



**Syllabus**

# **Databases**

---

**Mileto Di Marco**  
**Kristien Roels**  
**Bart Soete**

## Inhoudsopgave

<b>H12 - Normalisatie .....</b>	<b>3</b>
1 Wat is normalisatie?.....	3
2 Het normalisatieproces .....	4
2.1 De nulde normaalvorm (0NF).....	4
2.2 De eerste normaalvorm (1NF).....	4
2.3 De tweede normaalvorm (2NF).....	6
2.4 De derde normaalvorm (3NF).....	6
3 Voorbeeld.....	8
3.1 Nulde normaalvorm (0NF of UNF = Unnormalised Form).....	8
3.2 Eerste Normaalvorm (1NF).....	8
3.3 Tweede normaalvorm(2NF).....	9
3.4 Derde normaalvorm (3NF) .....	9

# H12 - Normalisatie

---

## 1 Wat is normalisatie?

Normalisatie is een techniek die ontwikkeld werd om ervoor te zorgen dat de opgemaakte gegevensstructuur efficiënt is. Het komt voor uit het werk van E.F. Codd en werd reeds in de jaren '70 geïntroduceerd.

De techniek van databasennormalisatie wordt in het bijzonder gebruikt in relationele databanken. Het geeft de methode weer om de logische structuur van gegevens in een informatiesysteem te beschrijven.

De voordelen van normalisatie zijn:

- Vrijwaring van ongewenste afhankelijkheid bij toevoegen, verwijderen en wijzigen van gegevens.
- Herstructurering van de gegevens wordt bij introductie van nieuwe gegevens geminimaliseerd. Deze gegevensonafhankelijkheid heeft tot gevolg dat de database weinig of geen effect zal hebben op reeds bestaande programma's die de database gebruiken.
- Er komen geen onnatuurlijke gedwongenheden in de gegevensstructuur voor.

We kunnen stellen dat normalisatie voornamelijk wordt gebruikt met volgende doeleinden:

- Vermijden van redundante (of overbodige) gegevens.
- Data wordt logisch gestockeerd in de database (data-afhankelijkheden zijn logisch)



## 2 Het normalisatieproces

Normalisatie omvat (in beginsel) 3 stappen die in volgorde dienen uitgevoerd te worden. Uitvoeren van de 2<sup>de</sup> stap impliceert dat de 1<sup>ste</sup> stap volledig werd uitgevoerd en hetzelfde geldt voor de volgende stappen. Na beëindiging van de 3<sup>de</sup> stap staan de gegevens zowel in eerste, tweede als in derde normaalvorm. Data die niet gestructureerd is, bevindt zich in de nulde normaalvorm.

De normaalvormen worden afgekort weergegeven door ONF, 1NF, 2NF en 3NF.

De meeste applicaties gebruiken databases die genormaliseerd zijn tot de 3<sup>de</sup> normaalvorm, de 4<sup>de</sup> en 5<sup>de</sup> normaalvorm zie je zelden, vandaar dat in dit cursusdeel tot de 3<sup>de</sup> normaalvorm wordt besproken.

We werken hier als voorbeeld met een orderboekje waar enkele bestellingen in staan genoteerd:

Datum : 20/04/2019
Klant : Guy Taar, Boomsesteenweg 35, 8000 Brugge
Bestelling :
1 Televisie (987456) met prijs 649,9 Euro / stuk
100 Pluggen (456123) met prijs 0,05 Euro / Stuk
10 Verdeeldozen (654987) met prijs 1,05 Euro / Stuk
Totaalprijs : 660,4 Euro
-----
Datum : 20/04/2019
Klant : Jan Janssen, Reversiestraat 75, 9000 Gent
Bestelling :
1 Kabelgoot (154325) met prijs 7,5 Euro / Stuk
1 Universele afstandsbediening (253535) met prijs 27,25 Euro / Stuk
1 Klavier (125835) met prijs 10,65 Euro/ Stuk
Totaalprijs : 45,4 Euro
-----
Datum : 21/04/2019
Klant : Willem Dewitte, Anjerstraat 5, 8000 Brugge
Bestelling :
20 Pluggen (456123) met prijs 0.05 Euro / Stuk
1 Grafische kaart (125387) met prijs 730 Euro / Stuk
Totaalprijs : 731 Euro

### 2.1 De nulde normaalvorm (ONF)

Ieder ongestructureerd gegevensbestand is in de nulde normaalvorm (ONF) ofwel niet genormaliseerd. Gegevens van verschillende soorten kunnen op elke regel voorkomen, waardoor deze niet in kolommen kunnen worden opgedeeld en de data niet te doorzoeken is. Een standaard tekstbestand is hier een voorbeeld van.

