Kwaliteit van de gegevens begint bij het begin (1)

Overzicht van soorten invoercontroles

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 een tweetal artikelen wordt een overzicht gegeven van de mogelijkheden en alternatieven om kwaliteit in de gegevens te verankeren.

Zonder een goed (bij de vraag passend) gegevensmodel is het niet mogelijk om gegevens goed op te slaan. Een goed gegevensmodel begint natuurlijk met een genormaliseerd model van de gegevens. Hierin staan meteen al veel controles impliciet gedefinieerd, zoals het datatype (numeriek, datum, etcetera), wel/niet verplicht, primaire/kandidaat sleutels en referentiële integriteit.

Redundant

Het gebruik van *subtypes* (een klant is ofwel een natuurlijk persoon of een rechtspersoon) geeft vervolgens veel mogelijkheden om de controles op gegevens nader te beschrijven. Bijvoorbeeld bij een natuurlijk persoon moet een geboortedatum (meer dan 18 jaar voor de systeemdatum) worden ingevuld, bij een rechtspersoon moet een verwijzing staan naar een natuurlijk persoon die de directeur/voorzitter/contactpersoon is van de organisatie [4]. Naast validaties kunnen ook Business Rules, zoals rekenregels, in het gegevensmodel worden opgenomen, zie het kader.

> Vaak worden bedragen overgenomen van een invoerdocument

Bijvoorbeeld: het uitstaand saldo van een klant is het totaal van alle facturen minus alle ontvangen betalingen. Door het uitstaand saldo als virtueel (berekend) attribuut in het gegevensmodel op te nemen wordt impliciet deze regel ook in het gegevensmodel opgenomen [2]. Zo'n gegeven is later een kandidaat om redundant opgeslagen te worden ten behoeve van een betere performance. Men moet er wel zeker van zijn dat deze redundante gegevens consistent zijn met de onderliggende basisgegevens. Dit kan door bijvoorbeeld de redundante gegevens alleen met database triggers te laten bijwerken. Het gebruik van domeinen is uitgebreid beschreven in [3]. De definitie van het domein met alle regels (voor validatie, opmaak, etcetera) wordt in het gegevensmodel vastgelegd. Een voorbeeld van een domein is een postcode: "Een veld van zes posities met een indeling vier cijfers en twee hoofdletters. Het moet bestaan in de postcode tabel (indien aanwezig). Bij afdrukken moet altijd een spatie tussen de letters en de cijfers worden afgedrukt."
Een domein maakt het mogelijk om alle regels waaraan een attribuut (dat gebaseerd is op het domein) moet voldoen op 1 plaats in het gegevensmodel te beschrijven, ook als dit domein op meer plaatsen wordt gebruikt.

Hieronder volgt eerst een overzicht van de invoercontroles, deze worden nader beschreven:

1. Type van de kolom (character, numeriek, datum/tijd, boolean);
2. Verplicht (NULL, NOT NULL) [7];
3. Lengte (minimum, maximum);
4. Bij numeriek: aantal decimalen toegestaan, minimum en maximum waarde;
5. Bij numeriek: Voldoet aan de 11-proef of vergelijkbare controles;
6. Bij datum: resolutie, minimum en maximum waarde [5];
7. Bij stingvelden: patroon (bijvoorbeeld postcode, telefoonnummer, kentekennummer, e-mail adres), maximum lengte, toegestane tekens;
8. CODES: hard gecodeerde codes, bijvoorbeeld M of V voor man of Vrouw;
9. Unieke waarden (primaire sleutel, kandidaat-sleutel);
10. Referentiële integriteit (buitensleutels);
11. Controle van waarde ten opzichte van andere waarden in hetzelfde record;
12. Controle van waarde ten opzichte van waarden in een andere tabel (bedrag van order ten opzichte van maximum toegestane orderbedrag bij de klant);
13. Controletellingen.

Controles op numerieke waarden

Op numerieke waarden kan een aantal controles worden uitgevoerd, bijvoorbeeld:

- Het type: alleen gehele getallen, een vast aantal decimalen of floating point;

- De hoogste of laagste waarde of eventueel individuele toegestane waarden;
- Controles tegen andere waarden in de database (bedrag kleiner dan een maximaal saldo);
- De 11 proef bij numerieke codes (zoals bij SOFInummer, zie http://cgi.dit.nl/sofi.cgi of Bankrekeningnummer, zie http://cgi.dit.nl/bank.cgi) kan ook voor systeemeigen codes (zoals grootboeknummers of artikelnummers) worden toegepast. Deze controle minimaliseert de kans op tikfouten, maar niet de kans op het inbrengen van een verkeerde maar wel geldige code (banknummer van de verkeerde crediteur).
- Zie ook voor informatie over artikelnummers www.ean.nl voor algemene artikelidentificatie, of www.isbn.nl voor de controle op ISBN-nummers van boeken.

Er zijn diverse andere mogelijkheden om bij het invoeren van numerieke waarden de kans op fouten te verkleinen. Omdat vooral de kans op fouten groot is bij het inbrengen van bedragen en omdat hierbij het nadeel van een fout groot is, zijn hier extra controles op hun plaats. Vaak worden bedragen overgenomen van een invoerdocument, bijvoorbeeld een rekeningafschrift van de bank. Door zowel de algemene gegevens van het bankafschrift (datum, totaal debet en credit) als de detailregels in te brengen kan het totaal van de detailregels vergeleken worden met de totaalregels van het bankafschrift. Is er een verschil dan is er ergens een fout gemaakt. Bij belangrijke financiële gegevens waar zo'n totaal niet voorhanden is, is het zinvol eerst handmatig een controletotaal van de in te voeren bedragen te maken en daarna dit totaal samen met de details in te voeren.

Controles op datums en periodes

Gebruik altijd het datum-type van het RDBMS voor datums. Dit voorkomt ongeldige datums in de database. Kijk verder naar de resolutie (nauwkeurigheid) van datum- en tijdvelden. Als alleen een datum relevant is en de tijd niet (datum van een factuur) dan moet de tijd ook op 00:00:00 staan of niet ingevuld zijn. Een willekeurige tijd (soms de systeemtijd bij invoer) maakt later het vergelijken van datums lastiger. De resolutie kan ook een hele maand zijn: Een nieuw salaris gaat vaak alleen op de eerste dag van de maand in [5].

Voor een datum kan vaak een range aangegeven worden waarbinnen deze datum moet vallen. Bijvoorbeeld de geboortedatum van een medewerker moet tenminste 16 jaar (en niet meer dan 65 jaar) voor de systeemdatum liggen. De datum van een order mag niet in de toekomst en niet verder dan vijf dagen in het verleden liggen. Voor zo'n orderdatum kan vaak de systeemdatum als default waarde worden gebruikt. Dit voorkomt tikwerk en mogelijke fouten.

De opmaak van een datum moet duidelijk zijn: 04-05-2004 zal in de USA als 5 april 2004 en in Nederland als 4 mei worden geïnterpreteerd. Gebruik daarom bij voorkeur een notatie in de vorm van Jaar-maand-dag uur:min:sec (bijvoorbeeld 2004-05-19 17:00:00).

Validatieregels, Invoercontroles of Business Rules.

Naast de term validatieregels of invoercontroles wordt tegenwoordig ook de term 'Business Rules' vaak (te pas en te onpas?) gebruikt. Deze termen houden in:
Validatieregels beschrijven de validaties, ofwel controles waaraan de gegevens moeten voldoen.
Invoercontroles zijn de uitwerking van de validatieregels in het systeem. Dit kan ofwel in de database, ofwel in de client, ofwel in beide worden geïmplementeerd.
Business rules beschrijven de niet technische validatieregels.

Een voorbeeld. In girotel (de PC-toepassing om giro-overschrijvingen te maken) moet het over te maken bedrag groter dan of gelijk aan 0,25 Euro zijn.
Het is duidelijk dat het bedrag > 0,00 Euro moet zijn. Dit is een technische invoercontrole (voorkom dat de gebruiker het bedrag vergeet in te vullen; wat zou het systeem moeten doen met een bedrag dat gelijk 0 of negatief is?).
Dat het bedrag ook nog >= 0,25 Euro moet zijn (en niet bijvoorbeeld 1 Euro) is dan een Business Rule.
De ontwerper kan de technische invoercontroles grotendeels zelf bedenken. Voor de controles op basis van Business Rules zal hij aanwijzingen van de gebruikers nodig hebben, meestal in de vorm van procedures of procesbeschijvingen.

Daarnaast beschrijven Business Rules ook andere zaken, zoals berekeningen (van kortingen, rente en boetes) en bijvoorbeeld workflow.
In dit artikel wordt uitsluitend de term 'invoercontroles' gebruikt.
De term 'Business Rules' (met quotes, dus als complete term) ingetikt in Google geeft heel wat 'inspiratie'. Zie bijvoorbeeld: www.businessrulesgroup.org en www.brcommunity.com/faqs.php
Van 7 t/m 10 Juni 2005 is er in Amsterdam een speciale conferentie over Business Rules.

Kijk op www.eurobizrules.org voor meer informatie.

Voor periodes (met een begin- en einddatum) of een algemene tijdsduur (de training duurt drie dagen) gelden dezelfde aspecten.

Controles op character strings

Character-velden kunnen gebruikt worden voor Codes, korte omschrijvingen of namen en lange tekstvelden.
Op codes (M/V voor Man/Vrouw, NL voor Nederland) kunnen veel controles worden gedefinieerd [3]:
- Gebruik voor codes altijd alleen hoofdletters;
- De codes kunnen hard gecodeerd zijn (M/V) of gevalideerd tegen de waarden in een tabel (landcode NL, klantnummer, artikelnummer, grootboeknummer, etcetera);

Invoercontroles in de database of in de toepassing

In een 2-tier of 3-tier systeem kunnen invoercontroles op elk van de tiers worden geïmplementeerd. Een 2-tier systeem bestaat uit een client-applicatie en een database-laag. Een 3-tier systeem bestaat uit een (thin)-client applicatie (vaak een browser met HTML of Java-coding), een middle-tier (applicatielaag) en een database-laag.

Bij een 2-tier systeem kunnen veel invoercontroles zowel in de applicatie als in de database-laag worden gebouwd. Het bouwen in de database-laag heeft als voordeel dat de controles altijd worden uitgevoerd, welk programma de gegevens ook aanlevert. Het nadeel is echter dat de controles pas op een laat moment worden uitgevoerd, namelijk als het scherm volledig is ingevuld en de gegevens naar de database worden gestuurd. Daarom is het uitvoeren van de controles in de client-applicatie gebruikersvriendelijker. De meeste controles kunnen dan worden uitgevoerd bij het verlaten van een veld.

Een alternatief is om de controles in beide tiers uit te voeren. In veel programmeeromgevingen is dit alleen mogelijk door de controles twee keer te coderen, dus met veel dubbel werk. Het ideaal in dit opzicht is een 4GL met een zeer complete repository (metamodel) waarin alle controles worden gedefinieerd. Vanuit deze repository moeten dan de database-definities en het systeem beide worden gegenereerd, inclusief alle (of zoveel mogelijk) controles in beide tiers.

Bij een 3-tier applicatie is de client-laag opgesplitst in een thin client tier (browser met HTML, Java, etcetera) en een middle tier. Hierin zullen de invoercontroles vaak worden opgesplitst:

- Simpele technische controles (verplicht, numeriek, vaste codes) kunnen op de thin client worden gebouwd;
- Controles tegen gegevens in de database (bestaat het artikelnummer, heeft de klant voldoende saldo) worden op de middle tier gebouwd of alleen in de database gedaan;
- Ook complexe validaties (zoals: indien persoon gehuwd en vrouw is, dan moet de geboortenaam ook ingevuld zijn) of flexibele controles aan de hand van een rule engine [8] worden in de middle tier uitgevoerd.

  De gebruiker tikt bij deze toepassingen meestal eerst een heel scherm vol. Pas dan worden de gegevens van het scherm naar de middle tier verstuurd. Het maakt voor de respons over fouten naar gebruiker geen verschil meer of de controle in de middle tier of in de database wordt uitgevoerd.

Bij controles door de database wordt een foutmelding door het RDBMS teruggemeld aan de applicatie. De applicatie moet de vaak technische, en Engelse, melding vertalen naar een begrijpelijke Nederlandse melding die aan de gebruiker getoond wordt. Alleen meldingen die de gebruiker zelf niet kan oplossen (bijvoorbeeld: de database is vol) hoeven niet vertaald te worden. De instructie om bij alle Engelstalige meldingen de applicatiebeheerder te waarschuwen moet voldoende zijn.

Zorg dat bij de technische foutmelding uit de database de gegevens die de applicatiebeheerder of DBA nodig heeft om het probleem te analyseren en herstellen (foutcode, routine waarin de fout is ontstaan, sleutelwaarden van de gegevens), niet verloren gaan. Deze gegevens kunnen vermeld worden op het scherm van de gebruiker en/of in een logfile.

- Codes kunnen een zeker formaat hebben (postcode, telefoonnummer, een e-mail adres moet een @ bevatten);
- Codes hebben vaak een vaste of maximale lengte;
- Een code als identificerend gegeven (primaire of kandidaat sleutels: landnummer in de landentabel) moet uniek zijn.
  Als het een refererend gegeven is (landcode in de klantentabel) moet het als sleutel in de gerefereerde tabel bestaan.

Op namen en omschrijvingen zijn controles lastiger te definiëren. Het probleem bij een tikfout in een naam (Janssen in plaats van Jansen) leidt meestal niet tot grote problemen (bij fouten in bedragen of codes is dit veel eerder het geval) maar staat wel slordig bij de ontvanger. In sommige gevallen kan een foute naam wel tot grote problemen leiden (aanklacht in een rechtszaak) en is een extra controle op de ingevoerde gegevens noodzakelijk.

Bij adres- en woonplaatsgegevens kan de kans op fouten worden verminderd door alleen de postcode en huisnummer in te brengen. De straatnaam en woonplaats worden dan telkens uit de postcodetabel opgehaald én visueel gecontroleerd tegen de opgegeven straatnaam en woonplaats. Dit is echter alleen zinvol bij een groot adressenbestand en meestal niet zinvol voor buitenlandse adressen. Bij namen kan men aangeven hoe hoofdletters of kleine letters gebruikt moeten worden. Het systeem kan dan automatisch een correctie uitvoeren als de naam geheel in hoofdletters of kleine letters is opgemaakt (janssen wordt Janssen). Een naam als McDonalds moet echter ook in het systeem ingevoerd kunnen worden.

Kijk verder hoe omgegaan wordt met diakritische tekens (namen als Daniëls). In een omgeving met meer computer-types (IBM Mainframe, UNIX-systemen, MS Windows, Browsers) kunnen deze tekens in diverse omgevingen verkeerd geïnterpreteerd worden [6].

Controles tegen andere gegevens in hetzelfde record

De hiervoor genoemde controles betreffen altijd 1 veld (op het scherm), kolom (in de database) of attribuut (in het gegevensmodel). Als alle gegevens van een record zijn ingevoerd is het ook mogelijk om controles uit te voeren op de gegevens van een record onderling, bijvoorbeeld:

- Een datum X (geboortedatum) moet altijd liggen voor een datum Y (huwelijksdatum);

- Indien de status van een persoon is "gehuwd", dan moet de huwelijksdatum ingevuld zijn;
- Als deze persoon een vrouw is moet tevens de geboortenaam (meisjesnaam) ingevuld zijn.

> Op namen en omschrijvingen zijn controles lastig te definiëren

Controles tegen andere gegevens in dezelfde database

De ingevoerde gegevens kunnen ook tegen andere gegevens in dezelfde database worden gevalideerd, bijvoorbeeld:

- Referentiële integriteit;
- Het totaalbedrag van een order plus het huidig uitstaand saldo van de betreffende klant mag het maximum saldo voor deze klant (of dit klanttype) niet overschrijden;
- Het aantal bestelde artikelen op een orderregel mag niet groter zijn dan de aanwezige voorraad.

Controles tegen gegevens in een andere database

De genoemde controles kunnen ook soms systeemgrenzen overschrijden. Bijvoorbeeld de klantgegevens worden bijgehouden in het klantsysteem en de orders en artikelen in het ordersysteem. Voor de controle op het overschrijden van het maximum saldo moet over de systeemgrens heen gekeken worden. Dit kan op verschillende manieren:

- Roep vanuit het ordersysteem een functie (RPC, Service in SOA) in het klantsysteem aan;
- Stuur een bericht naar het klantsysteem en wacht op een bericht terug of het OK is;
- Definieer in het ordersysteem een view op de klanttabel uit het klantsysteem. De klantgegevens lijken dan lokaal te staan en het RDBMS zal de controle transparant over de database-grens heen uitvoeren [1], [3].

In deel 2 zullen onder andere flexibele controles, correctheid en consistentie van gegevens besproken worden.

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.

Update

Performance Management applicaties voor BusinessObjects XI

Business Objects, leverancier van Business Intelligence-oplossingen, introduceert Performance Management applicaties voor BusinessObjects XI. In de applicaties zijn best practices, dashboards, scorecards, prestatie-indicatoren en analyses verenigd om organisaties te ondersteunen bij het optimaliseren van hun kritieke bedrijfsprocessen. De performance management-applicaties zijn beschikbaar voor verschillende sectoren waaronder retail, consumer package goods, telecommunicatie en de financiële sector. Door de inzet van de performance management-applicaties zou men beter inzicht in bedrijfsgegevens kunnen krijgen. De Sarbanes-Oxley Compliance Solution biedt financiële managers inzicht in financiële prestaties en processen die boven de interne risiconiveaus uitkomen.
Voor meer informatie: www.businessobjects.com

BMC maakt prestatiebeheer mogelijk voor DB2

BMC Software heeft een oplossing geïntroduceerd voor beheer en afstemming van de DB2 Universal Database (UDB). Met de nieuwe SmartDBA Performance Solution voor DB2 UDB versie 3.0 kunnen gebruikers onder Unix, Linux en Windows betere systeemprestaties bereiken. SmartDBA Performance Solution voor DB2 UDB betekent voortdurende bewaking en instandhouding van het noodzakelijke prestatieniveau van de database, minder onderhoud, zowel als het gaat om tijd als om frequentie, en betere anticipatie op toekomstige uitbreiding van applicaties. Op basis van nieuwe, ultramoderne technologie biedt SmartDBA Performance Solution voor DB2 UDB een verbeterde, geavanceerde vorm van prestatiebeheer, waarbij prestatiegegevens en -statistieken worden verzameld en opgeslagen in direct toegankelijke overzichten. Daarnaast maakt de gepatenteerde 'equalization'-methodiek het mogelijk ruwe data om te zetten in leesbare prestatie-informatie.
Meer informatie op www.bmc.com/datamanagement