



„SLOWLY CHANGING DIMENSION TYP 2“ IM RELATIONALEN UND MULTIDIMENSIONALEN DATA WAREHOUSE

Jordan, Claus . Consultant . 16. Mai 2006



1. EINFÜHRUNG IN DIE THEMATIK

Ein Data Warehouse dient als Langzeitarchiv für wichtige Unternehmensinformationen über mehrere Jahre hinweg. Der echte „Wert dieser alten Daten“ liegt in der Möglichkeit, aktuelle Kennzahlen mit denen der Vergangenheit zu vergleichen oder basierend auf Kennzahlen aus zurückliegenden Perioden (z.B. Monaten) Projektionen in die Zukunft zu berechnen.

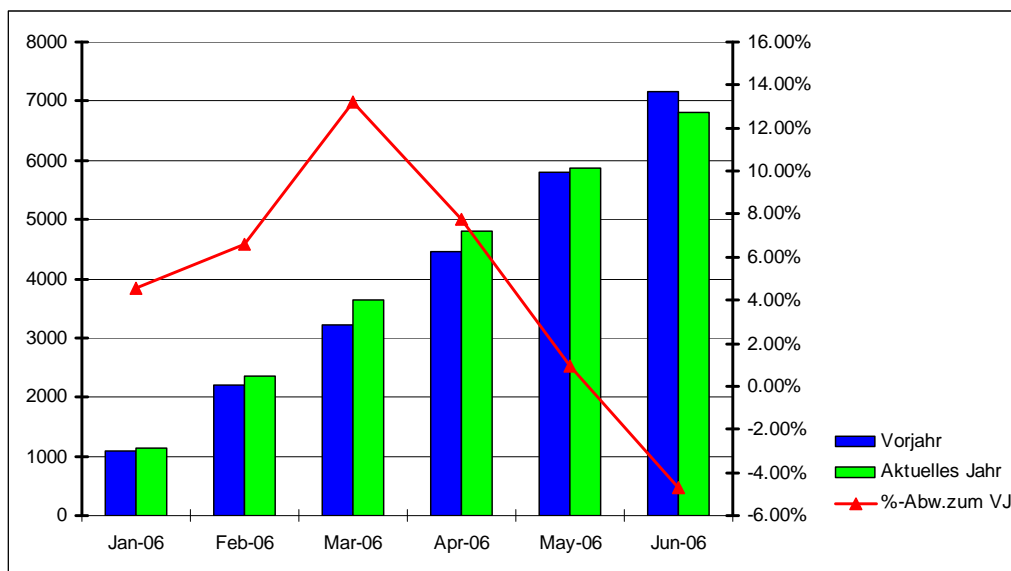
Vorraussetzung für die Akzeptanz eines Data Warehouse ist, dass sämtliche ausgewiesenen Kennzahlen betriebswirtschaftlich sinnvoll und richtig sind.

Dieser Beitrag beschäftigt sich mit der Umsetzung der in relationalen Data Warehouse Projekten am häufigsten verwendete Historisierungsmethode „Slowly Changing Dimension Type 2“ (kurz SCD2) und deren Relevanz auch in multidimensionalen Umgebungen.

In den Kapiteln 1 und 2 wird die Methode SCD2 und deren Umsetzung in relationaler Technologie an einem einfachen Beispiel erklärt. Kapitel 3 zeigt anhand desselben Beispiels die Umsetzung und Übernahme in die multidimensionale Technologie.

1.1 Beispiel: Year-to-Date Vorjahresvergleich

Sehr beliebt, weil aussagekräftiger - insbesondere zum Ende des Geschäftsjahres - ist eine „Year-to-Date“-Betrachtung wichtiger Kennzahlen. Dabei werden die Daten von Beginn des Geschäftsjahres bis zur aktuellen Periode aufaddiert. Häufig werden diese Daten mit den entsprechenden „Year-to-Date“ Daten des Vorjahres verglichen und entsprechende prozentuale Abweichungen berechnet. Die folgende grafische Darstellung zeigt „Year-to-Date“ Daten des aktuellen Jahres im Vergleich zum Vorjahr sowie die prozentuale Abweichung:



Dieser Vorperioden-Vergleich ist nur dann unproblematisch, wenn folgende Frage mit „Ja“ beantwortet werden kann:

Werden unverdichtete Granular-Daten miteinander verglichen?.

In den seltensten Fällen wird dies jedoch der Fall sein. Üblicherweise werden in Analysewerkzeugen oder in Berichten in irgendeiner Form verdichtete Daten benutzt. Unter



Umständen haben sich hierarchische Zuordnungen in einer Dimension über die betrachtete Zeitspanne geändert (z.B. ist ein Kunde in eine neue Region gezogen oder er wird seit kurzem einer anderen Kategorie zugeordnet).

1.2 Beispiel: Vorperiodenvergleich

Das folgende, bewusst sehr einfache Zahlenbeispiel soll helfen, diese Problematik zu verdeutlichen. In den folgenden Kapiteln wird stets auf dieses Beispiel verwiesen. Die folgende Tabelle zeigt die Umsätze pro Kunde für den Monat (Feb 2006) und für den Vormonat (Jan 2006):

Umsatz	Jan 2006	Feb 2006
Kunde A	10	15
Kunde B	20	10
Kunde C	10	20
Kunde D	25	12
Kunde E	10	13

Dieser Vergleich der dargestellten Granular-Daten für Monate und Kunden ist unproblematisch. Typischerweise werden diese Granular-Daten jedoch für Analysezwecke über Hierarchiestufen verdichtet. Im Vertriebscontrolling sollen die monatlichen Kundenumsätze beispielsweise über Verkaufsregionen verdichtet und in Form einer Vorperiodenvergleichsauswertung analysiert werden. Diese Auswertung könnte wie folgt aussehen:

Umsatz	Jan 2006	Feb 2006	
Region 1	30	25	← ?
Kunde A	10	15	
Kunde B	20	10	
Region 2	45	45	← ?
Kunde D	25	12	
Kunde E	10	13	
Kunde C	10	20	

Diese Auswertung scheint korrekte Daten zu enthalten. Es könnte jedoch sein, dass einer der dargestellten Kunden im letzten Monat in eine andere Region umgezogen ist. In diesem Fall würden möglicherweise die auf Regionsebene verdichteten Umsätze betriebswirtschaftlich nicht richtig ausgewiesen. Deshalb wird nachfolgend davon ausgegangen, dass der „Kunde C“ zum 1. Februar 2006 von „Region 1“ in „Region 2“ umgezogen ist.

