Relationale Algebra

Datenbanken I (Systemorientierte Informatik IV)
Sommersemester 2007

Gunar Fiedler (fiedler@is.informatik.uni-kiel.de)
Institut für Informatik
Arbeitsgruppe „Technologie der Informationssysteme“
Christian-Albrechts-Universität zu Kiel

Die relationale Algebra ist eine Anfragesprache für relationale Datenbanksysteme. Sie definiert eine Menge von Operationen, die jeweils eine gegebene Menge von Relationen in eine Ergebnisrelation transformieren. Die Operationen der relationalen Algebra führen also stets von Relationen auf eine Relation. Dadurch lassen sich die Operationen (fast) beliebig kombinieren. Eine Anfrage der relationalen Algebra ist eine Folge von Operationen, die ausgehend von den „Basisrelationen“ des aktuellen Zustands der Datenbank das Anfrageergebnis erzeugen.

In der Übung betrachten wir eine Teilmenge der in der Vorlesung diskutierten Operationen: die Mengenoperationen, die Selektion, die Projektion, die Umbenennung, den natürlichen Verbund und die Division.

1 Mengenoperationen

Relationen sind Mengen von Tupeln. Deshalb lassen sich die üblichen Mengenoperationen auf Relationen anwenden. Seien R un S zwei Relationen **über denselben Attributen**. Dann ist

$$\begin{aligned} R \cup S &= \{t | t \in R \vee t \in S\} \\ R \cap S &= \{t | t \in R \wedge t \in S\} \\ R \backslash S &= \{t | t \in R \wedge t \notin S\} \end{aligned}$$

Beispiele:

MITARBEITER

Name	Wohnort
Max Müller	Kiel
Tina Schmidt	Lübeck
Klaus Meyer	Kiel

STUDENTEN

Name	Wohnort
Max Müller	Kiel
Andre Petersen	Hamburg
Thomas Ebert	Rendsburg

MITARBEITER ∪ STUDENTEN

Name	**Wohnort**
Max Müller	Kiel
Tina Schmidt	Lübeck
Klaus Meyer	Kiel
Andre Petersen	Hamburg
Thomas Ebert	Rendsburg

MITARBEITER ∩ STUDENTEN

Name	**Wohnort**
Max Müller	Kiel

MITARBEITER \ STUDENTEN

Name	**Wohnort**
Tina Schmidt	Lübeck
Klaus Meyer	Kiel

Wichtig: die Mengenoperationen sind nur für Relationen mit denselben Attributen definiert!

2 Selektion

Mit Hilfe der Selektion werden auf Grundlage einer gegebenen aussagenlogischen Formel die Tupel aus einer Relation ausgewählt, die diese Formel erfüllen. Die Formel φ darf nur Aussagen über Attribute enthalten, die in R vorhanden sind[1].

$$\sigma_{\varphi}(R) = \{t | t \in R \wedge t \models \varphi\}$$

Beispiele:

$\sigma_{Wohnort='Kiel'}$(MITARBEITER)

Name	**Wohnort**
Max Müller	Kiel
Klaus Meyer	Kiel

MITARBEITER ∪ ($\sigma_{Wohnort='Rendsburg'}$(STUDENTEN))

Name	**Wohnort**
Max Müller	Kiel
Tina Schmidt	Lübeck
Klaus Meyer	Kiel
Thomas Ebert	Rendsburg

[1] siehe auch das Übungsskript „Grundlagen der Logik“

3 Projektion

Die Projektion erstellt aus einer gegebenen Relation eine neue Relation, indem sie nur eine Teilmenge der vorhandenen Attribute auswählt. Während die Selektion Tupel auswählt, also bildlich gesprochen „Zeilen entfernt“, wählt die Projektion Attribute aus, d.h. es werden „Spalten entfernt“. Die Liste der Attribute, die in die Zielrelation übernommen werden sollen, werden der Projektion als Parameter mitgegeben. Die Menge der Zielattribute muss natürlich eine (echte oder unechte) Teilmenge der Attribute der gegebenen Relation sein. Man beachte, dass Relationen Mengen sind. Falls durch die Projektion doppelte Tupel entstehen, fallen diese zu einem einzigen Tupel in der Zielrelation zusammen. Da wir Tupel als Funktionen definiert haben, die Attribute auf Werte abbilden, können wir die Projektion als Einschränkung des Definitionsbereichs der Funktion auf die gewünschten Attribute definieren.

