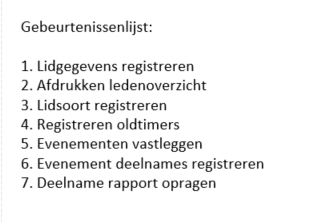
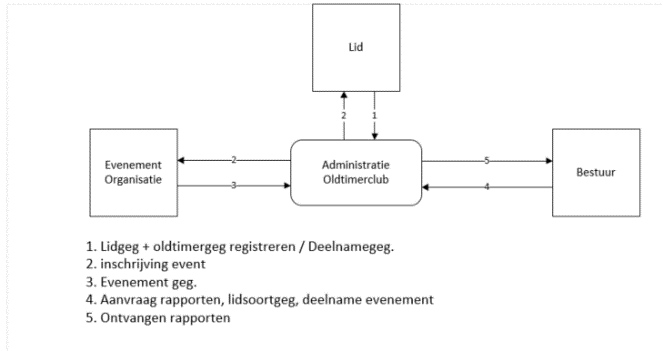
# DFD algemeen

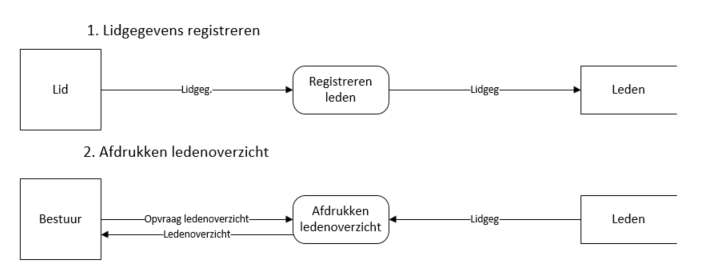
gebeurtenissenlijst = de tekst afgaan en alle werkwoorden aanduiden want deze zijn meestal gebeurtenissen die moeten opgenomen worden. Alternatieven die niet in de tekst staan zoals het betalen van uw personeel of het registeren er van maar dit is niet verplicht en moet ook gezien worden naar de gegeven tijd maar moeten wel worden opgenomen worden als er tijd genoeg is of dit simpel weg in de tekst staat.



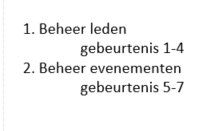
Context diagram = dit is eigenlijk een visualisatie van de gebeurtenissen lijst in het midden hebt je je proces met meestal de administratie van het onderwerp waarover het gaat in dat voorbeeld met dan de interfaces(personen of afdelingen) waar er in het voorbeeld gebruikt worden dus bv klant, bestuur, personeel,…. En dan tussen de administratie en de interfaces gegevensstromen(pijlen) met alle gebeurtenissen uit de gebeurtenissenlijst.



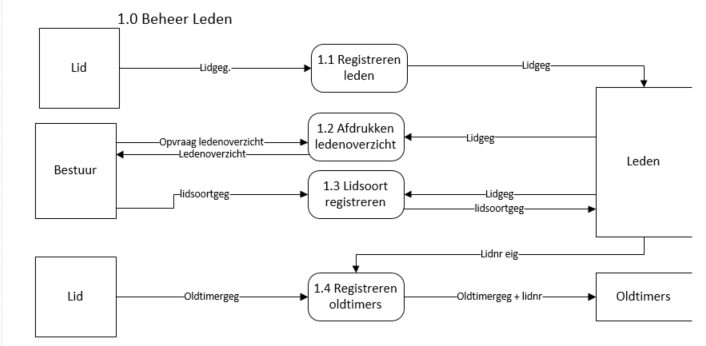
DFD’s = dit is eigenlijk een visualisatie van elke gebeurtennis apart van de gebeurtenissen lijst maar het verschil met het context diagram is dat hier elke gebeurtenis apart word bekeken en hier word er ook gebruik gemaakt van een gegevensopslag(database). Dit kan op verschillende manieren gebeuren bv iets uit een database steken of iets uit halen of beide, opvragen door een bepaalde persoon dan gat er een dubbele pijl eerst naar de proces de aanvraag en dan terug met de gegevens uit de database. Of het kan ook zijn dat er een aanvraag van iemand komt dat moet dit naar het proces gaan door middel van een aanvraag van de klant of personeel of, … en dan moet er data uit één of meerdere database komen en oftewel stop je deze gecombineerde data in een ander database of print je deze of laat je deze ergens zien dan moet deze gegevens terug worden gestuurd naar de persoon die de aanvraag deed. Zo zijn er veel mogelijkheden van situaties die kunnen voorvallen het belangrijkste is dat je bij deze logisch nadenkt en over elke stap goed nadenkt. Dus als er een aanvraag komt dat je niet direct data uit gaat halen maar eerst een aanvraag indient voor het te krijgen van de data die je wil krijgen.



