



Skript zur Vorlesung
Datenbanksysteme I
Wintersemester 2008/2009

Kapitel 4: Relationen-Kalkül

Vorlesung: Prof. Dr. Christian Böhm
Übungen: Annahita Oswald, Bianca Wackersreuther

Skript © 2005 Christian Böhm

<http://www.dbs.informatik.uni-muenchen.de/Lehre/DBS>



Begriff

Kalkül *das, auch der; -s, -e* <unter Einfluss von gleichbed. fr. calcul aus lat. calculus «Steinchen, Rechen-, Spielstein; Berechnung», Verkleinerungsform von lat. calx «(Spiel)stein; Kalk»>: etwas im Voraus abschätzende, einschätzende Berechnung, Überlegung.

Quelle: DUDEN - Das große Fremdwörterbuch

...**das Kalkül**

der Kalkül ...

Kalkül *der; -s, -e* <zu ¹Kalkül>: durch ein System von Regeln festgelegte Methode, mit deren Hilfe bestimmte mathematische Probleme systematisch behandelt u. automatisch gelöst werden können (Math.).

Quelle: DUDEN - Das große Fremdwörterbuch



Begriff

- Mathematik: Prädikatenkalkül
 - Formeln wie $\{x \mid x \in \mathbb{N} \wedge x^3 > 0 \wedge x^3 < 1000\}$
- Anwendung solcher Formeln für DB-Anfragen
 - Bezugnahme auf DB-Relationen im Bedingungsteil:
 $(x_1, y_1, z_1) \in \text{Mitarbeiter}, t_1 \in \text{Abteilungen}$
 - Terme werden gebildet aus Variablen, Konstanten usw.
 - Atomare Formeln aus Prädikaten der Datentypen:
 $=, <, >, \leq$, usw.
 - Atomare Formeln können mit logischen Operatoren zu komplexen Formeln zusammengefasst werden:
 $F_1 \wedge F_2, F_1 \vee F_2, \neg F_1, \exists x: F_1, \forall x: F_1$
- Bsp: Finde alle Großstädte in Bayern:
 $\{t \mid \text{Städte}(t) \wedge t[\text{Land}] = \text{Bayern} \wedge t[\text{SEinw}] \geq 500.000\}$
Hinweis: Städte(t) gleichbedeutend mit $t \in \text{Städte}$



Unterschied zur Rel. Algebra

- Relationale Algebra ist **prozedurale** Sprache:
 - Ausdruck gibt an, unter Benutzung welcher Operationen das Ergebnis berechnet werden soll
 - WIE
- Relationen-Kalkül ist **deklarative** Sprache:
 - Ausdruck beschreibt, welche Eigenschaften die Tupel der Ergebnisrelation haben müssen ohne eine Berechnungsprozedur dafür anzugeben
 - WAS
- Es gibt zwei verschiedene Ansätze:
 - **Tupelkalkül**: Variablen sind vom Typ *Tupel*
 - **Bereichskalkül**: Variablen haben *einfachen* Typ



Der Tupelkalkül

- Man arbeitet mit
 - Tupelvariablen: t
 - Formeln: $\psi(t)$
 - Ausdrücken: $\{t \mid \psi(t)\}$
- Idee: Ein Ausdruck beschreibt die Menge aller Tupel, die die Formel ψ **erfüllen** (wahr machen)
- Ein Kalkül besteht immer aus
 - Syntax: Wie sind Ausdrücke aufgebaut?
 - Semantik: Was bedeuten die Ausdrücke?



