

Normaliseren volgens Codd.

Reader voor het ID College Zoetermeer,
Unit ICT niveau 4

Geschreven door:

M. Püttmann
H. van Rheenen
L. Karsmaker

Inhoudsopgave

Normaliseren volgens Codd.....	1
Index van belangrijke termen	2
Inleiding	3
Waarom normaliseren	3
Wanneer niet normaliseren	3
Normaliseren	3
Attributen en Entiteiten.....	4
Primaire en externe sleutels.....	5
Nulde normaal vorm (0NV)	6
Eerste normaalvorm (1NV)	9
Tweede normaalvorm (2NV)	12
Derde normaalvorm (3NV)	14
Overige normaalvormen	16
Diagrammen	17
Relaties	17
Het Stroken diagram	18
Het Bachman diagram	18
Literatuur	19
Practicum/opgaven	20

Index van belangrijke termen

Atomair.....	9
Attribuut	
Attribuut	4
Attribuutwaarde	4
Stroken diagram.....	18
Bachman diagram	
Bachman diagram	18
Entiteiten.....	18
Relaties.....	17
constanten.....	6
Database	
Anomalie.....	3
Integere database	3
Niet integere database	3
Redundant	3
Entiteit	
Entiteit.....	4
Entiteit-ocurrence	4
Koppelentiteit	5
Relaties.....	17
Entiteiten	
Bachman diagram	18
informatiebehoefte.....	6
procesgegevens.....	6
Repeterende groep	6
Sleutel	
Extern	5
Primair	5
Sleutel.....	5
Sleutelattribuut.....	5
Stroken diagram	
Relaties.....	17
Sleutelattributen	18
Stroken diagram.....	18

Inleiding

Over het goed inrichten van een database bestaan geen vaste richtlijnen. Wat de optimale vorm is waarin gegevens opgeslagen kunnen is niet vastgelegd. Omdat het ontwerpen van een databasestructuur subjectief is kan je eindeloos over de aanpak discussiëren.

In 1970 kwam Dr. Edgar F. Codd¹ met een algemene methode om relationele databases te ontwerpen. Dit 'recept' beschrijft het normaliseren² van data (gegevens) in een relationele database. In de loop van de Jaren is Codd's methode voor het normaliseren van gegevens de ongeschreven standaard geworden. De methode van Codd's is later uitgebreid met een vierde en vijfde normaalvorm.

Waarom normaliseren

Een database is een verzameling gegevens. Zonder structuur is het niet meer dan een opeenhoping van data. Er bestaat weinig tot geen verband tussen de verschillende datagroepen. Het is zelfs zeer waarschijnlijk dat dezelfde gegevens meerdere keren in de database opgenomen zijn. Hierdoor kunnen in de database anomalieën (fouten) ontstaan tijdens invoegen, updaten of verwijderen van gegevens. De Database heet dan niet integer³. Dit is uiteraard niet wensbaar en kan worden voorkomen door gegevens redundant (gegarandeerd eenmalig) op te nemen door relaties in de database aan te brengen. Hoe deze relaties er uitzien wordt bepaald door het normalisatieproces.

Het doel van normaliseren is om een integere en efficiënte database te maken waar gegevens redundant opgenomen zijn. De aangebrachte structuur zorgt er voor dat op verschillende manieren naar de data gekeken kan worden.

Wanneer niet normaliseren

Normaliseren kan als gevolg hebben dat de database erg traag wordt. Teveel tabellen en relaties moet dan worden opgezocht. Je kan een database dus kapot normaliseren. Wanneer de snelheid van de database gaat leiden onder het normalisatieproces kun je het beter niet doen.

Een andere reden om niet te normaliseren is wanneer gebruikers zelf queries moeten kunnen maken in de database. Een enkelvoudige of simpele tabel levert een hoog gebruikersgemak. Een enkelvoudige tabel is ook makkelijk te archiveren.

Normaliseren

Er bestaan vijf normaalvormen. Wij zullen de stappen volgen om tot de derde normaalvorm te komen. De overige normaalvormen worden vrijwel nooit gebruikt. We zullen daarom er geen aandacht aan geven.

De normaalvormen nul tot en met drie.

