Datenbanken

Frühlingssemester 23

Manuel Flückiger
Universität Bern

über dieses Dokument

Das ist eine Zusammenfassung der Vorlesung “Datenbanken” von Prof. Dr. Thomas Studer im Frühlingssemester 2023. Du darfst sie so verwenden, wie sie dir am meisten beim Verständnis des Materials hilft. Verantwortlich dafür was hier drin steht ist Manuel Flückiger. Beachte, dass der Dozent dieses Dokument nicht geschrieben (und vielleicht auch nicht gelesen) hat. Falls du selbst Fehler entdeckst, oder Verbesserungsvorschläge hast, sind diese unter der Adresse

`manuel.flueckiger@students.unibe.ch`

zu melden. Bei dieser Zusammenfassung ist zu beachten, dass der Aufbau ein Plagiat von Theo Oggier’s Skripten vom Gymnasium Burgdorf sind. Es dient mir vorallem beim lernen von LaTeXund beim Verständnis der Vorlesung.

Anerkenntnis

Diese Zusammenfassung wird als Testvorbereitung des Stoffes gemacht und kann daher viele Fehler enthalten. Falls ihr Fehler findet, meldet diese bitte, dann korrigiere ich es für zukünftige Leser/innen Vielen Dank euch! Hilfreiche Beiträge kamen von:
eure Namen, wenn ihr helft ;).

Inhaltsverzeichnis

1 Die Mengenlehre

1.1 Objekte in der Mengenlehre

Objekte sind entweder

- atomare Objekte, d.h. unteilbare Objekte ohne interne Struktur oder
- n-Tupel der Form $(a_1, a_2, \ldots, a_n)$, wobei a_1 bis a_n Objekte sind $(n \geq 1)$.

Die Komponenten eines Tupels sind geordnet.

Konvention

Wir benutzen Kleinbuchstaben $a, b, c, \ldots$ um Objekte zu bezeichnen.

Konvention

Spielt bei einem n-Tupel die Anzahl der Komponenten keine Rolle (oder ist sie klar aus dem Kontext), so sprechen wir einfach von einem Tupel.

Definition 1.1: Projektion auf Komponenten

Sei $a = (a_1, \ldots, a_i, \ldots, a_n)$ ein n-Tupel, dann nennen wir a_i die i-te Komponente von a. Wir definieren für $a = (a_1, \ldots, a_n)$ und $1 \leq i \leq n$:

$$\pi_i(a) := a_i$$

Mit Hilfe der Projektionsfunktion $\pi_i(a)$ können wir die i-te Komponente aus einem Tupel a extrahieren.

Beispiel:
Seien a, b und c Objekte. Dann sind auch $t = (a, a, b)$ sowie $s = ((a, a, b), c)$ Objekte.

- $\pi_1(t) = a$
- $\pi_2(t) = a$
- $\pi_3(t) = b$
- $\pi_1(s) = (a.a.b)$
- $\pi_2(s) = c$
- $\pi_3(s) = b$

Definition 1.2: Gleichheit in der Mengenlehre

Seien $a = (a_1, \ldots, a_n)$ und $b = (b_1, \ldots, b_n)$ zwei n-Tupel. Es gilt

$$a = b \; g.d.w. \; \forall 1 \leq i \leq n (a_i = b_i).$$

Das heisst, zwei n-Tupel sind genau dann gleich, wenn Gleichheit für alle Komponenten gilt.

Beispiel:

- $(999, Max, Muster) = (999, Max, Muster)$
- $(999, Max, Muster) \neq (999, (Max, Muster))$

1.2 Mengen

Eine Menge ist eine ungeordnete Kollektion von Objekten. Falls das Objekt a zu einer Menge M gehört, sagen wir a ist ein Element von M und schreiben $a \in M$.
Eine endliche Menge M kann durch Aufzählen ihrer Elemente beschrieben werden. So besteht beispielsweise die Menge $M = \{a, b, c, d\}$ genau aus den Elementen a, b, c und d.
Bei Mengen geht es ausschliesslich um die Frage, welche Elemente in ihr enthalten sind. Häufigkeit und Reihenfolge der Elemente spielen keine Rolle.
$\{b, b, a, c, d\}$ und $\{a, b, c, d, d, d\}$
beschreiben dieselbe Menge.
Wir verwenden Grossbuchstaben $A, B, C, \ldots$ um Mengen zu bezeichnen.

