# Kapitel 3: Inhaltsübersicht

- Im Gegensatz zu den relativ komplizierten satzorientierten Datenmodellen (Netzwerkmodell und hierarchisches Modell) besticht das relationale Datenmodell durch seine sehr einfache Struktur auf Basis von zweidimensionalen Tabellen.
- Im ersten Teil von Kapitel 3 wird gezeigt, nach welchen formalen Regeln ein E/R-Diagramm in das **relationale Modell** umgewandelt werden kann.
- Der zweite Abschnitt beschäftigt sich mit der **relationalen Algebra**, die als mathematische Basis für SQL betrachtet werden kann.

# Grundlagen und Grundbegriffe (1)

- Seien $W_1, W_2, \ldots, W_n$ beliebige Mengen. Dann ist definiert als **kartesisches Produkt**

  $$W_1 \times W_2 \times \ldots \times W_n = \{(w_1, w_2, \ldots, w_n \mid w_i \in W_i \ (i = 1, \ldots, n)\}$$

- Im mathematischen Sinn ist eine n-stellige **Relation R** über $W_1, W_2, \ldots, W_n$ definiert als Teilmenge

  $$R \subseteq W_1 \times W_2 \times \ldots \times W_n$$ (n wird als Grad der Relation bezeichnet)

- Sei $A = \{a_1, a_2, \ldots, a_n\}$ eine Menge von Attributen (paarweise verschiedenen) und $a_i$ der Wertebereich dom $(a_i)$. Dann bezeichnet man als

  **Relationsschema** $R(A)$ mit $A = \{a_1, a_2, \ldots, a_n\}$

  wobei R der Name der Relation und A die Attributmenge ist.

# Grundlagen und Grundbegriffe (2)

- Zu jedem Attribut gibt es einen Wertebereich dom ($a_i$) mit der Menge der zulässigen Werte für dieses Attribut. Für den **Wertebereich der Relation** R gilt demnach:

$$R \subseteq \text{dom}(A) = \text{dom}(a_1) \text{ x dom } (a_2) \text{ x ... x dom}(a_n)$$

- Die Relation r ist eine aktuelle Ausprägung (**Instanz**) des Relationenschemas R (A) und enthält Tupel aus der Domain A:

$$r \subseteq \text{dom}(A) = \text{dom}(a_1) \text{ x dom } (a_2) \text{ x ... x dom}(a_n)$$

- Eine **Teilmenge K** $\subseteq$ A heißt indentifizierend für R(A) wenn für jede Relation r (= Instanz von R) gilt:

  **Seien x,y $\in$ r zwei beliebige Tupel mit x ≠ y.**

  **Für alle x,y gilt x.K ≠ y.K**

# Beispiel Artikelliste

Relationen-
name

Attributnamen (müssen eindeutig sein!)

Relationen-
schema

| Artikel | Art.-Nr. | Art.-Grp. | Bezeichnung | HEK |
|---|---|---|---|---|
| | 100005 | Server | ML 150 | 2.930 € |
| | 100015 | Server | DL 350 | 4.870 € |
| | 100230 | Server | DL 380 | 6.750 € |
| Tupel → | 200112 | Lizenzen | MS Office 2019 | 185 € |
| | 200230 | Lizenzen | Win Server 2019 | 575 € |

Relation:
Menge
aller
Tupel

Schlüssel = identifizierend und minimal

- Die Reihenfolge der Tupel ist beliebig (Mengeneigenschaft).
- Die Reihenfolge der Attribute ist grundsätzlich auch beliebig, sollte aber sinnvoll festgelegt werden (Lesbarkeit!) und dann auch konstant bleiben.

# Die 1. Normalform

- Eine Relation R (A) befindet sich in der **1. Normalform** (auch **1NF-Relation**) wenn gilt:
  Jedes a $\in$ A ist ein elementares (atomares) Attribut, d.h. der „Schnittpunkt" einer Zeile mit einer Spalte enthält immer nur genau einen einzigen Attributwert.
- Anders formuliert, sind weder zusammengesetzte noch mehrwertige Attribute zulässig.
- Negativbeispiel:

| Ang.Nr. | Name (Nach-, Vorname) | Sprachen | Sem. |
|---|---|---|---|
| 4711 | Weber, Claudia | {deutsch, englisch, spanisch} | 1 |
| 4712 | Müller, Peter | {deutsch, französisch} | 3 |
| 4713 | Maier, Manfred | {deutsch} | 5 |

# Lösungsmöglichkeiten für Negativbeispiel (1)

- Wiederholungsgruppen durch weitere Tupel auflösen?

