



PXL-Digital

Toegepaste Informatica

Data Essentials: Normalisatie

Lector

Carine Derkoningen

Isabelle Godfrind

Carina Medats

Kerstin Nys

I. Inhoudsopgave

| | | |
|-----|--|----|
| I. | Inhoudsopgave..... | 1 |
| II. | Normalisatie..... | 2 |
| 1 | Probleemstelling | 2 |
| 2 | Doelstelling | 4 |
| 3 | Mogelijke oplossing..... | 4 |
| 4 | Sleutels | 6 |
| 4.1 | Kandidaat-sleutel | 6 |
| 4.2 | Primaire sleutel (Primary Key) | 7 |
| 4.3 | Externe sleutel (Foreign Key) | 7 |
| 5 | Gegevens verzamelen..... | 10 |
| 5.1 | Verzamelen van gegevens | 10 |
| 5.2 | Nulde normaalvorm..... | 11 |
| 6 | Normalisatiestappen | 14 |
| 6.1 | Eerste normaalvorm. | 17 |
| 6.2 | Tweede normaalvorm..... | 21 |
| 6.3 | Derde normaalvorm..... | 23 |
| 6.4 | Integratie | 28 |
| 6.5 | Aanduiding externe sleutel (Foreign key) | 29 |

II. Normalisatie

1 Probleemstelling

Een bedrijf, school, vereniging ... verzamelt veel gegevens doormiddel van documenten, afspraken Er is echter niet altijd een overzicht van welke gegevens verzameld worden en ook niet waar welke gegevens verzameld worden.

Denken we hier bijvoorbeeld aan een bedrijf dat producten aankoopt en verkoopt. Hetzelfde product komt dus op verschillende documenten voor (aankoopfactuur, offertes, verkoopfactuur ...)

Er moet dus gezocht worden naar welke gegevens moeten worden vastgelegd en wat hun samenhang is. Als nu zonder nader onderzoek al deze gegevens worden opgeslagen in bestanden (XML, plain text, databank, . . .), kunnen zich tijdens het onderhoud van die bestanden vele problemen voordoen.

Veronderstel dat er een document is met de verkoopcijfers, waarin ook de aankoopprijs verwerkt is om de winst te kunnen berekenen. Een aankoopdocument bevat alleen alle aankoopproducten met hun aankoopprijs. Indien de aankoopprijs van een bepaald product verhoogt en de boekhouder wijzigt dit op het aankoopdocument maar hij vergeet hierbij het verkoopdocument. Hier ontstaat reeds een probleem: bij de berekening van de winst (het verschil in aankoop- en verkoopprijs) wordt enkel rekening gehouden met de aankoopprijs in het verkoopdocument. Hierdoor klopt de winst niet en ontstaan er problemen in de boekhouding.

Om het op een andere manier te schetsen volgt hier nog een voorbeeld om het probleem toe te lichten:

Stel dat bijvoorbeeld uit de informatieanalyse van een Project Management Systeem (PMS) blijkt dat er behoefte is aan de informatie in Tabel 1, waaruit kan worden afgeleid dat de informatiebehoefte betrekking heeft op de volgende gegevens:

- Projectnummer
- Projectnaam
- Budget
- Medewerkersnummer
- Medewerkersvoornaam
- Afdeling
- Uren

Hierbij kunnen Medewerkersnummer, Werknemersvoornaam, Afdeling en Uren meerdere keren voor komen.

Als deze gegevens op die manier opgeslagen worden, dan worden de gegevens van de medewerker herhaald bij ieder project waaraan door die medewerker gewerkt wordt. In dit voorbeeld wordt dit geïllustreerd door "005 Fred".

PROJECTOVERZICHT:

| PROJECT | BUDGET | MEDEWERKER | AFDELING | UREN |
|---------|--------|------------|---------------|------|
| 001 VDU | 1000 | 003 Ton | Analyse | 60 |
| | | 005 Fred | Programmering | 100 |
| 002 CRT | 800 | 004 Peter | Analyse | 200 |
| | | 005 Fred | Programmering | 50 |
| 003 | | | | |

tabel 1: Een voorstelling van een PMS

De volgende problemen kunnen optreden bij het wijzigen, toevoegen en verwijderen van gegevens:

- De schrijfwijze van de voornaam (Fred) zou bij elk project anders kunnen zijn, wat niet de bedoeling is.
- Wanneer Fred ontslag neemt moet zijn voornaam overal verwijderd worden. Het ligt voor de hand dat hierbij vergissingen gebeuren en zijn voornaam dus op een/meerdere plaatsen blijft staan, wat niet zou mogen.
- Telkens als vb. Fred aan een nieuw project gaat meewerken moet ook weer zijn afdeling (programmering) ingevuld worden. Dit leidt al vlug tot fouten.
- Als een naam verandert (van bv. een afdeling) dan moet die op alle plaatsen aangepast worden, met mogelijke vergissingen tot gevolg.

Een databank waar gegevens nodeloos in worden herhaald, is moeilijk te onderhouden en zal dus al vlug vol fouten staan.

2 Doelstelling

Het is een doel van data analysis om een zodanige structuur te vinden dat deze problemen niet kunnen optreden. Dit had men al vrij snel begrepen en reeds in de jaren '70 van vorige eeuw gebruikte men gegevensopslagplaatsen. Hier komen 2 methodes het meest aan bod: de hiërarchische database en de netwerkdatabase. De eerste is een boomstructuur, waarbij er steeds een vertakking is wanneer er een één op meer relatie kan gezien worden. De netwerkdatabase is daar een uitbreiding van en laat op elke vertakking meerdere takken op hetzelfde niveau toe.

In deze manier van opslaan van gegevens ontstaan er nog problemen als er zich wijzigingen in de structuur voordoen.

Het is de verdienste van Ted CODD (A Relational model of data for large shared data banks – E.F. Codd – Communications of the ACM – vol.13, pp.377/387 (1970)) dat hij al deze problemen heeft onderkend en een oplossingswijze hiervoor heeft aangereikt.

Deze oplossingswijze staat in de literatuur bekend onder de naam van “**normalisatiestappen van CODD**”.

