



H5 Relationeel model.

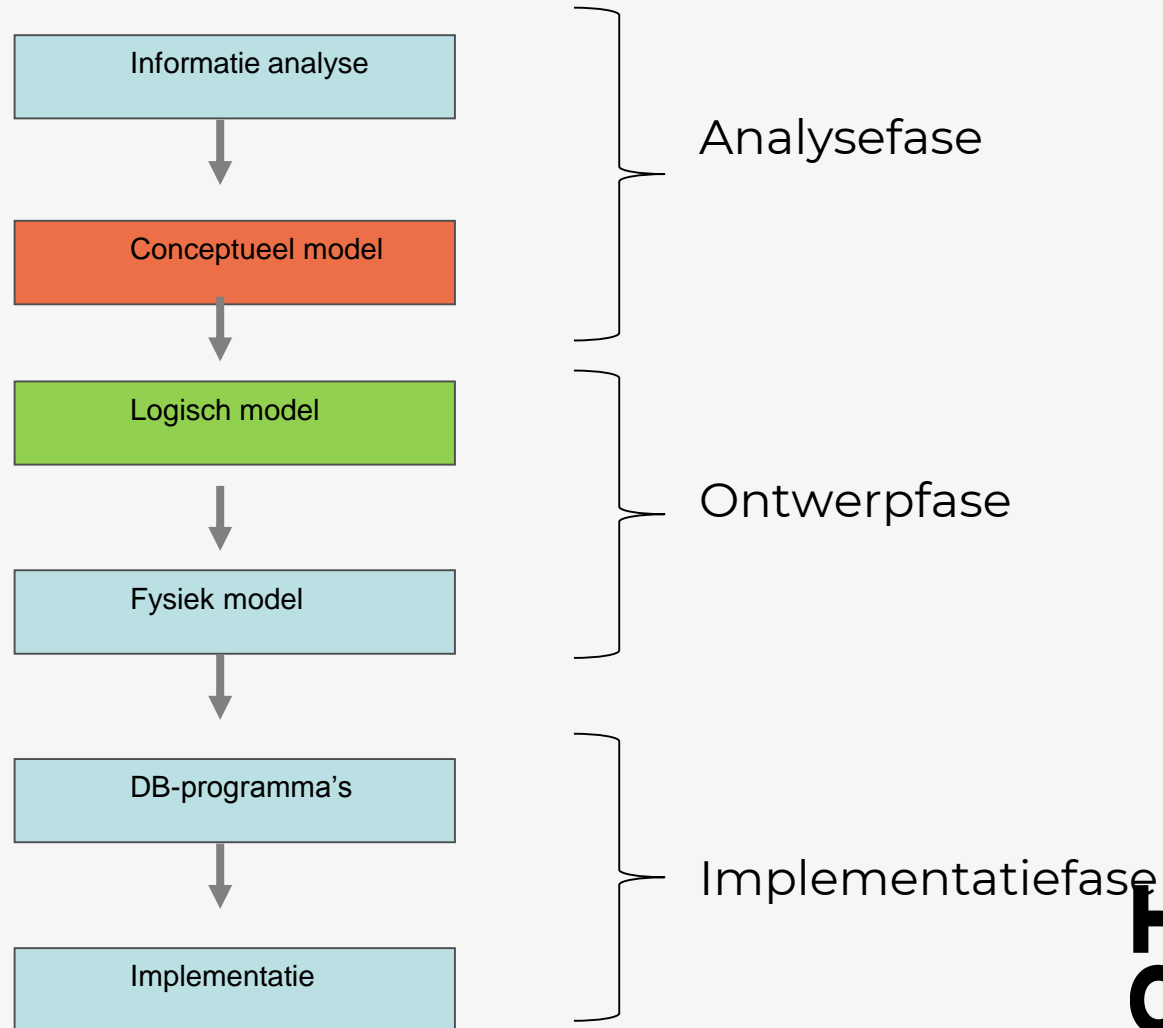
Relationeel Model
Mapping Conceptueel model → Relationeel model



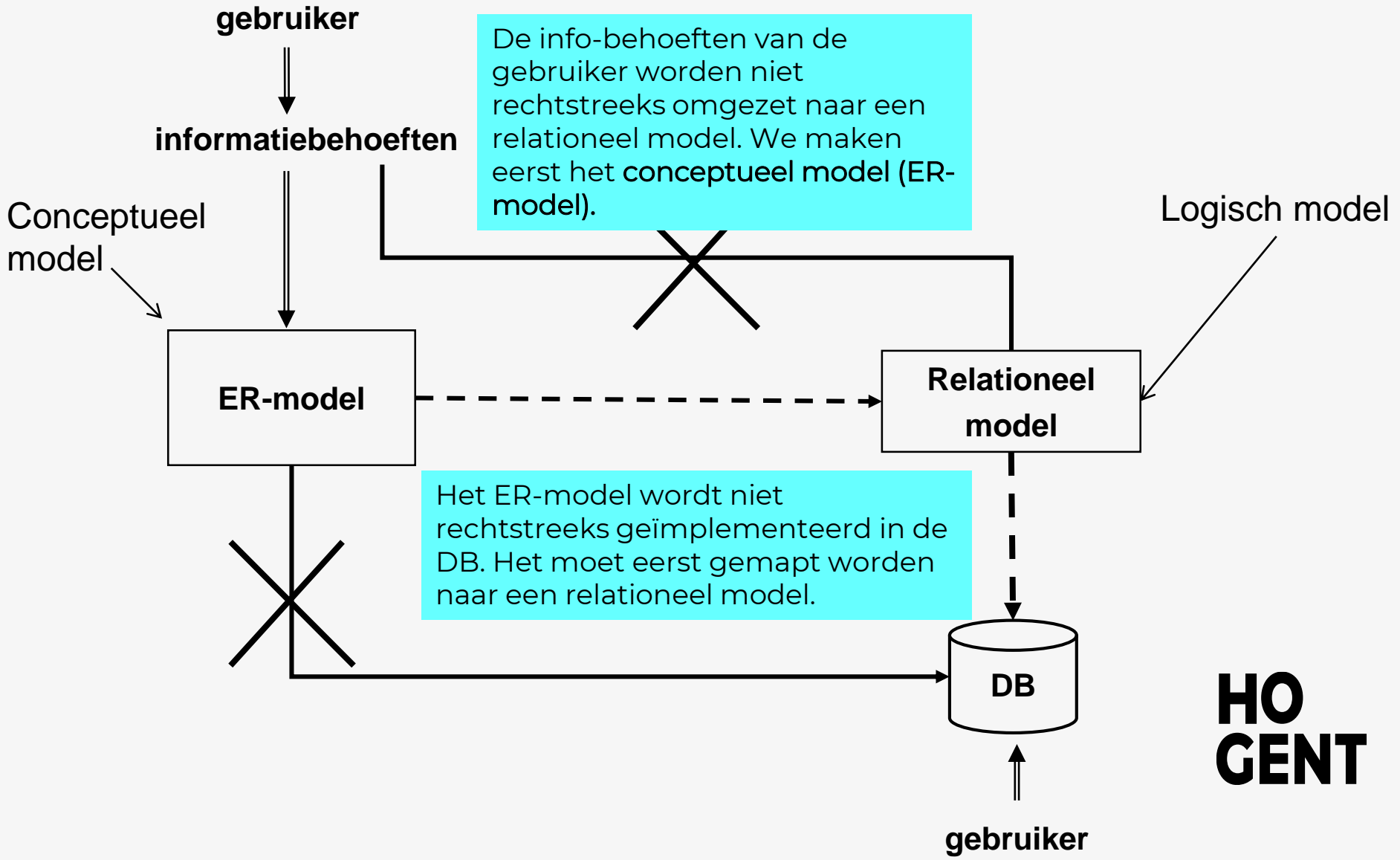
Relationeel model.