| Ang.Nr. | Vorname | Nachname | Sprache | Sem. |
|---|---|---|---|---|
| 4711 | Claudia | Weber | deutsch | 1 |
| 4711 | Claudia | Weber | englisch | 1 |
| 4711 | Claudia | Weber | spanisch | 1 |
| 4712 | Peter | Müller | deutsch | 3 |
| 4712 | Peter | Müller | französisch | 3 |
| 4713 | Manfred | Maier | deutsch | 5 |

- Zusätzliche Attribute für mehrwertige Attribute?

| Ang.Nr | Vorname | Nachname | 1. Sprache | 2. Sprache | 3. Sprache | Sem. |
|---|---|---|---|---|---|---|
| 4711 | Claudia | Weber | deutsch | englisch | spanisch | 1 |
| 4712 | Peter | Müller | deutsch | französisch | | 3 |
| 4713 | Manfred | Maier | deutsch | | | 5 |

# Lösungsmöglichkeiten für Negativbeispiel (2)

- Relation Studenten um das Attribut Sprache reduzieren

| Ang.Nr. | Vorname | Nachname | Sem. |
|---|---|---|---|
| 4711 | Claudia | Weber | 1 |
| 4712 | Peter | Müller | 3 |
| 4713 | Manfred | Maier | 5 |

- und eine zusätzliche Relation Sprachkenntnisse einführen:

| Ang.Nr. | Sprache |
|---|---|
| 4711 | deutsch |
| 4711 | englisch |
| 4711 | spanisch |
| 4712 | deutsch |
| 4712 | französisch |
| 4713 | deutsch |

## Umsetzung des E/R-Diagramms in ein relationales Schema

1. **Übertragung von Entity-Typen in ein Relationenschema**
   Der Name des Entity-Typs, seine Attribute und sein Schlüssel werden übernommen und als Relation abgebildet.

2. **Übertragung von Beziehungstypen in ein Relationenschema**
   Name, Zusatzattribute und die Schlüssel der an der Beziehung beteiligten Entity-Typen (**Fremdschlüssel**) werden nach bestimmten Regeln übernommen und ebenfalls als Relation abgebildet. D.h. im relationalen Modell wird nicht zwischen Entity-Typ und Beziehungs-Typ unterschieden!

3. **Schemaverfeinerung**
   Der entstandene Initial-Entwurf wird anschließend optimiert.

# Verfeinerung des initialen Entwurfs

- Bei dieser Optimierung geht es primär darum, die Anzahl der im ersten Entwurfsschritt erzeugten Relationen zu reduzieren.
- Es können allerdings nur Relationen eliminiert werden, die 1:1-, 1:n- oder n:1-Beziehungen darstellen. Auf <u>keinen</u> Fall dürfen n:m-Beziehungstypen entfernt werden, da dies zu schwerwiegenden Entwurfsfehlen (Anomalien) führt.
- „Goldene Regel“ bei der Eliminierung von Relationen

**Nur Relationen mit gleichem Schlüssel zusammenfassen!**

# Verfeinerung von 1:n - Beziehungen

- In dem Relationenschema des Objekttyps, bei dem die Komplexitätsangabe „n“ steht, wird der Schlüssel des Objekttyps mit der Komplexitätsangabe „1“ als sog. Fremdschlüssel eingetragen.
- Die Beziehung zwischen zwei konkreten Entities wird dann über identische Werte in den betreffenden Spalten der beiden Tabellen hergestellt.
- Beispiel: 1:n Beziehung zwischen Abteilungen und Mitarbeiter

| Abt.Nr | Abt.-Name |
|---|---|
| **4711** | Softwareentwicklung |
| **4712** | Vertrieb |
| **4713** | Buchhaltung |

| Ang.Nr | Name | Beruf | Abt.Nr |
|---|---|---|---|
| 42203 | Schmitt | Informatikerin | 4711 |
| 42213 | Peters | DV-Kaufmann | 4711 |
| 41217 | Müller | Bilanzbuchhalter | 4713 |

logische Verknüpfung über identische Attributwerte

# Verfeinerung von n:m - Beziehungen

- Eine n:m-Beziehung zwischen zwei Objekttypen erfordert grundsätzlich eine neue Relation im Relationenschema, in welche die Schlüssel der beteiligten Objekttypen und gegebenenfalls vorhandene Zusatzattribute aufgenommen werden.

