

6 Das BW-Datenmodell

Im vorangegangenen Kapitel wurden die maßgeblichen Datenmodelle in ihrer Reinform beschrieben. Dieses Kapitel erläutert das Datenmodell des BW. Dabei handelt es sich um eine Vielzahl unterschiedlicher BW-Objekte (vgl. Kapitel 4.3), welche für unterschiedliche Aufgaben eingesetzt werden können und zu diesem Zweck unterschiedliche Datenmodelle nutzen, kombinieren oder abwandeln. Dabei handelt es sich um:

- InfoObjekte und deren Master-Data-Konzept
- DataStore-Objekte
- BasisCubes

Als Oberbegriff dieser BW-Objekte wird die Bezeichnung *Datenziel* verwendet, da dies die BW-Objekte sind, die zur physischen Aufnahme analyserelevanter Daten eingesetzt werden. Den Schwerpunkt dieses Kapitels bildet die Erläuterung dieser Datenziele.

Den Abschluss dieses Kapitels bildet ein Hinweis darauf, wie Daten der Datenziele mit Bezug auf bzw. mit Abhängigkeit zu einem Quellsystem abgelegt werden können.

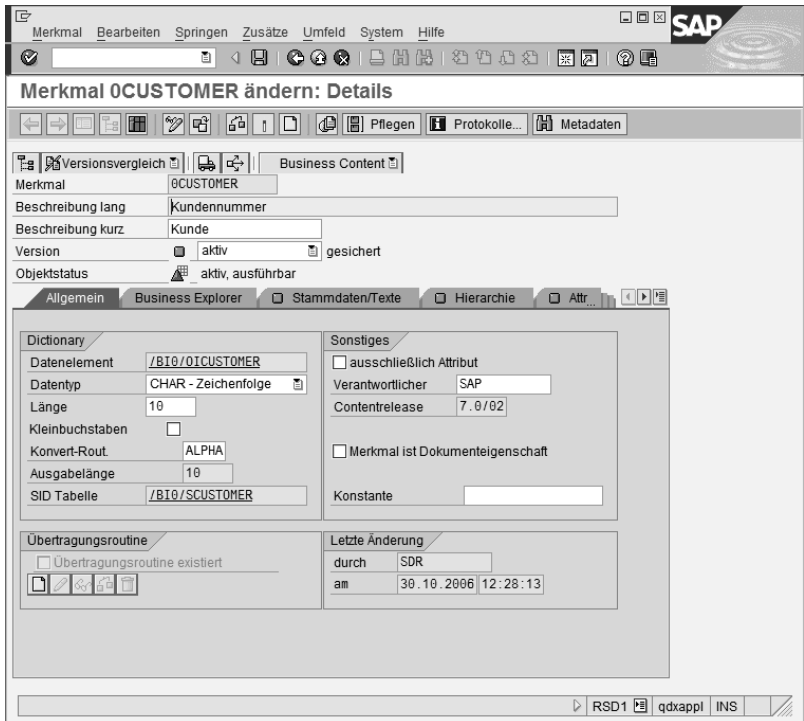
6.1 InfoObjekte

InfoObjekte bilden die Basis für die Definitionen aller anderen Datenziele. Alle Datenziele definieren sich vollständig über InfoObjekte und können nur dann angelegt werden, wenn die erforderlichen InfoObjekte existieren.

InfoObjekte sind durch einen systemweit eindeutigen technischen Namen (zum Beispiel OCUSTOMER) definiert, der durch eine Bezeichnung ergänzt ist (zum Beispiel »Kundennummer« für das InfoObjekt OCUSTOMER).

Neben dem technischen Namen und der Bezeichnung wird für jedes InfoObjekt ein Datentyp (zum Beispiel CHAR für einen alphanumerischen Datentyp) und die Datenlänge festgelegt. Diese Einstellungen werden in den allgemeinen Einstellungen jedes InfoObjektes vorgenommen (siehe Abb. 6–1).

Abb. 6–1
Allgemeine Einstellungen
von InfoObjekten



© SAP AG

Da alle Datenmodelle, in denen ein InfoObjekt verwendet wird, technisch in Form einer oder mehrerer transparenter Tabellen im ABAP Dictionary definiert werden (vgl. Kapitel 4.2.3), muss für jedes InfoObjekt im ABAP Dictionary ein entsprechendes Datenelement mit zugehöriger Domäne definiert sein. Unter diesem Datenelement wird das InfoObjekt in Elementen des ABAP Dictionary verwendet.

Das entsprechende Datenelement mit der dazugehörigen Domäne leitet sich aus dem Datentyp und der Datenlänge des InfoObjektes ab und folgt einer Namenskonvention (siehe unten), die den technischen Namen des InfoObjektes berücksichtigt (zum Beispiel das Datenelement /B10/0ICUSTOMER und die Domäne /B10/0CUSTOMER für das InfoObjekt 0CUSTOMER).

Beim Aktivieren der Metadaten eines InfoObjektes legt das BW Datenelement und Domäne automatisch im ABAP Dictionary an. Abbildung 6–2 stellt den Zusammenhang zwischen InfoObjekt und Datenelement beispielhaft am InfoObjekt OCUSTOMER dar.

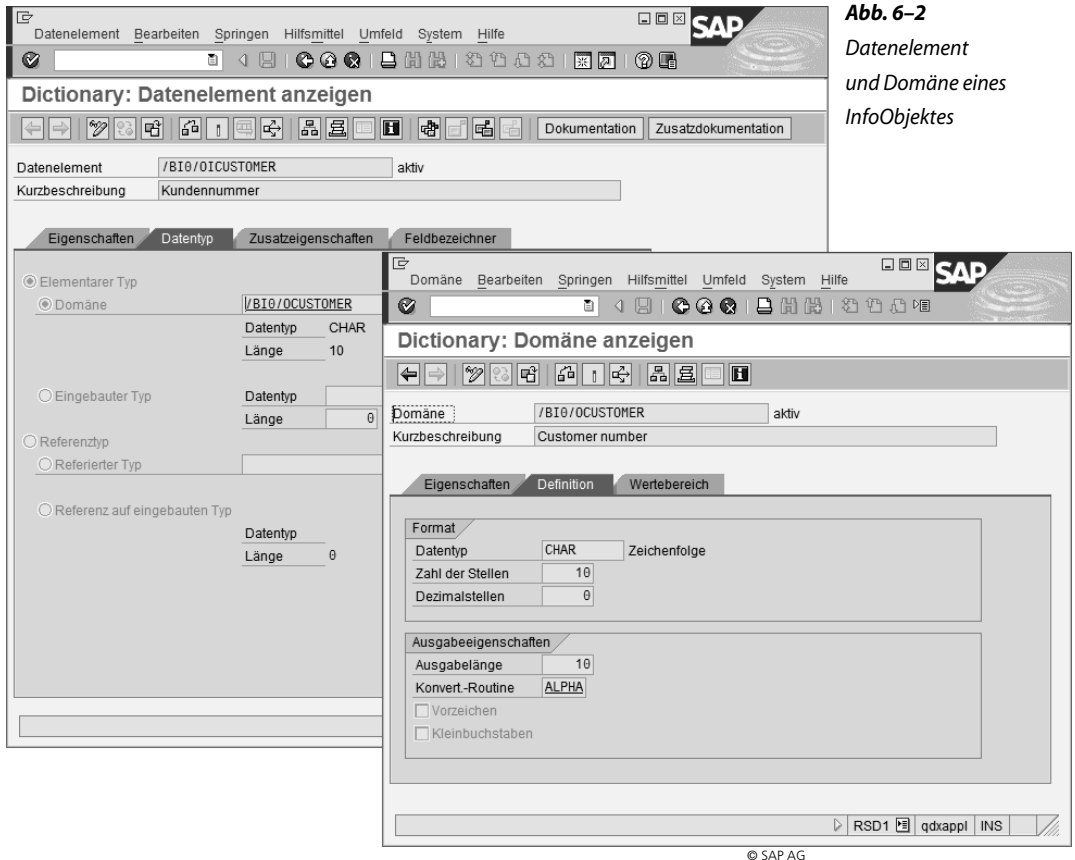


Abb. 6–2
Datenelement
und Domäne eines
InfoObjektes

Wird der Datentyp oder die Datenlänge eines InfoObjektes verändert, so wird die zugrunde liegende Domäne im ABAP Dictionary ebenfalls automatisch angepasst. Existieren bereits transparente Tabellen im ABAP Dictionary, die mit Daten gefüllt sind, so werden die Tabellen entsprechend der neuen Einstellungen auf der Datenbank angepasst. Dies ist nicht möglich, wenn die Anpassung (zum Beispiel bei einer Verkürzung der Datenlänge) mit einem Informationsverlust verbunden wäre.

Die nachfolgende Tabelle zeigt auf, welchen Namenskonventionen InfoObjekt-Namen, Datenelemente und Domänen folgen.

Namenskonvention

	Standard-InfoObjekt	Eigendefiniertes InfoObjekt
Name des InfoObjektes	0ttttttttt	{A-Z}tttttttt
Name des Datenelements	/BIO/0Ittttttttt	/BIC/OI{A-Z}tttttttt
Name der Domäne	/BIO/0tttttttttt	/BIC/0{A-Z}tttttttt
Beschreibung kurz	max. 20 Stellen	max. 20 Stellen
Beschreibung lang	max. 60 Stellen	max. 60 Stellen

InfoObjekt-Typen

Nach den bisherigen Ausführungen unterscheiden sich die Möglichkeiten von InfoObjekten nicht von den Möglichkeiten, die Datenelement und Domäne im ABAP Dictionary bieten. Darüber hinaus legen InfoObjekte jedoch eine Vielzahl weiterer Eigenschaften in den Metadaten des InfoObjektes ab.

Welche Informationen dies sind, ist abhängig vom jeweiligen InfoObjekt-Typ:

- Merkmale
- Zeiten
- Einheiten
- Kennzahlen

Nachfolgend werden die aufgeführten InfoObjekt-Typen erläutert und die mit ihnen verbundenen Eigenschaften beschrieben, sofern sie sich auf das Datenmodell beziehen.

6.1.1 Merkmale

Für InfoObjekte, die als Merkmal in den Datenmodellen des BW verwendet werden sollen, stehen die nachfolgend aufgeführten Datentypen zur Verfügung, die in den allgemeinen Einstellungen zum InfoObjekt festgelegt werden müssen:

Datentyp	Beschreibung	Zulässige Länge
CHAR	Ziffern und Buchstaben	Zeichenlänge 1 – 60
NUMC	nur Ziffern	Zeichenlänge 1 – 60
DATS	Datum	Zeichenlänge 8
TIMS	Zeit	Zeichenlänge 6

Darüber hinaus sind folgende Angaben des InfoObjektes für das Datenmodell von Bedeutung:

- Zulässigkeit von Kleinbuchstaben
- Konvertierungsroutine und Ausgabelänge
- SID-Tabelle

Diese Angaben werden nachfolgend erläutert.

Zulässigkeit von Zeichen

Im Falle von Merkmalen des Type CHAR existiert eine Reihe möglicher Stolperfallen, die aus der Verwendung unterschiedlicher nationaler Zeichensätze und Kleinbuchstaben resultieren.

Um derartigen Problemen möglichst vollständig zu entgehen, sind im BW per Default ausschließlich Zeichen mit den Hexadezimalwerten HEX21 bis HEX5A zugelassen; dies sind die Zeichen:

```
!"%&'()*+,-./:;<=>?_0123456789
ABCDEFGHIJKLMN O PQRSTU VWXYZ
```

Technisch unabdingbar ist ein Verbot von:

- Zeichen mit den Hexadezimalwerten HEX00 bis HEX1F.
- Zeichenfolgen, die mit einem ! beginnen (! ist ein Steuercode für die Eingabe von Werten in SAP-Systemen und löscht den Eingabewert).
- Zeichenfolgen, die nur das Zeichen # enthalten (analog zum Steuercode ! definiert das # eine Initialeingabe).

Alle weiteren Zeichen können im BW nach Bedarf explizit zugelassen werden. Hierfür existiert ein zweistufiges Konzept, das zum einen die *Zulässigkeit nationaler Zusatzzeichen* und zum anderen das *Zulassen aller weiteren Zeichen* ermöglicht.

In der ersten Stufe können nationale Zusatzzeichen¹ zugelassen werden, sofern es sich dabei um Großbuchstaben handelt. Die Einschränkung auf Großbuchstaben reflektiert die besonderen Probleme, die die Verwendung von Kleinbuchstaben mit sich bringen kann (s.u.).

Nationale Zusatzzeichen

Eben dadurch gestaltet sich jedoch das Zulassen nationaler Zusatzzeichen relativ komplex, sofern das System nicht als Unicode-System betrieben wird, sondern mit Codepages arbeitet: Ob ein bestimmter Zeichencode einen Großbuchstaben abbildet, ist in diesem Fall nämlich nicht durch den Zeichencode selbst, sondern durch den Zeichencode in Kombination mit der jeweils verwendeten Codepage definiert.

Um das Problem zu verdeutlichen, kann das große deutsche »Ö« als Beispiel herangezogen werden. Dieses scheint auf den ersten Blick

1. beispielsweise die deutschen Umlaute Ä, Ö oder Ü, aber auch das ß oder – für Dänemarkliebhaber – auch das Ø

eindeutig ein Großbuchstabe zu sein, jedoch stellt ebendieser Zeichencode in einer russischen Codepage ein kleines »sch« dar (um nur eines der vielen Beispiele zu nennen).

Welche Codepage verwendet wird, hängt von der jeweiligen Anmeldesprache eines Anwenders ab. Dadurch entsteht die Gefahr, dass beispielsweise unter der Verwendung einer deutschen Anmeldesprache (= deutsche Codepage) ein »Ö« als Merkmalswert gespeichert wird, das von einem Anwender mit russischer Anmeldesprache (= russische Codepage) als kleines »sch« wahrgenommen wird.

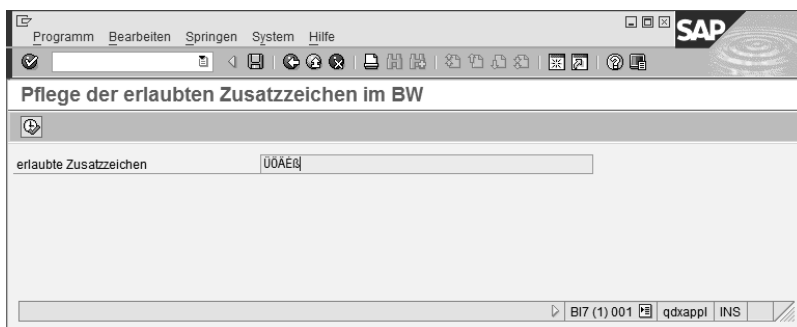
Um dies zu verhindern, dürfen nur diejenigen nationalen Zusatzzeichen genutzt werden, die in *allen installierten Codepages* einen Großbuchstaben darstellen. In der Praxis werden nur eine geringe Menge an Codepages² verwendet, so dass eine größere Anzahl an Großbuchstaben möglich wäre, ohne dass sich die entsprechenden Bitfolgen mit Kleinbuchstaben anderer Codepages überschneiden.



Werden nachträglich Sprachen installiert, so kann es sein, dass bereits Daten geladen wurden, die nach der Installation der Sprachen nicht mehr zulässig sind. Ein inkonsistenter Datenbestand ist die Folge. Gehen Sie daher mit der Erweiterung zulässiger Zeichen äußerst vorsichtig um und vermeiden Sie die Erweiterung nach Möglichkeit ganz, sofern Sie nicht über ein Unicode-System verfügen.

Diese zusätzlich möglichen Zeichen können mit Hilfe der Transaktion **RSKC** (systemweit) als Großbuchstaben zugelassen werden (siehe Abb. 6–3).

Abb. 6–3
Zulassen von
Sonderzeichen als
Großbuchstaben



© SAP AG

- Die Zeichen der englischen Sprache sind in allen Codepages enthalten. Die Zeichen der deutschen Sprache sind in Codepage 1100 (entspricht ISO 8859 bzw. Latin-1) enthalten. Da die Sprachen Deutsch und Englisch fest installiert sind, ist in jedem SAP-System zumindest Codepage 1100 installiert. Werden keine weiteren Sprachen genutzt, so muss auch keine weitere Codepage installiert sein.

Dieses Verfahren ist beispielsweise bei den üblichen europäischen Codepages angemessen, gerät bei Zeichensätzen mit außerordentlich vielen Zusatzzeichen³ jedoch schnell zur Farce.

Alternativ zum Einpflegen jedes erlaubten Zusatzzeichens besteht daher die Möglichkeit, grundsätzlich alle Zeichen zu erlauben, die in der Anmeldesprache des ladenden Users⁴ Großbuchstaben in der Codepage sind. Dies geschieht, indem anstelle der einzelnen Zusatzzeichen die Zeichenkette **ALL_CAPITAL** in der Transaktion **RSKC** eingetragen wird.

Die Zeichenfolge **ALL_CAPITAL** muss ohne weitere Ergänzungen in den Zusatzzeichen eingetragen werden. Versuchen Sie nicht, **ALL_CAPITAL** durch weitere Zeichen zu ergänzen, da es sonst bedeutungslos wird. Meldet das BW ein Zeichen trotz **ALL_CAPITAL** als ungültig, so ist durch eine Veränderung der zulässigen Zusatzzeichen keine Besserung zu erreichen. Überprüfen Sie stattdessen die Anmeldesprache des ladenden Users oder lassen Sie Kleinbuchstaben zu.



Speziell bei Unicode-Systemen werden alle Zeichen aller Sprachen durch eindeutige Bitfolgen mit variabler Länge repräsentiert. Das große deutsche »Ö« wird damit durch eine Bitfolge repräsentiert, die mit dem kleinen russischen »sch« nichts mehr zu tun hat.

Zwar gelten auch in Unicode-Systemen nur die per Default zulässigen Zeichen (s.o.), jedoch können durch die Eindeutigkeit der Bitfolgen Zusatzzeichen erlaubt werden, ohne Rücksicht auf andere Sprachen zu nehmen. Auch in diesem Fall können jedoch ausschließlich Großbuchstaben zugelassen werden.

Das BW prüft Zeichen in der Regel erst dann auf ihre Gültigkeit, wenn sie in die SID-Tabelle eines InfoObjekts geschrieben werden sollen. Falls Sie in der Definition des Datenflusses selbst die Gültigkeit einer Zeichenkette überprüfen wollen, so steht Ihnen hierfür der Funktionsbaustein **RSKC_CHAVL_CHECK** zur Verfügung.



Der restriktive Umgang mit *Kleinbuchstaben* im Schlüssel von InfoObjekten ist darin begründet, dass sie den Umgang mit entsprechenden Werten aus Sicht der Datenanalyse beträchtlich erschweren können.

Zulassen aller weiteren Zeichen

Beispielsweise könnte zur Selektion einer Kundennummer bei der Datenanalyse der String **18HB27F** oder auch **18hb27F** eingegeben werden. Hierbei ist zu klären, ob damit identische oder unterschiedliche Kunden beschrieben werden sollen.

Im Normalfall soll nicht zwischen Groß- und Kleinbuchstaben unterschieden werden. Hieraus resultiert die Default-Einstellung des

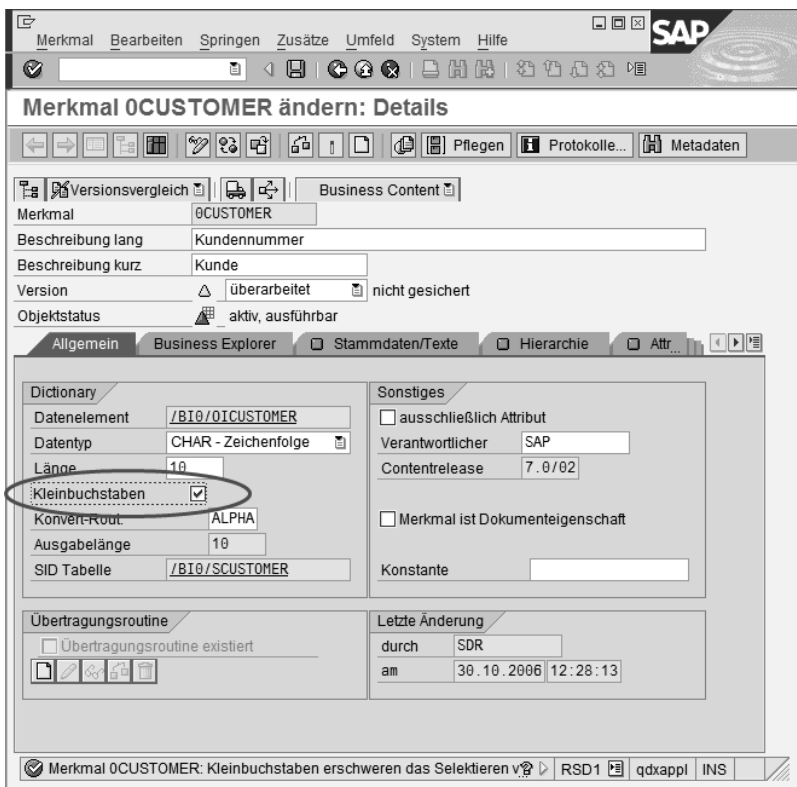
3. bspw. asiatische Zeichensätze

4. In der Regel wird dies der Hintergrund-User sein (siehe Abschnitt 15.1.1).

BW, dass alle Merkmalschlüssel nur mit Großbuchstaben versehen sein dürfen.

