop-off locations of rides.
- After a successful deployment in Munich the iBike company expands to other cities. Any given bike belongs to exactly one city. A city is identified by a name and has a population count. Modify the ER diagram accordingly.

Solution:

The following two diagrams show two possible solutions. Note that neither one is limiting a customer to only rent one bike at a time.



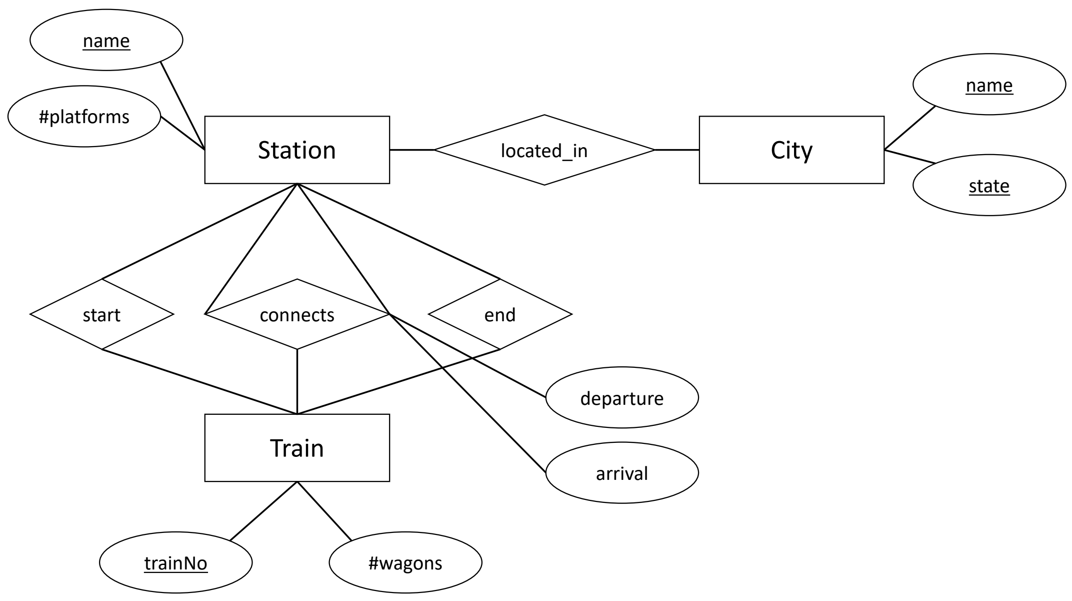


Homework 3

Consider the following description of a train connection system and create an entity relationship diagram for it:

- Train lines have unique train numbers. They have a starting station and an ending station. They have a specified number of wagons.
- The train from Munich to Hamburg becomes a different train line (different train number) when returning
- On the route, the train directly connects pairs of stations (e.g., the train from Munich to Berlin directly connects Munich and Nürnberg, and Nürnberg and Würzburg, ...) with a departure and arrival time between each pair.
- Each train station is located in a city

- A city can be identified by name and state
- Each station has a number of platforms and is identified by its name


name
#platforms
Station
located_in
City
name
state
start
connects
end
departure
arrival
Train
trainNo
#wagons