Fasen in de ontwikkeling van een DB

Stappen in de ontwikkeling van een databank:



Relationeel model



Relationeel model

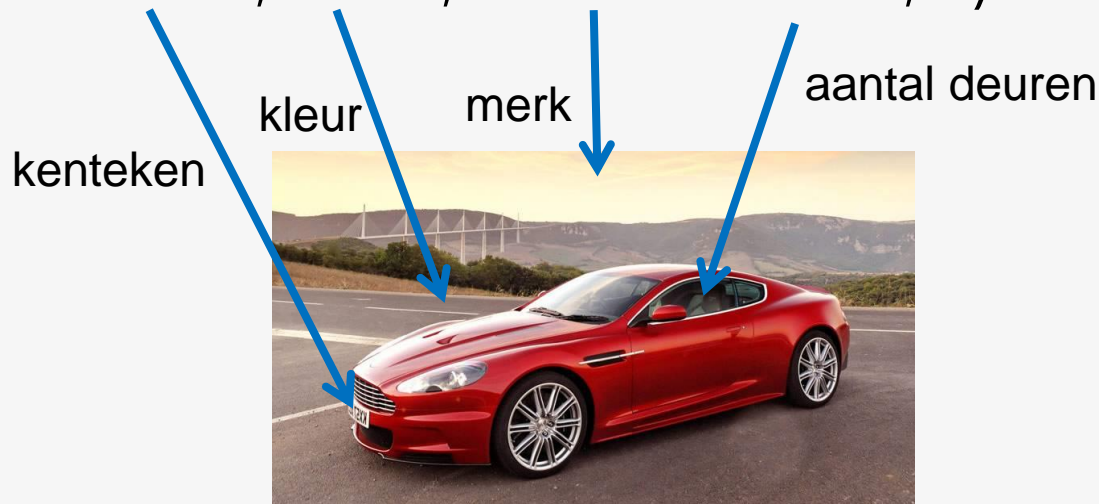
Een relationeel model bestaat uit **relaties** (tabellen) die met elkaar verbonden zijn via **vreemde sleutels**.

Terminologie:

- Tupel
- Attribuut
- Domein
- Relatie
- Attribuuotype
- Sleutels

Definities

- **Tupel (record)**
 - Een geordende lijst met waarden van kenmerken die een object beschrijven.
 - Een tupel is steeds uniek.
- Voorbeeld:
Een tupel voor het beschrijven van een auto:
(kx57 754, rood, Aston Martin, 2)



Definities

- **Attribuut**

- Een benoemd kenmerk van een tupel.
- Een attribuut van een tupel mag geen meerdere waarden hebben, de attribuutwaarde moet éénwaardig en atomair zijn.

- **Voorbeeld:**

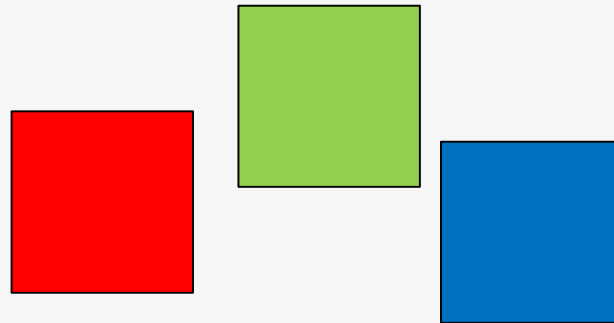
Het attribuut “*kleur*”, een benoemd kenmerk van een auto.

(kx57 754, **rood**, Aston Martin, 2)



Definities

- **Domein**
Een **beperkte** verzameling van **mogelijke (toegelaten) waarden** die voor de attributen in de tupels van een relatie kunnen worden gebruikt.
- **Voorbeeld:**
Het domein "*kleuren*" is een **beperkte** verzameling van alle mogelijke kleuren die een auto kan hebben: rood, groen, blauw,



Definities

- **Attribuuttype**

Elk attribuut is van een bepaald type. Dit type wordt afgeleid uit het domein.

- **Voorbeeld:**

Het attribuut “*aantal deuren*” duidt een aantal aan, het domein zijn alle getallen tussen 2 en 5. Bijgevolg moet het attribuuttype numeriek zijn.

Definities

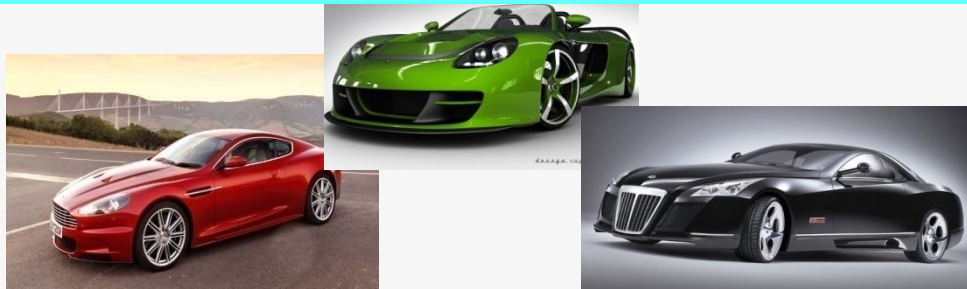
- **Relatie**

Een verzameling van tupels die gelijksoortige objecten beschrijven.

- Voorbeeld:

Een verzameling van tupels die auto's beschrijven.
{(kx57 754, rood, Aston Martin, 2), (xhd 352, groen, Porsche, 2), (123 klm, zwart, Maybach,2)}

Opgelet! Niet verwarren met een 'relatie' uit het ER-model!



Definities: samengevat

Klanten

1	Jansens	Jan	9000	Gent
2	Peeters	Piet	9050	Gentbrugge
3	...			

Diagram illustrating database concepts using a table of customers (Klanten):

- A red arrow points from the cell containing "Peeters" to the label "attribuutwaarde" (attribute value).
- A green arrow points from the entire row containing "Peeters" to the label "Tupel" (tuple).
- A blue arrow points from the entire table to the label "Relatie" (relation).

- **Relatie** = verzameling van tupels → **tabel**
- **Tupel** = geordende lijst met attribuutwaarden → **rij**
 - ✓ In een relatie is elke tupel uniek.
- **Attribuutwaarde** = inhoud van 1 veld → **cel**
 - ✓ Moet éénwaardig en atomair zijn.

ER-model $\Leftarrow == \Rightarrow$ Relationeel model

ER Model	Relationeel model
Entiteittype	Relatie (tabel)
Entiteiten	Tupels (record)
Attribuuttype	Attribuuttype
1-1; 1-N; N-N, 1 ^{ste} graad, 2e graad en >	1-1; 1-N, 1e graad en 2e graad (Geen hogere)

Regels

Elke relatie in het relationele model moet voldoen aan volgende regels:

- Elk tupel is **uniek**.
- Elk attribuut is **éénwaardig**.
- Elk attribuut is **atomair** (kan niet meer opgesplitst worden).
- Verbanden tussen relaties worden gelegd aan de hand van **vreemde sleutels**.

Elk tupel is uniek

- Sleutels
zorgen ervoor dat elk tupel uniek is.
- Welk attribuut maakt elk tupel in deze verzameling uniek?

{(kx57754, rood, Aston Martin, 2),
(xhd352, groen, Porsche, 2),
(123klm, zwart, Maybach, 2),
(456aze, rood, Maybach, 2)}

Elk tupel is uniek

- Uniek attribuut: nummerplaat

{ (kx57754, rood, Aston Martin, 2),
(xhd352, groen, Porsche, 2),
(123klm, rood, Maybach, 2),
(456aze, geel, Maybach, 2) }

→ nummerplaat = sleutel

Elk tupel is uniek

Er zijn verschillende soorten sleutels:

- Kandidaatsleutels
- Primaire sleutel
- Alternatieve sleutels

Sleutels

Kandidaatsleutel

- Is een minimale verzameling van attributen in de tupels van een relatie, waarvan de combinatie elk tupel, binnen die relatie, uniek kan identificeren.