- Beispiel: n:m Beziehung zwischen Projekte und Mitarbeiter

| Pro.Nr | Bezeichnung |
|---|---|
| **4711** | Artikeldatenaustausch |
| **4712** | Neue Finanzbuchhaltung |
| **4713** | Bilanzanalyse |

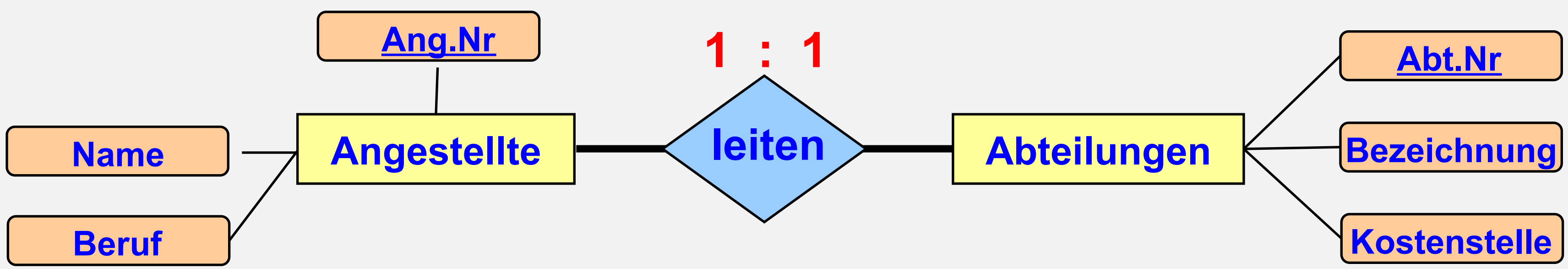
| Ang.Nr | Name | Beruf |
|---|---|---|
| 42203 | Schmitt | Informatikerin |
| 42213 | Peters | DV-Kaufmann |
| 41217 | Müller | Bilanzbuchhalter |

| Pro.Nr | Ang.Nr. | Anteil |
|---|---|---|
| 4712 | 41217 | 20% |
| 4712 | 42213 | 15% |
| 4711 | 42203 | 25% |

# Umsetzung von 1:1 - Beziehungen

- Bei 1:1 – Beziehungen besteht augenscheinlich die freie Wahl, welchem Objekttyp die Informationen aus dem Beziehungstyp zugeordnet werden. Meist ist die Lösung aber eindeutig.

- Beispiel: Beziehungstyp Abteilungsleiter (siehe Übungsblatt)




Angestellte: <Ang.Nr, Name, Beruf, Abt.Nr *(nur bei Abteilungsleitern)*>
Abteilungen: <Abt.Nr, Bezeichnung, Kostenstelle>

**oder**

Angestellte : < Ang.Nr, Name, Beruf, AbtNr >
Abteilungen: < Abt.Nr, Bezeichnung, Kostenstelle, Ang.Nr *(des Abteilungsleiters)*>

**Fremdschlüssel wo abbilden?**

# Vertiefung Fremdschlüssel (1)

- Als **Fremdschlüssel** (**foreign key**) bezeichnet man die Attribute eines Relationenschemas, die in einem anderen Relationenschema Schlüsseleigenschaft (Primärschlüssel oder sonstiger Schlüsselkandidat) haben. Der Fremdschlüssel **referenziert** dabei eine andere Relation bzw. Tabelle.

- Auszug aus dem Relationenschema DH
  - Studenten (MatrNr, Name, Sem)
  - **Vorlesungen** (VorlNr, Titel, SWS)

    **referenzierte Tabelle**
  - Professoren (PersNr, Name, Rang, Raum)
  - Studenten hören Vorlesungen (MatrNr, VorlNr)
  - Dozenten **halten** Vorlesungen: (PersNr, VorlNr)

    **referenzierende Tabelle**

# Vertiefung Fremdschlüssel (2)

- **Namensgleichheit** zwischen Fremdschlüssel und Schlüssel in der referenzierten Tabelle ist nicht notwendig; zwingend ist jedoch derselbe Wertebereich.

