

Goed nadenken waarom we vastleggen

# Omgaan met de geldigheid van gegevens

Marion Verwey

**De begrippen 'tijdslijnen' of 'geldigheid van gegevens' zijn regelmatig terugkerende items bij de ontwikkeling van applicaties en dan met name bij het opstellen van het logisch gegevensmodel. In de literatuur is hierover vanuit technisch oogpunt veel terug te vinden, maar vanuit functioneel oogpunt erg weinig.**

Deze functionele kant van de geldigheid van gegevens is vaak onderbelicht. Er wordt niet stil gestaan bij de vraag waarom we de geldigheid vastleggen en waarmee we functioneel rekening moeten houden. Ook moeten we ons afvragen voor welke gegevens we een geldigheidsperiode willen vastleggen. In de praktijk ziet men dat transactiedata veelal geen geldigheidsperiode kennen, terwijl masterdata en referentiegegevens deze veelal wel hebben. Een product zal bijvoorbeeld wel vaak een geldigheidsperiode hebben, maar een order bijvoorbeeld niet.

## Er kan verschil zitten tussen de onderlinge samenhang van de tijdslijnen

Dit artikel gaat vooral in op de functionele aspecten die komen kijken bij het introduceren van geldigheid van gegevens. Veel zal verklaard worden aan de hand van een voorbeeld met een eenvoudige situatie: over meneer X is geregistreerd dat hij vanaf 1 januari 2000 op adres A woont en dat hij op 1 augustus 2006 verhuist naar adres B.

Om de in dit artikel gebruikte begrippen helder te krijgen volgt hieronder een aantal definities.

**Geldigheidsperiode.** Een geldigheidsperiode is de periode waarbinnen (een groep van) gegevens geldig is. Dit wordt aangegeven door deze (groep van) gegevens te voorzien van een begin- en einddatum.

**Transactietijd.** Het tijdstip waarop de gegevens in een database opgeslagen zijn.

**Historie.** Een geldigheidsperiode die in het verleden geldig was.

Met andere woorden een geldigheidsperiode met een einddatum in het verleden. In ons voorbeeld adres A van meneer X.

**Versiehistorie.** De vorige waarde van een gegeven.

**Invoeren.** Onder invoeren wordt verstaan; het toevoegen van een nieuw gegeven of een nieuwe geldigheidsperiode. In het voorbeeld is het vastleggen van adres B van meneer X een invoer. Natuurlijk wordt onder invoeren ook het vastleggen in het systeem van een nieuwe meneer Y bedoeld.

**Wijzigen.** Met wijzigen wordt bedoeld het wijzigen binnen een al bestaande periode van een gegeven. Stel dat we in ons voorbeeld hadden vastgelegd dat meneer X vanaf 1 augustus op huisnummer 10 woont en dat dit later huisnummer 10a blijkt te zijn. Het veranderen van huisnummer 10 naar huisnummer 10a is dan een wijziging, het feit dat meneer X op 10a woont is namelijk geldig vanaf 1 augustus. Het betreft hier een registratieve fout die wordt gecorrigeerd.

## Geldigheid van gegevens

Als we naar de tijdslijnen van gegevens behorende bij een hoofdobject kijken, kunnen we concluderen dat er een verschil kan zitten tussen de onderlinge samenhang van de tijdslijnen van deze gegevensgroepen afzonderlijk. Als we bijvoorbeeld naar een natuurlijk persoon kijken, dan zien we dat deze slechts een woonadres tegelijkertijd kan hebben, maar op enig moment in de tijd wel meerdere nationaliteiten kan hebben.