Die betriebswirtschaftlich richtige Darstellung der Kennzahlen für dieses Szenario sieht nun wie folgt aus:

Umsatz	Jan 2006	Feb 2006	
Region 1	40	25	← OK
Kunde A	10	15	
Kunde B	20	10	
Kunde C	10		
Region 2	35	45	← OK
Kunde D	25	12	
Kunde E	10	13	
Kunde C		20	

Im Januar 2006 werden nun die der Realität entsprechenden Umsätze ausgewiesen, während sich erwartungsgemäß im Februar 2006 nichts geändert hat.



Betriebswirtschaftlich richtige Ausweisung von Kennzahlen wie im obigen Beispiel kann durch die Versionierung oder Historisierung von bestimmten Attributen erfolgen. In diesem Beispiel hat sich das Attribut „Region“ von „Kunde C“ im Laufe der Zeit geändert. Welche Dimensions-Attribute im konkreten Anwendungsfall versioniert werden sollen, muss vor Implementierung gemeinsam mit den Verantwortlichen der Fachabteilung diskutiert werden.

1.3 SCD 1, 2 oder 3

Neben der Historisierungsmethode „Slowly Changing Dimension Type 2“ (kurz SCD2) existieren noch die Methoden SCD1 und SCD3. Der Vollständigkeit halber wird im Folgenden auf die wesentlichen Unterschiede der drei Historisierungsmethoden eingegangen. SCD wird immer auf Attributsebenen (Tabellenspalten) angewendet.

SCD1:

Ein SCD1-Attribut welches sich geändert hat, wird überschrieben sobald es sich ändert. Es wird also keine neue Version des gesamten Datensatzes angelegt. Damit ist die Historie für immer verloren. Es handelt dabei um den Standardfall, d.h. Attribute die nicht vom Typ SCD2 oder SCD3 sind, gehören immer zum Typ SCD1.

SCD2:

Für den Fall, dass sich ein SCD2-Attribut ändert, wird eine neue Version des gesamten Datensatzes angelegt (siehe nachfolgendes Kapitel). Damit ist nachvollziehbar, welchen Inhalt ein Attribut zu jedem Zeitpunkt hatte.

SCD3:

Ein SCD3-Attribut wird wie bei SCD1 (siehe oben) überschrieben. Zusätzlich wird der alte Wert in einem separaten Attribut, z.B. LAST_VALUE_REGION, gespeichert. SCD3 wird in der Praxis sehr selten verwendet und wird deshalb in diesem Beitrag nicht weiter behandelt.

Die realitätsnahe SCD2-Vorgehensweise beinhaltet jedoch auch Nachteile:

- Möchte man die Umsätze von „Kunde C“, unabhängig von der Region analysieren, muss bei der Abfrage (Select Statement) darauf geachtet werden, dass die Daten der beiden Versionen von „Kunde C“ addiert werden.
- Soll in einer Abfrage die Anzahl der Kunden pro Jahr oder Monat ermittelt werden, darf der Kunde C nicht zweimal gezählt werden. Die Komplexität solcher Abfragen ist nicht zu unterschätzen.
- Die Komplexität der Lade- und Abfragelogik (auf die hier bewusst nicht eingegangen wird) und damit der Implementierungs-Aufwand für das Data Warehouse kann sich durch den Einsatz von SCD2 signifikant erhöhen.
- Das Datenvolumen der Dimensionstabelle steigt durch mehrere Versionen eines Datensatzes und durch zusätzliche Felder, wie VALID_FROM und VALID_TO, die für SCD2 benötigt werden.



2. Historisierung mit relationaler Technologie

Das oben beschriebene einfache zweidimensionale Beispiel „Umsätze nach Kunden und Monate“ wird nun mit relationaler Technologie realisiert. Es werden monatlich zuerst die Dimensionen und anschließend die Fakten geladen. Dabei wird beispielhaft für das Attribut REGION der Tabelle DIM_KUNDEN die Historisierungsmethode SCD2 gezeigt.

2.1 Dimensionstabelle für Kunden mit SCD2

Die Kunden-Stammdaten aus dem Vorsystem (z.B. SAP R/3) sollen monatlich in die Tabelle DIM_KUNDEN des Data Warehouse geladen werden. Das Vorsystem stellt zu Monatsbeginn entsprechende ASCII-Dateien zur Verfügung, die wie folgt aufgebaut sind:

ASCII-Datei vom 01. Februar 2006 (Kunden-Stammdaten Stand 31. Januar 2006):

K-Nr	K-Bezeichn.	Region
Kunde A	Müller GmbH	1
Kunde B	Beier AG	1
Kunde C	Franz AG	1
Kunde D	Stoll GmbH & Co. KG	2
Kunde E	Frey GmbH	2

ASCII-Datei vom 01. März 2006 (Kunden-Stammdaten Stand 28. Februar 2006):

K-Nr	K-Bezeichn.	Region
Kunde A	Müller GmbH	1
Kunde B	Beier AG	1
Kunde C	Franz AG	2
Kunde D	Stoll GmbH & Co. KG	2
Kunde E	Frey GmbH	2



In der Data Warehouse Dimensionstabelle für Kunden DIM_KUNDEN existiert ein künstlicher Schlüssel, der als „Surrogate-Key (SK)“ bezeichnet wird und gleichzeitig primärer Schlüssel (Primary Key) dieser Tabelle ist. Das Attribut „Region (REGION)“ steht für ein Verkaufsgebiet, hier 1 oder 2. Den beiden zusätzlich benötigten Datums-Attributen „gültig von (VALID_FROM)“ und „gültig bis (VALID_TO)“ kommt im Zusammenhang mit SCD2 besondere Bedeutung zu, da aus diesen Attributen ersichtlich ist ob die Version eines Kunden zu einem bestimmten Datum gültig ist oder nicht. Hinter dem Attribut „Business Key (BK)“ verbirgt sich der Schlüssel, bzw. die ID aus dem Vorsystem, zum Beispiel die Kundennummer, in unserem Beispiel „Kunde A“ oder „Kunde B“.

