

# INFORMATIEANALYSE VOLGENS HET ER-MODEL

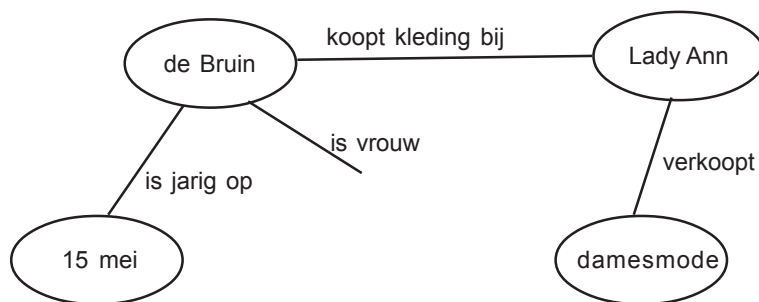


## Werkwijze Informatiemodellering

De kracht van de werkwijze voor informatiemodellering ligt in de communicatie van de informatie-analist met de gebruiker. In de lessen leren we hoe we tot een model moeten komen (werkwijze, recept) en hoe het model grafisch uit te beelden.

De werkwijze is als volgt voor te stellen:

- de informatie-analist en de gebruiker spreken over hoe welke informatie moet worden uitgewisseld in natuurlijke taalzinnen.
- de informatie-analist maakt hiervan plaatjes (entiteit-relationship-diagrammen) gaan over individuele feiten. Bijvoorbeeld:



- gaandeweg ontstaan vele voorbeelden van 'koopt bij', 'verkoopt' enzovoort relationships van entiteiten en van attributen
- de informatie-analist gaat de entiteiten, attributen en de relationships classificeren. Daarbij gebruikt de informatie-analist zoveel mogelijk de begrippen die de gebruiker aandraagt. Het kiezen van de klassen is heel moeilijk en cruciaal voor de kwaliteit van het model. Eén criterium is dat de voor een entiteittype onderkende relationshiptypen zo veel mogelijk op iedere entiteit van het entiteittype van toepassing zijn. Dit is bijvoorbeeld niet het geval als we klant en winkelbediende samenvoegen tot één entiteittype 'persoon'. Het bij dit entiteittype voorkomende attribuuttype 'salaris' is dan op heel wat personen niet van toepassing. Evenmin het relationshiptype 'werkt voor'. Een ander criterium is dat vermeden wordt gelijksoortige relationshiptypen bij verschillende entiteittypen te definiëren. Dit zou bijvoorbeeld het geval zijn wanneer onnodig voor mannelijke en vrouwelijke klanten aparte entiteittypen worden gekozen
- de informatie-analist maakt hiervan een generiek (gegeneraliseerd) model (entiteittypen, attribuuttypen en relationshiptypen, eventueel ook kardinaliteiten):



ET	klant
ID	naam
DE	naam
	geslacht
	verjaardag

ET	winkel
ID	naam
DE	naam
	assortiment

# INFORMATIEANALYSE VOLGENS HET ER-MODEL



Bedenk bij deze werkwijze het volgende:

- winkel is een voorbeeld van een entiteitstype. Een entiteitstype is een klasse gelijksoortige dingen waarover meer dan één soort informatie nodig is. Soort assortiment is daarentegen een attribuuttype. Een attribuuttype is een klasse van gelijksoortige statussen of kenmerken van een entiteit waarover behalve de eigen waarde (hier damesmode) geen informatie nodig is
- iedere relationship is atomair. Dat wil zeggen: ondersteunt één informatiebehoefte. Een informatiebehoefte mag niet verdeeld zijn over verschillende relationships
- er bestaan binaire relationships (2-aire), maar ook ternaire (3-aire), quartenaire (4-aire) enzovoort

Het model wordt uitgebreid met kardinaliteiten. De kardinaliteit geeft aan hoe vaak iets mag of moet voorkomen. In het model wordt dit als volgt aangegeven voor relationshiptypen:



Bij het entiteitstype klant is als kardinaliteitenpaar 0,n opgenomen. Het cijfer vóór de komma geeft de minimumkardinaliteit (toegestane waarden in het model zijn 0 en 1). Het cijfer na de komma geeft de maximumkardinaliteit (toegestane waarden zijn 1 en n). De minimumkardinaliteit 0 geeft aan dat voor een entiteit van het type klant minimaal 0 relationships van het type kledingkopen moeten voorkomen (dit komt neer op het niet verplicht voorkomen van deze relationships). De maximale kardinaliteit n geeft aan dat er hoogstens n (dus net zo veel als je wilt) relationships bij één entiteit van het type klant mogen voorkomen.

