



## Übung zur Vorlesung *Grundlagen: Datenbanken* im WS17/18

Harald Lang, Linnea Passing (gdb@in.tum.de)  
<http://www-db.in.tum.de/teaching/ws1718/grundlagen/>

### Blatt Nr. 03

Tool zum Üben der relationalen Algebra:  
<http://db.in.tum.de/people/sites/muehe/ira/>

#### Hausaufgabe 1

Beim konzeptuellen Entwurf hat man gewisse Freiheitsgrade hinsichtlich der Modellierung der realen Welt. Unter anderem hat man folgende Alternativen, die Sie an unserem Universitätsschema beispielhaft illustrieren sollten:

- Man kann ternäre Beziehungen in binäre Beziehungen transformieren.  
Betrachten Sie dazu die Beziehung *prüfen* und erläutern Sie die Vor- und Nachteile einer solchen Transformation.
- Man hat manchmal die Wahl, ein Konzept der realen Welt als Beziehung oder als Entitytyp zu modellieren. Erörtern Sie dies wiederum am Beispiel der Beziehung *priüfen* im Gegensatz zu einem eigenständigen Entitytyp *Prüfungen*.
- Ein Konzept der realen Welt kann manchmal als Entitytyp mit zugehörigem Beziehungstyp und manchmal als Attribut dargestellt werden. Ein Beispiel hierfür ist das Attribut *Raum* des Entitytyps *Professoren* im bekannten Uni Schema. Diskutieren Sie die Alternativen.

#### Lösung:

Ziel dieser Aufgabe ist es, alternative Entwürfe zu erstellen und bezüglich ihrer Anwendbarkeit zu analysieren. Unter Anwendbarkeit ist unter anderem zu verstehen, ob in der neuen Modellierung dieselben Informationseinheiten wie in der ursprünglichen abgebildet werden können, ob Konsistenzbedingungen eingehalten werden und ob die reale Welt in der modellierten Miniwelt sinnvoll wiedergegeben ist.

#### Erste Teilaufgabe: Transformation der ternären Beziehung in binäre Beziehungen

Abbildung 1 zeigt einen Ansatz, die ternäre Beziehung *prüfen* durch binäre Relationen auszudrücken. Durch die ursprüngliche Modellierung (links in der Abbildung) wird folgende Konsistenzbedingung ausgedrückt:

$$\text{prüfen} : \text{Studenten} \times \text{Vorlesungen} \rightarrow \text{Professoren} \quad (1)$$

Demgegenüber tritt bei der vorgeschlagenen Modellierung mittels binärer Relationen ein Semantikverlust auf. Durch die allgemeineren N:M-Beziehungen wird obige Konsistenzbedingung nicht mehr abgebildet. Somit ist das Modell der ternären Beziehung in diesem Fall ausdrucksstärker. Zwar lassen sich Prüfungsergebnisse in der alternativen Modellierung abbilden, allerdings geht Aussagekraft verloren. Abgebildet ist, dass Studenten über den Stoff von Vorlesungen geprüft werden, sowie dass Studenten von Professoren geprüft werden. Der Zusammenhang, welche Professoren welche Studenten in welchen Vorlesungen prüfen, ist aber nicht mehr ohne weiteres gegeben. Indirekt lösen lässt sich dies durch die

