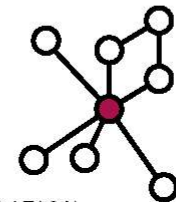


Teil 1: Deduktive Datenbanken



Gerd Stumme
Christoph Schmitz

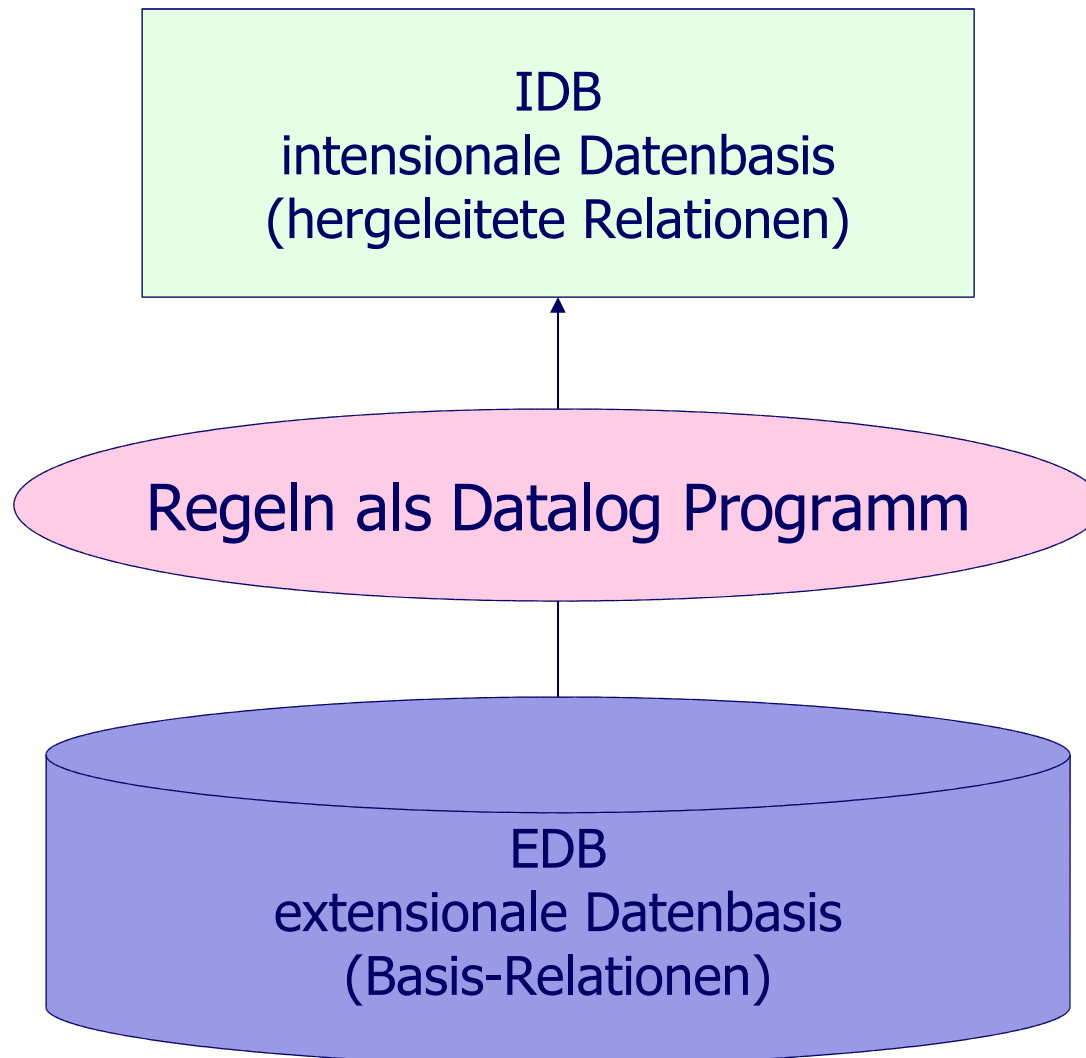
Wintersemester 2004/05



ENDOWED CHAIR OF THE HERTIE FOUNDATION
Knowledge and Data Engineering
DEPARTMENT OF MATHEMATICS & COMPUTER SCIENCE

Deduktive Datenbanken

Grundkonzepte einer deduktiven Datenbank



Terminologie

Die *extensionale Datenbasis (EDB)*, die manchmal auch Faktenbasis genannt wird. Die EDB besteht aus einer Menge von Relationen(Ausprägungen) und entspricht einer „ganz normalen“ relationalen Datenbasis.

Die *Deduktionskomponente*, die aus einer Menge von (Herleitungs-)Regeln besteht. Die Regelsprache heißt *Datalog* - abgeleitet von dem Wort *Data* und dem Namen der Logikprogrammiersprache *Prolog*.

Die *intensionale Datenbasis (IDB)*, die aus einer Menge von hergeleiteten Relationen(Ausprägungen) besteht. Die IDB wird durch Auswertung des Datalog-Programms aus der EDB generiert.

Datalog

Regel:

sokLV(T,S) :- vorlesungen(V,T,S,P), professoren(P, „Sokrates“,R,Z), >(S,2).

Äquivalenter Domänenkalkül-Ausdruck:

$$\{[t,s] \mid \exists v,p ([v,t,s,p] \in \text{Vorlesungen} \wedge \\ \exists n,r,z ([p,n,r,z] \in \text{Professoren} \wedge \\ n = \text{„Sokrates“} \wedge s > 2))\}$$

Grundbausteine der Regeln sind **atomare Formeln** **oder** **Literale**:

$$q(A_1, \dots, A_m).$$

q ist dabei der Name einer Basisrelation, einer abgeleiteten Relation oder eines eingebauten Prädikats: <,=,>,...

Beispiel: professoren(S, „Sokrates“,R,Z).

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

Vorlesungen			
VorlNr	Titel	SWS	gelesenV on
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

voraussetzen	
Vorgänger	Nachfolger
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

hören	
MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022

Assistenten			
PerslNr	Name	Fachgebiet	Boss
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetenbewegung	2127
3006	Newton	Keplersche Gesetze	2127
3007	Spinoza	Gott und Natur	2126

prüfen			
MatrNr	VorlNr	PersNr	Note
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2

Eine Datalog-Regel

$$p(X_1, \dots, X_m) :- q_1(A_{11}, \dots, A_{1m_1}), \dots, q_n(A_{n1}, \dots, A_{nm_n}).$$

Jedes $q_j(\dots)$ ist eine **atomare Formel**. Die q_j werden oft als *Subgoals* bezeichnet.

X_1, \dots, X_m sind **Variablen**, die mindestens einmal auch auf der rechten Seite des Zeichens $:-$ vorkommen müssen.

Logisch äquivalente Form obiger Regel:

$$p(\dots) \vee \neg q_1(\dots) \vee \dots \vee \neg q_n(\dots)$$

Wir halten uns an folgende Notation:

- **Prädikate** beginnen mit einem Kleinbuchstaben.
- Die zugehörigen **Relationen** - seien es **EDB-** oder **IDB-Relationen** - werden mit gleichem Namen, aber mit einem Großbuchstaben beginnend, bezeichnet.

Beispiel Datalog-Programm

Zur Bestimmung von (thematisch) verwandten Vorlesungspaaren

geschwisterVorl(N_1, N_2) :- voraussetzen(V, N_1),
voraussetzen(V, N_2), $N_1 < N_2$.

geschwisterThemen(T_1, T_2) :- geschwisterVorl(N_1, N_2),
vorlesungen(N_1, T_1, S_1, R_1),
Vorlesungen(N_2, T_2, S_2, R_2).

aufbauen(V, N) :- voraussetzen(V, N)

aufbauen(V, N) :- aufbauen(V, M), voraussetzen(M, N).

verwandt(N, M) :- aufbauen(N, M).

verwandt(N, M) :- aufbauen(M, N).

verwandt(N, M) :- aufbauen(V, N), aufbauen(V, M).

Voraussetzen: {[Vorgänger, Nachfolger]}

Vorlesungen: {VorlNr, Titel, SWS, gelesenVon]}

Analogie zur EDB/IDB in rel. DBMS

Basis-Relationen entsprechen der EDB.

Sichten entsprechen der IDB:

- „Aufbauen“ als Regeln in einem deduktiven DBMS:

aufbauen(V,N) :- voraussetzen(V,N)

aufbauen(V,N) :- aufbauen(V,M), voraussetzen(M,N).

- „Aufbauen“ als Sichtdefinition in DB2:

```
create view aufbauen(V,N) as
(select Vorgaenger, Nachfolger
 from voraussetzen
 union all
 select a.V, v.Nachfolger
 from aufbauen a, voraussetzen v
 where a.N = v.Vorgaenger)
```

```
select * from aufbauen
```


Der Wiener Kreis

Wissenschaftstheorie

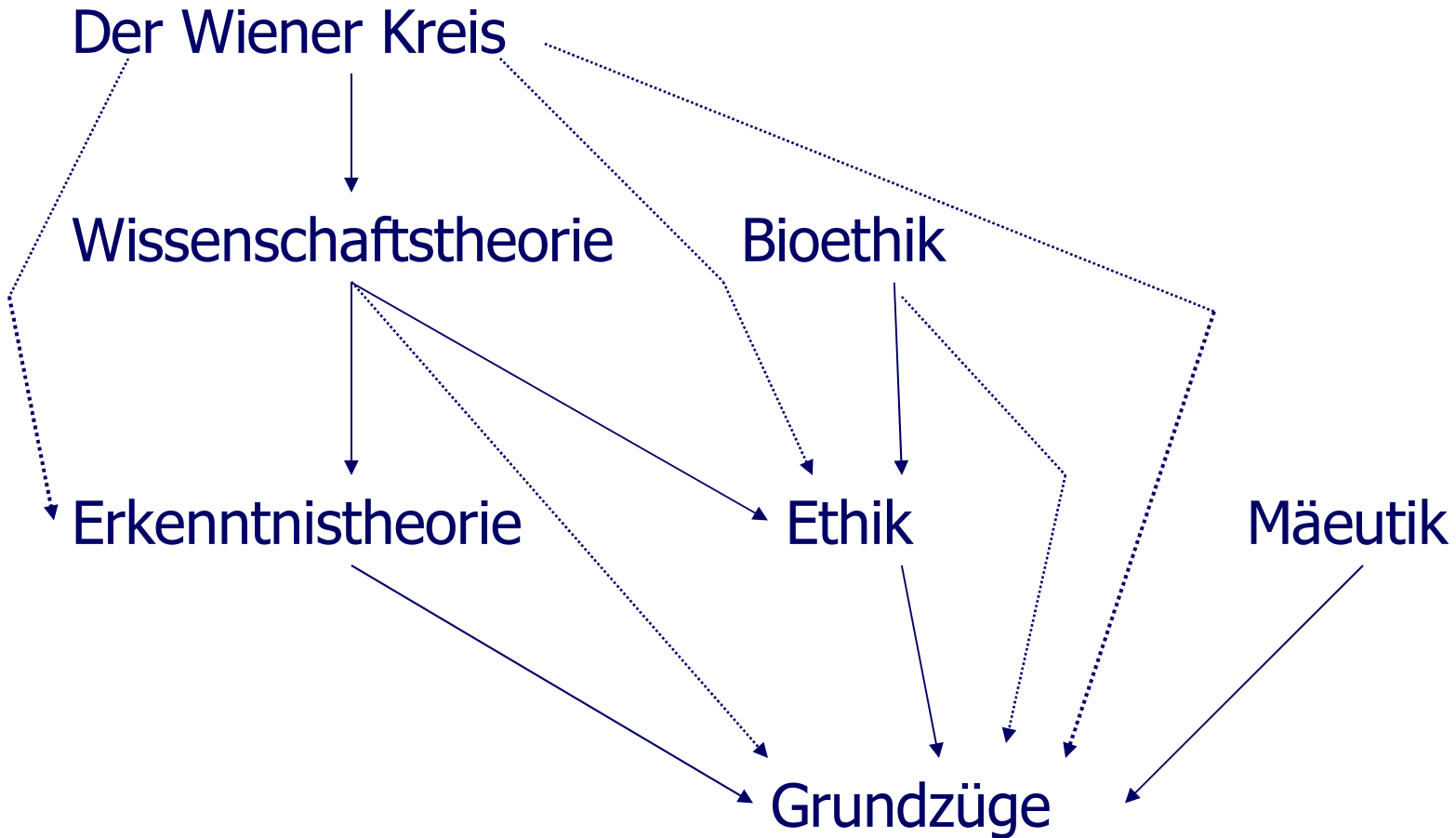
Bioethik

Erkenntnistheorie

Ethik

Mäeutik

Grundzüge



Rekursion in DB2/SQL99: gleiche Anfrage

with TransVorl (Vorg, Nachf)

as (**select** Vorgänger, Nachfolger **from** voraussetzen

union all

select t.Vorg, v.Nachfolger

from TransVorl t, voraussetzen v

where t.Nachf= v.Vorgänger)

select Titel **from** Vorlesungen **where** VorlNr **in**

(**select** Vorg **from** TransVorl **where** Nachf **in**

(**select** VorlNr **from** Vorlesungen

where Titel= `Der Wiener Kreis`))

Sicherheit von Datalog-Regeln

Es gibt unsichere Regeln, wie z.B.

$\text{ungleich}(X, Y) :- X \neq Y.$

Diese definieren unendliche Relationen.

Eine Datalog-Regel ist sicher, wenn alle Variablen im Kopf beschränkt (range restricted) sind. Dies ist für eine Variable X dann der Fall, wenn:

- die Variable im Rumpf der Regel in mindestens einem normalen Prädikat - also nicht nur in eingebauten Vergleichsprädikaten - vorkommt oder
- ein Prädikat der Form $X = c$ mit einer Konstante c im Rumpf der Regel existiert oder
- ein Prädikat der Form $X = Y$ im Rumpf vorkommt, und man schon nachgewiesen hat, dass Y eingeschränkt ist.

Ausdruckskraft von Datalog

Die Sprache Datalog, eingeschränkt auf nicht-rekursive Programme aber erweitert um Negation, wird in der Literatur manchmal als *Datalog*[¬]_{non-rec} bezeichnet

Diese Sprache *Datalog*[¬]_{non-rec} hat genau die gleiche Ausdruckskraft wie die relationale Algebra - und damit ist sie hinsichtlich Ausdruckskraft auch äquivalent zum relativen Tupel- und Domänenkalkül

Datalog mit Negation und Rekursion geht natürlich über die Ausdruckskraft der relationalen Algebra hinaus - man konnte in Datalog ja z.B. die transitive Hülle der Relation *Voraussetzen* definieren.

Datalog-Formulierung der relationalen Algebra-Operatoren

Selektion

$$\sigma_{SWS > 3}(\text{Vorlesungen}),$$

$query(V, T, S, R) : -\text{vorlesungen}(V, T, S, R), S > 3.$

$query(V, S, R) : -\text{vorlesungen}(V, \text{"Mäeutik"}, S, R).$

Projektion

$query(\text{Name}, \text{Rang}) : -\text{professoren}(\text{PersNr}, \text{Name}, \text{Rang}, \text{Raum}).$

Join

$$\Pi_{\text{Titel}, \text{Name}} (\text{Vorlesungen} \bowtie_{\text{gelesenVor}=\text{PersNr}} \text{Professoren})$$

$query(T, N) : -\text{vorlesungen}(V, T, S, R), \text{professoren}(R, N, Rg, Ra)$

Datalog-Formulierung der relationalen Algebra-Operatoren

Kreuzprodukt

$query(V1, V2, V3, V4, P1, P2, P3, P4) : -vorlesungen(V1, V2, V3, V4),$
 $professoren(P1, P2, P3, P4).$

Professoren \times Vorlesungen

Vereinigung

$\Pi_{PersNr, Name}(Assistenten) \cup \Pi_{PersNr, Name}(Professoren)$

$query(PersNr, Name) : -assistenten(PersNr, Name, F, B).$

$query(PersNr, Name) : -professoren(PersNr, Name, Rg, Ra).$

Datalog-Formulierung der relationalen Algebra-Operatoren

Mengendifferenz

$$\Pi_{VorlNr}(Vorlesungen) - \Pi_{Vorgänger}(Voraussetzen)$$

$vorlNr(V) : \neg vorlesungen(V, T, S, R).$

$grundlagen(V) : \neg voraussetzen(V, N).$

$query(V) : \neg vorlNr(V), \neg grundlagen(V).$