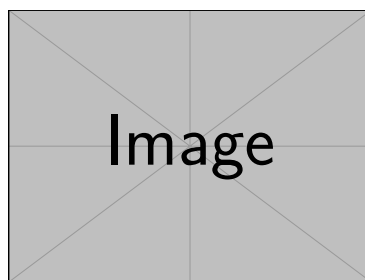


Notatki z wykładów

Matematyka – Analiza I



Twoje Imię i Nazwisko

Semestr zimowy 2025/2026

Hochschule Emden/Leer

Contents

1	Relationale Algebra	2
1.1	Relationale Algebra	2
1.1.1	Relationen	2
1.1.2	Vereinigungsverträglichkeit	3
1.1.3	Mengenoperationen für Relationen	4
1.1.4	Projektion	5
1.1.5	Umbenennung (Rename)	5
1.1.6	Auswahl (Select)	6
1.2	Ein Verknüpfungsoperator für Relationen	7
1.2.1	Verknüpfung von Tupeln (Konkatenation)	7
1.2.2	Kreuzprodukt	7
1.2.3	Übung	8
1.3	Aufgabenblatt 1	10
1.3.1	Aufgabe 1	10
1.3.2	Aufgabe 2	10
2	Aufgabenblätter	13
2.1	Aufgabe 2	14
2.1.1	Aufgabe 2.1	14
2.1.2	Aufgabe 2.2	15

Chapter 1

Relationale Algebra

1.1 Relationsale Algebra

Relationale Algebra to zbiór operacji, które przyjmują jedną lub więcej relacji (tabel) jako dane wejściowe i zwracają nową relację jako wynik. Wszystko w niej opiera się na zbiorach i operacjach matematycznych.

Die wichtigste operationen:

Table 1.1: Wichtige Operationen der Relationalen Algebra

Operation	Symbol	Beschreibung
Selektion	σ	Wählt Tupel aus, die eine gegebene Bedingung erfüllen.
Projektion	π	Wählt bestimmte Attribute (Spalten) einer Relation aus.
Vereinigung	\cup	Kombiniert die Tupel zweier Relationen mit gleicher Struktur (wie UNION in SQL).
Differenz	$-$	Liefert die Tupel, die in der ersten, aber nicht in der zweiten Relation vorkommen.
Kartesisches Produkt	\times	Bildet alle möglichen Kombinationen von Tupeln aus zwei Relationen.
Join (Verbund)	\bowtie	Verknüpft zwei Relationen über gleiche Attribute oder Bedingungen.

1.1.1 Relationen

Eine Relation R (Tabelle) ist eine Teilmenge des Kreuzproduktes $Att_1 \times \dots \times Att_n$. Dies wird $R \subseteq Att_1 \times \dots \times Att_n$ geschrieben.

Hinweis

Relacja to matematyczny model tabeli w relacyjnej bazie danych. Jest to zbiór tuples, które mają taką samą strukturę atrybutów:

$$R \subseteq Att_1 \times \dots \times Att_n$$

gdzie:

R - nazwa relacji (np. Student)

A_1, A_2, \dots, A_n nazwy atrybutów (Matrikelnummer, Name, Fachrichtung).

1.1.2 Vereinigungsverträglichkeit

2 Relationen sind Vereinigungsverträglich, wenn sie:

1. denselben Anzahl an Attributen haben
2. die entsprechenden Attribute A_i in R und B_i in S denselben Datentyp oder einen gemeinsamen Obertyp besitzen.

Hinweis

$$R \subseteq Att_1 \times \dots \times Att_n$$

$$S \subseteq Btt_1 \times \dots \times Btt_n$$

$$mitTyp(A_i) = Typ(B_i)$$

Beispiel

Table 1.2: Beispiel: Vereinigungsverträgliche Relationen

Relation R		Relation S		Relation C	
Matr-nummer	Name	Matr-nummer	Name	Name	Alter
101	Anna	103	Carla	Anna	21
102	Ben	104	David	Ben	22

Die relationen R und S sind verträglich, da Sie in einer tabelle dargestellt werden (verbunden). Die relationen R und C oder S und C sind net kompatibel, da die Attribute nicht gleich sind (siehe subsection 1.1.3)

1.1.3 Mengenoperationen für Relationen

Seien R und S vereinigungsverträglich, dann kann man neue Relationen berechnen:

1. **Schnittmenge** $R \cap S = \{r \mid r \in R \wedge r \in S\}$ - Einträge, die in beiden Relationen vorkommen
2. **Vereinigung** $R \cup S = \{r \mid r \in R \vee r \in S\}$ - Zusammenfassung aller Einträge der Relationen
3. **Differenz** $R - S = \{r \mid r \in R \wedge r \notin S\}$ - Suchen nach Einträgen, die nur in der ersten, aber nicht in der zweiten Relation vorkommen

Hinweis

Bei Relationen handelt es sich um Mengen, daher keine Zeile kommt doppelt vor!

