

# *DATABANKEN*

*C# programmeur*

# DATABANKEN – Planning

Vrij VM 16/10/2020	Normalisatie	Ontwerp database, tables ontwerpen / aanmaken
Vrij NM 16/10/2020	Normalisatie	relaties, indexen, unieke ID
Di VM 20/10/2020	Queries	DML: Select query, Joins
Di NM 20/10/2020	Queries	DML: Update en delete queries,
Do VM 22/10/2020	Queries	DML: Subqueries, single row functions
Do NM 22/10/2020	Werken met data	DDL
Vrij VM 23/10/2020	Werken met data	DDL
Vrij NM 23/10/2020	T SQL	T-Sql meest gebruikte functies
Di VM 27/10/2020	Stored procedures	Aanmaken, wijzigen en verwijderen
Di NM 27/10/2020	Advanced topics	Query performance; Index strategieën
Do VM 29/10/2020	Opdracht databanken	20 pt voor dagelijks werk
Do NM 29/10/2020	Examen Databanken	40pt EX (Theorie) + 40pt EX (Praktijk)

# DATABANKEN – INHOUD DEZE LES

## **Introductie**

- Gegevens versus informatie
- Soorten informatie
- Databanksystemen - DBMS

## **Relationele databanken**

- Wat is een relationele databank
- Terminologie:
  - entiteiten, attributen, relaties, tabellen,  
rijen (records), kolommen, PK, FK

## **Normalisatie (1NF, 2NF, 3NF)**

leren. durven. doen.



*DATABANKEN*

# *Introductie*

# gegevens vs informatie

**GEGEVENS:**

**10**

**14**

**15**

---

**INFORMATIE:**

Brussel	februari	maart	april
Gemiddelde temperatuur	<b>10</b>	<b>14</b>	<b>15</b>

# samengestelde gegevens

= verzameling van elementaire gegevens die bij elkaar horen

voorbeeld: **adres**

**adres** = (straat, huisnr, postnr, woonplaats)

## berekende gegevens

- worden afgeleid uit andere gegevens
- worden niet opgenomen in een database

voorbeeld: **totaalprijs**

**totaalprijs** = eenheidsprijs x aantal

# gegevens verzameling

bevat een aantal gegevens die logisch bij elkaar horen

voorbeeld: **Een kasboek**

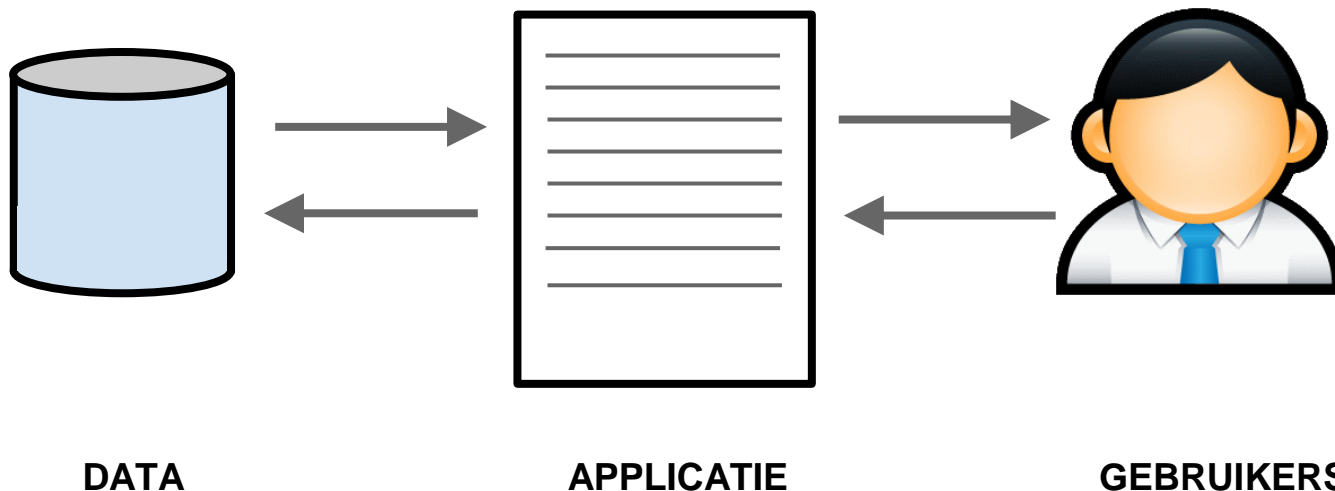
Een winkelier(ster) houdt zijn/haar kosten en inkomsten bij in **een kasboek**. De kasboek is hier een gegevensverzameling.



