

## 4 Gegevensanalyse

Informatieanalyse is een systematische benadering van de beschikbare informatie bij het bouwen van applicaties. Dit is nodig vanwege de toenemende complexiteit van softwaresystemen. In deze paragraaf behandelen we de twee methodieken van gegevensanalyse voor datamodellering:

- Top-down-analyse
- Normalisatie-analyse

### 4.1 Inleiding gegevensanalyse

Planning	Inleveren
	Opgaven 1 t/m 3

Databases zijn een essentieel onderdeel van de informatiemaatschappij. Een bedrijf of overheidsinstantie kan niet functioneren zonder databases. Om een database te bouwen begin je met een gegevensanalyse.

Gegevensanalyse stelt een organisatie in staat om interne en externe gegevens te analyseren en structureren. Deze analyses worden gebruikt voor de dagelijkse besturing van de organisatie en voor belangrijke besluitvorming binnen de organisatie.

Als uitgangspunt nemen we het informatiesysteem voor het fictieve bedrijf De WebShop. Dit bedrijf verkoopt muziek online. Zoals iedere organisatie draait De WebShop op informatiesystemen.

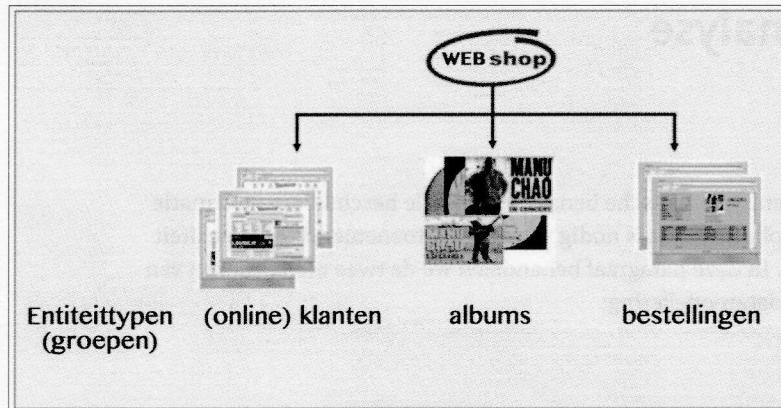
#### Top-down-model

Voor de analyse van informatie van de WebShop passen we een top-down-model toe. Een top-down-model gaat van globale naar gedetailleerde informatie. De elementen van een top-down-model zijn:

- entiteittypes
- entiteiten
- attributen
- sleutels

#### Entiteittypes

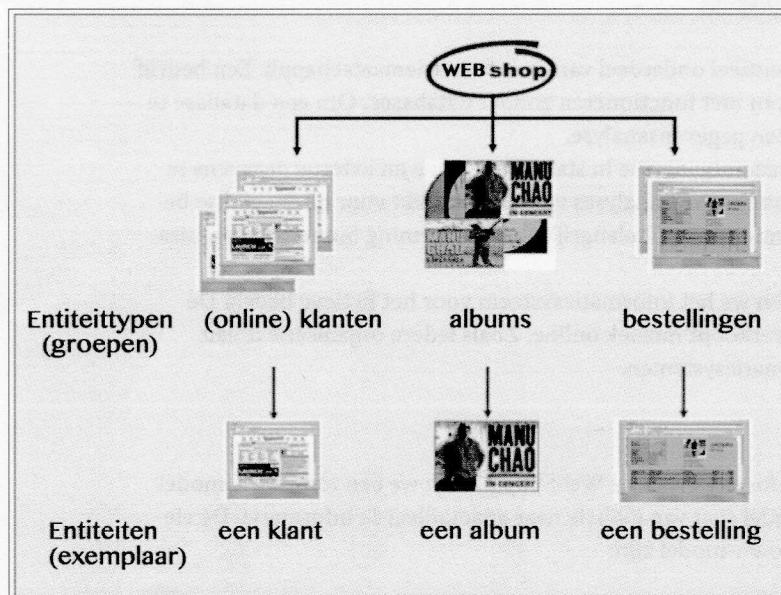
Entiteittypes zijn groepen waarover dezelfde soort informatie wordt bijgehouden. Zie bijvoorbeeld figuur 4.1.



Figuur 4.1 Entiteittypen

### Entiteiten

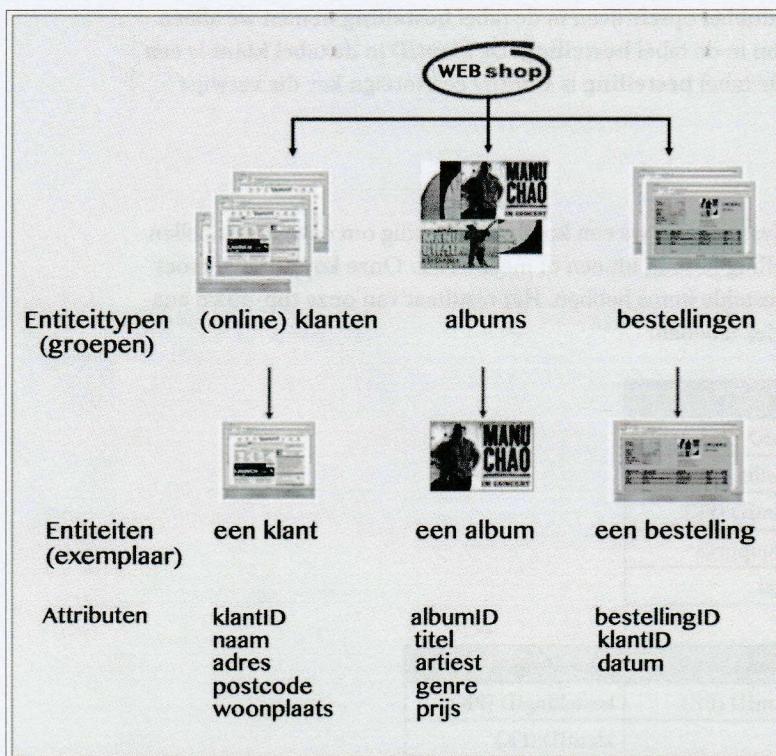
Een entiteit is een bepaald individu/exemplaar van een entiteittype. Bijvoorbeeld een klant is een entiteit van het entiteittype klanten.



Figuur 4.2 Entiteiten

### Attributen

Een attribuut is een *kenmerk* van een entiteit. Bijvoorbeeld naam is een attribuut van de entiteit klant, titel is een attribuut van album en bestellingnummer is een attribuut van een bestelling.



