Conceptueel model

HO
GENT

Fases in Databank Ontwerp

Het ontwerpen van een databank vertrekt vanuit de bedrijfsprocessen en bestaat uit 4 fases:

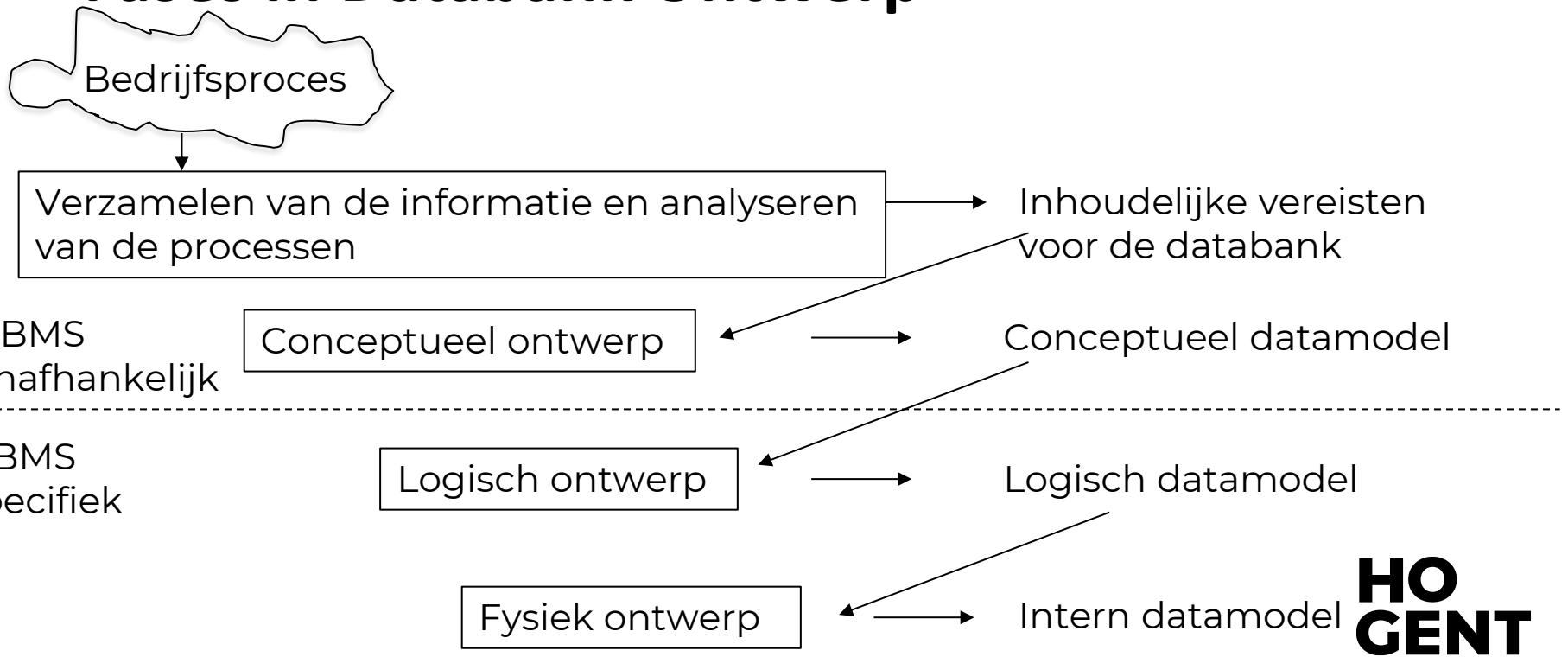
- Fase 1 = Verzamelen en analyseren van de functionele / inhoudelijke vereisten
- Fase 2 = Conceptueel ontwerp
- Fase 3 = Logisch ontwerp
- Fase 4 = Fysiek ontwerp

Mogelijke bedrijfsprocessen:

- het maken van facturen
- werkroosters en prestaties van werknemers
- voorraadbeheer
- puntenadministratie
- …

HO GENT

Fases in Databank Ontwerp



HO GENT

Fase 1 = Verzamelen en analyseren informatie

- Doel: de stappen en de benodigde data van het bedrijfsproces begrijpen.
  Wat nemen we op in de databank?

- Dit kan via
  - interviews met de opdrachtgever
  - analyse van bestaande formulieren en rapporten

- Vragen die moeten beantwoord worden
  - Welke data moet in de databank worden opgenomen?
  - Wat is de betekenis en context van alle data, symbolen, gebruikte coderingen?
  - Hoe zal de data worden verwerkt?
  - Wat is de beoogde functionaliteit?
  - Waarvoor zal de data gebruikt worden?

HO GENT

Fase 1 = Verzamelen en analyseren informatie

Voorbeeld: we willen info opslaan over schilderijen, schilders en musea.

- Relevante data is schilderijnaam, waarde van een schilderij, voor- en familienaam van de schilder, …
- Er moeten schilderijen kunnen toegevoegd worden, het schilderij verandert van museum, …
- Het moet mogelijk zijn om een overzicht te krijgen van de schilderijen. In hoofdzaak moet het mogelijk zijn te rapporteren welke musea welke schilderijen in hun bezit hebben, …

HO GENT

Fase 2 = Conceptueel ontwerp

- Het conceptueel model
  - is de abstractie van de data en de onderlinge verbanden
  - moet voldoende formeel en ondubbelzinnig zijn voor de DBontwerper.
  - moet gebruiksvriendelijk zijn
  - doorgaans een grafische representatie
  - basis voor communicatie en discussie tussen de gebruiker van het bedrijfsproces en de databankontwerper.
  - gebeurt onafhankelijk van enig databankmodel of applicatie. Anders te vroeg gekoppeld aan een bepaald databankmodel of een bepaalde applicatie.

HO GENT

Fase 2 = Conceptueel ontwerp

Voorbeeld: we willen info opslaan over schilderijen, schilders en eigenaars.

- De data zal worden georganiseerd rond de centrale concepten ‘SCHILDERIJ’, ‘SCHILDER' en ‘MUSEUM‘

- Gegevens die je wil opslaan
  - SCHILDERIJ: naam, kunststroming waartoe het schilderij behoort, gebruikte verftechniek, ….
  - SCHILDER: naam, nationaliteit, …….
  - MUSEUM: naam, plaats, land……

HO GENT

Fase 3 = Logisch ontwerp

- Type databank is bekend (relationele databank, NoSQL databank, hiërarchische databank, …)

- Het product zelf ligt nog niet vast
  - voor relationele databank: Microsoft SQL Server of MySQL of Oracle of DB2 of …
  - voor NoSQL document databank: MongoDB of CouchDB of …
  - voor hiërarschische databank: IMS of …

- !! Bij het opstellen van het conceptueel model en bij de overgang van het conceptueel model naar het logisch model is er mogelijk verlies van specificaties.
  → In een apart document bijhouden om te gebruiken bij de applicatie-ontwikkeling.

HO GENT

Fase 3 = Logisch ontwerp

Voorbeeld: we willen info opslaan over schilderijen, schilders en musea.

- Wij kiezen voor het relationeel databankmodel.
- De centrale concepten ‘SCHILDERIJ’, ‘SCHILDER’ en ‘MUSEUM’ worden omgezet en zullen later evolueren naar tabellen.
- Voorbeeld van informatie die niet in een conceptueel model opgenomen is:
  - Het geboortejaar van een SCHILDER moet kleiner zijn dan het jaar van overlijden;
  - Van elke SCHILDER moet het geboortejaar bekend zijn.