$$\pi_{A_1,\ldots,A_n}(R) = \{t|_{A_1,\ldots,A_n} \mid t \in R\}$$

Beispiele:
π_{Name}(STUDENTEN)

Name
Max Müller
Andre Petersen
Thomas Ebert

$\pi_{Wohnort}$(MITARBEITER)

Wohnort
Kiel
Lübeck

4 Umbenennung

Die Umbenennung gibt einem Attribut einen neuen Namen. Die Wertebereichsfunktion muss diese Umbenennung zulassen, d.h. die Datentypen des alten und des neuen Attributs müssen identisch sein. Außerdem darf der neue Attributname noch nicht in der Menge der Attribute der Relation enthalten sein. Sei $attr(R)$ die Menge der Attribute der Relation R:

$$\varrho_{A\to B}(R) = \{t|_{attr(R)\setminus\{A\}} \cup \{(B, t(A))\} \mid t \in R\}$$

Natürlich kann man mehrere Attribute „in einem Rutsch“ umbenennen. Dabei schreibt man die einzelnen Umbenennungen mit Komma getrennt als Parameter des Operators. Dies ist dann identisch mit der Nacheinanderausführung der einzelnen Umbenennungen.

Beispiel:
$\varrho_{Wohnort\to Ort}$(STUDENTEN)

Name	Ort
Max Müller	Kiel
Andre Petersen	Hamburg
Thomas Ebert	Rendsburg

5 Natürlicher Verbund

Der natürliche Verbund wird benutzt, um zwei Relationen zu verbinden. Die Attribute der beiden Relationen R und S lassen sich in drei Gruppen einteilen:

1. Attribute, die in R, aber nicht in S vorkommen
2. Attribute, die in S, aber nicht in R vorkommen
3. Attribute, die in beiden Relationen vorkommen

Die Attribute der dritten Kategorie bilden das „verbindende Element" der beiden Relationen. Wir erzeugen die Tupel t der Ergebnisrelation $R \bowtie S$ so, dass

1. wenn wir t auf die Attribute von R projizieren, ein gleiches Tupel in R existiert und
2. wenn wir t auf die Attribute von S projizieren, ein gleiches Tupel in S existiert.

Daraus folgt, dass wir alle Paare von Tupeln $r \in R$ und $s \in S$ betrachten, die in den gemeinsamen Attributen (Kategorie drei) die gleichen Werte haben. Wir verbinden die beide Tupel r und s zu t und fügen t zur Ergebnisrelation hinzu.

$$R \bowtie S = \{t \mid t|_{attr(R)} \in R \wedge t|_{attr(S)} \in S\}$$

Wenn wir uns einen Algorithmus zum Berechnen des natürlichen Verbunds zweier Relationen überlegen, können wir u.a. den „Nested-Loop-Join" anwenden. Seien $A_1, ..., A_n$ die gemeinsamen Attribute der Relationen R und S, dann berechnet sich der natürliche Verbund $T = R \bowtie S$ folgendermaßen:

```
T := ∅
FORALL r IN R DO
  FORALL s IN S DO
    IF r.A1 = s.A1 AND ... AND r.An = s.An THEN
      t := r ∪ s
      T := T ∪ {t}
    END IF
  END FOR
END FOR
RETURN T
```

Weitere (effizientere) Algorithmen zur Berechnung des natürlichen Verbunds werden wir später kennenlernen.