Beispiel

VK

Verkäufer	Produkt	Käufer
Meier	Hose	Schmidt
Müller	Rock	Schmidt
Meier	Hose	Schulz

VK2

Verkäufer	Produkt	Käufer
Müller	Hemd	Schmidt
Müller	Rock	Schmidt
Meier	Rock	Schulz

$VK \cup VK2$

Verkäufer	Produkt	Käufer
Meier	Hose	Schmidt
Müller	Rock	Schmidt
Meier	Hose	Schulz
Müller	Hemd	Schmidt
Meier	Rock	Schulz

$VK \cap VK2$

Verkäufer	Produkt	Käufer
Müller	Rock	Schmidt

$VK - VK2$

Verkäufer	Produkt	Käufer
Meier	Hose	Schmidt
Meier	Hose	Schulz

1.1.4 Projektion

Sei $R \subseteq \text{Att}_1 \times \dots \times \text{Att}_n$ eine Relation und B_1, \dots, B_j verschiedene Attribute aus der Menge $\{\text{Att}_1, \dots, \text{Att}_n\}$.

Dann ist die **Projektion** von R auf B_1, \dots, B_j , geschrieben als

$$\text{Proj}(R, [B_1, \dots, B_j]),$$

die Relation, die entsteht, wenn man aus R alle Spalten entfernt, die nicht in B_1, \dots, B_j enthalten sind.

Die Reihenfolge der Attribute B_1, \dots, B_j bestimmt zugleich die Reihenfolge der Spalten in der Ergebnisrelation.

Hinweis

Projekcja służy do wyboru określonych kolumn (atrybutów) z relacji. Odrzuca wszystkie pozostałe atrybuty i często też usuwa duplikaty, ponieważ relacja w matematycznym sensie to zbiór (a zbiór nie zawiera powtórzeń).

Beispiel

$\text{Proj}(VK, [K\ddot{a}ufer, Product])$		$\text{Proj}(VK, [Verk\ddot{a}ufer])$	
Käufer	Produkt	Verkäufer	
Schmidt	Hose	Meier	
Schmidt	Rock	Müller	
Schulz	Hose		

$\text{Proj}(VK \cap VK2, [Product])$	
Produkt	
Rock	

1.1.5 Umbenennung (Rename)

Sei R eine Relation, dann bezeichnet $\text{Ren}(R, T)$ eine Relation mit gleichem Inhalt wie R , die T genannt wird.

Hinweis

Tego używa się gdy tabela sama ze sobą musi być zestawiona. Jeśli mamy 2 razy nazwę tej samej tabeli i potem chcemy operować na Atrybutach tej tabeli to SQL nie wie o którą tabelkę nam chodzi. Dlatego robimy $TAB1$ i $\text{Ren}(TAB1, TAB2)$ i teraz pod $TAB1$ i $TAB2$ mamy tą samą tabelę i możemy operować na jej kolumnach

1.1.6 Auswahl (Select)

Sei R eine Relation, dann bezeichnet $Sel(R, Bed)$ eine Relation, die alle Zeilen aus R beinhaltet, die die Bedingung Bed erfüllen.

Syntax

Syntax der Bedingungen Bed : $Att_1 \text{ OPERATOR KONSTANTE}$

- OPERATOR - =, <>, <=, >=, <, >
- KONSTANTE - muss ein Wert des zum Attribut gehörenden Datentyps sein. Es kann auch ein Attribut aus anderer Spalte sein - hierbei muss der Typ des Attributs gleich sein, oder sie müssen einen Gemeinsamen Obertyp besitzen

Es besteht auch die Möglichkeit mehrere Bedingungen einzuführen:

- $Bed_1 \text{ AND } Bed_2$ - beide Bedingungen sollen erfüllt sein
- $Bed_1 \text{ OR } Bed_2$ - mindestens eine der Bedingungen soll erfüllt sein
- $NOT \text{ } Bed_1$ - die Bedingung soll nicht erfüllt sein
- (Bed_1) - die Bedingung in Klammern werden zuerst ausgewertet

Beispiel: Alle Verkäufe, die Meier gemacht hat

$Sel(VK, VK.Verkäufer = 'Meier')$

Verkäufer	Produkt	Käufer
Meier	Hose	Schmidt
Meier	Hose	Schlus

Beispiel: Alle Käufer, die bei Meier gekauft haben

$Proj(Sel(VK, VK.Verkäufer = 'Meier'), ['Käufer'])$

Käufer
Schmidt
Schlus

Beispiel: Alle Verkäuf, die Meier gemacht hat und die nicht den Kunden Schulz betreffen

1.2 Ein Verknüpfungoperator für Relationen

Bislang beziehen sich operationen auf einzelne Tabellen. Durch das kreuzprodukt können mehrerer, auch verschiedene Tabellen miteinander Verknüpft.