# database

Wordt gebruikt om gegevensverzamelingen te bewaren.

Een **database** is een gecentraliseerde, gestructureerde verzameling van gegevens die op een computer bewaard wordt.



# OEFENING KLANTENBESTAND

**Download de volgende bestanden van Github**

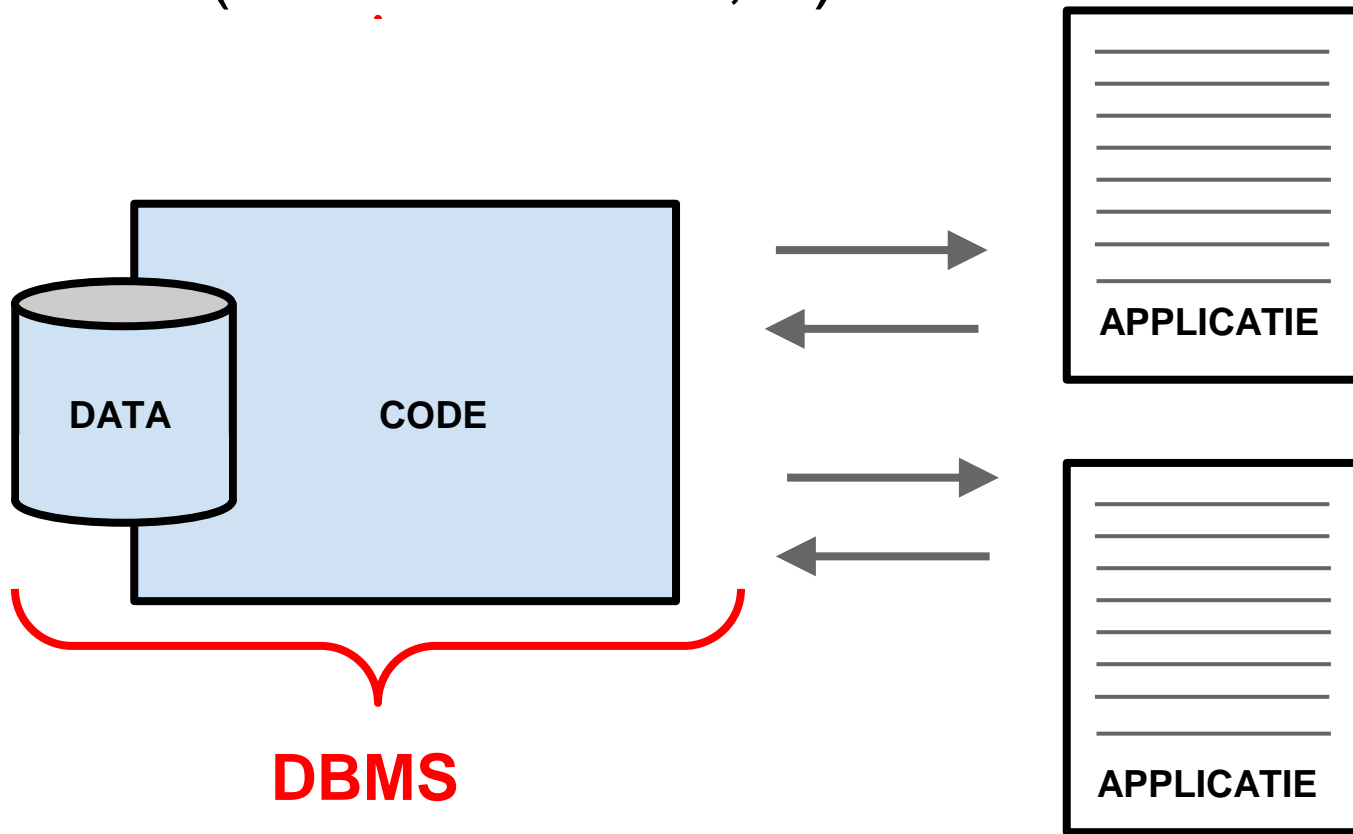
[https://github.com/CSharpSyntraWest/LESSEN\\_DATABANK](https://github.com/CSharpSyntraWest/LESSEN_DATABANK)

Onder Les 1:

- OEFENING\_KLANTENGEGEVENS.docx
- Oefening\_KlantenBestand.xlsx

# Database Management System (DBMS)

= Data + Software die de data aanspreekt en beheert (bv Ms Access,...)



