

Relationele databases

Normaliseert eer gij begint!

Normalisatie is van zeer groot belang bij het ontwikkelen van een database applicatie.

De kern van de zaak, de doelstelling : Efficiënt beheer van data. Overbodige - herhalende informatie vermijden.

Normalization is the process of efficiently organizing data in a database. There are two goals of the normalization process: eliminating redundant data (for example, storing the same data in more than one [table](#)) and ensuring data dependencies make sense (only storing related data in a table). Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored.

Er bestaan in totaal 5 normalisatie stappen - deze werden o.a. ontwikkeld door E.F. Codd (grondlegger relationele database theorie) en bijgestuurd/bekritiseerd, ... door heel wat anderen.

1NF - 2NF - 3NF - 4NF - 5NF (NF = Normaal Vorm)

In welke mate elk van deze niveau's wordt toegepast is geen exacte wetenschap. Ze zijn 'slechts' een 'belangrijke' leidraad tot een betere oplossing.

De originele theoriën zijn vrij zwaar om te verteren - hierna volgt een korte uiteenzetting waar enkel de belangrijkste elementen worden aangehaald en voorzien van een voorbeeld.

Verderop wordt dit ook in de praktijk toegepast met een voorbeeld.
meer volledige lezing voor de liefhebber :

- http://en.wikipedia.org/wiki/Database_normalization of
- <http://www.bkent.net/Doc/simple5.htm>
-

Stellingen ivm genormaliseerde vorm (Normalized Form = NF)

Theorie

1. Volgorde van rijen moet onbelangrijk zijn
2. Op elke kruising tussen rij en kolom mag slechts 1 waarde staan.
3. Geen herhalende rijen
4. Geen herhalende kolommen / volgorde kolommen moet onbelangrijk zijn

Voorbeeld : adreslijst

A° zie relational database 0.xls

Company	street	zip	city	contact	EmailAddress	function
Infortica	Kleine Doornstraat	2610	Wilrijk	Rob Paredis	rob@infortica.be	Zaakvoerder
				Janne Hermans	janna@infortica.be	Medewerker

Hier is een conflict met stelling 01

- indien de sorteervolgorde van de rijen wijzigt zal er informatie verloren gaan.

B° zie relational database 1.xls

Company	street	zip	city	contact	EmailAddress	function
Infortica	Kleine Doornstraat	2610	Wilrijk	Rob Paredis	rob@infortica.b	Zaakvoerder
				Janne Hermans	e janne@infortica. be	Medewerker

Hier is een conflict met stelling 2

op de kruising van rij1 en kolom contact, .. staan 2 waarden

- indien we bv een etiket willen drukken met gegevens van 2e persoon zal dit niet lukken.

- indien we een e-mail willen sturen, ... welk adres zal dan gebruikt worden, ...

C° zie relational database 2.xls

Company	street	zip	city	contact	EmailAddress	function
Infortica	Kleine Doornstraat	2610	Wilrijk	Rob Paredis	rob@infortica.be	Zaakvoerder
Infortica	Kleine Doornstraat	2610	Wilrijk	Janne Hermans	janne@infortica.be	Medewerker

Hier is een conflict met stelling 3

-indien de firma van adres wijzigt moeten meerdere records worden aangepast - dit leidt tot databasevervuiling en tijdsverspilling. (gebruikte schijfruimte wordt groter -- is echter geen argument meer)