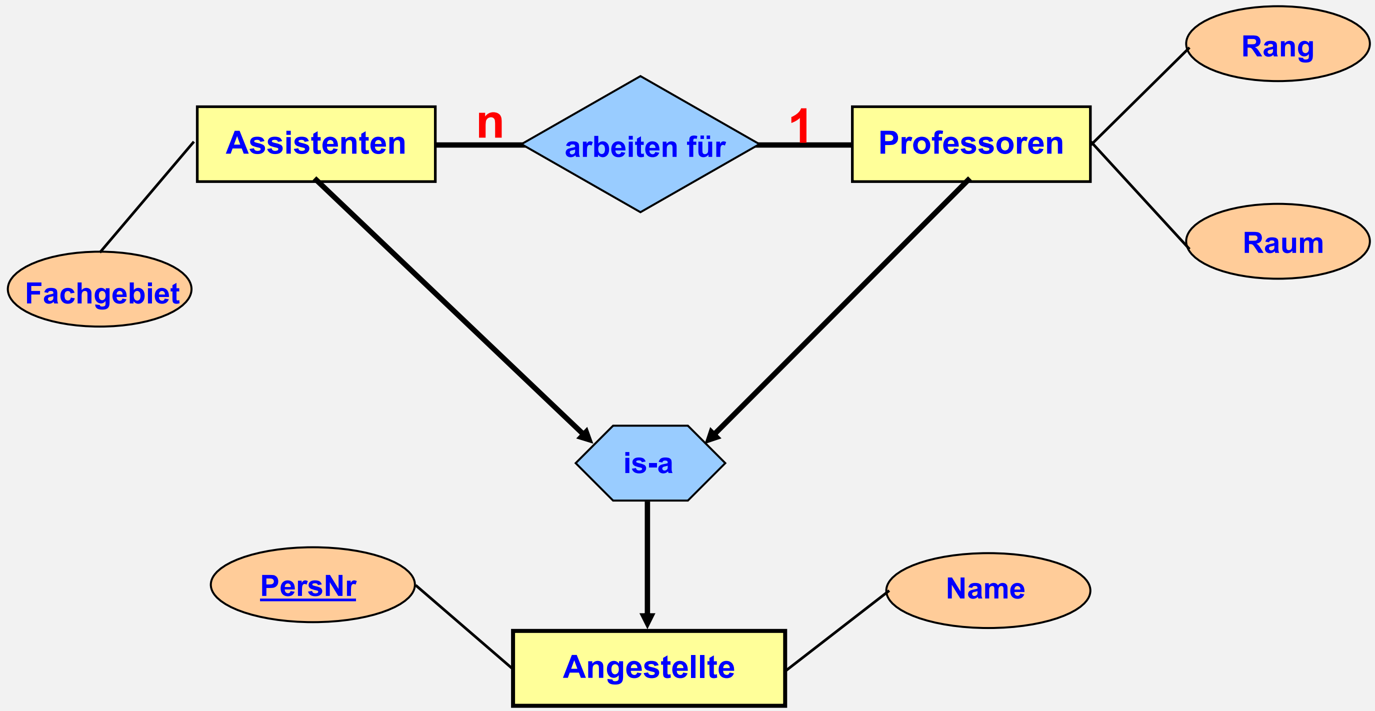
- Ein **Fremdschlüsselwert darf nur dann NULL** sein, wenn er nicht zum Primärschlüssel der referenzierenden Tabelle gehört.

- Ein Relationenschema kann durchaus **mehrere Fremdschlüssel** enthalten.

- In der Spalte des Fremdschlüssels des referenzierenden Tabelle können - außer NULL - nur Werte enthalten sein, die auch im Schlüssel der referenzierten Tabelle vorhanden sind.

# Relationale Modellierung der Generalisierung?

- Gegeben sei folgender Ausschnitt eines E/R-Diagramms:




- Folge: Die relevanten Informationen eines konkreten Entities sind auf zwei Tupel in verschiedenen Relationen verteilt. Das ist ineffizient und daher wird die Generalisierung nicht umgesetzt.

# Nullwerte

- In der Praxis kann es vorkommen, dass bestimmte Attribut-werte nicht eingetragen werden können, z.B. weil ein Wert
  - (noch) nicht bekannt ist, z.B. ein neuer Mitarbeiter ist noch keiner Abteilung zugeordnet
  - nicht anwendbar ist (der Geschäftsführer ist in keiner Abteilung)
- Daher muss es möglich sein, den Wertebereich von Attribu-ten um einen sogenannten Nullwert zu ergänzen.
- Dieser Nullwert
  - ist zu unterscheiden von der Zahl 0 und dem Leerzeichen
  - darf nicht für Schlüsselattribute zulässig sein.
  - kann bei Abfragen zu Problemen führen (dazu später mehr)

# Vermeidung von Nullwerten

- Bei 1:1- und 1:n-Beziehungen muss man berücksichtigen, dass einige oder auch sehr viele Entities der betreffenden Entity-Typen keine Beziehungen eingehen.
- Hierdurch entstehen Nullwerte in den Fremdschlüsseln, die es zu vermeiden gilt.
- Beispiel (in Anlehnung an Kem-09, S. 81):
  - Menschen können haben einen Erstwohnsitz, der in einem Bundesland liegt.
  - Menschen können Ministerpräsident eines Bundeslandes werden und sie können Mitglieder des Landtags eines Bundeslandes sein.
  - Bundesländer haben einen Menschen als Ministerpräsidenten
  - **Aufgabe: Zeichnen Sie ein entsprechendes E/R-Diagramm**