1.2.1 Verknüpfung von Tupeln (Konkatenation)

Seien R und S Relationen mit $r = \{r_1, \dots, r_n\} \in R$ und $s = \{s_1, \dots, s_n\} \in S$. Dann ist die Verknüpfung oder Konkatenation von r mit s , geschrieben $r \circ s$, definiert als $\{r_1, \dots, r_n, s_1, \dots, s_n\}$.

1.2.2 Kreuzprodukt

Seien R und S Relationen, dann ist das kreuzprodukt von R und S , geschrieben $R \times S$, sefiniert durch $R \times S = \{r \circ s \mid r \in R \text{ und } s \in S\}$

Hinweis

Konkatenacja to operacja na pojedynczych elementach, łączy je w jeden dłuższy element. Kreuzprodukt to operacja na zbiorach elementów, która generuje nową relację, która zawiera wszystkie moliwe kombinacje krotek z R i S

Konkatenation vs Kreuzprodukt

Seien

$$R = \{(a_1), (a_2)\} \quad \text{und} \quad S = \{(b_1), (b_2)\}.$$

Dann ist die **Konkatenation** einzelner Tupel definiert als:

$$(a_1) \circ (b_1) = (a_1, b_1)$$

Das **Kreuzprodukt** der Relationen R und S besteht aus allen möglichen Konkatenationen von Tupeln aus R und S :

$$R \times S = \{r \circ s \mid r \in R, s \in S\}$$

Konkret ergibt sich hier:

$$R \times S = \{(a_1, b_1), (a_1, b_2), (a_2, b_1), (a_2, b_2)\}$$

1.2.3 Übung

Relationen

Projekt		Aufgabe			Maschine		
ProNr	Name	AufNr	Arbeit	ProNr	Mname	Dater	AufNr
1.	Schachtel	1.	knicken	1	M1	2	1
2.	Behang	2.	kleben	1	M2	3	1
		3.	knicken	2	M1	3	2
		4.	färben	2	M3	2	3
					M1	1	4
					M4	3	4

1. Geben Sie die Namen aller möglichen Arbeiten an

$Proj(Aufgabe, [Arbeit])$

2. Geben Sie zu jedem Projektnamen die zugehörigen Arbeiten an. Das Ergebnis ist eine Relation mit den Attributen „Name“ und „Arbeit“.

$Proj(\text{Sel}(Projekt \times Aufgabe, Projekt.ProNr = Aufgabe.ProNr), [Name, Arbeit])$

Przykład dla Kreuzprodukt $Projekt \times Aufgabe$

Projekt \times Aufgabe

ProNr	Name	AufNr	Arbeit	ProNr
1	Schachtel	1	knicken	1
1	Schachtel	2	kleben	1
1	Schachtel	3	knicken	2
1	Schachtel	4	färben	2
2	Behang	1	knicken	1
2	Behang	2	kleben	1
2	Behang	3	knicken	2
2	Behang	4	färben	2

3. Welche Maschinen werden zum Knicken genutzt?

$Proj(\text{Sel}(Aufgabe \times Maschine, Aufgabe.AufNr = Maschine.AufNr$
AND $Aufgabe.Arbeit = 'knicken'), [Mname])$

4. Geben Sie zu jedem Projektnamen die Maschinen aus, die genutzt werden

$$\text{Proj}\left(\text{Sel}(\text{Projekt} \times \text{Aufgabe} \times \text{Maschine}, \text{Projekt.ProNr} = \text{Aufgabe.ProNr} \text{ AND } \text{Aufgabe.AufNr} = \text{Maschine.AufNr}), [\text{ProjName}, \text{Mname}]\right)$$

5. Geben Sie alle Projekte (deren Namen) aus, bei denen geknickt und gefärbt wird

$$\text{Proj}\left(\text{Sel}(\text{Projekt} \times \text{Aufgabe} \times \text{Ren}(\text{Aufgabe}, A2), \text{Projekt.ProNr} = \text{Aufgabe.ProNr} \text{ AND } \text{Projekt.ProNr} = A2.ProNr \text{ AND } \text{Aufgabe.Arbeit} = \text{'knicken'} \text{ AND } A2.Arbeit = \text{'färben'}), [\text{Name}]\right)$$

1.3 Aufgabenblatt 1

1.3.1 Aufgabe 1

Der Ren-Operator wird benötigt, wenn ein Kreuzprodukt einer Tabelle mit sich selbst (Self-Join) gebildet werden muss.

