

RELATIONALE ALGEBRA

Die relationale Algebra ist die grundlegende Datenmanipulationssprache zum ursprünglichen Relationenmodell und wurde gleichfalls von E.F. Codd beschrieben. Sie ist eine formale Sprache, die im Wesentlichen auf der Mengenalgebra basiert und von Codd um relationentypische Operationen ergänzt wurde.

Gegenstand der relationalen Algebra ist, dass sich auf eine oder mehrere Relationen spezielle Operationen definieren lassen, die als Ergebnis eine neue Relation liefern. Gleiches gilt auch, wenn diese Operationen in beliebiger Reihenfolge verknüpft und verschachtelt werden.

Beispielrelationen:

KUNDEN (Kunden aller Vertriebsabteilungen)

Kundnr	Name	Ort	Region
112	Schmidt	München	Süd
115	Richter	Bremen	Nord
123	Meier	Dresden	Ost
222	Schulze	Berlin	Mitte
333	Müller	Berlin	Mitte
345	Kunze	Bonn	West

KUNDEN1 (Kunden der Vertriebsabteilung A)

Kundnr	Name	Ort	Region
123	Meier	Dresden	Ost
222	Schulze	Berlin	Mitte
345	Kunze	Bonn	West

KUNDEN2 (Kunden der Vertriebsabteilung B)

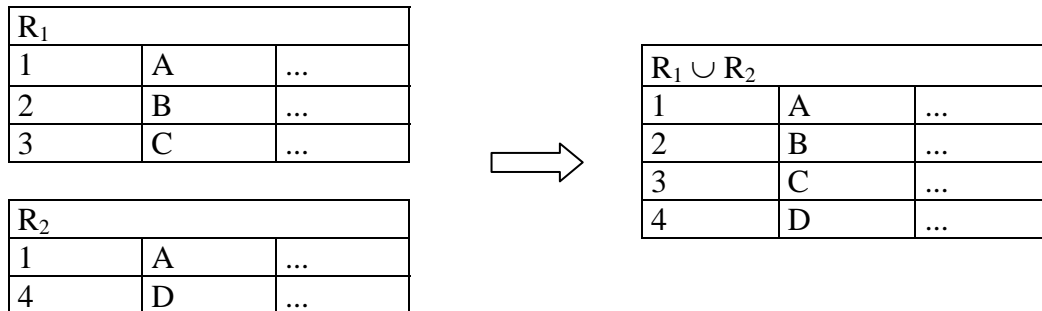
Kundnr	Name	Ort	Region
112	Schmidt	München	Süd
333	Müller	Berlin	Mitte
345	Kunze	Bonn	West

AUFTRAG

Auftrnr	Kundnr	Auftragdat	Betrag
99001	123	07.08.1999	125,50
99003	345	14.08.1999	1.500,00

VEREINIGUNG

Bei der **Vereinigung** (union) $R_1 \cup R_2$ wird die Menge der Tupel der Relation R_1 um die Menge der Tupel der Relation R_2 erweitert (oder umgekehrt). Die Ergebnisrelation enthält gleiche Tupel der ersten und zweiten Relation nur einmal.



Prinzipiskizze der Vereinigung zweier Relationen

Beispiel:

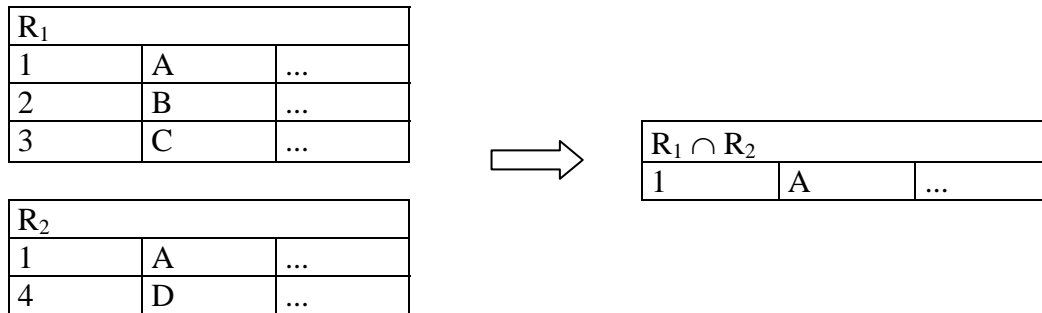
Die Vereinigung $KUNDEN1 \cup KUNDEN2$ ergibt die Relation der Kunden, die durch die Vertriebsabteilung A oder die Vertriebsabteilung B betreut werden.