DIM_KUNDEN nach dem Laden am 01. Februar 2006:

SK	BK	NAME	REGION	VALID_FROM	VALID_TO
101	Kunde A	Müller GmbH	1	01.04.2001	31.12.9999
102	Kunde B	Beier AG	1	01.06.2003	31.12.9999
103	Kunde C	Franz AG	1	01.07.2003	31.12.9999
104	Kunde D	Stoll GmbH & Co. KG	2	01.09.2004	31.12.9999
105	Kunde E	Frey GmbH	2	01.01.2006	31.12.9999

Anmerkung:

Die Kunden waren vor dem Laden bereits in der Tabelle DIM_KUNDEN vorhanden und wurden deshalb nicht modifiziert.



DIM_KUNDEN nach dem Laden am 01. März 2006:

SK	BK	NAME	REGION	VALID_FROM	VALID_TO
101	Kunde A	Müller GmbH	1	01.04.2001	31.12.9999
102	Kunde B	Beier AG	1	01.06.2003	31.12.9999
103	Kunde C	Franz AG	1	01.07.2003	31.01.2006 ←
104	Kunde D	Stoll GmbH & Co. KG	2	01.07.2003	31.12.9999
105	Kunde E	Frey GmbH	2	01.09.2004	31.12.9999
106	Kunde C	Franz AG	2	01.02.2006	31.12.9999 ←

Die markierten Zeilen zeigen die beiden Versionen von „Kunde C“ nach dem Laden am 01. März 2006. Eine neue Version wird immer dann angelegt, wenn sich ein nach SCD2 zu historisierendes Attribut (hier REGION) ändert. In unserem Beispiel hat sich am 1. Februar 2006 die Zuordnung zu der Region geändert. Es wird also eine neue Version für „Kunde C“ angelegt. Dabei muss das Attribut VALID_TO der alten Version auf den letzten gültigen Tag gesetzt werden und die Attribute VALID_FROM und VALID_TO der neuen Version entsprechend gesetzt werden. Die neue Version erhält einen neuen Surrogate-Key (SK).

2.2 Dimensionstabelle für Monate

Die Dimensionstabelle für die Monate wird ohne Historisierung wie folgt definiert.

DIM_MONATE nach dem Laden am 01. März 2006:

SK	BK	NAME	QUARTER	YEAR
...				
50	Jan_06	Januar 2006	Q1/2006	2006
60	Feb_06	Februar 2006	Q1/2006	2006
...				

2.3 Faktentabelle

Die Bewegungsdaten mit den Umsätzen werden nach den Dimensionen in das Data Warehouse geladen. Das Vorsystem stellt eine entsprechende ASCII-Dateien zur Verfügung, die wie folgt aussieht:

ASCII-Datei vom 01. Februar 2006 (Fakten für Januar 2006):

K-Nr	Monat	Umsatz
Kunde A	Januar 2006	10
Kunde B	Januar 2006	20
Kunde C	Januar 2006	10
Kunde D	Januar 2006	25
Kunde E	Januar 2006	10

ASCII-Datei vom 01. März 2006 (Fakten für Februar 2006):

K-Nr	Monat	Umsatz
Kunde A	Februar 2006	15
Kunde B	Februar 2006	10
Kunde C	Februar 2006	20
Kunde D	Februar 2006	12
Kunde E	Februar 2006	13



Die Data Warehouse Faktentabelle für die Umsatzzahlen besteht in unserem Beispiel aus lediglich drei Attributen. SK_KUNDE und SK_MONAT sind die Foreign-Keys auf die Primary Keys der beiden Dimensionstabellen DIM_KUNDEN und DIM_MONAT. Das dritte Attribut dient zur Speicherung der Kennzahl Umsatz.

Faktentabelle UMSATZ nach dem Laden am 01. Februar 2006:

SK_KUNDE	SK_MONAT	UMSATZ
..		
101	50	10
102	50	20
103	50	10
104	50	25
105	50	10

← Kunde C, Januar 2006, 10 \$

Faktentabelle UMSATZ nach dem Laden am 01. März 2006:

SK_KUNDE	SK_MONAT	UMSATZ
..		
101	50	10
102	50	20
103	50	10
104	50	25
105	50	10
101	60	15
102	60	10
104	60	12
105	60	13
106	60	20