# Vereisten voor een DBMS

- Simultaan gebruik van gegevens
- Beveiliging van toegang
- Beveiliging van gegevens
- Gedistribueerde gegevens (= gegevens zijn verdeeld over sub-databases maar worden centraal beheerd)
- Backup mogelijkheid van gegevens

# Voorbeelden van verschillende DBMS

- **Relationele databankSysteem(RDBMS)**

= DBMS gebaseerd op relationeel model

Bv: Microsoft SQL Server, Microsoft Access, Oracle, and MySql,...

Gebruiken **SQL (Structured Query Language)** om databank-gegevens op te halen en te manipuleren

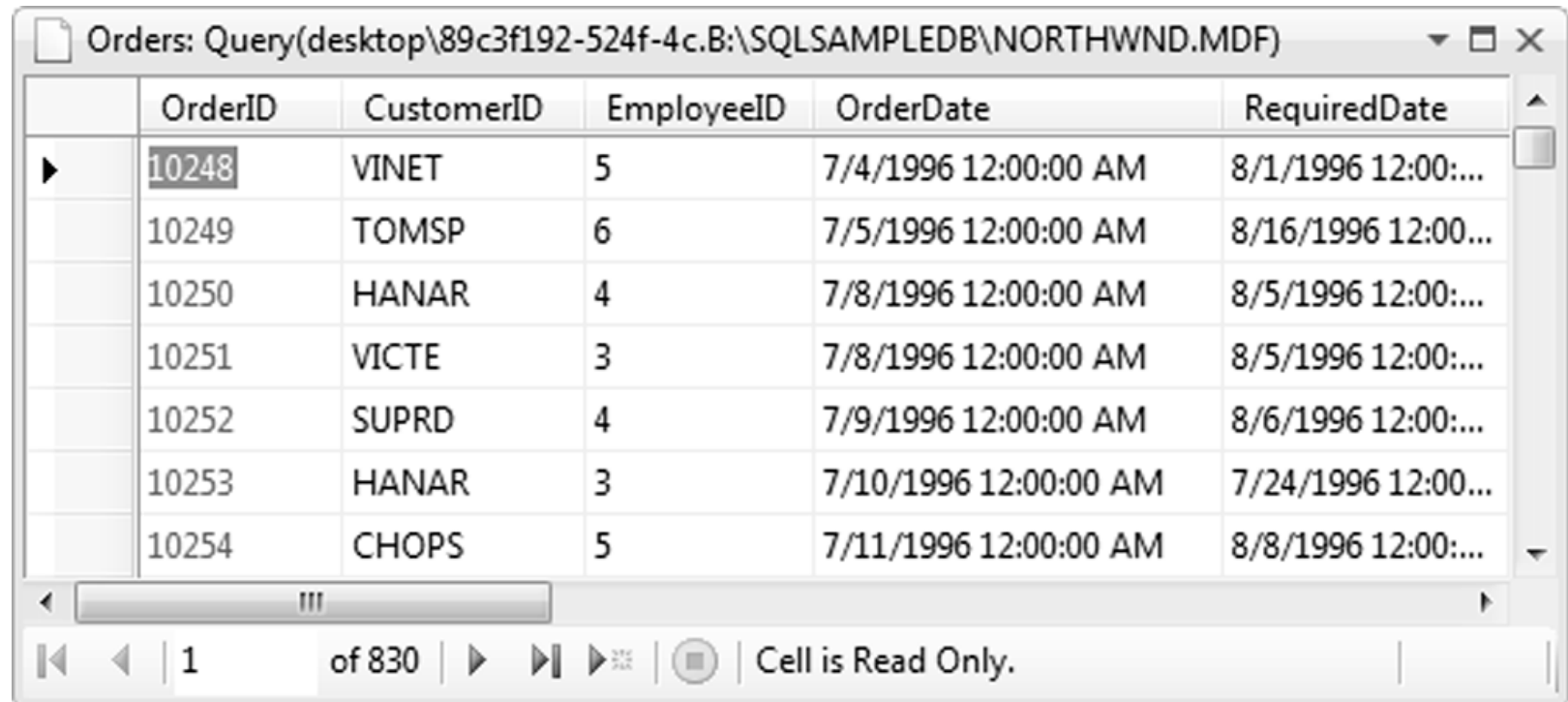
- Objectgeoriënteerde databanken
- Hierarchische databanken
- XML databanken