## Verschil informatie en gegevens

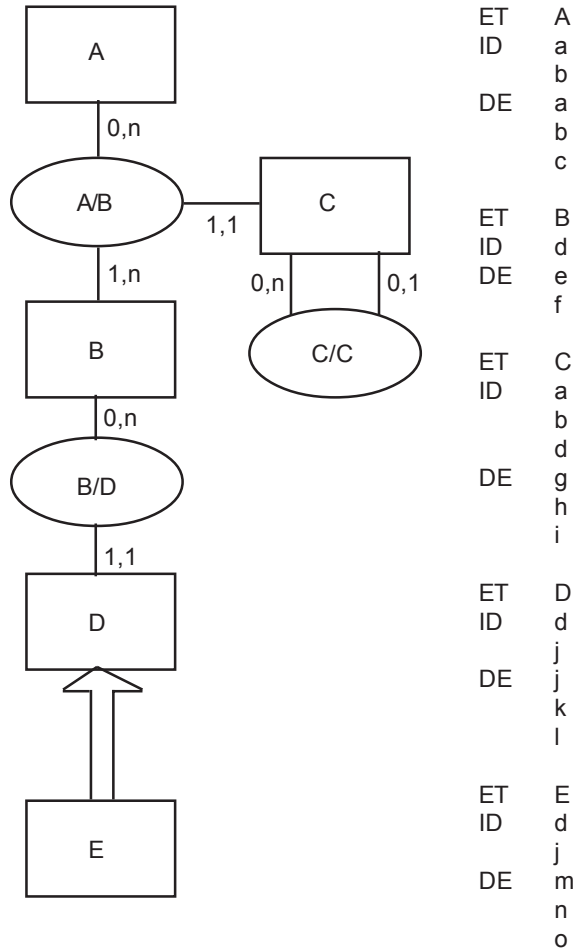
In het bovenstaande wordt uitdrukkelijk gesproken over informatiemodel, informatiebehoefte enzovoort. Er wordt niet over gegevens gesproken. Informatie en gegevens zijn twee verschillende begrippen. Met informatie wordt bedoeld de kennis die door middel van informatiesystemen overgedragen wordt. Met gegevens wordt bedoeld de weergave van de kennis. Hoe kennis weergegeven moet worden, is onderdeel van het ontwerp van een database. Het uitzoeken welke kennis een informatiesysteem moet ondersteunen is onderdeel van de analyse van dit systeem. De kennis is bijvoorbeeld het bedrag in guldens dat voor een artikel betaald moet worden. De weergave zou kunnen zijn '18'. De weergave zou ook kunnen zijn 'achtien' of '10010'(tweetalig stelsel). De keuze hiervoor ligt bij de ontwerper. De informatie-analist houdt zich hier niet mee bezig.

# INFORMATIEANALYSE VOLGENS HET ER-MODEL



## Syntaxis ER-model

De wijze waarop een informatiemodel grafisch en tekstueel moet worden uitgebeeld is:



RELATIONSHIP-TYPE A/B  
 DIMENSION 3  
 COLLECTION A  
 B  
 C  
 CARDINALITY A 0,n  
 B 1,n  
 C 1,1

RELATIONSHIP-TYPE B/D  
 DIMENSION 2  
 COLLECTION B  
 D  
 CARDINALITY B 0,n  
 D 1,1

RELATIONSHIP-TYPE C/C  
 DIMENSION 2  
 COLLECTION C (één keer opnemen)  
 CARDINALITY C 0,N  
 C 0,1

legenda:  
 ET staat voor entity type  
 ID staat voor identifier  
 DE staat voor description

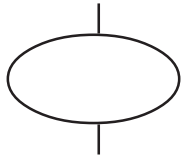
# INFORMATIEANALYSE VOLGENS HET ER-MODEL



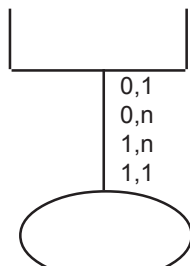
## Toelichting



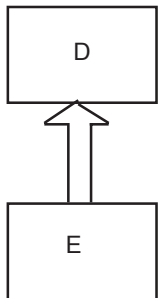
entiteittype; de naamgeving moet aangegeven worden door een zelfstandig naamwoord



relationshiptype; de naamgeving moet aangegeven worden door een heel werkwoord zoals werken, besturen, uitvoeren



kardinaliteit (minimum, maximum) en de toegestane waarden



subtype; entiteittype E is een subtype van entiteittype D

## IDENTIFIER

De opsomming van attribuuttypen die identificerend zijn voor het betreffende entiteittype. Hiertussen kunnen attribuuttypen van andere relationshiptypen voorkomen. Het betreft dan alle attribuuttypen waaruit de identifier van dat andere entiteittype bestaat. Deze attribuuttypen komen dan niet onder de DESCRIPTION voor. Zij zijn immers al vermeld onder de DESCRIPTION van dat andere entiteittype

## DESCRIPTION

De opsomming van attribuuttypen die eigenschappen van entiteiten van het betreffende entiteittype beschrijven

## DIMENSION

Geeft aan of een relationshiptype binair (2), ternair (3) enzovoort is. Unaire komen niet voor. Dit moet als een (status-) attribuut worden uitgebeeld.

## COLLECTION

Geeft een opsomming welke entiteittypen meedoen aan het relationshiptype. Eenzelfde entiteittype kan vaker meedoen. Hierdoor kan het aantal entiteittypen dat opgesomd staat bij COLLECTION kleiner zijn dan het aantal dat bij DIMENSION staat (voorbeeld zie C/C)

# INFORMATIEANALYSE VOLGENS HET ER-MODEL



## Toelichting vervolg

- CARDINALITY** Geeft voor ieder meedoen van entiteitstypen aan welke kardinaliteiten minimaal (0 of 1) en maximaal (1 of n) gelden. Een bepaald entiteitstype kan vaker meedoen aan een relationshiptype. Hierdoor kan het voorkomen dat hetzelfde entiteitstype vaker vermeld staat met dezelfde of juist verschillende kardinaliteiten. De tekstuele weergave en trouwens ook de grafische weergave geven dan geen duidelijkheid over voor welke betekenis welke kardinaliteiten gelden
- OCCURRENCE** Het voorkomen van een entiteit van een entiteitstype

## Classificatie van entiteiten

Het is niet altijd even duidelijk welke entiteitstypen in een model moeten worden opgenomen. Zullen we bijvoorbeeld in het geval van een dansschool één entiteitstype 'persoon' nemen of nemen we apart een entiteitstype 'man' en een entiteitstype 'vrouw'?

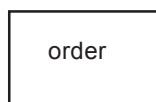
Nemen we in geval van een vereniging een entiteitstype 'lid' of nemen we een entiteitstype 'lidmaatschap' waarmee we dezelfde informatie volgen? Een antwoord hierop is veelal niet gemakkelijk te geven.

