



Begrippen bij het modelleren van de gegevens

Primaire sleutel (van een tabel) = De kolom of verzameling van kolommen waardoor een record van de tabel uniek wordt geïdentificeerd (bv. een patiëntnummer identificeert uniek een bepaalde patiënt). De primaire sleutel wordt in de opsomming van de kolommen vaak onderstreept. Een sleutel die slechts één kolom bestaat, noemen we een 'enkelvoudige sleutel' en een sleutel die uit meerdere kolommen bestaat een 'samengestelde sleutel' (bv. naam, geboortedatum, geboortetijd, geslacht en adres).

Verwijzende sleutel (van een tabel) = De kolom of verzameling van kolommen van een tabel die de sleutel is van een andere tabel waar een relatie mee is (bv. de huisartscode van de patiënt waarbij de huisartscode verwijst naar de huisartstabel).

Er kan onderscheid worden gemaakt tussen een logische sleutel (voor een patiënt is dit bijvoorbeeld naam, geboortedatum, geboortetijd, geslacht, adres), een kunstmatige sleutel (voor een patiënt is dit bijvoorbeeld patiëntnummer) en de surrogaatsleutel, zoals deze alleen in de tabellen wordt gebruikt (zoals een pointer).

Relatie (tussen tabellen) = Tabellen hebben een relatie met elkaar als de ene tabel over de sleutel van de andere tabel beschikt of andersom. Tabellen die een relatie met elkaar hebben, worden weergegeven door een lijn tussen deze tabellen.

Cardinaliteit (van relaties) = De cardinaliteit geeft aan hoeveel records van de ene tabel er kunnen voorkomen bij de andere tabel waar een relatie mee is en andersom.

Er wordt onderscheid gemaakt tussen de volgende cardinaliteiten:

- a. Een-op-eenrelatie: bij één record van de ene tabel hoort één record van de andere tabel.
- b. Een-op-N-relatie: bij één record van de ene tabel horen één of meerdere records van de andere tabel.
- c. N-op-M-relatie: bij één record van de ene tabel horen één of meerdere records van de andere tabel en bij één record van de andere tabel horen één of meerdere records van de ene tabel.

De N- of M-cardinaliteit wordt door middel van een kraaienvoet weergegeven. De N-op-M-relatie dus als > ---< .

Optionaliteit (van relaties)

De relatie is optioneel als er bij een record van de ene tabel geen record van de andere tabel aanwezig hoeft te zijn. De optionaliteit kan aan beide kanten van de relatie optreden.

De optionaliteit wordt door middel van een open rondje (O) weergegeven. Zie het voorbeeld van de persoon die geen rijbewijs hoeft te hebben.

De relatie tussen twee tabellen kan worden beschreven in tekst, waarbij de tekst met de klok mee moet worden gelezen.

Een-op-eenrelatie

Een persoon kan slechts gehuwd zijn met één ander persoon (volgens de Nederlandse wet).

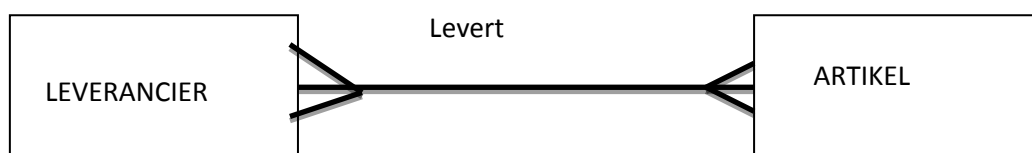
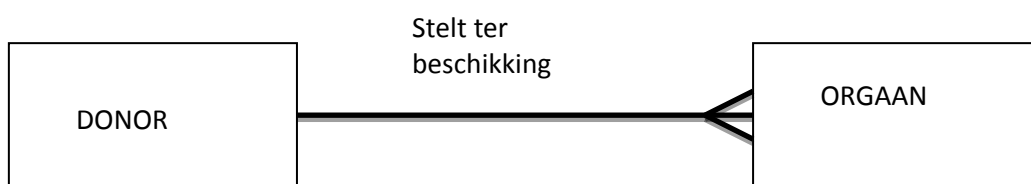
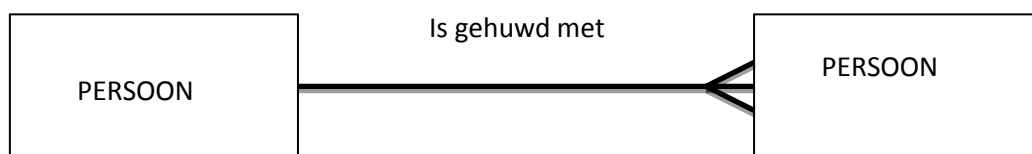
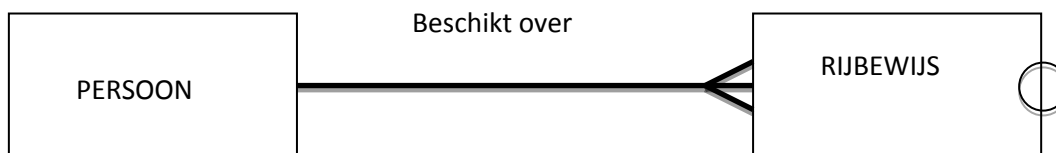
Een-op-N-relatie

