Kapitel DB:V (Fortsetzung)

V. Grundlagen relationaler Anfragesprachen

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

DB:V-70 Relational Algebra & Calculus © STEIN 2004-2018

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.

DB:V-71 Relational Algebra & Calculus © STEIN 2004-2018

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

DB:V-72 Relational Algebra & Calculus © STEIN 2004-2018

Einleitung (Fortsetzung)

Beispiele für Formeln:

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

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

Einleitung (Fortsetzung)

Beispiele für Formeln:

(a) Mitarbeiter(t) alternativ: $t \in Mitarbeiter$

Tupel *t* ist Element der Relation Mitarbeiter.

- (b) ¬Mitarbeiter((Smith, 1234, Weimar, 3334, 5))
- (c) $\exists t : Mitarbeiter(t) \land t.AbtNr = '5'$

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

DB:V-74 Relational Algebra & Calculus © STEIN 2004-2018

Einleitung (Fortsetzung)

Beispiele für Formeln:

(a) Mitarbeiter(t) alternativ: $t \in Mitarbeiter$

Tupel t ist Element der Relation Mitarbeiter.

(b) ¬Mitarbeiter((Smith, 1234, Weimar, 3334, 5))

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

(c) $\exists t : Mitarbeiter(t) \land t.AbtNr = '5'$

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

DB:V-75 Relational Algebra & Calculus © STEIN 2004-2018

Einleitung (Fortsetzung)

Beispiele für Formeln:

(a) Mitarbeiter(t) alternativ: $t \in Mitarbeiter$

(b) ¬Mitarbeiter((Smith, 1234, Weimar, 3334, 5))

(c) $\exists t : Mitarbeiter(t) \land t.AbtNr = '5'$

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

Tupel t ist Element der Relation Mitarbeiter.

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

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

DB:V-76 Relational Algebra & Calculus © STEIN 2004-2018

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) Mitarbeiter(t) alternativ: $t \in Mitarbeiter$

Tupel *t* ist Element der Relation Mitarbeiter.

(b) ¬Mitarbeiter((Smith, 1234, Weimar, 3334, 5))

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

(c) $\exists t : Mitarbeiter(t) \land t.AbtNr = '5'$

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

DB:V-77 Relational Algebra & Calculus © STEIN 2004-2018

Aufbau einer Formel / Grammatik der Sprache: Syntax I

Beispiele für Aussagen, Aussageformen, Atome:

- (a) Mitarbeiter(t) alternativ: $t \in Mitarbeiter$
- (b) ¬Mitarbeiter((Smith, 1234, Weimar, 3334, 5))
- (c) $\exists t : Mitarbeiter(t) \land t.AbtNr = '5'$

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

Aufbau einer Formel / Grammatik der Sprache: Syntax I

Beispiele für Aussagen, Aussageformen, Atome:

(a) Mitarbeiter(t) alternativ: $t \in Mitarbeiter$ Aussageform, Atom

- (b) ¬Mitarbeiter((Smith, 1234, Weimar, 3334, 5))
- (c) $\exists t : Mitarbeiter(t) \land t.AbtNr = '5'$

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

DB:V-79 Relational Algebra & Calculus © STEIN 2004-2018

Aufbau einer Formel / Grammatik der Sprache: Syntax I

Beispiele für Aussagen, Aussageformen, Atome:

(a) Mitarbeiter(t) alternativ: $t \in Mitarbeiter$

Aussageform, Atom

(b) ¬Mitarbeiter((Smith, 1234, Weimar, 3334, 5)) Aussage (falsch), keine atomare Formel

(c) $\exists t : Mitarbeiter(t) \land t.AbtNr = '5'$

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

DB:V-80 Relational Algebra & Calculus © STEIN 2004-2018

Aufbau einer Formel / Grammatik der Sprache: Syntax I

Beispiele für Aussagen, Aussageformen, Atome:

(a) Mitarbeiter(t) alternativ: $t \in Mitarbeiter$

Aussageform, Atom

(b) ¬Mitarbeiter((Smith, 1234, Weimar, 3334, 5))

Aussage (falsch), keine atomare Formel

(c) $\exists t : Mitarbeiter(t) \land t.AbtNr = '5'$

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

DB:V-81 Relational Algebra & Calculus © STEIN 2004-2018

Aufbau einer Formel / Grammatik der Sprache: Syntax I

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

- 1. Jedes Atom in Σ ist eine Formel.
- 2. Sind α und β Formeln, so sind es auch (α) , $\neg \alpha$, $\alpha \land \beta$, $\alpha \lor \beta$ und $\alpha \to \beta$.

