

Übung zur Vorlesung *Grundlagen: Datenbanken* im WS21/22
Christoph Anneser, Josef Schmeißer, Moritz Sichert, Lukas Vogel (gdb@in.tum.de)
<https://db.in.tum.de/teaching/ws2122/grundlagen/>

Blatt Nr. 02

Tool zum Üben der relationalen Algebra: <https://dbis-uibk.github.io/relax/calc>.
Für das Unischema im Dropdown links „Kemper Datenbanksysteme“ auswählen.

Hausaufgabe 1

Modellieren Sie ein Wahlsystem für die Bundestagswahl 2021. Das System soll Wahlberechtigte und WahlkandidatInnen in ihren jeweiligen Wahlkreisen und -bezirken verwalten. Da Wahlberechtigte einzelne KandidatInnen einer Liste mit ihrer Zweitstimme wählen können, soll es auch die Listen der jeweiligen Parteien modellieren. Es muss außerdem sicherstellen, dass das Wahlrecht eingehalten wird. Dazu gehört insbesondere, dass jede Person nur höchstens eine Stimme abgeben kann und die Wahl geheim ist (also auch das System selbst nicht wissen kann, wer wie abgestimmt hat). Betrachten Sie für die Modellierung die folgenden Beschreibungen:

Parteien Es gibt mehrere Parteien mit unterschiedlichen Namen. Jede Partei kann (muss aber nicht) in jedem Wahlkreis eine/n KandidatIn für die Erststimme und in jedem Bundesland eine Liste für die Zweitstimme aufstellen. Für „Parteien nationaler Minderheiten“ entfällt die 5%-Hürde.

Bundesländer Deutschland hat folgende Bundesländer: Baden-Württemberg, Bayern, Berlin, Brandenburg, Bremen, Hamburg, Hessen, Mecklenburg-Vorpommern, Niedersachsen, Nordrhein-Westfalen, Rheinland-Pfalz, Saarland, Sachsen-Anhalt, Schleswig-Holstein, Thüringen. Ein Bundesland enthält mehrere Wahlkreise. Pro Bundesland kann jede Partei mit einer Landesliste für die Zweitstimme antreten.

Wahlkreise Jeder Wahlkreis (z.B. „261 Esslingen“) ist genau einem Bundesland zugeordnet. Eine Partei kann hier eine/n KandidatIn für die Erststimme aufstellen. Allerdings kann es auch parteilose KandidatInnen geben, die nur für sich selbst antreten. Die Wahlberechtigten sind einem Wahlkreis zugehörig.

Landeslisten Eine Landesliste besteht aus mehreren KandidatInnen aufgestellt von (aber nicht notwendigerweise zugehörig zu) einer Partei. Sie ist spezifisch für ein Bundesland. Wahlberechtigte können mit ihrer Zweitstimme genau eine ganze Liste wählen. Nicht jede Partei muss für jedes Land eine Landesliste aufstellen. Landeslisten haben eine Reihenfolge.

KandidatInnen Ein/e KandidatIn kann in einem Wahlkreis antreten und dann mit der Erststimme gewählt werden, unabhängig davon, welcher Partei er/sie angehört (oder ob er/sie überhaupt in einer ist). Zusätzlich kann er/sie auf der Landesliste einer Partei stehen.

Wahlberechtigte Wahlberechtigte leben in einem Wahlkreis. Dort können sie eine Erst- und eine Zweitstimme abgeben. Mit der Erststimme können sie genau eine/n KandidatIn,

der/die in ihrem Wahlkreis antritt, wählen. Mit der Zweitstimme können sie eine Partei wählen, die für ihr Bundesland eine Landesliste aufgestellt hat. Sie können auch jede Stimme individuell ungültig abgeben. Stimmen können entweder in Person im Wahllokal oder per Briefwahl abgegeben werden.

Weitere Informationen zum Wahlrecht der Bundestagswahl finden Sie hier: <https://www.wahlrecht.de/bundestag/>.