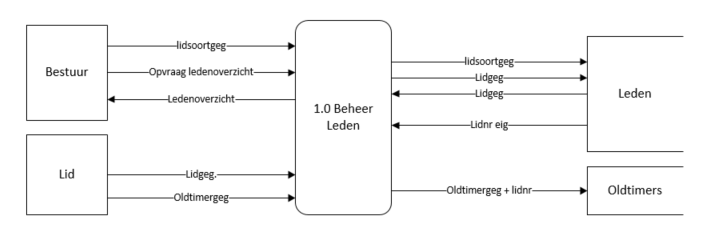
Gegroepeerde gebeurtenissenlijst = dit is eigenlijk het zelfde als de gewone gebeurtenissen lijst behalve dat deze zoals het al zegt gegroepeerd is. Vaak word het gegroepeerd volgens de soort gebeurtenis zoals bv alle registraties of alle betaling maar het kan ook gegroepeerd zijn op basis van waar over het gaat bv alles wat te maken heeft met de leden en daarna alles wat daarna te maken heeft met de evenementen.



Gelaagde DFD’s = dit is eigenlijk een samentrekking van de normale DFD’s en de gegroepeerde gebeurtenissenlijst hier worden alle DFD’s per hoofdpuntje van de gegroepeerde gebeurtenissen gezet dus bv als hier boven alle DFD’s over het beheer van de leden. Dus om dit te krijgen word er eigenlijk gewoon alle DFD’s genomen die je nodig hebt voor je huidige puntje en dan worden er alle dubbele interfaces of dubbele databases uit gehaal en de pijlen naar één algemene verzet. Als dit wegens onoverzichtelijkheid niet mogelijkheid worden ze behouden maar moeten ze wel in een gemeenschappelijk kleur worden aangeduid.



Level1 = dit is eigenlijk juist het zelfde als hierboven maar ipv alle processen apart te gebruiken word de algemene benaming gebruikt van de gegroepeerde gebeurtenissen lijst en word er dus per gegroepeerde gebeurtenissen maar één proces ipv veel meer van de gelaagde DFD’s



# Normalisatie

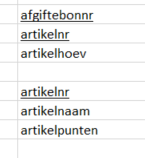
NV0 = dit is de eerste normaal vorm eerst heb je alle gegevens uit een document gehaald door elk ding dat je zien op te schijven. Om dan NV0 te krijgen schap je alle procesgegevens(het bereken van totalen of artikeltotalen) en alle info over het bedrijf(het adres van hebt bedrijf omdat je dit normaal gezien kent) dat je zelf bent.



NV1 = hier kies je eerst een primaire sleutel en dan splits je de gegevens die daar mee te maken hebben die slechts één keer voorkomen en dus dan moet je eigenlijk denken een …. Bevat meerdere … of een …. Bevat juist één …. . Als je deze tabel splits dan mag je niet vergeten bij de tweede tabel een foreig key te zetten zodat er geen verlies is van gegevens.



NV2 = hier word gekeken kan de afhankelijkheid van de waar er gebruikgemaakt word van een FK en PK. Meestal wilt dit zeggen dat er in de tweede tabel de PK gekopieert word en dan naar een tabel word gezet dat aftel enkel zo is voor die situaties of altijd het zelfde is in elke situaties bv een factuur daar heb je artikelnummer artikelprijs omsch, aantal enzo daar kan je bv artikelnummer kopieren en oftewel alles er uit halen en aantal laten staan of omgekeerd.



NV3 = hier doe je eigenlijk het zelfde als in NV2 maar hier ga je naar alles wat niet onderstreept is kijken dus eigenlijk dus een van de beste voorbeelden dat je postcode en gemeente in een aparte tabel zet en enkel postcode over houd in de oorspronkelijke tabel. Hier geef je dan ook als je klaar bent met alle tabellen aan elke tabel een logische naam.

