Relationales Datenmodell

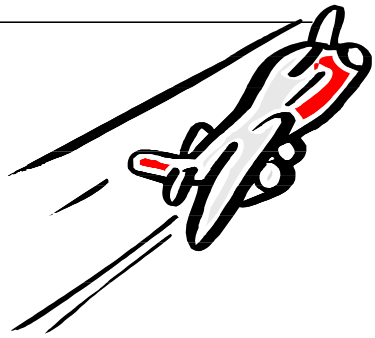
Ein **Datenmodell** hat zwei Bestandteile:

- Eine **mathematische Notation** zur Darstellung von Daten und Beziehungen.
- **Operationen** auf den Daten, um Abfragen und andere Manipulationen zu ermöglichen.

Dem **Relationen Datenmodell** (E.F. Codd, 1970) liegt die mengentheoretische Relation zugrunde.

Beispiele für typische relationale Datenbanken:

- SYSTEM R (IBM), DB/2 (IBM)
- INGRES (Univ. of California, Berkley)
- ORACLE, UNIFY, INFORMIX, SYBASE



Ein einführendes Beispiel:

Anzeigetafel an einem Flughafen:

FLUGNR	VON	NACH	ABFLUG	ANKUNFT
083	JFK	O'Hare	11:30a	1:43p
084	O'Hare	JFK	3:00p	5:55p
109	JFK	Los Ange- les	9:50p	2:25a
213	JFK	Boston	11:43a	12:45p
214	Boston	JFK	2:20p	3:12p

Typische Fragen an einem Flughafen:

- *Gibt es Flüge von New York nach Boston?*
- *Wann fliegt eine Maschine von New York nach Boston?*
- *Welche Ziele werden, startend in Frankfurt, angeflo- gen?*
- *Wann kommt der Flug mit Nummer 213 in Boston an?*

Formalisierung des Relationenmodells

Definition:

Ein ***Relationenschema R*** ist eine endliche Menge von ***Attributnamen*** $\{A_1, A_2, \ldots, A_n\}$.

Notation:

$R = \{A_1, A_2, \ldots, A_n\}$ ***oder*** $R(A_1, A_2, \ldots, A_n)$

Attributnamen können auch verkürzt nur als ***Attribute*** bezeichnet werden.

Definition:

Zu jedem Attribut A_i, $1 \leq i \leq n$, gibt es eine Menge D_i, den ***Wertebereich*** (domain) von A_i.

Notation:

$dom(A_i)$ ist der Wertebereich von A_i.

Beispiel:

Das Attribut **GESCHLECHT** hat den Wertebereich ***dom(GESCHLECHT) = {männlich, weiblich}***.

Definition:

Sei $D = D_1 \times D_2 \times D_3 \times \ldots \times D_n$ das kartesische Produkt der Domänen $D_1, D_2, D_3, \ldots, D_n$.

Eine **Relation** ***r*** auf einem **Relationenschema** ***R***, bezeichnet mit ***r(R)***, ist eine endliche Menge von Abbildungen $\{t_1, \ldots, t_n\}$ von ***R*** nach ***D***, wobei für jede Abbildung $t \in r$, der Wert $t(A_i)$ aus der Domäne D_i, $1 \leq i \leq n$, stammt.

Diese Abbildungen werden ***Tupel*** genannt. Der Wert eines Tupels ***t*** für ein Attribut ***A***, $t(A) = a$, heißt ***A-Wert*** von ***t***.

Ein Tupel $(v_1, v_2, \ldots, v_k)$ hat ***k*** **Komponenten**,
die ***i***-te Komponente ist dabei v_i .

Notation:

Eine Relation kann als **Tabelle** visualisiert werden:

- Die **Spalten** repräsentieren die **Attribute**.
- Die **Zeilen** repräsentieren die **Tupel**.

	Attribut 1	***Attribut 2***	***Attribut 3***
Tupel 1 ➔			
Tupel 2 ➔			
Tupel 3 ➔			

Beispiel: Flughafen (1)

Mögliches **Relationenschema:**

FLÜGE(FLUGNR, VON, NACH, ABFLUG, ANKUNFT)

Entsprechende **Tabelle:**

FLUGNR	VON	NACH	ABFLUG	ANKUNFT
083	JFK	O'Hare	11:30a	1:43p
084	O'Hare	JFK	3:00p	5:55p
109	JFK	Los Ange-les	9:50p	2:25a
213	JFK	Boston	11:43a	12:45p
214	Boston	JFK	2:20p	3:12p

Beispiel: Flughafen (2)

Bezüglich der **Domänen** gilt:

- ***dom (FLUGNR)*** =
  „Menge aller dreistelligen Dezimalzahlen“
- ***dom (VON) = dom (NACH) =***
  {JFK, O'Hare, Los Angeles, Boston, Atlanta}
- ***dom (ABFLUG) = dom (ANKUNFT) =***
  „Menge der Zeiten eines Tages“

Die Tabelle hat ***5* Tupel**. Eines davon ist ***t*** mit

- ***t(FLUGNR) = 084***
- ***t(VON) = O'Hare***
- ***t(NACH) = JFK***
- ***t(ABFLUG) = 3:00p***
- ***t(ANKUNFT) = 5:55p***

Konventionen und Notationen:

- Für **Attribute** werden **Großbuchstaben** vom **Anfang** des Alphabets verwendet, z.B. $\boldsymbol{A_i}$, ***B, C.***
- Für **Relationenschemata** werden **Großbuchstaben** vom **Ende** des Alphabets verwendet, z.B. $\boldsymbol{R_i}$, ***S, T, X***.
- Für **Relationen** werden **Kleinbuchstaben** verwendet, z.B. $\boldsymbol{r_i}$, ***s, t, x***

Natürlich werden in der Praxis primär Bezeichner benutzt, die der Anwendung entsprechen, z.B. „FLUGNR“, „STUDENTIN“, „NAME“ etc.

Bemerkungen:

- Relationen sind **Abstraktionen** von Teilen der realen Welt.
- **Relationen** sind **veränderlich**, sie ändern ihren Zustand in der Zeit
  ➔ Einfügen, Löschen, Ändern von **Tupeln**
- **Relationenschemata** sind **unveränderlich**
- Sind den Spalten einer Relation **Attributnamen** zugeordnet, so ist deren **Reihenfolge unwichtig**.
  (In der Definition der Relation auf S. 5 ist die Reihenfolge der Spalten wichtig.)

Beispiel:

Zwei Repräsentationen der gleichen Relation

FLUGNR	VON	NACH	ABFLUG	ANKUNFT
083	JFK	O'Hare	11:30a	1:43p
084	O'Hare	JFK	3:00p	5:55p
109	JFK	Los Ange-les	9:50p	2:25a
213	JFK	Boston	11:43a	12:45p
214	Boston	JFK	2:20p	3:12p

FLUGNR	ABFLUG	ANKUNFT	VON	NACH
083	11:30a	1:43p	JFK	O'Hare
084	3:00p	5:55p	O'Hare	JFK
109	9:50p	2:25a	JFK	Los Angeles
213	11:43a	12:45p	JFK	Boston
214	2:20p	3:12p	Boston	JFK

Definition:

Sei ***X*** eine Teilmenge von ***R***.

Unter der **Einengung** der Abbildung ***t*** auf die Attributmenge ***X***, genannt ***t(X)***, versteht man denjenigen ***X-Wert***, der jedem ***A*** $\in$ ***X*** genau die Werte aus ***dom(A)*** zuordnet, wie die Abbildung ***t***.

Beispiel:

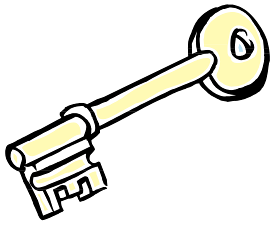
Gegeben sei das Tupel ***t*** mit

- ***t(FLUGNR) = 084***
- ***t(VON) = O'Hare***
- ***t(NACH) = JFK***
- ***t(Abflug) = 3:00p***
- ***t(ANKUNFT) = 5:55p***

Der ***VON-Wert*** von ***t*** ist ***t(VON) = O'Hare***.

Der **{VON, NACH}-Wert** von ***t*** ist das Tupel ***t'***, definiert als ***t'(VON) = O'Hare, t'(NACH) = JFK***.

Definition:



Ein ***Schlüssel*** (key) einer Relation ***r(R)*** ist eine Teilmenge ***K*** von ***R***, so dass für je zwei **verschiedene** Tupel $t_1, t_2 \in r$ gilt:

- $t_1(K) \neq t_2(K)$ und
- keine **echte** Teilmenge K' von K hat diese Eigenschaft.

K ist ein ***Oberschlüssel*** (super key) der Relation, falls ***K*** einen Schlüssel enthält.

Beispiel: Flughafen

FLÜGE

FLUGNR	VON	NACH	ABFLUG	ANKUNFT
083	JFK	O'Hare	11:30a	1:43p
084	O'Hare	JFK	3:00p	5:55p
109	JFK	Los Angeles	9:50p	2:25a
213	JFK	Boston	11:43a	12:45p
214	Boston	JFK	2:20p	3:12p

- ***{FLUGNR}*** und ***{VON, NACH}*** sind mögliche Schlüssel.
- ***{VON}*** alleine ist (wg. JFK) kein Schlüssel.
- ***{FLUGNR, VON}*** ist Oberschlüssel, aber kein Schlüssel, weil bereits ***{FLUGNR}*** Schlüssel ist.

Definition:

Schlüssel, die explizit zu einem Relationenschema angeführt sind, heißen **ausgezeichnete Schlüssel** (designated keys).

Im Allgemeinen wird **ein** Schlüssel als **Primärschlüssel** ausgezeichnet.

Dieser wird im Relationenschema durch **Unterstreichen** gekennzeichnet.