Beispiele für Aussagen, Aussageformen, Atome:

(a) Mitarbeiter(t) alternativ: $t \in Mitarbeiter$

Aussageform, Atom

(b) ¬Mitarbeiter((Smith, 1234, Weimar, 3334, 5))

Aussage (falsch), keine atomare Formel

(c) $\exists t : Mitarbeiter(t) \land t.AbtNr = '5'$

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

DB:V-82 Relational Algebra & Calculus © STEIN 2004-2018

Bewertung einer Formel / Interpretation der Sprache: Semantik I

Beispiele für die Bewertung von Formeln:

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

(c)
$$\mathcal{I}(\exists t : \overbrace{\mathsf{Mitarbeiter}(t)}^{\mathsf{Atom}} \land \overbrace{t.\mathsf{AbtNr} =' 5'}^{\mathsf{Atom}}) = 1$$

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

DB:V-83 Relational Algebra & Calculus © STEIN 2004-2018

Bewertung einer Formel / Interpretation der Sprache: Semantik I

- 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 α ein Wahrheitswert $\mathcal{I}(\alpha)$ zuweisen.

Beispiele für die Bewertung von Formeln:

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

(c)
$$\mathcal{I}(\exists t : \overline{\mathsf{Mitarbeiter}(t)} \land \overline{t.\mathsf{AbtNr}} = 1) = 1$$

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

DB:V-84 Relational Algebra & Calculus © STEIN 2004-2018

Bemerkungen:

- □ Ein Atom ist die einfachste Formel. Eine Formel stellt entweder eine Aussage oder eine Aussage form 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: das Atom entspricht dem Test, ob ein bestimmtes Wertetupel t oder ein bestimmter Attributwert x ein Element einer Relation r im Datenbankzustand $d(\mathcal{R}) = \{r_1, \ldots, r_p\}$ 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) ¬, ∧, ∨, → wie folgt definiert:

- Die Interpretationsfunktion wird mit \mathcal{I} bezeichnet und liefert für eine Formel α , 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.

DB:V-85 Relational Algebra & Calculus © STEIN 2004-2018

Freie und gebundene Variablen

Sei α eine Formel, die eine Variable x enthält. Dann sei vereinbart:

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

Freie und gebundene Variablen

Sei α eine Formel, die eine Variable x enthält. Dann sei vereinbart:

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

DB:V-87 Relational Algebra & Calculus © STEIN 2004-2018

Freie und gebundene Variablen

Sei α eine Formel, die eine Variable x enthält. Dann sei vereinbart:

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

DB:V-88 Relational Algebra & Calculus © STEIN 2004-2018

Freie und gebundene Variablen

Sei α eine Formel, die eine Variable x enthält. Dann sei vereinbart:

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

DB:V-89 Relational Algebra & Calculus © STEIN 2004-2018

Freie und gebundene Variablen

Sei α eine Formel, die eine Variable x enthält. Dann sei vereinbart:

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

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

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

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

3. Eine Formel $\exists x\alpha$ ist wahr, falls α bzgl. *einer* Instanziierung von x wahr wird. Eine Formel $\forall x\alpha$ ist wahr, falls α 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.

