

Normaliseren - Theorie

Normaliseren is het omzetten van een document (factuur, offerte, planning, enzovoort) in een databaseontwerp. Dat doe je via een aantal vaste stappen, de zogenaamde normaalvormen. Door deze stappen te volgen, weet je 100% zeker dat je een correct ontwerp krijgt voor je relationele database.

Algemeen

- Het komt voor dat je niets kunt doen voor een normaalvorm. Wanneer je tijdens een toets moet laten zien hoe je normaliseert, geef je dit duidelijk aan.
- Je mag zelf een primaire sleutel verzinnen voor een tabel wanneer nodig, dit kan het proces makkelijker maken.
- Bekijk ook het voorbeeld als naslagwerk: <https://www.youtube.com/watch?v=8wCfaxoj0e4>.
- We beginnen bij 0 want we zijn software ontwikkelaars 😊

0e normaalvorm

- Verzamel alle gegevens (*bijna altijd zelfstandige naamwoorden*) in één lijst.
 - We leggen geen statische gegevens vast (zaken die bijvoorbeeld op iedere factuur staan, zoals de naam van het bedrijf).
 - Er worden geen proces gegevens opgeslagen (zaken die je kunt berekenen, bijvoorbeeld het totaalbedrag, deze hoeft dus niet apart in de database te komen).
- Merk alvast de “herhalende groepen” aan. Op een factuur herken je die meestal aan de tabelvorm.
 - Bijvoorbeeld: een factuur waarop meerdere producten staan. Alle gegevens van een product zijn dan samen de “herhalende groep”.

1e normaalvorm

- Plaats alle niet-herhalende gegevens samen in een tabel.
 - Geef deze tabel een primaire sleutel.
- Plaats iedere herhalende groep in een losse tabel.
 - Geef iedere tabel een primaire sleutel.
 - Voeg bij herhalende groepen ook de primaire sleutel van de hoofdtabel toe.

2e normaalvorm

- Bekijk de herhalende groepen; zijn alle aanwezige attributen afhankelijk van beide primaire sleutels?
 - We noemen de beide primaire sleutels ook wel ‘samengestelde sleutel’.
 - Als dat niet zo is, plaats deze attributen dan in een nieuwe tabel.
 - Geef deze nieuwe tabel de primaire sleutel waar het attribuut wél van afhankelijk was.

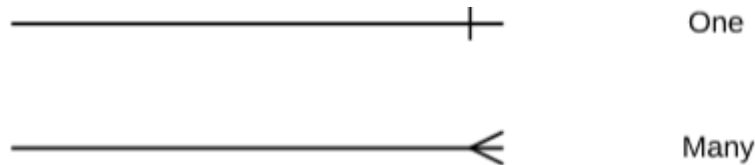
3e normaalvorm

- Bekijk iedere tabel nog eens. Zijn er nog attributen die niet bij de primaire sleutel horen?
 - Zo ja, haal deze dan uit de tabel naar een nieuwe tabel.
 - De nieuwe tabel krijgt een primaire sleutel.
 - Deze primaire sleutel blijft ook achter in de oude tabel als vreemde sleutel.

ERD – Theorie

Zie ook de losse PowerPoint over ERD. Verkort stappenplan:

- **Stap 1:** Voeg een ID toe als primary key bij de tabellen die nog geen duidelijke ID hebben.
- **Stap 2:** Controleer of alle tabellen een juiste foreign key hebben, waarmee ze verwijzen naar andere tabellen. Die krijgen vaak de naam {andere_tabel_naam_enkelvoud}_id. (bijv.: *user_id*, *post_id*)
- **Stap 3:** Teken de ERD en bedenk de relaties:



- **Stap 4:** Voeg de datatypes toe aan alle velden en/of maak een los Data Dictionary waarin je de datatypen beschrijft.

