Een relationele database ontwerpen door normalisatie

Agenda

- Relationeel Database ontwerp
  - Typisch ontwerpproces
  - Normaliseren
  - Concepten
  - Normaalvorm
- Normaliseren
  - Oefeningen

Relationeel Database ontwerp

Typisch ontwerpproces

1. Requirements verzamelen
   - Informatiebehoefte opdrachtgever in kaart brengen
2. Relaties uitschrijven
   - Requirements vertalen in tabellen
3. Relaties normaliseren naar gewenste niveau
   - Voorkomt potentiële problemen
     - duplicatie → inconsistentie
   - Helpt om bestaande databank te fixen

Relationeel Database ontwerp

Normaliseren

- **Input**: tabel(len) / relaties
- **Output**: meestal meer tabellen, van hogere kwaliteit
- Wanneer normaliseren?
  - Bij het ontwerp
  - Bestaande databank analyseren en fouten detecteren
  - Bij aanpassingen van een bestaande databank: impact controleren

Theorie

Concepten

- Benodigde concepten
  - Atomiciteit
  - Kandidaatsleutels
  - Functionele afhankelijkheden
  - Normalisatievormen
- We gebruiken als basisvoorbeeld deze relatie:

Boekenwinkel					
isbn	**titel**	**genre**	**fiction**	**klantid**	**betalingen**
99-99	Superman	strip	true	jos1	10, 5, 5
87-06	Bloembollen	tuin	false	jos1	23
32-00	Endymion	scifi	true	jef2	10
87-06	Bloembollen	tuin	false	eve1	18
12-72	Alien	scifi	true	ilse1	7

Theorie

Concepten - Atomiciteit

- Atomiciteit: waarden mogen niet opsplitsbaar zijn
- Tegenvoorbeelden
  - Arrays
  - Lists
  - JSON-objecten
  - Tabellen
  - …

Theorie

Concepten - Atomiciteit

- Voorbeeld niet-atomaire waarden: een tabel als waarde gebruiken

Boekenwinkel						
isbn	**titel**	**genre**		**fiction**	**klantid**	**betalingen**
99-99	Superman	comic book	engels	true	jos1	10, 5, 5
		strip	nederlands			
87-06	Bloembollen	tuin	nederlands	false	jos1	23
		garden	engels			

Theorie

Concepten - Atomiciteit

- Voorbeeld niet-atomaire waarden: een array als waarde gebruiken

Boekenwinkel					
isbn	**titel**	**genre**	**fiction**	**klantid**	**betalingen**
99-99	Superman	`["comic book", "strip"]`	true	jos1	10, 5, 5
87-06	Bloembollen	`["tuin", "garden"]`	false	jos1	23

Theorie

Concepten - Atomiciteit

- Voorbeeld niet-atomaire waarden: een JSON-object als waarde gebruiken

Boekenwinkel				
<u>isbn</u>	**titel**	**genre**	**<u>klantid</u>**	**betalingen**
99-99	Superman	`{ “name”: “comic book”, “fiction”: true }`	jos1	10, 5, 5
87-06	Bloembollen	`{ “name”: “tuin”, “fiction”: false }`	jos1	23

Theorie

Concepten - Atomiciteit

- Voorbeeld niet-atomaire waarden: waarden herhalen in het algemeen

Boekenwinkel					
isbn	**titel**	**genre**	**fiction**	**klantid**	**betalingen**
99-99	Superman	comic book	true	jos1	10, 5, 5
87-06	Bloembollen	tuin	false	jos1	23
32-00	Endymion	scifi	true	jef2	10
87-06	Bloembollen	tuin	false	eve1	18
12-72	Alien	scifi	true	ilse1	7

Theorie

Concepten - Atomiciteit

- Voorbeeld niet-atomaire waarden: kolommen herhalen als lapmiddel

Boekenwinkel							
isbn	**titel**	**genre**	**fiction**	**klantid**	**betaling 1**	**betaling 2**	**betaling 3**
99-99	Superman	comic book	true	jos1	10	5	5
87-06	Bloembollen	tuin	false	jos1	23	null	null
32-00	Endymion	scifi	true	jef2	10	null	null
87-06	Bloembollen	tuin	false	eve1	18	null	null
12-72	Alien	scifi	true	ilse1	7	null	null

Theorie

Concepten – Functionele Afhankelijkheden

- Notatie
  - B is **functioneel afhankelijk van** A
  - **A → B**
  - A en B zijn *sets* van attributen
- Intuïtieve betekenis
  - Als je A kent, ken je dan ook B zonder enige twijfel?
- Definitie
  - B is **functioneel afhankelijk** van A als er bij één waarde van A slechts één waarde van B voorkomt.

Theorie

Concepten – Functionele Afhankelijkheden

- Illustratie functionele afhankelijkheid

Boekenwinkel					
isbn	**titel**	**genre**	**fiction**	**klantid**	**betalingen**
99-99	Superman	comic book	true	jos1	10, 5, 5
87-06	Bloembollen	tuin	false	jos1	23
32-00	Endymion	scifi	true	jef2	10
87-06	Bloembollen	tuin	false	eve1	18
12-72	Alien	scifi	true	ilse1	7

(isbn)	→	(titel)	(fiction)	↛	(genre)
(isbn)	→	(genre)	(betalingen)	↛	(titel)
(isbn, klantid)	→	(betalingen)	(genre)	↛	(titel)
(isbn, klantid, titel)	→	(genre)	(genre, fiction)	↛	(titel)

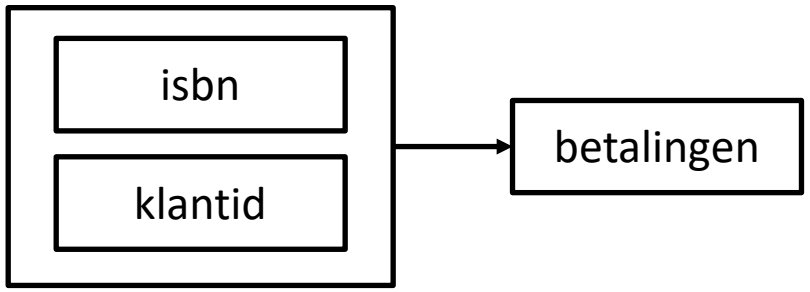
katholieke hogeschool associatie KU Leuven vives

Theorie

Concepten – Functionele Afhankelijkheden

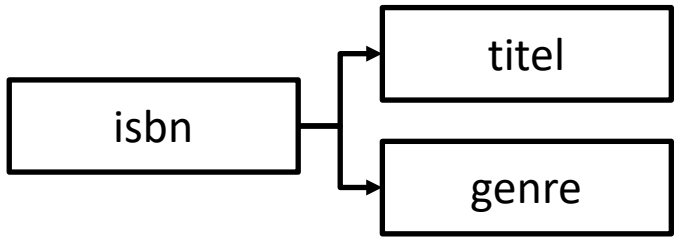
- Illustratie functionele afhankelijkheid

(isbn, klantid) → (betalingen)

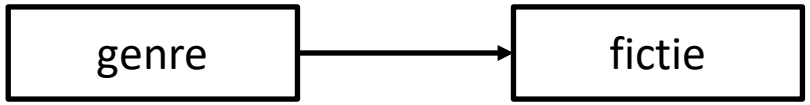


(isbn) → (titel)

(isbn) → (genre)



(genre) → (fictie)



Theorie

Concepten – Functionele Afhankelijkheden

- Functioneel afhankelijk, of **f.a.**
- Verschillende soorten functionele afhankelijkheden
  - **Volledig** f.a.
  - **Partieel** f.a.
  - **Transitief** f.a.

Theorie

Concepten – Volledig functioneel afhankelijk

- B is **volledig functioneel afhankelijk** van A als
  - B f.a. is van A
  - B **niet** f.a. is van een **stuk van A**

- Abstract voorbeeld
  - Als dit geldt: $(A,B,C) \rightarrow (D)$
  - Kunnen we dat een *volledig functionele afhankelijkheid* noemen als
    - $(A) \not\rightarrow (D)$ en $(B) \not\rightarrow (D)$ en $(C) \not\rightarrow (D)$ én
    - $(A,B) \not\rightarrow (D)$ en $(A,C) \not\rightarrow (D)$ en $(B,C) \not\rightarrow (D)$

Theorie

Concepten – Volledig functioneel afhankelijk - voorbeelddata

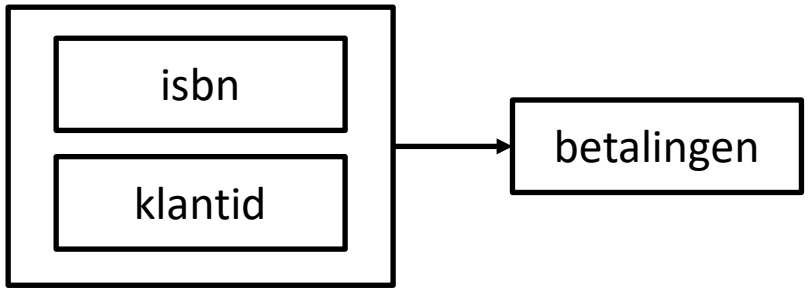
- Voorbeeldvraag 1: “is **(betalingen)** *volledig f.a.* van **(isbn, klantid)**?”
  - (isbn, klantid) → (betalingen) ?
  - (betalingen) **mag niet** f.a. zijn van (isbn) of (klantid)

Boekenwinkel					
isbn	**titel**	**genre**	**fiction**	**klantid**	**betalingen**
99-99	Superman	comic book	true	jos1	10, 5, 5
87-06	Bloembollen	tuin	false	jos1	23
32-00	Endymion	scifi	true	jef2	10
87-06	Bloembollen	tuin	false	eve1	18
12-72	Alien	scifi	true	ilse1	7

Theorie

Concepten – Volledig functioneel afhankelijk - voorbeelddata

- Voorbeeldvraag 1: “is **(betalingen)** *volledig f.a.* van **(isbn, klantid)**?”
  - ✓ (isbn, klantid) → (betalingen)
  - ✓ (isbn) ↛ (betalingen)
  - ✓ (klantid) ↛ (betalingen)



✓

volledig f.a.