Sleutels

Kandidaatsleutel

Voorbeeld:

Kandidaat Sleutels					
Results		Messages			
	ProductID	Name	ProductNumber	MakeFlag	Finish
1	1	Adjustable Race	AR-5381	0	0
2	2	Bearing Ball	BA-8327	0	0
3	3	BB Ball Bearing	BE-2349	1	0
4	4	Headset Ball Bearings	BE-2908	0	0
5	316	Blade	BL-2036	1	0
6	317	LL Crankarm	CA-5965	0	0
7	318	ML Crankarm	CA-6738	0	0
8	319	HL Crankarm	CA-7457	0	0

Sleutels

Primaire sleutel

- Uit de kandidaatsleutels wordt één sleutel gekozen. Dit is de primaire sleutel.
- De primaire sleutel moet **steeds ingevuld** zijn (NULL niet toegelaten).



Sleutels

Primaire sleutel

Kandidaat Sleutels					
	ProductID	Name	ProductNumber	MakeFlag	Finish
1	1	Adjustable Race	AR-5381	0	0
2	2	Bearing Ball	BA-8327	0	0
3	3	BB Ball Bearing	BE-2349	1	0
4	4	Headset Ball Bearings	BE-2908	0	0
5	316	Blade	BL-2036	1	0
6	317	LL Crankarm	CA-5965	0	0
7	318	ML Crankarm	CA-6738	0	0
8	319	HL Crankarm	CA-7457	0	0
9	320	Chainring Bolts	CB-2903	0	0
10	321	Chainring Nut	CN-6137	0	0
11	322	Chainring	CR-7833	0	0
12	323	Crown Race	CR-9981	0	0
13	324	Chain Stays	CS-2812	1	0
14	325	Decal 1	DC-8732	0	0

Primary Key

HO
GENT

Sleutels

Alternatieve sleutel

- Elke kandidaatsleutel die geen primaire sleutel geworden is, wordt een alternatieve sleutel genoemd.
- De alternatieve sleutel moet niet steeds ingevuld zijn (NULL toegelaten) ...
➔ maar ... indien in een tupel de alternatieve sleutel NULL-waarde heeft dan verliest de alternatieve sleutel zijn functie van 'sleutel'!

Verbanden tussen relaties

Vreemde sleutel



Verbanden tussen relaties

Vreemde sleutel

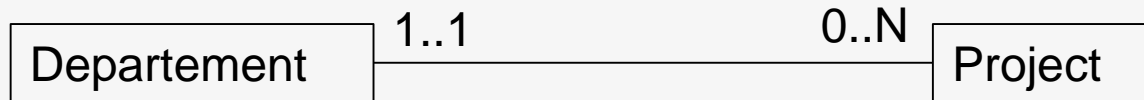
- Een vreemde sleutel (*foreign key*) is de **verbindende schakel** tussen twee relaties. Met een waarde uit een rij van de ene relatie kun je in een andere tabel de juiste tupel met gerelateerde gegevens opzoeken.
- De vreemde sleutel in de ene tabel **verwijst naar de primaire sleutel** uit de andere tabel.

Verbanden tussen relaties

Vreemde sleutel

Voorbeeld:

Een project wordt steeds uitgevoerd in een bepaald departement.



Departement (dnr, dnaam, dlokatie, ...)

Project (pnr, pnaam, pduur, *departement*)

departement in de relatie Project is de vreemde sleutel die verwijst naar de primaire sleutel dnr in de relatie Departement.

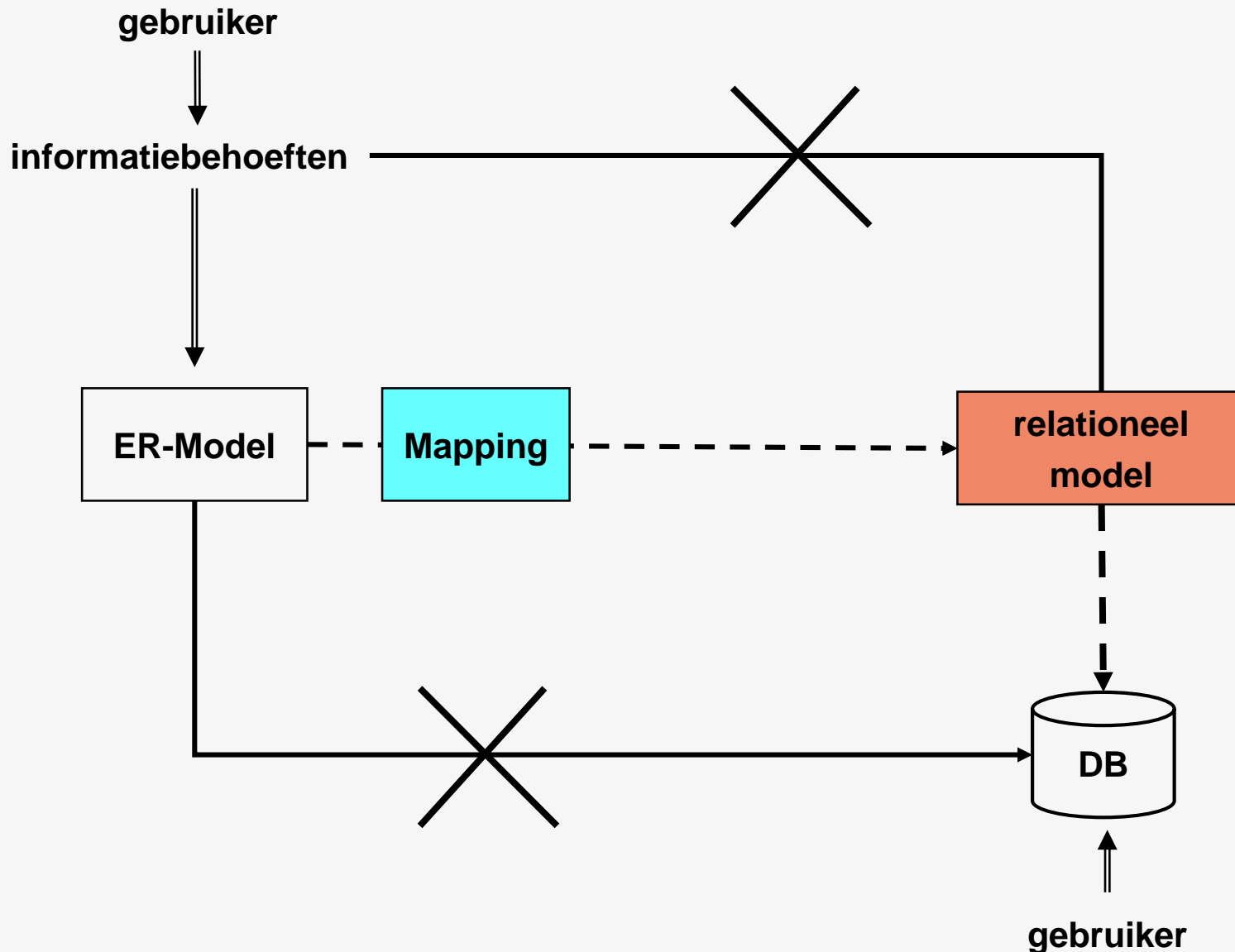


Mapping.

**HO
GENT**

Mapping

= het omzetten van een ER-model (conceptueel model) naar een relationeel model (logisch model)



Mapping

Stappenplan

1. Elk entiteitstype wordt een relatie (=tabel) (opgelet bij specialisatie!).
2. Enkelvoudige attribuuttypes overnemen.
3. Samengestelde attribuuttypes opsplitsen in enkelvoudige attribuuttypes.
4. Meerwaardige attributen in nieuwe relatie plaatsen.
5. Primaire sleutel bepalen (opgelet bij zwakke entiteiten!).
6. Voor elke relatie (verband) tussen entiteitstypen de vreemde sleutel(s) bepalen.
7. Integriteitregels bepalen van elke vreemde sleutel.

