Institut für Programmstrukturen und Datenorganisation (IPD)
Lehrstuhl für Systeme der Informationsverwaltung, Prof. Böhm

Kommunikation und Datenhaltung

Relationenmodell und Relationenalgebra

Überblick über den Datenhaltungsteil

- Einleitung
  - Motivation und Grundlagen
  - Architektur von Datenbanksystemen
- Datenbankanfragen
  - Relationenmodell und Relationenalgebra
  - Relationale Datenbanksprachen (SQL)
- Datenbankentwurf
  - ER- und EER-Modell
  - Abbildung von ER-Modellen auf das Relationenmodell
  - Relationaler Entwurf
  - Sprachen zur Datenbankdefinition
- Transaktionsverwaltung
- Anfrageoptimierung
- Datenbankanwendungsentwicklung

Einordnung dieses Kapitels

- Präsentation des Relationalen Modells (RM)
  - Grundlage relationaler Datenbanken
- „Daten(bank)modell“
- Etwas irreführender Begriff
  - Modellierung der Anwendungswelt (um deren Zustand in DB zu erfassen)
  - Nicht der DB (oder der Daten)
- Entwurfsmodelle vs. Realisierungsmodelle
  - Entwurfsmodell: Entity-Relationship Modell (ER), UML
    - Kommunikation zwischen DB-Entwickler und Anwender/Auftraggeber
  - Realisierungsmodell: RM, etc.
    - Umsetzung des Entwurfs

Einordnung dieses Kapitels

- Im Folgenden:
  - Grundlegende Überlegungen zu Datenmodellen
  - Vorstellung des RM
  - Kapitel 4: relationale Sprache SQL
- Wir nähern uns DBs von der Nutzerschnittstelle (vgl. Kapitel „Architektur“)
  - Entwurfsmodelle folgen später

Datenmodelle

Sinn/Zweck:

Modellierung der Anwendungswelt (um Zustand in DB zu erfassen)

Ein Datenmodell legt fest…

1. statische Eigenschaften der Anwendungsdomäne durch
   a) Objekte,
   b) Beziehungen
      inklusive der Standard-Datentypen,
      die Daten über die Beziehungen und Objekte darstellen können,
2. dynamische Eigenschaften der Anwendungsdomäne durch
   a) Operationen,
   b) Beziehungen zwischen Operationen,
3. Integritätsbedingungen für
   a) Objekte
   b) Operationen

Semantik von Datenmodellen

- Daten sind Repräsentanten von Zuständen
- Datenmodell beschreibt Anwendungswelt

$\Rightarrow$ Datenmodell beschreibt zulässige Zustände und zulässige Zustandsübergänge der Datenbasis

- Intuition:

  Semantik des Datenmodells: Welche Zustände kann Datenbasis annehmen?

- Festlegung der Semantik:
  - *Datenbankzustand* – Welche Zustände kann Datenbasis zu einem bestimmten Zeitpunkt annehmen?
  - Zustandsübergänge – Folge von Datenbank-Zuständen, die die Datenbasis über die Zeit hinweg annehmen kann

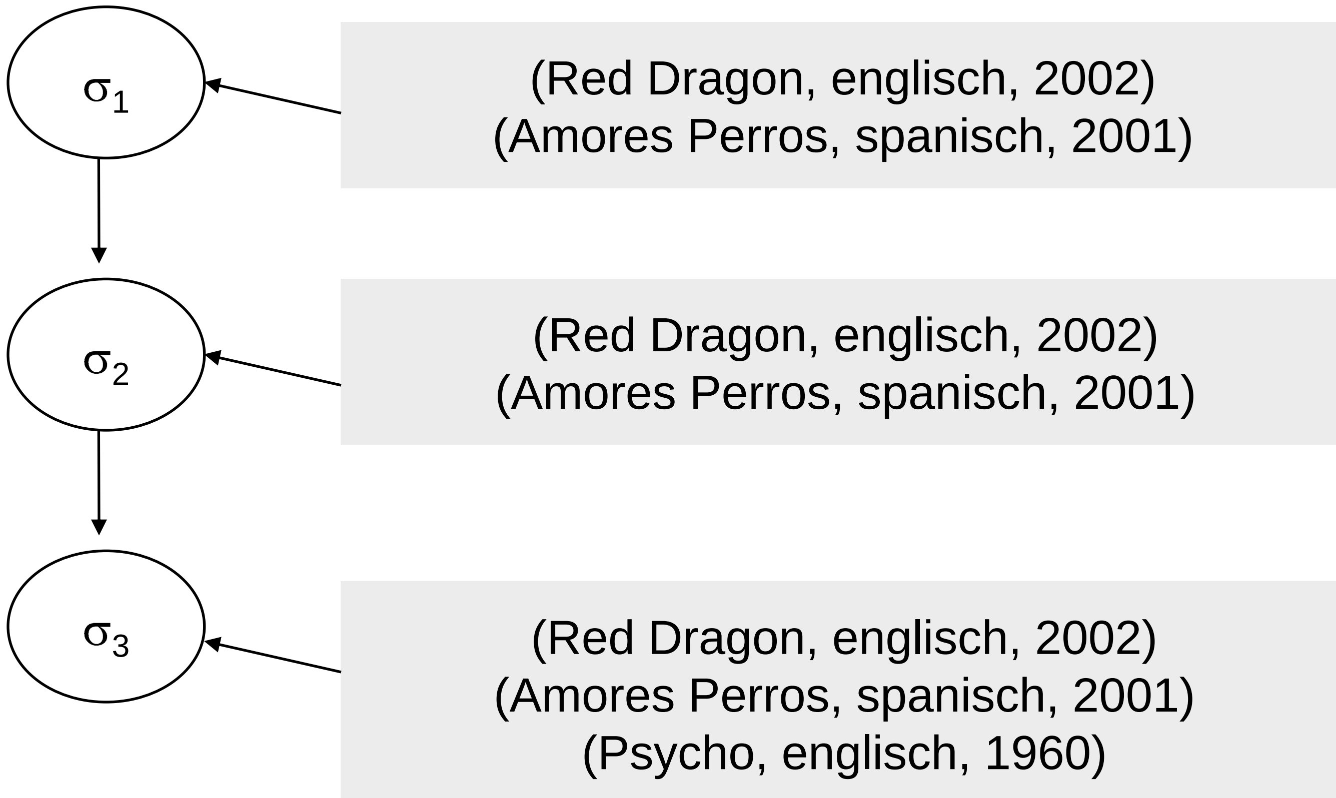
Zustandsfolgen

Einordnung

Datenmodelle

- Semantik
- Semantikbsp.
- Überblick

RM



Semantikfestlegung am Beispiel I

Einordnung
Datenmodelle
Semantik
Semantikbsp.
Überblick
RM

Ausgangspunkt:

