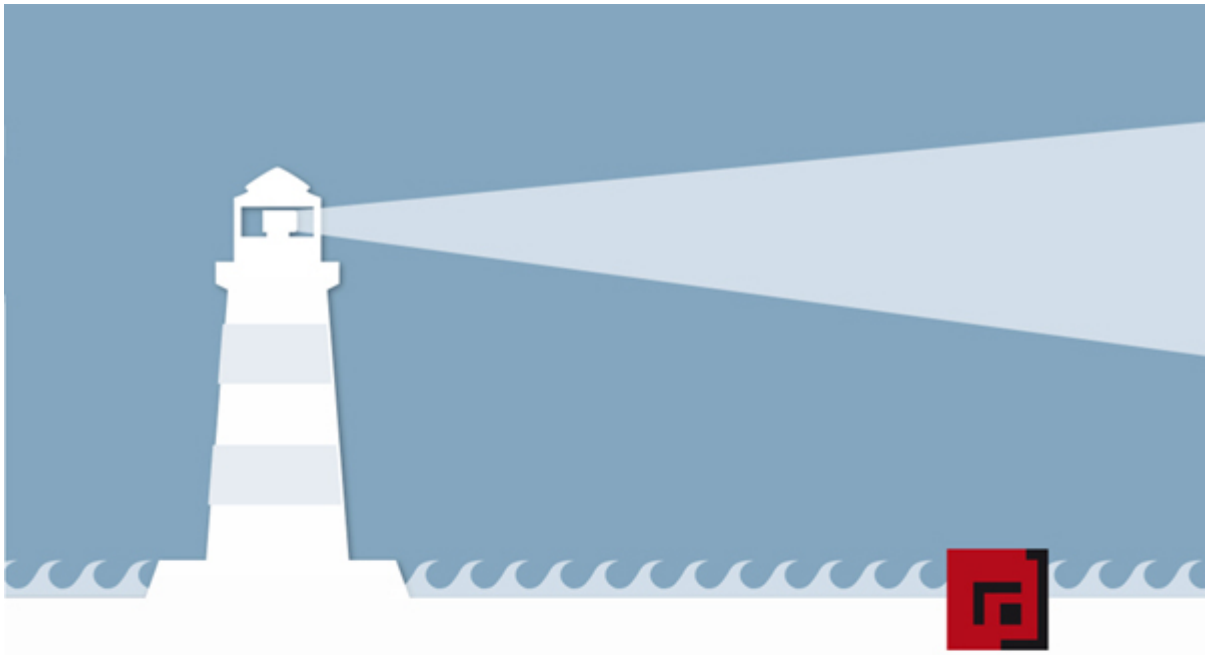


Michael Hahne

Modellierung von Business-Intelligence-Systemen

Leitfaden für erfolgreiche Projekte auf Basis flexibler Data-Warehouse-Architekturen

tdwi EUROPE

dpunkt.verlag

Inhaltsverzeichnis

4 Bestandteile und Varianten des Star-Schemas

Für die Speicherung mehrdimensionaler Datenstrukturen gibt es unterschiedliche Möglichkeiten auf der logischen Modellebene. Ausgehend von der konzeptionellen Beschreibung durch ein T-ADAPT-Modell (siehe Kap. 3) besteht für den Prozess der Gestaltung nun die Notwendigkeit, die logische Implementierungsebene zu bestimmen und das Modell entsprechend zu transformieren.

Eine schon seit vielen Jahren bewährte Technologie steht mit den mehrdimensionalen Datenbanken bereit. Diese auch als OLAP-Datenbank bezeichnete Systemkategorie ist durch Stärken im Bereich der Planung, Simulation und flexiblen Analyse gekennzeichnet. Oftmals wird von Benutzern die Flexibilität und Benutzerfreundlichkeit gelobt. Aus IT-Sicht sind diese Systeme wenig standardisiert, sodass es eine starke herstellerbezogene Abhängigkeit gibt. Die Größe der Modelle ist tendenziell eher limitiert als bei anderen Technologien, dafür können diese Modelle aber recht einfach vom Fachanwender geändert werden und weisen eine sehr gute Analyse-Performance auf. Nachteilig ist die teilweise wenig komprimierte Speicherung und die damit einhergehende begrenzte Größe.

Gerade in Bezug auf nahezu unbegrenzte Datenvolumina und sehr gute Skalierbarkeit haben sich die klassischen relationalen Datenbanksysteme bewährt. Hier können selbst sehr große Modelle sowohl hinsichtlich der Anzahl der Attribute als auch hinsichtlich des Datenvolumens problemlos verwaltet werden. Die damit einhergehende tendenziell schlechtere Performance wird seitens der Datenbankhersteller zunehmend durch spezielle Indextechniken und andere performancesteigernde Maßnahmen adressiert. Die weniger ausgeprägten Stärken im Bereich der Planung und Simulation sind bedingt durch die schlechte Unterstützung für die Erfassung von Daten, sodass vornehmlich Read-only-Lösungen in diesem Segment anzufinden sind.

Ähnlich wie die klassischen relationalen Datenbanken verstehen auch die immer beliebteren spaltenorientierten Datenbanken SQL als Anfragesprache. Hauptdifferenzierungsmerkmal ist die Speicherung der Daten, die nicht zeilenbasiert vorgenommen wird, sondern spaltenbasiert, was sowohl zu erheblichen Performance-Vorteilen führt als auch ein deutlich abgeflachtes

Wachstum der Tabellen nach sich zieht. Sie benötigen erheblich weniger physischen Platz als nicht komprimierte zeilenbasierte Speicherformen. Wegen dieser sehr positiven Eigenschaften sind auch klassische Datenbankanbieter dazu übergegangen, diese Technologie in ihre Produkte zu integrieren.

Nach dem Gesetz von Moore [Moore 1965] verdoppelt sich die Anzahl der Transistoren auf einem IC alle 18 Monate, die Informationsdichte auf einer Festplatte sogar alle 13 Monate nach dem Gesetz von Kryder [Walter 2005]. Nur für die Geschwindigkeit von Festplatten gibt es kein derartiges Wachstum. So bilden heute Festplatten immer mehr den eigentlichen Engpass in hochperformanten Umgebungen. Dies erklärt den deutlich zu spürenden Trend hin zu In-Memory-Konzepten. Gerade reine In-Memory-Datenbanken sind auf extreme Analyse-Performance hin ausgerichtet. Diese ist allerdings aus Kostengründen mit einem eingeschränkten Datenvolumen und bedingter Skalierbarkeit verbunden. Aufgrund der Plattformabhängigkeit gibt es keine Standards, jedoch finden In-Memory-Konzepte zunehmend Eingang in traditionelle Technologien. Intelligente Caching-Konzepte und Datenmanagementstrategien führen auch dort zu besserer Performance bei geringeren Kosten.