HO GENT

Fase 4 = Fysiek ontwerp

- Is de feitelijke implementatie van het logisch model.
- Je kiest eerst een product, ook DBMS genoemd (MySQL, Microsoft SQL Server, Oracle, …).
- Je implementeert het logisch model en zet dit om in datadefinitiecode (= DDL), die kan worden verwerkt door het DBMS.
- Technische details worden toegevoegd (datatypes van de attribuuttypes, …)
- Indien mogelijk worden ook de functionele beschrijvingen ‘vertaald’ naar databaseconcepten. Zo kunnen de bedrijfsregels rond correct geboortejaar en jaar van overlijden omgezet worden naar een integriteitsrestrictie.
- DBA kan ook aanbevelingen doen in verband met de performantie.
  → zie Relational Databases and Datawarehousing (2TI)

HO GENT

Fases in Databank Ontwerp

Verzamelen en analyseren van de vereisten

- domeinanalyse
- functionele analyse
- behoefteanalyse

Conceptueel ontwerp

- conceptueel model (bijvoorbeeld EER – diagram)
- functionele beschrijving

databasemodel-onafhankelijk

Logisch ontwerp

- logisch databankschema (bijvoorbeeld relationele databank)
- gedragsspecificaties

dbms-onafhankelijk

Fysiek ontwerp

- DDL-scripts
- implementatie van gedrag

HO GENT

Bron: Principes van Databases (G. De Tré)

Entity Relationship Diagram

HO
GENT

Inleiding

- Het Entity Relationship Diagram (ERD) werd geïntroduceerd en geformaliseerd door Peter Chen in 1976.

- Het is één van de populairste voorstellingswijzen voor het conceptueel gegevensmodel.

- Een Entity Relationship Diagram heeft de volgende bouwstenen:
  - Entiteittypes
  - Attribuuttypes
  - Relatietypes

HO GENT

Entiteittype

- Een entiteittype
  - bestaat in de reële wereld.
  - kan zowel abstract (een tentoonstelling, firma, cursus, job, …) als fysiek (schilderij, persoon, auto, huis, …) zijn.
  - is ondubbelzinnig gedefinieerd voor een bepaalde groep gebruikers.
  - karakteriseert een collectie van entiteiten
  - heeft een naam en inhoud en is identificeerbaar.
- Een entiteit is een instantie van een entiteittype.
- In het conceptueel model nemen we entiteittypes op (geen individuele entiteiten).

HO GENT

Attribuuttype

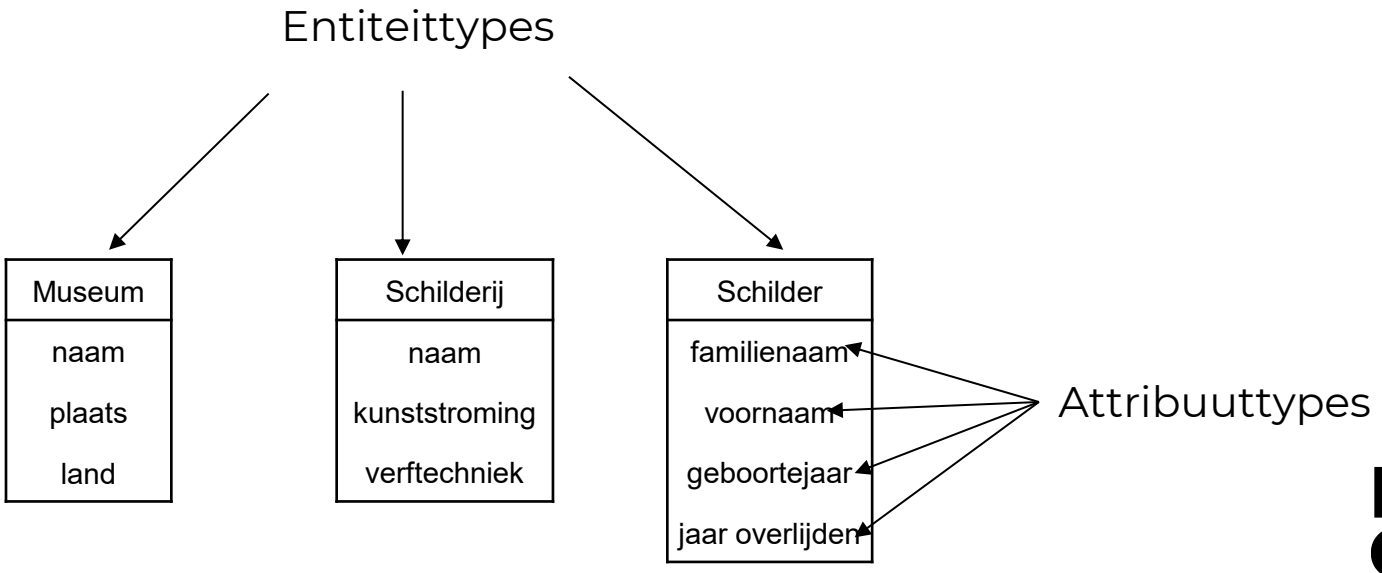
- Een attribuuttype
  - is een karakteristiek van een entiteittype
  - beschrijft het entiteittype

- Elke entiteit heeft een specifieke waarde voor elk attribuuttype.

HO GENT

Entiteittype en attribuuttype

- Visualisatie van de entiteittypes voor MUSEM, SCHILDERIJ en SCHILDER.



HO GENT

Entiteittype

- Voorbeelden van de entiteittypes voor MUSEM, SCHILDERIJ en SCHILDER.
  - Leonardo da Vinci, geboren in 1452 en gestorven in 1519 is een entiteit van het entiteittype SCHILDER.
  - Het Louvre in Parijs in Frankrijk is een entiteit van het entiteittype MUSEUM.
  - De Mona Lisa uit de Hoogrenaissance, met sfumato als verftechniek, is een entiteit van het entiteittype SCHILDERIJ

Museum
naam
plaats
land

Schilderij
naam
kunststroming
verftechniek

Schilder
familienaam
voornaam
geboortejaar
jaar overlijden

- Geef andere voorbeelden van entiteiten voor de entiteittypes MUSEM, SCHILDERIJ en SCHILDER.

Entiteittype en attribuuttype

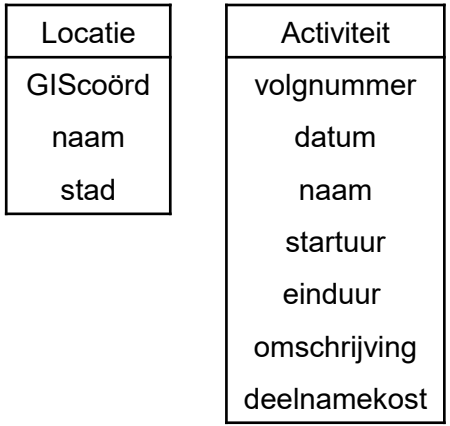
- Voor een onervaren databaseontwerper kan het onduidelijk zijn of een gegeven concept al dan niet als entiteittype moet worden gemodelleerd.

- Een entiteittype is **identificeerbaar** en moet **een inhoud** hebben.

HO GENT

Oefening Activiteit

- Je wil een activiteitendatabank creëren
- Elke ACTIVITEIT krijgt een uniek volgnummer, een datum, een naam, een startuur, vermoedelijk einduur, een korte omschrijving en een deelnamekost.
- Elke ACTIVITEIT start en eindigt op een bepaalde LOCATIE. Een LOCATIE heeft unieke GIScoördinaten, een naam en een stad.
- Welke entiteittypes en attribuuttypes herken je?



