

Prof. Dr. Agn es Voisard, Nicolas Lehmann

Datenbanksysteme, SoSe 2017

 bungsblatt 2

Tutor: Nicolas Lehmann

Tutorium 10

Boyan Hristov, Julian Habib

18. Mai 2017

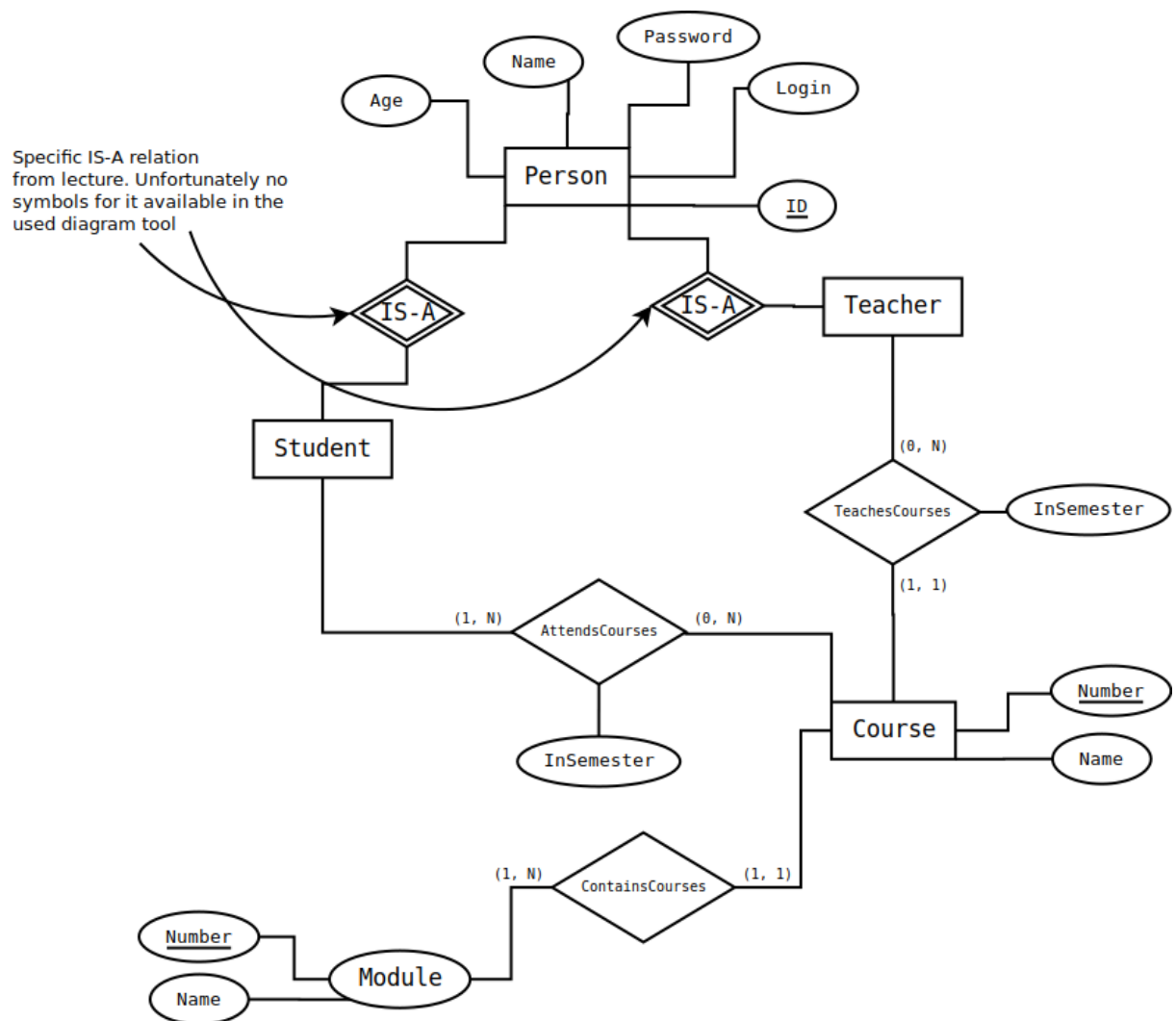
Link zum Git Repository: <https://github.com/BoyanH/Freie-Universitaet-Berlin/tree/master/Datenbanksysteme/Solutions/homework3>