Extra:

Als je werkt met Laravel dan moet je die naamgevingsconventies al gaan aanhouden in je ERD:

- Tabellen zijn *lower_snake_case* en meervoud. Bijv.: *users*, *posts*, *comments*
- De primaire sleutel in een tabel heet altijd 'id'
- Foreign keys hebben de naam van de andere tabel, in enkelvoud en dan gevolgd door *_id*. Bijv.: *user_id*, *post_id*, *comment_id*
- Een Koppeltabel (ookwel: Pivot Table) is altijd een samenstelling van de tabelnamen die het koppelt, enkelvoud, op alfabetische volgorde. Bijv.: *comment_post*, *post_user*

Door op deze manier te werken weten Models in Laravel automatisch in welke tabel ze moeten zoeken, en welke kolomnamen er in queries gebruikt moeten worden.

Oefeningen

Iedere oefening kun je normaliseren én verder uitwerken tot een ERD. De docent geeft steeds aan hoever je uitwerking moet gaan.

Oefening 1 – Factuur

| Factuur | | |
|-------------------|---|------------------|
| Klantnummer: 100 | | Factuurdatum: |
| Peter Hanson | | 25 November 2009 |
| Klavierstraat 15 | | Factuurnummer: |
| 1234 DD Nieuwstad | | DUI0925120 |
| Bestelling | | |
| Aantal | Omschrijving | Prijs |
| 1 | Boek: De Grote Vlucht - Andrew Stevens | 12,55 |
| 1 | Boek: Geen traan te veel - Jan van Apel | 17,99 |
| Totaal: | | 30,54 |

Oefening 2 – Bloedbank

| BLOEDAFGIFTE-OVERZICHT | |
|------------------------|-----------------------|
| Donornr | 900015 |
| Naam | Schovers |
| Roepnaam | Paul |
| Straat | Schutsestraat 12 |
| Plaats | Prinsenbeek |
| Postcode | 4841 EK |
| Geslacht | M |
| Bloedgroep | A |
| Rhesusfactor | Positief |
| Datum | Hoeveelheid in liters |
| 02-10-2011 | 0,45 |
| 22-10-2012 | 0,45 |
| 01-11-2013 | 0,50 |

Oefening 3 – Muziek

Je werkt in een ontwikkelteam voor een opdrachtgever, die wil een muzikspeler ontwikkelen. De desingafdeling heeft een interface ontworpen, zie bijlage. Aan jou de taak hiervan een genormaliseerd database-ontwerp aan te leveren. Laat alle normalisatiestappen zien.

(Dit is natuurlijk een screenshot uit Spotify - de tekst hierboven geeft aan hoe zoiets in het bedrijfsleven zou kunnen gaan - de opdracht is wel echt, maak een genormaliseerd database-ontwerp!)

AFSPEELLIJST

HeulMooi

Aangemaakt door Bart Roos • 108 nummers, 7 uur 31 min

AFSPELEN

VOLGERS 0

Filter

Downloaden

| TITEL | ARTIEST | ALBUM | | |
|-------------------------------|----------------|---------------------------|------------|------|
| ♥ God Only Knows - Remastered | The Beach Boys | Pet Sounds (Original ... | 2014-07-05 | 2:53 |
| ♥ The Scientist | Coldplay | A Rush of Blood to the... | 2014-07-05 | 5:10 |
| ♥ You Learn About It | The Gathering | Souvenirs | 2014-07-05 | 5:09 |
| ♥ Day Is Done | Johan | Pergola | 2014-07-05 | 4:18 |
| ♥ I Am A Camera | The Buggles | Adventures In Modern... | 2014-07-05 | 4:54 |
| ♥ Beautiful Girl | INXS | INXS Remastered | 2014-07-05 | 3:27 |

Oefening 4 – Huisarts

Een huisartspraktijk maakt nog gebruik van een oud systeem. Het systeem gaat worden vervangen door een nieuw systeem. Onderstaand is een scherm te zien van het oude systeem. Aan jou de taak om een genormaliseerde database te ontwerpen.

| | |
|--------------------------------|--|
| Lidnummer: | 0001 |
| Aanhef: | Fam. |
| Naam: | R.A. Volwerk |
| Adres: | Emmaweg 17 |
| Postcode/Woonplaats: | 4931 CZ Geertruidenberg |
| Telefoon: | 01621-18682 |
| Liddatum: | 29-11-1988 |
| Huisbezoek: | J |
| Huisbezoeker: | Corry |
| Datum van telefonisch contact: | 12-12-1988 26-01-1989 28-01-1989 01-02-1989 27-04-1989 |
| Datum huisbezoek: | 17-12-1988 02-06-1989 04-06-1989 |

| Gegevens patiënten | | | | | | |
|--------------------|--------------|-------------------|----------|------------|----------|-------|
| Nr. | Voorl. | Naam | Geb.dat. | Overl.dat. | Diagnose | Gesl. |
| 0001 | A. | Robert A. Volwerk | 01-12-70 | | 53 | M |
| 0002 | Jessica G.S. | Volwerk/Klein | 16-12-67 | | 71 | V |

