

# Kapitel DB:IV

## IV. Grundlagen relationaler Anfragesprachen

- Anfragen und Änderungen
- Relationale Algebra
- Anfragekalküle
- Relationaler Tupelkalkül
- Relationaler Domänenkalkül

# Anfragekalküle

## Einleitung

- Anfragealgebren spiegeln das Konzept von abstrakten Datenstrukturen wider; der Datentyp ist die Relation mit entsprechenden Operationen hierauf. Ein relationaler Ausdruck ist eine *prozedurale* Beschreibung, also eine genau festgelegte Folge von Operationen zur Berechnung einer Anfrage.

# Anfragekalküle

## Einleitung

- Anfragealgebren spiegeln das Konzept von abstrakten Datenstrukturen wider; der Datentyp ist die Relation mit entsprechenden Operationen hierauf. Ein relationaler Ausdruck ist eine *prozedurale* Beschreibung, also eine genau festgelegte Folge von Operationen zur Berechnung einer Anfrage.
- Anfragekalküle sind ein logikbasierter Ansatz zur Beschreibung der Ergebnismenge einer Anfrage.

Sie können als *deklarative* bzw. *nicht-prozedurale* Sprache aufgefasst werden. Insbesondere enthält eine Formel des Kalküls keine Information darüber, *wie* sie auszuwerten ist.

Für das relationale Modell betrachtet man folgende Kalküle:

1. relationaler Tupelkalkül
2. relationaler Domänenkalkül, auch Bereichskalkül genannt

# Anfragekalküle

## Einleitung (Fortsetzung)

### Beispiele für Formeln:

- (a)  $\text{Mitarbeiter}(t)$  alternativ:  $t \in \text{Mitarbeiter}$
- (b)  $\neg \text{Mitarbeiter}(\text{Smith}, 1234, \text{Weimar}, 3334, 5)$
- (c)  $\exists t : \text{Mitarbeiter}(t) \wedge t.\text{AbtNr} = '5'$

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

# Anfragekalküle

## Einleitung (Fortsetzung)

### Beispiele für Formeln:

(a)  $\text{Mitarbeiter}(t)$  alternativ:  $t \in \text{Mitarbeiter}$

Tupel  $t$  ist Element der Relation Mitarbeiter.

(b)  $\neg \text{Mitarbeiter}(\text{Smith}, 1234, \text{Weimar}, 3334, 5)$

(c)  $\exists t : \text{Mitarbeiter}(t) \wedge t.\text{AbtNr} = '5'$

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

# Anfragekalküle

## Einleitung (Fortsetzung)

### Beispiele für Formeln:

(a)  $\text{Mitarbeiter}(t)$  alternativ:  $t \in \text{Mitarbeiter}$

Tupel  $t$  ist Element der Relation Mitarbeiter.

(b)  $\neg \text{Mitarbeiter}(\text{Smith}, 1234, \text{Weimar}, 3334, 5)$

Tupel (Smith, 1234, Weimar, 3334, 5) ist nicht Element der Relation Mitarbeiter.

(c)  $\exists t : \text{Mitarbeiter}(t) \wedge t.\text{AbtNr} = '5'$

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

# Anfragekalküle

## Einleitung (Fortsetzung)

### Beispiele für Formeln:

(a)  $\text{Mitarbeiter}(t)$  alternativ:  $t \in \text{Mitarbeiter}$

Tupel  $t$  ist Element der Relation Mitarbeiter.

(b)  $\neg \text{Mitarbeiter}(\text{Smith}, 1234, \text{Weimar}, 3334, 5)$

Tupel (Smith, 1234, Weimar, 3334, 5) ist nicht Element der Relation Mitarbeiter.

(c)  $\exists t : \text{Mitarbeiter}(t) \wedge t.\text{AbtNr} = '5'$

Es gibt ein Tupel in der Relation Mitarbeiter, in dem AbtNr = '5' gilt.

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

# Anfragekalküle

## Einleitung (Fortsetzung)

- Die Kalkülsprache verwendet die Namen der Relationen und Attribute.
- Die Sätze der Kalkülsprache heißen *Formeln*.
- Die Grundbausteine der Formeln heißen *Atome*.
- Formeln, deren (Un)Wahrheit sich feststellen lässt, heißen *Aussagen*.

## Beispiele für Formeln:

(a)  $\text{Mitarbeiter}(t)$  alternativ:  $t \in \text{Mitarbeiter}$

Tupel  $t$  ist Element der Relation Mitarbeiter.

(b)  $\neg \text{Mitarbeiter}(\text{Smith}, 1234, \text{Weimar}, 3334, 5)$

Tupel (Smith, 1234, Weimar, 3334, 5) ist nicht Element der Relation Mitarbeiter.

(c)  $\exists t : \text{Mitarbeiter}(t) \wedge t.\text{AbtNr} = '5'$

Es gibt ein Tupel in der Relation Mitarbeiter, in dem AbtNr = '5' gilt.

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4



# Anfragekalküle

## Syntax I. Aufbau einer Formel / Grammatik der Sprache

Beispiele für Aussagen, Aussageformen, Atome:

- (a)  $\text{Mitarbeiter}(t)$  alternativ:  $t \in \text{Mitarbeiter}$
- (b)  $\neg \text{Mitarbeiter}(\text{Smith}, 1234, \text{Weimar}, 3334, 5)$
- (c)  $\exists t : \text{Mitarbeiter}(t) \wedge t.\text{AbtNr} = '5'$

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

# Anfragekalküle

## Syntax I. Aufbau einer Formel / Grammatik der Sprache

Beispiele für Aussagen, Aussageformen, Atome:

- (a)  $\text{Mitarbeiter}(t)$  alternativ:  $t \in \text{Mitarbeiter}$       Aussageform, Atom
- (b)  $\neg \text{Mitarbeiter}(\text{Smith}, 1234, \text{Weimar}, 3334, 5)$
- (c)  $\exists t : \text{Mitarbeiter}(t) \wedge t.\text{AbtNr} = '5'$

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

# Anfragekalküle

## Syntax I. Aufbau einer Formel / Grammatik der Sprache

Beispiele für Aussagen, Aussageformen, Atome:

- (a)  $\text{Mitarbeiter}(t)$  alternativ:  $t \in \text{Mitarbeiter}$  Aussageform, Atom
- (b)  $\neg \text{Mitarbeiter}(\text{Smith}, 1234, \text{Weimar}, 3334, 5)$  Aussage (falsch), keine atomare Formel
- (c)  $\exists t : \text{Mitarbeiter}(t) \wedge t.\text{AbtNr} = '5'$

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

# Anfragekalküle

## Syntax I. Aufbau einer Formel / Grammatik der Sprache

Beispiele für Aussagen, Aussageformen, Atome:

- (a)  $\text{Mitarbeiter}(t)$  alternativ:  $t \in \text{Mitarbeiter}$  Aussageform, Atom
- (b)  $\neg \text{Mitarbeiter}(\text{Smith}, 1234, \text{Weimar}, 3334, 5)$  Aussage (falsch), keine atomare Formel
- (c)  $\exists t : \text{Mitarbeiter}(t) \wedge t.\text{AbtNr} = '5'$  Aussage (wahr), keine atomare Formel

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

# Anfragekalküle

## Syntax I. Aufbau einer Formel / Grammatik der Sprache

Sei  $\Sigma$  eine Menge von Atomen aus einem Anfragekalkül, dann sind folgende Ausdrücke Formeln in diesem Kalkül:

1. Jedes Atom in  $\Sigma$  ist eine Formel.
2. Sind  $\alpha$  und  $\beta$  Formeln, so sind es auch  $(\alpha)$ ,  $\neg\alpha$ ,  $\alpha \wedge \beta$ ,  $\alpha \vee \beta$  und  $\alpha \rightarrow \beta$ .

Beispiele für Aussagen, Aussageformen, Atome:

- (a)  $\text{Mitarbeiter}(t)$  alternativ:  $t \in \text{Mitarbeiter}$       Aussageform, Atom
- (b)  $\neg\text{Mitarbeiter}(\text{Smith}, 1234, \text{Weimar}, 3334, 5)$       Aussage (falsch), keine atomare Formel
- (c)  $\exists t : \text{Mitarbeiter}(t) \wedge t.\text{AbtNr} = '5'$       Aussage (wahr), keine atomare Formel

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

# Anfragekalküle

## Semantik I. Bewertung einer Formel / Interpretation der Sprache

Beispiele für die Bewertung von Formeln:

$$(b) \quad \mathcal{I}(\underbrace{\neg \text{Mitarbeiter}(\text{Smith}, 1234, \text{Weimar}, 3334, 5)}_{\alpha}) = 0$$

$$(c) \quad \mathcal{I}(\underbrace{\exists t : \overbrace{\text{Mitarbeiter}(t)}^{\text{Atom}} \wedge \overbrace{t.\text{AbtNr} = '5'}^{\text{Atom}}}_{\alpha}) = 1$$

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

# Anfragekalküle

## Semantik I. Bewertung einer Formel / Interpretation der Sprache

1. Auf Basis des Datenbankzustandes  $d(\mathcal{R}) = \{r_1, \dots, r_p\}$  lässt sich einem Atom  $\alpha \in \Sigma$  ein Wahrheitswert, in Zeichen:  $\mathcal{I}(\alpha)$ , zuweisen.
2. Auf Basis der Wahrheitswerte der Atome lässt sich rekursiv gemäß der Semantik für  $\neg, \wedge, \vee, \rightarrow$  einer Formel  $\alpha$  ein Wahrheitswert  $\mathcal{I}(\alpha)$  zuweisen.

