



# Dynamisch Unterschiede in Datensätzen auf Feldebene finden

Sven-Olaf Kelbert

Nürnberg, 20.11.2012

# MT AG

## Daten und Fakten

Als anerkannter Partner der führenden Technologiehersteller verbinden wir die Agilität eines mittelständischen Unternehmens mit der Lösungskompetenz internationaler Beratungshäuser.

### Hauptsitz

Ratingen

### Niederlassungen

Hamburg, Dortmund, Frankfurt,  
Luxemburg

### Tochtergesellschaften

MT-ifs GmbH, MT-ics GmbH

### Beschäftigte (2012)

220 Festangestellte  
80 Freiberufler



# Agenda

1. Ausgangslage
2. Datenmodell
3. Wunsch
4. Umsetzung
5. Schwierigkeiten
6. Fazit



# Die Ausgangslage

4 | Dynamisch Unterschiede in Datensätzen auf Feldebene finden

# Ausgangslage

- Fonds-Gesellschaft
- FondsProfiler: Verwaltung von Fonds in Datenbank-System
  - Hinterlegung von umfangreichen Informationen
    - Restriktionen für die Anlage in Aktien, Fonds, Renten (Anleihen), Geldmarktpapiere, Währungen, Rohstoffe, Immobilien etc.
    - Anlagevorschriften für bestimmte Länder, Regionen und Kontinente
    - Eingrenzung auf bestimmte Börsen
    - Nutzung von Benchmarks und Indexen
    - Kostenstruktur
    - Management
    - verwaltungstechnische Angaben
  - Abbildung des Genehmigungs- und Auflageprozesses

# Ausgangslage

- Details zum Projekt:
  - Vortrag „Das größte APEX Projekt der Welt @ Union Investment“ (Niels de Bruijn)
  - Donnerstag, 9 Uhr, Istanbul



# Das Datenmodell

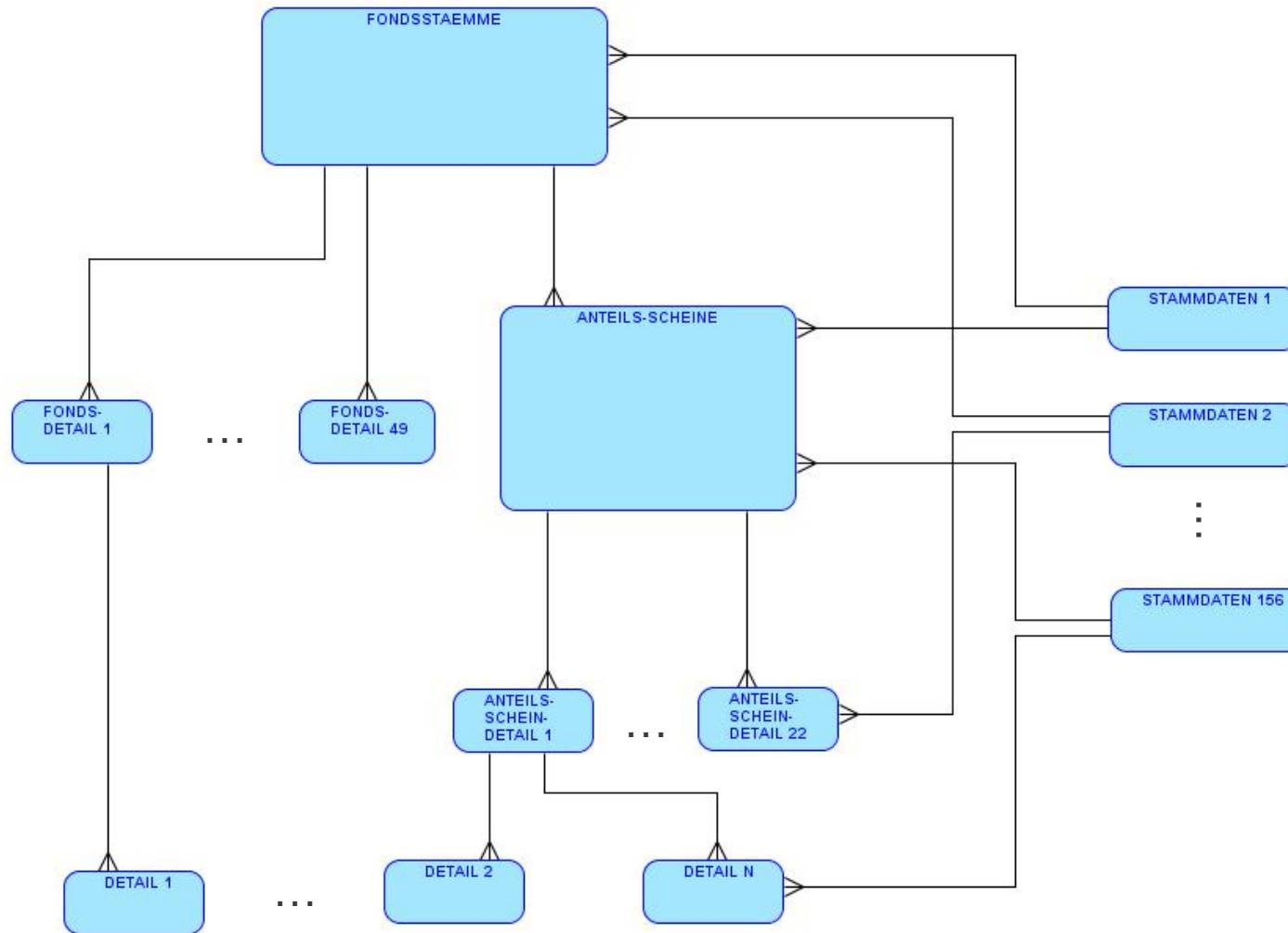
7 | Dynamisch Unterschiede in Datensätzen auf Feldebene finden