Figuur 4.3 Attributen

### Primaire sleutels (primary keys, PK)

Een sleutel is een attribuut dat een entiteit (uniek) identificeert. AlbumID is bijvoorbeeld een unieke identificatie van een album. Dit noemen we een *primary key* (PK) of primaire sleutel.

Onze top-down-analyse tot nu toe staat in de volgende drie tabellen:

Klant	Album	Bestelling
klantID (PK)	albumID (PK)	bestellingID (PK)
naam	titel	klantID (FK)
adres	artiest	datum
postcode	genre	
woonplaats	prijs	

### Verwijzende sleutels (foreign keys) FK

Een database mag geen dubbele gegevens hebben. Een bestelling heeft klantgegevens nodig en de klantgegevens zijn te vinden in de tabel **klant**. In plaats van dat

we alle klantgegevens dubbel opschrijven in de tabel **bestelling** nemen we alleen het attribuut *klantID* op in de tabel **bestelling**. De *klantID* in de tabel **klant** is een primary key, maar in de tabel **bestelling** is *klantID* een foreign key die verwijst naar de tabel **klant**.

### De koppeltabel

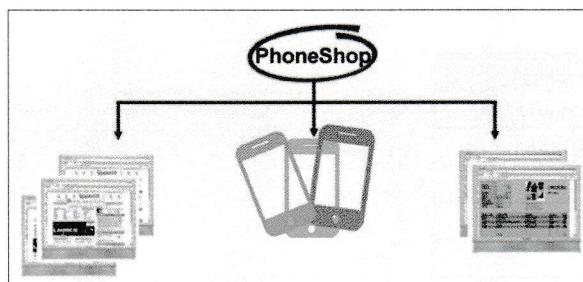
Bij een top-down-analyse hebben we een koppeltabel nodig om deze drie tabellen te koppelen. Een bestelling bestaat uit een of meer items. Onze koppeltabel moet dus gegevens van de bestelde items hebben. Het resultaat van onze top-down analyse zijn de volgende vier tabellen:

Item
itemID (PK)
bestellingID (FK)
albumID (FK)
verkoopprijs
aantal

Klant	Album	Bestelling
klantID (PK)	albumID (PK)	bestellingID (PK)
naam	titel	klantID (FK)
adres	artiest	datum
postcode	genre	
woonplaats	prijs	

- *Opgave 1*

Maak een top-down-analyse voor een PhoneShop die mobiele telefoons online verkoopt.



Figuur 4.4

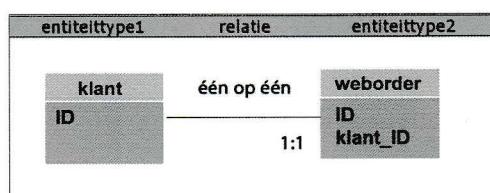
### Wat is een ERD?

Datamodellering is een grafische weergave van de relaties (samenvang) tussen de entiteitstypes. Deze relaties geven we schematisch weer in een Entiteit Relatie Diagram (ERD). Een Entiteit Relatie Diagram is een zeer abstracte beschrijving van een datamodel. Een ERD omvat de grafische elementen die je nodig hebt om de relaties tussen de entiteitstypes in het datamodel te beschrijven. We behandelen de volgende relaties:

- één-op-éénrelatie
- één-op-veelrelatie
- veel-op-veelrelatie

### Eén-op-éénrelatie (1:1)

De één-op-éénrelatie tussen twee tabellen geef je aan met een lijn tussen de tabellen en met de formule of *multiplicity*: (1:1)



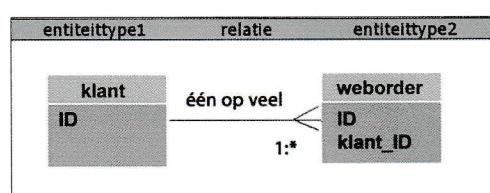
Figuur 4.5 Voorbeeld van één-op-éénrelatie

De relatie lezen we van links naar rechts als volgt: Eén klant mag één bestelling plaatsen.

De relatie lezen we van rechts naar links als volgt: Eén bestelling mag door één klant geplaatst worden.

### Eén-op-veelrelatie (1:\*)

De één-op-veelrelatie tussen twee tabellen geef je aan met een lijn tussen de tabellen en met de formule: (1:\*)



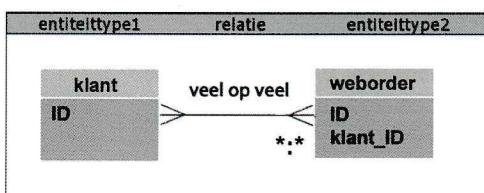
Figuur 4.6 Voorbeeld van één-op-veelrelatie

De relatie lezen we van links naar rechts als volgt: Eén klant mag één of meer bestellingen plaatsen.

De relatie lezen we van rechts naar links als volgt: Eén bestelling mag door één klant geplaatst worden.

### Veel-op-veelrelatie (\*:\*)

De veel-op-veelrelatie tussen twee tabellen geef je aan met een lijn tussen de tabellen en met de formule: (\*:\*)�



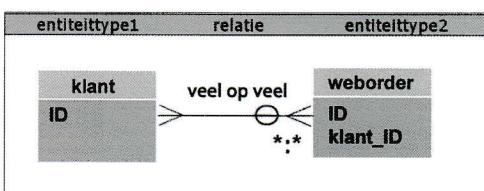
Figuur 4.7 Voorbeeld van veel-op-éénrelatie

De relatie lezen we van links naar rechts als volgt: Eén klant mag één of meer bestellingen plaatsen.

De relatie lezen we van rechts naar links als volgt: Een bestelling mag door één of meer klanten geplaatst worden.

### Nul-optionaliteit

Een nul-optionaliteit betekent dat er *niet* per se een relatie hoeft te zijn.



Figuur 4.8 Voorbeeld van een nul-optionaliteit

De relatie lezen we van links naar rechts als volgt: Eén klant mag geen, één of meer bestellingen plaatsen.

- *Opgave 2*

Hieronder zie je een ERD voor De WebShop. Maak het diagram af door de primary en foreign keys te tekenen en door de volgende relaties te tekenen.

- Een klant mag nul, één of meer bestellingen plaatsen.
- Een bestelling bestaat uit één of meer items.
- Een item bestaat uit een album.



Figuur 4.9

- **Opgave 3**

Maak de top-down-analyse uit opgave 1 af door er een ERD bij te maken voor de mobiele-telefoon-webshop.