Beispiele für die Bewertung von Formeln:

$$(b) \quad \underbrace{\mathcal{I}(\neg \text{Mitarbeiter}(\text{Smith}, 1234, \text{Weimar}, 3334, 5))}_{\alpha} = 0$$

$$(c) \quad \underbrace{\mathcal{I}(\exists t : \overbrace{\text{Mitarbeiter}(t)}^{\text{Atom}} \wedge \overbrace{t.\text{AbtNr} = '5'}^{\text{Atom}})}_{\alpha} = 1$$

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Bemerkungen:

- Ein Atom ist die einfachste Formel. Eine Formel stellt entweder eine Aussage oder eine Aussageform dar. Von einer Aussage lässt sich feststellen, ob sie wahr oder falsch ist; von einer Aussageform lässt sich nicht die Wahrheit bzw. Unwahrheit feststellen.
- In einem Anfragekalkül geschieht die Feststellung des Wahrheitswertes (= Interpretation, Semantik) einer atomaren Aussage auf Basis des Datenbankzustandes:

Ein Atom entspricht dem Test, ob ein bestimmtes Wertetupel  $t$  oder ein bestimmter Attributwert  $x$  ein Element einer Relation  $r$  im Datenbankzustand  $d(\mathcal{R}) = \{r_1, \dots, r_p\}$  ist.

- Der Wahrheitswert (= Interpretation, Semantik) einer komplexen Formel leitet sich in eindeutiger Weise von den Wahrheitswerten der Atome der Formel ab. Dabei ist die Verknüpfung von Wahrheitswerten mit den Junktoren (= Operatoren)  $\neg, \wedge, \vee, \rightarrow$  wie folgt definiert:

$\neg$	
$\alpha$	0 1
1	0

$\wedge$	$\beta$	0	1
$\alpha$	0	0	0
1	1	0	1

$\vee$	$\beta$	0	1
$\alpha$	0	0	1
1	1	1	1

$\rightarrow$	$\beta$	0	1
$\alpha$	0	1	1
1	1	0	1

- Die Interpretationsfunktion wird mit  $\mathcal{I}$  bezeichnet und liefert für eine Formel  $\alpha$ , die eine Aussage darstellt, ihren Wahrheitswert:  $\alpha \mapsto \mathcal{I}(\alpha), \mathcal{I}(\alpha) \in \{0, 1\}$ .
- Mit Formeln werden die Bedingungen einer Datenbankanfrage nachgebildet.



# Anfragekalküle

## Freie und gebundene Variablen

Sei  $\alpha$  eine Formel, die eine Variable  $x$  enthält. Dann sei vereinbart:

- (a) Ist  $\alpha$  ein Atom, so ist  $x$  eine *freie Variable*.
- (b) Das Vorkommen von  $x$  in  $(\alpha)$ ,  $\neg\alpha$ ,  $\alpha \wedge \beta$ ,  $\alpha \vee \beta$  und  $\alpha \rightarrow \beta$  ist *frei* oder *gebunden* – abhängig davon, ob es in  $\alpha$  frei oder gebunden ist.
- (c) Alle freien Vorkommen von  $x$  in  $\alpha$  sind gebunden in  $\exists x\alpha$  und  $\forall x\alpha$ .
- (d) In keiner Formel darf eine Variable sowohl frei als auch gebunden auftreten.

# Anfragekalküle

## Freie und gebundene Variablen

Sei  $\alpha$  eine Formel, die eine Variable  $x$  enthält. Dann sei vereinbart:

- (a) Ist  $\alpha$  ein Atom, so ist  $x$  eine *freie Variable*.
- (b) Das Vorkommen von  $x$  in  $(\alpha)$ ,  $\neg\alpha$ ,  $\alpha \wedge \beta$ ,  $\alpha \vee \beta$  und  $\alpha \rightarrow \beta$  ist *frei* oder *gebunden* – abhängig davon, ob es in  $\alpha$  frei oder gebunden ist.
- (c) Alle freien Vorkommen von  $x$  in  $\alpha$  sind gebunden in  $\exists x\alpha$  und  $\forall x\alpha$ .
- (d) In keiner Formel darf eine Variable sowohl frei als auch gebunden auftreten.

# Anfragekalküle

## Freie und gebundene Variablen

Sei  $\alpha$  eine Formel, die eine Variable  $x$  enthält. Dann sei vereinbart:

- (a) Ist  $\alpha$  ein Atom, so ist  $x$  eine *freie Variable*.
- (b) Das Vorkommen von  $x$  in  $(\alpha)$ ,  $\neg\alpha$ ,  $\alpha \wedge \beta$ ,  $\alpha \vee \beta$  und  $\alpha \rightarrow \beta$  ist *frei* oder *gebunden* – abhängig davon, ob es in  $\alpha$  frei oder gebunden ist.
- (c) Alle freien Vorkommen von  $x$  in  $\alpha$  sind gebunden in  $\exists x\alpha$  und  $\forall x\alpha$ .
- (d) In keiner Formel darf eine Variable sowohl frei als auch gebunden auftreten.

# Anfragekalküle

## Freie und gebundene Variablen

Sei  $\alpha$  eine Formel, die eine Variable  $x$  enthält. Dann sei vereinbart:

- (a) Ist  $\alpha$  ein Atom, so ist  $x$  eine *freie Variable*.
- (b) Das Vorkommen von  $x$  in  $(\alpha)$ ,  $\neg\alpha$ ,  $\alpha \wedge \beta$ ,  $\alpha \vee \beta$  und  $\alpha \rightarrow \beta$  ist *frei* oder *gebunden* – abhängig davon, ob es in  $\alpha$  frei oder gebunden ist.
- (c) Alle freien Vorkommen von  $x$  in  $\alpha$  sind gebunden in  $\exists x\alpha$  und  $\forall x\alpha$ .
- (d) In keiner Formel darf eine Variable sowohl frei als auch gebunden auftreten.

# Anfragekalküle

## Freie und gebundene Variablen

Sei  $\alpha$  eine Formel, die eine Variable  $x$  enthält. Dann sei vereinbart:

- (a) Ist  $\alpha$  ein Atom, so ist  $x$  eine *freie Variable*.
- (b) Das Vorkommen von  $x$  in  $(\alpha)$ ,  $\neg\alpha$ ,  $\alpha \wedge \beta$ ,  $\alpha \vee \beta$  und  $\alpha \rightarrow \beta$  ist *frei* oder *gebunden* – abhängig davon, ob es in  $\alpha$  frei oder gebunden ist.
- (c) Alle freien Vorkommen von  $x$  in  $\alpha$  sind gebunden in  $\exists x\alpha$  und  $\forall x\alpha$ .
- (d) In keiner Formel darf eine Variable sowohl frei als auch gebunden auftreten.

## Syntax II. Aufbau einer Formel / Grammatik der Sprache [Syntax I]

- 3. Ist  $\alpha$  eine Formel, so sind es auch  $\exists x\alpha$  und  $\forall x\alpha$  – wobei  $x$  eine Variable ist, die in  $\alpha$  frei vorkommt.

# Anfragekalküle

## Freie und gebundene Variablen

Sei  $\alpha$  eine Formel, die eine Variable  $x$  enthält. Dann sei vereinbart:

- (a) Ist  $\alpha$  ein Atom, so ist  $x$  eine *freie Variable*.
- (b) Das Vorkommen von  $x$  in  $(\alpha)$ ,  $\neg\alpha$ ,  $\alpha \wedge \beta$ ,  $\alpha \vee \beta$  und  $\alpha \rightarrow \beta$  ist *frei* oder *gebunden* – abhängig davon, ob es in  $\alpha$  frei oder gebunden ist.
- (c) Alle freien Vorkommen von  $x$  in  $\alpha$  sind gebunden in  $\exists x\alpha$  und  $\forall x\alpha$ .
- (d) In keiner Formel darf eine Variable sowohl frei als auch gebunden auftreten.

## Syntax II. Aufbau einer Formel / Grammatik der Sprache [Syntax I]

3. Ist  $\alpha$  eine Formel, so sind es auch  $\exists x\alpha$  und  $\forall x\alpha$  – wobei  $x$  eine Variable ist, die in  $\alpha$  frei vorkommt.

## Semantik II. Bewertung einer Formel / Interpretation der Sprache [Semantik I]

3. Eine Formel  $\exists x\alpha$  ist wahr, falls  $\alpha$  bzgl. *einer* Instanziierung von  $x$  wahr wird.  
Eine Formel  $\forall x\alpha$  ist wahr, falls  $\alpha$  bzgl. *aller* Instanziierungen von  $x$  wahr wird.  
Die Menge der möglichen Instanziierungen heißt *Universum*.

## Bemerkungen:

- ❑ Eine Formel mit freien Variablen stellt eine Aussageform dar. Eine Formel, die keine oder nur gebundene (= instanziierte) Variablen enthält, stellt eine Aussage dar.
- ❑ Die für eine Instanziierung zur Verfügung stehenden Wertetupel  $t$  bzw. Attributwerte  $x$  stammen aus einem Grundbereich, auch *Universum* genannt. Das Universum enthält *alle möglichen* Tupel bzw. Attributwerte, die in einem Datenbankzustand vorliegen können.
- ❑ Im Allgemeinen enthält das Universum unendlich viele Elemente.