Beispiel: Flughafen

FLÜGE ($\underline{\text{FLUGNR}}$, VON, NACH, ABFLUG, ANKUNFT)

Schlüssel

ER-Abbildung zu Relationen

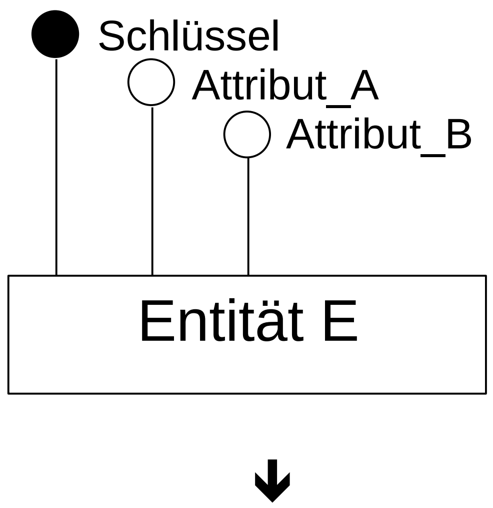
Motivation

- Das Entity-Relationship-Diagramm ist ein graphisches Modell, welches Zusammenhänge anschaulich macht.
- Das Entity-Relationship-Diagramm ist aber auf der **konzeptionellen** Ebene angesiedelt und daher nicht direkt zu verwenden.
- Es ist jedoch eine Abbildung in die verschiedenartigsten logischen Ebenen möglich, wie etwa dem *Relationalen Modell*, dem *Netzwerk-Modell* oder auch dem *Objektorientierten Modell.*
- Die derzeit am häufigsten anzutreffenden DBMS sind *Relationale* Datenbankmanagementsysteme (RDBMS), deshalb soll im Rahmen der Vorlesung die Abbildung eines ER-Diagramms in die relationalen Tabellen betrachtet werden.
- Diese Abbildung ist nicht eindeutig, womit es durchaus unterschiedliche relationale Modelle geben kann, die dieselben Sachverhalte ausdrücken.

Entitäten

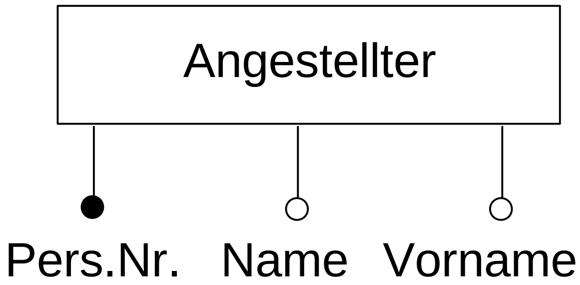
- Jede Entitätsmenge wird zu einer Relation (Tabelle), dessen Relationenschema aus allen Attributen der Entitätsmenge besteht.
- Jedes Tupel der Tabelle entspricht dann genau einer Entität in der Entitätsmenge.
- Etwaige Schlüssel werden übernommen und üblicherweise an den Anfang des Relationenschemas gestellt.

„Regel“:



E (<u>Schlüssel</u>, Attribut_A, Attribut_B)

Beispiel

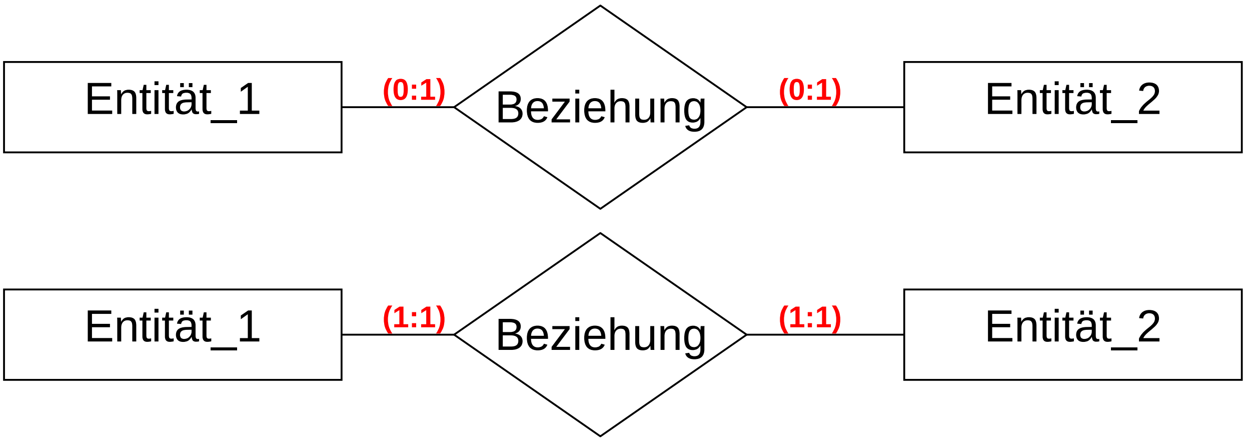


↓

ANGESTELLTER (<u>Pers.Nr.</u>, Name, Vorname)

Beziehungen

one-to-one



many-to-one

one-to-many

many-to-many

- Beziehungen (insbesondere alle many-to-many Beziehungen) werden ebenfalls zu einer Relation (Tabelle).

- Diese Relation umfasst, neben den Beziehungsattributen, die Schlüssel der an der Beziehung beteiligten Entitätsmengen.

- Der Schlüssel der Relation ist die Kombination der Schlüssel aller beteiligten Entitätsmengen.

- Eventuell ist eine Umbenennung (renaming) der Schlüsselattribute notwendig.

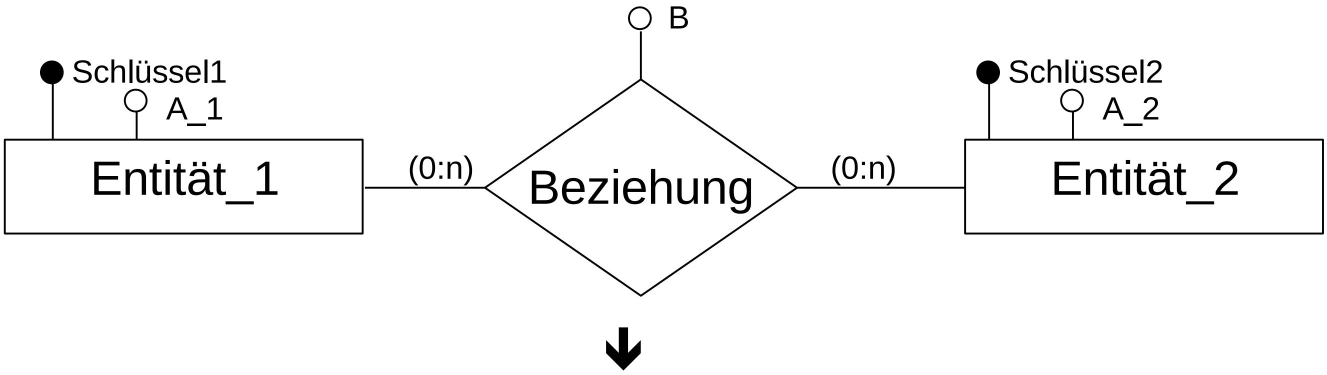
- Diese Art der Abbildung lässt sich auch für one-to-one bzw. one-to-many Beziehungen anwenden, allerdings entstehen dann Oberschlüssel.

- Für one-to-one, one-to-many und einige spezielle Beziehungen gibt es weitere Möglichkeiten der Abbildung (siehe später).

Many-to-many

„Regel“:

(Angegebene Attribute sind immer als Stellvertreter für eventuelle Attributsmengen zu sehen!)

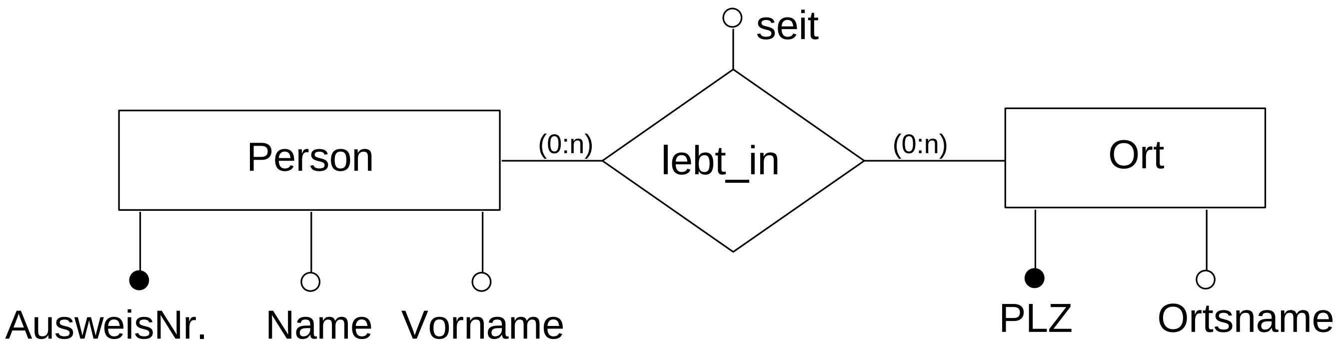


ENTITÄT_1 (<u>Schlüssel1</u>, A_1)

ENTITÄT_2 (<u>Schlüssel2</u>, A_2)

BEZIEHUNG (<u>Schlüssel1</u>, <u>Schlüssel2</u>, B)

Beispiel



↓

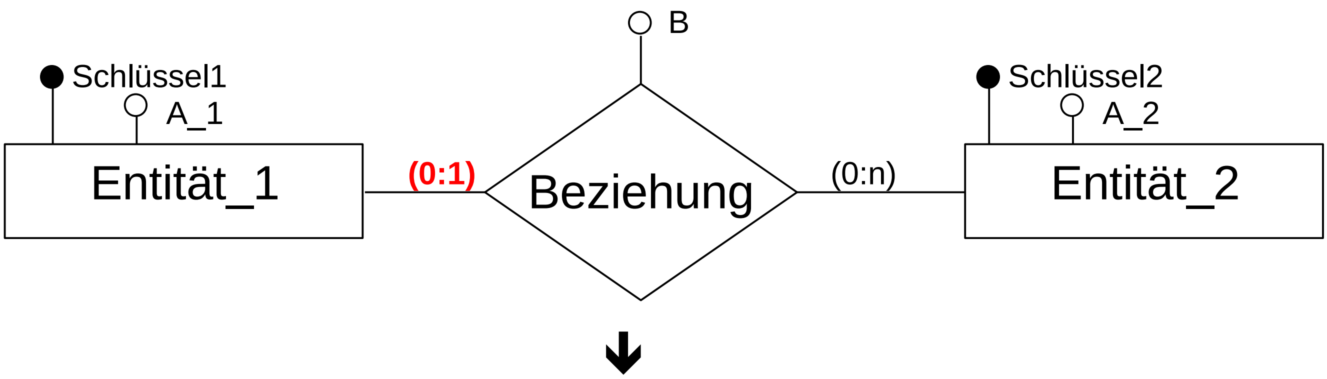
PERSON	(<u>AusweisNr.</u>, Name, Vorname)
ORT	(<u>PLZ</u>, Ortsname)
LEBT_IN	(<u>AusweisNr.</u>, <u>PLZ</u>, seit)

Many-to-one bzw. one-to-many

Bei Many-to-one bzw. One-to-many Beziehungen kann unter Umständen ein anderer Weg eingeschlagen werden.

