



PXL-Digital

Toegepaste Informatica

Data Essentials: ERD

Lector

Carine Derkoningen

Isabelle Godfrind

Carina Medats

Kerstin Nys

I. Inhoudsopgave

I.	Inhoudsopgave	1
II.	Entity Relationship Diagram.....	2
1	Elementen van een ERD.....	2
1.1	Entiteiten	2
1.2	Relaties	4
1.2.1	Kardinaliteit	4
1.2.2	Optionaliteit	7
1.2.3	Mogelijke relaties en naamgeving	9
1.2.4	Besluit.....	10
1.2.5	Verschillende notaties relaties.....	11
III.	Bibliografie.....	12

II. Entity Relationship Diagram

Een Entity-Relationship Diagram (ERD) is een diagram dat een grafische weergave vormt voor een hele reeks gegevens. Het is de bedoeling om op een eenvoudige manier 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).

Door het normaliseren ontstaan entiteitsklassen (de tabellen) en relaties (FK =>PK) en het ERD is een diagram dat de entiteiten, de relaties en de regels visueel weergeeft.

1 Elementen van een ERD

Er zijn verschillende notaties om een ERD voor te stellen. In deze cursus worden relaties in de ERD's met de Martin notatie aangegeven. Om het overzicht te bewaren wordt de weergave beperkt tot de entiteitsklasse (zie verder) en de primary keys. Maar je kan ook perfect in een ERD de attributen, in kaart gebracht door normalisatie, weergeven.

Om een ERD te tekenen wordt altijd vertrokken van de NV3 met aanduiding van de FK-relaties. Het zijn immers die relaties die hier grafisch worden weergegeven. De tool die tijdens de lessen gebruikt wordt is LucidChart.

1.1 Entiteiten

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

Het is belangrijk dat je begrijpt wat de verschillen zijn tussen een entiteitsklasse, een entiteit en een entiteitsinstantie. Een entiteitsklasse is een verzameling van entiteiten en wordt beschreven door de structuur of indeling van de entiteiten in die klassen. Voorbeeld Klanten (klantnummer). Een instantie van een entiteitsklasse is de representatie van een bepaalde entiteit, zoals 12345 voor 1 klant uit de groep Klanten en wordt beschreven door de waarden van attributen, in dit voorbeeld het klantnummer, van de entiteit. Er bestaan meestal vele instanties van een entiteit in een entiteitsklasse. Binnen de klasse Klanten zijn er bijvoorbeeld vele instanties, namelijk één voor elke klant die in de database voorkomt. In Figuur 1.1 zie je een aantal entiteitsklassen en twee bijbehorende instanties.

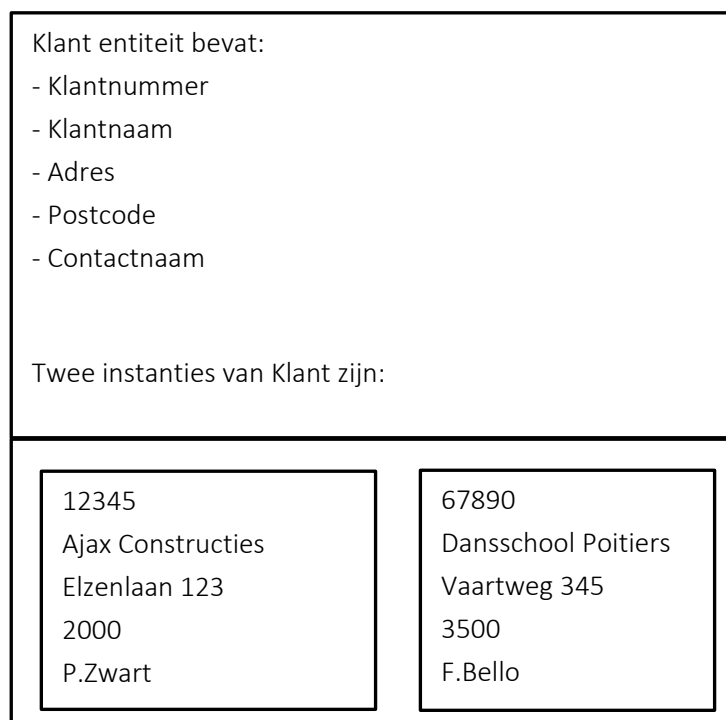
- (a) de entiteitsklasse Contacten; binnen die entiteitsklasse heeft 1 entiteit de attributen Contactnaam en TelefoonnummerContact. Elke instantie van de entiteit Contact heeft een waarde voor Contactnaam en TelefoonnummerContact.
- (b) de entiteitsklasse Postcodes met 2 instanties.
- (c) de entiteitsklasse Klanten