Als we op deze manier naar de geldigheid van gegevens binnen een gegevensgroep kijken, kunnen we drie soorten van samenhang onderkennen:

1. Overlappend: Meerdere voorkomens kunnen tegelijkertijd geldig zijn. Bijvoorbeeld, een persoon kan op een tijdstip meerdere nationaliteiten hebben;
2. Sequentieel, op ieder tijdstip is altijd maar een voorkomen geldig. Bijvoorbeeld, een persoon kan maar op een adres

tegelijkertijd wonen. Echter in de tijd gezien kan een persoon op meerdere adressen gewoond hebben;

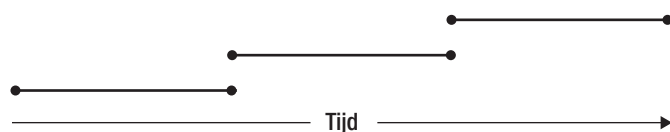
3. Met gaten, er zijn momenten waarop geen enkel voorkomt geldig is. Bijvoorbeeld, een persoon kan gedurende een periode werkeloos zijn, maar gedurende een andere periode meerdere banen tegelijkertijd hebben.

Bovenstaande soorten van samenhang kunnen grafisch op onderstaande wijze worden weergegeven:

1. Overlappend.



2. Sequentieel.



3. Met gaten (deze oplossing kan zowel in combinatie met 1 als met 2 voorkomen).



## Functionele voorwaarden

Voor iedere gegevensgroep waarvoor een geldigheidsperiode moet worden geregistreerd, moet men over de volgende vier aspecten nadenken en afspraken maken:

1. Is de einddatum een 'tot' of een 'tot en met' datum. Het is raadzaam om deze afspraak over het gehele systeem gelijk te houden;
2. Zijn de begin- en einddatums alleen een zuivere datumregistratie (jaar, maand, dag), of moet ook een tijdselement (uren, minuten, seconden) worden meegenomen. Gezien het feit dat de geldigheid vaak een functionele betekenis heeft, kunnen deze datums veelal zonder tijdselement worden geregistreerd. Als men besluit dat bij de registratie van een geldigheid toch het tijdselement moet worden meegenomen, dan moet men afspraken maken over wat het resultaat is van de vraag "wat is geldig op een peildatum". Stel namelijk dat we in ons voorbeeld van een adreswijziging registreren dat adres B van meneer X ingaat op 1 augustus om 17.00 uur. Bij de vraag "wat is het adres van meneer X op 1 augustus?" zal het systeem twee antwoorden opleveren. Op 1 augustus woont meneer X zowel op adres A als op adres B. Pas als we vragen naar het adres op een specifiek tijdstip krijgen we één antwoord;
3. Tot welke van de beschreven drie soorten de gegevensgroepen onderling behoren;

4. Is een einddatum functioneel nodig. Indien sprake is van sequentiële tijdslijnen zonder gaten, dan is de begindatum van een nieuwe periode automatisch de einddatum van de vorige periode.

De zaken waarmee we rekening moeten houden bij het toevoegen, wijzigen en verwijderen van gegevens met betrekking tot de geldigheidsperiodes worden verder uitgewerkt.

## Toevoegen

Alleen bij de 'sequentiële' tijdslijnen (2e soort samenhang) moeten we functionele afspraken maken. In dit geval is het noodzakelijk om te bepalen wat moet gebeuren als binnen de geldigheidsperiode van de nieuwe tijdslijn ook nog een oude tijdslijn geldig is. Met andere woorden, wat moet er gebeuren als er sprake is van een overlap in de geldigheidsperiode. Er zijn nu twee mogelijke (geautomatiseerde) oplossingen:

1. Weiger de nieuwe geldigheidsperiode, omdat er al een tijdslijn voorkomt binnen deze periode;
2. Pas de oude tijdslijn zodanig aan dat er geen overlap meer met de nieuwe tijdslijn bestaat.

Hieronder volgen twee voorbeelden voor het adres van meneer X. In het eerste voorbeeld gaan we uit van een verhuizing, dus het feit dat de nieuwe tijdslijn in principe achter de oude tijdslijn kan worden geplakt. Het tweede voorbeeld is ingewikkelder. In dit voorbeeld gaan we er vanuit dat de nieuwe tijdslijn geheel binnen een bestaande tijdslijn valt.

