

Hoofdstuk 4 : Entity Relationship Diagrammen

Een Entity-Relationship Diagram (ERD) is een diagram dat een grafische weergave vormt voor een hele reeks gegevens. Bedoeling is om eenvoudig en snel inzicht te krijgen in de benodigde informatie door verbanden tussen gegevens visueel weer te geven. Vervolgens kan dit ERD omgezet worden in een logisch gegevensmodel om dan uiteindelijk geïmplementeerd te worden in een database (= databank).

Een ERD is dus een diagram dat entiteiten, relaties en regels uit het logisch ontwerp visueel weergeeft.

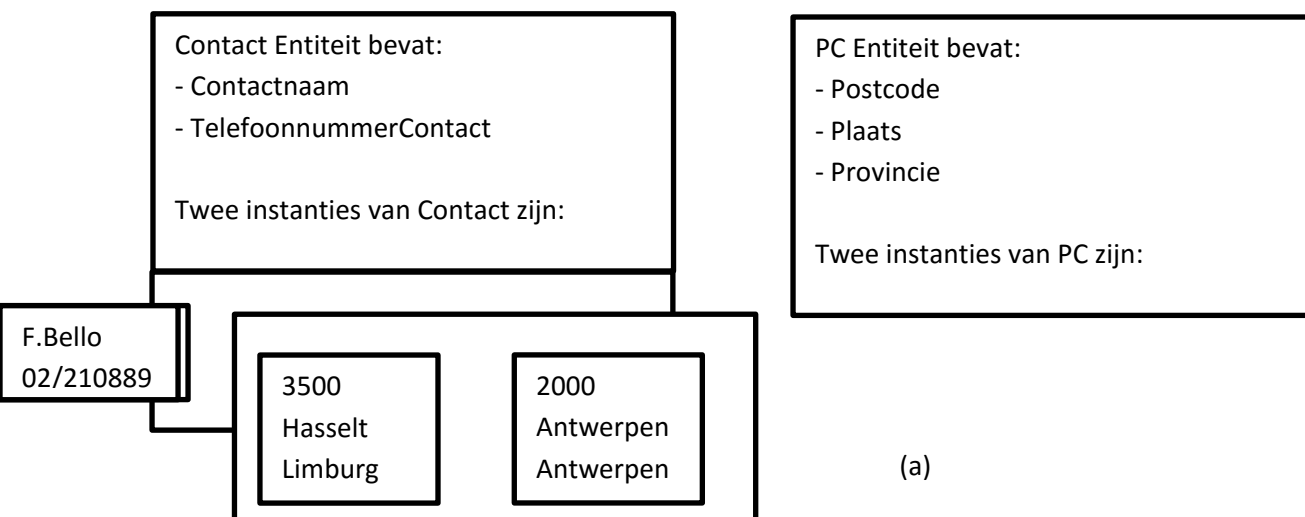
4.1 Elementen van een ERD

Er zijn verschillende notaties om een ERD voor te stellen. We beperken ons in deze cursus tot ERD's om de relaties tussen data stores, in kaart gebracht door normalisatie, aan te geven. Je kan ook perfect in een ERD de attributen weergeven. Daar komen we later op terug.