(a)



(b)



(c)

Figuur 1.1: Voorbeelden van entiteitsklassen met instanties

Entiteiten worden in een ERD voorgesteld door een rechthoek met daarin de entiteit (= de entiteitsklasse in enkelvoud). De naam van een entiteit begint altijd met een hoofdletter. Als symbool gebruik je een rechthoek.

Klant	
PK	klantnummer

1.2 Relaties

Entiteiten kunnen met elkaar in verband worden gebracht in zogenoemde relaties. In deze cursus is er enkel sprake van 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.

Zoals eerder gesteld komen na het normaliseren de relaties voort uit de FK => PK relaties die aangebracht worden na de derde normaalvorm. Als de entiteiten allemaal bekend zijn, kunnen de relaties eventueel al aangebracht worden door met een lijn de entiteiten met elkaar te verbinden. In deze cursus worden niet enkel de entiteiten met elkaar verbonden, maar wordt ook ingegaan hoe de relatie tussen de entiteiten is. Hiervoor wordt gekeken naar 2 dingen: de kardinaliteit en de optionaliteit.

1.2.1 Kardinaliteit

De kardinaliteit geeft aan hoeveel keer een bepaalde instantie van een attribuut uit de ene entiteit kan voorkomen in de entiteitsklasse van de andere entiteit.

Het bepalen van de kardinaliteit van de relatie tussen de entiteit met de FK en de entiteit met de PK wordt gedaan door 2 vragen te beantwoorden:

1. Als een waarde van het FK-attribuut uit de relatie uit die tabel genomen wordt, hoe vaak kan deze waarde voorkomen in de tabel waar dit het PK-attribuut is? Vermits het de PK in de tabel is, kan deze waarde hier dus maximum één keer voorkomen. De kardinaliteit van een relatie aan de kant van de PK in die relatie is dus steeds één.
2. Als een waarde van het PK-attribuut uit de relatie uit die tabel genomen wordt, hoe vaak kan deze waarde voorkomen in de tabel waar dit attribuut de FK in de relatie is? Hier kan niet

zomaar een antwoord gegeven worden en er moet dus echt naar de waarden gekeken worden. In de meeste gevallen gaat dit meerdere keren zijn. Heel uitzonderlijk kan dit ook één zijn. Het antwoord is de kardinaliteit aan de kant van de FK in de relatie.

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



(a) 1:1



(b) 1:N



(c) M:N

Figuur 1.2: Voorbeelden van ERD-relaties

1.2.1.1 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 1.2 (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.

1.2.1.2 1:N-relatie

In Figuur 1.2 (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.

1.2.1.3 M:N-relatie

In Figuur 1.2 (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!!!** Verder in deze cursus zal deze relatie dus ook niet meer aan bod komen.