← Kunde C, Januar 2006, 10 \$

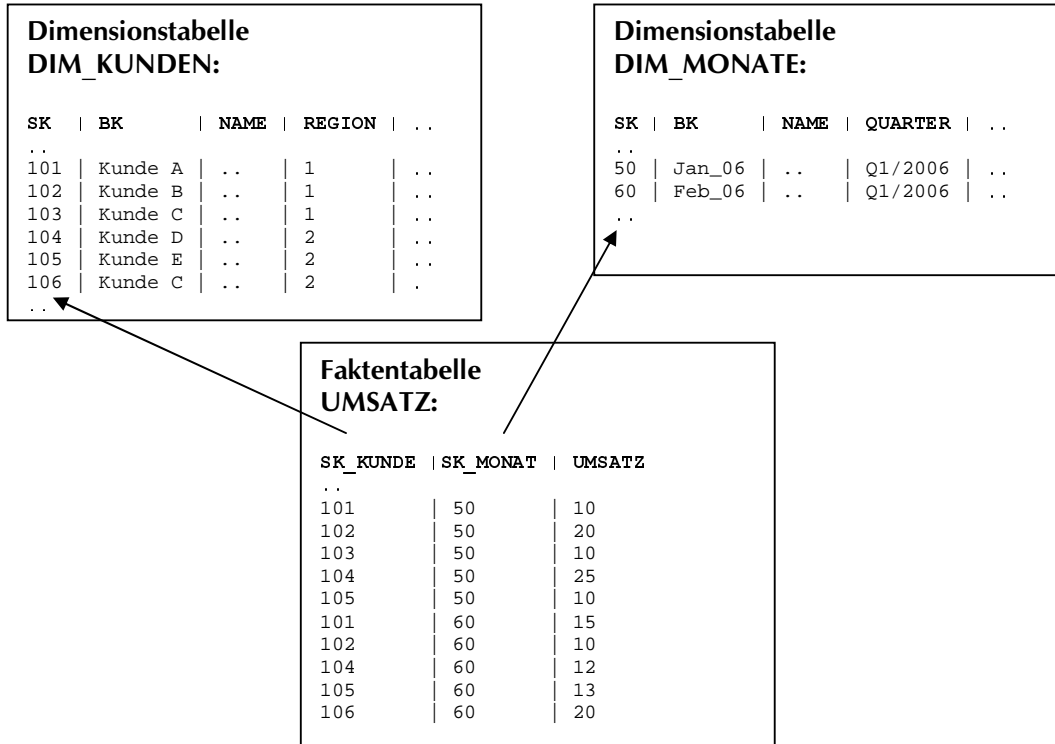
← Kunde C, Februar 2006, 20 \$

Die markierten Zeilen zeigen die Umsätze von „Kunde C“ im Januar und Februar 2006. Der Januar Umsatz verweist auf die erste Version (103) und der Februar Umsatz auf die zweite Version von „Kunde C“ (106). Data Warehouse – intern handelt es sich bei den unterschiedlichen Version um zwei verschiedene Kunden.



2.4 Star-Schema

Die beiden Dimensionstabellen und Faktentabelle stellen das folgende sehr einfache Star-Schema dar:



2.5 Beispiele für Abfragen des Endanwenders

Die folgenden Abfrage-Beispiele auf das zweidimensionale Star-Schema demonstrieren, wie sich SCD2 für das Attribut REGION der Dimension DIM_KUNDEN auf die Ergebnisse auswirkt.

2.5.1 Zeige den Umsatz von „Kunde C“ über alle Monate

Abfrage:

```
SELECT K.BK KUNDE ,
       sum(U.UMSATZ) UMSATZ
FROM   UMSATZ U,
       DIM_KUNDEN K
WHERE  U.SK_KUNDE = K.SK and
       K.BK = 'Kunde C'
GROUP BY
       K.BK
```

Ergebnis:

KUNDE	UMSATZ
Kunde C	30



Um die Summe der Umsätze für „Kunde C“ zu erhalten, muss über den Business-Key (BK) und nicht über den Surrogate-Key summiert werden.

2.5.2 Zeige den Umsatz pro Region und Monat

Abfrage:

```
SELECT M.NAME MONAT,
       K.REGION REGION,
       sum(U.UMSATZ) UMSATZ
FROM   UMSATZ U,
       DIM_KUNDEN K,
       DIM_MONATE M
WHERE  U.SK_KUNDE = K.SK and
       U.SK_MONAT = M.SK
GROUP BY
       M.NAME, K.REGION
```

Ergebnis:

MONAT	REGION	UMSATZ
Januar 2006	1	40
Februar 2006	1	25
Januar 2006	2	35
Februar 2006	2	45

Für diese Abfrage muss über die Regionen der Dimension DIM_KUNDEN gruppiert werden. Da für „Kunde C“ zwei Versionen existieren (eine gehört zu Region 1, die andere zu Region 2) werden die Regionssummen korrekt angezeigt. Ohne SCD2, würde diese Abfrage ein anderes, betriebswirtschaftlich „falsches“ Ergebnis liefern, da die Umsätze von „Kunde C“ vollständig in die Region 2 summiert würden.



3. Historisierung mit multidimensionaler Technologie

Im Gegensatz zur relationalen speichert die multidimensionale Technologie die Kennzahlen physikalisch nicht in Tabellen sondern in eigenen mehrdimensionalen Strukturen, die häufig als Würfel oder Cubes bezeichnet werden. Bei Einsatz von Oracle OLAP beispielsweise, erfolgt die Speicherung der Cubes in so genannten Analytic Workspaces (AW's), die als binäre Objekte in der Oracle Datenbank gespeichert und verwaltet werden. Ein Vorteil der multidimensionalen Speicherung ist die wesentlich performantere Abfrage hierarchisch verdichteter Kennzahlen. Unbedingt hervorzuheben ist zudem die erheblich einfachere Definition von abgeleiteten Kennzahlen wie beispielsweise die „Prozentuale Abweichung zum Vorjahr für den kumulierten Umsatz“.

„Warum überhaupt SCD2 und MOLAP?“

Eine Antwort auf diese Frage könnte lauten:

„Weil auch in Auswertungen mit Zeitbezug, die auf multidimensionaler Technologie (MOLAP) basieren, betriebswirtschaftlich „richtige“ Kennzahlen angezeigt werden sollen. MOLAP-Kennzahlen sollten mit den entsprechenden ROLAP-Kennzahlen über die gesamte Zeitachse vergleichbar sein, mit dem Ziel die Akzeptanz des Data Warehouse insgesamt zu steigern.“

Selbstverständlich ist es denkbar in der multidimensionalen Technologie bewusst auf SCD2 zu verzichten, obwohl in der relationalen Technologie SCD2 zum Einsatz kommt. Aus diesem Grund werden in den beiden folgenden Kapiteln beide Szenarien exemplarisch behandelt.