*DATABANKEN*

# Relationele databases

# Relationele Databanken

- Een relationele databank organiseert gegevens in 2-dimensionale tabellen die uit kolommen en rijen bestaan:
- Bv MS Access:



Orders: Query(desktop\89c3f192-524f-4c.B:\SQLSAMPLEDB\NORTHWND.MDF)

	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate
▶	10248	VINET	5	7/4/1996 12:00:00 AM	8/1/1996 12:00:...
	10249	TOMSP	6	7/5/1996 12:00:00 AM	8/16/1996 12:00...
	10250	HANAR	4	7/8/1996 12:00:00 AM	8/5/1996 12:00:...
	10251	VICTE	3	7/8/1996 12:00:00 AM	8/5/1996 12:00:...
	10252	SUPRD	4	7/9/1996 12:00:00 AM	8/6/1996 12:00:...
	10253	HANAR	3	7/10/1996 12:00:00 AM	7/24/1996 12:00...
	10254	CHOPS	5	7/11/1996 12:00:00 AM	8/8/1996 12:00:...

1 of 830 | Cell is Read Only.

# Voorbeeld TABLE Product

ProductID	Product	Enkcal	Eiwit	KH	Suikers	Vet
1	Brood, bruin	242	7	45,6	2,9	3,6
2	Tahin sesampasta	576	22	6	0	51,6
3	Melk, halfvolle	46	3,3	4,8	4,8	1,5
4	Cacaopoeder gezoet	385	4	87	78	2,5
5	Selderij, bleek	19	0,6	3,4	0	0,2
6	Mandarijn	42	0,8	9,5	9,5	0
7	Chili Sin Carne 1 portie	62	3,8	7,8	3,7	1,8
8	Room, 8% vet	107	2,9	5,9	5,1	8
9	Wijn, rood 12°	68	0,1	0,2	0,2	0
10	Rijst, bruin, gekookt	141	2,6	30,2	0	1,1
11	Linzen, gekookt	88	8,3	12,7	0,3	0,5
12	Wijn, rood 12°	67	0	0,3	0,3	0
13	Yoghurt, magere	52	4,6	6,4	6,4	0,8
14	Cashewnoten	575	18,8	22,5	0	46,3
15	Kiwi	56	1	12,4	8,5	0,2



# Relationele databases

- Voor een gebruiker (beheerder) bestaat een relationele databank uit een **verzameling van tabellen** (en niets anders dan tabellen)
- In een relationele databank worden **entiteiten** bijgehouden (*cursist, cursus*) en hun **attributen** (Voornaam, geboortedatum) en **relaties tussen die entiteiten** (*cursist Jan is ingeschreven voor cursus databanken*)
- Zowel entiteiten, attributen als relaties worden bijgehouden in tabellen

# Relationele databanken

**Cursustabel**

id	titel
1	databanken
2	SQL

**Cursisttabel**

id	naam
1	Jan
2	Rita

**Inschrijvingtabel**

id	cursistid	cursusid
1	1	1
2	2	1
3	1	2

Cursist Jan is ingeschreven voor de cursussen “databanken” en “SQL”. Cursiste Rita is ingeschreven voor de cursus “databanken”

# Metadata en data

## METADATA

= namen van tabellen,  
kolomnamen, datatype van  
elke kolom, security

bv:

**Cursus:** id (int, uniek) en titel (max 20 karakters)

**Cursist:** id (int, uniek) en naam (max 25 karakters)

**Inschrijving:** id (int, uniek),  
cursusid (int gekoppeld aan id in cursus), cursistid (int gekoppeld aan id in cursist)

**Iedereen** mag alle **gegevens** lezen. **Cursusgegevens** mogen alleen **gewijzigd** worden door **cursusbeheerders**.