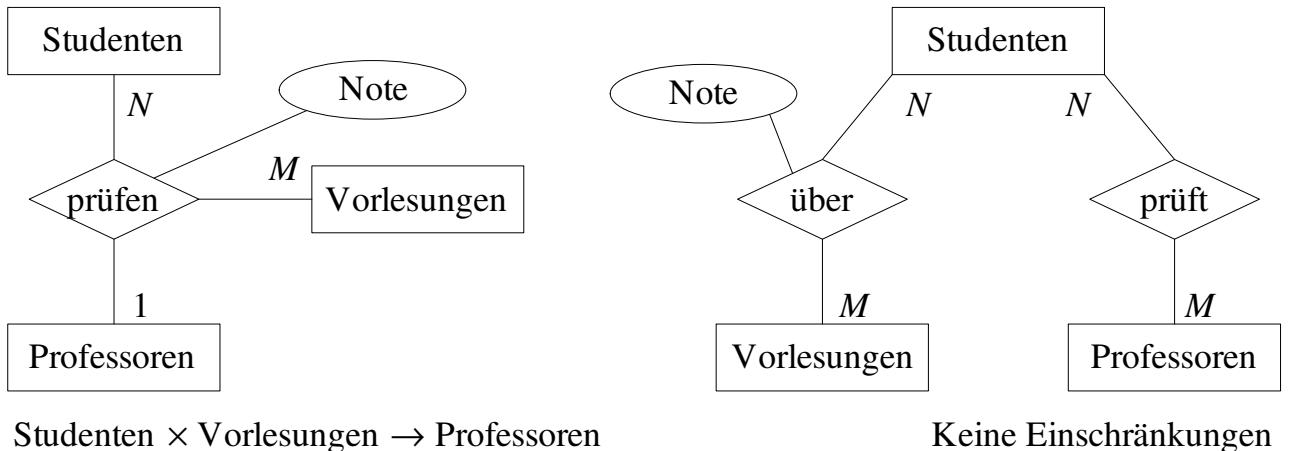


Abbildung 1: Auflösen der ternären Beziehung *prüfen* in binäre Beziehungen

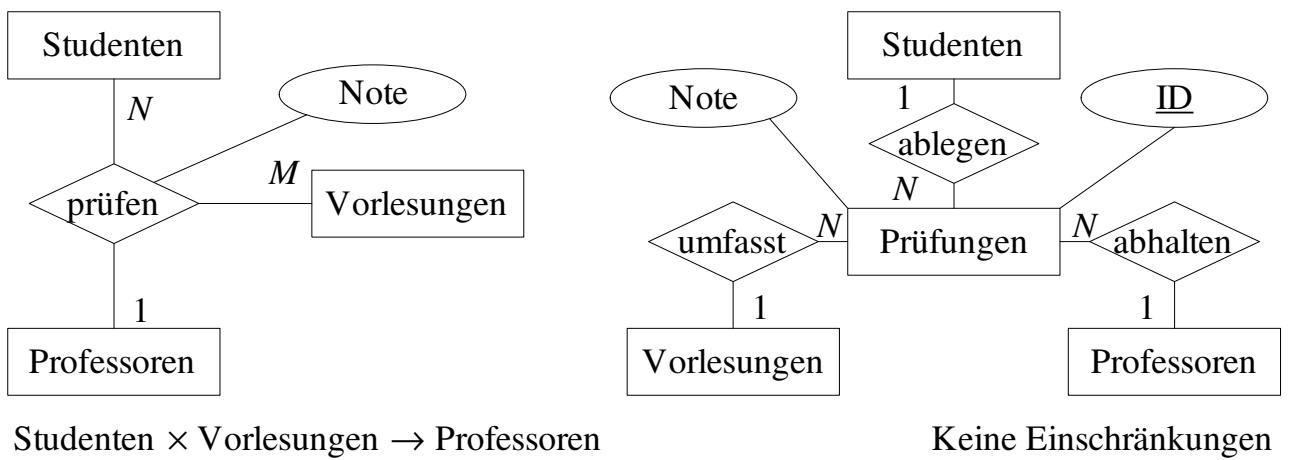


Abbildung 2: Modellierung von Prüfungen als Entitytyp

Aufnahme des zusätzlichen Attributs *Prüfungszeit* in die Relation *über* und auch in *prüft*. Da der zusätzlich aufgeführte Prüfungstermin eine Prüfung eindeutig festlegt, lässt sich die Information über eine Prüfung aus beiden Relationen erhalten. Allerdings muss für eine konsistente Extension sichergestellt werden, dass zu einem Eintrag in *über* auch ein passender Eintrag in *prüft* enthalten ist. Die gezeigte alternative Modellierung weist also klare Nachteile gegenüber der ursprünglichen ternären Beziehung auf.

