

Relaties

- is een verzameling waarin ieder element bestaat uit een vaste reeks elementen uit andere verzamelingen
- Bijvoorbeeld:
 - elke student bestaat uit elementen uit de verzamelingen voorletters, namen, geboorte-data enz.
- elementen mogen vaker dan eens voorkomen binnen een relatie (**MITS** hun rol verschilt)
- worden vaak als tabellen weergegeven
- beschrijft in een database een entiteit of de verbinding (relatie ;o)) tussen entiteiten

Domeinen

- Basisverzamelingen heten domeinen
- Niet alle elementen uit een domein hoeven voor te komen in een relatie
- een domein mag vaker binnen een relatie voorkomen, maar moet dan wel verschillende rollen hebben (bijvoorbeeld geboortedatum en examendatum)
- de rol die een domein binnen een relatie vervult is het attribuuttype (= veldnaam)

Relaties: uniciteit

- een relatie bevat geen duplicaat tupels
- elk veld dat of combinatie van velden (!) die een tuple uniek maakt is een kandidaatsleutel
- een relatie kan meerdere kandidaatsleutels hebben
- een kandidaatsleutel mag geen andere kandidaatsleutel bevatten en mag geen overbodige gegevens bevatten
- per relatie wordt 1 kandidaat sleutel aangewezen als primaire sleutel. Velden in de PK mogen nooit NULL zijn.

Relaties: volgorde

- Relaties zijn ongeordend!
- attribuuttypen zijn ongeordend!

Attributen

- zijn atomair zijn niet deelbaar zonder betekenis te verliezen
- ze moeten behoren tot het onderliggende domein (“aap” is geen postcode)
- Volledig functioneel afhankelijk van de volledige primaire sleutel

Afhankelijkeden

- een Oscar voor de beste acteur
 - is afhankelijk van acteur
 - is afhankelijk van film
 - is VOLLEDIG FUNCTIONEEL AFHANKELIJK van de combinatie film-acteur (=rol)

Relationele Database:

- vastgelegd in een schema = betekenis van de relaties
- Welke tabel bevat welke attributen
- Wat zijn de onderlinge relaties tussen tabellen

Foreign Key

- Legt vast dat er een relatie tussen twee tabellen bestaat
- als het domeintype van een veld overeenkomt met het domeintype van de PK van een andere tabel, heb je (mogelijk) een FK
- Referentiele integriteit: iedere waarde van een FK moet ook in de gerefereerde PK aanwezig zijn

ontwerpen van een database

- welke 3 stappen?
 - specificeren
 - modelleren
 - implementeren

Gegevensverzamelingen

- moeten aan regels (=normen) voldoen => genormaliseerd zijn
- Bij het normaliseren worden de gegevens die logisch bij elkaar horen (functioneel afhankelijk zijn) tot een eenheid (entiteit) samengevoegd
- meerdere entiteiten bij elkaar vormen de gegevensstructuur

Normaliseren

- Boyce-Codd normaalvorm
- Alle attributen van een entiteit zijn volledig functioneel afhankelijk van de volledige sleutel en tussen attributen zijn er geen functionele afhankelijkheden

Eisen relationele data

- gegevens zijn atomair
- geen herhalende kolommen in een tabel
- per tabel zijn de gegevens volledig functioneel afhankelijk van de volledige sleutel
- er zijn binnen een tabel geen gegevens (gedeeltelijk) afhankelijk van niet-sleutel attributen
- records zijn uniek
- (geen proces data)

- Het normaliseren bestaat uit een aantal stappen:
 - Inventariseer de gegevens
 - Verwijder de herhalende groepen en zet deze in een aparte tabel
 - Verwijder de gegevens die niet afhankelijk zijn van de gehele sleutel (en zet ze in een eigen tabel)
 - Verwijder de gegevens die afhankelijk zijn van niet-sleutelgegevens (en zet ze in een eigen tabel)

Inventariseer de gegevens

- Zoek de attributen
 - elementaire gegevens: kunnen niet verder opgedeeld worden zonder hun betekenis te verliezen
- Wat zijn sleutelgegevens?
- Zijn er proces gegevens?
 - niet bewaren (in principe)

In de praktijk

- ga je uit van informatiebehoeften
 - schermen, formulieren etc.
- elke info-behoefte wordt apart genormaliseerd
- daarna wordt er een model van gemaakt

Recept

- schrijf alle elementaire gegevens op
- verwijder proces gegevens
- groepeer de gegevens die logisch bij elkaar horen. Geef de groep een naam en ken een sleutel toe
- let steeds op de informatiebehoeften!

Recept

- Maak een diagram en vergelijk dat met de informatie behoeften. Controleer gegevens, sleutels en FK's

Recept

- controleer op afhankelijkheden van een deel van de sleutel
- controleer op afhankelijkheden tussen attributen
- controleer a.d.h.v. de informatiebehoefte of de nodige verbindingen tussen de groepen gelegd kunnen worden via FKs

Opdrachten

- Maak diagram n.a.v de informatie behoeften

Opdracht 1 informatiebehoefte 1

Firma Bla & Co Advocaten

Datum 5-10-2007

Stationsweg 666

1111 AA

Amsterdam

Klantnr: 12234

Factuur: 3456

Art.	Prijs	Aantal	Bedrag	
THX11 blok a 100 vel A4	2,00	10	20,00	
X15 ballpoint blauw	0,50	30	<u>15,00</u>	
				35,00
				BTW 7,00
				Totaal 42,00

Opdracht 2 informatiebehoefte 1

Hanze Hogeschool Groningen

Datum 5-10-2007

ILST

BIN

overzicht tentamens

student	cursus	datum	cijfer
parn	databases 1	1-10-2007	10
daab	databases 1	1-10-2007	0
sint	navigatie 10	2-10-2007	8

Opdracht 2 informatiebehoefte 2

Hanze Hogeschool Groningen

Datum 5-10-2007

ILST

BOV

Cursus overzicht

code	omschrijving	docent
-------------	---------------------	---------------

chem1	inleiding scheikunde	buur
-------	----------------------	------

chem2	nog een inleiding scheikunde	wiwa
-------	------------------------------	------

Opdracht 2 informatiebehoefte 3

Hanze Hogeschool Groningen

Datum 5-10-2007

ILST

BOV

studenten uit assen per docent

docent

student

buur

jan janssen

buur

jan klaassen

wiwa

kees van der velden

Opdracht 2 informatiebehoefte 4

Hanze Hogeschool Groningen

Datum 5-10-2007

ILST

BOV

studenten per mentor

mentor

student

kees van der velden (12-11-1990) jan janssen

buur

jan klaassen