- Ist die min-Kardinalität = 0, so muss das allgemeine Verfahren angewendet werden. Allerdings ist der Schlüssel der entstehenden Relation nun keine Kombination der Schlüssel der beteiligen Entitäten mehr.

„Regel“:

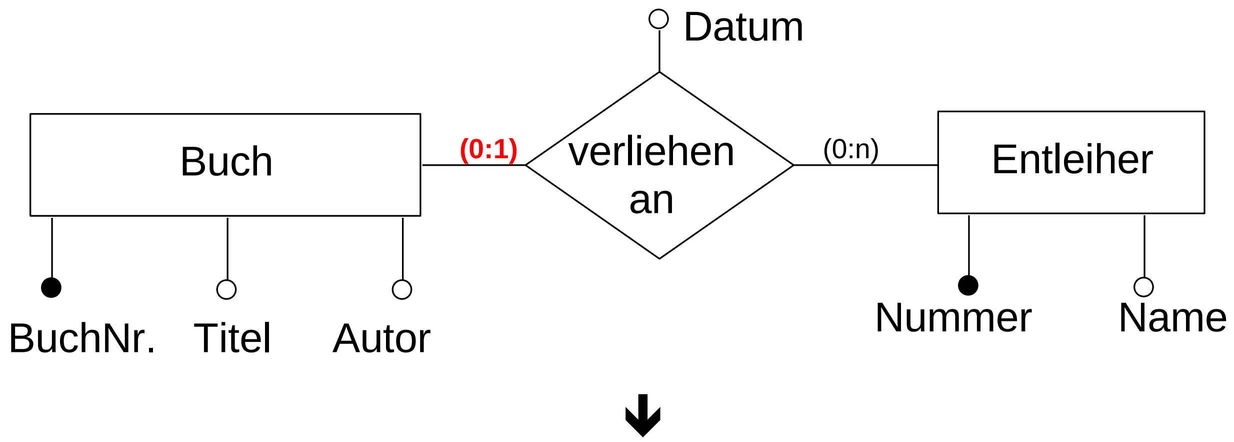


ENTITÄT_1 (<u>Schlüssel1</u>, A_1)

ENTITÄT_2 (<u>Schlüssel2</u>, A_2)

BEZIEHUNG (<u>Schlüssel1</u>, Schlüssel2, B)

Beispiel



BUCH (<u>BuchNr.</u>, Titel, Autor)

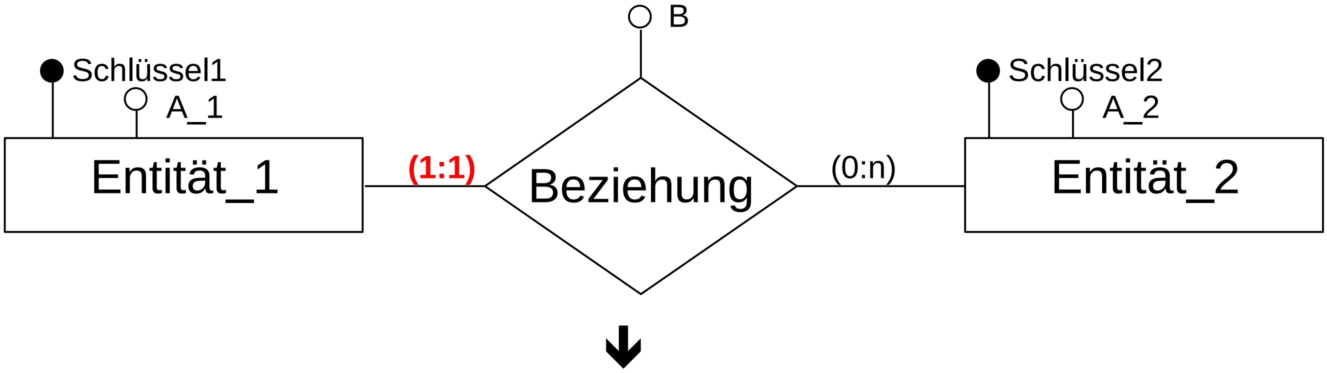
ENTLEIHER (<u>Nummer</u>, Name)

VERLIEHEN_AN (<u>BuchNr.</u>, Nummer, Datum)

- Ist die min-Kardinalität = 1, besteht also immer eine Beziehung, so werden der Relation der Entitätsmenge, die nur mit einer Entität der anderen Entitätsmenge verknüpft wird, einfach der Schlüssel dieser Entitätsmenge und die Beziehungsattribute hinzugefügt.

Hier sind nur noch zwei Relationen notwendig!

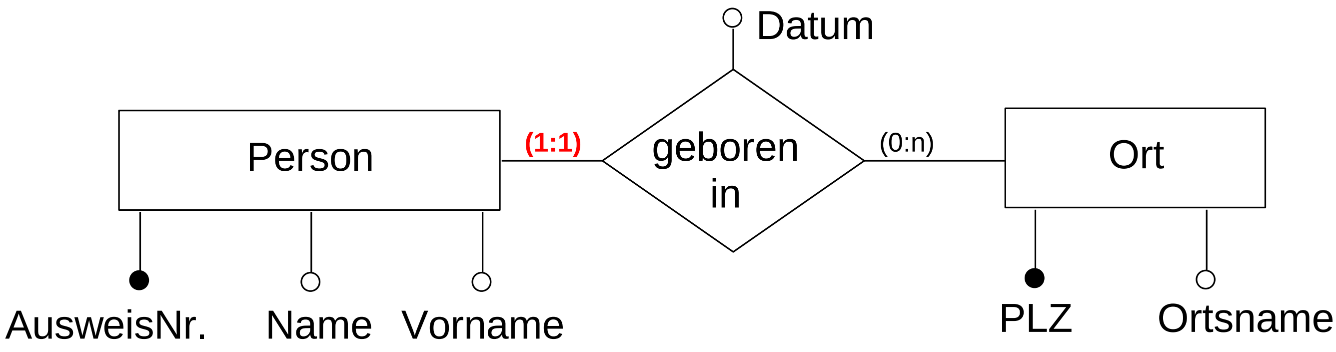
„Regel“:



ENTITÄT_1 (Schlüssel1, A_1, Schlüssel2 B)

ENTITÄT_2 (Schlüssel2, A_2)

Beispiel



↓

PERSON (<u>AusweisNr.</u>, Name, Vorname, PLZ, Datum)

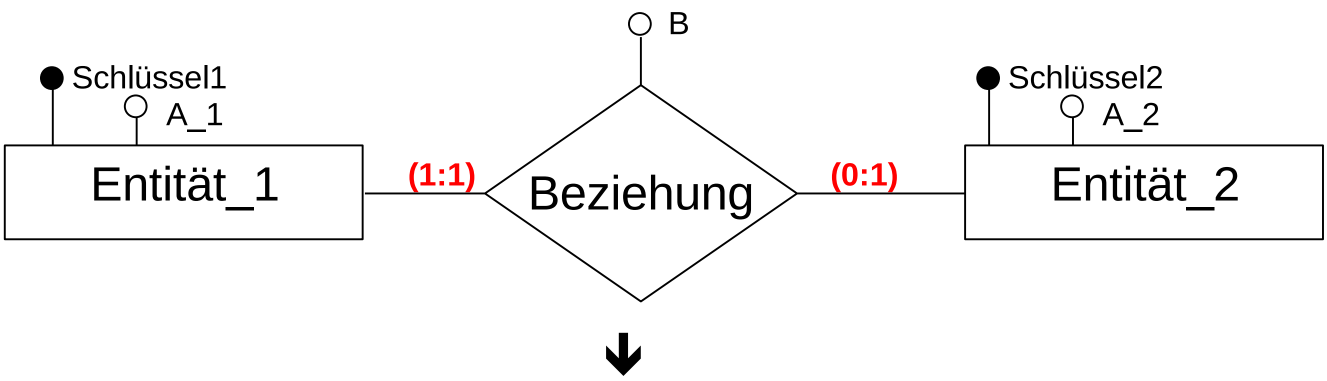
ORT (<u>PLZ</u>, Ortsname)

One-to-one

Eine **one-to-one** Beziehung kann wie eine one-to-many Beziehung in beide Richtungen betrachtet werden.

- Sind beide min-Kardinalitäten = 0, so muss das allgemeine Verfahren angewendet werden.
- Ist nur eine min-Kardinalität = 1, so wendet man die Abbildung der one-to-many Beziehung an.