### 2.2 De eerste normaalvorm (1NF)

Een relatie in de eerste normaalvorm bevat geen herhaalde velden (repeating groups)

De formele definitie luidt:

*“Een relatie is in de eerste normaalvorm, als het de eigenschap heeft dat geen van de domeinen elementen bevat, die zelf een verzameling met 2 of meer elementen zijn. “*

Elke **tabel** met gegevens die voldoen aan de definitie van een relatie is in de eerste normaalvorm (1NF). Wanneer gegevens aan een relatie voldoen zijn ze dus reeds genormaliseerd. Elke tabel wordt tevens voorzien van een **primaire sleutel**, die ervoor zorgt dat de datarij of record in de gegevenstabel uniek geïdentificeerd is.

Tabellen voldoen aan een aantal regels:

- Elk attribuut is atomair, en bevat dus 1 enkele waarde, en is van hetzelfde datatype.
- Ieder kolomnaam in een tabel heeft een unieke naam.
- Alle attributen blijven constant in de tijd.
- De volgorde van de rijen van de tabel (records) is onbelangrijk.

We starten met de gegevens op te sommen die mogelijk in onze database zouden moeten komen, in tabelvorm, waarbij we ieder kolom een unieke naam geven en komen tot volgend resultaat :

#### Tabel Orders:

Datum	Voor naam	Familienaam	Straat	Nummer	Postnummer	Gemeente	ProductID	Productnaam	Productprijs	Aantal
20/04/2019	Guy	Taar	Boomsesteenweg	38	8000	Brugge	987456	Televisie	649.9	1
20/04/2019	Guy	Taar	Boomsesteenweg	38	8000	Brugge	456123	Plug	0.05	100
20/04/2019	Guy	Taar	Boomsesteenweg	38	8000	Brugge	654987	Verdeelddoos	1.05	10
20/04/2019	Jan	Janssen	Reversiestraat	75	9000	Gent	154325	Kabelgoot	7.5	1
20/04/2019	Jan	Janssen	Reversiestraat	75	9000	Gent	253535	Univ. Afst.bed.	27.5	1
20/04/2019	Jan	Janssen	Reversiestraat	75	9000	Gent	125835	Klavier	10.65	1
21/04/2019	Willem	Dewitte	Anjerstraat	5	8000	Brugge	456123	Plug	0.05	20
21/04/2019	Willem	Dewitte	Anjerstraat	5	8000	Brugge	125387	Grafische kaart	730	1

De gegevens in bovenstaand voorbeeld bevat herhaalde velden omdat één order meerdere orderregels kan gaan bevatten en splitsen hierdoor de tabel Orders in Orders en OrderDetail.

We voorzien iedere deeltabel van een **primaire sleutel** en voorzien een *referentiesleutel* in de andere tabel(len) om de relaties vast te leggen en komen zo tot volgende vereenvoudiging:

#### Tabel Orders:

OrderID	Datum	Voor naam	Familienaam	Straat	Nummer	Postnummer	Gemeente
1	20/04/2019	Guy	Taar	Boomsesteenweg	38	8000	Brugge
2	20/04/2019	Jan	Janssen	Reversiestraat	75	9000	Gent
3	21/04/2019	Willem	Dewitte	Anjerstraat	5	8000	Brugge

#### Tabel OrderDetail:

OrderID	ProductID	Productnaam	Productprijs	Aantal
1	987456	Televisie	649.9	1
1	456123	Plug	0.05	100
1	654987	Verdeelddoos	1.05	10
2	154325	Kabelgoot	7.5	1
2	253535	Univ. Afst.bed.	27.5	1
2	125835	Klavier	10.65	1
3	456123	Plug	0.05	20
3	125387	Grafische kaart	730	1