¹ Dr. E.F. Codd, "A Relational Model for Large Shared Databanks", Communications van ACM, 1970

² Het proces van datanormalisatie is genoemd naar president Nixon die in 1970 de relaties met China normaliseerde.

³ Een database heet **integer** als er maar één manier is om een attribuutwaarde in de database te raadplegen.

- 0NV: Nulde normaalvorm – inventariseer de attributen,
- 1NV: Eerste normaalvorm – splits repeterende groep af,
- 2NV: Tweede normaalvorm – splits gedeeltelijk afhankelijke attributen af,
- 3NV: Derde normaalvorm – splits de onafhankelijke attributen af.

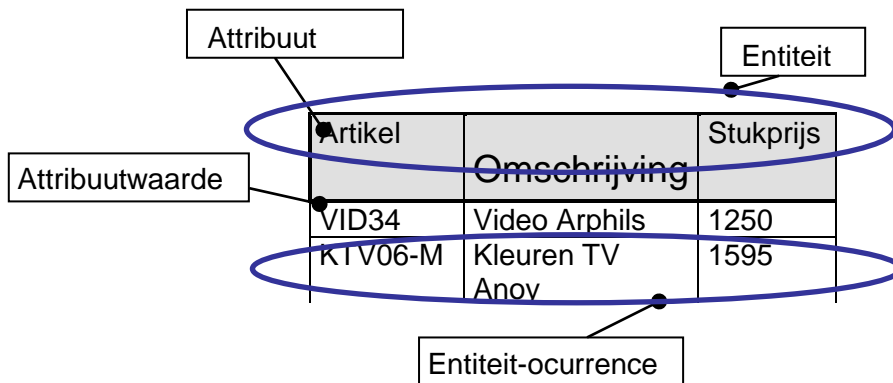
Verder heb je nog:

- BCNV: Boyce Codd normaalvorm,
- 4NV: Vierde normaalvorm,
- 5NV: Vijfde normaalvorm.

Attributen en Entiteiten

Een attribuut representeert een enkelvoudig object. Bijvoorbeeld 'een computer', 'een student', 'een liedje'. Een attribuut kan bepaalde attribuutwaarde aannemen. Het attribuut *computer* kan de attribuutwaarden *DELL 240* of *Compaq deskpro* aannemen. Het attribuut *student* kan de attribuutwaarden *Mohammed*, *Pieter* of *Jessika* aannemen. Een attribuut beschrijft dus gelijk gestemde objecten.

Meerdere attributen bij elkaar vormen een entiteit. Een entiteit is een bij elkaar horende groep van betekenisvolle attributen. De rij bijbehorende attribuutwaarden vormen samen de entiteit-ocurrence. Een entiteit geef je meestal een betekenisvolle naam die overeenkomt met de tabelnaam.

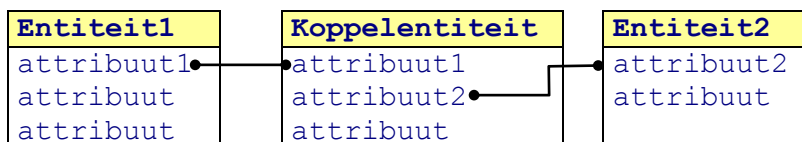


De verschillende onderdelen van een tabel.

Primaire en externe sleutels

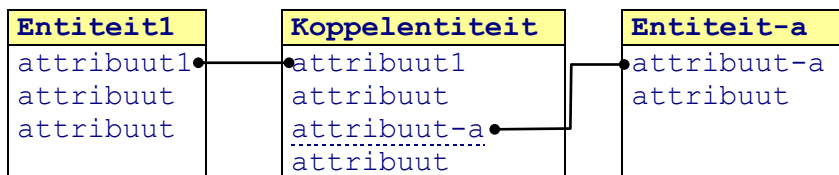
Verschillende entiteiten (of tabellen) kunnen een relatie met elkaar hebben. Deze relatie wordt via een sleutel aangegeven. Om elke entiteit-occurance te kunnen aanwijzen is een primaire sleutel nodig. Elke entiteit heeft precies één primaire sleutel. Een primaire sleutel kan bestaan uit een speciale attribuut, het sleutelattribuut, of uit een verzameling van sleutelattributen.

