Seminar: Imperfektion in Datenbanken WS03/04

Null-Werte in Relationalen Datenbanken

Thomas Bierhance

Einführung

- Null-Werte in DBen sind notwendiges Übel, da...
  - (1) das Wissen über die tatsächliche Welt imperfekt sein kann $\Rightarrow$ Wert existent aber unbekannt (z.B. Alter einer Person)
  - (2) die Datenstruktur die tatsächliche Welt nur imperfekt abbildet $\Rightarrow$ Wert nicht anwendbar (z.B. Datum der Führerscheinprüfung)
- Gegenstand der Forschung war und ist, wie mit diesen besonderen Werten in DBen umgegangen werden muss.

Beispiel zur Motivierung

- Zwei Relationen mit Null-Werten.
- Wie werden relationale Ausdrücke auf diesen Relationen ausgewertet?

R	A	B
	1	b
	2	Null
	Null	b'

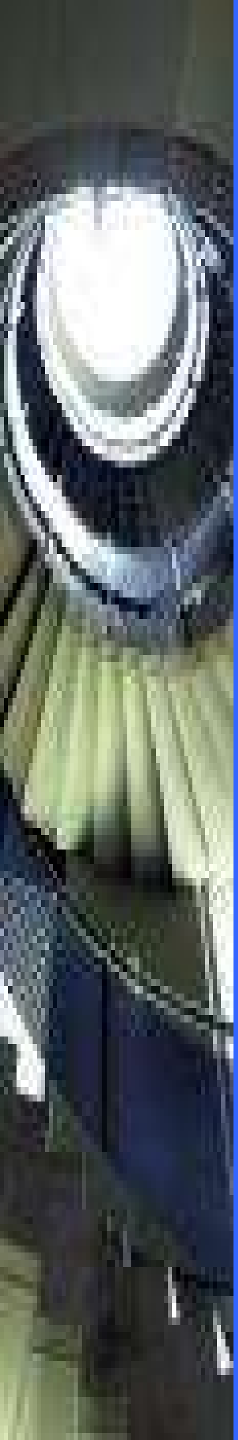
S	A	C
	2	c
	1	Null
	Null	Null

$$\sigma_{A>1}(R)$$

$$\sigma_{A>1 \wedge B=b'}(R)$$

$$\pi_{C}(S)$$

$$R \bowtie S$$



Erweiterung von Codd, 1979

- “Null” gleichbedeutend mit “Wert zur Zeit unbekannt”
- “Null” wird mit “ω“ bezeichnet
- Vorschlag zur Erweiterung der relationalen Operatoren um mit Null-Werten umgehen zu können.
- Der SQL-Standard hat im wesentlichen die Vorschläge von Codd übernommen.

Dreiwertige Logik

Kern von Codds Erweiterung ist ein dreiwertige Logik, die die Werte “Wahr”, “Falsch” und “unbekannt” beinhaltet.

$\wedge$	F	ω	T
F	F	F	F
ω	F	ω	ω
T	F	ω	T

$\vee$	F	ω	T
F	F	ω	T
ω	ω	ω	T
T	T	T	T

$\neg\omega = \omega \quad \neg T = F \quad \neg F = T$

Prinzip der Null-Substitution

- Ein Ausdruck *A* hat den Wahrheitswert ω genau dann, wenn...

  (1) jedes ω in *A* mit einem Wert ($\neq\omega$) ersetzt werden kann, so dass *A*=T

  (2) jedes ω in *A* mit einem Wert ($\neq\omega$) ersetzt werden kann, so dass *A*=F
- Hieraus folgen dann entsprechende Substitutionsregeln für *A*=*T* und A=F

Beispiel für die Null-Substitution

- $A>1$ für $\{\omega, b'\}$
  - $\omega>1=\omega$, da $1>1=F$ und $2>1=T$

- $A>1 \wedge B=b'$ für $\{2, \omega\}$
  - $(2>1 \wedge \omega=b') = \omega$, da
    - $(b=b')=F$ und $(b'=b')=T \Rightarrow (\omega=b')=\omega$
    - $T \wedge \omega = \omega$

R	A	B
	1	b
	2	ω
	ω	b'

Duplikatsentfernung und Konsequenzen

- Relationen dürfen ein Tupel nicht mehrfach enthalten.
- Um diese Regel einzuhalten, wird bei der Duplikatsentfernung vom Prinzip der Null-Substitution abgewichen.
- Abweichung sei zulässig, da Duplikatsentfernung und allgemeine Ausdrucksbewertung auf unterschiedlichen Ebenen befindlich.

Anpassung der Operatoren

- Operatoren deren Basis die Auswertung eines Wahrheitsausdrucks ist müssen angepasst werden. (Selektion, Theta-Verbund)

- Alle anderen Operatoren behandeln “ω” gemäß der Regel zur Duplikatsentfernung wie einen gewöhnlichen Domainenwert und müssen nicht weiter angepasst werden. (Vereinigung, Schnittmenge, Differenz, kartesisches Produkt und Projektion)

Anpassung von Theta-Verbund / Selektion

- Problem bei Theta-Verbund und Selektion:

  Das Selektionskriterium kann für einige Tupel unbestimmt sein (=ω). Was geschieht mit diesen Tupeln, für die das Kriterium "vielleicht" erfüllt ist?
- Lösung:
  - True-(Theta-Verbund/Selektion): Alle Tupel die sicher im Ergebnis enthalten sind.
  - Maybe-(Theta-Verbund/Selektion): Alle Tupel die vielleicht enthalten wären.

Tautologien in 2-/3-wertiger Logik

- Nach den Regeln der relationalen Algebra mit zweiwertiger Logik ist folgender Ausdruck eine Tautologie...

$$R_{[A \geq 50]} \cup R_{[A < 50]} = R$$

- Der gleiche Ausdruck mit Codds dreiwertiger Logik erhält diese Tautologie nicht...

$$R_{[A \geq 50]} \cup R_{[A < 50]} = R - R_{[A = \omega]}$$

Kritik an Codds Vorschlag

- Es werden keine Kriterien angegeben, die eine sinnvolle Erweiterung erfüllen sollte.
- Was ist die gemeinsame Basis für Duplikats-entfernung und Ausdrucksbewertung? Beispiel:
  - Der Schnittmengenoperator behandelt ω wie einen Domänenwert: $(\omega==\omega)=T$
    Die Schnittmenge ist aber ein Spezialfall vom Verbund, wo wiederum die 3-wertige Logik benutzt wird: $(\omega==\omega)=\omega$

Kritik an Codds Vorschlag, Grant 1977

- Sei S eine Relation auf {S#,Name,Status,City}
- Sei ein Tupel s = {25, Jones, ω, Paris}
- Ein relationaler Ausdruck

$$\sigma_{\text{Name=Jones} \wedge \text{Status=10}}(S) \cup \sigma_{\text{Status<>10} \wedge \text{Ort=Paris}}(S)$$

- Obwohl das Tupel s intuitiverweise im Ergebnis enthalten sein sollte, ist es nach Codds Vorschlag nicht enthalten.

Erweiterung von Imielinski/Lipski, 1984

- Definition einer allgemeinen Eigenschaft, die eine “sinnvolle” Erweiterung der relationalen Algebra erfüllen sollte.
  - Idee: Existiert eine Abbildung zwischen der Erweiterung und der klassischen relationalen Algebra, die bestimmte Eigenschaften erfüllt, ist die Erweiterung “korrekt”.
  - Diese Abbildung ist für die Korrektheit relevant, nicht für die Evaluierung von Ausdrücken.
- Erweiterung, die statt einfacher Null-Werte Variablen verwendet.

Repräsentations-System

- Definitionen
  - *T* – homogene Menge von Multitabellen (mögliche Datenbankzustände “mit Null-Werten” für ein Schema)
  - *R* – homogene Menge von Multirelationen (dito “ohne Null-Werte”)
  - P – Menge aller *R*
  - eine Abbildung *Rep*: $T \rightarrow \mathrm{P}$
  - Ω - eine Menge von relationalen Operatoren
  - (T, *Rep*, Ω) - ein Repräsentationssystem
  - *f* – ein relationaler Ausdruck bestehend aus Operatoren aus Ω (eine Abfrage)

Ideales Repräsentations Systeme

- Idealerweise sollte ein Repräsentations-System folgende Bedingung erfüllen:

  $$Rep(f(t))=f(Rep(t)) \quad \forall\ t \in T,\ f$$

- Die Repräsentation des Ergebnisses der Anfrage auf die Datenbank mit Null-Werten entspricht dem Ergebniss der Anfrage auf die Repräsentation der Datenbank.

- Diese Bedingung ist normalerweise nicht zu erfüllen.

Relaxiertes Repräsentations-System

- Da das ideale Repräsentations-System kaum zu erreichen ist, wird die Bedingung relaxiert.

- Anstatt der strikten Gleichheit wird die sogenannte Ω-Äquivalenz gefordert:

$$Rep(f(t)) \equiv_{\Omega} f(Rep(t)) \quad \forall\, t \in T,\, f$$

$$X \equiv_{\Omega} Y$$

$$\Leftrightarrow \ \cap g(X) = \cap g(Y) \quad \forall\, g$$

Anwendung auf Codds-Erweiterung

- Die Erweiterung von Codd wurde hinsichtlich ihrer Repräsentierbarkeit untersucht:

- “Korrekte” Auswertung von Anfragen der Form:
  - Projektion
  - Projektion-Selektion
- Keine “korrekte” Auswertung für Anfragen der Form
  - Projektion,Selektion,Vereinigung
  - Projektion,Verbund

Erweiterung: Virtuelle Tabellen

- Statt einfacher “Null-Werte” werden Variablen benutzt, die so auch die Gleichheit zwischen zwei unbekannten Werten darstellen können.
- Relationale Operatoren behandeln Variablen wie normalen Domänen-Werte.
- “Korrekte” Antworten für Anfragen der Form
  - Projektion, Selektion+, Vereinigung, Verbund
- Keine “korrekte” Antworten für Anfragen der Form
  - Projektion, Selektion

Fazit

- Es ist fraglich ob eine ideale Lösung für die Behandlung von Null-Werten existiert.
- Durch den vorgestellten Ansatz ist aber eine Bewertung von Erweiterungen möglich.
- Da SQL dem Vorschlag von Codd entspricht, wird eine Vielzahl von Anfragen heute nicht “korrekt” im Sinne dieser Bewertung beantwortet.

Literatur

- Codd, E.F. “Extending the database relational model to capture more meaning”, ACM Transactions on Database Systems, Vol 4, No 4, Dec 1979, pp. 379-434
- Imielinski, Lipski “Incomplete Information in Relational Databases”, ACM Transactions on Database Systems, Vol 31, No 4, Oct 1984, pp. 761-791
- Biskup, J. “A Foundation of Codd's Relational Maybe-Operations”, ACM Transactions on Database Systems, Vol 8, No 4, Dec 1983, pp. 608-636
- Grant, J. “Null values in a relational data base” Inform. Process. Lett. 5, 1977, pp. 156-157