Sollen Kleinbuchstaben jedoch zugelassen werden, so kann dies für jedes InfoObjekt individuell festgelegt werden (siehe Abb. 6–4).

Abb. 6–4
Zulässigkeit von
Kleinbuchstaben



Im Ergebnis sind alle Zusatzzeichen für das InfoObjekt erlaubt, es wird jedoch bei der Datenanalyse ebenfalls zwischen Klein- und Großbuchstaben unterschieden⁵.

In jedem Fall ist zu beachten, dass die genannten Restriktionen und Probleme sich nur auf die Verwendung von Kleinbuchstaben im Schlüsselwert eines Merkmals beziehen und nicht für die beschreibenden Texte eines Merkmals gelten (z.B. »Müller & Smørøbrød GmbH«). Für derartige Texte sind Kleinbuchstaben und Sonderzeichen jeglicher Art in jedem Fall zugelassen (siehe Kapitel 6.2.1).

5. Sonst werden Buchstaben bei der Eingabe von Selektionen automatisch in Großbuchstaben konvertiert.

Konvertierungsroutine und Ausgabelänge

Für die interne Verarbeitung und Speicherung von Daten nutzt das BW-Basisssystem eine *interne Darstellung*. Diese gibt beispielsweise vor, dass Uhrzeiten immer im Format JJJJMMTT oder eine Kostenstelle immer mit führenden Nullen gespeichert wird.

In der *externen Darstellung*, d.h. bei der Ein- und Ausgabe von Daten, werden häufig abweichende Formate verwendet, so dass eine Konvertierung der externen in die interne Darstellung und umgekehrt stattfinden muss. So ist z.B. bei der Eingabe der Kostenstelle 1000 eine Konvertierung in den entsprechenden 10-stelligen Wert 0000001000 durchzuführen.

Um eine konsistente Konvertierung zu gewährleisten, existieren im BW sogenannte Konvertierungsroutinen, die bei der Definition von InfoObjekten zu hinterlegen sind. Abbildung 6–1 auf Seite 60 stellt das InfoObjekt 0CUSTOMER dar, bei dem die Konvertierungsroutine ALPHA anzutreffen ist.

Ist keine Konvertierungsroutine zu einem InfoObjekt definiert, so wird die externe Darstellung unverändert in der internen Verarbeitung verwendet, so dass beispielsweise die Kostenstellen 0000001000 und 01000 als zwei unterschiedliche Merkmalsausprägungen betrachtet werden würden.

Durch den Einsatz von Konvertierungsroutinen wird insbesondere dem Anwender die Arbeit erleichtert, weil dieser ein Datum bspw. ebenso im Format "001.2008" wie im Format "012008" eingeben kann, und beide Eingaben gleichermaßen in das interne Format 2008001 konvertiert werden, wenn die Konvertierungsroutine **PERI7** im InfoObjekt hinterlegt ist⁶.

Für jede Konvertierungsroutine existieren zwei Funktionsbausteine:

- ein INPUT-Baustein für die Konvertierung des externen Formats in das interne Format
- ein OUTPUT-Baustein für die Konvertierung des internen Formats in das externe Format

Die Funktionsbausteine sind unter dem Namen `CONVERSION_EXIT_XXXXX_INPUT` beziehungsweise `CONVERSION_EXIT_XXXXX_OUTPUT` zu finden, wobei `XXXXX` jeweils durch den Namen der Konvertierungsroutine zu ersetzen ist.

6. Dies ist bspw. beim InfoObjekt 0FISCPER (Geschäftsjahr/Periode) der Fall.



Sollen die Konvertierungsroutinen ALPHA, NUMCV und GJAHR nachträglich bei einem InfoObjekt hinterlegt werden, obwohl bereits Stammdaten zu diesem InfoObjekt gespeichert sind, so muss das interne Format dieser Daten nachträglich in das Format der neuen Konvertierungsroutine umgewandelt werden. Für diese Umwandlung stand bis zum Release 3.x des SAP BW die Transaktion RSMDEXITON zur Verfügung. Mit dem Release 7.0 wurde diese Transaktion jedoch deaktiviert. Ein (nicht annähernd zufriedenstellender) Ersatz ist lediglich das Austauschen des InfoObjekts bspw. im Rahmen der Re-Modellierung (siehe Abschnitt 9.2).

*Konvertierungsroutinen
ALPHA, NUMC und GJAHR*

Eine besondere Beachtung verlangen Konvertierungsroutinen, wenn sie bestimmte interne Werte nicht in externe Werte umwandeln können, es also interne Werte gibt, die in der Definition der Konvertierungsroutine unzulässig sind. Zuvorderst zu nennen sind hier die Konvertierungsroutinen ALPHA, NUMCV und GJAHR

Als Beispiel kann der Wert 1000 in einem 5-stelligen InfoObjekt genannt werden, das vom Typ CHAR ist und über die Konvertierungsroutine ALPHA verfügt. In diesem Fall kann nie ein externer Wert den internen Wert 1000 ergeben – vielmehr ergibt sowohl die Eingabe von 01000 als auch von 1000 den internen Wert 01000.

Dieser Umstand kommt vor allem bei der Durchführung von Ladevorgängen zum Tragen. Hierbei ist die Anwendung einer Konvertierungsroutine nicht zwingend, d.h. Werte können auch als interne Werte gekennzeichnet und ohne Prüfung »durchgereicht werden«. Ob ein Merkmalswert als interner Wert betrachtet und durchgereicht oder vorher eine Konvertierungsroutine durchlaufen soll, wird in Inflow Layer durch die Einstellung der sogenannten Input-Konvertierung definiert (siehe Kapitel 15.2.3).

*Konvertierungsroutine
MATN1*

Als Exot unter den Konvertierungsroutinen ist die Routine MATN1 zu nennen, die im Content-InfoObjekt 0MATERIAL Einsatz findet (siehe Anhang D.2). Die Arbeitsweise dieser Routine ist nicht statisch definiert, sondern wird vielmehr im Customizing durch die Transaktion OMSL festgelegt.

Probleme bereitet dieser Exot vor allem dann, wenn Daten aus anderen SAP-Systemen extrahiert werden, in denen die Konvertierungsroutine MATN1 anders definiert ist, und die interne Darstellung einer Materialnummer sich im BW und im SAP-Quellsystem unterscheiden. Denn trotz Unterschieden geht das BW bei der Durchführung von Ladevorgängen standardmäßig davon aus, dass Materialnummern im internen Format ungeprüft durchgereicht werden können. In der Folge treten Fehler z.B. bei der Verbuchung derartiger Daten in BasisCubes auf.

Sofern keine Anpassung des Customizing an der Transaktion OMSL möglich ist, besteht die Lösung auch an dieser Stelle in der Anwendung der Input-Konvertierung.

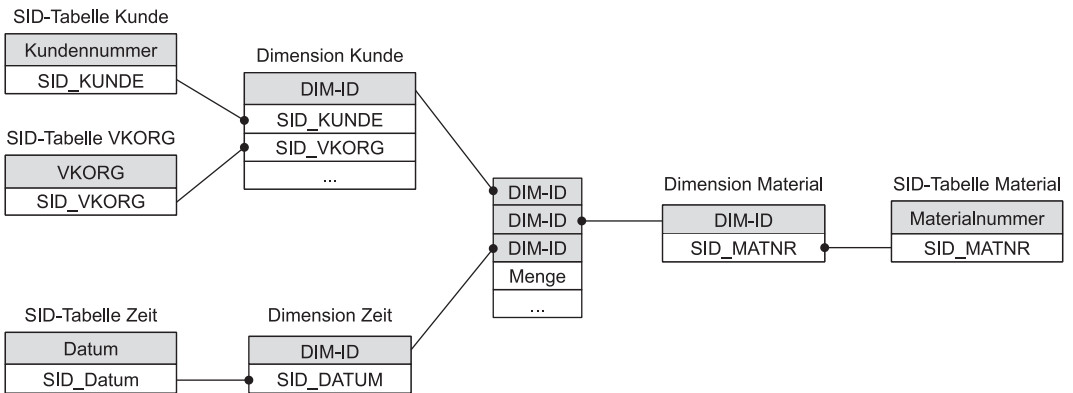
SID-Tabelle

Bei der Beschreibung des Star-Schemas wurde aufgezeigt, dass Dimensionen durch einen Dimensionsschlüssel (im BW als DIM ID bezeichnet) identifiziert werden und in der Dimension die Merkmalswerte abgelegt werden. Merkmale benötigten damit keine weitere Identifikation, sondern stellten sich lediglich durch ihren Wert dar.

Dieses Konzept wurde nicht ins BW übernommen. Stattdessen werden im BW alle Merkmalswerte durch eine Stammdaten-Identifikationsnummer (SID) identifiziert.

Damit ergeben sich gegenüber dem normalen Star-Schema Änderungen für das gesamte Datenmodell, da die echte Merkmalsausprägung nicht mehr in den Dimensionstabellen zu finden ist, sondern in den damit verknüpften SID-Tabellen der InfoObjekte (siehe Abb. 6–5).

Abb. 6–5
SID-Tabelle

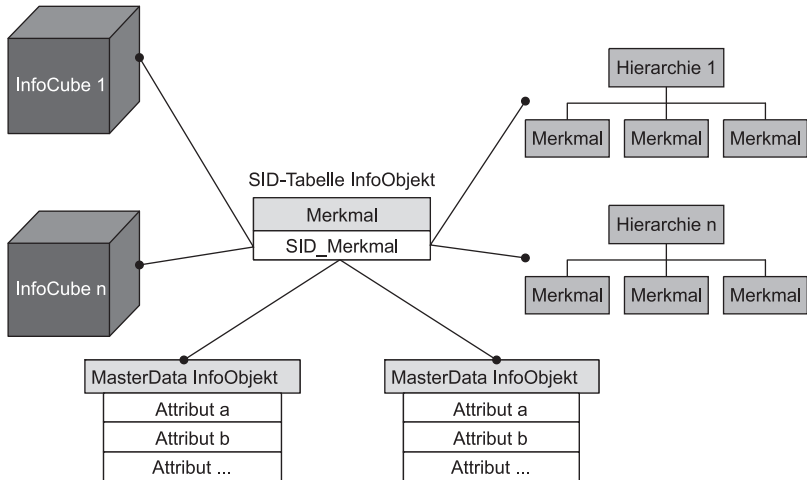


Durch die Einführung der SID-Tabellen beziehungsweise der SID werden im BW Merkmale vollständig von den BW-Objekten abgekapselt, welche die Merkmale nutzen. Damit ist es möglich, dieselbe SID-Tabelle eines Merkmales aus mehreren BW-Objekten zu referenzieren (siehe Abb. 6–6). Dazu gehören:

*Kapselung des
InfoObjektes*

- InfoCubes
- Hierarchien
- Master Data

Abb. 6-6
SID-Tabelle: Nutzung
in BW-Objekten



Diese Objekttypen werden in den nächsten Kapiteln detailliert beschrieben.

Die Tatsache, dass die SID immer vom Typ Integer (4 Byte Länge) ist, kann unter Umständen sogar trotz des aufwändigeren Datenmodells positiv für die Performance sein, da zum Beispiel eine 18-stellige Materialnummer nur noch als 4-Byte-Integerwert gespeichert wird.

Darüber hinaus bietet die SID die Möglichkeit, Merkmale mit einem zusammengesetzten Primärschlüssel zu verwalten.

*InfoObjekte mit
zusammengesetztem
Primärschlüssel
(Klammerung)*

Ein solcher Fall liegt zum Beispiel bei den Verkaufsdaten des Materialstammes im SAP ERP vor (Tabelle MVKE). Diese Daten werden nicht auf Ebene der Materialnummer abgelegt, sondern bilden sich aus der Kombination der Verkaufsorganisation, des Vertriebsweges und der Materialnummer. Diese bilden den Primärschlüssel der Tabelle MVKE im SAP ERP. Im Sprachgebrauch des BW heißt dies, dass die Materialnummer an die Verkaufsorganisation und den Vertriebsweg **geklammert** ist.

Ein derartiger Primärschlüssel kann in der SID-Tabelle berücksichtigt werden, ohne das Prinzip der SID zu verletzen. Dadurch vereinfacht sich aus Sicht der BW-Objekte, die ein InfoObjekt nutzen, der Zugriff auf dieses InfoObjekt, da in jedem Fall nur auf die SID-Tabelle des InfoObjektes zugegriffen werden muss.



Die Klammerung von InfoObjekten kann sich negativ auf die Performance bei der Datenaufbereitung sowie auch beim Reporting auswirken. Dies ist insbesondere dann der Fall, wenn mehrere InfoObjekte Teil der Klammerung sind. Setzen Sie daher die Klammerung nur dann ein, wenn dies durch das Datenmodell gerechtfertigt ist. Für alle weiteren Zwecke (Darstellung von Attributen, Hierarchien...) gibt es andere Funktionen im BW, die in den nachfolgenden Abschnitten beschrieben werden.

Die nachfolgende Tabelle stellt die SID-Tabelle für das InfoObjekt OMAT_SALES dar, das die Stammdaten des Materialstammes der Tabelle MVKE im BW abbildet.

Feldname	Datenelement	Erläuterung
SALESORG	/BIO/OISALESORG	Verkaufsorganisation
DISTR_CHAN	/BIO/OIDISTR_CHAN	Vertriebsweg
MAT_SALES	/BIO/OIMAT_SALES	Materialnummer an Verkauf geklammert
SID	RSSID	Stammdaten-ID
CHCKFL	RSDCHCKFL	Flag: Wert in Prüftabelle
DATAFL	RSDDATAFL	Flag: Wert in Dimension oder als Attribut vorhanden
INCLFL	RSDINCLFL	Flag: Wert ist in allen Inklusionstabellen eingebaut

Da ein InfoObjekt in mehreren anderen BW-Objekten zum Einsatz kommen kann, ist das Löschen von Einträgen der Master-Data-Tabellen (siehe unten) oder der SID-Tabelle dann kritisch, wenn aus BW-Objekten noch auf diese Einträge verwiesen wird. In diesen Fällen ist ein Löschen der Stammdaten nicht zulässig.

*Verwendungsnachweis
von SID*

Damit das BW-System informiert ist, ob Einträge der SID-Tabelle in Benutzung sind, werden in allen SID-Tabellen zwei entsprechende Flags protokolliert:

- DATAFL: Die SID wird in der Dimension eines InfoCubes oder als Attribut in den Master Data eines anderen InfoObjektes verwendet.
- INCLFL: Die SID wird in den Inklusionstabellen einer externen Hierarchie genutzt.

Sobald eine SID in einem InfoCube, den Master Data eines anderen InfoObjektes oder einer Hierarchie genutzt wird, wird das entsprechende Flag gesetzt (Wert »X«), und es ist nicht mehr möglich, die Stammdaten zu dieser SID zu löschen.

Der Nachteil dieses Konzepts liegt darin, dass die Flags zwar gesetzt werden, wenn eine SID verwendet wird, sie aber nicht wieder gelöscht werden, wenn die SID nicht mehr in Benutzung ist. Dies bedeutet, dass einmal verwendete Stammdaten nur mit zeitaufwändigen Prüfungen gelöscht werden können.

Der Name der SID-Tabelle wird durch das BW in Abhängigkeit vom InfoObjekt-Namen bestimmt. Die nachfolgende Tabelle zeigt die dabei verwendete Namenskonvention.

Namenskonvention

	Standard-InfoObjekt	Eigendefiniertes InfoObjekt
Name des InfoObjektes	Ottttttttt	{A-Z}ttttttt
Name der SID-Tabelle	/BIO/Stttttttttt	/BIC/S{A-Z}ttttttt

6.1.2 Zeiten

Neben Merkmalen sind für die Beschreibung betriebswirtschaftlicher Ordnungsgrößen auch *Zeiten* von Bedeutung. Grundsätzlich kommt sogenannten Zeitmerkmalen dieselbe Aufgabe zu wie Merkmalen. Dennoch sind im BW Zeiten als eigene InfoObjekt-Typen realisiert.

Auf diese Weise können beim Aufbau von Star-Schema-Modellen bereits Vorbelegungen für Zeitdimensionen vorgenommen, die Modelldefinition validiert und Konvertierungsroutinen (vgl. Kapitel 6.1.1) für die Definition des Staging bereitgestellt werden.

Während bei allen anderen InfoObjekt-Typen eigene InfoObjekte definiert werden können, können ausschließlich diejenigen Zeitmerkmale verwendet werden, die das BW mit dem BI Content ausliefert.

Die nachfolgende Tabelle zeigt die Zeitmerkmale des BI Content.

InfoObjekt	Beschreibung	Typ	Länge	Format
OCALDAY	Kalendertag	DATS	8	JJJJMMTT
OCALMONTH	Kalenderjahr/Monat	NUMC	6	JJJJMM
OCALMONTH2	Kalendermonat	NUMC	2	MM
OCALQUART1	Quartal	NUMC	1	Q
OCAL_QUARTER	Kalenderjahr/Quartal	NUMC	5	JJJJQ
OCALWEEK	Kalenderjahr/Woche	NUMC	6	JJJJWW
OCALYEAR	Kalenderjahr	NUMC	4	JJJJ
OFISCPER	Geschäftsjahr/Geschäftsmonat	NUMC	7	JJJJMMM
OFISCPER3	Geschäftsmonat	NUMC	3	MMM
OFISCVARNT	Geschäftsjahresvariante	CHAR	2	XX
OFISCYEAR	Geschäftsjahr	NUMC	4	JJJJ
OHALFYEAR1	Halbjahr	NUMC	1	H
OWEEKDAY1	Wochentag	NUMC	1	N

Wenn diese vorgegebenen Zeitmerkmale grundsätzlich den Anforderungen entsprechen, jedoch die Bezeichnung nicht gefällt, so sollte diese in der InfoObjekt-Pflege geändert werden. Das Umbenennen der Zeitmerkmale des BI Content stellt dabei weder im laufenden System noch in Bezug auf künftige Releasewechsel ein Problem dar.

Wenn grundsätzlich neue Zeitmerkmale angelegt werden sollen, so können normale Merkmale angelegt und diese als Zeitmerkmale verwendet werden. Aus inhaltlichen Gesichtspunkten stellt dies kein Problem dar, allerdings können die Vorteile echter Zeitmerkmale (Partitionierung der komprimierten Faktentabelle, Validierung auf gültige Werte und Typkonvertierung im Staging) nicht wahrgenommen werden.

Die Partitionierung der komprimierten Faktentabelle (siehe Kapitel 7.1) erfolgt nur anhand echter Zeitmerkmale aus dem BI Content. Vermeiden Sie es daher unbedingt, ausschließlich eigene Zeitmerkmale zu verwenden, da mit dem Verlust der Partitionierungsmöglichkeiten erhebliche Performanceprobleme verbunden sein können.



6.1.3 Kennzahlen

Ebenso wie bei Merkmalen/Zeitmerkmalen gilt die Definition des Datentyps als Grundeinstellung der Kennzahl. Da es sich bei Kennzahlen jedoch nicht um Ordnungsgrößen, sondern um Fluss- beziehungsweise Bestandsgrößen handelt, sind auch die Einstellungen grundsätzlich anders (siehe Abb. 6–7).

Abb. 6–7

*Kennzahldefinition:
Typ/Einheit*

Die nachfolgende Tabelle zeigt die zur Verfügung stehenden Datentypen.

Typ	Datentyp	mit Einheit	Beschreibung
Betrag	CURR	✓	Währungsfeld abgelegt als DEC
	FLTP	✓	Gleitpunktzahl mit 8 Byte Genauigkeit
Menge	QUAN	✓	Mengenfeld, zeigt auf Einheitenfeld mit Format UNIT
	FLTP	✓	Gleitpunktzahl mit 8 Byte Genauigkeit
Zahl	DEC		Rechen- oder Betragsfeld mit Komma und Vorzeichen
	FLTP		Gleitpunktzahl mit 8 Byte Genauigkeit
Integer	INT4		4-Byte-Integer, ganze Zahl mit Vorzeichen
Datum	DEC		Rechen- oder Betragsfeld mit Komma und Vorzeichen
	DATS		Datumfeld (JJJJMMTT), abgelegt als CHAR (8)
Zeit	DEC		Rechen- oder Betragsfeld mit Komma und Vorzeichen
	TIMS		Zeitfeld (hhmmss), abgelegt als CHAR (6)



Kennzahlen vom Typ DATS und TIMS werden im Datenbanksystem als Characterfeld abgelegt und eignen sich damit nicht für arithmetische Operationen. Aus diesem Grund müssen diese Kennzahlen bereits bei der Definition des InfoObjekts immer mit der Standardaggregation MAXIMUM oder MINIMUM angelegt werden (siehe Kapitel 6.1.3).

Ein besonderes Augenmerk bei der Definition von Kennzahlen ist auf die Verwendung von Einheiten zu legen.

Kennzahlen ohne Einheit

Kennzahlen vom Typ *Zahl*, *Integer*, *Datum* oder *Zeit* sind ohne Einheit. Der Umgang mit diesen Kennzahlen entspricht dem, was in den vorangegangenen Kapiteln zu Kennzahlen gesagt wurde. Sie sind zur Aufnahme in die Faktentabelle des Star-Schemas bestimmt und können dort mit Hilfe einfacher mathematischer Operationen gespeichert/addiert werden.



Die Typen *Datum* und *Zeit* sind nicht zu verwechseln mit Zeitmerkmalen! Hier handelt es sich nicht um Ordnungsmerkmale, sondern lediglich um Kennzahlen. Diese können dort übernommen oder auch addiert werden, eignen sich aber nicht als Ordnungsmerkmal.