### Datamodellering-software

Je zou voor het tekenen van een Entiteit Relatie Diagram (ERD) een freeware datamodellering-programma, zoals *Toad Data Modeler* van Quest Software kunnen gebruiken. De homepage van Quest Software vind je op:  
[www.toadworld.com/m/freeware/1469](http://www.toadworld.com/m/freeware/1469)

## 4.2 Normaliseren

Normaliseren is een bottom-up-aanpak om vanuit bekende informatie een gegevensmodel te ontwerpen. Door het normaliseren van gegevens voorkom je drie problemen:

The screenshot shows a web-based shopping cart interface for 'De WebShop'.

**Customer Information:**

Datum:	6-7-2007
Bestellingnr.:	101
Klantnr.:	20000
Naam:	K. Lant Audio
Adres:	Einsteinstraat 1
Postcode:	1993 AC
Woonplaats:	Arnhem
email:	fe@rrv.nl

**Product List:**

albumID	titel	artiest	genre	prijs	aantal	bedrag
1	Cafe Atlantico	Cesana Evora	World	3,00	1	3,00
2	Rumba Azul	Caetano Veloso	Latin	4,90	1	4,90
6	Der Her ist mein getreuer Hirt	Ton Koopman	Klasiek	5,50	1	5,50
9	Survivor	Destiny's Child	R&B	3,00	1	3,00
3	Oh Girl	The Chi-lites	Other	3,00	1	3,00
						TOTAAL
						19,40

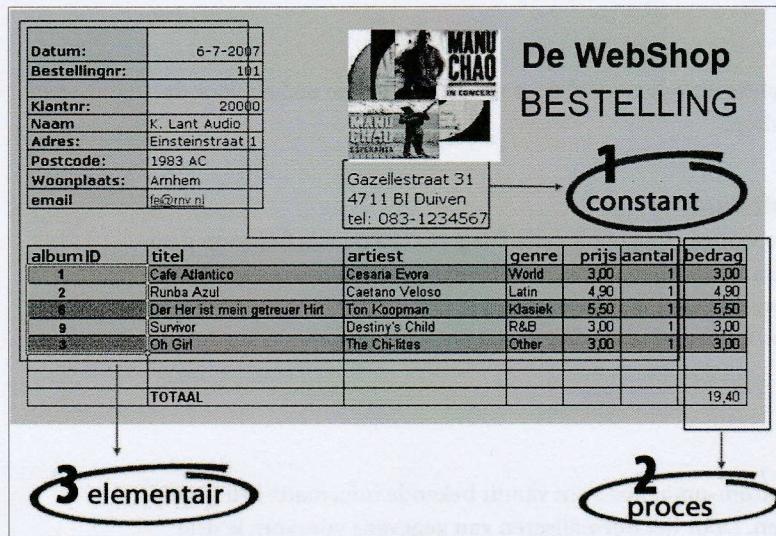
Figuur 4.10 Online bestelling

- 1 Overbodige gegevens: dat zijn gegevens die meermalen ingevoerd moeten worden.
- 2 Inconsistentie: stel dat het adres van een klant verandert en dat deze wijziging op meerdere plaatsen moet worden doorgevoerd. Als dat niet gebeurt, heeft dit een inconsistente (tegenstrijdige) gegevensverzameling tot gevolg.
- 3 Inefficiëntie: het is zeer inefficiënt om dezelfde gegevens op meerdere plaatsen bij te houden.

Dit soort problemen (overbodige, inconsistente en inefficiënte gegevens) kan worden voorkomen met *normaliseren*.

### Soorten gegevens

We nemen als voorbeeld de volgende gegevens van een online bestelling van De WebShop. Er zijn drie soorten gegevens:



Figuur 4.11 Soorten gegevens

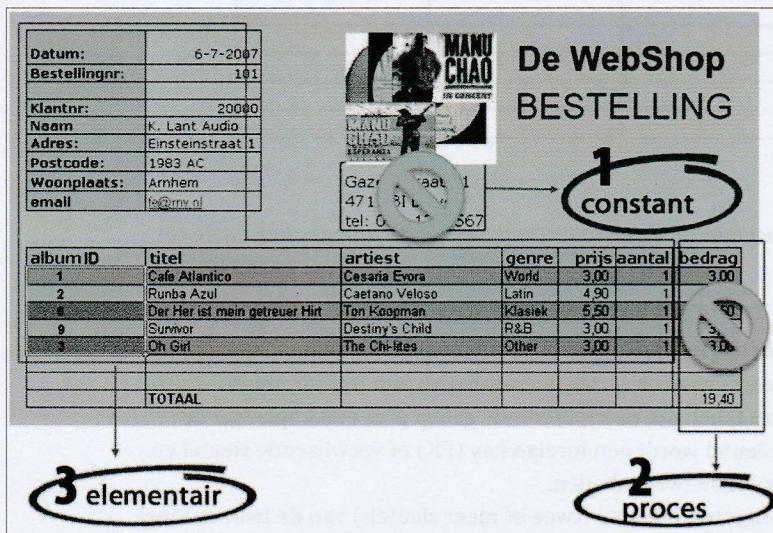
- Constante gegevens: dit zijn gegevens die zelden gewijzigd worden, bijvoorbeeld naam en adresgegevens van een bedrijf of een btw-percentage.
- Procesgegevens: procesgegevens kunnen uit andere gegevens afgeleid worden, bijvoorbeeld bedrag of totaal.
- Elementaire gegevens: dat zijn alle andere gegevens.

### 4.3 Normalisatieproces

Planning	Inleveren
Opgaven 4 t/m 7	

Voor het normalisatieproces beschrijf je de beschikbare gegevens in een tabel. Je let op herhalende gegevens. Deze gegevens noemen we de repeterende groep. Het normalisatieproces kent de volgende vormen:

- nulde normaalvorm
- eerste normaalvorm
- tweede normaalvorm
- derde normaalvorm



Figuur 4.12 De nulde normaalvorm

#### Nulde normaalvorm (0NF)

De nulde normaalvorm doen we in drie stappen:

*Stap 1:* Maak een tabel met alleen de elementaire gegevens. Constante gegevens zo als bedrijfsadres en telefoonnummer neem je niet mee. Ook procesgegevens zoals bedragen en totalen neem je niet mee.

*Stap 2:* Geef de repeterende groep aan. In een bestelling kunnen meerdere albumtitels worden besteld. Albums is de repeterende groep (RG).

*Stap 3:* Geef de primaire sleutel aan (PK). Een primaire sleutel identificeert een unieke entiteit. In dit geval identificeert *bestellingID* een unieke bestelling.

- *Opgave 4*

Maak een tabel met vier kolommen zoals in de volgende figuur. Maak de nulde normaalvorm uit de bestellinggegevens van De WebShop.

