

Normaliseren

In deze appendix wordt bekeken wat er moet gebeuren voordat een systeem kan worden gedefinieerd. Dit begint met een analyse van de gegevens die de basis vormen. Daarbij wordt gekeken naar het opsplitsen van de gegevens, zodat deze kunnen worden vastgelegd in verschillende tabellen. Om de gegevens uit de verschillende tabellen te kunnen koppelen, worden relaties tussen de gegevens in de tabellen gelegd. Het proces waarbij de gegevens worden gesplitst, wordt normaliseren genoemd.

Relaties

De afkorting DBMS staat voor databasemanagementsysteem, vaak wordt er nog de letter R voorgezet en die staat voor relationeel. Dat is ook de kern van het systeem. Er moet een relatie zijn tussen de gegevens die worden vastgelegd. Meestal is die relatie natuurlijk, een product wordt door een bepaalde fabrikant gemaakt, een familie woont op een bepaald adres of een bepaalde auto heeft één kenteken. Maar het komt ook voor dat er niet direct een bepaalde relatie is, deze wordt dan afgedwongen, bijvoorbeeld door een klant van een postorderbedrijf een klantnummer toe te kennen. Het kan zijn dat er één gegeven wordt gebruikt, zoals het klantnummer of een kenteken, maar het kan ook een combinatie zijn van bijvoorbeeld postcode en huisnummer. Gegevens die gebruikt kunnen worden om verschillende tabellen te koppelen, worden sleutelvelden genoemd. Het symbool van de sleutel wordt zowel in het pictogram van Acces als bij indexvelden gebruikt.

Functies

Om met gekoppelde tabellen te kunnen werken, moeten functies beschikbaar zijn. Denk hierbij aan het definiëren van de database, het invoeren, wijzigen en verwijderen van gegevens die in de verschillende tabellen zijn vastgelegd en het op verschillende manieren presenteren van de gegevens. Bij de uitvoer kan worden gedacht aan het genereren van lijsten met gegevens op het scherm, publiceren op internet of op papier. Dit zijn voor de hand liggende functies van het systeem.

Minder herkenbaar zijn functies als het beveiligen van de gegevens tegen gebruik en misbruik door onbevoegden. Dit wordt meestal geregeld door in het systeem vast te leggen wat de rechten van gebruikers zijn. Sommigen hebben alleen inzage, anderen mogen gegevens ook wijzigen en weer anderen mogen ook gegevens verwijderen. Dit beveiligen gebeurt op allerlei niveaus. Het zal duidelijk zijn dat klanten of concurrenten geen toegang kunnen krijgen tot de prijzen van artikelen die zij kunnen bestellen, maar wel hun eigen adresgegevens moeten kunnen wijzigen. Het aanpassen van de structuur van de database is een nog hoger niveau en zal alleen door de beheerders mogen worden gedaan.

Vaak wordt gebruikers van computers aangeraden back-ups te maken van de bestanden. Ook dat behoort tot de taken van een DBMS. Het moet mogelijk zijn regelmatig een back-up te maken van zowel de gegevens als de eromheen gebouwde bestanden, bijvoorbeeld voor het genereren van rapporten, en deze weer terug te zetten.

Wordt een database gebruikt voor bijvoorbeeld een webwinkel, dan moeten alleen bestellingen kunnen worden gedaan van artikelen die ook daadwerkelijk worden verkocht. Dat lijkt een open deur, maar er wordt mee bedoeld dat alleen artikelen die voorkomen in de database (herkenbaar aan een uniek artikelnummer) kunnen worden besteld. Dit betekent dat in de tabel waarin de bestellingen worden bijgehouden alleen artikelnummers mogen voorkomen die ook voorkomen in de tabel met de artikelen. Er moet dus een controle zijn op consistentie van de database.

Een andere controle is die op de invoer. Er moet gecontroleerd worden of de invoer wel voldoet aan de regels. Denk bijvoorbeeld aan de postcode van vier cijfers, een spatie en twee hoofdletters. Het formulier waarmee de gegevens moeten worden ingevoerd moet ook zo duidelijk zijn dat de gebruiker weet hoe de postcode moet worden ingevoerd. Soms moet de spatie getypt worden, in andere formulieren juist niet.

