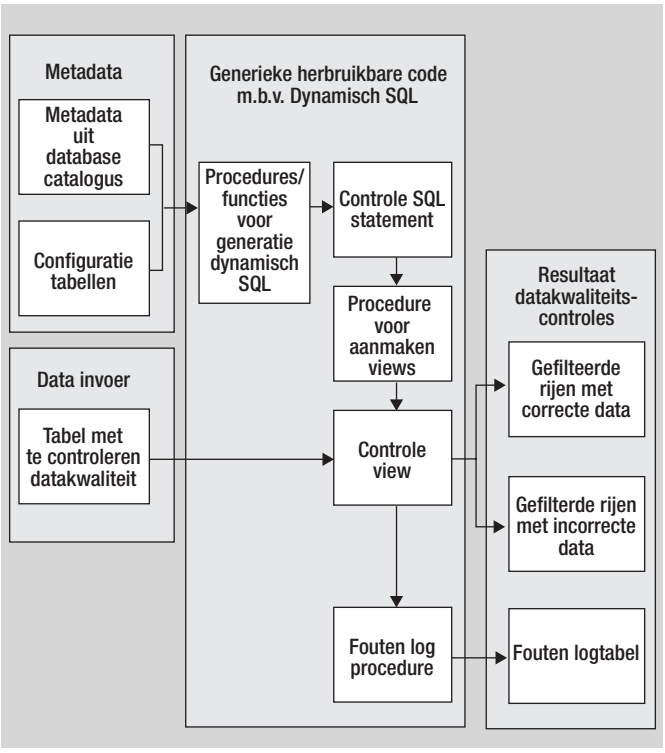
Ook zonder specifieke tools goede oplossing mogelijk

Datakwaliteitsborging met Oracle dynamisch SQL

Reinbert Hamstra

De bruikbaarheid van gegevens in een organisatie is rechtstreeks afhankelijk van de kwaliteit en betrouwbaarheid van die gegevens. Reeds bij het ontwerp van een informatiesysteem zal daarom moeten worden vastgesteld hoe de kwaliteit en betrouwbaarheid van de invoergegevens kunnen worden gecontroleerd en gewaarborgd.

Bij veel ETL-tools worden tegenwoordig nieuwe opties meegeleverd die speciaal op controle en handhaving van datakwaliteit zijn toegerust. Zo heeft Informatica PowerCenter bijvoorbeeld Informatica Data Explorer en heeft de meeste recente versie van Oracle Warehouse Builder (10g release 2) een Data profiler. Het is echter ook mogelijk om met veel eenvoudiger middelen datakwaliteit te controleren en te handhaven.
In dit artikel wordt een methode besproken om datakwaliteit te controleren en te handhaven in een Oracle-omgeving. Deze methode is herbruikbaar en schaalbaar en maakt uitsluitend gebruik van features die tegenwoordig standaard in iedere Oracle database aanwezig zijn. De complexiteit en kosten van een aparte datakwaliteits- of ETL-tool kunnen hiermee worden voorkomen.



Afbeelding 1: Structuuroverzicht generieke datakwaliteitscontrole.

Mogelijkheden Oracle database

Binnen de Oracle database zijn standaard enkele features aanwezig die het controleren en waarborgen van datakwaliteit faciliteren. Ten eerste wordt de mogelijke inhoud van een kolom beperkt door het datatype en de lengte van de kolom. Door constraints op tabellen toe te passen kan een aantal andere aspecten van datakwaliteit worden afgedwongen. Voor het bewaken van complexe logica kunnen triggers worden ingezet. Tijdens het laden zullen alleen de gegevens die aan de hiermee gestelde eisen voldoen worden geladen; de rest wordt afgewezen op het eerste aspect dat niet voldoet. Met deze methode is het mogelijk een basaal niveau van datakwaliteit te garanderen. Vaak is echter meer gedetailleerde informatie over de datakwaliteit vereist. Bovendien moeten de gegevens die niet voldoen vaak volgens bepaalde regels worden opgeschoond. Dit vereist een meer verfijnd mechanisme.