## DATA

1	databanken
2	SQL

1	Jan
2	Rita

1	1	1
2	2	1
3	1	2

# Tabellen

- Tabellen bestaan uit rijen (**records**) en kolommen (**velden**)
- Elke kolom in een tabel heeft een **unieke naam** (in die tabel)
- Een kolom bevat een bepaald **datatype**: int, float, tekst,...
- Records in een relationele databank hebben **geen volgorde**

# De primaire sleutel (Primary Key of PK)

Elke rij (record) in een tabel moet aangesproken worden via een sleutel

Moet **uniek** zijn voor elke record

- Uniek veld aanwezig? → **primaire sleutel**
- Combinatie van velden uniek? → **gecombineerde primaire sleutel**
- Andere gevallen: een **extra uniek veld** toevoegen. (dikwijls met autonummering)

# De Foreign Key (FK)

= een attribuut in de tabel die verwijst naar een andere tabel waar deze attribuut tevens voorkomt en de primary key vormt van die andere tabel.

**Voorbeeld:** Een klantnr in tabel facturen verwijst naar het klantnr in de tabel klanten. Het klantnr van de tabel klanten is de PK van de tabel.

# Entiteiten

= elke persoon, plaats, gebeurtenis, concept, enz... waarover de gebruiker gegevens wil bijhouden. -> worden omgezet tot tabellen

Voorbeelden van entiteiten:

Product

Klant

Leverancier

Bestelling

Factuur

# Attributen

Elke entiteit heeft één of meerdere kenmerken of attributen. -> worden kolommen in tabel

Enkele voorbeelden van attributen

Enkele attributen van entiteit **Klant**:

VoorNaam

FamilieNaam

Email

Enkele attributen van entiteit **Product**:

Naam

Beschrijving

...



# Relaties

Relaties geven het verband weer tussen entiteiten.

- **1:1 relaties (één op één relaties)** Elke instantie van een bepaalde entiteit komt maximaal met één instantie van de andere entiteit overeen.
- **1:N relaties (één op veel relaties)** Voor elke entiteit zijn er meerdere instanties van de andere entiteit mogelijk
- **N:N relaties (veel op veel relaties)** Voor elke entiteit zijn er meerdere instanties van de andere entiteit en vice versa

# Hoe gegevens verzamelen ?

- 1. Doel bepalen:** *Wat komt in de database, wat niet?  
Zo beperkt mogelijk houden.*
- 2. Bronnen bekijken:** *Hoe is de situatie nu?*
- 3. Goed luisteren**
- 4. Inventariseren:** *Breng eisen en wensen in kaart*
- 5. Analyseren:** *Werk je inventaris volledig uit*
- 6. Terugkoppelen:** *Betrek de klant regelmatig terug  
in het denkproces*

# OEFENING MULTIPLE CHOICE Db

**Download het volgende bestand van Github**

- OEFENING DATABASE MULTIPLE CHOICE.docx

leren. durven. doen.



## *DATABANKEN*

# Normalisatie

# Data anomalieën

<u>BookId</u>	BookName	CategoryId	CategoryName
1	Cooking Light	1001	Cooking
2	Prophecy	1002	Mystery & Thriller
3	Shift	1003	Business
4	The Confession	1002	Mystery & Thriller

## Insert anomalie

- Je kan geen nieuwe gegevens invoegen in de tabel vanwege een 'unrelated dependency'

## Delete Anomalie

- 1 gegeven weggeven geeft een onwenselijk verlies van andere gegevens

## Update Anomalie

- Aanpassing van een enkele gegevenswaarde vereist een aanpassing van meerdere rijen

# Databank normalisatie

Om voorgaande data-anomalieën te vermijden,  
Gaaf men relationele databanken normaliseren.

## Databank Normalisatie=

Het proces dat garandeert dat een databaseontwerp vrij is  
van problemen (data-anomalieën) die kunnen leiden tot:

- **verlies van gegevens**
- **integriteit van gegevens**

Eenvoudig gezegd: vermijd verlies van gegevens en  
zorg dat er geen rommel in de database kan komen!