Een donor stelt één of meerdere organen ter beschikking en een orgaan komt altijd van één (maar dan ook precies één) donor.

N-op-M-relatie

Een leverancier kan meerdere artikelen leveren en een artikel kan door meerdere leveranciers worden geleverd.

Al deze relaties kunnen natuurlijk in principe optioneel zijn aan beide kanten.



(Bron: OGN.)

Normalisatie

Het normaliseren van een database heeft als voordeel dat dubbele vastlegging wordt vermeden en dat het risico van het verschillend vastleggen van gegevens die eigenlijk gelijk behoren te zijn, wordt geminimaliseerd. Dit opsplitsen in kleinere tabellen is functioneel bij het invoeren, muteren en verwijderen van gegevens. Dit kan worden geïllustreerd aan de hand van het volgende voorbeeld:

Als de informatie van een leverancier en een artikel in één tabel worden vastgelegd, dan is het niet (eenvoudig) mogelijk om alleen informatie van een leverancier te registreren, terwijl nog niet bekend is welke artikelen deze levert (*insert anomalie*). Als het adres van de leverancier verandert, dan moet in alle records waar deze leverancier in wordt vermeld, het adres worden gemuteerd (*update anomalie*). Als het laatste artikel dat een leverancier nog levert, niet meer door deze leverancier wordt geleverd, dan wordt ook de informatie van de leverancier verwijderd (*delete anomalie*).

Er wordt onderscheid gemaakt in de volgende normaalvormen:

Eerste normaalvorm

Er zijn geen herhalende groepen meer in een tabel. Een tabel leveranciers met meerdere kolommen die elk aangeven welke artikelen er worden geleverd door deze leverancier, is dus niet in de nulde normaalvorm:

Leverancier = lev-code, lev-naam, lev-plaats, art-nr-1, art-naam-1, art-groepscode-1, art-groepsnaam-1, art-nr-2, art-naam-2, art-groepscode-2, art-nr-3, art-naam-3, art-groepsnaam-3, etc.

De volgende beschrijving is wel in de eerste normaalvorm:

Leverancier = lev-code, art-nr., lev-naam, lev-plaats, art-naam, art-groepscode, art-groepsnaam.

Tweede normaalvorm

Er komen geen niet-sleutelkolommen meer voor die geheel of gedeeltelijk functioneel afhankelijk zijn (dat wil zeggen dat de waarde van een kolom wordt bepaald door de waarde van een andere kolom) van de sleutel van de tabel.

De bovenstaande tabelbeschrijving is niet in de tweede normaalvorm omdat lev-naam functioneel afhankelijk is van lev-code (als de code van de leverancier bekend is, is immers ook de naam van de leverancier bekend) en art-naam, art-groepscode en art-groepsnaam functioneel afhankelijk zijn van art-nr. (als het artikelnummer bekend is, zijn immers ook de naam, artikelgroep en artikelgroepsomschrijving bekend).

De volgende (analoge) beschrijvingen zijn dus wel in de tweede normaalvorm:

- Leverancier = lev-code, lev-naam.
- Artikel = art-nr., lev-code, art-naam, art-groepscode, art-groepsnaam.