Tupelvariablen

- Tupelvariablen haben ein definiertes Schema:
 - $\text{Schema}(t) = (A_1: D_1, A_2: D_2, \dots)$
 - $\text{Schema}(t) = R_1$ (t hat dasselbe Schema wie Relation)
- Für Zugriff auf die Komponenten
 - $t[A]$ oder $t.A$ für einen Attributnamen $A \in \text{Schema}(t)$
 - oder auch $t[1]$, $t[2]$ usw.
- Tupelvariable kann in einer Formel ψ **frei** oder **gebunden** auftreten (s. unten)



Atome

- Es gibt drei Arten von Atomen:
 - $R(t)$ R ist Relationenname, t Tupelvariable
lies: t ist ein Tupel von R
 - $t.A \Theta s.B$ t bzw. s sind zwei Tupelvariablen mit passenden Attributen
lies: $t.A$ steht in Beziehung Θ zu ...
 - $t.A \Theta c$ t ist Tupelvariable und c eine passende Konstante

Θ Vergleichsoperator: $\Theta \in \{ =, <, \leq, >, \geq, \neq \}$



Formeln

Der Aufbau von Formeln ψ ist rekursiv definiert:

- **Atome:** Jedes Atom ist eine Formel
Alle vorkommenden Variablen sind **frei**
- **Verknüpfungen:** Sind ψ_1 und ψ_2 Formeln, dann auch:
 - $\neg \psi_1$ **nicht**
 - $(\psi_1 \wedge \psi_2)$ **und**
 - $(\psi_1 \vee \psi_2)$ **oder**Alle Variablen behalten ihren Status.
- **Quantoren:** Ist ψ eine Formel, in der t als **freie Variable** auftritt, sind auch Formeln...
 - $(\exists t)(\psi)$ **es gibt ein t , für das ψ**
 - $(\forall t)(\psi)$ **für alle t gilt ψ**die Variable t wird **gebunden**.



Formeln

- Gebräuchliche vereinfachende Schreibweisen:
 - $\psi_1 \Rightarrow \psi_2$ für $(\neg \psi_1) \vee \psi_2$ (**Implikation**)
 - $\exists t_1, \dots, t_k: \psi(t_1, \dots, t_k)$ für $(\exists t_1) (\dots ((\exists t_k) (\psi(t_1, \dots, t_k))) \dots)$
 - $(\exists t \in R) (\psi(t))$ für $(\exists t) (R(t) \wedge \psi(t))$
 - $(\forall t \in R) (\psi(t))$ für $(\forall t) (R(t) \Rightarrow \psi(t))$
 - Bei Eindeutigkeit können Klammern weggelassen werden
- Beispiel:
 - $(\forall s) (s.A \leq u.B \vee (\exists u)(R(u) \wedge u.C > t.D))$
 - t ist **frei**
 - s ist **gebunden**
 - u ist **frei beim ersten Auftreten und dann gebunden**



Ausdruck (Anfrage)

- Ein Ausdruck des Tupelkalküls hat die Form
$$\{t \mid \psi(t)\}$$
- In Formel ψ ist t die einzige freie Variable



Semantik

Bedeutung, die einem korrekt gebildeten Ausdruck durch eine Interpretation zugeordnet wird:

Syntax $\xrightarrow{\text{Interpretation}}$ Semantik

Tupelvariablen \longrightarrow konkrete Tupel

Formeln \longrightarrow true, false

Ausdrücke \longrightarrow Relationen



Belegung von Variablen

- Gegeben:
 - eine Tupelvariable t mit $\text{Schema}(t) = (D_1, D_2, \dots)$
 - eine Formel $\psi(t)$, in der t frei vorkommt
 - ein beliebiges konkretes Tupel r (d.h. mit Werten).
Es muß nicht zu einer Relation der Datenbank gehören
- Bei der Belegung wird jedes freie Vorkommen von t durch r ersetzt. Insbesondere wird $t.A$ durch den Attributwert von $r.A$ ersetzt.
- Man schreibt: $\psi(r \mid t)$



Beispiel

Gegeben sei folgendes Relationenschema:

Städte (SName: String, SEinw: Integer, Land: String)
Länder (LName: String, LEinw: Integer, Partei*: String)