3.1 MOLAP ohne SCD2

In diesem Abschnitt wird beschrieben, wie Dimensionen ohne SCD2, d.h. ohne Versionierung, in die multidimensionale Technologie übernommen werden. Als Basis dafür dienen die Star-Schema-Tabellen DIM_MONATE, DIM_KUNDEN und UMSATZ aus dem oben beschriebenen Beispiel.

3.1.1 Nicht versionierte Dimensionen

Die Speicherung einer Dimension in multidimensionaler Technologie erfolgt typischerweise so, dass die Elemente aller Hierarchiestufen in einer „Liste“ gespeichert werden. Über die entsprechende Parent-Child-Beziehung zwischen Dimensionseinträgen erfolgt die Zuordnung zum jeweils übergeordneten Element.

Die Kunden-Dimension ohne Versionierung aus unserem Beispiel wird in multidimensionaler Technologie, tabellarisch dargestellt wie folgt gespeichert:

MOLAP_KUNDEN	PARENTREL	DESC
1		Region 1
2		Region 2
Kunde A	1	Müller GmbH
Kunde B	1	Beier AG
Kunde D	2	Stoll GmbH & Co. KG
Kunde E	2	Frey GmbH
Kunde C	2	Franz AG

Kundenname (z.B. Franz AG) oder Regionsbezeichnung werden als Attribut (z.B. DESC) zum jeweiligen Dimensionseintrag gespeichert. Das hierarchisch übergeordnete Element wird in einer „Self-Relation“ (z.B. PARENTREL) gespeichert. Die Einträge für die Regionen und Kunden sind dabei gleichberechtigt. Die physische Speicherung erfolgt nicht in Tabellen!



Um die Tabelle DIM_KUNDEN in multidimensionaler Technologie am 01. März 2006 zu laden, müssen mit folgendem Select-Statement die gewünschten Kunden aus der relationalen Dimensionstabelle DIM_KUNDEN abgefragt werden.

Abfrage:

```
SELECT BK,  
       NAME,  
       REGION,  
       'Region ' || REGION REGION_DESC  
FROM   DIM_KUNDEN  
WHERE  VALID_TO   >= '01.März 2006' and  
       VALID_FROM <= '01.März 2006'
```

Ergebnis:

BK	NAME	REGION	REGION_DESC
Kunde A	Müller GmbH	1	Region 1
Kunde B	Beier AG	1	Region 1
Kunde C	Franz AG	2	Region 2
Kunde D	Stoll GmbH & Co. KG	2	Region 2
Kunde E	Frey GmbH	2	Region 2

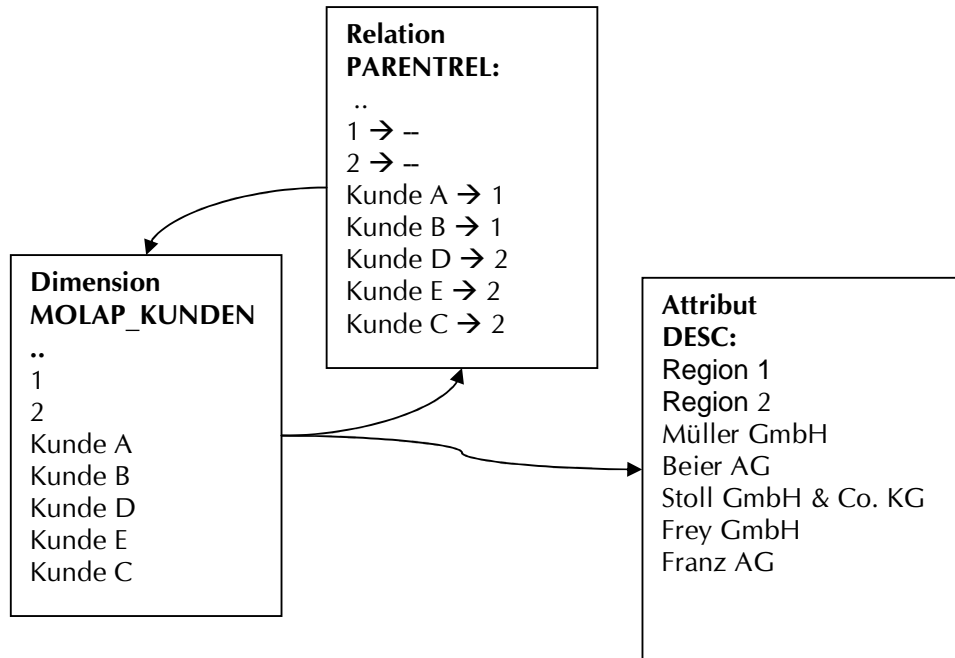
Die Einschränkung auf das Ladedatum (hier 01. März 2006) bewirkt, dass lediglich die an diesem Tag gültigen Versionen aus der Tabelle DIM_KUNDEN geladen werden. „Kunde C“ gehört laut der Tabelle DIM_KUNDEN am 1. März 2006 zur „Region 2“!

Nach der Extraktion erfolgt das Laden mit folgendem Mapping (Zuordnung) zwischen Abfrageergebnis (links) und multidimensionalen Objekten (rechts).

Relationale Attribute	Mapping	MOLAP-Objekte
REGION	→	MOLAP_KUNDEN
BK	→	MOLAP_KUNDEN
REGION	→	PARENTREL
NAME	→	DESC
REGION_DESC	→	DESC



In multidimensionaler Technologie (z.B. Analytic Workspace in Oracle OLAP) werden in diesem Beispiel folgende Objekte einschließlich Inhalt angelegt:



3.1.2 Kennzahlen (Fakten)

Die Speicherung der Kennzahlen (Fakten) in multidimensionaler Technologie erfolgt in mehrdimensionalen Feldern (Array's). Dabei sind die Dimensionen die Indizes zu den Kennzahlen. Normalerweise werden die Kennzahlen auf unterster Hierarchieebene (hier Kunde und Monat) in die multidimensionale Technologie übernommen und anschließend auf die übergeordneten Hierarchiestufen der jeweiligen Dimensionen verdichtet.