In het geval van man en vrouw als entiteitstype kan het bezwaar ontstaan dat we bij beide entiteitstypen grotendeels dezelfde attribuuttypen, relationshiptypen en regels aantreffen en bovendien dat we entiteiten van beide soorten op dezelfde wijze zullen volgen. Is dit het geval, dan kiezen we liever voor het entiteitstype persoon.

In het geval van de keuze tussen een entiteitstype lid en een entiteitstype lidmaatschap is de keuze lastiger. Het entiteitstype lid is een veel rechtstreekser beschrijving van de werkelijkheid, maar lidmaatschap heeft het voordeel dat iemand na stoppen met het lidmaatschap weer met een nieuw lidmaatschap kan beginnen zonder dat administratief de noodzaak bestaat om te kijken naar het oude lidmaatschap. Dit laatste kan onder omstandigheden een voordeel zijn. De keuze zal hier vooral vanuit praktisch oogpunt op het een of het ander vallen.

## Identificatie

Identificatie van een entiteitstype gebeurt door te kijken welk attribuuttype of welke combinatie van attribuuttypen van een entiteitstype voor iedere entiteit van dat type unieke waarden kent. Een eventuele combinatie moet minimaal zijn, dat wil zeggen je moet niet een attribuuttype kunnen weglaten en dan nog steeds unieke waarden hebben. Zie het onderstaande voorbeeld:



ET order  
ID ordernummer  
DE ordernummer  
orderdatum  
leverdatum  
aantal\_colli



ET vestiging  
ID plaatsnaam  
straat  
DE plaatsnaam  
straat  
naam  
oprichtingsdatum  
winkeloppervlakte

# INFORMATIEANALYSE VOLGENS HET ER-MODEL



Bij een identificatie kan eventueel ook een relationshiptype gebruikt worden. In dat geval moet voor dat relationshiptype als kardinaliteti 1,1 gelden. Alle identificerende attribuuftypen van het gerelateerde entiteittype worden in zo'n geval opgenomen in de identificatie. Let op, de attribuuftypen van het gerelateerde entiteittype worden niet in de description (DE) opgenomen. Zie onderstaand voorbeeld:



ET werknemer  
ID wno  
DE wno  
naam  
gebdat  
salaris

ET dossier  
ID wno  
d\_volgno  
DE d\_volgno  
doel  
beheerd door  
opengingsdatum

ET rapport  
ID wno  
d\_volgno  
r\_volgno  
DE r\_volgno  
onderwerp  
datum\_uitgeven

## Objectivering

Soms moeten we zaken vastleggen die niet op een entiteittype betrekking hebben, maar op een relationshiptype. Neem het volgende voorbeeld:



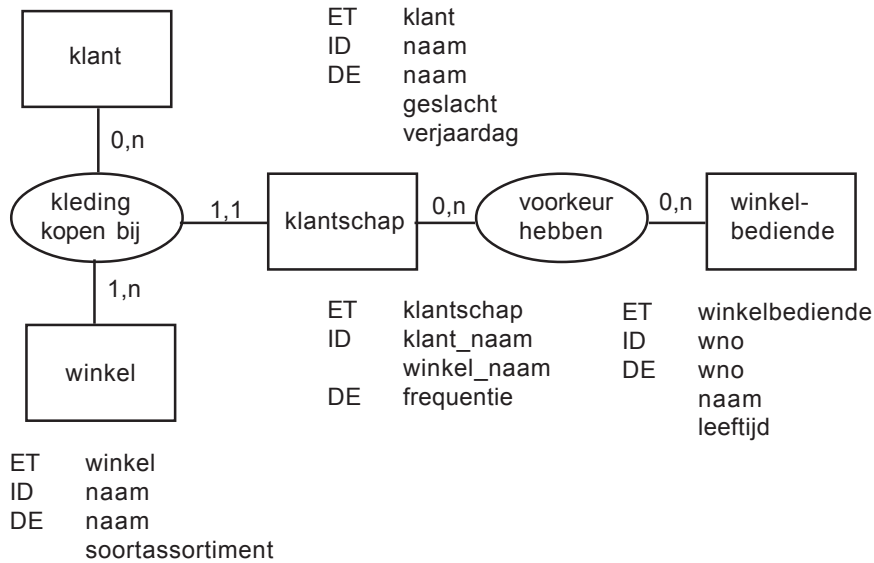
Willen we bijvoorbeeld de frequentie vastleggen dat een klant bij ieder van de winkels kleding koopt (per winkel zal die frequentie verschillend zijn), dan zouden we een attribuuftype moeten opnemen bij het relationshiptype 'kleding kopen' en dat mag niet.

Willen we per vestiging vastleggen welke winkelbediende(n) de voorkeur heeft (hebben) van de klant bij het kleding kopen in die vestiging, dan zouden we een relationshiptype tussen het entiteittype winkelbediende en het relationshiptype kleding kopen moeten opnemen. Ook dat mag niet (een relationshiptype mag geen relatie hebben met een ander relationshiptype). Als oplossing voor deze gevallen moet het relationshiptype kleding kopen geobjectiveerd worden.