Mapping

Regels voor het bepalen van de vreemde sleutel

Verbanden tussen relaties (tabellen) worden gelegd aan de hand van vreemde sleutels:

- Binair verband:
 - 1 op N verband: vreemde sleutel in relatie aan N-zijde
 - 1 op 1 verband: vreemde sleutel in éne of andere relatie
 - Veel op veel: aparte relatie (tabel) met beide vreemde sleutels
- Unair verband:
 - 1 op veel: vreemde sleutel in zelfde relatie (naam van rol 1-zijde!)
 - 1 op 1: vreemde sleutel in zelfde relatie
 - Veel op veel: nieuwe relatie met 2 vreemde sleutels (rolnamen gebruiken!)

Mapping

Integriteitsregels vreemde sleutel:

- naar welke **primaire sleutel** deze vreemde sleutel verwijst
- of de vreemde sleutel **verplicht** of **optioneel** is:
dit bepaal je aan de hand van de
minimumcardinaliteit
- of de vreemde sleutel **uniek** is (bij 1 op 1)

Mapping 1 op N binaire relatie



Vreemde sleutel in tabel aan N-zijde
→ verwijst naar 1 stafflid

Staff (staffNo, fName, lName, position, sex, DOB)

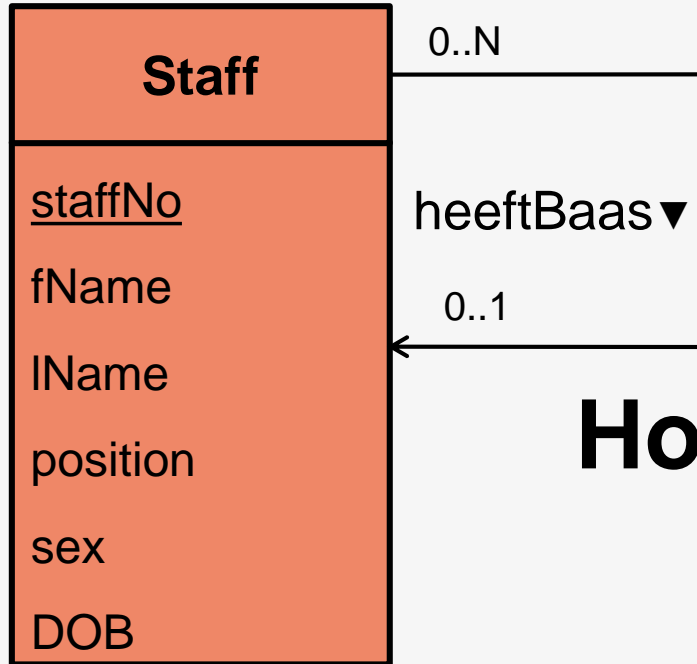
Min.card. = 1 →
verplicht

Client (clientNo, fName, lName, telNo, eMail, staffNo)

IR: vreemde sleutel *staffNo* verwijst naar *staffNo* uit *Staff*, is *verplicht*

**HO
GENT**

Mapping 1 op N recursief (unaire relatie)



▲heeftOndergeschikte

Hoe?

Rol 1-zijde opnemen als vreemde sleutel in zelfde tabel

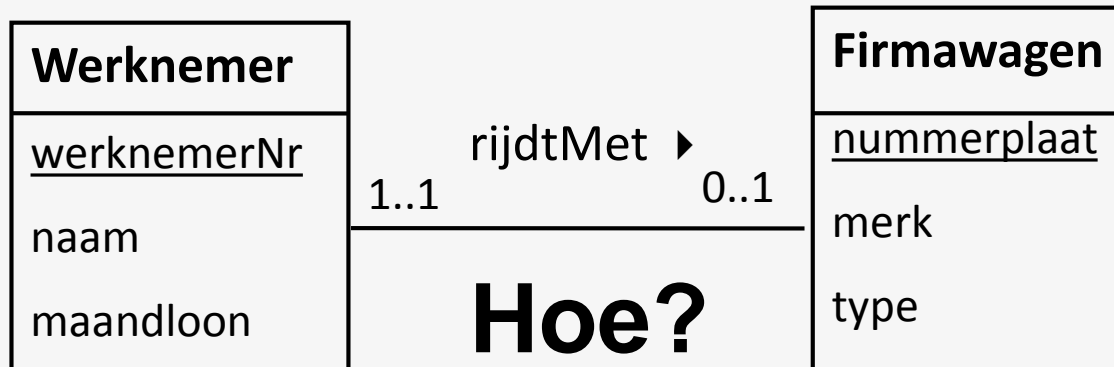
- N-zijde: ondergeschikte: zijn er veel
→ indien je dit zou opnemen als vreemde sleutel dan zou dit veld meerwaardig zijn → mag niet!
- 1-zijde: baas → een stafid heeft hoogstens 1 baas → dit opnemen als vreemde sleutel

Staff (staffNo, fName, lName, position, sex, DOB, baas)
IR: vreemde sleutel *baas* verwijst naar *staffNo*, is *optioneel*

Min.card. = 0
→ optioneel

HO
GENT

Mapping 1 op 1 relatie: verplichte deelname aan één zijde



Hoe?

Vreemde sleutel in tabel met optionele deelname:
vanuit firmawagen verwijzen naar werknemer
Omgekeerd is geen goed idee: indien men vanuit
werknemer zou verwijzen naar firmawagen → veel lege
velden aangezien niet elke werknemer een firmawagen
heeft

Werknemer (werknemerNr, voornaam, familienaam, maandloon)

Firmawagen (nummerplaat, merk, type, werknemer)

IR: Vreemde sleutel *werknemer* verwijst naar
werknemerNr uit *Werknemer*, is verplicht, *uniek*

1.1 relatie → uniek

Mapping 1 op 1 relatie: optionele deelname aan beide zijden



Ofwel vanuit firmawagen verwijzen naar werknemer.
Ofwel vanuit werknemer verwijzen naar firmawagen.

Werknemer (werknemerNr, voornaam, familienaam, maandloon)

Firmawagen (nummerplaat, merk, type, werknemer)

IR: Vreemde sleutel werknemer verwijst naar werknemerNr uit Werknemer, niet verplicht, uniek

OF

Werknemer (werknemerNr, voornaam, familienaam, maandloon, auto)

IR: vreemde sleutel auto, verwijst naar nummerplaat uit Firmawagen, optioneel, uniek

Firmawagen (nummerplaat, merk, type)

Mapping 1 op 1 relatie: optionele deelname aan beide zijden

Beste oplossing?

Werknemer (werknemerNr, voornaam, familienaam, maandloon)

Firmawagen (nummerplaat, merk, type, werknemer)

IR: Vreemde sleutel werknemer verwijst naar werknemerNr uit Werknemer, niet verplicht, uniek

Bovenstaande oplossing is de beste indien men aanneemt dat de meeste firmawagens aan een werknemer toebehoren en minder werknemers een firmawagen hebben.

Mapping veel op veel relatie

Hoe?



Nieuwe relatie met beide sleutels als vreemde sleutel (vormen samen primaire sleutel in nieuwe relatie) + relatie-attributen.