Entwerfen Sie ein ER-Modell für dieses System. Sie müssen dabei nur die in der Aufgabe erwähnten Sachverhalte berücksichtigen.

- a) Identifizieren Sie alle relevanten Entity-Typen und ihre Attribute.
- b) Beschreiben Sie alle notwendigen Relationships zwischen den Entity-Typen. Achten Sie darauf, dass das Wahlrecht eingehalten wird.
- c) Ergänzen Sie die Relationships mit Funktionalitätsangaben. Fügen Sie auch (min,max)-Angaben hinzu.
- d) Bildet Ihr Modell die Realität komplett ab? Welche Gegebenheiten kann Ihr Modell nicht ausdrücken? Warum nicht?

Lösung: Abbildung 1 zeigt ein mögliches ER-Diagramm. Abbildung 2 enthält zusätzlich Funktionalitätsangaben, Abbildung 3 zusätzlich (min,max)-Angaben.

- a) Aus der obigen Liste ergeben sich die zwingend notwendigen Entity-Typen. Die Beispiellösung modelliert zusätzlich die Erst- und Zweitstimme als getrennte Entity-Typen, da sie mit jeweils anderen Entity-Typen in Beziehung stehen. Der Listenplatz ist als getrennter Entity-Typ modelliert um über sein Attribut „Rang“ den Rang der Kandidatin/des Kandidaten auf einer Liste ausdrücken zu können.

Die Beschreibungen enthalten nicht alle Informationen über die erforderlichen Attribute der Entity-Typen, also müssen diese nach eigenem Ermessen hinzugefügt werden. Die meisten Entity-Typen haben z.B. einen Namen. Wahlkreise haben eine Nummer, Wahlberechtigte eine Anschrift (benötigt um die Wahlunterlagen versenden zu können) und KandidatInnen einen Beruf (der auf dem Stimmzettel stehen muss). In der Beispielmodellierung haben Stimmen zusätzlich eine „ist gültig“-Attribut, da das Wahlrecht zwischen einer ungültigen und einer nicht abgegebenen Stimme unterscheidet.

- b) Besonders zu beachten ist hier, dass es keine Beziehung zwischen Wahlberechtigten und ihrer Stimme geben darf. Die Systemadministratoren, die Programmierer oder der Staat selbst könnten sonst herausfinden wer wen gewählt hat: Die Wahl wäre, anders als vom Wahlrecht gefordert, nicht mehr geheim.
- c) Abbildung 2 zeigt das ER-Diagramm mit Funktionalitätsangaben, Abbildung 3 zeigt es mit (min,max)-Angaben.

Dabei sind (min, max)-Angaben ausdrucksstärker als Funktionalitätsangaben. Mit ihnen lässt sich z.B. ausdrücken, dass

- es in der aktuellen Bundestagswahl pro Wahlkreis maximal 64 Stimmkreise (in NRW) gab,
- eine Partei höchstens genau einmal in jedem der 16 Bundesländern antreten kann ((0, 16) an „Parteien“)