UNION (KUNDEN1, KUNDEN2)

Kundnr	Name	Ort	Region
112	Schmidt	München	Süd
123	Meier	Dresden	Ost
222	Schulze	Berlin	Mitte
333	Müller	Berlin	Mitte
345	Kunze	Bonn	West

DURCHSCHNITT

Der **Durchschnitt** (intersection) $R_1 \cap R_2$ ermittelt gleiche Tupel aus zwei Relationen und enthält jedes identische Tupel nur einmal (siehe Abb.).



Prinzipiskizze des Durchschnitts zweier Relationen

Beispiel:

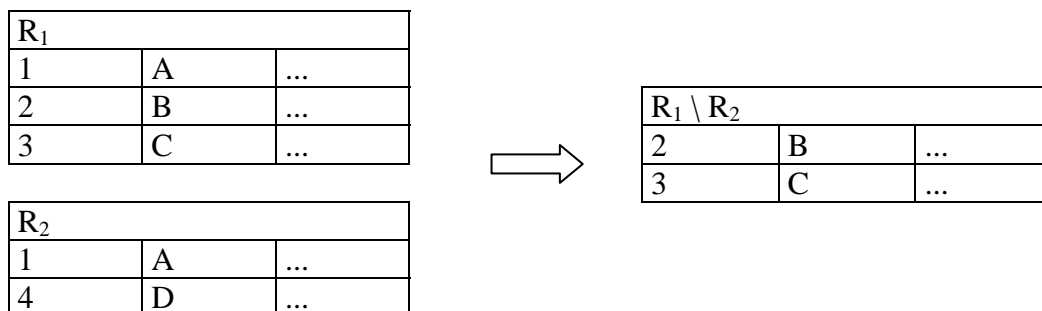
Der Durchschnitt $KUNDEN1 \cap KUNDEN2$ ergibt die Relation der Kunden, die von der Vertriebsabteilung A und der Vertriebsabteilung B betreut werden.

INTERSECTION (KUNDEN1, KUNDEN2)

Kundnr	Name	Ort	Region
345	Kunze	Bonn	West

DIFFERENZ

Die **Differenz** (difference) $R_1 \setminus R_2$ bildet eine Relation, die alle Tupel der Relation R_1 abzüglich der Tupel der Relation R_2 enthält.



Prinzipiskizze der Differenz zweier Relationen

Beispiel:

Die Differenz $KUNDEN \setminus KUNDEN1$ ergibt die Relation der Kunden, die nicht von der Vertriebsabteilung A betreut werden.

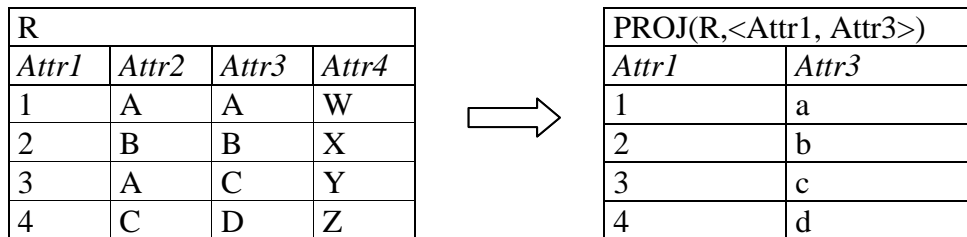
DIFFERENCE (KUNDEN, KUNDEN1)

Kundnr	Name	Ort	Region
112	Schmidt	München	Süd
115	Richter	Bremen	Nord
333	Müller	Berlin	Mitte

PROJEKTION

Die folgenden Operationen bringen jene Erweiterungen, die die Relationenalgebra von der gewöhnlichen Algebra unterscheiden. Zunächst sollen die zwei Operationen Projektion und Selektion erläutert werden, die in ihrer Grundform auf eine Relation anzuwenden sind.