D° zie relational datase 3.xls

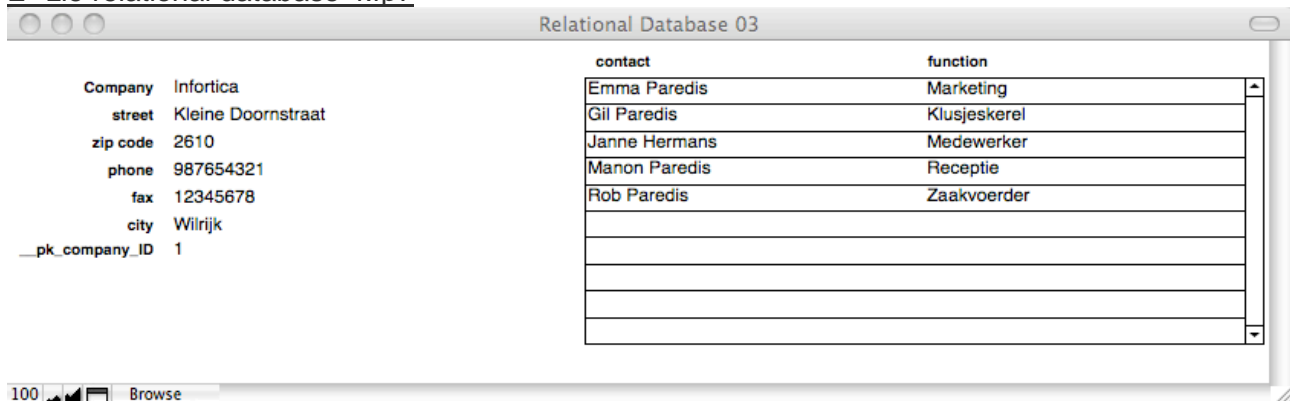
Company	street	zip	city	contact1	function1	contact2	function 2
Infortica	Kleine Doornstraat	2610	Wilrijk	Rob Paredis	Zaakvoerder	Janne Hermans	Medewerker

Hier is een conflict met stelling 4

- herhalende kolommen hebben leiden tot de volgende problemen.

- zoeken naar persoon met naam hermans moet op 2 kolommen gebeuren
- aantal contactpersonen per bedrijf is beperkt (tot 2)

E° zie relational database 4.fp7



Hier is voldaan aan alle stellingen (voor wat betreft afsplitsing bedrijven/contactenadressen)

in dit voorbeeld wordt er gebruik gemaakt van 2 tabellen (company & contact) die aan elkaar gekoppeld werden.