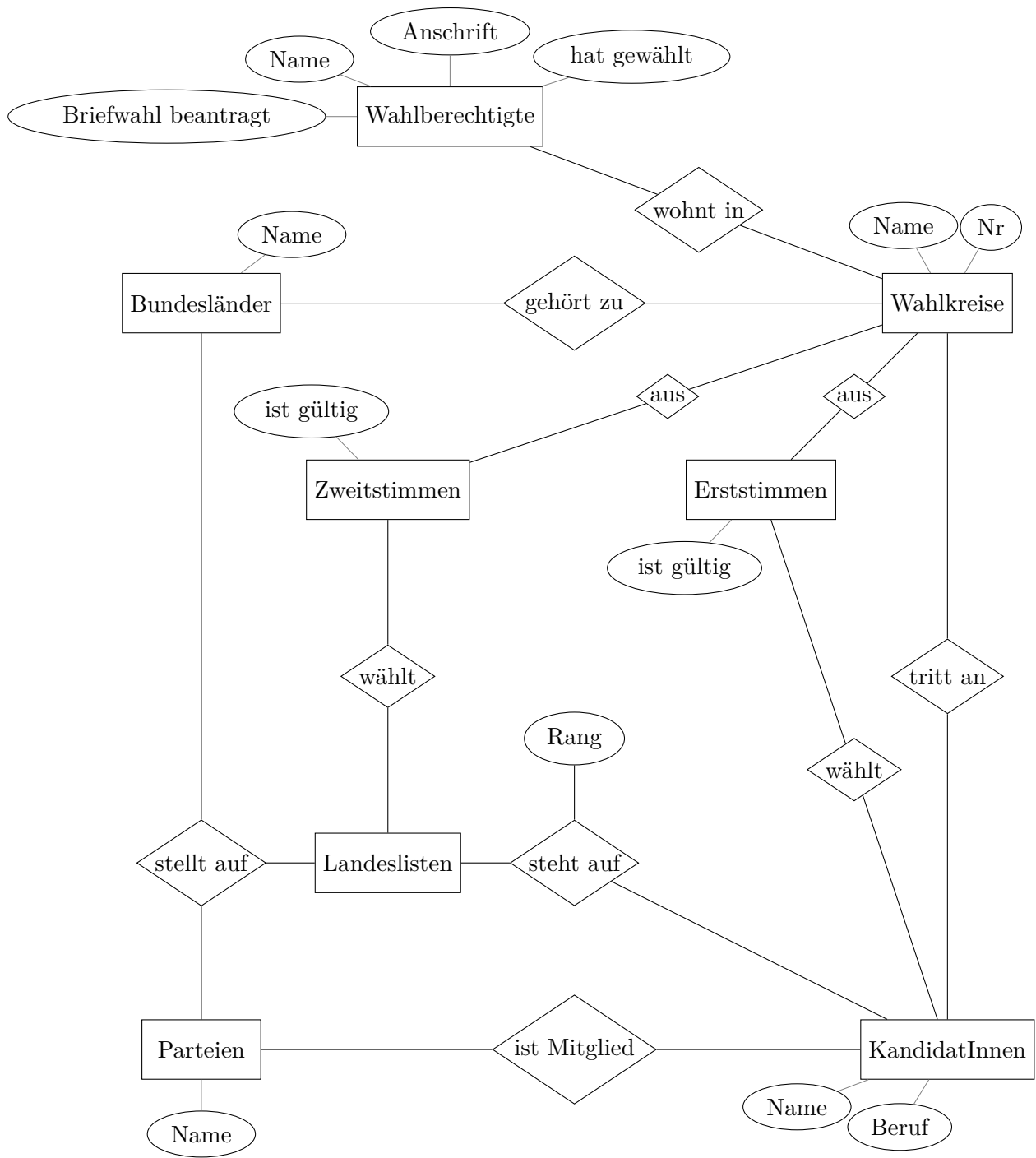


Abbildung 1: ER-Diagramm zur Bundestagswahl