# Fortführung des Beispiels

- Mach den bisher besprochenen Regeln zur Umwandlung von E/R-Modellen, darf man den Beziehungstyp als Fremd-schlüssel auch in der Relation abbilden, die den Entity-Typ repräsentiert, der „gegenüber der 1“ liegt.
- Hierdurch würde folgende Relation „Menschen“ entstehen:

| Nummer | Name | Wohnsitz | MdL von | MP von |
|---|---|---|---|---|
| 4711 | Müller | Baden-Württemberg | NULL | NULL |
| 4712 | Schmitt | Baden-Württemberg | NULL | NULL |
| 4713 | Weber | Baden-Württemberg | NULL | NULL |
| 4714 | Strobl | Baden-Württemberg | Baden-W. | NULL |
| 4715 | Kretschmann | Baden-Württemberg | Baden-W. | Baden-W. |

- Wie sollte der Nachteil diese Modellierung gelöst werden?

# Operatoren der relationalen Algebra

| | |
|---|---|
| $\sigma$ | **Selektion** |
| $\pi$ | **Projektion** |
| $\rho$ | **Umbenennung** |
| $\cup$ | **Vereinigung** |
| $\cap$ | **Durchschnitt** |
| $-$ | **Differenz** |
| $\div$ | **Division** |

| | |
|---|---|
| $\times$ | **Kreuzprodukt** |
| $\bowtie$ | **Join (Verbund)** |
| $\rtimes$ | **rechter Semi-Join** |
| $\ltimes$ | **linker Semi-Join** |
| ⟗ | **vollständiger äußerer Join** |
| ⟕ | **linker äußerer Join** |
| ⟖ | **rechter äußerer Join** |

# Notationen im Zusammenhang mit der relationale Algebra

- R,S — Namen von Relationen werden mit großen Buchstaben bezeichnet
- r,s — aktuelle Werte (Ausprägungen) von Relationen zu einem Zeitpunkt t
- A,B,C — Attributmengen
- a,b,c, — Attribute – einzelne Spalten einer Tabelle
- x,y — Tupel – die Zeilen einer Tabelle, die konkrete Entities repräsentieren
- x.a — a-Wert des Tupels x

# Selektion (1)

- Bei der Selektion werden diejenigen Tupel einer Relation gewählt, die eine bestimmte Selektionsbedingung erfüllen. Eine Selektion wird mit $\sigma$ bezeichnet und hat als Subskript das sogenannte Selektionsprädikat.

- Die Vorgehensweise kann man sich so veranschaulichen, dass die einzelnen Tupel der Ausgangsrelation sukzessive mit der Selektionsbedingung verglichen werden. Bei „wahr" wird das betreffende Tupel in die Ergebnisrelation kopiert.

- Das Selektionsprädikat ist eine Formel F bestehend aus
  - Attributnamen der Argumentrelation und / oder Konstanten
  - arithmetischen Vergleichsoperatoren =, <, $\leq$, >, $\geq$, $\neq$
  - logischen Operatoren $\wedge$, $\vee$, $\neg$

# Selektion (2)

- Das Ergebnis einer Selektion auf r ist wiederum eine Relation. Diese besteht aus allen Tupeln x, die die Bedingung F erfüllen

- Etwas formaler beschrieben gilt:
  $\sigma_F(r) = \{x \in r \mid x \text{ erfüllt } F\}$ mit F ist logischer Ausdruck wie zuvor definiert.

- Beispielsweise könnte eine Selektion zur Ermittlung der „Langzeitstudenten“ wie folgt definiert sein:

$$\sigma_{Sem>9}(\text{Studenten})$$

# Projektion

- Während bei der Selektion einzelne Tupel einer Relation extrahiert werden, geht es bei einer Projektion darum, Spalten, d.h. Attribute, auszuwählen.
- Die Projektion hat das Symbol $\pi$ und als Argument eine Menge von gültigen Attributnamen im Subskript.
- Es ist unbedingt zu beachten, dass durch eine Beschränkung auf eine Teilmenge der Attribute der Argumentrelation Duplikate auftreten können. Diese werden automatisch in der Ergebnisrelation „gestrichen".
- Beispiel: $\pi_{\text{Rang}}$ (Professoren)

# Vereinigung, Durchschnitt und Differenz (1)

- Zwei Relationen r und s mit identischem Schema, d.h. denselben Attributnamen und gleichen Wertebereichen, kann man durch die Vereinigung zu einer Relation zusammenfassen. Die Ergebnisrelation enthält alle Tupel **die entweder in r oder in s enthalten** sind.

- Unter dieser Voraussetzung kann man auch den Durchschnitt von r und s bilden, d.h. eine Ergebnisrelation erzeugen mit allen Tupeln, **die in r und auch in s enthalten** sind.