DB:V-91 Relational Algebra & Calculus © STEIN 2004-2018

Auswertung einer Anfrage

Gegeben:

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

Auswertung einer Anfrage

Gegeben:

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

Beispiel:

Mitarbeiter				
Name PersNr Wohnort ChefPersNr Abt				
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

Auswertung einer Anfrage

Gegeben:

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

Beispiel:

Mitarbeiter				
Name PersNr Wohnort ChefPersNr Abt				
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

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

Tupelkalkül

Auswertung einer Anfrage

Gegeben:

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

Beispiel:

Mitarbeiter				
Name PersNr Wohnort ChefPersNr AbtN				
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

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

Auswertung einer Anfrage

Gegeben:

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

Beispiel:

Mitarbeiter				
Name PersNr Wohnort ChefPersNr Abt				
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

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

Auswertung einer Anfrage

Gegeben:

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

Beispiel:

Mitarbeiter				
Name PersNr Wohnort ChefPersNr Ab				
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

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

Auswertung einer Anfrage (Fortsetzung)

Gegeben:

- ullet Anfrage $\{ \cdot \mid \alpha \}$ mit freien Variablen im Tupelausdruck \cdot und Formel α
- ullet 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 <u>freien</u> Variablen (= die Variablen in \cdot) werden hinsichtlich aller Tupel (im Tupelkalkül) bzw. aller Attributwerte (im Domänenkalkül) für die in der Datenbank befindlichen Relationen $\{r_1, \ldots, r_p\}$ instanziiert.
 - Durch die Instanziierung wird die Aussageform α zu einer Aussage.
- 3. Für jede Instanziierung der freien Variablen wird geprüft, ob die Formel α wahr (erfüllt) ist. Falls ja, setze $res = res \cup \{\cdot\}$
- 4. res enthält keine weiteren Elemente.

Auswertung einer Anfrage (Fortsetzung)

Gegeben:

- ullet Anfrage $\{ \cdot \mid \alpha \}$ mit freien Variablen im Tupelausdruck \cdot und Formel α
- \Box 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 <u>freien</u> Variablen (= die Variablen in \cdot) werden hinsichtlich aller Tupel (im Tupelkalkül) bzw. aller Attributwerte (im Domänenkalkül) für die in der Datenbank befindlichen Relationen $\{r_1, \ldots, r_p\}$ instanziiert.

Durch die Instanziierung wird die Aussageform α zu einer Aussage.

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

Auswertung einer Anfrage (Fortsetzung)

Gegeben:

- ullet Anfrage $\{\ \cdot\ |\ lpha\}$ mit freien Variablen im Tupelausdruck $\ \cdot\$ und Formel lpha
- \Box 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 <u>freien</u> Variablen (= die Variablen in \cdot) werden hinsichtlich aller Tupel (im Tupelkalkül) bzw. aller Attributwerte (im Domänenkalkül) für die in der Datenbank befindlichen Relationen $\{r_1, \ldots, r_p\}$ instanziiert.

Durch die Instanziierung wird die Aussageform α zu einer Aussage.

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

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.

DB:V-101 Relational Algebra & Calculus © STEIN 2004-2018

Anfragen [Domänenkalkül]

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

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

- $\lnot t_1, \ldots, t_n$ sind freie, t_{n+1}, \ldots, t_{n+m} sind gebundene Tupelvariablen.
- $\neg A_1, \dots, A_n$ sind Attribute der Relationen bzgl. derer die t_i instanziiert sind.

DB:V-102 Relational Algebra & Calculus © STEIN 2004-2018

Anfragen [Domänenkalkül]

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

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

- $\lnot t_1, \ldots, t_n$ sind freie, t_{n+1}, \ldots, t_{n+m} sind gebundene Tupelvariablen.
- $\neg A_1, \dots, A_n$ sind Attribute der Relationen bzgl. derer die t_i instanziiert sind.
- 1. r(t) alternativ: $t \in r$ ist ein Atom, wobei t eine Tupelvariable und t eine Relation bezeichnet. r(t) ist wahr für eine Instanziierung von t, falls diese Instanziierung ein Tupel in t ist.

