



Normaliseren

Inleiding

In deze les gaan we de stappen van het normaliseren van nulde normaalvorm (0NV) tot en met de derde normaalvorm (3NV) bespreken aan de hand van een volledig uitgewerkt voorbeeld.

De normalisering stappen

Er zijn 5 stappen in het normaliseren, maar meestal worden alleen de eerste 3 uitgevoerd.

1. De eerste regel bestaat uit drie deel stappen:
 - A. Herhalende groepen gegevens in aparte tabellen onderbrengen en voorzien van een primaire sleutel.
 - B. Proces gegevens verwijderen uit de tabel.
 - C. Samengestelde gegevens splitsen in aparte atomaire gegevens.
2. Kolommen afhankelijk maken van de gehele primaire sleutel.
3. Kolommen onafhankelijk van elkaar maken en alleen afhankelijk van de primaire sleutel.

Deze regels worden toegepast op een verzameling gegevens om zo uiteindelijk tot een aantal tabellen te komen die in de 3^{de} normaalvorm staan.

Primaire sleutels zijn gegevens die uniek zijn voor een specifieke verzameling feiten. Vaak worden hiervoor codes of nummers gebruikt. Denk aan het stamboeknummer van een leerling op school, het paspoortnummer van elke paspoort, het persoonsnummer op Aruba, burger servicenummer in Nederland, het social security number van de VS, het kentekennummer van een auto etc. Al deze nummers en codes zijn uniek voor één persoon, auto, etc.

Samengestelde gegevens zijn gegevens die eigenlijk uit meerdere gegevens zijn opgebouwd. Bijvoorbeeld het adres. Deze bestaat op Aruba uit Stad, Straatnaam/wijknaam (barrio naam) en het huisnummer. Soms is het echter niet noodzakelijk om een samengesteld gegeven verder op te splitsen. Een geboortedatum hoeft (meestal) niet verder te worden opgesplitst in dag, maand en jaar.

De gegevens die niet verder kunnen worden opgesplitst heten **atomaire gegevens**. Zo is een dag een dag en deze is niet verder te verdelen in andere gegevens.

Proces gegevens bevatten informatie die berekent moet worden. Deze worden niet opgenomen in de tabellen van een database. Ze worden



Normaliseren

iedere keer opnieuw berekend om zo de meest actuele waardes te bevatten.

Voorbeeld 1

In het volgende voorbeeld worden de gegevens voor een bedrijf genormaliseerd waar data moet worden bijgehouden zoals: welke cursussen het personeel volgt/gevolgd heeft, wat hun functie is, in welke vestiging (gebouw) zij werken, het salaris etc.

0NV

De **0^{de} normaalvorm** is een niet genormaliseerde verzameling van alle gegevens die in de te maken DB moeten worden bijgehouden.

Voor dit voorbeeld zijn alle gegevens verzameld en ziet de 0^{de} normaalvorm er zo uit:

tblPersoneel(PersoneelsID, Voornaam, Achternaam, Geboortedatum, Geslacht, Functienaam, DatumInDienst, Vestiging, Vestigingsadres, Salaris, CursusID1, Cursusnaam1, Cursusdatum1, CursusID2, CursusNaam2, Cursusdatum2)

De tabel naam is tblPersoneel en bevat alle gegevens die van het personeel moeten worden bijgehouden in de DB.

Gevuld zou de tabel er bijvoorbeeld zo uit kunnen zien:

PersoneelsID	Voornaam	Achternaam	Geboortedatum	Geslacht	Functienaam	DatumInDienst	Vestiging	Vestigingsadres	Salaris	CursusID1	Cursusnaam1	Cursusdatum1	CursusID2	CursusNaam2	Cursusdatum2
10	Pietje	Puk	December 7, 1980	M	Kok	June 1, 2001	Noord	Bubali 22	2600	100	Toetjes maken	April 10, 2004			
12	Marie	Dubois	June 12, 1989	V	Serveerster	January 1, 2002	Noord	Bubali 22	2400	120	Tafel dekken	September 11, 2003			
13	Chantal	de Vries	July 21, 1984	V	Serveerster	August 15, 2004	Oranjestad	Steenweg 44a	2450	120	Tafel dekken	March 1, 2005			
14	Maria	Carlson	February 14, 1991	V	Kok	September 1, 2005	Oranjestad	Steenweg 44a	2600	100	Toetjes maken	January 15, 2007			
10	Pietje	Puk	December 7, 1980	M	Kok	June 1, 2001	Noord	Bubali 22	2600				140	Garneren	July 23, 2004

1NV

Om deze tabel in de 1^{ste} normaalvorm te zetten moet de eerste regel van het normaliseren worden uitgevoerd. Deze bestaat uit de volgende stappen:

1. Herhalende groepen gegevens in aparte tabellen onderbrengen en voorzien van een primaire sleutel.
2. Gegevens splitsen in aparte velden.



