Hoe Microsoft Slowly Changing Dimensions ondersteunt

Waar DTS ophoudt gaat SSIS 2005 verder

Paul Hover

Met de komst van SQL Server 2005 Integration Services (kortweg SSIS) heeft Microsoft geprobeerd het laden van dimensionele data in een datawarehouse wat te vereenvoudigen.

In bijna elk datawarehouse worden voorzieningen geboden voor slowly changing dimensions, hetgeen met de vorige versie van SQL Server telkens van de grond af aan opgebouwd moest worden. Met name de data-integratie-tool in SQL Server 2000, genaamd Data Transformation Services (DTS), liet op dat vlak veel te wensen over.

Verbeteringen

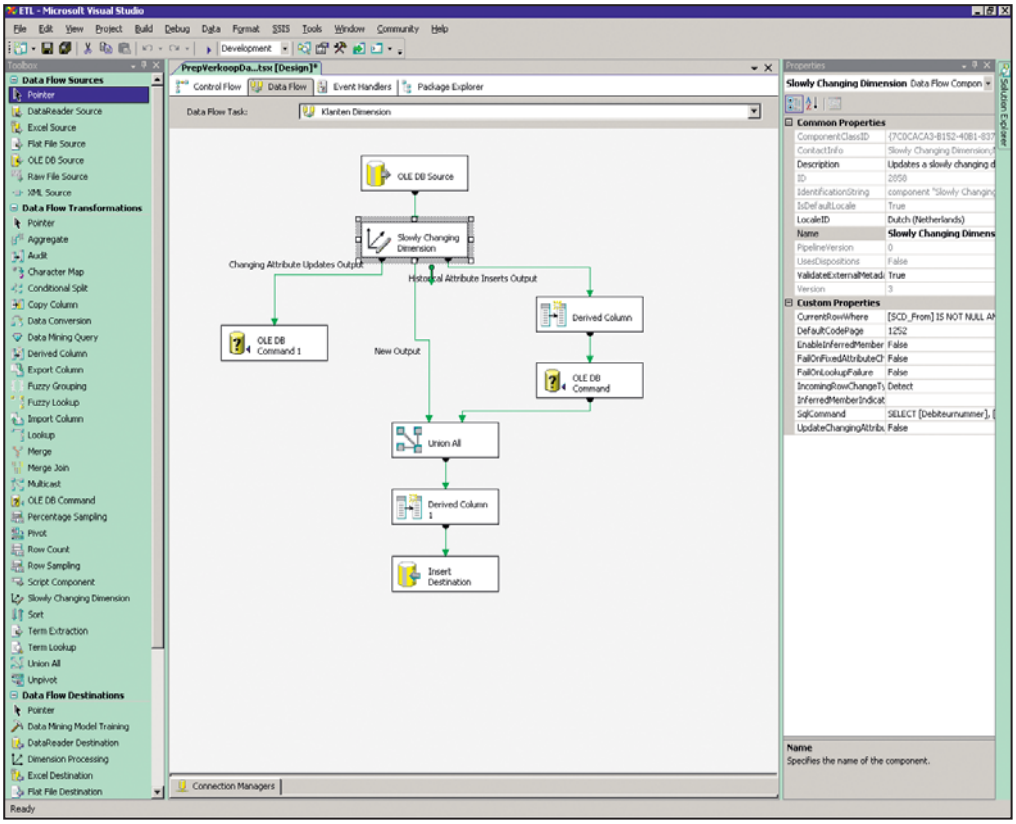
DTS is een tool waarmee data uit de ene bron kunnen worden overgezet naar een andere bron. Deze bronnen kunnen variëren van tekstbestanden, Excel- en Access-bestanden tot RDBMS'en als Oracle en natuurlijk SQL Server. Tijdens het overzetten van de data kunnen er door middel van scripts allerlei transformaties plaatsvinden. Deze Extract, Transform en Load (ETL) handelingen vormen een belangrijk onderdeel van een datawarehouse-ontwerp. De definitie van de databronnen, de datastromen en de transformatielogica worden in DTS opgeslagen in zogenaamde packages. Deze packages kunnen zowel binnen SQL Server als op een filesystem worden opgeslagen.