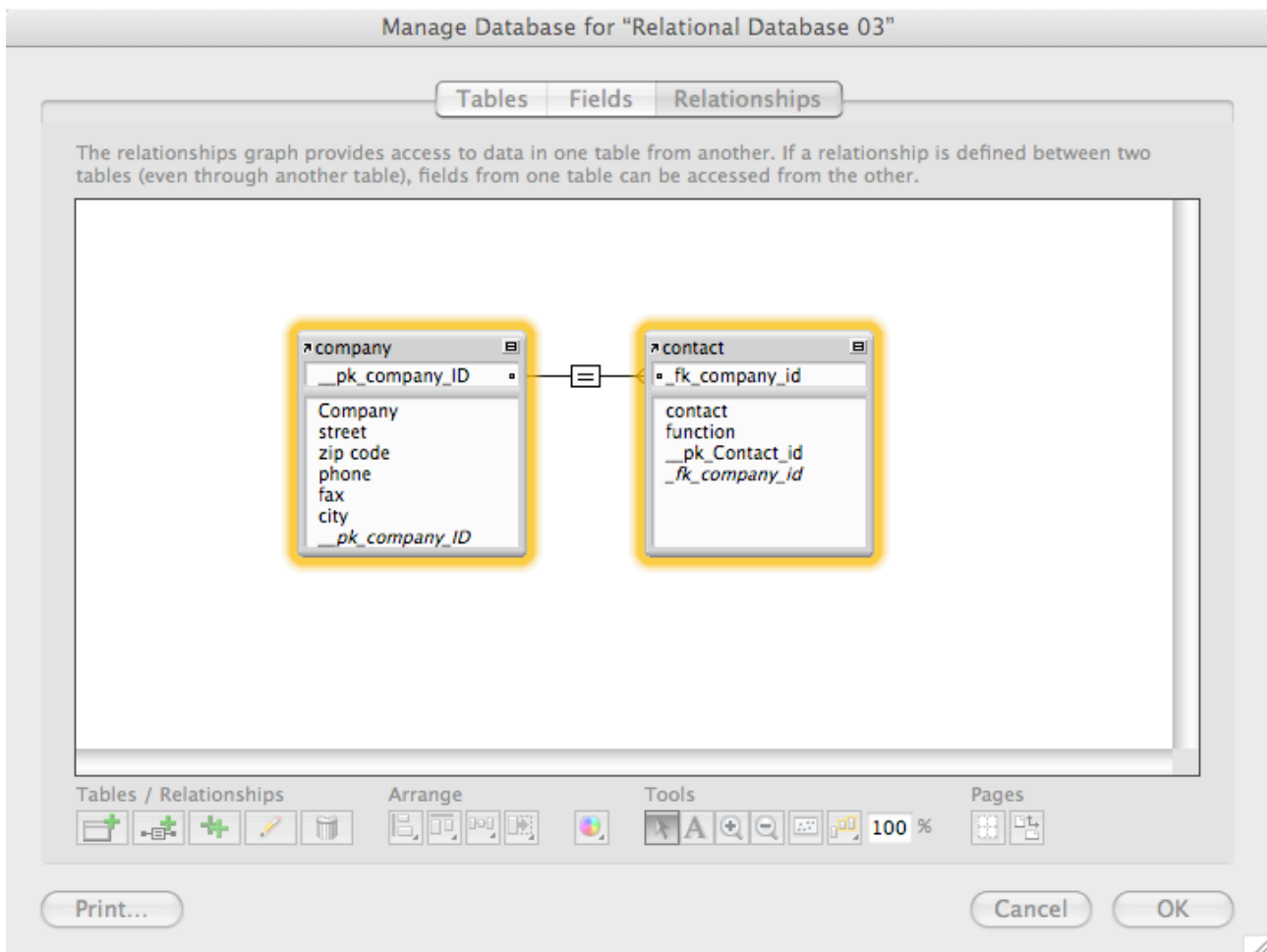
Praktijk

Om tot dit resultaat te komen moeten de volgende stappen genomen worden.

1. elke tabel moet voorzien worden van een PRIMAIRE sleutel

- a. Dit is een kolom/veld met unieke inhoud - bij voorkeur gevuld door FileMaker, niet door gebruikers
- b. de naamgeving is vrijblijvend - er wordt echter steeds vaker gebruik gemaakt van deze opbouw : __pk_tabelname_id
 - i. __ zorgt ervoor dat de primaire sleutel bovenaan staat in gesorteerde lijsten met velden.
 - ii. pk staat voor Primary Key (soms wordt ook _K1 gebruikt, ... Key One)

- iii. tablenamee zorgt voor herkenbaarheid van tabel in de naamgeving
 - iv. id duidt op het feit dat 1 en ander automatisch wordt ingevuld.
2. de tabellen worden gekoppeld in een 1 op veel relatie (1 company - veel contacts)
- a. om deze koppeling te kunnen maken moet er in de contact tabel een secundaire sleutel gemaakt worden
 - b. dit is een veld dat ingevuld moet zijn maar niet noodzakelijk door een unieke waarde
 - c. de naamgeving is vrijblijvend - er wordt echter steeds vaker gebruik gemaakt van deze opbouw : `_fk_tabelname_id`
 - i. `_` zorgt ervoor dat de secundaire sleutel bovenaan staat - NA de primaire sleutel - in gesorteerde lijsten met velden.
 - ii. `fk` staat voor Foreign Key (soms wordt ook `_K2` gebruikt, ... Key Two)
 - iii. tablenamee zorgt voor herkenbaarheid van tabel in de naamgeving
 - iv. id duidt op het feit dat 1 en ander automatisch wordt ingevuld.
3. na het aanmaken van de tabellen (meer tabellen in 1 database!) worden deze gekoppeld
- a. dit kan door pk te slepen naar de fk
 - b. een dubbelklik op de ontstane verbindinglijn geeft enkele opties weer.



Edit Relationship

A relationship defines the set of matching related records in one table for each record in another table. Select the pair of fields to be used to find matching records. To create complex relationship criteria, use additional pairs of fields.

Table: Table:

company

- __pk_company_ID
- city
- Company
- fax
- phone
- street
- zip code

=

contact

- __pk_Contact_id
- _fk_company_id**
- contact
- function

company	contact
__pk_company_ID	= _fk_company_id

company	contact
<input type="checkbox"/> Allow creation of records in this table via this relationship <input type="checkbox"/> Delete related records in this table when a record is deleted in the other table <input type="checkbox"/> Sort records <input type="button" value="Specify..."/>	<input checked="" type="checkbox"/> Allow creation of records in this table via this relationship <input type="checkbox"/> Delete related records in this table when a record is deleted in the other table <input type="checkbox"/> Sort records <input type="button" value="Specify..."/>

Om vanuit het scherm met 'companies' ook contacten te kunnen aanmaken is het belangrijk de optie 'allow creation of records via this relationship' aan te vinken aan de zijde van 'contact'