Boekenwinkel					
isbn	**titel**	**genre**	**fiction**	**klantid**	**betalingen**
99-99	Superman	comic book	true	jos1	10, 5, 5
87-06	Bloembollen	tuin	false	jos1	23
32-00	Endymion	scifi	true	jef2	10
87-06	Bloembollen	tuin	false	eve1	18
12-72	Alien	scifi	true	ilse1	7

Theorie

Concepten – Volledig functioneel afhankelijk - voorbeelddata

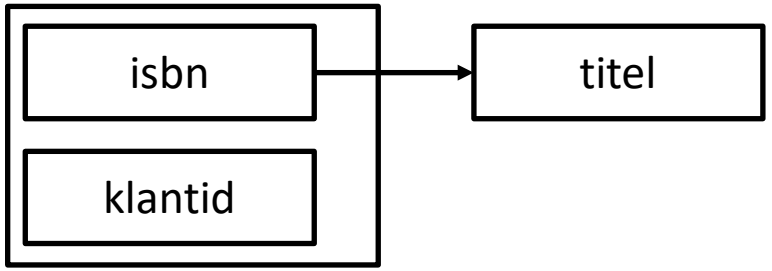
- Voorbeeldvraag 2: “is **(titel)** *volledig f.a.* van **(isbn, klantid)**?”
  - (isbn, klantid) → (titel) ?
  - (titel) **mag niet** f.a. zijn van (isbn) of (klantid)

Boekenwinkel					
isbn	**titel**	**genre**	**fiction**	**klantid**	**betalingen**
99-99	Superman	comic book	true	jos1	10, 5, 5
87-06	Bloembollen	tuin	false	jos1	23
32-00	Endymion	scifi	true	jef2	10
87-06	Bloembollen	tuin	false	eve1	18
12-72	Alien	scifi	true	ilse1	7

Theorie

Concepten – Volledig functioneel afhankelijk - voorbeelddata

- Voorbeeldvraag 2: “is **(titel)** *volledig f.a.* van **(isbn, klantid)**?”
  - ✓ (isbn, klantid) → (titel)
  - ✓ (klantid) ↛ (titel)
  - ✗ (isbn) → (titel)



niet volledig f.a.

Boekenwinkel					
isbn	**titel**	**genre**	**fiction**	**klantid**	**betalingen**
99-99	Superman	comic book	true	jos1	10, 5, 5
87-06	Bloembollen	tuin	false	jos1	23
32-00	Endymion	scifi	true	jef2	10
87-06	Bloembollen	tuin	false	eve1	18
12-72	Alien	scifi	true	ilse1	7

Theorie

Concepten – Volledig functioneel afhankelijk - voorbeelddata

- Alle **volledig** functionele afhankelijkheden in *boekenwinkel*:

(isbn) → (titel)

(isbn) → (genre)

(isbn) → (fiction)

(isbn, klantid) → (betalingen)

(genre) → (fiction)

Boekenwinkel					
isbn	**titel**	**genre**	**fiction**	**klantid**	**betalingen**
99-99	Superman	comic book	true	jos1	10, 5, 5
87-06	Bloembollen	tuin	false	jos1	23
32-00	Endymion	scifi	true	jef2	10
87-06	Bloembollen	tuin	false	eve1	18
12-72	Alien	scifi	true	ilse1	7

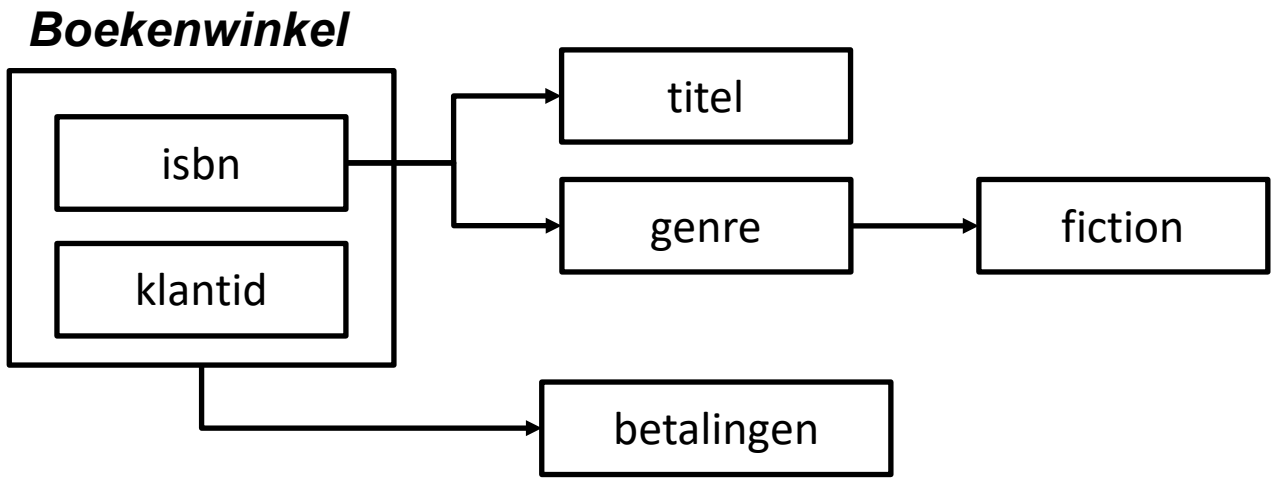
Theorie

Concepten – Volledig functioneel afhankelijk - voorbeelddata

- Alle **volledig** functionele afhankelijkheden in *boekenwinkel*:

(isbn)	→	(titel)		(isbn, klantid)	→	(betalingen)
(isbn)	→	(genre)		(genre)	→	(fiction)
(isbn)	→	(fiction)				

Impliciete f.a. worden visueel niet getoond
Afleidbaar uit andere f.a.



Theorie

Concepten – Partieel functioneel afhankelijk

- Steunt op **kandidaatsleutels**
- B is **partieel functioneel afhankelijk** van A als
  - A → B
  - B niet in *een* kandidaatsleutel zit
  - A een stuk is van *een* kandidaatsleutel
- Abstract voorbeeld
  - Als dit geldt:
    - `(X,Y) → (Z)`
    - `(X) → (Z)`
    - `(X,Y)` is een kandidaatsleutel
    - `(Z)` zit niet in een kandidaatsleutel
  - Dan is Z partieel f.a. van (X,Y)
- Opmerking: kan dus enkel voorkomen bij *samengestelde sleutels*

Theorie

Concepten – Partieel functioneel afhankelijk - voorbeelddata

Kandidaatsleutels?

(isbn,klantid)

Partiële f.a.?

(isbn) is **deel** van ***een*** kandidaatsleutel

(isbn)	→	(titel)
(isbn)	→	(genre)
(isbn)	→	(fiction)

Boekenwinkel					
isbn	**titel**	**genre**	**fiction**	**klantid**	**betalingen**
99-99	Superman	comic book	true	jos1	10, 5, 5
87-06	Bloembollen	tuin	false	jos1	23
32-00	Endymion	scifi	true	jef2	10
87-06	Bloembollen	tuin	false	eve1	18
12-72	Alien	scifi	true	ilse1	7

Theorie

Concepten – Transitief functioneel afhankelijk

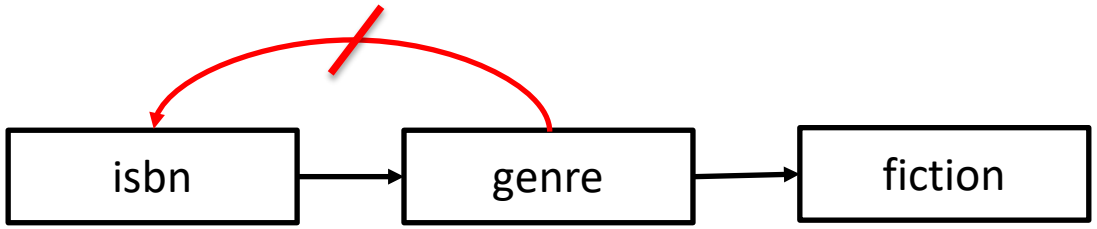
- Neem A, B en C: 3 verschillende sets van attributen (mogen overlappen)
- C is **transitief functioneel afhankelijk** van A als
  - A → B
  - B ↛ A
  - B → C

Theorie

Concepten – Transitief functioneel afhankelijk - voorbeelddata

✓ (fiction) **transitief f.a. van** (isbn)

- (isbn) → (genre)
- (genre) ↛ (isbn)
- (genre) → (fiction)



Boekenwinkel					
isbn	**titel**	**genre**	**fiction**	**klantid**	**betalingen**
99-99	Superman	comic book	true	jos1	10, 5, 5
87-06	Bloembollen	tuin	false	jos1	23
32-00	Endymion	scifi	true	jef2	10
87-06	Bloembollen	tuin	false	eve1	18
12-72	Alien	scifi	true	ilse1	7

Theorie

Concepten – Transitief functioneel afhankelijk - voorbeelddata

Stel **Person(id, email, name)**

Kandidaatsleutels: (id) en (email)

- (id) → (email)
- (email) → (id)
- (id) → (name)
- (email) → (name)

✗ (name) **transitief f.a.** van (id_person)

Person		
id	**email**	**name**
1	Eveline.Lauwers@vives.be	Eveline Lauwers
2	Koen.Peeters@mail.com	Koen Peeters
3	Mo_Vanassche@provider.be	Mo Vanassche
4	Koen@Peeters.com	Koen Peeters