Beispiele:

VORLESUNG

Kuerzel	**Bezeichnung**
SysInf IV	Datenbanken I
Info III	Softwaretechnologie
Info II	Algorithmen und Datenstrukturen
SysInf I	Digitale Systeme

HOERT

Name	**Kuerzel**
Max Müller	SysInf IV
Max Müller	Info III
Andre Petersen	Info II
Andre Petersen	Info III

STUDENTEN $\bowtie$ HOERT

Name	**Wohnort**	**Kuerzel**
Max Müller	Kiel	SysInf IV
Max Müller	Kiel	Info III
Andre Petersen	Hamburg	Info II
Andre Petersen	Hamburg	Info III

STUDENTEN $\bowtie$ HOERT $\bowtie$ VORLESUNG

Name	**Wohnort**	**Kuerzel**	**Bezeichnung**
Max Müller	Kiel	SysInf IV	Datenbanken I
Max Müller	Kiel	Info III	Softwaretechnologie
Andre Petersen	Hamburg	Info II	Algorithmen und Datenstrukturen
Andre Petersen	Hamburg	Info III	Softwaretechnologie

$\pi_{Name,Bezeichnung}((\sigma_{Wohnort='Kiel'}(\text{STUDENTEN})) \bowtie \text{HOERT} \bowtie \text{VORLESUNG})$

Name	**Bezeichnung**
Max Müller	Datenbanken I
Max Müller	Softwaretechnologie

Der Student Max Müller hört die Veranstaltungen mit den Kürzeln „SysInf IV“ und „Info III“. Wenn wir die Relationen `STUDENTEN` und `HOERT` verbinden, ist `Name` das einzige gemeinsame Attribut. Wir schauen also alle Tupel in `STUDENTEN` an. Für jeden gefundenen Studenten schauen wir uns alle Tupel in `HOERT` an. Wenn wir einen Studenten und ein `HOERT`-Tupel finden, die im Attribut `Name` übereinstimmen, dann verbinden wir die beiden Tupel und fügen das neue Tupel zum Ergebnis hinzu. Da der Student Thomas Ebert keine Veranstaltung hört, fällt er aus dem Ergebnis heraus.

Auf die gleiche Art und Weise können wir das so entstandene Ergebnis mit der Relation `VORLESUNG` verbinden. `Kuerzel` ist das verbindende Attribut. Da die Veranstaltung „SysInf I“ von keinem Studenten gehört wird, taucht sie in der Ergebnisrelation nicht auf.

Falls die beiden Relationen R und S keine gemeinsamen Attribute haben, wird das Kreuzprodukt beider Relationen gebildet, d.h. jedes Tupel aus R wird mit jedem Tupel aus S verknüpft.

6 Division

Der Divisionsoperator erlaubt die kompakte Formulierung von „für-alle“-Anfragen. Betrachten wir folgende Relationen:

VORLESUNG

Kuerzel	Bezeichnung
SysInf IV	Datenbanken I
Info III	Softwaretechnologie
Info II	Algorithmen und Datenstrukturen
SysInf I	Digitale Systeme

HOERT

Name	Kuerzel
Max Müller	SysInf IV
Max Müller	Info III
Max Müller	Info II
Max Müller	SysInf I
Andre Petersen	Info II
Andre Petersen	Info III

Wenn man jetzt die Anfrage „Welcher Student hört alle Vorlesungen“ stellt, dann suchen wir die Namen, für die für jedes Kürzel in der Relation `VORLESUNG` ein passendes Tupel in der Relation `HOERT` existiert (in unserem Beispiel ist dies Max Müller.) Das leistet der Divisionsoperator:

HOERT $\div$ ($\pi_{Kuerzel}$(VORLESUNG))