In koppelentiteiten wordt meestal verwezen naar de sleutelattributen van de entiteiten die ze koppelen. De primaire sleutel van een koppelentiteit is meestal de verzameling van deze (verwijzende) sleutelattributen.



In entiteit1 is de primaire sleutelattribuut1. In de koppelentiteit is de primaire sleutel de attributen attribuut1+attribuut2. De koppeling met Entiteit2 vindt plaats via primaire sleutels.

Een externe sleutel is een attribuut uit een entiteit die zelf niet een sleutelattribuut is, maar wel verwijst naar een sleutelattribuut in een andere entiteit. Een vreemde sleutel koppelt dus alleen entiteiten met elkaar.



In koppelentiteit is de primaire sleutel attribuut1. In de Entiteit-a is de primaire sleutel attribuut-a. De koppeling tussen Koppelentiteit en Entiteit-a vindt plaats via de externe sleutel attribuut-a in Koppelentiteit. In entiteit-a is attribuut-a een primaire sleutel.

Nulde normaal vorm (ONV)

ONV:

De nulde normaalvorm krijg je door alle attributen op te schrijven (inventariseren) die uit de analyse van de informatiebehoefde komt. Bepaal het sleutelattribuut die deze gegevens uniek identificeren. Geef aan de groep een betekenisvolle naam. Geef de repeterende groep aan door deze te laten inspringen.

De te nemen stappen:

- Stap: Bepaal de constanten en procesgegevens
- Stap: Inventariseer de attributen
- Stap: Geef aan de entiteit een betekenisvolle naam.
- Stap: Bepaal de sleutelattribuut.
- Stap: Bepaal de repeterende groep.

Uitgangspunt van het normaliseren is steeds de informatiebehoefte van de toekomstige gebruiker. Welke attributen je gaat opnemen wordt bepaald door de gegevens die de klant wil zien.

Voorbeelden zijn een inschrijfformulier, een lijst van voetbaluitslagen, een overzicht van alle CD's die je bezit, de rekening van de Gamma, de factuur van de aannemer voor het verbouwen van je huis.

Niet alle gegevens die op een formulier aanwezig zijn hoeven te worden opgenomen. Gegevens die via een proces kunnen worden aangemaakt, bijvoorbeeld het berekenen van BTW of de datum van vandaag, neem je niet op. Ook constanten die nooit veranderen, bijvoorbeeld je eigen adres, neem je niet op. Kortom alleen unieke en veranderlijke gegevens binnen de informatiebehoefde worden opgenomen als attribuut in de database.

Er kunnen ook attributen voorkomen waarvan de attribuutwaarden meerdere keren gebruikt worden. Deze repeterende groep zetten we bij elkaar in een eigen entiteit.

Tot slot geven we de entiteit een betekenisvolle naam zodat we direct kunnen zien welke gegevens zijn opgenomen in de tabel.

Aan de hand van een Inschrijvingsformulier zullen we het normaliseren doorlopen. De vraag is; hoe ziet een genormaliseerde database er in deze informatiebehoefde uit?

ATV sch 701		INSCHRIJVING		14FEB06	
STUDENTNR: 1234		Naam:		Rheenen JC van	
		Adres:		Hooicamp 9	
		Woonplaats:		Houten	
CURSUSCODE	CURSUSNAAM	STARTDATUM	GEBOUW	LOCATIE	GEBOUWNAAM
SO	Systeemontw.	12 mei 06	2	Utrecht	Paviljoen
SA	Systeemanalyse	21 jun 06	1	Utrecht	Toren
PR	Programmeren	11 aug 06	7	Enschede	Bastille
IA	Informatieanalyse	16 okt 06	4	Zwolle	Vrijhof

Stap: bepaal de constanten en procesgegevens

We zoeken eerst de constanten en procesgegevens.

Normaliseren volgens Codd

In ons inschrijvingsformulier zijn dat de schoolcode (701) en de datum van inschrijving (14FEB06)

ATV sch 701	INSCHRIJVING	14FEB06
-------------	--------------	---------

Het gegeven 'ATV sch 701' is een constante. De datum '14FEB06' is een proces.