Annahme: Die Klasse der Objekte und die Klasse der Mengen sind disjunkt.
Dies bedeutet, dass Mengen keine Objekte sind. Somit kann eine Menge nicht Element einer (anderen) Menge sein. Für Mengen A und B ist also $A \in B$ nicht möglich. Wir treffen diese Annahme, damit wir uns nicht um Paradoxien der Mengenlehre kümmern müssen.

Beispiel:
Seien a, b, und c Objekte.

- Die leere Menge ist eine Menge, sie wird auch mit $\emptyset$ bezeichnet.
- $\{a, b\}$ ist eine Menge
- $\{a, (b, c)\}$ ist eine Menge.
- $\{a, \{b, c\}\}$ ist keine Menge (gemäss unserer Definition).

Definition 1.3: Gleichheit von Mengen

Zwei Mengen sind gleich, falls sie dieselben Elemente enthalten. Formal heisst das

$$A = B \text{ } g.d.w. \text{ } \forall x(x \in A \Leftrightarrow x \in B).$$

Statt A und B sind gleich sagen wir auch, A und B sind identisch.

Definition 1.4: Teilmengen

Eine Menge A heisst Teilmenge einer Menge B (wir schreiben dafür $A \subseteq B$), falls jedes Element von A auch ein Element von B ist. Das heisst

$$A \subseteq B \text{ } g.d.w. \text{ } \forall x(x \in A \Rightarrow x \in B).$$

A heisst *echte Teilmenge* von B (in Zeichen $A \subsetneq B$, falls

$$A \subseteq B \text{ } und \text{ } A \neq B.$$

Bemerkung:

Für zwei Mengen A und B gilt somit

$$A = B \text{ } g.d.w. \text{ } A \subseteq BundB \subseteq A.$$

Definition 1.5: Prädikate

Ein Prädikat $\varphi(x)$ beschreibt eine Eigenschaft, welche Objekten zu- oder abgesprochen werden kann. Der Ausdruck $\varphi(a)$ sagt, dass das Objekt a die durch $\varphi(x)$ beschriebene Eigenschaft hat. Wir sagen dann a erfüllt φ.

Beispiel:

- $\varphi(x) = x$ ist eine gerade Zahl
- $\varphi(x) = x$ so, dass $\exists y \in \mathbb{N}$ mit $x = 2y$
- $\varphi(x) = x$ ist rot

Komprehension
Annahme: Für jedes Prädikat $\varphi(x)$ gibt es eine Menge A, so dass für alle Objekte x gilt.

$$x \in A \text{ } g.d.w. \text{ } \varphi(x).$$

Wir verwenden folgende Schreibweise, um eine durch Komprehension gebildete Menge zu definieren

$$A := \{x \mid \varphi(x)\}$$

und sagen A ist die Menge von allen x, welche φ erfüllen.

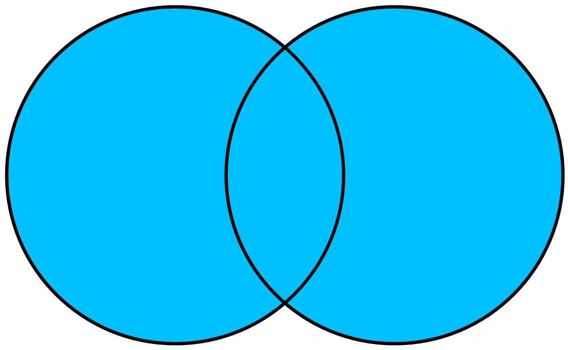
Beispiel:
Die Menge

$$A := \{x \mid \exists y \in \mathbb{N} \; mit \; x = 2y\}$$

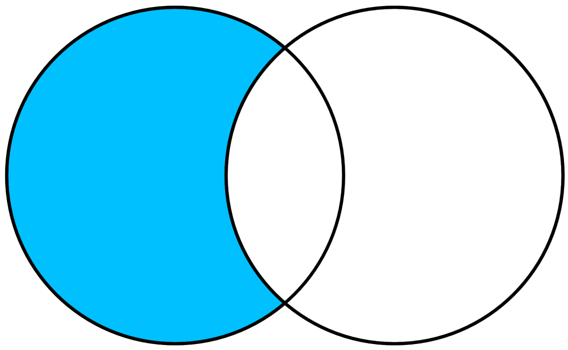
ist die Menge derjenigen x für die es eine natürliche Zahl y gibt mit $x = 2y$.
Das heisst, A ist die Menge der geraden natürlichen Zahlen.
Weniger formal: $A = \{0, 2, 4, 6, 8, 10, ...\}$