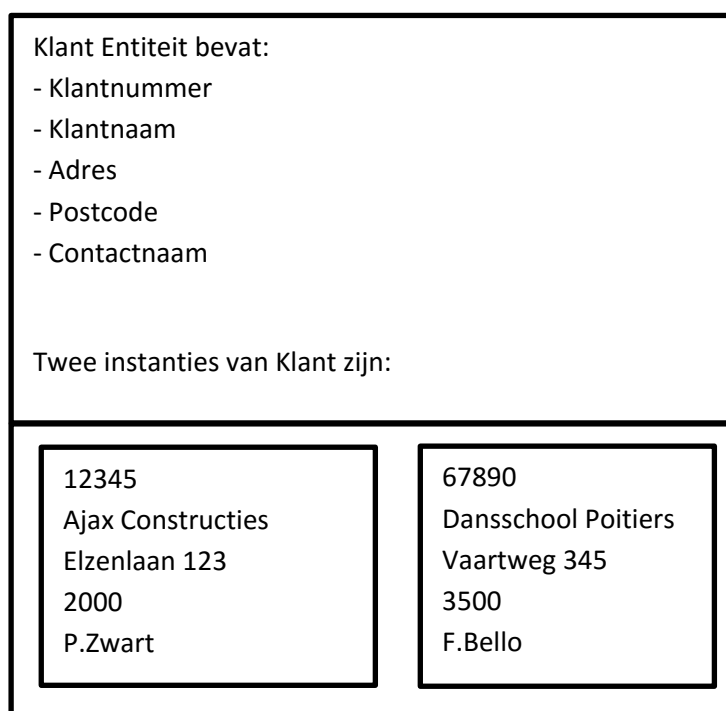
4.1.1 Entiteiten

Een entiteit kan meestal afgeleid worden uit het resultaat na normalisatie. Elke groep stelt een entiteit voor. Voorbeelden van entiteiten zijn Medewerker Marie Donker, Klant 12345, Verkooporder 1000, Verkoper Jan Smit en Product A42000. Entiteiten worden gegroepeerd in bepaalde entiteitsklassen. De Medewerker-entiteitsklasse is bijvoorbeeld de verzameling van alle medewerker-entiteiten (Marie Donkers, . . .).

Het is belangrijk dat je begrijpt wat de verschillen zijn tussen een entiteitsklasse en een entiteitsinstantie. Een entiteitsklasse is een verzameling van entiteiten en wordt beschreven door de structuur of indeling van de entiteiten in die klassen. Een instantie van een entiteitsklasse is de representatie van een bepaalde entiteit, zoals Klant 12345 en wordt beschreven door de waarden van attributen van de entiteit. Er bestaan meestal vele instanties van een entiteit in een entiteitsklasse. Binnen de klasse Klant zijn er bijvoorbeeld vele instanties, namelijk een voor elke klant die in de database voorkomt. In Figuur 4.1 zie je een aantal entiteitsklassen en twee bijbehorende instanties.



(b)



(c)

Figuur 4.1: Voorbeelden van entiteitsklassen met instanties

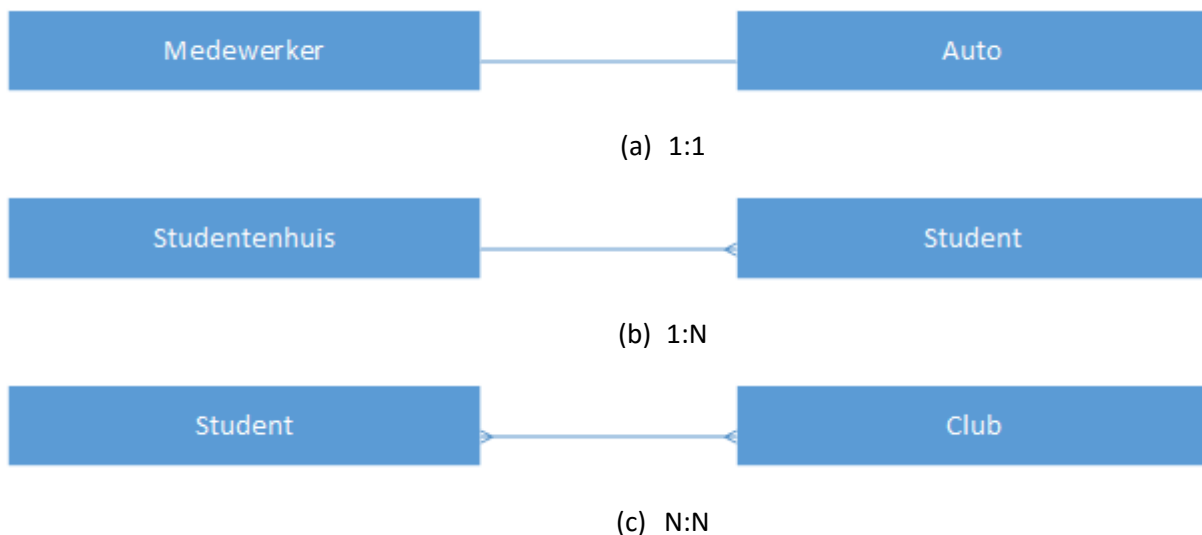
Entiteiten worden in een ERD voorgesteld door een rechthoek met daarin de naam van de entiteitsklasse in enkelvoud. We spreken af dat we de naam van een entiteitsklasse ook altijd beginnen met een hoofdletter. Als symbool gebruik je een rechthoek.