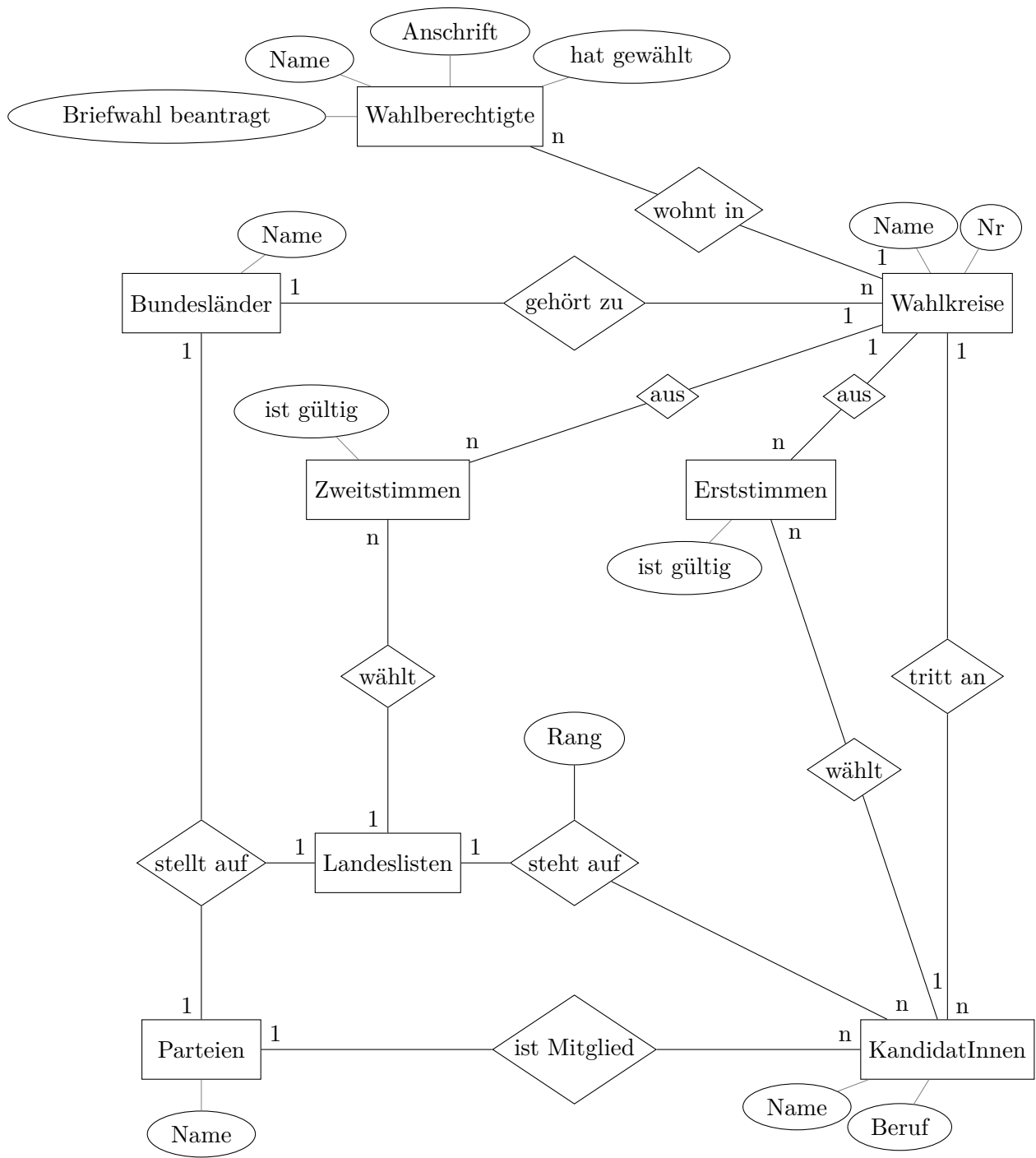


Abbildung 2: ER-Diagramm zur Bundestagswahl mit Funktionalitätsangaben