Durch eine **Projektion** werden bestimmte Attribute einer Relation ausgewählt. Bei der Darstellung in Tabellenform entspricht dies der Auswahl von Spalten. Das Ergebnis der Projektion ist selbst wieder eine Relation.



Prinzipskizze der Projektion

Wird die Projektion auf Nichtschlüsselattribute geführt, müssen gleichzeitig alle jetzt mehrfach vorhandenen gleichen Tupel bis auf je eines ebenfalls gestrichen werden.

Beispiel:

Auf die Relation der Kunden aller Vertriebsabteilungen soll eine Projektion ausgeführt werden, die als Ergebnis die Namen und Wohnorte aller Kunden liefert:


PROJ (KUNDEN,<Name, Ort>)

Name	Ort
Schmidt	München
Richter	Bremen
Meier	Dresden
Schulze	Berlin
Müller	Berlin
Kunze	Bonn

SELEKTION

Die **Selektion** (restriction) wählt alle Tupel in einer Relation aus, die einer bestimmten Bedingung genügen. In der Tabellendarstellung führt die Selektion zu einer Auswahl von Zeilen. Die relationale Operation Selektion darf nicht mit dem SQL-Befehl SELECT verwechselt werden.

R			
Attr1	Attr2	Attr3	Attr4
1	A	A	W
2	B	B	X
3	A	c	Y
4	C	d	Z



REST(R, Attr2='A')			
Attr1	Attr2	Attr3	Attr4
1	A	a	W
3	A	c	Y

Prinzipskizze der Selektion

Eine Bedingung kann sich auf ein oder mehrere Attribute beziehen. Bei der Selektion werden die entsprechenden Merkmalsausprägungen überprüft und jedem Tupel der Relation die Aussage „wahr“ oder „falsch“ zugeordnet. Als Vergleichsoperator innerhalb der Bedingung kommen dabei =, <, <=, >, >= sowie „ungleich“ in Frage. Mehrere Bedingungen sind untereinander mit den logischen Operatoren „UND“ (AND), „ODER“ (OR) und „NICHT“ (NOT) verknüpfbar.

Beispiel:

Für die Relation der Kunden aller Vertriebsabteilungen sollen die zwei Bedingungen Region=„Ost“ und Region=„Mitte“ im Rahmen einer Selektion verknüpft werden. Dann ergibt das Ergebnis der Selektion alle Kunden, die entweder in der Region Ost oder der Region Mitte wohnen.

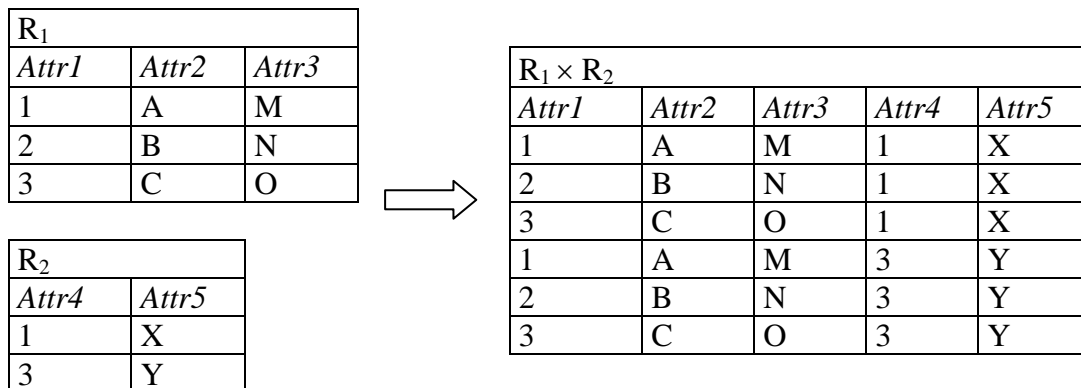
REST (KUNDEN, Region=„Ost“ OR Region=„Mitte“)

Kundnr	Name	Ort	Region
123	Meier	Dresden	Ost
222	Schulze	Berlin	Mitte
333	Müller	Berlin	Mitte

KARTESISCHES PRODUKT

Die Operation des kartesischen Produktes wurde bereits bei der Definition des relationalen Datenmodells eingeführt. Dabei wurde eine Relation als Teilmenge des kartesischen Produktes von mehreren Mengen gebildet.