HO GENT

Relatietype

- Entiteittypes kunnen onderlinge verbanden hebben:
  - een SCHILDERIJ is geschilderd door een SCHILDER
  - een SCHILDERIJ behoort toe aan een MUSEUM
  - een STUDENT volgt een aantal CURSUSsen, …
- Er kunnen één, twee, drie of meer entiteittypes betrokken zijn in een relatie
  - één entiteittype: een SCHILDER kan een afstammeling zijn van een andere SCHILDER
  - twee enteittypes: een SCHILDERIJ werd geschilderd door één of meerdere SCHILDER(s)
  - drie entiteittypes: een DOKTER schrijft een GENEESMIDDEL voor aan een PATIËNT

HO GENT

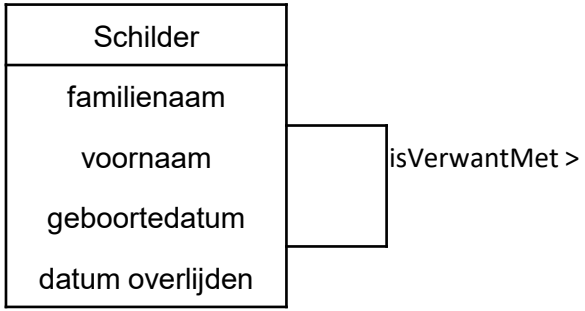
Relatietype

- Een **relatietype** is een verzameling van relaties tussen instanties van één, twee of meer al dan niet verschillende entiteittypes. Men spreekt respectievelijk van een unair (n = 1), binair (n = 2), ternair (n = 3) of n-air (n > 3) relatietype.
  Elk relatietype wordt gekenmerkt door een naam.
- In Databases beperken we ons tot unaire en binaire relatietypes.

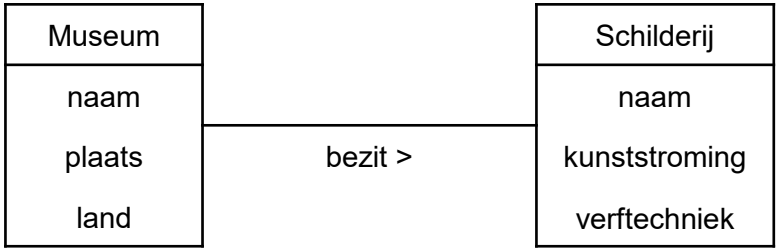
HO GENT

Relatietype

- De **graad** van een relatietype = het aantal verschillende entiteittypes die deelnemen aan het relatietype
  - Unaire relatie → 1 entiteittype
  - Binaire relatie → 2 entiteittypes
- Voorbeeld van een unaire of recursieve relatie:



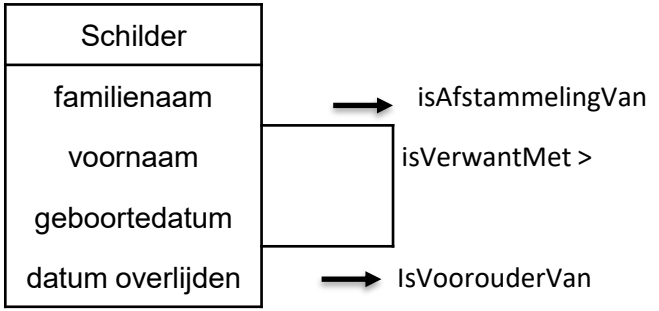
- Voorbeeld van een binaire relatie:

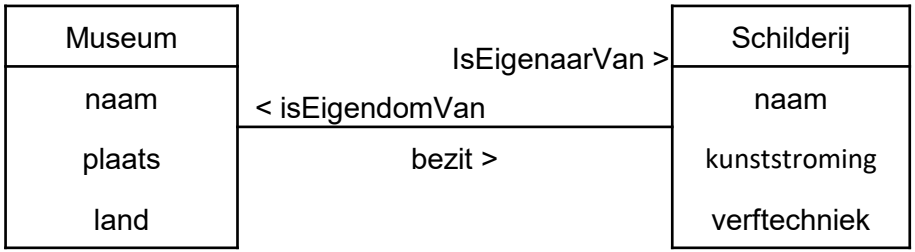


HO GENT

Relatietype

- De **rollen** van een relatietype beschrijven geeft de rol aan die een deelnemende entiteit van het entiteitstype speelt in de relatie.
- In een unaire of recursieve relatie:
  - één entiteitstype neemt meer dan één keer deel aan de relatie
  - => de rolnaam is essentieel voor het onderscheiden van de betekenis die elke deelnemende entiteit speelt
- In een binaire relatie:
  - de naam van elk deelnemend entiteitstype kan worden gebruikt als rolnaam

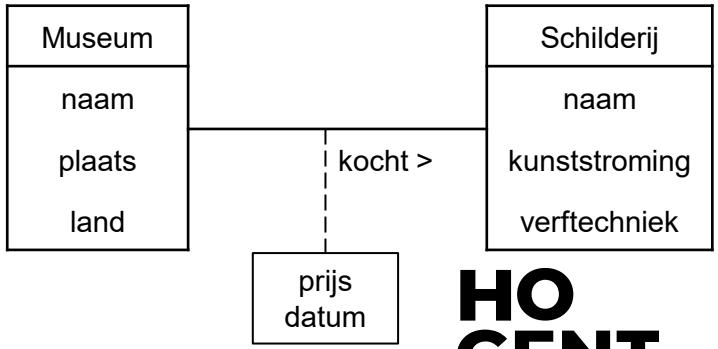




HO GENT

Relatie-attribuut

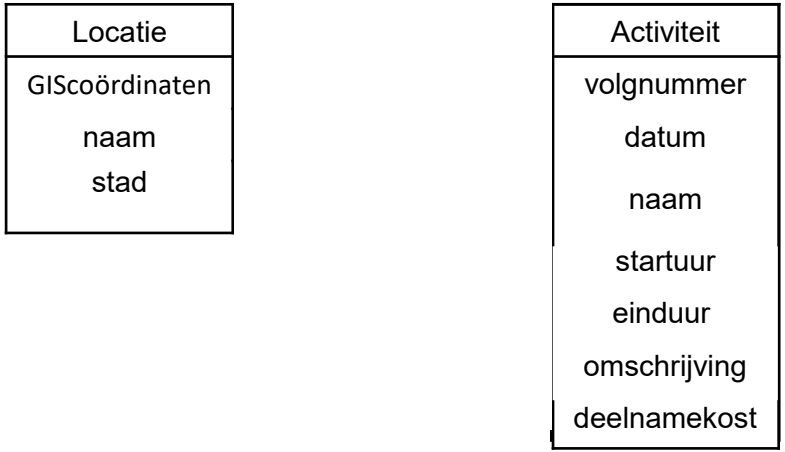
- Ook relatietypes kunnen eigenschappen hebben: wanneer een kenmerk een eigenschap is van het relatietype en niet van één van de betrokken entiteittypes. We spreken van een **relatie-attribuut**.
- **Voorbeeld**: het bedrag dat uitdrukt voor welke prijs een MUSEUM het SCHILDERIJ heeft aangekocht. Deze prijs hoort bij de 'kocht'-relatie tussen MUSEUM en SCHILDERIJ en is geen eigenschap van het MUSEUM (die verschillende SCHILDERIJen kan gekocht hebben) of van het SCHILDERIJ (dat eventueel meer dan eens verkocht werd).
- Idem voor de datum waarop het MUSEUM het SCHILDERIJ heeft verworven.