De hier beschreven methode implementeert dit op een generieke wijze en maakt slechts gebruik van standaard functionaliteit die in iedere Oracle database aanwezig. Deze methode heeft als belangrijkste voordeel dat de datakwaliteitscontroles volledig configureerbaar zijn. Dit betekent dat voor toepassing van de datakwaliteitscontroles op nieuwe tabellen geen code handmatig hoeft te worden toegevoegd. Verder wordt de informatie over fouten in een generieke vorm opgeslagen in de database en niet in losse files op het operating systeem, zoals dat bij SQL loader gebeurt.
De generieke oplossing maakt gebruik van Oracle dynamisch SQL. Een SQL statement wordt opgebouwd aan de hand van een

bepaalde configuratie. Deze configuratie is vastgelegd in configuratietabellen; deze bevatten per veld/kolom de datakwaliteitscontroles die op de data moeten worden uitgevoerd. Voor elke kolom kunnen meerdere soorten controles worden opgegeven.
Op basis van de kolomnamen uit de database-catalogus en de inhoud van de configuratietabellen worden SQL statements opgebouwd die de datakwaliteit controleren. Dit wordt gerealiseerd met behulp van generieke functies en procedures die de SQL statements voor elke gewenste tabel/kolom-combinatie kunnen opbouwen.
De datakwaliteitscontroles vinden plaats in controlekolommen met daarin specifieke SQL-expressies. Uit de waarde van controlekolommen blijkt of aan de gestelde controles is voldaan of niet. De controlekolommen bevatten bijvoorbeeld de waarde 0 indien een veld correct gevuld is en een waarde groter dan 0 indien dit niet het geval is. De exacte waarde van de controlekolom geeft gedetailleerde informatie over de datakwaliteitsproblemen die zijn geconstateerd.
Aan de hand van de controlekolommen is gemakkelijk per kolom en per rij vast te stellen of aan de controles is voldaan. Dit kan men gebruiken om rijen of kolomwaarden op basis van datakwaliteit te selecteren. In het onderstaande is de beschreven werkwijze verder uitgewerkt.

Typen datakwaliteitscontroles.
Om de datakwaliteit van de invoergegevens te verifiëren worden verschillende typen controles uitgevoerd. We geven hier ter illustratie typen controles die voor de hand liggen, zie afbeelding 2. Elk type controle krijgt een controlegetal uit een exponentiële reeks toegekend. De functie van het controlegetal zal worden toegelicht. Naast de gedemonstreerde controles is elk type controle dat uit te voeren is met Oracle SQL in principe geschikt om toe te voegen.

Configuratietabellen.
Om de configuratie van de uiteenlopende typen controles efficiënt en eenvoudig op te slaan krijgt elk type controle een aparte configuratietabel. Dit heeft als voordeel dat de dataconsistentie van de configuratietabellen met eenvoudige constraints en triggers is af te dwingen. Aan de hand van de inhoud van de configuratietabellen wordt per kolom een controle-expressie geconstrueerd.

Controlekolom.
Voor elke kolom in de te controleren gegevens wordt in SQL een controlekolom geconstrueerd. Uit de numerieke waarde van de controlekolom is af te leiden of aan de geconfigureerde controles is voldaan. Indien aan alle controles is voldaan krijgt de controlekolom de waarde 0. Indien niet aan een controle wordt voldaan wordt een voor deze controle specifiek controlegetal bij de controlekolom opgeteld. De controlegetallen voor de verschillende controles vormen een exponentiële reeks. Dit garandeert dat ze kunnen worden opgeteld tot één waarde waaruit de verschillende oorspronkelijke waarden zijn te herleiden. Hiermee is het mogelijk om met één controlekolom het resultaat van meerdere controles te representeren.
Stel dat in de configuratietabellen voor een specifiek veld is vastgelegd dat de waarde uniek moet zijn en moet voorkomen in een referentietabel. Indien de veldwaarde voor een bepaalde rij niet uniek is en niet voorkomt in de referentietabel, wordt de waarde van de controlekolom voor die rij 110.
Het is van belang om een standaardnaamgeving te hanteren voor de controlekolommen, zodat ze gemakkelijk herkenbaar zijn en voor verdere stappen ook kunnen worden opgehaald uit de database-catalogus. Zo kan een vaste prefix aan elke originele kolomnaam worden toegevoegd, zodat voor de kolom 'naam' de controlekolom 'c_naam' wordt.