Datenmodell, für welches wir die Semantik festlegen wollen (stark vereinfachte Version des relationalen Modells):

- Atomare Typen: **integer**, **string**
- Typkonstruktoren:
  - **datensatz ::= tuple**(sel:**atomarerTyp**, ...,sel:**atomarerTyp**)
  - **menge ::= set**(**datensatz**)

Selektor (Feldname)

- „Polymorphes Typsystem" – Instanziierung = Definition eines konkreten („monomorphen") Typsystems
  - Hintergrund: Generizität von Datenbanken
    - Komplexe Systeme
    - Wirtschaftlichkeit durch große Installationszahl
  - Erfordert generische Datenmodelle, die auf die jeweiligen Anwendungsdomänen zugeschnitten werden können

Semantikfestlegung am Beispiel II

Einordnung
Datenmodelle
Semantik
Semantikbsp.
Überblick
RM

Beispiel einer Modellierung mit diesem Modell:

- Eine DVD wird beschrieben durch Namen, Sprache, Erscheinungsjahr
- Typ DVD ::= tuple(Name:string, Sprache:string, Jahr:integer)
- Typ DVDs ::= set(DVD)
- DVD und DVDs sind Beispiele für monomorphe Typen

Anlegen einer DB-Variable („Relation“):

- Um eine Datenbasis zu erhalten müsste man theoretisch den Typen noch instanziieren (Vgl. Variablendeklarationen in Programmiersprachen: Var_Name:Var_Typ)
- Dieser letzte Schritt erübrigt sich in relationalen DBs – sämtliche Tupel desselben Typs werden in einer Relation zusammengefasst, die implizit den Namen des Typs übernimmt, d.h., es gilt implizit DVDs:DVDs

Möglicher Datenbasiszustand:

(DB-Variable DVDs)

(Red Dragon, englisch, 2002)
(Amores Perros, spanisch, 2001)
(Psycho, englisch, 1960)

Semantikfestlegung am Beispiel III

Einordnung
Datenmodelle
Semantik
Semantikbsp.
Überblick
RM

Für Datenmodell lässt sich Semantik rekursiv definieren:

1. Für jeden Datentyp wird eine Trägermenge μ festgelegt (mögliche Zustände dieses Typs)

- μ(**integer**) = **Z** (die ganzen Zahlen **Z**),
- μ(**string**) = C* (Folgen von Zeichen aus C={a, b, ..., z, A, B, ..., Z}),
- μ(**set**(z)) = $2^{\mu(z)}$
  (die Potenzmenge über den Werten des Parameterdatentyps z, oder, anders ausgedrückt, die Mengen aller Teilmengen von möglichen Werten in μ(z)),
- μ(**tuple**(sel:z_1, ..., sel:z_n)) = $\mu(z_1) \times \ldots \times \mu(z_n)$
  (das kartesische Produkt der Parameterwertebereiche).

Semantikfestlegung am Beispiel IV

2. Semantik der zeitlichen Entwicklung (Zustandsfolge)

- Intuition:
  - Sehr einfaches Datenmodell ⇒ keine Möglichkeiten, die Menge der zulässigen Zustände/Zustandsübergänge einzuschränken
  - zu jedem Zeitpunkt ist der gesamte durch den Datentypen festgelegte Zustandsraum zulässig

- Modellierung der Datenbankentwicklung:

$$\hat{\sigma} = <\sigma_0, \sigma_1, \ldots, \sigma_i, \ldots>$$

- Bedeutung einer DB-Variablen ($\mathcal{T}$: Zeitachse):

$$\hat{\sigma}(\mathrm{db}): \mathcal{T} \to \mu(\mathbf{typ}(\mathrm{db}))$$

Semantikfestlegung am Beispiel V

Einordnung
Datenmodelle
Semantik
Semantikbsp.
Überblick
RM

- Beispiel der DVD-Datenbank:
  - μ(**typ**(DVDs)) =
    μ(set(tuple(Name:string, Sprache:string, Jahr:integer)))
    $= 2^{C^{*} \times C^{*} \times Z}$
  - $\hat{\sigma}(\text{DVDs})\ \mathcal{T} \rightarrow 2^{C^{*} \times C^{*} \times Z}$

- Ein konkreter Zustandswert zum Zeitpunkt 42:
  $\hat{\sigma}$(DVDs)(42)={(Red Dragon, englisch, 2002), (Amores Perros, spanisch, 2001)}

Integritätsbedingungen

Einordnung
Datenmodelle
Semantik
Semantikbsp.
Überblick
RM

- Waren im Beispiel-Datenmodell nicht vorhanden
  - Schränken sowohl Zustände wie auch Zustandsübergänge weiter ein
- Datenbankzustände – Beispiel: „Kein Mitarbeiter verdient mehr als sein Vorgesetzter."
- Zustandsfolgen – Beispiel: „Gehälter dürfen nur ansteigen."

Rückblick: Anfang dieses Kapitels

Einordnung
Datenmodelle
Semantik
Semantikbsp.
Überblick
RM

- Datenmodelle zur Modellierung der Anwendungsdomäne (letztlich zur Erfassung in der DB)
- Unterscheidung von Entwurfsmodellen und Realisierungsmodellen; - dieses Kapitel: Realisierungsmodelle
- Datenmodell besteht aus 3 Teilen:
  - statische Eigenschaften (Objekte und Beziehungen)
  - dynamische Eigenschaften (Operatoren sowie Beziehungen zwischen Operatoren)
  - Integritätsbedingungen
- Semantik eines Datenmodells = zulässige Zustände/ zulässige Zustandsübergänge einer Datenbasis zu jedem Zeitpunkt
  - DB-Schema = Datentyp
  - mögliche Zustände = Domäne dieses Datentyps
  - Relation entspricht im Prinzip einer deklarierten Variablen dieses Typs – aber: bei relationalen Systemen: Deklaration implizit; Variable (=Relation) heißt wie der definierte Typ

Praxis der (Realisierung-)Datenmodelle

Einordnung
Datenmodelle
Semantik
Semantikbsp.
Überblick
RM

- Entwicklung eines Datenmodells verlangt Kompromisse:
  - Universelle Anwendbarkeit vs. Zuschnitt auf spezielle Modellierungsbedürfnisse
  - Ausdrucksmächtigkeit vs. Effizienz der Implementierung
- Heute eingesetzte Datenmodelle:
  - Hierarchisches und Netzwerk-Datenmodell (Altanwendungen)
  - Relationales Datenmodell (Sieg des Kompromisses)
  - Objektorientiertes Datenmodell (für anspruchsvollere Modellierungsbedürfnisse)
  - Objektrelationales Modell (Synthese der besten Eigenschaften)
  - XML (für Datenaustausch, Internetanwendungen)