# Datenmodell – Überblick

- zahlreiche Informationen → umfangreiches Datenmodell
- Basistabelle für ein Fondsprofil: FONDSSTAEMME
- Fondsstamm kann in mehrere Anteilsscheine aufgeteilt sein
- Fondsstamm sowie Anteilsscheine besitzen diverse Detaildaten



# Datenmodell



# Datenmodell – Fakten

- Fondsstamm
  - 50 Detail-Tabellen
- Anteilsschein
  - 22 Detail-Tabellen
- Detail-Tabellen
  - weitere Detail-Tabellen mit weiteren Detaillierungen über mehrere Ebenen
- 156 Stammdaten-Tabellen
  
- insgesamt 344 Tabellen mit 7307 Spalten



# Der Wunsch

11 | Dynamisch Unterschiede in Datensätzen auf Feldebene finden

# Fonds-Kopie

- Kopie eines kompletten Fondsprofils mit allen Detail-Informationen
  - Anlage einer neuen Version eines Fondsprofils
    - Änderung von Anlagerestriktionen
    - Änderung von verwaltungstechnischen Daten (z.B. Kontonummer)
  - Erhaltung der Original-Version eines Fondsprofils
    - Revisionsicherheit
  - Anlage neuer Fonds auf Basis eines bestehenden Fonds
    - Templating

# Delta-Report

- Vergleich von zwei Versionen eines Fondsprofils
- Feststellung der Unterschiede
- Identifizierung von Eingabefehlern
- Übertragung von Änderungen in weitere Versionen eines Fondsprofils
  
- Fondsprofiler besteht aus ca. 150 Masken
  - manueller Vergleich der Daten nicht möglich
  - automatisierter, programmatischer Vergleich nötig



# Die Umsetzung

# dynamische Umsetzung

- alle Felder hart verdrahtet vergleichen
  - hohe Anzahl der zu vergleichenden Datenbank-Felder
  - spätere Datenmodell-Änderungen
- Entwicklung einer dynamischen Lösung
  - übersichtlich
  - funktioniert bei zukünftigen System-Anpassungen automatisch weiter

# Prinzip

- Start mit Tabelle FONDSSTAEMME
  - finden aller Unterschiede
- rekursive Verzweigung in alle Detail-Tabellen
  - über Data Dictionary
  - über FKs
  - Rückgabe und Weiterverarbeitung der dort gefundenen Unterschiede



# Hauptfunktion

```
function get_differenzen_table
  (p_table_name in varchar2,
   p_pk_constraint_name in varchar2,
   p_pk_column_name in varchar2,
   p_pk_id_akt in number,
   p_pk_id_hist in number,
   p_fond_version in number,
   p_master_detail_kz in varchar2)
return t_differenzen
pipelined;
```

# Pipelined Table Function

- pipelined table function

- Aufruf

```
select *  
  from table(get_differenzen_table(...));
```

- Rückgabe ist eine Daten-Tabelle

- „Datensätze“ werden zeilenweise zurückgegeben über

```
pipe row (...);
```

# Rückgabe-Objekt

- Objekt-Typ mit allen Informationen zu Unterschieden
  - Name der Tabelle und Spalte
  - Name der Maske und des Feldes
  - Werte in den beiden Versionen
  - diverse Hilfsdaten für den Abgleich
    - Wert und Name der PK-Spalten,
    - Wert und Name der FK-Spalten für Stammdatentabellen
  - Änderungslog