> Het daadwerkelijk overpompen en transformeren van data vindt plaats in de Data Flow

Al het bovenstaande geldt ook voor SSIS en ook hier wordt de definitie van de datastroom met alles wat daar verder bij komt kijken opgeslagen in packages. Waar DTS ophoudt gaat SSIS verder en biedt het een aantal (flinke) voordelen. Allereerst zijn de performance en schaalbaarheid sterk verbeterd, maar met name het ontwerpen van packages biedt nieuwe mogelijkheden. Zo is voorzien in een echte ontwikkelomgeving genaamd Business Intelligence Development Studio (kortweg BIDS). BIDS is in werkelijkheid gewoon Visual Studio en daarmee hebben we onder andere ook meteen de beschikking over goede mogelijkheden voor debugging en source control.
Tijdens het ontwerp van SSIS-packages in BIDS zijn het proces en de transformaties strikt van elkaar gescheiden. Het verloop van het proces en de samenhang tussen de verschillende componenten worden ondergebracht in de Control Flow. Het daadwerkelijk overpompen en transformeren van de data vindt plaats in de Data Flow. Zo blijft het ontwerp van het package overzichtelijk en kan bepaalde logica optimaal hergebruikt worden. En waar in DTS bijna alle transformaties moeten worden uitgevoerd met scripts kan in SSIS gebruik gemaakt worden van vele standaard Data Flow componenten, waarmee een grote diversiteit aan transformaties eenvoudig is te realiseren. Zo zijn er bijvoorbeeld componenten voor lookups, sorteren, het splitsen van de datastroom en het definiëren van afgeleide kolommen. Ondersteuning voor scripts is uiteraard nog steeds aanwezig. Het zelf maken van componenten is relatief eenvoudig indien er wordt beschikt over .Net programmeerkennis. Voorbeelden en sjablonen worden meegeleverd voor zowel Visual Basic als C#. De component waar ik hier wat dieper op wil ingaan betreft de Slowly Changing Dimension transformatie.

SCD-ondersteuning in SSIS

Op het eerste gezicht lijkt de SCD-transformatie sterk op de andere transformaties die we in een dataflow kunnen gebruiken. Vanuit de toolbox binnen BIDS wordt de SCD-component binnen onze dataflow gesleept en wordt deze gekoppeld aan een dataflow source component. Na dubbelklikken op de SCD-component wordt een wizard gestart die ons door het hele configuratieproces leidt.
In de eerste stap van deze wizard wordt gevraagd om een dimensietabel te selecteren uit de lijst van beschikbare tabellen in de gekozen databron. Tevens wordt gevraagd wat de business key is. Dit is de key zoals deze gebruikt wordt in het bron-systeem. Er mag slechts één business key worden aangewezen. Het datawarehouse gebruikt eigen sleutelvelden.
Op het volgende scherm dient minimaal één attribuut te worden aangewezen dat door de SCD-transformatie in de gaten gehouden moet worden (afbeelding 3). Per attribuut moet worden aangegeven hoe die controle moet plaatsvinden. 'Fixed' betekent dat het attribuut niet meer gewijzigd mag wor-



Afbeelding 1: Componenten in Visual Studio.

den. Op het volgende scherm kan worden aangegeven of het detecteren van wijzigingen in 'fixed' attributen moet leiden tot het afbreken van de transformatie. Deze optie is alleen beschikbaar indien er attributen zijn aangemerkt als fixed. 'Changing' houdt in dat wijzigingen in het attribuut worden behandeld als Type 1. Wijzigingen in de brondata zullen worden doorgevoerd in de doeltabel, maar zonder historie. 'Historical' doet hetzelfde maar dan als Type 2 wijziging. Er kunnen meerdere attributen worden aangemerkt als historical en bij een gedetecteerde wijziging in één daarvan zal een nieuw record worden aangemaakt.