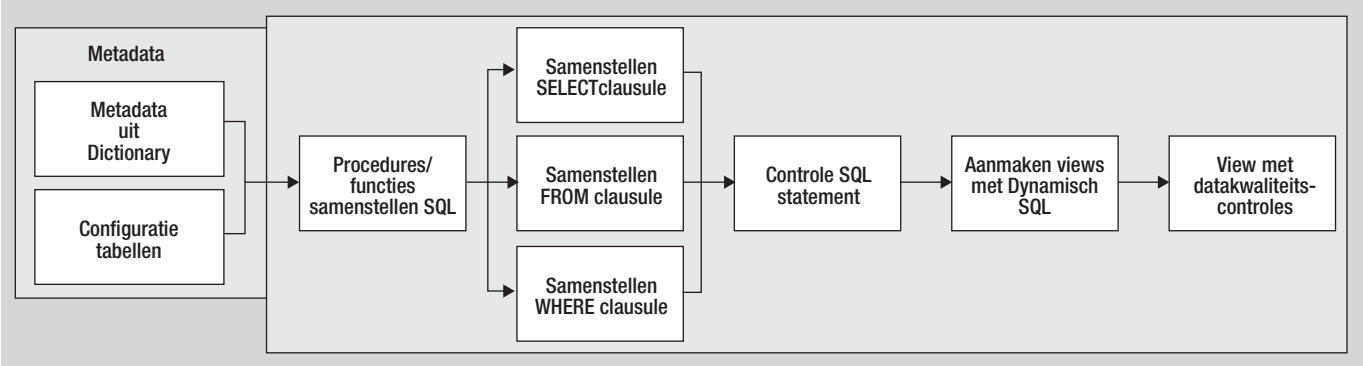
Controle	Omschrijving controle	Controlegetal
Verplicht	Veld moet verplicht een waarde bevatten, leeg veld is niet toegestaan.	$10^0 = 1$
Uniek	Veld of combinatie van velden moet uniek zijn.	$10^1 = 10$
Lookup	Veld of combinatie van velden moet voorkomen in gespecificeerde referentietabel.	$10^2 = 100$

Afbeelding 2: Voorbeeldtypen datakwaliteitscontroles met controlegetallen.

Controle SQL.
Voor elke te controleren tabel wordt een controle SQL statement opgebouwd uit een SELECT, een FROM en een WHERE clausule. In de SELECT clausule worden alle originele kolommen en de bijbehorende controlekolommen opgenomen. In de FROM clausule wordt in ieder geval de tabel die moet worden gecontroleerd opgenomen, plus tabellen waarmee een referentiële controle moet worden uitgevoerd. De WHERE clausule verbindt de tabellen uit de FROM clausule door middel van een outer-join conditie. Afbeelding 3 toont een algemeen SQL statement dat wordt samengesteld voor de datakwaliteitscontrole. Een expressie van een controlekolom levert 0 op in geval dat er geen controles zijn geconfigureerd of dat aan alle controles is voldaan voor een bepaalde veldwaarde.

```
select kolom a
     , controle-expressie kolom a
     , kolom b
     , controle-expressie kolom b
       etcetera...
from <te controleren tabel> tab
   , <lookup tabel>         lkp
where tab.sleutelkolom = lkp.sleutelkolom(+)
```

Afbeelding 3: Voorbeeld SQL statement voor controle van datakwaliteit.