DB:V-103 Relational Algebra & Calculus © STEIN 2004-2018

Anfragen [Domänenkalkül]

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

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

- $\lnot t_1, \ldots, t_n$ sind freie, t_{n+1}, \ldots, t_{n+m} sind gebundene Tupelvariablen.
- $\neg A_1, \dots, A_n$ sind Attribute der Relationen bzgl. derer die t_i instanziiert sind.
- 1. r(t) alternativ: $t \in r$ ist ein Atom, wobei t eine Tupelvariable und t eine Relation bezeichnet. r(t) ist wahr für eine Instanziierung von t, falls diese Instanziierung ein Tupel in t ist.
- 2. $_{i}t_{i}.A_{i}$ op $t_{j}.A_{j}$ " ist ein Atom mit $op \in \{=, <, \leq, >, \geq, \neq\}$. 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. $_{i}t_{i}.A_{i}$ op c^{*} ist ein Atom mit $op \in \{=, <, \leq, >, \geq, \neq\}$. t_{i} bezeichnet eine Tupelvariable, A_{i} ein Attribut aus der Relation hinsichtlich der t_{i} instanziiert ist und $c \in dom(A_{i})$ ist eine Konstante.

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	
Χ	1	Köln	5	
Υ	2	Hamburg	5	
Z	3	Weimar	4	
New	8	Weimar	4	

Anfrage

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

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	
Χ	1	Köln	5	
Υ	2	Hamburg	5	
Z	3	Weimar	4	
New	8	Weimar	4	

Anfrage

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

Relationenalgebra

 \sim \mathcal{TAFEL}

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	
Χ	1	Köln	5	
Υ	2	Hamburg	5	
Z	3	Weimar	4	
New	8	Weimar	4	

Anfrage

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

Relationenalgebra

 \sim TAFEL

Tupelkalkül

```
 \{(t_1.\mathsf{Name}, t_1.\mathsf{Wohnort}) \mid \\ \mathsf{Mitarbeiter}(t_1) \ \land \ \exists t_2(\mathsf{Abteilung}(t_2) \ \land \ t_2.\mathsf{AbtName} = '\mathsf{Forschung}' \ \land \ t_2.\mathsf{Nr} = t_1.\mathsf{AbtNr}) \}
```

Bemerkungen:

Die Qualifizierung der freien Tupelvariablen vor dem Trennsymbol "|" bei der Mengenbildung entspricht der Projektion π in der relationalen Algebra.

Beispiel: $\{(t_1.\mathsf{Name},\ldots) \mid \ldots\}$

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

Beispiel: t_2 .AbtName =' Forschung'

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

Beispiel: $t_2.Nr = t_1.AbtNr$

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
Χ	1	Köln	5
Υ	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."

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
Χ	1	Köln	5
Υ	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

 $\pi_{\mathsf{PNr},\mathsf{AbtNr},}$ $((\rho_{\mathsf{PNr}\leftarrow\mathsf{Nr},}(\sigma_{\mathsf{Ort}='\mathsf{Weimar'}}(\mathsf{Projekt}))) \bowtie_{\mathsf{AbtNr}=\mathsf{Nr}} \mathsf{Abteilung} \bowtie_{\mathsf{Manager}=\mathsf{PersNr}} \mathsf{Mitarbeiter})$

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	
Χ	1	Köln	5	
Υ	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

 $\pi_{\mathsf{PNr},\mathsf{AbtNr},}$ $((\rho_{\mathsf{PNr}\leftarrow\mathsf{Nr},}(\sigma_{\mathsf{Ort}='\mathsf{Weimar'}}(\mathsf{Projekt}))) \bowtie_{\mathsf{AbtNr}=\mathsf{Nr}} \mathsf{Abteilung} \bowtie_{\mathsf{Manager}=\mathsf{PersNr}} \mathsf{Mitarbeiter})$

Tupelkalkül

 \sim TAFEL

Beispiel 3 [Domänenkalkül]

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

Abt	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	
Χ	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."

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	

Arbeite	ArbeitetInProjekt		
PersNr	ProjektNr		
1234	1		
1234	2		
6668	3		
4534	1		

Projekt				
Name Nr Ort			AbtNr	
Χ	1	Köln	5	
Υ	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

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

DB:V-113 Relational Algebra & Calculus © STEIN 2004-2018

Beispiel 3 [Domänenkalkül]

	Mitarbeiter				
Name PersNr		Wohnort	ChefPersNr	AbtNr	
Smith 1234		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
Χ	1	Köln	5
Υ	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

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

Tupelkalkül

 \sim TAFEL

Bemerkungen:

□ Interpretation des \forall -Quantors in der folgenden Formel: α = Mitarbeiter(t_1) $\wedge \forall t_3 \beta$

Semantik: " α ist erfüllt für diejenigen Mitarbeiter(tupel t_1), bei denen für *alle* Tupel t_3 die Teilformel β erfüllt ist."