Normaliseren

Om de eerste stap goed te kunnen doen moet eerst bepaald worden welk gegeven als primaire sleutel gekozen moet worden voor de hele tabel in de ONV.

In de tabel in ONV is het gegeven *personeelsID* het veld dat elke persoon, met al zijn of haar bijbehorende gegevens, uniek definieert.

Elk personeelslid zou nul, één of meerdere cursussen kunnen volgen. Dit betekent dat er per personeelslid, nul of meer cursussen moet kunnen worden geregistreerd. Ofwel, de gegevens van een cursus zullen herhaald worden voor elke cursus die een personeelslid volgt.

Elke cursus wordt uniek geïdentificeerd door het gegeven cursusID. En elke cursus die gevolgd wordt door een personeelslid wordt uniek geïdentificeerd door de combinatie van de gegevens personeelsID en cursusID samen.

Er vanuit gaande dat het gegeven vestigingsadres niet verder opgesplitst hoeft te worden, levert het toepassen van de stappen van de 1NV de volgende tabellen op:

tblPersoneel(PersoneelsID, Voornaam, Tussenvoegsel, Achternaam, Geboortedatum, Geslacht, Functienaam, DatumInDienst, Vestiging, Vestigingsadres, Vestigingspostcode, Vestigingsplaats, Salaris)

tblPersoneelscursussen(PersoneelsID, CursusID, Cursusnaam, Cursusdatum)

De gevulde tabel Personeelscursussen zou er zo uit kunnen zien:

PersoneelsID	CursusID	Cursusnaam	Cursusdatum
10	100	Toetjes maken	April 10, 2004
12	120	Tafel dekken	September 11, 2003
13	120	Tafel dekken	March 1, 2005
14	100	Toetjes maken	January 15, 2007
10	140	Garneren	July 23, 2004

Opvalt dat de cursusnamen worden herhaalt daar deze afhankelijk zijn van de CursusID, dat deel uitmaakt van de gehele primaire sleutel.

De tblPersoneelscursussen heeft twee gegevens die samen de primaire sleutel vormen, dit wordt een **samengestelde sleutel** genoemd.

Deze twee tabellen staan in de 1NV.



Normaliseren

De relatie tussen de twee tabellen is een 1:n (1 op n) relatie. Namelijk, één personeelslid kan 0, 1 of meerdere cursussen volgen/gevolgd hebben.

De gegevens die als primaire sleutels functioneren worden voorop in de tabel geschreven en onderstreept.

2NV

Om de twee tabellen van de 1NV over te zetten in de 2^{de} NV moet de 2^{de} regel van het normaliseren worden toegepast. Deze is:

Kolommen afhankelijk maken van de gehele primaire sleutel.

Onder kolommen (van de tabel) wordt verstaan de gegevens/veldnamen. Anders gezegd, alle gegevens moet door de primaire sleutel uniek worden geïdentificeerd.

In de tabel tblPersoneel worden alle gegevens uniek geïdentificeerd (zijn afhankelijk van) de primaire sleutel PersoneelsID. Echter in de tweede tabel, tblPersoneelscursussen, is dit niet het geval. Het gegeven cursusnaam is niet afhankelijk van het sleutel gegeven PersoneelsID. Het wordt uniek geïdentificeerd door het gegeven CursusID. Dit betekent dat volgens de regel van 2^{de} NV het gegeven cursusnaam in een aparte tabel dient te worden geplaatst. Toepassing van deze regel leidt tot de volgende groep tabellen:

tblPersoneel(PersoneelsID, Voornaam, Tussenvoegsel, Achternaam, Geboortedatum, Geslacht, Functienaam, DatumInDienst, Vestiging, Vestigingsadres, Vestigingspostcode, Vestigingsplaats, Salaris)

tblPersoneelscursussen(PersoneelsID, CursusID, Cursusdatum)

tblCursussen(CursusID, Cursusnaam)

Deze tabellen staan in de 2^{de} NV.