Agenda – Relationenmodell und -algebra

- Relationenmodell
  - Veranschaulichung
  - Formalisierung
  - Integritätsbedingungen
- Relationenalgebra
  - Einleitung
  - Projektion
  - Selektion
  - Join
  - Mengenoperationen
  - Vollständigkeit

Abgrenzung der Darstellung

Einordnung
Datenmodelle
RM
Struktur
Formal
Integrität
Algebra

- Im Folgenden: Darstellung/Formalisierung des Relationenmodells
- Wir gehen davon aus, ein Relationenschema gegeben zu haben und legen dessen Semantik fest
  - Verschieben das Thema „Datendefinition“ auf Kap 8
- Alternative Darstellung (z.B. „Verteilte Datenhaltung“, Abeck, Lockemann, Schiller, Seitz):
  - Einführung des Relationenmodells über ein Polymorphes Typsystem, mit dessen Hilfe sich ein Relationenschema (= monomorpher Typ) definieren lässt
  - Diese Darstellung schließt also eine Formalisierung der Datendefinition mit ein

$\Rightarrow$ Wir formalisieren nur die Nutzung des Relationenmodells – diese ist essentiell: Abbildung von SQL-Anfragen auf das Relationenmodell zum Zweck der „algebraischen Optimierung“

Beispiele für Relationen

Personen

PANr	Vorname	Nachname	PLZ	Ort	GebDatum
4711	Andreas	Heuer	18209	DBR	31.10.1958
5588	Gunter	Saake	39106	MD	05.10.1960
6834	Michael	Korn	39104	MD	24.09.1974
8832	Tamara	Jagellovsk	38106	BS	11.11.1973
9999	Christa	Loeser	69121	HD	10.05.1969

Pers_Telefon

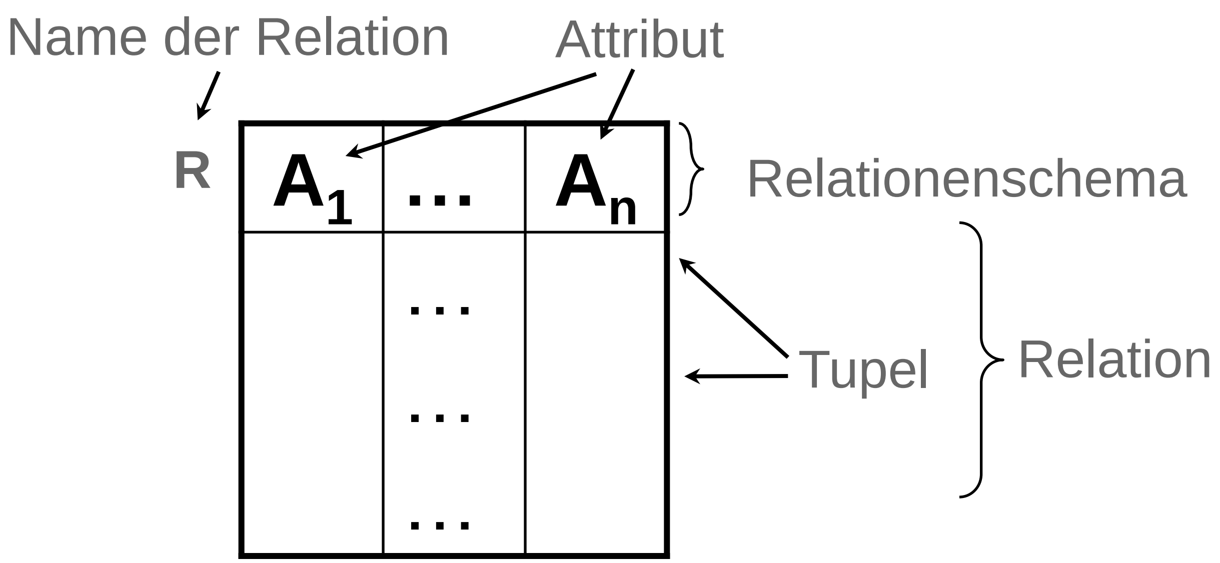
PANr	Telefon
4711	038203-12230
4711	0381-498-3401
5588	0391-345677
5588	0391-5592-3800
9999	06221-400177

- Tabelle = „Relation“

Relationenmodell

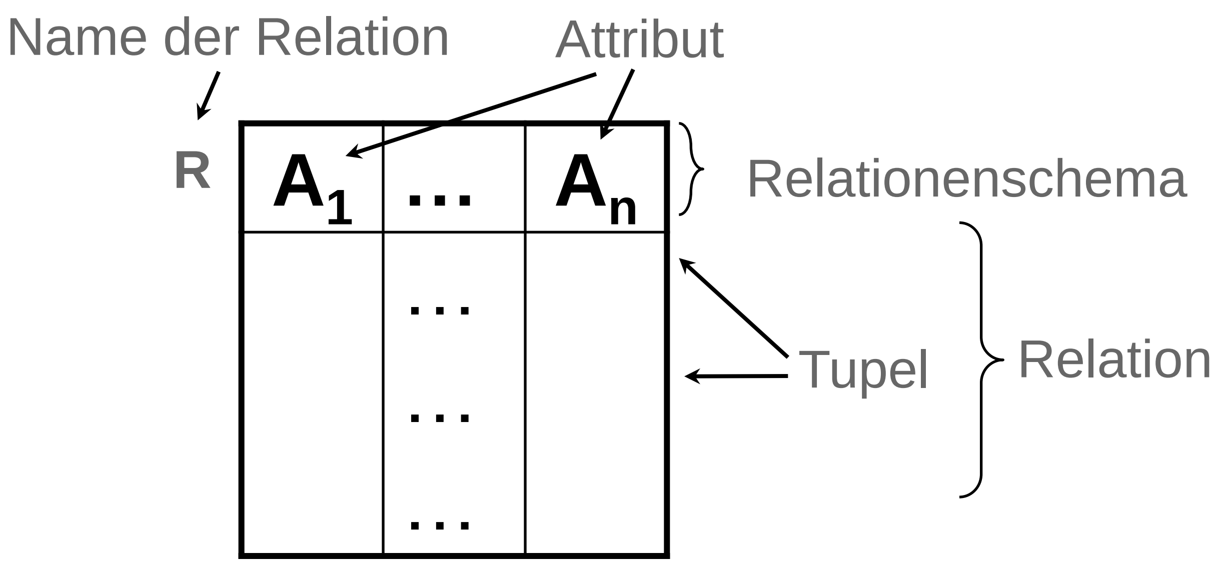
Einordnung
Datenmodelle
RM
Struktur
Formal
Integrität
Algebra

- Erste Zeile: Relationenschema (=Datentyp)
  - Erfasst statische Eigenschaften der Anwendungsdomäne – Objekte und Beziehungen (letzteres zusammen mit referenziellen Integritätsbedingungen, s.u.)
- Spaltenüberschriften: Attribut
  - Gemeinsame Eigenschaften von Objekten/Beziehungen
  - Name + Wertebereich (integer, etc.)