Definition 1.6: Operationen auf Mengen

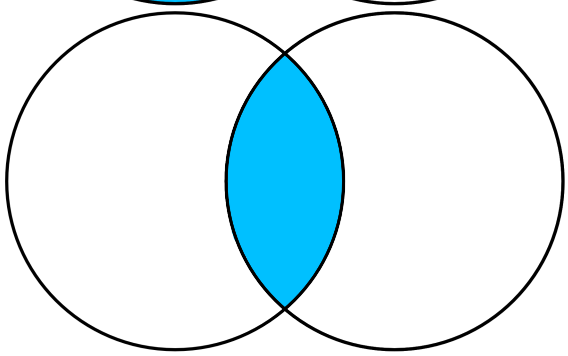
Vereinigung $x \in A \cup B$
g.d.w. $x \in A$ *oder* $x \in B$



Differenz $x \in A \setminus B$
g.d.w. $x \in A$ *und* $x \notin B$



Vereinigung $x \in A \cap B$
g.d.w. $x \in A$ *oder* $x \in B$



Bemerkung:

Die Vereinigung, die Differenz und der Schnitt zweier Mengen existieren (als neue Mengen), da sie durch das Schema der Komprehension gebildet werden können.

Definition 1.7: Relation

Eine Menge R heisst n-stellige (oder n-äre) Relation über Mengen $A_1, ..., A_n$, falls

$$R \subseteq \{(x_1, ..., x_n) \mid x_1 \in A_1 \textit{ und } \dots \textit{ und } x_n \in A_n\}.$$

Für eine n-stellige Relation R über Mengen $A_1, \dots, A_n$ gilt somit: Jedes Element von R ist ein n-Tupel $(x_1, \dots, x_n)$ mit $x_i \in A_i$ für alle $1 \leq i \leq n$.

Beispiel:
Seien a, b und c atomare Objekte.

- $\{a, b, c\}$ ist keine Relation.
- $\{(a), (b), (c)\}$ ist eine Relation.
- $\{(a, a, a), (b, c, a), (b, c, c)\}$ ist eine Relation.
- $\{(a, a), (a, b, c), (c, c)\}$ ist keine Relation.

Definition 1.8: Kartesisches Produkt

Eine Menge R heisst n-stellige (oder n-äre) Relation über Mengen $A_1, ..., A_n$, falls

$$R \subseteq \{(x_1, ..., x_n) \mid x_1 \in A_1 \textit{ und } \dots \textit{ und } x_n \in A_n\}.$$

Für eine n-stellige Relation R über Mengen $A_1, \dots, A_n$ gilt somit: Jedes Element von R ist ein n-Tupel $(x_1, \dots, x_n)$ mit $x_i \in A_i$ für alle $1 \leq i \leq n$.

Beispiel:
Sei

- R die 1-stellige Relation $R = \{(a), (b), (c)\}$ und
- S die 2-stellige Relation $S = \{(1, 5), (2, 6)\}$. Dann ist $R \times S$ die 3-stellige Relation

$$R \times S = (a, 1, 5), (a, 2, 6), (b, 1, 5), (b, 2, 6), (c, 1, 5), (c, 2, 6).$$

Bemerkung:

Unsere Definition nennt man auch flaches kartesisches Produkt. Das bedeutet, dass das kartesische Produkt einer m-stelligen mit einer n-stelligen Relation eine $(m + n)$-stellige Relation ist.

Bemerkung:

Besteht R aus h-vielen Elementen und S aus k-vielen Elementen, so enthält das kartesische Produkt $(h * k)$-viele Elemente.