Theorie

Normaalvorm

- Normaalvormen zijn *regels*
- Een **relatie** (tabel) is conform de normaalvorm als die eraan voldoet
- Dienen om
  - Bestaande databases te checken op potentiële fouten
  - Ontwerp nieuwe databases te versterken
- *Normaliseren* is het transformatieproces van tabellen naar een gewenste normaalvorm
  - We beschouwen telkens *één tabel per keer*
  - *Elke* tabel kan op zichzelf aan een normaalvorm voldoen

Theorie

Normaalvorm overzicht

- Normaalvormen (of NF, Normal Forms)
  - 0NF (unnormalized normal form - UNF)
  - 1NF (first normal form, of eerste normaalvorm)
  - 2NF
  - 3NF
  - BCNF (Boyce-Codd normal form)
  - 4NF
  - 5NF
  - DKNF (Domain-key normal form)
  - 6NF

Theorie

Normaalvorm overzicht

- Normaalvormen
  - 0NF
  - 1NF
  - 2NF
  - 3NF
  - BCNF
  - 4NF
  - 5NF
  - DKNF
  - 6NF

- Steeds strikter
  - een tabel staat niet in 3NF als ze niet in 2NF staat, etc
- Steeds moeilijker om te analyseren en mee te werken
- Typisch **meer tabellen** hoe hoger de normaalvorm
  - Vaak **minder efficiënt** in een database
- Elke NF lost specifieke ontwerpproblemen op

Theorie

Normaalvorm overzicht

- Normaalvormen
  - 0NF
  - 1NF
  - 2NF
  - 3NF
  - BCNF
  - 4NF
  - 5NF
  - DKNF
  - 6NF

- We beperken ons tot 3NF
- Goed evenwicht tussen complexiteit en foutbestendig maken
- In de praktijk het vaakst gebruikt

Theorie

Normaalvorm overzicht

- Wat kunnen we ermee aanvangen?
  - Checken of een tabel *voldoet* aan een normaalvorm
  - Indien het niet voldoet, weten we wat het *probleem* is
- Probleem geïdentificeerd in een tabel?
  - Verplaats het naar een nieuwe tabel
  - Originele tabel nu (potentieel) wel conform de normaalvorm
- Resultaat?
  - Evenveel of meer tabellen die (potentieel) wél voldoen aan de gewenste normaalvorm
  - We kunnen **steeds** de originele tabel **reconstrueren**

Normaliseren

Normaalvorm 0

- Definitie
  - *Alles kan, zolang het in tabelformaat staat*
- Problemen
  - Je kan alle data in 1 tabel proppen
  - Er kunnen niet-atomaire waarden voorkomen
  - Weinig afgedwongen structuur
  - Geen garanties in termen van duplicaten, consistentie, redundantie, …

Normaliseren

Normaalvorm 0 - voorbeelddata

✓ Relatie *Boekenwinkel* staat in tabulair format

– 0NF vereist niet veel meer dan dat

✓ *Deze tabel is conform 0NF*

Boekenwinkel					
isbn	**titel**	**genre**	**fiction**	**klantid**	**betalingen**
99-99	Superman	comic book	true	jos1	10, 5, 5
87-06	Bloembollen	tuin	false	jos1	23
32-00	Endymion	scifi	true	jef2	10
87-06	Bloembollen	tuin	false	eve1	18
12-72	Alien	scifi	true	ilse1	7

Normaliseren

Normaalvorm 1

- Definitie
  - *Er mogen enkel atomaire waarden in de tabel voorkomen*
  - *Tabel moet geldige relatie zijn: geen duplicate tupels, ...*
- Wat lost dit op t.o.v. 0NF?
  - Enkel atomaire waarden: meer afgedwongen structuur
  - Een tabel in 1NF is een *basistabel* die voldoet aan relationele algebra

Normaliseren

Normaalvorm 1 - voorbeelddata

✓ Tabel is conform 0NF

Er komen niet-atomaire waarden voor

✗ *Tabel is niet conform 1NF*

Boekenwinkel					
isbn	**titel**	**genre**	**fiction**	**klantid**	**betalingen**
99-99	Superman	comic book	true	jos1	10, 5, 5
87-06	Bloembollen	tuin	false	jos1	23
32-00	Endymion	scifi	true	jef2	10
87-06	Bloembollen	tuin	false	eve1	18
12-72	Alien	scifi	true	ilse1	7

Normaliseren

Van normaalvorm 0 naar 1

- Stel: tabel X staat in 0NF
- Potentieel probleem
  - **Niet-atomaire waarden**
- Splits probleemattributen af naar nieuwe tabel
  - Zorg voor (nieuwe, correcte) primaire sleutel
- Zorg dat je originele data kan **reconstrueren**
  - Kopieer primaire sleutel uit originele tabel mee naar nieuwe tabel

Normaliseren

Van normaalvorm 0 naar 1 - voorbeelddata

Splits niet-atomaire waarden af
Bepaal primaire sleutel van nieuwe tabel

- Kopieer originele PK
- Hier nieuw attribuut nodig (volgnr)

Boekenwinkel					
isbn	titel	genre	fiction	klantid	betalingen
99-99	Superman	comic book	true	jos1	10, 5, 5
87-06	Bloembollen	tuin	false	jos1	23
32-00	En				10
87-06	Bl				18
12-72	Al				7

Boekenwinkel				
isbn	titel	genre	fiction	klantid
99-99	Superman	comic book	true	jos1
87-06	Bloembollen	tuin	false	jos1
32-00	Endymion	scifi	true	jef2
87-06	Bloembollen	tuin	false	eve1
12-72	Alien	scifi	true	ilse1

Aankoop			
isbn	klantid	volgnr	betaling
99-99	jos1	1	10
99-99	jos1	2	5
99-99	jos1	3	5
87-06	jos1	1	23
32-00	jef2	1	10
87-06	eve1	1	18
12-72	ilse1	1	7

Normaliseren

Van normaalvorm 0 naar 1 - voorbeelddata

- Start
  - Boekenwinkel (isbn, titel, genre, fiction, klantid, betalingen)
  - Staat in 0NF, maar niet in 1NF: probleemattribuut
- Splits probleemattribuut af naar nieuwe relatie “Aankoop”
  - Aankoop (isbn, titel, volgnr, betaling)
  - Het attribuut wordt nu “enkelvoudig”, atomair
  - Originele kandidaatsleutel gekopieerd
  - Zorg dat nieuwe relatie een correcte kandidaatsleutel heeft
    - (isbn, klantid) niet uniek, introduceer (volgnr)
- Resultaat
  - Boekenwinkel (isbn, titel, genre, fiction, klantid)
  - Aankoop (isbn, klantid, volgnr, betaling)
  - Beide relaties staan in 1NF: geen niet-atomaire attributen

Normaliseren

Relaties in 1NF

Boekenwinkel				
isbn	**titel**	**genre**	**fiction**	**klantid**
99-99	Superman	comic book	true	jos1
87-06	Bloembollen	tuin	false	jos1
32-00	Endymion	scifi	true	jef2
87-06	Bloembollen	tuin	false	eve1
12-72	Alien	scifi	true	ilse1

Aankoop			
isbn	**klantid**	**volgnr**	**betaling**
99-99	jos1	*1*	10
99-99	jos1	*2*	5
99-99	jos1	*3*	5
87-06	jos1	*1*	23
32-00	jef2	*1*	10
87-06	eve1	*1*	18
12-72	ilse1	*1*	7

Normaliseren

Normaalvorm 2

- Definitie
  - *De tabel is conform 1NF*
  - *Een attribuut dat **niet** in een kandidaatsleutel voorkomt mag **niet** partieel functioneel afhankelijk zijn van een kandidaatsleutel*

Normaliseren

Normaalvorm 2

- Wat lost 2NF op t.o.v. 1NF?
  - Redundantie wordt weggewerkt
  - Opslagrestrictie wordt weggewerkt

Boekenwinkel				
isbn	**titel**	**genre**	**fiction**	**klantid**
99-99	Superman	comic book	true	jos1
87-06	Bloembollen	tuin	false	jos1
32-00	Endymion	scifi	true	jef2
87-06	Bloembollen	tuin	false	eve1
12-72	Alien	scifi	true	Ilse1
~~33-87~~	~~Loofbomen~~	~~tuin~~	~~false~~	~~null~~

Normaliseren

Normaalvorm 2 - voorbeelddata

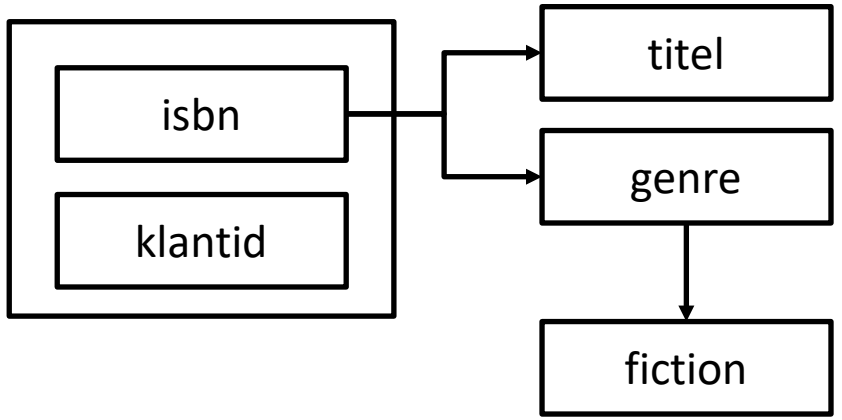
✓ Boekenwinkel is conform 1NF (in 0NF en enkel atomaire waarden)

(isbn) → (titel)
(isbn) → (genre)
(isbn) → (fiction)

(isbn, klantid) is een kandidaatsleutel:
(titel), (genre), (fiction) zijn er ***partieel f.a.*** van