Sicherlich kann die Frage nach der geeigneten Technologie nicht losgelöst von den Anforderungen und den Kosten betrachtet werden. Ein Abwägen der Total Cost of Ownership (TCO) in Relation zu den Anforderungen und dem Nutzen ist in jedem Fall notwendig. Für alle Technologien, die auf dem relationalen Modellparadigma basieren, sind allerdings die Ansätze der Modellierung sehr ähnlich und können insbesondere unter dem Schlagwort der Star-Schema-Modellierung subsumiert werden.

4.1 Einfaches Star-Schema

Zur Abbildung mehrdimensionaler Datenstrukturen in relationalen Systemen hat sich mittlerweile ein Standard entwickelt, der unter dem Sammelbegriff Star-Schema bekannt ist. Hierunter ist eine facettenreiche Vielzahl von Varianten einer relationalen Modellklasse zu verstehen, deren Ursprung in der mehrdimensionalen Betrachtungsweise liegt.

4.1.1 Grundform des Star-Schemas

In dem grundsätzlichen Ansatz des Star-Schemas werden die quantifizierenden Informationen in einer zentralen Tabelle gehalten, die Faktentabelle genannt wird. Die Ablage der qualifizierenden Informationen erfolgt in Form von Satellitentabellen, die sternförmig um die Faktentabelle herum angeordnet sind und Dimensionstabellen genannt werden. Der identifizierende Schlüssel in der Faktentabelle ist dabei der zusammengesetzte Schlüssel, bestehend aus den Fremdschlüsseln, die die Primärschlüssel aller Dimensionstabellen referenzieren.

Während in der Faktentabelle die Bewegungsdaten enthalten sind, beinhalten die Dimensionstabellen die Stammdaten und beschreiben die Bewegungssätze. Die Dimensionstabellen haben drei wesentliche Aufgaben:

- Sie beschreiben die Fakten, um daraus sinnvolle Aussagen entstehen zu lassen.
- In ihnen sind die Suchkriterien festgelegt, nach denen Fakten sinnvoll auswertbar sind.
- Sie definieren die Hierarchien, entlang derer die Verdichtungsstufen für die Auswertungen festgelegt werden können.

In der Grundform des Star-Schemas sind alle Kennzahlen als Spalten in einer großen Faktentabelle abgelegt. Als einfaches Beispiel für ein Modell in Form eines Star-Schemas soll das Beispiel der Anwendung im Marketingbereich (bzw. Vertriebsbereich) dienen. Die abstrakte Sicht ist in Abbildung 4–1 dargestellt, in der die Kennzahlen, aufgegliedert nach Produkten und Vertriebswegen, für einzelne Zeitperioden dargestellt werden.

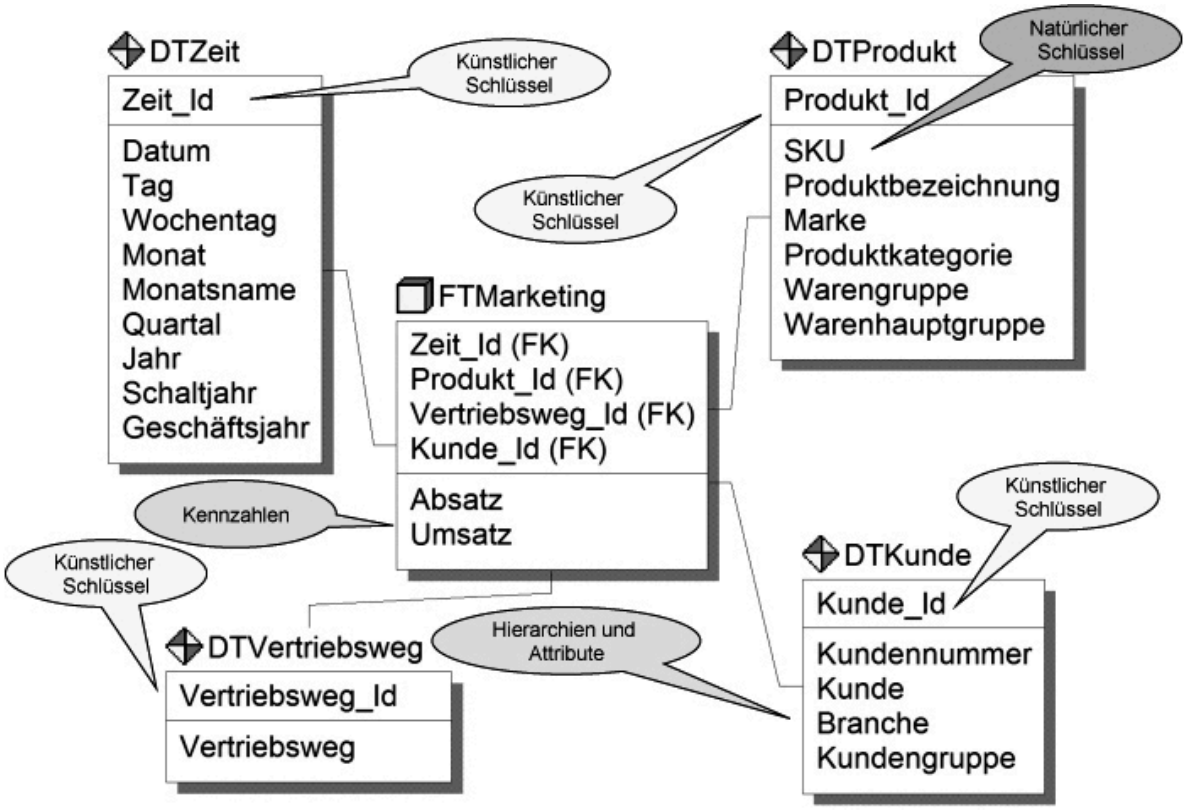


Abb. 4–1 *Entitäten, Schlüssel und Beziehungen im Star-Schema*

Mit diesem Modell, das in den weiteren Ausführungen detaillierter erörtert wird, ist es möglich, Fragen der folgenden Art zu beantworten:

- Wie hoch ist der Gesamtumsatz für ein Produkt?
- Wie groß ist der Anteil am Gesamtumsatz für einen Vertriebsweg?
- Welche Produkte ergeben wann und über welchen Vertriebsweg den größten Anteil am Gesamtumsatz?
- Welche Artikel einer Warengruppe verkaufen sich unterproportional?

Die Granularität der Auswertungen und die Möglichkeiten der Verdichtung ergeben sich aus der Modellierung der Dimensionstabellen, die nachfolgend dargestellt wird.

Star-Schema-Modelle haben wenige Tabellen und sind durch wenige einfache Beziehungen zwischen diesen Tabellen gekennzeichnet. Aufgrund der Orientierung an der Auswertung anstatt der Erfassung der Daten sind diese Modelle auch leicht nachvollziehbar. Die Organisierung der Daten entsprechend der Geschäftssicht des Anwenders ist dabei wichtig, um diese Modelle optimal nutzen zu können.