Kennzahlen mit fixer Einheit

Für Kennzahlen vom Typ Betrag oder Menge muss entweder eine fixe oder eine variable Einheit definiert werden. Die Definition von Einheiten gewinnt im Bereich Extraktion und Staging beziehungsweise der Datenanalyse an Bedeutung, wenn die abgespeicherten Kennzahlen-

werte in Bezug zu einer Einheit gesetzt oder (im Fall von Währungseinheiten) in andere Einheiten umgerechnet werden sollen (zum Beispiel Umrechnung alter DM-Daten in EUR).

Soll die Einheit einer solchen Kennzahl fest vorgegeben werden (zum Beispiel EUR, USD...), so kann dies bei der Pflege der Kennzahl erfolgen (siehe Abb. 6–8).

The screenshot shows the SAP 'Kennzahl 0AMOUNT ändern: Details' dialog box. The 'Typ/Einheit' tab is selected, showing the following configuration:

- Kennzahl:** 0AMOUNT
- Beschreibung lang:** Betrag
- Beschreibung kurz:** Betrag
- Version:** aktiv
- Objektstatus:** aktiv, ausführbar
- Typ/Datentyp:**
 - Betrag
 - Menge
 - Zahl
 - Integer
 - Datum
 - Zeit
- Datentyp:** CURR - Währungsfeld, abgelegt als DEC
- Währung/Mengeneinheit:**
 - Feste Währung:** EUR
 - Feste Mengeneinheit:** Europäischer Euro
 - Einheit / Währung:** (empty field)

© SAP AG

Abb. 6–8
Kennzahlen mit
fixer Einheit

Durch die feste Vorgabe einer Einheit muss die Einheit später im Datenmodell nicht mehr explizit gespeichert werden, da sie der Definition der Kennzahl entnommen werden kann.

Die Verwendung von Kennzahlen mit fixer Einheit setzt voraus, dass ggf. beim Speichern der Daten eine Umrechnung stattfindet, wenn die gelieferten Daten nicht ebenfalls mit derselben festen Einheit arbeiten. Diese Umrechnung ist möglich und wird in Kapitel 19.3.7 näher erläutert.



Soll eine Kennzahl mit unterschiedlichen Einheiten im Datenmodell gespeichert werden, so muss bei der Definition der Kennzahl ein weiteres InfoObjekt angegeben werden, welches überall dort, wo die Kennzahl abgelegt wird, zusätzlich die dazugehörige Einheit ablegt. Zu diesem Zweck kann ein Einheiten-InfoObjekt (zum Beispiel 0CURRENCY) angegeben werden (vgl. Abb. 6–7 auf Seite 73).

*Kennzahlen mit variabler
Einheit*

Verwenden mehrere Kennzahlen gleichzeitig dasselbe Einheiten-InfoObjekt, so werden die Kennzahlen im Datenmodell immer dann in unterschiedlichen Datensätzen abgelegt, wenn sie andere Einheiten aufweisen. Dies hat Auswirkungen auf das Datenvolumen, jedoch sind die Daten in jedem Fall korrekt abgelegt.

Aggregationsverhalten von Kennzahlen

Bewegungsdaten werden auf dem Weg von den OLTP-Systemen über die BasisCubes bis zur Datenanalyse mehrmals verdichtet. Dabei werden die Kennzahlen auf eine höher verdichtete Detaillierungsstufe aggregiert.

Standardaggregation

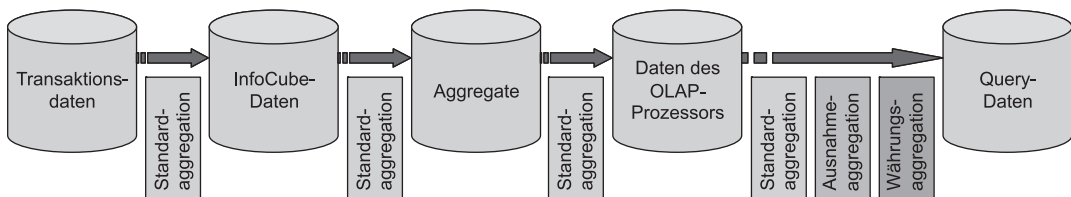
Als Standardaggregation wird eine einfache Summation oder die Ermittlung von Minimal-/Maximalwerten verwendet. Diese Art der Aggregation kann bereits auf Datenbankebene durchgeführt werden, so dass die Implementierung einfach und performant ist. Darüber hinaus bilden diese Aggregationsformen die Grundlage für weitere Berechnungen.

Aufbauend auf der Standardaggregationen können im Rahmen der Datenanalyse weitere Aggregationen von Bedeutung sein. Dabei handelt es sich um komplexe Aggregationsverfahren mit einem Bezug auf andere InfoObjekte (Ausnahmeaggregation) sowie um Währungsaggregation.

Diese Aggregationen werden speziell für die Datenanalyse bei BasisCubes (siehe Kapitel 6.3) zur Verfügung gestellt und im Anschluss an die Standardaggregationen, aber noch vor der Übergabe der Daten an das Analysetool durchgeführt. Abbildung 6–9 stellt die Stufen des Aggregationsverhaltens dar.

Abb. 6–9

Stufen des Aggregationsverhaltens



Bei anderen Datenzielen als den BasisCubes wird ausschließlich die Standardaggregation durchgeführt. Dies ist darin begründet, dass BasisCubes konzeptionell als Grundlage für das Reporting vorgesehen sind, andere Datenziele aber nur zweitrangig der Datenanalyse dienen.

Nachfolgend werden die Möglichkeiten beschrieben, die mit Ausnahmeaggregation und Währungsaggregation zur Verfügung stehen.

Mit Hilfe der Ausnahmeaggregation kann eine komplexere Aggregation von Kennzahlen ausgeführt werden. Typische Beispiele für eine Ausnahmeaggregation ist die Berechnung

- des durchschnittlichen Lagerbestands pro Monat
- des durchschnittlichen Umsatzes pro Kunde
- der Anzahl an Aufträgen pro Monat

Derartige Operationen sind nur unter der Angabe eines anderen InfoObjektes sinnvoll, das als Bezugsgröße verwendet wird. Als Bezugsgrößen können sowohl normale InfoObjekte (sachliche Aggregation) als auch Zeitmerkmale (zeitliche Aggregation) verwendet werden.

Die Ausnahmeaggregation wird pro Kennzahl in der InfoObjekt-Pflege definiert (siehe Abb. 6–10).

Ausnahmeaggregation

The screenshot shows the SAP 'Kennzahl ZNONKUM anlegen: Details' dialog box. The 'Aggregation' tab is active, showing the following settings:

- Kennzahl: ZNONKUM
- Beschreibung lang: Kennzahl mit Ausnahmeaggregation
- Beschreibung kurz: Kennz. m. Ausn.Aggr.
- Version: neu (nicht gesichert)
- Objektstatus: inaktiv, nicht ausführbar

The 'Aggregation' section is expanded, showing:

- Aggregation: SUM
- Ausnahmeaggregation: Durchschnitt (alle Werte)
- Agg Bezugsmerkmal: @CALDAY (Kalendertag)

The 'Fluss-/Bestandsgrößen' section is also expanded, showing:

- Flußgröße
- Bestand mit Bestandveränderung (Bestandveränderung:)
- Bestand mit Zu- und Abgang (Zugang: , Abgang:)

The status bar at the bottom shows 'BI7 (2) 001 qdxappl INS'.

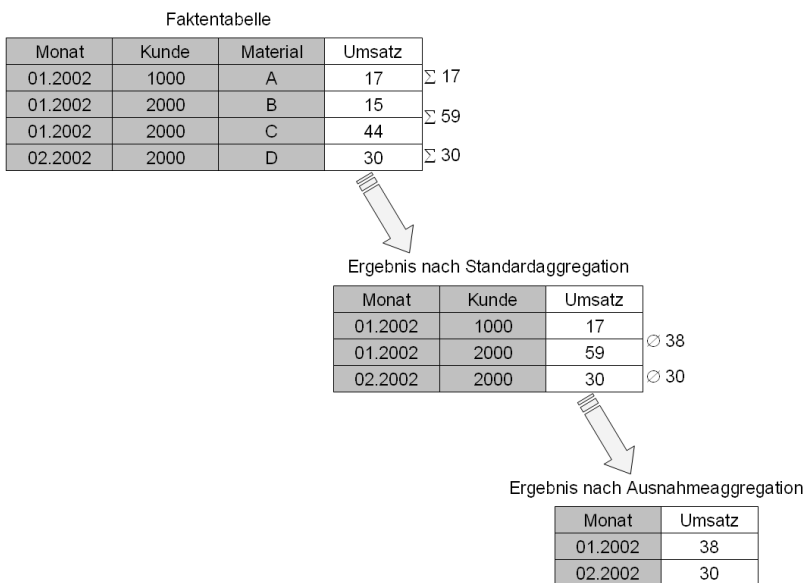
Abb. 6–10

Kennzahlen mit Ausnahmeaggregation

Bei der Anwendung der Ausnahmeaggregation führt die Analytical Engine zuerst die Standardaggregation und erst im Anschluss die Ausnahmeaggregation für das jeweilige Bezugsmerkmal durch.

Abb. 6–11

Aggregationsstufen
der Analytical Engine



Der Vorgang der Ausnahmeaggregation wird in Abbildung 6–11 beispielhaft an der Kennzahl UMSATZ dargestellt. Die Kennzahl besitzt die Standardaggregation SUMMATION und die Ausnahmeaggregation DURCHSCHNITT mit dem Bezugsmerkmal KUNDE. Die Analytical Engine stellt diese Kennzahl (durchschnittlicher Umsatz pro Kunde) pro Monat für die Datenanalyse bereit.

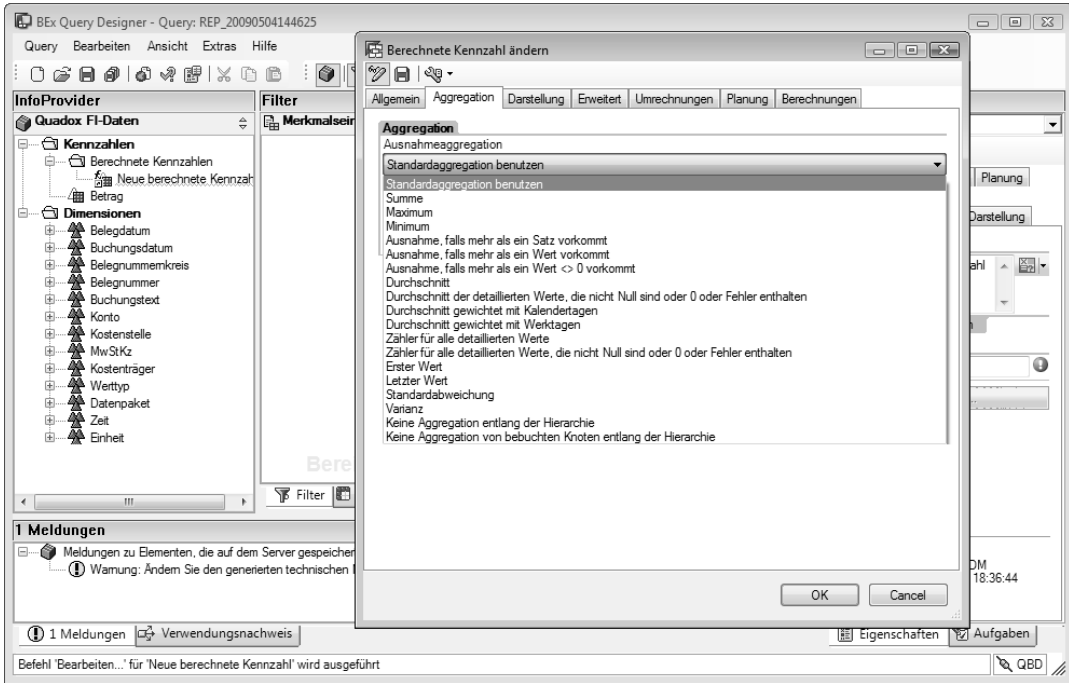
Bezieht sich eine Ausnahmeaggregation wie in Abbildung 6–10 auf eine Flussgröße, so kann alternativ zur Definition einer eigenen Kennzahl in der Faktentabelle eines Cubes auch eine berechnete Kennzahl im Query-Designer definiert werden. Die Ausnahmeaggregation ist in diesem Fall in den Eigenschaften der berechneten Kennzahl zu definieren (siehe Abbildung 6–12).

Eine besondere Form der Ausnahmeaggregation stellen Bestandsveränderungen dar. Dabei handelt es sich um Bestandskennzahlen, die nicht im Datenmodell gespeichert, sondern lediglich auf Basis von Bestandsveränderungen errechnet werden können, die im BasisCube abgelegt sind..

Ausnahmeaggregation
mit Bestandsveränderung



Bestandskennzahlen werden im Gegensatz zu allen anderen Kennzahlen nicht physisch im Datenmodell des BasisCubes abgelegt, sondern vollständig aus Kennzahlen berechnet, welche Bestandsveränderungen darstellen.



© SAP AG

Abhängig davon, in welcher Form die Bestandserhöhungen bzw. -minderungen in einem BasisCube abgelegt sind, wird zwischen zwei Arten von Bestandskennzahlen unterschieden:

- **Bestandskennzahlen mit Bestandsveränderungen:** Die Bestandskennzahl errechnet sich aus genau einer anderen Kennzahl, die sowohl Bestandszugänge als auch Bestandsabgänge abbildet.
- **Bestandskennzahlen mit Zu- und Abgängen:** Die Bestandskennzahl errechnet sich aus genau zwei anderen Kennzahlen. Dabei bildet eine der beiden Kennzahlen die Bestandszugänge ab und die andere Kennzahl die Bestandsabgänge.

Je nach Art der vorliegenden Bestandskennzahl müssen in der Definition des InfoObjektes diejenigen Kennzahlen festgelegt werden, welche die Bestandsveränderungen bzw. Zu- und Abgänge beinhalten (siehe Abb. 6–13). Die Kennzahlen, die die Bestandsveränderungen darstellen, müssen zwingend in jeden BasisCube aufgenommen werden, in dem die Bestandskennzahl definiert ist.

Abb. 6–12
Kennzahlen mit
Ausnahmeaggregation
im Query-Designer

Abb. 6-13
Ausnahmeaggregation
mit Bestandsveränderung

Kennzahl ZNONKUM anlegen: Details

Kennzahl: ZNONKUM

Beschreibung lang: Kennzahl mit Ausnahmeaggregation

Beschreibung kurz: Kennz. m. Ausn.Aggr.

Version: neu (nicht gesichert)

Objektstatus: inaktiv, nicht ausführbar

Typ/Einheit | Aggregation | Weitere Eigenschaften

Aggregation

Aggregation: SUM

Ausnahmeaggregation: Durchschnitt (gewichtet mit Anzahl Tagen)

Fluss-/Bestandsgrößen

Flußgröße

Bestand mit Bestandsveränderung

Bestandsveränderung: ZBESTVA (Bestandsmenge)

Bestand mit Zu- und Abgang

Zugang: []

Abgang: []

BI7 (2) 001 | qdxappl | INS

© SAP AG

Aus Sicht des Bestands ist es unerheblich, welche Art von Bestandskennzahl verwendet wird. Allerdings ist zu bedenken, dass eine Trennung der Bestandsveränderungen in Zu- und Abgänge neben der Betrachtung des Bestandes auch eine Betrachtung von Zu- und Abgängen ermöglicht, da die entsprechenden Kennzahlen im BasisCube verfügbar sind.

Bezugsmerkmale

Anders als bei der normalen Ausnahmeaggregation werden bei Bestandskennzahlen nicht nur ein beliebiges Bezugsmerkmal, sondern auch genau ein zeitliches Bezugsmerkmal und optional mehrere sachliche Bezugsmerkmale definiert.

Als zeitliches Bezugsmerkmal kann ausschließlich ein Zeit-Info-Objekt aus der Zeitdimension gewählt werden. Die Bezugsmerkmale zu einer Bestandskennzahl werden nicht bei der Pflege des InfoObjektes (wie bei der normalen Ausnahmeaggregation), sondern pro BasisCube festgelegt. Die Festlegung der Bezugsmerkmale wird in der Pflege des BasisCubes unter *Zusätze* → *Bestandsparameter pflegen* vorgenommen (siehe Abb. 6-14). Dies kann nur dann erfolgen, wenn der BasisCube keine Daten enthält.

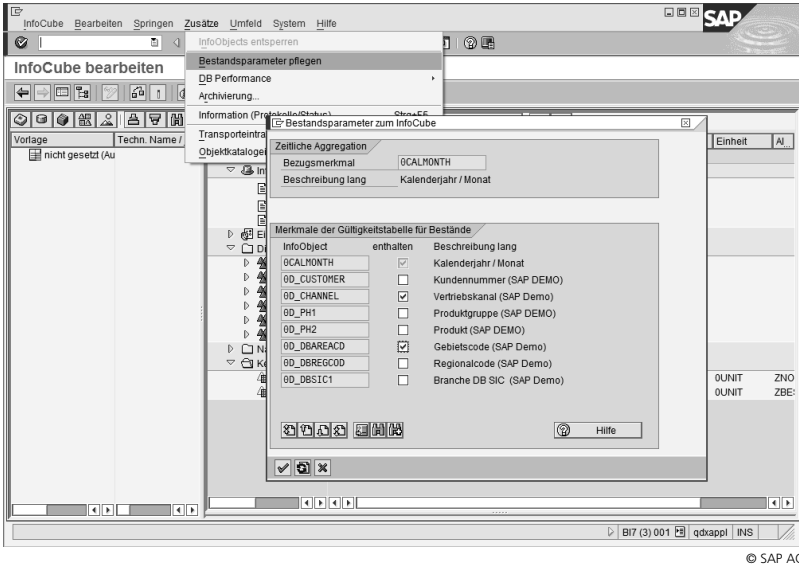


Abb. 6-14

Bezugsmerkmale von BestandsCubes

Die Kardinalität der Bezugsmerkmale beeinflusst maßgeblich die Komplexität der SQL-Kommandos, die für Analysen ausgeführt werden müssen und hat damit unmittelbar Auswirkung auf die Performance der Datenanalyse. Verwenden Sie daher nach Möglichkeit keine InfoObjekte mit mehr als 20–30 Merkmalsausprägungen als Bezugsmerkmal.



Zusätzlich zur Festlegung der Bezugsmerkmale muss zum entsprechenden BasisCube eine Gültigkeitstabelle gepflegt werden, die Zeitintervalle für die möglichen Ausprägungen der Bezugsmerkmale beschreibt (z.B. die Berechnung des durchschnittlichen Lagerbestands für Werk 2000 und Materialgruppe 002 vom 01.01.2007 bis 01.03.2007). Dies erfolgt mit Hilfe der Transaktion RSDV.

Die Zeitintervalle müssen vor dem Laden von Daten in einen BasisCube gepflegt werden. Sind bereits Daten in einem BasisCube vorhanden, so kann die Gültigkeitstabelle mit Hilfe des Programms RSDG_CUBE_VALT_MODIFY angepasst werden.

Im Anschluss an alle Standardaggregationen und die Ausnahmeaggregation wird die Währungsaggregation ausgeführt. Soll also eine Währungsumrechnung im Rahmen der Datenanalyse durchgeführt werden, so wird diese erst unmittelbar vor der Datenübergabe an das Analysetool durchgeführt. Findet bei der Analyse keine Währungsumrechnung statt, so entfällt die Währungsaggregation.

Währungsaggregation

6.1.4 Einheiten

Einheiten-InfoObjekte werden immer als Zusatzinformation für einen Betrag oder eine Menge verwendet. Damit steht fest, dass es sich bei einer Einheit immer entweder um eine Währung oder um eine Mengeneinheit handelt und die möglichen Ausprägungen bei jeder Einheit immer dieselben sind (EUR, USD bzw. Stück, Kilogramm, Paletten usw.).

Aus diesem Grund ist beim Anlegen eines Einheiten-InfoObjektes ausschließlich zu definieren, wie es heißt und ob es eine Einheit oder eine Währung ist (siehe Abb. 6–15).



Technisch stehen hinter allen Einheiten die beiden InfoObjekte OCURRENCY oder OUNIT, welche die zur Verfügung stehenden Einheiten vorgeben und von allen Einheiten referenziert werden (Näheres zur Referenzierung von InfoObjekten in Kapitel 6.2.4).

Abb. 6–15
Anlegen von
Einheiten-InfoObjekten

6.2 Master Data

Für die aktuelle und stichtagsbezogene Darstellung von Attributen zu Merkmalen existiert im BW das Konzept der *Master Data*. Dieser Begriff ist bereits bei der Beschreibung der SID-Tabelle häufiger gefallen und wird nun eingehend erläutert.

Bei den Master Data handelt es sich um eine erweiterte Form der Nutzung von Merkmals-InfoObjekten, die mit der Speicherung von Stammdaten in einem SAP-ERP-System zu vergleichen ist.

Master Data umfassen:

- Texte eines Merkmals
- Stammdaten (Attribute) eines Merkmals
- externe Hierarchien

Die Dateninhalte der Master Data werden in der Regel aus den operativen Quellsystemen extrahiert (zum Beispiel Namen und Attribute von Kunden), können jedoch auch direkt im BW gepflegt werden.

6.2.1 Texte

Bei Texten handelt es sich um beschreibende Informationen zu Merkmalsausprägungen (zum Beispiel Kundennamen, Produktbezeichnungen), die auch Kleinbuchstaben und Sonderzeichen enthalten können.