> Er kunnen meerdere attributen worden aangemerkt als historical

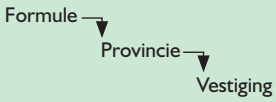
Op het volgende scherm (afbeelding 4) moet worden aangegeven hoe Type 2 wijzigingen worden bijgehouden. Dit kan met een enkele kolom die aangeeft wat het actuele record is, maar een betere oplossing is om start- en einddatum te gebruiken. Hiermee kunnen we te allen tijde exact reproduceren welk record bij een bepaald feit hoort uit onze feitentabel. Deze kolommen moeten wel al aanwezig zijn in de doeltabel.
De dropdown-lijstjes voor Start Date en End Date bevatten ook alleen kolommen van het type datetime. Een kolom kan ook alleen voor één van beide worden gebruikt, dus om van deze optie gebruik te kunnen maken dienen er twee kolommen voor dit doel aanwezig te zijn. De laatste dropdown biedt de mogelijkheid om aan te geven met welke datum/tijd de kolommen een update moeten krijgen. Het ligt meestal voor de hand om ContainerStartTime te gebruiken, aangezien dit de meest nauwkeurige benadering van het huidige tijdstip is. StartTime is minder geschikt, omdat dit de aanvangstijd van het package is en er hier voor misschien nog enkele (tijdrovende) taken zijn uitgevoerd.

Inferred

Als laatste volgt nog een vraag over 'Inferred Member Support'. Hiermee kunnen we een voorziening treffen voor feiten in onze feitentabel waaraan nog geen records uit onze dimensietabel gekoppeld zijn, wat bijvoorbeeld het geval kan zijn wanneer de feitentabel gevuld wordt voordat de dimensietabel (volledig) gevuld is. Indien van deze optie gebruik gemaakt wordt, dan zal tijdens de transformatie gekeken worden of er bestaande records in de doeltabel zitten, waarvan de business key overeenkomt met een (completer) record in de bron. Deze records worden behandeld als een Type 1 wijziging en krijgen een update met de nieuwe informatie. Het markeren van records als 'inferred' kan gebeuren met een aparte (boolean) statuskolom of door de attribuutwaarden leeg te laten.

Slowly Changing Dimensions in het kort

In een datawarehouse kan het voorkomen dat wijzigingen in de brondata leiden tot ongewenste resultaten. Beschouw eens het volgende (bekende) voorbeeld. In het datawarehouse wordt informatie opgeslagen over omzetcijfers van supermarkten. Voor de supermarkten is een klantdimensie ontworpen met een aantal niveau's. Als bovenste niveau heeft men de supermarktformule gedefinieerd (bijvoorbeeld C1000, Jumbo, enzovoort). Daaronder heeft men een geografisch niveau gedefinieerd in de vorm van provincie. Als laagste niveau bevat de dimensie de supermarktvestigingen.