Gerecht (gerechtNr, naam)

Ingrediënt (IngrediëntNr, naam, prijs)

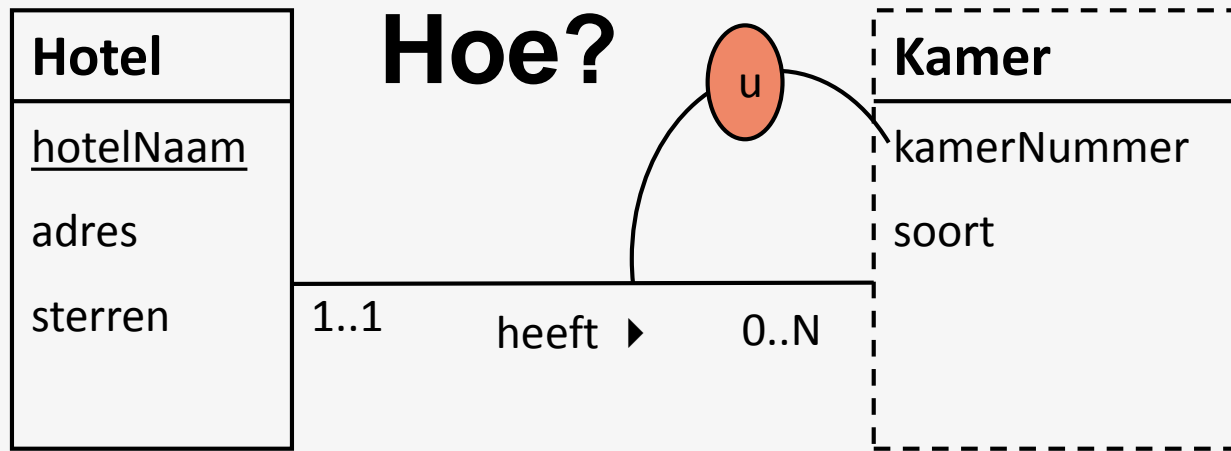
Ingrediënten/Gerecht (gerechtNr, IngrediëntNr, hoeveelheid)

- *gerechtNr*: vreemde sleutel, verwijst naar *gerechtNr* uit *Gerecht*, *verplicht*
- *IngrediëntNr*: vreemde sleutel, verwijst naar *IngrediëntNr* uit *Ingrediënt*, *verplicht*

Verplicht: want maakt deel uit van primaire sleutel!

**HO
GENT**

Mapping zwak entiteitstype



Kijken naar identificatie in ERD: hier wordt een kamer geïdentificeerd aan de hand van:

- zijn relatie met Hotel → sleutel van Hotel = hotelnaam
- kamernummer

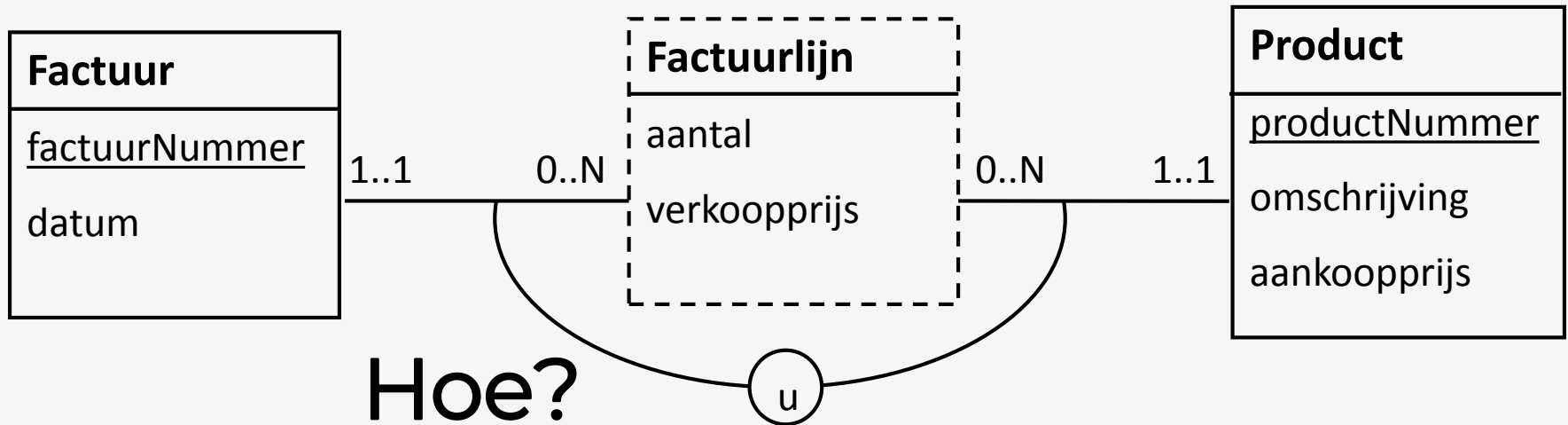
Niet vergeten om samengesteld attribuut adres op te splitsen!!

Hotel (hotelnaam, straat, huisnummer, postcode, gemeente, land, sterren)

Kamer (hotelnaam, kamernummer, soort)

vreemde sleutel *hotelnaam*, verwijst naar *hotelnaam* uit *Hotel*, verplicht

Mapping zwak entiteitstype



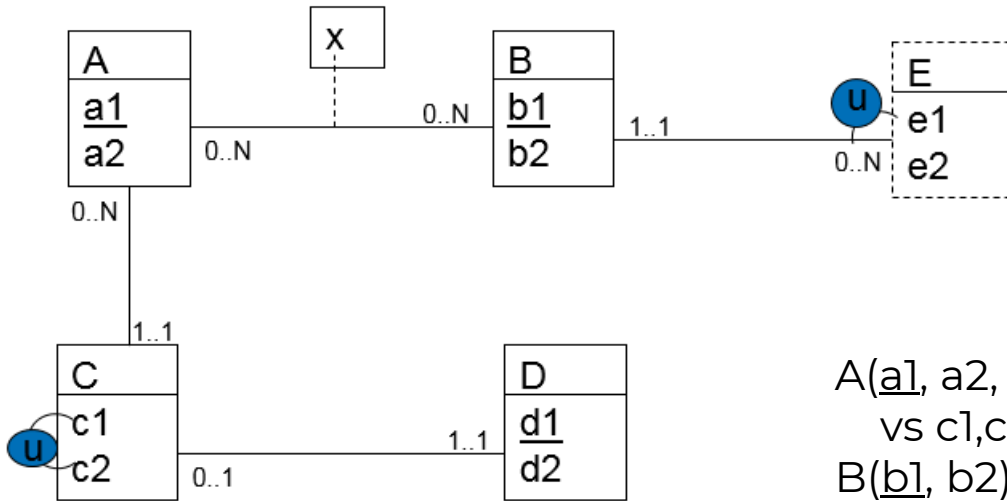
Factuurlijn (factuurNr, productNr, aantal, verkoopprijs)

- vreemde sleutel *factuurNr*, verwijst naar *factuurNummer* uit *Factuur*, *verplicht*
- vreemde sleutel *productNr*, verwijst naar *productNummer* uit *Product*, *verplicht*

Product (productNummer, omschrijving, aankoopprijs)

Factuur (factuurNummer, datum)

Mapping: oefening



A(a1, a2, c1, c2)

vs c1,c2 verwijst naar C, verplicht

B(b1, b2)

AB (a1, b1, x)

vs a1 verwijst naar A, verplicht

vs b1 verwijst naar B, verplicht

E(b1, e1, e2)

vs b1 verwijst naar B, verplicht

C (c1, c2, d1)

vs d1 verwijst naar D, verplicht, uniek

D (d1, d2)

Mapping meerwaardige attributen



Hoe?

Het meerwaardig attribuut schrappen en er een nieuw entiteitstype van maken. Veel-op-veel tussen beide entiteitstypes. Betekenisloze sleutel toevoegen in nieuw entiteitstype. Attribuut naam of omschrijving toevoegen.

Mappingregels van veel-op-veel toepassen.