✗ Boekenwinkel is niet conform 2NF

Boekenwinkel				
<u>isbn</u>	**titel**	**genre**	**fiction**	**<u>klantid</u>**
99-99	Superman	comic book	true	jos1
87-06	Bloembollen	tuin	false	jos1
32-00	Endymion	scifi	true	jef2
87-06	Bloembollen	tuin	false	eve1
12-72	Alien	scifi	true	ilse1



Normaliseren

Van normaalvorm 1 naar 2

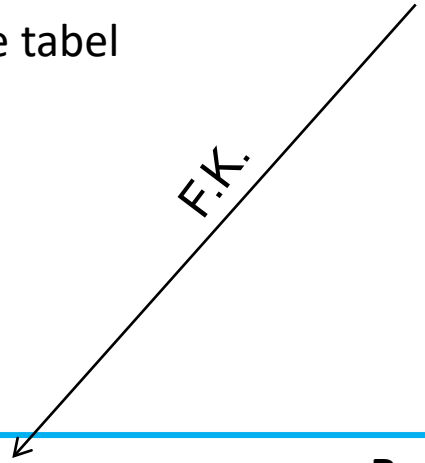
- Stel: tabel X staat in 1NF
- Probleem
  - A is subset / deel van kandidaatsleutel
  - B is niet deel van een kandidaatsleutel (niet-sleutelattribuut)
  - A → B
  - Dus: **B is partieel f.a. van een kandidaatsleutel**
- Upgrade X naar 2NF
  - Verwijder B uit X en verplaats naar nieuwe tabel Y
  - Bronattributen van de partiele f.a. worden de nieuwe primaire sleutel
    - Kopieer ze (A) naar Y
  - Foreign key aanmaken van X.A naar Y.A

Normaliseren

Van normaalvorm 1 naar 2 - voorbeelddata

(titel, genre, fiction) is partieel f.a. van primaire sleutel (isbn, klantid) via (isbn)
Splits probleemattributen af naar nieuwe tabel
Kopieer bronattribuut van partiele f.a. als primaire sleutel naar nieuwe tabel

Boekenwinkel				
isbn	**titel**	**genre**	**fiction**	**klantid**
99-99	Superman	comic book	true	jos1
99-99	Superman	comic book	true	jos1
99-99	Superman	comic book	true	jos1
87-06	Bloembollen	tuin	false	jos1
32-00	Endymion	scifi	true	jef2
87-06	Bloembollen	tuin	false	eve1
12-72	Alien	scifi	true	ilse1



Boekenwinkel	
isbn	**klantid**
99-99	jos1
87-06	jos1
32-00	jef2
87-06	eve1
12-72	ilse1

Boek			
isbn	**titel**	**genre**	**fiction**
99-99	Superman	comic book	true
32-00	Endymion	scifi	true
87-06	Bloembollen	tuin	false
12-72	Alien	scifi	true

Normaliseren

Van normaalvorm 1 naar 2 - voorbeelddata

Het boek ‘Loofbomen’ kan nu wel bijgehouden worden.

Boek			
isbn	titel	genre	fiction
99-99	Superman	comic book	true
32-00	Endymion	scifi	true
87-06	Bloembollen	tuin	false
12-72	Alien	scifi	true
33-87	Loofbomen	tuin	true

Normaliseren

Van normaalvorm 1 naar 2 - voorbeelddata

- Relaties op dit moment

Boek			
isbn	**titel**	**genre**	**fiction**
99-99	Superman	comic book	true
32-00	Endymion	scifi	true
87-06	Bloembollen	tuin	false
12-72	Alien	scifi	true

Aankoop			
isbn	**klantid**	**volgnr**	**betaling**
99-99	jos1	1	10
99-99	jos1	2	5
99-99	jos1	3	5
87-06	jos1	1	23
32-00	jef2	1	10
87-06	eve1	1	18
12-72	ilse1	1	7

Boekenwinkel	
isbn	**klantid**
99-99	jos1
87-06	jos1
32-00	jef2
87-06	eve1
12-72	ilse1

Normaliseren

Van normaalvorm 1 naar 2 - voorbeelddata

- Speciaal geval
  - Als bij normalisatie een relatie A een subset is van een andere relatie B
  - Verwijder relatie A
- In dit voorbeeld
  - Boekenwinkel ⊆ Aankoop
  - Geen toegevoegde waarde!
  - Verwijder Boekenwinkel

Aankoop			
isbn	**klantid**	**volgnr**	**betaling**
99-99	jos1	1	10
99-99	jos1	2	5
99-99	jos1	3	5
87-06	jos1	1	23
32-00	jef2	1	10
87-06	eve1	1	18
12-72	ilse1	1	7

Boekenwinkel	
isbn	**klantid**
99-99	jos1
87-06	jos1
32-00	jef2
87-06	eve1
12-72	ilse1

Normaliseren

Relaties in 2NF

Aankoop			
isbn	**klantid**	**volgnr**	**betaling**
99-99	jos1	1	10
99-99	jos1	2	5
99-99	jos1	3	5
87-06	jos1	1	23
32-00	jef2	1	10
87-06	eve1	1	18
12-72	ilse1	1	7

Boek			
isbn	**titel**	**genre**	**fiction**
99-99	Superman	comic book	true
32-00	Endymion	scifi	true
87-06	Bloembollen	tuin	false
12-72	Alien	scifi	true

Normaliseren

Normaalvorm 3

- Definitie
  - *De tabel is conform 2NF*
  - *Niet-sleutelattributen zijn **niet transitief f.a.** van de kandidaatsleutels*

Normaliseren

Normaalvorm 3

- Wat lost 3NF op t.o.v. 2NF?
  - Redundantie wordt weggewerkt
  - Opslagrestrictie wordt weggewerkt

Boek			
isbn	**titel**	**genre**	**fiction**
99-99	Superman	comic book	true
32-00	Endymion	scifi	true
87-06	Bloembollen	tuin	false
12-72	Alien	scifi	true
~~null~~	~~null~~	~~fantasie~~	~~true~~

Normaliseren

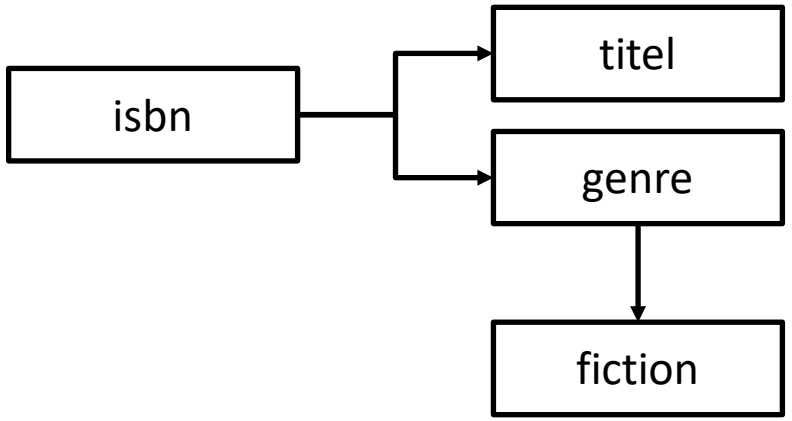
Normaalvorm 3 - voorbeelddata

✓ tabel is conform 2NF (in 1NF, geen part. f.a.)

(fiction) is **transitief f.a.** van een kandidaatsleutel

- (isbn) → (genre)
- (genre) ↛ (isbn)
- (genre) → (fiction)

✗ *Deze tabel is **niet** conform 3NF*



Boek			
isbn	**titel**	**genre**	**fiction**
99-99	Superman	comic book	true
32-00	Endymion	scifi	true
87-06	Bloembollen	tuin	false
12-72	Alien	scifi	true

Normaliseren

Van normaalvorm 2 naar 3

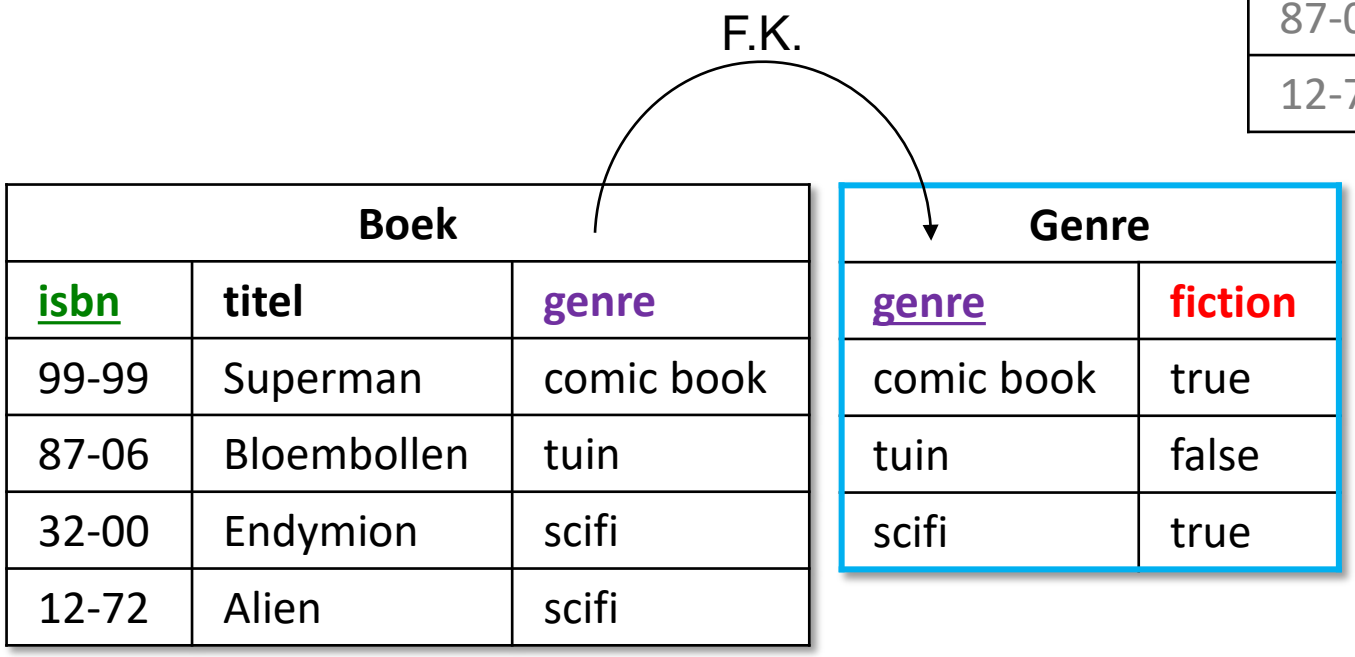
- Stel: tabel X staat in 2NF
- Probleem
  - A is een kandidaatsleutel
  - $A \rightarrow B$, $B \not\rightarrow A$, $B \rightarrow C$
  - Dus: **C is transitief f.a. van A**
- Upgrade X naar 3NF
  - Verwijder C uit X en verplaats naar nieuwe tabel Y
  - Kopieer B naar Y en gebruik als primaire sleutel
  - Foreign key van X.B naar Y.B