De schoolcode zal niet veranderen. Deze kan je dus zelfs voorgedrukt op het inschrijvingsformulier plaatsen. De datum van inschrijving kan automatisch gegenereerd worden door het systeem. We nemen deze dus niet op als attributen.

Opmerking: Het zou kunnen voorkomen dat je bij meerdere scholen kan inschrijven. In zo'n geval neem je de schoolcode natuurlijk wel op. Dit geldt ook voor de inschrijfdatum als je die nogmaals ergens anders zou willen gebruiken. Het normaliseren gebeurt dan op basis van een bredere informatiebehoefte. Bijvoorbeeld omdat er meerdere formulieren bestaan.

Stap: Inventariseer de attributen

Nu we de constanten en procesgegevens bepaald hebben kunnen we de attributen inventariseren. We vinden:

STUDENTNR Naam Adres Woonplaats CURSUSCODE CURSUSNAAM STARTDATUM GEBOUW LOCATIE GEBOUWNAAM

De inventarisatie van de attributen.

Stap: geef de entiteit een betekenisvolle naam.

Hiermee hebben we al bijna de nulde normaalvorm te pakken. Deze entiteit (groep van attributen) geven we de betekenisvolle naam *Student*.

Stap: bepaal de primaire sleutel.

Welk attribuut is zo uniek zodat we alle gegevens die we opnemen hiermee kunnen terug vinden? In ons geval kunnen we het attribuut *STUDENTNR* gebruiken als primaire sleutel. Immers, geen student heeft dezelfde studentnummer. Het studentnummer zal nooit veranderen. Eenmaal toegekend behoud je dit nummer. Kennen we het studentnummer dan is alles van deze student te vinden. Dit attribuut kunnen we dus prima gebruiken als sleutelattribuut. We geven dit aan door dit attribuut te onderstrepen. Omdat dit sleutelattribuut de enige is in deze entiteit is het tevens de primaire sleutel.

Normaliseren volgens Codd

Student
<u>Studentnr</u>
Naam
Adres
Woonplaats
Cursuscode
Cursusnaam
Startdatum
Gebouw
Locatie
Gebouwnaam

Het attribuut 'Studentnr' is de primaire sleutel.

Stap: Bepaal de repeterende groep

We zien dat de student van Rheenen meerdere cursussen volgt. Deze starten op verschillende dagen in verschillende gebouwen:

We kunnen dus meerdere attribuutwaarden aan wijzen op het formulier die behorende bij een attribuut. Dit is de repeterende groep.

CURSUSCODE	CURSUSNAAM	STARTDATUM	GEBOUW	LOCATIE	GEBOUWNAAM
SO	Systeemontw.	12 mei 06	2	Utrecht	Paviljoen
SA	Systeemanalyse	21 jun 06	1	Utrecht	Toren
PR	Programmeren	11 aug 06	7	Enschede	Bastille
IA	Informatieanalyse	16 okt 06	4	Zwolle	Vrijhof

De repeterende attributen in het inschrijfformulier.

De bijbehorende attributen laten we inspringen om dit aan te geven.

We hebben nu de nulde normaalvorm gevonden.

ONV:

Student
<u>Studentnr</u>
Naam
Adres
Woonplaats
Cursuscode
Cursusnaam
Startdatum
Gebouw
Locatie
Gebouwnaam

Eerste normaalvorm (1NV)

1NV :

0NV+alle attributen atomair.

Van de 0NV entiteit scheiden we de repeterende groep. De relatie tussen de entiteiten verbinden we door het sleutelattribuut over te nemen in de nieuwe entiteit. We maken van de sleutel weer een primaire sleutel.

De te nemen stappen:

Stap: Bepaal of de attributen atomair zijn

Stap: Splits de repeterende groep af.

Stap: Geef aan de afgesplitste entiteit een betekenisvolle naam.

Stap: Maak van de sleutel een primaire sleutel

Stap: Bepaal of de attributen atomair zijn

We gaan eerst na of alle attributen atomair⁴ zijn. Dat wil zeggen, elk attribuut is *enkelwaardig* en *ondeelbaar*.