- Für zwei Relationen r und s mit identischem Schema kann man auch die Mengendifferenz r– s bilden, d.h. die Menge aller Tupel, **die in der Relation r aber nicht in s** vorkommen.

# Vereinigung, Durchschnitt und Differenz (2)

- Beispiel:
  "Finde alle Professoren mit C4-Gehalt, die mindestens eine Vorlesung halten" formuliert in der relationalen Algebra:

$$\pi_{PersNr}(Vorlesungen)) \cap \pi_{PersNr}(\sigma_{Rang=C4}(Professoren))$$

- Gegeben seien die folgenden Relationen:

| r | a | b | c |
|---|---|---|---|
| | a1 | b1 | c1 |
| | a1 | b1 | c2 |
| | a2 | b2 | c2 |

| s | a | b | c |
|---|---|---|---|
| | a2 | b2 | c1 |
| | a2 | b2 | c2 |

**Bestimmen Sie bitte $r \cup s$ , $r \cap s$ und $r - s$**

# Kreuzprodukt (kartesisches Produkt)

- Das **Kreuzprodukt** zweier Relationen r und s ist definiert als die Menge aller möglichen Paare von Tupeln aus r und s und wird mit r x s bezeichnet.

- Das Schema der Ergebnisrelation ist eine **Vereinigung der Attributmengen** von r und s.

- Beim Kreuzprodukt kann es vorkommen, dass die zugrunde-liegenden Argumentrelationen Attribute mit identischem Namen beinhalten. Diese sind entweder umzubenennen oder durch den Relationennamen gefolgt von einem Punkt ein-deutig zu identifizieren Dieses bezeichnet man auch als **qualifizierender Attributname**.

# Einfaches Beispiel zum Kreuzprodukt

- Gegeben seien die folgenden Relationen:

| r | a | b | c |
|---|---|---|---|
| | a1 | b1 | c1 |
| | a1 | b1 | c2 |
| | a2 | b2 | c2 |

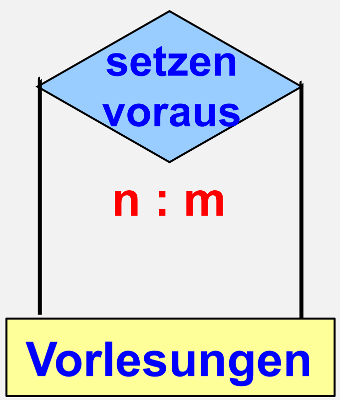
| s | d |
|---|---|
| | d1 |
| | d2 |

Bitte bestimmen Sie

| r x s | a | b | c | d |
|---|---|---|---|---|
| | a1 | b1 | c1 | d1 |
| | a1 | b1 | c1 | d2 |
| | a1 | b1 | c2 | d1 |
| | a1 | b1 | c2 | d2 |
| | a2 | b2 | c2 | d1 |
| | a2 | b2 | c2 | d2 |

# Komplexes Beispiel zum Kreuzprodukt (1)

- Gesucht werden die indirekten Vorgänger (also die Vorvorgänger) einer Vorlesung. Umgangssprachlich sind das also diejenigen Vorlesungen, die man besuchen muss, um die Voraussetzungen zu Besuch der eigentlich gewünschten Vorlesung zu erfüllen. Beispiel:




| Vorgänger | Nachfolger |
|---|---|
| Mathe I | Mathe II |
| Mathe II | Mathe III |
| Mathe I | DB I |
| DB I | DB II |

- In diesem Fall ist es erforderlich, dieselbe Relation mehrfach in einer Abfrage zu verwenden. Hierbei werden – zumindest logisch – zwei vollständige Kopien der Relation erzeugt.

# Komplexes Beispiel zum Kreuzprodukt (2)

- Aus den Tupeln der Ergebnisrelation werden anschließend die-jenigen selektiert, für die die beiden folgenden Bedingungen gelten:
  - **V2.Nachfolger = DB II und**
  - **V1.Nachfolger = V2.Vorgänger**
- Auf dieser Ergebnisrelation muss dann noch eine Projektion auf das Attribut V1.Vorgänger erfolgen. Das auf das Beispiel bezogene Ergebnis ist dann ein Tupel mit dem Attribut Mathe I.
- Hieraus ergibt sich als Ausdruck auf Basis der relationalen Algebra

$$\pi_{\text{V1.Vorgänger}}\,(\sigma_{\text{V2.Nachfolger = DB II} \wedge \text{V1.Nachfolger= V2.Vorgänger}}\,(\rho_{V1}(\text{voraussetzen}) \times \rho_{V2}(\text{voraussetzen})))$$

# Umbenennung von Relationen und Attributen

- Der ρ–Operator wurde im vorangegangenen Beispiel zur Umbenennung der Relation voraussetzen verwendet.
- Es ist mit dem ρ–Operator auch möglich, einzelne Attribute umzubenennen. Da Relationen i.a. mehrere Attribute besitzen, muss das umzubenennende Attribut im Subskript angegeben werden.
- Beispiel: Mit

  $$\rho_{\text{Boss} \leftarrow \text{Professoren.PersNr}}\ (\text{Assistenten})$$

  wird der Fremdschlüssel auf die Personalnummer des vorgesetzten Professors in der Relation Assistenten in Boss umbenannt.