4.1.2 Relaties

Entiteiten kunnen met elkaar in verband worden gebracht in zogenoemde relaties. We beperken ons hierbij tot binaire relaties omdat deze het meest voorkomen. Een binaire relatie is een relatie tussen 2 entiteiten. Het aantal entiteiten dat aan een relatie deelneemt, is de graad van de relatie. Relaties met een graad groter dan 2 worden in de praktijk ontleed in een aantal binaire relaties.

Laat ons beginnen hoe we cardinaliteiten kunnen voorstellen in een ERD.



Figuur 4.2: Voorbeelden van ERD-relaties

4.1.2.1 1:1-relatie

In een 1:1-relatie (spreek uit “één op één”) is één instantie van één bepaalde entiteit gerelateerd met één instantie van een andere entiteit. In Figuur 4.2 (a) koppelt de relatie één Medewerker aan één Auto. Volgens het diagram is er aan geen enkele werknemer meer dan één auto toegekend en is geen enkele auto toegekend aan verscheidene werknemers.

4.1.2.2 1:N-relatie

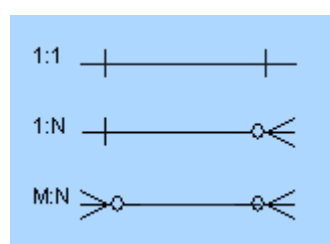
In Figuur 4.2 (b) zie je het tweede type relatie: 1:N (spreek uit “één op N” of “één op veel”). In de relatie staat één bepaalde instantie van Studentenhuis in verband met vele instanties van Student. In een studentenhuus wonen vele studenten, maar een bepaalde student bewoont slechts één studentenhuus.

4.1.2.3 M:N-relatie

In Figuur 4.2 (c) zie je het derde type binaire relaties: M:N (spreek uit “M op N” of “veel op veel”). De relatie Student-Club verbindt instanties van Student met instanties van Club. Een student kan lid worden van meer dan één club en een club kan vele studenten als lid hebben. Dit type mag na het normaliseren niet meer voorkomen!!!

4.1.2.4 Optionaliteit

De *optionaliteit* van een relatie geeft aan, hoeveel elementen van een betrokken entiteitstype minimaal in relatie moeten staan met één instantie van de andere entiteitstypes. Daarbij zijn de toegelaten waarden: nul of één. De kant waar er verplicht record(s) aanwezig moeten zijn krijgt een streep haaks op de lijn. Niet-verplicht wordt aangegeven met een cirkel ipv een streep.



Figuur 4.3: Voorbeeld van minstens één relatie



Figuur 4.4: Voorbeeld van optionele relatie

Bij de 1 : N-relaties in Figuur 4.2 (b) zeggen we enkel “Een instantie van de entiteit Student kan bij maximaal 1 instantie Studentenhuis zijn aangesloten”. Maar dus niet dat elke instantie van Student bij Studentenhuis moet voorkomen. Als je duidelijk wilt aangeven dat elke instantie van een entiteit iets minstens één moet hebben, geef je dit aan met een verticale streep. In Figuur 4.3 eis je, in tegenstelling tot Figuur 4.2 (b), dat iedere instantie van Student gekoppeld is met exact één instantie van Studentenhuis en elke instantie van Studentenhuis gelinkt is met minstens één instantie van Student.