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

□ Die Quantoren können verschoben werden, solange sich keine Variablenbindungen ändern und die Ordnung zwischen den ∃- und ∀-Quantoren erhalten bleibt:

```
 \{(t_1.\mathsf{Name}) \mid \mathsf{Mitarbeiter}(t_1) \land \\ \forall t_3 \exists t_4 (\neg(\mathsf{Projekt}(t_3) \land (t_3.\mathsf{AbtNr} =' 5')) \lor \\ (\mathsf{ArbeitetInProjekt}(t_4) \land t_4.\mathsf{ProjektNr} = t_3.\mathsf{Nr} \land t_4.\mathsf{PersNr} = t_1.\mathsf{PersNr})) \}
```

DB:V-115 Relational Algebra & Calculus © STEIN 2004-2018

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?"

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

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

DB:V-117 Relational Algebra & Calculus © STEIN 2004-2018

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_{\mathsf{Name}\leftarrow\mathsf{Haendler}}(\mathsf{Angebote})) \div (\pi_{\mathsf{Name}}(\mathsf{Buchhaendler}))
```

Tupelkalkül

```
 \begin{split} &\{(t_1.\mathsf{Titel}) \mid \mathsf{Buecher}(t_1) \land \\ &\forall t_2(\neg \mathsf{Buchhaendler}(t_2) \lor \quad \mathsf{oder:} \ \forall t_2(\mathsf{Buchhaendler}(t_2) \to \\ &\exists t_3(\mathsf{Angebote}(t_3) \ \land \ t_3.\mathsf{Haendler} = t_2.\mathsf{Name} \ \land \ t_3.\mathsf{Titel} = t_1.\mathsf{Titel})) \} \end{split}
```

DB:V-118 Relational Algebra & Calculus © STEIN 2004-2018

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

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

Tupelkalkül

```
\{(t_1.\mathsf{Titel}) \mid \mathsf{Buecher}(t_1) \land \\ \forall t_2(\neg \mathsf{Buchhaendler}(t_2) \lor \\ \exists t_3(\mathsf{Angebote}(t_3) \land t_3.\mathsf{Haendler} = t_2.\mathsf{Name} \land t_3.\mathsf{Titel} = t_1.\mathsf{Titel}))\}
```

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

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

Tupelkalkül