Normaliseren

Van normaalvorm 2 naar 3 - voorbeelddata

(fiction) is transitief f.a. van primaire sleutel (isbn) via (genre)
Splits probleemattribuut af naar nieuwe tabel
Kopieer tussenattribuut als primaire sleutel naar nieuwe tabel

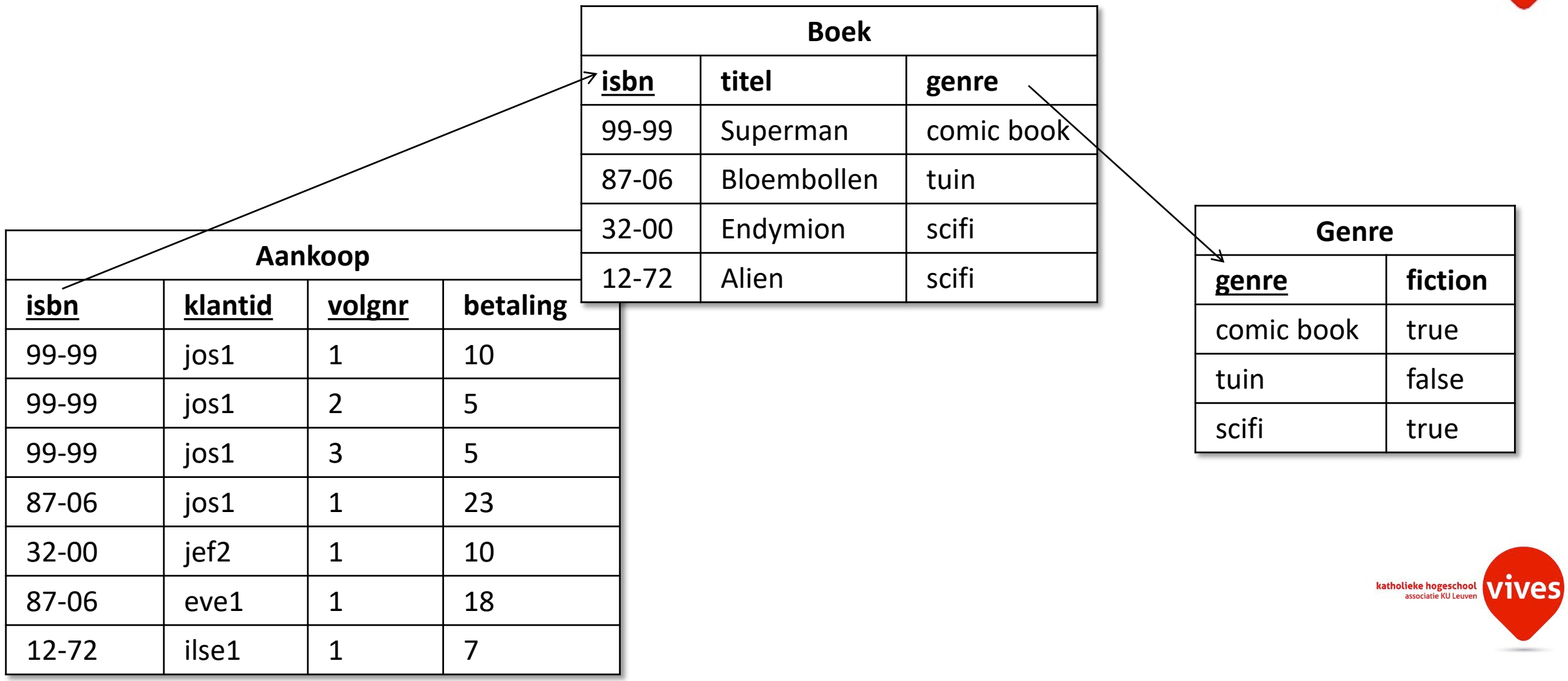
Boek			
isbn	**titel**	**genre**	**fiction**
99-99	Superman	comic book	true
32-00	Endymion	scifi	true
87-06	Bloembollen	tuin	false
12-72	Alien	scifi	true



katholieke hogeschool associatie KU Leuven vives

Normaliseren

NF3 - voorbeelddata



Boek

isbn	titel	genre
99-99	Superman	comic book
87-06	Bloembollen	tuin
32-00	Endymion	scifi
12-72	Alien	scifi

Aankoop

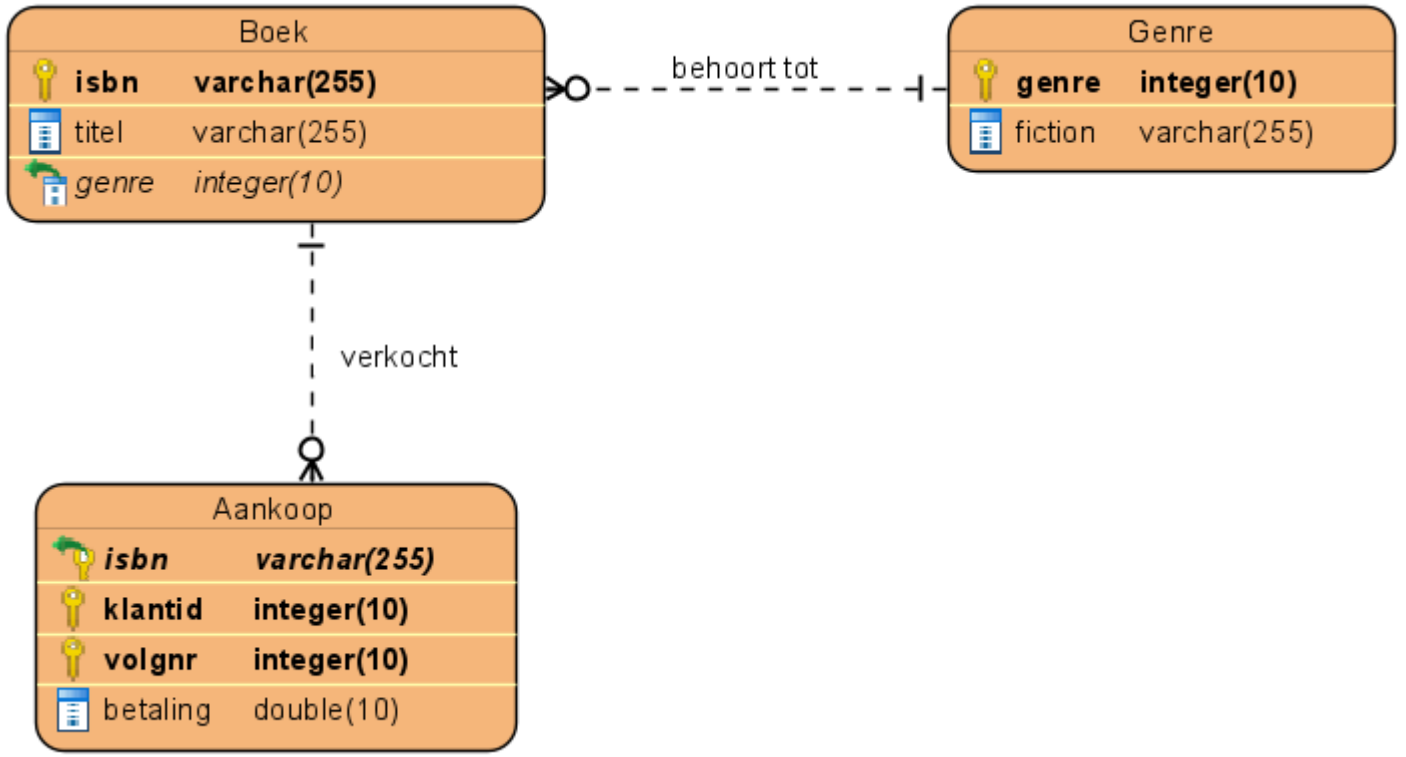
isbn	klantid	volgnr	betaling
99-99	jos1	1	10
99-99	jos1	2	5
99-99	jos1	3	5
87-06	jos1	1	23
32-00	jef2	1	10
87-06	eve1	1	18
12-72	ilse1	1	7

Genre

genre	fiction
comic book	true
tuin	false
scifi	true

Normaliseren

ERD - voorbeelddata



Normaliseren

Eerste hulp bij onderhoudsproblemen

- **Integriteit** bij *inserts:* een klant of boek toevoegen zonder aankoop?
- **Integriteit** bij *deletions*: de laatste aankoop verwijderen ⇨ klant kwijt
- **Redundantie**: info over klant en boek komt *meerdere keren* voor
- **Consistentie**: info boek wijzigen kan enkel correct *door meerdere updates te doen*