Bemerkung:

Assoziativität
Seien R, S und T Relationen. Es gilt

$$(R \times S) \times T = R \times (S \times T).$$

Diese Eigenschaft erlaubt es uns, die Klammern wegzulassen und einfach $R \times S \times T$ zu schreiben.

Flaches Produkt vs. übliche mathematische Definition
üblicherweise wird in der mathematischen Mengenlehre das kartesische Produkt anders definiert, nämlich durch

$$R \times S := \{(a, b) \mid a \in R \ und \ b \in S\}.(1)$$

Damit ist $R \times S$ immer eine 2-stellige Relation.
Im Gegensatz zu unserem kartesischen Produkt erfüllt das Produkt aus (1) das Assoziativgesetzt nicht. Für R uns S aus dem vorherigen Beispiel finden wir dann

$$R \times S := \{((a), (1, 5)), ((a), (2, 6)), ((b), (1, 5)), ((b), (2, 6)), ((c), (1, 5)), ((c), (2, 6))\}.$$

Die Elemente aus $R \times S$ sind 2-Tupel (Paare) bestehend aus einem 1-Tupel und einem 2-Tupel.

2 Das Relationenmodell

Problem.
Wir nehmen an, wir wollen die Daten zu einer von Menge von Autos in unserer Datenbank halten. FÃ¼r jedes Auto soll dessen **Marke** und **Farbe** abgespeichert werden.

2.1 Attribute & DomÃ¤ne

Wir kÃ¶nnen jedes Auto als Paar $(Marke, Farbe)$ darstellen. Eine Menge von Autos entspricht somit einer Menge von solchen Paaren. Das heisst wir kÃ¶nnen eine Menge von Autos durch eine 2-stellige Relation reprÃ¤sentieren.

$$Autos := \{(Opel, silber), (VW, rot), (Audi, schwarz)\}.$$

Wir kÃ¶nnen diese Relation als Tabelle mit zwei Spalten darstellen:

Autos

Marke	*Farbe*
Opel	*silber*
VW	*rot*
Audi	*schwarz*

wobei

Attribut Name der Spalte $(Marke, Farbe)$

DomÃ¤ne MÃ¶gliche Werte, die ein Attribut annehmen kann

2.2 Null Werte

$Null$ wird verwendet, wenn der Wert eines Attributs unbekannt ist,

weil wir den Wert nicht kennen oder

oder, weil das Attribut keinen Wert besitzt.

Konvention

$Null$ gehÃ¶rt zu jeder DomÃ¤ne.

in der Tabellenform schreiben wir hÃ¤ufig âˆ’ anstelle von $Null$.

Bemerkung:

Semantik von Null
Da wir *Null* verwenden, um auszudrÃ¼cken, dass der Wert eines Attributs unbekannt ist, erhÃ¤lt Null eine spezielle Semantik bezÃ¼glich der Gleichheit:

$$Es\ gilt\ \textbf{nicht},\ dass\ Null = Null. \quad (1)$$

Wenn wir zwei unbekannte Werte vergleichen, so wissen wir eben nicht, ob sie gleich sind. Deshalb verwenden wir eine Semantik fÃ¼r die (1) der Fall ist.

Definition 2.1: Schwache Gleichheitsrelation

An einigen Stellen werden wir zwei *Null* Werte als gleichwertig betrachten mÃ¼ssen. Dazu fÃ¼hren wir folgende schwache Gleichheitsrelation zwischen zwei atomaren Objekten a und b ein:

$$a \simeq b\ g.d.w.\ a = b\ oder\ (a\ ist\ Null\ und\ b\ ist\ Null).$$

FÃ¼r zwei n-Tupel definieren wir analog:

$$(a_1, \dots, a_n) \simeq (b_1, \dots, bn)\ g.d.w.\ a_i \simeq b_i\ f\ddot{u}r\ alle\ 1 \leq i \leq n.$$

Beispiel:
Seien a, b, und c paarweise verschiedene Objekte, die verschieden von *Null* sind.