# Anfragekalküle

## Auswertung einer Anfrage

Gegeben:

- Anfrage  $\{ \cdot \mid \alpha \}$  mit freien Variablen im Tupelausdruck  $\cdot$  und Formel  $\alpha$
- Datenbankzustand  $d(\mathcal{R}) = \{r_1, \dots, r_p\}$



# Anfragekalküle

## Auswertung einer Anfrage

Gegeben:

- Anfrage  $\{ \cdot \mid \alpha \}$  mit freien Variablen im Tupelausdruck  $\cdot$  und Formel  $\alpha$
- Datenbankzustand  $d(\mathcal{R}) = \{r_1, \dots, r_p\}$

Beispiel:

$d(\mathcal{R}) =$

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

### Anfrage

„Liefere die Namen der Mitarbeiter, die in Abteilung 5 arbeiten.“

### Relationenalgebra

### Tupelkalkül

# Anfragekalküle

## Auswertung einer Anfrage

Gegeben:

- Anfrage  $\{ \cdot \mid \alpha \}$  mit freien Variablen im Tupelausdruck  $\cdot$  und Formel  $\alpha$
- Datenbankzustand  $d(\mathcal{R}) = \{r_1, \dots, r_p\}$

Beispiel:

$d(\mathcal{R}) =$

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

### Anfrage

„Liefere die Namen der Mitarbeiter, die in Abteilung 5 arbeiten.“

### Relationenalgebra

$\pi_{\text{Name}}(\sigma_{\text{AbtNr}='5'}(\text{Mitarbeiter}))$

### Tupelkalkül

# Anfragekalküle

## Auswertung einer Anfrage

Gegeben:

- Anfrage  $\{ \cdot \mid \alpha \}$  mit freien Variablen im Tupelausdruck  $\cdot$  und Formel  $\alpha$
- Datenbankzustand  $d(\mathcal{R}) = \{r_1, \dots, r_p\}$

Beispiel:

$d(\mathcal{R}) =$

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

### Anfrage

„Liefere die Namen der Mitarbeiter, die in Abteilung 5 arbeiten.“

### Relationenalgebra

$\pi_{\text{Name}}(\sigma_{\text{AbtNr}='5'}(\text{Mitarbeiter}))$

### Tupelkalkül

$\{ \cdot \mid \alpha \} \sim \{ (t.\text{Name}) \mid \text{Mitarbeiter}(t) \wedge t.\text{AbtNr} = '5' \}$

# Anfragekalküle

## Auswertung einer Anfrage

Gegeben:

- Anfrage  $\{ \cdot \mid \alpha \}$  mit freien Variablen im Tupelausdruck  $\cdot$  und Formel  $\alpha$
- Datenbankzustand  $d(\mathcal{R}) = \{r_1, \dots, r_p\}$

Beispiel:

$d(\mathcal{R}) =$

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

### Anfrage

„Liefere die Namen der Mitarbeiter, die in Abteilung 5 arbeiten.“

### Relationenalgebra

$\pi_{\text{Name}}(\sigma_{\text{AbtNr}='5'}(\text{Mitarbeiter}))$

### Tupelkalkül

$\{ \cdot \mid \alpha \} \sim \{ (t.\text{Name}) \mid \text{Mitarbeiter}(t) \wedge t.\text{AbtNr} = '5' \} = \{ (\text{Smith}), (\text{Wong}) \}$

# Anfragekalküle

## Auswertung einer Anfrage

Gegeben:

- Anfrage  $\{ \cdot \mid \alpha \}$  mit freien Variablen im Tupelausdruck  $\cdot$  und Formel  $\alpha$
- Datenbankzustand  $d(\mathcal{R}) = \{r_1, \dots, r_p\}$

Beispiel:

$d(\mathcal{R}) =$

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

### Anfrage

„Liefere die Namen der Mitarbeiter, die in Abteilung 5 arbeiten.“

### Relationenalgebra

$$\pi_{\text{Name}}(\sigma_{\text{AbtNr}='5'}(\text{Mitarbeiter}))$$

### Tupelkalkül

$$\{ \cdot \mid \alpha \} \sim \{ (t.\text{Name}) \mid \overbrace{\text{Mitarbeiter}(t)}^{\text{Atom}} \wedge \overbrace{t.\text{AbtNr}='5'}^{\text{Atom}} \} = \overbrace{\{(Smith), (Wong)\}}^{\text{Ergebnisrelation}}$$

# Anfragekalküle

## Auswertung einer Anfrage (Fortsetzung)

Gegeben:

- Anfrage  $\{ \cdot \mid \alpha \}$  mit freien Variablen im Tupelausdruck  $\cdot$  und Formel  $\alpha$
- Datenbankzustand  $d(\mathcal{R}) = \{r_1, \dots, r_p\}$

Schema zur Konstruktion der **Ergebnisrelation**  $res$  für Anfrage  $\{ \cdot \mid \alpha \}$ :

1.  $res = \emptyset$
2. Die freien Variablen (= die Variablen in  $\cdot$ ) werden hinsichtlich aller Tupel (im Tupelkalkül) bzw. aller Attributwerte (im Domänenkalkül) für die in der Datenbank befindlichen Relationen  $\{r_1, \dots, r_p\}$  instanziiert.

Durch die Instanziierung wird die Aussageform  $\alpha$  zu einer Aussage.

3. Für jede Instanziierung der freien Variablen wird geprüft, ob die Formel  $\alpha$  wahr (erfüllt) ist. Falls ja, setze  $res = res \cup \{ \cdot \}$
4.  $res$  enthält keine weiteren Elemente.

# Anfragekalküle

## Auswertung einer Anfrage (Fortsetzung)

Gegeben:

- Anfrage  $\{ \cdot \mid \alpha \}$  mit freien Variablen im Tupelausdruck  $\cdot$  und Formel  $\alpha$
- Datenbankzustand  $d(\mathcal{R}) = \{r_1, \dots, r_p\}$

Schema zur Konstruktion der **Ergebnisrelation**  $res$  für Anfrage  $\{ \cdot \mid \alpha \}$ :

1.  $res = \emptyset$
2. Die freien Variablen (= die Variablen in  $\cdot$ ) werden hinsichtlich aller Tupel (im Tupelkalkül) bzw. aller Attributwerte (im Domänenkalkül) für die in der Datenbank befindlichen Relationen  $\{r_1, \dots, r_p\}$  instanziiert.

Durch die Instanziierung wird die Aussageform  $\alpha$  zu einer Aussage.

3. Für jede Instanziierung der freien Variablen wird geprüft, ob die Formel  $\alpha$  wahr (erfüllt) ist. Falls ja, setze  $res = res \cup \{ \cdot \}$
4.  $res$  enthält keine weiteren Elemente.

# Anfragekalküle

## Auswertung einer Anfrage (Fortsetzung)

Gegeben:

- Anfrage  $\{ \cdot \mid \alpha \}$  mit freien Variablen im Tupelausdruck  $\cdot$  und Formel  $\alpha$
- Datenbankzustand  $d(\mathcal{R}) = \{r_1, \dots, r_p\}$

Schema zur Konstruktion der **Ergebnisrelation**  $res$  für Anfrage  $\{ \cdot \mid \alpha \}$ :

1.  $res = \emptyset$
2. Die freien Variablen (= die Variablen in  $\cdot$ ) werden hinsichtlich aller Tupel (im Tupelkalkül) bzw. aller Attributwerte (im Domänenkalkül) für die in der Datenbank befindlichen Relationen  $\{r_1, \dots, r_p\}$  instanziiert.

Durch die Instanziierung wird die Aussageform  $\alpha$  zu einer Aussage.

3. Für jede Instanziierung der freien Variablen wird geprüft, ob die Formel  $\alpha$  wahr (erfüllt) ist. Falls ja, setze  $res = res \cup \{ \cdot \}$
4.  $res$  enthält keine weiteren Elemente.



# Kapitel DB:IV

## IV. Grundlagen relationaler Anfragesprachen

- Anfragen und Änderungen
- Relationale Algebra
- Anfragekalküle
- Relationaler Tupelkalkül
- Relationaler Domänenkalkül

# Relationaler Tupelkalkül

## Konzepte [\[Domänenkalkül\]](#)

1. Tupelvariablen, die sich auf **Relationen**  $r_i \in d(\mathcal{R})$ ,  $d(\mathcal{R}) = \{r_1, \dots, r_p\}$ , beziehen und mit jedem Tupel aus  $r_i$  instanziiert werden können.
2. Formeln, mit denen sich auf Basis der Tupelvariablen Zusammenhänge zwischen Attributen formulieren lassen.

# Relationaler Tupelkalkül

Anfragen [Domänenkalkül]

Anfrage im relationalen Tupelkalkül mit Variablen  $t_1, t_2, \dots, t_n, t_{n+1}, \dots, t_{n+m}$  :

$$\{t \mid \alpha\} \quad \text{allgemein :} \quad \{(t_1.A_1, t_2.A_2, \dots, t_n.A_n) \mid \alpha\}$$

- $t_1, \dots, t_n$  sind freie,  $t_{n+1}, \dots, t_{n+m}$  sind gebundene Tupelvariablen.
- $A_1, \dots, A_n$  sind Attribute der Relationen bzgl. derer die  $t_i$  instanziiert sind.
- $\alpha$  ist eine logische Formel, wobei die Menge der Atome  $\Sigma$ , aus denen  $\alpha$  besteht, wie folgt definiert ist:

# Relationaler Tupelkalkül

## Anfragen [Domänenkalkül]

Anfrage im relationalen Tupelkalkül mit Variablen  $t_1, t_2, \dots, t_n, t_{n+1}, \dots, t_{n+m}$  :

$$\{t \mid \alpha\} \quad \text{allgemein : } \{(t_1.A_1, t_2.A_2, \dots, t_n.A_n) \mid \alpha\}$$

- $t_1, \dots, t_n$  sind freie,  $t_{n+1}, \dots, t_{n+m}$  sind gebundene Tupelvariablen.
- $A_1, \dots, A_n$  sind Attribute der Relationen bzgl. derer die  $t_i$  instanziiert sind.
- $\alpha$  ist eine logische Formel, wobei die Menge der Atome  $\Sigma$ , aus denen  $\alpha$  besteht, wie folgt definiert ist:
  1. „ $r(t)$ “    alternativ: „ $t \in r$ “  
ist ein Atom, wobei  $t$  eine Tupelvariable und  $r$  eine Relation bezeichnet.  $r(t)$  ist wahr für eine Instanziierung von  $t$ , falls diese Instanziierung ein Tupel in  $r$  ist.