Als we kijken naar ons voorbeeld van het adres van meneer X: op het moment dat de adreswijziging naar adres B binnenkomt, dienen we de geldigheid van adres A af te sluiten. Aangezien deze actie evident is, kan het beëindigen van het oude adres automatisch plaats vinden.

Als we nogmaals kijken naar het adres van meneer X. Stel we hebben netjes in opeenvolgende periodes adres A en adres B geregistreerd. Vervolgens krijgen we de melding dat het adres van meneer X van 1 september 2006 tot 1 november 2006 adres C is. De vraag die nu rijst is wat moet er met het adres B gebeuren? Aangezien men niet op twee adressen tegelijkertijd kan wonen kan dit bericht twee betekenissen hebben. Namelijk adres B moet per 31 augustus 2006 worden beëindigd, adres C is tot 1 november 2006 geldig en adres B is weer vanaf 1 november 2006 geldig. Echter het kan ook betekenen dat het na 1 november 2006 niet duidelijk is waar meneer X woont. Omdat een dergelijk bericht op meerdere manieren kan worden geïnterpreteerd is het niet vanzelfsprekend dat deze wijziging volledig geautomatiseerd kan worden doorgevoerd.

## Wijzigen en corrigeren

Omdat bij gebruik van tijdslijnen de begindatum vaak een onderdeel is van de functionele sleutel, dienen we onderscheid te maken tussen twee vormen van wijzigen:

1. Wijzigen van gegevens binnen een bestaande geldigheidsperiode;
2. Wijzigen van de geldigheidsperiode zelf.

In het eerste geval is de functionele sleutel van de gegevens die moeten worden gewijzigd bekend en zal het wijzigen (corrigeren) geen problemen opleveren.

De tweede soort wijziging kan problemen geven. We gaan namelijk een functionele sleutel wijzigen. Bij dit soort wijzigingen dienen we ook de punten die bij het invoeren genoemd zijn in acht te nemen.

## Rekening dient te worden gehouden met de complexiteit van geldigheid van master/detail-relaties

We gaan in ons voorbeeld weer uit van het adres van meneer X. Als we de melding krijgen dat er een foutje zit in zijn huidige adres (dat is dus adres C) dan kunnen we dat probleemloos verwerken. Hij blijkt bijvoorbeeld niet op nummer 10 te wonen maar op nummer 10c. Echter als de melding aangeeft dat hij met ingang van 1 september op het adres met nummer 10c woont, dan kan dit betekenen dat een wijziging van het huisnummer heeft plaatsgevonden (omdat bijvoorbeeld de huizen in zijn straat hernummerd zijn), of dat hij op 1 september van het huis op nummer 10 naar het huis op nummer 10c is verhuisd.

### Verwijderen

Net zoals bij het toevoegen zijn het de sequentiële tijdslijnen die enige extra aandacht behoeven bij het verwijderen. Er mogen namelijk geen gaten in de tijdslijn ontstaan en dus moet bij het verwijderen van een tijdslijn worden aangegeven wat er met de begin- en einddatum van de andere tijdslijnen moet gebeuren.

We kijken opnieuw naar het adres van meneer X en naast zijn adressen A en B is tevens geregistreerd dat hij vanaf 1 oktober 2006 op adres C woont. Stel dat de tijdslijn van adres B wordt verwijderd, dan moet de vraag worden gesteld waar meneer X van 1 augustus 2006 tot 1 oktober 2006 woonde.

Het verwijderen van adres B kan een drietal betekenissen hebben. Van 1 augustus tot 1 oktober woont meneer X op een voor ons onbekend adres of adres A is geldig tot 1 oktober of adres C is geldig vanaf 1 augustus. Welke van deze drie mogelijkheden geldt moet dus door een gebruiker worden bepaald.