1.2.1.4 Uitgewerkt voorbeeld

Van het voorbeeld uit de cursus normalisatie worden deze 2 entiteitsklassen genomen:

Projectmedewerkers: Projectnummer FK =>Projecten(Projectnummer)
Medewerkersnummer FK=>Medewerkers(Medewerkersnummer)
Uren

Medewerkers: Medewerkersnummer
Medewerkersvoornaam
Afdeling FK=>Afdelingen(Afdeling)



Figuur 1.5 Voorstelling entiteiten

De relatie tussen deze twee entiteiten is in Projectmedewerker de FK medewerkersnummer met de PK medewerkersnummer in de entiteit Medewerker. Nu worden bovenstaande vragen toegepast op deze 2 entiteiten:

1. Als een medewerkersnummer genomen wordt uit tabel Projectmedewerkers, hoe vaak kan datzelfde medewerkersnummer voorkomen in de tabel Medewerkers? Het antwoord op deze vraag, bepaalt de kardinaliteit aan de kant van de entiteit Medewerker.

- Als een medewerkersnummer genomen wordt uit de tabel Medewerker, hoe vaak kan datzelfde medewerkersnummer voorkomen in de tabel Medewerkers? Het antwoord op die vraag, bepaalt de kardinaliteit aan de kant van de entiteit Projectmedewerker.

Het antwoord op de vragen is eenvoudig als twee voorbeeldtabellen met de gegevens van de documenten worden ingevuld.

Medewerkers		
medewerkersnummer	medewerkersvoornaam	afdeling
003	Ton	Analyse
004	Peter	Programmering
005	Fred	Analyse

Projectmedewerkers		
projectnummer	medewerkersnummer	uren
001	003	60
001	005	100
002	004	200
002	005	50

Vraag 1: vermits het medewerkersnummer de PK is in de tabel Medewerkers, kan éénzelfde nummer maar één keer voorkomen in die tabel. Eender welk medewerkersnummer uit de tabel Projectmedewerkers zal maar één keer te vinden zijn in de tabel Medewerkers. Dus de kardinaliteit aan de kant van de Medewerker is één.

Vraag 2: het medewerkersnummer 005 uit de tabel Medewerkers komt meerdere keren voor in de tabel Projectmedewerkers. Het antwoord op de vraag of een waarde voor het attribuut medewerkersnummer uit de tabel Medewerkers één of meerdere keren voorkomt in de tabel Projectmedewerkers is dus meerdere. De kardinaliteit aan de kant van de Projectmedewerker is dus veel.

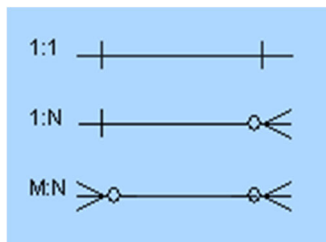


Figuur 1.6: Voorstelling relatie met kardinaliteit: N:1

1.2.2 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 i.p.v. een streep.



Figuur 1.7: Voorbeeld van minstens één relatie



Figuur 1.8: Voorbeeld van een optionele relatie aan de zijde van kardinaliteit 1



Figuur 1.9: Voorbeeld van een optionele relatie aan de zijde van kardinaliteit veel

Bij de 1 : N-relaties in Figuur 1.8 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 1.7 eis je, in tegenstelling tot Figuur 1.8, 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 1.8 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. In figuur 1.9 toon je dat iedere student gekoppeld is aan minstens 1 studentenhuis, maar dat ieder studententhuis meerdere of geen studenten heeft (dat laatste is misschien wat raar, maar dit kan bij een nieuwbouw).

Veelal kan je de optionaliteit afleiden uit de opmerkingen bij een oefening (cfr gesprek met klant).

Toegepast op het uitgewerkte voorbeeld kardinaliteit:



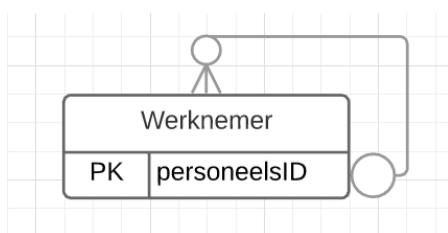
Figuur 1.10: relatie met kardinaliteit en optionaliteit

Een medewerker die toegewezen is aan een project, moet aanwezig zijn in de tabel Medewerkers.