Aufgaben:

ProzessNr	Name	VorgängerNr
1	Schneiden	-
2	Waschen	1
3	Biegen	2
4	Bohren	2
5	Malen	4

Gib die Aufgaben und deren Vorgänger aus.

$\text{Proj}(\text{Sel}(\text{Aufgaben} \times \text{Ren}(\text{Aufgaben}, A2), \text{Aufgabe.VorgängerNr} = A2.\text{ProzessNr}),$
 $[\text{Aufgabe.Name}, A2.\text{Name}])$

1.3.2 Aufgabe 2

Relationen

Student	
MatNr	Name
1	Meier
2	Meyer
3	Maier

Gericht		
GNr	Name	Art
1	Pizza	Haupt
2	TomatenSuppe	Vor
3	Schnitzel	Haupt
4	Reis	Beilage
5	Pudding	Nach

Bewertung		
MatrNr	GNr	Sterne
1	2	3
1	4	2
2	1	4
3	3	3

1. Geben Sie alle Arten von Gerichten aus.

$\text{Proj}(\text{Gericht}, [\text{Art}])$

Ergebnistabelle

Art
Haupt
Vor
Beilage
Nach

2. Geben Sie die Namen aller Hauptgerichte (mit der Art „Haupt“) aus.

$$\text{Proj}\left(\text{Sel}(\text{Gericht}, [\text{Art} = \text{'Haupt'}]), [\text{Name}]\right)$$

Ergebnistabelle

Name
Pizza
Schnitzel

3. Geben Sie eine Liste aller einzelnen Bewertungen aus (Ausgabe: Name des Gerichts, Sterne).

$$\text{Proj}\left(\text{Sel}(\text{Gericht} \times \text{Bewertung}, \text{Gericht.GNr} = \text{Bewertung.GNr}), [\text{Name}, \text{Sterne}]\right)$$

Ergebnistabelle

Name	Sterne
Pizza	4
TomatenSuppe	3
Schnitzel	3
Reis	2

4. Geben Sie die Namen aller Gerichte aus, die der Student Meier bewertet hat.

$$\begin{aligned} &\text{Proj}\left(\text{Sel}(\text{Student} \times \text{Gericht} \times \text{Bewertung}, \text{Student.MatNr} = \text{Bewertung.MatNr} \right. \\ &\quad \text{AND } \text{Bewertung.GNr} = \text{Gericht.GNr} \text{ AND } \text{Student.Name} = \text{'Meier'}), \\ &\quad \left. [\text{Gericht.Name}]\right) \end{aligned}$$

Ergebnistabelle

Name
TomatenSuppe
Reis

5. Geben Sie alle Bewertungen aus (Name Student, Name Gericht, Sterne), die min-

destens vier Sterne haben.

$\text{Proj}\left(\text{Sel}(\text{Student} \times \text{Gericht} \times \text{Bewertung}, \text{Student.MatNr} = \text{Bewertung.MatrNr} \text{ AND } \text{Bewertung.GNr} = \text{Gericht.GNr} \text{ AND } \text{Bewertung.Sterne} \geq 4), [\text{Student.Name}, \text{Gericht.Name}, \text{Sterne}]\right)$

Ergebnistabelle

Name Student	Name Gericht	Sterne
Meyer	Pizza	4

6. Geben Sie aus, welche Studierenden das Schnitzel bewertet haben.

$\text{Proj}\left(\text{Sel}(\text{Student} \times \text{Gericht} \times \text{Bewertung}, \text{Student.MatNr} = \text{Bewertung.MatNr} \text{ AND } \text{Bewertung.GNr} = \text{Gericht.GNr} \text{ AND } \text{Gericht.Name} = \text{'Schnitzel'}), [\text{Student.Name}]\right)$

Ergebnistabelle

Student Name
Maier

7. Geben Sie aus, welcher Studierende mindestens zwei Bewertungen abgegeben hat.

$\text{Proj}\left(\text{Sel}(\text{Student} \times \text{Bewertung} \times \text{Ren}(\text{Bewertung}, B2), \text{Student.MatrNr} = \text{Bewertung.MatrNr} \text{ AND } \text{Student.MatrNr} = B2.\text{MatrNr} \text{ AND } \text{Bewertung.GNr} \neq B2.\text{GNr}), [\text{Student.Name}]\right)$