Als we naar de nulde normaalvorm kijken zouden we de attributen *adres*, *woonplaats* en *datum* nog kunnen splitsen in *adres+huisnummer*, *postcode+woonplaats* en *dag+maand+jaar*. Of we dit doen hangt af van het gebruik dat we gaan maken van deze attributen. Is het nuttig om deze attributen verder te splitsen omdat we de verschillen onderdelen gaan gebruiken? Zo ja, dan kunnen we het attribuut verder opdelen. Zo niet dan levert een splitsing ons alleen een complexere en langzamere database op.

In ons inschrijvingsformulier levert het verder splitsen van deze attributen geen voordeel op. Zoals ze hier staan zijn ze dus atomair.

⁴ Het woord atomair is afgeleid van het Griekse woord *atomos* dat betekend "kan niet meer in kleiner delen gesplitst worden". De filosofie van het atomisme zoals dat door de Griek Democitos (460-340 B.C.) geloofde dat materie is opgebouwd uit zeer kleine niet meer deelbare elementen de 'atomos'. Deel je een element verder dan verliest deze zijn typische eigenschap. Ons woord voor atoom is van deze filosofie afgeleid.

Stap: Splits de repeterende groep af.

We splitsen de repeterende groep af in een eigen entiteit. Omdat in de oorspronkelijke entiteit de sleutel door het sleutelattribuut `Studentnr` bepaald wordt nemen we deze over. De nieuwe entiteit is op deze manier gekoppeld aan de eerste. Hiermee verzekeren we de integriteit van de database. Alle gegevens zijn door de koppeling nog steeds te bereiken.

Student	
<u>Studentnr</u>	Studentnr
Naam	Cursuscode
Adres	Cursusnaam
Woonplaats	Startdatum
	Gebouw
	Locatie
	Gebouwnaam

De repeterende groep is afgesplitst naar een eigen entiteit

Stap: Geef aan de entiteit een betekenisvolle naam

We bedenken voor deze repeterende groep een goede naam. Omdat het de gegevens bevat die over de te volgen cursussen gaan zullen we deze entiteit de naam `Opleiding` geven.

Student	Opleiding
<u>Studentnr</u>	Studentnr
Naam	Cursuscode
Adres	Cursusnaam
Woonplaats	Startdatum
	Gebouw
	Locatie
	Gebouwnaam

De nieuwe entiteit krijgt een betekenisvolle naam.

Stap: Maak van de sleutel een primaire sleutel.

De overgenomen sleutel uit de eerste entiteit is niet voldoende om als primaire sleutel te dienen. Een student kan immers meerdere cursussen volgen. De primaire sleutel zal een samengestelde sleutel moeten zijn bestaande uit meerdere sleutelattributen. We kiezen het attribuut `Cursuscode` als tweede sleutelattribuut. Deze heeft unieke attribuutwaarden. We onderstrepen deze ook.

Opmerking: Een (primaire) sleutel moet je zodanig kiezen dat er zo min mogelijk repeterende groepen voorkomen t.o.v. deze sleutel. Het liefst natuurlijk helemaal geen repeterende groep. Meestal kun je een combinatie nemen van de sleutel van de oorspronkelijke entiteit en het attribuut dat in de repeterende groep de sleutelrol vervult.

Normaliseren volgens Codd

De eerste normaalvorm ziet er dan als volgt uit:

1NV:

Student	Opleiding
<u>Studentnr</u>	<u>Studentnr</u>
Naam	<u>Cursuscode</u>
Adres	Cursusnaam
Woonplaats	Startdatum
	Gebouw
	Locatie
	Gebouwnaam

Tweede normaalvorm (2NV)

2NV

1NV+elk attribuut dat maar voor een *deel* van de sleutel afhangt wordt in een eigen entiteit afgesplitst.

De te nemen stappen:

Stap: Bepaal de attributen die maar voor een deel van de sleutel afhangt

Stap: Splits de attributen af in een eigen entiteit met sleutel

Stap: geef de entiteit een betekenisvolle naam

Stap: Bepaal de attributen die maar voor een deel van de sleutel afhangt

Binnen de eerste normaalvorm kunnen attributen zitten die maar voor een deel van de (primaire) sleutel afhangen. Deze groepen zijn dus niet uniek bepaald door de gekozen sleutelattributen. Zo is het attribuut *cursusnaam* niet afhankelijk van *studentnr* maar wel weer van het sleutelattribuut *cursuscode*. De attributen *cursuscode* en *cursusnaam* vormen hiermee een aparte entiteit.