Bijzonderheden patiënt 1: Robert is actief in de werkgroep 53 en de redactie van de OKé-krant.
Tevens is hij contactpersoon in de regio Brabant.

Bijzonderheden patiënt 2: Bij Jessica is op haar elfde jaar diagnose 71 gesteld.

Het systeem registreert voor een familie de datums van het telefonisch contact en de datum van het huisbezoek. Aan een familie kunnen verschillende familieleden worden gekoppeld. Per familielid kan er opmerking worden geplaatst (bijzonderheid).

Oefening 5 – Verzekering

Een verzekeringsmaatschappij stuurt het volgende overzicht op

| | | |
|---------------------------------------|---------------------------|---------|
| Factuurnummer 53 | | |
| Jaar 2010 | | |
| Relatienummer VP-112 | | |
| Naam J. van Stoffelen | | |
| Straat W.F. Hermanslaan 55 | | |
| Postcode en plaats 2548 AH Teteringen | | |
| Polisnr. | Omschrijving | Premie |
| 1234 | Algemene ziektekosten | € 2400 |
| 2234 | Inboedelverzekering | € 100 |
| 3323 | Autoverzekering | € 300 |
| 3334 | Rechtsbijstandverzekering | € 200 + |
| | Totaal | €3000 |
| | Korting | € 300 - |
| | Te betalen | €2700 |

De korting wordt niet bepaald als percentage o.i.d., maar is een vast bedrag dat door de verkoper van de verzekering per klant wordt vastgesteld.

Oefening 6 – Docent en klassenlijsten

| | | | | | | | |
|---------------|-------------|---------------|----------------------------|----------|------------|----------|--------|
| CURSUSCODE: | PHP/MySQL | Omschrijving: | Leren programmeren met PHP | | | | |
| CURSUSPLAATS: | Breda | | | | | | |
| CURS_DATUM: | 04-12-2013 | | | | | | |
| <hr/> | | | | | | | |
| DOC_CODE: | SC | | | | | | |
| DOC_NAAM: | SCHALKEN | | | | | | |
| DOC_TELNR: | 04674-63673 | | | | | | |
| <hr/> | | | | | | | |
| CURSISTNR | NAAM | ROEPNAAM | STRAAT EN NR | POSTCODE | PLAATS | GESLACHT | CIJFER |
| 002 | KRIMPEN | TANJA | TILBURGSEWEG 2 | 4222 BB | GOIRLE | V | 6 |
| 064 | BROEKEN | BRAM | DRIMMELSEWEG 8 | 4395 XX | MADE | M | 8 |
| 088 | VOS | HENK | BESBEEMD 64 | 4142 CE | OOSTERHOUT | M | 9 |

Oefening 7 – Bestelling bij boekwinkel

Bestelnummer : 92033

Datum : 16-03-2019

| ISBN | Auteur | Aantal | Prijs | Bedrag |
|--------|---------|--------|-------|--------|
| 1213 | Smit | 10 | 70 | 700 |
| 2312 | Berger | 10 | 20 | 450 |
| 3214 | Voskuil | 5 | 40 | 200 |
| Totaal | | | | 1350 |