De gevulde tabellen Personeelscursussen en Cursussen zouden er als volgt uit kunnen zien:

PersoneelsID	CursusID	Cursusdatum
10	100	April 10, 2004
12	120	September 11, 2003
13	120	March 1, 2005
14	100	January 15, 2007
10	140	July 23, 2004



Normaliseren

CursusID	Cursusnaam
100	Toetjes maken
120	Tafel dekken
140	Garneren

De tabel tblPersoneelscursussen is opgesplitst in twee tabellen. Eén die de cursussen registreert die door het personeelslid gevolgd zijn/worden. En een tabel die de cursussen zelf definieert.

De relatie tussen tblPersoneelscursussen en tblCursussen is een 1:n relatie. Namelijk elke cursus is door meerdere personeelsleden te volgen. Eén personeelslid volgt één bepaalde cursus op één datum.

De basis gegevens, in dit geval alleen cursusnaam, wordt nu maar één keer opgeslagen in de gehele database. Er zouden nog meer gegevens van een cursus bijgehouden kunnen worden. Denk aan, cursus duur, cursus niveau en cursuslocatie, etc.

3NV

Voor het omzetten van de tabellen uit de 2NV naar de 3NV moet de volgende regel worden toegepast op de tabellen van de 2NV:

Kolommen onafhankelijk van elkaar maken en alleen afhankelijk van de primaire sleutel.

Dit wil zeggen dat de gegevens die **niet** tot de primaire sleutel behoren. **Niet afhankelijk** mogen zijn van andere gegevens die niet tot de primaire sleutel behoren.

Bij de tabellen in de 2NV zijn er twee groepen gegevens in de tabel tblPersoneel, die niet aan deze regel voldoen.

Het gegeven Functienaam en de gegevens Vestiging, Vestigingsadres, Vestigingspostcode en Vestigingsplaats zijn niet afhankelijk van de primaire sleutel, PersoneelsID, in deze tabel.

Om dit te verbeteren moeten er twee nieuwe tabellen worden toegevoegd. Eén voor het bijhouden van de verschillende functies en één voor de informatie over een vestiging.

Toepassen van deze regel levert de volgende groep tabellen op:

tblPersoneel(PersoneelsID, Voornaam, Tussenvoegsel, Achternaam, Geboortedatum, Geslacht, FunctieID, DatumInDienst, VestigingID, Salaris)



Normaliseren

tblVestigingen(VestigingID, Vestiging, Vestigingsadres, Vestigingspostcode, Vestigingsplaats)

tblFuncties(FunctieID, Functienaam)

tblPersoneelscursussen(PersoneelsID, CursusID, Cursusdatum)

tblCursussen(CursusID, Cursusnaam)

Voor de twee nieuwe tabellen zijn er nieuwe primaire sleutels gemaakt, VestigingsID en FunctieID. Deze identificeren de gegevens in beide tabellen uniek.

Deze tabellen staan nu in de 3^{de} normaalvorm

De relatie tussen tblFunctie en tblPersoneel is nu een 1:n relatie. Eén functie zou door meerdere personeelsleden uitgevoerd kunnen worden. Eén personeelslid heeft één functie.

De relatie tussen tblVestigingen en tblPersoneel is nu een 1:n relatie. In elke vestiging werken meerdere personeelsleden. Eén personeelslid werkt bij één vestiging.

Nu het normaliseren klaar is zou men deze tabellen in het database programma kunnen gaan invoeren. Er moet echter nog één ding gedaan worden en dat is het bepalen van de datatype van de verschillende velden van de tabellen.

Opmerking:

In een database worden géén **procesgegevens** bijgehouden. Procesgegevens zijn gegevens die berekend worden uit gegevens in de database. Denk aan totalen, gemiddelden, percentages, etc. Deze worden iedere keer opnieuw berekend. Dit zorgt ervoor dat steeds de meest correcte informatie wordt weergegeven.



Normaliseren

De gevulde tabellen in de 3^{de} Normaalvorm zouden er als volgt uit kunnen zien:

PersoneelsID	Voornaam	Achternaam	Geboortedatum	Geslacht	FunctieID	DatumInDienst	VestigingID	Salaris
10	Pietje	Puk	December 7, 1980	M	500	June 1, 2001	112	2600
12	Marie	Dubois	June 12, 1989	V	510	January 1, 2002	112	2400
13	Chantal	de Vries	July 21, 1984	V	510	August 15, 2004	114	2450
14	Maria	Carlson	February 14, 1991	V	500	September 1, 2005	114	2600

FunctieID	FunctieNaam
500	Kok
510	Serveerster

VestigingID	Vestiging	Vestigingsadres
112	Noord	Bubali 22
114	Oranjestad	Steenweg 44a

PersoneelsID	CursusID	Cursusdatum
10	100	April 10, 2004
12	120	September 11, 2003
13	120	March 1, 2005
14	100	January 15, 2007
10	140	July 23, 2004

CursusID	Cursusnaam
100	Toetjes maken
120	Tafel dekken
140	Garneren

De oranje gekleurde kolommen zorgen voor de links (relaties) tussen de verschillende tabellen. In de tabellen waar ze geen primaire sleutel zijn worden ze de Vreemde/Secundaire/Externe Sleutel (Foreign Key) genoemd. Een Externe Sleutel is een veld in de tabel die de Primaire Sleutel is in een andere tabel.