Als wesentliche erste Strukturkomponente mehrdimensionaler Datenstrukturen wurden die Kennzahlen aufgeführt. Diese numerischen

Messgrößen werden im Star-Schema in den Faktentabellen gespeichert. Diese haben durchaus sehr viele Zeilen, sind aber nicht unbedingt sehr breit. Im Regelfall hat die Faktentabelle einen zusammengesetzten Primärschlüssel, der aus den Fremdschlüsseln zu den Dimensionstabellen besteht.

Ein Star-Schema ist ein relationales Modell, in dem die Kennzahlen in einer zentralen Faktentabelle gehalten werden und gespeichert sind. Die Ablage der qualifizierenden Informationen erfolgt in Form von Satellitentabellen, die sternförmig um die Faktentabelle herum angeordnet sind und Dimensionstabellen genannt werden.

Die Strukturierung der Kennzahlen erfolgt durch die Dimensionen und in diesen jeweils durch die granulare Ebene der Verdichtung. In den Dimensionstabellen sind alle beschreibenden Angaben zu Dimensionselementen sowie zu den Konsolidierungsstufen gespeichert. Damit umfassen diese Tabellen die unterschiedlichsten Attribute und Hierarchieinformationen. Sie sind daher in ihrer Struktur recht breit angelegt, haben aber im Regelfall nicht sehr viele Zeilen. Der Primärschlüssel in den Dimensionstabellen ist ein einfacher nicht zusammengesetzter künstlicher Schlüssel. Dimensionstabellen sind aus Performance-Gründen im Regelfall denormalisiert, eine Betrachtung der Aspekte der im Bedarfsfall optionalen Normalisierung erfolgt später.

Innerhalb eines Star-Schemas wird die Größe maßgeblich von der Faktentabelle determiniert, da diese sehr groß ist. Eine gute Faustregel besagt, dass die Größe einer Dimensionstabelle nicht mehr als 10 bis 20 Prozent der Größe der Faktentabelle haben sollte. Dies kann mit der Abfrage-Performance begründet werden, die insbesondere bei der Verknüpfung von mehreren großen Tabellen im Join tendenziell schlechter ist.[1]

Eine Dimensionstabelle im Star-Schema sollte im Regelfall nicht größer als 10 bis 20 Prozent der Faktentabelle sein.

Neben dem Aspekt der Schlüssel im Star-Schema ist auch die Frage nach den Beziehungen und der referenziellen Integrität interessant. Im einfachen Star-Schema sind dies genau die Fremdschlüsselbeziehungen der Faktentabelle zu den Dimensionstabellen. Ist die referenzielle Integrität eingeschaltet, können in die Faktentabelle nur Datensätze aufgenommen werden, zu denen ein passender Eintrag in den jeweiligen Dimensionstabellen existiert. Dies impliziert, dass im Rahmen der Bewirtschaftung zunächst die Dimensionstabellen zu füllen sind, bevor anschließend die Faktentabelle beladen werden kann. Die dadurch gewährleistete Konsistenz ist ein großer

Vorteil, der allerdings mit den Nachteilen des zusätzlichen Aufwands bei der Bewirtschaftung durch die Wartung der Datenbankindizes abzuwägen ist. Ist die Grundlage des Star-Schemas ein sauberes Core Data Warehouse, wie in der Referenzarchitektur beschrieben, ist der Aspekt der Datenkonsistenz bereits in dieser Schicht adressiert und die referenzielle Integrität auf der Ebene des Data Mart ist nicht notwendig. Ist jedoch eine Architektur mit isolierten Data Marts gegeben, ist es durchaus sinnvoll, die Wahrung der Datenkonsistenz nicht allein in die Hand der ETL-Prozesse zu geben.

In einer mehrschichtigen BI-Architektur mit einem Core Data Warehouse als Grundlage für die Bewirtschaftung von Data Marts ist die Nutzung referentieller Integrität im Star-Schema nicht notwendig, da dies bereits in der harmonisierten bereinigten Datenschicht gewährleistet wird.

Unabhängig davon, ob die Auswertungen eines Star-Schemas werkzeuggestützt oder per Hand erfolgen, kommt als Abfragesprache im Wesentlichen SQL zum Einsatz. Eine Abfrage an ein klassisches Star-Schema, zunächst ohne Berücksichtigung von Aggregaten, hat einen prinzipiellen Aufbau, der in Abbildung 4–2 exemplarisch für das bisher verwendete Star-Schema dargestellt ist.

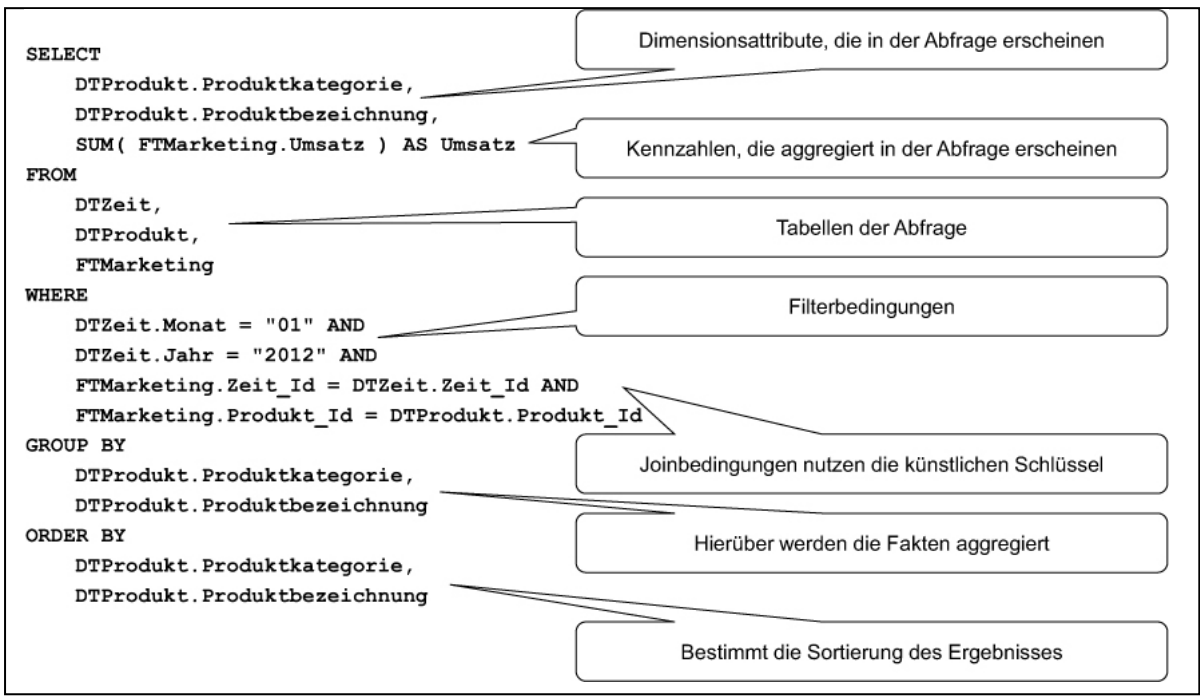


Abb. 4–2 *Abfrage über Fakten (Kennzahlen)*

Diese SQL-Anweisung liefert als Ergebnismenge genau die in der Faktentabelle gespeicherten Werte der Kennzahlen bzw. Fakten, auf die die einschränkenden Bedingungen der *Where-Klausel* zutreffen. Die *Join-Klausel* ist in SQL eine

explizite Formulierung der Joins der Dimensionstabellen mit der Faktentabelle und kann ebenso implizit durch Bedingungen in der *Where-Klausel* formuliert werden. Die explizite Darstellung durch den Join-Operator von SQL ist aber deutlicher und weniger fehleranfällig. Die *Order-Klausel* verändert nur die dargestellte Reihenfolge dieser Tupel der Ergebnismenge.