Er zijn echter wat problemen met 1NF. De informatie over een product kan uitsluitend in het systeem worden opgenomen, wanneer er een order voor bestaat. Hierdoor zijn er problemen met de 3 basisoperaties die we op een database willen uitvoeren, nl. het toevoegen, wijzigen en verwijderen :

- Informatie over een product kan niet toegevoegd worden, tenzij er een order voor dit product bestaat.
- Wanneer een orderregel verwijderd wordt, gaat informatie over het product verloren.
- In elke orderregel voor een bepaald product wordt de productinformatie herhaald, wanneer dit product al is opgenomen. Voor één wijziging van bijvoorbeeld productprijs, moeten bijgevolg evenveel records gewijzigd worden als dat er orders zijn voor dit product.

### 2.3 De tweede normaalvorm (2NF)

In een relatie die in de 2<sup>de</sup> normaalvorm staat, is elk veld of attribuut afhankelijk van de gehele sleutel en niet van slechts een gedeelte van de sleutel. Een tabel in de 2<sup>de</sup> normaalvorm is eveneens in de 1<sup>ste</sup> normaalvorm.

Deze stap in het normalisatieproces is uitsluitend nodig in relaties met samengestelde sleutels.

De formele definitie luidt:

*“Een relatie R is in de 2<sup>de</sup> normaalvorm wanneer het in de eerste normaalvorm staat en elk niet- primair attribuut van R volledig functioneel afhankelijk is van elke kandidaat sleutel van R.”*

We kunnen dit nu gaan toepassen op bovenstaand voorbeeld in 1<sup>ste</sup> normaalvorm. We zien in de tabel OrderDetail dat ‘Aantal’ afhankelijk is van de combinatie ‘OrderID’ en ‘ProductID’. De gehele sleutel is nodig om deze beide velden uniek te identificeren. Bij een ander order kunnen deze gegevens namelijk totaal andere waarden bevatten, wanneer hetzelfde product bijvoorbeeld in een andere hoeveelheid wordt besteld.

‘Productnaam’ en ‘Productprijs’ zijn beide afhankelijk van ‘ProductID’ maar niet van ‘OrderID’. Voor een volgend order waar terug dit product zal besteld worden zullen deze waarden van omschrijving en prijs terug dezelfde zijn (wanneer we ervan uitgaan dat de prijs niet wijzigt).

We splitsen bijgevolg de tabel OrderDetail in tabellen OrderDetail en Product. Voor de tabel Product kunnen we ProductID als primaire sleutel gaan gebruiken.

**Tabel Product:**

ProductID	Productnaam	Productprijs
987456	Televisie	649.9
456123	Plug	0.05
654987	Verdeelddoos	1.05
154325	Kabelgoot	7.5
253535	Univ. Afst.bed.	27.5
125835	Klavier	10.65
125387	Grafische kaart	730

**Tabel OrderDetail:**

OrderID	ProductID	Aantal
1	987456	1
1	456123	100
1	654987	10
2	154325	1
2	253535	1
2	125835	1
3	456123	20
3	125387	1

Er zijn echter nog problemen bij deze 2<sup>de</sup> normaalvorm. De relatie met ordernummer in de tabel Orders vertoont hetzelfde probleem met de Klantgegevens als we hierboven konden vaststellen met Productgegevens. Hiervoor passen we de 3<sup>de</sup> normaalvorm toe.

### 2.4 De derde normaalvorm (3NF)

In een relatie die in de derde normaalvorm staat, is elk veld uitsluitend afhankelijk van de sleutel en niet afhankelijk van enig ander veld in de relatie. Een record in de 3<sup>de</sup> normaalvorm is eveneens in de 2<sup>de</sup> normaalvorm.

De formele definitie luidt:

*“Een relatie R is in de 3<sup>de</sup> normaalvorm, wanneer deze in de 2<sup>de</sup> normaalvorm staat en elk niet-primair attribuut van R niet transitief afhankelijk is van een willekeurige kandidaat sleutel van R.”*

Wanneer dit toegepast wordt komen we uiteindelijk tot 4 tabellen, nl. Order, OrderDetail, Klant en Product.