Een ander aspect van invoercontrole is de controle door een andere persoon. Fouten maken blijft een menselijke eigenschap en de kans dat een fout wordt opgemerkt is groter als meer mensen de gegevensinvoer controleren. Ook het gebruik van speciale formulieren voor verschillende gegevensinvoer helpt bij het voorkomen van fouten. Denk aan een gemeentelijke administratie. Als voor nieuwgeborenen een ander formulier wordt gebruikt dan voor nieuwe inwoners, kan beter worden gecontroleerd of een geboortedatum logischerwijs klopt (en is de kans dat een baby bij aanmelding een verkeerd geboortjaar krijgt een stuk kleiner). Deze controle valt onder het begrip integriteit, de controle op de juistheid van de gegevens.

De zin van splitsen

Waarom wordt vaak de moeite genomen om gegevens op te splitsen en op te slaan in verschillende tabellen? Het is helemaal niet zo prettig om met klantnummers te werken. Vaak weten de klanten het nummer niet terug te vinden of moeten pasjes worden gemaakt waarop dit nummer staat. Zeker in kleinere organisaties bestaat nogal eens de gewoonte om alle gegevens in één grote tabel op te slaan, dat is een stuk eenvoudiger. Nadenken over de structuur van een tabel en dan ook nog over het soort gegevens dat in de tabel wordt vastgelegd, is niet nodig. Echter, na verloop van tijd ontstaat het risico dat dezelfde gegevens op verschillende plaatsen worden vastgelegd en dat een wijziging van de gegevens niet overal wordt doorgevoerd. Twee termen beschrijven deze situatie:

- **Redundantie** Dezelfde gegevens worden meer dan eens vastgelegd.
- **Inconsistentie** Gegevens zijn niet meer met elkaar in overeenstemming.

Normaliseren

In 1970 heeft E.F. Codd (overleden in 2003) in een artikel met de titel “A relational model of data for large shared databanks”, een beschrijving van het relationele model gepubliceerd. In 1985 heeft hij in twaalf regels vastgelegd hoe een ideale database eruit moet zien, maar voegde daaraan toe dat geen enkel systeem aan alle twaalf regels tegelijk zal voldoen.

In hetzelfde stuk is ook de opzet beschreven om informatie zo te analyseren dat tabellen kunnen worden gemaakt die het ideaal zo dicht mogelijk benaderen. Dit proces wordt normaliseren genoemd. Hierbij wordt in vier stappen bekeken hoe de gegevens zo kunnen worden geordend dat er een logische structuur ontstaat waarmee de tabellen van de database kunnen worden gedefinieerd. Bij het normaliseren wordt uitgegaan van de praktijk, bijvoorbeeld een factuur voor een aantal gekochte producten.

Nulde normaalvorm

Op de factuur komt een aantal vaste gegevens voor, maar ook variabele gegevens zoals de producten die zijn aangekocht. Bekeken vanuit het bedrijf dat de goederen heeft verkocht is de naam van het bedrijf een vast gegeven, dit wordt niet opgenomen in de database. Evenmin worden adres en woonplaats van het verkopende bedrijf in de database opgenomen.



Filialen

Gaat het om een winkelketen met filialen die allemaal één verkoopsysteem gebruiken, dan is het uiteraard wel de bedoeling om op te nemen in welk filiaal de artikelen zijn verkocht. Dit geeft meteen de kracht van een relationeel systeem aan.

Daarnaast zijn er ook berekeningen te ontdekken, bijvoorbeeld de totaalprijs per artikel wanneer van één artikel meer exemplaren zijn gekocht, de btw-berekeningen (eventueel per artikel) voor de totale factuur en natuurlijk het totaalbedrag. Dergelijke gegevens worden aangeduid met de term procesgegevens en worden niet opgenomen in de database. Maar bij facturen waarbij sprake kan zijn van twee of zelfs drie verschillende btw-tarieven moet wel een btw-code worden opgenomen.



