

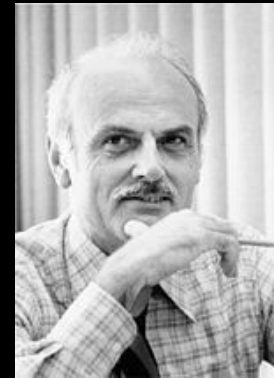
Normalisatie

Methodisch modelleren

- Bottom up
 - Normaliseren (gegevensanalyse)
- Top down
 - Entity relationship (informatie-analyse)

Relationeel model

- Ted Codd (1923 – 2003)
- “A Relational Model of Data for Large Shared Data Banks” (1970)
- Langzame overgang van hiërarchisch DMBS naar RDBMS
 - IBM System R met SEQUEL (later SQL)
- Wiskundige basis samen met Chris Date
 - Normaliseren



Tabel vs. Relatie

- Een RDBMS slaat data op in vorm van relaties
- Een relatie is een genormaliseerde, tweedimensionale tabel

Tabel, maar geen relatie

Course	Course name	Level	Module	Module name	Status	Credits
B47	Applied computer science	Bachelor	B471 B472 B473 B474 B475	Programming 1 Data Essentials Desktop OS Programming 2 Data Advanced	Basic	8
					Medium	11
B94	Data Science	Master	B941 B942 B471	Data Analytics Cloud Services Programming 1	Advanced	14
					Basic	8

- Spreadsheetformaat (geen relationele tabel)
- Herhalende deelverzameling: 1 course name met N module names

Relationele tabel (= relatie)

graad (degree)

Kardinaliteit van de tabel

KlantNr	Voornaam	Achternaam	Adres	Plaats
134	Desiderius	Erasmus	Nijverheidskaai 170	Brussel
135	Georges	Lemaître	Bondgenotenlaan 100	Leuven
136	Gabrielle	Petit	Quai Notre Dame 47	Doornik
Etc.	Etc.	Etc.	Etc.	Etc.

cel (atomaire waarde)

kolom = veld

rij = record = tuple

Relationele tabel

- Relatie is een verzameling van records
- Volgorde niet relevant voor het model

Relationele tabel

- Een RDBMS slaat data op in vorm van relaties
- Een relatie is een tweedimensionale tabel met volgende eigenschappen
 - Kolommen bevatten gegevens over attributen van de entiteit
 - Rijen bevatten gegevens over één entiteit (= instantie)
 - Cellen van een tabel bevatten enkelvoudige gegevens
 - Alle items in een kolom zijn van dezelfde soort
 - Elke kolom heeft een unieke naam
 - De volgorde van de kolommen is van geen belang
 - De volgorde van de rijen is van geen belang

Relationele tabel

- Niet alle tabellen zijn relaties, maar de termen worden vaak door elkaar gebruikt
 - Tabel / rij / kolom = bestand / record / veld = relatie / tuple / attribuut

Algemeen

- Gegevens
 - Elementaire gegevens
 - Samengestelde gegevens
 - Procesgegevens
- Instantie of voorkomen (occurrence)
 - Verzameling van waarden voor 1 tupel
 - Wordt geïdentificeerd door een waarde van een sleutel (key)

Soorten sleutels

- Een sleutel of key bestaat uit één of meerdere kolommen van een relatie, die een rij identificeren
- Een sleutel identificeert exact 1 rij
- Een samengestelde sleutel is een sleutel die meerdere attributen bevat
- Een relatie heeft één primary key (primaire sleutel), de andere sleutels worden kandidaatsleutels genoemd

Primary key

- Voor het voorstellen van rijen in een tabel
- Voor het organiseren van tabelopslag
- Voor het genereren van indexen

Niet-genormaliseerde tabellen

- Informatie wordt meerdere keren herhaald (redundantie)
 - Plaatsverlies
 - Integriteit niet gegarandeerd
 - Wijzigingen → programmeren
- Bij invoeren nieuwe rij: lege waarden (NULL)
- Bij verwijderen van informatie worden ook andere nuttige gegevens verwijderd

Adviseur	Tel. adviseur	Student	Afdeling	Afdelingshoofd
Jansen	221.2345	Park	Boekhouding	Groen
Kwekeboom	223.4444	Steen	Informatie	Meester
Rozeboom	281.3944	Johansen	Boekhouding	Groen
Rozeboom	281.3944	Gonzales	Boekhouding	Groen
Jansen	221.2345	Riks	Boekhouding	Groen