```
 \begin{split} &\{(t_1.\mathsf{Titel}) \mid \mathsf{Buecher}(t_1) \land \\ &\forall t_2(\neg \mathsf{Buchhaendler}(t_2) \lor \\ &\exists t_3(\mathsf{Angebote}(t_3) \land t_3.\mathsf{Haendler} = t_2.\mathsf{Name} \land t_3.\mathsf{Titel} = t_1.\mathsf{Titel})) \} \end{split}
```

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 vewendet 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 α , β , γ , die Prädikatsbezeichner P, Q, die Quantoren \forall , \exists , die Variablenbezeichner t, x, y, z, und die Junktoren \neg , \land , \lor , \rightarrow gehören zur Objektsprache. Das \approx -Zeichen ist ein Zeichen der Metasprache und steht für "ist logisch äquivalent mit". Es gelten u.a. folgende Zusammenhänge:

$$\forall x P(x) \approx \not\exists x (\neg P(x))$$

$$\alpha \to \beta \approx \neg \alpha \lor \beta$$

$$\neg (\alpha \land \beta) \approx \neg \alpha \lor \neg \beta$$

$$(\alpha \land \beta) \to \gamma \approx \neg \alpha \lor \neg \beta \lor \gamma$$

DB:V-121 Relational Algebra & Calculus © STEIN 2004-2018

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 erfüllt 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.

 \rightarrow Domäne einer Formel α

Sichere Anfragen (Fortsetzung)

Definition 4 (Domäne einer Formel α)

Der Bereich bzw. die Domäne einer Formel α ist die Menge aller Konstanten in α 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}

DB:V-123 Relational Algebra & Calculus © STEIN 2004-2018

Sichere Anfragen (Fortsetzung) [Domänenkalkül]

Folgende Anfrage liefert eine unendliche Zahl von Ergebnissen:

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

- 1. Gehört $t=(c_1,c_2,\ldots,c_n)$ in das Anfrageergebnis, so wird gefordert, dass $\{c_1,c_2,\ldots,c_n\}$ Teilmenge der Domäne von α ist.
 - ⇒ Die Suche zur Beantwortung der Anfrage ist auf die Domäne beschränkt.
- 2. Für jede Teilformel $\exists t\beta$ wird gefordert, dass β höchstens für Elemente aus der Domäne von β 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 β für alle Elemente aus der Domäne von β erfüllt ist.
 - $\Rightarrow \beta$ ist für alle Elemente, die nicht in seiner Domäne sind, immer erfüllt

Sichere Anfragen (Fortsetzung) [Domänenkalkül]

Folgende Anfrage liefert eine unendliche Zahl von Ergebnissen:

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

- 1. Gehört $t=(c_1,c_2,\ldots,c_n)$ in das Anfrageergebnis, so wird gefordert, dass $\{c_1,c_2,\ldots,c_n\}$ Teilmenge der Domäne von α ist.
 - ⇒ Die Suche zur Beantwortung der Anfrage ist auf die Domäne beschränkt.
- 2. Für jede Teilformel $\exists t\beta$ wird gefordert, dass β höchstens für Elemente aus der Domäne von β 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 β für alle Elemente aus der Domäne von β erfüllt ist.
 - $\Rightarrow \beta$ ist für alle Elemente, die nicht in seiner Domäne sind, immer erfüllt

Sichere Anfragen (Fortsetzung) [Domänenkalkül]

Folgende Anfrage liefert eine unendliche Zahl von Ergebnissen:

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

- 1. Gehört $t=(c_1,c_2,\ldots,c_n)$ in das Anfrageergebnis, so wird gefordert, dass $\{c_1,c_2,\ldots,c_n\}$ Teilmenge der Domäne von α ist.
 - ⇒ Die Suche zur Beantwortung der Anfrage ist auf die Domäne beschränkt.
- 2. Für jede Teilformel $\exists t\beta$ wird gefordert, dass β höchstens für Elemente aus der Domäne von β 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 β für alle Elemente aus der Domäne von β erfüllt ist.
 - $\Rightarrow \beta$ ist für alle Elemente, die nicht in seiner Domäne sind, immer erfüllt

Sichere Anfragen (Fortsetzung) [Domänenkalkül]

Folgende Anfrage liefert eine unendliche Zahl von Ergebnissen:

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

- 1. Gehört $t=(c_1,c_2,\ldots,c_n)$ in das Anfrageergebnis, so wird gefordert, dass $\{c_1,c_2,\ldots,c_n\}$ Teilmenge der Domäne von α ist.
 - ⇒ Die Suche zur Beantwortung der Anfrage ist auf die Domäne beschränkt.
- 2. Für jede Teilformel $\exists t\beta$ wird gefordert, dass β höchstens für Elemente aus der Domäne von β 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 β für alle Elemente aus der Domäne von β 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 α deshalb stark vereinfacht, weil *vereinbart* wird, dass Instanziierungen der Variablen mit Werten von außerhalb der Domäne keinen Einfluss auf die Erfüllbarkeit von α 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 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.

DB:V-128 Relational Algebra & Calculus © STEIN 2004-2018

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.

DB:V-129 Relational Algebra & Calculus © STEIN 2004-2018

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.

DB:V-130 Relational Algebra & Calculus © STEIN 2004-2018

Anfragen [Tupelkalkül]

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

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

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

DB:V-131 Relational Algebra & Calculus © STEIN 2004-2018

Anfragen [Tupelkalkül]

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

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

- $\neg x_1, \dots, x_n$ sind freie, x_{n+1}, \dots, x_{n+m} sind gebundene Domänenvariablen.
- f lpha ist eine logische Formel, wobei die Menge der Atome Σ , aus denen α 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.

DB:V-132 Relational Algebra & Calculus © STEIN 2004-2018

Anfragen [Tupelkalkül]

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

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

- $\neg x_1, \dots, x_n$ sind freie, x_{n+1}, \dots, x_{n+m} sind gebundene Domänenvariablen.
- 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 op x_j " ist ein Atom mit $op \in \{=, <, \leq, >, \geq, \neq\}$. x_i, x_j bezeichnen Domänenvariablen, die über den Wertebereichen der zugeordneten Attribute instanziiert sind.
- 3. " x_i op c" ist ein Atom mit $op \in \{=, <, \leq, >, \geq, \neq\}$. x_i bezeichnet eine Domänenvariable, die über dem Wertebereich des zugeordneten Attributes instanziiert ist, und $c \in dom(A_i)$ ist eine Konstante aus dem gleichen Wertebereich.

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	
Χ	1	Köln	5	
Υ	2	Hamburg	5	
Z	3	Weimar	4	
New	8	Weimar	4	

Anfrage

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

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	
Χ	1	Köln	5	
Υ	2	Hamburg	5	
Z	3	Weimar	4	
New	8	Weimar	4	

Anfrage

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

Relationenalgebra

 \sim \mathcal{TAFEL}

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	
Χ	1	Köln	5	
Υ	2	Hamburg	5	
Z	3	Weimar	4	
New	8	Weimar	4	

Anfrage

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

Relationenalgebra

$$\sim$$
 TAFEL

Domänenkalkül