Die Einstellung bezüglich der Texte eines InfoObjektes wird in der InfoObjekt-Pflege über die Registerkarte *Stammdaten/Texte* vorgenommen (siehe Abb. 6–16 auf Seite 84).

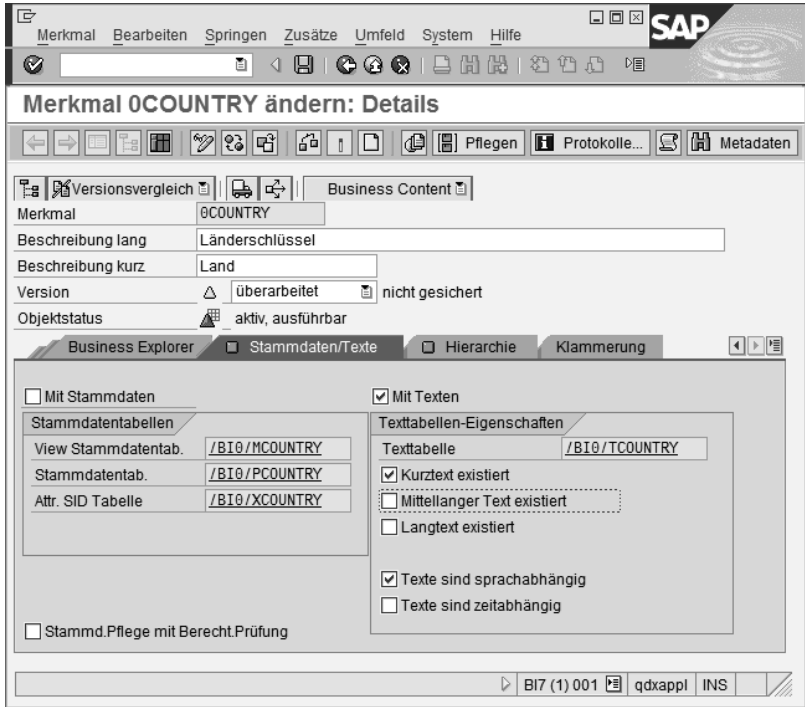
Dabei ist nicht nur festzulegen, ob ein Merkmal über Texte verfügen soll, sondern es sind auch Angaben zur Art der Texte zu machen.

Dies umfasst die Länge der Textinformationen (Kurztext, mittellanger Text, Langtext). Es ist mindestens eine Auswahl zu treffen. Dabei sollte immer darauf geachtet werden, dass das Quellsystem, aus dem die Textinformationen geladen werden, auch in der Lage ist, die ausgewählten Textinformationen zu liefern. So ist es nur dann sinnvoll, sämtliche Optionen zu wählen, wenn das Quellsystem auch alle Texte in kurzer, mittellanger und langer Form vorhält.

Textlänge

	Textlänge
Kurztext	20 Zeichen
Mittellanger Text	40 Zeichen
Langtext	60 Zeichen

Abb. 6-16
Eigenschaften von
Texttabellen



© SAP AG

Sprachabhängigkeit

Des Weiteren sind Angaben über die Sprachabhängigkeit von Textinformationen zu treffen. Bei aktivierter Sprachabhängigkeit wird in der Texttabelle des InfoObjektes zusätzlich ein Schlüsselfeld für die Sprache angelegt. Wie auch bei der Textlänge ist es nur dann sinnvoll, Texte sprachabhängig zu hinterlegen, wenn das Quellsystem, aus dem die Texte gelesen werden, die Texte auch in mehreren Sprachen liefern kann.

Da sich für jede zusätzlich abgelegte Sprache die Anzahl der Datensätze in der Texttabelle vervielfachen kann, sollte die Sprachabhängigkeit nur dann gewählt werden, wenn sie für das Reporting einen tatsächlichen Mehrwert darstellt.



Der BI Content definiert zahlreiche InfoObjekte (zum Beispiel auch den Kunden) mit sprachabhängigen Texten. Überlegen Sie hier bitte, ob Sie diese Einstellung tatsächlich benötigen oder die Einstellung zurücknehmen können.

Zeitabhängigkeit

Da Texte wie Stammdaten gespeichert und regelmäßig aktualisiert werden, erscheinen sie bei der Datenanalyse immer mit den jeweils aktuellen Ausprägungen. Sollen Texte hingegen stichtagsbezogen dargestellt werden (vgl. Kapitel 5.1.2), so kann dies über zeitabhängige Texte realisiert werden.

Ist diese Option aktiviert, so wird der Texttabelle als zusätzliches Schlüsselfeld ein Gültigkeitszeitraum hinzugefügt. Da mit jedem Laden von Daten ein neuer Gültigkeitszeitraum hinzukommt, kann sich die Anzahl der Datensätze in der Texttabelle schnell vervielfachen. Da ein derartiges Wachstum negative Auswirkungen auf die Performance hat, sollten zeitabhängige Texttabellen nur bei kleinen Texttabellen eingesetzt werden, und auch nur dann, wenn es aus Sicht der Datenanalyse einen wirklichen Mehrwert darstellt.

Der Stichtag von Texten richtet sich nicht nach dem Änderungsdatum der Texte im operativen System, sondern nach dem Datum des Ladevorganges im BW. Ein nachträglicher Aufbau stichtagsbezogener Texte ist daher nicht möglich.



Namenskonvention

Auf der Basis des InfoObjekt-Namens wird bei der Aktivierung der Metadaten eines InfoObjektes automatisch eine transparente Tabelle für die Aufnahme der Textinformationen angelegt.

	Standard-InfoObjekt	eigendefiniertes InfoObjekt
Name des InfoObjektes	0tttttttt	{A-Z}ttttttt
Name der Texttabelle	/BI0/Ttttttttt	/BIC/T{A-Z}ttttttt

Die Struktur der Tabelle leitet sich aus der Art der Textinformationen ab. Eine Texttabelle mit allen Optionen (alle Textlängen, Sprachabhängigkeit, Zeitabhängigkeit) würde für das InfoObjekt 0COUNTRY im ABAP Dictionary definiert sein wie in Abb. 6–17 dargestellt.

The screenshot shows the SAP ABAP Dictionary interface for a transparent table. The table name is /BI0/TCOUNTRY, which is active. The short description is 'Texts: Char. Country'. The 'Felder' (Fields) tab is selected, showing a list of fields with their properties:

Feld	Key	Initi.	Datenelement	Datentyp	Länge	DezSt.	Kurzbeschreibung
COUNTRY	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	/BI0/0ICOUNTRY	CHAR	3		0 Länderschlüssel
LANGU	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	LANGU	LANG	1		0 Sprachenschlüssel
DATETO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	RSDATETO	DATS	8		0 Datum gültig bis
DATEFROM	<input type="checkbox"/>	<input checked="" type="checkbox"/>	RSDATEFROM	DATS	8		0 Datum gültig ab
TXTSH	<input type="checkbox"/>	<input checked="" type="checkbox"/>	RSTXTSH	CHAR	20		0 Beschreibung kurz
TXTMD	<input type="checkbox"/>	<input checked="" type="checkbox"/>	RSTXTMD	CHAR	40		0 Beschreibung mittel
TXTLG	<input type="checkbox"/>	<input checked="" type="checkbox"/>	RSTXTLG	CHAR	60		0 Beschreibung lang

The interface also shows navigation options like 'Suchhilfe' and 'Eingebauter Typ'.

Abb. 6–17

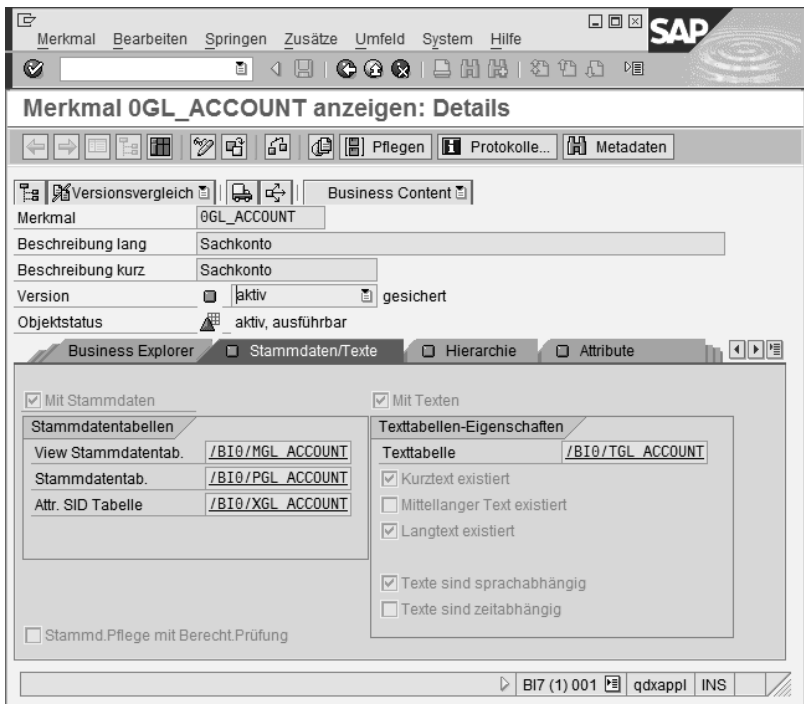
Texttabellen im
ABAP Dictionary

6.2.2 Stammdaten eines Merkmals

Bei den Stammdaten eines Merkmals (in diesem Zusammenhang auch als Basismerkmal bezeichnet) handelt es sich um zusätzliche Informationen (Attribute) zu Merkmalsausprägungen. Im Gegensatz zu Texten können diese Attribute nicht nur als beschreibende Information verwendet werden, sondern auch als eigenständige Elemente der Datenanalyse, über die selektiert und navigiert werden kann.

Ebenso wie die Einstellungen bezüglich der Texte eines InfoObjektes werden die Einstellungen für Stammdaten in der InfoObjekt-Pflege über die Registerkarte *Stammdaten/Texte* vorgenommen (siehe Abb. 6–18).

Abb. 6–18
Eigenschaften von
Stammdaten



© SAP AG

Im Rahmen der InfoObjekt-Pflege werden eine Reihe von transparenten Tabellen erzeugt, die für die Aufnahme und Verwaltung der Stammdaten zuständig sind. Die Struktur dieser Tabellen ist davon abhängig, welche Einstellungen beim InfoObjekt getroffen werden. Ausschlaggebend dafür sind Einstellungen über

- Klammerung
- zeitkonstante Attribute
- zeitabhängige Attribute
- Navigationsattribute

Die Klammerung wurde bereits bei der Erläuterung der SID-Tabelle als mögliche Abbildung eines zusammengesetzten Primärschlüssels erwähnt.

Durch die Festlegung einer Klammerung wird in allen relevanten Stammdatentabellen eines InfoObjektes das geklammerte InfoObjekt zusätzlich als Feld des Primärschlüssels aufgenommen. Die Festlegung einer Klammerung wird in der InfoObjekt-Pflege in der Registerkarte *Klammerung* vorgenommen (siehe Abb. 6–19).

Klammerung

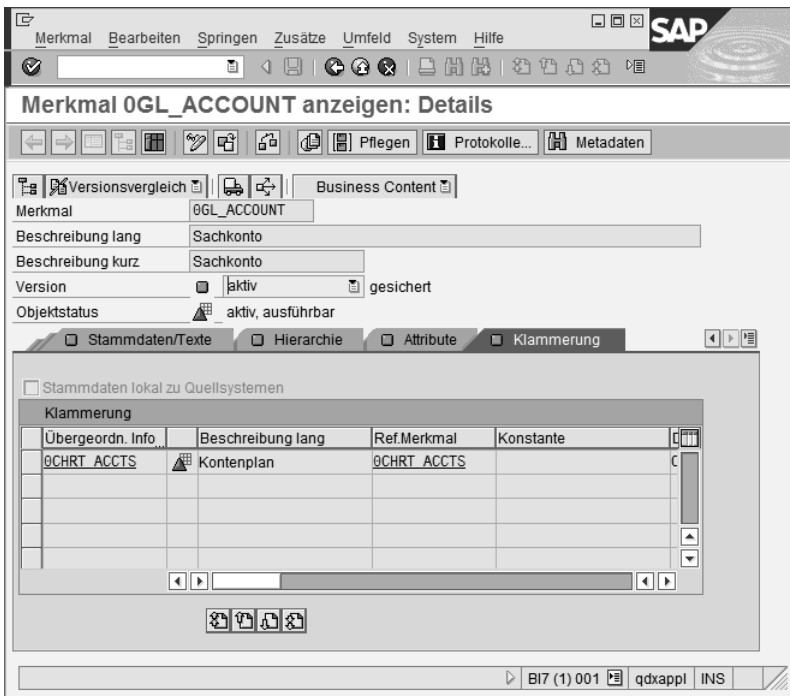


Abb. 6–19

*Klammerung
von InfoObjekten*

Ein hilfreiches Feature der Klammerung ist die Option, Stammdaten lokal zum Quellsystem anzulegen. Ist diese Option aktiviert, so wird die Quellsystem-ID als Klammerung hinzugefügt und bei Ladevorgängen automatisch gefüllt. Dies ist dann sinnvoll, wenn Stammdaten für dasselbe Merkmal aus unterschiedlichen Quellsystemen geladen werden und diese Quellsysteme gleiche Merkmalswerte liefern, obwohl damit unterschiedliche Objekte beschrieben werden.



Durch die Attribute eines InfoObjektes werden die Informationen bestimmt, welche die Stammdaten zusätzlich zu dem InfoObjekt bieten. Diese Informationen sind dabei zunächst ausschließlich als Anzeigemöglichkeit bestimmt (zum Beispiel, um neben der Kundennummer auch dessen Postleitzahl anzuzeigen) und nicht geeignet, um in Analy-

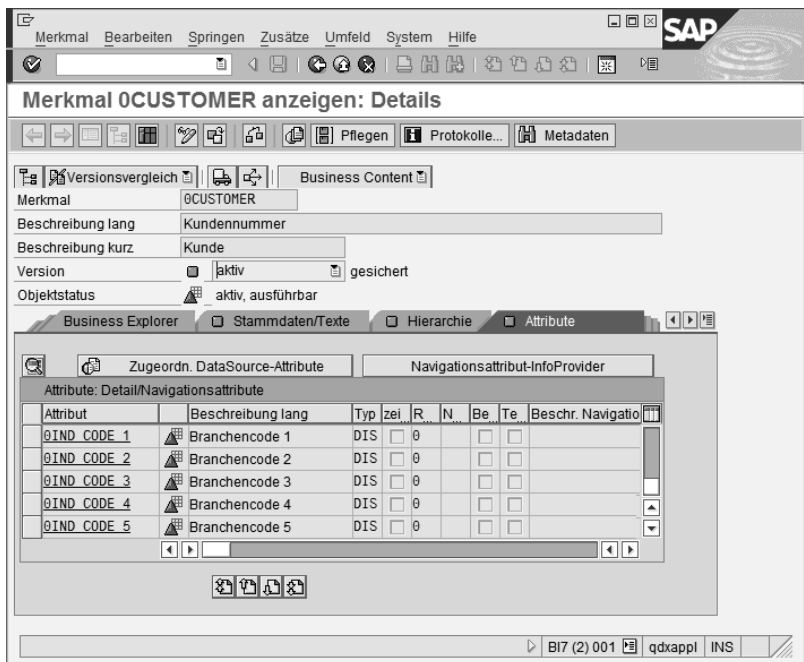
Zeitkonstante Attribute

sen als Filter eingesetzt zu werden oder um einen Drill Down auf diese Attribute durchzuführen (vergleichbar mit Texten). Aus diesem Grund werden diese Attribute als *Anzeigeattribute* bezeichnet.

Da die Attributdaten jeweils in der aktuellen Form zum InfoObjekt vorliegen und keine Abhängigkeit zu einer Zeitachse aufweisen, sind sie *zeitkonstant*.

Bei Attributen handelt es sich um andere existierende InfoObjekte, die dem Master-Data-InfoObjekt zugeordnet werden (siehe Abb. 6–20). Jedes InfoObjekt, das über Stammdaten verfügt, kann weitere Info-Objekte aufnehmen und damit als Master-Data-InfoObjekt eingesetzt werden.

Abb. 6–20
Anzeigeattribute eines
InfoObjektes



© SAP AG

Dabei kann es möglich sein, dass es sich bei einem zugeordneten Attribut um ein InfoObjekt handelt, das seinerseits ebenfalls Attribute besitzt. Diese Attribute können damit auch als Attribut des Master-Data-InfoObjektes genutzt werden, um tief verschachtelte Attribut-Abhängigkeiten aufzubauen. Dies gilt allerdings nur für Anzeige-, nicht für Navigationsattribute (siehe Abb. 6–20).

Die Stammdatentabelle zu einem InfoObjekt wird automatisch bei der Aktivierung der Metadaten des InfoObjektes angelegt und folgt der nachstehend beschriebenen Namenskonvention.

	Standard-InfoObjekt	eigendefiniertes InfoObjekt
Name des InfoObjektes	Otttttttttt	{A-Z}tttttttt
Stammdatentabelle (zeitkonstant)	/BIO/Ptttttttttt	/BIC/P{A-Z}tttttttt

Die zeitkonstanten Anzeigeattribute eines InfoObjektes werden in Klarform (das heißt in Form ihrer tatsächlichen Ausprägung und nicht in Form der SID) in der zeitkonstanten Stammdatentabelle des InfoObjektes gespeichert.

The screenshot shows the SAP ABAP Dictionary interface for the table `/BIO/PGL_ACCOUNT`. The table is active and its structure is displayed in a table view. The fields listed are:

Feld	Key	init.	Datenelement	Datentyp	Länge	DezSt.	Kurzbeschreibung
CHRT_ACCTS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	/BIO/OICHT ACCTS	CHAR	4		0 Kontenplan
GL_ACCOUNT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	/BIO/OIGL_ACCOUNT	CHAR	10		0 Sachkonto
OBJVERS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	RSOBJVERS	CHAR	1		0 Objektversion
CHANGED	<input type="checkbox"/>	<input checked="" type="checkbox"/>	RSRCHANGEFLAG	CHAR	1		0 Änderungsflag (1 eingefügt / D gelöscht)
BAL_FLAG	<input type="checkbox"/>	<input checked="" type="checkbox"/>	/BIO/OIBAL_FLAG	CHAR	1		0 Kennzeichen: Sachkonto ist Bestandskonto
INCST_FLAG	<input type="checkbox"/>	<input checked="" type="checkbox"/>	/BIO/OIINCST_FLAG	CHAR	2		0 Kennzeichen: Sachkonto ist Erfolgskonto
LOGSYS	<input type="checkbox"/>	<input checked="" type="checkbox"/>	RSDLGYS	CHAR	10		0 Quellsystem
SEM_POSIT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	/BIO/OISEM_POSIT	CHAR	16		0 Planposition
SOURSYSTEM	<input type="checkbox"/>	<input checked="" type="checkbox"/>	RSSOURSYSID	CHAR	2		0 Quellsystem ID
GLACCEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	/BIO/OIGLACCEXT	CHAR	18		0 Bilanz/GuV-Position

Abb. 6–21

Struktur zeitkonstanter
Master-Data-Tabellen

In Abbildung 6–21 wird die Stammdatentabelle des InfoObjektes `OGI_ACCOUNT` dargestellt, wie sie im ABAP Dictionary angelegt ist. An der Stammdatentabelle ist zu erkennen, dass das InfoObjekt `OCHRT_ACCTS` an das InfoObjekt `OGI_ACCOUNT` geklammert ist (Primärschlüssel) und die InfoObjekte `OBAL_FLAG`, `INCST_FLAG` usw. als Attribute verwendet werden.

Ebenso wie Texte zeitabhängig gespeichert werden können, besteht auch bei Attributen die Möglichkeit der stichtagsbezogenen Speicherung. Die Einstellung zur Zeitabhängigkeit für Attribute wird in der InfoObjekt-Pflege über »Detail/Navigationsattribute« eingestellt, indem die Option »zeitabhängig« der Attribute aktiviert wird.

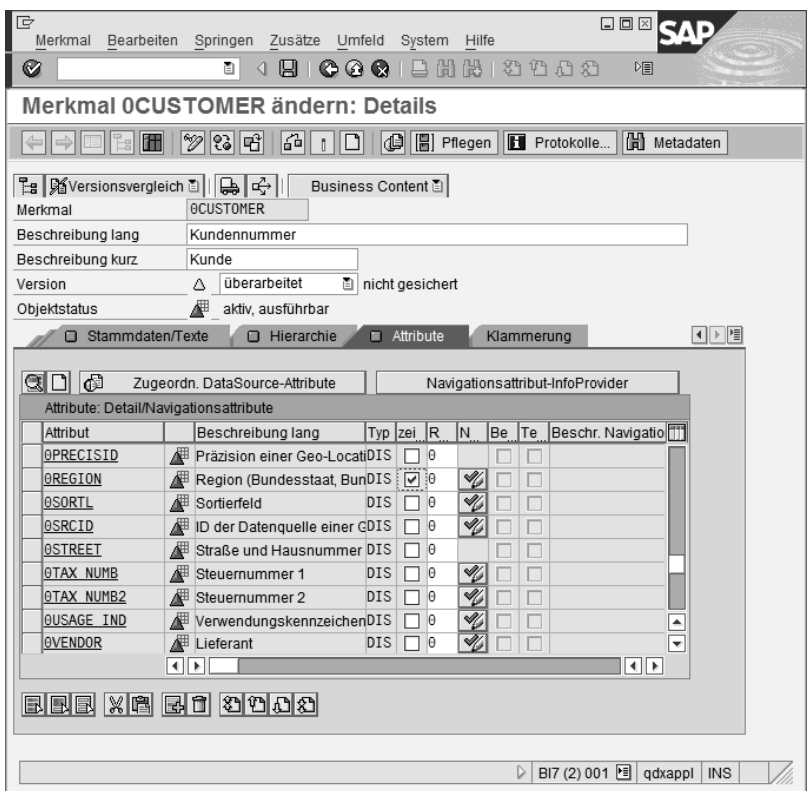
Anders als bei den Texten wird die Zeitabhängigkeit von Attributen nicht für alle Attribute getroffen, sondern kann für jedes Attribut individuell festgelegt werden (siehe Abb. 6–22).