#### Tabel Klant:

KlantID	Voornaam	Familienaam	Straat	Nummer	Postnummer	Gemeente
1	Guy	Taar	Boomsesteenweg	38	8000	Brugge
2	Jan	Janssen	Reversiestraat	75	9000	Gent
3	Willem	Dewitte	Anjerstraat	5	8000	Brugge

#### Tabel Orders:

OrderID	Datum	KlantID
1	20/04/2019	1
2	20/04/2019	2
3	21/04/2019	3

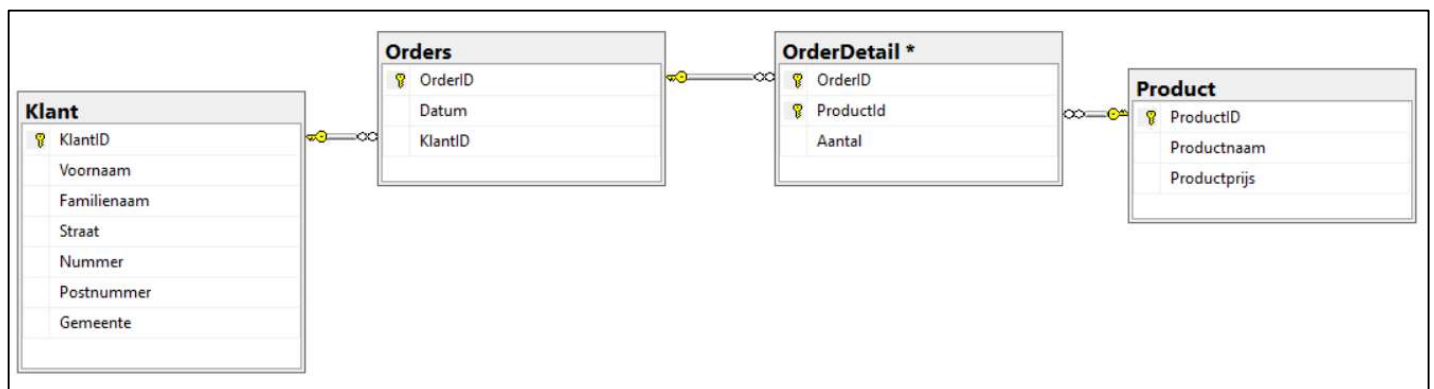
#### Tabel Product:

ProductID	Productnaam	Productprijs
987456	Televisie	649.9
456123	Plug	0.05
654987	Verdeeldoos	1.05
154325	Kabelgoot	7.5
253535	Univ. Afst.bed.	27.5
125835	Klavier	10.65
125387	Grafische kaart	730

#### Tabel OrderDetail:

OrderID	ProductID	Aantal
1	987456	1
1	456123	100
1	654987	10
2	154325	1
2	253535	1
2	125835	1
3	456123	20
3	125387	1

Hieronder vind je hiervoor het databasediagram:



### 3 Voorbeeld

Onderstaand een uitwerking voor Projectwerk.

#### 3.1 Nulde normaalvorm (ONF of UNF = Unnormalised Form)

Tabel Projecten (**ProjectID**, Projecttitel, Budget, WerknemerID, Werknemer, FunctieID, Functie, Weddeschaal)

##### Projecten

ProjectID	Projecttitel	Budget	WerknemerID	Werknemer	FunctieID	Functie	Weddeschaal
AB123	Normalisatie	23500	S01	Jan Jansen	N1	Directie	800
			S04	Guy Taar	N3	Administratie	620
			S02	An Vannevel	N3	Administratie	620
AB268	Testen	15000	S03	Hans Genten	N1	Directie	800
			S01	Jan Jansen	N1	Directie	800

#### 3.2 Eerste Normaalvorm (1NF)

Tabel Projecten wordt opgesplitst in 2 tabellen :

- Tabel Projecten (**ProjectID**, Projecttitel, Budget)
- Tabel ProjectWerknemers (**ProjectID**, **WerknemerID**, Werknemer, FunctieID, Functie, Weddeschaal)