Relationenmodell

- Inhalt der Tabelle: Relation (=Instanz des Typs, also Zustand einer Variablen des Typ/Relationenschemas)
  - Anschaulich: zulässige Zustände = Teilmenge des kart. Produkts
  - Aber nicht ganz korrekt: Relation legt Reihenfolge der Attribute nicht fest (formale Def. folgt) (Warum nicht/ Vorteil?)
- Zeilen der Tabelle: Tupel



Formalisierung: Attribute und Domänen

Einordnung
Datenmodelle
RM
Struktur
Formal
Integrität
Algebra

- U nichtleere, endliche Menge: Universum der Attribute
- $A \in U$: Attribut
- $D \in \{D_1, \ldots, D_m\}$
  Menge endlicher, nicht-leerer Mengen
  - jedes D_i: Domäne
- Total definierte Funktion dom: $U \rightarrow D$
- dom(A): Domäne von A
  $w \in \mathrm{dom}(A)$: Attributwert für A

Attribute und Domänen: Beispiel

Einordnung
Datenmodelle
RM
Struktur
Formal
Integrität
Algebra

- U = {Name, Alter, Haarfarbe, ...}
- $\{D_1, \ldots, D_m\}$ =
  {
  {1, 2, 3, …},
  {wahr, falsch},
  {schwarz, rot, blond}, ...
  }
- dom(Alter) = {1, 2, 3, …},
  dom(Haarfarbe) = {schwarz, rot, blond}
- ‚schwarz' möglicher Attributwert für Haarfarbe

Formalisierung: Relationenschema und Tupel

Einordnung
Datenmodelle
RM
Struktur
Formal
Integrität
Algebra

- $R \subseteq U$: Relationenschema

- Tupel t in $R = \{A_1, ..., A_n\}$ ist Abbildung

$$t : R \rightarrow \cup_{i=1}^{m} D_i$$

- Es gilt: $t(A) \in dom(A)$
  ($t(A)$ ist eine Restriktion von t auf $A \in R$)

- Für $X \subseteq R$ analog: $t(X)$ ist X-Wert von t

Relationenschema und Tupel: Beispiel

Einordnung
Datenmodelle
RM
Struktur
Formal
Integrität
Algebra

r

Name	Alter	Haarfarbe
Andreas	43	blond
Gunter	42	blond
Michael	25	schwarz

- R = {Alter, Haarfarbe, Name}
- Relation r besteht aus Tupeln / Abbildungen t_1, t_2, t_3 mit t_1(Name)='Andreas', t_1(Alter)=43, t_1(Haarfarbe)=blond, t_2(Name)='Gunter', t_2(Alter)=42, ...
- t_1({Name, Alter}) = {Name: Andreas, Alter: 43}
  (Notation beinhaltet Namen des Attributs wegen der nicht festgelegten Reihenfolge)

Formalisierung: Relation

- Relation r über R = $\{A_1, \ldots, A_n\}$ (kurz: r(R)) ist endliche Menge von Tupeln
- Abkürzung: r(R) – r ist Relation von R.
- Menge aller Relationen über R:

  REL(R) := {r | r(R)};

  d. h. Menge aller r, für die gilt:
  r ist Relation von R.

Relation: Beispiel

Einordnung
Datenmodelle
RM
Struktur
Formal
Integrität
Algebra

- Gegeben Relation r

Name	Alter	Haarfarbe
Andreas	43	blond
Gunter	42	blond
Michael	25	schwarz

- r $\in$ REL({Name, Alter, Haarfarbe})
- r $\notin$ REL({Name, Vorname})

Formalisierung: Datenbankschema und Datenbank

Einordnung
Datenmodelle
RM
Struktur
Formal
Integrität
Algebra

- Menge von Relationenschemata $S:=\{R_1, ..., R_p\}$: Datenbankschema

- Datenbank über S:
  Menge von Relationen
  $d:=\{r_1, ..., r_p\}$, und $r_i(R_i)$

- Datenbank d über S: d(S)
  - Bezeichnung

- Relation $r \in d$: Basisrelation

Konsequenzen aus Relationen-Definition

- Relation ist als Menge von Tupeln definiert
- Relationen sind
  - duplikatfrei,
  - ungeordnet,
  - endlich,
  - in sogenannter **1. Normalform**, d.h., alle Attribute haben atomare Typen.
- Jetzt: Integritätsbedingungen
  - Menge der möglichen Instanzen eines Relationenschemas weiter einschränken

Integritätsbedingungen: Schlüssel

- Motivation: Identifizierende Attributmenge
  - Duplikatfreiheit: Keine 2 Tupel können in ALLEN Attributen übereinstimmen
  - Vielleicht wissen wir aber, dass 2 Tupel schon auf einer Teilmenge nicht gleich sein können (Bsp.: Sozialversicherungsnummer)
- Identifizierende Attributmenge:
  $K := \{B_1, \ldots, B_k\} \subseteq R$:
  $\forall\ t_1, t_2 \in r : t_1 \neq t_2 \rightarrow \exists\ B \in K : t_1(B) \neq t_2(B)$
- Schlüssel:
  Minimale identifizierende Attributmenge
  {Vorname, Nachname, PLZ, Geburtsdatum} und {PANr} für Personen {PANr, Telefon} für Pers_Telefon.

- Primattribut: Element eines Schlüssels.
- Primärschlüssel: Ausgezeichneter Schlüssel.

Lokale Integritätsbedingung I

- Jetzt: Formalisierung der Semantik von Integritätsbedingungen; zunächst diejenigen, die sich auf ein Relationsschema beziehen („lokal")
- Abbildung aller möglichen Relationen zu einem Schema auf true oder false, formal:

  $b : \{r \mid r(R)\} \rightarrow \{\text{true}, \text{false}\};\ b \in B$

- Beispiel für lokale Integritätsbedingung: Schlüsselbedingung
- Erweitertes Relationenschema: $\mathcal{R}$:= (R, B)
  - Relationenschema erweitert um lokale Integritätsbedingungen
- Abkürzung: r($\mathcal{R}$) – r ist Relation von R, und b(r) = true für alle $b \in B$.