Name
Max Müller

Formal gesprochen: es existieren zwei Relationen R und S, wobei die Attributmenge der Relation S eine echte Teilmenge der Attributmenge von R ist: $attr(S) \subsetneq attr(R)$. Das Ergebnis der Division ist eine Relation über den Attributen, die in R, aber nicht in S vorkommen ($attr(R \div S) = attr(R) \backslash attr(S)$). Diese Relation enthält genau die Tupel t, die aus einem Tupel $r \in R$ durch Projektion auf $attr(R) \backslash attr(S)$ entstehen, so dass man dieses Tupel mit allen Tupeln $s \in S$ ergänzen kann, um wieder ein Tupel aus R zu erzeugen. Mit anderen Worten: das Ergebnistupel t steht in R mit allen Tupeln der Relation S „in Beziehung“:

$$R \div S = \{t \mid attr(t) = attr(R) \backslash attr(S) \land \{t\} \bowtie S \subseteq R\}$$

7 Anfragebeispiele

Wir wenden nun die vorgestellten Operationen an, um Anfragen an ein Beispielschema zu stellen. Wichtig: wir stellen Anfragen immer gegen ein Datenbankschema, nicht gegen einen konkreten Datenbankzustand. Die Auswertung der Anfrage erfolgt

stets bzgl. eines konkreten Datenbankzustandes. Unsere Anfrage muss aber für alle gültigen Zustände unseres Schemas funktionieren.
Wir benutzen folgendes Beispielschema (Primärschlüssel sind unterstrichen):

```
{
STUDENT({MatrikelNr,Name,Wohnort}),
MITARBEITER({BearbeiterNr,PersonalNr,Name,Wohnort}),
VORLESUNG({VorlesungsNr,Bezeichnung}),
DOZENT({BearbeiterNr,PersonalNr,VorlesungsNr}),
HOERT({MatrikelNr,VorlesungsNr,Wiederholung}),
FINDETSTATT({VorlesungsNr,Zeit,RaumNr}),
RAUM({RaumNr,Bezeichnung})
}
```

Folgende Fremdschlüssel sind definiert:

$DOZENT[BearbeiterNr, PersonalNr] \subseteq MITARBEITER[BearbeiterNr, PersonalNr]$
$DOZENT[VorlesungsNr] \subseteq VORLESUNG[VorlesungsNr]$
$HOERT[MatrikelNr] \subseteq STUDENT[MatrikelNr]$
$HOERT[VorlesungsNr] \subseteq VORLESUNG[VorlesungsNr]$
$FINDETSTATT[VorlesungsNr] \subseteq VORLESUNG[VorlesungsNr]$
$FINDETSTATT[RaumNr] \subseteq RAUM[RaumNr]$

1. Anfrage „Gib die Bezeichnung der Vorlesung '080104'.“
Die Daten zu Vorlesungen stehen in der Relation `VORLESUNG`. '080104' ist eine Vorlesungsnummer einer konkreten Vorlesung, also müssen wir diese konkrete Vorlesung selektieren. Wir interessieren uns nur für die Bezeichnung dieser Vorlesung, also müssen wir das Ergebnis auf das Attribut `Bezeichnung` projizieren:

$$\pi_{Bezeichnung}(\sigma_{VorlesungsNr='080104'}(VORLESUNG))$$

2. Anfrage „Gib die Namen aller Studenten, die die Veranstaltung '080104' hören, zusammen mit den Namen aller Dozenten der Veranstaltung '080104'.
Die Daten der Studenten stehen in der Relation `STUDENTEN`, die Teilnahme in der Relation `HOERT`. Wenn wir aus `HOERT` die Tupel für die Veranstaltung '080104' selektieren, erhalten wir die Matrikelnummern der an '080104' teilnehmenden Studenten. Wenn wir dieses Zwischenergebnis mit der Relation `STUDENTEN` verbinden und anschließend projizieren, erhalten wir die Namen dieser Studenten. Analog verfahren wir mit `DOZENT` und `MITARBEITER`. Beide Relationen zusammen bilden das Ergebnis der Anfrage.

