

Van datamodel naar data-ontwerp

ERD naar tabellen

Relationeel model

- Relationele databank bevat enkel genormaliseerde relaties
- Foreign keys
 - Samenhang tussen gegevens
 - Verwijzen naar primary keys van relaties

Data klaarmaken voor relationeel model

- Top down
 - Toepassen omzettingsregels ERD
- Bottom up
 - Normalisatiestappen Codd

Datamodel (ERD)

- Soorten relaties tussen entiteiten
 - 1:1 (1 op 1)
 - 1:N (1 op meer)
 - N:M (meer op meer)
- Relaties uitgedrukt door lijnen met kraaienpoten

Data-ontwerp (tabelstructuur)

- Heeft enkel 1 (parent) op N (child) relaties
 - N kan 1 zijn
- Relaties uitgedrukt door foreign keys in tabellen

Relaties tussen tabellen geeft informatie

STUDENTEN

Studentnummer	Achternaam	Voornaam	E-mailadres
100	Tack	Pol	p.tack@ehb.be
200	Kumps	Jessie	j.kumps@ehb.be
300	Haantjes	Felix	f.haantjes@ehb.be
400	Pluym	Adèle	a.pluym@ehb.be

VAKKEN

Vaknummer	Vaknaam	Trimester	Studiepunten
10	Programmeren 1	1	4
20	Programmeren 2	2	6
30	Data Essentials	2	6
40	Data Advanced	3	3
50	Android	6	3

CIJFERS

Studentnummer	Vaknummer	Cijfer
100	10	15
100	40	7
200	20	14
300	30	16
400	40	12
400	50	8

Relaties tussen tabellen geeft informatie

STUDENTEN

Studentnummer	Achternaam	Voornaam	E-mailadres
100	Tack	Pol	p.tack@ehb.be
200	Kumps	Jessie	j.kumps@ehb.be
300	Haantjes	Felix	f.haantjes@ehb.be
400	Pluym	Adèle	a.pluym@ehb.be

VAKKEN

Vaknummer	Vaknaam	Trimester	Studiepunten
10	Programmeren 1	1	4
20	Programmeren 2	2	6
30	Data Essentials	2	6
40	Data Advanced	3	3
50	Android	6	3

CIJFERS

Studentnummer	Vaknummer	Cijfer
100	10	15
100	40	7
200	20	14
300	30	16
400	40	12
400	50	8

Omzettingsregels ERD

Stappenplan

1. Entiteiten en attributen omzetten naar respectievelijk tabellen met primary key en kolommen
2. Relaties omzetten naar foreign keys
3. Super- en subtypes omzetten

Stap 1: Algemene regels

- Elke entiteit (enkelvoud) wordt een tabel (meervoud)
- Samengestelde attributen opsplitsen in enkelvoudige kolommen
- Identifier wordt primary key

Stap 1: voorbeeld

- In ERD:



- Na omzetting:

MEDEWERKERS (mednr, mednaam, geboortedatum)

→ mednr is primary key

Stap 2: Omzetten relaties

- Relaties worden uitgedrukt door middel van foreign keys
- Per relatie bepalen waar FK(s) moeten worden toegevoegd

1:1 relatie

- De primary key van de ene tabel wordt als foreign key opgenomen in de andere tabel
- In welke tabel is onbelangrijk
- Oplossing zoeken waarbij NULL-waarden worden vermeden

Voorbeeld omzetting 1:1

- In ERD:

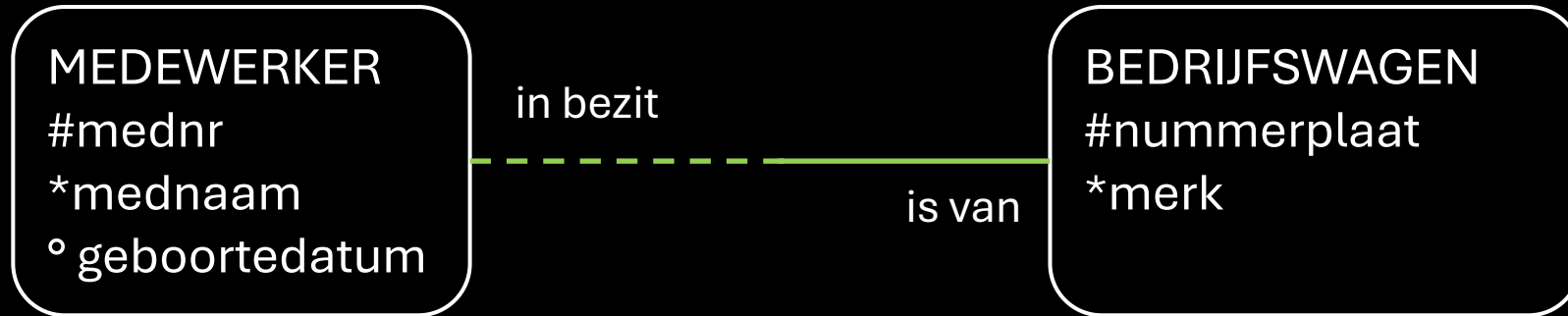


- Na omzetting (stap 1):

MEDEWERKERS (mednr, mednaam, geboortedatum)
BEDRIJFSWAGENS (nummerplaat, merk)

- Relatie wordt foreign key
 - Mednr bij bedrijfswagen of nummerplaat bij medewerkers?

Voorbeeld omzetting 1:1



OPTIE 1:

MEDEWERKERS (mednr, mednaam, geboortedatum, nummerplaat
BEDRIJFSWAGENS (nummerplaat, merk

OPTIE 2:

MEDEWERKERS (mednr, mednaam, geboortedatum
BEDRIJFSWAGENS (nummerplaat, merk, mednr

→ in dit geval de beste oplossing

1:N relatie

- De primary key van de entiteit langs de 1-kant van de relatie wordt opgenomen als foreign key in de entiteit langs de N-kant

Voorbeeld omzetting 1:N

- In ERD:



- Na omzetting:

KLANTEN (klantnr, klantnaam, e-mail)

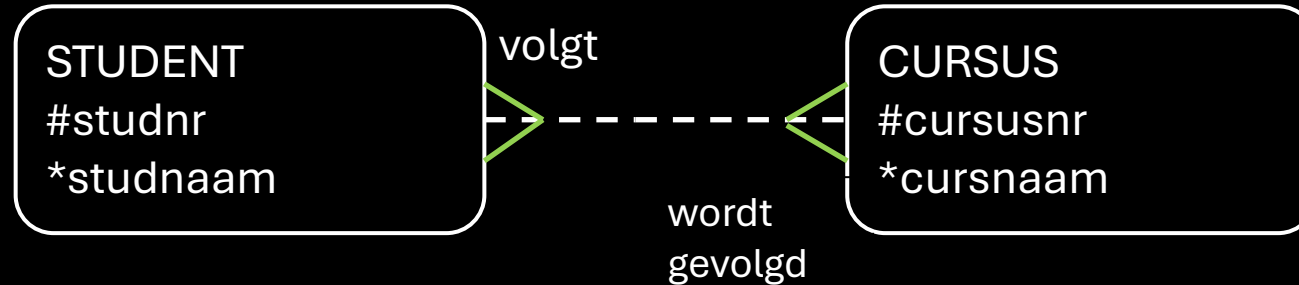
FACTUREN (factuurnr, factuurdatum, bedrag, klantnr)

N:M relatie

- Voor elke meer-op-meer relatie moet een nieuwe tabel (tussentabel) worden aangemaakt
 - Ook wel associative table, junction table, join table, facts table, ...
- De primary key van de tussentabel bestaat uit de identifier van de 2 entiteiten van de N:M relatie

Voorbeeld omzetting N:M

- In ERD:



- Na omzetting:

STUDENTEN (studnr, studnaam)

INSCHRIJVINGEN (studnr, cursusnr)

CURSUSSEN (cursusnr, cursusnaam)

ID-afhankelijke relatie

- Bij een ID-afhankelijke relatie wordt de primary key van de parent tabel toegevoegd aan de child tabel (= de zwakke entiteit)
- De foreign key wordt deel van de samengestelde primary key van de child tabel

Voorbeeld omzetting ID-afhankelijke relatie

- In ERD:



- Na omzetting:

CAMPUSSEN (locatie, naam)
AUDIS (locatie, audinr, aantal_stoelen)

Recursieve relaties

- Volgen dezelfde regels als andere relaties
- Verschil is dat de foreign key in dezelfde tabel wordt toegevoegd i.p.v. een andere tabel

Rekursieve 1:N relatie

- Foreign key wordt toegevoegd aan de tabel
- Deze foreign key duidt de rol aan binnen de relatie

Voorbeeld omzetting recursieve 1:N relatie

- In ERD:



- Na omzetting:

MEDEWEKERS (mednr, mednaam, geboortedatum, leidinggevende)
→ leidinggevende is foreign key naar mednr in dezelfde tabel

Rekursieve N:M relatie

- Voor een recursieve N:M relatie wordt een nieuwe tabel (tussentabel) aangemaakt
- De primary key van de tussentabel is tweemaal de identifier van de entiteit naar waar wordt verwezen

Voorbeeld omzetting recursieve N:M relatie

- In ERD:



- Na omzetting:

MEDEWERKERS (mednr, mednaam)

VERVANGINGEN (mednr, vervanger)

→ opgelet: elke kolom moet een unieke naam hebben

Stap 3: Super- en subtypes

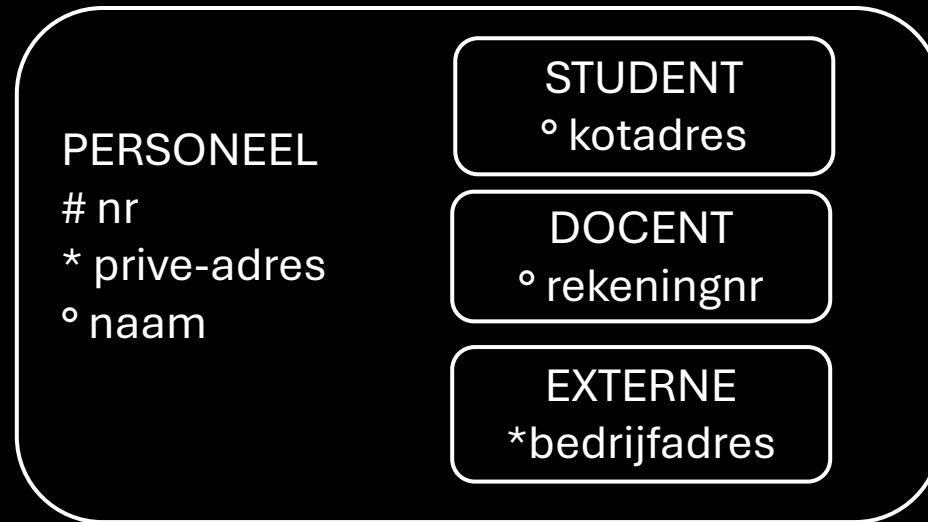
- Meerdere oplossingen mogelijk
 - Optie 1: attributen opsplitsen in aparte tabellen
 - Optie 2: aparte tabellen, met herhaling van attributen
 - Optie 3: alle attributen in 1 tabel
- Optie kiezen op basis van use case, rekening houdend met
 - NULL-waardes
 - Data-duplicatie
 - Performantie

Super- en subtypes: Optie 1

- Aparte tabel voor elk subtype en supertype
- Tabel supertype bevat alle gemeenschappelijke attributen
- Tabel subtype bevat enkel de specifieke attributen voor het subtype
- Primary key van supertype wordt overgenomen in elk subtype
- Supertype kan worden aangevuld met een attribuut dat verwijst naar het subtype waartoe de instantie behoort

Voorbeeld super- en subtypes: Optie 1

- In ERD:



- Na omzetting:

PERSONEELSLEDEN(nr, privé_adres, naam)
STUDENTEN(nr, kotadres)
DOCENTEN(nr, bureel)
EXTERNEN(nr, bedrijfadres)

Super- en subtypes: Optie 1

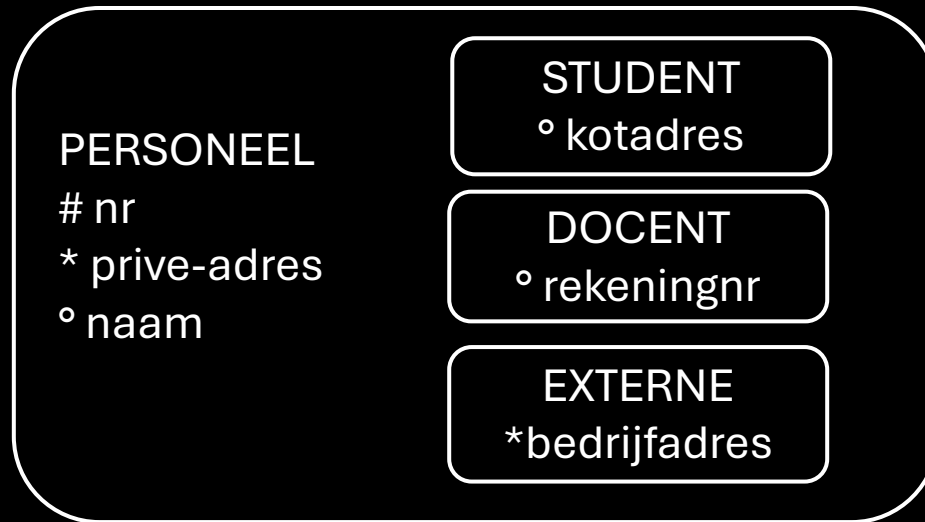
- Voordelen
 - Duidelijke opsplitsing tussen entiteiten
 - Minder NULL-waarden
- Nadeel
 - Eigenschappen van instanties verspreid over meerdere tabellen