Boekenwinkel					
isbn	**titel**	**genre**	**fiction**	**klantid**	**betalingen**
99-99	Superman	comic book	true	jos1	10, 5, 5
87-06	Bloembollen	tuin	false	jos1	23
~~32-00~~	~~Endymion~~	~~scifi~~	~~true~~	~~jef2~~	~~10~~
87-06	Bloembollen	tuin	false	eve1	18
~~12-72~~	~~Alien~~	~~scifi~~	~~true~~	NULL	NULL
NULL	NULL	NULL	NULL	~~eve1~~	NULL

katholieke hogeschool associatie KU Leuven vives

Normaliseren

Proces als oefening

- Gegeven: een tekst, waaruit attributen voorkomen
- Doel: relaties creëren waarin alle attributen zitten
- *Per relatie R doe je dit*
  - Bepaal *alle* kandidaatsleutels van R
  - Check in welke normaalvorm R staat
  - Staan die nog niet in de gevraagde normaalvorm?
    - Identificeer *één* probleem en los dit op
      - Dit resulteert in 1 of meerdere tabellen, waarvoor je het proces herhaalt
    - Herhaal zolang je een relatie hebt die nog niet in de gevraagde NF staat

- *Komt zelden voor: als je twee relaties $A \subseteq B$ hebt, verwijder dan A*

Oefening

Voorbeeld 1: normaliseringsproces

- Opdracht: normaliseer de gegeven relatie naar 3NF
- Attributen worden gegeven, waarvoor betekenis gekend is
- We starten naïef: alle attributen in 1 tabel

? 0NF

? 1NF

? 2NF

? 3NF

Logboek
werknemernummer
naam
geboortedatum
geslacht
nummer departement
naam departement
nummer departementshoofd
nummer directe chef
nummers van de projecten waarvoor hij gewerkt heeft
aantal uren dat hij voor zo'n project gewerkt heeft
namen van de projecten
locaties van de projecten
begindatums van de projecten
einddatums van de projecten

Oefening

Voorbeeld 1: normaliseringsproces

- We hebben momenteel 1 relatie: *Log*
- We voeren het normalisatieproces voor deze relatie uit
  - Bepaal kandidaatsleutels
  - In welke NF?
  - Nog niet in **3NF**? Los op!

? 0NF
? 1NF
? 2NF
? 3NF

Log
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor
ids_project
hours_projects
names_projects
locations_projects
start_dates_projects
end_dates_projects

Oefening

Voorbeeld 1: normaliseringsproces

- Staat in 0NF?
  - ✓ 0NF (tabelvorm)

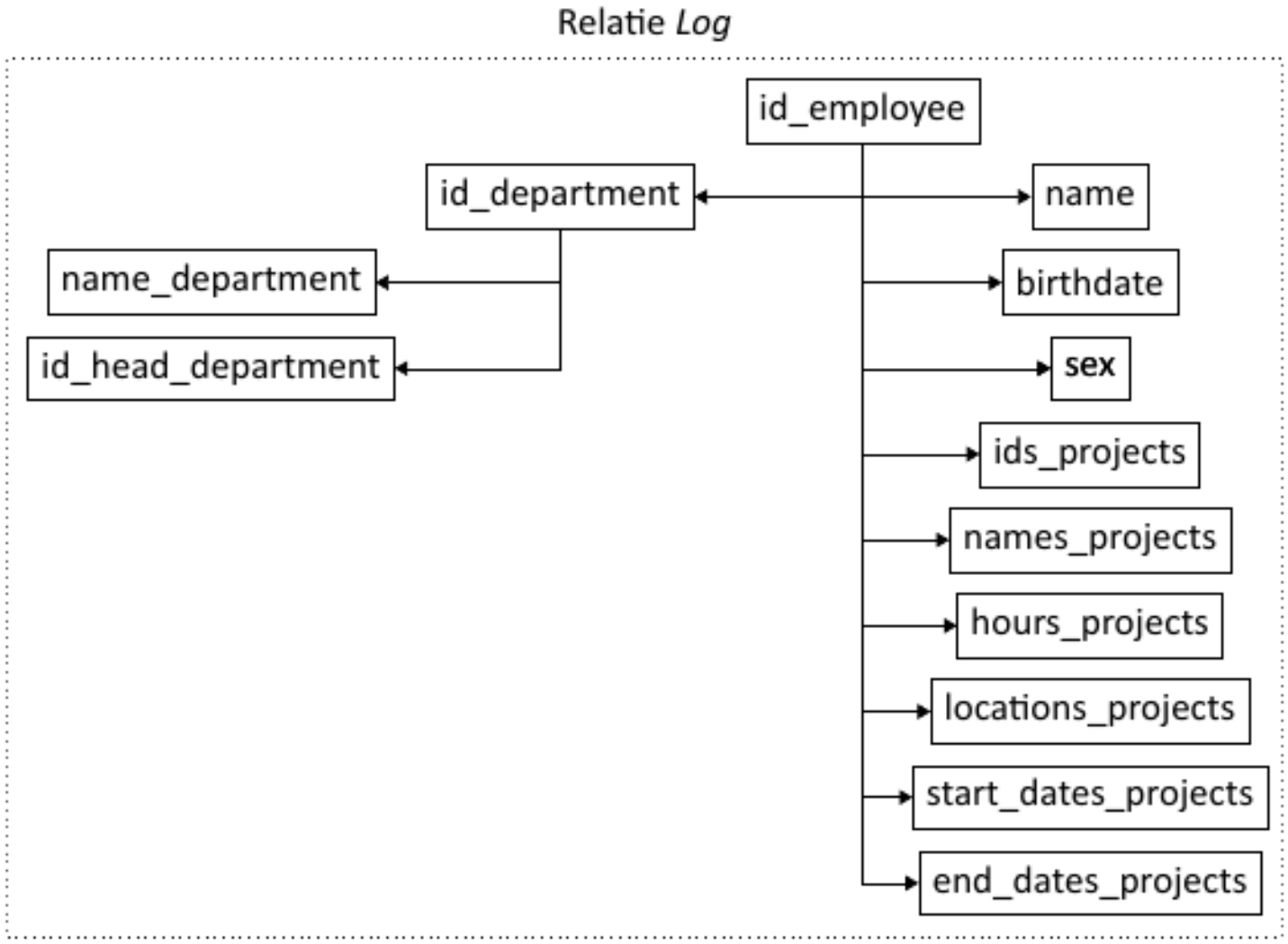
✓ 0NF
? 1NF
? 2NF
? 3NF

Log
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor
ids_project
hours_projects
names_projects
locations_projects
start_dates_projects
end_dates_projects

Oefening

Voorbeeld 1: normaliseringsproces

f.a. bepalen
(kandidaatsleutels bepalen)



Oefening

Voorbeeld 1: normaliseringsproces

1. Kandidaatsleutels?
   - (id_employee)

✓ 0NF
? 1NF
? 2NF
? 3NF

Log
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor
ids_project
hours_projects
names_projects
locations_projects
start_dates_projects
end_dates_projects

Oefening

Voorbeeld 1: normaliseringsproces

1. Kandidaatsleutels?
   - (id_employee)
2. Normaalvorm?

✗ Staat in 1NF?

- Er zijn niet-atomaire waarden!
  - Project-gerelateerde attributen
  - Project-gerelateerd attribuut, maar employee-gerelateerd

✓ 0NF
? 1NF
? 2NF
? 3NF

Log
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor
ids_project
hours_projects
names_projects
locations_projects
start_dates_projects
end_dates_projects

Oefening

Voorbeeld 1: normaliseringsproces

1. Kandidaatsleutels?
   - (id_employee)
2. Normaalvorm?
   - ✗ Staat in 1NF?
     - Er zijn niet-atomaire waarden!
       - Project-gerelateerde attributen
       - Project-gerelateerd attribuut, maar employee-gerelateerd
3. Actie: ga over naar 1NF
   1. Splits niet-atomaire waarden af
   2. Herzie primaire sleutel

✓	0NF
✗	1NF
✗	2NF
✗	3NF

Log
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor
ids_project
hours_projects
names_projects
locations_projects
start_dates_projects
end_dates_projects

Oefening

Voorbeeld 1: normaliseringsproces

4. Attributen aangepast en ontdubbeld

✓ 0NF
? 1NF
? 2NF
? 3NF

Log
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor
id_project
hours_project