- $a \simeq a$
- $a \not\simeq a$
- $a \not\simeq Null$
- $Null \simeq Null$
- $(a, b) \simeq (a, b)$
- $(a, Null, c) \simeq (a, Null, c)$
- $(a, Null, c) \not\simeq (a, b, c)$

2.3 Relationsschema & Relationale Datenbank

Relationenschemata (oder einfach nur Schemata) spezifizieren die bei Relationen verwendeten Attribute und DomÃ¤nen. Es handelt sich dabei um Sequenzen der Form

$$(A_1 : D_1, \dots, A_n : D_n),$$

wobei $A_1, \dots, A_n$ Attribute mit den jeweiligen DomÃ¤nen $D1, \dots, Dn$ sind.

Beispiel:
$(Marke : text, Baujahr : integer, Farbe : color_enum, Fahrer : text)$

Konvention

- Wir schreiben manchmal

  $$(A_1, \dots, A_n) \textit{ anstelle von } (A_1 : D_1, \dots, A_n : D_n),$$

  wenn sich die DomÃ¤nen unmittelbar aus dem Kontext ergeben oder unwichtig sind.
- Wir sagen R ist eine Relation Ã¼ber $A_1, \dots, A_n$, wenn R eine Relation Ã¼ber den dazugehÃ¶renden DomÃ¤nen $D_1, \dots, D_n$ ist.
- Wir sagen dann auch R ist eine Instanz des Schemas $(A_1, \dots, A_n)$.

Bemerkung:

Als *relationales Datenbank-Schema* (oder kurz *DB-Schema*) bezeichnen wir die Menge aller verwendeten Relationenschemata.

Definition 2.2: Relationale Datenbank

Als *relationale Datenbank* (oder kurz *relationale DB*) bezeichnen wir das verwendete relationale Datenbank-Schema zusammen mit den momentanen Werten der Relationen.
Eine relationale Datenbank besteht somit aus

- einem DB-Schema und
- den aktuellen Instanzen aller Schemata des DB-Schemas.

Wir sprechen in diesem Zusammenhang auch von einer *Instanz* eines DB-Schemas und meinen damit die Menge der aktellen Instanzen aller Schemata des DB-Schemas.

Definition 2.3: Projektion auf Attribute

Es seien $A_1, \dots, A_n$ Attribute,

$$S = (A_1, \dots, A_n)$$

ein Relationenschema und R eine Instanz von S.

- Ist t ein n-Tupel, das zu R gehÃ¶rt, so schreiben wir $t[A_i]$ fÃ¼r den Wert von t bei Attribut A_i. FÃ¼r $(a_1, \dots, a_i, \dots, a_n) \in R$ heisst das $(a_1, \dots, a_i, \dots, a_n)[A_i] = a_u$.
- Ist $K = (A_{i_1}, \dots, A_{i_m})$ eine Sequenz von Attributen, so definieren wir fÃ¼r ein n-

Tupel $(a_1, \ldots, a_i, \ldots, a_n) \in R$ $(a_1, \ldots, a_i, \ldots, a_n)[K] := (a_{i_1}, \ldots, a_{i_m})$.

Beispiel:

Betrachten wir die Relation

Autos

Marke	Farbe	Baujahr	Vorname	Nachname
Opel	*silber*	2010	*Tom*	*Studer*
Opel	*schwarz*	2010	*Eva*	*Studer*
VW	*rot*	2014	*EvaStuder*	
Audi	*schwarz*	2014	*EvaMeier*	

Es sei nun

$$t := (Opel, schwarz, 2010, Eva, Studer).$$

Damit gilt

$$t[(Marke, Farbe)] = (Opel, schwarz)$$

und

$$t[(Nachname, Baujahr)] = (Studer, 2010).$$

Bemerkung:

FÃ¼r $s, t \in R$ bedeutet $s[K] = t[K]$, dass die Werte von s und t in allen Attributen aus K Ã¼bereinstimmen.

2.4 SchlÃ¼ssel

Problem.

Wie kÃ¶nnen wir in einer Instanz R von S die einzelnen Elemente unterscheiden?

Dazu wÃ¤hlen wir einen sogenannten *PrimÃ¤rschlÃ¼ssel*. Dies ist eine Sequenz von Attributen

$$K = (A_{i_1}, \ldots, A_{i_m}).$$

Dann verlangen wir fÃ¼r alle Instanzen R von S und alle $s, t \in R$, dass $s[K] = t[K] \Rightarrow s \simeq t$.