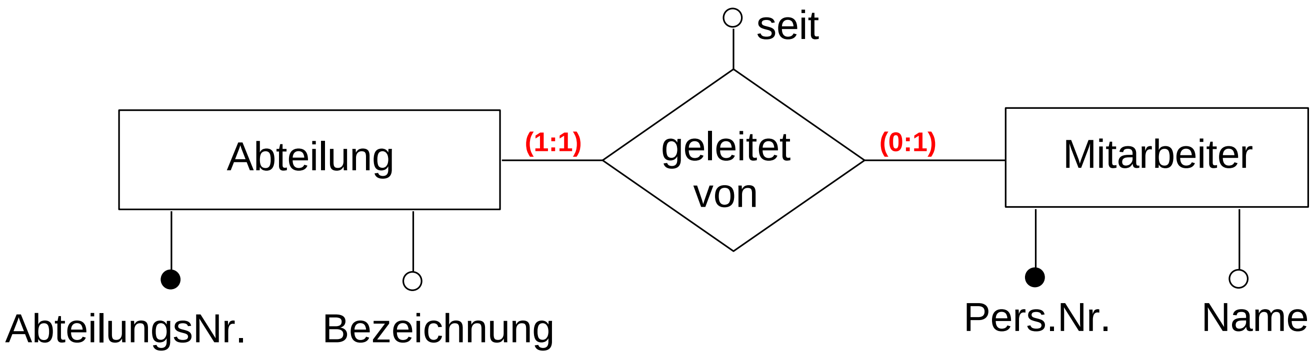
„Regel“:



ENTITÄT_1 (Schlüssel1, A_1, Schlüssel2, B)

ENTITÄT_2 (Schlüssel2, A_2)

Beispiel



↓

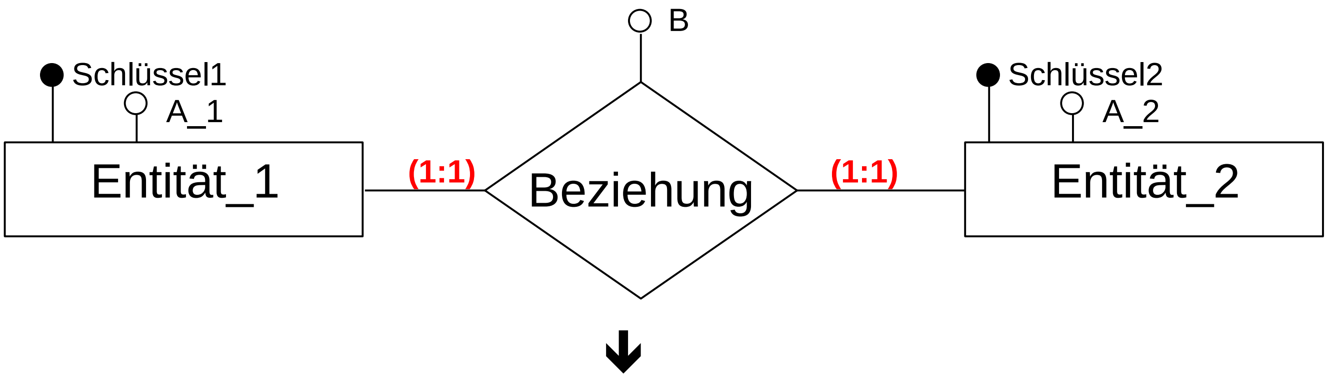
ABTEILUNG (<u>AbteilungsNr.</u>, Bezeichnung, Pers.Nr., seit)

MITARBEITER (<u>Pers.Nr.</u>, Name)

- Sind beide min-Kardinalitäten = 1, so werden in der Regel beide Entitätsmengen zusammengefasst. Das resultierende Relationenschema ist die Vereinigung der Attribute der Entitätsmengen und der Beziehungsattribute.

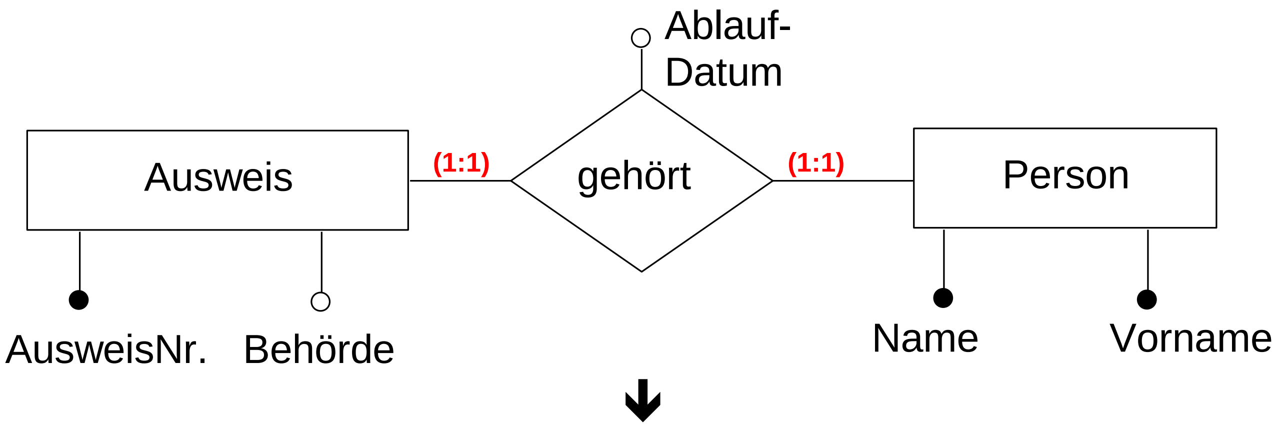
Es gibt nur noch eine Relation!

„Regel“:



ENTITÄT_1_2 (<u>Schlüssel1</u>, A_1, Schlüssel2, A_2, B)

Beispiel



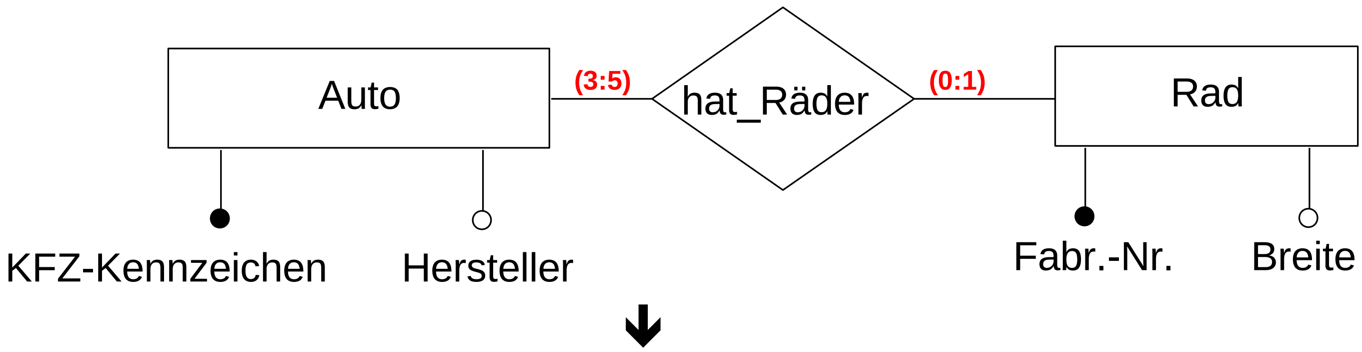
PERSON (<u>AusweisNr.</u>, Behörde, Ablaufdatum, Name, Vorname)

Weitere Sonderfälle

Bestimmte min/max-Kardinalitäten lassen sich auch direkt in der resultierenden Relation darstellen.

Die nicht existenten Beziehungen werden durch Nullwerte aufgeführt.

Beispiel



(Hier sind RAD1 – RAD3 verbindlich, also NOT NULL, während RAD4 und RAD5 durchaus Nullwerte beinhalten dürfen.)

AUTO (<u>KFZ-Kennzeichen</u>, Hersteller, RAD1, ... RAD5)

RAD (<u>Fabr.-Nr.</u>, Breite)

Abbildung der Generalisierung

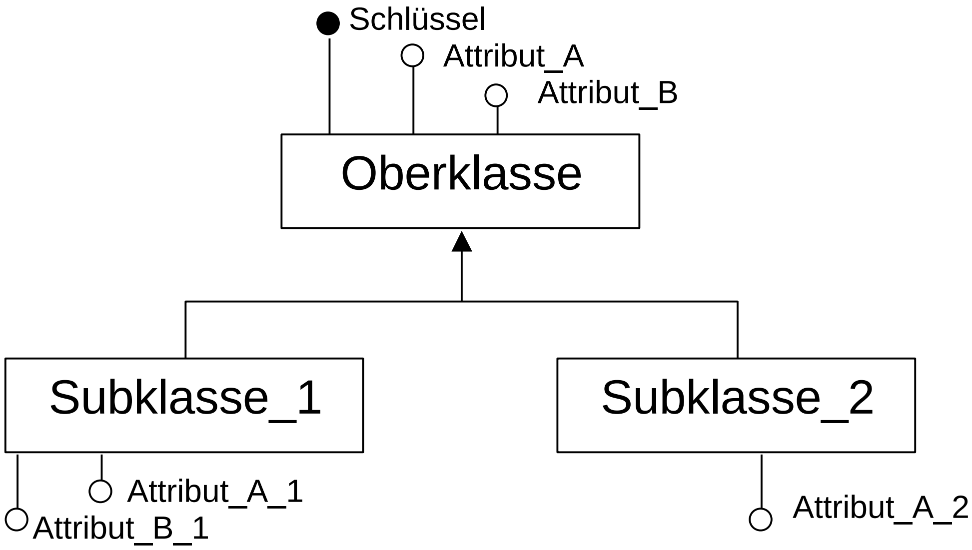
Es gibt **drei Möglichkeiten** Generalisierungen auf Tabellen abzubilden.

1. Möglichkeit

Bei dieser weniger häufig anzutreffenden Alternative werden alle Entitätsmengen zu eigenständigen Relationen, die alle für sie relevanten Informationen beinhalten. Die Subklassen enthalten neben ihren neuen Attributen noch alle Attribute ihrer Oberklasse.

Der Vorteil der Performance überwiegt nur in wenigen Fällen gegenüber der entstehenden Redundanz, deren Gefahren der Inkonsistenz und des zusätzlichen Speicherbedarfs.