Lokale Integritätsbedingung II

Einordnung
Datenmodelle
RM
Struktur
Formal
Integrität
Algebra

- $\text{SAT}_R(B) := \{r \mid r(\mathit{R})\}$
  Menge aller Relationen
  über erweitertem Relationenschema *R*
  (‚SAT' ≡ ‚satisfy')

- Man definiert ein <u>lokal erweitertes Datenbankschema</u> (mehrere Relationen) als Menge lokal erweiterter Relationenschemata

Lokale Integritätsbedingung – Beispiel

- r

Name	Alter	Haarfarbe
Andreas	43	blond
Gunter	42	blond
Michael	25	schwarz

- r‘

Name	Alter	Haarfarbe
Andreas	43	blond
Andreas	42	blond
Michael	25	schwarz

- r, r‘ $\in$ REL({Name, Alter, Haarfarbe})
- b_n sei Festlegung, daß Name Schlüssel von R
- $b_n(r)$ = true, $b_n(r')$ = false
- $r \in SAT_{\{Name,\ Alter,\ Haarfarbe\}}(\{b_n\})$,
  $r' \notin SAT_{\{Name,\ Alter,\ Haarfarbe\}}(\{b_n\})$.

Integritätsbedingungen: Fremdschlüsselbeziehung

Personen

PANr	Vorname	Nachname	PLZ	Ort	GebDatum
4711	Andreas	Heuer	18209	DBR	31.10.1958
5588	Gunter	Saake	39106	MD	05.10.1960
6834	Michael	Korn	39104	MD	24.09.1974
8832	Tamara	Jagellovsk	38106	BS	11.11.1973
9999	Christa	Loeser	69121	HD	10.05.1969

Pers_Telefon

PANr	Telefon
4711	038203-12230
4711	0381-498-3401
5588	0391-345677
5588	0391-5592-3800
9999	06221-400177

Formalisierung: Fremdschlüsselbeziehung

- Fremdschlüssel:
  $X(R_1) \rightarrow Y(R_2)$

  $\{t(X) \mid t \in r_1\} \subseteq \{t(Y) \mid t \in r_2\}$,
  und
  Y ist Schlüssel von R_2
- Semantik von Integritätsbedingungen, die sich auf mehrere Relationenschemata beziehen („globale Integritätsbedingungen"), analog zum lokalen Fall: Abbildung eines Datenbankschemas auf {true, false}
  - Beachte: Globale Integritätsbedingungen müssen sich auf Datenbankschemata beziehen, nicht auf Relationenschemata
- Schlüssel und Fremdschlüssel sind die einzigen Integritätsbedingungen im relationalen Modell.

Zusammenfassung: Begriffe Relationenmodell I

Begriff	Informale Bedeutung
Attribut	Spalte einer Tabelle
Wertebereich	mögliche Werte eines Attributs (auch Domäne)
Attributwert	Element eines Wertebereichs
Relationenschema	Menge von Attributen
Relation	Menge von Zeilen einer Tabelle
Tupel	Zeile einer Tabelle
Datenbankschema	Menge von Relationenschemata
Datenbank	Menge von Relationen (Basisrelationen)
Basisrelation	die in Datenbank aktuell vorhandene Relation zu bestimmten Relationenschemata

Zusammenfassung: Begriffe Relationenmodell II

Einordnung
Datenmodelle
RM
Struktur
Formal
Integrität
Algebra

Begriff	Informale Bedeutung
Schlüssel	minimale Menge von Attributen, deren Werte ein Tupel einer Relation eindeutig identifizieren
Primärschlüssel	ein beim Datenbankentwurf ausgezeichneter Schlüssel
Fremdschlüssel	Attributmenge, die in einer anderen Relation Schlüssel ist
Fremdschlüssel-bedingung	Alle Attributwerte des Fremdschlüssels tauchen in der anderen Relation als Werte des Schlüssels auf
Nicht-Primattribute	Attribute, die nicht in einem Schlüssel auftauchen.

Agenda – Relationenmodell und -algebra

- Relationenmodell
  - Veranschaulichung
  - Formalisierung
  - Integritätsbedingungen
- Relationenalgebra
  - Einleitung
  - Projektion
  - Selektion
  - Join
  - Mengenoperationen
  - Vollständigkeit

Anfragesprachen

Funktionale Anforderungen an ein DBMS:

Funktionen für das
- Entgegennehmen,
- Abspeichern,
- Ändern,
- Löschen,
- Auswählen,
- Bereitstellen

von Daten.

} Anfragesprache

(Dazu kommen eine Reihe nicht-funktionaler Anforderungen, z. B. Performanz)

- Anfragesprache für relationale DBMS: SQL
  - Zunächst: Beschränkung auf Operatoren zum Abfragen der in Relationen enthaltenen Information.

Anfragealgebra

- Wir betrachten in diesem Kapitel aber anderen Mechanismus: Relationale Algebra
- Mathematik:
  Algebra definiert durch Wertebereich und auf diesem definierte Operatoren.
- Für Datenbankanfragen:
  Inhalte der Datenbank sind Werte, und Operatoren definieren Funktionen zum Berechnen von Anfrageergebnissen.
- Relationale Algebra bildet theoretische Grundlage von Anfragesprachen
  - Hintergrund: Anfrageoptimierung (Wahl einer möglichst effizienten Ausführungsreihenfolge der Operatoren)
  - SQL-Anfragen werden auf Ausdrücke der relationalen Algebra abgebildet
  - Assoziativität und Kommutativität ermöglichen Aussagen über Äquivalenz von Ausdrücken (Korrektheit des Ergebnisses!)