# INFORMATIEANALYSE VOLGENS HET ER-MODEL



Dit gaat als volgt:

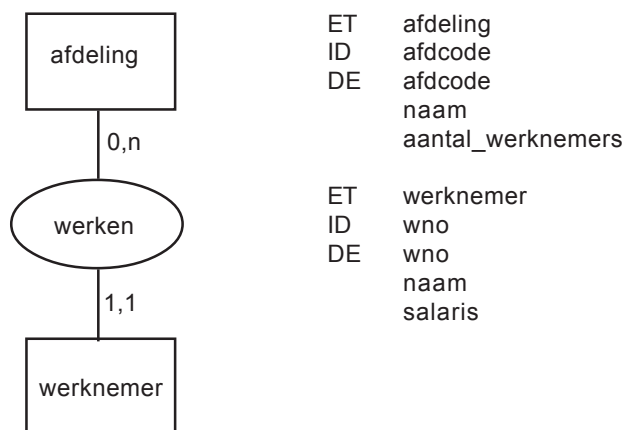


Of iets een objectivering is, valt te zien aan:

- de kardinaliteit 1,1
- dat er maar één klantschap kan voorkomen bij één klant/winkel-combinatie
- dat klantschap geen eigen identificatie heeft (meestal), maar de identificatie van klant en winkel gebruikt

## Afleidbaarheid

Soms komt in een model iets afleidbaars voor. Zie het volgende voorbeeld:



Het attribuuttype 'aantal\_werknemers' is afleidbaar. Dit aantal kan immers bepaald worden door het aantal relationships van een bepaalde afdeling met er werkende werknemers te tellen. We laten het attribuuttype 'aantal\_werknemers' ondanks dat NIET weg. Het model beschrijft immers de informatiebehoefte. Als het 'aantal\_werknemers' een informatiebehoefte is mogen we het niet weglaten. Ook al is het afleidbaar. Dit gaat overigens alleen maar op voor informatiebehoefte die expliciet voor het te bouwen systeem dan wel de beschouwde activiteiten worden onderkend. Het is niet de bedoeling om 'ad-hoc' vragen die zouden kunnen worden gesteld in het model op te nemen (vb. dus niet de medewerkers boven de dertig jaar, onder de twintig enzovoort).

# INFORMATIEANALYSE VOLGENS HET ER-MODEL



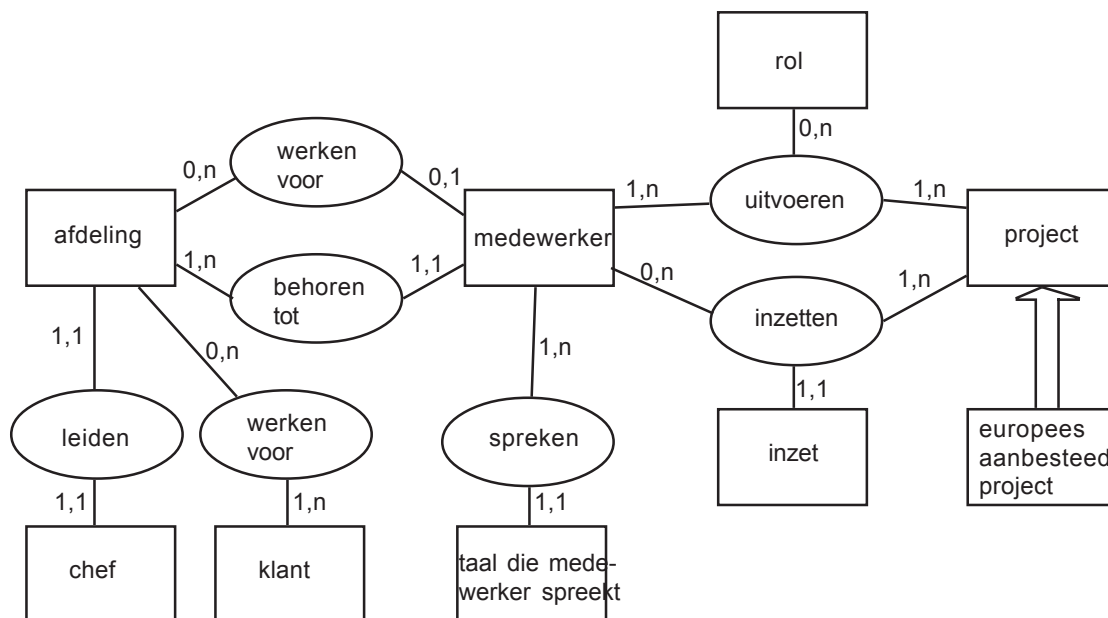
## Omzetten Datamodel in een Database model

In de onderstaande tekst wordt getoond hoe we stap voor stap van een datamodel (informatie-model) een ontwerp van een database (database model) maken (in de vorm van een relationeel schema dat voldoet aan de eisen van het relationele model).

De procedure komt in het kort neer op:

- atap 1 entiteittype (ET) wordt een tabel (relatie volgens het relationele model)
- stap 2 identifier (ID) wordt een primary key
- stap 3 attribuuttype wordt een kolom (attribuut volgens het relationele model)
- stap 4 relationshiptype wordt foreign key(s) + eventueel een tabel
- stap 5 subtype wordt een tabel + een foreign key op de tabel van het supertype
- stap 6 besluit of afleidbaarheid weg moet of niet

We gebruiken het onderstaande ER-model als voorbeeld:



ET	afdeling	ET	klant	ET	taal die medewerker spreekt	ET	rol
ID	afdelingnummer	ID	klantnummer	ID	medewerkernummer	ID	soortrol
DE	afdelingnummer naam adres	DE	klantnummer naam adres	DE	taal	DE	soortrol omschrijving
ET	chef	ET	medewerker	ET	inzet	ET	project
ID	chefnummer	ID	medewerkernummer	ID	medewerkernummer	ID	projectnummer
DE	chefnummer naam adres	DE	medewerkernummer naam adres geboortedatum	DE	projectnummer inzetnummer startdatum geplande duur geplande einddatum werkelijke duur werkelijke einddatum	DE	projectnummer startdatum geplande einddatum werkelijke einddatum werkelijke duur geplande duur
						ET	europees aanbesteed project
						ID	projectnummer
						DE	bedrag



