Accumulating Snapshots – Mächtige aber herausfordernde Faktentabellen

Alexander Voigt
areto consulting gmbh
Köln

Schlüsselworte:

Data Warehouse, Faktentabellendesign, Accumulating Snapshots

Einleitung

Viele Geschäftsprozesse lassen sich als Serie von wohldefinierten Fortschritten beschreiben. Im Enterprise DWH finden wir diese Daten in transaktionalen Faktentabellen mit nur wenigen Modifikationen zur OLTP Quelle. Diese Form der Datenmodellierung ist für viele Analysezwecke gut geeignet, weist aber Schwächen bei der Auswertung von Kennzahlen auf, die sich aus der Korrelation von Daten zwischen den einzelnen Fortschritten ergeben, beispielsweise der Zeitdauer zwischen zwei Meilensteinen. Hier kann ein auf der transaktionalen Faktentabelle basierender Accumulating Snapshot helfen. Dieser verfolgt den Geschäftsprozess als Ganzes und stellt mit vorberechneten Kennzahlen zur Zeitdauer einzelner Abschnitte, aber auch mit Soll- Ist- Vergleichen, Analysepotential für das Messen der Geschäftsprozesseffizienz. Diese Form der Faktentabelle kann prinzipiell im Enterprise DWH, wie auch im Mart modelliert werden. Die Bereitstellung schon im DWH begünstigt jedoch die die Konsistenz und Performance darauf aufbauender Marts.

Situation

Faktentabellen in Form von Accumulating Snapshots sind in vielen Data Warehouse Umgebungen vorhanden. Unter den lokalen Faktentabellen sind sie dann aber selten. Das hat zur Folge, dass im Gegensatz zu häufiger modellierten transaktionalen Faktentabellen die Umsetzungen in ihrer Qualität und Leistungsfähigkeit bei Analysen breit gestreut sind. Beispiele sind Situationen, in denen der Schritt eine transaktionale Faktentabelle aufzubauen übersprungen worden ist und nur die schon korrelierten Daten zur Auswertung zur Verfügung gestellt werden. Oder die Daten werden teilkorreliert abgespeichert, in einer Form wo einzelne Meilensteine schon korreliert sind, für den Gesamtprozess jedoch noch immer korrelierte Subqueries benötigt werden. Insgesamt sind die ETL Zyklen für solche Faktentabellen auch häufig bei den Langläufern zu finden. Diese Situation ist beidseitig für Anwender und IT unbefriedigend.

Problemen begegnen

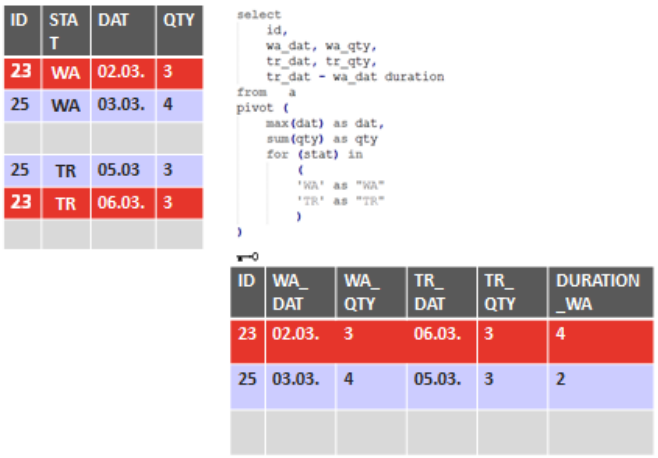
Der Sinn und Zweck eines Accumulating Snapshot steht und fällt zumeist mit der Fähigkeit einen Geschäftsprozess auf Datensatzebene auszuwerten und der Konstruktion auf Basis einer transaktionalen Faktentabelle im DWH. Bedingung eins wird häufig verletzt, weil der Gesamtprozess entweder sehr viele Teilschritte umfasst und oder nicht-lineare Zwischenschritte oder Probleme mit der Kardinalität zwischen Teilprozessen bestehen. Bei der Begegnung dieser Probleme ist es unumgänglich mit dem Fachbereich Definitionen für das Verhalten und für gültige Referenzen zu

finden. Springt ein Prozess im Meilenstein zurück, muss eine Definition dafür existieren, welches Datum die neue Referenz des nun aktiven Meilensteins oder bei Neubetreten des nächsten Abschnitts ist. Regeln, die vom Endanwender auch nachvollziehbar sind, kann und darf nur der Fachbereich liefern. Gibt es zu viele Subprozesse, würde die Aufnahme aller Prozessfortschritte zu einem sehr breiten Accumulating Snapshot führen, da jeder Meilenstein eigene Fremdschlüssel und Kennzahlen hinzufügt. Zusammenfassen von Teilprozessen kann hier helfen, ist aber wiederum nur in Abstimmung mit dem Fachbereich durchführbar. Bei Problemen mit Kardinalitäten ist es unter Umständen sinnvoller separate Snapshots aufzubauen oder Teilprozesse aggregiert auszuweisen. Sind diese Fragen geklärt, stellen sich bei der konkreten Entwicklung Fragen, bei denen sich der ETL Entwickler in den Freiheitsgraden der Performance, Konsistenz, Transparenz, Wartbarkeit und Flexibilität bewegt. Der Großteil des Vortrags behandelt diese Punkte, weil für die Makro-Fragestellungen schon Literatur existiert.