# Durchlaufen der Spalten

- alle Spalten der übergebenen Tabelle aus dem Data Dictionary heraussuchen und per Loop durchlaufen

```
select column_name,  
       data_type  
  from user_tab_columns  
 where table_name = p_table_name
```

- Unterscheidung
  - normale Spalte
  - FK-Spalte

# Werte selektieren (normale Spalte)

- Typ NUMBER, VARCHAR2

```
execute immediate
  'select tab_akt.'||rec.column_name||', '||
  '          tab_hist.'||rec.column_name||
  '  from '||p_table_name||' tab_akt, '||p_table_name||' tab_hist '||
  ' where tab_akt.'||p_pk_column_name||' =
          nvl('||to_char(n_id1)||', '||to_char(n_id2)||')'||
  ' and tab_hist.'||p_pk_column_name||' =
          nvl('||to_char(n_id2)||', '||to_char(n_id1)||')'
into v_wert_akt, v_wert_hist;
```

- Typ DATE

```
execute immediate
  'select to_char(tab_akt.'||rec.column_name||', ''DD.MM.YYYY HH24:MI:SS''), '||
  '          to_char(tab_hist.'||rec.column_name||', ''DD.MM.YYYY HH24:MI:SS'')'||
  '  from '||p_table_name||' tab_akt, '||p_table_name||' tab_hist '||
```

# Daten vorhanden?

- Aktionen nach der Kopie des Fondsprofils
  - Löschung eines Detail-Datensatzes
    - Parameter p\_pk\_id\_akt = null
  - Hinzufügen eines Detail-Datensatzes
    - Parameter p\_pk\_id\_hist = null
  - Veränderung des Detail-Datensatzes
    - Parameter p\_pk\_id\_akt <> null, p\_pk\_id\_hist <> null

Ausgangs-Version	Vergleichs-Version
<kein Datensatz vorhanden>	xyz

Ausgangs-Version	Vergleichs-Version
abc	xyz

# Ausgabe der Differenzen

```
v_differenz := t_differenz (p_table_name,  
                           rec.column_name,  
                           v_wert_akt,  
                           v_wert_hist,  
                           ...);  
  
pipe row (v_differenz);
```

# Bearbeitung von FK-Spalten

- rekursiver Aufruf der Hauptfunktion für die Master-Tabelle

```
select *  
  from table(get_differenzen_table(v_master_table_name,  
                                   v_master_constraint_name,  
                                   v_pk_column_name,  
                                   to_number(v_wert_akt),  
                                   to_number(v_wert_hist),  
                                   p_fond_version,  
                                   'M' ))
```

- Ergebnis: Delta-Report für Master-Tabelle

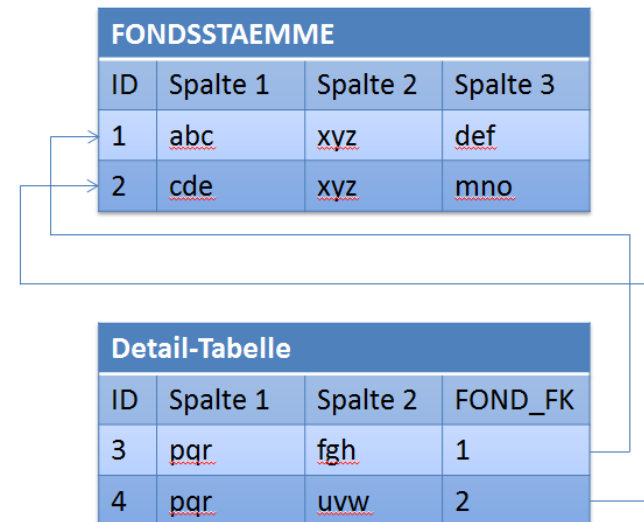


# Abarbeitung der Detail-Tabellen

- Verzweigung in sämtliche Detail-Tabellen der aktuellen Tabelle
- Prüfung anhand des Data Dictionaries auf FKs
- Durchlaufen der Detail-Tabellen per Rekursion
  
- Anzahl der Detail-Datensätze für die zu vergleichenden Master-Datensätze
  - jeweils kein Detail-Datensatz → nichts zu vergleichen
  - jeweils genau ein Detail-Datensatz → direkter Vergleich
  - mehrere Detail-Datensätze vorhanden → komplexerer Vergleich