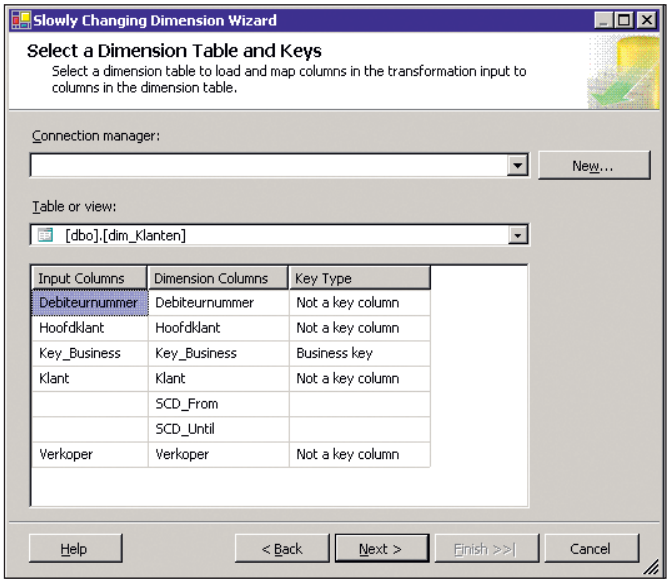


De data voor deze dimensie worden geladen vanuit het bronsysteem. Als we geen historie vasthouden van deze data, dan kan een aantal ongewenste effecten optreden. Als een supermarktondernemer besluit om van formule te wisselen, dan zou zijn historische omzet na de eerstvolgende update van het datawarehouse plotseling onder de nieuwe formule vallen, hetgeen uiteraard niet de bedoeling is.
Om aan dit soort (en andere) problemen tegemoet te komen is het principe van Slowly Changing Dimensions (SCD) bedacht. Daar onderscheiden we inmiddels een aantal types in, waarvan de meestgebruikte zijn:

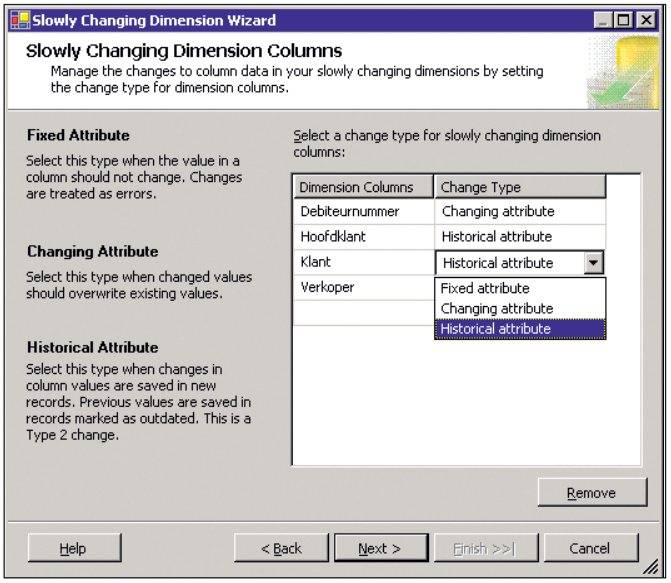
- Geen SCD: bij iedere verwerking wordt de volledige dimensie opnieuw opgebouwd
- Type 1: gewijzigde records krijgen een update en nieuwe worden toegevoegd, maar er wordt geen historie vastgehouden;
- Type 2: bij een wijziging wordt een nieuw record aangemaakt, waarbij ieder record wordt voorzien van een start- en eindtijdstip, zodat achteraf exact is te reconstrueren welk record op een bepaald moment van toepassing was;
- Type 3: voor de gewenste attributen wordt een extra kolom toegevoegd, zodat de huidige en de vorige staat van het attribuut worden bijgehouden (meerdere wijzigingen van hetzelfde attribuut worden niet bijgehouden).

In de praktijk worden vooral Type 1 en Type 2 toegepast (soms gecombineerd). Type 1 biedt de beste performance, terwijl Type 2 volledige historie vasthoudt. Als we in ons voorbeeld gebruik maken van Type 2, dan zou bij het raadplegen van de cijfers de omzet van de betreffende supermarkt correct verdeeld worden over de twee winkelformules.

Na een overzichtscherm kunnen we de wizard afsluiten en maakt de wizard een vrij complex uitziende set van dataflow-componenten aan. Deze componenten behandelen in een aantal verschillende takken alle opties zoals we die in de wizard hebben aangegeven. Zo wordt er voor de changing attributen een tak aangemaakt met een OLE DB Command waarin een SQL Update statement staat. Voor historical attributen worden twee takken aangemaakt: één voor updates van bestaande records (met een afgeleide kolom voor het bepalen van de huidige datum/tijd) en één voor nieuwe records. De nieuwe records en de nieuwe versies van de bestaande records worden daarna samengevoegd en toegevoegd aan de doeltabel.



Afbeelding 2: Wizard met selectie van dimensietabel.