# Relationaler Tupelkalkül

## Anfragen [Domänenkalkül]

Anfrage im relationalen Tupelkalkül mit Variablen  $t_1, t_2, \dots, t_n, t_{n+1}, \dots, t_{n+m}$  :

$$\{t \mid \alpha\} \quad \text{allgemein :} \quad \{(t_1.A_1, t_2.A_2, \dots, t_n.A_n) \mid \alpha\}$$

- $t_1, \dots, t_n$  sind freie,  $t_{n+1}, \dots, t_{n+m}$  sind gebundene Tupelvariablen.
- $A_1, \dots, A_n$  sind Attribute der Relationen bzgl. derer die  $t_i$  instanziiert sind.
- $\alpha$  ist eine logische Formel, wobei die Menge der Atome  $\Sigma$ , aus denen  $\alpha$  besteht, wie folgt definiert ist:
  1. „ $r(t)$ “ alternativ: „ $t \in r$ “  
ist ein Atom, wobei  $t$  eine Tupelvariable und  $r$  eine Relation bezeichnet.  $r(t)$  ist wahr für eine Instanziierung von  $t$ , falls diese Instanziierung ein Tupel in  $r$  ist.
  2. „ $t_i.A_i \text{ op } t_j.A_j$ “  
ist ein Atom mit  $\text{op} \in \{=, <, \leq, >, \geq, \neq\}$ .  $t_i, t_j$  bezeichnen Tupelvariablen und  $A_i, A_j$  bezeichnen Attribute aus den Relationen hinsichtlich derer  $t_i$  bzw.  $t_j$  instanziiert sind.
  3. „ $t_i.A_i \text{ op } c$ “  
ist ein Atom mit  $\text{op} \in \{=, <, \leq, >, \geq, \neq\}$ .  $t_i$  bezeichnet eine Tupelvariable,  $A_i$  ein Attribut aus der Relation hinsichtlich der  $t_i$  instanziiert ist und  $c \in \text{dom}(A_i)$  ist eine Konstante.

## Bemerkungen:

- „ $t_i.A_i \text{ op } t_j.A_j$ “ ist wahr für eine Instanziierung von  $t_i$  und  $t_j$ , falls „ $t_i.A_i \text{ op } t_j.A_j$ “ ein Element der mit  $op$  bezeichneten Relation ist.

Beispiel:  $t_1.\text{ChefPersNr} < t_1.\text{PersNr}$ , mit  $t_1.\text{ChefPersNr} = 9876$  und  $t_1.\text{PersNr} = 9998$

- „ $t_i.A_i \text{ op } c$ “ ist wahr für eine Instanziierung von  $t_i$ , falls „ $t_i.A_i \text{ op } c$ “ ein Element der mit  $op$  bezeichneten Relation ist.

Beispiel:  $t_6.\text{AbtNr} = '5'$ , mit  $t_6.\text{AbtNr} = 5$

# Relationaler Tupelkalkül

## Beispiel 1 [Domänenkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

### Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

# Relationaler Tupelkalkül

## Beispiel 1 [Domänenkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

## Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

## Relationenalgebra

$\pi_{\text{Name, Wohnort}}(\text{Mitarbeiter} \bowtie_{\text{AbtNr}=\text{Nr}} (\sigma_{\text{AbtName}='Forschung'}(\text{Abteilung})))$



# Relationaler Tupelkalkül

## Beispiel 1 [Domänenkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

## Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

## Relationenalgebra

$$\pi_{\text{Name, Wohnort}}(\text{Mitarbeiter} \bowtie_{\text{AbtNr}=\text{Nr}} (\sigma_{\text{AbtName}='Forschung'}(\text{Abteilung})))$$

## Tupelkalkül

$$\{(t_1.\text{Name}, t_1.\text{Wohnort}) \mid \text{Mitarbeiter}(t_1) \wedge \exists t_2(\text{Abteilung}(t_2) \wedge t_2.\text{AbtName} = 'Forschung' \wedge t_2.\text{Nr} = t_1.\text{AbtNr})\}$$

## Bemerkungen: [Domänenkalkül]

- Die Attribute der freien Tupelvariablen (vor dem Trennsymbol „|“ bei der Mengenbildung) entsprechen der Projektion  $\pi$  in der relationalen Algebra.

Beispiel:  $\{(t_1.\text{Name}, \dots) \mid \dots\}$

- Eine Bedingung, die sich auf ein Attribut und eine Konstante bezieht, entspricht einer Selektion,  $\sigma$ , in der relationalen Algebra.

Beispiel:  $t_2.\text{AbtName} = \text{'Forschung'}$

- Eine Bedingung bzgl. zweier Attribute, die sich auf Tupel aus verschiedenen Relationen bezieht, entspricht einem Verbund (Join),  $\bowtie$ , in der relationalen Algebra.

Beispiel:  $t_2.\text{Nr} = t_1.\text{AbtNr}$

# Relationaler Tupelkalkül

## Beispiel 2 [Domänenkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

### Anfrage

„Liefere für jedes Projekt in Weimar dessen Nummer, die Nummer der durchführenden Abteilung sowie Name und Wohnort des Abteilungsmanagers.“

# Relationaler Tupelkalkül

## Beispiel 2 [Domänenkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

### Anfrage

„Liefere für jedes Projekt in Weimar dessen Nummer, die Nummer der durchführenden Abteilung sowie Name und Wohnort des Abteilungsmanagers.“

### Relationenalgebra

$$\pi_{\text{PNr, AbtNr, Name, Wohnort}} \left( \left( \rho_{\text{PNr} \leftarrow \text{Nr}, \text{PName} \leftarrow \text{Name}} \left( \sigma_{\text{Ort}='Weimar'}(\text{Projekt}) \right) \right) \bowtie_{\text{AbtNr}=\text{Nr}} \text{Abteilung} \right) \bowtie_{\text{Manager}=\text{PersNr}} \text{Mitarbeiter} \right)$$

# Relationaler Tupelkalkül

## Beispiel 2 [Domänenkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

### Anfrage

„Liefere für jedes Projekt in Weimar dessen Nummer, die Nummer der durchführenden Abteilung sowie Name und Wohnort des Abteilungsmanagers.“

### Relationenalgebra

$$\pi_{\text{PNr, AbtNr, Name, Wohnort}} \left( \left( \rho_{\text{PNr} \leftarrow \text{Nr}, \text{PName} \leftarrow \text{Name}} \left( \sigma_{\text{Ort} = \text{'Weimar'}}(\text{Projekt}) \right) \bowtie_{\text{AbtNr} = \text{Nr}} \text{Abteilung} \bowtie_{\text{Manager} = \text{PersNr}} \text{Mitarbeiter} \right) \right)$$

### Tupelkalkül

$$\{ (t_1.\text{Name}, t_1.\text{Wohnort}, t_3.\text{Nr}, t_3.\text{AbtNr}) \mid \text{Mitarbeiter}(t_1) \wedge \text{Projekt}(t_3) \wedge t_3.\text{Ort} = \text{'Weimar'} \wedge \exists t_2 (\text{Abteilung}(t_2) \wedge t_2.\text{Nr} = t_3.\text{AbtNr} \wedge t_2.\text{Manager} = t_1.\text{PersNr}) \}$$

# Relationaler Tupelkalkül

## Beispiel 3 [Domänenkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

### Anfrage

„Liefere die Namen der Mitarbeiter, die in *allen* Projekten der Abteilung 5 arbeiten.“

# Relationaler Tupelkalkül

## Beispiel 3 [Domänenkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

### Anfrage

„Liefere die Namen der Mitarbeiter, die in *allen* Projekten der Abteilung 5 arbeiten.“

### Relationenalgebra

$\pi_{\text{Name}} \left( \left( \text{ArbeitetInProjekt} \div \rho_{\text{ProjektNr} \leftarrow \text{Nr}} \left( \pi_{\text{Nr}} \left( \sigma_{\text{AbtNr}=5'} \left( \text{Projekt} \right) \right) \right) \right) \bowtie \text{Mitarbeiter} \right)$

# Relationaler Tupelkalkül

## Beispiel 3 [Domänenkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

## Anfrage

„Liefere die Namen der Mitarbeiter, die in *allen* Projekten der Abteilung 5 arbeiten.“

## Relationenalgebra

$$\pi_{\text{Name}} \left( \left( \text{ArbeitetInProjekt} \div \rho_{\text{ProjektNr} \leftarrow \text{Nr}} \left( \pi_{\text{Nr}} \left( \sigma_{\text{AbtNr} = '5'} \left( \text{Projekt} \right) \right) \right) \right) \bowtie \text{Mitarbeiter} \right)$$

## Tupelkalkül

$$\{(t_1.\text{Name}) \mid \text{Mitarbeiter}(t_1) \wedge \forall t_3 (\neg (\text{Projekt}(t_3) \wedge t_3.\text{AbtNr} = '5') \vee \text{oder: } \forall t_3 ((\text{Projekt}(t_3) \wedge t_3.\text{AbtNr} = '5') \rightarrow \exists t_4 (\text{ArbeitetInProjekt}(t_4) \wedge t_4.\text{ProjektNr} = t_3.\text{Nr} \wedge t_4.\text{PersNr} = t_1.\text{PersNr})) \vee \dots)\}$$



## Bemerkungen:

- Interpretation des  $\forall$ -Quantors in der folgenden Formel:  $\alpha = \text{Mitarbeiter}(t_1) \wedge \forall t_3 \beta$

Semantik: „ $\alpha$  ist erfüllt für diejenigen Mitarbeiter(tupel  $t_1$ ), bei denen für *alle* Tupel  $t_3$  die Teilformel  $\beta$  erfüllt ist.“

Beachte, dass  $t_3$  an *alle* Tupel des Universums bzw. des Datenbankzustandes  $d(\mathcal{R})$  gebunden wird (also nicht nur an die Tupel von Projekt) und bzgl. *aller* möglichen Instanziierungen die Formel  $\beta$  erfüllen muss.