Normaliseren

- Normalisatie elimineert wijzigingsanomalieën
 - Verwijderanomalie
 - Invoeganomalie
 - Update-anomalie
- Anomalieën wegwerken door relaties op te delen in relaties met elk hun eigen thema
- Splitsen van relaties veroorzaakt meestal referentiële integriteitvoorwaarden

Normaliseren

- Data organiseren in tabellen zodat
 - de data-integriteit bewaard wordt
 - aanpassingen (insert, update, delete) eenvoudig zijn (geen redudantie)
 - rijen in een tabel te identificeren zijn a.d.h.v. de primary key
 - bij elkaar horende rijen gerelateerd zijn via een foreign key

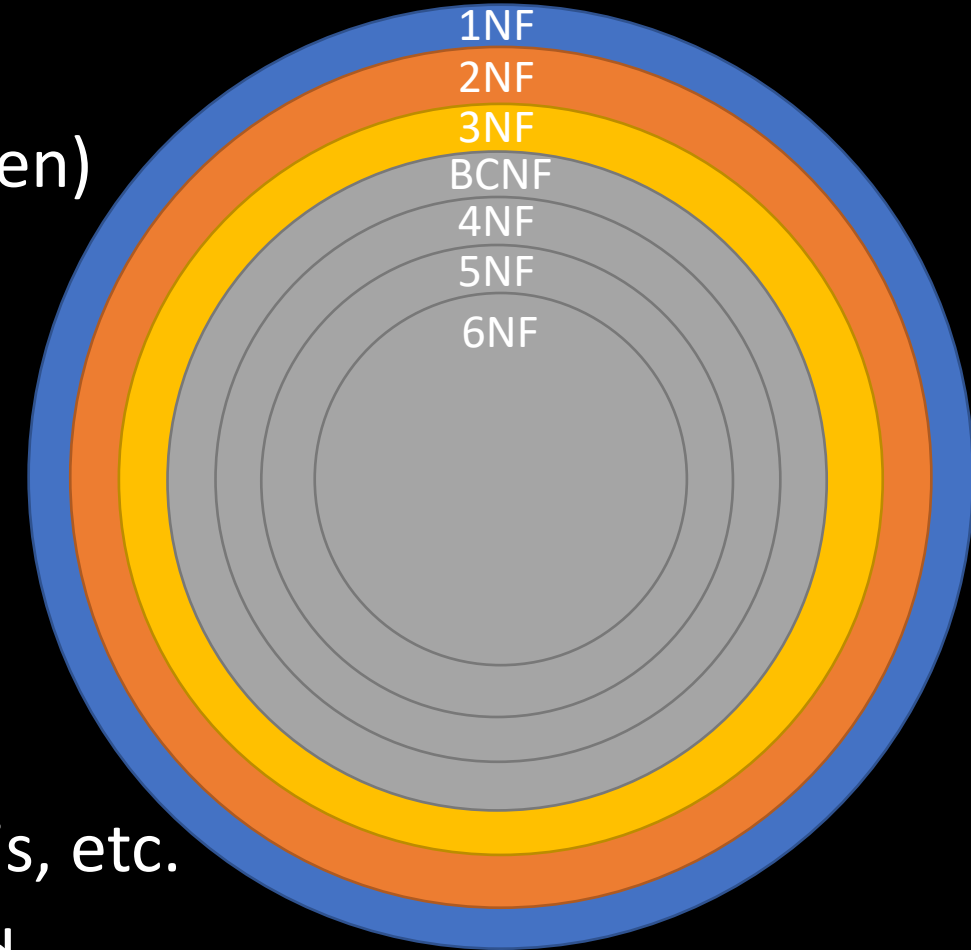
Normaliseren als techniek

- Omvat een aantal stappen (= normaalvormen)

- 1NF
- 2NF
- 3NF
- BCNF
- 4NF
- 5NF
- 6NF

Meestal voldoende
“relatie is genormaliseerd”

- 2NF veronderstelt dat de dataset al in 1NF is, etc.
- Centraal begrip: functionele afhankelijkheid



Functionele afhankelijkheid

- $A \rightarrow B$
- B is functioneel afhankelijk van A
- A is de determinant
- Voorbeeld:
 - personeelsnummer \rightarrow naam
 - afdelingnummer \rightarrow {afdelingnaam, afdelingplaats}