Zeitabhängige Attribute

Um diese individuelle Zeitabhängigkeit zu realisieren, existiert neben der oben beschriebenen Stammdatentabelle für zeitkonstante Attribute eine weitere Stammdatentabelle für zeitabhängige Attribute.

	Standard-InfoObjekt	eigendefiniertes InfoObjekt
Name des InfoObjektes	0ttttttttt	{A-Z}ttttttt
Stammdatentabelle (zeitabhängig)	/BIO/Qttttttttt	/BIC/Q{A-Z}ttttttt

Abb. 6-22
Zeitabhängigkeit von
Attributen



Je nachdem, ob ein Attribut als zeitkonstant oder zeitabhängig definiert wird, wird es entweder in die eine oder andere Stammdatentabelle aufgenommen.

Ebenso wie bei der Tabelle für zeitkonstante Merkmale werden die Attribute in Klartext abgelegt, allerdings unter Bezug auf einen Zeitraum. Abbildung 6-23 stellt die zeitabhängige Stammdatentabelle des InfoObjektes ZMASTRDTA dar, in der das InfoObjekt OPLANT enthalten ist.

Die beiden Stammdatentabellen für zeitabhängige und zeitkonstante Attribute werden von BW nicht direkt, sondern über einen View⁷ gelesen. Dieser View wird durch das BW automatisch angelegt,

um den Lesezugriff programmtechnisch einfacher zu gestalten (auch für den Zugriff aus eigenen ABAP-Programmen).

	Standard-InfoObjekt	eigendefiniertes InfoObjekt
Name des InfoObjektes	Otttttttttt	{A-Z}tttttttt
View Stammdatentabelle	/BIO/Mtttttttttt	/BIC/M{A-Z}tttttttt

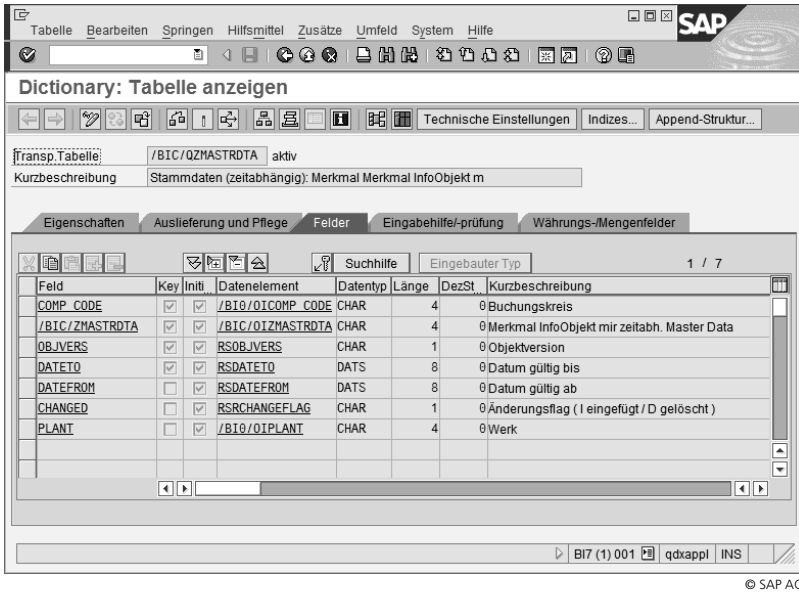


Abb. 6-23

Struktur zeitabhängiger
Master-Data-Tabellen

Normale Anzeigeattribute können nicht zur Navigation oder als Filter in Queries verwendet werden. Zu diesem Zweck können Attribute als Navigationsattribut definiert werden.

Navigationsattribute

Da die Navigation über die Stammdatentabellen, in denen Attribute im Klartext abgelegt sind, nicht performant genug sein würde, werden diejenigen Attribute, die als Navigationsattribute definiert sind, zusätzlich in weiteren Stammdatentabellen angelegt. In diesen werden die Attributinhalt in Form ihrer SID abgelegt.

Sowohl zeitkonstante als auch zeitabhängige Attribute können als Navigationsattribut definiert werden. Dementsprechend existieren zwei Stammdatentabellen mit SID-Werten.

7. Ein View ist eine virtuelle Tabelle, deren Daten aus der relationalen Verknüpfung einer oder mehrerer transparenter Tabellen bezogen werden.

	Standard-InfoObjekt	eigendefiniertes InfoObjekt
Name des InfoObjektes	0ttttttttt	{A-Z}ttttttt
Stammdaten-SID (zeitkonstante Navigationsattribute)	/BIO/Xttttttttt	/BIC/X{A-Z}ttttttt
Stammdaten-SID (zeitabhängige Navigationsattribute)	/BIO/Yttttttttt	/BIC/Y{A-Z}ttttttt



Aufgrund der bisherigen Erläuterungen haben Navigationsattribute keine Nachteile gegenüber Anzeigeattributen. Dies ist aus Sicht des Datenmodells auch zutreffend, allerdings werden im Rahmen der Ladevorgänge bei Master Data Prüfroutinen für die SID-Einträge von Navigationsattributen durchlaufen, die bei Anzeigeattributen nicht notwendig sind. Navigationsattribute können sich daher negativ auf die Performance des Staging auswirken.

6.2.3 Explizite Anzeigeattribute

Ein Sonderfall bei der Definition von InfoObjekten ist die Option, InfoObjekte nur als Attribut zu definieren. Derartige InfoObjekte können nicht in Dimensionstabellen aufgenommen werden und nicht als Navigationsattribute zu anderen InfoObjekten definiert werden, sondern ausschließlich als Anzeigeattribut anderer InfoObjekte genutzt werden.

Explizite Anzeigeattribute eines InfoObjekts werden nur in der Tabelle für zeitkonstante bzw. zeitabhängige Attribute, nicht jedoch in den Tabellen für Navigationsattribute abgelegt. Beim Laden der Stammdaten eines InfoObjekts müssen demnach zwar die SID-Werte für dessen Navigationsattribute, aber nicht für dessen explizite Anzeigeattribute ermittelt werden, was sich positiv auf die Performance der Ladevorgänge auswirkt.

6.2.4 Referenzierende Merkmale

Beim Anlegen eines Merkmals besteht die Möglichkeit, dieses Merkmal mit einer Referenz auf ein anderes Merkmal anzulegen. In diesem Fall gelten für das referenzierende Merkmal alle technischen Einstellungen des Referenzmerkmals. Dies betrifft Attribute, Stammdaten, Texte, Hierarchien, Datentyp, Länge, Klammerungen, Kleinbuchstaben und Konvertierungsroutinen.

Dabei ist zu beachten, dass die technischen Einstellungen nicht in ein neues Merkmal kopiert werden⁸, sondern das InfoObjekt physisch

8. Dies wäre der Fall, wenn das Merkmal mit der Vorlage eines anderen Merkmals angelegt würde. In diesem Fall können die Einstellungen noch nachträglich geändert werden.

dasselbe Datenelement, dieselben SID- und weitere Tabellen nutzt wie das referenzierte InfoObjekt. Dies macht sich zum Beispiel darin bemerkbar, dass nach dem Laden der Stammdaten für ein Referenzmerkmal automatisch für das referenzierende Merkmal dieselben Stammdaten zur Verfügung stehen.

Vom Referenzmerkmal unabhängig ist das referenzierende Merkmal lediglich in Bezug auf die betriebswirtschaftliche Semantik. Dies betrifft die Beschreibung, Darstellung, Auswahl von Textlängen, Berechtigungsrelevanz, Konstantenzuweisung im Staging und die Bezeichnung von Attributen.

Ein typisches Beispiel für den Einsatz eines Referenzmerkmals sind Partnerrollen. Eine Partnerrolle besagt bspw., dass in einem Kundenauftrag nicht nur der Auftraggeber, sondern auch der Regulierer (Rechnungsempfänger) und der Warenempfänger erfasst werden. Diese Partnerrollen können identisch, aber auch unterschiedlich ausgeprägt sein, stellen aber in jedem Fall eine Kundennummer dar.

Partnerrollen

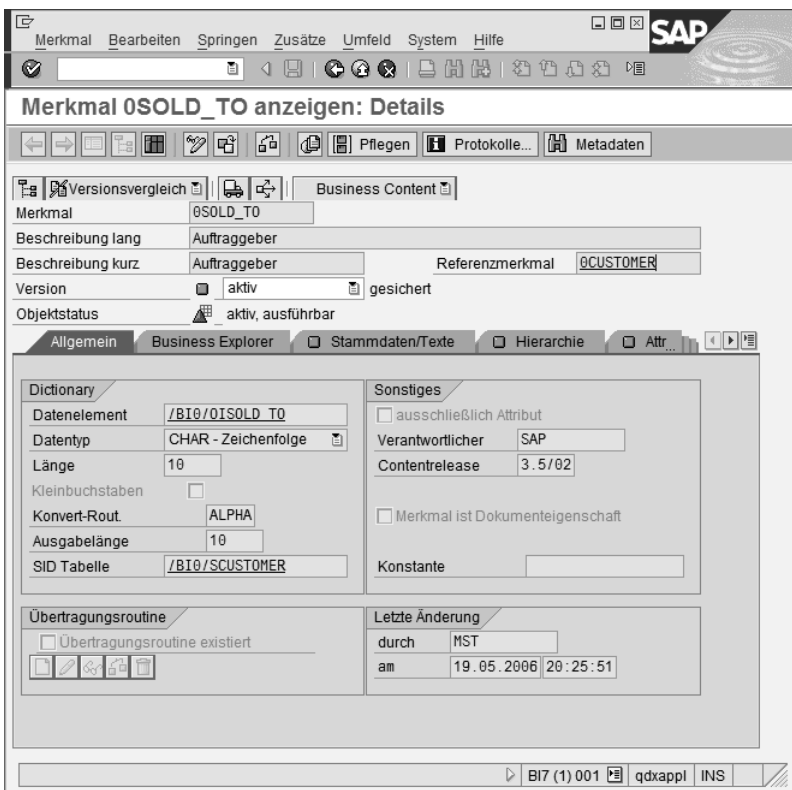


Abb. 6-24
Referenzierung von
InfoObjekten

Ohne Referenzmerkmale müssten drei InfoObjekte (Auftraggeber, Regulierer, Warenempfänger) angelegt und jeweils mit denselben Stammdaten versorgt werden. Dies ist aber insofern nicht sinnvoll, als allen drei Partnerrollen derselbe Kundenstamm zugrunde liegt.

Aus diesem Grund wird zum Beispiel im BI Content ein InfoObjekt 0CUSTOMER ausgeliefert, das die Funktion eines Referenzmerkmals hat. Stammdaten, Texte, Hierarchien werden ausschließlich für dieses InfoObjekt ins BW geladen. Daneben existieren weitere InfoObjekte für die Partnerrollen, die auf 0CUSTOMER referenzieren und damit dessen Werte, Attribute und Texte verwenden. Abb. 6–24 auf Seite 93 stellt die Referenzierung des InfoObjektes 0CUSTOMER durch das Content-InfoObjekt 0SOLD_T0 dar.

6.2.5 Externe Hierarchien

Eine Hierarchie ist eine Gliederung und Zusammenfassung von Merkmalswerten eines Merkmals nach bestimmten Kriterien. Sie werden immer in Bezug auf genau ein InfoObjekt definiert.

Jede Hierarchie besteht aus genau einem obersten Hierarchieknoten (der *Hierarchiewurzel*) und weiteren darunter liegenden Hierarchieknoten, welche die Verzweigungen der Hierarchie definieren. Innerhalb der Hierarchie sind die unterschiedlichen Ausprägungen des Merkmals enthalten, so dass im Reporting die Merkmalswerte durch die Struktur, welche die Hierarchie vorgibt, gruppiert werden können.

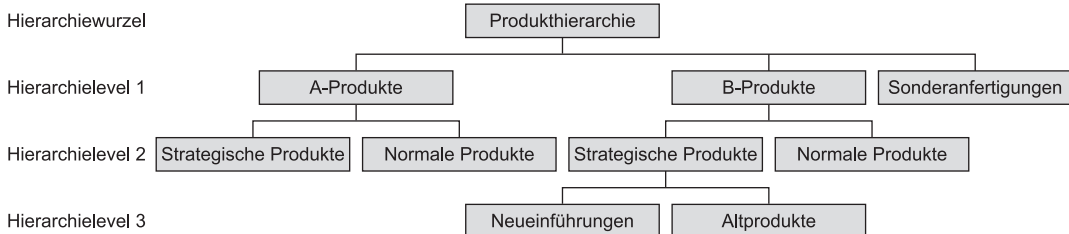


Externe Hierarchien bieten hervorragende Möglichkeiten, um Hierarchien flexibel und in kurzer Zeit auch manuell zu definieren. In Bezug auf die Performance bringt der Einsatz externer Hierarchien aber auch Nachteile mit sich. Es gibt noch weitere Möglichkeiten, um hierarchische Strukturen abzubilden. Lesen Sie dazu das Kapitel 8.3 – »Modellierung von Hierarchien«, S. 194.

Abb. 6–25
Aufbau einer externen Hierarchie

Je nach Entfernung der Hierarchieknoten von der Hierarchiewurzel wird von Hierarchieleveln gesprochen. In Abbildung 6–25 ist beispielhaft eine mögliche Hierarchie zum InfoObjekt 0MATERIAL dargestellt.

Produktthierarchie aus Sicht des Produktmanagers



Bei den Hierarchieknoten einer Hierarchie werden zwei Knotentypen unterschieden:

- **Nicht bebuchbare Knoten:** Diese geben die Struktur der Hierarchie vor und stellen damit eine Art Hülse für die Merkmalswerte dar.
- **Bebuchbare Knoten:** Diese sind die Merkmalsausprägungen des InfoObjektes, für das die externe Hierarchie angelegt ist.

Bei der Datenanalyse werden die Kennzahlwerte aller Merkmale, die unter einem Hierarchieknoten eingeordnet sind, auf diesen Knoten aggregiert. Dabei ist es möglich, unterhalb von nicht bebuchbaren Knoten bebuchbare Knoten anzulegen und umgekehrt.

Die nicht bebuchbaren Knoten der Hierarchie können willkürlich aufgebaut werden und können in ihrer Beschreibung ein- und mehrdeutig und in ihrer Verästelung unterschiedlich tief gestaltet sein (unbalanciert). Die Struktur einer externen Hierarchie kann damit vollständig an fachlichen Vorgaben ausgerichtet werden, ohne technischen Einschränkungen der Struktur unterworfen zu sein.

Nicht bebuchbare Knoten

Die nicht bebuchbaren Knoten unterteilen sich in zwei Arten:

- Textknoten
- fremde Merkmalsknoten

Textknoten sind die einfachste Form eines nicht bebuchbaren Hierarchieknotens. Ein Textknoten wird ausschließlich durch einen beliebigen Knotennamen und eine beliebige Beschreibung definiert.

Textknoten

Die Werte für den Knotennamen beziehungsweise seine Bezeichnung hängen von keinen anderen InfoObjekten ab und können vollständig frei gebildet und verschachtelt werden. Abbildung 6–26 stellt ein Beispiel für die Definition einer Hierarchiestruktur mit Hilfe von Textknoten dar.

Im Normalfall werden Textknoten zur Beschreibung der Hierarchiestruktur verwendet. Es besteht jedoch auch die Möglichkeit, statt statischer Bezeichnungen die Textinhalte anderer InfoObjekte in den Knoten zu hinterlegen. Diese werden bei der Verwendung der Hierarchie aus den Texttabellen der InfoObjekte nachgelesen und sind somit immer aktuell, auch wenn sich Bezeichnungen ändern.

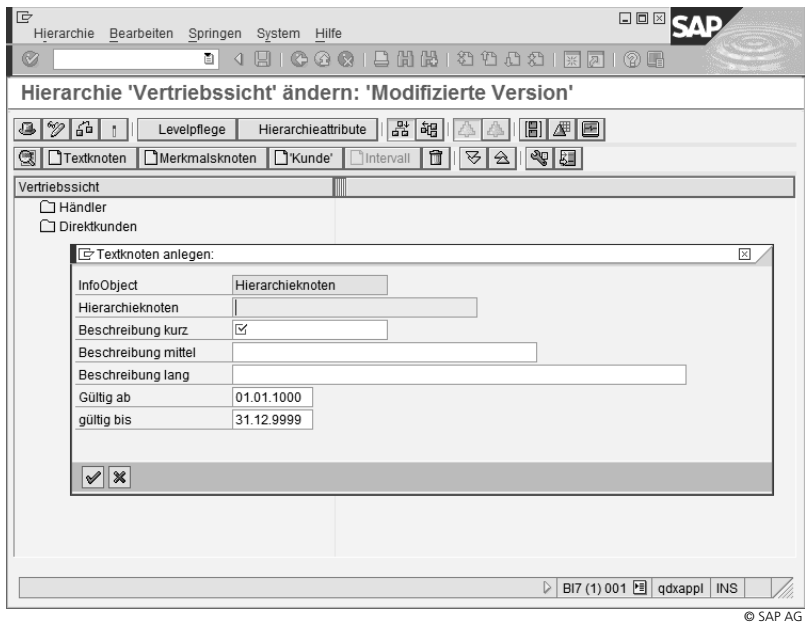
Fremde Merkmalsknoten

In der Hierarchiepflege werden zu diesem Zweck sogenannte *Fremde Merkmalsknoten* eingesetzt. Bei der Nutzung dieses Knotentyps muss zu jedem Hierarchieknoten das gewünschte InfoObjekt und ein Merkmalswert angegeben werden.

Bebuchbare Knoten stellen die Einordnung von Merkmalswerten eines InfoObjektes in die Struktur seiner Hierarchie dar. Die Einordnung der Merkmalswerte in die Hierarchiestruktur bestimmt damit, welche Merkmalsausprägungen die Summe eines Hierarchieknotens bilden.

Bebuchbare Knoten

Abb. 6-26
Textknoten einer
Hierarchie



Bebuchbare Knoten können das unterste Level in einem Zweig der Hierarchie bilden. In diesem Fall spricht man von einem *Hierarchieblatt*. Es ist aber auch möglich, dass unterhalb von bebuchbaren Knoten weitere Knoten (bebuchbare Knoten, Textknoten, fremde Merkmalsknoten) angelegt werden.

Hierarchieintervalle

Eine Sonderform der bebuchbaren Knoten sind Hierarchieintervalle. Während bei bebuchbaren Knoten ein exakter Merkmalswert angegeben werden muss, der diesen Knoten definiert, kann bei einem Hierarchieintervall ein Nummernkreis von Merkmalswerten definiert werden.

Dies ist insbesondere dann sinnvoll, wenn Merkmalswerte durch Nummernkreise bestimmt werden, die mit den Hierarchieknoten einhergehen. Unter dieser Voraussetzung ist es sogar möglich, Vorkehrungen für die Einordnung von noch nicht angelegten Stammdaten zu treffen. Damit muss bei einem Neuzugang von Stammdaten keine Anpassung der Hierarchie erfolgen.

Versionsabhängigkeit

Bei Bedarf kann eine Hierarchie in mehreren Versionen gleichzeitig vorhanden sein. Die Pflege der Hierarchiestruktur wird dann immer für die jeweils aktive Version vorgenommen. Auch bei der Datenanalyse steht nur die jeweils aktive Version der Hierarchie zur Verfügung.

Zur Speicherung von Versionen werden keine versionsabhängigen Tabellen oder Ähnliches generiert. Alle Versionen werden zusammen in denselben transparenten Tabellen abgelegt, so dass der Einsatz mehrerer Versionen insbesondere bei großen Hierarchien zu Performanceproblemen führen kann.



Der Einsatz versionsabhängiger Hierarchien kann sinnvoll sein, wenn planungs- oder simulationsähnliche⁹ Aufgaben bei der Datenanalyse wahrgenommen werden.

Aus der fachlichen Notwendigkeit der Zeitabhängigkeit von Stammdaten leitet sich die Zeitabhängigkeit auch für Hierarchien ab. Die Zeitabhängigkeit von Hierarchien ist vergleichbar mit der Zeitabhängigkeit von Master Data und weist damit einen wesentlichen Unterschied zur Versionsabhängigkeit auf: Alle Stichtage sind im Reporting gleichzeitig verfügbar. Es bedarf dazu keiner Aktivierung von Stichtagen, wie dies bei Versionen der Fall ist.

Zeitabhängigkeit

Bei der Zeitabhängigkeit kann zwischen der Zeitabhängigkeit der Gesamthierarchie und der Zeitabhängigkeit der Hierarchiestruktur gewählt werden.