CODD heeft deze oplossingswijze geschreven in het kader van zijn theorie over "relational data bases" en nooit bedoeld voor het bepalen van de gegevensstructuur van een verzameling. Toch is het praktische nut van zijn normalisatiestappen juist hierbij erg groot gebleken.

3 Mogelijke oplossing

Normalisatie is een techniek

- waarbij een verzameling zodanig in groepen verdeeld wordt,
- dat er zich geen onregelmatigheden bij het onderhoud van de verzameling meer kunnen voordoen.

Deze techniek van normaliseren wordt best altijd bij gegevensanalyse gebruikt.

Met behulp van de normalisatiestappen van CODD, kortweg normaliseren genoemd, zijn we in staat om een willekeurige informatiebehoefte te verdelen in een aantal groepen.

Zo een *genormaliseerde groep* bevat altijd een vast aantal gegevens en kan dus als een tabel worden voorgesteld.

In Tabel 2 zie je de genormaliseerde groep Medewerkers, waarbij het gegeven 'Nummer' iedere rij een unieke id geeft of ook wel sleutel genoemd.

| Medewerkers | | | |
|-------------|-------|----------|----------|
| Nummer | Naam | Voornaam | Postcode |
| 001 | Henk | Jan | 3500 |
| 002 | Peter | Els | 3600 |
| 003 | Ton | Piet | 3770 |
| 004 | Peter | Peter | 3740 |
| 005 | Fred | Ilse | 3800 |

Tabel 2: Voorbeeld van een genormaliseerde groep

Alle attributen Naam, Voornaam en Postcode zijn functioneel afhankelijk van het Nummer (d.w.z.: als er gesproken wordt over een ander (medewerkers)nummer, dan wordt een andere medewerker bedoeld en verandert dus de naam, voornaam en postcode). Er zijn geen functionele afhankelijkheden aanwezig tussen de attributen onderling. De groep is dus genormaliseerd en kan ook als volgt worden weergegeven:

Medewerker (Nummer, Naam, Voornaam, Postcode)

Hierbij worden per groep tussen de haken de gegevens opgesomd waaruit de groep bestaat en wordt de sleutel onderstreept. Bij een samengestelde sleutel zullen dus meerdere gegevens onderstreept zijn. Voorbeeld van een groep met samengestelde sleutel:

Geldafhaling (Datum, Tijd, Bankkaart, Bedrag)

Het is de gewoonte om de sleutel het eerst te vermelden, maar noodzakelijk is dat niet.

In tegenstelling tot de voorstellingswijze in tabelvorm kunnen op deze manier geen afzonderlijke voorkomens apart worden aangegeven. Deze laatste voorstellingswijze geeft dus alleen de groepsnaam (attribuut) aan, terwijl met een tabel de afzonderlijke voorkomens (instanties) getoond kunnen worden.

Meestal is een voorstelling van de verschillende entiteiten voldoende om aan te geven hoe een gegevensverzameling is opgebouwd. Maar soms is een voorstelling van de afzonderlijke voorkomens nodig om dit precies te begrijpen.

Vooraleer er dieper ingaan wordt op normaliseren, wordt nog even dieper ingegaan op sleutels aangezien deze een belangrijke rol spelen bij normalisatie.

4 Sleutels

4.1 Kandidaat-sleutel

Referentienummers worden vaak gebruikt om dingen te identificeren. Factuurnummers, onderdeel codes, serienummers zijn daar algemeen bekende voorbeelden van. Vaak worden combinaties van letters en cijfers gebruikt; bijvoorbeeld bij autonummers. Bij de gegevensverwerking worden sleutels om dezelfde reden gebruikt. Sleutels dienen om records (=rij van data in een database tabel bestaande uit een enkele waarde uit elke kolom) te identificeren zodat er naar gerefereerd kan worden of zodat ze benaderd kunnen worden. Sleutel en sleutelwaarde zijn fundamentele begrippen uit de gegevensverwerking. Het zal daarom duidelijk zijn dat binnen de relationele benadering, faciliteiten moeten bestaan om met sleutels te kunnen werken. In feite hebben relationele systemen speciale regels ten aanzien van sleutels.

Een sleutel moet gevormd worden uit één of meer attributen binnen het record of de relatie.

Een kandidaat-sleutel moet steeds voldoen aan 2 voorwaarden:

- Een kandidaat-sleutel is een combinatie van attributen uit de tabel, waarvoor de attribuutwaarden op een unieke wijze een record van alle andere records uit de tabel onderscheidt. Met andere woorden: een kandidaat-sleutel moet uniek zijn. Het is altijd mogelijk om in een relationeel systeem zo een unieke sleutel aan te wijzen, omdat er geen duplicaat records bestaan.
- Een volgende regel zegt, dat wanneer een willekeurig attribuut wordt weggelaten uit de kandidaat-sleutel, de eigenschap van het uniek identificeren dan verloren moet gaan. Dit betekent dat elke kandidaat-sleutel volgende attributen moet bevatten om elk record uniek te identificeren, maar dat het geen redundante attributen mag bevatten.

Voor elke tabel in de database moet er minstens één kandidaat-sleutel gedefinieerd worden. Vaak is het mogelijk dat meerdere kandidaat-sleutels kunnen beschouwd worden, dat wil zeggen: meerdere verschillende combinaties van attributen die een unieke sleutel opleveren. Dit wordt verderop in de cursus meer in detail besproken.

In vele gevallen zullen per tabel meerdere kandidaat-sleutels bestaan. De kandidaat-sleutel die hieruit wordt gekozen ter unieke identificatie van de record wordt de primaire sleutel genoemd.

4.2 Primaire sleutel (Primary Key)

Een primaire sleutel is een kandidaat-sleutel waarin geen van de elementen een 'null-waarde' kan bevatten. Primaire sleutels zijn bedoeld om een eenvoudige en duidelijke representant te zijn van de echte objecten. Pogingen om de waarde van de primaire sleutels te wijzigen, moeten zorgvuldig bewaakt worden. Dit, omdat primaire sleutels weer kunnen voorkomen in andere delen van de database. De primaire sleutel doet in zo een geval dienst als een verwijzing. Elke verandering aan de primaire sleutel zal in de meeste gevallen betekenen, dat elders voorkomende waarden van de sleutel gewijzigd moet worden.