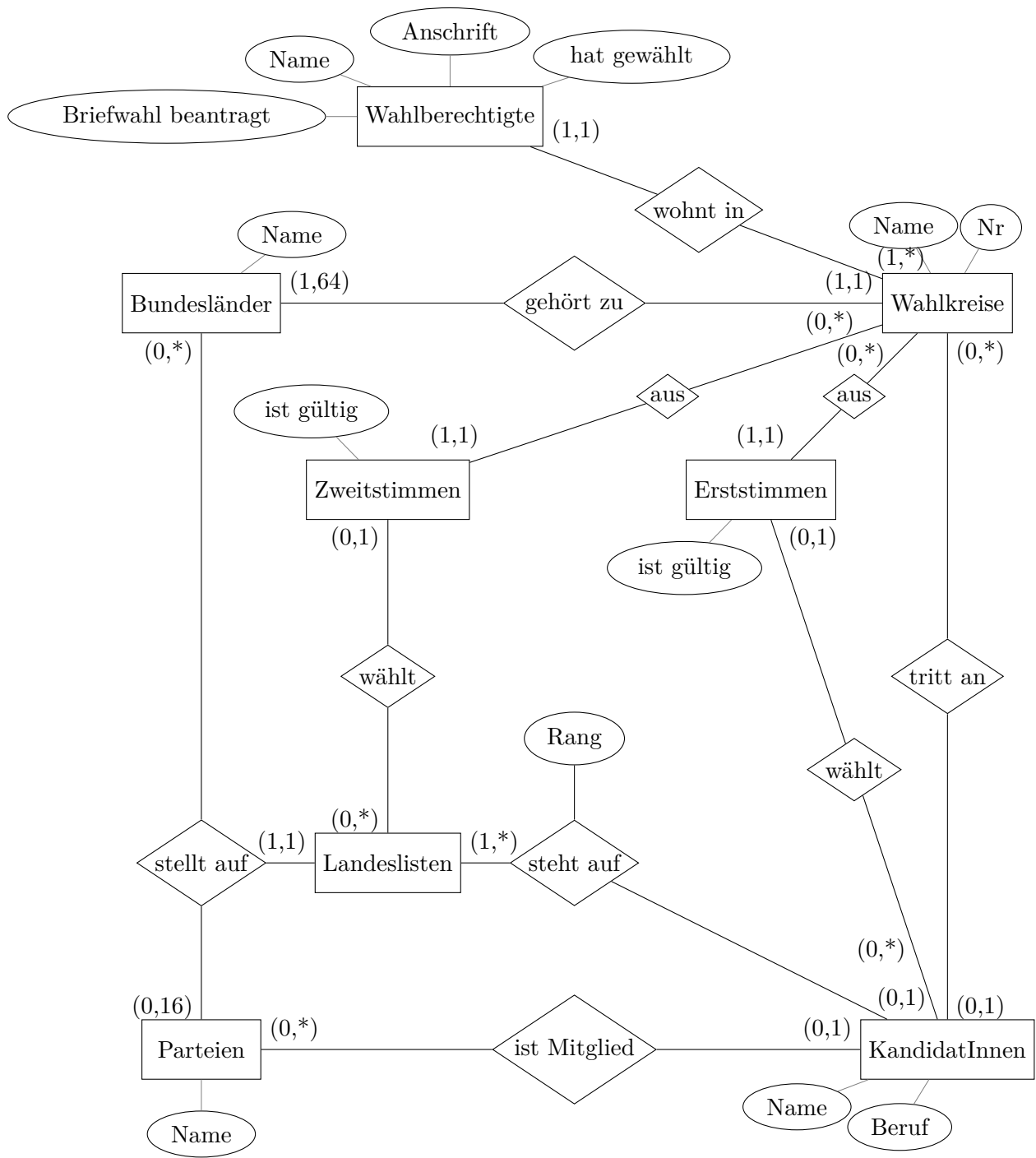
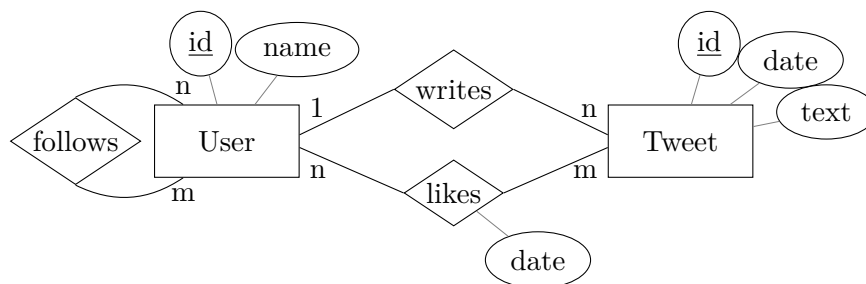


Abbildung 3: ER-Diagramm zur Bundestagswahl mit (min,max)-Angaben

- eine Partei keine Listen aufstellen ((0,16) an „Parteien“) - und ein Bundesland keine Liste haben ((0,*) an „Bundesländern“) muss (jeweils in der ternären Beziehung „stellt auf“),
 - ein/e KandidatIn nicht Mitglied einer Partei sein muss, und
 - ein/e KandidatIn nicht auf einer Liste stehen muss (jeweils (0,1) an „KandidatIn“).
- d) Dieses Modell bildet natürlich nicht die Realität in allen Einzelheiten ab. Hier einige Begebenheiten, die wir entweder nicht modelliert haben, oder die ein ER-Diagramm alleine nicht abbilden kann:
- KandidatInnen selbst sind wahlberechtigt
 - Man kann mit der Erststimme nur eine/n KandidatIn aus dem eigenen Wahlkreis wählen
 - Man kann mit der Erststimme nur eine/n KandidatIn wählen, die/der sich auf das Direktmandat bewirbt
 - Ein/e KandidatIn kann sich nur in seinem/ihrem Wahlkreis um das Direktmandat bewerben
 - Wenn die Erststimme für eine/n parteilose/n KandidatIn ist, der/die in den Bundestag einzieht, verfällt die Zweitstimme.
 - Unter den DirektkandidatInnen eines Wahlkreises darf es nicht zwei BewerberInnen derselben Partei geben.
 - Stimmen werden in einem Wahllokal abgegeben, oder per Briefwahl. Je nachdem gehört die Stimme zu einem jeweils anderen Wahlbezirk. Mehrere Wahlbezirke können sich ein Wahllokal teilen.
 - Auch wenn man Briefwahl beantragt hat, kann man noch in einem Wahllokal wählen, insofern man seine Briefwahlunterlagen mitbringt.

Hausaufgabe 2

Gegeben sei folgendes ER-Diagramm, das User, deren Tweets, Likes und Follows modelliert:



- Übertragen Sie das ER-Modell in ein relationales Schema.
- Verfeinern Sie das relationale Schema durch Elimination von Relationen.
- Kann das Attribut *date* des Entity-Typs *Tweet* stattdessen der Relationship *writes* zugeordnet werden? Kann das Attribut *date* der Relationship *likes* dem Entity-Typen *Tweet* zugeordnet werden?

Lösung:

a)

User : { [id,name] }
Tweet : { [id,date, text] }
follows : { [follower_id, follows_id] }
writes : { [tweet_id, user_id] }
likes : { [user_id, tweet_id, date] }

b) Die Relationen *Tweet* und *writes* können zusammengefasst werden, da sie den gleichen Schlüssel haben und die Relationship eine 1-zu-n-Beziehung ist. Eine weitere Verfeinerung ist nicht möglich.

User : { [id,name] }
Tweet : { [id, user_id, date, text] }
follows : { [follower_id, follows_id] }
likes : { [user_id, tweet_id, date] }

c) Das Attribut *date* kann auch *writes* zugeordnet werden. Insbesondere nach der Verfeinerung sind alle Attribute von *writes* und von *Tweet* in einer Relation zusammengefasst. Für das relationale Schema ist es also egal, ob das Attribut ursprünglich zu *writes* oder zu *Tweet* gehört hat.

Das Attribut *date* der Relation *likes* kann keinem anderen Entity-Typen zugeordnet werden, da weder *User* noch *Tweet* wegen ihrer n-zu-m-Beziehung ein eindeutiges „like_date“-Attribut haben können.

Hausaufgabe 3

Gegeben seien die beiden Relationen $R : \{[a, b]\}$ und $S : \{[b, c]\}$. Ersetzen Sie den folgenden Ausdruck der relationalen Algebra durch einen äquivalenten, in dem keine Joins vorkommen. Für diese Aufgabe zählt das Kreuzprodukt (\times) nicht als Join.

$$R \triangleright S$$

Lösung:

$$\begin{aligned} R \triangleright S &= R - R \times S \\ &= R - \Pi_{a,b}(R \bowtie S) \\ &= R - \Pi_{a,b}(\sigma_{R.b=S.b}(R \times S)) \end{aligned}$$