```
 \{(x_1,x_3) \mid \exists x_2 \exists x_4 \exists x_5 \ \exists y_1 \exists y_2 \exists y_3 \\ (\mathsf{Mitarbeiter}(x_1,x_2,x_3,x_4,x_5) \ \land \ \mathsf{Abteilung}(y_1,y_2,y_3) \ \land \ y_1 = \mathsf{'Forschung'} \ \land \ y_2 = x_5) \}
```

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
Χ	1	Köln	5
Υ	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

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

Relationenalgebra

 \sim TAFEL

Domänenkalkül Mögliche Konvention: nicht-freie Variablen sind per Default ∃-quantifiziert.

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

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

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
Χ	1	Köln	5
Υ	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

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

Relationenalgebra

 \sim TAFEL

Domänenkalkül Abkürzung: Konstanten als Parameter.

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

(Mitarbeiter(x_1, x_2, x_3, x_4, x_5) \land Abteilung('Forschung', y_2, y_3) \land $y_2 = x_5$)

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
Χ	1	Köln	5
Υ	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

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

Relationenalgebra

$$\sim$$
 TAFEL

```
Domänenkalkül Abkürzung: Unifikation von Domänenvariablen \{(x_1, x_3) \mid (\text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \land \text{Abteilung}(\text{Forschung}', x_5, y_3))\}
```

Bemerkungen:

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

Beispiel: $y_1 = '$ Forschung'

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

Beispiel: $x_5 = y_2$

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
Χ	1	Köln	5
Υ	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."

DB:V-141 Relational Algebra & Calculus © STEIN 2004-2018

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
Χ	1	Köln	5
Υ	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_{\mathsf{PNr},\mathsf{AbtNr},} ((\rho_{\mathsf{PNr}\leftarrow\mathsf{Nr},}(\sigma_{\mathsf{Ort}='\mathsf{Weimar'}}(\mathsf{Projekt}))) \bowtie_{\mathsf{AbtNr}=\mathsf{Nr}} \mathsf{Abteilung} \bowtie_{\mathsf{Manager}=\mathsf{PersNr}} \mathsf{Mitarbeiter})
```

DB:V-142 Relational Algebra & Calculus © STEIN 2004-2018

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
Χ	1	Köln	5
Υ	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

 $\pi_{\mathsf{PNr},\mathsf{AbtNr},} ((\rho_{\mathsf{PNr}\leftarrow\mathsf{Nr},} (\sigma_{\mathsf{Ort}='\mathsf{Weimar'}}(\mathsf{Projekt}))) \bowtie_{\mathsf{AbtNr}=\mathsf{Nr}} \mathsf{Abteilung} \bowtie_{\mathsf{Manager}=\mathsf{PersNr}} \mathsf{Mitarbeiter})$

Domänenkalkül

 \sim TAFEL

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
Χ	1	Köln	5
Υ	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."