### Raadplegen van tijdslijnen

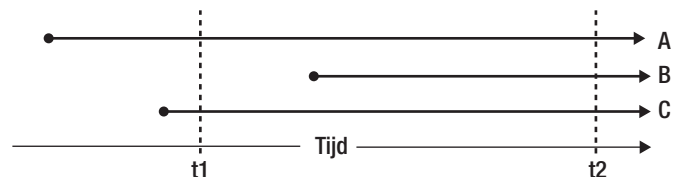
Het opvragen van gegevens kan heel complex worden, vooral als we ook rekening willen gaan houden met de transactietijd (versiehistorie) van gegevens – opvragen van versiehistorie wordt hier verder buiten beschouwing gehouden. Er kunnen vier manieren worden onderkend om gegevens op te vragen:

1. Wat is nu geldig (dit is ook de meest gebruikte manier van opvragen);
2. Wat was geldig op een datum in het verleden;

3. Wat is geldig sinds een datum in het verleden (manier 3 is een specifieke invulling van de hiergenoemde algemene manier);
4. Wat is geldig in een bepaalde periode (van deze manier van opvragen zal minimaal gebruik gemaakt worden.

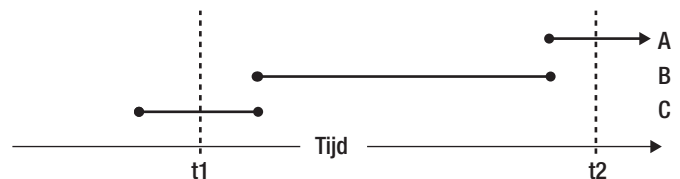
Hieronder wordt aan de hand van de eerder getoonde schema's een voorbeeld gegeven van de bovenstaande manieren van opvragen. In dit voorbeeld stelt  $t_1$  op de tijdslijn een datum in het verleden voor en  $t_2$  de datum nu.

Overlappend.



1. Deze vraag levert als antwoord A, B en C.
2. Deze vraag levert als antwoord A en C.
3. Deze vraag levert als antwoord A, B en C.
4. Deze vraag levert als antwoord A, B en C.

Sequentieel.



1. Deze vraag levert als antwoord A op.
2. Deze vraag levert als antwoord C op.
3. Deze vraag levert als antwoord A, B en C op.
4. Deze vraag levert als antwoord A, B en C op.

Met gaten.



1. Deze vraag levert als antwoord A op.
2. Deze vraag levert als antwoord C op.
3. Deze vraag levert als antwoord A en C op.
4. Deze vraag levert als antwoord A en C op.

### Tijdslijnen en master/detail-relaties

Behalve de hierboven beschreven problematiek wat betreft de geldigheid binnen één entiteit, dient ook rekening te worden gehouden met de complexiteit van geldigheid van master/detail-relaties. Of met andere woorden, is een adres nog geldig als de bijbehorende persoon is overleden?

Voor het oplossen van deze problematiek bestaan drie mogelijkheden:

1. Bij het zoeken naar een geldig detail dienen we ook altijd te kijken of de bijbehorende parent ook geldig is;
2. Bij het afsluiten (vullen van datum einde) van de master dienen we ook altijd de bijbehorende details (automatisch) af te sluiten;
3. Er is geen eenduidige regel. In dit geval dienen we bij het opvragen van gegevens aan te geven of we moeten controleren op de geldigheid van de details en bij het beëindigen van gegevens altijd aan te geven of we de details ook moeten beëindigen.

Als voorbeeld nemen we weer meneer X. Stel meneer X overlijdt op 1 december en het enige wat we registreren is zijn overlijdensdatum. We beëindigen niet zijn adres.

Als we dan gaan zoeken naar het adres van meneer X op 5 december en we kijken alleen naar het adres, dan kan de indruk worden gewekt dat meneer X op 5 december nog steeds op adres B woont. Daarom moet in dit geval ook altijd naar de geldigheid van de parent (in dit geval de overlijdensdatum van meneer X) worden gekeken.

Een andere mogelijkheid is om als meneer X overlijdt ook met een zijn adres te beëindigen, echter als zijn vrouw op hetzelfde adres geregistreerd staat en het adres wordt beëindigd is mevrouw X opeens adresloos. Deze situatie is in dit geval dan niet wenselijk. Vandaar dat we functioneel, voor iedere parent/child-relatie met geldigheidsperioden moeten bekijken, wat met de child moet gebeuren op het moment dat de parent wordt beëindigd.