Relationenalgebra I

- Attribute ausblenden: Projektion, Zeichen „π“
  in [...]: welche Spalten behalten,
  in (...): auf welche Relation anwenden;

- Tupel heraussuchen: Selektion, Zeichen „σ“
  in [...]: unter welchen Bedingungen,
  in (...): auf welche Relation anwenden;

- Relationen verknüpfen: Verbund (Join), Zeichen “⋈“
  Tupel über gleichbenannten Attributen und Werten aneinanderhängen;

Relationenalgebra II

- Relationen vereinigen: Vereinigung, Zeichen „U“; Tupel aus beiden Relationen sammeln, Duplikate eliminieren
- Relationen voneinander abziehen: Differenz; Zeichen „–“ Tupel aus der ersten Relation herausnehmen, falls sie auch in der zweiten Relation vorkommen
- Attribute umbenennen: Zeichen „ß“ einen Attributnamen in einen anderen umbenennen (wichtig für $\bowtie$ und U, –)

Laufendes Beispiel

Ausleih

Invnr	Name
4711	Meyer
1201	Schulz
0007	Müller
4712	Meyer

Buch

Invnr	Titel	ISBN	Autor
0007	Dr. No	3-125	James Bond
1201	Objektbanken	3-111	Heuer
4711	Datenbanken	3-765	Vossen
4712	Datenbanken	3-891	Ullman
4717	Pascal	3-999	Wirth

Projektion I

- Beispiel 1: Projektion auf ein Attribut

$$\pi_{[Name]}(Ausleih)$$

Invnr	Name
4711	Meyer
1201	Schulz
0007	Müller
4712	Meyer

⇒

Name
Meyer
Schulz
Müller

- Doppelte Ergebnistupel werden eliminiert.
  → Mengensemantik!

Projektion II

- Syntax

$$\pi_{[attributmenge]}(relation)$$

bzw.

$$\pi_{attributmenge}(relation)$$

- Semantik

$$\pi_{X}(r) := \{t(X) \mid t \in r\}$$

für $r(R)$ und $X \subseteq R$ Attributmenge in R

Projektion III

Einordnung
Datenmodelle
RM
Struktur
Algebra
Einleitung
Projektion
Selektion
Join
Mengen-op.
Vollst.

- Beispiel 2: Projektion auf Attributmenge

$$\pi_{[\text{Invnr, ISBN}]}(\text{Buch})$$

Invnr	Titel	ISBN	Autor
0007	Dr. No	3-125	James Bond
1201	Objektbanken	3-111	Heuer
4711	Datenbanken	3-765	Vossen
4712	Datenbanken	3-891	Ullman
4717	Pascal	3-999	Wirth

Invnr	ISBN
0007	3-125
1201	3-111
4711	3-765
4712	3-891
4717	3-999

Projektion IV

- Einfache Optimierungsregel:
  Bei vielen Projektionen hintereinander reicht die zuletzt ausgeführte auch allein
- Beispiel

$$\pi_{[\text{Invnr}]}(\pi_{[\text{Invnr, ISBN}]}(\text{Buch}))$$

ergibt optimiert

$$\pi_{[\text{Invnr}]}(\text{Buch})$$

Laufendes Beispiel

Buch

Invnr	Titel	ISBN	Autor
0007	Dr. No	3-125	James Bond
1201	Objektbanken	3-111	Heuer
4711	Datenbanken	3-765	Vossen
4712	Datenbanken	3-891	Ullman
4717	Pascal	3-999	Wirth

$\pi_{[\text{Invnr}]}(\pi_{[\text{Invnr, ISBN}]}(\text{Buch}))$

Invnr	ISBN
0007	3-125

Invnr	ISBN
0007	3-125
1201	3-111

Invnr	ISBN
0007	3-125
1201	3-111
4711	3-765

...

Invnr	ISBN
0007	3-125
1201	3-111
4711	3-765
4712	3-891
4717	3-999

Projektion V

- Wieso sind derartige Optimierungen wichtig?
- Für Anfrageausführung:
  Alternativen sind zwar äquivalent,
  aber unterschiedlich teuer in der Ausführung.
- Optimierung nutzt Äquivalenz aus,
  um günstige Ausführung zu finden.

Projektion VI

- Ersetzung von

$$\pi_{[Invnr]}(\pi_{[Invnr,\ ISBN]}(Buch))$$

durch

$$\pi_{[Invnr]}(Buch)$$

ist immer vorteilhaft.

- Bei anderen Transformationen hängt Vorteilhaftigkeit vom Datenbankzustand ab, z. B. Join-Reihenfolge (s.u.)

Selektion I

Einordnung
Datenmodelle
RM
Struktur
Algebra
Einleitung
Projektion
Selektion
Join
Mengen-op.
Vollst.

- Beispiel

$$\sigma_{[\text{Name} < \text{'N'}]}(\text{Ausleih})$$

Invnr	Name
4711	Meyer
1201	Schulz
0007	Müller
4712	Meyer

⇒

Invnr	Name
4711	Meyer
0007	Müller
4712	Meyer

Selektion II

Einordnung
Datenmodelle
RM
Struktur
Algebra
Einleitung
Projektion
Selektion
Join
Mengen-op.
Vollst.

- Syntax

$$\sigma_{[bedingung]}(relation)$$

bzw.

$$\sigma_{bedingung}(relation)$$

- Semantik

$$\sigma_F(r) := \{t \mid t \in r \wedge F(t) = true\}$$

Selektion III

- 2 Typen von Selektionsbedingungen
  - F Konstanten-Selektion

    Attribut θ Konstante

    boolesches Prädikat θ ist = oder ≠, bei linear geordneten Wertebereichen auch < , ≤ , ≥ oder >
  - F Attribut-Selektion

    Attribut 1 θ Attribut 2
- F logische Verknüpfung mehrerer Konstanten- oder Attribut-Selektionen mit ∧, ∨ oder ¬ (boolesche Ausdrücke)

Selektion IV

- Einfache Optimierungsregeln:
  - Selektionen lassen sich in der Reihenfolge beliebig vertauschen,
    Beispiel:

    $$\sigma_{\text{Invnr} = 4711}(\sigma_{\text{Name} \leq \text{'N'}}(\text{Ausleih}))$$
    $$= \sigma_{\text{Name} \leq \text{'N'}}(\sigma_{\text{Invnr} = 4711}(\text{Ausleih}))$$

    - Diskussion/ Vorgriff: Warum sollten die Kosten dieser Alternativen sich unterscheiden?

  - Beispiel 2: $\sigma_{\text{Name} \leq \text{'N'} \wedge \text{Invnr} = 4711}(\text{Ausleih})) = ?$
    - Hintergrund: Warum ist die obige Formulierung wenig wünschenswert?

Selektion V

- Optimierungsregel (Forts.)
  - Beispiel 2: $\sigma_{Name \leq 'N' \wedge Invnr = 4711}(Ausleih)$
    $= \sigma_{Name \leq 'N'}(\sigma_{Invnr=4711}(Ausleih))$
  - D. h. logische Verknüpfungen überflüssig: Eine Relationenalgebra ist auch mit atomaren Konstanten- und Attributselektoren bereits vollständig.