# Der relationale Verbund (Join)

- Der Nachteil des Kreuzproduktes zweier Relationen mit n bzw. m Tupeln besteht darin, dass die Ergebnismenge n * m Tupel enthält. Die meisten dieser Tupel sind völlig irrelevant und "vergrößern" nur unnötigerweise die Ergebnisrelation.

- Umgangssprachlich bedeutet ein Join
  - **Zusammensetzen von Tabellen** bezüglich gleicher Spalten und
  - **Zusammenfügen der Tupel**, die denselben Wert bezüglich der betrachteten Spalten haben.

- Man unterscheidet hierbei
  - Natürlicher Verbund (Natural-Join)
  - Allgemeiner Verbund (Theta-Join)
  - Äußerer Verbund (Outer-Join)
  - Semi-Join

# Natürlicher Verbund (auch Inner-/ Natural-Join)

- Gegeben seinen zwei Relationen r(A) und s(B) mit $A \cap B \neq \emptyset$,

  wobei r

  - m + k Attribute $a_1,a_2,...,a_m,b_1,.b_2,...b_k$ enthält

  und s

  - k + n Attribute $b_1,b_2,...,b_k,c_1,c_2,...,c_n$ enthält.

dann hat $r \bowtie s$ die Stelligkeit **m + n + k**.
Hierbei wird angenommen, dass die Attribute $a_i$ und $c_j$ für $1 <= i <= m$ und $1 <= j<= n$ jeweils paarweise unterschiedlich sind. M.a.W. haben r und s genau $b_1,..,b_k$ gleichbenannte Attribute.

# Natürlicher Verbund (auch Inner-/ Natural-Join)

- In diesem Fall ist $r \bowtie s$ wie folgt definiert:

  $r \bowtie s = \pi a_1,\ldots a_m, r.b_1,\ldots r.b_k, c_1,\ldots c_n \, (\sigma_{r.b1=s.b1 \wedge \ldots \wedge r.bk=s.bk}(r \times s))$

- Genauer betrachtet wird also das Kreuzprodukt von r und s gebildet und aus der Ergebnisrelation nur diejenigen Tupel selektiert, deren Attributwerte für gleichbenannte Attribute der beiden Argumentrelationen r und s identisch sind. Weiterhin werden gleichbenannte Attribute in das Ergebnis nur einmal übernommen, was die nachgeschaltete Projektion bewirkt.

- Weiterhin gilt
  - $r \bowtie s = s \bowtie r$ (Kommutativgesetz)
  - $(r \bowtie s) \bowtie t = r \bowtie (s \bowtie t)$ (Assoziativgesetz)

# Beispiel zum natürlichen Verbund

- Gegeben seien folgende Relationen:

| r | a | b | c |
|---|---|---|---|
| | a1 | b1 | c1 |
| | a2 | b2 | c2 |
| | a1 | b3 | c3 |

| s | a | d |
|---|---|---|
| | a1 | d1 |
| | a1 | d2 |
| | a2 | d3 |
| | a2 | d4 |
| | a2 | d5 |
| | a3 | d5 |

- Bitte bestimmen Sie $r \bowtie s$.

# Allgemeiner Join (Theta-Join)

JOIN alle Operatoren auf alle Felder möglich (bis auf IST GLEICH) siehe: https://help.sap.com/docs/SAP_BUSINESSOBJECTS_BUSINESS_INTELLIGENCE_PLATFORM/512fca6758c4495bb6a50fe3e1e4b892/464ee9ff6e041014910aba7db0e91070.html

- Beim zuvor behandelten natürlichen Join wurden alle gleichbenannten Attribute der Argumentrelationen betrachtet und Tupel, die bei diesen Attributen gleiche Werte aufwiesen, in die Ergebnisrelation übernommen.
- Allerdings ist für einen Join nicht Namensgleichheit der Spalten notwendig, sondern nur die Gleichheit der Wertebereiche. Dies bezeichnet man auch als **Verbundkompatibilität**.
- Der Theta-Join erlaubt die Spezifikation eines beliebigen Joinprädikates $\theta$.
- Außerdem sind beim Theta-Join folgende Vergleichsoperatoren zulässig: $\theta \in \{ =, >, <, <=, >=, \neq \}$

# Allgemeiner Join (Theta-Join)

- Gleich benannte Attribute werden durch den Namen ihrer Ursprungsrelation näher spezifiziert.