- Die Quantoren können verschoben werden, solange sich keine Variablenbindungen ändern und die Ordnung zwischen den  $\exists$ - und  $\forall$ -Quantoren erhalten bleibt:

$$\{(t_1.\text{Name}) \mid \text{Mitarbeiter}(t_1) \wedge \forall t_3 \exists t_4 (\neg(\text{Projekt}(t_3) \wedge (t_3.\text{AbtNr} = '5')) \vee (\text{ArbeitetInProjekt}(t_4) \wedge t_4.\text{ProjektNr} = t_3.\text{Nr} \wedge t_4.\text{PersNr} = t_1.\text{PersNr}))\}$$

# Relationaler Tupelkalkül

## Beispiel 4 [Domänenkalkül]

Buecher	
Titel	Verlag
Harry Potter	Princeton
Heuristics	Addison
Glücksformel	dpunkt
Datenbanken	Springer

Buchhaendler		
Name	Stadt	PLZ
Lehmann	Berlin	99011
Meiersche	Aachen	42100
Amazon	Köln	52100

Angebote	
Titel	Haendler
Harry Potter	Lehmann
Harry Potter	Meiersche
Harry Potter	Amazon
Datenbanken	Amazon
Glücksformel	Amazon
Glücksformel	Lehmann

### Anfrage

„Welche Titel sind bei *allen* Buchhändlern im Angebot?“

# Relationaler Tupelkalkül

## Beispiel 4 [Domänenkalkül]

Buecher	
Titel	Verlag
Harry Potter	Princeton
Heuristics	Addison
Glücksformel	dpunkt
Datenbanken	Springer

Buchhaendler		
Name	Stadt	PLZ
Lehmann	Berlin	99011
Meiersche	Aachen	42100
Amazon	Köln	52100

Angebote	
Titel	Haendler
Harry Potter	Lehmann
Harry Potter	Meiersche
Harry Potter	Amazon
Datenbanken	Amazon
Glücksformel	Amazon
Glücksformel	Lehmann

### Anfrage

„Welche Titel sind bei *allen* Buchhändlern im Angebot?“

### Relationenalgebra

$(\rho_{\text{Name} \leftarrow \text{Haendler}}(\text{Angebote})) \div (\pi_{\text{Name}}(\text{Buchhaendler}))$

# Relationaler Tupelkalkül

## Beispiel 4 [Domänenkalkül]

Buecher	
Titel	Verlag
Harry Potter	Princeton
Heuristics	Addison
Glücksformel	dpunkt
Datenbanken	Springer

Buchhaendler		
Name	Stadt	PLZ
Lehmann	Berlin	99011
Meiersche	Aachen	42100
Amazon	Köln	52100

Angebote	
Titel	Haendler
Harry Potter	Lehmann
Harry Potter	Meiersche
Harry Potter	Amazon
Datenbanken	Amazon
Glücksformel	Amazon
Glücksformel	Lehmann

### Anfrage

„Welche Titel sind bei *allen* Buchhändlern im Angebot?“

### Relationenalgebra

$(\rho_{\text{Name} \leftarrow \text{Haendler}}(\text{Angebote})) \div (\pi_{\text{Name}}(\text{Buchhaendler}))$

### Tupelkalkül

$\{(t_1.\text{Titel}) \mid \text{Buecher}(t_1) \wedge$   
 $\forall t_2(\neg \text{Buchhaendler}(t_2) \vee \text{oder: } \forall t_2(\text{Buchhaendler}(t_2) \rightarrow$   
 $\exists t_3(\text{Angebote}(t_3) \wedge t_3.\text{Haendler} = t_2.\text{Name} \wedge t_3.\text{Titel} = t_1.\text{Titel}))\}$

# Relationaler Tupelkalkül

## Beispiel 4 [Domänenkalkül]

Buecher	
Titel	Verlag
Harry Potter	Princeton
Heuristics	Addison
Glücksformel	dpunkt
Datenbanken	Springer

Buchhaendler		
Name	Stadt	PLZ
Lehmann	Berlin	99011
Meiersche	Aachen	42100
Amazon	Köln	52100

Angebote	
Titel	Haendler
Harry Potter	Lehmann
Harry Potter	Meiersche
Harry Potter	Amazon
Datenbanken	Amazon
Glücksformel	Amazon
Glücksformel	Lehmann

### Anfrage

„Welche Titel sind bei *allen* Buchhändlern im Angebot?“

### Relationenalgebra

$(\rho_{\text{Name} \leftarrow \text{Haendler}}(\text{Angebote})) \div (\pi_{\text{Name}}(\text{Buchhaendler}))$

### Tupelkalkül

$\{(t_1.\text{Titel}) \mid \text{Buecher}(t_1) \wedge$   
 $\forall t_2(\neg \text{Buchhaendler}(t_2) \vee$   
 $\exists t_3(\text{Angebote}(t_3) \wedge t_3.\text{Haendler} = t_2.\text{Name} \wedge t_3.\text{Titel} = t_1.\text{Titel}))\}$

# Relationaler Tupelkalkül

## Beispiel 4 [Domänenkalkül]

Buecher	
Titel	Verlag
Harry Potter	Princeton
Heuristics	Addison
Glücksformel	dpunkt
Datenbanken	Springer

Buchhaendler		
Name	Stadt	PLZ
Lehmann	Berlin	99011
Meiersche	Aachen	42100
Amazon	Köln	52100

Angebote	
Titel	Haendler
Harry Potter	Lehmann
Harry Potter	Meiersche
Harry Potter	Amazon
Datenbanken	Amazon
Glücksformel	Amazon
Glücksformel	Lehmann

### Anfrage

„Welche Titel sind bei *allen* Buchhändlern im Angebot?“

### Relationenalgebra

$(\rho_{\text{Name} \leftarrow \text{Haendler}}(\text{Angebote})) \div (\pi_{\text{Name}}(\text{Buchhaendler}))$

### Tupelkalkül

$\{(t_1.\text{Titel}) \mid \text{Buecher}(t_1) \wedge$   
 $\forall t_2(\neg \text{Buchhaendler}(t_2) \vee$   
 $\exists t_3(\text{Angebote}(t_3) \wedge t_3.\text{Haendler} = t_2.\text{Name} \wedge t_3.\text{Titel} = t_1.\text{Titel}))\}$

## Bemerkungen:

- Interpretation des  $\forall$ -Quantors in der folgenden Formel:  $\alpha = \text{Buecher}(t_1) \wedge \forall t_2 \beta$

Semantik: „ $\alpha$  ist erfüllt für diejenigen Buecher(tupel  $t_1$ ), bei denen für *alle* Tupel  $t_2$  die Teilformel  $\beta$  erfüllt ist.“

Beachte, dass  $t_2$  an *alle* Tupel des Universums bzw. des Datenbankzustandes  $d(\mathcal{R})$  gebunden wird (also nicht nur an die Tupel von Buchhaendler) und bzgl. *aller* möglichen Instanziierungen die Formel  $\beta$  erfüllen muss.

## Bemerkungen:

- Bei (formalen, logischen, natürlichen) Sprachen unterscheidet man zwischen Sätzen aus der Sprache selbst und der Formulierung von Zusammenhängen *über* solche Sätze. Sätze aus der Sprache selbst dienen uns zur Kommunikation mittels dieser Sprache; die Symbole, die verwendet werden, um solche Sätze zu formulieren, gehören zur Objektsprache.

Symbole, die verwendet werden, um *über* Sätze zu sprechen, die in der Objektsprache formuliert sind, gehören zur Metasprache.

- Die Formelbezeichner  $\alpha, \beta, \gamma$ , die Prädikatsbezeichner  $P, Q$ , die Quantoren  $\forall, \exists$ , die Variablenbezeichner  $t, x, y, z$ , und die Junktoren  $\neg, \wedge, \vee, \rightarrow$  gehören zur Objektsprache. Das  $\approx$ -Zeichen ist ein Zeichen der Metasprache und steht für „ist logisch äquivalent mit“.

Es gelten u.a. folgende Äquivalenzen:

$$\neg(\alpha \vee \beta) \approx \neg\alpha \wedge \neg\beta \quad (\text{deMorgan})$$

$$\neg(\alpha \wedge \beta) \approx \neg\alpha \vee \neg\beta$$

$$\alpha \rightarrow \beta \approx \neg\alpha \vee \beta \quad (\text{Implikation})$$

$$(\alpha \wedge \beta) \rightarrow \gamma \approx \neg\alpha \vee \neg\beta \vee \gamma$$

$$\forall x P(x) \approx \neg \exists x (\neg P(x)) \quad (\text{Quantoren})$$

$$\neg \forall x P(x) \approx \exists x (\neg P(x))$$

$$\forall x (\neg P(x)) \approx \neg \exists x P(x)$$

$$\neg \forall x (\neg P(x)) \approx \exists x P(x)$$



# Relationaler Tupelkalkül

## Sichere Anfragen

Unter (semantisch) sicheren Anfragen versteht man Formeln eines Anfragekalküls, die für jeden Datenbankzustand  $d(\mathcal{R}) = \{r_1, \dots, r_p\}$  nur für eine endliche Menge von Variableninstanziierungen erfüllbar sind.

Beispiel für eine nicht-sichere Anfrage:

$\neg r(x)$  : „Alle Instanzen von  $x$  (im Universum), die nicht in  $r \in d(\mathcal{R})$  sind.“

Durch die Forderung bestimmter syntaktischer Einschränkungen kann man die semantische Sicherheit für eine Teilmenge der semantisch sicheren Anfragen auf einfache Art bestimmen.