##### Projecten

ProjectID	Projecttitel	Budget
AB123	Normalisatie	23500
AB268	Testen	15000

##### ProjectWerknemers

ProjectID	WerknemerID	Werknemer	FunctieID	Functie	Weddeschaal
AB123	S01	Jan Jansen	N1	Directie	800
AB123	S04	Guy Taar	N3	Administratie	620
AB123	S02	An Vannevel	N3	Administratie	620
AB268	S03	Hans Genten	N1	Directie	800
AB268	S01	Jan Jansen	N1	Directie	800



### 3.3 Tweede normaalvorm(2NF)

Tabel ProjectWerknemers wordt opgesplitst in :

- Tabel ProjectWerknemers (**ProjectID**, **WerknemerID**)
- Tabel Werknemers (**WerknemerID**, Werknemer, Functie, Weddeschaal)

We krijgen 3 tabellen :

#### Projecten

<b>ProjectID</b>	Projecttitel	Budget
AB123	Normalisatie	23500
AB268	Testen	15000

#### ProjectWerknemers

<b>ProjectID</b>	<b>WerknemerID</b>
AB123	S01
AB123	S04
AB123	S02
AB268	S03
AB268	S01

#### Werknemers

<b>WerknemerID</b>	Werknemer	FunctieID	Functie	Weddeschaal
S01	Jan Jansen	N1	Directie	800
S04	Guy Taar	N3	Administratie	620
S02	An Vannevel	N3	Administratie	620
S03	Hans Genten	N1	Directie	800

### 3.4 Derde normaalvorm (3NF)

Tabel Werknemers wordt opgesplitst in :

- Tabel Werknemers (**WerknemerID**, Werknemer, *FunctieID*)
- Tabel Functies (**FunctieID**, Functie, Weddeschaal)

We krijgen dus als eindresultaat 4 tabellen :

#### Projecten

<b>ProjectID</b>	Projecttitel	Budget
AB123	Normalisatie	23500
AB268	Testen	15000

#### ProjectWerknemers

<b>ProjectID</b>	<b>WerknemerID</b>
AB123	S01
AB123	S04
AB123	S02
AB268	S03
AB268	S01

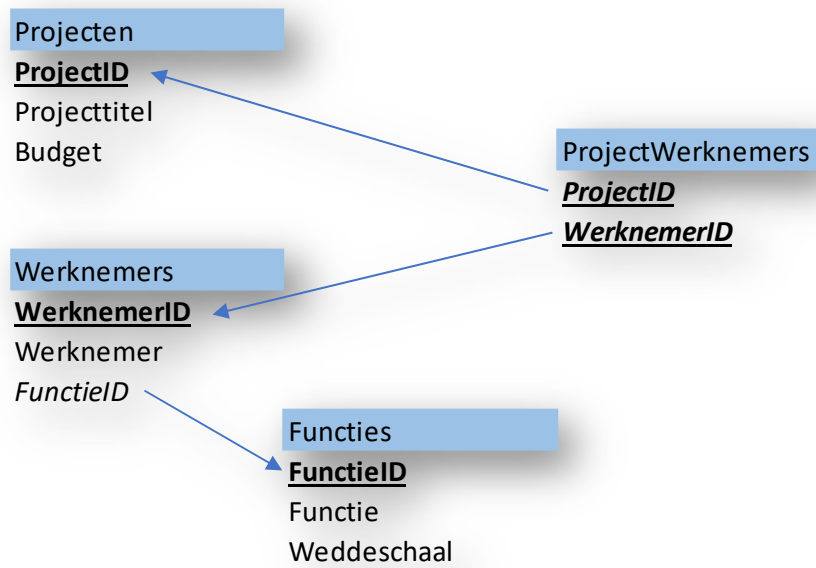
#### Werknemers

<b>WerknemerID</b>	Werknemer	FunctieID
S01	Jan Jansen	N1
S04	Guy Taar	N3
S02	An Vannevel	N3
S03	Hans Genten	N1

#### Functies

<b>FunctieID</b>	Functie	Weddeschaal
N1	Directie	800
N3	Administratie	620

Een andere voorstelling van de bekomen datastructuur :



Meer informatie over normalisatie vind je op : <http://nl.wikipedia.org/wiki/Databasenormalisatie>