Je kan ook aangeven dat een relatie optioneel is, dit doe je met een bolletje of met de O van Optioneel. In Figuur 4.4 toon je dat iedere instantie van Student gekoppeld is aan geen of hoogstens één instantie van Studentenhuis, en elke instantie van Studentenhuis gelinkt is met minstens één instantie van Student.

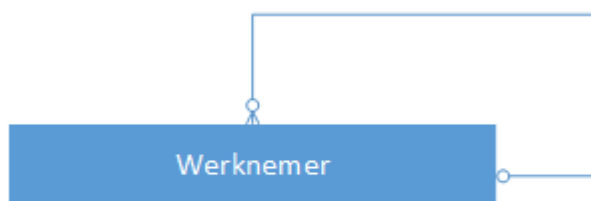
4.1.2.5 Mogelijke relaties en naamgeving

Bovenvermelde figuren van relaties zijn de meest voorkomende. Maar je kan ook een relatie leggen tussen één en dezelfde entiteit zoals getoond in Figuur 4.5. Een werknemer, dus een instantie van Werknemer, kan een andere werknemer als baas hebben en een werknemer kan geen of meerdere werknemers onder hem hebben staan werken.

Dit betekent in praktijk dat de entiteit Werknemer een attribuut overste of dergelijke zal bevatten. Het kan perfect zijn dat één entiteit meerdere attributen heeft die een relatie hebben met eenzelfde, andere, entiteit.

Figuur 4.6 toont de entiteit Trein waarvan we weten dat deze de attributen bestuurder en begeleider heeft. Zowel bestuurder als begeleiders zijn werknemers. Maar een trein wordt bestuurd door exact één bestuurder en moet minstens één, mogelijk meer begeleiders hebben.

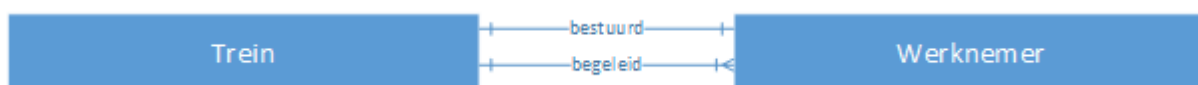
Welke relatie slaat nu op welk attribuut? Het is toegestaan om de relaties een naam te geven zoals in Figuur 4.7.



Figuur 4.5: Voorbeeld van een relatie met slechts 1 entiteit



Figuur 4.6: Voorbeeld van twee relaties tussen twee entiteiten



Figuur 4.7: Voorbeeld van twee relatie tussen twee entiteiten met namen voor de relaties

Besluit:

Voor deze cursus spreken we af om steeds zowel de cardinaliteit als de optionaliteit te vermelden.