HO GENT

Oefening Activiteiten

- Je wil een activiteitendatabank creëren
- Elke ACTIVITEIT krijgt een uniek volgnummer, een datum, een naam, een startuur, vermoedelijk einduur, een korte omschrijving en een kostprijs om deel te nemen. De combinatie van datum en naam is ook uniek.
- Elke ACTIVITEIT start en eindigt op een bepaalde LOCATIE. Een LOCATIE heeft unieke GIScoördinaten (breedtegraad en lengtegraad), een naam en een stad.
- Geef de relatietype(s)



Attribuuttype

Het ER-model kent een aantal mogelijkheden om attribuuttypes verder te karakteriseren:

- Enkelvoudige versus samengestelde attribuuttypes
- Enkelwaardige versus meerwaardige attribuuttypes
- Afgeleide attribuuttypes
- Kandidaatsleutelattribuuttypes

HO GENT

Attribuuttype

Enkelvoudige versus samengestelde attribuuttypes

- Samengesteld attribuuttype: het attribuuttype kan nog opgesplitst worden. Bijvoorbeeld het attribuuttype ‘adres’ kan samengesteld zijn uit een ‘straat’, een ‘nummer’, een ‘postcode’ en een 'woonplaats'.
  Wij werken in het conceptueel model steeds op het niveau van enkelvoudige attribuuttypes.
- Afhankelijk van de context zullen attribuuttypes soms verder opgesplitst worden of niet. Bijvoorbeeld als het niet belangrijk is dat 'straat' of 'woonplaats' afzonderlijk moet gekend zijn, dan wordt 'adres' een enkelvoudig attribuuttype. In dat geval kan niet met de afzonderlijke delen (straat, stad, …) gewerkt worden.

HO GENT

Attribuuttype

Enkelwaardige versus meerwaardige attribuuttypes

- Enkelwaardig attribuuttype: het attribuuttype heeft één waarde. Bijvoorbeeld het attribuuttype ‘museum’ van ‘SCHILDERIJ’ en de attribuuttypes ‘geboortejaar’ en ‘jaar overlijden’ van ‘SCHILDER’.
- Meerwaardig attribuuttype: het attribuuttype kan (meerdere) waarden bevatten. Bijvoorbeeld een ‘SCHILDER' kan meerdere talen spreken of meerdere hobbys hebben. In dat geval zijn ‘talen’ en ‘hobby's’ meerwaardige attribuuttypes.
- In een ERD mogen beide voorkomen (zie later).
  Binnen Databases vermijden we meerwaardige attributen in het ERD.

HO GENT

Attribuuttype

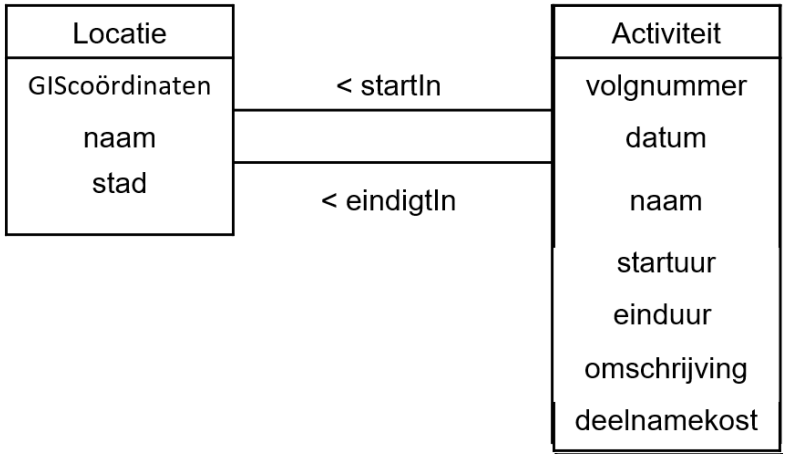
Afgeleide attribuuttypes

- De waarde van een afgeleid attribuuttype kan berekend worden op basis van waarden van andere attribuuttypes.
- Een typisch voorbeeld is ‘leeftijd’: de waarde kan berekend worden als het verschil van de huidige datum en de waarde van het attribuuttype ‘geboortedatum’
- Afgeleide attribuuttypes worden **niet opgeslagen** in de databank, omdat dit een potentiële bron van inconsistentie kan zijn.
  Dit wordt vervangen door de **basisinformatie** waaruit de waarde van het attribuuttype kan berekend worden.

HO GENT

Oefening Activiteiten

- Je wil een activiteitendatabank creëren
- Elke ACTIVITEIT krijgt een uniek volgnummer, een datum, een naam, een startuur, vermoedelijk einduur, een korte omschrijving en een kostprijs om deel te nemen.
- Elke ACTIVITEIT start en eindigt op een bepaalde LOCATIE. Een LOCATIE heeft unieke GIScoördinaten (breedtegraad en lengtegraad), een naam en een stad.
- Herken je
  - enkelvoudige attributen
  - samengestelde attributen
  - eenwaardige attributen
  - meerwaardige attributen



Attribuuttype

Kandidaatsleutelattributen

- Één attribuut of meerdere attributen samen die de entiteiten van een entiteittype op een unieke, irreducibele manier identificeren, vormen een **kandidaatsleutel** van het entiteittype.
  Irreducibiliteit wil zeggen dat er geen uniciteit mag gelden als men één of meerdere attributen weglaat.
- De attributen die deel uitmaken van een kandidaatsleutel noemt men de **kandidaatsleutelattributen**.
- Er kunnen meerdere **kandidaatsleutels** zijn. Later wordt uit de kandidaatsleutels één sleutel gekozen als primaire sleutel.

HO GENT

Attribuuttype

Kandidaatsleutelattributen

Voorbeeld: Stel er bestaan geen twee musea met eenzelfde combinatie van naam en plaats => naam en plaats zijn **kandidaatsleutelattributen** en vormen (samen) de kandidaatsleutel.

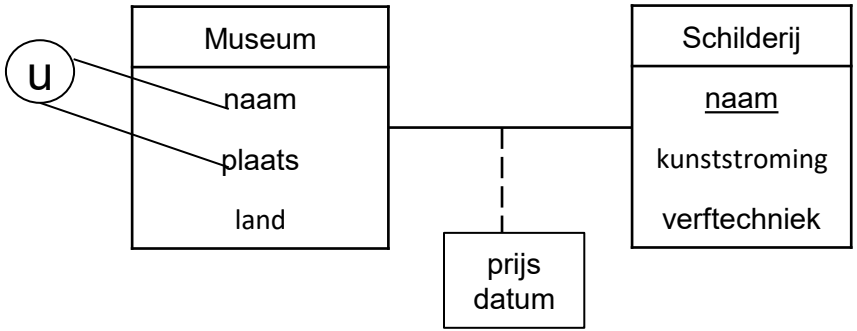
De combinatie van naam, plaats en land is geen kandidaatsleutel want de combinatie van naam en plaats op zich is al uniek => naam, plaats én land zijn niet irreducibel, maw land is overbodig om uniciteit af te dwingen.