→ Domäne einer Formel  $\alpha$

# Relationaler Tupelkalkül

## Sichere Anfragen (Fortsetzung)

### Definition 4 (Domäne einer Formel $\alpha$ )

Der Bereich bzw. die Domäne einer Formel  $\alpha$  ist die Menge aller Konstanten in  $\alpha$  vereinigt mit der Menge aller Attributwerte der Relationen  $r$ ,  $r \in \alpha$ .

Beispiel:

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Domäne der Formel „ $\neg$  Mitarbeiter( $t$ )“ :

{Smith, Wong, Zelaya, Weimar, Köln, Erfurt, 4, 5, 1234, 3334, 8886, 9876, 9998}

# Relationaler Tupelkalkül

## Sichere Anfragen (Fortsetzung) [Domänenkalkül]

Folgende Anfrage liefert eine unendliche Zahl von Ergebnissen:

$$\{t \mid \neg \text{Mitarbeiter}(t)\}$$

Eine Anfrage  $\{t \mid \alpha\}$  des Tupelkalküls ist sicher, falls folgende Bedingungen als erfüllt angenommen werden:

1. Gehört  $t = (c_1, c_2, \dots, c_n)$  in das Anfrageergebnis, so wird gefordert, dass  $\{c_1, c_2, \dots, c_n\}$  Teilmenge der Domäne von  $\alpha$  ist.  
 $\Rightarrow$  Die Suche zur Beantwortung der Anfrage ist auf die Domäne beschränkt.
2. Für jede Teilformel  $\exists t\beta$  wird gefordert, dass  $\beta$  höchstens für Elemente aus der Domäne von  $\beta$  erfüllbar sein kann.  
 $\Rightarrow \beta$  ist für alle Elemente, die nicht in seiner Domäne sind, unerfüllbar.
3. Für jede Teilformel  $\forall t\beta$  wird gefordert, dass  $\forall t\beta$  dann und nur dann erfüllt ist, wenn  $\beta$  für alle Elemente aus der Domäne von  $\beta$  erfüllt ist.  
 $\Rightarrow \beta$  ist für alle Elemente, die nicht in seiner Domäne sind, immer erfüllt.

# Relationaler Tupelkalkül

## Sichere Anfragen (Fortsetzung) [Domänenkalkül]

Folgende Anfrage liefert eine unendliche Zahl von Ergebnissen:

$$\{t \mid \neg \text{Mitarbeiter}(t)\}$$

Eine Anfrage  $\{t \mid \alpha\}$  des Tupelkalküls ist sicher, falls folgende Bedingungen als erfüllt angenommen werden:

1. Gehört  $t = (c_1, c_2, \dots, c_n)$  in das Anfrageergebnis, so wird gefordert, dass  $\{c_1, c_2, \dots, c_n\}$  Teilmenge der Domäne von  $\alpha$  ist.  
 $\Rightarrow$  Die Suche zur Beantwortung der Anfrage ist auf die Domäne beschränkt.
2. Für jede Teilformel  $\exists t\beta$  wird gefordert, dass  $\beta$  höchstens für Elemente aus der Domäne von  $\beta$  erfüllbar sein kann.  
 $\Rightarrow \beta$  ist für alle Elemente, die nicht in seiner Domäne sind, unerfüllbar.
3. Für jede Teilformel  $\forall t\beta$  wird gefordert, dass  $\forall t\beta$  dann und nur dann erfüllt ist, wenn  $\beta$  für alle Elemente aus der Domäne von  $\beta$  erfüllt ist.  
 $\Rightarrow \beta$  ist für alle Elemente, die nicht in seiner Domäne sind, immer erfüllt.

# Relationaler Tupelkalkül

## Sichere Anfragen (Fortsetzung) [Domänenkalkül]

Folgende Anfrage liefert eine unendliche Zahl von Ergebnissen:

$$\{t \mid \neg \text{Mitarbeiter}(t)\}$$

Eine Anfrage  $\{t \mid \alpha\}$  des Tupelkalküls ist sicher, falls folgende Bedingungen als erfüllt angenommen werden:

1. Gehört  $t = (c_1, c_2, \dots, c_n)$  in das Anfrageergebnis, so wird gefordert, dass  $\{c_1, c_2, \dots, c_n\}$  Teilmenge der Domäne von  $\alpha$  ist.  
 $\Rightarrow$  Die Suche zur Beantwortung der Anfrage ist auf die Domäne beschränkt.
2. Für jede Teilformel  $\exists t\beta$  wird gefordert, dass  $\beta$  höchstens für Elemente aus der Domäne von  $\beta$  erfüllbar sein kann.  
 $\Rightarrow \beta$  ist für alle Elemente, die nicht in seiner Domäne sind, unerfüllbar.
3. Für jede Teilformel  $\forall t\beta$  wird gefordert, dass  $\forall t\beta$  dann und nur dann erfüllt ist, wenn  $\beta$  für alle Elemente aus der Domäne von  $\beta$  erfüllt ist.  
 $\Rightarrow \beta$  ist für alle Elemente, die nicht in seiner Domäne sind, immer erfüllt.

# Relationaler Tupelkalkül

## Sichere Anfragen (Fortsetzung) [Domänenkalkül]

Folgende Anfrage liefert eine unendliche Zahl von Ergebnissen:

$$\{t \mid \neg \text{Mitarbeiter}(t)\}$$

Eine Anfrage  $\{t \mid \alpha\}$  des Tupelkalküls ist sicher, falls folgende Bedingungen als erfüllt angenommen werden:

1. Gehört  $t = (c_1, c_2, \dots, c_n)$  in das Anfrageergebnis, so wird gefordert, dass  $\{c_1, c_2, \dots, c_n\}$  Teilmenge der Domäne von  $\alpha$  ist.  
 $\Rightarrow$  Die Suche zur Beantwortung der Anfrage ist auf die Domäne beschränkt.
2. Für jede Teilformel  $\exists t \beta$  wird gefordert, dass  $\beta$  höchstens für Elemente aus der Domäne von  $\beta$  erfüllbar sein kann.  
 $\Rightarrow \beta$  ist für alle Elemente, die nicht in seiner Domäne sind, unerfüllbar.
3. Für jede Teilformel  $\forall t \beta$  wird gefordert, dass  $\forall t \beta$  dann und nur dann erfüllt ist, wenn  $\beta$  für alle Elemente aus der Domäne von  $\beta$  erfüllt ist.  
 $\Rightarrow \beta$  ist für alle Elemente, die nicht in seiner Domäne sind, immer erfüllt.

## Bemerkungen:

- ❑ Semantische Sicherheit ist eine Eigenschaft, die im Einzelfall leicht zu zeigen sein kann, die aber in der Allgemeinheit nicht automatisch nachprüfbar ist. Die Ursache dafür liegt in der Unentscheidbarkeit der Prädikatenlogik erster Stufe mit Arithmetik.
- ❑ Mit Hilfe der Domäne wird die semantische Analyse (= die Analyse der Erfüllbarkeit) der Formel  $\alpha$  deshalb stark vereinfacht, weil *vereinbart* wird, dass Instanziierungen der Variablen mit Werten von außerhalb der Domäne keinen Einfluss auf die Erfüllbarkeit von  $\alpha$  haben. Diese Vereinbarung entspricht einer „Closed World Assumption“. Da die Domäne endlich ist und sich syntaktisch einfach konstruieren lässt, ist unter der Closed World Assumption semantische Sicherheit in endlicher Zeit überprüfbar.
- ❑ Sobald ein (Datenbank-)System annimmt bzw. voraussetzt, dass die drei genannten Bedingungen erfüllt sind, interpretiert es eine Formel aus Sicht der Closed World Assumption. Die Bedingungen verhindern, dass unendlich viele Variableninstanziierungen evaluiert werden müssen, um die Erfüllbarkeit einer Formel zu analysieren. Beachte, dass auch Anfragen, die ein endliches Ergebnis liefern, die Evaluierung unendlich vieler Variableninstanziierungen erfordern können. Bei solchen Anfragen liegt die „Unendlichkeit“ in der Zeit – nicht in der Größe der Ergebnismenge.

# Kapitel DB:IV

## IV. Grundlagen relationaler Anfragesprachen

- Anfragen und Änderungen
- Relationale Algebra
- Anfragekalküle
- Relationaler Tupelkalkül
- **Relationaler Domänenkalkül**



# Relationaler Domänenkalkül

## Konzepte [\[Tupelkalkül\]](#)

1. Domänenvariablen, die sich auf **Attribute**  $A$  in den Relationenschemata  $\mathcal{R} \in \mathcal{R}$  beziehen und mit jedem Wert aus dem Wertebereich  $dom(A)$  von  $A$  instanziiert werden können.
2. Formeln, mit denen sich auf Basis der Domänenvariablen Zusammenhänge zwischen Attributen formulieren lassen.

## Bemerkungen:

- ❑ SQL, *Structured Query Language*, basiert auf dem relationalen Tupelkalkül und wurde von IBM Research, San Jose, Kalifornien, entwickelt.
- ❑ QBE, *Query By Example*, basiert auf dem relationalen Domänenkalkül und wurde von IBM Research, Yorktown Heights, New York, entwickelt. Diese Entwicklung fand fast zeitgleich mit der Entwicklung von SQL in San Jose statt.
- ❑ QBE war eine der ersten graphischen Anfragesprachen für Datenbanksysteme und ist bei IBM als Interface-Option für DB2 erhältlich.