* bei Koalitionsregierungen: jeweils eigenes Tupel pro Partei

- $\psi(t) = (t.\text{Land} = \text{Bayern} \wedge t.\text{SEinw} \geq 500.000)$
mit $\text{Schema}(t) = \text{Schema}(\text{Städte})$
 - $r_1 = (\text{Passau}, 49800, \text{Bayern})$:
 $\psi(r_1 \mid t) = (\text{Bayern} = \text{Bayern} \wedge 49800 \geq 500.000)$
 - $r_2 = (\text{Bremen}, 535.058, \text{Bayern})$:
 $\psi(r_2 \mid t) = (\text{Bremen} = \text{Bayern} \wedge 535.058 \geq 500.000)$



Interpretation von Formeln

Interpretation $I(\psi)$ analog zu syntaktischem Aufbau

- Anm: Alle Variablen sind durch konkrete Tupel belegt
- Atome:
 - $R(r)$: $I(R(r)) = \mathbf{true} \Leftrightarrow r$ ist in R enthalten
 - $c_i \Theta c_j$: $I(c_i \Theta c_j) = \mathbf{true} \Leftrightarrow$ der Vergleich ist erfüllt
- Logische Operatoren:
 - $\neg\psi$: $I(\neg\psi) = \mathbf{true} \Leftrightarrow I(\psi) = \mathbf{false}$
 - $\psi_1 \wedge \psi_2$: $I(\psi_1 \wedge \psi_2) = \mathbf{true} \Leftrightarrow I(\psi_1) = \mathbf{true}$ und $I(\psi_2) = \mathbf{true}$
 - $\psi_1 \vee \psi_2$: $I(\psi_1 \vee \psi_2) = \mathbf{true} \Leftrightarrow I(\psi_1) = \mathbf{true}$ oder $I(\psi_2) = \mathbf{true}$



Beispiele

- Atome:
 - $I(\text{Städte}(\text{Passau}, 49.800, \text{Bayern})) = \text{true}$
 - $I(49.800 \geq 500.000) = \text{false}$
- Logische Operatoren:
 - $I(\neg 49.800 \geq 500.000) = \text{true}$
 - $I(\text{Städte}(\text{Passau}, 49.800, \text{Bayern}) \vee 49.800 \geq 500.000) = \text{true}$
 - $I(\text{Städte}(\text{Passau}, 49.800, \text{Bayern}) \wedge 49.800 \geq 500.000) = \text{false}$



Interpretation von Quantoren

- Interpretation $I((\exists s)(\psi))$ bzw. $I((\forall s)(\psi))$:
 - In ψ darf **nur** s als freie Variable auftreten.
 - $I((\exists s)(\psi)) = \mathbf{true} \Leftrightarrow$ ein Tupel $r \in D_1 \times D_2 \times \dots$ existiert, daß bei Belegung der Variablen s die Formel ψ gilt:
$$I(\psi(r \mid s)) = \mathbf{true}$$
 - $I((\forall s)(\psi)) = \mathbf{true} \Leftrightarrow$ für alle Tupel $r \in D_1 \times D_2 \times \dots$ gilt die Formel ψ .
- Beispiele:
 - $I((\exists s)(\text{Städte}(s) \wedge s.\text{Land} = \text{Bayern})) = \mathbf{true}$
 - $I((\forall s)(s.\text{Name} = \text{Passau})) = \mathbf{false}$



Interpretation von Ausdrücken

- Interpretation von Ausdruck $I(\{t|\psi(t)\})$ stützt sich
 - auf Belegung von Variablen
 - und Interpretation von Formeln
- Gegeben:
 - $E = \{t \mid \psi(t)\}$
 - t die einzige freie Variable in $\psi(t)$
 - $\text{Schema}(t) = D_1 \times D_2 \times \dots$
- Dann ist der Wert von E die Menge aller* (denkbaren) Tupel $r \in D_1 \times D_2 \times \dots$ für die gilt:

$$I(\psi(r \mid t)) = \mathbf{true}$$

*Grundmenge sind hier **nicht** nur die gespeicherten Tupel aus der DB



Beispiel-Anfragen

Gegeben sei folgendes Relationenschema:

Städte (SName: String, SEinw: Integer, Land: String)
Länder (LName: String, LEinw: Integer, Partei*: String)

* bei Koalitionsregierungen: jeweils eigenes Tupel pro Partei

- Finde alle Großstädte (SName, SEinw, Land) in Bayern:

$\text{Schema}(t) = \text{Schema}(\text{Städte})$

$\{t \mid \text{Städte}(t) \wedge t.\text{Land} = \text{Bayern} \wedge t.\text{SEinw} \geq 500.000\}$

- In welchem Land liegt Passau?

$\text{Schema}(t) = (\text{Land}:\text{String})$

$\{t \mid (\exists u \in \text{Städte})(u.\text{Sname} = \text{Passau} \wedge u.\text{Land} = t.\text{Land})\}$

- Finde alle Städte in CDU-regierten Ländern:

$\text{Schema}(t) = \text{Schema}(\text{Städte})$

$\{t \mid \text{Städte}(t) \wedge (\exists u \in \text{Länder})(u.\text{Lname} = t.\text{Land} \wedge u.\text{Partei} = \text{CDU})\}$



Beispiel-Anfragen

Gegeben sei folgendes Relationenschema:

Städte (SName: String, SEinw: Integer, Land: String)
Länder (LName: String, LEinw: Integer, Partei*: String)

* bei Koalitionsregierungen: jeweils eigenes Tupel pro Partei

- Welche Länder werden von der SPD allein regiert?

$\text{Schema}(t) = \text{Schema}(\text{Länder})$

$\{t \mid \text{Länder}(t) \wedge (\forall u \in \text{Länder})(u.\text{LName} = t.\text{LName} \Rightarrow u.\text{Partei} = \text{SPD})\}$

- Gleichbedeutend mit:

$\text{Schema}(t) = \text{Schema}(\text{Länder})$

$\{t \mid \text{Länder}(t) \wedge (\forall u \in \text{Länder}) \neg (u.\text{LName} = t.\text{LName} \wedge u.\text{Partei} \neq \text{SPD})\}$



Sichere Ausdrücke

- Mit den bisherigen Definitionen ist es möglich, unendliche Relationen zu beschreiben:
 - $\text{Schema}(t) = \{\text{String}, \text{String}\}$
 - $\{t \mid t.1 = t.2\}$
 - Ergebnis: $\{(A,A), (B,B), \dots, (AA,AA), (AB,AB), \dots\}$
- Probleme:
 - Ergebnis kann nicht gespeichert werden
 - Ergebnis kann nicht in endlicher Zeit berechnet werden
- Definition:

Ein Ausdruck heißt *sicher*, wenn jede Tupelvariable nur Werte einer gespeicherten Relation annehmen kann, also positiv in einem Atom $R(t)$ vorkommt.



Der Bereichskalkül

- Tupelkalkül: Tupelvariablen t (ganze Tupel)
- Bereichskalkül: Bereichsvariablen $x_1:D_1, x_2:D_2, \dots$
 für einzelne Attribute
 (Bereich=Wertebereich=Domäne)

Ein **Ausdruck** hat die Form:

$$\{x_1, x_2, \dots \mid \psi(x_1, x_2, \dots)\}$$

Atome haben die Form:

- $R_1(x_1, x_2, \dots)$: Tupel (x_1, x_2, \dots) tritt in Relation R_1 auf
- $x \Theta y$: x, y Bereichsvariablen bzw. Konstanten
 $\Theta \in \{=, <, \leq, >, \geq, \neq\}$

Formeln analog zum Tupelkalkül



Beispiel-Anfragen

Städte (SName: String, SEinw: Integer, Land: String)

Länder (LName: String, LEinw: Integer, Partei*: String)

*bei Koalitionsregierungen: jeweils eigenes Tupel pro Partei

- In welchem Land liegt Passau?