1 Aufgabe

a) Relationales Modell

```
Person(ID::integer, Age::integer, Name::character varying(20), Password::character varying(40),  
      Login::character varying(40))  
Teacher(ID::integer)  
Student(ID::integer)  
Course(Number::integer, Name :: character varying(50))  
Module(Number::integer, Name :: character varying(50))  
  
PersonIsATeacher(PersonID, TeacherID)  
PersonIsAStudent(PersonID, StudentID)  
TeachesCourses(PersonID, CourseNumber, InSemester)  
ContainsCourses(ModuleNumber, CourseNumber)  
AttendsCourses(StudentID, CourseNumber)
```

ER Diagramm in umgekehrter min-max Chen Notation



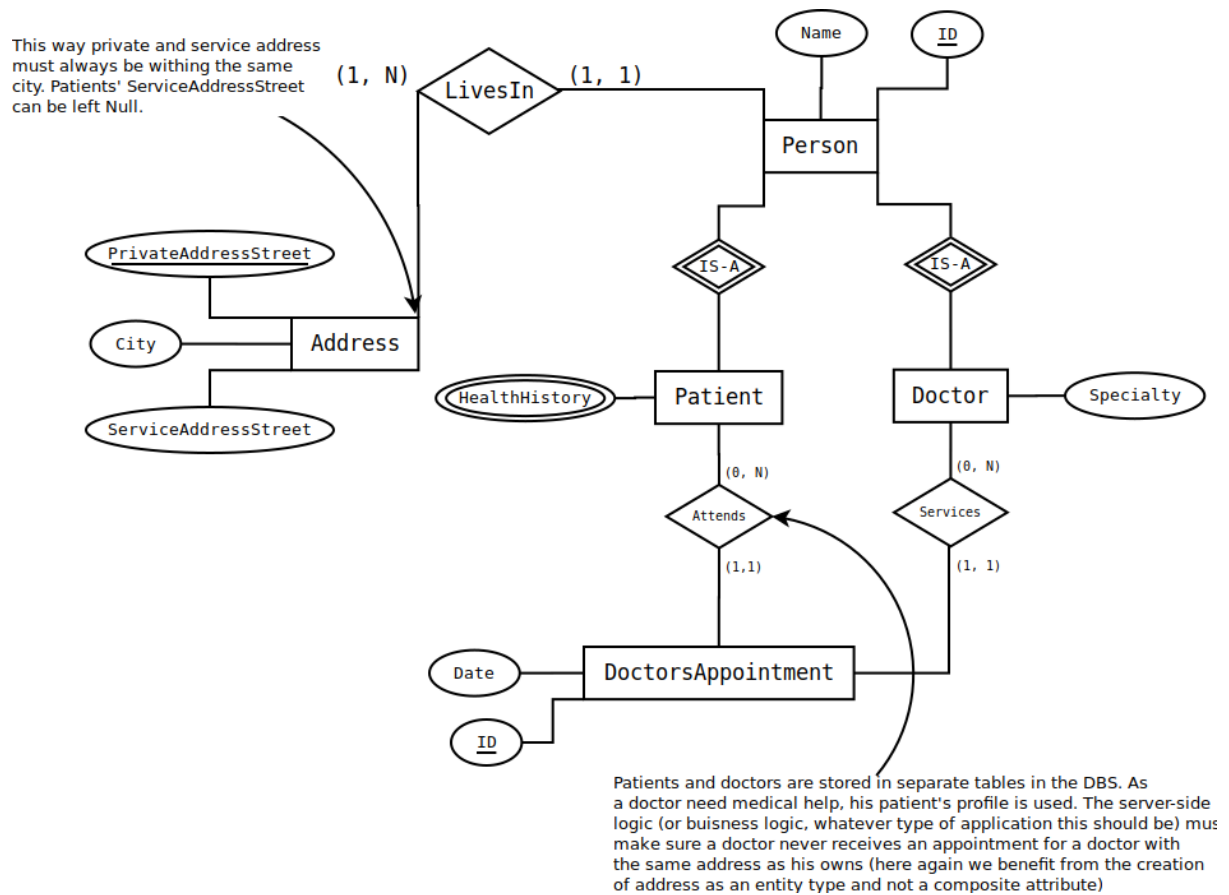
b) Relationales Modell

```

Person(ID::integer, Name::character varying(20))
Doctor(ID::integer, Specialty::character varying(30), Name::character varying(20))
Patient(ID::integer, HealthHistory::character varying(40) ARRAY, Name::character varying(20))
DoctorsAppointment(ID::integer, Date::date)
Address(PrivateAddressStreet::character varying(20), City::character varying(20),
        ServiceAddressStreet::character varying(20))

LivesIn(PersonID, AddressPrivateAdressStreet)
Attends(PersonID, DoctorsAppointmentID)
Services(PersonID, DoctorsAppointmentID)
  