# INFORMATIEANALYSE VOLGENS HET ER-MODEL



Het ontwerp van de uiteindelijke database structuur voor dit voorbeeld is:

tabel	kolommen
chef	<u>chefnummer</u> , chefnaam, chefadres
afdeling	<u>afdelingnummer</u> , afdelingnaam, afdelingadres, <chefnummer>
medewerker	<u>medewerkernummer</u> , medewerkernaam, medewerkeradres, geboortedatum, <werkt_voor_afdelingnummer>, <behoort_tot_afdelingnummer>
medewerkertaal	<u>&lt;medewerkernummer&gt;</u> , taal
klant	<u>klantnummer</u> , klantnaam, klantadres
afdeling_klant	<afdelingnummer>, <klantnummer>
project	<u>projectnummer</u> , projectstartdatum, geplande projecteinddatum, werkelijkeprojecteinddatum, werkelijkeduur, geplandeduur
inzet	<projectnummer>, <medewerkernummer>, inzetnummer, inzetstartdatum, geplande_inzeteinddatum, werkelijke_inzeteinddatum
eupeesaanbestedeproject	<projectnummer>, bedrag
rol	<u>soortrol</u> , omschrijving
med_rol_pro	<soortrol>, <medewerkernummer>, <project nummer>

legenda:

- onderstreping betekent deze kolom(men) behoort (behoren) tot de primary key
- < xxx > betekent deze kolom(men) behoort (behoren) tot een foreign key

Hieronder volgen de stappen in detail:

## Stap 1 Entiteittype wordt een tabel

Voor elk entiteittype wordt een tabel (relatie genoemd in het relationele model) opgenomen. Hieronder zijn ook begrepen de entiteittype die een objectivering zijn (zie taak in het voorbeeld) van een n-op-m relationship en subtypen (zie europees aanbesteed project in het voorbeeld):

tabel	kolommen
chef	
afdeling	
medewerker	
medewerkertaal	
klant	
project	
inzet	
eupeesaanbestedeproject	
rol	

# INFORMATIEANALYSE VOLGENS HET ER-MODEL



## Stap 2 Identifier wordt een primary key

Iedere tabel krijgt als primary key het attribuuttype (of de attribuuttypen) die voorkomen onder ID in de tekstuele beschrijving van het entiteittype waar de tabel op gebaseerd is.

Dit geldt ook voor subtypen en objectiveringen:

tabel	kolommen
chef	<u>chefnummer</u>
afdeling	<u>afdelingnummer</u>
medewerker	<u>medewerkernummer</u>
medewerkertaal	<u>medewerkernummer, taal</u>
klant	<u>klantnummer</u>
project	<u>projectnummer</u>
inzet	<u>projectnummer, medewerkernummer</u>
eupeesaanbestedeproject	<u>projectnummer</u>
rol	<u>soortrol</u>

## Stap 3 Attribuuttype wordt een kolom

Neem in iedere ontstane tabel de attribuuttypen uit de description (DE) van het entiteittype waar de tabel op gebaseerd is op als kolommen (attributen volgens het relationele model):

tabel	kolommen
chef	<u>chefnummer</u> , chefnaam, chefadres
afdeling	<u>afdelingnummer</u> , afdelingnaam, afdelingadres
medewerker	<u>medewerkernummer</u> , medewerkernaam, medewerkeradres, geboortedatum
medewerkertaal	<u>medewerkernummer, taal</u>
klant	<u>klantnummer</u> , klantnaam, klantadres
project	<u>projectnummer</u> , projectstartdatum, geplandeprojecteinddatum, werkelijkeprojecteinddatum, werkelijkeduur, geplandeduur
inzet	<u>projectnummer, medewerkernummer, inzetnummer</u> , inzetstartdatum, geplande_inzeteinddatum, werkelijke_inzeteinddatum
eupeesaanbestedeproject	<u>projectnummer</u> , bedrag
rol	<u>soortrol</u> , omschrijving

## Stap 4 en 5 relationshiptype wordt foreign key(s) + eventueel een tabel / subtype wordt een tabel + een foreign key op de tabel van het supertype

### *n-op-1 relationshiptypen*

Heeft het relationshiptype (bijvoorbeeld 'werken voor' en 'behoren tot') vanuit het entiteittype in onderzoek (bijvoorbeeld 'afdeling') een maximumkardinaliteit n, dan hoeft in de tabel voor dit entiteittype geen foreign key te worden opgenomen.

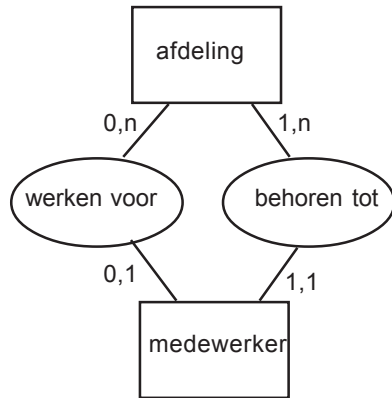
Heeft het relationshiptype (bijvoorbeeld 'werken voor' en 'behoren tot') vanuit het entiteittype in onderzoek (bijvoorbeeld 'werknemer') een maximumkardinaliteit 1, dan moet in de tabel voor dit entiteittype ('werknemer') voor dit relationshiptype een foreign key (<afdelingnummer>) worden opgenomen. Deze foreign key bestaat uit de kolom(men) van de primary key van de tabel waarmee de betreffende tabel door middel van het relationshiptype is geassocieerd (hier 'afdeling'). Zet de kolom(men)

# INFORMATIEANALYSE VOLGENS HET ER-MODEL



van de foreign key tussen haken < >.

Wanneer er zoals in het voorbeeld meer dan één relationshiptype tussen de entiteitstypen voorkomt, dan moet de naam van de foreign key uitgebreid worden om te kunnen onderscheiden op welke relationshiptype de foreign key gebaseerd is. Zie de voorvoegsels 'werkt\_voor\_' en 'behoort\_tot\_'.