Für obiges Beispielskript einer Abfrage an das Marketingmodell ergibt sich z. B. das folgende Ergebnis:

```
Produktkategorie     Produktbezeichnung   Umsatz
================================================
Consumer Electronic  PA-1100              5.800
Consumer Electronic  TU-1000              2.000
High Fidelity        PM-300               1.000
High Fidelity        PM-500               1.500
```

Neben den Abfragen an die Fakten, die sicherlich die wesentliche Form von Abfragen an ein Star-Schema darstellen, gibt es auch die Notwendigkeit reiner Dimensionsabfragen etwa für Wertlisten als Grundlage für Selektionen bzw. Filter. Dies soll an der Produktdimension aus Abbildung 4–3 veranschaulicht werden.

Abb. 4–3 *Produktdimensionstabelle*

Ist etwa eine Liste der Produktkategorien aus obiger Produktdimension gewünscht, kann dies durch eine *Select-distinct*-Abfrage realisiert werden. Die Abfrage und das korrespondierende Ergebnis für diesen Fall ergeben sich aus Abbildung 4–4, die eine solche einfache Abfrage an eine Dimension beschreibt. Diese Form einer Abfrage wird auch *Browsedimension* genannt.

```
SELECT DISTINCT
    DTProdukt.Produktkategorie
FROM
    DTProdukt
ORDER BY
    DTProdukt.Produktkategorie
```

```
Produktkategorie
====================
Consumer Electronic
High Fidelity
```

Abb. 4–4 *Einfache Abfrage über Dimensionen (Browsedimension)*

In hierarchischen Dimensionsstrukturen sind vielfältige, durchaus auch komplexere Abfragen vorstellbar. Wird das Beispiel von oben erweitert, so ist es durchaus denkbar, dass für eine spezielle Produktkategorie alle Produkte aufgelistet werden sollen. Diese Abfrage ist in Abbildung 4–5 für die Produktkategorie *High Fidelity* dargestellt.

```
SELECT DISTINCT
    DTProdukt.Produktkategorie,
    DTProdukt.Produktbezeichnung
FROM
    DTProdukt
WHERE
    DTProdukt.Produktkategorie = "High Fidelity"
ORDER BY
    DTProdukt.Produktbezeichnung
```

```
Produktkategorie        Produktbezeichnung
=========================================
High Fidelity           PM-300
High Fidelity           PM-500
```

Abb. 4–5 *Komplexere Abfrage über Dimensionen*

Auf weitere Formen der Abfragen im Fall von Aggregaten und im Fall der rekursiv definierten hierarchischen Beziehungen wird in Kapitel 6 und 7 noch eingegangen.

4.1.2 Abbildung von Kennzahlen und Kennzahlensystemen

Im Ansatz des klassischen Star-Schemas und der bisher diskutierten Varianten erfolgt die Speicherung der betrachteten Kennzahlen, die als eigentliche qualifizierte Wertgrößen für das Modell eine herausragende Bedeutung haben, in der Faktentabelle. Das Kennzahlensystem des Marketingbereichs ist im Folgenden die Grundlage für die Analyse der Möglichkeiten zur Abbildung von Kennzahlensystemen im Star-Schema. Die Abbildung des exemplarischen Kennzahlensystems führt zu der in Abbildung 4–6 dargestellten Faktentabelle.

Abb. 4–6 *Faktentabelle im Kennzahlenmodell*

In dieser Darstellung tritt das wesentliche Defizit dieser Modellierungsvariante des sogenannten Kennzahlenmodells hervor: Jede Kennzahl steht für sich und die Abhängigkeiten, die sich aus der Semantik des Kennzahlensystems ergeben, gehen dabei verloren. Die rechnerischen Abhängigkeiten sind somit ebenfalls nicht darstellbar, sondern sind Teil des Datenbeschaffungsprozesses.

Ein alternativer Ansatz zur Modellierung von Kennzahlensystemen ergibt sich aus deren Struktur als Menge von Knoten, verbunden über gerichtete Kanten, der zufolge dieses System als Dimension des mehrdimensionalen Modells aufgefasst werden kann. Die Zuordnung eines Faktensatzes zu einer Kennzahl erfolgt dann über diese Kennzahlendimension. Da es sich dabei auch um eine Kontozuordnung handeln kann, wird auch gerne vom Kontenmodell gesprochen. Das impliziert, dass in der Faktentabelle nur eine künstliche Faktenspalte für den Wert einer über die Dimensionsausprägung in der Kennzahlendimension näher bestimmten Kennzahl vorhanden ist. Diese Form der Darstellung, wie in Abbildung 4–7 verdeutlicht, hat den Vorteil, dass die hierarchische Information zur Navigation im Kennzahlensystem erhalten bleibt.

Abb. 4–7 *Faktentabelle im Kontenmodell*

In den Darstellungen der Abbildung von Kennzahlensystemen wird im Allgemeinen davon ausgegangen, dass die Beziehungen der Kennzahlen untereinander, die sich aus den Abhängigkeiten in der Berechnung ergeben, auch gleichzeitig die Beziehung zur Navigation widerspiegeln. In der dargestellten Form einer Kennzahlendimension ist das implizit ebenso. Da

jedoch die Berechnung der Kennzahlen ohnehin Teil des Prozesses der Datenbeschaffung und kein direkter Modellbestandteil ist, kann diese Annahme fallen gelassen werden. Die Dimensionshierarchie repräsentiert die Navigationspfade. Da sich aber aus der Hierarchie demzufolge keine Berechnungsvorschriften ableiten lassen, handelt es sich bei dieser Form der Modellierung automatisch um ein Star-Schema mit gespeicherten Aggregaten, auf die in Kapitel 7 näher eingegangen wird.

Die Umsetzung des einen Modells in das andere ist dabei eine häufig anzutreffende Anforderung, da prozessübergreifende Auswertungen auf gleichartigen Typen von Faktentabellen aufsetzen sollten.

Zur Abbildung von Fakten stehen im Star-Schema die Optionen Kennzahlenmodell und Kontenmodell zur Verfügung. Im Kennzahlenmodell sind die Fakten als einzelne Spalten der Faktentabelle repräsentiert, im Kontenmodell gibt es nur eine künstliche Wertspalte, deren Bedeutung sich über die Kontozuordnung bzw. eine Kennzahldimension ergibt. Das Kontenmodell ist sehr flexibel, führt aber unter Umständen im Frontend zu komplexeren Abfragen.