```

ER Diagramm in umgekehrter min-max Chen Notation



2 Aufgabe

- $\Pi_{\text{Vorname, Nachname}} (\sigma_{\text{Alter} < 30})$
- $\Pi_{\text{Datum}} (\sigma_{\text{Temperatur} > \text{Regenmenge} \vee \text{Temperatur} > \text{Sonnenscheindauer})$
- $\Pi_{\text{Kreditkartennummer}} (\sigma_{\text{Name} = \text{'Emirates'} \wedge \text{Datum} \geq \text{'02.03.2016'} \wedge \text{Datum} \leq \text{'07.06.2016'}} (\text{Passagier} \bowtie_{\text{ID} = \text{Passagier-ID}} \text{Flug} \bowtie_{\text{Fluggesellschaft-ID} = \text{ID}} \text{Fluggesellschaft}))$
- $\Pi_{\text{Name}} (\sigma_{\text{Temperatur} < 0} (\text{Wetter} \bowtie_{\text{Wetter::Datum} = \text{Flug::Datum}} \text{Flug} \bowtie_{\text{Fluggesellschaft-ID} = \text{ID}} \text{Fluggesellschaft}))$
- $\Pi_{\text{Vorname, Nachname}} (\sigma_{\neg(\text{Temperatur} < 20 \wedge \text{Regenmenge} > 10 \wedge \text{Sonnenscheindauer} < 6)} (\text{Wetter} \bowtie_{\text{Wetter::Datum} = \text{Flug::Datum}} \text{Flug} \bowtie_{\text{Passagier-ID} = \text{ID}} \text{Passagier}))$

3 Aufgabe

Um die Aufgabe richtig zu lösen, muss man Attribute (Spalten) umbenennen können. Sonst kann man nicht äußern, dass z.B. Nachname und Datum gleich sein müssen, ID aber nicht. Nach der Annahme, dass wir die Operation zum Umbenennen benutzen können, kann man die Folgende aufgaben so lösen (Diese Operation ist offiziell in der relationalen Algebra enthalten).

Wir haben die einzelne Teile in separate Queries geteilt, damit man besser verstehen kann was genau passiert.

- a) $q1 = \sigma_{textAlter > 30} (Passagier \bowtie_{ID = Passagier-ID} Flug)$
 $q2 = \Pi_{Vorname, Nachname, Datum, Fluggesellschaft-ID} (q1)$
 $q3 = q2 \bowtie (\delta \text{ 'Vorname' } \rightarrow \text{ 'Vorname-duplicate' } q2)$
 $q4 = q3 - \sigma_{Vorname = Vorname-duplicate} (q3)$
 $q5 = \Pi_{Vorname, Nachname} (q4)$

$q1$ nimmt alle Passagiere, die über 30 Jahre alt sind und die dazugehörige Flüge. $q2$ nimmt nur die Attributen Vorname, Nachname, Datum und Fluggesellschaft-ID davon. Dies wird gemacht, damit dann bei $q3$ alle mögliche Kombinationen von Passagieren, die im selben Flug geflogen sind und die selbe Nachname haben (wir haben angenommen, das ein Flug durch Datum und Fluggesellschaft-ID eindeutig charakterisiert wird) erzeugt werden. Dann sorgt $q4$ dass Entitäten mit gleiche Vorname und Vorname-duplicate gelöscht werden. Fall mindestens 2 Personen mit dem selben Nachname gibt, dann gibt es mindestens 4 Entitäten, die originelle und diese mit vertauschte Vorname und Vorname-duplicate. Nur diese mit vertauschte (unterschiedliche) Vorname und Vorname-duplicate werden im $q4$ selektiert. Danach wird in $q5$ die Vorname und Nachname dieser Entitäten selektiert.