Wird die Gesamthierarchie als zeitabhängig gekennzeichnet, so gilt diese Abhängigkeit für die Hierarchiewurzel und wird auf alle darunter liegenden Hierarchieknoten übertragen. Je nach Auswahl des Stichtags können dabei vollständig unterschiedliche Hierarchien genutzt werden.

Zeitabhängigkeit der Gesamthierarchie

Bei der zeitabhängigen Hierarchiestruktur wird pro Knoten (bebuchbare und nicht bebuchbare Knoten) festgelegt, für welchen Zeitraum er an der angegebenen Stelle der Hierarchie stehen soll. Dieser Typ der Zeitabhängigkeit eignet sich vor allem dann, wenn die Hierarchie grundsätzlich stabil besteht und im Zeitverlauf nur einzelne Änderungen in der Hierarchie vorgenommen werden.

Zeitabhängigkeit der Hierarchiestruktur

Externe Hierarchien haben das Präfix »extern« im Namen deshalb erhalten, weil sie losgelöst von den Master Data eines InfoObjektes sind, das heißt, sie können quasi außerhalb der Stammdaten des InfoObjektes gepflegt werden. Dennoch beziehen sich externe Hierarchien auf genau ein InfoObjekt.

Externe Hierarchie im Datenmodell

Jedes InfoObjekt mit Master Data kann mehrere externe Hierarchien besitzen. Die Eigenschaften aller externen Hierarchien eines

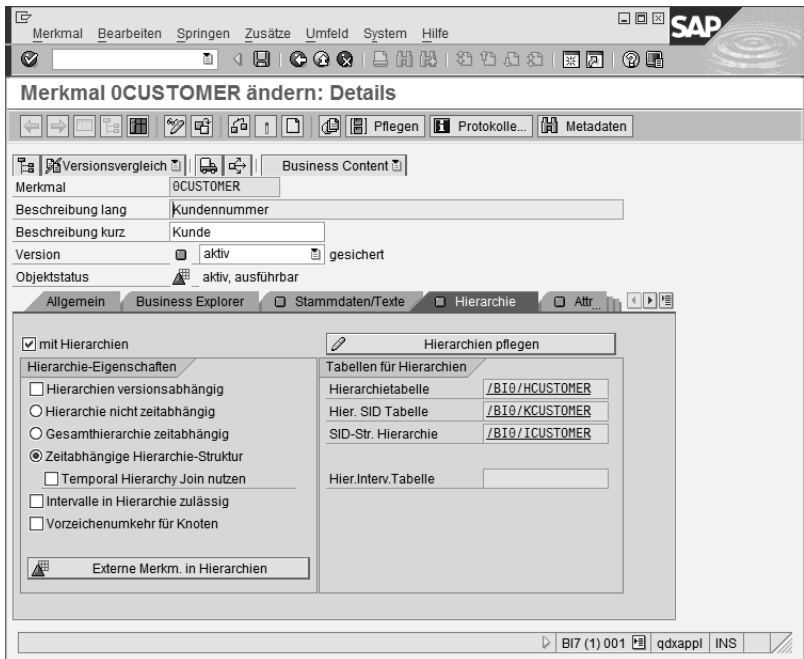
9. Im BW existieren keine Funktionen, die explizit zum Zweck der Planung oder Simulation entwickelt wurden. Die bestehenden Funktionen können in begrenztem Umfang für solche Aufgaben genutzt werden, ersetzen jedoch nicht Produkte wie SEM, SCM oder APO, welche explizite Planungs- und Simulationsaufgaben wahrnehmen sollen.

InfoObjektes werden in der InfoObjekt-Pflege festgelegt. Bei diesen Eigenschaften handelt es sich um die Festlegung,

- ob ein InfoObjekt überhaupt externe Hierarchien besitzen kann.
- ob die Hierarchien versionsabhängig sind.
- ob die Gesamthierarchien oder die Hierarchiestrukturen zeitabhängig sind.
- ob Intervalle in der Hierarchie zugelassen sein sollen.

Abbildung 6–27 stellt die Festlegung dieser Hierarchieeigenschaften in der InfoObjekt-Pflege dar.

Abb. 6–27
Hierarchieeigenschaften
von InfoObjekten



© SAP AG

Die Tatsache, dass die Hierarchieeigenschaften in der InfoObjekt-Pflege festgelegt werden, hat zur Folge, dass die Eigenschaften aller externen Hierarchien zu einem InfoObjekt identisch sind. Es ist zum Beispiel nicht möglich, zu einem InfoObjekt eine externe Hierarchie zu erstellen, deren *Gesamtstruktur* zeitabhängig ist, und eine, deren *Hierarchiestruktur* zeitabhängig ist.

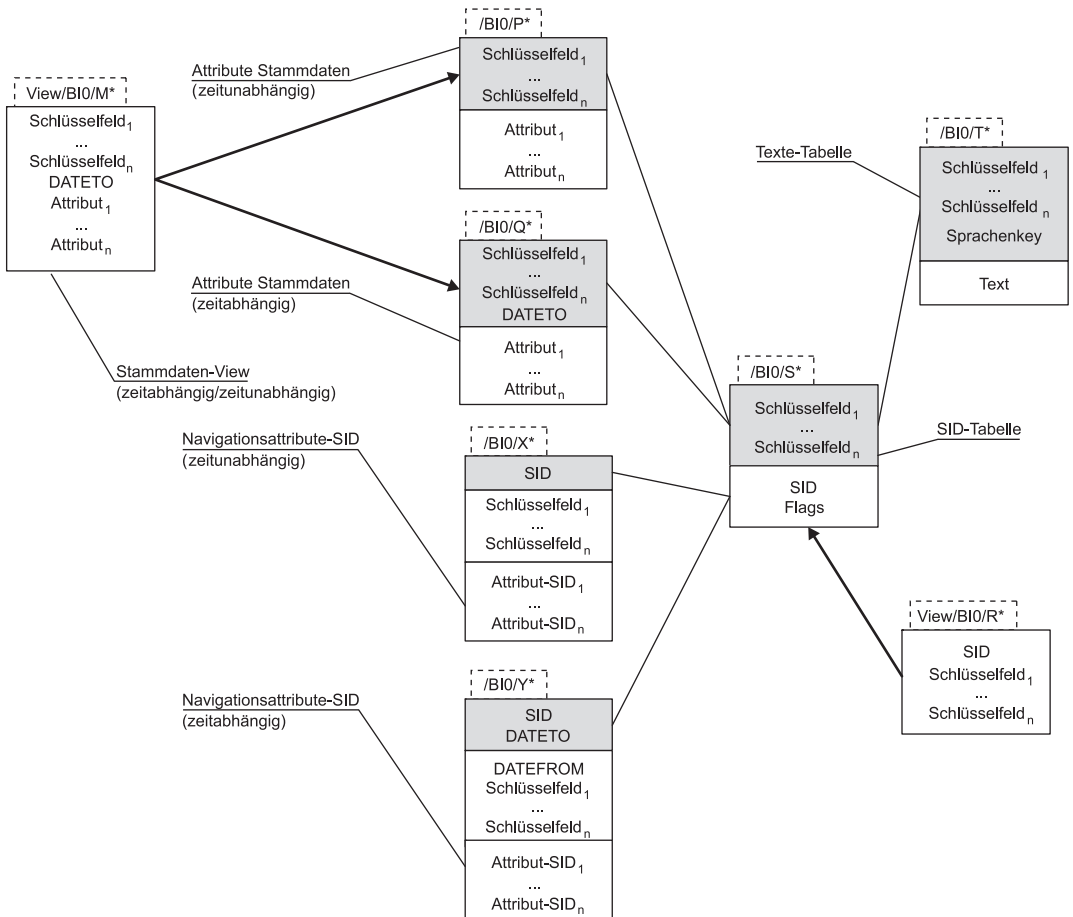


Externe Hierarchien können nur für Basismerkmale, nicht aber für referenzierende Merkmale angelegt werden. Da referenzierende Merkmale dieselben Stammdatentabellen wie ihre Referenzmerkmale nutzen, können externe Hierarchien auch bei allen Referenzmerkmalen verwendet werden. Beim Aufbau einer Kundenhierarchie muss die Hierarchie zum Beispiel nur zum Kunden (OCUSTOMER) angelegt werden. Dieselbe Hierarchie steht damit auch bei den referenzierenden Merkmalen (zum Beispiel Auftraggeber, Regulierer, Lieferant) zur Verfügung.

6.2.6 Zusammenfassung

Das Datenmodell der Master Data ist sehr komplex. Dennoch ist das Verständnis dieses Modells unabdingbar für die Modellierung weiterer Datenziele. Aus diesem Grund werden die Zusammenhänge in Abbildung 6–28 noch einmal zusammengefasst.

Abb. 6–28
 Tabellenstruktur
 von Master Data



Namenskonvention

Die nachfolgende Übersicht fasst die Namenskonventionen der Tabellen, die zur Speicherung der Master Data verwendet werden, zusammen:

	Standard-InfoObjekt	eigendefiniertes InfoObjekt
Name des InfoObjektes	0ttttttttt	{A-Z}tttttttt
Texttabelle	/BIO/Tttttttttt	/BIC/T{A-Z}tttttttt
Stammdatentabelle (zeitkonstant)	/BIO/Pttttttttt	/BIC/P{A-Z}tttttttt
Stammdaten-SID (zeitkonstante Navigationsattribute)	/BIO/Xttttttttt	/BIC/X{A-Z}tttttttt
Stammdatentabelle (zeitabhängig)	/BIO/Qttttttttt	/BIC/Q{A-Z}tttttttt
Stammdaten-SID (zeitabhängige Navigationsattribute)	/BIO/Yttttttttt	/BIC/Y{A-Z}tttttttt
View Stammdatentabelle	/BIO/Mttttttttt	/BIC/M{A-Z}tttttttt
Hierarchietabelle	/BIO/Httttttttt	/BIC/H{A-Z}tttttttt
Umschlüsselung Hierarchiewerte auf SID-Werte	/BIO/Kttttttttt	/BIC/K{A-Z}tttttttt
Struktur der Hierarchieknoten	/BIO/Ittttttttt	/BIC/I{A-Z}tttttttt
Hierarchieintervalle	/BIO/Jttttttttt	/BIC/J{A-Z}tttttttt

Die aufgeführten Tabellen werden nur dann vollständig für ein InfoObjekt angelegt, wenn alle Optionen der Master Data genutzt werden (Stammdaten und Texte vorhanden, Navigationsattribute in zeitkonstanter und zeitabhängiger Form vorhanden). Abhängig von den Einstellungen des InfoObjektes werden einzelne Tabellen nicht benötigt und dementsprechend nicht im ABAP Dictionary angelegt.

6.3 DataStore-Objekte

Die bisherigen Ausführungen zum Datenmodell bezogen sich auf den Bereich der Datendefinitionen (InfoObjekte für Merkmale und Kennzahlen) sowie der Master Data, d.h. der Stammdaten-tragenden Datenbereiche eines Datenmodells.

Analog zu den flachen Strukturen, die in Kapitel beschrieben wurden, fassen DataStore-Objekte Merkmale und Kennzahlen in Tabellenform zusammen.

DataStore-Objekte können damit zur Aufnahme von Bewegungsdaten genutzt werden, was für die Vielzahl aller Fälle auch der prädes-

tinierte Einsatzbereich von DataStore-Objekten ist. Grundsätzlich können DataStore-Objekte aufgrund ihrer einfachen Struktur jedoch jegliche andere Informationen wie z.B. Steuerungsdaten, Protokolle u.v.m. enthalten.

Das BW bietet drei unterschiedliche Typen von DataStore-Objekten:

- DataStore-Objekte für direktes Schreiben
- DataStore-Objekte ohne Delta-Bildung (schreiboptimierte DataStore-Objekte)
- DataStore-Objekte mit Delta-Bildung (Standard-DataStore-Objekte)

Die Standard-DataStore-Objekte sind mit sehr umfangreichen Funktionen für das Staging verbunden, die erst im Rahmen von Kapitel 17 erläutert werden.

DataStore-Objekte können für die Datenanalyse genutzt werden, sie sind jedoch unter Berücksichtigung der Performance keinesfalls dafür geeignet. Nutzen Sie DataStore-Objekte zu allen nur erdenklichen Zwecken, doch vermeiden Sie die Verwendung von DataStore-Objekten zur Datenanalyse.



In jedem Fall stellen DataStore-Objekte eine Tabellenstruktur dar, die entsprechende Tabellendefinitionen im ABAP Dictionary bilden (vgl. Kapitel 4.2.3).

Namenskonvention

Die nachfolgende Tabelle stellt dar, nach welcher Namenskonvention die transparente Tabelle zur Aufnahme von Daten jeweils im ABAP Dictionary angelegt wird.

	Conten DataStore-Objekt	eigendefiniertes DataStore-Objekt
DataStore-Objekt	0tttttt	{A-Z}ttttt
Tabelle im ABAP Dictionary	/BIO/Attttt00	/BIC/A{A-Z}ttttt00

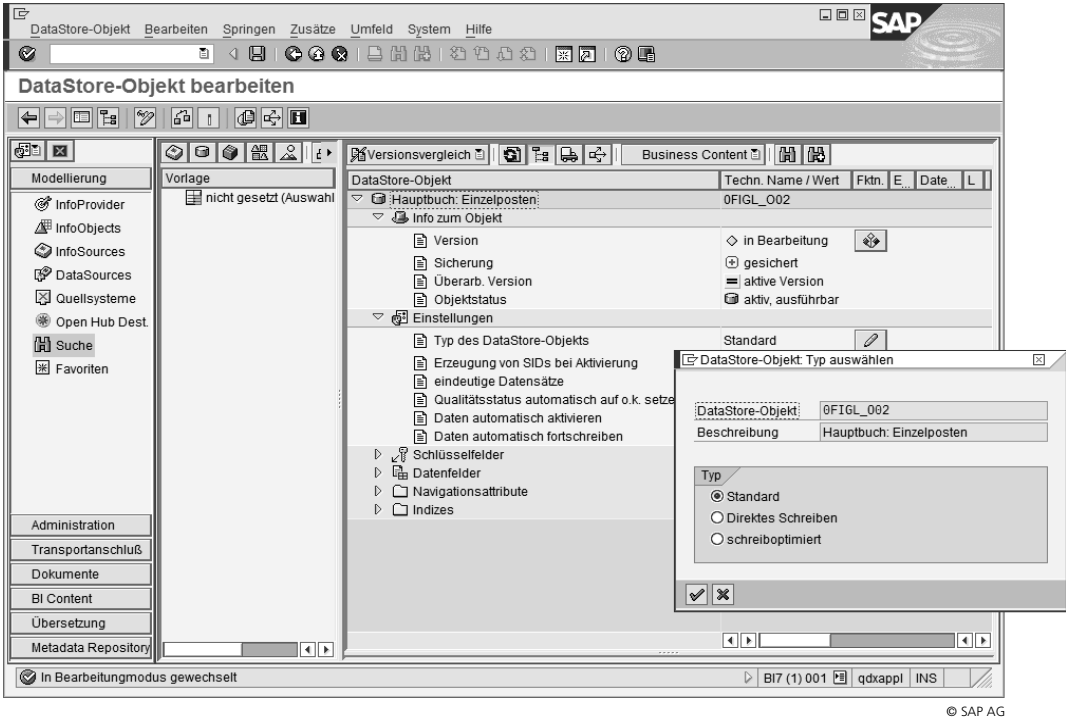
Die Darstellung dieser Namenskonvention wird in Kapitel 17.1.2 um weitere Tabellen ergänzt, die bei Standard-DataStore-Objekten zur Ermittlung von Deltainformationen benötigt werden.

Welcher Art ein DataStore-Objekt sein soll, wird bei der Pflege des DataStore-Objekts in den Einstellungen festgelegt (siehe Abb. 6–29).

Festlegen des DataStore-Objekt-Typen

Die einfachste Form eines DataStore-Objektes ist das *DataStore-Objekt für direktes Schreiben* (im BW 3.x als transaktionales ODS-Objekt bezeichnet). DataStore-Objekte für direktes Schreiben sind nicht mit Funktionen für das Staging versehen, können nicht für das

DataStore-Objekt für direktes Schreiben



© SAP AG

Abb. 6–29
 Festlegen des DataStore-
 Objekt-Typen

Reporting¹⁰ genutzt werden und sind in erster Line als bessere Form von transparenten Tabellen (vgl. Kapitel 4.2) zu betrachten, die zwar Bestandteil der Metadaten des BW sind, denen aber keine tiefgreifenden Aufgaben im BW zukommen.

DataStore-Objekte für direktes Schreiben werden stattdessen als Datenziel für Analyseprozesse des Analysis Process Designer und der Data Mining Workbench (Transaktionen RSANWB bzw. RSDMWB) eingesetzt und können auch in eigenen Programmen genutzt werden. Auf diese Weise dienen DataStore-Objekte für direktes Schreiben als Datenbehälter für unterschiedlichste Daten, die auf unterschiedliche Weise generiert und auf ebenso unterschiedliche Weise weiterverarbeitet werden können.

Die Inhalte von DataStore-Objekten für direktes Schreiben (und nur von diesen!) können durch selbstentwickelte Programme in beliebiger Weise erzeugt, verändert und gelesen werden. Eleganter ist der Zugriff mit Hilfe der folgenden RFC-fähigen Funktionsbausteine, die auch über Systemgrenzen hinweg aufgerufen werden können und eine Sperrverwaltung mit sich bringen:

10. Transaktionale ODS-Objekte können über einen Umweg für Auswertungen genutzt werden, indem sie mit Hilfe eines InfoSets ausgewertet werden (siehe Kapitel 11.2).

```
RSDRI_ODSO_INSERT_RFC
RSDRI_ODSO_UPDATE_RFC
RSDRI_ODSO_MODIFY_RFC
RSDRI_ODSO_DELETE_RFC
```

Das nachfolgende Beispiel schreibt Datensätze in ein DataStore-Objekt für direktes Schreiben (Q_DSTORE), das durch die InfoObjekte OCUSTOMER (Kundennummer) und OMATERIAL (Materialnummer) im Schlüssel und durch die InfoObjekte OSALES (Verkaufswert) und OCCURRENCY (Währung) in den Datenfeldern definiert ist.

```
REPORT ZQX_DATASTORE_BEISPIEL.
DATA:
  l_numrows LIKE bapi6116xx-numrows,
  l_s_data LIKE bapi6116da,           "Struktur der Tabelle
                                     l_t_data
  l_t_data TYPE TABLE OF bapi6116da. "Tabelle

DATA: BEGIN OF l_wa_data.
INCLUDE STRUCTURE /bic/aQ_DSTORE.
DATA: buff(1000).
DATA: END OF l_wa_data.
  l_wa_data-customer = '4711'.       "Kundennummer 4711
  l_wa_data-material = ,0815'.      "Materialnummer 0815
  l_wa_data-sales = 100.             "Verkaufswert 100
  l_wa_data-currency = 'EUR'.       "Euro

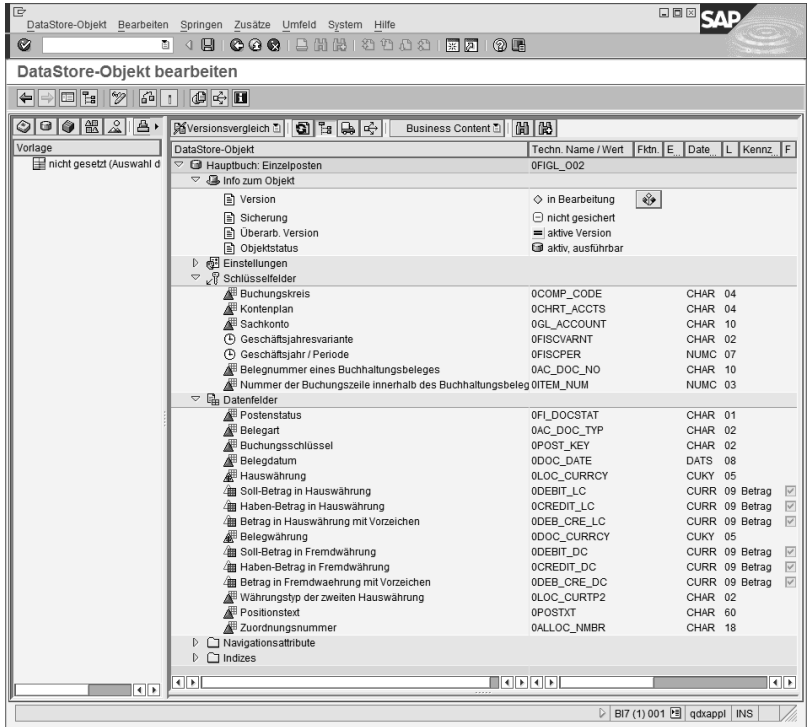
  l_s_data-data = l_wa_data.
APPEND l_s_data TO l_t_data.        "In interne Tabelle einfügen

CALL FUNCTION 'RSDRI_ODSO_INSERT_RFC'
EXPORTING
  i_odsobject = 'Q_DSTORE'
IMPORTING
  e_numrows = l_lumrows             "Eingefügte Sätze
TABLES
  i_t_data = l_t_data "Interne Tabelle mit Daten
EXCEPTIONS
  data_target_not_ods = 1
  ods_type_not_transactional = 2
  active_table_name_not_found = 3
  record_key_already_exists = 4
  array_insert_failed = 5
  internal_error = 6
OTHERS = 7.

IF sy_subrc NE 0.
  WRITE:/ ,Einfügen oder Ändern fehlgeschlagen'.
ELSE.
  WRITE:/ ,Erfolgreich eingefügte Sätze: , , l_numrows.
ENDIF.
```

Das Datenmodell jedes DataStore-Objekts definiert sich durch *Schlüsselfelder* und *Datenfelder*. In beiden Fällen handelt es sich bei den Feldern um InfoObjekte (siehe Abb. 6–30).