$\{x_3 \mid \exists x_1, x_2: (\text{Städte}(x_1, x_2, x_3) \wedge x_1 = \text{Passau}) \}$

oder auch

$\{x_3 \mid \exists x_2: (\text{Städte}(\text{Passau}, x_2, x_3)) \}$

- Finde alle Städte in CDU-regierten Ländern:

$\{x_1 \mid \exists x_2, x_3, y_2: (\text{Städte}(x_1, x_2, x_3) \wedge \text{Länder}(x_1, y_2, \text{CDU})) \}$

- Welche Länder werden von der SPD allein regiert?

$\{x_1 \mid \exists x_2: (\text{Länder}(x_1, x_2, \text{SPD}) \wedge \neg \exists y_3: (\text{Länder}(x_1, x_2, y_3) \wedge y_3 \neq \text{SPD})) \}$

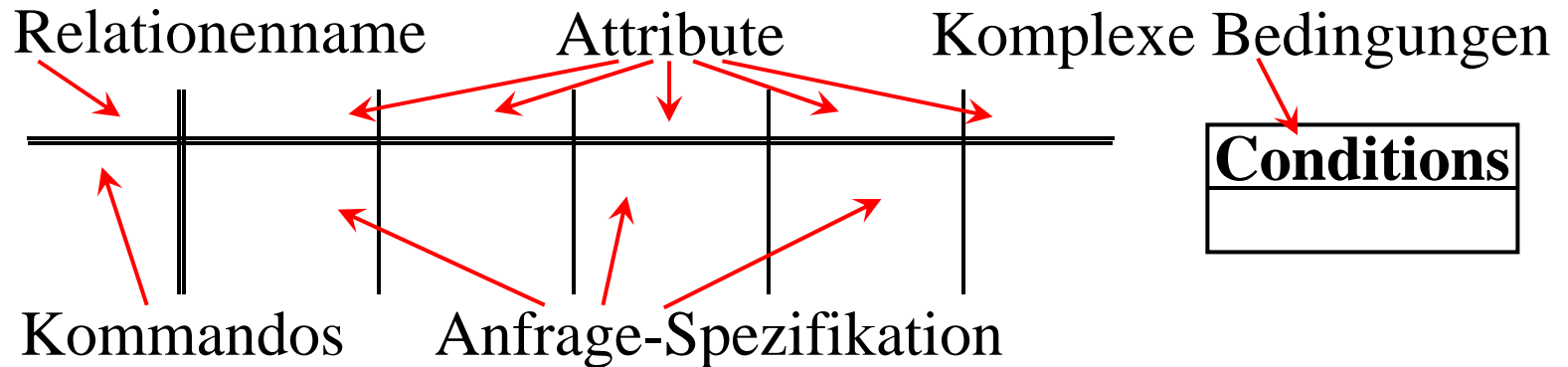


Query By Example (QBE)

- Beruht auf dem Bereichskalkül
- Ausdrücke nicht wie in SQL als Text
- Dem Benutzer wird am Bildschirm ein Tabellen-Gerüst angeboten, das mit Spezial-Editor bearbeitet werden kann
- Nach Eintrag von Werten in das Tabellengerüst (Anfrage) füllt das System die Tabelle
- Zielgruppe: Gelegentliche Benutzer



Query By Example (QBE)



Sprachelemente:

- Kommandos, z.B. **P.** (print), **I.** (insert), **D.** (delete) ...
- Bereichsvariablen (beginnen mit ‘_’): **_x**, **_y**
- Konstanten (Huber, Milch)
- Vergleichsoperatoren und arithmetische Operatoren
- Condition-Box: Zusätzlicher Kasten zum Eintragen einer Liste von Bedingungen (**AND**, **OR**, kein **NOT**)