Die alternative Modellierung einer ternären Beziehung durch mehrere binäre kann (abhängig von den zu modellierenden Anforderungen) im Allgemeinen folgende Nachteile aufweisen:

- Es tritt ein Semantikverlust auf.
- Es besteht die Möglichkeit, inkonsistente Datenbankzustände zu generieren. Gegebenenfalls ist eine Konsistenzüberprüfung der Datenbank erforderlich.
- Die reale Welt wird in der Miniwelt unzureichend wiedergegeben.

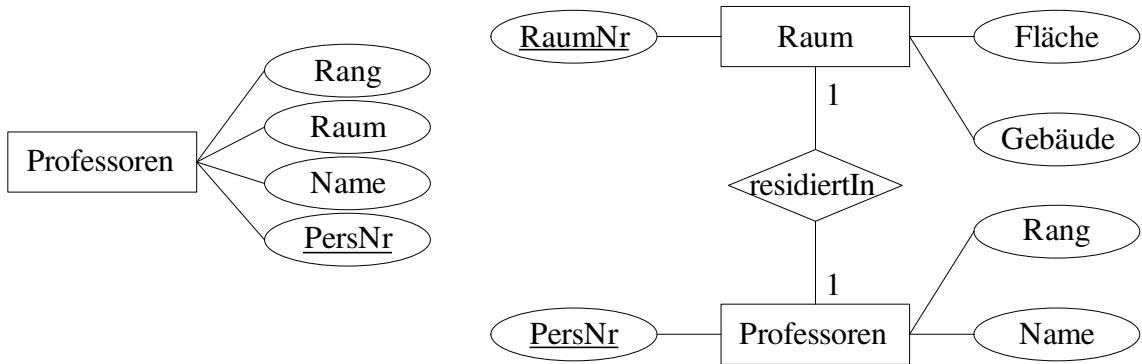


Abbildung 3: Zwei alternative Modellierungen um auszudrücken, dass Professoren ein Raum zugeordnet ist

### Zweite Teilaufgabe: Modellierung als Entitytyp anstelle einer Beziehung

Abbildung 2 zeigt eine alternative Modellierung der *prüfen*-Beziehung über einen Entitytyp *Prüfungen*. Auch in diesem Fall tritt erneut ein Semantikverlust auf. Es ist möglich, dass in der Modellierung mittels Entitytyp eine Prüfung existiert, zu der z.B. noch kein Prüfer feststeht, bzw. der Prüfer nicht mehr existiert. Möchte man dies in der Modellierung ausdrücken, müsste man zur  $(min, max)$ -Notation übergehen, mittels derer man fordern kann, dass eine Prüfung genau je einmal in den Relationen *ablegen*, *abhalten* und *umfasst* auftritt. Außerdem kann auch obige Konsistenzbedingung (1) nicht zugesichert werden. Zwar legt eine *Prüfungen*-Instanz über die angesprochenen Relationen den Studenten / die Studentin, die geprüfte Vorlesung und den / die prüfende(n) Professor(in) fest. Allerdings bestimmt die Kombination *Studenten*  $\times$  *Vorlesungen* nun nicht mehr *Professoren*. Denn es ist durch den Entwurf nicht ausgeschlossen, dass es beispielsweise zwei unterschiedliche Prüfungen gibt, die der Student Fichte im Fach Ethik ablegt. Nur einmal lässt er sich bei Professor Sokrates und ein andermal bei Professorin Curie darüber prüfen. Andererseits lassen sich manche Aspekte der Modellierung mittels Entity genauer erfassen, als dies in der ursprünglichen Modellierung der Fall ist. So ist in obigem Beispiel etwa spezifiziert, dass pro Prüfung genau eine Vorlesung geprüft wird.