ET afdeling  
ID afdelingnummer  
DE afdelingnummer  
naam  
adres

ET medewerker  
ID medewerkernummer  
DE medewerkernummer  
naam  
adres  
geboortedatum

## tabel

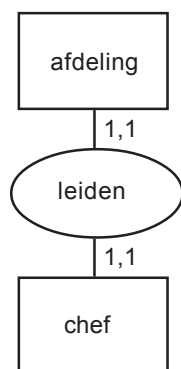
afdeling  
medewerker

## kolommen

afdelingsnummer, afdelingnaam, afdelingadres  
medewerkernummer, medewerkernaam, medewerker-  
adres, geboortedatum, <werkt\_voor\_afdelingnummer>,  
<behoort\_tot\_afdelingnummer>

## 1-op-1 relationshiptypen

Bij een relationshiptype die voor beide gerelateerde entiteitstypen een maximumkardinaliteit van 1 heeft (in het voorbeeld 'leiden' tussen afdeling en chef), mag willekeurig gekozen worden in welk van de twee tabellen (die van afdeling of die van chef) een foreign key wordt opgenomen om de relationship te ondersteunen. Voor de rest geldt het zelfde als bij n-op1 relationshiptypen:



ET afdeling  
ID afdelingnummer  
DE afdelingnummer  
naam  
adres

ET chef  
ID chefnnummer  
DE chefnnummer  
naam  
adres

## tabel

chef  
afdeling

## kolommen

chefnummer, chefnaam, chefadres, <afdelingnummer>  
afdelingnummer, afdelingnaam, afdelingadres

of

chef  
afdeling

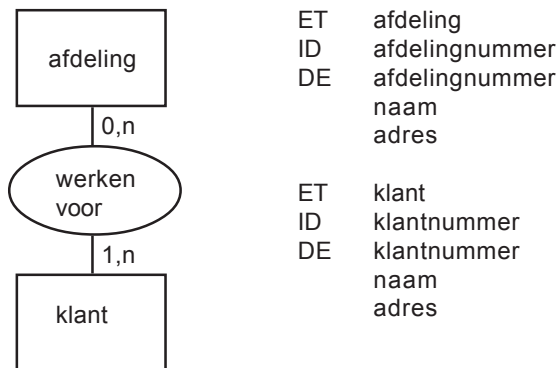
chefnummer, chefnaam, chefadres  
afdelingnummer, afdelingnaam, afdelingadres, <chef-  
nummer>

# INFORMATIEANALYSE VOLGENS HET ER-MODEL



## *n-op-m relationshiptypen*

Voor een relationshiptype die voor beide gerelateerde entiteitstypen een maximumkardinaliteit heeft van n (in het voorbeeld bij 'werken voor' tussen klant en afdeling), moet een aparte tabel worden opgenomen. Deze tabel heeft als primary key de attributen van de ID's van de gerelateerde entiteitstypen samen (hier dus afdelingnummer en klantnummer). De kolommen voor de attribuuttypen van ieder van de gerelateerde entiteitstypen vormen ieder op zich een foreign key (hier dus <afdelingnummer> en <klantnummer>). De tabel ('afdeling\_klant') bestaat dus geheel uit foreign keys die samen de primary key vormen. Andere kolommen zijn er niet:



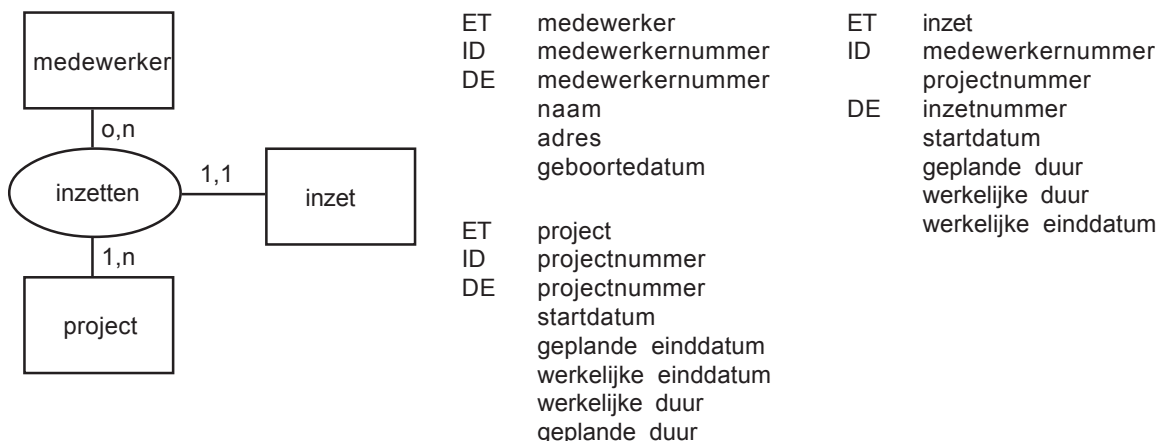
### tabel

### kolommen

afdeling	<u>afdelingnummer</u> , afdelingnaam, afdelingadres, <chefnummer>
klant	<u>klantnummer</u> , klantnaam, klantadres
afdeling_klant	<afdelingnummer>, <klantnummer>

## *objectivering*

In het geval van een objectivering (zie als voorbeeld het entiteitstype 'inzet') bestaat de primary key uit de combinatie van de primary keys van de via het relationshiptype, waarop de objectivering berust (hier 'inzetten'), gerelateerde tabellen (hier zijn dit de primary keys 'medewerkernummer' en 'projectnummer'). Ieder van deze primary keys in de combinatie moet tevens een foreign key worden. Mocht een objectivering geen primary key hebben die gebaseerd is op de primary keys van de gerelateerde tabellen (dit zou het geval zijn geweest als in het ER-model inzetnummer als ID gekozen was), dan moeten de primary keys van de gerelateerde tabellen alsnog opgenomen worden, maar dan alleen als foreign keys:



