Relationale Algebra / Bag Algebra

Operationen

Selection	Projection	Natural Join	Duplicate Elimination
$\sigma_{Row="value"}$	$\pi_{Row}(table)$	(table) $\bowtie$ (table)	δ(expression)
$\sigma_{Name='Miller'}$	$\pi_{Name,E-Mail}(user)$	U $\bowtie$ L	$\delta(\sigma_{Name='Miller'})$

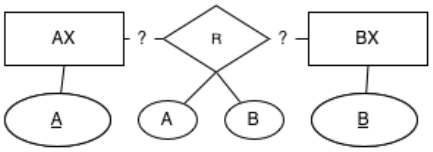
Mengenoperationen

Durchschnitt (intersection)	Vereinigung (union)	Differenz (difference)
$r \cap s$	$r \cup s$	$r \setminus s$
$= r \setminus (r \setminus s) = (r \sqcup s) \setminus (r \cup s)$	$= (r \setminus s) \sqcup s = (r \sqcup s) \setminus (r \cap s)$	
min. Anzahl der mehrf. Tupeln (BA)	max. Anzahl der mehrf. Tupeln (BA)	

r	||°°°°°
s	||||°°°
$r \cap s$	||°°°
$r \cup s$	||||°°°°°
$r \sqcup s$	|||||°°°°°°°°
$r \setminus s$	°°

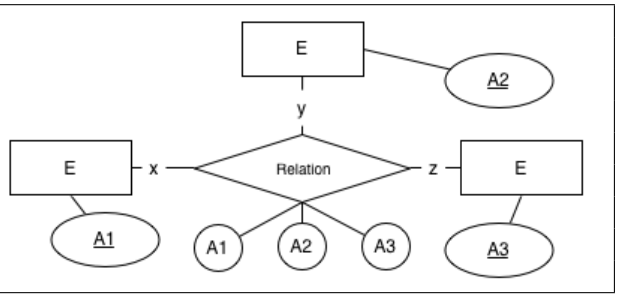
Der Unterschied zwischen Bag Algebra (BA) und Relationaler Algebra (RA) ist, in RelAlg existieren keine doppelten Einträge. Die Bag Algebra jedoch erlaubt dies. Auch existiert der Operator $\sqcup$ (union all) nur in der Bag Welt.

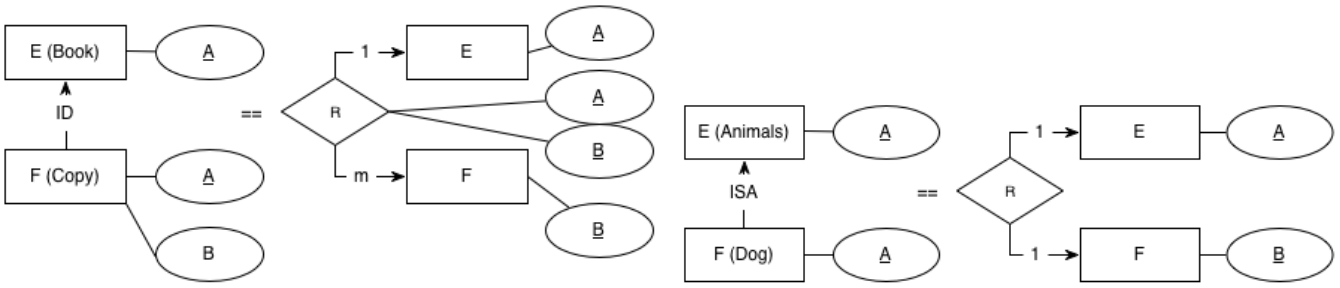
DB Design



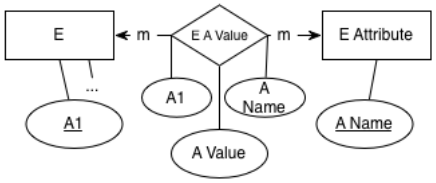
m	{A,B}	m
1	{B}	m
m	{A}	1
1	{A},{B}	1

X	Y	Z	Unique not 1
m	m	m	{1,2,3}
m	m	1	{1,2}
m	1	m	{1,3}
m	1	1	{1,2}+{1,3}
1	m	m	{2,3}
1	m	1	{2,3}+{1,2}
1	1	m	{2,3}+{1,3}
1	1	1	{2,3}+{1,3}+{1,2}

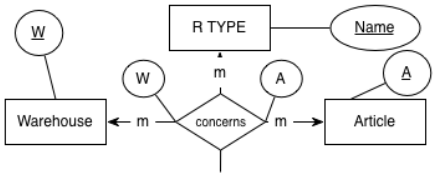




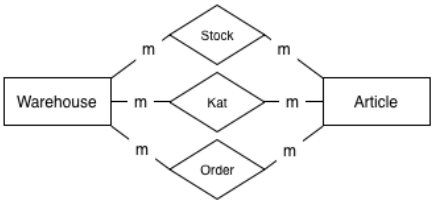
ID / ISA: Vergleich von existenziell abhängiger Version mit unabhängiger Version.