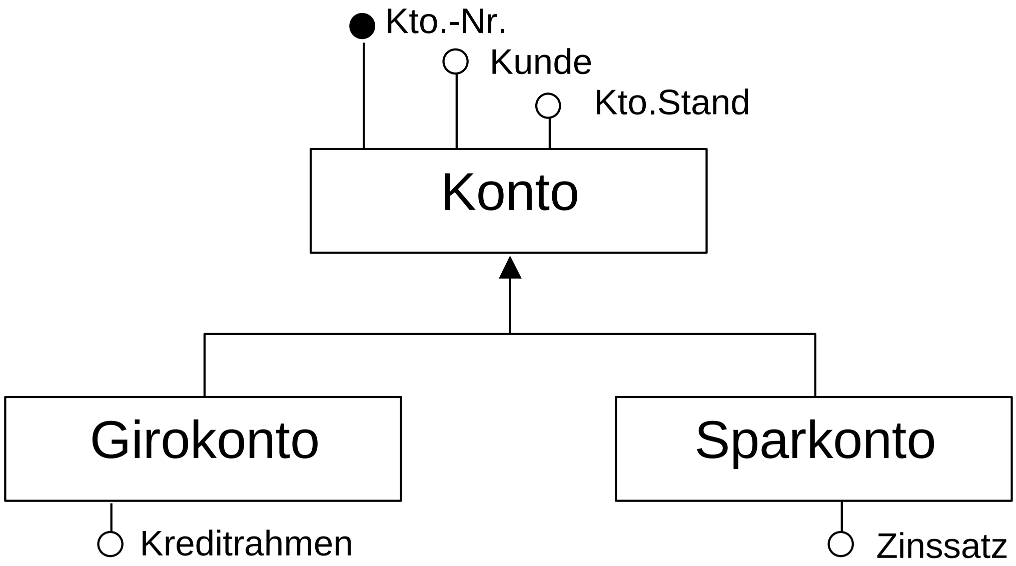
„Regel“:



⬇

OBERKLASSE	(<u>Schlüssel</u>, Attribut_A, Attribut_B)
SUBKLASSE_1	(<u>Schlüssel</u>, Attribut_A, Attribut_B, Attribut_A_1, Attribut_B_1)
SUBKLASSE_2	(<u>Schlüssel</u>, Attribut_A, Attribut_B, Attribut_A_2)

Beispiel



⬇

KONTO (<u>Kto.Nr.</u>, Kunde, Kto.Stand)

GIROKONTO (<u>Kto.Nr.</u>, Kunde, Kto.Stand, Kreditrahmen)

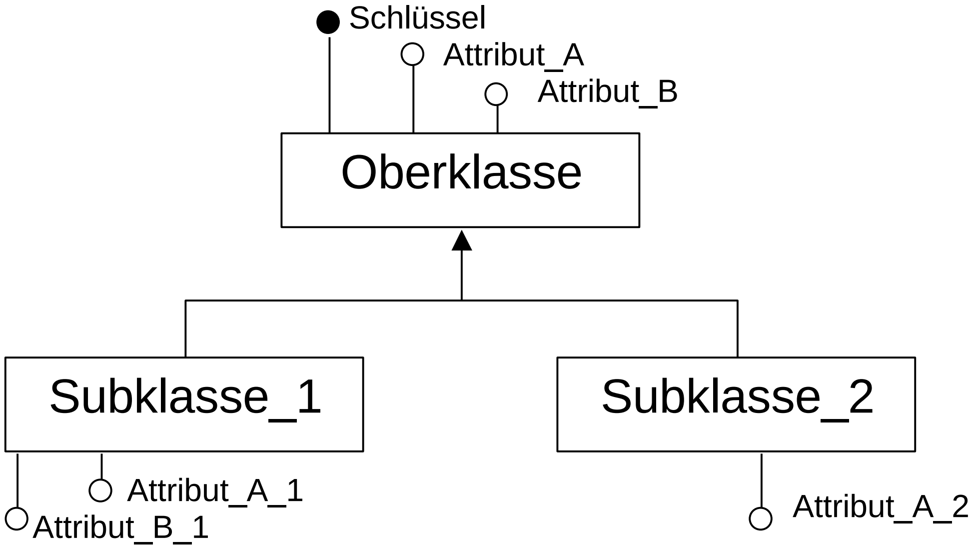
SPARKONTO (<u>Kto.Nr.</u>, Kunde, Kto.Stand, Zinssatz)

2. Möglichkeit

Bei der zweiten (gebräuchlicheren) Alternative wird die Generalisierung wie eine normale Beziehung abgebildet.

Die Entitätsmengen, die die Subklassen bilden, beinhalten in ihrer entstehenden Relation neben ihren eigenen Attributen nur noch den Schlüssel der Oberklassen-Entitätsmenge. Damit lassen sich alle Daten einer Subklassen-Entität durch einen natürlichen Verbund (natural join) gewinnen.

„Regel“:



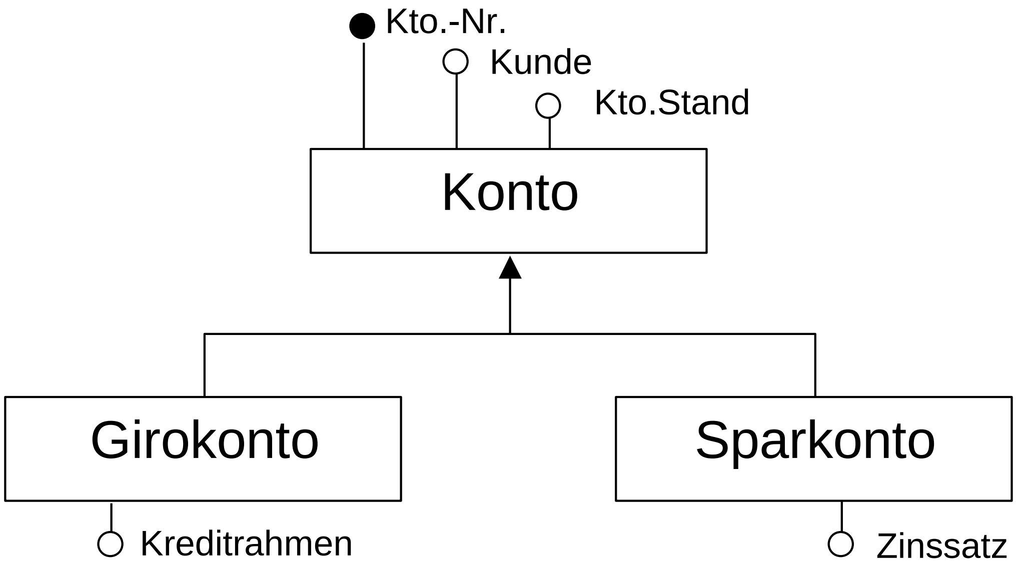
⇓

OBERKLASSE (<u>Schlüssel</u>, Attribut_A, Attribut_B)

SUBKLASSE_1 (<u>Schlüssel</u>, Attribut_A_1, Attribut_B_1)

SUBKLASSE_2 (<u>Schlüssel</u>, Attribut_A_2)

Beispiel



↓

KONTO (<u>Kto.Nr.</u>, Kunde, Kto.Stand)

GIROKONTO (<u>Kto.Nr.</u>, Kreditrahmen)

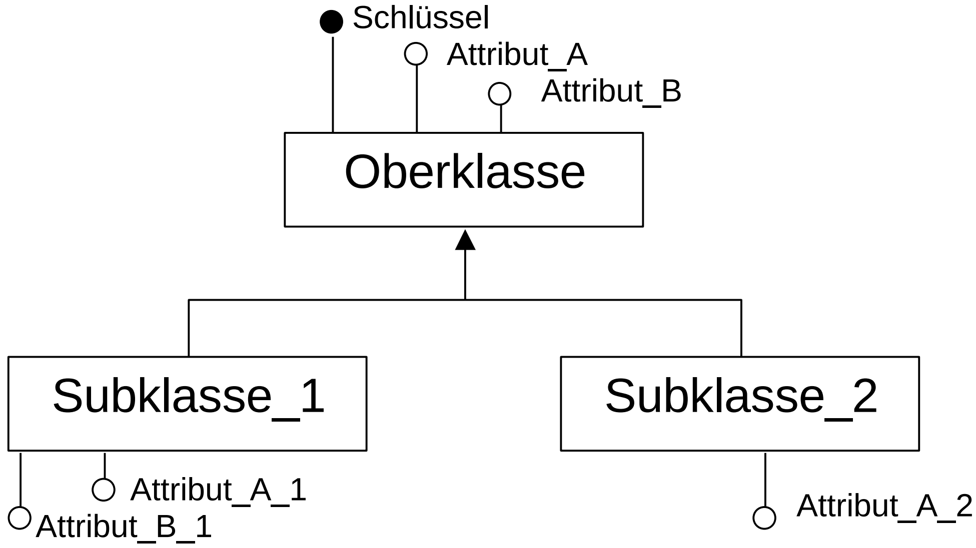
SPARKONTO (<u>Kto.Nr.</u>, Zinssatz)

3. Möglichkeit

Man erstellt eine Relation, die als Schema die Vereinigung der Attribute aller Subklassen und der Oberklasse hat.

Die Attribute, die eine bestimmte Entität nicht hat, werden durch Nullwerte ersetzt.

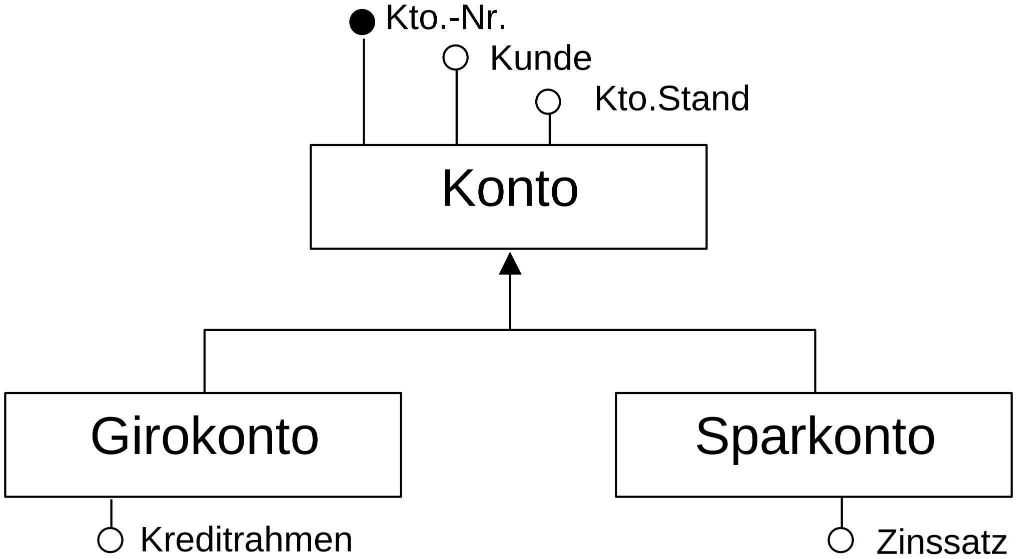
„Regel“:



⇓

KLASSE(<u>Schlüssel</u>, Attribut_A, Attribut_B, Attribut_A_1, Attribut_B_1, Attribut_A_2)