# Relationaler Domänenkalkül

Anfragen [\[Tupelkalkül\]](#)

Anfrage im relationalen Domänenkalkül mit Variablen  $x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m}$  :

$$\{(x_1, x_2, \dots, x_n) \mid \alpha\}$$

- $x_1, \dots, x_n$  sind freie,  $x_{n+1}, \dots, x_{n+m}$  sind gebundene Domänenvariablen.
- $\alpha$  ist eine logische Formel, wobei die Menge der Atome  $\Sigma$ , aus denen  $\alpha$  besteht, wie folgt definiert ist:

# Relationaler Domänenkalkül

## Anfragen [Tupelkalkül]

Anfrage im relationalen Domänenkalkül mit Variablen  $x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m}$  :

$$\{(x_1, x_2, \dots, x_n) \mid \alpha\}$$

- $x_1, \dots, x_n$  sind freie,  $x_{n+1}, \dots, x_{n+m}$  sind gebundene Domänenvariablen.
- $\alpha$  ist eine logische Formel, wobei die Menge der Atome  $\Sigma$ , aus denen  $\alpha$  besteht, wie folgt definiert ist:
  1. „ $r(x_{r_1}, x_{r_2}, \dots, x_{r_k})$ “ alternativ: „ $(x_{r_1}, x_{r_2}, \dots, x_{r_k}) \in r$ “  
ist ein Atom, wobei die  $x_{r_i}$  Domänenvariablen und  $r$  eine Relation über  $k$  Attribute bezeichnet.  $r(x_{r_1}, x_{r_2}, \dots, x_{r_k})$  ist wahr für eine Instanziierung von  $(x_{r_1}, x_{r_2}, \dots, x_{r_k})$ , falls diese Instanziierung ein Tupel in  $r$  ist.

# Relationaler Domänenkalkül

## Anfragen [\[Tupelkalkül\]](#)

Anfrage im relationalen Domänenkalkül mit Variablen  $x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m}$  :

$$\{(x_1, x_2, \dots, x_n) \mid \alpha\}$$

- $x_1, \dots, x_n$  sind freie,  $x_{n+1}, \dots, x_{n+m}$  sind gebundene Domänenvariablen.
- $\alpha$  ist eine logische Formel, wobei die Menge der Atome  $\Sigma$ , aus denen  $\alpha$  besteht, wie folgt definiert ist:
  1. „ $r(x_{r_1}, x_{r_2}, \dots, x_{r_k})$ “ alternativ: „ $(x_{r_1}, x_{r_2}, \dots, x_{r_k}) \in r$ “  
ist ein Atom, wobei die  $x_{r_i}$  Domänenvariablen und  $r$  eine Relation über  $k$  Attribute bezeichnet.  $r(x_{r_1}, x_{r_2}, \dots, x_{r_k})$  ist wahr für eine Instanziierung von  $(x_{r_1}, x_{r_2}, \dots, x_{r_k})$ , falls diese Instanziierung ein Tupel in  $r$  ist.
  2. „ $x_i \text{ op } x_j$ “  
ist ein Atom mit  $\text{op} \in \{=, <, \leq, >, \geq, \neq\}$ .  $x_i, x_j$  bezeichnen Domänenvariablen, die über den Wertebereichen der zugeordneten Attribute instanziiert sind.
  3. „ $x_i \text{ op } c$ “  
ist ein Atom mit  $\text{op} \in \{=, <, \leq, >, \geq, \neq\}$ .  $x_i$  bezeichnet eine Domänenvariable, die über dem Wertebereich des zugeordneten Attributes instanziiert ist, und  $c \in \text{dom}(A_i)$  ist eine Konstante aus dem gleichen Wertebereich.

# Relationaler Domänenkalkül

## Beispiel 1 [Tupelkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

### Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

# Relationaler Domänenkalkül

## Beispiel 1 [Tupelkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

## Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

## Relationenalgebra

$\pi_{\text{Name, Wohnort}}(\text{Mitarbeiter} \bowtie_{\text{AbtNr}=\text{Nr}} (\sigma_{\text{AbtName}='Forschung'}(\text{Abteilung})))$

# Relationaler Domänenkalkül

## Beispiel 1 [Tupelkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

## Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

## Relationenalgebra

$\pi_{\text{Name, Wohnort}}(\text{Mitarbeiter} \bowtie_{\text{AbtNr}=\text{Nr}} (\sigma_{\text{AbtName}='Forschung'}(\text{Abteilung})))$

## Domänenkalkül

$\{(x_1, x_3) \mid \exists x_2 \exists x_4 \exists x_5 \exists y_1 \exists y_2 \exists y_3$   
 $(\text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \wedge \text{Abteilung}(y_1, y_2, y_3) \wedge y_1 = 'Forschung' \wedge y_2 = x_5)\}$



# Relationaler Domänenkalkül

## Beispiel 1 [Tupelkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

## Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

## Relationenalgebra

$\pi_{\text{Name, Wohnort}}(\text{Mitarbeiter} \bowtie_{\text{AbtNr}=\text{Nr}} (\sigma_{\text{AbtName}='Forschung'}(\text{Abteilung})))$

**Domänenkalkül** Schreibkonvention: nicht-freie Variablen sind per Default  $\exists$ -quantifiziert.

$\{(x_1, x_3) \mid \exists x_5 \exists y_1 \exists y_2$

$(\text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \wedge \text{Abteilung}(y_1, y_2, y_3) \wedge y_1 = \text{'Forschung'} \wedge y_2 = x_5)\}$

# Relationaler Domänenkalkül

## Beispiel 1 [Tupelkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

## Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

## Relationenalgebra

$\pi_{\text{Name, Wohnort}}(\text{Mitarbeiter} \bowtie_{\text{AbtNr}=\text{Nr}} (\sigma_{\text{AbtName}='Forschung'}(\text{Abteilung})))$

**Domänenkalkül** Abkürzung: Konstanten als Parameter.

$\{(x_1, x_3) \mid \exists x_5 \exists y_2$   
 $(\text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \wedge \text{Abteilung}('Forschung', y_2, y_3) \wedge y_2 = x_5)\}$

# Relationaler Domänenkalkül

## Beispiel 1 [Tupelkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

## Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

## Relationenalgebra

$\pi_{\text{Name, Wohnort}}(\text{Mitarbeiter} \bowtie_{\text{AbtNr}=\text{Nr}} (\sigma_{\text{AbtName}='Forschung'}(\text{Abteilung})))$

**Domänenkalkül** Abkürzung: Unifikation von Domänenvariablen.

$\{(x_1, x_3) \mid$   
 $(\text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \wedge \text{Abteilung}('Forschung', x_5, y_3))\}$

## Bemerkungen: [Tupelkalkül]

- Die freien Domänenvariablen (vor dem Trennsymbol „|“ bei der Mengenbildung) entsprechen der Projektion  $\pi$  in der relationalen Algebra.

Beispiel:  $\{(x_1, x_3) \mid \dots\}$

- Eine Bedingung, die sich auf eine Domänenvariable und eine Konstante bezieht, entspricht einer Selektion,  $\sigma$ , in der relationalen Algebra.

Beispiel:  $y_1 = \text{'Forschung'}$

- Eine Bedingung bzgl. zweier Domänenvariablen, die sich auf zwei verschiedene Relationen beziehen, entspricht einem Verbund (Join),  $\bowtie$ , in der relationalen Algebra.

Beispiel:  $x_5 = y_2$

- Schreibkonvention: Die Notation des  $\exists$ -Quantors in der Formel  $\alpha$  ist nur für diejenigen nicht-freien Variablen notwendig, die in einer Bedingung auftauchen. D.h., für diejenigen nicht-freien Variablen in  $\alpha$ , die ohne Existenzquantor notiert sind (obwohl sie der Exaktheit wegen müssten), wird implizit der Existenzquantor angenommen.

# Relationaler Domänenkalkül

## Beispiel 2 [Tupelkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

### Anfrage

„Liefere für jedes Projekt in Weimar dessen Nummer, die Nummer der durchführenden Abteilung sowie Name und Wohnort des Abteilungsmanagers.“

# Relationaler Domänenkalkül

## Beispiel 2 [Tupelkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

### Anfrage

„Liefere für jedes Projekt in Weimar dessen Nummer, die Nummer der durchführenden Abteilung sowie Name und Wohnort des Abteilungsmanagers.“

### Relationenalgebra

$$\pi_{\text{PNr}, \text{AbtNr}, \text{Name}, \text{Wohnort}} \left( \left( \rho_{\text{PNr} \leftarrow \text{Nr}, \text{PName} \leftarrow \text{Name}} \left( \sigma_{\text{Ort} = \text{'Weimar'}} (\text{Projekt}) \right) \right) \bowtie_{\text{AbtNr} = \text{Nr}} \text{Abteilung} \right) \bowtie_{\text{Manager} = \text{PersNr}} \text{Mitarbeiter}$$

# Relationaler Domänenkalkül

## Beispiel 2 [Tupelkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

### Anfrage

„Liefere für jedes Projekt in Weimar dessen Nummer, die Nummer der durchführenden Abteilung sowie Name und Wohnort des Abteilungsmanagers.“

### Relationenalgebra

$$\pi_{\text{PNr, AbtNr, Name, Wohnort}} \left( \left( \left( \rho_{\text{PNr} \leftarrow \text{Nr}, \text{PName} \leftarrow \text{Name}} \left( \sigma_{\text{Ort} = \text{'Weimar'}} (\text{Projekt}) \right) \right) \bowtie_{\text{AbtNr} = \text{Nr}} \text{Abteilung} \right) \bowtie_{\text{Manager} = \text{PersNr}} \text{Mitarbeiter} \right)$$

### Domänenkalkül

$$\{(x_1, x_3, z_2, z_4) \mid \exists x_2 \exists x_4 \exists x_5 \exists y_1 \exists y_2 \exists y_3 \exists z_1 \exists z_3 \\ (\text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \wedge \text{Abteilung}(y_1, y_2, y_3) \wedge \text{Projekt}(z_1, z_2, z_3, z_4) \wedge \\ z_3 = \text{'Weimar'} \wedge z_4 = y_2 \wedge y_3 = x_2)\}$$