Konvention

Wir geben den PrimÃ¤rschlÃ¼ssel an, indem wir beim Relationenschema diejenigen Attribute unterstreichen, welche zum gewÃ¤hlten PrimÃ¤rschlÃ¼ssel gehÃ¶ren.

Beispiel:
Betrachten wir die Relation

Autos

Marke	Farbe	Baujahr	Vorname	Nachname
Opel	*silber*	2010	*Tom*	*Studer*
Opel	*schwarz*	2010	*Eva*	*Studer*
VW	*rot*	2014	*Eva*	*Studer*
Audi	*schwarz*	2014	*Eva*	*Meier*

Es sind bspw. die beiden folgenden PrimÃ¤rschlÃ¼ssel mÃ¶glich:

$$(Marke, Farbe) \text{ und } (Baujahr, Vorname, Nachname). \quad (3)$$

Falls wir $(Marke, Farbe)$ als PrimÃ¤rschlÃ¼ssel wÃ¤hlen, so geben wir das Schema wie folgt an:

$$(\underline{Marke}, \underline{Farbe}, Baujahr, Vorname, Nachname).$$

Sprechende SchlÃ¼ssel

In einer echten Datenbankanwendung sind wahrscheinlich beide mÃ¶glichen PrimÃ¤rschlÃ¼ssel aus (3) ungeeignet. Es ist nÃ¤mlich gut mÃ¶glich, dass wir spÃ¤ter dieser Relation weitere Autos hinzufÃ¼gen mÃ¶chten. Da kann es dann sein, dass ein zweiter roter VW eingefÃ¼gt werden soll oder ein weiteres Auto mit Baujahr 2010 und dem Fahrer Tom Studer.
In der Praxis wird oft ein zusÃ¤tzliches Attribut, z.B. *AutoId*, hinzugefÃ¼gt, welches als PrimÃ¤rschlÃ¼ssel dient. Dieses Attribut hat nur den Zweck, die verschiedenen Elemente der Relation eindeutig zu bestimmen. Es beschreibt aber keine echte Eigenschaft von Autos. Wir nennen einen solchen PrimÃ¤rschlÃ¼ssel einen *nicht-sprechenden* SchlÃ¼ssel.
Ein *sprechender* SchlÃ¼ssel hingegen hat eine logische Beziehung zu einem oder mehreren Attributen des Schemas.
Es ist gute Praxis *keine* sprechenden SchlÃ¼ssel zu verwenden, da diese die Tendenz haben zu zerbrechen.
Das heisst, frÃ¼her oder spÃ¤ter wird eine Situation auftreten, in der ein neues Tupel eingefÃ¼gt werden soll, dessen sprechender SchlÃ¼ssel bereits ein Tupel in der Relation bezeichnet. Oder es kann sein, dass das System der SchlÃ¼sselgenerierung komplett geÃ¤ndert wird. Beispiele dazu sind:

- das System der AHV-Nummern (eindeutige Personennummer der Alters- und Hinterlassenenversicherung) in der Schweiz, welches 2008 geÃ¤ndert wurde,
- die Internationale Standardbuchnummer (ISBN), fÃ¼r die 2005 ein revidierter Standard eingefÃ¼hrt wurde.

Problem.

Autos

Marke	Farbe	Baujahr	Vorname	Nachname
Opel	*silber*	2010	*Tom*	*Studer*
Opel	*schwarz*	2010	*Eva*	*Studer*
VW	*rot*	2014	*Eva*	*Studer*
Audi	*schwarz*	2014	*Eva*	*Meier*

In dieser Tabelle sind die Daten zu Eva Studer doppelt abgespeichert.

Besser: zwei Zabellen

Autos

Marke	Farbe	Baujahr	FahrerId
Opel	*silber*	2010	1
Opel	*schwarz*	2010	2
VW	*rot*	2014	2
Audi	*schwarz*	2014	3

Personen

PersId	Vorname	Nachname
1	*Tom*	*Studer*
2	*Eva*	*Studer*
3	*Eva*	*Meier*

FahrerId *referenziert* die Tabelle Personen
FahrerId ist ein *FremdschlÃ¼ssel* fÃ¼r Personen FOLIE 25