HO GENT

Attributen

Kandidaatsleutelattributen

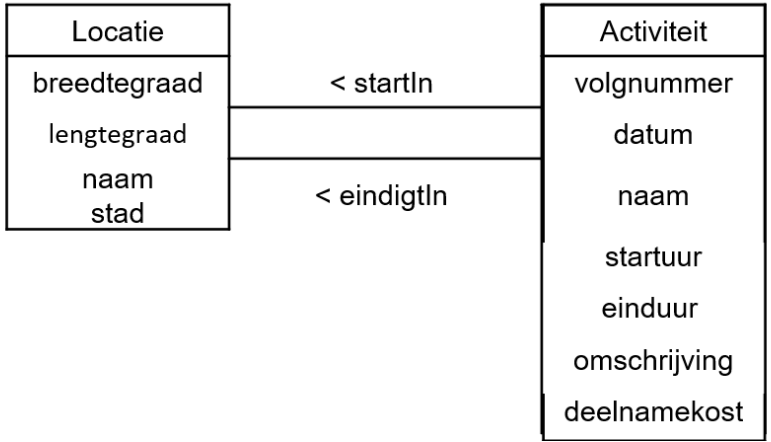
- Alle enkelvoudige kandidaatsleutels (bestaat uit 1 attribuuttype) worden onderlijnd.
- Indien een kandidaatsleutel uit meerdere attribuuttypes bestaat (samengestelde kandidaatsleutel), duiden we dit dan aan met de 'u'-constraint.



HO GENT

Oefening Activiteiten

- Je wil een activiteitendatabank creëren
- Elke ACTIVITEIT krijgt een uniek volgnummer, een datum, een naam, een startuur, vermoedelijk einduur, een korte omschrijving en een kostprijs om deel te nemen. De combinatie van datum en naam is ook uniek.
- Elke ACTIVITEIT start en eindigt op een bepaalde LOCATIE. Een LOCATIE heeft unieke GIScoördinaten (breedtegraad en lengtegraad), een naam en een stad.
- Geef de kandidaatsleutel(s)



Cardinaliteiten

- Elk relatietype kan worden gekarakteriseerd in termen van cardinaliteit = het aantal entiteiten dat kan deelnemen aan de relatie.
- Elk relatietype heeft een minimum- en een maximumcardinaliteit.
- Cardinaliteit betekent aantal en wordt uitgedrukt als een getal.
- De cardinaliteiten moeten afgetoetst worden met de opdrachtgever! Deze zijn vaak afhankelijk van de bedrijfsregels.
  WIJ MODELLEREN WAT WE WETEN. We veronderstellen niet!
- !! Verkeerd gekozen cardinaliteiten kunnen ook voor minder kwalitatieve applicaties zorgen.

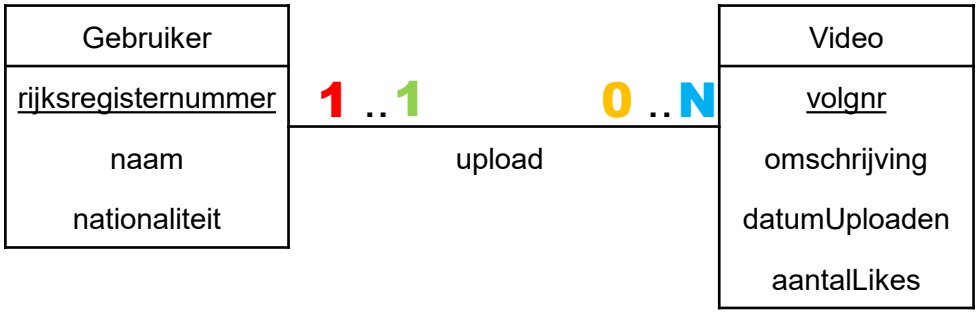
Cardinaliteiten

- Maximumcardinaliteit = het maximum aantal entiteiten van het entiteittype dat op een gegeven tijdstip **kan** deelnemen aan een relatie van het relatietype. Mogelijke waarden zijn 1 of N.
  - 1: één entiteit kan in relatie staan met maximum 1 (andere) entiteit via dit relatietype
  - N: één entiteit kan in relatie staan met N (andere) entiteiten via dit relatietype. N is een willekeurig geheel getal groter dan 1.
- Minimumcardinaliteit = het minimum aantal entiteiten van het entiteittype dat op elk tijdstip **moet** voorkomen in een relatie van het relatietype. Mogelijke waarden zijn 0 of 1.
  - 0: sommige entiteiten nemen niet deel aan de relatie. De relatie is optioneel voor dat entiteittype.
  - 1: een entiteit moet altijd in relatie staan met minimum één andere entiteit

HO GENT

Cardinaliteiten

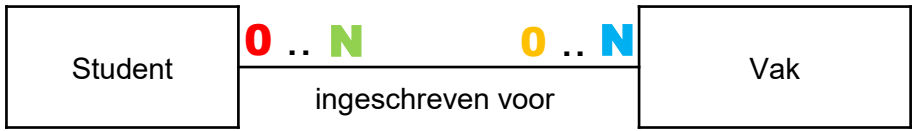
- Maximumcardinaliteit
  - Kan één GEBRUIKER meer dan één VIDEO geüpload hebben? Ja => **N**
  - Kan één VIDEO geüpload zijn door meer dan één GEBRUIKER? Neen => **1**
- Minimumcardinaliteit
  - Moet één GEBRUIKER ten minste één VIDEO geüpload hebben? Neen => **0**
  - Moet één VIDEO geüpload zijn door ten minste één GEBRUIKER? Ja => **1**



HO GENT

Cardinaliteiten

- Maximumcardinaliteit
  - Kan één STUDENT ingeschreven zijn voor meer dan één VAK ? Ja => **N**
  - Kan één VAK gevolgd worden door meer dan één STUDENT? Ja => **N**
- Minimumcardinaliteit
  - Moet één STUDENT ten minste voor één VAK ingeschreven zijn? Neen => **0**
  - Moet één VAK ten minste één STUDENT hebben? Neen => **0**



HO GENT

Cardinaliteiten

- Maximumcardinaliteit
  - Kan één STUDENT toegekend zijn aan meer dan één BACHELORPROEF? Neen => **1**
  - Kan één BACHELORPROEF toegekend zijn aan meer dan één STUDENT? Neen => **1**
- Minimumcardinaliteit
  - Moet één STUDENT toegekend zijn aan ten minste één BACHELORPROEF? Neen => **0**
  - Moet één BACHELORPROEF ten minste toegekend zijn aan één STUDENT? Neen => **0**

HO GENT

Cardinaliteiten

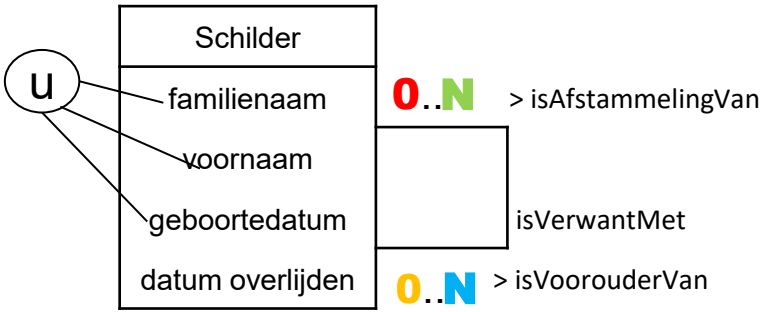
- Voorbeeld van bedrijfsspecifieke cardinaliteiten.

