

Übung zur Vorlesung *Einführung in die Informatik 2 für Ingenieure (MSE)*

Alexander van Renen (renen@in.tum.de)

<http://db.in.tum.de/teaching/ss17/ei2/>

Lösungen zu Blatt 8

Tool zum Üben der relationalen Algebra: <http://www-db.in.tum.de/~muehe/ira/>.

SQL-Schnittstelle: <http://hyper-db.com/interface.html>.

Aufgabe 1: Relationale Modellierung

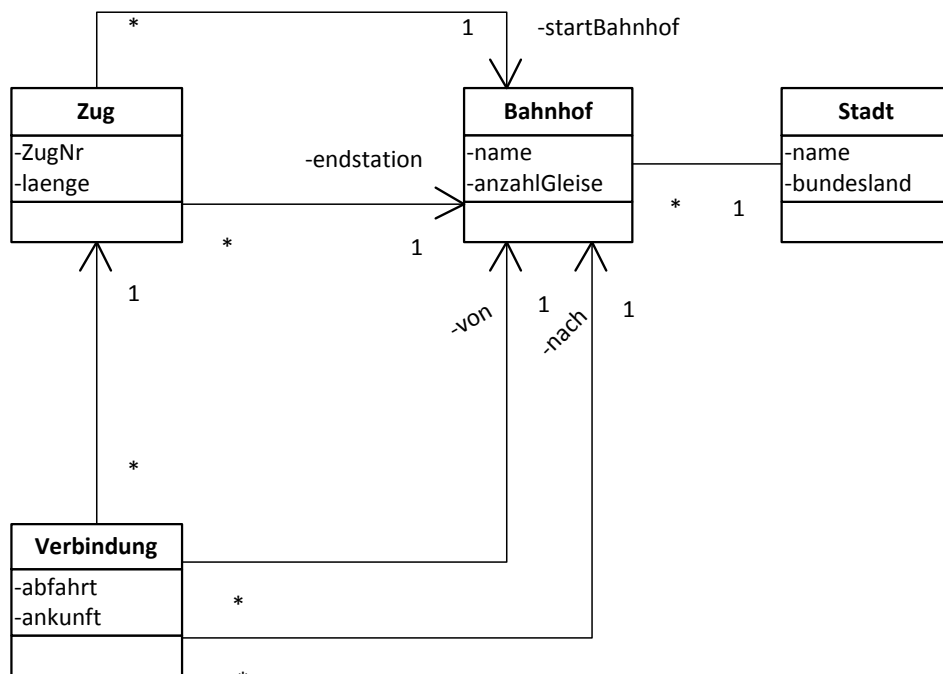


Abbildung 1: UML Modell von Zugverbindungen

Setzen Sie das UML-Modell von Zugverbindungen aus Abbildung 1 in ein relationales Modell um. Wandeln Sie dazu zunächst die Klassen mit ihren Attributen in Relationen um. Anschließend beachten Sie noch die Beziehungen, die Sie entweder mit zusätzlichen Attributen (Spalten) in den bestehenden Relationen umsetzen können oder aber mit zusätzlichen Relationen. Zuletzt überlegen Sie sich, welche Attribute jeweils einen Primärschlüssel für die Relationen darstellen und unterstreichen diese.

Lösung 1

Zuerst modellieren wir die Klassen als Relationen und bestimmen passende Schlüssel. Im Fall der Verbindung müssen wir einen künstlichen Schlüssel einführen, da nicht einmal die Kombination aus Abfahrt und Ankunft eindeutig ist:

- Züge** : $\{\{\underline{\text{ZugNr: integer}}, \text{Laenge: integer}\}\}$ (1)
- Bahnhöfe** : $\{\{\underline{\text{Name: string}}, \text{AnzahlGleise: integer}\}\}$ (2)
- Städte** : $\{\{\underline{\text{Name: string}}, \text{Bundesland: string}\}\}$ (3)
- Verbindung** : $\{\{\underline{\text{VerbindungsNr: integer}}, \text{Abfahrt: date}, \text{Ankunft: date}\}\}$ (4)

Nun kümmern wir uns um die Beziehungen, die wir in einem ersten Schritt alle als eigenständige Relationen modellieren:

- liegtIn** : $\{\{\underline{\text{Bahnhof: string}}, \text{Stadt: string}, \text{Bundesland: string}\}\}$ (5)
- startetIn** : $\{\{\underline{\text{ZugNr: integer}}, \text{Bahnhof: string}\}\}$ (6)
- endetIn** : $\{\{\underline{\text{ZugNr: integer}}, \text{Bahnhof: string}\}\}$ (7)
- verbindetVon** : $\{\{\underline{\text{VerbindungsNr: integer}}, \text{Bahnhof: string}\}\}$ (8)
- verbindetNach** : $\{\{\underline{\text{VerbindungsNr: integer}}, \text{Bahnhof: string}\}\}$ (9)
- verbindetMit** : $\{\{\underline{\text{VerbindungsNr: integer}}, \text{Zug: integer}\}\}$ (10)

Als letztes verfeinern wir das relationale Schema, indem wir Relationen zusammenfassen. Dabei werden Relationen für Beziehungen mit Relationen für Klassen zusammengefasst, falls diese den gleichen Schlüssel haben und es sich dabei um eine 1:N, N:1 oder 1:1 Beziehung handelt.

So kann Relation (5) in (2) aufgenommen werden. (6) und (7) werden mit (1) zusammengefasst. Und (8), (9) und (10) werden alle mit (4) zusammengefasst. Das ergibt folgendes Schema:

- Züge** : $\{\{\underline{\text{ZugNr: integer}}, \text{Laenge: integer}, \text{StartBahnhof: string}, \text{ZielBahnhof: string}\}\}$
- Bahnhöfe** : $\{\{\underline{\text{Name: string}}, \text{AnzahlGleise: integer}, \text{Stadt: string}, \text{Bundesland: string}\}\}$
- Städte** : $\{\{\underline{\text{Name: string}}, \text{Bundesland: string}\}\}$
- Verbindung** : $\{\{\underline{\text{VerbindungsNr: integer}}, \text{Abfahrt: date}, \text{Ankunft: date}, \text{VonBahnhof: string}, \text{NachBahnhof: string}, \text{ZugNr: integer}\}\}$

Aufgabe 2: Relationenalgebra

Formulieren Sie die folgenden Anfragen auf dem Universitätsschema in Relationenalgebra.

Lösung 2

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

$$R := \Pi_{\text{VorlNr}, \text{Titel}}(\text{Vorlesungen} \bowtie (\text{hören} \bowtie (\sigma_{\text{Name}='Xenokrates'}(\text{Studenten}))))$$

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

$$R := \Pi_{\text{v2.Titel}}(\rho_{\text{v2}}(\text{Vorlesungen}) \bowtie_{\text{v2.VorlNr=Vorgänger}} (\text{voraussetzen} \bowtie_{\text{v1.VorlNr=Nachfolger}} (\sigma_{\text{v1.Titel}='Wissenschaftstheorie'}(\rho_{\text{v1}}(\text{Vorlesungen}))))))$$

Aufgabe 3: SQL

Formulieren Sie folgende Anfragen auf dem Universitätsschema in SQL.

Lösung 3

- (a) Finden Sie die *Studenten*, die Sokrates aus *Vorlesung(en)* kennen.

```
select s.Name, s.MatrNr
from Studenten s, hoeren h, Vorlesungen v, Professoren p
where s.MatrNr = h.MatrNr
and h.VorlNr = v.VorlNr
and v.gelesenVon = p.PersNr
and p.Name = 'Sokrates';
```

DISTINCT wäre nett, um Duplikate zu unterdrücken ist aber nicht explizit in der Aufgabe gefordert.

- (b) Finden Sie die *Assistenten* von *Professoren*, die den *Studenten* Fichte unterrichtet haben – z.B. als potentielle Betreuer seiner Diplomarbeit.

```
select a.Name, a.PersNr
from Assistenten a, Vorlesungen v, hoeren h, Studenten s
where a.Boss = v.gelesenVon
and v.VorlNr = h.VorlNr
and h.MatrNr = s.MatrNr
and s.Name = 'Fichte';
```

- (c) Geben Sie die Namen der *Professoren* an, die Xenokrates aus *Vorlesungen* kennt.

```
select p.PersNr, p.Name
from Professoren p, hoeren h, Vorlesungen v, Studenten s
where p.PersNr = v.gelesenVon
and v.VorlNr = h.VorlNr
and h.MatrNr = s.MatrNr
and s.Name = 'Xenokrates';
```

- (d) Welche *Vorlesungen* werden von *Studenten* im Grundstudium (1.-4. Semester) gehört? Geben Sie die Titel dieser *Vorlesungen* an.

```
select v.Titel
from Vorlesungen v, hoeren h, Studenten s
where v.VorlNr = h.VorlNr
and h.MatrNr = s.MatrNr
and s.Semester between 1 and 4;
```