PersNr	Naam	AfdelingNr	AfdelingNaam	AfdelingPlaats
134	Desiderius Erasmus	100	Opleiding	Brussel
135	Georges Lemaître	200	Analyse	Leuven
136	Gabrielle Petit	300	Informatica	Brussel
137	Pieter De Somer	100	Opleiding	Brussel

Transitieve afhankelijkheden

- $A \rightarrow B \rightarrow C$
- Afhankelijkheid van een sleutel “via” een ander attribuut
- Transitiviteitsregel: Als $A \rightarrow B$ en $B \rightarrow C$, dan geldt $A \rightarrow C$

Transitieve afhankelijkheden

- Transitiviteitsregel: Als $A \rightarrow B$ en $B \rightarrow C$, dan geldt $A \rightarrow C$
- Voorbeeld:
 - personeelsnummer \rightarrow afdelingnummer
 - afdelingnummer \rightarrow {afdelingnaam, afdelingplaats}
 - personeelsnummer \rightarrow {afdelingnaam, afdelingplaats}
 - personeelsnummer \rightarrow afdelingnummer \rightarrow {afdelingnaam, afdelingplaats}

PersNr	Naam	AfdelingNr	AfdelingNaam	AfdelingPlaats
134	Desiderius Erasmus	100	Opleiding	Brussel
135	Georges Lemaître	200	Analyse	Leuven
136	Gabrielle Petit	300	Informatica	Brussel
137	Pieter De Somer	100	Opleiding	Brussel

EXTRA Functionele afhankelijkheid: axiomas

- Reflexiviteitsregel: Als Y een subset is van X , dan $X \rightarrow Y$
 - $Y \subseteq X \Rightarrow X \rightarrow Y$
- Uitbreidingsregel: Als $X \rightarrow Y$, dan $XZ \rightarrow YZ$
 - $\{X \rightarrow Y\} \mid = XZ \rightarrow YZ$
- Transitiviteitsregel: Als $X \rightarrow Y$ en $Y \rightarrow Z$, dan $X \rightarrow Z$
 - $\{X \rightarrow Y, Y \rightarrow Z\} \mid = X \rightarrow Z$

Normalisatie: Voorafgaand (0NF)

- 0NF = gegevens in niet-genormaliseerde vorm
- Opsplitsen van samengestelde gegevens in elementaire gegevens
- Verwijderen van procesgegevens als deze kunnen berekend worden a.d.h.v. beschikbare informatie

Normalisatie: 1NF

- Een relatie is in 1NF als alle attributen atomisch zijn (= er zijn geen herhalende deelverzamelingen)
- Werkwijze
 - elimineer eventueel dubbele kolommen uit de relatie
 - kies de sleutel zodat er geen herhalende deelverzamelingen zijn (kijk hiervoor naar FA)

Normalisatie: 1NF

PartNr	PartName	Stock	FactuurNr	KlantNr	Code	Aantal
142	Potlood	600	17123	4629	1	15
			19198	3751	3	17
830	Gom	100	17123	4629	1	13
			19198	3751	3	21
			18763	5003	5	12
779	Pen	330	16765	3751	3	13

- Kies PartNr en FactuurNr als primaire sleutel
 $\{partnr, factuurnr\} \rightarrow \{partname, stock, klantnr, code, aantal\}$
- (partnr, factuurnr, partname, stock, klantnr, code, aantal)

Normalisatie: 1NF

FactuurNr	PartNr	PartName	Stock	KlantNr	Code	Aantal
17123	142	Potlood	600	4629	1	15
19198	142	Potlood	600	3751	3	17
17123	830	Gom	100	4629	1	13
19198	830	Gom	100	3751	3	21
18763	830	Gom	100	5003	5	12
16765	779	Pen	330	3751	3	13



Normalisatie: 2NF

- Een relatie is in 2NF als ze in 1NF is en elk attribuut functioneel afhankelijk is van de volledige sleutel
- Stappenplan
 - Zorg dat de relatie in 1NF is
 - Vind attributen die niet functioneel afhankelijk zijn van de volledige sleutel
 - Vorm een aparte groep voor ieder deel van de sleutel waarvan deze attributen afhankelijk zijn
 - Neem in iedere groep de attributen met bijhorende sleuteldeel op
 - Verwijder deze attributen uit de oorspronkelijke groep

