Kwaliteit van de gegevens begint bij het begin (2)

Mogelijkheden voor verbetering van kwaliteit

Toon Loonen

Datakwaliteit begint bij de eerste invoer van gegevens of eerder bij het opstellen van het gegevensmodel en het beschrijven en bouwen van de invoercontroles. In dit laatste van een tweetal artikelen wordt een overzicht gegeven van de mogelijkheden om kwaliteit in de gegevens te verankeren.

Door default-waarden te definiëren voor een gegeven kan invoer sneller en eenvoudiger worden, waarmee ook invoerfouten kunnen worden vermeden. Wel bestaat het gevaar dat te gemakkelijk deze default-waarde wordt overgenomen. De default-waarden kunnen hard gecodeerd zijn (systeemdatum voor de orderdatum) of uit een tabel komen (een gegenereerd volgnummer, het percentage korting dat afhankelijk is van de eerdere afname door deze klant).
Daarnaast kan het systeem soms geautomatiseerd zelf correcties uitvoeren, bijvoorbeeld bij het hoofdletter gebruik in naam- en adresvelden. Een alternatief is het geven van een waarschuwing als een gegeven (naam McDonald) niet aan een verwachte controle voldoet of bij een overschrijding van een aantal of bedrag. Hierna kan de gebruiker toch bewust besluiten de ingevoerde gegevens op te slaan.

Flexibele controles

Bij de meeste systemen worden de invoercontroles hard gecodeerd in de database en/of de applicatie. Als deze controles regelmatig (kunnen) wijzigen moeten ze zo gebouwd worden dat wijziging van de controle mogelijk is zonder wijziging in de programmatuur. Bij een controle op bijvoorbeeld het laagste bedrag in een girotel-transactie (zie kader) zou dit bedrag in een tabel met systeem-constanten opgenomen kunnen worden. Bij het wijzigen van deze waarde behoeft niet de programmatuur te worden aangepast, maar alleen de betreffende waarde in de systeemtabel.
Door codes op te nemen in een codetabel (in plaats van hard coderen) kan gemakkelijk een nieuwe code toegevoegd worden. Zo'n tabel bevat de kolommen Codetype, Code en Omschrijving, zie de tabel links onderaan deze pagina.
Johan van der Graaf beschrijft hoe invoercontroles en andere regels in een Rule Engine kunnen worden vastgelegd [8].
Op deze wijze kunnen zeer flexibel nieuwe invoercontroles (en berekeningen, etcetera) worden vastgelegd of bestaande controles worden gewijzigd.

Codetype	Code	Omschrijving
Geslacht	M	Man
Geslacht	V	Vrouw
Klantsoort	P	Particulier
Klantsoort	B	Bedrijf
Klantsoort	O	Overheid

Wijzigingen in de validatieregels

Validatieregels kunnen wijzigen, of vaak aangescherpt worden. Bij het in productie nemen van een aangescherpte controle moet tevens gecontroleerd worden of de reeds aanwezige gegevens aan de aangescherpte controles voldoen. Er wordt zonodig een lijst gemaakt van de gegevens die niet aan de nieuwe controles voldoen. Aan de hand van deze lijst kunnen de foutieve gegevens (handmatig of met bijvoorbeeld SQL-update scripts) gecorrigeerd worden.

Goede werkvoorbereiding

De stap die voorafgaat aan het invoeren is ook van belang voor de kwaliteit van de gegevens:

- Zorg voor duidelijke invoerdocumenten waarop de in te voeren gegevens overzichtelijk verzameld zijn;
- Zorg dat hierop de gegevens zo compleet mogelijk verzameld worden zodat niet, tijdens het invoeren, nog andere bronnen geraadpleegd moeten worden om de invoer compleet te maken;
- Zorg voor duidelijke regels voor het maken van codes (userid's, artikelcodes etcetera) zodat niet een tweede code aangemaakt wordt voor hetzelfde object omdat deze niet gevonden werd onder de eerste code;
- Zorg voor duidelijke procedures en workflow rond de gegevensverwerking;
- Zorg voor goede autorisaties. Voorkom dat een junior medewerker zelf codes gaat bedenken en inbrengen voor foutief begrepen situaties.

Mogelijkheden tijdens de invoer

Er zijn nog meer mogelijkheden om de kwaliteit van de gegevens te verhogen:

- Voorkom (fouten bij) intikken door gegevens automatisch over te nemen vanuit een ander bronsysteem. Bijvoorbeeld: een order wordt door de klant op een website ingegeven of bankafrekeningen worden via een bestand, aangeleverd door de bank, ingelezen;
- Gebruik leespennen of barcodes in plaats van het overtikken van codes;
- Zorg voor goede opmaak van grote numerieke gegevens zoals banknummers of bedragen door hierin spaties of punten op te nemen (vergelijk de leesbaarheid van het bedrag 123456789,00 met 123.456.789,00);
- Gebruik gemakkelijke codes: een landencode als NL of BE leidt minder snel tot fouten dan de codes 31 respectievelijk 32;
- Gebruik controletotalen op bedragen. Bij afwezigheid van geschikte numerieke gegevens kan een telling van het aantal verwerkte records worden gebruik voor een volledigheids-controle;
- Controleer belangrijke invoer door visuele controle of door de gegevens twee keer in te voeren.