4.3 Externe sleutel (Foreign Key)

Een externe sleutel is een veld in een relationele tabel die met een primaire sleutel van een tabel overeenkomt. De foreign key kan gebruikt worden om te refereren naar andere records in tabellen. Bijvoorbeeld, stel dat er twee tabellen zijn: een tabel Klant die alle klantgegevens bevat en een Order tabel met alle klantenorders. Het is de bedoeling dat alle bestellingen moeten worden gekoppeld aan een klant die al in de tabel Klant voorkomt. Om dit te doen, wordt een externe sleutel in de Order tabel opgenomen en verwijst deze naar de primaire sleutel van de tabel Klant.

Een tabel kan meerdere foreign keys hebben en elke foreign key kan naar een andere tabel verwijzen.

Neem terug bovenstaand voorbeeld van Klant en Order. In Order zou ook nog bijgehouden kunnen worden welke verkoper deze Order opgenomen heeft. In dat geval zal een externe sleutel in de Order tabel wijzen naar de primaire sleutel van de tabel Personeel.

Een externe sleutel kan ook verwijzen naar de primaire sleutel in de eigen tabel. Bijvoorbeeld, in de tabel Personeel zou de kolom 'chef' kunnen voorkomen. 'Chef' verwijst dan naar de primaire sleutel van de tabel Personeel of is null omdat een Chef ook een personeelslid is.

Onjuiste foreign / primary key relaties of het niet handhaven van deze relaties zijn vaak de bron van veel database en "data modellering" problemen.

Opdracht 1: Veronderstel dat de producten in een onderneming voorzien zijn van een kleurcode voor de afmeting en in twee verschillende lengtes worden gemaakt. Bepaal de kandidaatsleutels en de primaire sleutel in volgende tabel.

| PROD# | AFMETING | KLEUR | LENGTE |
|-------|----------|-------|--------|
| P01 | 8 | ROOD | 50 |
| P02 | 10 | GROEN | 50 |
| P03 | 12 | BLAUW | 50 |
| P04 | 14 | GEEL | 50 |
| P05 | 8 | ROOD | 100 |
| P06 | 10 | GROEN | 100 |
| P07 | 12 | BLAUW | 100 |
| P08 | 14 | GEEL | 100 |

Opdracht 2: Duid in onderstaande tabellen de primaire en externe sleutels aan.

Schilderij

| S_ID | Naam | Artiest | Periode | Waarde | Eigenaar |
|------|----------------------|---------|---------|------------|----------|
| S01 | Vissershuis | A04 | 1882 | 16.000.000 | Boijmans |
| S02 | De balletles | A02 | 1872 | 8.500.000 | Louvre |
| S03 | Mona Lisa | A01 | 1499 | 75.000.000 | Louvre |
| S04 | Namiddag te Oostende | A03 | 1881 | 200.000 | KMSK |

Artiest

| A_ID | Naam | Voornaam | Geboren | Gestorven |
|------|----------|----------|---------|-----------|
| A01 | Da Vinci | Leonardo | 1452 | 1519 |
| A02 | Degas | Edgar | 1834 | 1917 |
| A03 | Ensor | James | 1860 | 1949 |
| A04 | Monet | Claude | 1840 | 1926 |

Eigenaar

| Naam | Plaats | Land |
|----------|-----------|-----------|
| Boijmans | Rotterdam | Nederland |
| Louvre | Parijs | Frankrijk |
| KMSK | Antwerpen | België |

5 Gegevens verzamelen

5.1 Verzamelen van gegevens

De eerste vraag is steeds: "Welke gegevens moeten er in de database opgenomen worden? ".

Bijgevolg wordt gestart met het noteren van alle gegevens(groepen). Op deze manier wordt er niets vergeten. Een gegeven is bijvoorbeeld de voornaam van een medewerker.

Bij het noteren vertrekt men vaak van documenten, maar dit is niet noodzakelijk. Een gesprek met de persoon voor wie de database wordt opgesteld kan ook de basis zijn. In de praktijk wordt meestal een combinatie van beide uitgevoerd.

In de oefeningen bij deze cursus wordt veelal vertrokken van documenten, aangevuld met opmerkingen. Zie de opmerkingen als het resultaat van een gesprek met eerder genoemd persoon.

Als voorbeeld gebruiken we de gegevens van een bedrijf dat softwareprogramma's maakt en elke opdracht als een project ziet. Voor elke project wordt een document opgesteld waarop eigen bedrijfsgegevens, een projectnaam, de verantwoordelijke medewerkers met hun afdeling, hun afdelingschef en de uren per medewerker staan. Verder staan er ook een totaal budget en een aantal uren op het document.

Document 1:

| | | | |
|----------------------|--------------------------|-------------|-------------|
| Bedrijf oan | Dokterspraktijk NietZiek | | |
| Oanstraat zn | Gezondheidsstraat 8 | | |
| 3500 Hasselt | 3600 Genk | | |
| Project | | | |
| 001 VDU | | | |
| Medewerker | Afdeling | Chef | Uren |
| 003 Ton | Analyse | Johan | 60 |
| 005 Fred | Programmering | Ron | 100 |
| aantal uren | | | 160 |
| totaal budget | | | 1000 |

Document 2:

| | |
|--------------|-------------------|
| Bedrijf oan | Apotheek Pillen |
| Oanstraat zn | Medicijnstraat 10 |
| 3500 Hasselt | 3500 Hasselt |

Project

002 CRT

| Medewerker | Afdeling | Chef | Uren |
|----------------------|---------------|-------|------|
| 004 Peter | Analyse | Johan | 200 |
| 005 Fred | Programmering | Ron | 50 |
| aantal uren | | | 250 |
| totaal budget | | | 800 |

De gegevens opsommen levert het volgende lijstje op:

bedrijfsnaam
bedrijfsstraat+nr
bedrijfspostcode
bedrijfsgemeente
klantnaam
klantadres
klantpostcode
klantgemeente
project nummer+naam
medewerker nummer+voornaam
afdeling
chef
uren
aantal_uren
totaal_budget

5.2 Nulde normaalvorm

Bij een blik op bovenstaande lijst kunnen er nog enkele vragen gesteld worden:

- moeten de eigen bedrijfsgegevens in de database opgenomen worden?