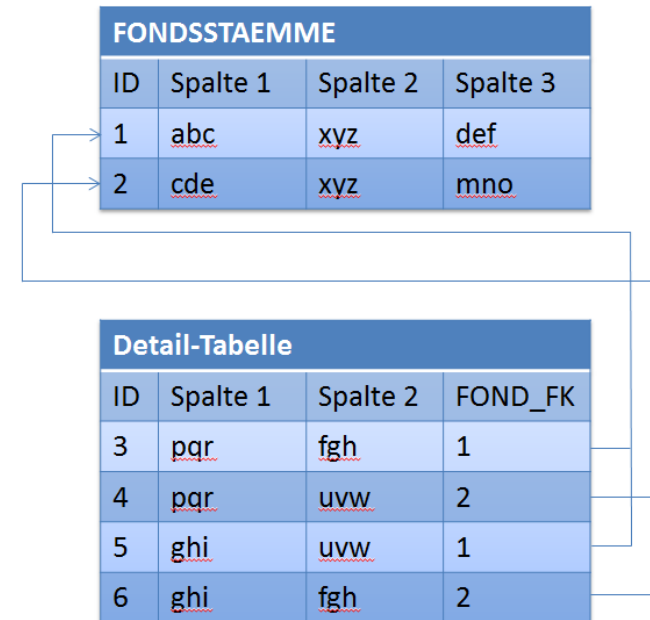
# direkter Vergleich

- Rekursions-Schritt
  - Aufruf von `get_differenzen_table` mit den Detail-Datensätzen statt mit den Haupt-Datensätzen
  - Integration der Ergebnisse in das Ergebnis der aufrufenden Funktion



# komplexerer Vergleich

- Problem:
  - Welcher Detail-Datensatz ist bei der Kopie aus welchem Datensatz entstanden?
  - Vergleich IDs 3/4 und 5/6 oder Vergleich IDs 3/6 und 4/5?
- Lösung: Mapping-Tabelle für Kopie
  - ID des Ursprungs-Datensatzes
  - ID der daraus entstandenen Kopie



# Abarbeitungsfluss

- Durchlaufen der Detail-Tabellen
  - Rückgabe der Unterschiede für die Detail-Tabelle
  - Integration in die Differenzen der übergeordneten Tabelle
  - evtl. Verzweigungen in Stammdaten-Tabellen
  - Zusammenführen aller Differenzen im 1. Aufruf der Hauptfunktion
  - Ausgabe aller Differenzen als Daten-Tabelle
  - Anzeige in der APEX-Maske über einfaches Select



# Schwierigkeiten

# Verzweigung im Data Dictionary

- dynamische Verzweigung durch das Datenmodell
  - nächste Detail-Tabelle finden
  - Masterdatensatz zu einer FK-Spalte finden

```
select uc.r_constraint_name
  from user_cons_columns ucc, user_constraints uc
 where ucc.column_name = 'FOND_DEPB_ID,
       and ucc.constraint_name = uc.constraint_name
       and uc.table_name = 'FONDSSTAEMME,
       and uc.constraint_type = 'R,
```

- sehr inperformant!

# Materialized View

```
create materialized view CONS_COLUMNS_MV
refresh force start with to_date('01-01-2012', 'dd-mm-yyyy') next TRUNC(SYSDATE) + 1
as
select detail_ac.table_name detail_table_name,
       detail_pk_ac.constraint_name detail_pk_constraint,
       detail_pk_acc.column_name detail_pk_column_name,
       detail_ac.constraint_name detail_fk_constraint,
       detail_acc.column_name detail_column_name,
       master_ac.table_name master_table_name,
       detail_ac.r_constraint_name master_pk_constraint,
       master_acc.column_name master_column_name
from all_cons_columns master_acc,
     all_constraints master_ac,
     all_cons_columns detail_acc,
     all_constraints detail_ac,
     all_cons_columns detail_pk_acc,
     all_constraints detail_pk_ac
where detail_acc.constraint_name = detail_ac.constraint_name
      and detail_ac.constraint_type = 'R'
      and detail_ac.r_constraint_name = master_acc.constraint_name
      and detail_ac.r_constraint_name = master_ac.constraint_name
      and detail_pk_ac.constraint_type = 'P'
      and detail_pk_ac.table_name = detail_ac.table_name
      and detail_pk_acc.constraint_name = detail_pk_ac.constraint_name
      and <Owner in allen Tabellen> = <Owner>;
```