Super- en subtypes: Optie 2

- Enkel tabellen voor subtypes
- Gegevens van het supertype herhalen voor alle subtypes

Voorbeeld super- en subtypes: Optie 2

- In ERD:



- Na omzetting:

STUDENTEN(nr, privé_adres, naam, kotadres)
DOCENTEN(nr, privé_adres, naam, rekeningnr)
EXTERNEN(nr, privé_adres, naam, bedrijfadres)

Super- en subtypes: Optie 2

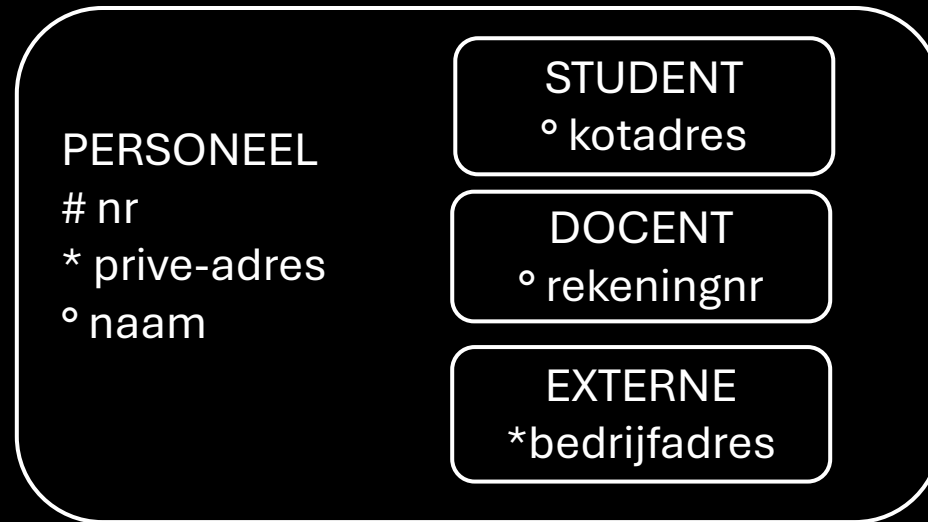
- Voordeel
 - Alle gegevens van een specifieke instantie zitten in één tabel
- Nadeel
 - Gemeenschappelijke gegevens worden herhaald
- Interessant als een instantie niet in meerdere subtypes kan voorkomen

Super- en subtypes: Optie 3

- Supertype en subtypes in één tabel

Voorbeeld super- en subtypes: Optie 3

- In ERD:



- Na omzetting:

PERSONEELSLEDEN(nr, privé-adres, naam, kotadres, rekeningnr, bedrijfadres)
→ Categorie?

Super- en subtypes: Optie 3

- Voordeel
 - Performantiewinst bij hele grote datasets (> 1m records)
- Nadeel
 - Attributen van subtypes kunnen nooit verplicht zijn
 - Veel NULL-waarden voor specifieke kolommen
 - Specifieke categorie van een instantie?
- Interessant als
 - Subtypes geen of niet veel specifieke eigenschappen hebben
 - Performantie een systeemkritieke vereiste is

Bepalen van de primary key

- DBMS gebruikt primary key om tabel te indexeren
- Primary key:
 - Kort
 - Bij voorkeur numeriek
 - Wordt niet gewijzigd
- Vaak surrogaatsleutel (id)
 - Voordeel: snel indexeerbaar
 - Nadeel: geen betekenis voor gebruiker

Een goede primary key

- Is enkelvoudig (geen meerwaardig attribuut)
- Heeft enkel unieke waarde
- Kan geen NULL waarde bevatten
- Mag geen waarde zijn die privacy- of securityregels overtreedt
 - Bv.: Rijksregisternummer
- Verplicht
- Minimum aantal velden om uniekheid te bepalen