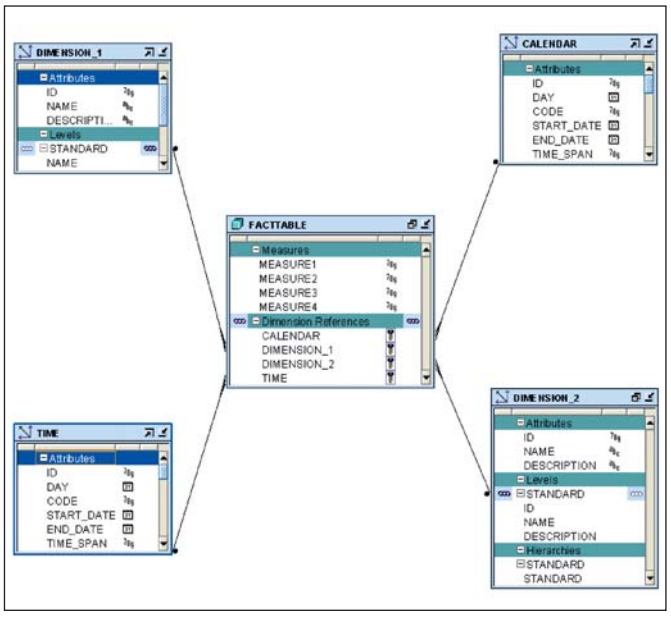
Normalisatie van tijdelement verkleint de overhead

Timestamps in het datawarehouse

René Kuipers

Veel bedrijfsgegevens zijn tijdgerelateerd, misschien zelfs alle. Een accurate bepaling van het moment van een gebeurtenis kan bedrijfskritisch zijn. Niet alleen voor rapportages, maar niet zelden ook in juridisch opzicht. In een compliant omgeving is het zelfs verplicht om gegevens van juiste timestamping te voorzien.

Timestamps worden nogal eens verward met Datum en Tijd. Maar dit is niet hetzelfde. Een timestamp verwijst ook naar het gegeven of de dag een feestdag is, of het vóór of juist ná de middag was. Of iets in een weekend gebeurde etcetera. Er zijn veel meer afleidingen mogelijk van een datum dan alleen maar de datum. Deze extra attributen zijn van bijzonder belang voor rapportages, omdat ze extra inzicht verschaffen.
Timestamps verwijzen ook naar een tijdstip, als in een deel van 24 uur. Een dag bestaat uit 86400 seconden. Het opslaan van meetgegevens met een juiste timestamp en op de juiste manier uitgevoerd, is cruciaal voor het verkrijgen van inzicht in business processen en verschaft een helder en historisch correct beeld van business data en informatie.



Afbeelding 1: Een Fact-tabel met vier dimensies.

Datawarehouse-architecten, -ontwerpers en -ontwikkelaars moeten van deze noodzaak doordrongen zijn. Daarnaast is een goed ontwerp beter te beheren en te managen. Tevens komt het ten goede aan de performance van het overall datawarehouse-systeem. Dit artikel beschrijft een werkbare en bewezen methode om met datums, tijden en timestamps om te gaan in een datawarehouse-omgeving.

Het accuraat opslaan van datum- en tijdgegevens is van vitaal belang voor het juist rapporteren binnen een datawarehouse-omgeving. Regelmatig liggen slechte ontwerpbeslissingen aan de basis van complexe structuren van vaak meerdere datum-dimensies die niet alleen breed (veel kolommen) maar ook lang (veel rijen) zijn. Deze combinatie leidt vaak tot slechte performance van de rapportage-omgeving en frustratie bij de systeem-gebruikers vanwege de gebruikersonvriendelijkheid: zij moeten zelf de kennis hebben om te bepalen welke dimensie moet worden gebruikt. Dat is niet iets voor de eindgebruiker: zij houden zich bezig met business-termen, niet met dimensies. Het zijn de datawarehouse-ontwerpers die hier met een goed ontwerp (zowel functioneel als technisch) dienen te komen.

Praktijkvoorbeeld

In het datawarehouse van een organisatie zijn meetgegevens opgeslagen. Het gaat hier om vitale informatie omtrent de performance van verkochte apparatuur en de vraag of aan contractuele afspraken met klanten wordt voldaan. Het is van groot belang om incidenten op de apparatuur tot op het laagst mogelijke niveau te traceren en te analyseren. De resolutie van de meetgegevens is één seconde.
Om het complex te maken zijn er meerdere kalenders in het spel. Er worden ook fysiek meerdere kalenderdimensies gebouwd: één voor de ISO-kalender, één voor de standaard-kalender etcetera. Dit alles omdat de rapportagebehoefte van de eindgebruikers evolueert over de tijd. Binnen deze organisatie heeft de datawarehouse-ontwerper ervoor gekozen om meerdere kalenderdimensies te bouwen: één voor elk (functioneel) kalendertype. Elke kalenderdimensie wordt gebouwd op het laagste niveau (1 seconde) en is voor 10 jaar data voorbereid.
De effecten van deze ontwerpbeslissing waren dramatisch:
10 jaar x 365,25 dagen x 24 uur x 60 minuten x 60 seconden = 315.576.000 records.