Afbeelding 4: Constructie en dynamisch uitvoeren SQL statement.

Werknemersnummer	Naam	Leidinggevende
7369	SMITH	
7499	JONES	7369
7499		9999
7902	FORD	9999

Afbeelding 5: Externe tabel met medewerkers en leidinggevenden (WERKNEMERS).

Te controleren tabel	Te controleren kolom
WERKNEMERS	WERKNEMERSNUMMER
WERKNEMERS	NAAM

Afbeelding 6: Configuratietabel voor verplichte velden.

Dynamisch SQL.

Het hiervoor beschreven SQL statement kan met PL/SQL procedures en functies worden opgebouwd. Hierbij worden de tabel- en kolomnamen uit de database-catalogus opgehaald en gekoppeld aan de informatie uit de configuratietabellen. Zo wordt een SELECT, FROM en WHERE clausule samengesteld op basis van de opgegeven configuratie voor een bepaalde tabel, zie afbeelding 4.

Het samengestelde SQL statement kan nu runtime worden uitgevoerd op de database met behulp van dynamisch SQL. Vanwege de voordelen tijdens testen maken we gebruik van views met daarin de gegenereerde SQL statements. Op de techniek van het procedureel samenstellen van SQL statements en het dynamisch uitvoeren daarvan gaan we in dit artikel niet verder in. Hiervoor verwijzen we naar algemeen gangbare Oracle-documentatie.

Voorbeeld datakwaliteitscontrole

Ter illustratie van de datakwaliteitscontrole maken we gebruik van een externe tabel met gegevens over medewerkers en leidinggevenden zoals weergegeven in afbeelding 5. Stel dat de volgende regels gelden voor deze gegevens:

- werknemersnummer moet gevuld zijn;
- werknemersnummer moet uniek zijn;
- naam moet gevuld zijn;
- leidinggevende moet, indien gevuld, voorkomen als werknemersnummer;
- leidinggevende moet uniek zijn (deze regel is overigens niet erg realistisch).

Uniciteitsleutel	Te controleren tabel	Te controleren kolom
UNIEKE-SLEUTEL-01	WERKNEMERS	WERKNEMERSNUMMER
UNIEKE-SLEUTEL-02	WERKNEMERS	LEIDINGGEVENDE

Afbeelding 7: Configuratietabel voor unieke velden of combinaties van velden.

Voor het vastleggen van elk type uit te voeren controle gebruiken we een aparte configuratietabel. In dit voorbeeld zijn dat een configuratietabel voor verplichte velden (afbeelding 6), een configuratietabel voor unieke velden of combinaties van velden (afbeelding 7) en een configuratietabel voor referentiële relaties (afbeelding 8). Op basis van bovenstaande configuratie wordt als eerste een controleview aangemaakt met in de SELECT clausule voor iedere originele kolom tevens een controlekolom, zie afbeelding 9.

In de FROM en WHERE clausule wordt een outer-join gelegd tussen de originele tabel en lookup tabellen vanwege de controle op referentiële integriteit. In dit voorbeeld zijn de originele tabel en de lookup tabel fysiek dezelfde tabel. Voor de lookup tabellen wordt de benodigde kolom of kolommencombinatie gegroepeerd om te garanderen dat altijd slechts één rij per sleutelwaarde wordt geselecteerd. Dit is nodig als een nog ongecontroleerde lookup tabel dubbele rijen bevat. Zie afbeelding 10.

Het resultaat van het controle SQL statement staat weergegeven in afbeelding 11. In rij 3 is het veld naam niet gevuld. Dit resul-

Referentie-sleutel	Te controleren tabel	Te controleren kolom	Referentie-tabel	Referentiekolom
VREEMDE-SLEUTEL-01	WERKNEMERS	LEIDINGGEVENDE	WERKNEMERS	WERKNEMERSNUMMER

Afbeelding 8: Configuratietabel referentiële integriteit.