4.1.3 Attribute in Dimensionen

Wie bereits dargestellt sind zwei Arten von Attributen zu unterscheiden: die eher technisch orientierten für die Navigation entlang der Ebenen von Konsolidierungshierarchien und die fachlich bestimmten Attribute, die einen Großteil der analytischen Möglichkeiten eines Modells ausmachen. Für die Implementierung im Star-Schema spielt die Art des abzubildenden Attributs keine Rolle, da jedes Attribut durch eine eigene Spalte in der Dimensionstabelle repräsentiert wird. Die Abbildung von Attributen in der Dimensionstabelle ist exemplarisch in Abbildung 4–8 für den Fall der Zeitdimension dargestellt.

Abb. 4–8 *Attribute in der Dimensionstabelle*

Neben den im Beispiel aufgeführten fachlichen Attributen erfolgt auch die Darstellung der Attribute für Ebene und Generation in Form von eigenen Spalten in der Tabelle.

Im Star-Schema ist der Unterschied zwischen den Spalten für Hierarchiestufen und denen für Attribute, egal welcher Form, nicht erkennbar, da alle Spalten gleichberechtigt nebeneinander stehen. Die strenge Differenzierung zwischen diesen Komponenten mehrdimensionaler Modelle, wie sie auf semantischer Modellebene propagiert wird, ist im Star-Schema nicht ohne zusätzliche Informationen möglich.

Das logische Modell eines Star-Schemas gibt wenig Aufschluss über die Bedeutung und die Beziehungen insbesondere der Attribute in einer Dimensionstabelle. Daher ist für eine semantische Beschreibung zusätzlich ein T-ADAPT-Modell zwingend notwendig.

4.2 Modellierung von Dimensionshierarchien

Die verschiedenen Formen von hierarchischen Strukturen in Dimensionen auf semantischer Ebene standen in Kapitel 3 im Vordergrund. Nun erfolgt die Darstellung der Abbildungsmöglichkeiten dieser Strukturen im Star-Schema. Als Grundlage zur Veranschaulichung dient der Marketingbereich des Anwendungsbeispiels.

4.2.1 Flache Strukturen

Die Dimension *Vertriebsweg* hat als Dimensionselemente ausschließlich die Knoten *Partner, Katalog* und *E-Shop*, die in keiner hierarchischen Beziehung zueinander stehen. Für die Verknüpfung der Dimensionstabellen mit der Faktentabelle dienen im Allgemeinen künstliche Primärschlüssel. Für den Vertriebsweg ergibt sich die Dimensionstabelle in Abbildung 4–9.

Vertriebsweg_Id	Vertriebsweg
1	Partner
2	Katalog
3	E-Shop

Abb. 4–9 *Flache Struktur der Dimensionstabelle Vertriebsweg*

Der in diesem Beispiel verwendete Primärschlüssel ist lediglich eine fortlaufende Nummer, die die Eindeutigkeit der Datensätze gewährleistet und damit keine semantische Information trägt.

4.2.2 Balancierte Baum- und Waldstrukturen

Die Zeitdimension hat in fast allen Data-Warehouse-Modellen eine herausragende Bedeutung, da die zeitliche Qualifizierung des betrachteten Zahlenmaterials essenziell ist. Die kalendarische Sicht ist eine klassische Form einer balancierten Struktur. Für die Abbildung in einem Star-Schema gibt es verschiedene Varianten. Zunächst erfolgt die Abbildung in einer Dimensionstabelle (siehe Abb. 4–10). Dann entspricht jede Konsolidierungsebene der Hierarchie einer eigenen Spalte in der Dimensionstabelle. Die Hierarchiestufen der Verdichtungswege in der Dimension sind in dieser Form die Spalten *Monat, Quartal* und *Jahr* in der Tabelle, wobei von einer Granularität auf Monatsebene ausgegangen wird.

In der skizzierten Form der Dimensionstabelle für die Zeit ist der Primärschlüssel wieder ein einfacher künstlicher numerischer Schlüssel. In einer anderen Variante der Modellierung wird ein zusammengesetzter Primärschlüssel, bestehend aus Komponenten für jede Konsolidierungsstufe, verwendet. Dann sind automatisch auch Schlüsselinformationen zu den

Verdichtungsebenen bereits in der Faktentabelle abgelegt. In vielen Fällen erspart dies in der Abfrage einen Join mit der Dimensionstabelle.

Zeit_Id	Monat_Jahr	Monat	Monatsname	Quartal_Jahr	Quartal	Jahr
1	01.2014	1	Januar	1.2014	Q1	2014
2	02.2014	2	Februar	1.2014	Q1	2014
3	03.2014	3	März	1.2014	Q1	2014
4	04.2014	4	April	2.2014	Q2	2014
…	…	…	…	…	…	…
12	12.2014	12	Dezember	4.2014	Q4	2014
13	01.2015	1	Januar	1.2015	Q1	2015
14	02.2015	2	Februar	1.2015	Q1	2015
…	…	…	…	…	…	…

Abb. 4–10 *Balancierte Baumstruktur der Zeitdimension*

4.2.3 Unbalancierte Strukturen

Eine weitere Klasse von Baum- und Waldstrukturen sind die Strukturen mit unterschiedlich langen Wegen von der Wurzel zu den Blättern, die unter dem Begriff unbalanciert oder unausgeglichen eingeführt wurden. Die dritte Dimension in dem betrachteten Beispiel, die Produktdimension, ist ein guter Kandidat für eine solche Struktur, denn teilweise werden die Produkte neben der Strukturierung nach Produkt- bzw. Warengruppen auch in Untergruppen einsortiert. Eine exemplarische Ausprägung der Dimensionstabelle könnte die in Abbildung 4–11 dargestellte Form haben.

Produkt_Id	Produkt	Untergruppe	Gruppe	Hauptgruppe
1	100Ω	Widerstände	Bauteile	Elektronik
2	1kΩ	Widerstände	Bauteile	Elektronik
…	…	…	…	…
51	250µF	Kondensatoren	Bauteile	Elektronik
52	500µF	Kondensatoren	Bauteile	Elektronik
…	…	…	…	…
101	PA600	NULL	Verstärker	High Fidelity
102	PAX300	NULL	Verstärker	High Fidelity
103	PAX450	NULL	Verstärker	High Fidelity
…	…	…	…	…

Abb. 4–11 *Unbalancierte Produktdimension*

An den NULL-Einträgen in der Spalte der Untergruppe ist die Eigenschaft der Unbalanciertheit erkennbar. Die angegebene Struktur stellt eine unausgeglichene Waldstruktur dar. Zu einer Baumstruktur wird diese durch Hinzunahme einer weiteren Spalte, die allerdings für jede Zeile der Tabelle die Ausprägung *Alle Produkte* hätte und damit nicht wirklich sinnvoll ist.