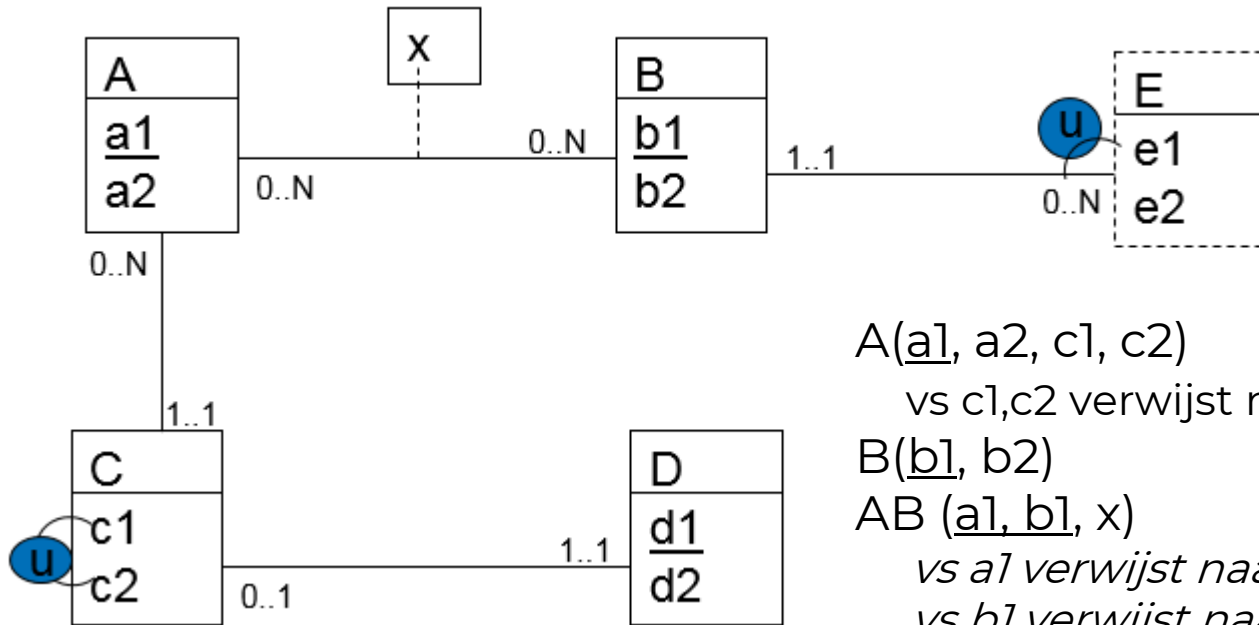
Boeken/Auteur (auteur, boek)

- *auteur*: vreemde sleutel, verwijst naar *auteurID* uit *Auteur*, *verplicht*
- *boek*: vreemde sleutel, verwijst naar *isbn* uit *Boek*, *verplicht*

Auteur (auteurID, voornaam, familienaam)

Boek(isbn, titel, aantalBlz)

Mapping: oefening



d2 is meerwaardig !

$A(\underline{a1}, a2, c1, c2)$

vs c1,c2 verwijst naar C, verplicht

$B(\underline{b1}, b2)$

$AB(\underline{a1}, \underline{b1}, x)$

vs a1 verwijst naar A, verplicht

vs b1 verwijst naar B, verplicht

$E(\underline{b1}, \underline{e1}, e2)$

vs b1 verwijst naar B, verplicht

$C(\underline{c1}, \underline{c2}, d1)$

vs d1 verwijst naar D, verplicht, uniek

$D(\underline{d1})$

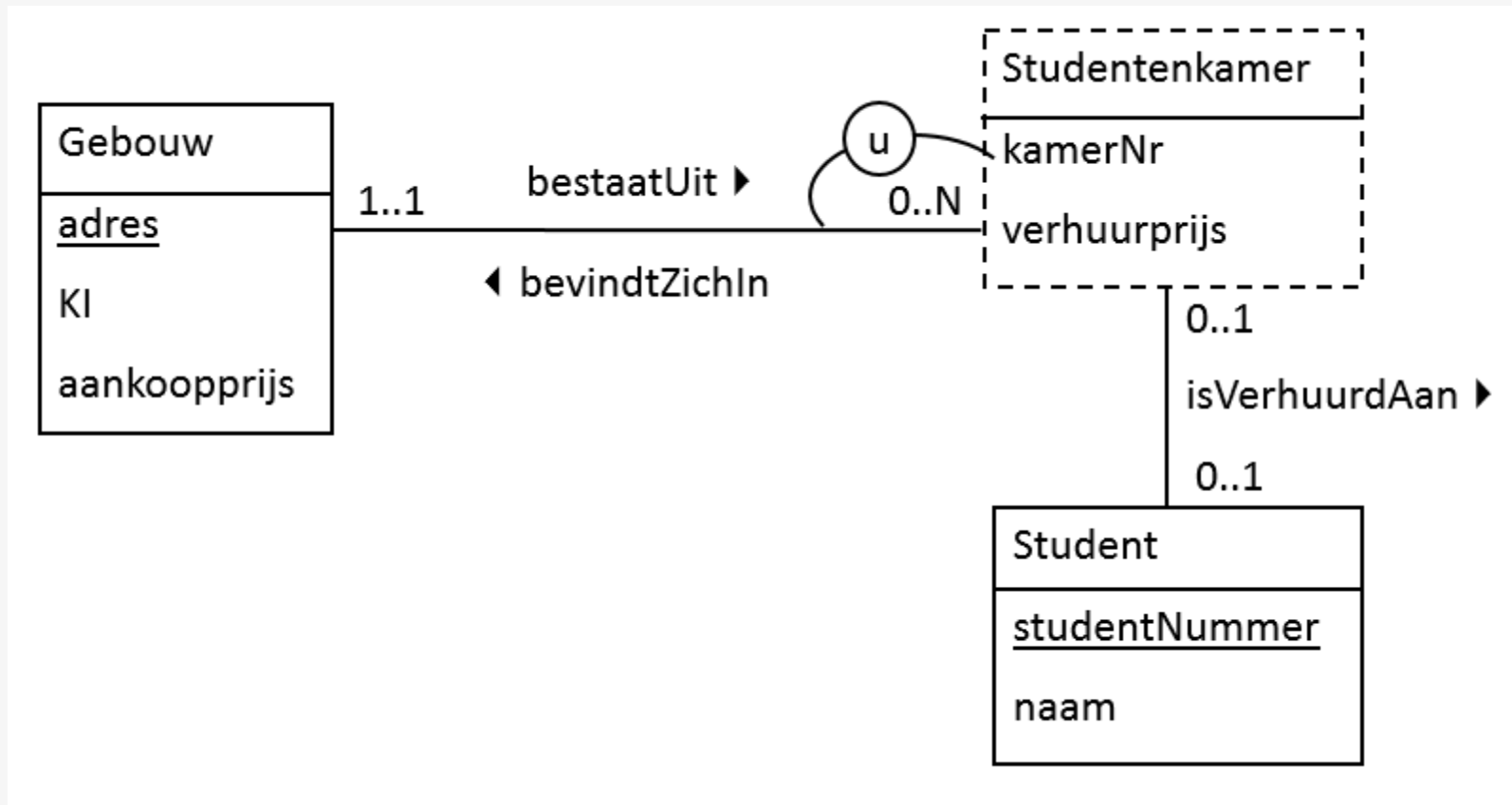
$D2(\underline{ID}, naam)$

$D-D2(\underline{d1}, \underline{ID})$

vs d1 verwijst naar D, verplicht

vs ID verwijst naar D2, verplicht

Mapping: keuze van primaire sleutel



Mapping: keuze van primaire sleutel

Probleem ...

Gebouw (straat, huisnr, postcode, gemeente, land, KI)

Studentenkamer (straat, huisnr, postcode, gemeente, land, kamerNummer, verhuurprijs)

- straat, huisnr, postcode, gemeente, land: vreemde sleutel, verwijst naar de sleutel van *Gebouw*, is *verplicht*

Student (studentNummer, voornaam, familienaam, straat, huisnr, postcode, gemeente, land, kamerNummer)

- straat, huisnr, postcode, gemeente, land, kamerNummer: vreemde sleutel, verwijst naar de sleutel van *Studentenkamer*, *optioneel*

Mapping: keuze van primaire sleutel

Andere mogelijkheid ...