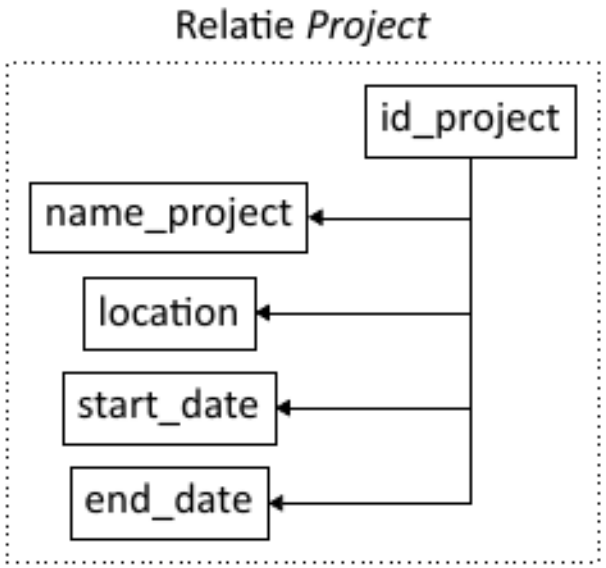
✓ 0NF
? 1NF
? 2NF
? 3NF

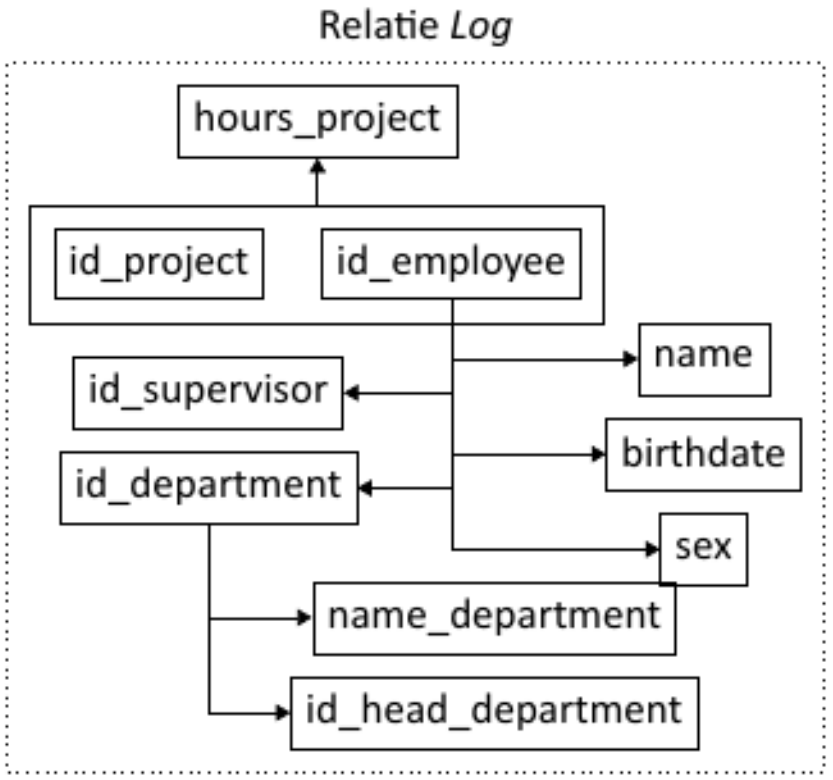
Project
id_project
name
location
start_date
end_date

Oefening

Voorbeeld 1: normaliseringsproces

- f.a. bepalen





Oefening

Voorbeeld 1: normaliseringsproces

4. Attributen aangepast en ontdubbeld
5. Primaire sleutel (id_employee) klopt nu niet meer.
   Pas aan: (id_emplyee, id_project)

✓	0NF
?	1NF
?	2NF
?	3NF

Log
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor
id_project
hours_project

✓	0NF
?	1NF
?	2NF
?	3NF

Project
id_project
name
location
start_date
end_date

Oefening

Voorbeeld 1: normaliseringsproces

4. Attributen aangepast en ontdubbeld
5. Primaire sleutel (id_employee) klopt nu niet meer.
   Pas aan: (id_emplyee, id_project)
6. Normaalvorm?
   – *Log* staat nu in 1NF: enkel atomaire waarden
   – *Project* staat in 1NF

✓	0NF
✓	1NF
?	2NF
?	3NF

Log
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor
id_project
hours_project

✓	0NF
✓	1NF
?	2NF
?	3NF

Project
id_project
name
location
start_date
end_date

Oefening

Voorbeeld 1: normaliseringsproces

- Project in 2NF?
  - 1 kandidaatsleutel: (id_project)
  - ✓ 2NF (enkelvoudige kandidaatsleutel)

✓	0NF
✓	1NF
✗	2NF
?	3NF

Log
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor
id_project
hours_project

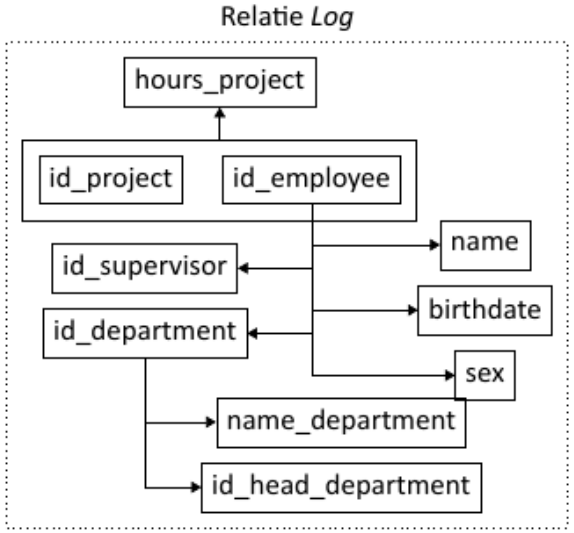
✓	0NF
✓	1NF
?	2NF
?	3NF

Project
id_project
name
location
start_date
end_date

Oefening

Voorbeeld 1: normaliseringsproces

- Project in 2NF?
  - 1 kandidaatsleutel: (id_project)
  - ✓ 2NF (enkelvoudige kandidaatsleutel)
- Log in 2NF?
  - 1 kandidaatsleutel: (id_employee, id_project)
  - ✗ 2NF (partiële f.a. aanwezig)



✓	0NF
✓	1NF
✗	2NF
?	3NF

Log
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor
id_project
hours_project

✓	0NF
✓	1NF
✓	2NF
?	3NF

Project
id_project
name
location
start_date
end_date

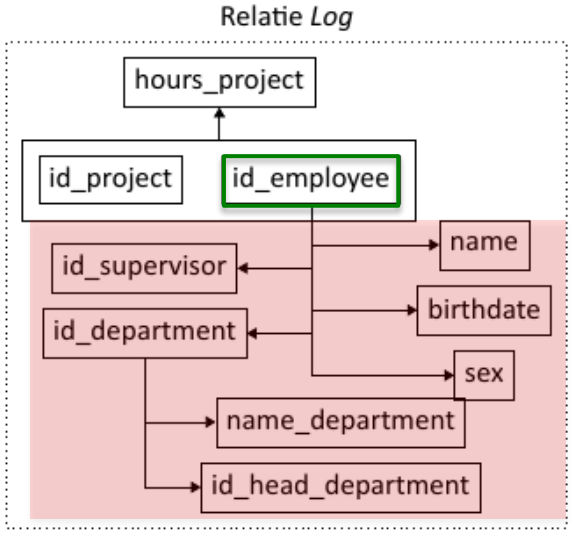
Oefening

Voorbeeld 1: normaliseringsproces

- Normaliseer *Log* naar 2NF
  - Kandidaatsleutel(s)
    - (id_employee, id_project)
  - Partiële f.a. gevonden!
    - (id_employee) → (name, birthdate, sex, id_department, name department, id_head_ departement, id_supervisor)
  - Splits probleemattributen af
    - Kopieer deel van de kandidaatsleutel mee waarvan probleemattributen afhankelijk zijn
    - Dat deel wordt kandidaatsleutel van nieuwe tabel

✓ 0NF
✓ 1NF
✗ 2NF
? 3NF

Log
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor
id_project
hours_project



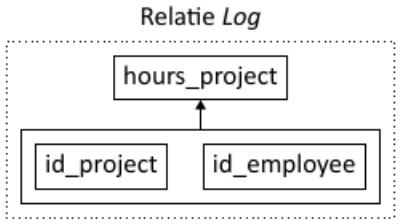
Oefening

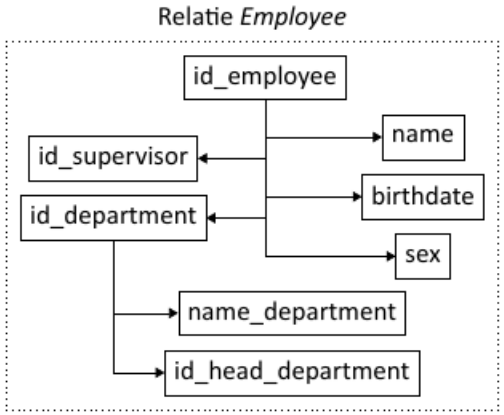
Voorbeeld 1: normaliseringsproces

- Resultaat normalisatie Logboek
  - Twee nieuwe relaties
  - Deze elk door het proces halen
- *Log* in 2NF?
  - ✓ hours_project is **niet f.a. van enkel** id_employee of id_project
- *Employee* in 2NF?
  - ✓ Geen samengestelde kandidaatsleutel

		Log
✓	0NF	
✓	1NF	id_employee
✓	2NF	id_project
?	3NF	hours_project

		Employee
		id_employee
		name
		birthdate
✓	0NF	sex
✓	1NF	id_department
✓	2NF	name_department
?	3NF	id_head_department
		id_supervisor





Oefening

Voorbeeld 1: normaliseringsproces

- Huidige stand van zaken: drie relaties in 2NF
- Check 3NF voor *elk* van deze relaties

✓	0NF
✓	1NF
✓	2NF
?	3NF

Project
id_project
name
location
start_date
end_date

✓	0NF
✓	1NF
✓	2NF
?	3NF

Log
id_employee
id_project
hours_project

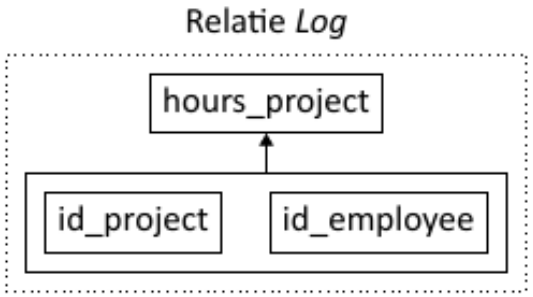
✓	0NF
✓	1NF
✓	2NF
?	3NF

Employee
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor

Oefening

Voorbeeld 1: normaliseringsproces

- Log in 3NF?
  - ✓ geen transitieve f.a.