### Dritte Teilaufgabe: Alternative Modellierungen über Attribute oder Beziehungstypen

Abbildung 3 zeigt zwei Modellierungsansätze, die ausdrücken, dass jedem Professor ein Raum zugewiesen ist. Links die Darstellung mittels Attribut, rechts über die Beziehung *residiertIn*. Generell ist eine Modellierung über eine Beziehung mit einem eigenständigen Entity (*Raum*) dann angebracht, wenn entsprechend detaillierte Informationen zu einem Raum nötig sind. Dies kann z.B. dann der Fall sein, wenn die Anwendungssicht der Abteilung Gebäudetechnik in das Modell integriert werden muss. Möchte man die Raumdaten für jeden Professor abfragen, dann zieht diese Modellierung in der Regel eine weniger effiziente Anfrageauswertung nach sich.

## Hausaufgabe 2

Formulieren Sie folgende Anfragen auf dem bekannten Universitätsschema in der relationalen Algebra:

- Finden Sie die *Vorlesungen*, die keine Hörer haben.

- b) Finden Sie die *Studenten*, die alle *Vorlesungen* hören.

**Lösung:**

- a) Finden Sie die *Vorlesungen*, die keine Hörer haben.

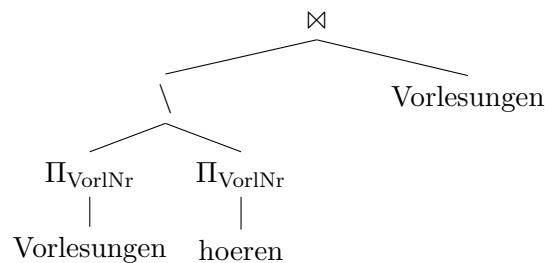
### Formulierung in relationaler Algebra

$$(\Pi_{VorlNr}(\text{Vorlesungen}) \setminus \Pi_{VorlNr}(\text{hören})) \bowtie \text{Vorlesungen}$$

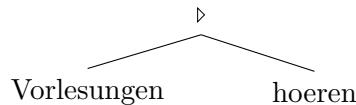
oder

$$\text{Vorlesungen} \triangleright \text{hören}$$

In Operatorbaumdarstellung:



oder

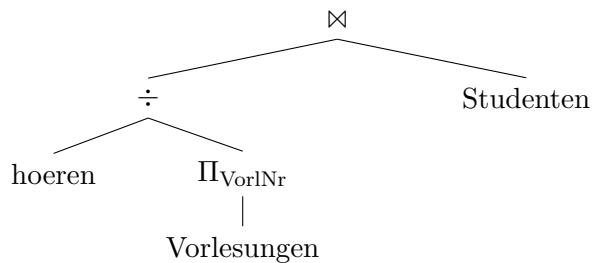


- b) Finden Sie die *Studenten*, die alle *Vorlesungen* hören.

### Formulierung in relationaler Algebra

$$(\text{hoeren} \div \Pi_{VorlNr}(\text{Vorlesungen})) \bowtie \text{Studenten}$$

In Operatorbaumdarstellung:



### Gruppenaufgabe 1 (nicht Zuhause vorbereiten)

Gegeben sei die ER-Modellierung von Zugverbindungen in Abbildung 4. Beachten Sie: **verbindet** modelliert ein **Teilstück** einer Verbindung, d.h. auf der Strecke München → Hamburg gibt es einen Eintrag für die Teilstrecke von München nach Nürnberg, einen Eintrag für Nürnberg nach Würzburg, einen Eintrag für die Teilstrecke Würzburg nach Göttingen und einen Eintrag von Göttingen nach Hamburg.