- Maximumcardinaliteit
  - Kan één WERKNEMER meer dan één PROJECT leiden? Ja => **N**
  - Kan één PROJECT geleid worden door meer dan één WERKNEMER? Neen => **1**
- Minimumcardinaliteit
  - Moet één WERKNEMER ten minste één PROJECT leiden? Neen => **0**
  - Moet één PROJECT geleid worden door ten minste één WERKNEMER? Ja => **1**

HO GENT

Cardinaliteiten

- Maximumcardinaliteit
  - Kan één SCHILDER afstammen van meer dan één SCHILDER? Ja => **N**
  - Kan één SCHILDER voorouder zijn van meer dan één SCHILDER? Ja => **N**
- Minimumcardinaliteit
  - Moet één SCHILDER afstammen van ten minste één SCHILDER? Neen => **0**
  - Moet één SCHILDER voorouder zijn van ten minste één SCHILDER? Neen => **0**



HO GENT

Cardinaliteiten

- Relatietypes worden vaak gekarakteriseerd door de maximumcardinaliteit van elk van de rollen.

- In het geval van een unaire of binaire relatie geeft dit aanleiding tot:
  - 1-op-1 relatie → 1:1
  - 1-op-veel-relatie → 1:N of N:1
  - veel-op-veel-relatie → M:N of N:M

- Een verplichte minimumcardinaliteit wijst op bestaansafhankelijkheid (zie later).

HO GENT

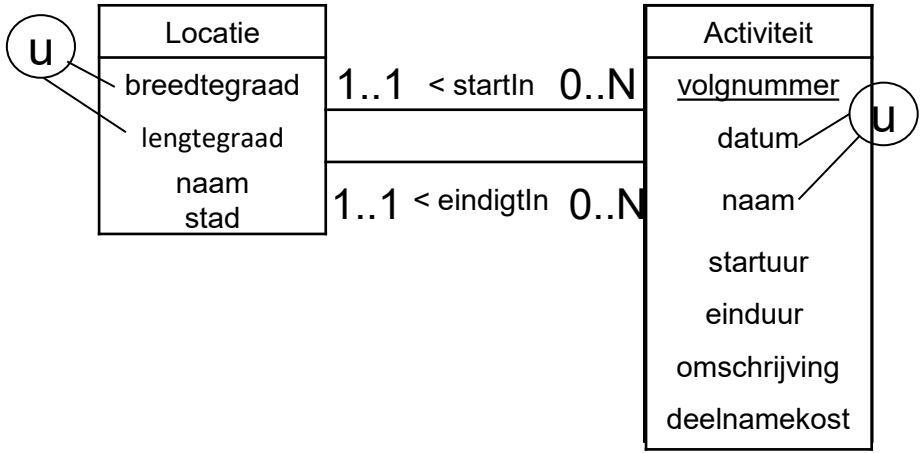
Cardinaliteiten

- Voorbeelden
  - Een STUDENT kan voor minimaal 0 en voor maximaal M VAKken ingeschreven zijn.
    Voor een VAK kunnen minimaal 0 en maximaal N STUDENTen ingeschreven zijn.
    → M:N (een veel-op-veel-relatie)
  - Een BACHELORPROEF kan aan minimaal 0 en aan maximaal 1 STUDENT toegekend zijn.
    Een STUDENT kan minimaal 0 en maximaal 1 BACHELORPROEF uitwerken.
    → 1:1 (een één-op-één-relatie)
  - Een PROJECT wordt geleid door juist 1 WERKNEMER.
    Een WERKNEMER kan min 0 en max N PROJECTen leiden.
    → 1:N (een één-op-veel-relatie)

HO GENT

Oefening Activiteiten

- Je wil een activiteitendatabank creëren
- Elke ACTIVITEIT krijgt een uniek volgnummer, een datum, een naam, een startuur, vermoedelijk einduur, een korte omschrijving en een kostprijs om deel te nemen. De combinatie van datum en naam is ook uniek.
- Elke ACTIVITEIT start en eindigt op een bepaalde LOCATIE. Een LOCATIE heeft unieke GIScoördinaten (breedtegraad en lengtegraad), een naam en een stad.
- Geef de relatietype(s)



Oefeningen

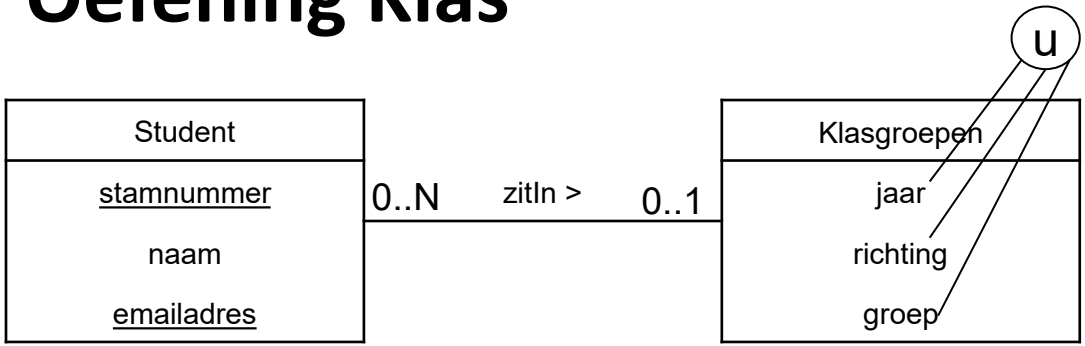
HO
GENT

Oefening Klas

- Een student kan worden geïdentificeerd aan de hand van zijn stamnummer of aan de hand van zijn e-mailadres. Een student heeft een naam.
- Een klasgroep heeft niet verplicht meerdere studenten.
- Een student zit maximaal in 1 klasgroep.
- De klasgroepen zijn genummerd per jaar en richting

HO GENT

Oefening Klas

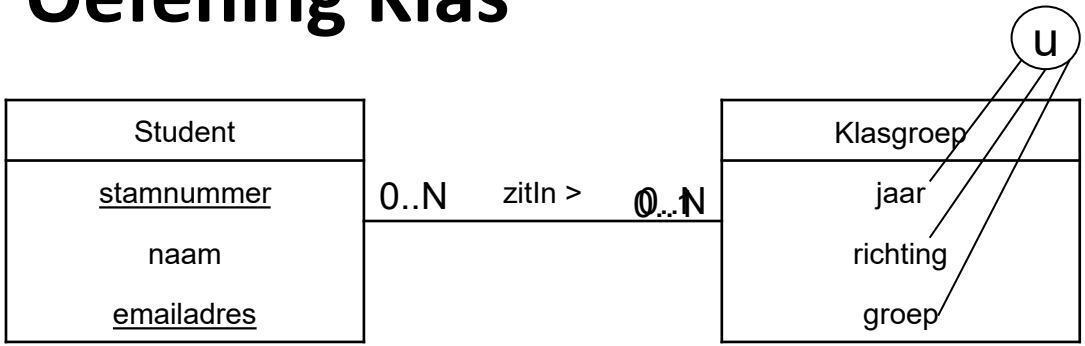


Pas het ERD aan

- Een klasgroep wordt uniek geïdentificeerd aan de hand van de combinatie van jaar, richting en groep.