Gebouw (gebouwID, straat, huisnr, postcode, gemeente, land, KI)

Studentenkamer (gebouwID, kamerNummer, verhuurprijs)
gebouwID: vreemde sleutel, verwijst naar *gebouwID* uit *Gebouw*, is verplicht

Student (studentNummer, voornaam, familienaam, gebouwID, kamerNummer)
gebouwID, kamerNummer: vreemde sleutel, verwijst naar *gebouwID, kamerNummer* uit *Studentenkamer*, optioneel

Mapping: keuze van primaire sleutel

Richtlijn!

- Voorkom dat iemand anders kan beslissen over het al dan niet uniek zijn van een gekozen primaire sleutel
 - Is het ISBN van een boek werkelijk uniek? ...
 - Is een barcode werkelijk uniek? ...
 - Zal de barcode van een product steeds dezelfde blijven? ...
- Zorg dat jij als ontwerper steeds controle hebt over de gekozen primaire sleutel.
- Bij twijfel, zelf een sleutel creëren!

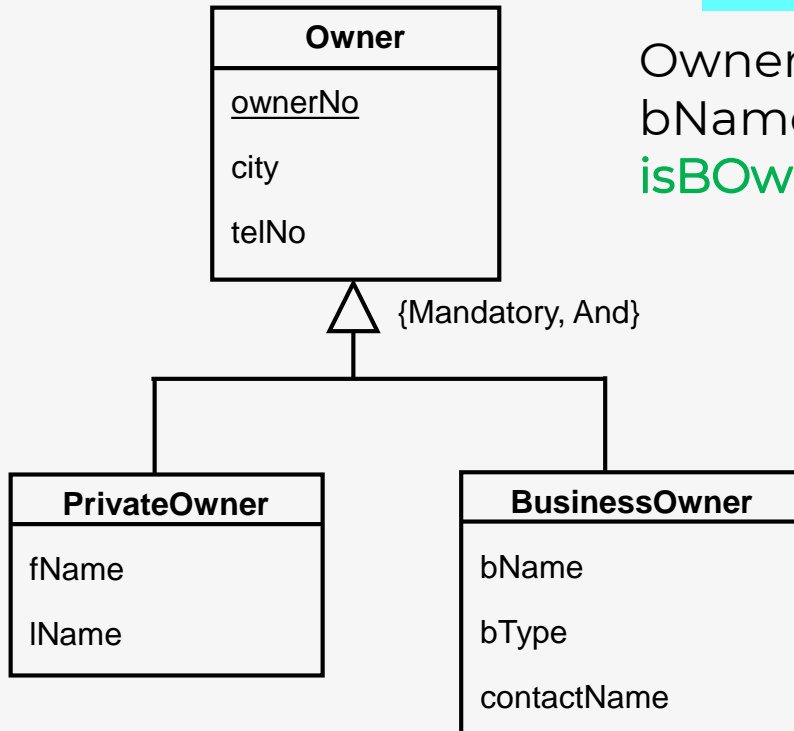
Mapping van specialisatie

Participatie constraint	Disjoint constraint	Relaties
Mandatory	And	1 relatie met hierin booleans om de subtypes te onderscheiden
Optional	And	2 relaties: 1 voor het supertype, 1 voor de subtypes met booleans om de subtypes te onderscheiden
Mandatory	Or	1 aparte relatie per subtype
Optional	Or	voor elk type een relatie

Mapping van specialisatie

Mandatory, And

Mandatory, And: 1 relatie met alle attributen van supertype en subtypes + booleans om subtypes te onderscheiden.



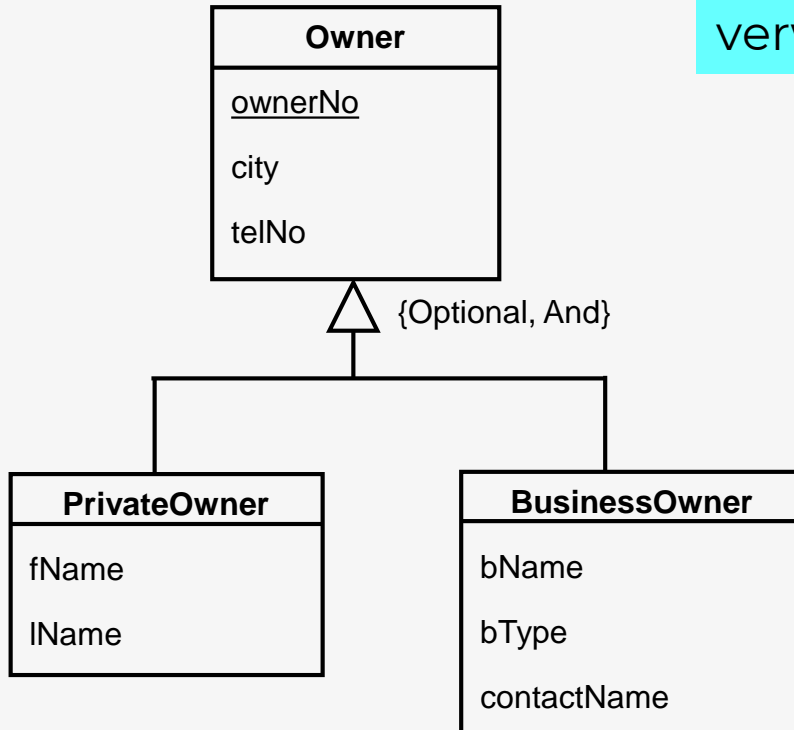
Owner(ownerNo, city, telNo, fName, lName, bName, bType, contactname, **isPOwner**, **isBOwner**)

- Mandatory: → bijkomende regel opnemen: 1 van beide booleans moet 'yes' zijn.
- And: Kan bekomen worden door beide booleans op 'yes' te zetten.
- Nadeel van deze oplossing: indien een owner enkel private owner is of enkel business owner → veel null-waarden.

Mapping van specialisatie

Optional, And

Optional, And: 2 relaties: 1 voor supertype en 1 voor alle subtypes + flags om subtypes te onderscheiden. Vanuit de subtabel verwijzen naar de supertabel.



Owner(ownerNo, city, telNo)

OwnerDetails(ownerNo, fName, lName, bName, bType, contactname, **isPOwner**, **isBOwner**)

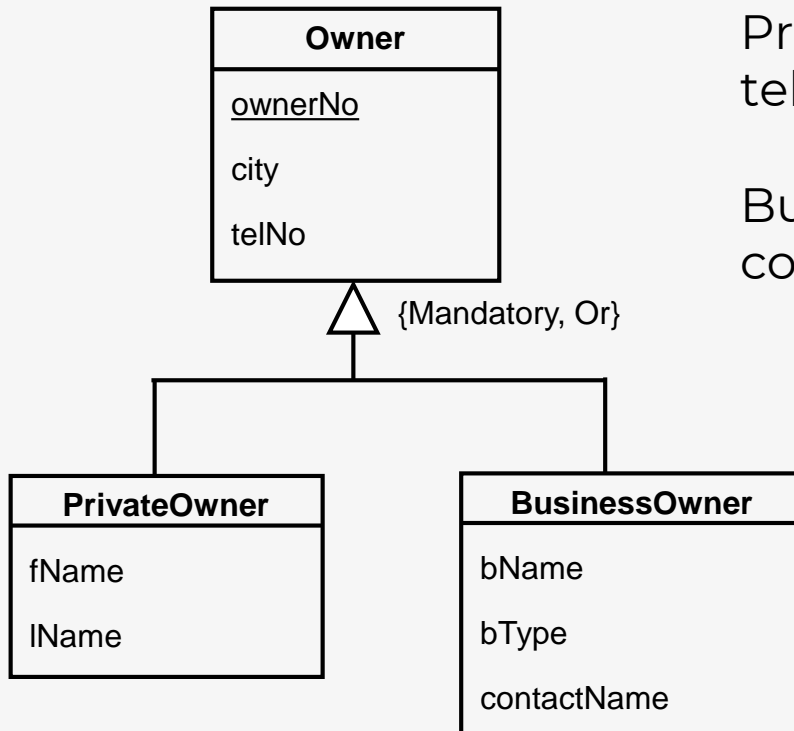
VS ownerNo verwijst naar owner.ownerNo, verplicht.