- aantal uren is eigenlijk de som van de uren van de medewerkers – moeten beiden worden bijgehouden? (=procesgegevens)

Om de gegevens in de nulde normaalvorm te zetten, wordt uitgegaan van de voorheen opstelde lijst gegevens en worden alle eigen gegevens en de procesgegevens verwijderd/doorgehaald. En indien dit niet onmiddellijk gebeurd is worden bepaalde gegevens verder uitgesplitst: naam en voornaam, postcode en gemeente

Dit levert volgend resultaat:

~~bedrijfsnaam~~
~~bedrijfsstraat+nr~~
~~bedrijfspostcode~~
~~bedrijfsgemeente~~
 klantnaam
 klantadres
 klantpostcode
 klantgemeente
 projectnummer
 projectnaam
 medewerkernummer
 medewerkervoornaam
 afdeling
 chef
 medewerker_uren
~~aantal_uren~~
 totaal_budget

| Project nummer | Project naam | Budget | Medewerkers nummer | Medewerkers voornaam | Afdeling | Chef | Uren | klant_naam | klant_adres | klant_postcode | klant_gemeente |
|----------------|--------------|--------|--------------------|----------------------|---------------|-------|------|------------|---------------------|----------------|----------------|
| 001 | VDU | 1000 | 003 | Ton | Analyse | Johan | 60 | NietZiek | Gezondheidsstraat 8 | 3600 | Genk |
| | | | 005 | Fred | Programmering | Ron | 100 | | | | |
| 002 | CRT | 800 | 004 | Peter | Analyse | Johan | 200 | Pillen | Medicijnstraat 10 | 3500 | Hasselt |
| | | | 005 | Fred | Programmering | Ron | 50 | | | | |
| | | | | | | | | | | | |

Tabel 3: Een ongenormaliseerd model

Kort samengevat: Om NVO te verkrijgen kan het volgende gesteld worden:

- 1.1. Inventariseer alle elementaire gegevens.
- 1.2. Verwijder alle procesgegevens.
- 1.3. Verwijder alle eigen bedrijfsgegevens.

Stap 1.1 Inventariseer alle elementaire gegevens

Alleen elementaire gegevens mogen geïnventariseerd worden. Voor ieder aanwezig samengesteld gegeven moet dus minstens bekend zijn uit welke elementaire gegevens dit is samengesteld. Ieder elementair gegeven behoort een naam te krijgen, waarmee het zich van alle andere gegevens onderscheidt. Of deze opsplitsing zinvol is, hangt ervan af of we ons zinvolle bewerkingen op de afzonderlijke componenten kunnen voorstellen.

Vb1: moet straat en huisnummer als één gegeven worden beschouwd of niet?

Vb2: de score van een voetbalwedstrijd kan worden beschouwd als een samengesteld gegeven, namelijk een geordend paar van twee natuurlijke getallen, maar als het berekenen van het totaal aantal gescoorde doelpunten in één seizoen belangrijk is maak je hier best twee elementaire gegevens van.

Stap 1.2 Verwijder alle procesgegevens

Soms kan een gegeven door berekening uit andere gegevens worden afgeleid. In feite is er dan geen sprake van een keuze uit een verzameling: de keuze ligt vast na het maken van keuzes voor andere gegevens. Dergelijke gegevens worden berekende gegevens of procesgegevens genoemd. Ze kunnen integraal deel uitmaken van de informatiebehoefte van een bedrijfsproces.

Procesgegevens moeten wel apart genoteerd worden, maar worden in de normalisatiestappen niet meegenomen mits aan de volgende voorwaarden is voldaan:

- Alle voor de berekening benodigde gegevens zijn aanwezig.
- De voor de berekening benodigde gegevens bevatten op het tijdstip waarop de berekening moet worden uitgevoerd nog de juiste waarden. Voorbeelden van procesgegevens zijn:
 - de dag van de week kan worden berekend uit de datum, maar is niettemin nuttig om te bepalen of het om een verlofdag gaat.
 - ‘aantal calorieën’ is afleidbaar uit ‘hoeveelheid vetten’, ‘hoeveelheid koolhydraten’ en ‘hoeveelheid eiwitten’, als je weet dat deze drie energiebronnen respectievelijk 9kcal/g,

4kcal/g en 5kcal/g leveren. Nochtans staan op de verpakking van veel voedingswaren alle vier de gegevens vermeld.

- ‘aantal dienstjaren’ is berekenbaar uit ‘datum indiensttreding’, op voorwaarde dat de huidige datum als bekend verondersteld wordt.
-

Stap 1.3 Verwijder alle eigen bedrijfsgegevens

Meestal worden de gegevens van de hoofdzetel van een bedrijf niet mee opgenomen als elementaire gegevens. Indien het bedrijf echter verschillende vestigingen heeft, kan het wel nodig zijn om deze in de database te zetten.

Na deze 3 stappen zijn de gegevens klaar om genormaliseerd te worden.

Om de uitleg van de normalisatiestappen niet te complex te maken wordt verder enkel van het eerste stuk van tabel 3 gebruik gemaakt. De klantgegevens worden buiten beschouwing gelaten.

6 Normalisatiestappen

Het normaliseren vindt plaats in drie stappen. Vanuit een ongenormaliseerde informatiebehoefte worden de genormaliseerde groepen bepaald. De stappen vormen een recept, dat -indien goed opgevolgd - altijd hetzelfde eindproduct geeft. Het product is niet afhankelijk van de kok die het bereidt.

De drie stappen van het recept luiden:

1. Verwijder de zich herhalende deelverzamelingen (NV1)
2. Verwijder de attributen die functioneel afhankelijk zijn van slechts een gedeelte van de sleutel. (NV2)
3. Verwijder de attributen die ook functioneel afhankelijk zijn van andere (niet-sleutel) attributen. (NV3)

In al deze stappen wordt gesproken over ‘verwijderen’. Dit zou er op kunnen duiden dat gegevens echt verwijderd moeten worden en dus geen deel meer uitmaken van het geheel. Dat is natuurlijk niet de bedoeling, want dan zou met behulp van de genormaliseerde groep(en) nooit meer de oorspronkelijke informatiebehoefte vervaardigd kunnen worden.

Het 'verwijderen' in deze normalisatiestappen houdt in:

- het verwijderen uit de oorspronkelijke groep,
- maar het tegelijkertijd **creëren van een nieuwe groep**. Er mag dus niets écht verwijderd worden.

Iedere stap heeft slechts betrekking op één groep. Als er meerdere groepen zijn dan moet iedere stap voor alle groepen afzonderlijk worden uitgevoerd.

De drie stappen zullen nu eerst zonder veel toelichting getoond worden. Hiervoor is het inmiddels bekende projectoverzicht iets aangepast. Na deze korte behandeling wordt iedere stap afzonderlijk uitvoerig toegelicht in de hierna volgende paragrafen.

Deze ongenormaliseerde informatiebehoefte in Tabel 3 (eerste deel) kan als volgt worden voorgesteld:

| | |
|------------|---------------------|
| Projecten: | Projectnummer |
| | Projectnaam |
| | Budget |
| | Medewerkersnummer |
| | Medewerkersvoornaam |
| | Afdeling |
| | Chef |
| | Uren |

Stap 1: Verwijder de zich herhalende deelverzamelingen

De groepen die na deze eerste stap ontstaan worden aangeduid met **eerste normaalvorm (1NV)**. Deze eerste normaalvorm van het projectoverzicht bestaat uit de volgende eerste normaalvormgroepen:

| | |
|---------------------|--------------------------|
| Projecten: | <u>Projectnummer</u> |
| | Projectnaam |
| | Budget |
| Projectmedewerkers: | <u>Projectnummer</u> |
| | <u>Medewerkersnummer</u> |
| | Medewerkersvoornaam |
| | Afdeling |
| | Chef |
| | Uren |

De oorspronkelijke ongenormaliseerde groep is nu opgedeeld in twee groepen waarin géén deelverzamelingen meer voorkomen die zich herhalen. Per groep is de sleutel onderstreept.

Stap 2: Verwijder de attributen die functioneel afhankelijk zijn van slechts een gedeelte van de sleutel.

Na deze stap wordt gesproken over de **tweede normaalvorm (2NV)**.

Deze tweede normaalvorm bestaat uit de volgende groepen:

| | |
|---------------------|---|
| Projecten: | <u>Projectnummer</u> Projectnaam Budget |
| Projectmedewerkers: | <u>Projectnummer</u> <u>Medewerkersnummer</u> Uren |
| Medewerkers: | <u>Medewerkersnummer</u> Medewerkersvoornaam Afdeling Chef |

Alle attributen zijn in elk van deze groepen functioneel afhankelijk van de volledige sleutel.

Stap 3: Verwijder de attributen die ook functioneel afhankelijk zijn van andere (niet-sleutel) attributen.

Na deze stap wordt natuurlijk gesproken over de **derde normaalvorm (3NV)**. Deze derde normaalvorm bestaat voor het projectoverzicht uit de volgende groepen:

| | |
|---------------------|---|
| Projecten: | <u>Projectnummer</u> Projectnaam Budget |
| Projectmedewerkers: | <u>Projectnummer</u> <u>Medewerkersnummer</u> Uren |
| Medewerkers: | <u>Medewerkersnummer</u> Medewerkersvoornaam Afdeling |

Afdelingen: Afdeling
Chef

Niet alle groepen worden in iedere stap nader opgedeeld. Zo is de groep "Project" na de eerste stap niet meer gewijzigd. Voor deze groep geldt dus dat de eerste normaalvorm gelijk was aan de tweede en de derde normaalvorm.

Uit deze waarneming vallen twee zaken af te leiden.

1. Het normalisatieproces kan ergens stoppen. Er is sprake van een zogenaamde ***laatste normaalvorm (LNV)***, daar waar het normalisatieproces stopt voor een bepaalde groep.
2. Een groep in de eerste normaalvorm kan voldoen aan de tweede en de derde normaalvorm. Omgekeerd is het zo, dat iedere groep in 3NV altijd voldoet aan 2NV en iedere groep in 2NV altijd voldoet aan 1NV. De drie normaalvormen zijn deelverzamelingen van elkaar.

In het volgende deel wordt elke stap uitgebreider besproken.

6.1 Eerste normaalvorm.

Stap 1: Verwijder de zich herhalende deelverzamelingen.

Dit is een stap, die vaak moeilijker gevonden wordt dan de overige stappen. Het is ook een stap, die niet eenduidig is. Dat wil zeggen dat het resultaat verschillend kan zijn. Het eindresultaat van de drie normalisatiestappen is wél eenduidig, maar de af te leggen weg kennelijk niet.

Om enige steun te bieden bij het vinden van de eerste normaalvorm, is ook hiervoor een recept ontwikkeld:

Doe het volgende totdat er géén nieuwe groepen meer ontstaan:

- 1 Geef de sleutel van de groep aan.
- 2 Geef de deelverzameling aan die een herhaald aantal keren voorkomt.
- 3 Herhaal de sleutelgegevens van de oorspronkelijke groep samen met de gegevens van de zich herhalende deelverzameling als een nieuwe groep.
- 4 Verwijder de zich herhalende deelverzameling uit de oorspronkelijke groep.

Zich herhalende deelverzamelingen kunnen genest voorkomen. D.w.z. dat er een zich herhalende deelverzameling bevindt binnen een andere deelverzameling, die ook een herhaald aantal keren voorkomt. De eerste normalisatiestap moet zover worden doorgevoerd totdat alle zich herhalende deelverzamelingen zijn opgedeeld.

Stap 1.1 Geef de sleutel van de groep aan

Deze stap is bepalend voor de wijze waarop de eerste normalisatiestap verloopt.

Afhankelijk van de keuze van de sleutel zullen er al of niet (geneste) herhalingen voorkomen. Deze wellicht wat cryptische woorden zullen aan de hand van het voorbeeld nader toegelicht worden.

Na de inventarisatie van de elementaire gegevens is het projectoverzicht als volgt voorgesteld:

| | |
|------------|----------------------|
| Projecten: | <u>Projectnummer</u> |
| | Projectnaam |
| | Budget |
| | Medewerkersnummer |
| | Medewerkersvoornaam |
| | Afdeling |
| | Chef |
| | Uren |

Er is blijkbaar gekozen voor het gegeven Projectnummer als sleutel. Want de eerste normaalvorm wordt gevormd door twee groepen, Project en Medewerker. Men had ook de samengestelde sleutel Projectnummer en Medewerkersnummer kunnen kiezen, maar dan waren er geen zich herhalende deelverzamelingen meer aanwezig geweest.

De ervaring heeft geleerd dat het verstandig is om het meest overkoepelende attribuut als sleutel te kiezen, omdat dan de normalisatiestappen het eenvoudigst zijn. Zo een overkoepelend attribuut is bijvoorbeeld bij een factuur het factuurnummer. Indien mogelijk wordt ook best gestart met een enkelvoudige sleutel. Om deze redenen kiezen we in ons voorbeeld voor Projectnummer als sleutel.

Stap 1.2 Geef de deelverzameling aan die een herhaald aantal keren voorkomt

Afhankelijk van de sleutelkeuze in stap 1.1 zullen nu nul, één of meer zich herhalende deelverzamelingen aanwezig zijn.

| | | |
|------------|----------------------|---|
| Projecten: | <u>Projectnummer</u> | |
| | Projectnaam | |
| | Budget | |
| | Medewerkersnummer | } komt een herhaald aantal keer voor. |
| | Medewerkersvoornaam | |
| | Afdeling | |
| | Chef | |
| | Uren | |

Stap 1.3 Herhaal de sleutelgegevens van de oorspronkelijke groep samen met de gegevens van de zich herhalende deelverzameling als een nieuwe groep

Nu wordt een nieuwe groep gevormd. Deze moet bestaan uit:

de zich herhalende deelverzameling + **de sleutel van de oorspronkelijke groep.**

Deze laatste wordt in de nieuwe groep opgenomen om de koppeling met de oorspronkelijke groep in stand te houden.

Zo worden de volgende twee groepen verkregen:

| | | | |
|---------------------|----------------------|---|---|
| Projecten: | <u>Projectnummer</u> | | |
| | Projectnaam | | |
| | Budget | | |
| | Medewerkersnummer | } | komt een herhaald aantal keer voor. |
| | Medewerkersvoornaam | | |
| | Afdeling | | |
| | Chef | | |
| | Uren | | |
| Projectmedewerkers: | Projectnummer | | |
| | Medewerkersnummer | | |
| | Medewerkersvoornaam | | |
| | Afdeling | | |
| | Chef | | |
| | Uren | | |

Let op:

- Op dit ogenblik is de oorspronkelijke groep nog compleet.
- In de nieuwe groep is nog géén sleutel bepaald.

Stap 1.4 Verwijder de zich herhalende deelverzameling uit de oorspronkelijke groep

Nu wordt pas de oorspronkelijke groep aangepast en vindt de eigenlijke ‘*verwijdering*’ plaats. Door dit in deze stappen te doen kan men zich ervan overtuigen dat alle gegevens die verwijderd worden ook daadwerkelijk in de nieuwe groep zijn opgenomen. De sleutel, die in de nieuwe groep herhaald is, wordt natuurlijk niet verwijderd.

| | |
|---------------------|--------------------------|
| Projecten: | <u>Projectnummer</u> |
| | Projectnaam |
| | Budget |
| Projectmedewerkers: | <u>Projectnummer</u> |
| | <u>Medewerkersnummer</u> |
| | Medewerkersvoornaam |
| | Afdeling |
| | Chef |
| | Uren |

Stap 1.1 (2^{de} iteratie)

De keuze van de samengestelde sleutel ‘Projectnummer’ en ‘Medewerkersnummer’ is misschien niet direct duidelijk. Indien alleen Medewerkersnummer als sleutel gekozen was, dan kwamen Uren en Projectnummer nog een herhaald aantal keren voor. Bij de nu gekozen sleutel zijn er géén zich herhalende deelverzamelingen meer aanwezig. Stap 1 van het normalisatieproces is voltooid en de eerste normaalvorm is bepaald.

| | |
|---------------------|--------------------------|
| Projecten: | <u>Projectnummer</u> |
| | Projectnaam |
| | Budget |
| Projectmedewerkers: | <u>Projectnummer</u> |
| | <u>Medewerkersnummer</u> |
| | Medewerkersvoornaam |
| | Afdeling |
| | Chef |
| | Uren |

6.2 Tweede normaalvorm.

Stap 2: Verwijder de attributen die functioneel afhankelijk zijn van slechts een gedeelte van de sleutel.

Om te komen van de eerste tot de tweede normaalvorm, moeten de attributen die slechts van een *gedeelte van de sleutel* afhankelijk zijn in een aparte groep worden opgenomen. Alleen groepen met een samengestelde sleutel komen hiervoor in aanmerking, want alleen bij een samengestelde sleutel kan een attribuut afhankelijk zijn van een gedeelte van de sleutel.

Eigenlijk is de sleutel van groepen, die nog niet in tweede normaalvorm zijn, geen goede sleutel voor alle attributen, want een sleutel mag volgens de definitie géén overbodige gegevens bevatten. Voor sommige attributen, is dat echter wel het geval. Het zijn dan ook deze attributen die samen met het deel van de sleutel dat voor hun geen overbodige gegevens bevat, een afzonderlijke groep gaan vormen.

Het recept voor de tweede normaalvorm luidt als volgt:

- 2.1. Geef de attributen aan die niet functioneel afhankelijk zijn van de volledige sleutel.
- 2.2. Herhaal de attributen die afhankelijk zijn van een deel van de sleutel samen met dat deel van de sleutel in een nieuwe groep.
- 2.3. Verwijder deze attributen uit de oorspronkelijke groep.

Stap 2.1 Geef de attributen aan die niet functioneel afhankelijk zijn van de volledige sleutel

Om dit te kunnen doen moet van ieder attribuut de vraag beantwoord worden:

"Welk gegeven of combinatie van gegevens identificeert dit attribuut op een éénduidige wijze?"

In het voorbeeld "Projectoverzicht" komt de *groep Projecten* niet in aanmerking, daar die groep niet beschikt over een samengestelde sleutel. Binnen de *groep Projectmedewerkers* zijn o.a. de volgende functionele afhankelijkheden te onderkennen:

Attribuut Medewerkersvoornaam is functioneel afhankelijk van sleutel *Medewerkersnummer*

Attribuut Afdeling is functioneel afhankelijk van sleutel *Medewerkersnummer*

Attribuut Chef is functioneel afhankelijk van sleutel *Medewerkersnummer*

Attribuut Uren is functioneel afhankelijk van sleutel *Projectnummer + Medewerkersnummer*

De attributen Medewerkersvoornaam, Afdeling en Chef zijn slechts functioneel afhankelijk van een deel van de sleutel, het Medewerkersnummer.

Stap 2.2 Herhaal de attributen die afhankelijk zijn van een deel van de sleutel samen met dat deel van de sleutel in een nieuwe groep.

Het kan gebeuren dat een samengestelde sleutel in meerdere delen gesplitst kan worden en dat van ieder deel afzonderlijk attributen functioneel afhankelijk zijn. Er moeten dan meerdere groepen gevormd worden.

In dit voorbeeld ontstaat slechts één nieuwe groep: Medewerkers.

De aldus aangegeven groepen moeten nu gevuld worden. Dit moet zodanig gebeuren, dat iedere nieuwe groep voldoet aan de eisen van de tweede normaalvorm:

- er mogen dus géén herhalingen aanwezig zijn
- en alle attributen moeten functioneel afhankelijk zijn van de volledige sleutel.

In het voorbeeld ontstaat nu de groep Medewerkers

- met als sleutel Medewerkersnummer,
- en als attributen Medewerkersvoornaam, Afdeling, Chef.

Na deze stap is de oorspronkelijke groep Projectmedewerkers opgebouwd uit de volgende groepen:

Projectmedewerkers: Projectnummer
Medewerkersnummer
Medewerkersvoornaam
Afdeling
Chef
Uren

Medewerkers: Medewerkersnummer
Medewerkersvoornaam
Afdeling
Chef

Stap 2.4 Verwijder deze attributen uit de oorspronkelijke groep

Doordat de oorspronkelijke groep nog intact is, wordt hier heel zorgvuldig te werk gegaan bij het verwijderen van de attributen uit deze groep. De sleutel van de oorspronkelijke groep mag niet aangetast worden. Het zijn dus enkel de afhankelijke attributen die verwijderd worden.

We hebben nu in totaal de volgende groepen gekregen:

| | |
|---------------------|--------------------------|
| Projecten: | <u>Projectnummer</u> |
| | Projectnaam |
| | Budget |
| Projectmedewerkers: | <u>Projectnummer</u> |
| | <u>Medewerkersnummer</u> |
| | Uren |
| Medewerkers: | <u>Medewerkersnummer</u> |
| | Medewerkersvoornaam |
| | Afdeling |
| | Chef |

6.3 Derde normaalvorm.

Stap 3: Verwijder attributen die ook functioneel afhankelijk zijn van andere (niet-sleutel) attributen.

Bij de stap naar de derde normaalvorm moeten de attributen

- die functioneel afhankelijk zijn van de volledige sleutel (2NV),
- maar ook nog functioneel afhankelijk zijn van andere attributen,

in aparte groepen worden opgenomen.

Dit soort afhankelijkheden, tussen attributen onderling, wordt wel aangeduid als *transitieve afhankelijkheden*.

Uiteraard kunnen alleen groepen met meerdere attributen deze soort afhankelijkheden bevatten.

De afhankelijkheid tussen de attributen onderling moet wél een functionele afhankelijkheid zijn om over te gaan tot de vorming van een nieuwe groep. Afhankelijkheden, waarbij het ene attribuut op de een of andere manier samenhangt met een ander attribuut, zijn hiervoor geen reden. Een voorbeeld van dit soort (losse, toevallige) afhankelijkheden is het feit dat de datum-in-dienst groter moet zijn dan de geboortedatum van een medewerker. Dit is echter géén functionele afhankelijkheid.

De *afhankelijkheden* die *transitief* genoemd worden zijn altijd van het type 'sleutel attribuut' en deze moeten in een aparte groep worden opgenomen.

Het recept voor de derde normaalvorm is het volgende:

- 3.1. Geef de attributen aan die ook functioneel afhankelijk zijn van andere (niet-sleutel) attributen.
- 3.2. Vorm een aparte groep voor ieder attribuut of combinatie van attributen, waar andere attributen functioneel van afhankelijk zijn.
- 3.3. Duid in de nieuwe groep de sleutel aan.
- 3.4. Verwijder de attributen van de nieuwe groep(en) uit de oorspronkelijke groep, maar laat de sleutel van die nieuwe groep in de oorspronkelijke groep als attribuut staan.

***Stap 3.1* Geef de attributen aan die ook functioneel afhankelijk zijn van andere (niet-sleutel) attributen**

Om dit te kunnen doen moet van ieder attribuut worden vastgesteld of er nog één of meer andere attributen zijn die het beschouwde attribuut uniek identificeren.

In het voorbeeld "Projectoverzicht" is in de groep Projecten geen functionele afhankelijkheid aanwezig tussen de attributen Projectnaam en Budget. De groep Projectmedewerkers bevat slechts één attribuut en komt dus niet in aanmerking. Maar in de groep Medewerkers:

- Is het attribuut Chef functioneel afhankelijk van het attribuut Afdeling.
- de attributen Medewerkersvoornaam en Afdeling zijn onderling onafhankelijk.

In feite is hier vastgesteld dat iedere medewerker één chef heeft en één afdeling. Maar tevens dat iedere afdeling ook één chef heeft en dat dus de chef van de medewerker de chef is van de afdeling van die medewerker.

***Stap 3.2* Vorm een aparte groep voor ieder attribuut of combinatie van attributen, waar andere attributen functioneel van afhankelijk zijn**

Er kunnen meerdere transitieve afhankelijkheden aanwezig zijn. Alle unieke attributen of attribuutcombinaties, die als sleutel fungeren in transitieve afhankelijkheden, geven aanleiding tot de vorming van verschillende groepen.

In het voorbeeld ontstaat slechts één nieuwe groep: Afdelingen.

Stap 3.3 Duid in de nieuwe groep de sleutel aan.

De in stap 3.2 geïnterpreteerde groepen moeten nu van sleutels en attributen worden voorzien.

De op deze manier nieuw gevormde groepen moeten wél aan de definitie van de derde normaalvorm voldoen:

- zij mogen dus geen herhalingen bevatten
- en alle attributen moeten functioneel afhankelijk zijn van de volledige sleutel
- en onderling geen functionele afhankelijkheden bevatten.

De nieuw gevormde groep Afdelingen bestaat uit twee gegevens, "Afdeling" en "Chef". Wat is nu de sleutel: Afdeling of Chef?

Een sleutel moet éénduidig identificeren en dus unieke waarden bevatten. Er kunnen zich verschillende situaties voordoen. De gebruiker zal moeten aangeven welke situatie voor hem geldig is.

Situatie a: Iedere afdeling heeft één chef en iedere chef is chef van één afdeling.

| Afdeling | Chef |
|----------|------|
| A | 1 |
| B | 2 |
| C | 3 |

Door het tekenen van een tabel met afzonderlijke voorkomens is direct te zien dat er sprake is van twee kandidaatsleutels. Welke er gekozen wordt is niet belangrijk.

Situatie b: Iedere afdeling heeft één chef, maar een chef kan van meerdere afdelingen chef zijn.

| Afdeling | Chef |
|----------|------|
| A | 1 |
| B | 2 |
| C | 1 |

Het enige gegeven dat nu een unieke waarde bevat is Afdeling. Dit is dus de sleutel in deze situatie.

Situatie c: Iedere afdeling heeft meerdere chefs, maar iedere chef is slechts chef van één afdeling.

| Afdeling | Chef |
|----------|------|
| A | 1 |
| A | 2 |
| B | 3 |
| B | 4 |

Nu is Chef de sleutel.

Situatie d: Iedere afdeling heeft meerdere chefs en iedere chef kán chef zijn van meerdere afdelingen.

| Afdeling | Chef |
|----------|------|
| A | 1 |
| A | 2 |
| B | 3 |
| B | 1 |

Geen van beide gegevens bevat nu nog unieke waarden. De combinatie van beide gegevens is wél uniek. Beide gegevens vormen een samengestelde sleutel.

De gebruiker heeft aangegeven dat in zijn omgeving situatie a geldt, maar dat situatie b niet uitgesloten wordt geacht. Om deze reden is attribuut Afdeling tot sleutel gekozen.

Na deze stap is de oorspronkelijk 2NV groep Medewerker nu opgebouwd uit de volgende groepen:

Medewerkers: Medewerkersnummer
Medewerkersvoornaam
Afdeling
Chef

Afdelingen: Afdeling
Chef

Stap 3.4 Verwijder de attributen van de nieuwe groep(en) uit de oorspronkelijke groep, maar laat de sleutel van die nieuwe groep in de oorspronkelijke groep als attribuut staan.

Uit de nu nog intact zijnde oorspronkelijke groep

- moeten de attributen van de nieuw ontstane groep(en) verwijderd worden.
- en de sleutel van de nieuwe groep blijft als attribuut in de oorspronkelijke groep gehandhaafd.

Dit geeft de volgende groepen, die elk op zich in de 3-de normaalvorm zijn:

Projecten: Projectnummer
Projectnaam
Budget

Projectmedewerkers: Projectnummer
Medewerkersnummer
Uren

Medewerkers: Medewerkersnummer
Medewerkersvoornaam
Afdeling

Afdelingen: Afdeling
Chef

6.4 Integratie

Soms wordt vertrokken van 2 of meerdere verschillende documenten waarvan de gegevens in 1 database moet opgenomen worden, bijvoorbeeld een factuur en een inkoopdocument. In dat geval wordt er een normalisatie gemaakt van de documenten apart. Na het bekomen van de derde normaalvormen, worden deze samengevoegd tot één geheel.

Hiervoor worden de derde normaalvormen naast elkaar gezet en wordt gekeken of er in de verschillende normalisaties tabellen voorkomen met dezelfde primary key. Indien dat zo is, worden die tabellen samengevoegd tot 1 tabel in de integratie. Let op dat er geen attributen verloren gaan!

Voorbeeld:

Bij de normalisatie van de factuur staat in NV3 de tabel Producten (productnummer, productnaam, verkoopprijs) en bij de normalisatie van het inkooporder staat in NV3 de tabel Producten(productnummer, productnaam, inkoopprijs) . Vermits deze beide tabellen dezelfde primary key hebben, wordt dit in de integratie de tabel Producten(productnummer, productnaam, inkoopprijs, verkoopprijs).

Verder worden in de integratie alle andere tabellen ook opgenomen, die slechts vanuit 1 normalisatie komen.

Uitwerkingen hiervan komen tijdens het maken van de oefeningen en de groepsopdracht aan bod.

6.5 Aanduiding externe sleutel (Foreign key)

Bij normaliseren wordt voor elke groep telkens de primary key aangeduid. Ook is er in NV1 sprake van primary key meenemen, in NV2 deel van sleutel meenemen naar nieuwe groep en in NV3 sleutel als attribuut in oorspronkelijke groep laten staan. Dit alles teneinde de relaties te behouden tussen de attributen.

In het deel Sleutels was er niet enkel sprake van de primaire sleutel, maar ook van de externe sleutel. De externe sleutel of foreign key, is het attribuut dat verwijst naar een primaire sleutel of zijn de attributen die verwijzen naar een volledige primaire sleutel.

Na het normaliseren moet dus nu nog enkel de foreign key (FK) aangeduid worden:

| | | |
|---------------------|--------------------------|-------------------------------------|
| Projecten: | <u>Projectnummer</u> | |
| | Projectnaam | |
| | Budget | |
| Projectmedewerkers: | <u>Projectnummer</u> | FK =>Projecten(Projectnummer) |
| | <u>Medewerkersnummer</u> | FK=>Medewerkers(Medewerkersnummer) |
| | Uren | |
| Medewerkers: | <u>Medewerkersnummer</u> | |
| | Medewerkersvoornaam | |
| | Afdeling | FK=>Afdelingen(Afdeling) |
| Afdelingen: | <u>Afdeling</u> | |
| | Chef | FK=>Medewerkers(Medewerkersnummer). |

Bij de tabel Afdelingen wordt Chef als foreign key aangeduid. Een Chef is immers ook een medewerker en heeft dus ook een medewerkersnummer.

Indien er na de derde normaalvorm nog een integratie gebeurt, worden de relaties pas aangeduid in de integratie.