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
Χ	1	Köln	5
Υ	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

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

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
Χ	1	Köln	5
Υ	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

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

Domänenkalkül

 \sim TAFEL

Bemerkungen:

□ Die Quantoren können verschoben werden, solange sich keine Variablenbindungen ändern und die Ordnung zwischen den ∃- und ∀-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 \\ (\mathsf{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \ \land \ (\neg \mathsf{Projekt}(z_1, z_2, z_3, z_4) \ \lor \ \neg (z_4 = 5) \ \lor \\ (\mathsf{ArbeitetInProjekt}(y_1, y_2) \ \land \ y_2 = z_2 \ \land \ y_1 = x_2))) \}
```

- Beispiele für alternative Formeln, die denselben Sachverhalt "Für alle Projekte der Abteilung 5 gilt . . . " modellieren, also logisch äquivalent sind:
 - 1. $\forall z_1 \forall z_2 \forall z_3 \forall z_4 \ ((\mathsf{Projekt}(z_1, z_2, z_3, z_4) \land (z_4 = 5)) \rightarrow \dots$
 - 2. $\forall z_1 \forall z_2 \forall z_3 \forall z_4 \ (\neg(\mathsf{Projekt}(z_1, z_2, z_3, z_4) \land z_4 = 5) \lor \ldots$
 - 3. $\forall z_1 \forall z_2 \forall z_3 \forall z_4 \ (\neg \mathsf{Projekt}(z_1, z_2, z_3, z_4) \lor \neg (z_4 = 5) \lor \dots$
 - 4. $\forall z_1 \forall z_2 \forall z_3 \ (\mathsf{Projekt}(z_1, z_2, z_3, 5) \rightarrow \ldots$
 - 5. $\forall z_1 \forall z_2 \forall z_3 \ (\neg \mathsf{Projekt}(z_1, z_2, z_3, 5) \lor \ldots$

Beispiel 4 [Tupelkalkül]

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

Bucl	hhaendler	
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?"

DB:V-148 Relational Algebra & Calculus © STEIN 2004-2018

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

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

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

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

Domänenkalkül

```
 \{(x_1) \mid \exists x_2(\mathsf{Buecher}(x_1, x_2) \land \\ \forall y_1 \forall y_2 \forall y_3(\neg \mathsf{Buchhaendler}(y_1, y_2, y_3) \lor \mathsf{oder} \colon \forall y_1 \forall y_2 \forall y_3(\mathsf{Buchhaendler}(y_1, y_2, y_3) \to \\ \exists z_1 \exists z_2(\mathsf{Angebote}(z_1, z_2) \land z_2 = y_1 \land z_1 = x))) \}
```

Sichere Anfragen [Tupelkalkül]

Folgende Anfrage liefert eine unendliche Zahl von Ergebnissen:

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

Sichere Anfragen [Tupelkalkül]

Folgende Anfrage liefert eine unendliche Zahl von Ergebnissen:

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

Sei die <u>Domäne einer Formel</u> α wie zuvor definiert. Dann ist eine Anfrage $\{(x_1,x_2,\ldots,x_n)\mid \alpha\}$ des Domänenkalküls sicher, falls folgende Bedingungen erfüllt sind:

- 1. Gehört (c_1, c_2, \ldots, c_n) in das Anfrageergebnis, so muss $\{c_1, c_2, \ldots, c_n\}$ Teilmenge der Domäne von α sein.
 - ⇒ Die Suche zur Beantwortung der Anfrage ist auf die Domäne beschränkt.
- 2. Für jede Teilformel $\exists x \beta$ muss gelten, dass β 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 β für alle Elemente aus seiner Domäne erfüllt ist.
 - $\Rightarrow \beta$ ist für alle Elemente, die nicht in seiner Domäne sind, immer erfüllt.

Anfragekalküle

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

DB:V-153 Relational Algebra & Calculus © STEIN 2004-2018

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.

DB:V-154 Relational Algebra & Calculus © STEIN 2004-2018