Standardfragen die zu Beantworten sind:

- Wie gestalte ich mit den zur Verfügung stehenden Mittels / Tools den Prozess der Korrelation möglichst performant
    - In Hinsicht auf die Gesamtmenge der Daten?
    - In Hinsicht auf Integration der Change Data?
- Wie erhalte ich mir Transparenz und Wartbarkeit der Prozesse?
- Bereitet man schon in den transaktionalen Fakten explizit die spätere Korrelation vor?

Als Grundregel gilt es OUTER JOIN soweit wie möglich zu vermeiden. Besteht der Input aus nur einer transaktionalen Faktentabelle bietet sich die Korrelation über Pivot an. Je nach verwendetem ETL oder ELT Tool sieht die Umsetzung unterschiedlich aus. Kann das ausgeführte SQL frei definiert werden, bietet der Einsatz der Pivot-Clause für transparenten Code an, während ansonsten ANSI Pivot über CASE WHEN benutzt werden kann. Diese Methode verhindert aufwändige JOINS der Faktentabelle mit sich selbst und unser SQL erfüllt noch immer die Voraussetzung für Partition Change Tracking. Selbst wenn der Accumulating Snapshot nicht als Materialized View realisiert wird, ist es empfehlenswert sich soweit wie möglich an diesen Kriterien zu orientieren, da die Anforderungen weitgehend kongruent mit den Prinzipien guten Faktentabellendesigns sind.



ID	STAT	DAT	QTY
23	WA	02.03.	3
25	WA	03.03.	4
25	TR	05.03	3
23	TR	06.03.	3

```
select
    id,
    wa_dat, wa_qty,
    tr_dat, tr_qty,
    tr_dat - wa_dat duration
from   a
pivot (
    max(dat) as dat,
    sum(qty) as qty
    for (stat) in
        (
        'WA' as "WA"
        'TR' as "TR"
        )
)
```

ID	WA_DAT	WA_QTY	TR_DAT	TR_QTY	DURATION_WA
23	02.03.	3	06.03.	3	4
25	03.03.	4	05.03.	3	2

Abb. 1: Korrelationen über verschiedene Wege bieten jeweils eigene Vor- und Nachteile.

Sind Quelldaten über mehrere Faktentabellen verteilt, funktioniert dieser Ansatz nicht mehr. Trotzdem gibt es Design-Möglichkeiten, die uns eine Vielzahl an Optionen offenlassen. Ein einfaches aber effektives Mittel ist das Erstellen einer Kopf-Tabelle / Process Hub. Diese enthält pro Geschäftsprozess-Instanz einen Datensatz samt UK und prozessübergreifende Informationen. Die Tabelle wird im späteren Verlauf für einen effektiven Drill-Through zwischen Teil-/Subprozessen verwendet, die in verschiedenen Faktentabellen verfolgt werden. Zusätzlich spielt die Tabelle beim Lookup von historischen Dimensionsschlüsseln eine entscheidende Rolle, gerade beim Aufbau der transaktionalen Faktentabellen. Hier kommen Anforderungen an die Darstellung nach historischer Korrektheit hinzu. Da der Gesamtprozess sich über eine Zeitdauer erstreckt, stellt sich die Frage, nach welchem Zeitbezug nun die historische Zuordnung von Dimensionsfremdschlüsseln geschehen soll. Wenn möglich, sollte dieses Datum bei Prozessbeginn bekannt sein und sich dann nicht mehr ändern. Der Grund ist der Wunsch die Anzahl zukünftig durchführender Updates gering zu halten in Verbund mit der Unmöglichkeit für alle, über einen Zeitraum verteilten, Subprozesse historisch korrekte Darstellung zu gewährleisten. Dieses Referenzdatum wird in die Kopf-Tabelle aufgenommen und spielt bei neu eintreffenden Informationen zum Prozess-Fortschritt für den Lookup der Dimensions-Keys eine Rolle. Vorbereitend für die spätere Korrelation wird in die transaktionale Faktentabelle nicht nur der historisch korrekte Fremdschlüssel zum Ausführungstag der Transaktion aufgenommen, sondern auch für das Bezugsdatum des Gesamtprozesses. Korreliert man nun die transaktionalen Faktentabellen sind schon die korrekten Fremdschlüssel zu den Dimensionen enthalten ohne zusätzlichen Aufwand zu produzieren.

Gleichzeitig ist die historisch korrekte Dimensionszuordnung in Bezug auf einen speziellen Meilenstein vorhanden. Diese Information kann noch aus den transaktionalen Faktentabellen gewonnen werden.