$$\pi_{Name}((\sigma_{VorlesungsNr='080104'}(HOERT)) \bowtie STUDENTEN) \cup$$
$$\pi_{Name}((\sigma_{VorlesungsNr='080104'}(DOZENT)) \bowtie MITARBEITER)$$

3. Anfrage Angenommen, der Name identifiziert eine Person eindeutig. „Gib die Personen, die Dozent einer Veranstaltung sind und sich parallel dazu für diese Veranstaltung als Student angemeldet haben.“
Wir verbinden die `DOZENT`-Relation mit der `MITARBEITER`-Relation und projizieren anschließend auf die Attribute `Name` und `VorlesungsNr`, so bekommen wir die Namen der Dozenten einer Veranstaltung. Analog verfahren wir mit den eingeschriebenen Studenten. Der Durchschnitt beider Mengen enthält die Personen, die gleichzeitig Dozent und Student einer Vorlesung sind.

$$\pi_{Name,VorlesungsNr}(DOZENT \bowtie MITARBEITER) \cap$$
$$\pi_{Name,VorlesungsNr}(HOERT \bowtie STUDENT)$$

4. Anfrage „Finde Überschneidungen, d.h. gib die Namen der Studenten zusammen mit der entsprechenden Zeit aus, so dass dieser Student zu diesem Zeitpunkt in zwei Räumen präsent sein muss.“
Wir bilden für jeden Studenten Paare von Teilnahmen an Vorlesungsdurchführungen und verwerfen die Paare, deren Zeiten unterschiedlich sind. Teilnahmen an Vorlesungsdurchführungen erhalten wir durch das Verbinden der Relationen `HOERT` und `FINDETSTATT`. Da Fremdschlüssel bzgl. der Relation `VORLESUNG` definiert sind, können wir die Relation `VORLESUNG` weglassen[2]. Wir benötigen das Attribut `MatrikelNr` aus `HOERT` und die Attribute `Zeit` und `RaumNr` aus `FINDETSTATT`. Wir führen diese Anfrage zweimal aus, beim zweiten mal benennen wir alle Attribute bis auf die Matrikelnummer um. Anschließend selektieren wir alle Tupel, deren Zeiten gleich, deren Räume aber verschieden sind. Diese Menge verbinden wir mit der `STUDENT`-Relation und projizieren alles außer dem Namen und der Zeit aus.

$$\begin{array}{l}
\pi_{Name,Zeit}(\\
\quad \sigma_{Zeit=Zeit2 \wedge RaumNr!=RaumNr2}(\\
\quad\quad \pi_{MatrikelNr,Zeit,RaumNr}(HOERT \bowtie FINDETSTATT) \\
\quad\quad \bowtie \\
\quad\quad (\varrho_{Zeit \rightarrow Zeit2, RaumNr \rightarrow RaumNr2}(\\
\quad\quad\quad \pi_{MatrikelNr,Zeit,RaumNr}(HOERT \bowtie FINDETSTATT) \\
\quad\quad)) \\
\quad) \\
\quad \bowtie \\
\quad \text{STUDENT} \\
)
\end{array}$$

[2]Überlegen Sie sich, was passiert, wenn die Fremdschlüssel nicht definiert wären.

5. Anfrage „Gib die Studenten, die bei allen Dozenten eine Veranstaltung hören.“ Zunächst benötigen wir eine Zuordnung von Studenten (Matrikelnummern) zu Dozenten (Bearbeiternummer, Personalnummer). Anschließend teilen wir diese Relation durch die Relation mit allen Dozenten (Bearbeiternummer, Personalnummer) und erhalten das gewünschte Ergebnis.

$$(\pi_{MatrikelNr,BearbeiterNr,PersonalNr}(HOERT \bowtie DOZENT))$$
$$\div$$
$$(\pi_{BearbeiterNr,PersonalNr}(DOZENT))$$

Selbststudium Veranschaulichen Sie sich die Auswertung der Anfragen an einem Beispielzustand.