Selektion VI

- Einfache Optimierungsregeln (Forts.):
  - Manchmal lassen sich Projektion und Selektion vertauschen;
    - Ist $\pi_{Invnr}(\sigma_{Name \leq 'N'}(Ausleih))$ $= \sigma_{Name \leq 'N'}(\pi_{Invnr}(Ausleih))$?
    - Voraussetzung für Vertauschbarkeit: Selektionsattribute kommen in Projektionsliste vor.
    - Beispiel: $\pi_{Name}(\sigma_{Name \leq 'N'}(Ausleih))$

Verbund

- Syntax des (natürlichen) Verbundes (englisch: natural join)

  Relation 1 $\bowtie$ Relation 2

  - Semantik

    $r_1 \bowtie r_2 :=$

    $\{t \mid t(R_1 \cup R_2) \wedge. [\forall i \in \{1, 2\}. \exists t_i \in r_i : t_i = t(R_i)]\}$

- Verbund verknüpft Tabellen über gleichbenannten Spalten bei gleichen Attributwerten.

Laufendes Beispiel

Ausleih

Invnr	Name
4711	Meyer
1201	Schulz
0007	Müller
4712	Meyer

Buch

Invnr	Titel	ISBN	Autor
0007	Dr. No	3-125	James Bond
1201	Objektbanken	3-111	Heuer
4711	Datenbanken	3-765	Vossen
4712	Datenbanken	3-891	Ullman
4717	Pascal	3-999	Wirth

Verbund: Beispiel

- Ausleih ⋈ Buch

  ergibt

Name	Invnr	Titel	ISBN	Autor
Müller	0007	Dr. No	3-125	James Bond
Schulz	1201	Objektbanken	3-111	Heuer
Meyer	4711	Datenbanken	3-765	Vossen
Meyer	4712	Datenbanken	3-891	Ullman

- Nicht ausgeliehenes Pascal-Buch verschwindet: Tupel, die keinen Partner finden (dangling tuples), werden eliminiert.

Verbund und kartesisches Produkt I

- Semantik Join

  $r_1 \bowtie r_2 :=$
  $\{t \mid t(R_1 \cup R_2) \wedge [\forall i \in \{1, 2\}. \exists t_i \in r_i : t_i = t(R_i)]\}$

- Was ist, wenn $R_1 \cap R_2$ leer?

Verbund und kartesisches Produkt II

- $\pi_{[Autor]}(Buch) \bowtie \pi_{[Invnr]}(Ausleih)$ entartet zu kartesischem Produkt.

Autor	Invnr
James Bond	4711
James Bond	1201
James Bond	0007
James Bond	4712
Heuer	4711
Heuer	1201
Heuer	0007
Heuer	4712
Vossen	4711
...	...

Kartesisches Produkt:

alle Paarungen, die möglich sind.

Eigenschaften Verbund

- Verbund kommutativ:
  $r_1 \bowtie r_2 = r_2 \bowtie r_1$
- Verbund assoziativ:
  $(r1 \bowtie r2) \bowtie r3 = r1 \bowtie (r2 \bowtie r3)$
- Daher erlaubt: $\bowtie_{i=1}^{p} r_i$
- Beispiel dafür, daß Join-Reihenfolge wichtig:

r_1

B
b

r_2

B	C
a	c1
…	…
a	c100000

r_3

B	D
a	d1
…	…
a	d100000

Theta-Join

- Bisher können wir nur auf Gleichheit verbinden („Equi-Join“) – Beispiel, bei dem dies nicht ausreicht?
- Allgemeiner Verbund-Operator?
- Häufige Kombination von kart. Produkt und Selektion motiviert Definition des abgeleiteten Operators Theta-Join:

$$R \bowtie_\theta S := \sigma_\theta (R \times S)$$

- Beachte (Def.): Hier findet keine Projektion statt
- Vorteil:
  - Einfachere Formulierung von Anfragen,
  - Implementierung kann umfangreiche Zwischenrelation durch unmittelbare Auswertung der Selektionsbedingung vermeiden.

Halbverbindung (Semi-Join)

- Effekt der natürlichen Verbindung $R \bowtie S$:
  - Kombination derjenigen Tupel, die „Partner“ in jeweils anderer Relation haben,
  - Elimination der „partnerlosen“ (engl.: dangling) Tupel.
- Manchmal interessiert nur Existenz, nicht aber Attributwerte, eines „Partners“ (spezielle Selektion!)
  - Beispiel: Welche Bücher hat ‚Müller‘ gerade ausgeliehen?
- Halbverbindung $R \ltimes S$ selektiert Tupel aus R, die an natürlicher Verbindung teilnehmen würden:

$$R \ltimes S := \pi_{\text{Attribute}(R)} (R \bowtie S) = R \bowtie \pi_{\text{Attribute}(R)} (S)$$

Mengenoperationen und Umbenennung

Buch1

Autor1
James Bond
Heuer
Vossen
Ullman
Wirth

Buch2

Autor2
Witt
Vossen
Silberschatz
Meier
Wirth

- Umbenennung
  $ß_{[neu \leftarrow alt]}$(relation) (bzw. $ß_{neu \leftarrow alt}$(relation))
  ändert Attributnamen von alt in neu
  - Beispiel: $ß_{[Autor1 \leftarrow Autor2]}$(Buch2)
  - Durch Umbenennung nun Vereinigung, Differenz und Durchschnitt möglich
    - Mengenoperationen erfordern, dass Relationenschemata der beteiligten Relationen gleich sind!

Mengenoperationen: Vereinigung

- Relation 1 U Relation2
  - Beispiel: Buch1 U $ß_{[\text{Autor1} \leftarrow \text{Autor2}]}$(Buch2)

Autor1
James Bond
Heuer
Vossen
Ullman
Wirth
Witt
Silberschatz
Meier

Mengenoperationen: Differenz

- Relation 1 – Relation2
  - Beispiel: Buch1 – $ß_{[\text{Autor1} \leftarrow \text{Autor2}]}$(Buch2)

Autor1
James Bond
Heuer
Ullman

Mengenoperationen: Durchschnitt

Einordnung
Datenmodelle
RM
Struktur
Algebra
Einleitung
Projektion
Selektion
Join
Mengen-op.
Vollst.

- Relation 1 ∩ Relation 2
  - Beispiel: Buch1 ∩ $ß_{[Autor1 \leftarrow Autor2]}$(Buch2)

Autor1
Vossen
Wirth

Mengenoperationen, Umbenennung I

- Umbenennung ermöglicht
  - Verbunde, wo bisher kartesische Produkte ausgeführt wurden (unterschiedliche Attribute werden gleich benannt),
  - kartesische Produkte, wo bisher Verbunde ausgeführt wurden (gleiche Attribute werden unterschiedlich genannt),
  - Mengenoperationen

Mengenoperationen, Umbenennung II

- Was ist der Natural Join dieser Relationen ohne Umbenennung?
  (Hinweis: Was ist Schema des Join-Ergebnisses?)

