

Databases 1, Labsessie 1

Opgelet: Deze uitwerkingen zijn niet gegarandeerd correct, en zijn ook niet gegarandeerd de enige correcte uitwerkingen.

Algemene opmerking vooraf: De gegeven set opgaven is wat de docenten verwachten dat je kunt maken in 2×45 minuten. Dit is tijdens de labsessie niet gehaald en betekent dat er nog hard moet gewerkt worden om tempo te leren maken!

1 Voorkennis

1a. $K(\text{Jan}, \text{Ans}) \wedge M(\text{Jan})$

1b. $(\exists x : G(x, \text{Jan}) : K(x, \text{Kees}) \wedge M(\text{Kees}))$

Tijdens de labsessie is gebleken dat niet iedereen kwantoren heeft gebruikt. In deze expressies mogen geen vrije variabelen voorkomen. Zomaar een x gebruiken zonder die via een kwantor te binden mag niet.

1c. $(\exists x : G(x, \text{Klazien}) : K(x, \text{Ans}) \wedge \neg M(\text{Klazien})) \wedge$
 $(\forall x, y : G(x, y) : (K(x, \text{Ans}) \wedge \neg M(y)) \Rightarrow y = \text{Klazien})$

Bij deze vraag is tijdens de labsessie gebleken dat sommigen alleen maar een eis hebben opgeschreven die moet gelden voor mogelijke schoondochters, maar geen existentiële kwantor die zegt dat de schoondochter ook moet bestaan. Sommigen hebben ook schoonzoons verboden. Dat was niet de bedoeling.

1d. $(\exists x : K(\text{Erik}, x) \wedge K(\text{Wil}, x) : \forall y : K(\text{Erik}, y) \wedge K(\text{Wil}, y) : y = x)) \wedge M(\text{Erik}) \wedge M(\text{Wil})$

2a. b is grootmoeder van a

2b. Alleen personen van verschillend geslacht mogen getrouwd zijn.

3. $(\forall x, y : (\exists z :: K(x, z) \wedge K(y, z)) : \neg G(x, y))$

= { verwissel de condities binnen de universele quantor }

$(\forall x, y : G(x, y) : \neg(\exists z :: K(x, z) \wedge K(y, z)))$

= { er bestaat geen = voor alle geldt niet }

$(\forall x, y : G(x, y) : (\forall z :: \neg(K(x, z) \wedge K(y, z))))$

= { verwissel condities binnen de universele quantor opnieuw }

$(\forall x, y : G(x, y) : (\forall z : K(x, z) : \neg K(y, z)))$

Betekenis: Echtparen moeten verschillende ouders hebben.

2 Structuur van een database

1. De man uit ieder huwelijk komt in de personentabel voor.

Hierbij willen we nog even opmerken dat *natuurlijke taal* natuurlijk moet klinken. Het moet een voor een groot publiek begrijpelijke uitdrukking zijn, geen specialistische taal over elementen en records en dergelijke.

2. Er zijn geen twee personen met dezelfde naam.

3. $(\forall x, y : x \in \text{huwelijken} \wedge x[\{\text{naamman}, \text{datum}\}] = y[\{\text{naamman}, \text{datum}\}] : x = y)$

4. $(\forall x : x \in \text{personen} \wedge x[\text{geslacht}] = V : (\exists y : y \in \text{huwelijken} : y[\text{naamvrouw}] = x[\text{naam}]))$

3 Database Design

1. De functionele afhankelijkheden die niet gelden:

$\emptyset \rightarrow \{\text{naam}\}$

$\emptyset \rightarrow \{\text{geboortedatum}\}$

$\emptyset \rightarrow \{\text{geslacht}\}$

$\emptyset \rightarrow \{\text{naam, geboortedatum}\}$

$\emptyset \rightarrow \{\text{naam, geslacht}\}$

$\emptyset \rightarrow \{\text{geboortedatum, geslacht}\}$

$\emptyset \rightarrow \{\text{naam, geboortedatum, geslacht}\}$

$\{\text{geslacht}\} \rightarrow \{\text{naam}\}$

$\{\text{geslacht}\} \rightarrow \{\text{geboortedatum}\}$

$\{\text{geslacht}\} \rightarrow \{\text{naam, geboortedatum}\}$

$\{\text{geslacht}\} \rightarrow \{\text{naam, geslacht}\}$

$\{\text{geslacht}\} \rightarrow \{\text{geboortedatum, geslacht}\}$

$\{\text{geslacht}\} \rightarrow \{\text{naam, geboortedatum, geslacht}\}$

Merk op dat een functionele afhankelijkheid met \emptyset aan de linkerkant alleen maar geldt in een lege tabel of een tabel met maar 1 tupel. Let ook op dat de vraag gesteld is over de gegeven *instance*. Het doet er dus niet toe wat er “in het algemeen” moet gelden in een persoonstabel.

2. De functionele afhankelijkheden van 1 attr naar 1 attr in Figure 7.1:

$\{\text{branch-name}\} \rightarrow \{\text{branch-name}\}$

$\{\text{branch-city}\} \rightarrow \{\text{branch-city}\}$

$\{\text{assets}\} \rightarrow \{\text{assets}\}$

$\{\text{customer-name}\} \rightarrow \{\text{customer-name}\}$

$\{\text{loan-number}\} \rightarrow \{\text{loan-number}\}$

$\{\text{amount}\} \rightarrow \{\text{amount}\}$

$\{\text{branch-name}\} \rightarrow \{\text{branch-city}\}$

$\{\text{branch-name}\} \rightarrow \{\text{assets}\}$

$\{\text{loan-number}\} \rightarrow \{\text{branch-name}\}$

$\{\text{loan-number}\} \rightarrow \{\text{branch-city}\}$

$\{\text{loan-number}\} \rightarrow \{\text{assets}\}$

$\{\text{loan-number}\} \rightarrow \{\text{customer-name}\}$

$\{\text{loan-number}\} \rightarrow \{\text{amount}\}$

$\{\text{assets}\} \rightarrow \{\text{branch-name}\}$

$\{\text{assets}\} \rightarrow \{\text{branch-city}\}$

4 Relationale Algebra

1a. $\Pi_{\text{branch-name}}(\text{loan})$

1b. $\sigma_{\text{assets} > 1.000.000}(\text{branch})$

1c. $\Pi_{\text{customer-name}}(\sigma_{\text{customer-city} = \text{“Eindhoven”}}(\text{customer}))$

1d. $\Pi_{\text{branch-name}}(\text{account}) \cup \Pi_{\text{branch-name}}(\text{loan})$

1e. $\Pi_{\text{branch-name}}(\text{branch}) - \Pi_{\text{branch-name}}(\text{loan})$

2a. Geef de rekeningnummers van alle rekeningen met een saldo boven de 10.000.

2b. Geef alle steden waar klanten wonen maar geen filiaal is.

2c. Geef de naam van alle klanten die een rekening hebben met een saldo boven de 10.000.