Student	Opleiding
<u>Studentnr</u>	<u>Studentnr</u>
Naam	<u>Cursuscode</u>
Adres	Cursusnaam
Woonplaats	Startdatum
	Gebouw
	Locatie
	Gebouwnaam

De attribuut 'Cursusnaam' hangt voor een deel af van de primaire sleutel.

Opmerking: De attributen *gebouw*, *locatie* en *gebouwnaam* zijn zelfs niet afhankelijk van een deel de sleutel. De attributen die helemaal niet afhangen van een sleutel worden in de derde normaalvorm behandeld.

Stap: Splits de attributen af in een eigen entiteit

De attributen *cursuscode* en *cursusnaam* vormen een eigen entiteit. De sleutelattribuut *cursuscode* die in de oorspronkelijke entiteit geen primaire sleutel was is het in de nieuwe wel. Om de koppeling met de oorspronkelijke entiteit te behouden laten we een kopie van het sleutelattribuut *cursuscode* achter.

Student	Opleiding	
<u>Studentnr</u>	<u>Studentnr</u>	<u>Cursuscode</u>
Naam	<u>Cursuscode</u>	Cursusnaam
Adres	Startdatum	
Woonplaats	Gebouw	
	Locatie	
	Gebouwnaam	

De attributen 'cursuscode' en 'cursusnaam' vormen een eigen entiteit.

Normaliseren volgens Codd

Opmerking: In de nieuwe entiteit zijn alle attributen afhankelijk van een enkelvoudige sleutel; de primaire sleutel.

Stap: geef de entiteit een betekenisvolle naam

We geven deze nieuwe entiteit ook de betekenisvolle naam *curusus*.

Hiermee hebben we de tweede normaalvorm bereikt.

2NV:

Student	Opleiding	Cursus
<u>Studentnr</u>	<u>Studentnr</u>	<u>Cursuscode</u>
Naam	<u>Cursuscode</u>	Cursusnaam
Adres	Startdatum	
Woonplaats	Gebouw	
	Locatie	
	Gebouwnaam	

Derde normaalvorm (3NV)

3NV:

2NV+elk attribuut dat niet afhangt van een andere sleutelattribuut splitsen we af in een eigen entiteit.

De te nemen stappen:

Stap: Bepaal de attributen die niet afhankelijk zijn van een sleutelattribuut.

Stap: Split de attributen af in een eigen entiteit.

Stap: Bepaal het sleutelattribuut.

Stap: Geef aan de entiteit een betekenisvolle naam

Stap: Geef de externe sleutel aan.

Stap: Bepaal de attributen die niet afhankelijk zijn van een sleutelattribuut.

Er kunnen attributen bestaan die niet afhankelijk zijn van sleutelattributen maar van andere gewone attribuut. Zo is in ons voorbeeld de attributen *gebouw*, *gebouwnaam* en *locatie* niet afhankelijk van *studentnr* of *cursuscode*. Deze vormen een daarmee een afzonderlijke entiteit.

Student	Opleiding	Cursus
<u>Studentnr</u>	<u>Studentnr</u>	<u>Cursuscode</u>
Naam	<u>Cursuscode</u>	Cursusnaam
Adres	Startdatum	
Woonplaats	Gebouw	
	Locatie	
	Gebouwnaam	

De attributen 'gebouw', 'locatie' en 'gebouwnaam' zijn onafhankelijk van welk sleutelattribuut dan ook.

Stap: Splits de attributen af in een eigen entiteit.

Deze onafhankelijke attributen worden afgesplitst in een eigen entiteit.

Student	Opleiding	Cursus	
<u>Studentnr</u>	<u>Studentnr</u>	<u>Cursuscode</u>	Gebouw
Naam	<u>Cursuscode</u>	Cursusnaam	Locatie
Adres	Startdatum		Gebouwnaam
Woonplaats			

De attributen die niet afhankelijk zijn van enig sleutel splitsen we af in een eigen entiteit.

Stap: Bepaal het sleutelattribuut.