Het kan zijn dat er medewerkers zijn die niet toegewezen worden aan een project, denk maar aan de mensen van de dienst HR. Zij komen dus wel voor in de Medewerkerstabel, maar niet in de Projectmedewerkerstabel.

1.2.3 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 1.11. 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 werken.



Figuur 1.11: voorbeeld van een relatie met slechts 1 entiteit.

Dit betekent in praktijk dat de entiteit Werknemer een attribuut overste of dergelijke zal bevatten.

Denk maar aan de verhouding werknemer en manager in de Oracle tabel employees in de cursus SQL.

Figuur 1.12 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 1.13.



Figuur 1.12: Voorbeeld van twee relaties tussen twee entiteiten.



Figuur 1.13: Voorbeeld van twee relaties tussen twee entiteiten met namen voor de relaties.

1.2.4 Besluit

In deze cursus worden steeds zowel de kardinaliteit als de optionaliteit vermeld. In de oefeningen worden ook telkens beide aangeduid.

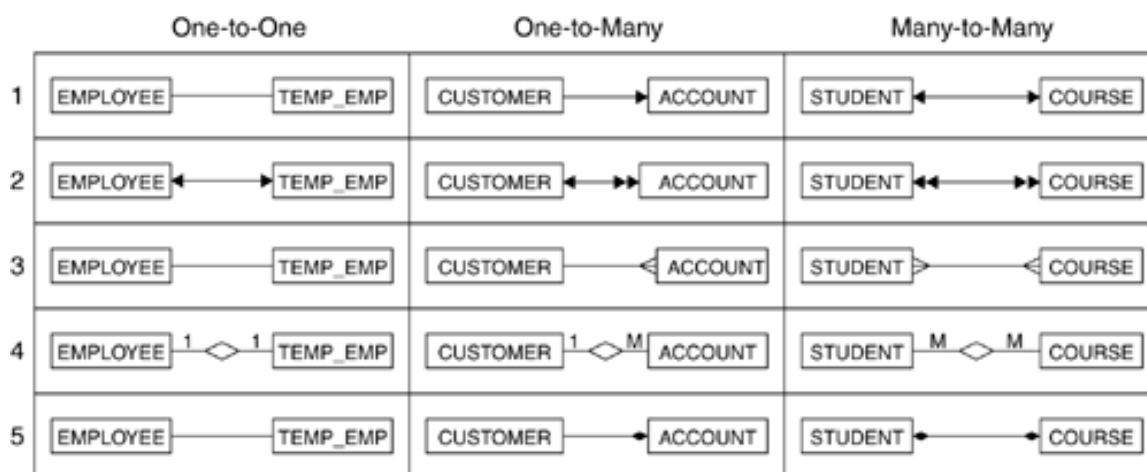
De mogelijkheden zijn:

	minstens 1, maar kunnen er ook meer zijn
	0, 1 of veel
	verplicht 1
	0 OF 1

1.2.5 Verschillende notaties relaties

Zoals in het begin van deze cursus aangegeven, zijn er verschillende notaties om de kardinaliteiten in een ERD aan te duiden. Voor de volledigheid hier 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]



Figuur 1.14: Vijf verschillende notaties voor ERD's

De afspraak in Data Essentials is om in de oefeningen (ook op het examen) de **3: Martin-notatie** te gebruiken om discussies te vermijden.

Zoals uit figuur 1.14 op te maken is, is zo een afspraak zeer belangrijk, want de voorstelling van de 1-op-1 relatie van Bachmann(2) is identiek aan de veel-op-veel relatie van Ross(1). In het bedrijfsleven wordt er als bijlage van een analyse altijd een symbolenlijst met de conventies opgenomen bij het ERD.

III. 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 Crompvoets 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=JqoZAQAAIAAJ>.
- [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.