Oefening 8 – Modellenbureau

| | | | | | | | | | | |
|------------|------------|----------|-------|----------|------------|----------|----------------|------------|------------|------------|
| Opdrachtnr | Bedrijfsnr | Bedrijf | Adres | Postcode | Plaats | Telefoon | Contactpersoon | Woonplaats | Datum | |
| Modelnr | Naam | Roepnaam | Adres | Postcode | Woonplaats | Telefoon | Geb_datum | Geslacht | Kleur_haar | Kleur_ogen |
| Modelnr | Naam | Roepnaam | Adres | Postcode | Woonplaats | Telefoon | Geb_datum | Geslacht | Kleur_haar | Kleur_ogen |
| Modelnr | Naam | Roepnaam | Adres | Postcode | Woonplaats | Telefoon | Geb_datum | Geslacht | Kleur_haar | Kleur_ogen |
| Modelnr | Naam | Roepnaam | Adres | Postcode | Woonplaats | Telefoon | Geb_datum | Geslacht | Kleur_haar | Kleur_ogen |
| Modelnr | Naam | Roepnaam | Adres | Postcode | Woonplaats | Telefoon | Geb_datum | Geslacht | Kleur_haar | Kleur_ogen |
| Modelnr | Naam | Roepnaam | Adres | Postcode | Woonplaats | Telefoon | Geb_datum | Geslacht | Kleur_haar | Kleur_ogen |
| Modelnr | Naam | Roepnaam | Adres | Postcode | Woonplaats | Telefoon | Geb_datum | Geslacht | Kleur_haar | Kleur_ogen |
| Modelnr | Naam | Roepnaam | Adres | Postcode | Woonplaats | Telefoon | Geb_datum | Geslacht | Kleur_haar | Kleur_ogen |
| Modelnr | Naam | Roepnaam | Adres | Postcode | Woonplaats | Telefoon | Geb_datum | Geslacht | Kleur_haar | Kleur_ogen |

Oefening 9 – Vrachtbrief

Vrachtnr 32323
Autonr 10
Kenteken KR-11-VO
Merk DAF
Weeknr 22

Vrachten

| Klantnr | Naam | Plaats | Colli | Gewicht in ton |
|----------------------------------|-----------|-----------|-------|----------------|
| 21 | Malheures | Parijs | 29 | 1,20 |
| 34 | Flamand | Brussel | 34 | 1,34 |
| 32 | Polenter | Brugge | 34 | 1,34 |
| 25 | Klaustal | Bonn | 30 | 1,21 |
| 9 | Timmer | Eindhoven | 31 | 1,29 |
| Totaal gereden kilometers | | | | 1340 km |

Oefening 10 – Kassabon

Kassabonnummer : 200131321

Datum : 2-10-2019

Kassa : 12

Kassière : Klaas

| Artikelnr | Omschrijving | Aantal | Prijs | Totaal |
|-----------|--------------|--------|-------|--------|
| 1231 | banaan | 3 | 1,20 | 3,60 |
| 3212 | paprika | 1 | 0,20 | 0,20 |
| 3432 | koffie | 3 | 2,50 | 7,50 |

Totaal : 11,30

Oefening 11 – Highscore

| | |
|--------|-----------------|
| Date | : 12-01-2020 |
| Game | : Battlefield V |
| Server | : Europa01 |

| PlayerID | Playername | Highscore | Date | Timeplayed |
|----------|---------------|-----------|----------|------------|
| A0001 | MegaShooter | 12000 | 1-1-2020 | 10:04:05 |
| A0041 | SuperBlaaster | 11190 | 3-1-2020 | 18:12:10 |

Oefening 12 – Cijferlijst

Studentnummer : A102200

Naam : Piet Jansen

Straat + nummer : Straat 1

Woonplaats: Woonplaats

CIJFERLIJST

Docentcode : AB00

Naam : Ad Boer

Telefoonnummer : 06 - 11111111

| Vakcode | Omschrijving | Periode | Cijfer |
|---------|----------------------|---------|--------|
| A01 | C# Basic programming | 1 | 7 |
| A02 | PHP Basics | 2 | 9 |

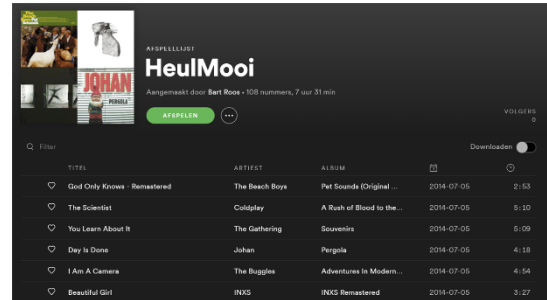
Uitwerkingen

Hier laten we van een van de oefeningen zien hoe we deze kunnen uitwerken a.d.h.v. de normalisatiestappen.