ONV	1NV	2NV	3NV
Tabel 1			
(PK) bestellingID			
datum			
klantID			
naam			
adres			
postcode			
woonplaats			
email			
albumID			
titel			
artiest			
genre			
prijs			
aantal			

Figuur 4.13

Het resultaat is een tabel met de elementaire gegevens, een primary key en de repeterende groep.

### De eerste normaalvorm (1NV)

Van de nulde normaalvorm naar de eerste normaalvorm zijn twee stappen.

*Stap 1:* Maak een tweede tabel met de repeterende groep plus een kopie van de primaire sleutel. Deze sleutel wordt een foreign key (FK) of verwijzende sleutel en vormt de koppeling tussen de twee tabellen.

*Stap 2:* Bepaal de samengestelde sleutel (twee of meer sleutels) van de tweede tabel.

Bijvoorbeeld:

Hier maak je een tweede tabel met een kopie van *bestellingID* plus de repeterende groep (RG). Vervolgens maak je een samengestelde sleutel met *bestellingID* plus *albumID*. In dit geval identificeert onze samengestelde sleutel een album uniek in een bestelling in tabel 2.

- *Opgave 5*

Maak de eerste normaalvorm uit de bestellinggegevens van De WebShop. Het resultaat moet er als volgt uitzien:

ONV	1NV	2NV	3NV
<b>Tabel 1</b>			
(PK) bestellingID			
datum			
klantID			
naam			
adres			
postcode			
woonplaats			
email			
<b>Tabel 2</b>			
(FK) bestellingID			
(FK) albumID			
titel			
artiest			
genre			
prijs			
aantal			

Figuur 4.14

Het resultaat is een tweede tabel met de repeterende groep plus een kopie van de primary key.

### De tweede normaalvorm (2NV)

Maak een derde tabel met een kopie van het tweede deel van de samengestelde sleutel in tabel 2 plus de bijbehorende attributen. Het tweede deel van de samengestelde sleutel is *albumID*. Deze sleutel vormt de koppeling tussen tabel 2 en tabel 3.

- *Opgave 6*

Maak de tweede normaalvorm uit de bestelling gegevens van De WebShop. Het resultaat moet er als volgt uitzien:

ONV	1NV	2NV	3NV
Tabel 1			
(PK) bestellingID			
datum			
klantID			
naam			
adres			
postcode			
woonplaats			
email			
Tabel 2	Tabel 3		
(FK) bestellingID	(FK) albumID	(FK) albumID	
(FK) albumID	titel	titel	
	artiest	artiest	
	genre	genre	
	prijs	prijs	
aantal			

Figuur 4.15

Hier wordt tabel 3 gemaakt met een kopie van *albumID* en de bijbehorende attributen (alle attributen van album).

## De derde normaalvorm (3NV)

Kijk of er een tabel is die verder opgesplitst kan worden. Als dat zo is, maak je een vierde tabel met de overtollige attributen. Maak een koppeling tussen de twee tabellen met een primaire sleutel en een verwijzende sleutel. Bijvoorbeeld, tabel 1 bevat zowel klant als bestellinggegevens. In de volgende stap splitsen we tabel 1 in tabel 1 en tabel 4.

- *Opgave 7*

Maak de derde normaalvorm uit de bestelling gegevens van De WebShop. Het resultaat moet er als volgt uitzien:

Figuur 4.16

Het resultaat is tabel 4 met *klantID* en bijbehorende attributen van de klant. Maak *klantID* de primary key in tabel 4 en de foreign key in tabel 1.

Bij het benoemen van de tabellen kijken we per tabel naar de inhoud. We zien dat: tabel 1 is de tabel **bestelling**

tabel 2 is de tabel item

tabel 3 is de tabel album

tabel 4 is de tabel klant

## Samenvatting

In een datamodel mogen we primary keys en foreign keys kopiëren om de tabellen te kunnen linken, maar we mogen geen dubbele gegevens hebben. We mogen bijvoorbeeld geen albumgegevens in een tweede of derde tabel hebben. Als een gegeven gewijzigd moet worden, dan doen we dat maar in één tabel.

Zo zorgen we ervoor dat de gegevens altijd consistent zijn. Er is een uitzondering op het niet kopiëren van gegevens – dat zijn prijzen. Bijvoorbeeld bij de tabel **item** is de verkoopprijs een kopie van de prijs in de tabel **album**. We moeten onderscheid maken tussen de prijs van een album en de verkoopprijs, want de prijs van een album kan altijd veranderen, maar de verkoopprijs blijft hetzelfde.

Bovenstaand proces kun je ook als volgt samenvatten:

bestelling (bestellingID, klantID, datum)

bestelitem (bestellingID, albumID, verkoopprijs, aantal)

album (**albumID**, titel, artiest, genre, prijs)

klant (klantID, naam, adres, postcode, woonplaats, email)

#### 4.4 Opgaven normalisatie

Planning	Inleveren
	Opgaven 8 t/m 13

- *Opgave 8*

In de volgende figuur zie je een bestelling van de PhoneShop.

PhoneShop						
Postbus 20000AA						
Uw bestelling						
Jan Smit		Datum	24 mei 2017			
Telefoonstraat 1		klant-nr	56			
223NN		bestelling-nr	100			
Telecomstad						
<b>Serienummer</b>	<b>Merk</b>	<b>Model</b>	<b>Geheugen</b>	<b>Prijs</b>	<b>Aantal</b>	<b>Totaal</b>
18726343636	Samsung	S7 Zwart	5 MB	679	1	679
34565433456	iPhone	6S	5 MB	769	1	769
<b>Subtotaal</b>						1448
<b>BTW 20%</b>						289,60
<b>Totaal</b>						1723,12

Figuur 4.17

Maak hieronder de normalisatie van de nulde normaalvorm tot en met de derde normaalvorm van de bestelgegevens van de PhoneShop.

Figuur 4.18

- *Opgave 9*

De Wijnshop verkoopt kwaliteitswijn aan een klantenbestand met wijnliefhebbers:

- Elke factuur heeft een factuurnummer.
  - Elke klant heeft een klantnummer.
  - De artikelcode identificeert een unieke wijn.

De Wijnshop heeft momenteel een factureringssysteem dat gebaseerd is op papier.

Hieronder zie je een exemplaar van een factuur.