4.2.4 Parallele Hierarchien

In Abschnitt 3.2.2 diente die Zeitdimension der Verdeutlichung paralleler Hierarchien. Dieses Beispiel soll nun auf den Fall des Star-Schemas angewendet werden. Wie auch bei den anderen diskutierten Strukturformen wird im Fall der parallelen Hierarchie jede Konsolidierungsebene durch eine eigene Spalte der Dimensionstabelle für alle parallelen Hierarchien repräsentiert. Demzufolge

kann die Zeitdimension im Star-Schema wie in der Tabelle in Abbildung 4–12 dargestellt implementiert werden.

Zeit_Id	Monat_Jahr	Monat	Monatsname	…	Jahr	Geschäftsjahr
1	01.2014	1	Januar	…	2014	2013
2	02.2014	2	Februar	…	2014	2013
3	03.2014	3	März	…	2014	2013
4	04.2014	4	April	…	2014	2014
…	…	…	…	…	…	…
12	12.2014	12	Dezember	…	2014	2014
13	01.2015	1	Januar	…	2015	2014
14	02.2015	2	Februar	…	2015	2014
…	…	…	…	…	…	…

Abb. 4–12 *Dimensionstabelle mit paralleler Hierarchie*

An der Tabelle selbst ist dann aber nicht mehr ablesbar, welche Spalten zu welcher Hierarchie gehören. Wie auch bei den anderen Strukturen ist die Information über die Konsolidierungsebenen ebenfalls nicht mehr direkt aus einer Tabelle ableitbar.

4.2.5 Anteilige Verrechnung und Heterarchien

Bei den Strukturformen, in denen die Bedingung der 1:n-Beziehungen zwischen den Konsolidierungsebenen fallen gelassen wird, sind zwei Varianten zu differenzieren. Im einfachsten Fall basiert die Verdichtung auf der üblichen Annahme der Summation, die in der zweiten Form ebenso fallen gelassen wird. Für Letzteres wurde in Abschnitt 2.3.6 das Beispiel der Konsolidierung entlang einer Hierarchie von Tochtergesellschaften in eine Holding aufgeführt. Generell

sind Beteiligungsverhältnisse ein typisches Anwendungsszenario für anteilige Verrechnungen.

In der Implementierung im Star-Schema ist die Bedingung einer 1:n-Beziehung impliziert fixiert, da für ein Tupel jeder Attributwert eindeutig ist. Üblicherweise erfolgt die Realisierung dieser Beziehung zwischen Ebenen im Star-Schema in Form zweier eigenständiger Dimensionen.

Für die Realisierung von Heterarchien der aufgeführten Form mit einer nicht standardmäßigen Verdichtung, die ebenfalls über zwei eigenständige Dimensionen erfolgen muss, ist eine Abfrage verdichteter Werte mit normalen Abfragen nicht möglich. Es sind andere Formen der Berechnung zu berücksichtigen, etwa durch vorberechnete Aggregatwerte. Das Thema Aggregate wird in Kapitel 7 wieder aufgegriffen.

4.3 Normalisierung von Dimensionen

Die bisher dargestellten Star-Schema-Modelle haben die Eigenschaft, dass ihre Dimensionstabellen in erster und zweiter Normalform vorliegen. In der Theorie und Praxis relationaler Datenbanksysteme hat aber die dritte Normalform eine besondere Bedeutung, die dadurch gekennzeichnet ist, dass neben den Eigenschaften der ersten und zweiten Normalform kein Nichtschlüsselattribut transitiv vom Primärschlüssel abhängt.

Die Überführung einer Dimensionstabelle in die dritte Normalform soll im Folgenden am Beispiel der Produktdimension dargestellt werden. Dazu sei die in Abbildung 4–13 dargestellte, um zahlreiche Attribute erweiterte Dimensionstabelle zugrunde gelegt.

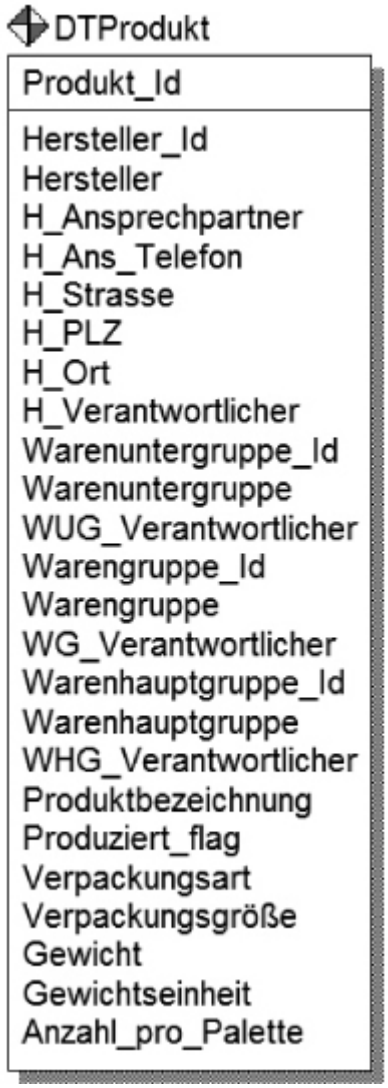


Abb. 4–13 *Herkömmliche Produktdimension*

Die vielen Attribute in der Dimension zeigen auch die vielfältigen Möglichkeiten von Attributen auf unterschiedlichen Konsolidierungsebenen. So ist etwa der WG-Verantwortliche ein Attribut der Ebene der Warengruppe, nicht jedoch der anderen Hierarchieebenen. Die Spalte für den Hersteller eines Produkts repräsentiert dabei eine parallele Hierarchie, zu der auch die Attribute mit »H_« beginnend gehören.

Aus dieser Dimensionstabelle werden nun bis auf die Attribute auf der untersten Ebene und die Schlüssel zu den Ebenen der nächsthöheren Verdichtungsstufen alle Attribute herausgebrochen. Konkret sind alle weiteren Attribute bezogen auf die Hersteller und die Warenuntergruppe sowie alle Attribute der Warengruppe bzw. Warenhauptgruppe aus der eigentlichen Dimensionstabelle entfernt. Durch diesen Schritt der Normalisierung, dessen

Ergebnis in Abbildung 4–14 zusammengefasst ist, wird die Tabelle in mehrere über Beziehungen verbundene Tabellen umgeformt.