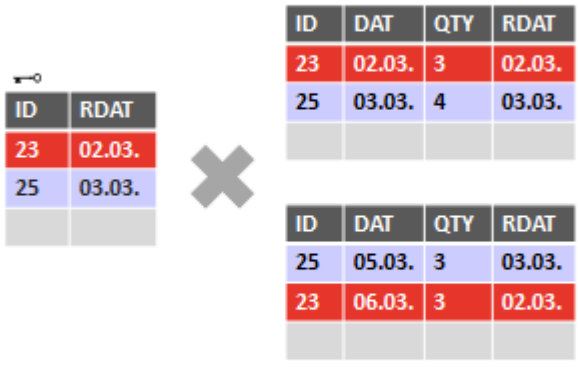


Abb. 2: Eine Kopf-Tabelle dient als Andockpunkt für benötigte Subprozesse und Subprozess-übergreifende Informationen

Ein weiterer Baustein ist die Partitionsstrategie. Ein sinnvoller Ansatzpunkt ist das Aufrechterhalten einer deterministischen Beziehung zwischen der Partition der Basistabelle und der Korrelationstabelle, wenn die Anforderungen dies Erlauben. Hier droht ein Trade-Off zwischen Abfragen auf die transaktionalen Faktentabellen und dem Refresh des Accumulating Snapshots. Der Refresh profitiert von prozessschrittübergreifender Referenz als Partitionskriterium, während Partition Pruning für Abfragen auf die Basistabellen gewünscht wird. Composite Partitioning ist hier in die Entscheidung mit einzubeziehen, wobei das Primärpartitionskriterium das prozessübergreifende Referenzdatum sein muss, um bei der Korrelation noch von Fast Refresh profitieren zu können.

Ein verbleibender Performance-Killer bleibt der in LEFT OUTER JOIN. Auch dieser hindert den Einsatz des effizienten Partition Change Tracking. Das Anlegen von Embryo Sätzen für noch nicht erreichte Meilensteine eines Prozesses kann auch dieses Hindernis entfernen.

Seit Oracle 12C existiert eine neue Variante des Fast Refresh, der Synchronous Refresh. Dieser ist ein technischer Mix aus Log-Based-Refresh, Partition Change Tracking und Partition Exchange Load. Die Anforderungen sind leicht erweitert, decken sich aber weitestgehend mit denen von PCT. Diese Refresh-Methode bietet noch einmal einen Performance-Gewinn gegenüber den üblichen Methoden und verschiebt somit die Balance zwischen dem DIY- Ansatz und Materialized View.

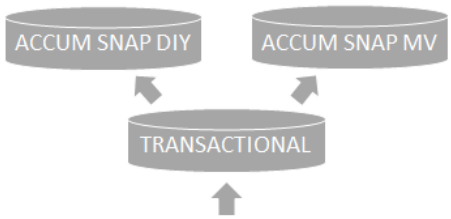


Abb. 3: In Oracle 12C verbesserte Fast Refresh Methoden machen die Realisation und Ausrichtung als/auf Materialized Views interessanter

Fazit

Transaktionale Faktentabellen komplexer Prozesse können häufig sinnvoll mit einem Accumulating Snapshot ergänzt werden und dadurch Effizienzauswertungen auf den Prozess erleichtern. Ein Accumulating Snapshot sollte eine Zeile und nur eine Zeile pro Prozess-Instanz haben und er sollte auf Basis der transaktionalen Faktentabelle gebildet werden, um Konsistenz in der weiteren Verwendung zu gewährleisten. Zum Beispiel wenn der Accumulating Snapshot schon im Enterprise DWH aufgebaut wird, so dass in den Marts die Daten standardisiert korreliert benutzt werde. Es ist nicht egal, wie die Korrelation der transaktionalen Daten vorgenommen wird. Simplicity is key! Dieser Schritt sollte schon mit Hinblick auf das Ziel geschehen. Das heißt, es sollte klar sein, wie der Load des Accumulating Snapshot geschehen soll und daraufhin optimiert werden. Die Partitionsstrategie gehört in dieses Thema hinein und unter Umständen gilt es hier Trade Offs zu bewerten.

Sind die Geschäftsprozesse nicht-linear oder gibt es Rücksprünge, muss fachlich deterministisch geklärt sein, wie damit zu verfahren ist. Insbesondere bei Rücksprüngen ist es wichtig, Regeln für das richtige Bezugsdatum der einzelnen Meilensteine zu haben. Oracle bietet insbesondere mit dem

Synchronous Refresh effiziente Methoden für schnelle Refresh-Zyklen bei Materialized Views an, so dass diese Form der Realisierung an Attraktivität gewonnen hat.

Kontaktadresse:

Alexander Voigt

areto consulting gmbh

Julius-Bau-Straße, 2

D51063 Köln

Telefon: +49 221 66 95 75-0

Fax: +49 221 66 95 75-99

E-Mail alexander.voigt@areto-consulting.de

Internet: http://www.areto-consulting.de