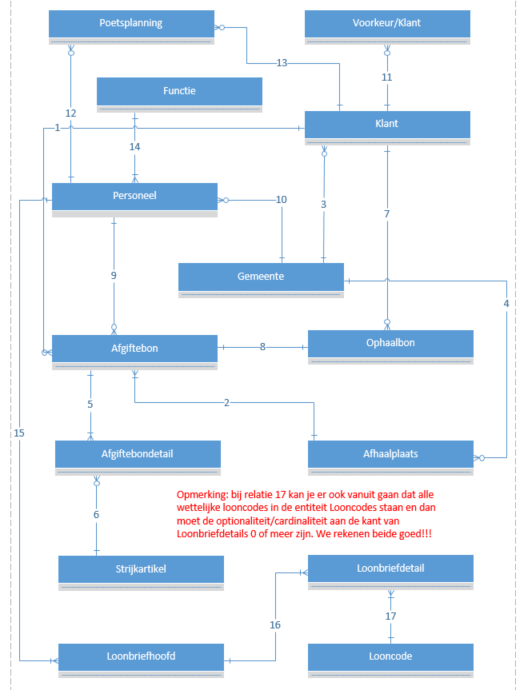


Integratie = dit is eigenlijk alle NV3 van een oefening samen gepakt en de tabellen met ongeveer of de zelfde gegevens samen in één tabel zetten zodat je niet zit met dubbele gegevens. Hier kan je ook PK en FK aanduiden dit kan helpen als je daarna ERD’s moet er van maken.



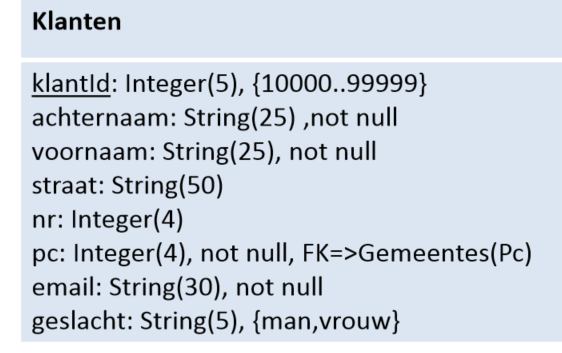
# ERD’s

ERD’s = dit is eigenlijk een visualisatie van de normalisatie. hier word de tabellen uit de integratie of de NV3 genomen dan word er gekeken heeft deze tabel een FK zo ja wilt dit zeggen dat deze tabel een verband heeft met een andere tabel van waar deze FK een PK is dan moet je gaan kijken een …. Heeft die 0,1 of meerder gegevens. Bv een klant heeft juist één postcode en één postcode heeft geen een of meerdere klanten. Dit is vaak gewoon logische redeneren en gaan kijken waar ligt juist het verband tussen twee tabellen.



# Logisch gegevensmodel

Logisch gegevensmodel = dit is eigenlijk zodat je info simpel in een database kan stoppen. Dit kan je best leren aan de hand van een voorbeeld. Hier heeft klantid minstens de waarde 10000 en maximum de waarde 99999 en heeft maximum 5 karakters omdat 99999 5 karakters lang is. Achternaam max 25 lang en mag niet leeg zijn, voornaam hetzelfde, straat maximum 50 karakters lang, huisnummer max 4 karakters lang dus max 9999, pc hetzelfde maar mag ook niet leeg zijn en word de FK ook aangegeven voor de verwijzing tussen tabellen, email max 30 lang en niet leeg, en geslacht maximum 5 karakters en als keuze man of vrouw 5 max lang omdat vrouw 5 karakters lang is en als je een keuze meegeeft dan moet je er altijd voor zorgen dat je maximum waarde gelijk is aan het woord dat het langste is in die keuze.



mogelijke datatypes

integers

string

date

decimal

eerst () eerste getal geeft minimum wat er minstens aantal karakters moet staan tweede maximum aantal karakters

maar kan ook gewoon één getal worde ingegeven dan word het van0 tot u maximum dat je hebt ingegeven

daarna

bv string(15) string met maximum 15 karakters

string(1..30) string met minimum 1 waarde en maximum 30 karakters

int(2),{1..24},not null = string met maximum twee karakters en de waarde zit tussen 1 en 24 en het moet altijd worden ingevuld worden

string (4..20),{cash,overschrijving} een string met als mogelijkheden cash of overschrijving en dus een minimum van 4 karakters van cash dus altijd het woord met minste karakters

decimal(10,2) een getal met max 10 karakters en met twee na de komma dus 8 voor er 2 na

date hoeft geen info bijgegeven te worden