Derde normaalvorm

Er komen geen niet-sleutelkolommen meer voor die functioneel afhankelijk zijn van een ander niet-sleutelattribuut (of andere niet-sleutelattributen). De bovenstaande tabelbeschrijving is niet in de derde normaalvorm, omdat art-groepsnaam functioneel afhankelijk is van art-groepscode (als de code van de artikelgroep bekend is, is immers ook de naam van de groep bekend).

De volgende (analoge) beschrijvingen zijn dus wel in de derde normaalvorm:

- Leverancier = lev-code, lev-naam.
- Artikel = art-nr., lev-code, art-naam, art-groepscode, lev-code.
- Artikelgroep = art-groepscode, art-groepsnaam.

Andere normaalvormen

De nulde normaalvorm betekent dat het een gestructureerd bestand betreft dat in kolommen is in te delen. De Boyce Codd-normaalvorm en de vierde en vijfde normaalvorm komen vrijwel nooit voor.

Kijk voor meer informatie op de Wikipedia-pagina [Database normalization](#).

ACID

Om te garanderen dat databasetransacties betrouwbaar verlopen, moeten alle onderliggende databaseacties voldoen aan de ACID-eigenschappen:

A = Atomic (Atomair).

C = Consistent (Consistent).

I = Isolated (Geïsoleerd).

D = Durable (Duurzaam).

Bekijk de Wikipedia-pagina [ACID](#) voor een uitgebreidere beschrijving van dit onderwerp.

Per ACID-eigenschap volgt hierna een (praktijk)voorbeeld om deze te illustreren:

Atomair

Deze eigenschap betreft de mate waarin het DBMS garandeert dat een transactie ofwel geheel wordt uitgevoerd, ofwel geheel nietig is.

Een (daadwerkelijk gebeurd) voorbeeld waarbij een transactie niet atomair was, is het afschrijven van een bedrag van de rekening en het opwaarderen van het beltegoed. Door een interne storing bij de opwaarderingsdienst van de telecomprovider (we zullen geen namen noemen) werd het bedrag wel van de rekening afgeschreven, maar het beltegoed werd niet opgewaardeerd. Meerdere pogingen om op te waarderen resulteerden wel in een debitering van de rekening, maar niet in de creditering van het beltegoed.

Consistent

Een transactie creëert ofwel een nieuwe geldige staat of herstelt de staat die er was (in geval van een fout of een probleem). Dit impliceert dat na de transactie alle integriteitsregels van de database moeten gelden.

Een (daadwerkelijk gebeurd) voorbeeld waarbij een transactie niet consistent was, betreft het boeken van een vliegticket met hotel in Mallorca dat in twee verschillende stappen gebeurt. Er wordt in eerste instantie een leuk hotel gevonden in de hoofdstad van Mallorca (Palma) en vervolgens wordt de vliegticket geboekt. Als de reis daadwerkelijk wordt ondernomen, blijkt dat het vliegticket landt in La Palma (Canarisch eiland) en dat het hotel dat werd geboekt zich in Palma (Mallorca) bevond. Er is dus een niet-consistente reis geboekt: tijdens deze geïntegreerde boeking had deze verkeerde combinatieboeking dus moeten worden geweigerd of de klant had tijdens het boeken moeten worden geholpen om het land/eiland van de bestemming van de vliegticket gelijk te kiezen aan het land/eiland van het hotel.

Geïsoleerd

Transacties worden geïsoleerd van elkaar uitgevoerd, dat wil zeggen dat transacties die tegelijkertijd worden uitgevoerd, geen inzicht hebben in elkaars tussenresultaten.

Een voorbeeld van een transactie die niet geïsoleerd wordt uitgevoerd, is dat twee personen (vrijwel) gelijktijdig een stoel boeken in een vliegticket, waardoor er twee passagiers zijn die dezelfde stoel hebben geboekt en daarop komen te zitten.

Duurzaam

Een transactie is niet duurzaam als een voltooide transactie later niet ongeldig kan worden gemaakt. Het resultaat moet worden opgeslagen, moet beschikbaar zijn en het moet kunnen worden hersteld.

Een voorbeeld hiervan is dat er bij een belastingdienst (we zullen wederom niet melden welke) 175.000 belastingaangiftes moesten worden bewerkt. Deze bewerking werd echter verkeerd gedaan, waardoor de belastingaangiftes niet meer te gebruiken waren. Het was handig geweest om voor deze (desastreus verlopende) bewerking een back-up te maken.