

Entity-Relationship Diagram (ERD) Modelleren

Huub de Beer

Eindhoven, 4 juni 2011

Informatiesysteemontwikkeling en modelleren in fasen

1. Gegevensstromen naar en van het systeem: DFD
2. Gegevens die in het systeem beschikbaar moeten zijn: gegevensmodel met ERD
3. Gegevensmodel omzetten naar een database
4. Gegevens in een database stoppen
5. Gegevens in de database gebruiken

DFDs, de invoer/uitvoer van het systeem en ERDs

Het ontwerpen van een database

- ▶ Met behulp van een contextdiagram geef je aan wat buiten het systeem zit (buitenwerelden), welke informatie er in gaat (invoer) en welke informatie het systeem “produceert” (uitvoer)
- ▶ Om die invoer te verwerken en de uitvoer te produceren moeten bepaalde gegevens in het systeem bewaard worden. (vaak buffers in het gedetailleerde DFD)
- ▶ Het vaststellen welke informatie bewaard moet worden, heet het vaststellen van het datamodel. We doen dat met behulp van een Entity-Relationship Diagram (ERD; acroniem wordt steeds duidelijker)
- ▶ Je maken van een gegevensmodel met behulp van een ERD noemen we ook wel het ontwerpen van een database

Voorbeeld: casus VeloVeniVici

In het kort (zie ook Eckartnet):

- ▶ registreren van klanten met behulp van naam, adres en geboortejaar
- ▶ verhuren van fietsen; betalen van borg en huur: administratie
- ▶ voorraad fietsen en onderdelen; leveranciergegevens?
- ▶ lijst met vermoedelijk gestolen fietsen voor de politie
- ▶ het verkopen van fietsen aan klanten en als oud ijzer: administratie

Entiteiten en attributen

Entiteit

Een ding, persoon, object waarvan de gegevens in de database opgenomen moeten worden. (Dit kunnen ook abstracte dingen zijn)

Attribuut

De (belangrijke) eigenschappen van een entiteit noemen we de attributen van die entiteit

Entiteiten en attributen (II)

Voorbeelden

- ▶ Klanten, fietsen, onderdelen, leveranciers en verhuuring zijn entiteiten
- ▶ Van een klant zijn de naam, het adres en het geboortjaar belangrijk. Dat zijn dus attributen
- ▶ Klanten hebben ook een huidskleur, haarkleur en hoogst genoten opleiding. In deze casus is dat niet relevant: dat zijn dus geen attributen. (In een casus van een dermatoloog misschien weer wel)
- ▶ De verhuurde fiets en de huurder (klant) zijn geen attributen, maar zijn natuurlijk wel belangrijk

Relaties tussen entiteiten

Relatie

Entiteiten kunnen een relatie met elkaar hebben; die relatie is belangrijke informatie in het datamodel

Attribuut

Een relatie kan ook attributen hebben.

Voorbeelden

- ▶ Een verhuuring is een relatie tussen een bepaalde klant en de gehuurde fiets. Attributen van een verhuuring kunnen de huurdatum, de borg en de huurtijd zijn.
- ▶ Een fiets is geleverd door een bepaalde leverancier; tussen die fiets en die leverancier bestaat dus een relatie.

Generaliseren tot typen

- ▶ In de database bewaren we de gegevens van entiteiten.
- ▶ Bij het ontwerpen van de database willen we het niet alleen over die daadwerkelijke entiteiten hebben (zoals de klant Jan, geboren op 3 mei 1977 te Eindhoven) maar over alle mogelijke klanten.
- ▶ We generaliseren vergelijkbaren entiteiten tot een entiteitstype: de entiteiten Jan, Piet, Marie, Ailya, enz. behoren tot het entiteitstype *Klant*.
- ▶ Evenzo generaliseren we daadwerkelijke attributen tot attribuuttypen (3 mei 1977 behoort tot het attribuuttype *geboortedatum*) en relaties tot relatietypen (Fiets nr. 2344 is geleverd door Gazelle behoort tot het relatietype *geleverd_door*).

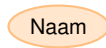
Het tekenen van een Entity-Relationship diagram

Symbolen

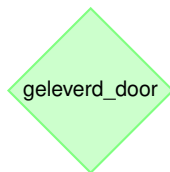
Entiteitstype zn. ev.



Attribuuttype zn. (ev.)

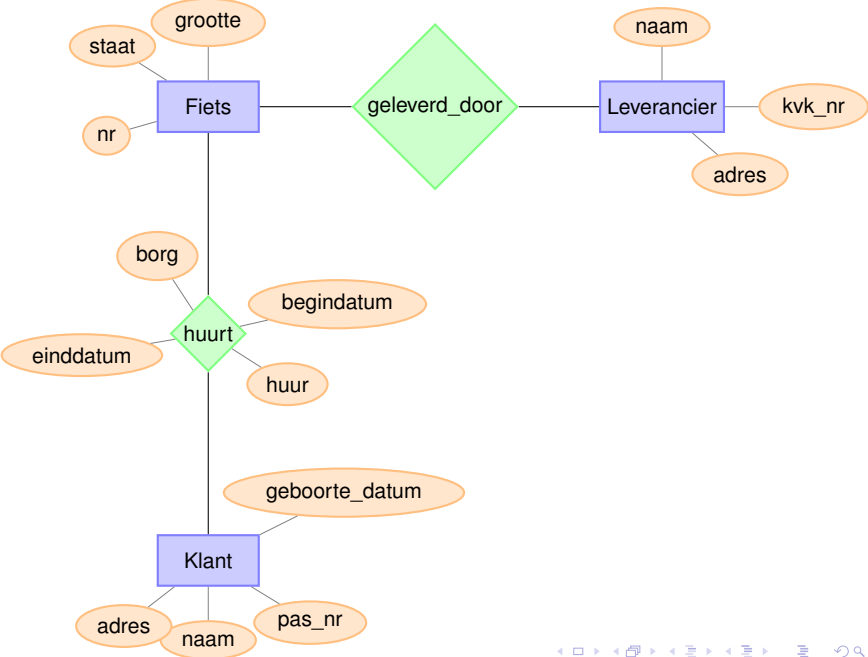


Relatietype ww. (+ vZ)



- ▶ Schrijf spaties als “_”
- ▶ Verbind relatietypen, entiteitstypen en attribuuttypen met lijnen

Voorbeeld



Tot slot

- ▶ In een DFD geef je aan welke informatie door het systeem *stroomt*, dat kun je als dynamisch kenschetsen
- ▶ Processen verwerken informatie; personen “zitten” als het ware in die processen.
- ▶ In een ERD geef je aan welke informatie bewaard wordt: informatie in een database is *statisch*.
- ▶ Informatie zit in een database, of niet.
- ▶ Hoe die informatie in de database komt, of welke eisen er aan gesteld worden, vind je niet terug in een ERD.
- ▶ Geen links tussen attribuuttypen onderling of entiteitstypen onderling.
- ▶ Attribuuttypen zijn gekoppeld aan entiteitstypen of relatiestypen
- ▶ Relatiestypen koppelen entiteitstypen.
- ▶ Een entiteitstype kan een relatie met zichzelf hebben; of met meer dan een andere entiteitstype.

Na het slot

- ▶ Ken het verschil tussen entiteiten en entiteitstypen, attributen en attribuuttypen, relaties en relatietypen
- ▶ In dagelijkse communicatie gebruiken we dat gewoon door elkaar.
- ▶ Maar ben je wel bewust van het verschil.