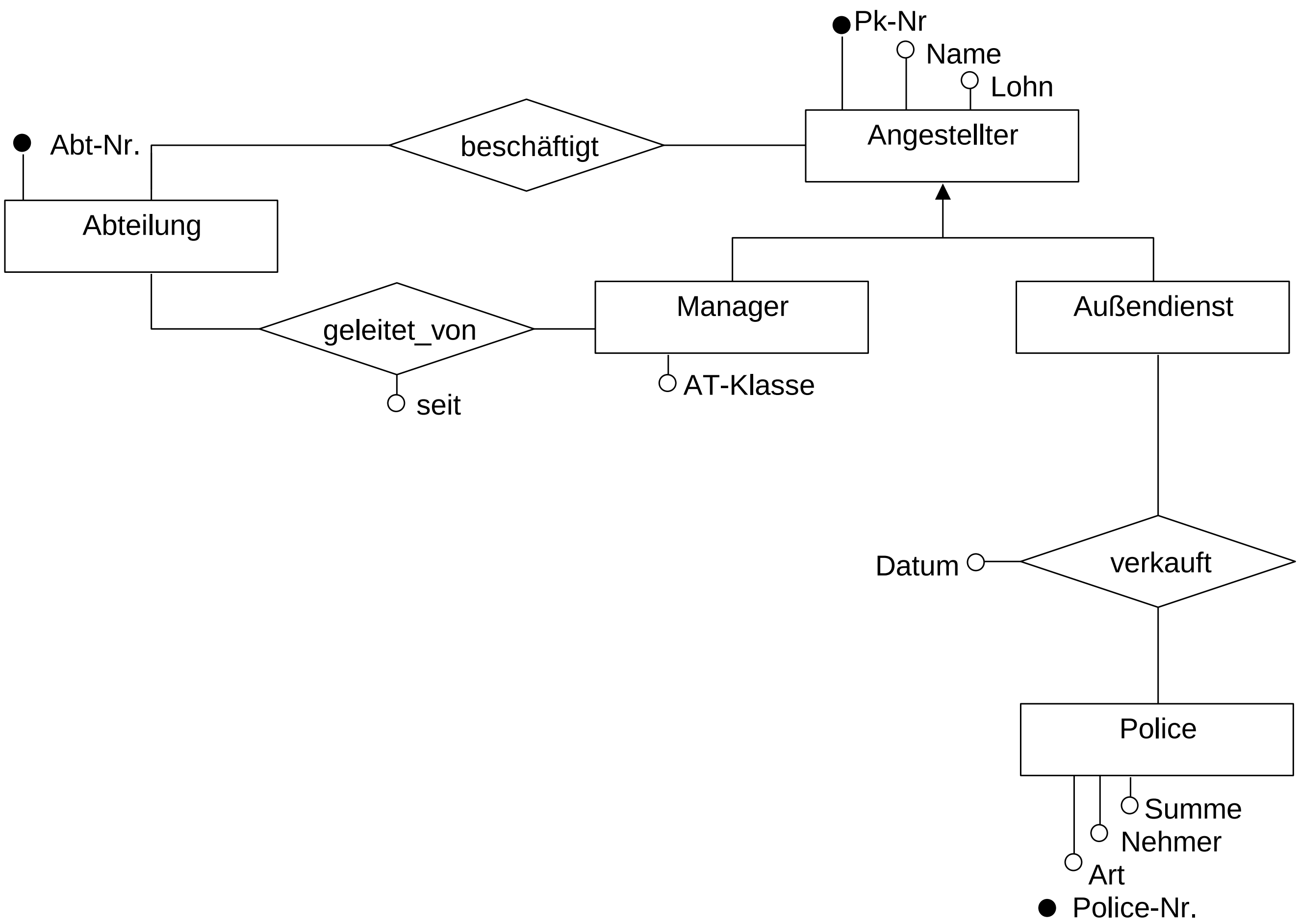
Beispiel



⬇

KONTO (<u>Kto.Nr.</u>, Kunde, Kto.Stand, Kreditrahmen, Zinssatz)

Abschließendes Beispiel



ANGESTELLTER (<u>PK-NR</u>, NAME, LOHN)

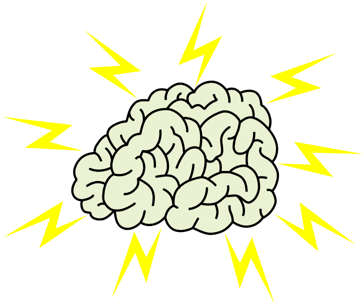
AUSSENDIENST (<u>PK-NR</u>)

ABT_MANAGER (<u>ABT-NR</u>, SEIT, PK-NR, AT-KLASSE)

BESCHÄFTIGT (<u>ABT-NR</u>, <u>PK-NR</u>)

POLICE (<u>POLICE-NR</u>, ART, NEHMER, SUMME,
DATUM, PK-NR)

Operationen auf Relationen

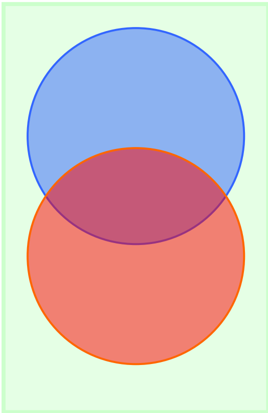


Relationale Algebra (Codd, 1972)

Die **Relationale Algebra** stellt die Operationen zur Verfügung, mit denen auf die Relationen zugegriffen werden kann.

Dadurch können beispielsweise **Anfragen** (Queries) an eine Datenbank gestellt werden.

Mengenoperationen



Die klassischen Mengenoperationen

- **Vereinigung**
- **Durchschnitt**
- **Differenz**

sind jeweils nur über der **gleichen Attributmenge** definiert. Die Schemata der zu vereinigenden Relationen müssen demnach alle identisch sein.

Beispiel:

Gegeben sind **zwei Relationen r** und **s**:

r

A	**B**	**C**
a_1	b_1	c_1
a_1	b_2	c_1
a_2	b_1	c_2

s

A	**B**	**C**
a_1	b_2	c_1
a_2	b_2	c_1
a_2	b_2	c_2

Durchschnitt:

$r \cap s$ =

A	**B**	**C**
a_1	b_2	c_1

Differenz:

$r - s$ =

A	**B**	**C**
a_1	b_1	c_1
a_2	b_1	c_2

Bemerkung:

Der **Schnitt** zweier Relationen lässt sich durch die **Differenz** ausdrücken, denn es gilt:

$$r \cap s = r - (r - s)$$

Beispiel: (Fortsetzung)

r	**(A**	**B**	**C)**
	a_1	b_1	c_1
	a_1	b_2	c_1
	a_2	b_1	c_2

s	**(A**	**B**	**C)**
	a_1	b_2	c_1
	a_2	b_2	c_1
	a_2	b_2	c_2

Vereinigung:

	$r \cup s$ =	**(A**	**B**	**C)**
		a_1	b_1	c_1
		a_1	b_2	c_1
ohne Duplikate		~~a_1~~	~~b_2~~	~~c_1~~
		a_2	b_2	c_1
		a_2	b_1	c_2
		a_2	b_2	c_2

Spezielle Operationen

1) **Selektion**
   zur Auswahl bestimmter Tupeln
2) **Projektion**
   zur Auswahl bestimmter Attribute
3) **Join**
   zur Verknüpfung von Relationen
4) **Division**

Selektion σ

Die **Selektion** ist die Auswahl von **Zeilen** (Tupeln):

Symbol für die Selektion

$$\sigma_{\underline{A=a}}(r)$$

Relation

Auswahlkriterium

Es gilt: $\sigma_{A=a}(r) = \{t \in r \,/\, t(A)=a\}$

Bemerkung:

Der Selektionsoperator ist **kommutativ**, d.h. es gilt:

$$\sigma_{A=a}(\sigma_{B=b}(r)) = \sigma_{B=b}(\sigma_{A=a}(r))$$

Beispiel: Flughafen

„Alle Flüge, die am Flughafen 'JFK' starten."

$\sigma_{VON='JFK'}(FLÜGE)$

FLUGNR	*VON*	*NACH*	*ABFLUG*	*ANKUNFT*
083	JFK	O'Hare	11:30a	1:43p
109	JFK	L. A.	9:50p	2:25a
213	JFK	Boston	11:43a	12:45p

Verkettung von Selektionen

Beispiel: Flughafen

„Alle Flüge von JFK nach Boston“

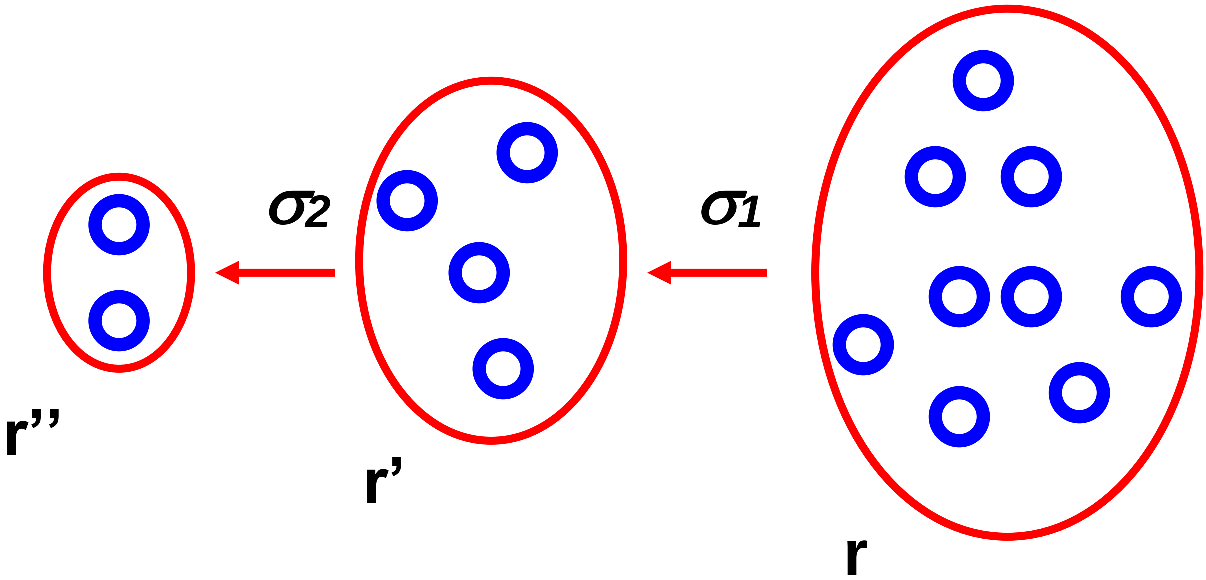
$$mein\_Flug = \sigma_{NACH=\text{“Boston“}} (\sigma_{VON=\text{“JFK“}}(FLÜGE))$$

Selektion 2 **Selektion 1**

mein_Flug

FLUGNR	*VON*	*NACH*	*ABFLUG*	*ANKUNFT*
213	JFK	Boston	11:43a	12:45p

Selektionen **verringern** die Zahl der Tupel:



Erweiterung der Selektion:

Sei Θ eine Menge von Vergleichsoperatoren (z.B. $\{<, \leq, =, \geq, >\}$) und θ ein Operator aus dieser Menge, so ermöglicht dies einen Vergleich mit **Konstanten** oder **anderen Attributen:**

$$\sigma_{A \theta a}(r) = \{t \in r \,/\, t(A) \theta a\}$$

$$\sigma_{A \theta B}(r) = \{t \in r \,/\, t(A) \theta\, t(B)\}$$

wobei ***A*** und ***B*** Attribute der Relation ***r*** sind und $a \in dom(A)$.