# Databank ontwerp: 3 eerste normalizaties

Een goed ontwerp beantwoordt aan de drie normaalvormen:

**1ste normaalvorm:** velden mogen alleen enkelvoudige attributen bevatten

**2de normaalvorm:** in een tabel met een samengestelde primaire sleutel moet elk attribuut afhankelijk zijn van de volledige primaire sleutel

**3de normaalvorm:** attributen mogen alleen afhankelijk zijn van de primaire sleutel, niet van de andere attributen

# Eerste Normaal Vorm (1NF)

Om aan de eerste normaalvorm te voldoen, mag geen niet op dezelfde kolom en rij dezelfde rij van een tabel meerdere waarden bevatten staan

Bv: de volgende tabel is NIET in 1 NF want kolom PhoneNumber bevat op rij 2 en 3 meerdere waarden:

<u>Id</u>	FirstName	LastName	PhoneNumber
1	Jane	Doe	(503) 555-6874
2	John	Doe	(509) 555-7969, (509) 555-7970
3.	Howard	Steel	(604) 555-3392, (604) 555-3393

---



## Tweede Normaal Vorm (2 NF)

- Een tabel voldoet aan de 2<sup>de</sup> Normaalvorm, moet het eerst voldoen aan 1NF
- Daarenboven moeten alle niet-key kolommen functioneel afhankelijk zijn van de volledige PK

Bv: de volgende tabel is NIET in 2 NF want de niet-key kolommen zijn functioneel afhankelijk van een gedeelte van de PK:

<u>OrderId</u>	<u>CustomerId</u>	OrderDate	CustomerName
101	1	10/1/2010	Jane Doe
102	2	10/5/2010	John Doe
103	1	10/4/2010	Jane Doe

---

## Derde Normaal Vorm (3 NF)

- Een tabel voldoet aan de 3<sup>de</sup> Normaalvorm, moet het eerst voldoen aan 2NF
- 3NF vereist ook dat er geen functionele afhankelijkheid is tussen niet-key attributen (d.i kolommen)

Bv: de volgende tabel is NIET in 3 NF want 2 niet-key kolommen zijn functioneel afhankelijk van elkaar (SupplierID & Fax)

<u>ItemId</u>	SupplierId	Fax
101	100	(514) 555-2955
102	11	(514) 555-9022
103	525	(313) 555-5735

# Voorbeeld normaliseren: De bestelbon



**PIZZERIA**

**LUIGI**

**Order nr:** 12034  
**Datum:** 13/01/2014

Klantnr: 00259

Alberto Pinacolada  
Steenbrugstraat 135  
1930 Zaventem

artikelnr	omschrijving	aantal	prijs	totaal
10015	Pizza Polo	1	€7,00	€7,00
10280	Pizza Margeritte	3	€6,50	€19,50
10147	Pizza Mozzarella en tomaat	2	€6,50	€13,00

**TOTAAL**

€49,50

# Voorbereiding: de nulde normaalvorm

## ORDERS

ordernr

orderdatum

klantnr

klantnaam

adres

postcode

plaats

artnr

artomschrijving

aantal

prijs

regeltotaal

eindtotaal

Eerst moeten we de informatiebehoefte gaan inventariseren.

Alle gegevens op de bestelbon onder elkaar zetten. Bovenaan plaatsen we in hoofdletters de naam van deze lijst.

De aldus bekomen lijst wordt ook wel de **nulde** normaalvorm (**0 NV**) genoemd.

# 1ste normaalvorm

## ORDERS

ordernr  
orderdatum  
klantnr  
klantnaam  
adres  
postcode  
plaats  
artnr  
artomschrijving  
aantal  
prijs

~~regeltotaal  
eindtotaal~~

## Stap 1:

Zijn alle attributen  
enkelvoudig ?