## Versiehistorie

Hiervoor is beschreven hoe we kunnen omgaan met de geldigheid van gegevens. Met andere woorden; wat is geldig op een specifiek tijdstip. Voor sommige situaties is het kunnen beantwoorden van deze vraag niet voldoende. In deze gevallen dient naast de vraag "wat denken we nu dat vorige maand geldig was", ook de vraag beantwoord te worden "wat dachten we vorige week, dat vorige maand geldig was?" Deze vraag zou een ander antwoord op kunnen leveren. Om een en ander duidelijk te maken nemen we als voorbeeld weer meneer X met de adressen A en B. Stel dat we op 4 augustus de melding krijgen dat meneer X al vanaf 15 juli (in plaats van 1 augustus) op adres B woont. Nu stellen we 5 augustus de vraag: waar woonde meneer X op 21 juli? Als antwoord krijgen we dan adres B. Hadden we deze vraag echter op 3 augustus gesteld dan hadden we als antwoord adres A gekregen. Als we op 5 augustus vragen "waar dachten we 3 augustus dat meneer X woont op 21 juli", dan krijgen we als antwoord adres A.

## Conclusies

Bij het ontwikkelen van een systeem waarin 'tijdslijnen' of 'geldigheid van gegevens' moet worden geregistreerd, moet men

zich bewust zijn van het feit dat op functioneel niveau goed moet worden nagedacht wat we met deze 'geldigheid van gegevens' willen bewerkstelligen. De impact van deze eis is niet, zoals het nu vaak wordt benaderd, puur een technische aangelegenheid.

We moeten ons terdege afvragen wat we hiermee willen. De aspecten waarmee we functioneel rekening moeten houden is over wat voor een soort tijdslijn (sequentieel, overlappend of met gaten) we spreken en hoe we in deze gevallen omgaan met invoeren, wijzigen, verwijderen en raadplegen van gegevens. Het registreren van 'tijdslijnen' of 'geldigheid van gegevens' heeft alleen toegevoegde waarde als er voor een object ook een functionele reden is voor het registreren hiervan. Het registreren om het registreren is vanwege de mogelijke functionele onduidelijkheid en technische complexiteit af te raden.

## Literatuur

1. R. Veldwijk, *Tien Geboden voor een beter database-ontwerp*. Database Magazine 3/1995.
2. R. Snoddgrass, *Developing Time-Oriented Database Applications in SQL*. Morgan Kaufmann Publishers, Inc., San Francisco, July 1999.
3. M. Chisholm, *Eindelijk grote belangstelling voor Master Data*. Database Magazine 4/2006.

## Marion Verwey

Drs. M. Verwey (marion.verwey@capgemini.com) is Informatie analist bij Capgemini.

## Online archief Database Magazine

Online archief

Online archief

Trefwoorden:  zoek Zoektips

U bent op dit moment niet ingelogd. [\[inloggen\]](#)

Extra zoekcriteria:

Database Magazine

Alle magazines

Zoek in:

Alle velden

Titel

Auteur

Omschrijving

Jaar

Bladnummer

Datum:

van: Januari 1993

tot: Maart 2007

Aantal artikelen per pagina:

25

[Array Publications ©](#) | [disclaimer](#) | [privacy statement](#)

Database Magazine-lezer opgelet! Artikelen over onderwerpen als Datawarehousing, SQL, ETL, Business Intelligence, Relationale databases, modellering en nog veel meer vindt u in het Online Archief van Array Publications. Vaktijdschriften als Storage Magazine, Database Magazine, IT Service Magazine, Java Magazine en ons Oracle vakblad Optimize hebben hun artikelenarchief online gezet. Met een Google-achtige zoekstructuur vindt u snel wat u zoekt op [www.dbm.nl](http://www.dbm.nl)