Normalisatie: 2NF

1NF

FactuurNr	PartNr	PartName	Stock	KlantNr	Code	Aantal
17123	142	Potlood	600	4629	1	15
19198	142	Potlood	600	3751	3	17
17123	830	Gom	100	4629	1	13
19198	830	Gom	100	3751	3	21
18763	830	Gom	100	5003	5	12
16765	779	Pen	330	3751	3	13

2NF

<u>FactuurNr</u>	KlantNr	Code
17123	4629	1
19198	3751	3
18763	5003	5
16765	3751	3

<u>PartNr</u>	PartName	Stock
142	Potlood	600
830	Gom	100
779	Pen	330

<u>FactuurNr</u>	<u>PartNr</u>	Aantal
17123	142	15
19198	142	17
17123	830	13
19198	830	21
18763	830	12
16765	779	13

Normalisatie: 3NF

- Een relatie is in 3NF als ze in 2NF is en geen transitieve afhankelijkheden heeft
- Stappenplan
 - Zorg dat de relatie 2NF is
 - Vind attributen die FA zijn van een ander attribuut dat geen kandidaatsleutel is
 - Vorm een aparte groep voor alle attributen waarvan andere attributen FA zijn
 - Neem in iedere groep de attributen met de bijhorende sleutel op
 - Verwijder de attributen uit de oorspronkelijke groep

Normalisatie: 3NF

2NF

<u>FactuurNr</u>	KlantNr	Code
17123	4629	1
19198	3751	3
18763	5003	5
16765	3751	3

3NF

<u>KlantNr</u>	Code
4629	1
3751	3
5003	5

<u>FactuurNr</u>	KlantNr
17123	4629
19198	3751
18763	5003
16765	3751

Normalisatie: 3NF

<u>KlantNr</u>	Code
4629	1
3751	3
5003	5

<u>FactuurNr</u>	KlantNr
17123	4629
19198	3751
18763	5003
16765	3751

<u>FactuurNr</u>	<u>PartNr</u>	Aantal
17123	142	15
19198	142	17
17123	830	13
19198	830	21
18763	830	12
16765	779	13

<u>PartNr</u>	PartName	Stock
142	Potlood	600
830	Gom	100
779	Pen	330

Normalisatie: BCNF

- Een relatie is in BCNF als ze in 3NF is en als alle determinanten een kandidaatsleutel zijn
- Stappenplan
 - Zorg dat de relatie 3NF is
 - Identificeer FA en elke kandidaatsleutel
 - Is er een FA die een determinant heeft die geen kandidaatsleutel is?
 - Verplaats de kolommen van deze FA naar een nieuwe relatie
 - Maak de determinant van deze FA tot sleutel van de nieuwe relatie
 - Laat een kopie van de determinant in de originele relatie staan
 - Maakt een referentiële integriteitvoorwaarde die de originele en de nieuwe relatie met elkaar koppelt (foreign key)
 - Herhaal vorige stap tot elke determinant van elke relatie een kandidaatsleutel is

Normalisatie: BCNF

- Gegeven: tabel in 3NF
- Meerdere studenten kunnen inschrijven voor een cursus
- Elke student kan inschrijven voor meerdere cursussen
- Voor elke cursus zijn meerdere docenten aangeduid
- Elke docent begeleidt slechts één cursus
- docent → cursus

<u>StudentNr</u>	<u>Cursus</u>	Docent
15	Java	Ruben
15	Data	Wim
16	Java	David
17	C#	Jan
18	Java	Ruben

Normalisatie: BCNF

3NF

<u>StudentNr</u>	<u>Cursus</u>	Docent
15	Java	Ruben
15	Data	Wim
16	Java	David
17	C#	Jan
18	Java	Ruben

BCNF

<u>StudentNr</u>	<u>Cursus</u>	Docent
15	Java	Ruben
15	Data	Wim
16	Java	David
17	C#	Jan
18	Java	Ruben

<u>Docent</u>	<u>Cursus</u>
Ruben	Java
Wim	Data
David	Java
Jan	C#

Normalisatie: 4NF

- Een relatie is in 4NF als ze in BCNF is en als de relatie geen meerwaardige afhankelijkheden heeft
- Een attribuut C is meerwaardig afhankelijk van A indien voor elke waarde van A geldt dat bij elke waarde van B dezelfde verzameling van waarden van C bestaat