HO GENT

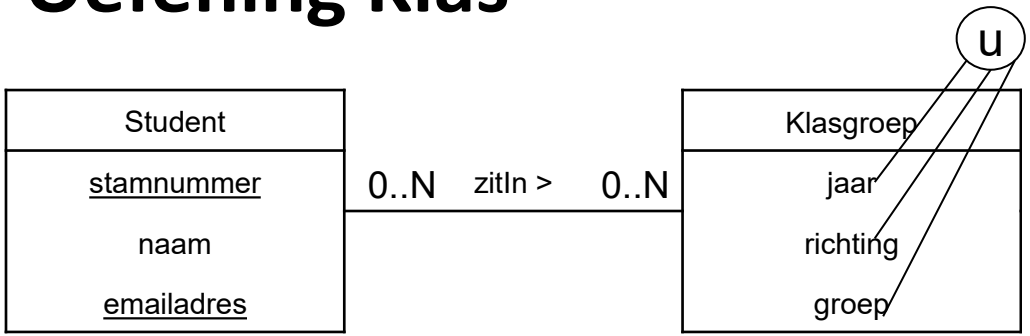
Oefening Klas



Pas het ERD aan

- Een student kan in meerdere klasgroepen zitten.

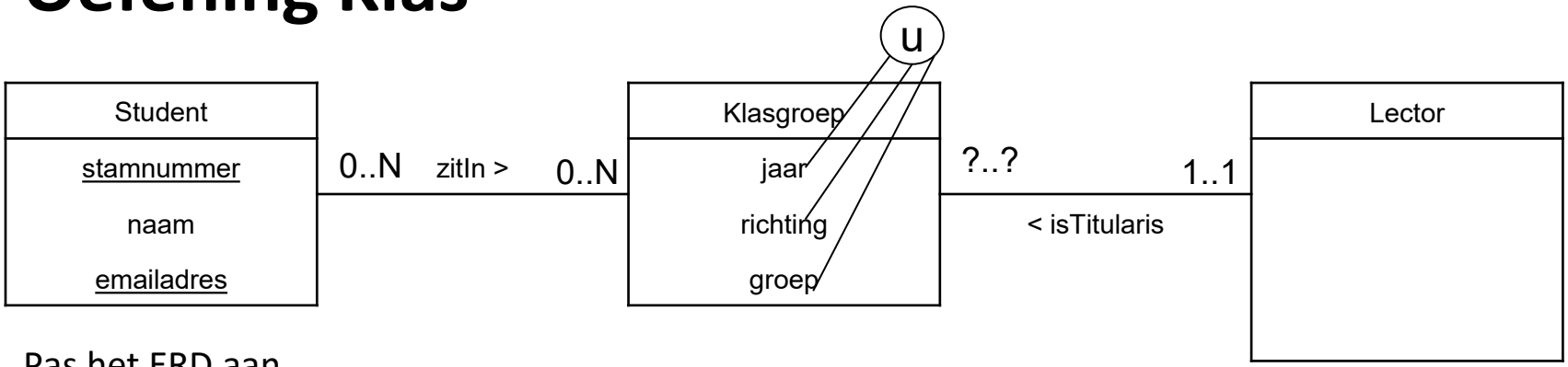
Oefening Klas



Pas het ERD aan

- Elke klasgroep heeft juist één lector als titularis.

Oefening Klas

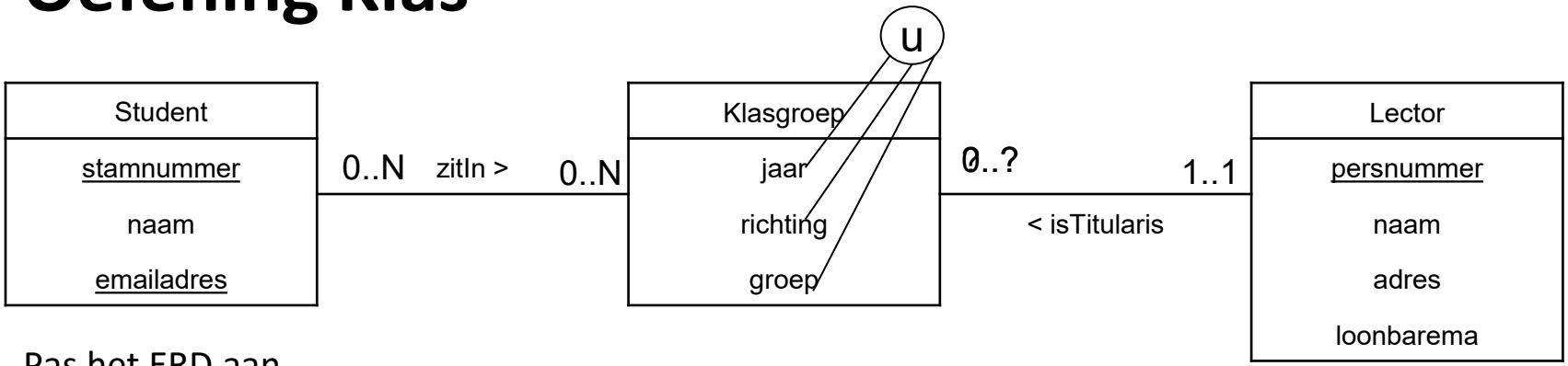


Pas het ERD aan

- Van een lector worden een uniek personeelsnummer, naam, adres en loonbarema bijgehouden

HO GENT

Oefening Klas

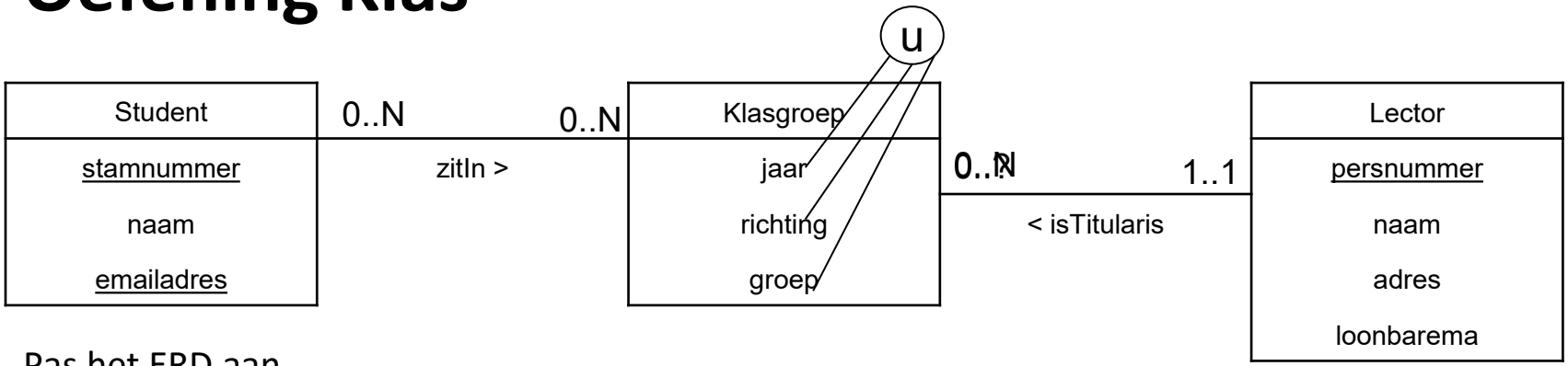


Pas het ERD aan

- Een lector hoeft geen titularis te zijn

HO GENT

Oefening Klas



Pas het ERD aan

- Een lector kan titularis zijn van meerdere klassen

Oefening Bibliotheek

- Een bibliotheek wil een databank ontwerpen voor het bijhouden van informatie over de boeken die er aanwezig zijn.
- Van elk boek is maar 1 exemplaar aanwezig in de bibliotheek.
- Er moet kunnen opgevraagd worden of een boek aanwezig is of niet en zo ja in welke rij het kan gevonden worden.
- Van elk boek moet volgende info kunnen opgevraagd worden: ISBN (unieke identificatie van een boek), titel, auteur(s), aantal blz, prijs, korte inhoud.
- Elk BOEK heeft minstens 1 auteur. Van die auteur kennen we een unieke naam.