- b) $q1 = Passagier \bowtie_{ID = Passagier-ID} Flug$
 $q2 = \Pi_{Alter, Datum, Fluggesellschaft-ID} (q1)$
 $q3 = q2 \bowtie (\delta \text{ 'Alter' } \rightarrow \text{ 'Alter-duplicate' } q2)$
 $q4 = q2 - \sigma_{Alter = Alter-duplicate} (q2)$
 $q5 = \Pi_{Datum, Fluggesellschaft-ID} (q4)$
 $q6 = q1 - q5$
 $q7 = \Pi_{Datum, Fluggesellschaftsname} (q6 \bowtie_{FluggesellschaftID = ID} Fluggesellschaft)$

Nach der selben Logik haben wir in $q4$ Alle Flüge, auf dem mindestens zwei Personen das selbe Alter haben. In $q5$ nehmen wir nur Datum und Fluggesellschaft-ID (Das Superkey einer Flug, nach Annahme) damit wir in $q6$ alle Flüge bekommen können, die nur Passagiere mit verschiedenem Alter haben. Dann wird in $q7$ die Fluggesellschaft Tabelle dazugenommen, damit Datum und Fluggesellschaftsname zurückgegeben werden können.

Quelle - eine analoge Frage im StackOverflow: <https://stackoverflow.com/questions/43567351/relational-algebra-natural-join-of-table-q2-with-rename-of-table-q2-for-passege>

4 Aufgabe

```

2 <!DOCTYPE html>
3 <html>
4 <head>
5   <meta charset="utf-8"></meta>
6   <title>Apple Aktie</title>
7   <style type="text/css">

9     #linechart, #barchart {

11       width: 600px;
12       height: 400px;
13     }

15   </style>
16   <script type="text/javascript" language="javascript" src="./bower_components/jquery/
    dist/jquery.min.js"></script>
17   <script type="text/javascript" language="javascript" src="./bower_components/Flot/
    jquery.flot.js"></script>

```

```

18 </head>
19 <body>
20 <div id="linechart"></div>
21 <div id="barchart"></div>
22 <script type="text/javascript">

24 function parseDataToObject(data) {

26     var rows = data.split('\r\n').map(function (item) {
27         return item.replace('; ', ' ').split(' ');
28     });
29     var attributes = rows[0];
30     var parsedObject = {};

32     rows = rows.slice(1);
33     attributes.forEach(function (attr, idx) {

35         parsedObject[attr] = rows.map(function (row) {

37             if(attr === 'Datum') {

39                 var crntDate = new Date(),
40                 crntDecade = crntDate.getFullYear().toString().slice(0,2);
41                 valueArr = row[idx].split('.');
42                 day = valueArr[0]*1,
43                 month = valueArr[1]*1 - 1,
44                 year = (crntDecade + valueArr[2])*1,
45                 date = new Date(year, month, day);

47                 return date.getDate()*1;
48             }
49             return row[idx]*1;
50         });
51     });

53     return parsedObject;
54 }

56 function mergeArrayEntities(arrA, arrB) {

58     var copyA = arrA.slice(0),
59     copyB = arrB.slice(0);

61     if(copyA.length !== copyB.length) {

63         throw new Error('Only arrays with equal length can be merged!');
64     }

66     return copyA.map(function (item, idx) {

68         return [item, copyB[idx]];
69     });
70 }

72 $.get('./data/apple.csv', function(data) {

74     var parsedData = parseDataToObject(data);

76     $.plot($('#linechart'),
77     [

79         {color: '#f00', data: mergeArrayEntities(parsedData.Datum, parsedData.Tief)},
80         {color: '#0f0', data: mergeArrayEntities(parsedData.Datum, parsedData.Hoch)},
81         {color: '#ff0', data: mergeArrayEntities(parsedData.Datum, parsedData.
            Tagesendwert)}

83     ]);

```

```

85     $.plot($('#barchart'),
86     [
87
88         {color: '#f00', data: mergeArrayEntities(parsedData.Datum, parsedData.
Handelsvolumen)}
89
90     ],
91     {
92         series: {
93             bars: {
94                 show: true
95             }
96         },
97         bars: {
98             align: "center",
99             barWidth: 0.5
100         }
101     });
102
103 });
104
105 </script>
106 </body>
107 </html>

```