Das **kartesische Produkt** (product) aus zwei Relationen wird gebildet, indem jedes Tupel der ersten Relation mit jedem Tupel der zweiten Relation kombiniert wird. Alle Attribute der beteiligten Relationen werden vollständig in die Ergebnisrelation übernommen. Dabei wird jede Kombinationsmöglichkeit der beiden Relationen gebildet.



Prinzipiskizze des kartesischen Produktes

In der Abbildung entsteht bei der Bildung des kartesischen Produktes aus einer Relation mit 3 Tupel und einer Relation mit 2 Tupel eine Ergebnisrelation, die 6 Tupel enthält (3×2). Analog würde das kartesische Produkt von zwei Relationen mit beispielsweise 100 Tupel in der einen und 50 Tupel in der anderen Ausgangsrelation zu 5000 Tupel in der Ergebnisrelation führen. Diese Vervielfachung der Tupel in der Ergebnisrelation des kartesischen Produktes ist ein typisches Merkmal dieser Operation.

Beispiel:

Es soll das kartesische Produkt zwischen der Relation KUNDEN1 und AUFTRAG gebildet werden. Die Ergebnisrelation enthält alle Attribute der beiden Relationen und eine Kombination der Tupel der Relation KUNDEN1 mit jedem Tupel der Relation AUFTRAG.

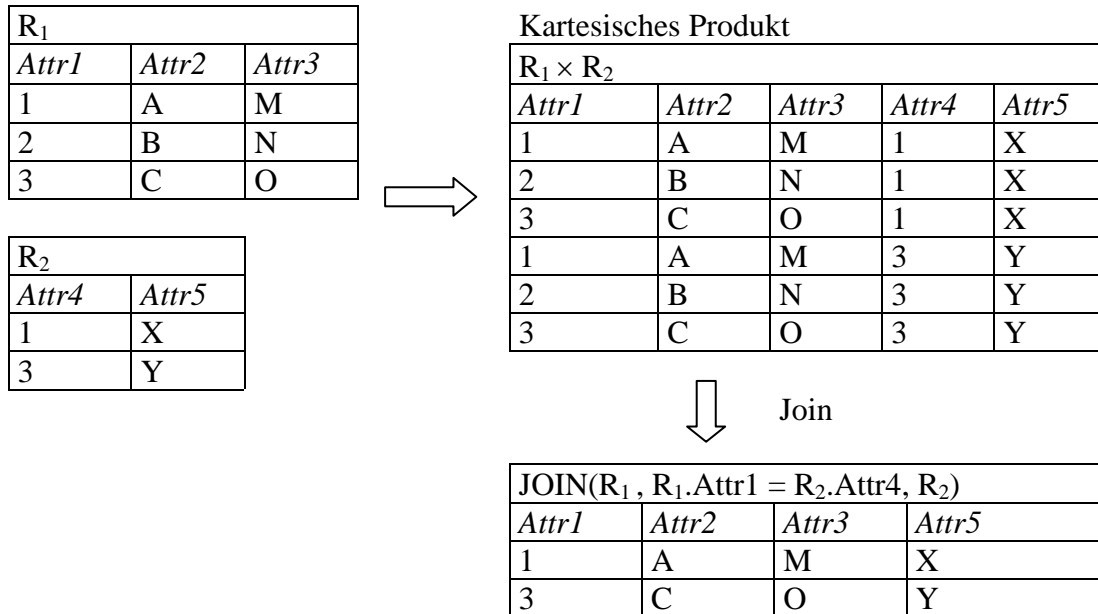
PRODUCT (KUNDEN1, AUFTRAG)

Kundnr	Name	Ort	Region	Auftrnr	Kundnr	Auftragdat	Betrag
123	Meier	Dresden	Ost	99001	123	07.08.1999	125,50
222	Schulze	Berlin	Mitte	99001	123	07.08.1999	125,50
345	Kunze	Bonn	West	99001	123	07.08.1999	125,50
123	Meier	Dresden	Ost	99003	345	14.08.1999	1.500,00
222	Schulze	Berlin	Mitte	99003	345	14.08.1999	1.500,00
345	Kunze	Bonn	West	99003	345	14.08.1999	1.500,00

VERBUND (JOIN)

Mit dem Verbund (Join) werden Relationen miteinander verknüpft. Dabei werden zwei Relationen ähnlich wie beim kartesischen Produkt zusammengefügt, allerdings nur für solche Tupel, in der zwei bestimmte Attributwerte in einer gewissen Beziehung zueinander stehen.