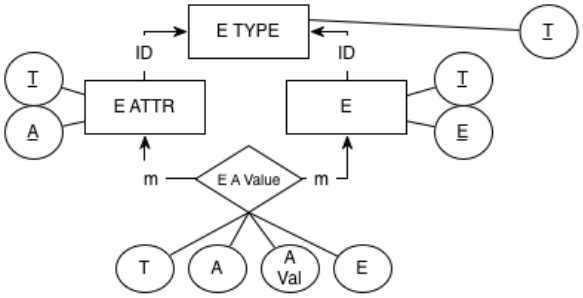
Dynamische Attribute



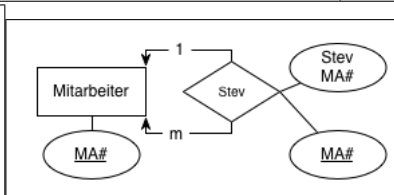
Dynamische Relationen

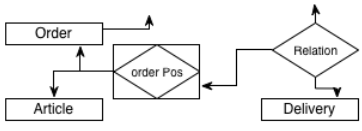


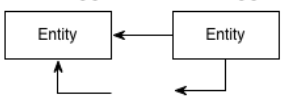
vgl. statische Relationen



ISA - Batterie (dynamische ISA Vererbung)







SQL-Befehle

	SQL	Algebraische Schreibweise / Beschreibung
SELECT, FROM WHERE	SELECT Name, Vorname FROM Besucher	$\Pi_{Nam,Vorname}(Besucher)$
	SELECT Name, Vorname FROM Besucher WHERE Name='Müller'	$\Pi_{Nam,Vorname}(\sigma_{Name='Müller'}(Besucher))$
LIKE	WHERE Name LIKE 'Pe%'	Alle Name die mit 'Pe' beginnen
	WHERE name LIKE '_da%'	Alle Namen die 'da' nach dem ersten Zeichen enthalten
BETWEEN	WHERE Name BETWEEN 'Bucher' AND 'Schmid'	Alle Name zwischen Bucher und Schmid
	WHERE Hausnummer BETWEEN 1 AND 10	Alle Hausnummern zwischen 1 und 10 (inkl. 1 und 10!)
AS (Scope variables)	SELECT x.Name, x.Vorname, y.Ort FROM Besucher AS x, Gast AS y WHERE x.PLZ = y.PLZ	AS nicht unbedingt nötig, alternativ geht auch FROM Besucher x, Gast y
(NOT) EXISTS	SELECT x.Name, x.Vorname, x.Ort FROM Besucher AS x WHERE NOT EXISTS (SELECT y.Vorname, y.Name FROM Besucher AS y WHERE y.Name = y.Name AND x.Vorname < y.Vorname)	ALLE Besucher, ausser die, deren Nachnamen nochmals vorkommt, von denen wird der, mit dem grössten ("höchste") Vornamen genommen.
SUM	SELECT SUM(Lagerbestand) FROM Sortiment	Summe aller Werte von "Lagerbestand"
	SELECT SUM (DISTINCT Lagerbestand) FROM Sortiment	Summe aller Werte ohne Duplikate
COUNT	SELECT COUNT(*) FROM Besucher	Zählt alle Tupel in der Tabelle Besucher
	SELECT COUNT(DISTINCT *) FROM Artikel	Alle Tupel, Ohne Duplikate
	SELECT COUNT(Name,Vorname)	Geht nicht, stattdessen:
	SELECT COUNT(*) FROM (SELECT Name, Vorname From Besucher)	
GROUP BY	SELECT Restaurant, MAX(Frequenz) FROM Gast GROUP BY Restaurant	Höchste Frequenz eines Restaurants
	SELECT Restaurant FROM Gast GROUP BY Restaurant HAVING COUNT(*) > 5	Alle Restaurants die mehr als 5 Gäste haben
	SELECT x.Restaurant FROM Gast AS x WHERE 5 < (SELECT COUNT(*) FROM Gast AS y WHERE y.Restaurant = x.Restaurant)	Gleiches Statement, ohne GROUP BY

CREATE

```
CREATE TABLE Name
    (
    A1 INTEGER NOT NULL,
    A2 INTEGER NOT NULL,
    Attr1Name VARCHAR(30) NOT NULL,
    Attr2Name DATE NOT NULL,
    Attr3Name TIME NOT NULL,
    Attr4Name TIMESTAMP NOT NULL,
    PRIMARY KEY (A1),
    FOREIGN KEY (A2) REFERENCES TABLE TableName,
    UNIQUE (A1,...)
    )
```