# Relationaler Domänenkalkül

## Beispiel 3 [Tupelkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

### Anfrage

„Liefere die Namen der Mitarbeiter, die in *allen* Projekten der Abteilung 5 arbeiten.“



# Relationaler Domänenkalkül

## Beispiel 3 [Tupelkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

### Anfrage

„Liefere die Namen der Mitarbeiter, die in *allen* Projekten der Abteilung 5 arbeiten.“

### Relationenalgebra

$\pi_{\text{Name}} \left( \left( \text{ArbeitetInProjekt} \div \rho_{\text{ProjektNr} \leftarrow \text{Nr}} \left( \pi_{\text{Nr}} \left( \sigma_{\text{AbtNr}='5'} \left( \text{Projekt} \right) \right) \right) \right) \bowtie \text{Mitarbeiter} \right)$

# Relationaler Domänenkalkül

## Beispiel 3 [Tupelkalkül]

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

## Anfrage

„Liefere die Namen der Mitarbeiter, die in *allen* Projekten der Abteilung 5 arbeiten.“

## Relationenalgebra

$\pi_{\text{Name}} ((\text{ArbeitetInProjekt} \div \rho_{\text{ProjektNr} \leftarrow \text{Nr}}(\pi_{\text{Nr}}(\sigma_{\text{AbtNr} = '5'}(\text{Projekt})))) \bowtie \text{Mitarbeiter})$

## Domänenkalkül

$\{(x_1) \mid \exists x_2 \exists x_3 \exists x_4 \exists x_5$   
 $(\text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \wedge \forall z_1 \forall z_2 \forall z_3 \forall z_4$   
 $(\neg \text{Projekt}(z_1, z_2, z_3, z_4) \vee \neg(z_4 = 5) \vee$  oder:  $((\text{Projekt}(z_1, z_2, z_3, z_4) \wedge (z_4 = 5)) \rightarrow$   
 $\exists y_1 \exists y_2 (\text{ArbeitetInProjekt}(y_1, y_2) \wedge y_2 = z_2 \wedge y_1 = x_2))))\}$

## Bemerkungen:

- **Wiederholung.** Die Quantoren können verschoben werden, solange sich keine Variablenbindungen ändern und die Ordnung zwischen den  $\exists$ - und  $\forall$ -Quantoren erhalten bleibt:

$$\{(x_1) \mid \exists x_2 \exists x_3 \exists x_4 \exists x_5 \forall z_1 \forall z_2 \forall z_3 \forall z_4 \exists y_1 \exists y_2 \\ (\text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \wedge (\neg \text{Projekt}(z_1, z_2, z_3, z_4) \vee \neg(z_4 = 5) \vee \\ (\text{ArbeitetInProjekt}(y_1, y_2) \wedge y_2 = z_2 \wedge y_1 = x_2)))\}$$

- Beispiele für alternative Formeln, die denselben Sachverhalt „Für alle Projekte der Abteilung 5 gilt ...“ modellieren, also logisch äquivalent sind:

1.  $\forall z_1 \forall z_2 \forall z_3 \forall z_4 ((\text{Projekt}(z_1, z_2, z_3, z_4) \wedge (z_4 = 5)) \rightarrow \dots$
2.  $\forall z_1 \forall z_2 \forall z_3 \forall z_4 (\neg(\text{Projekt}(z_1, z_2, z_3, z_4) \wedge z_4 = 5) \vee \dots$
3.  $\forall z_1 \forall z_2 \forall z_3 \forall z_4 (\neg \text{Projekt}(z_1, z_2, z_3, z_4) \vee \neg(z_4 = 5) \vee \dots$
4.  $\forall z_1 \forall z_2 \forall z_3 (\text{Projekt}(z_1, z_2, z_3, 5) \rightarrow \dots$
5.  $\forall z_1 \forall z_2 \forall z_3 (\neg \text{Projekt}(z_1, z_2, z_3, 5) \vee \dots$

# Relationaler Domänenkalkül

## Beispiel 4 [Tupelkalkül]

Buecher	
Titel	Verlag
Harry Potter	Princeton
Heuristics	Addison
Glücksformel	dpunkt
Datenbanken	Springer

Buchhaendler		
Name	Stadt	PLZ
Lehmann	Berlin	99011
Meiersche	Aachen	42100
Amazon	Köln	52100

Angebote	
Titel	Haendler
Harry Potter	Lehmann
Harry Potter	Meiersche
Harry Potter	Amazon
Datenbanken	Amazon
Glücksformel	Amazon
Glücksformel	Lehmann

### Anfrage

„Welche Titel sind bei *allen* Buchhändlern im Angebot?“

# Relationaler Domänenkalkül

## Beispiel 4 [Tupelkalkül]

Buecher	
Titel	Verlag
Harry Potter	Princeton
Heuristics	Addison
Glücksformel	dpunkt
Datenbanken	Springer

Buchhaendler		
Name	Stadt	PLZ
Lehmann	Berlin	99011
Meiersche	Aachen	42100
Amazon	Köln	52100

Angebote	
Titel	Haendler
Harry Potter	Lehmann
Harry Potter	Meiersche
Harry Potter	Amazon
Datenbanken	Amazon
Glücksformel	Amazon
Glücksformel	Lehmann

### Anfrage

„Welche Titel sind bei *allen* Buchhändlern im Angebot?“

### Relationenalgebra

$(\rho_{\text{Name} \leftarrow \text{Haendler}}(\text{Angebote})) \div (\pi_{\text{Name}}(\text{Buchhaendler}))$

# Relationaler Domänenkalkül

## Beispiel 4 [Tupelkalkül]

Buecher	
Titel	Verlag
Harry Potter	Princeton
Heuristics	Addison
Glücksformel	dpunkt
Datenbanken	Springer

Buchhaendler		
Name	Stadt	PLZ
Lehmann	Berlin	99011
Meiersche	Aachen	42100
Amazon	Köln	52100

Angebote	
Titel	Haendler
Harry Potter	Lehmann
Harry Potter	Meiersche
Harry Potter	Amazon
Datenbanken	Amazon
Glücksformel	Amazon
Glücksformel	Lehmann

### Anfrage

„Welche Titel sind bei *allen* Buchhändlern im Angebot?“

### Relationenalgebra

$(\rho_{\text{Name} \leftarrow \text{Haendler}}(\text{Angebote})) \div (\pi_{\text{Name}}(\text{Buchhaendler}))$

### Domänenkalkül

$\{(x_1) \mid \exists x_2(\text{Buecher}(x_1, x_2) \wedge$   
 $\forall y_1 \forall y_2 \forall y_3 (\neg \text{Buchhaendler}(y_1, y_2, y_3) \vee$  oder:  $\forall y_1 \forall y_2 \forall y_3 (\text{Buchhaendler}(y_1, y_2, y_3) \rightarrow$   
 $\exists z_1 \exists z_2 (\text{Angebote}(z_1, z_2) \wedge z_2 = y_1 \wedge z_1 = x_1))\})\}$

# Relationaler Domänenkalkül

## Sichere Anfragen [\[Tupelkalkül\]](#)

Folgende Anfrage liefert eine unendliche Zahl von Ergebnissen:

$$\{(x_1, x_2, x_3, x_4, x_5) \mid \neg \text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5)\}$$

# Relationaler Domänenkalkül

## Sichere Anfragen [Tupelkalkül]

Folgende Anfrage liefert eine unendliche Zahl von Ergebnissen:

$$\{(x_1, x_2, x_3, x_4, x_5) \mid \neg \text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5)\}$$

Sei die Domäne einer Formel  $\alpha$  wie zuvor definiert. Dann ist eine Anfrage  $\{(x_1, x_2, \dots, x_n) \mid \alpha\}$  des Domänenkalküls sicher, falls folgende Bedingungen als erfüllt angenommen werden:

1. Gehört  $(c_1, c_2, \dots, c_n)$  in das Anfrageergebnis, so muss  $\{c_1, c_2, \dots, c_n\}$  Teilmenge der Domäne von  $\alpha$  sein.  
 $\Rightarrow$  Die Suche zur Beantwortung der Anfrage ist auf die Domäne beschränkt.
2. Für jede Teilformel  $\exists x\beta$  muss gelten, dass  $\beta$  höchstens für Elemente aus seiner Domäne erfüllbar sein kann.  
 $\Rightarrow \beta$  ist für alle Elemente, die nicht in seiner Domäne sind, unerfüllbar.
3. Für jede Teilformel  $\forall x\beta$  muss gelten, dass  $\forall x\beta$  dann und nur dann erfüllt ist, wenn  $\beta$  für alle Elemente aus seiner Domäne erfüllt ist.  
 $\Rightarrow \beta$  ist für alle Elemente, die nicht in seiner Domäne sind, immer erfüllt.



# Relationaler Domänenkalkül

## Ausdrucksstärke der Kalküle

Folgende drei Sprachen besitzen die gleiche Ausdruckskraft:

1. die relationale Algebra
2. der relationale Tupelkalkül, eingeschränkt auf sichere Anfragen
3. der relationale Domänenkalkül, eingeschränkt auf sichere Anfragen

## Bemerkungen:

- ❑ Der Beweis erfolgt induktiv über den Aufbau der Ausdrücke in der jeweiligen Sprache. Unter anderem spezifiziert man äquivalente Ausdrücke des Tupelkalküls zu den Basisoperatoren der relationalen Algebra.
- ❑ Weil der (sichere) relationale Tupelkalkül und der (sichere) relationale Domänenkalkül die gleiche Ausdruckskraft wie die relationale Algebra besitzen, sind sie auch relational vollständig.
- ❑ Die Aussage, dass der relationale Tupelkalkül und der relationale Domänenkalkül relational vollständig sind, bedarf nicht der Einschränkung auf sichere Anfragen.