- And: Kan bekomen worden door beide booleans op 'yes' te zetten.
- Optional: enkel tabel Owner invullen.

Mapping van specialisatie

Mandatory, Or

Mandatory, Or: geen tabel voor supertype, wel een aparte tabel voor elk subtype.



PrivateOwner(pOwnerNo, fName, lName, city, telNo)

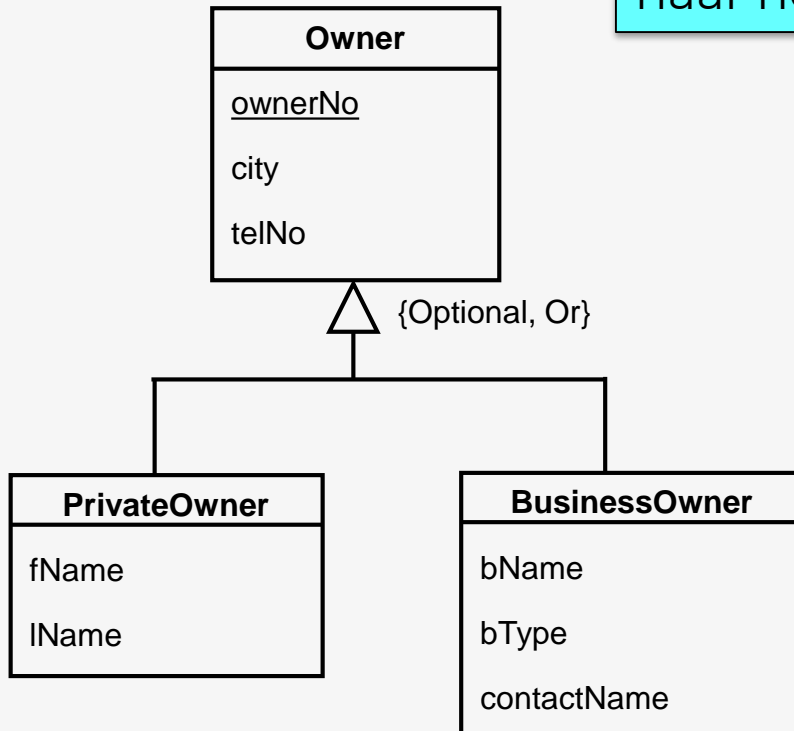
BusinessOwner(bOwnerNo, bName, bType, contactName, city, telNo)

- Mandatory: je kan geen andere owners dan private of business creeëren in dit model.
- Or: voor beide soorten is een aparte tabel.

Mapping van specialisatie

Optional, Or

Optional, Or: tabel voor elk type: zowel voor supertype als elk subtype. De subtypes verwijzen naar het supertype.



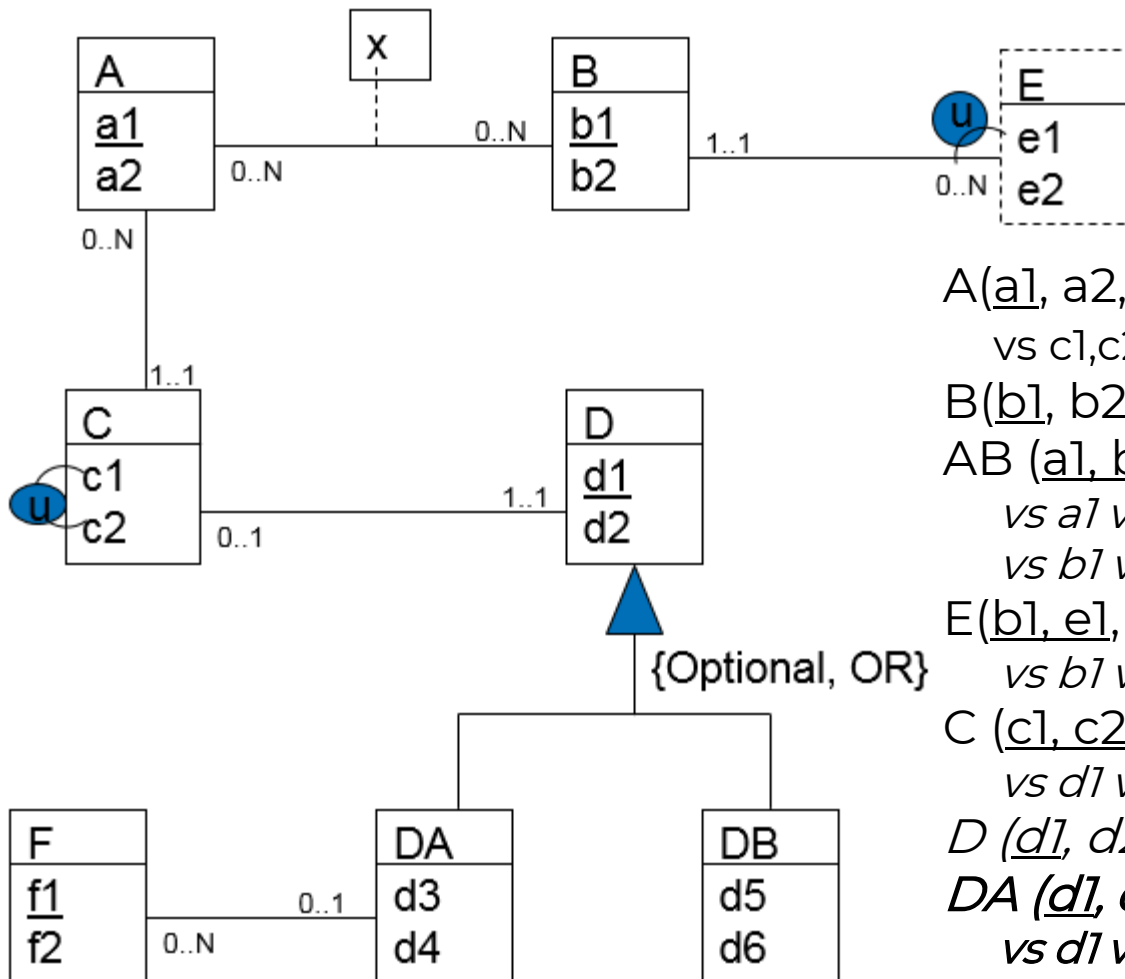
Owner(ownerNo, city, telNo)

PrivateOwner(pOwnerNo, fName, lName)
VS pOwnerNo verwijst naar
Owner.ownerNo, verplicht.

BusinessOwner(bOwnerNo, bName,
bType, contactName)
VS bOwnerNo verwijst naar
Owner.ownerNo, verplicht.

- Optional: Er kunnen andere owners dan private of business gecreëerd worden in de tabel Owner.
- Or: aparte tabellen voor elk subtype.

Mapping: oefening



A(a1, a2, c1, c2)

vs c1,c2 verwijst naar C, verplicht

B(b1, b2)

AB (a1, b1, x)

vs a1 verwijst naar A, verplicht

vs b1 verwijst naar B, verplicht

E(b1, e1, e2)

vs b1 verwijst naar B, verplicht

C (c1, c2, d1)

vs d1 verwijst naar D, verplicht, uniek

D (d1, d2)

DA (d1, d3, d4)

vs d1 verwijst naar D, verplicht

DB (d1, d5, d6)

vs d1 verwijst naar D, verplicht*

F (f1, f2, d1)

vs d1 verwijst naar DA, optioneel

Structuurbependingen

- Relationeel model bevat enkel 0 of 1 op veel verbanden:
 - Veel op veel verbanden: opsplitsen in twee 1 op veel verbanden
 - 1 op 1 verband: beperkingsregels toepassen: 'uniek'
- Minimum cardinaliteit 1:
 - bij max. cardinaliteit 1: beperkingsregel: mag niet null zijn
 - bij max. cardinaliteit veel: kan niet afgedwongen worden