Als we naar de attributen en hun attribuutwaarden kijken dan is het attribuut *gebouw* uniek. Deze kunnen we gebruiken als primaire sleutel in de nieuwe entiteit.

GEBOUW	LOCATIE	GEBOUWNAAM
2	Utrecht	Paviljoen
1	Utrecht	Toren
7	Enschede	Bastille
4	Zwolle	Vrijhof

Het attribuut 'gebouw' is uniek.

Om de koppeling in stand te houden voegen we *gebouw* weer toe aan de entiteit *Opleiding* waar hij vandaan kwam.

Student	Opleiding	Cursus	
<u>Studentnr</u>	<u>Studentnr</u>	<u>Cursuscode</u>	<u>Gebouw</u>
Naam	<u>Cursuscode</u>	Cursusnaam	Locatie
Adres	Startdatum		Gebouwnaam
Woonplaats	Gebouw		

Van het attribuut 'gebouw' maken we de primaire sleutel. We herstellen de koppeling met entiteit 'Opleiding' door de attribuut 'Gebouw' weer terug te plaatsen.

Stap: Geef aan de entiteit een betekenisvolle naam

Ook nu moeten we een betekenisvolle naam bedenken voor de nieuwe entiteit. Omdat hier bijgehouden word in welk gebouw de cursus plaatsvindt geven we de entiteit de naam *Lesgebouw*.

Student	Opleiding	Cursus	Lesgebouw
<u>Studentnr</u>	<u>Studentnr</u>	<u>Cursuscode</u>	<u>Gebouw</u>
Naam	<u>Cursuscode</u>	Cursusnaam	Locatie
Adres	Startdatum		Gebouwnaam
Woonplaats	Gebouw		

We geven de nieuwe entiteit de naam 'Lesgebouw'.

Stap: geef de externe sleutel aan.

De koppeling tussen de oorspronkelijke entiteit en de nieuwe loopt via het attribuut *gebouw*. Maar het attribuut *gebouw* is in deze entiteit geen sleutelattribuut. Om de koppeling aan te geven maken we van *gebouw* een externe sleutel. Deze geven we aan door het attribuut te onderstippelen.

Student	Opleiding	Cursus	Lesgebouw
<u>Studentnr</u>	<u>Studentnr</u>	<u>Cursuscode</u>	<u>Gebouw</u>
Naam	<u>Cursuscode</u>	Cursusnaam	Locatie
Adres	Startdatum		Gebouwnaam
Woonplaats	<u>Gebouw</u>		

Het attribuut 'gebouw' in de entiteit 'Opleiding' is een externe sleutel. Deze is gekoppeld met de primaire sleutel in de entiteit 'Lesgebouw'.

De derde normaalvorm is dan bereikt.

3NV:

Student	Opleiding	Cursus	Lesgebouw
<u>Studentnr</u>	<u>Studentnr</u>	<u>Cursuscode</u>	<u>Gebouw</u>
Naam	<u>Cursuscode</u>	<u>Cursusnaam</u>	Gebouwnaam
Adres	<u>Startdatum</u>		Locatie
Woonplaats	<u>Gebouw</u>		

Overige normaalvormen

Boyce-Codd Normaalvorm (BCNF): 3NV + elk attribuut is een sleutel.

Vierde normaalvorm (4NV), 3NV + geen relatie mag meerdere afhankelijkheden hebben.

Er zijn nog meerdere normaalvormen zoals de **Vijfde normaalvorm** en de **domeinsleutel normaalvorm**. Deze dienen voor speciale situaties en zijn lastiger toe te passen.

In de meeste gevallen al deze normaalvormen van weinig tot geen nut.

Diagrammen

Relaties

In een Strokendiagram en een Bachmandiagram beschrijf je de relaties tussen de entiteiten op een grafische manier. Er zijn een beperkt aantal relaties te onderscheiden:

één-op-één relatie
 één-op-veel relatie
 veel-op-één relatie
 veel-op-veel relatie

De relaties één-op-veel en veel-op-één zijn natuurlijk spiegelbeelden van elkaar. Zij worden meestal samengenomen.

In een diagram worden de verschillende relaties met pijlen grafisch aangegeven:

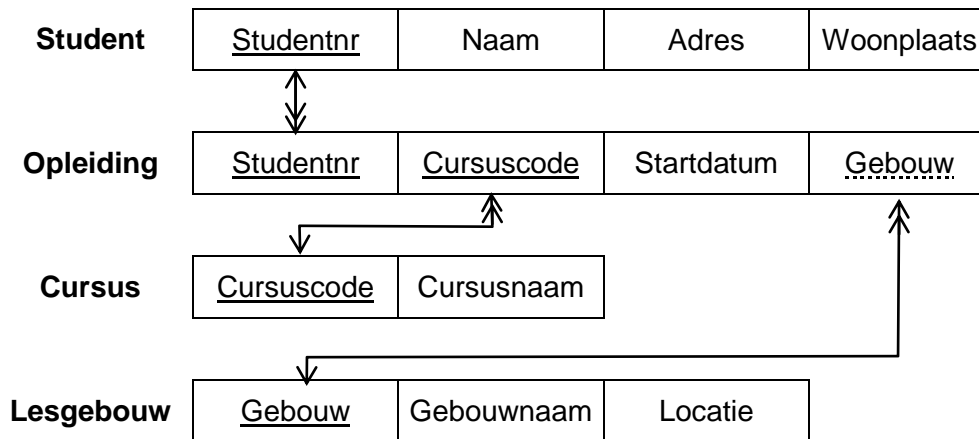
één-op-één relatie één \longleftrightarrow één
 één-op-veel relatie één \longleftrightarrow veel
 veel-op-veel relatie veel \longleftrightarrow veel

Opmerking: Er zijn meerdere manieren om relaties grafisch aan te geven. Wij gebruiken deze pijlen methode.

Het Stroken diagram

In een strokendiagram wordt de *relatie tussen sleutelattributen* (primaire en externe sleutels) van verschillende entiteiten aangegeven.

Van de door ons genormaliseerd inschrijvingsformulier kunnen we het volgende strokendiagram tekenen:



Één student kan vele cursussen volgen.

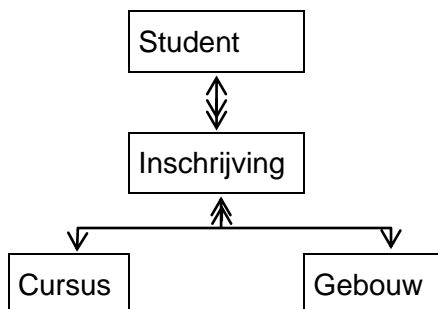
Elke cursus heeft maar één cursuscode, maar er kunnen vele cursussen zijn.

Er zijn vele gebouwen waarin kan worden leggegeven, maar in maar één gebouw een bepaalde cursus volgt.

Het Bachman diagram

In een Bachman diagram worden de *relaties tussen de entiteiten* weergegeven.

Voor ons genormaliseerd voorbeeld van het inschrijvingsformulier kunnen we het volgende Bachman diagram maken:



Een student kan meerdere cursussen volgen.

Op een cursus kan meerdere studenten inschrijven.

Elke cursus wordt gegeven in één gebouw.

Alle studenten volgen dezelfde cursus op dezelfde plek.

Literatuur

[Boek]

Databaseontwikkeling Access 97, Orientatie op de informatieanalyse deel 2,
les Korpershoek, Ben Groenendijk, Bert Pinkster
Academic services, 2000, ISBN 90 395 0889 5

[URL1]: *Home.student.utwente.nl*

http://home.student.utwente.nl/s.p.ekkebus/portfolio/files/Paper_DB_normalisation.pdf

[URL3]: DatabaseAnswers.com

http://www.databaseanswers.com/normal_forms.htm

[

[URL4]: DatabaseAnswers.com

http://www.databaseanswers.com/codds_rules.htm

[URL5]: www.yapf.net

<http://www.yapf.net/faq.php?cmd=100&itemid=700#8>

[URL6]: www.phphulp.nl

<http://www.phphulp.nl/php/tutorials/3/150/259/>

[URL7]: www.sum-it.nl

<http://www.sum-it.nl/cursus/dbdesign/hollands/logis010.php3>

Practicum/opgaven