Data Dictionary

De Data Dictionary is een lijst van alle velden die in de database gebruikt worden waar bij wordt aangegeven wat voor soort gegeven een veld moet bevatten en welke grootte deze moet hebben.

De onderstaande lijst beschrijft de verschillende datatype die gebruikt kunnen worden in een database. Ze kunnen wel enigszins variëren afhankelijk van het DBMS dat gebruikt wordt.

Text (short text): Alle combinaties van letters en cijfers met een maximale lengte van 255 tekens.

Memo (long text): Idem aan Text, alleen is de maximale lengte nu 65.535 tekens.

Number: Cijfers voor het maken van berekeningen. Er zijn verschillende soorten cijfers. Bijvoorbeeld byte, integer, long integer, decimal, etc.



Normaliseren

Date/Time: Voor een datum en een tijd. Er kan gerekend worden met de velden.

Currency: Numerieke gegevens die als geldwaarde worden weergegeven.

Autonumber: Ieder record krijgt een nieuw (opvolgend uniek)nummer

Yes/No: Een logisch veld (boolean). Kan maar twee waarden bevatten, yes of no.

De Data Dictionary voor dit voorbeeld ziet er als volgt uit:

Veldnaam	Datatype	Lengte
PersoneelsID	AutoNumber	Integer (default)
Voornaam	Short Text	20 characters
Tussenvoegsel	Short Text	10 characters
Geboortedatum	Date	Short date
Geslacht	Short Text	1 character (M/F)
DatumInDienst	Date	Short Date
Salaris	Currency	Decimal (default)
VestigingsID	Autonumber	Integer (default)
Vestiging	Short Text	25 characters
Vestigingsadres	Short Text	50 characters
Vestigingspostcode	Short Text	15 characters
Vestigingsplaats	Short text	20 characters
CursusID	Autonumber	Integer (default)
CursusDatum	Date	Short Date
Cursusnaam	Short Text	25 characters
FunctieID	Autonumber	Integer (default)
Functienaam	Short Text	25 characters

Indexen

Voor het sorteren en zoeken in de tabellen worden indexen aangemaakt in een DBMS. Kenmerken van indexen zijn:

- Indexen worden gebruikt om gegevens in volgorde te zetten
- Je kan een index op meerdere velden in je tabel aanbrengen
- Indexen maakt het zoeken naar informatie sneller
- Veel indexen maakt het programma langzamer, bij wijzigingen van informatie

Er wordt automatisch al een index gemaakt op elk veld dat hoort tot de primaire sleutels.

Oefen opdracht 1



Normaliseren

Hieronder zie je een tabel in de ONV. Probeer deze zelf te normaliseren naar de 3NV.

tblOngenormaliseerd(KlantID, Klantnaam, KlantTelnr, Klantfaxnr, Klantwebsite, KlantEmail, Branchecode, Branchenaam, OrderD1, Besteldatum1, Productcode1, Productnaam1, Hoeveelheid1, PrijsPerProduct1, Korting1, Totaalprijs1, OrderD2, Besteldatum2, Productcode2, Hoeveelheid2, PrijsPerproduct2, Korting2, Totaalprijs2)

Bepaal zelf goede namen voor de tabellen. Denk eraan, de tabelnaam is de context van de gegevens en feiten.

Oefen opdracht 2

Stel je wil een database maken van je muziek verzameling. Je hebt muziek op verschillende media. Denk aan vinyl, tape, CD of digitale formaten zoals mp3 of FLAC. Verder wil je weten wat de naam van het album is, de datum van het album, de datum waarop het album is vrij gegeven, de speeltijd van het album, de namen van de liedjes, de artiest(en)/groep die het lied produceren, het genre van het lied, wie de componist is.

Normaliseer de informatie van de 0^{de} normaalvorm tot en met de 3^{de} normaalvorm. Laat hier bij alle tabellen zien voor elke normaalvorm.