Factuur				
De Wijnshop		Klant-nr	123	
Spuistraat 22		Frans en Ellen Dijkstra		
8877TT Amsterdam		Servatiusstraat 3		
tel:3456789		2385ZX Zoeterwoude		
Factuur-nr	55			
Factuurdatum	02/02/17			
Code	Artikel	Aantal	Prijs/stuk	Bedrag
1	Selection Maître de Chais	5	50	250
44	Vaqueyras rouge	2	40	80
33	Rosé des Demoiselles	10	20	200
		Subtotaal		530
		BTW	20%	106
		Te betalen		636

Figuur 4.19 Een factuur van De Wijnshop

De Wijnshop wil een database ontwikkelen voor het factureren van zijn klanten. Bestudeer bovenstaand document en normaliseer de gegevens tot en met de derde normaalvorm. Maak een datamodel op basis van de factuur van De Wijnshop.

Figuur 4.20

- *Opgave 10*

Maak een Entiteit Relatie Diagram (ERD) met de relaties tussen de resulterende tabellen van De Wijnshop.

- *Opgave 11*

Boekhandel MAXX bestelt boeken van een aantal uitgevers:

- Elke bestelling heeft een bestelnummer.
- Elke uitgever heeft een uniek uitgevernummer.
- ISBN (International Standard Book Number) is een uniek nummer dat iedere boektitel identificeert.

Boekhandel MAXX heeft momenteel een ordersysteem dat gebaseerd is op papier.

Hieronder zie je een exemplaar van een bestelling.

Boekhandel MAXX						
Marnixstraat 22 2299GF Amsterdam						
Bestelling						
Bestelling-nr	987		Datum	24 mei 2017		
Uitgever-nr	22					
Naam	EduBoek					
Adres	Postbus 48					
Plaats	Culemborg					
ISBN	TITEL	AUTEUR	Aantal	Prijs/stuk	Bedrag	
1238765434	Werkboek MS Word	Reats e.a.	5	50	250	
9376523412	Basis JavaScript	Cremers R	2	40	80	
8373652425	Wiskunde I	Mulder S	10	20	200	
		Subtotaal			530	
		BTW		20%	106	
		Te betalen			636	

Figuur 4.21 Een bestelling van Boekhandel MAXX

Boekhandel MAXX wil een database ontwikkelen voor het bestellen van boeken. Bestudeer bovenstaand document en normaliseer de gegevens tot en met de derde normaal vorm.

Figuur 4.22

- *Opgave 12*

Maak een Entiteit Relatie Diagram (ERD) met de relaties tussen de tabellen van Boekhandel MAXX.

- *Opgave 13*

Maak een datamodel voor de gegevens uit de factuur voor Tele-4-klanten uit de volgende figuur.

Tele-4			
Postbus 2 2299GF			
<b>Uw factuur nr. 903</b>			
P. De Boer Telefoonstraat 1 223NN Telecomstad	Datum	24 mei 2017	
Abonnement: 0612345678			
Bundel	Duur	Prijs	Totaal
Standaard		32,99	32,99
Servicenummers	2 min	0,25	0,5
Bellen naar het buitenland	10 min	0,33	3,3
<b>Subtotaal</b>			<b>36,79</b>
<b>BTW 20%</b>			<b>7,36</b>
<b>Totaal</b>			<b>44,148</b>

Figuur 4.23

Figuur 4.24

- *Opgave 14*  
Maak een Entiteit Relatie Diagram (ERD) met de relaties tussen de tabellen voor het Tele-4-datamodel.

## 4.5 Project Vacaturebank

## Voorkennis

Kennis van en ervaring met webapplicatieontwikkeling met behulp van HTML en JavaScript is vereist. Evenals kennis van een in een serveromgeving gebruikte taal (zoals PHP) en een DBMS (zoals MySQL).

## Algemeen

Dit project gaat over het ontwerpen van een webapplicatie met een database voor een vacaturebank. Dit project voer je individueel uit. Voor dit project wordt ervan uitgegaan dat je hoofdstuk 1 tot en met 4 uit dit boek hebt bestudeerd. Raadpleeg zo nodig bronnen.

## Project

PowerJobs is een ICT-recruitment-bedrijf dat operationeel is in de Randstad. PowerJobs wil een web-database-applicatie ontwikkelen voor een online vacaturebank toegankelijk voor bedrijven en voor sollicitanten.

### Situatie

PowerJobs is een middelgroot ICT-recruitment-bedrijf met relaties met grote bedrijven en de overheid. Op dit moment gaat de meeste werving en selectie via reclames in de kranten. PowerJobs wil een moderne online vacaturebank voor zijn klanten en sollicitanten.

### Jouw functie

Je bent werkzaam als junior applicatieontwikkelaar bij SoftROC en hebt de opdracht gekregen om de webapplicatie PowerJobs te ontwerpen en realiseren. De applicatie moet het hele proces van rekruteren automatiseren.

### De webapplicatie Vacaturebank

Het doel van de applicatie is dat de bedrijven zelf hun vacatures kunnen plaatsen en de sollicitanten zelf kunnen solliciteren.

De sollicitant moet een overzicht van open vacatures kunnen opvragen en daarop solliciteren.

Het adverterende bedrijf moet de profielen van de sollicitanten kunnen zien.  
De sollicitant moet de profielen van de adverterende bedrijven kunnen zien.

PowerJobs factureert het adverterende bedrijf per advertentie: voor elke sollicitatie een vaste commissie van 5% van het jaarsalaris.

De webapplicatie moet in de volgende behoeften voorzien:

- De applicatie moet responsive zijn.
- Een homepage is openbaar en moet een overzicht van drie tot zes open vacatures tonen.
- Een inlogscript met sessies voor het inloggen van de sollicitanten en bedrijven.
- De sollicitant en het bedrijf beheren hun eigen wachtwoord en profiel.
  - Account aanmaken
  - Wachtwoord vergeten
  - Inloggen
  - Profiel wijzigen
- Een welkomspagina voor de sollicitant met:
  - Melding: 'U bent ingelogd als NAAM-SOLICITANT'
  - overzicht open vacatures
  - link naar sollicitatieformulier
  - link naar sollicitatie-history-rapport
  - link naar uitlogpagina
- Een welkomspagina voor bedrijven met:
  - Melding: 'U bent ingelogd als NAAM-BEDRIJF'
  - link naar sollicitanten
  - link naar online facturen
  - link naar advertentie plaatsen
  - link naar uitlogpagina

- Een welkomspagina voor administrator:
  - Melding: ‘U bent ingelogd als NAAM-ADMIN’
  - link naar bedrijventabel met creat/read/update/delete
  - link naar sollicitantentabel met CRUD (Create, Read, Update en Delete)
  - link naar facturentabel met CRUD
  - link naar overzicht sollicitanten per vacature
  - link naar uitlogpagina

### Sollicitatie-history

De sollicitant moet zijn sollicitatie-history online kunnen zien zoals hieronder:

PowerJobs Sollicitatiehistory					
Sollicitant	Bedrijf	Vacature	Functie	Datum	Commissie
G. Sanchez	KPN	100	Developer	1-1-2016	€ 50,00
		101	Webdesigner	15-1-2016	€ 50,00
		102	Database Adm	20-1-2016	€ 50,00
			subtotaal		€ 150,00
	KLM	200	Developer	10-1-2016	€ 50,00
		201	Webdesigner	11-1-2016	€ 50,00
		202	Database Adm	23-1-2016	€ 50,00
			subtotaal		€ 150,00
TOTAAL					€ 300,00

### Online facturen

Bedrijven wensen ook hun online facturen kunnen zien, zoals hieronder:

PowerJobs Factuur					
Bedrijf	Vacature	Sollicitant	Functie	Datum	Commissie
KPN	100	G. Sanchez	Developer	1-1-2016	€ 50,00
		T. Umut	Developer	13-1-2016	€ 50,00
TOTAAL					€ 100,00

## Analysefase

Vóórdat je begint met coderen moet je eerst de volgende analyse en documenten inleveren.

- Behoefteanalyse
- Plan van aanpak (PVA)
- Functioneel ontwerp
- Technisch ontwerp

Maak gebruik van de volgende sjablonen bij de analysefase van dit project.

### Sjabloon 1: Behoefteanalyse

#### Algemeen

Deze sjabloon geeft een algemene beschrijving van de onderdelen van een behoefteanalyse.

#### De kern

Geef een korte beschrijving van het project in maximaal vijf regels. Bijvoorbeeld: informatie over het project, de deelnemers en verantwoordelijkheden, deadlines enzovoort.

#### Bedrijfsnaam

Geef een korte beschrijving van het bedrijf van de opdrachtgever.

#### Aanleiding

De aanleiding om een nieuwe applicatie te realiseren. Beschrijf hier de vraag, de wens, het probleem of 'de uitdaging' van de opdrachtgever.

#### De applicatie

Algemene beschrijving van de applicatie. Bijvoorbeeld, de te bouwen applicatie moet de volgende productinformatie bevatten, enzovoort.

#### De doelen van de applicatie

Maak een beschrijving van de doelen van de applicatie. Wat wil de opdrachtgever met het project bereiken? Bijvoorbeeld, hogere verkoopcijfers, naamsbekendheid, efficiënter werkprocessen.

#### Doelgroepen

Beschrijving van de doelgroepen van de applicatie. Bijvoorbeeld:  
*Klant:* dit is een gebruiker die producten kan bestellen.

#### Vormgeving

Beschrijving van de vormgeving van de applicatie. Bijvoorbeeld, logo's, afbeeldingen.

### Communicatie

Datgene wat je met de gebruikers wilt delen moet je tekstueel of met afbeeldingen op de homepage of de bedrijfspagina (About Us) van de applicatie plaatsen. Bijvoorbeeld, informatie over het bedrijf:

[bedrijfsnaam] is een bedrijf dat gespecialiseerd is in enzovoort. Locatie of Google-map met routebeschrijving.

### Vaak gestelde vragen (FAQ)

Bijvoorbeeld: Hoe kan ik mijn bestelling ruilen?

Informatie over extra korting en weekend-specials, enzovoort.

### Programma van eisen/user-stories

Een programma van eisen (PvE) beschrijft de interactie tussen de applicatie en de gebruikers. Het PvE bestaat uit *users* (gebruikers) van de website en uit *user-stories*. Dit zijn de gebruikerseisen of verhalen van wat de applicatie moet doen.

#### De users

Bij users geef je de gebruikers aan die dagelijks betrokken zullen zijn bij de operationele applicatie. Bijvoorbeeld:

- *Medewerker*: dit is een gebruiker die na het inloggen producten kan bestellen.  
De user kan ook een profiel aanmaken en profielgegevens aanpassen.
- *Administrator*: dit is een gebruiker die de content van de applicatie kan wijzigen.  
Enzovoort.

#### De user-stories

Maak in overleg met de opdrachtgever een duidelijke beschrijving van de functionaliteit. Gebruik hiervoor user-stories in de vorm van korte beschrijvingen van de functionaliteit van wat de applicatie moet doen.

Hieronder zie je een voorbeeld van een wazige en onduidelijke omschrijving van een functionaliteit:

De applicatie moet simpel genoeg zijn zodat een relatief intelligente user zich in de kortste tijd de applicatie eigen kan maken.

Wat is simpel genoeg? Wat is relatief intelligent? Wat is de kortste tijd?

Hieronder zie je een voorbeeld van een duidelijke omschrijving van een functionaliteit:

Wanneer de user op de tekst 'wachtwoord vergeten?' klikt, stuurt de applicatie een e-mail terug met het nieuwe wachtwoord.

User-stories definiëren wat opgebouwd moet worden. Bijvoorbeeld: Als <role> wil ik <functionaliteit> zodat <business waarde>. Maak een tabel zoals hieronder met een overzicht van de user-stories.

# moskow

User-story	Prioriteit Hoog/ Laag	Als <role>	Wil ik <functionaliteit>	Zodat <business waarde>	Getest / Verbeteractie
o1	H	Als klant	Wil ik een bedrijfspagina met NAW-gegevens en een Googlemap routebeschrijving	Zodat ik het bedrijf kan lokaliseren.	
o2	H	Als admin	Wil ik kunnen inloggen	Zodat ik de content van de database kan wijzigen	
o3	L	Als medewerker	Will ik de content kunnen muteren	Zodat de gegevens in de applicatie actueel blijven	

### Analyse user-stories

Alle user-stories moeten geanalyseerd worden op redundantie, relevantie en efficiëntie. Bij ieder user-story stel je de volgende vragen:

- Is deze role relevant voor de applicatie?
- Zijn er te veel aannames in de user-story?
- Is de gewenste actie de beste actie?
- Is er een betere manier om aan de eisen te voldoen?
- Controleer op interdependency (onderlinge afhankelijkheid) van stories?

### Sjabloon 2: Plan van aanpak (PvA)

#### Inleiding

In de inleiding beschrijf je de opdrachtgever, de opdracht en de opdrachtaanvoerder (uit de behoefteaanalyse).

#### Doelstelling

Beschrijf hier de doelstelling: Wat wil de opdrachtgever met het project bereiken? Bijvoorbeeld, hogere verkoopcijfers, naamsbekendheid, efficiënter werkprocessen (uit de behoefteaanalyse).

**Op te leveren resultaten**

Beschrijf hier het projectresultaat: Wat gaat je team maken? Benoem concreet wat het is, maak het 'zichtbaar', 'tastbaar'. Datum voor het aanleveren van het product.