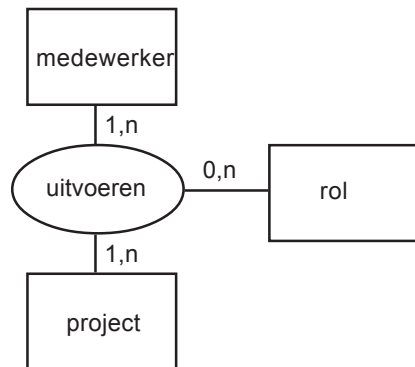
# INFORMATIEANALYSE VOLGENS HET ER-MODEL



tabel	kolommen
medewerker	<u>medewerkernummer</u> , medewerkernaam, medewerker-adres, geboortedatum, <werkt_voor_afdelingnummer>, <behoort_tot_afdelingnummer>
project	<u>projectnummer</u> , projectstartdatum, geplandeprojecteinddatum, werkelijkeprojecteinddatum, werkelijkeduur, geplandeduur
inzet	<u>&lt;projectnummer&gt;</u> , <u>&lt;medewerkernummer&gt;</u> , inzetnummer, inzetstartdatum, geplande_inzeteinddatum, werkelijke_inzeteinddatum

## zuiver ternair

Bij een zuiver ternair relationshiptype (zie als voorbeeld 'uitvoeren'), die voor de drie betrokken entiteitstypen een maximumkardinaliteit n heeft, moet een aparte tabel voor dit relationshiptype opgenomen worden. Het bepalen van de primary key en de drie foreign keys gaat op dezelfde wijze als bij een n-op-m relationshiptype:



ET medewerker  
 ID medewerkernummer  
 DE medewerkernummer  
 naam  
 adres  
 geboortedatum

ET rol  
 ID soortrol  
 DE soortrol  
 omschrijving

ET project  
 ID projectnummer  
 DE projectnummer  
 startdatum  
 geplande einddatum  
 werkelijke einddatum  
 werkelijke duur  
 geplande duur

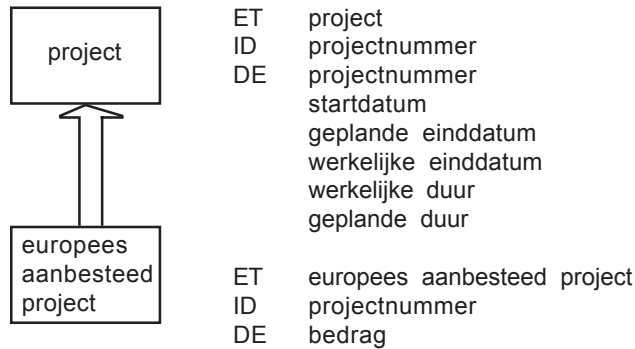
tabel	kolommen
medewerker	<u>medewerkernummer</u> , medewerkernaam, medewerker-adres, geboortedatum, <werkt_voor_afdelingnummer>, <behoort_tot_afdelingsnummer>
project	<u>projectnummer</u> , projectstartdatum, geplandeprojecteinddatum, werkelijkeprojecteinddatum, werkelijkeduur, geplandeduur
rol	<u>soortrol</u> , omschrijving
med_rol_pro	<u>&lt;soortrol&gt;</u> , <u>&lt;medewerkernummer&gt;</u> , <u>&lt;projectnummer&gt;</u>

# INFORMATIEANALYSE VOLGENS HET ER-MODEL



## subtypen

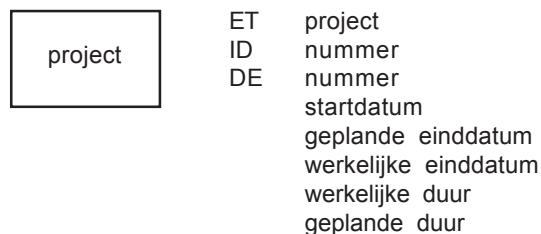
In iedere tabel die is gemaakt voor een subtype (als voorbeeld 'europees aanbesteed project') vormt de primary key ('projectnummer') tevens een foreign key op de tabel die voor het supertype (hier 'project') is gemaakt:



tabel	kolommen
project	<u>projectnummer</u> , projectstartdatum, geplandeprojecteinddatum, werkelijke projecteinddatum, werkelijkeduur, geplandeduur
europeesaanbesteedproject	< <u>projectnummer</u> >, bedrag

## Stap 6 Besluit of afleidbaarheid weg moet of niet

Bij het ontwerpen van databases wordt er meestal naar gestreefd zo min mogelijk kolommen op te nemen waarvan de inhoud kan worden afgeleid uit andere zaken (meestal kolommen) in de database. Men schrijft soms wel voor om afleidbaarheid altijd te vermijden. Dit is soms erg onpraktisch. Het relationele model schrijft zelfs voor afleidbaarheid altijd weg te laten tenzij informatie verloren gaat. Een goede database ontwerper weegt de voordelen van het opnemen van afleidbare kolommen (soms zijn zelfs hele tabellen afleidbaar) af tegen de nadelen. Een voordeel is dat het afleiden zelf wordt voorkomen. Afleiden is soms veel rekenwerk. Een nadeel is dat wanneer de database niet goed beheerd wordt, de inhoud van de kolommen elkaar potentieel kan tegenspreken. Een voorbeeld van het opnemen of weglaten van afleidbaarheid is het al dan niet opnemen van 'werkelijke duur' en 'geplande duur':



tabel	kolommen
project	<u>projectnummer</u> , projectstartdatum, geplandeprojecteind- datum, werkelijkeprojecteinddatum, werkelijkeduur, geplandeduur
of	
project	<u>projectnummer</u> , projectstartdatum, geplandeprojecteind- datum, werkelijkeprojecteinddatum