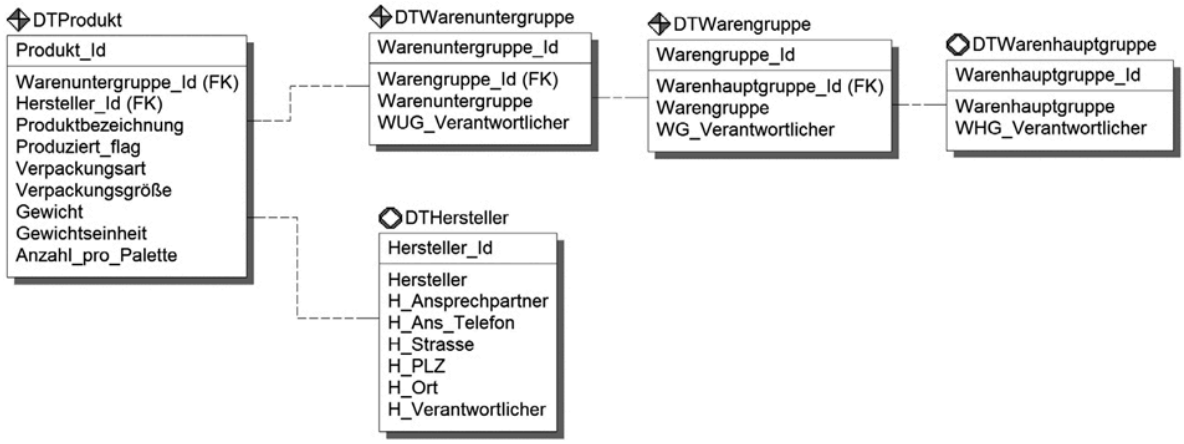


Abb. 4–14 *Normalisierte Produktdimension*

An der Abbildung ist sehr gut die Parallelität zur typisierten Darstellung der Dimension erkennbar, in der die Knoten für die Konsolidierungsebenen den normalisierten Dimensionstabellen entsprechen. Dies ist allerdings nicht der Regelfall, denn die Entscheidung Attribut vs. Dimensionsebene ist einer der Freiheitsgrade in der Modellierung.

Wichtig ist bei diesem Modellierungsansatz, dass bei Abfragen von Attributen, die nicht in der eigentlichen Dimensionstabelle gespeichert sind, die Einbeziehung der weiteren Tabellen notwendig ist. Trotz der gewonnenen Übersichtlichkeit der Dimensionsstruktur leidet die Performance bei dieser Modellierungsvariante zumindest bei entsprechend großen Datenmengen.

Für diesen dargestellten Fall der Abfrageanforderungen bietet sich der Ansatz der Partitionierung an, in dem die Attribute der höheren Ebenen in die zu den unteren Ebenen gehörigen Tabellen dupliziert werden. Hierdurch entsteht eine bewusste Redundanz, die aber Vorteile in der Abfrage-Performance impliziert. Die Ableitung eines partitionierten Modells aus der Darstellung der Produktdimension ist in Abbildung 4–15 wiedergegeben und unterscheidet sich nur durch die redundanten Attributspalten von dem normalisierten Modell.

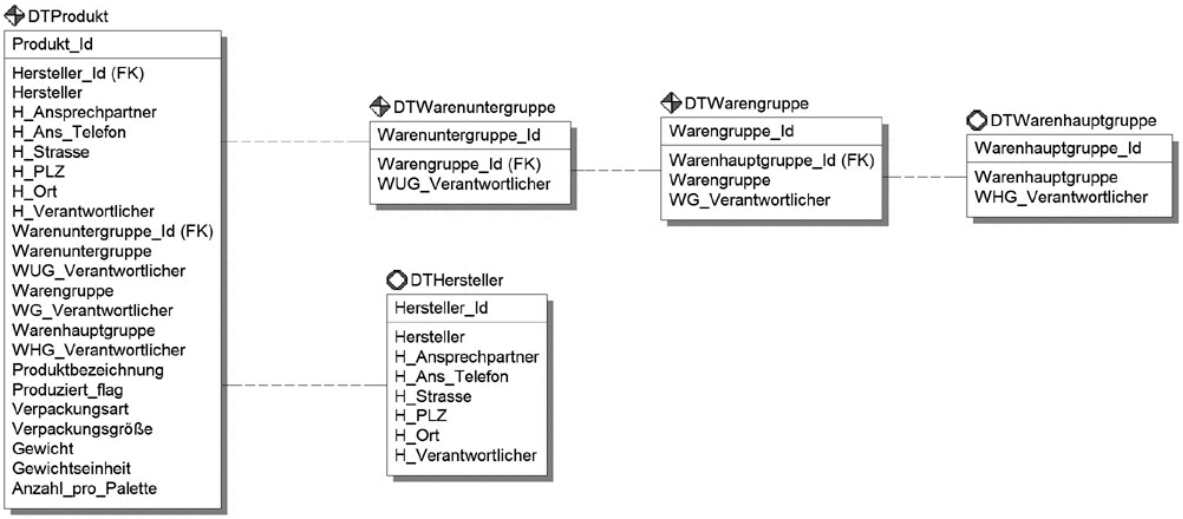


Abb. 4–15 *Partitionierte Dimensionstabelle*

Die Verwendung von normalisierten oder partitionierten Dimensionstabellen führt in Verbindung mit der Verwendung von Aggregattabellen zu einer weiteren Designvariante des Star-Schemas und wird in Kapitel 8 vertieft.

4.4 Übergang von T-ADAPT zum logischen Modell

Alle Data Marts sollten zunächst fachlich modelliert vorliegen, bevor die Gestaltung auf der logischen Ebene erfolgt. Ausgangsbasis ist damit ein T-ADAPT-Modell, aus dem im nachfolgenden Schritt ein logisches Modell abzuleiten ist. Für den Fall der klassischen relationalen Datenbanken als Implementierungsplattform hat sich das Star-Schema bewährt. In einem Prozess der Transformation ist also zunächst aus einem gegebenen semantischen Modell das logische Star-Schema abzuleiten.

Dieser Prozess geht dann aber noch weiter in die physische Modellierung. Auch technisch bedingte Aspekte der Modellierung, wie z. B. Housekeeping-Attribute und Aspekte der Datenbewirtschaftung, müssen jedoch in dieses Modell integriert werden. Ein erster Transformationsschritt führt also im Regelfall zu einer guten Basis, die aber nur als Grundlage für ein weiter angereichertes und verfeinertes Star-Schema-Modell dienen kann.

4.4.1 Transformation von Dimensionen

In der Dimensionsmodellierung in ADAPT bzw. T-ADAPT sind die beiden Typen elementbestimmte und ebenenbestimmte Dimension zu unterscheiden. Für die

einfachen flachen elementbestimmten Dimensionen ist die Übertragung in ein Star-Schema recht einfach, da die Dimensionstabelle neben dem künstlichen Primärschlüssel nur noch das abhängige Attribut zur Aufnahme der Dimensionselemente umfasst. Neben der Struktur der Tabelle, beschrieben durch die DDL (Data Definition Language), ergibt die Transformation aber auch noch die Notwendigkeit, die Ausprägungen dieser Tabelle zu bestimmen, da diese bereits im ADAPT-Modell festgelegt sind. Daher ergibt sich auf der logischen Modellebene die Situation wie in Abbildung 4–16 beschrieben.