Abb. 6–30
Modellierung von
DataStore-Objekten



© SAP AG

6.3.1 Schlüsselfelder

Die Daten in DataStore-Objekten leiten sich nicht primär aus den Anforderungen für die Analyse ab, sondern aus ihrer Beziehung zu den operativen Datenstrukturen. Danach muss es möglich sein, auch Transaktionsveränderungen richtig verarbeiten zu können. Voraussetzung dafür ist die Fähigkeit, bestehende Daten überschreiben zu können (zum Beispiel muss bei einer Änderung des Auftragsstatus der letzte Status durch den aktuellen ersetzt werden).

Dies ist nur dann möglich, wenn diejenigen Felder, die eine Transaktion bestimmen (zum Beispiel Auftragsnummer und Auftragsposition), als solche definiert sind, damit alle anderen Felder (zum Beispiel Auftragsstatus) in Relation dazu gesetzt werden können.

Um diese Abhängigkeit zu erreichen, werden diejenigen Felder, die eine Transaktion bestimmen, als Schlüsselfelder in DataStore-Objekten definiert. Jeder Datensatz eines DataStore-Objekts ist damit ein-

deutig durch die Kombination der Schlüsselfelder zu identifizieren. In der zugrunde liegenden transparenten Tabelle bilden die Schlüsselfelder den Primärschlüssel der Tabelle.

Die Definition der Schlüsselfelder eines DataStore-Objekts setzt genaue Kenntnisse der im Quellsystem vorliegenden Strukturen voraus. Es ist dringend davon abzuraten, Schlüsselfelder zu vermuten oder auszuprobieren, da Fehler bei der Definition zu schwer nachvollziehbaren Fehlern führen können. Definieren Sie die Schlüsselfelder eines DataStore-Objekts erst, wenn Sie sich sicher sind, dabei keinen Fehler zu begehen.



Bei der Definition der Schlüsselfelder sind folgende Einschränkungen zu beachten:

- Es müssen zwischen mindestens einem und maximal 16 Schlüsselfelder definiert werden.
- Die Länge der Schlüsselfelder darf in Summe 780 Byte nicht übersteigen.
- Es können keine Kennzahlen als Schlüsselfeld verwendet werden.

6.3.2 Datenfelder

Alle Felder, die aus operativer Sicht abhängig von den Schlüsselfeldern sind (zum Beispiel Statusfelder von Aufträgen), werden bei DataStore-Objekten als Datenfelder bezeichnet. Dabei handelt es sich im Falle von Bewegungsdaten um alle Felder, die nicht Teil des Primärschlüssels (Schlüsselfelder) sind – also sowohl Merkmale als auch Kennzahlen.

Die Verwendung von Datenfeldern ist an folgende Rahmenbedingungen gebunden:

- Es können maximal 749 Datenfelder definiert werden.
- Die maximale Datensatzlänge der DataStore-Tabelle (inkl. Schlüsselfelder) darf 1918 Byte nicht überschreiten.

6.4 BasisCubes

BasisCubes bilden die Grundlage der analysefähigen Datenmodelle im BW. Zwar können Daten in unterschiedlichen BW-Objekten abgelegt werden, doch ist es den BasisCubes vorbehalten, Merkmale und Kennzahlen performant und mit der vollen Funktionsvielfalt des BW für Analysezwecke bereitzustellen.

BasisCubes sind grundsätzlich nach dem Snowflake-Schema aufgebaut, das jedoch BW-spezifisch angepasst und erweitert wurde, um

zum Beispiel nichtbalancierte Hierarchien oder die stichtagsbezogene Historisierung von Attributen zu ermöglichen. Die SAP bezeichnet das Datenmodell der BasisCubes daher als *Developed Star-Schema* (Erweitertes Star-Schema).

Die Erweiterung des normalen Star-Schemas bietet umfangreiche Möglichkeiten bei der Datenmodellierung, erhöht jedoch gleichzeitig den Komplexitätsgrad des Datenmodells, so dass die Modellierung im BW nicht ohne tiefere Kenntnisse des Datenmodells vorgenommen werden sollte.

Nachfolgend werden die Rolle der *Faktentabelle*, der *Stammdaten-IDs* und der *Dimensionen* von BasisCubes erläutert. Anschließend wird die spezielle Form der *Realtime InfoCubes* erläutert.

6.4.1 Faktentabelle

Während das Konzept des normalen Star-Schemas auf nur einer zentralen Faktentabelle basiert, definiert das BW beim Anlegen eines BasisCubes automatisch *zwei* Faktentabellen:

- eine »normale« Faktentabelle
- eine komprimierte Faktentabelle

Beide Faktentabellen sind in ihrer Struktur identisch, jedoch für unterschiedliche Aufgaben im »Lebenszyklus« analytischer Daten vorgesehen.

Die normale Faktentabelle

Die normale Faktentabelle ist die erste Station für alle neuen Daten in einem BasisCube. Sie ist bereits in Grundzügen für den analytischen Zugriff auf ihre Datenbestände vorbereitet und damit funktional vollwertig, um Datenbestände analytisch zu verwerten.

Dennoch ist die normale Faktentabelle nur als *Zwischenstation* konzipiert, die es in erster Linie ermöglichen soll, die Daten einzelner Ladevorgänge zu administrieren¹¹ (Daten eines Ladevorgangs werden nachfolgend als *Request* bezeichnet). Dies wird unter anderem erreicht, indem zu jedem Datensatz der Faktentabelle die ID des Requests (die sog. Request-ID), mit dem er in die Faktentabelle geschrieben wurde, in einer eigenen Dimension abgelegt wird. Dadurch werden Daten in der Faktentabelle mit einer Detaillierungsstufe abgelegt, die zwar nicht aus betriebswirtschaftlicher, aber aus technischer Sicht erforderlich ist.

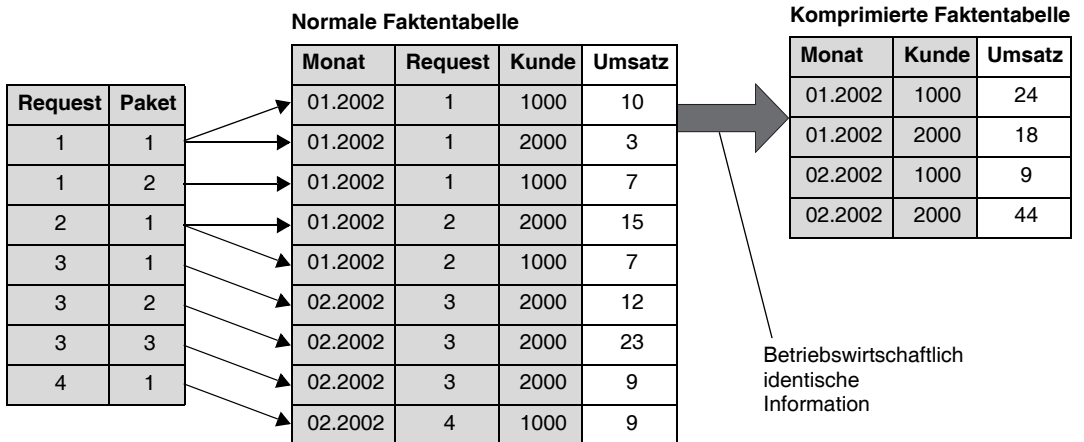
11. Teil dieser Administrationsaufgaben ist zum Beispiel die Weitergabe neuer Daten in andere Datenziele oder das nachträgliche Löschen eines Requests aus der normalen Faktentabelle.

Um Daten zusätzlich möglichst performant in die normale Faktentabelle schreiben zu können, verfügt die normale Faktentabelle nicht über einen Primärschlüssel im Sinne des ABAP Dictionary¹². Werden die Daten eines Requests in mehrere Pakete unterteilt, so werden Daten beim Füllen der Faktentabellen nicht über die Grenzen der Pakete aggregiert, so dass die Daten der BasisCubes auf Ebene der einzelnen Datenpakete detailliert sind¹³.

Je nach Datenmodell des BasisCubes, Häufigkeit der Ladevorgänge und Zusammensetzung der geladenen Daten kann sich die Detaillierung der normalen Faktentabelle gravierend auf das Datenvolumen der BasisCubes auswirken, so dass eine Datenpaket-übergreifende Aggregation der Daten, zusammen mit einem Wegfall der Request-ID, das Datenvolumen der Faktentabelle um ein Vielfaches verringern kann, ohne aus betriebswirtschaftlicher Sicht Nachteile mit sich zu bringen (siehe Abb. 6–31).

Die komprimierte Faktentabelle

Abb. 6–31
Komprimierung der Faktentabelle



Als Gegenstück zur normalen Faktentabelle existiert daher bei jedem BasisCube eine komprimierte Faktentabelle, die weniger auf die Administration und Durchführung von Ladevorgängen, sondern vielmehr auf analytische Performance optimiert ist. Dies betrifft sowohl die Speicherstruktur¹⁴ als auch die Verdichtung der Faktentabelle durch Wegfall der Request-Id.

12. In der Transaktion SE11 wird zwar ein Primärschlüssel angezeigt, in der Definition der Datenbanktabelle ist der Schlüssel jedoch nicht vorhanden.

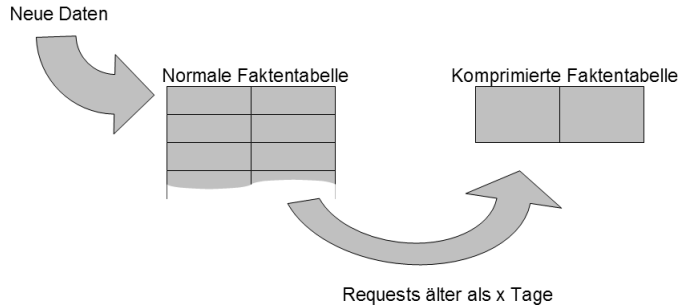
13. Die Detaillierung auf einzelne Datenpakete ist zwar nicht im Datenmodell der normalen Faktentabelle abgebildet, jedoch ist es aufgrund des fehlenden Primärschlüssels möglich, dass mehrere Sätze zu identischen Schlüsselwerten in der Faktentabelle abgelegt werden.

Zusammenspiel der
Faktentabellen

Die Nutzung der komprimierten Faktentabelle ist optional. Ohne explizite Einstellungen im BW wird die komprimierte Faktentabelle nicht verwendet. Die Übergabe von Daten aus der normalen Faktentabelle in die komprimierte Faktentabelle erfolgt im Rahmen der sogenannten Komprimierung¹⁵.

Bei der Komprimierung verschiebt das BW Daten aus der normalen Faktentabelle in die komprimierte Faktentabelle (siehe Abb. 6–32).

Abb. 6–32
Zusammenspiel normaler
und komprimierter
Faktentabellen



Aus Sicht der Datenanalyse ist es transparent, ob und wie viele Requests in der komprimierten Faktentabelle zusammengefasst sind, da die Analytical Engine beide Tabellen berücksichtigt und die Inhalte automatisch zusammenfasst.



Werden im Rahmen von Large-Scale-Architekturen Daten des BasisCube an ein anderes System weitergegeben, so wird die Request-ID für die Ermittlung der neuen Requests benötigt, die seit der letzten Datenlieferung hinzugekommen sind (Delta-Verfahren). In diesem Fall können nur diejenigen Requests in die Komprimierung eingeschlossen werden, die bereits an das andere System geliefert wurden.

Durchführen der
Komprimierung

Die Komprimierung einer Faktentabelle kann in der InfoCube-Administration durchgeführt werden (siehe Abb. 6–33). Dabei wird auf Basis der Request-ID festgelegt, welche Requests komprimiert werden sollen, d.h., es kann wahlweise die gesamte Faktentabelle oder nur ein Teil der älteren Requests komprimiert werden.



Der Vorgang der Komprimierung hat Einfluss auf den laufenden Betrieb und beeinträchtigt die Datenanalyse erheblich. Führen Sie die Komprimierung daher nicht während Analysezeiten durch.

14. Detaillierte Erläuterungen zur Speicherstruktur aus dem Gesichtspunkt des Performance-Tuning werden in den Kapitel 7.3.1 (»Range-Partitionierung«, S. 147) und Kapitel 7.3.2 (»Clustering«, S. 153) gegeben.
15. Das Komprimieren wird von der SAP stellenweise auch als Kondensieren bezeichnet.

Die Komprimierung von Daten über Request-ID und Pakete führt in der komprimierten Faktentabelle oftmals zu Datensätzen, in denen alle Kennzahlen mit dem Wert Null belegt sind. Sofern es sich nicht um einen Cube mit Bestandskennzahlen oder Kennzahlen mit Ausnahmeaggregation handelt (also nur Kennzahlen mit Aggregationsverhalten »SUM«), können diese Datensätze aus inhaltlichen Gesichtspunkten gelöscht werden.

Eliminierung von Nullwerten

Das Löschen von Nullwerten erfolgt im Anschluss an die Komprimierung automatisch, sofern die Check-Box »Mit Null-Elimination« aktiviert werden (vgl. Abb. 6–33).

Das Eliminieren von Nullwerten kann zu verwaisten Dimensionseinträgen führen, durch die die erhofften Performancevorteile der Nullwert-Eliminierung nicht in vollem Umfang erzielt werden. Führen Sie daher bei Cubes mit Nullwert-Eliminierung regelmäßige Dimensionstrimming durch (siehe Kapitel 32.2).

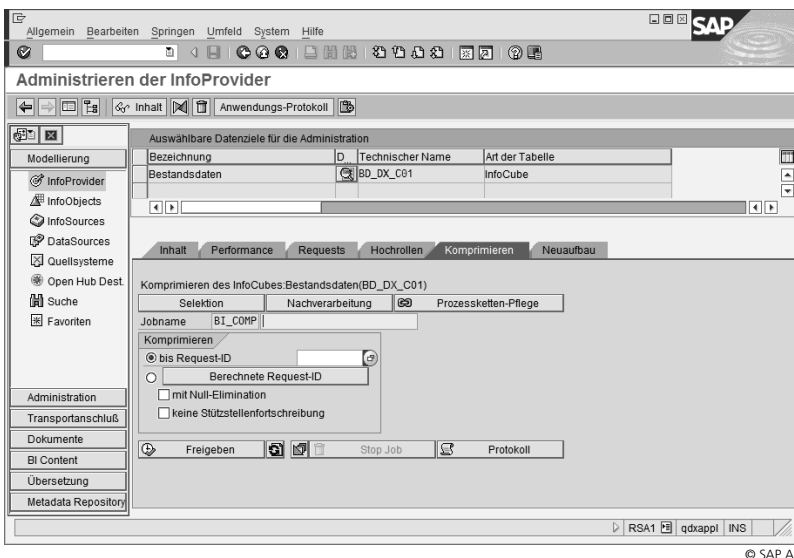


Abb. 6–33
Komprimierung von BasisCubes

Die Nullwert-Eliminierung kommt lediglich für diejenigen Nullwerte zum Tragen, die nach der Aktivierung der Nullwert-Eliminierung entstehen. Wird die Nullwert-Eliminierung nicht bereits beim Design eines Cubes aktiviert, so müssen ggf. bereits bestehende Nullwerte explizit gelöscht werden. Dieser Vorgang wird in Kapitel 9.2.3 erläutert.

Speziell bei der Komprimierung von BasisCubes mit Bestandskennzahlen ist die Verdichtung der normalen Faktentabelle in die komprimierte Faktentabelle nur die halbe Miete; denn jeder Bestand muss ausgehend von der initialen Bestandsübernahme errechnet wer-

Die komprimierte Faktentabelle bei BestandsCubes

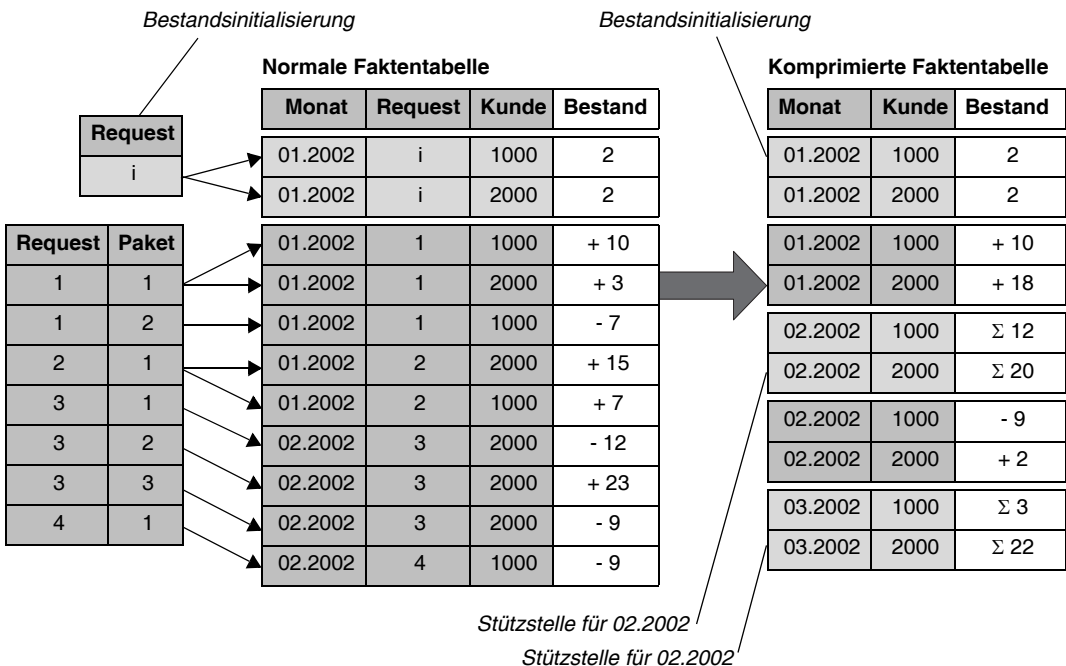
den. Soll zum Beispiel der Lagerbestand eines Materials im Januar 2004 ermittelt werden, so müssen sämtliche Bestandsveränderungen seit der initialen Bestandsübernahme (die u.U. schon im Jahr 1999 gewesen sein kann) zu der Berechnung herangezogen werden.

Die Selektion einer Buchungsperiode 01.2004 in einer Auswertung würde somit trotzdem das Lesen aller vorhergehenden Perioden auf der Datenbank bedeuten, was insbesondere das Konzept der datenbankseitigen Partitionierung über den Haufen werfen würde (siehe Kapitel 7.3.1).

Aus diesem Grund werden bei der Komprimierung von Bestands-Cubes sogenannte Stützstellen berechnet und in der Faktentabelle gespeichert, in denen die abschließenden Bestände des entsprechenden Zeitmerkmals abgelegt sind. Auf diese Weise kann sich bei der Selektion der Periode 01.2004 auch das Datenbanksystem auf das Lesen dieser Periode beschränken (siehe Abb. 6–34¹⁶).

Abb. 6–34
Komprimierung
der Faktentabelle bei
BestandsCubes

Die Komprimierung der Faktentabelle bringt somit bei Bestands-Cubes durch die Berechnung von Stützstellen noch einen weiteren wesentlichen Performancegewinn.



16. Die dargestellte Anordnung von Daten und Stützstellen ist vereinfacht. In der technischen Umsetzung werden die Stützstellen in einem gesonderten Speicherbereich zusammengefasst.

Als Besonderheit bei BestandsCubes besteht die Möglichkeit, die Aktualisierung von Bestands-Stützstellen (siehe Kapitel 6.4.1) bei der Komprimierung zu verhindern.

*Aktualisierung von
Stützstellen bei
BestandsCubes*

Dies ist dann erforderlich, wenn Bestandsveränderungen in einen BasisCube geladen wurden, deren zeitlicher Bezug noch vor der Bestandsinitialisierung liegt. Dieses Szenario stellt in der Praxis eher einen Ausnahmefall dar, so dass auch die Option zur Unterdrückung der Stützstellenfortschreibung als Sonderfall betrachtet werden sollte.

Bei der Aktivierung der Metadaten eines BasisCubes werden durch das BW automatisch die beiden Faktentabellen im ABAP Dictionary angelegt. Die Namenskonventionen dieser beiden Tabellen sind in der nachfolgenden Tabelle dargestellt.

Namenskonvention

	Standard-BasisCube	eigendefinierter BasisCube
Name des BasisCubes	Otttttttt	{A-Z}tttttttt
Faktentabelle	/BIO/F0tttttttt	/BIC/F{A-Z}tttttttt
Faktentabelle (kompr.)	/BIO/E0tttttttt	/BIC/E{A-Z}tttttttt
Struktur über alle Merkmale/Kennzahlen	/BIO/V0ttttttttT	/BIC/V{A-Z}ttttttttT

6.4.2 SID-Einträge in Dimensionstabellen

Der Bezug zwischen Stammdaten und Dimensionstabellen wird durch die SID der jeweiligen InfoObjekte hergestellt. In den Dimensionstabellen werden ausschließlich SID, niemals aber Merkmalswerte abgelegt (vgl. Abb. 6–5 auf Seite 69).