Beispiel: Flughafen

flugzeiten	(Flugnr	Abflug	Ankunft)
	084	3:00p	5:55p
	109	9:40p	2:42a
	117	10:05p	1:43a
	213	11:43a	12:45p
	214	2:20p	3:12p

„Alle Flüge, die vor 1:00p ankommen"

$\sigma_{\textbf{ANKUNFT} \leq \textbf{1:00p}}$ ***(flugzeiten) =***

(Flugnr	Abflug	Ankunft)
109	9:40p	2:42a
109	10:05p	1:43a
109	11:43a	12:45p

Selektion σ allgemein

$$\sigma_F(r)$$

F **stellt eine Formel dar, für die gilt:**

1. **Operatoren sind Konstanten oder Attributnamen**
2. **Zulässig sind die Vergleichsoperatoren <, =, >, $\leq$, $\neq$ und $\geq$.**
3. **Zulässig sind die logischen Operatoren $\wedge$ (UND), $\vee$ (ODER) und $\neg$ (NICHT).**

Beispiel:

$$\sigma_{NACH=\text{“Boston“} \wedge VON=\text{“JFK“}}(FLÜGE))$$

Projektion π

Die **Projektion** ist die Auswahl von **Spalten**. Sei $X \subseteq R$.

Symbol für die Projektion

$$\pi_X(r)$$

Attributmenge

Relation

Es gilt: $\pi_X(r) = \{t(X) \,/\, t \in r\}$.

Bemerkung:

Der Projektionsoperator ist **kommutativ bezüglich der Selektion**, wenn die Attribute, auf die selektiert wird, in den projizierten Attributen enthalten sind:

$$\pi_{A,B}(\sigma_{A=a}(r)) = \sigma_{A=a}(\pi_{A,B}(r))$$

Beispiel: Flughafen

flugzeiten

(Flugnr	Abflug	Ankunft)
084	3:00p	5:55p
109	9:40p	2:42a
117	10:05p	1:43a
213	11:43a	12:45p
214	2:20p	3:12p

„Alle Abflug und Ankunftszeiten!“

$\pi_{\{\text{Abflug, Ankunft}\}}$ ***(flugzeiten) =***

(Abflug	Ankunft)
3:00p	5:55p
9:40p	2:42a
10:05p	1:43a
11:43a	12:45p
2:20p	3:12p

Beispiele:

1. Gib mir alle Flüge, die von Boston abfliegen.

   $\sigma_{\text{VON=“Boston“}}$ (FLUEGE)
   $\rightarrow$ {(214, Boston, JFK, 2:20p, 3:12p)}

2. Gib mir nur die Flugnummern aller Flüge, die von Boston abfliegen.

   $\pi_{\text{FLUGNR}}(\sigma_{\text{VON=“Boston“}}$ (FLUEGE)) $\rightarrow$ {(214)}

Bemerkung:

Projektionen **verringern** die Zahl der Attribute. Bei der Durchführung einer Projektion können **doppelte** Tupel entstehen. Da das Ergebnis aber eine **Menge** und keine Multimenge ist, müssen **doppelte** Tupel **gestrichen** werden.

Beispiel:

r **(*A B C*)**

A	B	C
1	2	3
2	2	3

$\pi_{B,C}(r)$ **(*B C*)**

B	C
2	3

Hintereinanderausführung von Projektionen

Wenn zwei Projektionen hintereinander ausgeführt werden, beinhaltet die letztere die erstere Projektion:

Wenn π_y auf dem Ergebnis von $\pi_X(r)$ ausgeführt wird, ist das Ergebnis das gleiche, wie wenn π_y direkt auf r angewendet worden wäre.

Für eine Folge von Projektionen muss daher nur die letzte angewendet werden.

Falls $X_1 \subseteq X_2 \subseteq ... \subseteq X_m \subseteq R$, dann gilt:

$$\pi_{X1} (\pi_{X2} (... (\pi_{Xm} (r)) ...)) = \pi_{X1}(r)$$

Natürlicher Verbund (Natural Join)

Der **Verbundoperator** (Join) verknüpft zwei Relationen auf ihren **gemeinsamen** Attributen.

Seien r(R), s(S) Relationen und $R \cup S = T$, dann wird der **natürliche Verbund** $r \bowtie s = q(T)$ definiert durch:

$$r \bowtie s = \{\, t \,/\, \exists\, t_r \in r \wedge \exists\, t_s \in s : t_r = t(R) \wedge t_s = t(S) \,\}$$

Der Verbundoperator ist **kommutativ.**

Falls $R \cap S = \{\}$, dann ist $r \bowtie s$ das kartesische Produkt, geschrieben als $r \times s$.

Da $R \cap S$ eine Untermenge sowohl von R als auch von S ist, gilt $t_r(R \cap S) = t_s(R \cap S)$. Damit ist jedes Tupel in q eine Kombination eines Tupels von r mit einem Tupel von s mit gleichen $(R \cap S)$-Werten.

Beispiel:

r	(A	B	C)
	a	b	c
	d	b	c
	b	c	f
	c	a	d

s	(B	C	D)
	b	c	d
	b	c	e
	a	d	b

$q = r \bowtie s =$

(A	B	C	D)
a	b	c	d
a	b	c	e
d	b	c	d
d	b	c	e
c	a	d	b

Definition des kartesischen Produktes

Falls $R \cap S = \{\}$, dann ist $r \bowtie s$ das kartesische Produkt, geschrieben als $r \times s$.

Beispiel:

r	(A	B
	a_1	b_1
	a_2	b_1

s	(C	D)
	c_1	d_1
	c_2	d_1
	c_2	d_2

$r \times s =$

(A	B	C	D)
a_1	b_1	c_1	d_1
a_1	b_1	c_2	d_1
a_1	b_1	c_2	d_2
a_2	b_1	c_1	d_1
a_2	b_1	c_2	d_1
a_2	b_1	c_2	d_2

Beispiel: Flugdienst (1)

Gegeben sind die beiden folgenden Relationen:

benutzbar

(FLUGNR	**MASCHINE)**
083	727
083	747
084	727
084	747
109	707

zugelassen

(PILOT	**MASCHINE)**
Sim-	707
Sim-	727
Barth	747
Hill	727
Hill	747

„Welche Kombinationsmöglichkeiten gibt es?“

➔ ***optionen = benutzbar*** $\bowtie$ ***zugelassen***

optionen

(FLUG	**MASCHINE**	**PILOT)**
083	727	Simmons
083	727	Hill
083	747	Barth
083	747	Hill
084	727	Simmons
084	727	Hill
084	747	Barth
084	747	Hill
109	707	Simmons

Der natürliche Verbund kann nur angewandt werden, wenn beide Relationen ***r*** und ***s*** Spalten haben, die durch Attribute **benannt** sind.

Um $r \bowtie s$ zu berechnen, geht man wie folgt vor:

1. Berechne $r \times s$.
2. Für jedes Attribut ***A***, das in ***r*** und in ***s*** vorkommt, werden aus $r \times s$ diejenigen Tupel ausgewählt, bei denen die Werte von ***r.A*** und ***s.A*** gleich sind.
3. Für jedes Attribut ***A*** (wie bei 2.), wird die Spalte ***s.A*** aus der Projektion gestrichen.

Seien r, s, q Relationen, dann gilt für den Join Operator:

- Kommutativ: $\boldsymbol{r \bowtie s = s \bowtie r}$
- Assoziativ: $\boldsymbol{(r \bowtie s) \bowtie q = r \bowtie (s \bowtie q)}$
- sowie: $\boldsymbol{r \bowtie r = r}$

$\Rightarrow \quad \boldsymbol{r \bowtie s = (r \bowtie r) \bowtie s = r \bowtie (r \bowtie s)}$

Gleichverbund (Equijoin)

Definition:

Seien ***r(R), s(S)*** Relationen. $A_i \in R$, $B_i \in S$, und $dom(A_i) = dom(B_i)$, $1 \leq i \leq n$.

Der **Gleichverbund** von ***r*** und ***s*** über die Attribute $A_1,\dots,A_n$ und $B_1,\dots,B_n$, notiert als $r\ [A_1 = B_1, \dots, A_n = B_n]\ s$ ist die Relation:

$$q(RS) = \{t \mid \exists t_s \in s \wedge \exists t_r \in r \text{ mit } t(R) = t_r,\ t(S) = t_s \text{ und } t(A_i) = t(B_i),\ 1 \leq i \leq n\}$$

$$\text{mit } R \cap S = \emptyset$$

Alternative Notation:

Symbol für den Join

Relation 1 $$r \bowtie_{A_1=B_1,\dots,\,A_n=B_n} s$$ **Relation 2**

Attributverknüpfungen

Beispiel: Flughafenpersonal (1)

route	*(FLUG*	*VON*	*NACH)*
	084	O'Hare	JFK
	109	JFK	Los Angeles
	117	Atlanta	Boston
	213	JFK	Boston
	214	Boston	JFK

stationiert	*(PILOT*	*FLUGHAFEN)*
	Terhune	JFK
	Temple	Atlanta
	Taylor	Atlanta
	Tarbell	Boston
	Todd	Los Angeles
	Truman	O'Hare

route [VON = FLUGHAFEN] stationiert = kannfliegen

kannfliegen	*(FLUG*	*VON*	*NACH*	*PILOT*	*FLUGHAFEN)*
	084	O'Hare	JFK	Truman	O'Hare
	109	JFK	Los Angeles	Terhune	JFK
	117	Atlanta	Boston	Temple	Atlanta
	117	Atlanta	Boston	Taylor	Atlanta
	213	JFK	Boston	Terhune	JFK
	214	Boston	JFK	Tarbell	Boston

- Der **Gleichverbund** erweitert den Join-Operator um die Fähigkeit von Vergleichen zwischen Spalten mit verschiedenen Attributnamen.