Oefening 3 – muziek

0NF

1. We maken een lijst van alle gegevens (alle zelfstandige naamwoorden)
2. We halen statische gegevens weg (zoals de woorden “Afspeellijst” en “Filter”)
3. We verzinnen waar nodig alvast namen voor deze gegevens:



| | Uitleg |
|---|---|
| 1. Titel_Afspeellijst 2. Naam_Gebruiker 3. RG(o Titel_Liedje o Artiest_Liedje o Titel_Album_Liedje o Datum_Toegevoegd_Liedje o Duur_Liedje) | 4. “HeulMooi” is de afspeellijsttitel 5. “Bart Roos” is hier de maker van deze lijst o We zien meerdere liedjes met ieder een artiest, album, datum dat het toegevoegd werd aan de lijst en een duur. o Voor de duidelijkheid zetten we _Liedje bij de namen, omdat het attribuut daarbij hoort. |

4. We markeren met RG(...) wat de herhalende gegevens zijn (Repeating Groups)

Het aantal nummers is een totaal, dat is een procesgegeven wat we later kunnen uitrekenen.

Het aantal volgers is ook een totaal en dus een procesgegeven. We zien verder helemaal niets over volgers en kunnen dat dus niet normaliseren. We mogen dat voor nu vergeten en normaliseren alleen wat er is.

1NF

1. Verplaats de herhalende gegevens (Repeating Groups) naar een aparte tabel.
2. Verzin eventueel zelf primaire sleutels (primary keys). Ik verzin hier Id_Afspeellijst als primaire sleutel van de afspeellijst
(Primaire sleutels toevoegen mag altijd als dat zinnig is)

3. Dit is het resultaat nu:

| | | |
|-----------------|--------------------|----------------|
| Id_Afspeellijst | Titel_Afspeellijst | Naam_Gebruiker |
|-----------------|--------------------|----------------|

| | | | | | |
|-----------------|--------------|----------------|--------------------|-------------------------|-------------|
| Id_Afspeellijst | Titel_Liedje | Artiest_Liedje | Titel_Album_Liedje | Datum_Toegevoegd_Liedje | Duur_Liedje |
|-----------------|--------------|----------------|--------------------|-------------------------|-------------|

4. In de tweede tabel die we nu hebben maken we een samengestelde sleutel. Dat is een primaire sleutel die bestaat uit twee velden (gemarkeerd hierboven): Id_Afspeellijst en Titel_Liedje.
5. De titel van een liedje als sleutel, is dat wel handig? Is dat uniek voor ieder liedje? De titel “Tonight” is van een liedje van Kings of Leon, maar ook Elton John, Blondie, Jonas Brothers en nog veel

meer. **We voegen daarom een zelfverzonnen primaire sleutel toe aan deze nieuwe tabel. Deze gaan we wel uniek houden:**

| | | |
|------------------------|--------------------|----------------|
| <u>Id_Afspeellijst</u> | Titel_Afspeellijst | Naam_Gebruiker |
|------------------------|--------------------|----------------|

| | | | | | | |
|------------------------|------------------|--------------|----------------|--------------------|-------------------------|-------------|
| <u>Id_Afspeellijst</u> | <u>Id_Liedje</u> | Titel_Liedje | Artiest_Liedje | Titel_Album_Liedje | Datum_Toegevoegd_Liedje | Duur_Liedje |
|------------------------|------------------|--------------|----------------|--------------------|-------------------------|-------------|

Dankzij de samengestelde sleutel verliezen we niet de koppeling tussen de liedjes en de afspeellijst waar ze bijhoren.

2NF

1. Van het liedje zijn de titel, de artiest, het album en de duur niet afhankelijk van de VOLLEDIGE samengestelde sleutel. Ze horen alleen bij de sleutel van het liedje: Id_Liedje.