Über den Einsatz der SID wird vielfach diskutiert, denn das Schreiben der unmittelbaren Stammdatenschlüssel in die Dimension wäre ähnlich wirkungsvoll und darüber hinaus einfach zu realisieren. Der Einsatz der SID bringt jedoch eine Reihe von Vorteilen mit sich:

- Der kurze 4-Byte-Schlüssel der SID kann bei der Datenanalyse einen Performancevorteil gegenüber langen Merkmalsausprägungen darstellen.
- Durch die Referenzierung auf die SID ist die Kapselung der InfoObjekte (und deren Master Data) möglich. Master Data können damit InfoCube-übergreifend genutzt werden.
- Durch die Referenzierung auf die SID der InfoObjekte stehen bei der Nutzung des Datenmodells auch die Funktionen der Master Data, insbesondere deren Texte, zeitabhängige Informationen und externe Hierarchien, zur Verfügung.

Die Erweiterung des Snowflake-Schemas um die SID hat weitreichende Folgen für die Struktur des Datenmodells. Insbesondere die Nutzung von Navigationsattributen, Texten, Zeitabhängigkeiten und externen Hierarchien grenzt sich das BW von den meisten anderen Data-Warehouse-Systemen ab, die derartige Strukturen nicht in diesem Umfang verwenden.

6.4.3 Dimensionen

Die Relation zwischen einer Faktentabelle und ihren Dimensionstabellen wird mit Hilfe künstlicher Schlüssel, der sogenannten DIM-IDs (Dimension Identification), hergestellt. Bei DIM-IDs handelt es sich um systemgenerierte, vier Byte lange IDs, die bei der Verbuchung von Daten in einen BasisCube durch das BW gebildet werden.

Jeder BasisCube verfügt über maximal 16 Dimensionen. Dreizehn dieser Dimensionen sind für jeden BasisCube individuell definierbar. Jede dieser Dimensionen kann maximal 248 InfoObjekte aufnehmen. Damit ergibt sich eine maximale Anzahl von $13 \times 248 = 3224$ InfoObjekten pro BasisCube (zuzüglich der Attribute jedes InfoObjektes).

Drei der Dimensionen sind für alle BasisCubes fest vorgegeben: die Dimensionen Paket, Zeit und Einheit.

Paketdimension

Inhalt der Paketdimension ist die *Request-ID*. Dabei handelt es sich um eine technische Detaillierungsstufe, welche den Ladevorgang beschreibt, über den Daten in einen BasisCube gelangt sind.

Die Detaillierung der Daten auf Ebene der Request-ID ermöglicht es, Daten einzelner Ladevorgänge gezielt zu administrieren (zum Beispiel Daten aus BasisCubes zu löschen, falls sie sich als fehlerhaft herausstellen sollten), solange sie noch in der normalen Faktentabelle enthalten sind.

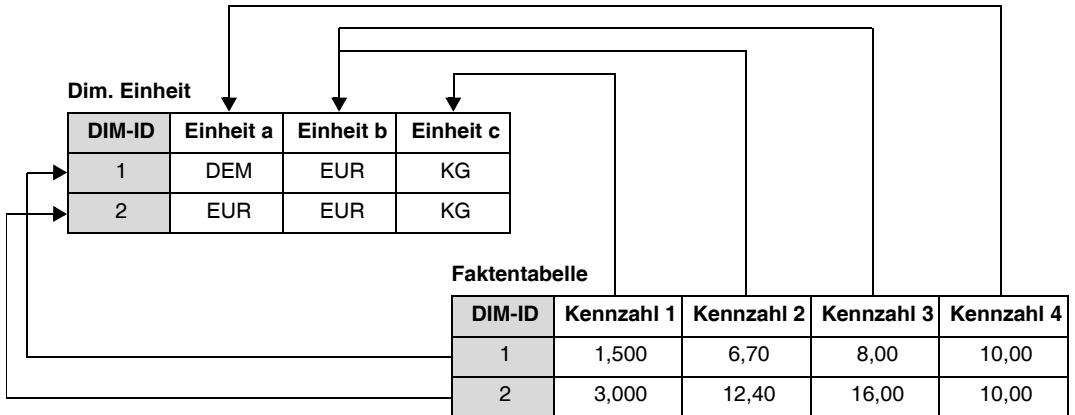
Die Paketdimension ist auch Bestandteil der komprimierten Faktentabelle, wird dort jedoch lediglich durch einen einzigen Initialsatz dargestellt und hat keine weitere Funktion.

Zeitdimension

Die Zeitdimension dient zur Aufnahme von Zeitmerkmalen in einen BasisCubes, sofern es sich dabei um die vom BW bereitgestellten Standard-Zeitmerkmale wie 0CALMONTH oder 0FISCPER handelt (vgl. Kapitel 6.1.2). Eigendefinierte Zeitmerkmale können in dieser Dimension nicht abgelegt werden.

Einheitendimension

Alle Kennzahlen, die eine Menge oder einen Betrag beschreiben, müssen als InfoObjekt mit Bezug zu einem Einheiten-InfoObjekt definiert werden. Während die Kennzahlen in der Faktentabelle abgelegt werden, werden die Einheiten-InfoObjekte in der Einheitendimension abgelegt (siehe Abb. 6–35).



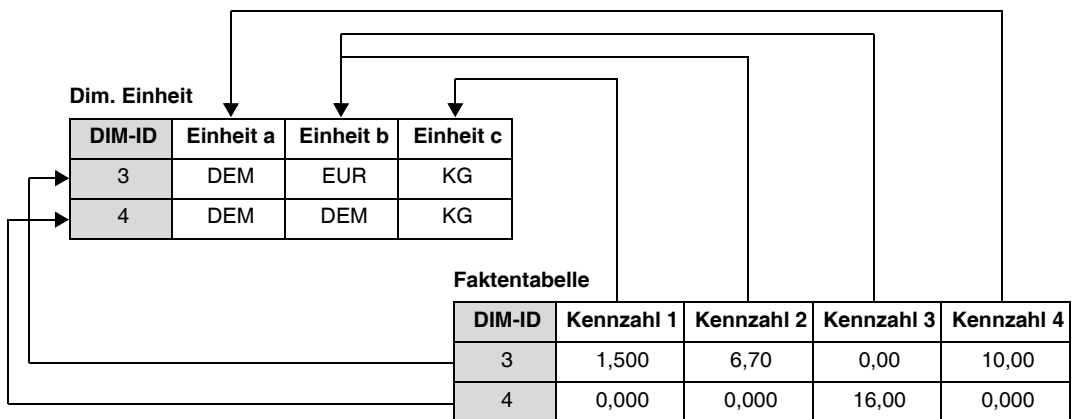
Auf diese Weise wird jede Kennzahl mit einer individuell definierbaren Einheit abgelegt¹⁷.

Dabei ist es auch möglich, dass mehrere Kennzahlen auf dasselbe Einheiten-InfoObjekt referenzieren (in den Abbildungen sind dies die Kennzahlen 2 und 3). Dies sollte dann angewendet werden, wenn diese Kennzahlen im Regelfall identische Einheiten aufweisen.

Weisen solche Kennzahlen doch einmal unterschiedliche Einheiten auf, so lassen sich die Daten dennoch korrekt abbilden, wie Abbildung 6–36 zeigt.

Abb. 6–35

Verwendung der
Einheitendimension (1)



Nachteilig an einer solchen Situation ist der Umstand, dass in der Faktentabelle mehr Datensätze erzeugt werden, als notwendig wären (durch eine Referenzierung von Kennzahl 2 und Kennzahl 3 auf unter-

Abb. 6–36

Verwendung der
Einheitendimension (2)

17. Die Einheit wird, wie zuvor beschrieben, *nicht* in der Dimensionstabelle abgelegt. Dies ist in Abb. 6–35 nur vereinfacht dargestellt. Stattdessen wird die SID für die jeweilige Einheit in der Dimensionstabelle abgelegt.

schiedliche Einheiten könnte dies vermieden werden). Solche Situationen sollten daher nur in Ausnahmefällen vorzufinden sein, um das Datenvolumen eines BasisCubes nicht unnötig zu vergrößern.

Namenskonvention

Bei der Aktivierung der Metadaten eines BasisCubes werden durch das BW automatisch alle erforderlichen Tabellen für die Dimensionen im ABAP Dictionary angelegt. Welche dies sind, ist in der nachfolgenden Tabelle dargestellt.

	Standard-BasisCube	eigendefinierter BasisCube
Name des BasisCubes	Otttttttt	{A-Z}tttttttt
Paketdimension	/BIO/D0ttttttttP	/BIC/D{A-Z}ttttttttP
Zeitdimension	/BIO/D0ttttttttT	/BIC/D{A-Z}ttttttttT
Einheitendimension	/BIO/D0ttttttttU	/BIC/D{A-Z}ttttttttU
1. Dimension	/BIO/D0tttttttt1	/BIC/D{A-Z}tttttttt1
....	/BIO/D0tttttttt..	/BIC/D{A-Z}tttttttt..
13. Dimension	/BIO/D0ttttttttD	/BIC/D{A-Z}ttttttttD

6.4.4 Line-Item-Dimensionen

Bei einer Line-Item-Dimension handelt es sich um eine spezielle Form einer Dimensionstabelle. Hier verweist die Faktentabelle nicht erst auf eine Dimensionstabelle, die wiederum über eine SID auf die Stammdatentabelle eines InfoObjekts verweist. Vielmehr wird die SID, die eigentlich in den Dimensionstabellen steht, direkt in die Faktentabelle aufgenommen. Als Folge darf eine Line-Item-Dimension nur aus genau einem einzigem InfoObjekt bestehen (im Vergleich zu maximal 253 InfoObjekten bei normalen Dimensionen).

Line-Item-Dimensionen dienen zur Verbesserung der Performance und können immer dann eingesetzt werden, wenn eine Dimension lediglich aus einem InfoObjekt besteht.

Ein typischer BasisCube mit einigen »normalen« Dimensionen und einer Line-Item-Dimension ist damit im ABAP Dictionary folgendermaßen definiert:

Feld	Datenelement	Typ	Prüftabelle	Beschreibung
KEY_ZCUBEP	RSDIMID	INT4	/BIC/DZCUBEP	DIM ID
KEY_ZCUBET	RSDIMID	INT4	/BIC/DZCUBET	DIM ID
KEY_ZCUBEU	RSDIMID	INT4	/BIC/DZCUBEU	DIM ID
KEY_ZCUBE1	RSDIMID	INT4	/BIC/DZCUBE1	DIM ID
KEY_ZCUBE2	RSDIMID	INT4	/BIC/DZCUBE2	DIM ID →

Feld	Datenelement	Typ	Prüftabelle	Beschreibung
KEY_ZCUBE3	RSDIMID	INT4	/BIC/DZCUBE3	DIM ID
KEY_ZCUBE4	RSDIMID	INT4	/BIC/DZCUBE4	DIM ID
KEY_ZCUBE5	RSSID	INT4		SID (Line-Item-Dimension)
KEY_ZCUBE6	RSDIMID	INT4	/BIC/DZCUBE6	DIM ID
/BIC/ZABSATZ01	/BIC/OIZABSATZ01	QUAN		Absatzkennzahl 1
/BIC/ZABSATZ02	/BIC/OIZABSATZ02	QUAN		Absatzkennzahl 2
/BIC/ZUMSATZ01	/BIC/OIZUMSATZ01	CURR		Umsatzkennzahl 1
/BIC/ZUMSATZ02	/BIC/OIZUMSATZ02	CURR		Umsatzkennzahl 2

In der Dokumentation der SAP wird der Einsatz von Line-Item-Dimensionen auf die Fälle beschränkt, in denen das abzubildende Info-Objekt annähernd so viele Ausprägungen hat, wie die Faktentabelle an Sätzen aufzuweisen hat. Mit der Line-Item-Dimension sollen Performanceprobleme bei sehr großen und stetig wachsenden Dimensionen beseitigt werden.

Für solche Zwecke wurde die Line-Item-Dimension zwar ursprünglich entwickelt, die Einschränkung darauf ist jedoch nicht notwendig. Line-Item-Dimensionen sind in jedem Fall vorteilhaft für die Performance und sollten immer eingesetzt werden, wenn die Anzahl der InfoObjekte es ermöglicht, pro Dimension nur ein InfoObjekt zu definieren.

6.4.5 Realtime InfoCubes

Realtime InfoCubes sind eine besondere Form der BasisCubes, die ursprünglich in Zusammenhang mit der integrierten Planung¹⁸ des BW eingesetzt werden (bis zur BW-Version 3.x hießen sie noch transaktionale Cubes).

Die Funktionen des BPS stellen spezielle Anforderungen an die Speicherstruktur der verwendeten Cubes. So werden im Rahmen der Planungs- und Simulationsfunktionen des BPS nicht nur Daten gelesen, sondern auch zurückgeschrieben. Dabei handelt es sich nicht um Massendaten, wie sie im Rahmen von Extraktion und Staging anzutreffen sind, sondern um einzelne Datensätze, die von mehreren Benutzern gleichzeitig in den Cube geschrieben und unter Umständen sofort wieder daraus gelesen werden.

Ähnliche Anforderungen stellen auch die im BW 7 bereitstehenden Verfahren für die zeitnahe Analyse von Daten, die mittels Push-Verfah-

18. Zum damaligen Zeitpunkt handelte es sich um das SEM BPS, das mit der Version 3.5 in das BW integriert wurde.

ren oder über Realtime-fähige Datenquellen in kurzen Abständen in die Cubes geschrieben werden (siehe hierzu Kapitel 22).

Die Funktionen des BW BPS entsprechen ebenso wie Push-Verfahren und hochfrequente Aktualisierung von Daten nicht dem ursprünglichen Konzept der BasisCubes, so dass hierfür spezielle Erweiterungen vorgenommen wurden, die zu dem Typ des Realtime InfoCubes geführt haben.

*Technische
Besonderheiten*

Die technische Herausforderung bei einem Realtime InfoCube liegt dabei darin, mehrere schreibende Zugriffe parallel aufzunehmen. Dies bedingt im Falle von Oracle- und DB2/UDB-Datenbanksystemen eine andere Indizierung der Faktentabellen (siehe Kapitel 7.2.1) und kleinere Extend-Sizes. Im Falle aller anderen Datenbanksysteme ist die Speicherstruktur und die Indizierung von transaktionalen Cubes und Standard-BasisCubes identisch.



Der Einsatz von Realtime InfoCubes ist ausschließlich für die Funktionen der integrierten Planung und für Echtzeit-Szenarien erforderlich. Alle BasisCubes, bei denen diese Funktionen nicht zum Einsatz kommen sollen, sollten Sie nicht als Realtime InfoCubes anlegen, da dies Nachteile für die Performance der Datenanalyse zur Folge hat.

Abb. 6–37

*Anlegen eines
transaktionalen
BasisCubes*

Um einen BasisCube als transaktional zu definieren, muss die entsprechende Option bereits beim Anlegen des Cubes gewählt werden (siehe Abb. 6–37).

The screenshot shows the SAP Data Warehousing Workbench interface. The main window is titled 'Data Warehousing Workbench: Modellierung'. On the left, there is a navigation tree with 'InfoProvider' selected. The main area displays the 'InfoCube bearbeiten' dialog box. The dialog has the following fields and options:

- InfoCube:** RCUBE1
- InfoArea:** QUADOX
- Realtime-Cube:** (checked)
- InfoProviderTyp:**
 - Standard-InfoCube (with 'real-time' checked)
 - VirtualProvider
 - basierend auf Datentransferprozess für Direktzugriff
 - basierend auf einem BAPI
 - basierend auf einem Funktionsbaustein
- Systemtyp (Namenraum):**
 - SAP (von SAP ausgeliefert)
 - CUS (beim Kunden generiert)
 - PAR (Partner)
- Deltafähigkeit:** Name der Delta Cache Klasse (empty field)

Sollen Standard-BasisCubes nachträglich in einen transaktionalen Cube umgewandelt werden, so kann dies mit Hilfe des Programms SAP_CONVERT_NORMAL_TRANS erfolgen.

Sollen Realtime InfoCubes für die Funktionen der integrierten Planung genutzt werden, so muss das Ladeverhalten des Cubes zusätzlich entsprechend definiert werden. Das Ladeverhalten kann im laufenden Betrieb umgeschaltet (und wieder zurückgeschaltet) werden (siehe Abb. 6–38).

Integrierte Planung

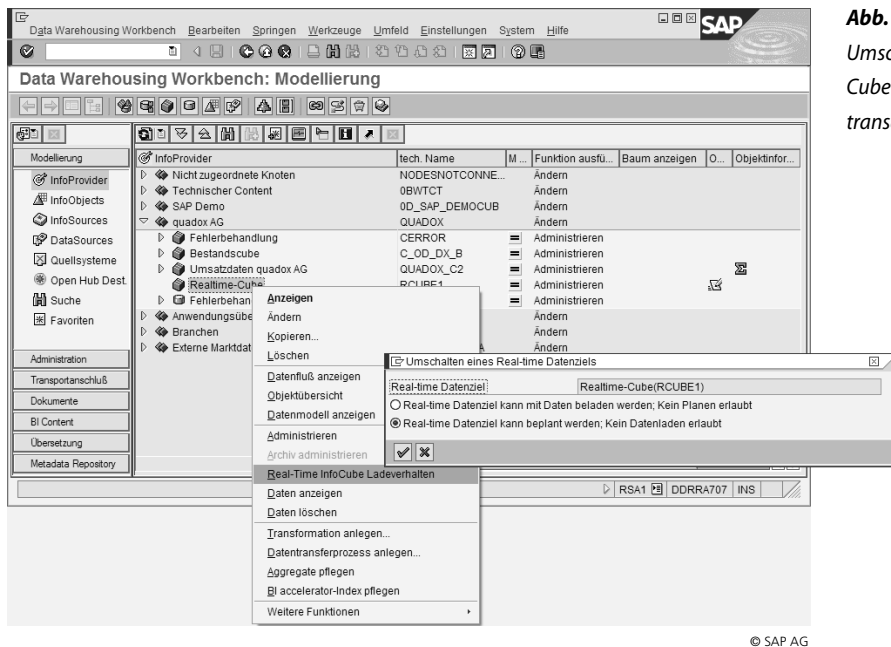


Abb. 6–38

*Umschalten des
Cube-Status bei
transaktionalen Cubes*

Je nach Ladeverhalten kann ein Realtime InfoCube entweder nur über die Prozesse der Staging Engine¹⁹ oder über den Planungsprozessor der integrierten Planung befüllt werden, nie jedoch von beiden gleichzeitig.

6.5 Quellsystemabhängige Daten

In einigen Fällen ist es sinnvoll, Stammdaten oder Bewegungsdaten mit Bezug auf das Quellsystem zu speichern, das diese Daten liefert. Für diesen Zweck ist das InfoObjekt OSOURSYSTEM vordefiniert, das eine

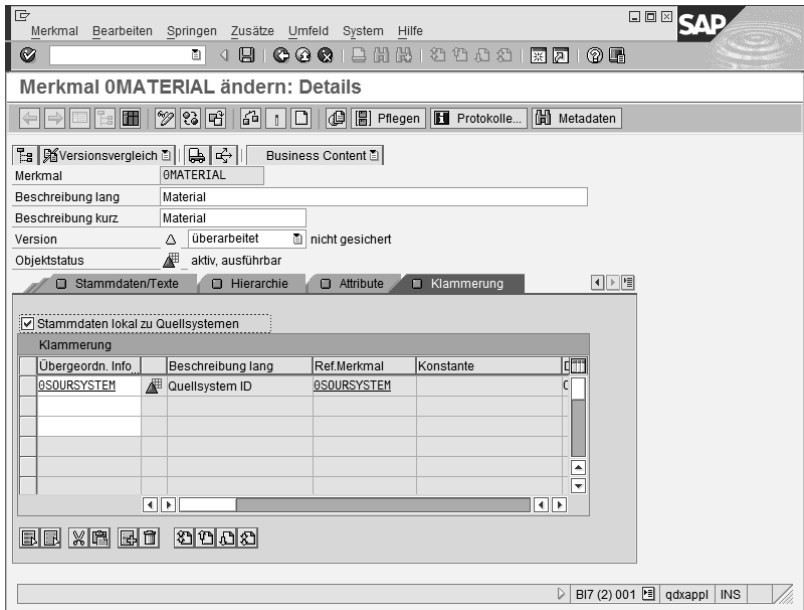
19. Siehe im entsprechenden Buchabschnitt ab Seite 311

Identifikation des Quellsystems enthält und in das jeweilige Datenziel aufgenommen werden kann.

Quellsystemabhängige
InfoObjekte

Im Fall von Stammdaten-InfoObjekten, muss der Bezug zum Quellsystem durch eine Klammerung mit dem InfoObjekt OSOURSYSTEM hergestellt werden. Für diesen Fall ist in der InfoObjekt-Pflege die Option *Stammdaten lokal zum Quellsystem* vorgesehen, über die das BW die Klammerung selbst herstellt (siehe Abb. 6–39).

Abb. 6–39
Quellsystemabhängige
InfoObjekte



© SAP AG

Die Quellsystemabhängigkeit stellt im Falle von InfoObjekten keine zusätzliche Information dar, sondern erzeugt eine vollständige Abhängigkeit aller Stammdaten zur Quellsystem-ID. Beachten Sie vor der Definition einer Quellsystemabhängigkeit, dass beispielsweise die Klammerung der InfoObjekts OMATERIAL an OSOURSYSTEM zur Folge hat, dass es keine Materialnummer 4711 mehr geben kann, sondern nur noch die Materialnummern SYS1/4711, SYS2/4711 usw. Es handelt sich dabei um physisch absolut unabhängige Materialnummern, was bei der Datenanalyse zu Problemen führen kann (bspw. bei ABC-Analysen).