- Fügen Sie bei den Beziehungen Funktionalitätsangaben hinzu.
- Übertragen Sie das ER-Modell in ein relationales Schema.
- Verfeinern Sie das relationale Schema soweit möglich durch Eliminierung von Relationen.

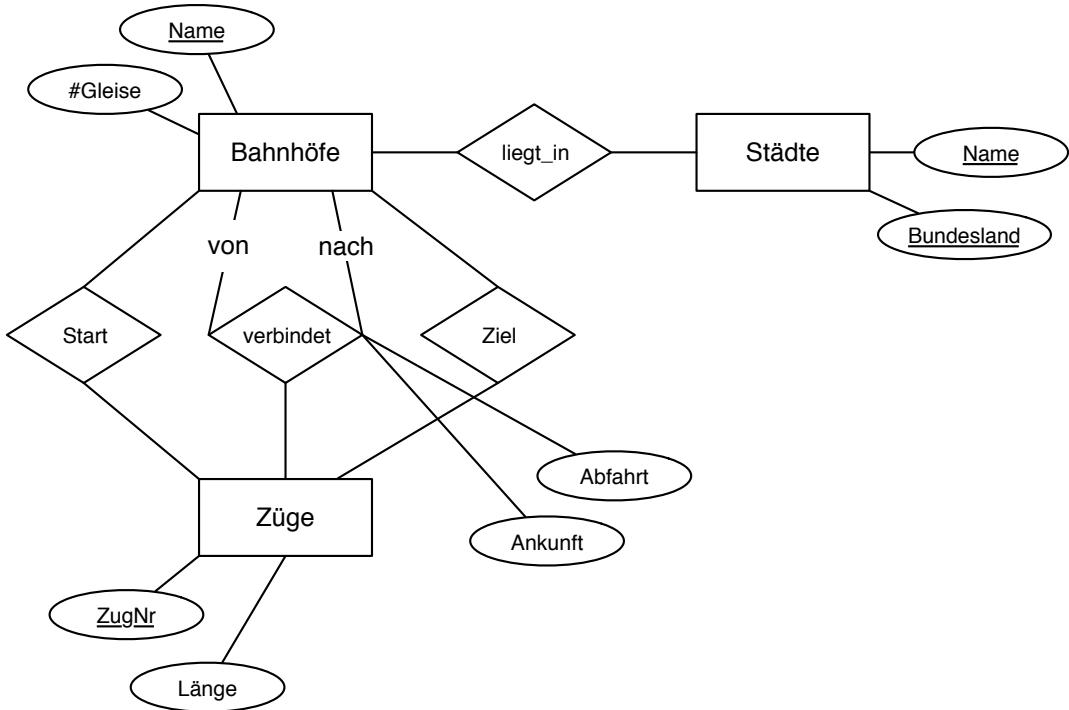


Abbildung 4: ER-Modellierung von Zugverbindungen

**Lösung:**

**a) Eintragen der Multiplizitäten**

Abbildung 5 zeigt das ER-Diagramm mit eingetragenen Funktionalitätsangaben.

**b) Erstellen des relationalen Schemas**

Die initiale Überführung ergibt folgende Relationen für die Entitytypen:

$$\text{Städte} : \{[\underline{\text{Name}} : \text{string}, \underline{\text{Bundesland}} : \text{string}]\} \quad (2)$$

$$\text{Bahnhöfe} : \{[\underline{\text{Name}} : \text{string}, \underline{\#Gleise} : \text{integer}]\} \quad (3)$$

$$\text{Züge} : \{[\underline{\text{ZugNr}} : \text{integer}, \underline{\text{Länge}} : \text{integer}]\} \quad (4)$$

Für die Beziehungstypen werden folgende Relationen erstellt:

$$\text{liegt_in} : \{[\underline{\text{BName}} : \text{string}, \underline{\text{SName}} : \text{string}, \underline{\text{Bundesland}} : \text{string}]\} \quad (5)$$

$$\text{Start} : \{[\underline{\text{ZugNr}} : \text{integer}, \underline{\text{BName}} : \text{string}]\} \quad (6)$$

$$\text{Ziel} : \{[\underline{\text{ZugNr}} : \text{integer}, \underline{\text{BName}} : \text{string}]\} \quad (7)$$

$$\text{verb bindet} : \{[\underline{\text{VonBahnhof}} : \text{string}, \underline{\text{NachBahnhof}} : \text{string}, \underline{\text{ZugNr}} : \text{integer}, \underline{\text{Abfahrt}} : \text{date}, \underline{\text{Ankunft}} : \text{date}]\} \quad (8)$$

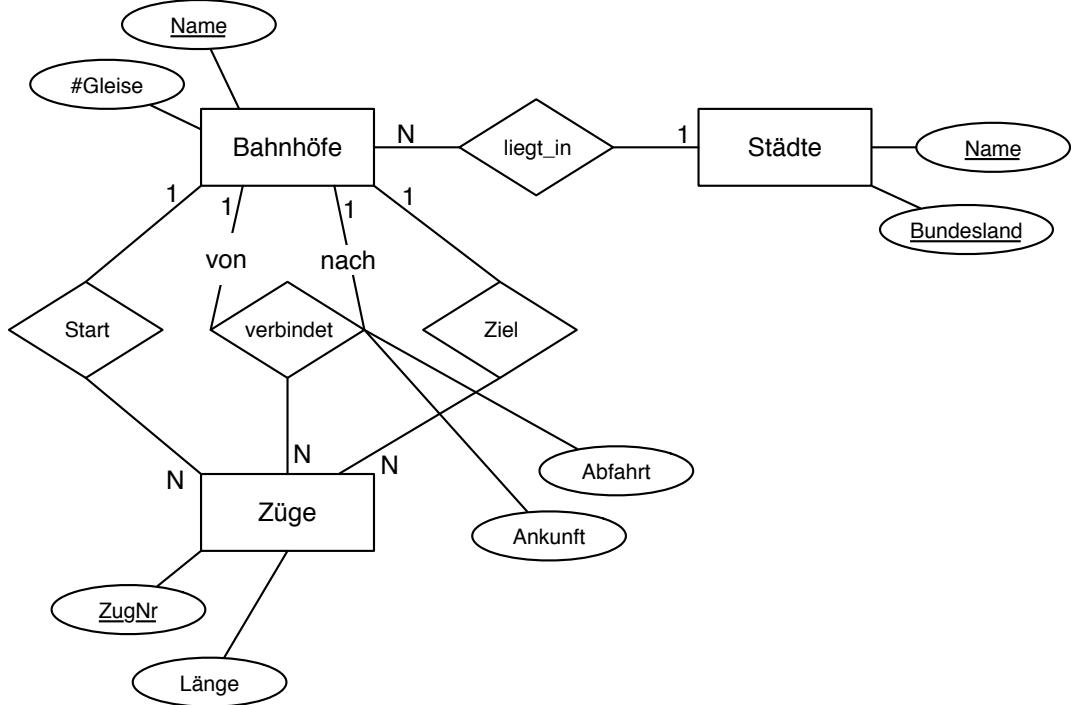


Abbildung 5: ER-Modellierung von Zugverbindungen mit Angabe der Funktionalitäten

### c) Verfeinerung des relationalen Schemas

Als Nächstes wird das relationale Schema verfeinert, indem Relationen zusammengefasst werden.

Dabei werden Relationen für binäre Beziehungstypen mit Relationen für Entitytypen zusammengefasst, falls diese gleiche Schlüssel besitzen und es sich dabei um 1:N, N:1 oder 1:1 Beziehungen handelt.