Beispiel-Dialog

- Beginn: leeres Tabellengerüst

|--|--|--|--|

- *Benutzer* gibt interessierende Relation und **P.** ein
Kunde P.

|--|--|--|--|

- *System* trägt Attributnamen der Relation ein
Kunde KName KAdr Kto

Kunde	KName	KAdr	Kto
-------	-------	------	-----

- *Benutzer* stellt Anfrage

Kunde	KName	KAdr	Kto
	P.	P.	<0

- *System* füllt Tabelle mit Ergebnis-Werten

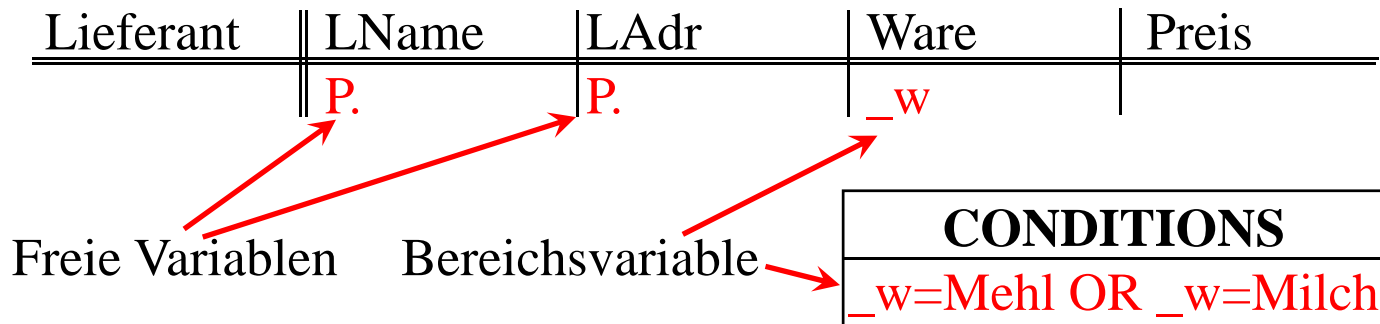
Kunde	KName	KAdr
	Huber	Innsbruck
	Maier	München

evtl. weitere Tabelle (Join)



Anfragen mit Bedingungen

- Welche Lieferanten liefern Mehl oder Milch?



- Bedeutung:
 $\{x_1, x_2 \mid \exists w, x_4: \text{Lieferant}(x_1, x_2, w, x_4) \wedge (w = \text{Mehl} \vee w = \text{Milch})\}$
- Kommando **P.** für print bzw. auch für die Projektion



Anfragen mit Bedingungen

- Welche Lieferanten liefern Brie und Perrier, wobei Gesamtpreis 7,00 € nicht übersteigt?

Lieferant	LName	LAdr	Ware	Preis
	P. _L		Brie	_y
	_L		Perrier	_z

CONDITIONS

$_y + _z \leq 7.00$

- Bedeutung:

$\{l \mid \exists x_1, x_2, y, z: \text{Lieferant}(l, x_1, \text{Brie}, y) \wedge$
 $\text{Lieferant}(l, x_2, \text{Perrier}, z) \wedge y + z \leq 7.00\}$



Join-Anfragen

- Welcher Lieferant liefert etwas das Huber bestellt hat?

Lieferant	LName	LAdr	Ware	Preis
	P.		_w	

Auftrag	KName	Ware	Menge
	Huber	_w	

- Bedeutung:
 $\{x_1 \mid \exists x_2, w, x_4, y_3: \text{Lieferant}(x_1, x_2, w, x_4) \wedge \text{Auftrag}(\text{Huber}, w, y_3)\}$
- Beachte:
Automatische Duplikat-Elimination in QBE



Join-Anfragen

Meist ist für Ergebnis neues Tabellengerüst nötig:

- Beispiel: Bestellungen mit Kontostand des Kunden
- Falsch (leider nicht möglich):

Kunde	KName	KAdr	Kto
	P. _n		P.
Auftrag	KName	Ware	Menge
	_n	P.	P.