Die Umsätze aus unserem Beispiel mit der nichtversionierten Kunden-Dimension werden in multidimensionaler Technologie wie folgt gespeichert:

MOLAP_UMSATZ		MOLAP_MONATE	
		50 (Jan 2006)	60 (Feb 2006)
M O L A P K U N D E N	1 (Region 1)	30	25
	Kunde A	10	15
	Kunde B	20	10
	2 (Region 2)	45	45
	Kunde D	25	12
	Kunde E	10	13
	Kunde C	10	20

Die Speicherung erfolgt nicht in Tabellen!



Um die Fakten in multidimensionale Technologie zu laden, muss die Tabelle UMSATZ mit folgendem Select-Statement abgefragt werden:

Abfrage:

```
SELECT K.BK KUNDEN ,
       U.SK_MONAT MONAT ,
       sum(U.UMSATZ)UMSATZ
FROM   UMSATZ U ,
       DIM_KUNDEN K
WHERE  U.SK_KUNDE = K.SK
GROUP BY
       K.BK, U.SK_MONAT
```

Ergebnis:

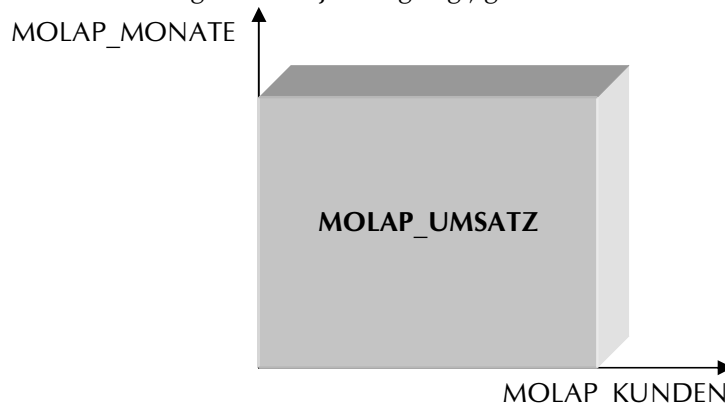
KUNDEN	MONAT	UMSATZ
Kunde A	50	10
Kunde A	60	15
Kunde B	50	20
Kunde B	60	10
Kunde C	50	10
Kunde C	60	20
Kunde D	50	25
Kunde D	60	12
Kunde E	50	10
Kunde E	60	13

In diesem Select-Statement werden die Daten aus der Faktentabelle über den Business Key (hier Kundennummer) verdichtet.

Nach der Extraktion erfolgt das Laden mit folgendem Mapping zwischen Abfrageergebnis (links) und multidimensionalem Objekt (rechts).

Relationale Attribute	Mapping	MOLAP-Objekte
KUNDE	→	MOLAP_KUNDEN
MONAT	→	MOLAP_MONATE
UMSATZ	→	MOLAP_UMSATZ

In multidimensionaler Technologie (z.B. Analytic Workspace in Oracle OLAP) wird für die Kennzahl Umsatz folgendes Objekt angelegt, geladen und verdichtet:





3.1.3 Fazit

Diese auch in relationaler Technologie weit verbreitete Vorgehensweise, nämlich kein SCD2 anzuwenden, liefert in diesem konkreten Beispiel auf Regionsebene für die Vorperioden betriebswirtschaftlich "falsche" Ergebnisse.

Diese könnten vermieden werden, wenn nach dem Laden der Umsätze für „Feb 2006“ die Vergangenheitsumsätze (hier alle Vormonate bis einschließlich „Jan 2006“) nicht verdichten werden, sondern nur der Umsatz von „Feb 2006“. Ein gravierender Nachteil dieser Vorgehensweise ist, dass dann die Summe der Kunden einer Region nicht zwingend dem Regionsumsatz entspricht, was wiederum zu Irritationen und Akzeptanzproblemen führen kann. Außerdem kann ein versehentliches Verdichten des gesamten Datenbestands die mühevoll aufgebaute Historie zerstören.

3.2 MOLAP mit SCD2

In diesem Abschnitt wird beschrieben, wie Dimensionen nach SCD2 versioniert in die multidimensionale Technologie übernommen werden. Als Basis dafür dienen die Starschema-Tabellen DIM_MONATE, DIM_KUNDEN und UMSATZ aus dem oben beschriebenen Beispiel.

3.2.1 Versionierte Dimensionen

Die Speicherung einer nach SCD2 versionierten Dimension erfolgt genau gleich wie bei einer nicht versionierten (siehe Kapitel 3.1.1).

Die versionierte Kunden-Dimension aus unserem Beispiel wird in multidimensionaler Technologie tabellarisch dargestellt wie folgt gespeichert:

MOLAP_KUNDEN	PARENTREL	DESC	
1	--	Region 1	
2	--	Region 2	
101	1	Müller GmbH	
102	1	Beier AG	
103	1	Franz AG	←
104	2	Stoll GmbH & Co. KG	
105	2	Frey GmbH	
106	2	Franz AG	←

Dabei wird für jeden relationalen Dimensionseintrag ein entsprechender Eintrag in der Dimension in der multidimensionalen Technologie gespeichert. Da nun die Kundennummer (z.B. „Kunde C“) nicht mehr eindeutig ist, wird als Schlüssel der künstliche Schlüssel (Surrogate Key) aus der Tabelle DIM_KUNDEN verwendet. Die Kundennummer bzw. die Regions-ID kann als ergänzendes Attribut mitgespeichert werden.

Um die Tabelle DIM_KUNDEN in die multidimensionale Technologie zu laden, werden mit folgendem Select-Statement die gewünschten Kunden aus der relationalen Dimensionstabelle DIM_KUNDEN abgefragt.