So kann Relation (5) in (3) aufgenommen werden. (6) wird mit (4) zusammengefasst. Auch die Ziel-Relation (7) wird mit der *Züge*-Relation (4) zusammengefasst, d.h.

$$(5) \mapsto (3), (6) \mapsto (4), (7) \mapsto (4)$$

Damit ergibt sich folgendes Schema:

```

Städte : {[Name : string, Bundesland : string]}
Bahnhöfe : {[Name : string, #Gleise : integer,
             SName : string, Bundesland : string]}
Züge : {[ZugNr : integer, Länge : integer,
          StartBahnhof : string, ZielBahnhof : string]}
verbindet : {[VonBahnhof : string, NachBahnhof : string,
              ZugNr : integer, Abfahrt : date, Ankunft : date]}

```

Im vorliegenden Fall ist die Zugnummer eindeutig für eine Verbindung. Ein ICE, der die Städte München (*StartBahnhof*) und Berlin (*ZielBahnhof*) verbindet, hat somit eine eindeutige Zugnummer für diese Verbindung, die über mehrere Zwischenbahnhöfe erfolgen

kann. Fährt der Zug zurück, erhält er eine andere Nummer zugewiesen. Dadurch sind die Kombinationen (*ZugNr*, *VonBahnhof*) und (*ZugNr*, *NachBahnhof*) zwei mögliche Schlüssel für die Relation *verbindet*.

### Gruppenaufgabe 2 (nicht Zuhause vorbereiten)

iBike ist ein neuer Anbieter von free-floating bike sharing in München, d.h. die Kunden können die Fahrräder an beliebigen Orten im Stadtgebiet ausleihen und auch wieder abstellen.

Modellieren Sie das iBike-System, das *Kunden*, *Fahrräder* und *individuelle Fahrten* (von wann bis wann, von wo nach wo, Preis) verwaltet, als ER-Diagramm. Tragen Sie min-max- und Funktionalitätsangaben ein.

Stellen Sie die folgenden Anfragen an das iBike-System in relationaler Algebra:

- a) Welche Fahrräder wurden noch nie gefahren?
- b) Was ist der letzte bekannte Standort vom Fahrrad mit der Kennung '123'?
- c) Geben Sie eine Liste der Fahrräder aus, mit denen der Kunde 'Alfons Kemper' schon gefahren ist.
- d) Geben Sie eine Liste der Fahrräder aus, mit denen der Kunde 'Alfons Kemper' **noch nicht** gefahren ist.
- e) Wer ist schon mit **allen** Fahrrädern gefahren?
- f) Am Standort '48.1718164,11.5510174' wurden sehr viele Fahrräder abgestellt (das ist mitten im Olympiasee). iBike möchte wissen, ob jemand mehr als ein Fahrrad dort abgestellt hat. Dann ist dieser Kunde vermutlich ein Vandale.

### Lösung:

Die Anfragen unterscheiden sich je nach Modellierung, daher hier nur allgemeine Hinweise:

- a) Antisemijoin zwischen der Fahrräder- und der Fahrtenrelation
- b) entweder MAX-Aggregation oder Antijoin mit der Joinbedingung 'ZeitpunktA größer ZeitpunktB'
- c) Selektion auf Kundenrelation, Semijoin mit Fahrtenrelation und ggf. (wenn man noch mehr Daten als die Fahrrad-Kennung ausgeben will) Semijoin mit Fahrräderrelation
- d) entweder Mengenoperation (alle Fahrräder minus die in Teilaufgabe c) ausgegebenen) oder Antisemijoin Kunde–Fahrten und anschließend ggf. Semijoin mit Fahrräderrelation
- e) relationale Division (Erklärung folgt in der nächsten Vorlesung, bei Wikipedia, ...)
- f) Zuerst Liste der Fahrräder mit letztem Standort im Olympiasee finden (siehe Teilaufgabe b)). Dann mit Kunden joinen. Dann Selfjoin mit Bedingung 'gleicher Kunde aber unterschiedliches Fahrrad'. Die Kunden aus diesem Ergebnis sind die potentiellen Vandalen.