Kolom	Controles	Code controlekolom

Werknemers-nummer	Verplicht, uniek	case
		  when tab.werknemersnummer
		  is null
		  then 1
		  else 0
		end
		+
		case
		  when count(1)
		  over (partition by
		  tab.werknemersnummer
		  order by rownum) > 1
		  then 10
		  else 0
		end

Naam	Verplicht	case
		  when tab.naam is null
		  then 1
		  else 0
		end

Leiding-gevende	Uniek, referentieel	case
		  when count(1)
		  over (partition by
		  tab.leidinggevende
		  order by rownum) > 1
		  then 10
		  else 0
		end
		+
		case
		  when tab.leidinggevende is
		  not null
		  and lkp.werknemersnummer
		  is null
		  then 100
		  else 0
		end

Afbeelding 9: Code controlekolommen; kolommen van de te controleren tabel beginnen met 'tab' en kolommen van de lookup tabel met 'lkp'.

```
from werknemers tab
    ,(select werknemersnummer
      from werknemers
      group by werknemersnummer) lkp
where tab.leidinggevende = lkp.werknemersnummer(+)
```

Afbeelding 10: Code FROM en WHERE clausule.

teert in het bijbehorende controlegetal voor verplichte velden ($10^0 = 1$). De rijen 2 en 3 hebben dezelfde waarde in het veld werknemersnummer. Alleen de laatste waarde wordt hier als fout aangemerkt omdat de eerste keer dat een waarde voorkomt er nog geen sprake is van een dubbele waarde. Zodoende krijgt alleen rij 3 voor het veld naam de controlewaarde voor unieke velden ($10^1 = 10$). De rijen 3 en 4 hebben een waarde in veld leidinggevende die niet voorkomt in het veld werknemersnummer; bovendien is deze waarde niet uniek. Beide rijen krijgen voor het veld leidinggevende de controlewaarde ($10^2 = 100$). Bovendien wordt daar voor de laatste rij de controlewaarde voor unieke velden bij opgeteld ($10^1 = 10$). Opgeteld levert dat de controlewaarde 110 op.

Inpassen in ETL-proces

Het resultaat van de datakwaliteitscontroles staat voor elke gecontroleerde tabel in een gegenereerde controle-view. Inpassing van deze datakwaliteitscontrole in een ETL-proces vereist integratie op drie punten: invoer van gegevens; filteren van gegevens; foutrapportage. Zie afbeelding 12.

Gegevensinvoer. De beschreven methode van datakwaliteitscontrole gaat uit van tabelstructuren in een Oracle database. Dit kunnen feitelijk tabellen, views of externe tabellen zijn zolang de tabel- en kolomnamen maar op te halen zijn uit de databasecatalogus.

De datakwaliteitscontroles moeten plaatsvinden op de te controleren gegevens, voor de verdere verwerking gaat plaatsvinden. De datakwaliteitscontrole is de eerste stap van verwerking. Vaak zullen de controle-views worden geplaatst op externe tabellen die worden gebruikt voor het inlezen van gegevens in de database.

Filteren van gegevens. Afhankelijk van het ETL-proces zullen in meer of mindere mate opgeschoonde gegevens nodig zijn voor verdere verwerking. Het kan zijn dat alleen de foutloze rijen mogen worden ingelezen, maar ook dat de foutieve kolommen leeg moeten worden gelaten.