**Risicoanalyse**

Beschrijf hier: Wat kan het resultaat in de weg staan? Waardoor kan de kwaliteit minder worden? Of welke onverwachte gebeurtenissen kunnen voor vertraging zorgen? Welke maatregelen neem je dan (vermijden, bestrijden, verzekeren, accepteren) en wie is voor de uitvoering verantwoordelijk?

**Projectrisico's**

- een derde partij levert niet of levert slecht;
- contractuele zaken;
- lijnwerkzaamheden naast de projecttaken;
- projectcultuur, of het ontbreken hiervan;
- gebrek aan kennis en vaardigheden;
- cultuurverschillen tussen klant en leverancier;
- de mate waarin de producteisen op goede wijze kunnen worden opgesteld;
- de mate waarin de producteisen kunnen worden gehaald, gebruikmakend van bestaande faciliteiten, methoden en technieken;
- de moeilijkheidsgraad en de complexiteit van het project.

Inventariseer je risico's als volgt:

Risico	Maatregel	Verantwoordelijk

**Benodigdheden**

Beschrijf hier wat de benodigdheden zijn. Bijvoorbeeld: kantoorruimte, kantoorsoftware, ontwikkelsoftware, apparatuur enzovoort.

### Begroting

Geef hier de belangrijkste posten en verdeel ze per fase. Maak deze begroting in een Excel-sheet. Bijvoorbeeld:

<b>Begroting</b>				
<Projectnaam>		Bedragen in euro's		
<Naam opsteller dit document>				
<Datum>				
<b>Definitiefase</b>	<b>wie?</b>	<b>uren</b>	<b>tarief</b>	<b>totaal</b>
Overleg eindgebruikers	projectleider	2	120	240
	functioneel ontwerper	2	80	160
Overleg experts	projectleider	2	120	240
	interne experts			
Projectmanagement				
...				
subtotaal definitiefase				640
<b>Ontwerp fase</b>				
Reiskosten	teamleden			
Ontwerpschetsen	ontwerper			
Presentatie ontwerpen	ontwerper			
Materialen				
Content	tekstschrijver			
Behoefteaanalyse	programmeur			
Functioneel ontwerp	programmeur			
Technisch ontwerp	programmeur			
Vormgeving	ontwerper			
Projectmanagement				
+++				
subtotaal ontwerp fase				

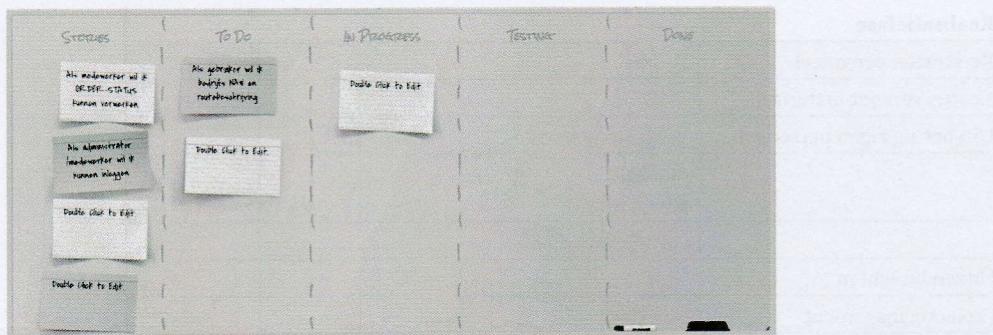
<b>Realisatiefase</b>				
Reiskosten personeel				
Kosten vervoer materiaal				
Uitvoering eigen personeel	programmeur			
	database-administrator			
	vormgevers			
Uitzendkrachten				
Projectmanagement				
+++				
subtotaal realisatiefase:				
<b>Implementatiefase</b>				
Reiskosten personeel				
Afvoer materialen	projectteam			
Implementatie uitvoeren	projectteam			
Opruimen materialen	projectteam			
Eindrapportage	projectleider			
Accountantsverklaring				
Projectmanagement				
+++				
Subtotaal implementatiefase				
<b>Resumé</b>				
subtotaal ontwerpfas				
subtotaal realisatiefase				
subtotaal implementatie				
<b>Subtotaal</b>				
onvoorzien 10%				
<b>Totaal</b>				

Deze begroting is de globale begroting van het gehele project <projectnaam>. Aan het begin van elke nieuwe fase zal een gedetailleerde begroting aangeleverd worden van die fase.

#### Fasering en planning van tussenresultaten

Gebruik een software-ontwikkelmethodiek om de ontwikkelfases te plannen. Bijvoorbeeld SCRUM is een eenvoudige maar effectieve methodiek waar je het programmeren van je user-stories kunt plannen.

Je vindt online een aantal SCRUM-boarden zoals Trello.com. Hieronder zie je een voorbeeld van een SCRUM-board met user-stories.