- Richtig:

Kunde	KName	KAdr	Kto
	_n		_k
Auftrag	KName	Ware	Menge
	_n	_w	_m

Abkürzung!

Bestellung	Name	Was	Wieviel	Kontostand
P.	P. _n	P. _w	P. _m	P. _k



Anfragen mit Ungleichung

- Wer liefert Milch zu Preis zw. 0,50 € und 0,60 €?
- Variante mit zwei Zeilen:

Lieferant	LName	LAdr	Ware	Preis
P.	<u>L</u>		Milch	≥ 0.5
	<u>L</u>		Milch	≤ 0.6

- Variante mit Condition-Box

Lieferant	LName	LAdr	Ware	Preis
P.			Milch	<u>p</u>

CONDITIONS
$\text{p} \geq 0.5 \text{ AND } \text{p} \leq 0.6$



Anfragen mit Negation

- Finde für jede Ware den billigsten Lieferanten

Lieferant	LName	LAdr	Ware	Preis
P.			_w	_p
¬			_w	< _p

- Das Symbol **¬** in der ersten Spalte bedeutet:
Es gibt kein solches Tupel
- Bedeutung:
 $\{x_1, x_2, w, p \mid \neg \exists y_1, y_2, y_3: \text{Lieferant}(x_1, x_2, w, p) \wedge \text{Lieferant}(y_1, y_2, w, y_3) \wedge y_3 < p\}$



Einfügen

- Einfügen von einzelnen Tupeln
 - Kommando **I.** für INSERT

Kunde	KName	KAdr	Kto
I.	Schulz	Wien	0

- Einfügen von Tupeln aus einem Anfrageergebnis
 - Beispiel: Alle Lieferanten in Kundentabelle übernehmen

Kunde	KName	KAdr	Kto
I.	_n	_a	0

Lieferant	LName	LAdr	Ware	Preis
	_n	_a		



Löschen und Ändern

- Löschen aller Kunden mit negativem Kontostand

Kunde	KName	KAdr	Kto
D.			< 0

- Ändern eines Tupels (**U.** für UPDATE)

Kunde	KName	KAdr	Kto
	Schulz	Wien	U. 100

- oder auch:

Kunde	KName	KAdr	Kto
	Meier	_a	_k
U.	Meier	_a	_k - 110

- oder auch mit Condition-Box



Vergleich

QBE	Bereichskalkül
Konstanten	Konstanten
Bereichsvariablen	Bereichsvariablen
leere Spalten	paarweise verschiedene Bereichsvariablen, \exists -quantifiziert
Spalten mit P.	freie Variablen
Spalten ohne P.	\exists -quantifizierte Variablen

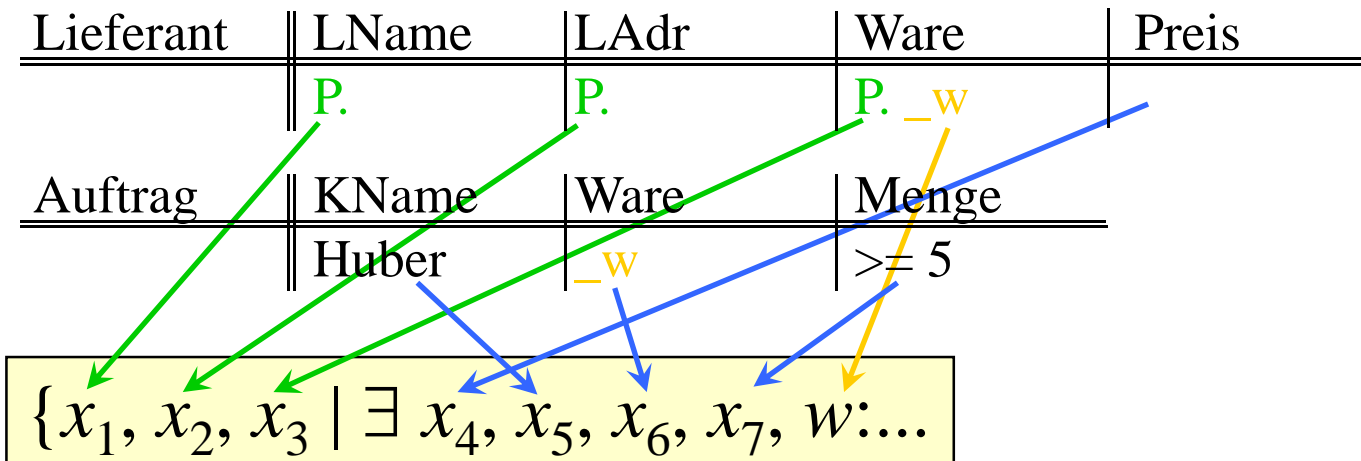
Anmerkung: QBE ist relational vollständig, jedoch ist für manche Anfragen der relationalen Algebra eine Folge von QBE-Anfragen nötig



Umsetzung einer QBE-Anfrage

(ohne Negation)

- Erzeuge für alle Attribute A_i aller vorkommenden Tabellen-Zeilen der Anfrage eine Bereichsvariable x_i
- Steht bei Attribut A_i das Kommando **P.**, dann schreibe x_i zu den freien Variablen ($\{\dots x_i, \dots \mid \dots\}$), sonst binde x_i mit einem \exists -Quantor ($\{\dots \mid \exists \dots, x_i, \dots\}$)
- Binde alle Variablen der Anfrage mit einem \exists -Quantor





Umsetzung einer QBE-Anfrage

Lieferant	LName	LAdr	Ware	Preis
	P.	P.	P. _w	
Auftrag	KName	Ware	Menge	
	Huber	_w	>= 5	

- Füge für jede vorkommende Relation R ein Atom der Form $R(x_i, x_{i+1}, \dots)$ mit \wedge an die Formel Ψ an

$\{x_1, x_2, x_3 \mid \exists x_4, x_5, x_6, x_7, w: \text{Lieferant}(x_1, x_2, x_3, x_4) \wedge \text{Auftrag}(x_5, x_6, x_7) \dots$

- Steht bei A_i ein Zusatz der Form **Const** bzw. \leq **Const** etc., dann hänge $x_i = \text{Const}$ bzw. $x_i \leq \text{Const}$ mit \wedge an Formel.

$\{x_1, x_2, x_3 \mid \exists x_4, x_5, x_6, x_7, w: \text{Lieferant}(x_1, x_2, x_3, x_4) \wedge \text{Auftrag}(x_5, x_6, x_7) \wedge x_5 = \text{Huber} \wedge x_7 \geq 5$



Umsetzung einer QBE-Anfrage

Lieferant	LName	LAdr	Ware	Preis
	P.	P.	P. $_w$	
Auftrag	KName	Ware	Menge	
	Huber	$_w$	≥ 5	

- Gleiches Vorgehen bei Zusätzen der Form $_Variable$ bzw. $\leq _Variable$ usw:

$$\{x_1, x_2, x_3 \mid \exists x_4, x_5, x_6, x_7, w: \text{Lieferant}(x_1, x_2, x_3, x_4) \wedge \text{Auftrag}(x_5, x_6, x_7) \wedge x_5 = \text{Huber} \wedge x_7 \geq 5 \wedge w = x_3 \wedge w = x_6\}$$

- Ggf. wird der Inhalt der Condition-Box mit \wedge angehängt.
- Meist lässt sich der Term noch vereinfachen:

$$\{x_1, x_2, w \mid \exists x_4, x_5, x_7: \text{Lieferant}(x_1, x_2, w, x_4) \wedge \text{Auftrag}(\text{Huber}, w, x_7) \wedge x_7 \geq 5\}$$