Dat zijn driehondervijftienmiljoenvijfhonderdzesenzeventigduizend records! Per kalenderdimensie meer dan 315 miljoen records; dat is best veel. Met drie kalenderdimensies zit je dan op bijna 1 miljard records. Dit is een typisch voorbeeld van een verkeerde ontwerpbeslissing. Waarom niet één enkele, brede kalenderdimensie en een aparte tijddimensie? Dit is voordelig voor zowel het aantal records (en dus query performance) alsook voor de ontwikkeltijd om aanpassingen door te voeren.

Het probleem

Elke fact-record is gelinkt aan één of meerdere dimensies, afbeelding 1 illustreert dit. Als er meerdere datumdimensies zijn bevat elk fact-record meerdere referenties naar datumgerelateerde dimensies: één voor de klantendimensie, één voor de productdimensie, enzovoort. Wanneer er meerdere kalenderdimensies zijn bevat elk fact-record net zoveel relaties met kalenderdimensies als er kalenderdimensies zijn. Dit betekent ook dat wanneer er een kalenderdimensie bijkomt (omdat er binnen de bedrijfsdefinitie een nieuw kalendertype is bedacht, bijvoorbeeld: productiekalender, life cycle kalender), elke fact-tabel moet worden uitgebreid met (een) extra kolom(men). Deze kolommen (ook de historie) moeten in alle fact-tabellen worden gevuld met data en het bestaande ETL-proces voor het laden van alle fact-tabellen moet worden aangepast om in de toekomst deze nieuwe kolommen ook te laden.
Kortom, er is veel ontwikkel- en testtijd nodig vanwege de introductie van een nieuw kalendertype. Dit is een langdurig proces dat erg gevoelig is voor fouten terwijl de business vraag zo eenvoudig is: 'Ik wil dit kalendertype kunnen gebruiken.'

De oplossing

Uiteindelijk hebben alle attributen van elke kalenderdimensie betrekking op hetzelfde unieke geïdentificeerde gegeven: een seconde op een bepaalde dag. Een betere manier om dit aan te pakken dan de eerdere geschetste aanpak bestaat uit twee punten:

1. Verbreed de *enige* kalenderdimensie met extra attributen die van toepassing zijn op het nieuwe kalendertype en voeg alle attributen van de bestaande dimensies bij elkaar;
2. Splits de tijdcomponent af van de datumcomponent en bouw hiervoor een zelfstandige tijddimensie (normaliseer het tijdelement).

Datum	Weekdag	Dag	Maand	Jaar	Weekend_indicator
1 Januari 2007	Maandag	1	Januari	2007	Nee
2 Januari 2007	Dinsdag	2	Januari	2007	Nee
3 Januari 2007	Woensdag	3	Januari	2007	Nee
4 Januari 2007	Donderdag	4	Januari	2007	Nee
5 Januari 2007	Vrijdag	5	Januari	2007	Nee
6 Januari 2007	Zaterdag	6	Januari	2007	Ja
7 Januari 2007	Zondag	7	Januari	2007	Ja
...	...	...	...	...	...

Afbeelding 2: Voorbeeld van een kalenderdimensie.

Het gevolg van deze aanpak zal zijn dat elk fact-record twee verwijzingen heeft naar timestamp-gerelateerde dimensies: één voor de datum en één voor de tijd (in het voorbeeld had elk fact-record er drie vanwege het aantal kalenderdimensies).
Ook datamodel-technisch is dit een goede en flexibele aanpak: voor toekomstige kalenderdefinities volstaat het om de bestaande kalenderdimensie uit te breiden met de benodigde attributen en deze beschikbaar te stellen aan eindgebruikers, zodat zij ze kunnen gebruiken in de rapportages. Er zijn geen aanpassingen nodig aan bestaande fact-tabellen in de ETL om deze te laden enzovoort.