  Der Gleichverbund wiederholt die Spalten der verbundenen Attribute.

- Der **natürliche Verbund** (natural join) wiederholt die Spalten der verbundenen Attribute **nicht**.

Thetaverbund (Thetajoin)

Definition:

Seien ***r(R), s(S)*** Relationen mit $R \cap S = \{\,\}$, d.h. es gibt keine gemeinsamen Attribute von R und S. Sei $A \in R$, $B \in S$, und die Elemente aus ***dom(A)*** seien mit den Elementen aus ***dom(B)*** mittels der Operatoren aus Θ vergleichbar. Sei weiterhin $\theta \in \Theta$

Der ***Thetaverbund r [AθB]s*** ist die Relation:

$$q(RS) = \{t \mid \exists t_s \in s \wedge \exists t_r \in r \text{ mit } t(R) = t_r,\; t(S) = t_s \text{ und } t(A_i)\ \theta\ t(B_i),\ 1 \leq i \leq n\}$$

Alternative Notation:

Symbol für den Join

Relation 1

$$r \bowtie_{A_1 \theta B_1, \ldots, A_n \theta B_n} s$$

Relation 2

Attributverknüpfungen

Beispiel: Flugzeiten (1)

Gegeben sind die folgenden Relationen über die Flugzeiten von ***a*** nach ***b*** bzw. ***b*** nach ***c***.

flugzeiten_ab

(FLUGNR	ABFLUG	ANKUNFT)
060	9:40a	11:45a
091	12:50p	2:47p
112	4:05p	6:15p
306	8:30p	10:25p
420	9:15p	11:11p

beachte: andere Attributnamen

flugzeiten_bc

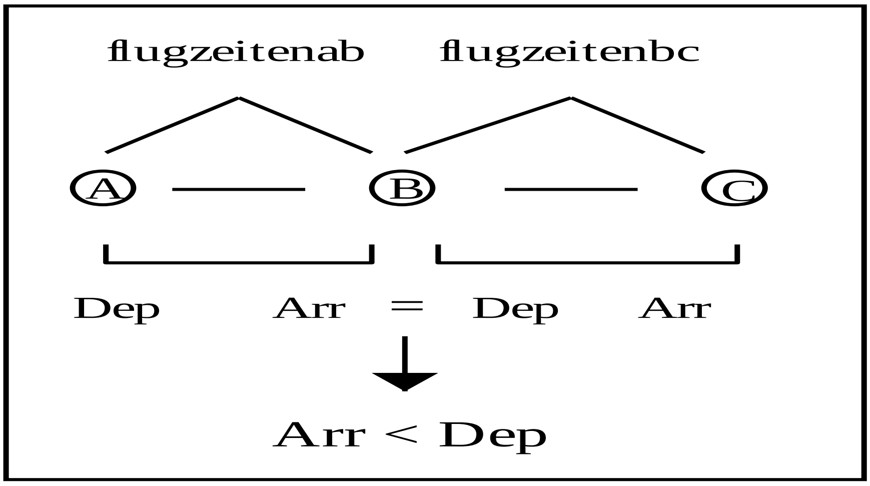
(FLUGNR'	ABFLUG'	ANKUNFT')
011	8:30a	9:52a
060	12:25p	1:43p
156	4:20p	5:40p
158	7:10p	8:35p

Durchführung des folgenden Thetajoins

$$\textbf{verb} = \textbf{flugzeiten\_ab} \underset{ANKUNFT<ABFLUG}{\bowtie} \textbf{flugzeiten\_bc}$$

oder

verb = flugzeiten_ab [ANKUNFT<ABFLUG] flugzeiten_bc



verb	(FLUGNR	ABFLUG	ANKUNFT	FLUGNR'	ABFLUG'	ANKUNFT')
	060	9:40a	11:45a	060	12:25p	1:43p
	060	9:40a	11:45a	156	4:20p	5:40p
	060	9:40a	11:45a	158	7:10p	8:35p
	091	12:50p	2:47p	156	4:20p	5:40p
	091	12:50p	2:47p	158	7:10p	8:35p
	112	4:05p	6:15p	158	7:10p	8:35p

Ausdrücke der Relationenalgebra (RA-Queries)

Beispiel

b:

INVNR	AUTOR	TITEL	VERLAG
1	Date	Intro DBS I	AW
2	Date	Intro DBS II	AW
3	Ullman	Princ. DBS	CSP
4	Kroenke	DB Proc.	SRA

l:

LNR	NAME	ADRESSE
500	Müller	Köln
550	Meier	Bonn
600	Schulz	Aachen

elhn:

INVNR	LNR	RDAT
1	550	300601
2	550	310501
3	600	310501

„Zeige alle Bücher von Autor ‚Date'."

$E_1 = \sigma_{AUTOR='DATE'}(B)$

„Zeige alle Verlage (von denen Bücher vorhanden sind)."

$E_2 = \pi_{VERLAG}(B)$

„Zeige die Rückgabedaten der Bücher mit den Inventar-nummern 1 bzw. 2."

$E_3 = \pi_{RDAT}(\sigma_{INVNR='1' \vee INVNR='2'}(ELHN))$

„Zeige die Adressen der Leser, die zur Zeit ein Buch entlie-hen haben."

$E_4 = \pi_{ADRESSE}(L \bowtie ELHN)$

„Zeige die Buchtitel und Lesernamen jedes aktuellen Ent-leihvorgangs."

$E_5 = \pi_{TITEL,NAME} (B \bowtie L \bowtie ELHN)$

Umbenennung von Attributen (renaming)

Durch das kartesische Produkt können doppelte Attributnamen entstehen. Da dies nicht erlaubt ist, sieht die relationale Algebra eine Möglichkeit vor, Attribute umzubenennen.

Sei r(R) eine Relation mit A ein Attribut in R, B kein Attribut in R und $R' = (R - A) \cup B$, dann ist

$$\delta_{A \rightarrow B} (r) = r'(R')$$

definiert als die Relation

$r'(R') = \{t' \mid \exists\, t \in r \text{ mit } t'(R - A) = t(R - A) \wedge t'(B) = t(A)\}$,

wobei dom(A) = dom(B) gilt.

Beispiel zur Umbenennung:

Kartesisches Produkt ***r(A,B)*** $\times$ ***s(B,C)***

Im Ergebnis käme das Attribut ***B*** zweimal vor:

r × s	**(A**	**B**	**B**	**C)**
	…	**…**	**…**	**…**

Das ist **nicht zulässig**!

Lösung:

Umbenennen der Attribute:

r × s	**(r.A**	**r.B**	**s.B**	**s.C)**
	…	**…**	**…**	**…**

Division

Definition:

Seien **r(R)** und **s(S)** Relationen mit $\mathbf{S \subseteq R}$. Sei R' **= R – S**.
Dann ist die **Division** $\boldsymbol{r \div s}$ die Relation

$$\mathbf{r'(R') = \{t \mid \forall t_S \in s\ \exists t_r \in r \text{ mit } t_r(R') = t \text{ und } t_r(S) = t_S\}.}$$

S ist das kleinere Schema und Teilmenge des Schemas R.
„$\forall t_S$" bedeutet, dass für jedes t_s ein t_r in r gefunden werden muss.

Bemerkung:

Die Division ist **nicht kommutativ**.

$$\mathbf{r \div s \neq s \div r}$$

Beispiel: Flugplan

Gegeben sind die folgenden Relationen:

q

(MASCHINE)
707
727
747

s

(MASCHINE)
707

fliegt

(PILOT	MASCHINE)
Desmond	707
Desmond	727
Desmond	747
Doyle	707
Doyle	727
Davis	707
Davis	727
Davis	747
Davis	1011
Dow	727

„Welcher Pilot kann <u>alle</u> Typen aus q bzw. s fliegen?"

fliegt ÷ q

(PILOT)
Desmond
Davis

fliegt ÷ s

(PILOT)
Desmond
Doyle
Davis

Für $r(R)$, $s(S)$, $S \subseteq R$, $R' = R\text{-}S$ gilt:

$R \div S$ ist die maximale Untermenge r' von $\pi_{R'}(R)$, für die gilt, dass $r' \times s$ in r enthalten ist.

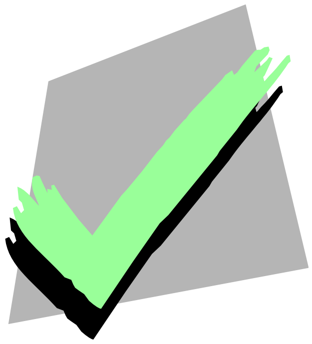
Beweis:

Die Division lässt sich durch die anderen Operationen der **Relationalen Algebra** definieren:

$$\mathbf{r \div s = \pi_{R'}(r) - \pi_{R'}((\pi_{R'}(r) \times s) - r)}$$

$$\mathbf{r \div s = \cap_{t \in s}\ \pi_{R'}(\sigma_{\forall Si \in S\ Si = t(Si)}(r))}$$

Zusammenfassung



Die Relationale Algebra besteht aus sechs atomaren **Grundoperationen**:

- ✓ Vereinigung (union)
- ✓ Mengendifferenz (set difference)
- ✓ Kartesisches Produkt (cartesian product)
- ✓ Projektion (projection)
- ✓ Selektion (selection)
- ✓ Umbenennung von Attributen (renaming of attributes)

Folgende Operationen lassen sich durch diese Grundoperationen ausdrücken:

- ✓ Durchschnitt (intersection)
- ✓ Division (quotient)
- ✓ Gleichverbund (equi-join)
- ✓ Natürlicher Verbund (natural join)
- ✓ Thetaverbund (theta-join)

Jeder Ausdruck der legal aus diesen Operatoren gebildet wurde, ist ein algebraischer Ausdruck.

Wenn ein algebraischer Ausdruck E und die Werte der in E enthaltenen Relationen gegeben sind, kann E zu einer einzigen Relation ausgewertet werden. E repräsentiert eine Abbildung von einer Menge von Relationen zu einer einzigen Relation.

⇨ Ergebnis eines algebraischen Ausdrucks ist immer eine Relation