Controles achteraf

Gebruik batch-gewijze controles op gegevens door achteraf nog op vreemde situaties te controleren. Zet bijvoorbeeld alle woonplaatsen eens gesorteerd op een rij: hoeveel keer is dezelfde woonplaats verkeerd gespeld, hoe vaak wordt er slordig met hoofd- of kleine letters in adressen en namen omgesprongen? Hiermee kan de kwaliteit van de invoer worden gemeten. Daarna kan deze informatie worden gebruikt om de bestaande gegevens te corrigeren en (zeker zo belangrijk) maatregelen te nemen, om dit soort fouten te voorkomen door het opnemen van extra werkinstructies of controles bij de invoer.

Een extra controle is mogelijk door de gegevens, na de invoer, terug te koppelen naar de bron van deze gegevens, bijvoorbeeld een bevestigingsbericht van de bestelling, opdracht of registratie. Hiermee kan aan de bron, door de klant/opdrachtgever, nog een keer gecontroleerd worden of de gegevens correct in het systeem zijn ingevoerd.

Correctheid en consistentie

Het is belangrijk om onderscheid te maken tussen de begrippen ‘correctheid’ en ‘consistentie’ van de gegevens. Consistentie kan op vele manieren met invoercontroles worden afgedwongen. Maar dat wil nog niet zeggen dat de gegevens correct zijn. Algehele correctheid kan nooit door het systeem worden afgedwongen, omdat het systeem geen kennis heeft van de werkelijkheid. Alleen consistentie met andere gegevens in het systeem of met geprogrammeerde validatieregels kan worden afgedwongen. Tikfouten in namen zijn daarom onvermijdelijk. Volledigheid is ook een vorm van kwaliteit die alleen kan worden afgedwongen als er regels zijn waartegen de volledigheid kan worden getoetst. Dit kan bijvoorbeeld met de hiervoor beschreven controle-tellingen op bedragen of door een telling van het aantal ingevoerde records te vergelijken met het aantal regels in het brondocument.

Voor één systeem is het nog mogelijk om (tot op zekere hoogte) de consistentie van de gegevens te waarborgen. Over meer systemen heen wordt dit echter veel moeilijker. De gegevens van een personeelslid of klant (of verwijzingen daarnaar) staan vaak in diverse systemen van een organisatie. Om over deze systemen heen de kwaliteit (consistentie, volledigheid en zoveel mogelijk correctheid) te waarborgen stelt ons voor veel technische en waarschijnlijk nog veel meer procedurele en menselijke uitdagingen. De volgende stappen kunnen hierbij helpen:

- Breng eerst de systemen van de organisatie in kaart;
- Van elk systeem wordt het doel en gegevensmodel geïnventariseerd.

CASE tool en genereren van de fysieke database-definities

Het logisch gegevensmodel kan worden vastgelegd in een CASE tool, zoals Power Designer. Hierbij is het mogelijk om vanuit de CASE tool het fysieke model en de database/definities te genereren. Een uitwerking hiervan is op www.dbm.nl/site/Projecten/specials.htm te vinden.

In het fysieke gegevensmodel (het RDBMS) kunnen de invoercontroles nog op verschillende manieren worden geïmplementeerd: declaratief of met database triggers. Veel controles kunnen declaratief worden gedefinieerd, dat wil zeggen: ze worden in de definitie van de tabellen en kolommen vastgelegd. Dit geldt bijvoorbeeld voor het type gegeven (numeriek, verplicht), voor toegestane waarden en referentiële integriteit.

Voor complexere regels (bedrag van een order mag niet groter zijn dan ...) zal in de CASE tool vaak een verbale beschrijving nodig zijn. In de database is dan een database trigger nodig om de controle op de database-laag te kunnen uitvoeren. Deze controles zullen meestal geprogrammeerd moeten worden en niet vanuit het logisch gegevensmodel gegenereerd kunnen worden.

De implementatie en mogelijkheden van een trigger zijn in de diverse producten verschillend. Er zijn triggers die voor het event (insert, update of delete) afgaan en/of die na het betreffende event afgaan. Sommige producten kennen een trigger die bij de commit afgaat. Daarmee is een controle mogelijk die verifieert of er bij een order tenminste 1 orderregel bestaat. Met alleen referentiële integriteit of post insert triggers is deze controle in de database niet mogelijk. Als alternatief moet de controle dan in de applicatie worden uitgewerkt.

Code-bijlagen

Op de website van Database Magazine zijn extra bijlagen te vinden betreffende de twee artikelen van Toon Loonen over Datakwaliteit. (zie ook DB/M 4, 2005).
U kunt de volgende documenten c.q. bestanden downloaden:

Kwaliteit-begint-bijlage: betreft het Genereren van database definities met Power Designer 11.
Dit is een toelichting op de (gezipte) bijlagen, met daarin een uitwerking van het logisch model en twee keer het fysieke model.

cdm.zip: Power Designer HTML-rapport van het logisch model.
pdm_mssql.zip: Power Designer HTML-rapport van het fysiek model voor MS SQL Server met 1 tabel voor klant.
pdm_oracle.zip: Power Designer HTML-rapport van het fysiek model voor Oracle met 2 tabellen voor klant.

Kijk op www.dbm.nl/site/Projecten/specials.htm

Indien een gegeven in meer systemen wordt geregistreerd, ga dan na of het mogelijk is om deze gegevens:
- een keer in te voeren en daarna naar de andere systemen te dupliceren [8];
- of op 1 plaats te registreren en andere systemen (via service calls, distributed databases of wat de techniek maar te bieden heeft) van deze gegevens gebruik te laten maken [1].

Het belang van controles

Invoercontroles dienen in de eerste plaats om de consistentie van de gegevens te waarborgen. Maar zij hebben vooral ook een functie om problemen in de latere verwerking te voorkomen. Een bedrag gelijk aan 0 in een betalingsopdracht zou bij het verwerken van de opdracht mogelijk tot onverwacht gedrag van de programmatuur kunnen leiden, mogelijk loopt de verwerking hierop zelfs vast. In dat geval moet bij de invoer reeds een controle gebouwd worden zodat alle latere processen de gegevens goed kunnen verwerken. Omgekeerd kan het fysiek ontwerp en de bouw van de volgende processen leiden tot het definiëren van aangescherpte controles op de invoer. Denk bijvoorbeeld aan het voorkomen van delen door 0.
Het zijn niet alleen de processen van het eigen systeem die deze controles nodig hebben. De volgende processen kunnen ook onderdeel vormen van een ander systeem. Zo zal een ordersysteem gebruik kunnen maken van de gegevens in het klantenregistratiesysteem. Het datawarehouse maakt weer gebruik van het klanten- en ordersysteem. Al deze systemen zijn gebaat bij een consistente, volledige en zoveel mogelijk correcte invoer aan de voordeur van deze systemen.

Controles brengen extra kosten met zich mee. Extra handelingen, zoals een totaaltelling maken van de bedragen voor de invoer, kosten extra tijd van de medewerkers. Het definiëren en bouwen van invoercontroles kost extra werk van de systeemontwikkelaars. Deze laatste kosten zijn eenmalig en daarom snel terug te verdienen. De eerstgenoemde handelingen komen bij elke invoer weer terug en zullen dus snel hoger oplopen. Daarom zal er telkens een afweging gemaakt moeten worden wanneer deze kosten wel en wanneer ze niet 'rendabel' zijn. Bedenk hierbij dat het herstellen van fouten ook veel tijd en geld kost. Daarnaast kunnen fouten ook imago schade veroorzaken.

Conclusies

De kwaliteit van de gegevens in de database is voor een groot deel afhankelijk van de kwaliteit van het gegevensmodel en de kwaliteit van de invoer. Deze kwaliteit kan verhoogd worden door:
- alle regels, van eenvoudige invoervalidaties tot complexe Business Rules, in het gegevensmodel van het systeem op te nemen;
- zo mogelijk vanuit dit model zowel de database-laag als de client-laag te genereren;
- aanvullende procedures (goede werkinstructies, controletellingen, visuele controles van de invoer) in te stellen, indien de gegevens niet op een andere manier gecontroleerd kunnen worden.

De kosten van het extra werk van deze controles moeten natuurlijk opwegen tegen het belang van de gegevens.

Literatuur

1. *Loonen. Gegevenskoppelingen in een 4GL/RDBMS omgeving. Database Magazine 1996/8.*
2. *Loonen. Gebruik van afgeleide gegevens in ontwerp en bouw. Database Magazine 1997/1.*
3. *Loonen. Gegevensmodel van een distributed data dictionary. Database Magazine 1997/4.*
4. *Loonen. Modelleren van subtypes. Database Magazine 1999/1.*
5. *Loonen. Datum en tijd in het logisch en fysiek gegevensmodel. Database Magazine 2001/6.*
6. *Loonen. Gebruik van speciale tekens in de database. Database Magazine 2002/8.*
7. *Loonen. NULL in het logisch en fysiek gegevensmodel. Database Magazine 2004/1,4.*
8. *Van der Graaf. Het adaptief ontwerpen van bedrijfsregels. Database Magazine 2003/3,4,6.*

Toon Loonen

Toon Loonen (toon.loonen@capgemini.com) is werkzaam bij Capgemini.