Ja -> 1 NV

Berekende velden  
worden (meestal)  
verwijderd

# 1ste normaalvorm

## ORDERS

ordernr

orderdatum

klantnr

klantnaam

adres

postcode

plaats

artnr

artomschrijving

aantal

prijs

**Stap 2: Geef de sleutel aan**

(Onderlijn het uniek veld dat het order kan identificeren)

# 1ste normaalvorm

## ORDERS

ordernr

orderdatum

klantnr

klantnaam

adres

postcode

plaats

RG artnr

RG artomschrijving

RG aantal

RG prijs

**Stap 3:** Geef de deilverzameling aan die een herhaald aantal keren voorkomt t.o.v. de primaire sleutel.

Deilverzameling

= RG

= Repeterende Groep

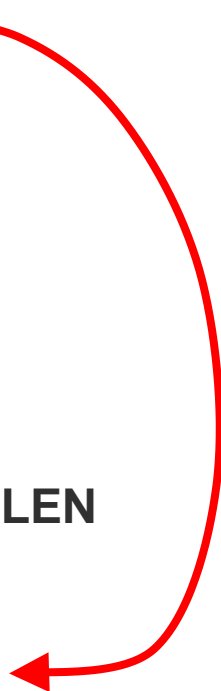
# 1ste normaalvorm

## ORDERS

ordernr  
orderdatum  
klantnr  
klantnaam  
adres  
postcode  
plaats

## BESTELDE\_ARTIKELEN

ordernr  
artnr  
artomschrijving  
aantal  
prijs



Stap 4: Stop de repeterende groepen in een nieuwe lijst en voeg er de sleutel van de oorspronkelijke groep bij



# 1ste normaalvorm

## ORDERS

ordernr  
orderdatum  
klantnr  
klantnaam  
adres  
postcode  
plaats

## BESTELDE\_ARTIKELEN

ordernr  
artnr  
artomschrijving  
aantal  
prijs

Stap 5: Bepaal de sleutel van de nieuwe tabel

Ter herinnering:

**1ste normaalvorm:**  
velden mogen alleen  
enkelvoudige attributen  
bevatten

# 2de Normaalvorm

## ORDERS

ordernr

orderdatum

klantnr

klantnaam

adres

postcode

plaats

In een tabel met een samengestelde primaire sleutel moet elk attribuut afhankelijk zijn van de volledige primaire sleutel

## BESTELDE\_ARTIKELEN

ordernr

artnr

aantal

## ARTIKELEN

artnr

artomschrijving

prijs

# 3de Normalvorm

## ORDERS

ordernr

orderdatum

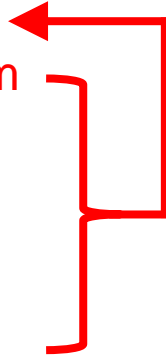
**klantnr**

klantnaam

adres

postcode

plaats



Attributen mogen alleen afhankelijk zijn van de primaire sleutel, niet van de andere attributen

## BESTELDE\_ARTIKELEN

ordernr

artnr

aantal

## ARTIKELEN

artnr

artomschrijving

prijs

# 3de Normalvorm

## ORDERS

ordernr

orderdatum

klantnr

Attributen mogen **alleen** afhankelijk zijn van de primaire sleutel, niet van de andere attributen

## KLANTEN

klantnr

klantnaam

adres

postcode

plaats

## BESTELDE\_ARTIKELEN

ordernr

artnr

aantal

## ARTIKELEN

artnr

artomschrijving

prijs

# Oefeningen normalisatie

Download het bestand van Github

[https://github.com/CSharpSyntraWest/LESSEN\\_DATABANK](https://github.com/CSharpSyntraWest/LESSEN_DATABANK) onder LES 1

Oefeningen - normalisatie.pdf

# OEFENING BIEREN DATABASE

## KLASSIKALE OEFENING

**Installeer SQL SERVER 2019 (Developer edition of SQL Express) (indien Netwerk dit aankan)**

**Github/Databanken/INSTALL/**

**Download de volgende bestanden van Github/Databanken/LES1**

- BierenDb.mdf
- BierenDb\_log.ldf

**In Sql server management studio (na installaties):  
we bekijken de inhoud van de bieren database**

# Samenvatting

## Introductie

- Gegevens versus informatie
- Soorten informatie
- Databanksystemen - DBMS

## Relationele databanken

- Wat is een relationele databank
- Terminologie:

entiteiten, attributen, relaties, tabellen,  
rijen (records), kolommen, PK, FK

## Normalisatie (1NF, 2NF, 3NF)

# VRAGEN?