minstens 1, maar kunnen er ook meer zijn



0, 1 of veel



verplicht 1



0 OF 1

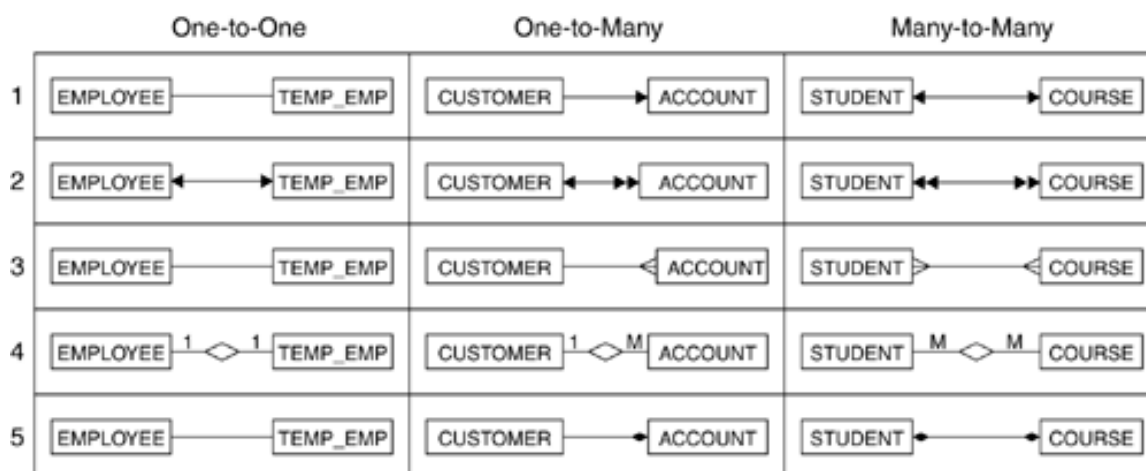
4.2 Diagrammen praktisch

Zoals in het begin van het hoofdstuk verteld, zijn er verschillende notaties voor ERD's. Figuur 4.8 toont de vijf meest voorkomende notaties.

1. Ross-notatie: Ronald G. Ross [9]
2. Bachmann-notatie: Charles William Bachman [1]
3. Martin-notatie: James Martin [7]
4. Chen-notatie: Peter Pin-Shan Chen [3]
5. Rumbaugh-notatie: James Rumbaugh [10]

Het is daarom aan te raden om als bijlage van een analyse altijd een symbolenlijst met conventies op te nemen om vergissingen te voorkomen. Want in Figuur 4.8 merk je dat de voorstelling van de 1-op-1 relatie van Bachmann identiek is aan de Veel-op-veel relatie van Ross.

De figuren in dit hoofdstuk zijn getekend met Visio. Je kan ook ERD's tekenen met Dia maar die gebruikt weer een andere notatie. Je bent niet verplicht om deze notatie te volgen, maar indien je afwijkt van deze notatie, meld dit dan duidelijk en voeg de gebruikte symbolen met uitleg toe als bijlage.



Figuur 4.8: Vijf verschillende notaties voor ERDs.

Bibliografie

- [1] Charles William Bachman. "Data structure diagrams". In: Database (1969), p4-10.
- [2] Guido Bakema, Jan Pieter Zwart en Harm van der Lek. Fully Communication Oriented Information Modeling (FCO-IM). Bommelj'e Cromptvoets en partners (BCP), 2002.
- [3] Peter Pin shan Chen. "The Entity-Relationship Model: Toward a Unified View of Data". In: ACM Transactions on Database Systems 1 (1976), p. 9-36.
- [4] Edgar Frank Codd. "A Relational Model of Data for Large Shared Data Banks". In: Commun. ACM 13.6 (1970), p. 377-387.
- [5] Isabelle Godfrind. Dataorganisatie fundamentals. Xios Hogeschool Limburg, 2012.
- [6] Instruct. Informatica Online. Dec 2007. url : <http://www.instruct-online.nl/>.
- [7] James Martin en Clive Finkelstein. Information Engineering. Savant institute, 1981.
- [8] Morgan Masters. An Introduction to Data Flow Diagrams. Jan 2012. Url: <http://www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/articleId/2009/An-Introduction-to-Data-Flow-Diagrams.aspx>
- [9] Ronald G. Ross. Entity modeling: techniques and application. Database Research Group, 1987. Isbn: 9780941049009. url : <http://books.google.be/books?id=JqoZAQAIAAJ>.
- [10] James Rumbaugh. "A Parallel Asynchronous Computer Architecture for Data Flow Programs". Proefschrift. M.I.T. Laboratory for Computer Science, Cambridge, MA, mei 1975.
- [11] Ed Yourdon. Structured Analysis: Dataflow Diagrams. Jun 2011. Url : http://www.yourdon.com/strucanalysis/wiki/index.php?title=Chapter_9.