Abfrage:

```
SELECT SK,  
       BK,  
       NAME,  
       REGION, 'Region ' || REGION REGION_DESC  
FROM   DIM_KUNDEN
```



Ergebnis:

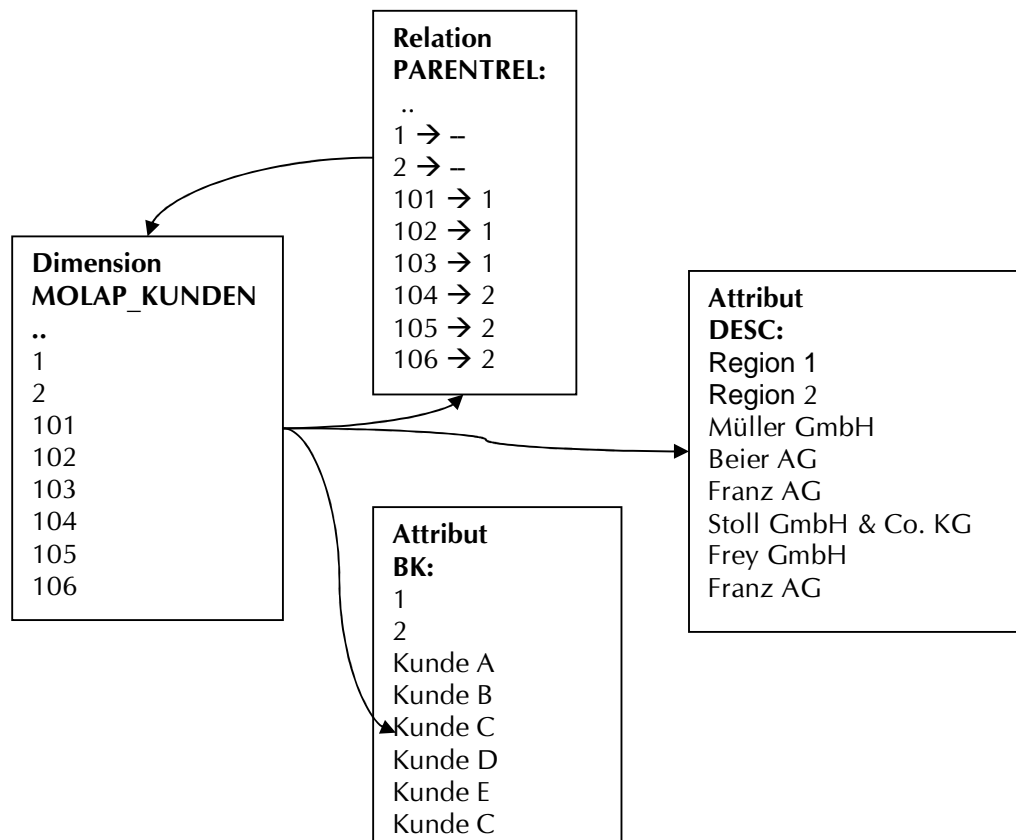
SK	BK	NAME	REGION	REGION_DESC
101	Kunde A	Müller GmbH	1	Region 1
102	Kunde B	Beier AG	1	Region 1
103	Kunde C	Franz AG	1	Region 1
104	Kunde D	Stoll GmbH & Co. KG	2	Region 2
105	Kunde E	Frey GmbH	2	Region 2
106	Kunde C	Franz AG	2	Region 2

Nach der Extraktion erfolgt das Laden mit folgendem Mapping zwischen Abfrageergebnis (links) und multidimensionalen Objekten (rechts).

Relationale Attribute	Mapping	MOLAP-Objekte
REGION	→	MOLAP_KUNDEN
SK	→	MOLAP_KUNDEN
BK	→	BK
REGION	→	PARENTREL
NAME	→	DESC
REGION_DESC	→	DESC

Das Attribut „REGION“ wird zweimal als Quelle benötigt: Einmal als Dimensionswert in der Dimension MOLAP_KUNDEN und einmal für die Parent-Child-Relation PARENTREL.

In multidimensionaler Technologie (z.B. Analytic Workspace in Oracle OLAP) werden in diesem Beispiel folgende Objekte angelegt und geladen:





3.2.2 Kennzahlen (Fakten)

Die Daten für die Kennzahl „Umsatz“ werden, wie bereits im Kapitel 3.1.2 beschrieben, auf unterster Hierarchieebene (hier Kunde und Monat) in die multidimensionale Technologie übernommen. Danach erfolgt die Verdichtung auf die übergeordneten Hierarchieebene (hier Regionen).

Die Umsätze aus unserem Beispiel mit der versionierten Kunden-Dimension wird in multidimensionaler Technologie wie folgt gespeichert:

MOLAP_UMSATZ		MOLAP_MONATE	
		50 (Jan 2006)	60 (Feb 2006)
M O L A P K U N D E N	1 (Region 1)	40	25
	101 (Kunde A)	10	15
	102 (Kunde B)	20	10
	103 (Kunde C)	10	
	2 (Region 2)	35	45
	104 (Kunde D)	25	12
	105 (Kunde E)	10	13
	106 (Kunde C)		20

Die Speicherung erfolgt nicht in Tabellen!

Um die Fakten in multidimensionale Technologie zu laden, muss die Tabelle UMSATZ mit folgendem Select-Statement abgefragt werden:

Abfrage:

```
SELECT SK_KUNDE KUNDE ,  
       SK_MONAT MONAT ,  
       UMSATZ  
FROM   UMSATZ
```

Ergebnis:

KUNDE	MONAT	UMSATZ	
101	50	10	
101	60	15	
102	50	20	
102	60	10	
103	50	10	←
106	60	20	←
104	50	25	
104	60	12	
105	50	10	
105	60	13	



Nach der Extraktion erfolgt das Laden mit folgendem Mapping zwischen Abfrageergebnis (links) und multidimensionalem Objekt (rechts).