# Rekursion und pipelined table function

- normale rekursive Funktion
  - Rückgabewert kann beliebig weiter verarbeitet werden
- rekursive pipelined table function
  - Rückgabe ist Datentabelle

```
select *
  from table(get_differenzen_table(v_master_table_name,
                                   v_master_constraint_name,
                                   v_pk_column_name,
                                   to_number(v_wert_akt),
                                   to_number(v_wert_hist),
                                   p_fond_version,
                                   'M' ))
```



# Weiterreichen der Detail-Daten

- Datentabelle aus rekursivem Aufruf muss weitergereicht werden in Datentabelle der aufrufenden Funktion
- für jeden Datensatz aus dem Select  
`pipe row (v_differenz);`
- Gefahr, dass Datensätze ansonsten verloren gehen
- Ergebnis-Datensätze aus den unteren Ebenen der Rekursion werden in jeder Ebene wieder per pipe row in das Ergebnis der nächsthöheren Ebene integriert.

# Ausnahmen

- keine Regel ohne Ausnahme...
- Meta-Daten (Strukturbäume) als statische Ausnahmen programmiert

# Nachziehen von Änderungen

- Zusatz-Funktionalität:
  - festgestellte Differenzen zwischen Datensätzen automatisiert angleichen
- Fall 1: in zu aktualisierender Version fehlt der Datensatz komplett
  - Hinzufügen des entsprechenden Datensatzes (nur einmal, nicht für jede Spalte)
  - dynamische Liste aller Spalten für INSERT zusammenstellen
- Fall 2: in ursprünglicher Version fehlt der Datensatz komplett
  - Löschen des entsprechenden Datensatzes
  - vorher rekursives Löschen aller vorhandenen Details
- Fall 3: beide Datensätze vorhanden
  - Update auf die unterschiedlichen Felder
  - UK-Verletzungen abfangen



# Fazit

# Fazit

- technischer Delta-Report sehr einfach zu erzeugen
  - dynamische Selects
  - Data Dictionary
- Probleme mit Dynamik
  - Delta-Report ohne Anpassungen für fachliche Anwender schwer lesbar
  - interne Spalten ausblenden
  - hart verdrahtete speziell Vorgehensmodelle für bestimmte Tabellen
- Anpassungen an der Dynamik
  - machen Programmcode weniger gut lesbar
  - zerstören Dynamik
  - helfen Endanwender



**Fragen?**

# Besuchen Sie auch unsere weiteren Vorträge auf der DOAG 2012

Dienstag, 12 Uhr, Raum Riga

Dienstag, 13 Uhr, Raum Seoul

Dienstag, 14 Uhr, Raum Stockholm

Dienstag, 15 Uhr, Raum Kopenhagen

Dienstag, 16 Uhr, Raum Stockholm

Mittwoch, 13 Uhr, Raum Riga

Mittwoch, 15 Uhr, Raum Riga

Mittwoch, 16 Uhr, Raum Seoul

Donnerstag, 09 Uhr, Raum Istanbul

Donnerstag, 14 Uhr, Raum Konf. EG

Donnerstag, 15 Uhr, Raum Istanbul

Donnerstag, 16 Uhr, Raum Oslo

Dynamisch Unterschiede in Datensätzen auf Feldebene finden by S.O. Kelbert

Route to ASM by Ernst Leber

Automatische Generierung der ETL-Prozesse OWB vs. ODI by Irina Gotlibovych

Wiederverwendung von bestehendem PL/SQL Code in ADF Anwendungen by Hendrik Gossens

„Managed Code“ mit OWB – Methoden und Wege by Bernhard Rosenberger

Dateizugriff mit new I/O 2 by Wolfgang Nast

WebServices in Java SE und EE by Wolfgang Nast

Das Mysterium OPatch by Volker Mach

Das größte APEX Projekt der Welt @ Union Investment by Niels de Bruijn

Testen mit Pfefferminzgeschmack by Birgit Kratz

APEX goes UNIT Testing by Oliver Lemm

SOA verspielt – rekursive BPEL Prozesse by Guido Neander



# Vielen Dank.

## MT AG

Balcke-Dürr-Allee 9  
40882 Ratingen

Telefon: +49 (0) 21 02 309 61-0  
Telefax: +49 (0) 21 02 309 61-10

E-Mail: [info@mt-ag.com](mailto:info@mt-ag.com)  
[www.mt-ag.com](http://www.mt-ag.com)