Normalisatie: 4NF

- Gegeven: Tabel in BCNF
- Meerwaardige afhankelijkheid:
Voor elke waarde van persnr en
voor elke waarde van taalkennis
komen steeds alle waarden van
programmeerkennis voor

<u>PERSNR</u>	TAALKENNIS	PROGRAMMEERKENNIS
15	Nederlands	C++
15	Nederlands	SQL
15	Duits	C++
15	Duits	SQL

Normalisatie: 4NF

BCNF

<u>PERSNR</u>	TAALKENNIS	PROGRAMMEERKENNIS
15	Nederlands	C++
15	Nederlands	SQL
15	Duits	C++
15	Duits	SQL

4NF

PersNr	Programmeerkennis
15	C++
15	SQL

PersNr	Taalkennis
15	Nederlands
15	Duits

Normalisatie: overzicht

- Elke tabel met gegevens die voldoet aan de definitie van een relatie is in 1NF
- Een relatie is in 2NF als alle attributen die niet in de sleutel zijn opgenomen, afhankelijk zijn van de gehele sleutel (geen gedeeltelijke afhankelijkheid)
- Een relatie is in 3NF indien ze in 2NF is en geen transitieve afhankelijkheid kent
- Een relatie is in BCNF als elke determinant een kandidaatsleutel is
- Een relatie is in 4NF als ze in BCNF staat en geen meerwaardige afhankelijkheden kent

Relaties synthetiseren

- Gegeven: de FA tussen de attributen
- Vraag: Welke relaties moeten gevormd worden?
- Voorbeeld: A en B zijn twee attributen
 - Indien $A \rightarrow B$ en $B \rightarrow A$
 - A en B hebben één-op-één relatie
 - Indien $A \rightarrow B$, maar B niet $\rightarrow A$
 - A en B hebben meer-op-één relatie
 - Indien A niet $\rightarrow B$ en B niet $\rightarrow A$
 - A en B hebben een meer-op-meer relatie

1-op-1 relatie tussen attributen

- Attributen met een één-op-één relatie moeten minstens in één tabel samen voorkomen
- Noem deze relatie R en de attributen A en B:
 - A of B moet de sleutel van R zijn
 - Een attribuut kan worden toegevoegd aan R als het functioneel afhankelijk is van A of B
 - Een attribuut dat niet functioneel afhankelijk is van A of B kan niet aan R worden toegevoegd
 - A en B moeten beide voorkomen in R, maar horen niet samen in andere relaties voor te komen
 - A of B moeten consequent worden gebruikt om het attributenpaar in andere relaties dan R te vertegenwoordigen

1-op-N relatie tussen attributen

- Attributen met een één-op-meer relatie mogen in dezelfde relatie voorkomen
- Veronderstel dat A bepaalt B in tabel R
 - A moet de sleutel van R zijn
 - Een attribuut mag aan R worden toegevoegd als het functioneel afhankelijk is van A
 - Een attribuut dat niet door A is bepaald, mag niet aan R worden toegevoegd

N-op-M relatie tussen attributen

- Attributen met een meer-op-meer relatie mogen in dezelfde relatie voorkomen
- Stel dat de attributen A en B beide in relatie R voorkomen
 - De sleutel van R moet (A, B) zijn
 - Een attribuut mag aan R worden toegevoegd als het door (A, B) wordt bepaald
 - Een attribuut mag niet aan R worden toegevoegd als het niet door (A, B) wordt bepaald
 - Als door toevoeging van een nieuw attribuut, C, de sleutel wordt uitgebreid tot (A, B, C), wordt het onderwerp van de relatie gewijzigd
 - Ofwel behoort C niet tot R ofwel moet de naam van R worden gewijzigd aan het nieuwe onderwerp

Gedenormaliseerde ontwerpen

- Een gedegenormaliseerd ontwerp is beter als een genormaliseerd ontwerp onnatuurlijk, onhandig of niet performant is
- Voorbeeld
 - Genormaliseerde relatie
 - KLANT(klantnummer, klantnaam, postcode)
 - POSTCODES(postcode, gemeente)
 - Gedenormaliseerde relatie
 - KLANT(klantnummer, klantnaam, postcode, gemeente)