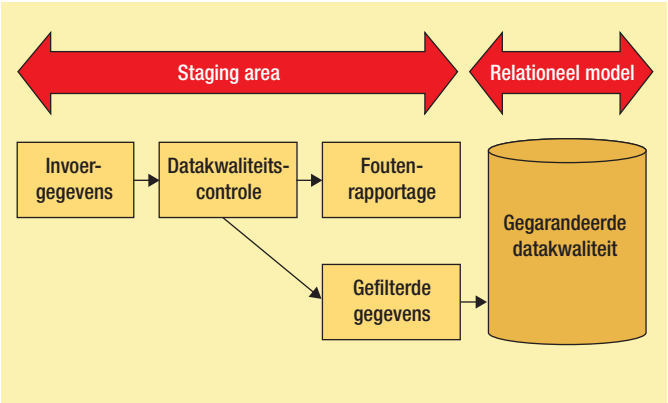
Omdat de gedemonstreerde controle-view voor elke kolom een controlekolom bevat, zijn de foutieve veldwaarden en rijen gemakkelijk op te halen. Daarbij maken we gebruik van een tweede SQL statement dat geconstrueerd wordt aan de hand van de databasecatalogus. Door standaardnaamgeving van controle views en kolommen zijn de namen hiervan gemakkelijk uit de databasecatalogus op te halen. Zo kan een SQL statement worden samengesteld dat bijvoorbeeld alleen de rijen selecteert waarvoor alle controlekolommen 0 zijn (geen foutieve waarde). Ook kunnen

Rij	Werknemersnummer		Naam		Leidinggevende	
	Waarde	Controlewaarde	Waarde	Controlewaarde	Waarde	Controlewaarde
1	7369	0	SMITH	0		0
2	7499	0	JONES	0	7369	0
3	7499	10		1	9999	100
4	7902	0	FORD	0	9999	110

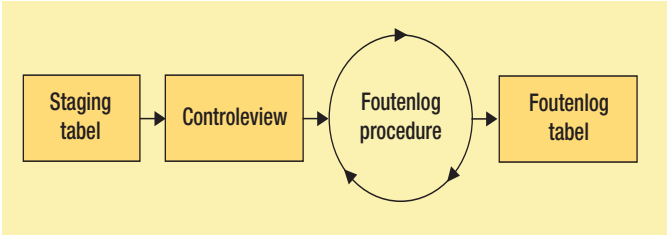
Afbeelding 11: Controleresultaat externe tabel 'Medewerkers'.

foutieve veldwaarden met behulp van de corresponderende controlekolom leeg worden gemaakt. De manier van filtering kan geheel worden aangepast aan de behoeften van het ETL-proces.

Foutrapportage. Voor de rijen en velden waarvoor niet aan de controles is voldaan is in veel gevallen een foutrapportage vereist. Hiervoor moet de informatie over fouten uit de controlekolommen wordt getransformeerd naar een genormaliseerde log-tabel met een standaardformaat. Hierin komen velden als tabelnaam, kolomnaam, kolomwaarde en geconstateerde fout voor. Om deze velden te vullen maken we gebruik van een procedure die voor elke veld de controle-view doorloopt en de gegevens van de



Afbeelding 12: Inpassing datakwaliteitscontroles in een ETL-applicatie.



Afbeelding 13: Wegschrijven geconstateerde fouten naar de foutentabel; de fouten log-procedure schrijft de geconstateerde fouten per kolom weg.

controlekolom transformeert en wegschrijft naar een log-tabel, zie afbeelding 13. Dit betekent wel dat de controle-view voor elke kolom een keer wordt uitgelezen. De performance-bezwaren hiervan kunnen worden weggenomen door de controle-view als materialized view vorm te geven.

Samenvatting

In dit artikel is een methode besproken om datakwaliteitscontroles te implementeren in een Oracle-omgeving. Deze methode maakt slechts gebruik van features die in elke Oracle database beschikbaar zijn. Voor wie over een Oracle database beschikt is het daardoor met deze methode mogelijk om datakwaliteit te waarborgen, zonder de inzet van ETL-tools of datakwaliteits-tools.

Reinbert Hamstra (Reinbert.Hamstra@AtosOrigin.com) is Consultant Datawarehousing bij Atos Origin.