Code

Meestal wordt niet het echte percentage 0%, 6% of 21% opgenomen, maar een code, bijvoorbeeld 0, 1 en 2. Aan deze codes wordt het echte percentage gekoppeld. Wordt een wijziging van het btw-percentage ingevoerd, dan hoeft niet bij elk artikel het percentage te worden gewijzigd. Het spreekt voor zich dat ook hier een aparte tabel voor wordt gemaakt.

De berekening zal door een instructie bij het genereren van een rapport worden uitgevoerd. Door de verschillende elementaire gegevens die relevant zijn te noteren, ontstaat de zogenoemde nulde normaalvorm.

Voor de factuur kan worden gedacht aan de volgende gegevens: datum, factuurnummer, klantnummer, klantnaam, btw-percentage, artikelnummer, aantal van artikel, artikelnaam, artikelprijs. De laatste vier gegevens komen op iedere regel voor een gekocht artikel terug. Dat wordt een repeterende groep (*repeating group*) genoemd. Om dit duidelijk aan te geven, wordt vaak de volgende notatie gebruikt: datum, factuurnummer, klantnummer, klantnaam, btw-percentage RG(artikelnummer, aantal van artikel, artikelnaam, artikelprijs en btw-code). Hierbij is factuurnummer onderstreept, het is de sleutel tot de gegevens van deze factuur.

Eerste normaalvorm

De artikelen worden door meer klanten gekocht en het kan voorkomen dat dezelfde klant hetzelfde artikel op dezelfde dag nog een keer aanschaft. Door deze gegevens te koppelen aan het factuurnummer, dat immers bij iedere nieuwe factuur anders en uniek is, kan worden vastgelegd bij welke factuur de bestelling hoort. Dit betekent dat de gegevens van de artikelen (die zich in de repeterende groep bevinden) moeten worden gekoppeld aan het factuurnummer. De notatie voor dit deel van de database wordt dan: factuurnummer, artikelnummer, aantal van artikel, artikelnaam, artikelprijs en btw-code. Nu zijn het factuurnummer en het artikelnummer onderstreept om aan te geven dat daarmee het gekochte artikel op die factuur is terug te vinden. Dit wordt een samengestelde sleutel genoemd. Alleen het factuurnummer instellen als sleutel is nu niet genoeg, omdat daarmee meer factuurregels worden gevonden. Het is juist de bedoeling om met een sleutel een uniek gegeven terug te vinden.

In de notatie voor de eerste normaalvorm wordt de repeterende groep met de aangepaste sleutel apart genoteerd.

Eerste normaalvorm

factuurnummer, datum, klantnummer, klantnaam, btw-percentage
factuurnummer, artikelnummer, aantal van artikel, artikelnaam, artikelprijs en
btw-code

Tweede normaalvorm

De volgende stap is het losmaken van gegevens die niet van de hele sleutel afhankelijk zijn. De gegevens aantal van artikel, artikelnaam en artikelprijs hangen niet van het factuurnummer af, maar alleen van het artikelnummer. Deze gegevens worden nu apart opgenomen, met de sleutel.

artikelnummer, artikelnaam, artikelprijs en btw-code

Tweede normaalvorm

factuurnummer, datum, klantnummer, klantnaam, btw-percentage
factuurnummer, artikelnummer, aantal van artikel
artikelnummer, artikelnaam, artikelprijs en btw-code

Derde normaalvorm

Het btw-percentage hangt niet van het factuurnummer af en het heeft evenmin iets te maken met de verschillende artikelen. Er moet een aparte tabel voor worden gemaakt. Hierin komen maar twee velden voor: btw-code en btw-percentage. De sleutel is btw-code. Dit betekent dat in de eerste opsomming btw-percentage kan verdwijnen.

Derde normaalvorm

factuurnummer, datum, klantnummer, klantnaam
factuurnummer, artikelnummer, aantal van artikel
artikelnummer, artikelnaam, artikelprijs en btw-code
btw-code, btw-percentage

Deze uitsplitsing levert voldoende mogelijkheden op om met nette tabellen te werken.