

Die Nadel im Heuhaufen

Oliver Lemm
MT AG
Ratingen

Schlüsselworte

APEX, PL/SQL, Reports, SQL.

Motivation

Bei großen Datenmodellen kann man schnell die Übersicht verlieren und wenn es dann noch darum geht zu erkennen wo sich Daten in der Zwischenzeit geändert haben wird es noch schwerer. Will man noch einen Schritt weiter gehen, dass nicht nur der DBA ermitteln kann wo sich Daten, wann und vom wem geändert haben wird es noch schwerer.

Genau diese Problematik soll gelöst werden. Ein Anwender soll kleinste Datenveränderungen in riesigen Datenmodellen binnen Sekunden ausgewertet bekommen. Eine solche Mechanik soll möglichst keinen Entwicklungsaufwand erzeugen, auch wenn das Datenmodell angepasst wird.

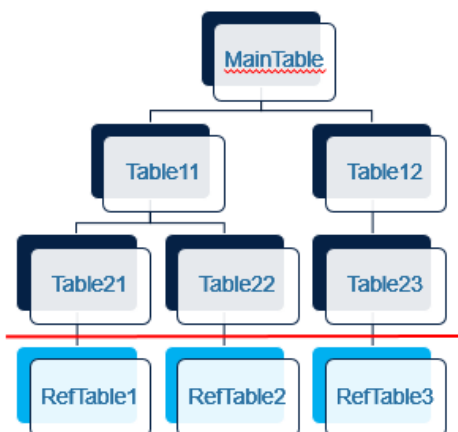
Insgesamt soll jede Änderung im Kontext der Anwendung dargestellt werden um Änderungen noch besser verständlich darzustellen.

Der Weg

Über die Zeit wurden verschiedene Ansätze zur Lösung des Problems eingesetzt, die im Einzelnen wie folgt aussahen:

Anfangs wurde eine komplett auf einem PL/SQL Package basierte Lösung entwickelt. Hier wurde gezielt pro Tabelle im Datenmodell eine Funktion im Package bereitgestellt. Eine solche Lösung war zwar einfach zu verstehen, aber aufwändig zu pflegen, da bei jeder Änderung im Datenmodell das Package nachzupflegen war. Zusätzlich wurde ein solches Package sehr groß und war bei vielen Datenbankänderungen, gerade am Anfang eines Projekts, sehr zeitintensiv.

Beim zweiten Ansatz wurde eine Lösung gewählt die neben einer Packagelogik Metadaten benötigte. Hierbei wurden im Datenmodell beim Erzeugen von neuen Datensätze eine Tabelle befüllt die jeweils die Information enthielt welche Datensätze aus welchen anderen Datensätze erzeugt wurden. Dadurch wurde die Vergleichsmechanik wesentlich schneller, aber es blieb das Problem, dass bei Daten die neu erstellt wurden und bei Daten aus Spalten die später hinzukamen, Vergleiche nicht möglich waren.



Im dritten Ansatz wurde dann die Entscheidung getroffen die Vergleiche nicht mehr mittels fest definierter Funktionen zu wählen, sondern mittels des Oracle Database Repositories die Struktur zu ermitteln. Hierbei wurden die Fremdschlüssel zum Auflösen der Verknüpfungen genutzt um die relevanten Tabellen zu identifizieren. Wichtig war dabei, dass die Referenzdaten nicht mit in die Vergleiche eingingen. Hierdurch wurde der Code enorm reduziert, aber es gab Fremdschlüssel die zirkuläre Abhängigkeiten definierten und immer wieder Anpassungen notwendig machten.

Abb. 1: Tabellenstruktur

Die Lösung

Der endgültige Ansatz, welcher die heutige Lösung darstellt, basiert auf Metadaten gepaart mit einer dynamischen Methodik die Werte im Datenmodell zu vergleichen. Es werden dabei Metadaten hinterlegt, welche die Tabellen und die relevanten Fremdschlüssel enthalten.

Weitere Metadaten werden definiert, welche beschreiben wo ein Feld in der Anwendung gepflegt wird und in welcher Tabelle in der Datenbank es hinterlegt ist.

Basierend auf diesen Metadaten wird pro Feld eine dynamische Abfrage gebildet und zweimal ausgeführt um die Werte in einer Datenversion und zur Vergleichsversion zu ermitteln.

Herausforderungen

Beim endgültigen Ansatz kam zuerst die Frage auf, ob ein solcher dynamischer Lösungsansatz in der Lage ist innerhalb von Sekunden für bis zu 1000 Eigenschaften eine Antwort zu bieten.

Zusätzlich mussten Fragen der Gleichheit von Werten beantwortet werden, da teils ein Datensatz in einem Vergleich nicht vorlag. Ist zum Beispiel eine nicht vorhandene Zeile gleich wie ein NULL Wert? Und bei Attributen die in tabellarischer Form vorliegen, muss definiert werden nach welchen Merkmalen Einträge verglichen werden.

Eine weitere Herausforderung bestand in der Problematik, dass je nach Kontext unterschiedliche Eigenschaften vorliegen. Würde man zum Beispiel Fahrzeuge vergleichen und dabei ein Auto mit einem Schiff vergleichen, sollte nicht jedes Bauteil als Unterschied ausgewiesen werden. Vielmehr sollte der unterschiedliche Fahrzeugtyp dargestellt werden und alle darunterliegenden Details nicht weiter verglichen werden.

Ergebnis

Um die Unterschiede, welche zwischen Daten ermittelt wurden, zu präsentieren bietet sich APEX mit einem Interactive Report an. Damit Vergleiche die durchgeführt werden nicht bei jeder Filterung im Interactive Report erneut ausgeführt werden, wurden Collections verwendet, welche bei jedem komplett neuen Vergleich befüllt werden. Dabei wurden neben den unterschiedlichen Datensätzen auch Informationen abgelegt, welche zeigen wo innerhalb der Anwendung die jeweiligen Daten verändert wurden. Der Benutzer hat dabei die Möglichkeit die jeweilige Seite der Anwendung aufzurufen und gleichzeitig die unterschiedlichen Werte dieser Seite angezeigt zu bekommen. So behält der Benutzer vor allem in großen Anwendungen die Übersicht, welche Änderungen wo und von wem durchgeführt wurden.

Fazit

Es ist also möglich in Datenmodellen mit über 100 Tabellen dynamisch gezielt Daten zu vergleichen. In der Entwicklung ist ein solcher Ansatz zwar nicht trivial, aber durch die Metadaten können Datenmodelländerungen leicht und schnell direkt angepasst werden, ohne die eigentliche Logik anpassen zu müssen. Trotz der Komplexität in der Entwicklung, fallen Fehler in den Metadaten oder der Logik schnell auf, da diese sofort dazu führen, dass der Vergleich nicht durchgeführt werden kann. Würde man hingegen einen statischen Ansatz benutzen, so müsste bei jeder Datenmodelländerung die Logik nachgepflegt werden und evtl. vorhandene Fehler in der Vergleichslogik würden jeweils nur beim Vergleich des Attributes an dieser Stelle auffallen.

Kontaktadresse:
Oliver Lemm
MT AG
Balcke-Dürr-Allee, 9
D-40882 Ratingen

Telefon: +49 (0) 2102 30961-0
Fax: +49 (0) 12-345 6788
E-Mail: oliver.lemm@mt-ag.de
Internet: www.mt-ag.de
Blog: <http://oliverlemm.blogspot.de/>
Twitter: [@OliverLemm](https://twitter.com/OliverLemm)