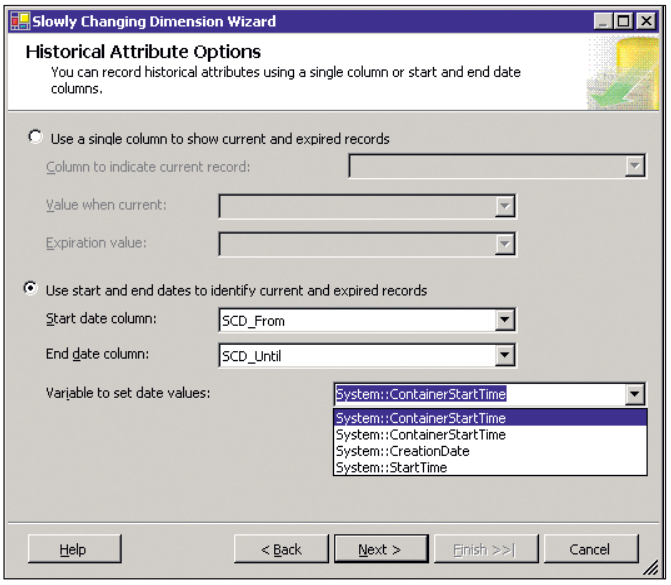


Afbeelding 3: Wizard met attribuutaanwijzing.

Deze set van dataflow-componenten kan naar eigen behoefte worden aangepast en uitgebreid, maar hiermee vervalt de mogelijkheid om de wizard opnieuw te gebruiken met behoud van de eerdere keuzes. Door dubbel te klikken op de SCD-transformatie wordt de wizard opnieuw gestart met de eerder gemaakte keuzes al ingevuld, maar let op. De handmatige wijzingen en toevoegingen in de gegenereerde flow worden hierdoor weer overschreven! En aangezien de wizard een flow creëert waarin er uitsluitend gebruik gemaakt wordt van standaard componenten is het ook prima mogelijk om zelf een complete SCD-flow te bouwen, maar door dit te doen gaat wel een belangrijk voordeel van SSIS boven DTS verloren.

Tot slot

Ondanks alle verbeteringen is ook SSIS geen perfect product.



Afbeelding 4: Wizard met Type 2 wijzigingen.

Zo is het onbegrijpelijk dat voor het schrijven van scripts alleen gebruik gemaakt kan worden van Visual Basic. Met de nauwe .Net integratie zou men verwachten dat hier ook C# gebruikt zou kunnen worden. De dropdown box waar de gewenste scripttaal gekozen kan worden bevat vreemd genoeg slechts één keuze. Ook de knullige editor waarin expressies moeten worden geschreven laat veel te wensen over. Met name bij syntaxfouten in expressies is vaak lastig te achterhalen waar het probleem zit. Daarnaast treden soms rare bugs op met data viewers en raken af en toe opgeslagen packages corrupt. Hopelijk zullen deze punten worden aangepakt in de nieuwe versie die volgend jaar uitgebracht zal worden. Een aangekondigde verbetering in SQL Server 2008 met betrekking tot SCD's is het gebruik van zogenaamde *persistent lookups*. Dit zorgt ervoor dat het opzoeken van business keys in relatie tot grote dimensietabellen een stuk sneller gaat.
Hopelijk geeft deze uiteenzetting van SSIS en de nieuwe Slowly Changing Dimension component een beeld van hoe het leven van een ETL-developer weer wat makkelijker gemaakt is.

Paul Hover MCT en MCITP BI (paul.hover@atosorigin.com) is Business Intelligence Consultant bij Atos Origin.

Kimball

De theorie van de slowly changing dimensions is opgesteld door de Amerikaanse datawarehouse-deskundige Ralph Kimball. Hij heeft meerdere toonaangevende boeken geschreven over het ontwerp van datawarehouses en houdt zich al sinds 1982 bezig met het onderwerp. Tegenwoordig heeft hij een eigen consultancybureau en verzorgt hij lezingen over de hele wereld. Hij wordt door velen gezien als *de* autoriteit op dit gebied.