- Ein Theta-Join zwischen zwei Relationen r und s ist identisch zu dem kartesischen Produkt gefolgt von einer Selektion. Es gilt

  $$r\ [a\ \theta\ b]\ s = \sigma\theta(r \times s)$$

  wobei $\theta$ das Joinprädikat ist und $a \in A$ sowie $b \in B$.

- **Aufgabe** zum Schema Duale Hochschule: Gesucht sind die Assistenten, welche mehr Geld verdienen, als ihre unmittelbar vorgesetzten Professoren:

  Assistenten.Gehalt > Professoren.Gehalt && Assistenten.ChefPersNr = Professor.PersNr

# Outer-Joins

- Bei den bisher vorgestellten Join-Operatoren gehen diejenigen Tupel der Argumentrelationen verloren, die keinen "Join-Partner" haben. Diese Join-Operatoren bezeichnet man auch als **innere Joins**.

- Daneben gibt es sog. **äußere Joins**, bei denen "partnerlose" Tupel der linken oder rechten bzw. beider Argumentrelationen in die Ergebnisrelation übernommen werden. Man unterscheidet
  - den linken äußeren Join, bei dem die Tupel der linken Argumentrelation auf jeden Fall erhalten bleiben
  - den rechten äußeren Join, bei dem die Tupel der rechten Argumentrelation auf jeden Fall erhalten bleiben
  - Den (vollständigen) äußeren Join: In diesem Fall bleiben die Tupel beider Argumentrelationen auf jeden Fall erhalten.

# Semi-Joins

- Umgangssprachlich formuliert ist das Ergebnis des linken (bzw. rechten) Semi-Joins die Menge aller Tupel der linken (bzw. rechten) Argumentrelation in unveränderter Form, die einen potentiellen Joinpartner in der rechten (bzw. linken) Argumentrelation hätten.

- Formal definiert: Sei A die Menge der Attribute von r und B die Menge der Attribute von s. Dann heißt

  $r \ltimes s = \pi_A(r \bowtie s)$ linker Semi-Join (bzw. Semi-Join von r mit s)

  $r \rtimes s = \pi_B(r \bowtie s)$ rechter Semi-Join (Semi-Join von s mit r)

- Beispiele Outer- und Semi-Joins in den Übungsaufgaben

# Die relationale Division (1)

- Das Ergebnis der relationalen Division R÷S enthält nur die Attribute von R, die nicht in S enthalten sind. Unter diesen Voraussetzungen ist ein Tupel t in R÷S enthalten, falls es für jedes Tupel $t_s$ aus S ein Tupel $t_r$ aus R gibt, so dass die beiden Bedingungen
  - $t_r.S = t_s.S$
  - $T_r(R-S) = t$

  erfüllt sind.

Ergebnis: nur noch spalte a, da b sowohl in r als auch in s ist; nur noch a1 aus der 3. Zeile, da da nicht b1 und b2 drin ist; Frage ist Zeile 5?

- Abstraktes Beispiel:

r

| a | b |
|---|---|
| a1 | b1 |
| a1 | b2 |
| a1 | b3 |
| a2 | b2 |
| a2 | b3 |

÷

s

| b |
|---|
| b1 |
| b2 |

=

r÷s

| a |
|---|
| a1 |
| |

# Die relationale Division (2)

- Praxisbeispiel: Durch eine Anfrage sollen die Matrikelnummern derjenigen Studenten ermittelt werden, die alle vierstündigen Vorlesungen belegt haben.
  - Zunächst müssen die VorlNr der vierstündigen Vorlesungen ermittelt werden:

    $$R = \pi_{\text{VorlNr}}(\sigma_{\text{SWS=4}}(\text{Vorlesungen}))$$

  - Anschließend lassen sich aus der Relation hören die MatrNr der Studenten ermitteln, die alle in R enthaltenen Vorlesungen gehört haben. Das erfolgt mit Hilfe des Divisionsoperators:

    $$\text{Hören} \div \pi_{\text{VorlNr}}(\sigma_{\text{SWS=4}}(\text{Vorlesungen}))$$

- Für die Durchführung der Division zweier Relationen $R \div S$ muss gelten, dass S eine Teilmenge von R ist – also $S \subseteq R$ erfüllt. Zusätzlich ist das Schema der Ergebnisrelation mit R – S definiert.