Ergebnistabelle

Student Name
Meier

Chapter 2

Aufgabenblätter

2.1 Aufgabe 2

Relationen

Student	
MatNr	Name
1	Meier
2	Meyer
3	Maier

Klausur			
KNr	Name	Datum	Zeit
1	Java 1	2024-01-14	10:00:00
2	Einführung Inf.	2024-01-16r	08:00:00
3	Mathematik 1	2024-01-20	13:00:00
4	Medieninformatik	2024-01-20	08:00:00
5	Audio/Video	2024-01-28	15:30:00

Bewertung

MatrNr	KNr	Versuch
1	2	1
1	4	2
2	1	2
3	3	3

2.1.1 Aufgabe 2.1

Listing 2.1: Erstellen der Tabellen STUDENT, KLAUSUR und ANMELDUNG

```

CREATE TABLE STUDENT(
    MatrNr INTEGER,
    Name VARCHAR(5),
    PRIMARY KEY (MatrNr)
);

CREATE TABLE KLAUSUR(
    KNr INTEGER,
    Name VARCHAR(25),
    Datum DATE,
    Zeit TIME,
    PRIMARY KEY (KNr)
);

CREATE TABLE ANMELDUNG(
    MatrNr INTEGER,
    KNr INTEGER,
    Versuch INTEGER,
    CONSTRAINT FK_MatrNr
        FOREIGN KEY (MatrNr)
        REFERENCES STUDENT(MatNr),
    CONSTRAINT FK_KNr
        FOREIGN KEY (KNr)

```

REFERENCES KLAUSUR(KNr)

);

Listing 2.2: Einfügen von Datensätzen in die Tabellen STUDENT, KLAUSUR und ANMELDUNG

```
INSERT INTO STUDENT VALUES(1,'Meier');
INSERT INTO STUDENT VALUES(2,'Meyer');
INSERT INTO STUDENT VALUES(3,'Maier');

INSERT INTO KLAUSUR VALUES(1,'Java 1','2024-01-14','10:00:00');
INSERT INTO KLAUSUR VALUES(2,'Einführung Inf.','2024-01-16','08:00:00');
INSERT INTO KLAUSUR VALUES(3,'Mathematik 1','2024-01-20','13:00:00');
INSERT INTO KLAUSUR VALUES(4,'Medieninformatik','2024-01-20','08:00:00');
INSERT INTO KLAUSUR VALUES(5,'Audio/Video','2024-01-28','15:30:00');

INSERT INTO ANMELDUNG VALUES(1,2,1);
INSERT INTO ANMELDUNG VALUES(1,4,2);
INSERT INTO ANMELDUNG VALUES(2,1,2);
INSERT INTO ANMELDUNG VALUES(3,3,3);
```

2.1.2 Aufgabe 2.2

Listing 2.3: Einfügen von Datensätzen in die Tabellen STUDENT, KLAUSUR und ANMELDUNG

```
-- 1. Geben Sie die Namen aller Studierenden aus.
SELECT klausur.Name FROM student;

-- 2. Geben Sie die Namen aller Klausuren aus, die um 08:00 Uhr geschrieben
    werden.
SELECT klausur.name FROM klausur WHERE zeit = '08:00:00';

-- 3. Geben Sie eine Liste aller Erstanmeldungen (nur 1. Versuch) fuer eine
    Klausur aus (Ausgabe: Name der Klausur, Name des Studierenden).
SELECT klausur.name, student.name FROM klausur, student, anmeldung WHERE
    anmeldung.versuch = 1 AND klausur.knr = anmeldung.knr AND student.matrnr
    = anmeldung.matrnr;

-- 4. Geben Sie die Namen aller Klausuren aus, fuer die sich die Studentin
    "Meier" angemeldet hat.
```



```
SELECT klausur.name FROM klausur, student, anmeldung WHERE student.name  
Like 'Meier' AND klausur.knr = anmeldung.knr AND student.matrnr =  
anmeldung.matrnr;
```

*-- 5. Geben Sie die Namen aller Studierenden aus, die mindestens zwei
Klausuren im letzten Versuch (3. Versuch) schreiben.*

```
SELECT student.name  
FROM student, anmeldung a1, anmeldung a2  
WHERE student.matrnr = a1.matrnr  
AND student.matrnr = a2.matrnr  
AND a1.knr <> a2.knr  
AND a1.versuch >= 3  
AND a2.versuch >= 3;
```