Dimensie-ontwerp: de standaard kalender

Het is binnen een organisatie lang niet altijd helder wat 'de' kalender is. Het is heel gebruikelijk dat er meerdere kalenders bestaan: de normale (Juliaanse) kalender; ISO-kalender; productiekalender; fiscale kalender; zelf ontworpen kalenders (branchespecifiek/bedrijfsspecifiek).
Een gangbare kalenderdimensie wordt vaak Time-dimension genoemd, wat eigenlijk onjuist is. Het bevat datums, geen tijdstippen. De benaming Calendar-dimension is beter. Het laagste niveau in de kalenderdimensie is een dag. De overige attributen kunnen naar believen worden gevuld. Veel database-leveranciers hebben scripts om een dergelijke dimensie aan te maken. Zie afbeelding 2. Vanwege het 'standaard' karakter van zo'n dimensie kan het zijn dat niet alle attributen van toepassing zijn,

Datum	Weekdag	Dag	Maand	Jaar	Weekend_indicator	ISOWeeknummer
1 Januari 2007	Maandag	1	Januari	2007	Nee	200701
2 Januari 2007	Dinsdag	2	Januari	2007	Nee	200701
3 Januari 2007	Woensdag	3	Januari	2007	Nee	200701
4 Januari 2007	Donderdag	4	Januari	2007	Nee	200701
5 Januari 2007	Vrijdag	5	Januari	2007	Nee	200701
6 Januari 2007	Zaterdag	6	Januari	2007	Ja	200701
7 Januari 2007	Zondag	7	Januari	2007	Ja	200701
8 Januari 2007	Maandag	8	Januari	2007	Nee	200702
...	...	...	...	...	...	...

Afbeelding 3: De kalenderdimensie uitgebreid met een attribuut ISO-Weeknummer.

of dat benodigde attributen niet aanwezig zijn. Daarom is het wenselijk om zelf met de kalenderdimensie(s) aan de slag te gaan.

Dimensie-ontwerp: de ISO-kalender

Een datum valt in een bepaalde week. De internationale standaardenorganisatie ISO biedt regels om te bepalen in welke ISO-week een datum valt. Deze ISO-weeknummers kunnen afwijken van de weeknummers zoals reguliere kalenders ze opleveren. Internationaal zijn ISO-weeknummers altijd gelijk voor een bepaalde datum.

Veel RDBMS-leveranciers hebben standaard functies om de ISO-week van een bepaalde datum te bepalen.
Zo is het in een Oracle-omgeving mogelijk om met behulp van onderstaand statement de ISO-week voor een datum te bepalen:

```
SQL> column ISOWEEK format a8
SQL> select to_char(to_date('01-01-2007',
         'dd-mm-yyyy'),'iyyyiw') ISOWEEK from dual;

ISOWEEK
--------
200701
```

Zo valt 1 Januari 2007 in ISO-week 200701. ISO heeft de regels en definities opgesteld hoe een ISO-weeknummer te bepalen. Dit kan soms vreemde effecten hebben. Bijvoorbeeld in het geval van de ISO-Week van 1 Januari 2006:

```
SQL> column ISOWEEK format a8
SQL> select to_char(to_date('01-01-2006',
         'dd-mm-yyyy'),'iyyyiw') ISOWEEK from dual;

ISOWEEK
--------
200552
```

1 Januari 2006 valt in ISO-week 200552. Dus, als ISO-weekrapportage van belang is voor de business, moet de kalenderdimensie deze manier van rapporteren ondersteunen.
Aangezien de ISO-week een attribuut is van een dag, en dus van een bepaalde datum, kan het als extra kolom aan de reeds bestaande datumdimensie worden toegevoegd. Deze ziet er dan uit zoals in afbeelding 3.
Van elke datum in de kalenderdimensie wordt het ISO-week-attribuut toegevoegd. Op deze manier is het eenvoudig om te rapporteren op ISO-week, naast de reeds bestaande attributen van de datum.

Dimensie-ontwerp: de Zelf-Ontworpen-Kalender

Zoals eerder aangetoond kunnen alle attributen die betrekking hebben op één dag worden opgeslagen in de standaard kalenderdimensie. In het geval van bedrijfsspecifieke kalenders is dit ook het geval, bijvoorbeeld een productiekalender. Deze zou kunnen beschrijven dat de eerste dag van een maand altijd in een nieuwe productieweek valt. Voor de rest begint week 1 van de productiekalender altijd op 1 januari, ongeacht de dag van de week. Aangezien de definitie is gebouwd op de granule van 1 dag en omdat dit ook de granule van de standaard kalenderdimensie is, kunnen de attributen van de productiekalender worden toegevoegd aan die kalenderdimensie. Als de logica van het bepalen van de productieweek attributen helder is, kan de kalenderdimensie met de volgende kolommen worden uitgebreid. Dit alles heeft nog steeds geen impact op de fact-tabellen en is dus snel en beheerst te realiseren, zie afbeelding 4. Op deze manier kunnen de drie attributen van de productiekalender aan de bestaande kalenderdimensie worden toegevoegd, aangezien ze ook naar de eigenschappen van een dag verwijzen. Hierdoor wordt de kalenderdimensie breder.