Relationale Attribute	Mapping	MOLAP-Objekte
KUNDE	→	MOLAP_KUNDEN
MONAT	→	MOLAP_MONATE
UMSATZ	→	MOLAP_UMSATZ

3.2.3 Umsatz pro Kunde oder pro Kundenversion?

Häufig sind die Umsätze eines bestimmten Kunden über einen größeren Zeitraum gewünscht. In der relationalen Technologie werden die Fakten im Select-Statement über den Business Key (z.B. „Kunde C“) verdichtet um die Summe aller Versionen eines Kunden zu erhalten (siehe dazu Kapitel 2.5.1).

In multidimensionaler Technologie würde man die Umsätze der zu analysierenden Kunden sortiert ausgeben. Dadurch werden alle Versionen eines Dimensionseintrags aufgelistet wie folgendes Beispiel zeigt:

BK	DESC	Jan 2005	Feb 2005
Kunde A	Müller GmbH	10	15
Kunde B	Beier AG	20	10
Kunde C	Franz AG	10	
Kunde C	Franz AG		20
Kunde D	Stoll GmbH & Co. KG	25	12
Kunde E	Frey GmbH	10	13

Mit Hilfe von Gruppierungsfunktionen im Front-End über das Attribut BK können die Summen pro Kunde zur Laufzeit berechnet werden.

Eine andere Möglichkeit wäre, in der multidimensionalen Technologie eine weitere Hierarchie für die Kunden-Dimension zu definieren, die die einzelnen Kunden-Versionen zum Kunden zusammenführt. Die Umsätze von „Kunde C“ ergeben sich aus der Summe der von „Kunde C (bis Jan-06)“ und „Kunde C (ab Feb-06)“, wie die folgende Tabelle veranschaulicht:

SK	BK	DESC	Jan 2005	Feb 2005
501	Kunde A	Müller GmbH	10	15
101	Kunde A	Müller GmbH	10	15
502	Kunde B	Beier AG	20	10
102	Kunde B	Beier AG	20	10
503	Kunde C	Franz AG	10	20
103	Kunde C	Franz AG	10	
106	Kunde C	Franz AG		20
504	Kunde D	Stoll GmbH & Co. KG	25	12
104	Kunde D	Stoll GmbH & Co. KG	25	12
505	Kunde E	Frey GmbH	10	13
105	Kunde E	Frey GmbH	10	13

In diesem Fall ist es notwendig, für jeden Kunden einen weiteren Dimensionseintrag anzulegen. Die führt im schlechtesten Fall zur Verdoppelung der Einträge.



3.2.4 Fazit

Der größte Vorteil von SCD2 - auch in der multidimensionalen Technologie - ist, dass auch bei vergangenheitsbezogener Analyse betriebswirtschaftlich sinnvolle Ergebnisse angezeigt werden. Am einfachsten zu realisieren ist dieser Ansatz, wenn ein SCD2-basiertes Star- oder Snowflake-Schema im relationalen Data Warehouse als Quelle dient. Dieser Fall wurde beispielhaft vorgestellt und zeigt, dass die Daten in der multidimensionalen Technologie den Ergebnissen der Abfrage auf das Data Warehouse entsprechen. Durch die so erzwungene Übereinstimmung zwischen multidimensionaler und relationaler Technologie innerhalb eines Data Warehouse kann die Richtigkeit der Daten – und somit die Akzeptanz der Anwender – mit einfachen Mitteln sichergestellt werden.

Beim Design des multidimensionalen Datenmodells ist jedoch zu berücksichtigen, dass der Einsatz von SCD2 auch Nachteile in sich birgt. So können versionierte Dimensionen rasch wachsen. Ein Nebeneffekt vieler Versionen ist, dass bei Auswertungen für einen „Kunden“ viele Zeilen angezeigt werden, die meist keinen Wert (NULL) aufweisen. Deshalb sollte das Front-End (OLAP-Werkzeug) unbedingt die Funktion der „Nullzeilen-Unterdrückung“ aufweisen. Aus diesem Gründen ist es ratsam SCD2 in der multidimensionalen Technologie begrenzt und wenn, dann ganz bewusst einzusetzen.

Selbstverständlich kann SCD2 in der multidimensionalen Technologie auch dann angewendet werden, wenn kein relationales Data Warehouse mit SCD2-Implementierung vorliegt. In diesem Fall muss die SCD2-Logik beim Laden in die multidimensionale Technologie definiert werden. Dieser Ansatz ist nicht zu empfehlen, da die gängigen ETL-Standardwerkzeuge nur die relationale Implementierung von SCD2 unterstützen.

4. Zusammenfassung

Die Historisierungsmethode „Slowly Changing Dimension Typ 2“ (SCD2) ist sowohl mit relationaler als auch mit multidimensionaler Technologie implementierbar. Es muss jedoch sorgfältig analysiert werden, welche Dimensionen historisiert bis zu welcher Hierarchiestufe in die multidimensionale Technologie übernommen werden, da neben positiven auch negative Nebenwirkungen auftreten können.

Claus Jordan arbeitet seit 1993 als Consultant im Umfeld Business Intelligence und Data Warehousing. Sein Schwerpunkt liegt im Bereich der multidimensionalen OLAP-Technologien.

Vielen Dank an **Joachim Wehner** für die konstruktive Unterstützung während der Erstellung und den inhaltlichen und technischen Review dieses Beitrags.

Claus Jordan
Trivadis GmbH
Industriestrasse 4
70565 Stuttgart

claus.jordan@trivadis.com
Telefon: +49 (0) 162 - 2959643



5. Glossar

- MOLAP: Multidimensionales **OnLine Analytical Processing**, Daten werden nicht in Tabellen, sondern in speziellen multidimensionalen Strukturen gespeichert. Beispiel „Analytic Workspace“ in Oracle OLAP.
- ROLAP: Relationales **OnLine Analytical Processing**, Daten werden in Tabellen gespeichert.
- SCD2: Slowly Changing Dimension Typ 2.
- AW: Analytic Workspace von Oracle OLAP.