# INFORMATIEANALYSE VOLGENS HET ER-MODEL



## Opstellen Bachmandiagrammen

### Algemeen

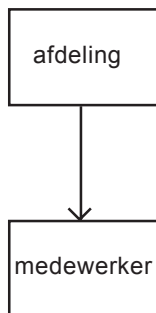
Het verband tussen tabellen in een relationele databaas wordt weergegeven met behulp van foreign keys. Wordt een ontwerp van een relationele database erg groot, dan is niet goed meer te overzien waar de foreign keys naar verwijzen. Een hulpmiddel om de foreign keys overzichtelijk is beeld te krijgen is het opstellen van een Bachmandiagram.

### Bachmandiagram

In het Bachmandiagram krijgt iedere tabel een rechthoek met de naam van de tabel erin. Iedere foreign key wordt weergegeven door een pijl die begint bij de rechthoek van de tabel waar de foreign key naar verwijst en met een punt bij de rechthoek van de tabel waar de foreign key in voorkomt.

### Voorbeeld

Een voorbeeld hiervan is het volgende:



#### tabel

afdeling  
medewerker

#### kolommen

afdelingnummer, afdelingnaam, afdelingadres  
medewerkernummer, medewerkernaam, medewerker-  
adres, geboortedatum, <afdelingnummer>,

### Aard van de relatie

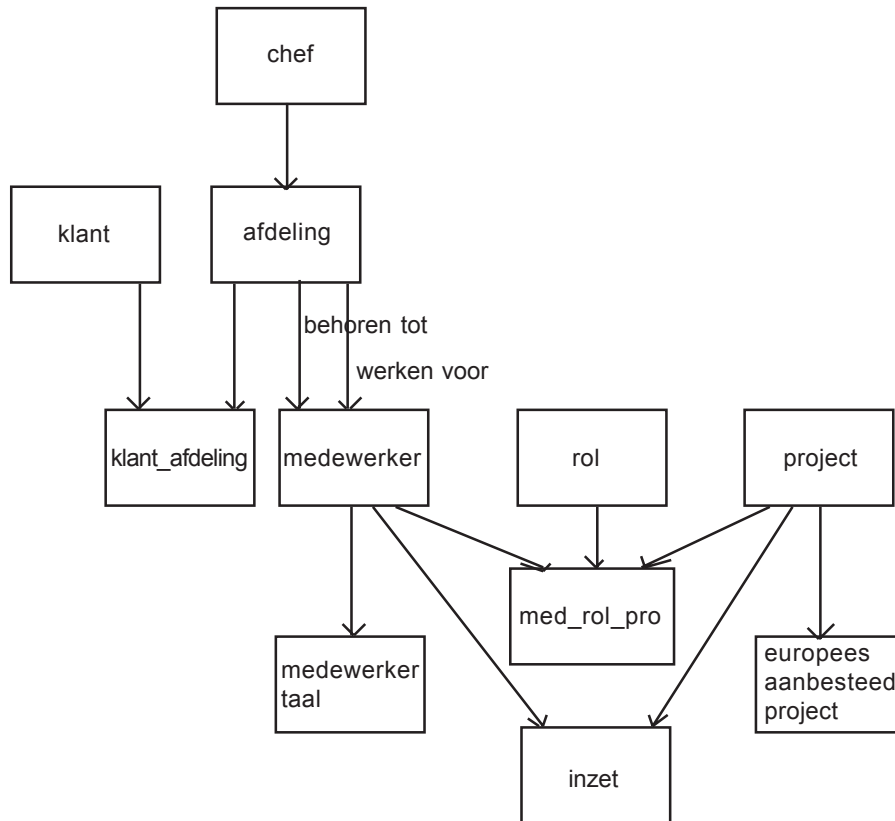
De pijl in het Bachmandiagram geeft een 1:n relatie tussen twee tabellen weer. Het bovenstaande is echter ook van toepassing op tabellen waar een 1:1 relatie tussen voorkomt. Ook daar wordt de pijlpunt gericht op de rechthoek van de tabel waar de foreign key in voorkomt.

# INFORMATIEANALYSE VOLGENS HET ER-MODEL



## Groot voorbeeld

Hierna volgt het Bachmandiagram voor het voorbeeld dat bij het omzetten van een datamodel naar een database ontwerp is gebruikt:



Het Bachmandiagram is gebaseerd op de volgende databasestructuur:

tabel	kolommen
chef	<u>chefnummer</u> , chefnaam, chefadres
afdeling	<u>afdelingnummer</u> , afdelingnaam, afdeling adres, <chefnummer>
medewerker	<u>medewerkernummer</u> , medewerkernaam, medewerkeradres, geboortedatum, <werkt_voor_afdelingnummer>, <behoort_tot_afdelingnummer>
medewerkertaal	<medewerkernummer>, <u>taal</u>
klant	<u>klantnummer</u> , klantnaam, klantadres
afdeling_klant	<afdelingnummer>, <klantnummer>
project	<u>projectnummer</u> , projectstartdatum, geplandeprojecteinddatum, werkelijke projecteinddatum, werkelijkeduur, geplandeduur
inzet	<projectnummer>, <medewerkernummer>, <u>inzetnummer</u> , inzetstartdatum, geplande_inzeteinddatum, werkelijke_inzeteinddatum
europeesaanbesteedproject	<projectnummer>, bedrag
rol	<u>soortrol</u> , omschrijving
med_rol_pro	<soortrol>, <medewerkernummer>, <projectnummer>