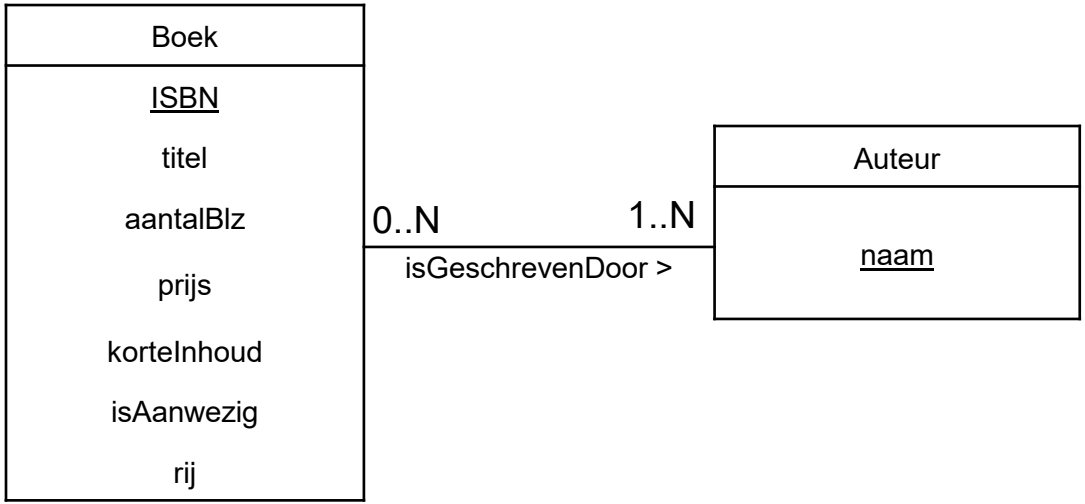
HO GENT

Oefening Bibliotheek

Boek
ISBN
titel
auteurs
aantalBlz
prijs
korteInhoud
isAanwezig
rij

HO GENT

Oefening Bibliotheek



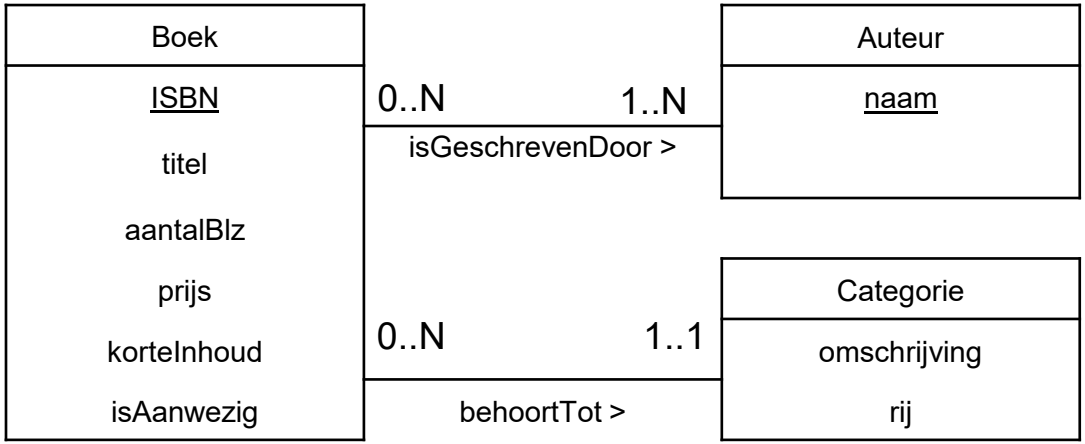
HO GENT

Oefening Bibliotheek

Breid het ERD uit:

- Elk boek behoort tot een bepaalde categorie (historische roman, thriller, ….).
- Elke categorie heeft een eigen plaats (rij) in de bibliotheek.

Oefening Bibliotheek



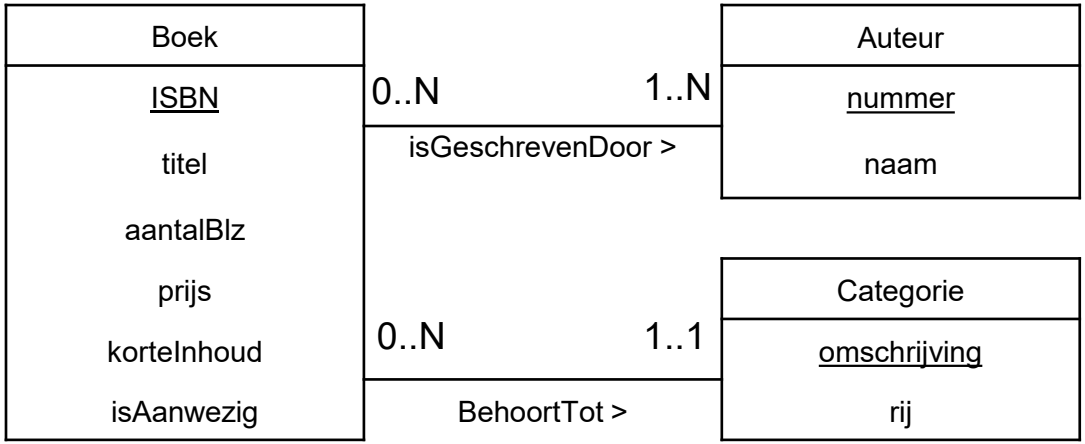
HO GENT

Oefening Bibliotheek

- Elk entiteittype moet zich kunnen identificeren.
- Hier hebben we geen identifier voor categorie.
- Als ontwerper mag je nooit zelf iets beslissen maar moet je overleggen met de opdrachtgever!
- Na overleg blijkt dat
  - een categorie geïdentificeerd wordt aan de hand van zijn omschrijving
  - naam van een auteur is geen goede identificatie → een oplopend nummer
- Pas het ERD aan!

HO GENT

Oefening Bibliotheek



HO GENT

Opletten met nummers als id!!!

- De bedoeling van de kandidaatsleutel is om iedere entiteit van een bepaald type uniek te identificeren
- Door een willekeurig nummer toe te voegen, stel je de databank die volgt op het ontwerp bloot aan mogelijk fouten

Je kan dezelfde auteur meermaals toevoegen onder een ander nummer

Goede vuistregel

- Gebruik enkel info die je aan de gebruikers van de DB en de DB-applicaties kan vragen
- Gebruik nummers alleen als die ook in de echte wereld bestaan of zullen bestaan
  - Nummerplaat, chassis-nummer
  - ISBN-nummer
  - Studentnummer
  - …

Goede vuistregel

- Gebruik nummers alleen als die ook in de echte wereld bestaan of zullen bestaan
  - Niet ~~categorienummer~~
  - Niet ~~auteurnummer~~
  - Niet ~~...~~
  - Anderzijds is het soms een gemakkelijke oplossing
  - Maar dan moet je in je applicatie(s) proberen vermijden dat records dubbel worden toegevoegd

Bronnen

HO
GENT

Bronnen

- Principles of database management (W. Lemahieu, S. Vanden Broucke, B. Baesens)
  - 3.1 + 3.2.1 + 3.2.2 + 3.2.3 + 3.2.4 + 3.2.6
- Principes van databases (G. De Tré)

HO GENT