Dimensies splitsen: normaliseer het tijdelement

Hiervoor is de situatie geschetst waarin de tijd een component was in de kalenderdimensie. Ook werd aangetoond dat deze wijze van opslag tot een enorm aantal records in de kalenderdimensie kan leiden. Het is daarom beter om de tijd (aangezien deze voor iedere dag hetzelfde is) te normaliseren en hiervoor een eigen dimensie te ontwerpen. Immers, een tijdstip heeft ook een aantal tijdspecifieke attributen ('ochtend', 'tijdens werkuren') Hieronder volgt een calculatievoorbeeld.
Tien jaar kalenderdata op het detail van een seconde leveren meer dan 315 miljoen records op in de kalenderdimensie. Wanneer het tijdelement wordt afgesplitst, bevat de kalenderdimensie nog maar 10 x 365,25 = 3653 records. De tijddimensie bevat nog slechts 86400 records. Beide zijn significant kleiner dan de gecombineerde dimensie. Dit zal een positief effect hebben op de query performance. Daarnaast is het beheer op twee kleine tabellen eenvoudiger dan het beheer op een enorme tabel. Ze nemen ook minder opslagruimte in beslag. Verder bevat elk fact-record te allen tijde slechts twee referenties naar datum- en tijddimensies. Ook dit is voordelig ten opzichte van de situatie dat er voor elk kalendertype een aparte dimensie bestaat.

Datum	Productie-weeknr	Dag van productieweek	Productie-kalenderkwartaal
1 Januari 2007	1	1	Q1
2 Januari 2007	1	2	Q1
3 Januari 2007	1	3	Q1
4 Januari 2007	1	4	Q1
5 Januari 2007	1	5	Q1
6 Januari 2007	1	6	Q1
7 Januari 2007	1	7	Q1
8 Januari 2007	2	1	Q1
...	...	...	...

Afbeelding 4: Attributen van de productiekalender toegevoegd aan de bestaande kalenderdimensie.

> Van elke datum in de kalenderdimensie wordt het ISO-weekattribuut toegevoegd

Een derde voordeel ligt in het feit dat aggregaties op de feittabellen die slechts betrekking hebben op dagen, weken of maanden nu onafhankelijk van het tijdelement kunnen plaatsvinden, omdat deze niet van belang is in de aggregatie.

Een voorbeeld

Stel, we berekenen de som van een meting gegroepeerd per ISO-Week. In het oorspronkelijke scenario (datum en tijd in 1 dimensie) betekent 1 week het aantal van 604.800 records. (7x24x60x60). Dit aantal records moet worden gescand voor hun technische sleutel voor de join met de fact-tabel. En dit aantal geldt voor elke week in de aggregatie. In de genormaliseerde situatie, waarbij datum en tijd aparte dimensies zijn, beslaat 1 week 7 records. Het aantal op te halen ID's voor de join is een factor 86.400 per week kleiner. Dit zal zeker ten goede komen aan de join performance en dus de query performance.

Conclusie

Het uitbreiden van één enkele datumdimensie met attributen uit andere datumgerelateerde dimensies, in tegenstelling tot het hebben van meerdere datumgerelateerde dimensies, leidt tot een overzichtelijker, beheersbaarder en beter presterende rapportageomgeving die beter geaccepteerd wordt door eindgebruikers.
Er is niet meer werk nodig om nieuwe kalenderdefinities toe te voegen aan de bestaande, anders dan het uitbreiden en vullen van de reeds bestaande kalenderdimensie. ETL voor het laden van de fact-tabellen kan ongewijzigd blijven en ook de fysieke structuur hoeft niet te veranderen. Deze implementatie leidt tot kortere ontwikkeltijden. Het elimineren van meerdere dimensies die alle betrekking hebben op dezelfde granule, verkleint de hoeveelheid data waar men een query op moet doen en het is voordelig voor query performance.
Het normaliseren van het tijdelement in een eigen dimensie brengt het aantal records in de respectievelijke dimensies aanzienlijk terug en verkleint de overhead bij analyses die niet zozeer tijd- maar wel datumgerelateerd zijn, omdat het tijdelement geen rol speelt in deze aggregaties.

René Kuipers
Ing. R.J.L. Kuipers (rene.kuipers@ciber.nl) is senior BI-consultant en datawarehouse-architect bij CIBER.