2. We verplaatsen deze gegevens dus naar een nieuwe tabel en kopiëren ook de sleutel mee:

| | | |
|------------------------|--------------------|----------------|
| <u>Id_Afspeellijst</u> | Titel_Afspeellijst | Naam_Gebruiker |
|------------------------|--------------------|----------------|

| | | |
|------------------------|------------------|-------------------------|
| <u>Id_Afspeellijst</u> | <u>Id_Liedje</u> | Datum_Toegevoegd_Liedje |
|------------------------|------------------|-------------------------|

| | | | | |
|------------------|--------------|----------------|--------------------|-------------|
| <u>Id_Liedje</u> | Titel_Liedje | Artiest_Liedje | Titel_Album_Liedje | Duur_Liedje |
|------------------|--------------|----------------|--------------------|-------------|

3. De Datum_Toegevoegd_Liedje hoort bij nader inzien wel bij de samengestelde sleutel (*bij sommige playlists voegt iemand het liedje vandaag toe en bij andere staat het liedje al een tijdje in de playlist*)

3NF

1. In de 3^e normaalvorm kijken we bij iedere tabel of alle attributen afhankelijk zijn van de sleutel.
2. De naam van de gebruiker is niet afhankelijk van de afspeellijst en gaat dus naar een nieuwe tabel.
3. We verzinnen hier zelf een primaire sleutel en nemen die over:

| | | |
|------------------------|--------------------|---------------------|
| <u>Id_Afspeellijst</u> | Titel_Afspeellijst | <u>Id_Gebruiker</u> |
|------------------------|--------------------|---------------------|

| | | |
|------------------------|------------------|-------------------------|
| <u>Id_Afspeellijst</u> | <u>Id_Liedje</u> | Datum_Toegevoegd_Liedje |
|------------------------|------------------|-------------------------|

| | | | | |
|------------------|--------------|----------------|--------------------|-------------|
| <u>Id_Liedje</u> | Titel_Liedje | Artiest_Liedje | Titel_Album_Liedje | Duur_Liedje |
|------------------|--------------|----------------|--------------------|-------------|

| | |
|---------------------|----------------|
| <u>Id_Gebruiker</u> | Naam_Gebruiker |
|---------------------|----------------|

Nu hebben we 4 genormaliseerde tabellen. Dit is een prima basis voor ons databaseontwerp.

Extra informatie

Je mag na de vorige normaalvorm stoppen, maar we leggen hier toch nog wat uit over andere oplossingen:

- Mogelijk vindt je Titel_Album_Liedje ook niet afhankelijk van de sleutel. Dan zou het kunnen dat je nog een tabel krijgt:

| | | |
|------------------------|--------------------|---------------------|
| <u>Id_Afspeellijst</u> | Titel_Afspeellijst | <u>Id_Gebruiker</u> |
|------------------------|--------------------|---------------------|

| | | |
|------------------------|------------------|-------------------------|
| <u>Id_Afspeellijst</u> | <u>Id_Liedje</u> | Datum_Toegevoegd_Liedje |
|------------------------|------------------|-------------------------|

| | | | | |
|------------------|--------------|----------------|-----------------|-------------|
| <u>Id_Liedje</u> | Titel_Liedje | Artiest_Liedje | <u>Id_Album</u> | Duur_Liedje |
|------------------|--------------|----------------|-----------------|-------------|

| | |
|---------------------|----------------|
| <u>Id_Gebruiker</u> | Naam_Gebruiker |
|---------------------|----------------|

| | |
|-----------------|-------------|
| <u>Id_Album</u> | Titel_Album |
|-----------------|-------------|

Met zo'n tabel voor albums komt er ruimte om ook meer albuminformatie op te slaan. Liedjes hoeven daar via de foreign key (Id_Album) dan enkel naar te verwijzen

- In de opdracht stonden ook afbeeldingen. Die hoeft je voor deze opdracht niet in je databaseontwerp op te nemen, maar mocht je dat voor een app toch willen:
 - In de applicatie laat je gebruikers uploaden naar een bepaald pad (bijv.: *images/21o4k1wes.jpg*).
 - In je database (en ook ontwerp) sla je een afbeelding op als string (die het pad naar de afbeelding bevat. Bijv.: *images/21o4k1wes.jpg*)
- Wanneer je klaar bent met normaliseren is het enorm belangrijk dat je jouw databaseontwerp gaat **testen** op fouten. Dat kun je gewoon op papier doen:
 - Bedenk voorbeelden van data voor deze database. Dat verschilt per casus.
 - Probeer ze een plek te geven in de bovenstaande tabellen: