



Übung zur Vorlesung *Grundlagen: Datenbanken* im WS18/19

Moritz Sichert, Lukas Vogel (gdb@in.tum.de)

<https://db.in.tum.de/teaching/ws1819/grundlagen/>

Blatt Nr. 02

Tool zum Üben der relationalen Algebra:

<https://tools.db.in.tum.de/ira/>

Hausaufgabe 1

Modellieren Sie ein Wahlsystem für die 18. Landtagswahl in Bayern. Das System soll Wahlberechtigte und Wahlkandidaten in ihren jeweiligen Wahl- und Stimmkreisen verwalten. Da Wahlberechtigte einzelne Kandidaten 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:

Partei Es gibt mehrere Parteien mit unterschiedlichen Namen. Jede Partei kann in jedem Stimmkreis eine Kandidatin für die Erststimme und in jedem Wahlkreis eine Liste für die Zweitstimme aufstellen.

Wahlkreis Bayern hat zum Zeitpunkt der 18. Landtagswahl genau folgende Wahlkreise: Oberbayern, Niederbayern, Oberpfalz, Oberfranken, Mittelfranken, Unterfranken und Schwaben. Ein Wahlkreis enthält mehrere Stimmkreise. Pro Wahlkreis kann jede Partei mit einer Liste für die Zweitstimme antreten.

Stimmkreis Jeder Stimmkreis (z.B. „117 Freising“) ist genau einem Wahlkreis zugeordnet. Eine Partei kann hier eine Kandidatin für die Erststimme aufstellen. Allerdings kann es auch parteilose Kandidaten geben, die nur für sich selbst antreten. Die Wahlberechtigten und deren Stimmen werden in den Stimmkreisen verwaltet.

Liste Eine Liste besteht aus mehreren Kandidaten einer Partei. Sie ist spezifisch für einen Wahlkreis. Wahlberechtigte können mit ihrer Zweitstimme eine ganze Liste oder einen bestimmten Kandidaten einer Liste wählen.

Kandidatin Eine Kandidatin kann in ihrem Stimmkreis antreten und dann mit der Erststimme gewählt werden, unabhängig davon, welcher Partei sie angehört (oder ob sie überhaupt in einer ist). Zusätzlich kann sie auf der Liste einer Partei stehen. Das bedeutet, dass sie von Wahlberechtigten mit der Zweitstimme direkt gewählt werden kann.

Wahlberechtigte Wahlberechtigte leben in einem Stimmkreis, in dem sie eine Erst- und eine Zweitstimme abgeben können. Mit der Erststimme können sie genau eine Kandidatin, die in ihrem Stimmkreis antritt, wählen. Mit der Zweitstimme können sie entweder eine Kandidatin ihrer zugehörigen Wahlkreisliste oder die Liste selbst ohne die Auswahl einer bestimmten Kandidatin wählen. Sie können auch jede Stimme individuell ungültig abgeben.

Weiter Informationen zum bayrischen Wahlrecht finden Sie hier: <https://www.wahlrecht.de/landtage/bayern.htm>.

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 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. Stimmkreise haben eine Nummer, Wahlberechtigte eine Anschrift (benötigt um die Wahlunterlagen versenden zu können) und Kandidaten einen Beruf (der auf dem Stimmzettel stehen muss). In der Beispiellösung 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 einer 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.

Von besonderem Interesse sind die Relationen zwischen Zweitstimme, Liste und Listenplatz. Da man mit seiner Zweitstimme *entweder* einen Kandidaten auf einer Liste *oder* eine ganze Liste wählt, muss es eine Beziehung zu beiden Entitätstypen geben. Die Anforderung, dass nur eine der beiden Beziehungen bestehen darf, lässt sich mit einem ER-Diagramm nicht ausdrücken.

Die beiden ternären Beziehungen lesen sich wie folgt: „Eine Partei stellt pro Wahlkreis bis zu eine Liste auf“ und „Eine Kandidatin kann auf bis zu einer Liste einen Listenplatz haben. Eine Liste hat beliebig viele Listenplätze“. Beide Beziehungen könnte man jeweils auch äquivalent durch zwei binäre Beziehungen abbilden.

- 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

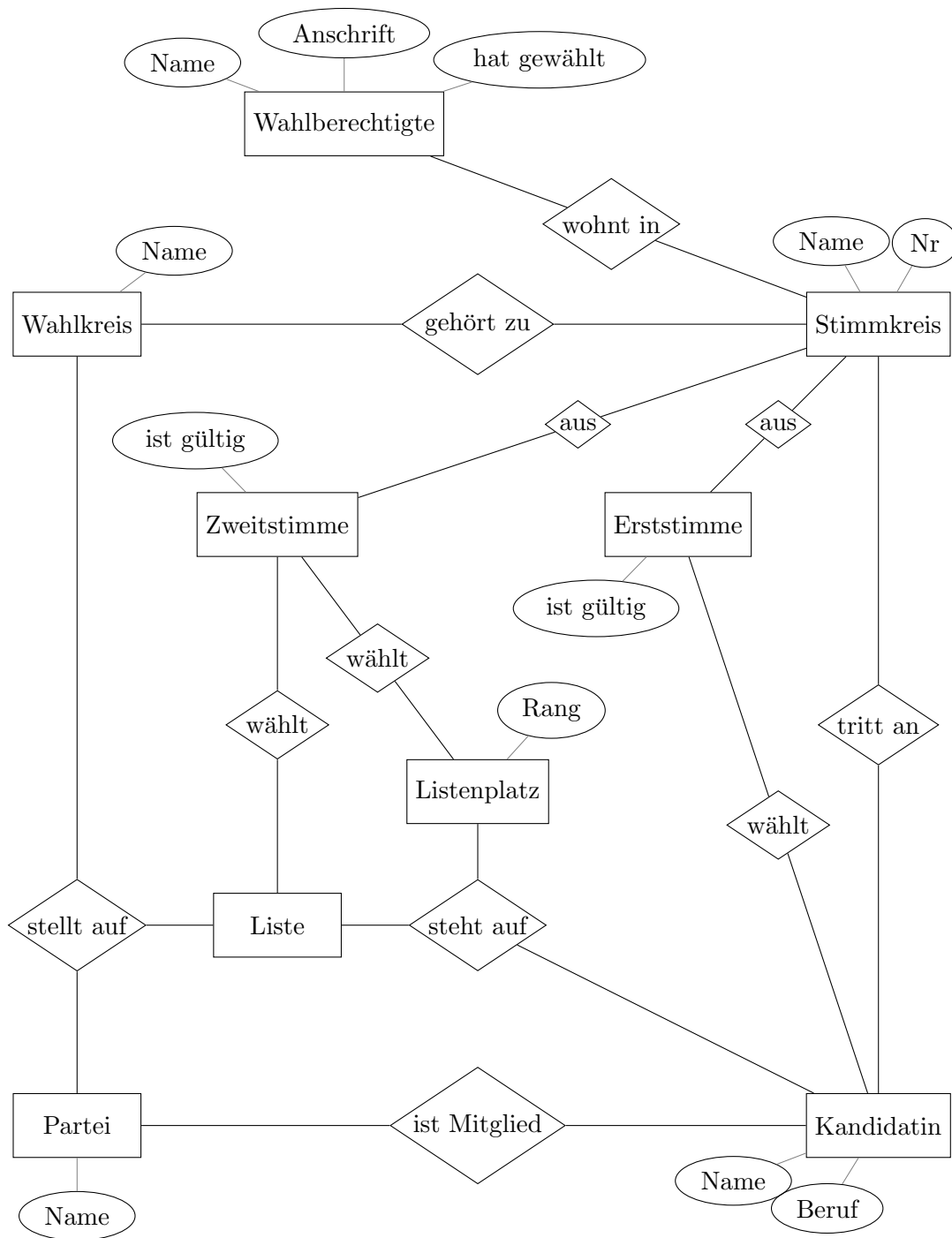


Abbildung 1: ER-Diagramm zur Landtagswahl

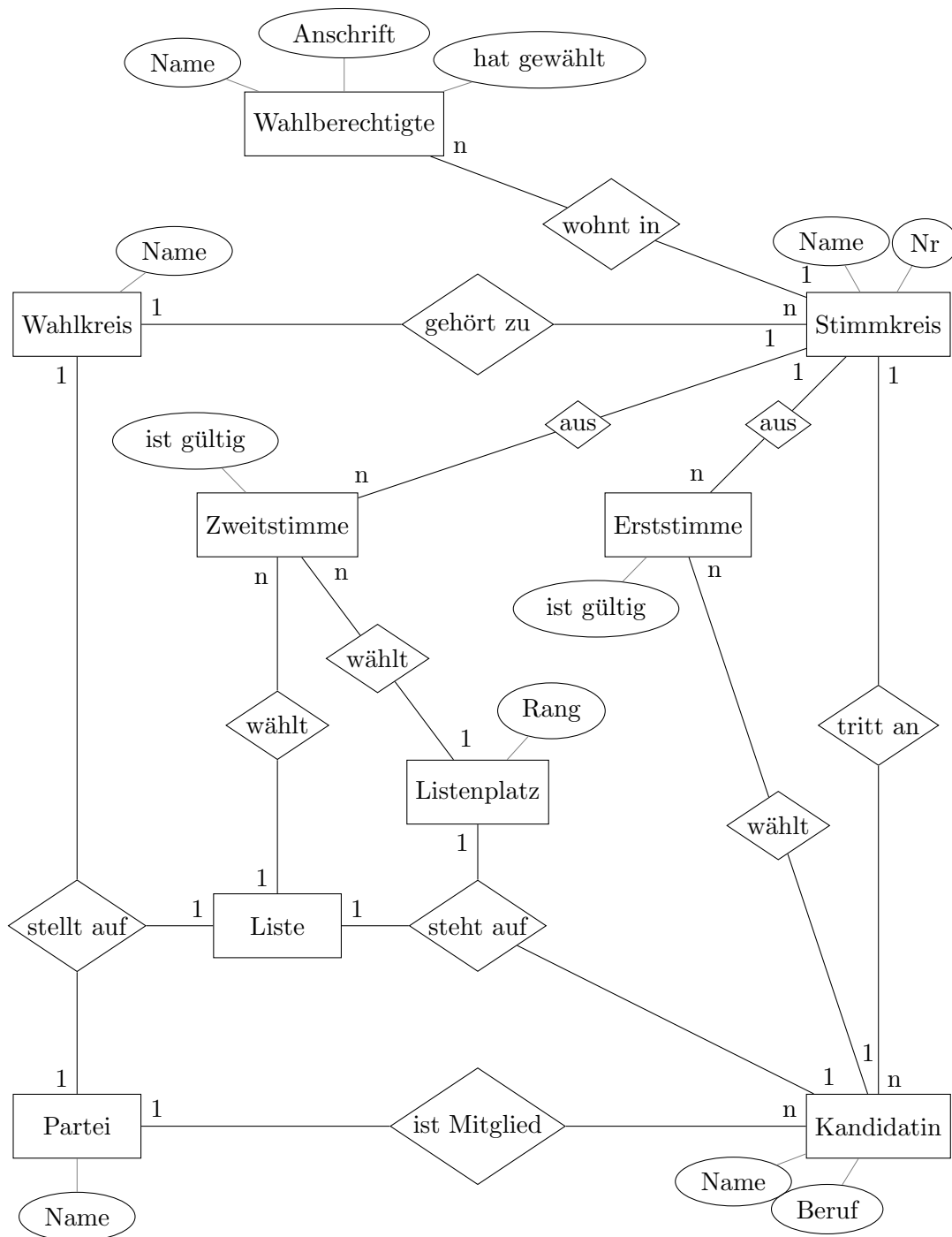


Abbildung 2: ER-Diagramm zur Landtagswahl mit Funktionalitätsangaben

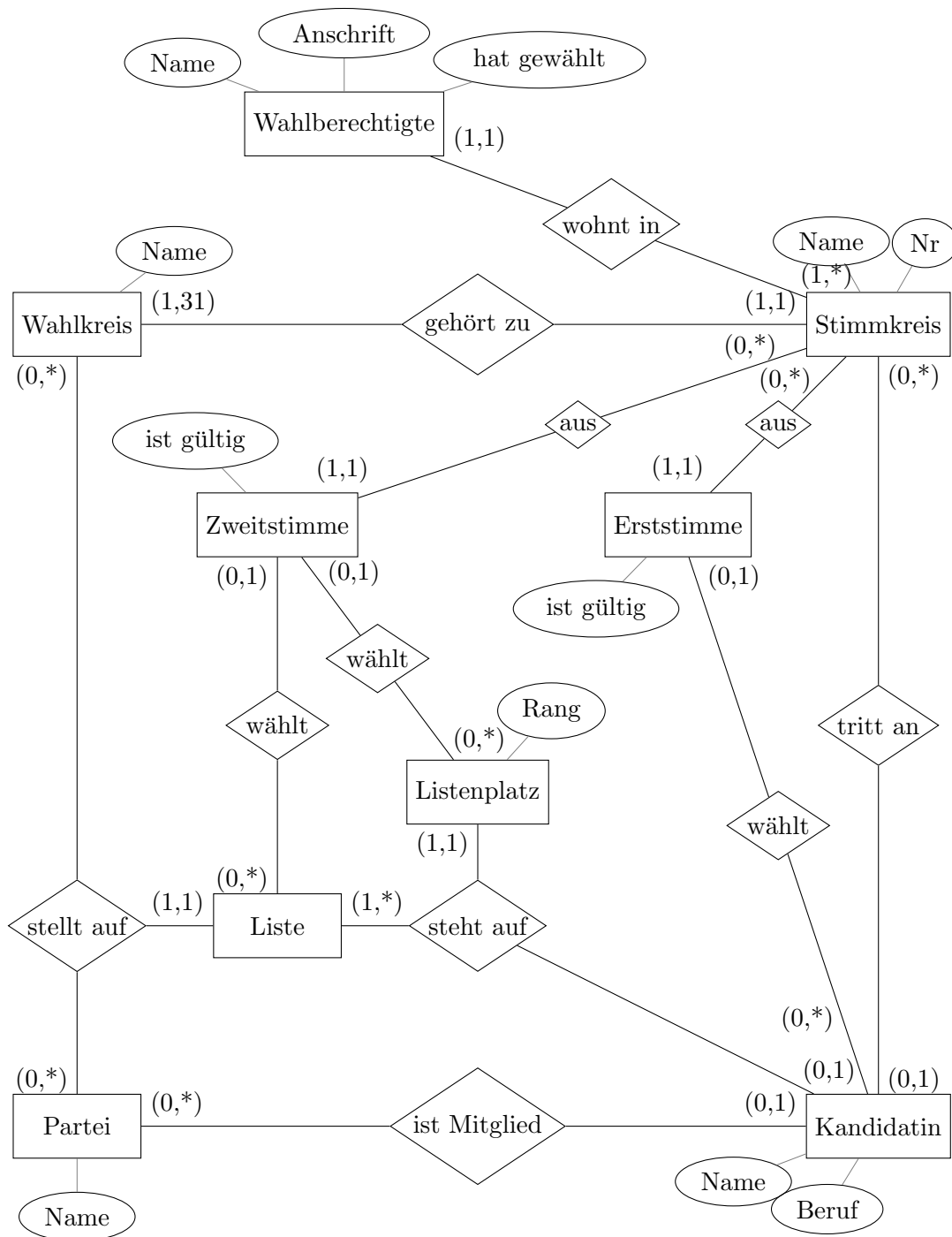


Abbildung 3: ER-Diagramm zur Landtagswahl mit (min,max)-Angaben

- es in der aktuellen Landtagswahl pro Wahlkreis maximal 31 Stimmkreise (in Oberbayern) gab,
 - eine Partei keine Listen aufstellen ((0, *) an „Partei“) - und ein Wahlkreis keine Liste haben ((0, *) an „Wahlkreis“) muss (jeweils in der ternären Beziehung „stellt auf“),
 - eine Kandidatin nicht Mitglied einer Partei sein muss, und
 - eine 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:
- Kandidaten selbst sind wahlberechtigt
 - Eine Kandidatin, die sich in ihrem eigenen Stimmkreis um das Direktmandat bewirbt darf in diesem nicht auf der Wahlkreisliste stehen (in anderen Stimmkreisen des Wahlkreises aber schon)
 - Man kann mit der Erststimme nur eine Kandidatin aus dem eigenen Stimmkreis wählen
 - Man kann mit der Erststimme nur eine Kandidatin wählen, die sich auf das Direktmandat bewirbt
 - Eine Kandidatin kann sich nur in ihrem Stimmkreis um das Direktmandat bewerben
 - Mit der Zweitstimme wählt man *entweder* einen Kandidaten auf einer Liste *oder* eine ganze Liste, aber nicht beides
 - Briefwahl
 - Hochrechnungen und Auszählungen dürfen erst nach Schließung der Wahllokale veröffentlicht werden. So wird sichergestellt, dass kein Wähler beeinflusst wird. Das System sollte daher garantieren, dass niemand vorher Zwischenergebnisse einsehen kann (auch keine Systemadministratoren oder Entwickler, da diese selbst wahlberechtigt sind).

Hasaufgabe 2

Unkontrollierte Redundanz ist unerwünscht. Können Sie sich eine sinnvolle Einsatzmöglichkeit für eine durch das DBMS kontrollierte Redundanz vorstellen?

Lösung:

Redundante Datenhaltung zur Performancesteigerung

Kontrollierte redundante Datenhaltung kann aus Performancegesichtspunkten sinnvoll sein. Als Beispiel betrachte man die Personenverwaltung einer Universität. So sind etwa Verwaltungsangestellte, Professoren, wissenschaftliche Mitarbeiter und Studenten Angehörige einer Universität, deren personenbezogene Daten in einem Datenbankverwaltungssystem abgelegt werden. Anfragen, die an die Datenbasis gestellt werden, können sich auf alle Personen beziehen (z.B. Einladung zu einer Universitätsfeier), auf bestimmte Personengruppen (z.B. Erstellen der Lohnabrechnung für alle Angestellten) oder auch nur selektiv auf Teilgruppen (z.B. Einladung zu einer Informationsveranstaltung für alle Studenten im 4. Semester). Die einzelnen Personengruppen sind aber unterschiedlich groß. So ist davon

auszugehen, dass es deutlich mehr Studenten als wissenschaftliche Mitarbeiter und mehr wissenschaftliche Mitarbeiter als Professoren gibt. Häufig auftretende Anfragen, die sich nur auf Professoren beziehen, sind dann ineffizient, wenn alle Personendaten zentral in einer Relation abgespeichert sind. In diesem Fall wäre das Überprüfen aller Universitätsangehörigen und das Auswählen der Professoren notwendig.

Speichert man in einer Relation all die Informationen, die sich auf alle Personen beziehen (Name, Anschrift, etc.), und in spezialisierten Relationen die Informationen für Studenten (Name, Anschrift, Semester, Studienrichtung), Professoren (Name, Anschrift, Rang, Raum) usw., so können beide Anfragetypen effizient ausgewertet werden: Es ist dann möglich, allgemeine Anfragen, die sich auf alle Universitätsangehörigen beziehen, wie auch selektive Anfragen, die sich nur auf Personengruppen beziehen, getrennt zu behandeln. Die redundante Datenspeicherung, in diesem Fall also das doppelte Ablegen allgemeiner personenbezogener Daten, wie Name und Anschrift, muss vom Datenbanksystem kontrolliert werden. Um einen konsistenten, d.h. stimmigen Dateninhalt zu gewährleisten, müssen Änderungen des Datenbestands (Einfügen, Löschen oder Modifikation einzelner Datensätze) sowohl in den spezialisierten Datensätzen, wie auch in dem globalen Datensatz der Personendaten durchgeführt werden. Dieses Vorgehen wird auch als Generalisierung bezeichnet.

Replizierte Datenhaltung für mobile Anwendungen

Replizierte Datenbanken spielen insbesondere auch bei mobilen Anwendungen eine wichtige Rolle. So werden Teile der globalen Datenbank auf den Laptops von Außendienstmitarbeitern repliziert und regelmäßig (z.B. abends nach den Kundengesprächen) synchronisiert.

Replizierte Datenhaltung bei verteilten Datenbanken

Redundante Datenhaltung ist mitunter auch bei verteilten Informationssystemen erforderlich. Für Anfragen, die von einem verteilten System verarbeitet werden, lässt sich eine höhere Performance ggf. dadurch erreichen, dass häufig abgefragte Daten, die zudem relativ statisch sind, d.h. selten modifiziert werden, an den Knoten im Netzwerk repliziert werden, an denen die Anfrageauswertung erfolgt. Das heißt, die Anfragebearbeitung erfolgt nahe bei den Daten und eine teure Übermittlung von Daten wird so gut es geht vermieden. Je nach Anwendungsfall muss dann natürlich die Konsistenz, d.h. Gültigkeit oder auch "Frische", der replizierten Daten durch das verteilte Datenbanksystem überwacht werden.

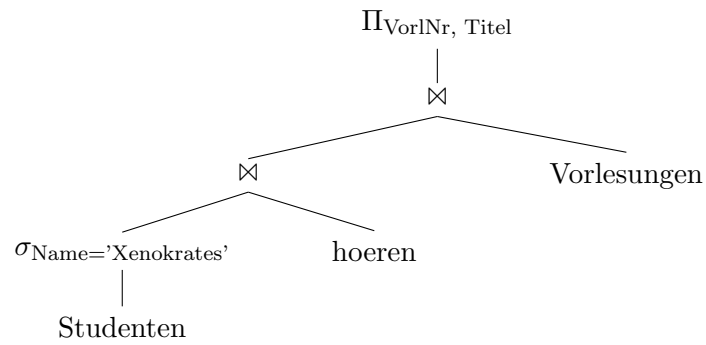
Hausaufgabe 3

Formulieren Sie die folgenden Anfragen auf dem bekannten Universitätsschema in Relationenalgebra. Geben Sie die Lösungen in der Operatorbaum-Darstellung an.

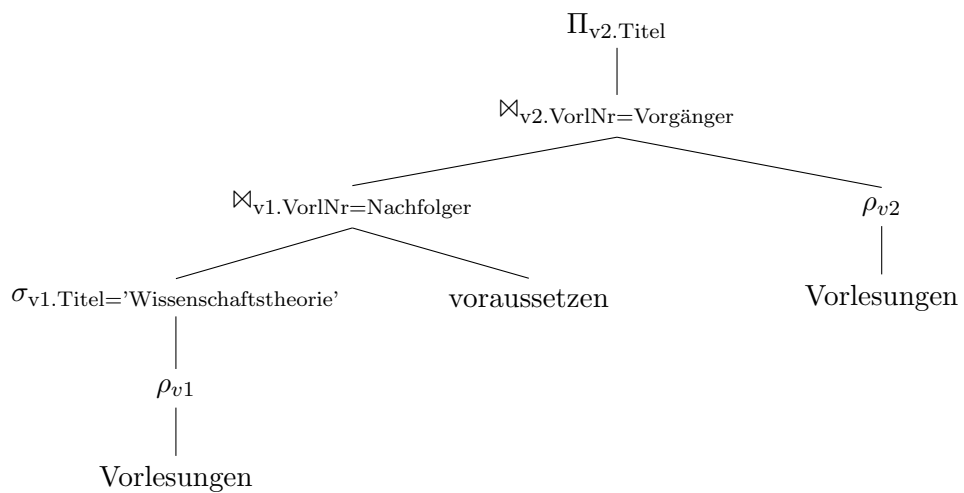
- a) Geben Sie alle *Vorlesungen* an, die der *Student* Xenokrates gehört hat.
- b) Geben Sie die Titel der direkten Voraussetzungen für die *Vorlesung* Wissenschaftstheorie an.
- c) Geben Sie Paare von *Studenten*(-Namen) an, die sich aus der *Vorlesung* Grundzüge kennen.

Lösung:

- a) Geben Sie alle *Vorlesungen* an, die der *Student* Xenokrates gehört hat.



b) Geben Sie die Titel der direkten Voraussetzungen für die *Vorlesung* Wissenschaftstheorie an.



c) Geben Sie Paare von *Studenten*(-Namen) an, die sich aus der *Vorlesung* Grundzüge kennen.