Der sogenannte Natürliche Join (Natural Join) wird genau wie der Equi-Join gebildet. Der Unterschied besteht jedoch darin, daß die Ergebnisrelation gleiche Attributspalten nur einmal beinhaltet. Die in der Abbildung noch doppelt vorhandenen (gleiche) Attribute Attr1 und Attr4 werden in die Ergebnisrelation des Natural-Join nur einmal aufgenommen.



Prinzipskizze des Natural-Join

Beispiel:

Die Operation des Natural-Join soll genutzt werden, um die Relation KUNDEN1 mit der Relation AUFTRAG zu verbinden. Als Join-Attribut kommt nur die Kundnr in Frage, da beide Attribute die gleiche Domäne haben. Eine Verbindung der Tupel wird nur möglich, wenn der Wert der Kundnr der Relation KUNDEN1 gleich dem Wert der Kundnr in der Relation AUFTRAG ist. Die Ergebnisrelation enthält alle Attribute der beiden Relationen, wobei das gleiche Attribut Kundnr nur einmal erscheint.

JOIN (KUNDEN1, KUNDEN1.Kundnr=AUFTRAG.Kundnr, AUFTRAG)

Kundnr	Name	Ort	Region	Auftrnr	Auftragdat	Betrag
123	Meier	Dresden	Ost	99001	07.08.1999	125,50
345	Kunze	Bonn	West	99003	14.08.1999	1.500,00

BEISPIEL NORMALISIERUNG

Die Datenbasis eines Handelsunternehmens soll in einer Datenbank zentralisiert werden. Eine Analyse ergibt die folgenden Feststellungen:

Für einen Mitarbeiter ist die Personalnummer (Pnr), sein Name und Familienstand (Fst) erfasst. Er ist in einer Abteilung tätig, für die eine Abteilungsnummer (Anr) und ein Abteilungsname (Aname) geführt werden. Der Mitarbeiter verkauft eine bestimmte Anzahl (Menge) unterschiedlicher Artikel. Jeder Artikel besitzt eine Artikelnummer (Artnr), eine Artikelbezeichnung (Artbez) und einen Verkaufspreis (Preis). Jeder Mitarbeiter des Unternehmens kann jeden Artikel verkaufen.

Pnr	Name	Fst	Artnr	Artbez	Preis	Menge	Anr	Aname
11	Busch	led.	T111	HD 9GB	349	3	A1	Lager
			T333	Kabel	18	3		
			T444	LJ 601	275	1		
15	Wald	verh.	P586	PC P2	1495	1	A2	Verkauf
			T444	LJ 601	275	1		
16	Wiese	led.	T111	HD 9GB	349	1	A1	Lager
18	Teich	verh.	P586	PC P2	1495	2	A2	Verkauf

Sie sehen vorstehend den ersten, völlig unstrukturierten Lösungsansatz zum Aufbau der Datensätze. Entwickeln Sie eine bessere Lösung, indem Sie schrittweise den Prozess der Normalisierung von der ersten bis zur dritten Normalform (3NF) durchführen.

Personall

1. Normalform (1NF)

Pnr	Name	Fst	Artnr	Artbez	Preis	Menge	Anr	Aname
11	Busch	led.	T111	HD 9GB	349	3	A1	Lager
11	Busch	led.	T333	Kabel	18	3	A1	Lager
11	Busch	led.	T444	LJ 601	275	1	A1	Lager
15	Wald	verh.	P586	PC P2	1495	1	A2	Verkauf
15	Wald	verh.	T444	LJ 601	275	1	A2	Verkauf
16	Wiese	led.	T111	HD 9GB	349	1	A1	Lager
18	Teich	verh.	P586	PC P2	1495	2	A2	Verkauf