✓	0NF
✓	1NF
✓	2NF
✓	3NF

Project
id_project
name
location
start_date
end_date

✓	0NF
✓	1NF
✓	2NF
?	3NF

Log
id_employee
id_project
hours_project

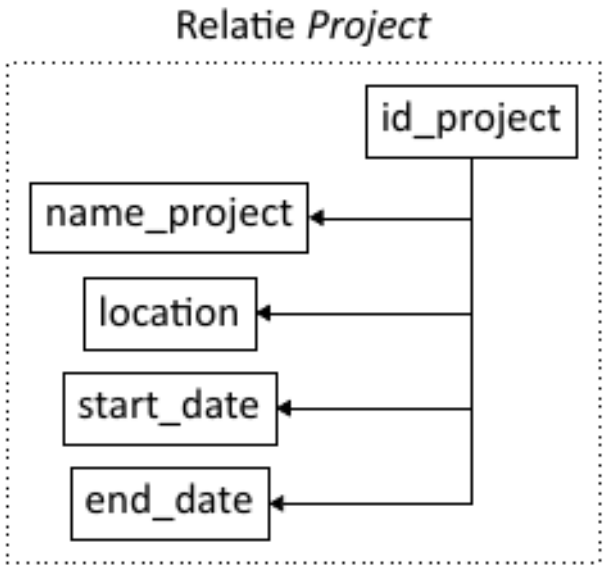
✓	0NF
✓	1NF
✓	2NF
?	3NF

Employee
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor

Oefening

Voorbeeld 1: normaliseringsproces

- Project in 3NF?
  - ✓ geen transitieve f.a.



✓	0NF
✓	1NF
✓	2NF
✓	3NF

Project
id_project
name
location
start_date
end_date

✓	0NF
✓	1NF
✓	2NF
✓	3NF

Log
id_employee
id_project
hours_project

✓	0NF
✓	1NF
✓	2NF
?	3NF

Employee
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor

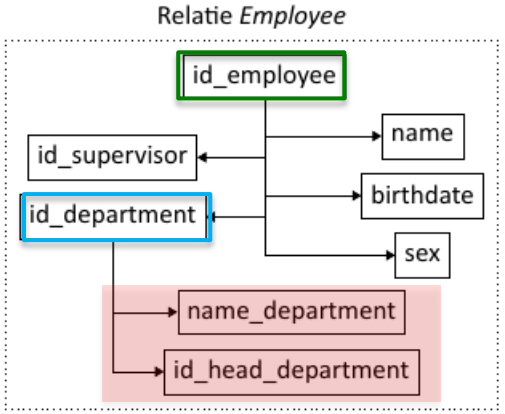
Oefening

Voorbeeld 1: normaliseringsproces

- Employee in 3NF?
  - transitieve f.a.?
    - name_departement
      - ✓ (id_employee) → (id_departement)
      - ✓ (id_ departement) → (name_departement)
      - ✓ (id_ departement) ↛ (id_employee)
    - id_ head_departement
    - ✓ (id_employee) → (id_ departement)
      - ✓ (id_ departement) → (id_ head_departement)
      - ✓ (id_ departement) ↛ (id_employee)
    - ✗ 'Employee' niet in 3NF

✓ 0NF
✓ 1NF
✓ 2NF
✗ 3NF

Employee
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor



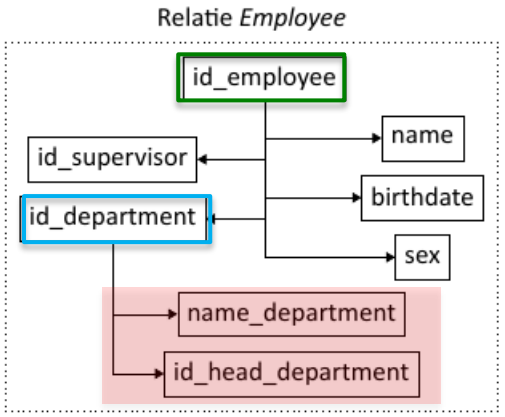
Oefening

Voorbeeld 1: normaliseringsproces

- Splits de probleemattributen af
  - Los de twee transitieve f.a. problemen in één keer op
- Neem gepaste attributen mee als sleutel
  - Transitieve f.a.: A→B→C
  - C: probleemattributen
  - B: neem mee als sleutel

✓ 0NF
✓ 1NF
✓ 2NF
✗ 3NF

Employee
id_employee
name
birthdate
sex
id_department
name_department
id_head_department
id_supervisor



Oefening

Voorbeeld 1: normaliseringsproces

- Resultaat normalisatie *Employee* naar 3NF
  - Twee nieuwe relaties
  - Deze elk door het proces halen
- *Employee* in 3NF?
  - ✓ 2NF (geen samengestelde kandidaatsleutel)
  - ✓ Geen transitieve f.a. meer
- *Departement* in 3NF?
  - ✓ 2NF (geen samengestelde kandidaatsleutel)
  - ✓ Geen transitieve .f.a.

✓ 0NF
✓ 1NF
✓ 2NF
✓ 3NF

Employee
id_employee
name
birthdate
sex
id_department
id_supervisor

✓ 0NF
✓ 1NF
✓ 2NF
✓ 3NF

Department
id_department
name_department
id_head_department

Oefening

Voorbeeld 1: normaliseringsproces

Tussentijdse resultaten

- Check elke relatie op 3NF afzonderlijk
  - Alles staat in 2NF, reeds gechecked
  - Nergens nog een transitieve afhankelijkheid te zien

  ✓ proces compleet: alle relaties staan in 3NF

		Employee
		id_employee
✓	0NF	name
✓	1NF	birthdate
✓	2NF	sex
✓	3NF	id_department
		id_supervisor

		Department
✓	0NF	id_department
✓	1NF	name_department
✓	2NF	id_head_department
✓	3NF	

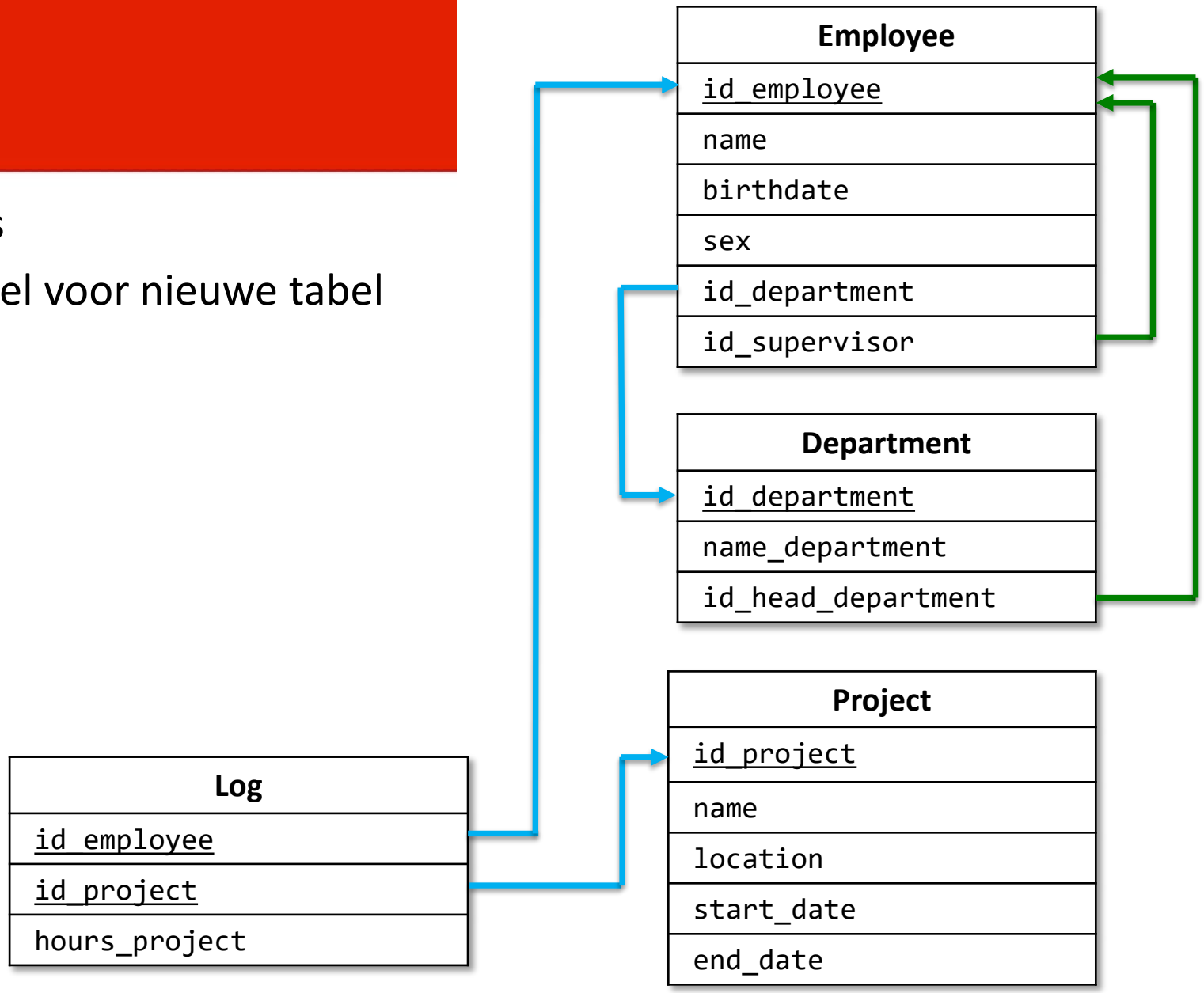
		Log
✓	0NF	id_employee
✓	1NF	id_project
✓	2NF	hours_project
✓	3NF	

		Project
✓	0NF	id_project
✓	1NF	name
✓	2NF	location
✓	3NF	start_date
		end_date

Oefening

Voorbeeld 1: normaliseringsproces

- Foreign keys volgen uit het process
  - Bij kopie van attributen als sleutel voor nieuwe tabel
  - Uit betekenis van de attributen



Oefening

Voorbeeld 1: ERD representatie