Buch1

Autor
James Bond
Heuer
Vossen
Ullman
Wirth

Buch2

Autor
Witt
Vossen
Silberschatz
Meier
Wirth

Mengenoperationen, Umbenennung III

- Beispiel:

Buch1

Autor
James Bond
Heuer
Vossen
Ullman
Wirth

Buch2

Autor
Witt
Vossen
Silberschatz
Meier
Wirth

- Wir wollen Paare bilden (James Bond, Witt), (James Bond, Vossen), ..., (Heuer, Witt), ...
- Vorgehen: Umbenennung und Natural Join.

Mengenoperationen, Umbenennung IV

Einordnung
Datenmodelle
RM
Struktur
Algebra
Einleitung
Projektion
Selektion
Join
Mengen-op.
Vollst.

- Erinnerung:
  Tupel – Abbildung $t : R \rightarrow \cup_{i=1}^{m} D_i$
  $R=\{A_1, ..., A_n\}$
- Formal für $r_1(R)$ und $r_2(R)$.
- Umbenennung
  $ß_{B \leftarrow A(r)} := \{t' \mid \exists\, t \in r: t'(R-\{A\})=t(R-\{A\}) \wedge t'(B)=t(A)\}$
- Vereinigung $r_1 \cup r_2 := \{t \mid t \in r_1 \vee t \in r_2\}$
- Differenz $r_1 - r_2 := \{t \mid t \in r_1 \wedge t \notin r_2\}$
- Durchschnitt $r_1 \cap r_2 := \{t \mid t \in r_1 \wedge t \in r_2\}$
  Durchschnitt $\cap$
  wegen $r_1 \cap r_2 = r_1 - (r_1 - r_2)$ überflüssig.

Unabhängigkeit und Vollständigkeit I

- Minimale Relationenalgebra:

  $\Omega = \pi, \sigma, \bowtie, ß, \cup$ und $-$

- Ω ist unabhängig:
  Kein Operator kann weggelassen werden, ohne Ausdrucksmächtigkeit zu verlieren.
- Andere unabhängige Menge:
  $\bowtie$ durch x ersetzen.
- Warum wichtig?
  - Redundanzfreiheit
    für formale Überlegungen vorteilhaft.
  - Minimalität bequemer, wenn es darum geht, Vollständigkeit nachzuweisen.

Unabhängigkeit und Vollständigkeit II

Einordnung
Datenmodelle
RM
Struktur
Algebra
Einleitung
Projektion
Selektion
Join
Mengen-op.
Vollst.

- Wir definieren $\Omega = \{\pi, \sigma, \bowtie, ß, \cup$ und $-\}$ als Operationenmenge der Relationenalgebra
  - Festlegung der Ausdrucksmächtigkeit
- Relationale Vollständigkeit: Anfragesprache ist relational vollständig, falls sie genauso mächtig ist wie Ω.
  - Wie könnte man dies formal zeigen?
- Strenge relationale Vollständigkeit:
  Zu jedem Term mit Operatoren aus Ω
  gibt es einen einzigen Ausdruck in der betrachteten Anfragesprache

Problem: Quantoren

- Existenzquantor ist in Selektionsbedingungen implizit enthalten
- Allquantor?
- Kann simuliert werden (Ausdruckmächtigkeit!)
- Darstellung am Beispiel der Division (Umkehroperation des kart. Produkts)
- Division: Seien $R_2 \subseteq R_1$, $R' = R_1 - R_2$

$$r_1 \div r_2 := \{t \mid \forall\, t_2 \in r_2\ \exists\, t_1 \in r_1 : t_1(R') = t \wedge t_1(R_2) = t_2\} = r'(R')$$

Division: Beispiel

r_1

Pilot	Flugzeug
Snoopy	707
Snoopy	727
Snoopy	747
Meyer	707
Meyer	727
Müller	707
Müller	727
Müller	747
Müller	777
Lüdenscheid	727

r_2

Flugzeug
707
727
747

r_3

Flugzeug
707

$r_1 \div r_2$

Pilot
Snoopy
Müller

$r_1 \div r_3$

Pilot
Snoopy
Meyer
Müller

Division aus Ω herleitbar

- $r_1(R_1)$ und $r_2(R_2)$ gegeben mit $R_2 \subseteq R_1$, $R' = R_1 - R_2$.
- R_1={Pilot, Flugzeug}, R_2={Flugzeug}
- Division von r_1 durch r_2

$$r_1 \div r_2 = \pi_{R'}(r_1) - \pi_{R'}((\pi_{R'}(r_1) \bowtie r_2) - r_1)$$

$$r_1 \div r_2 = \pi_{Pilot}(r_1) - \pi_{Pilot}((\pi_{Pilot}(r_1) \bowtie r_2) - r_1)$$

r_1

Pilot	Flugzeug
Snoopy	707
Snoopy	727
Snoopy	747
Meyer	707
Meyer	727
Müller	707
Müller	727
Müller	747
Müller	777
Lüdenscheid	727

r_2

Flugzeug
707
727
747

Zusammenfassung

- Algebra für den Zugriff auf relationale Datenbanken wurde vorgestellt (relationale Algebra)
- Standard-Formalismus – Referenzpunkt für andere Anfragemechanismen
- Ausdrucksmächtigkeit: keine Rekursion

σ	:	**relation** $\times\ \Theta \rightarrow$ **relation**	(Selektion)
π	:	**relation** $\times$ attributfolge $\rightarrow$ **relation**	(Projektion)
$\times$	:	**relation** $\times$ **relation** $\rightarrow$ **relation**	(Kartesisches Produkt)
$\cup$	:	**relation** $\times$ **relation** $\rightarrow$ **relation**	(Vereinigung)
/	:	**relation** $\times$ **relation** $\rightarrow$ **relation**	(Differenz)
$\cap$	:	**relation** $\times$ **relation** $\rightarrow$ **relation**	(Durchschnitt)
$\bowtie_\theta$	:	**relation** $\times\ \Theta\ \times$ **relation** $\rightarrow$ **relation**	(Theta-Verbindung)
$\bowtie$	:	**relation** $\times$ **relation** $\rightarrow$ **relation**	(natürliche Verbindung)
$\div$	:	**relation** $\times$ **relation** $\rightarrow$ **relation**	(Division)

Θ: Universum logischer Bedingungen; plus: Umbenennung ß