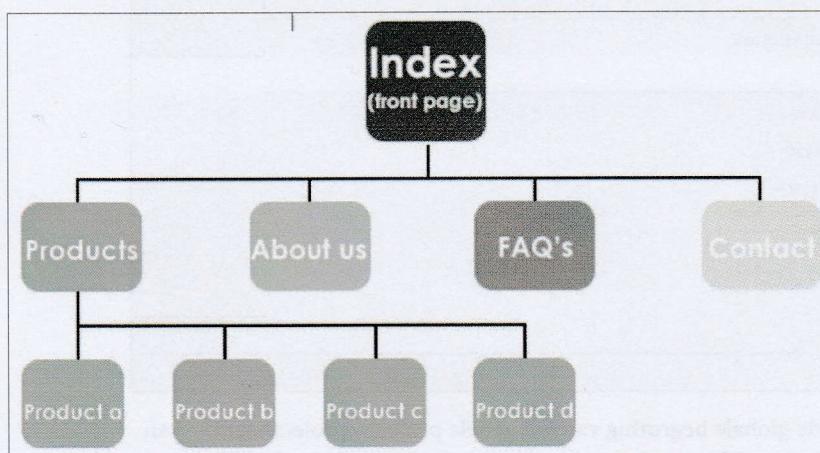


Figuur 4.25

### Sjabloon 3: Functioneel ontwerp

#### Sitemap van de applicatie

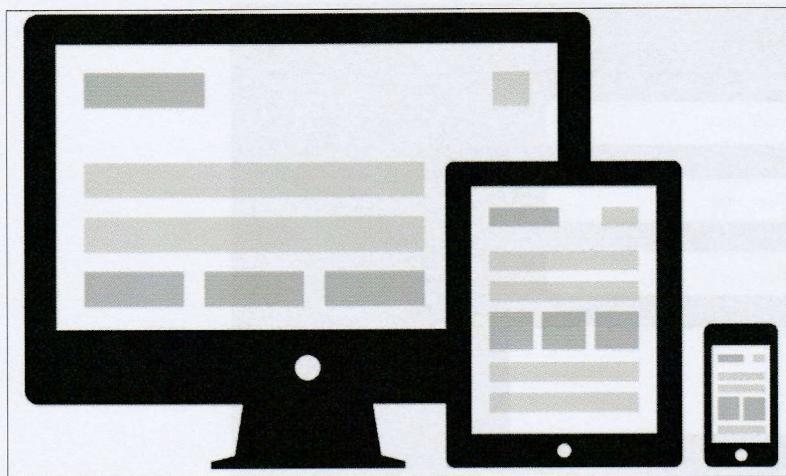
De eerste stap bij het maken van een functioneel ontwerp is het tekenen van een sitemap. Maak een sitemap waarin je de structuur van de site aangeeft. Aan de hand van de sitemap kun je zien of alle gewenste onderdelen van de informatiebehoefte uitgewerkt zijn. Geef de paginanamen en de navigatie van je applicatie aan. Hieronder zie je een voorbeeld van een sitemap.



Figuur 4.26 Voorbeeld van een sitemap

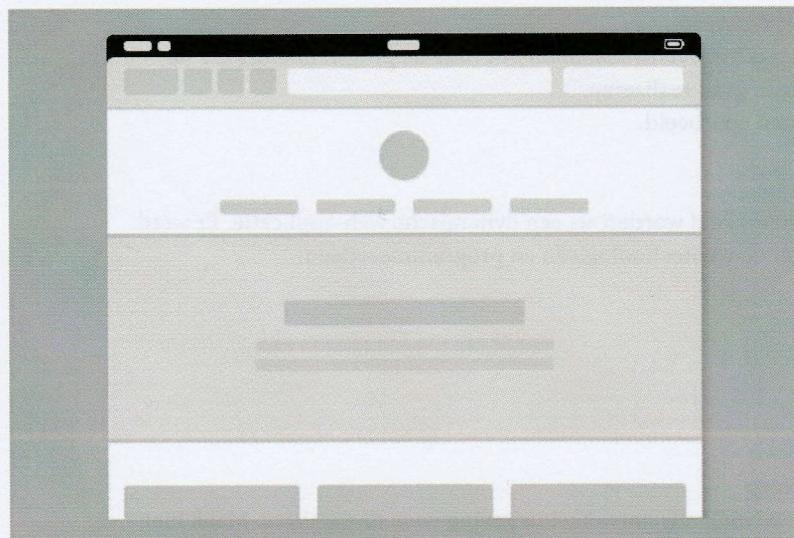
#### Draadmodel van het paginaontwerp

In het tweede deel van het functioneel ontwerp maak je een draadmodel van je paginaontwerp waarin je bijvoorbeeld de banner, navigatie en content-elementen van de pagina aangeeft. Hier geef je je responsive design weer.



Figuur 4.27

Hieronder zie je een voorbeeld van een draadmodel van een paginaontwerp.



Figuur 4.28

#### Formulierontwerp

Het derde deel van het functioneel ontwerp, indien gewenst, is het formulierontwerp. Een voorbeeld van een formulierontwerp zie je hieronder.

The screenshot shows a contact form template with a dark grey background. At the top, it says 'Contactformulier'. Below that is a horizontal line. The form fields are as follows:

- Naam:** A text input field with placeholder text 'placeholder text'.
- Adres:** A text input field with placeholder text 'placeholder text'.
- email:** A text input field with placeholder text 'placeholder text'.
- A large empty text area for a message.
- A 'Submit' button at the bottom left.
- Two 'Form field hint' buttons on the right side of the input fields.

Figuur 4.29

#### Sjabloon 4: Technisch ontwerp

Het technisch ontwerp is het overdrachtsmoment naar de programmeurs die de website en de applicaties gaan realiseren.

Hieronder geven we een voorbeeld.

#### Programmeertalen

De applicatie moet ontwikkeld worden als een dynamische web-applicatie. Er werd gekozen voor de volgende webtechnologieën en programmeertalen:

- HTML5
- CSS3
- Javascript
- PHP
- MySQL

#### Fysiek databaseontwerp

Er werd gekozen voor MySQL als ontwikkelomgeving voor de database. Maak hier een figuur met ERD.

#### Sjabloon 5: Realiseer de applicatie

Bij het realiseren (coderen) van je applicatie moet je eerst een data-dictionary van je database maken zoals in het volgende voorbeeld. Daarna maak je de database aan.

<b>Entiteitnaam</b>	Consultant	
<b>Definitie</b>	Een consultant van ROConsultants	
<b>Attribuutnaam</b>	<b>Datatype</b>	<b>Toelichting</b>
Empcode	Varchar(5)	De code van de employé
Voorletters	Varchar(10)	De voorletters van de employ'r
Tussenvoegsels	Varchar(10)	De tussenvoegsels van een employé
Achternaam	Varchar(25)	De achternaam van een employé
Adres	Varchar(25)	Het (post)adres van de employé
Postcode	Varchar(7)	De postcode die bij het adres hoort
Woonplaats	Varchar(20)	De woonplaats van de employé
Gebruikersnaam	Varchar(20)	De gebruikersnaam van de employé
Wachtwoord	Varchar(20)	Het wachtwoord van de employé
<b>Relatie</b>	<b>Toelichting</b>	
R1	Een consultant heeft geen, een of meerdere kosten	

### Sjabloon 6: Testrapport

Na of tijdens de realisatie van je applicatie test je alle user-stories.

Maak een testlog met een beschrijving van de geteste user-stories uit je behoeft-analyse. Bijvoorbeeld, geef in de rechterkolom aan of de user-story goed werkt of verbeteracties uitgevoerd moeten worden.

User story	Prioriteit Hoog/ Laag	Als <role>	Wil ik <functionaliteit>	Zodat <business waarde>	Getest / Verbeter-actie
o1	H	Als klant	Wil ik een bedrijfs-pagina met NAW-gegevens en een Googlemap routebeschrijving	Zodat ik het bedrijf kan lokaliseren.	Werkt goed
o2	H	Als admin	Wil ik kunnen in-loggen	Zodat ik de content van de database kan wijzigen	Inlog-script niet veilig
o3	L	Als medewerker	Will ik de content kunnen muteren	Zodat de gegevens in de applicatie actueel blijven	Werkt goed