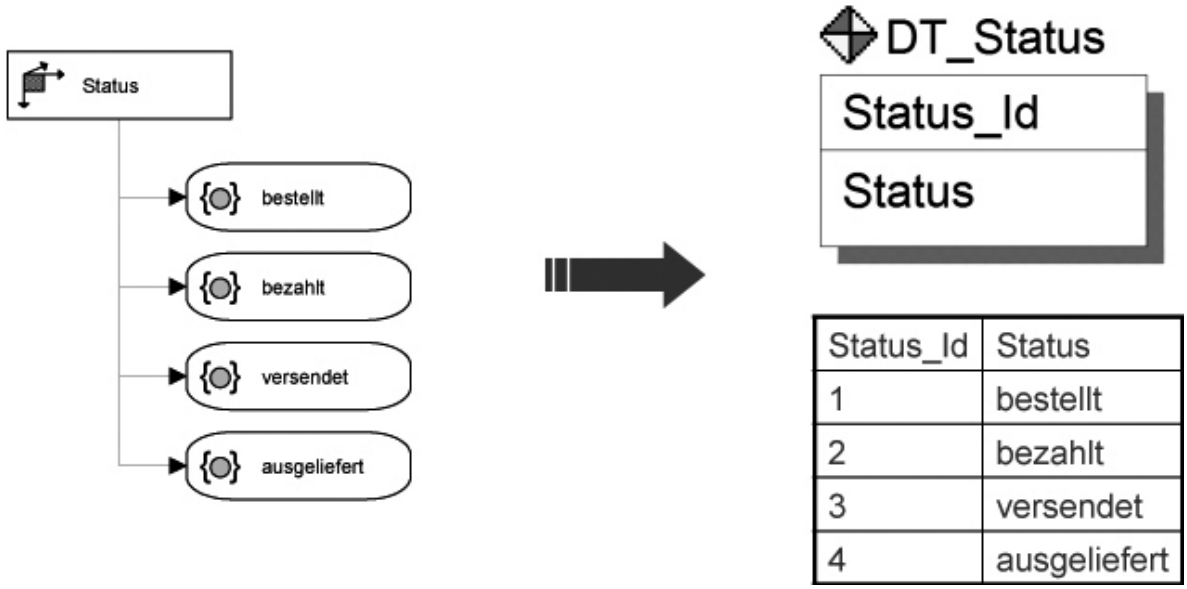
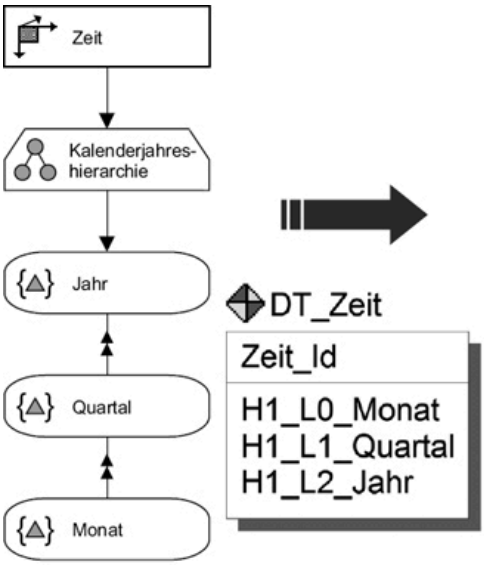


Abb. 4–16 *Flache Struktur einer elementbestimmter Dimension*

Die zweite Form einer Dimension ist durch Konsolidierungsstufen in Hierarchien determiniert. Im einfachsten Fall handelt es sich dabei um balancierte Baumstrukturen in der Hierarchie. Ein typisches Beispiel ist die Zeitdimension aus Abbildung 4-17 mit der Kalenderjahreshierarchie und den Verdichtungsstufen Monat, Quartal, Jahr.

Die Struktur der Dimensionstabelle erhält nun für jede Ebene der Hierarchie eine Spalte sowie den künstlichen Primärschlüssel. Für jedes Blatt, also jedes Element auf der untersten Ebene, in diesem Fall der Monat, gibt es eine Zeile in der Tabelle. Dies geht so nicht aus dem ADAPT-Modell hervor, dient jedoch der Illustration des Aufbaus der Dimensionstabelle. In Abbildung 4–17 sind die Namen der Attribute so gewählt, dass aus diesen die Herkunft der Attribute erkennbar ist. So ist etwa das Attribut, das die Ebene Quartal repräsentiert, als *H1_L1_Quartal* benannt, da dies die Ebene 1 in der Hierarchie 1 darstellt. Dies ist keine generelle Empfehlung, sondern dient nur der Illustration. Aufgrund der möglichen Änderung von Hierarchien ist eine sprechende Benennung von

Spalten in der Dimensionstabelle im Zeitablauf ggf. nicht mehr korrekt. Genau genommen gibt es nur eine Hierarchie in diesem Fall, jedoch ist das Verfahren nicht anders bei mehreren parallelen Hierarchien. Alle Ebenen aller Hierarchien führen zu einem Attribut in der Dimensionstabelle. In der Struktur der Dimensionstabelle ist allerdings die Herkunft einer Spalte im Allgemeinen nicht mehr erkennbar. In existierenden implementierten Modellen können unter Umständen mit Data-Profiling-Hypothesen erstellt werden, eine wirkliche Aussagekraft hat aber nur ein semantisches Modell als Dokumentation, das T-ADAPT-Modell.



Zeit_Id	H1_L0_Monat	H1_L1_Quartal	H1_L2_Jahr
1	Januar	Q1	2014
2	Februar	Q1	2014
3	März	Q1	2014
4	April	Q2	2014
5	Mai	Q2	2014
…	…	…	…
12	Dezember	Q4	2014
13	Januar	Q1	2015
14	Februar	Q1	2015
…	…	…	…
24	Dezember	Q4	2015

Abb. 4–17 *Transformation balancierter Baumstrukturen*

4.4.2 Abbildung von Attributen

Nicht nur Hierarchiestufen müssen in der Dimensionstabelle abgebildet werden, sondern es sind auch Attribute aus dem T-ADAPT-Modell zu berücksichtigen. In Abbildung 4–18 wird das Beispiel der Zeitdimension wieder aufgegriffen und um Attribute erweitert. Die Granularität ist nun die Ebene Tag, die Kalenderhierarchie ist aber weiterhin eine balancierte Baumstruktur. Exemplarisch für die vielen denkbaren Attribute sind nur der *Wochentag* (Montag, Dienstag etc.) und das *Schaltjahr* aufgeführt. Jedes Attribut hängt dabei eindeutig an einer Ebene.

Bei der Transformation in das logische Star-Schema-Modell erfolgt nun die Abbildung von Attributen ebenso wie die von Hierarchiestufen auf Spalten in der Dimensionstabelle. Auch hier dient die sprechende Benennung der Attributspalten nur zu Zwecken der Illustration. So verdeutlicht der Name

H1_L0_A_Wochentag, dass es sich um ein Attribut an der Ebene 0 der Hierarchie 1 handelt, also in diesem Fall die Ebene Tag der Kalenderjahreshierarchie.

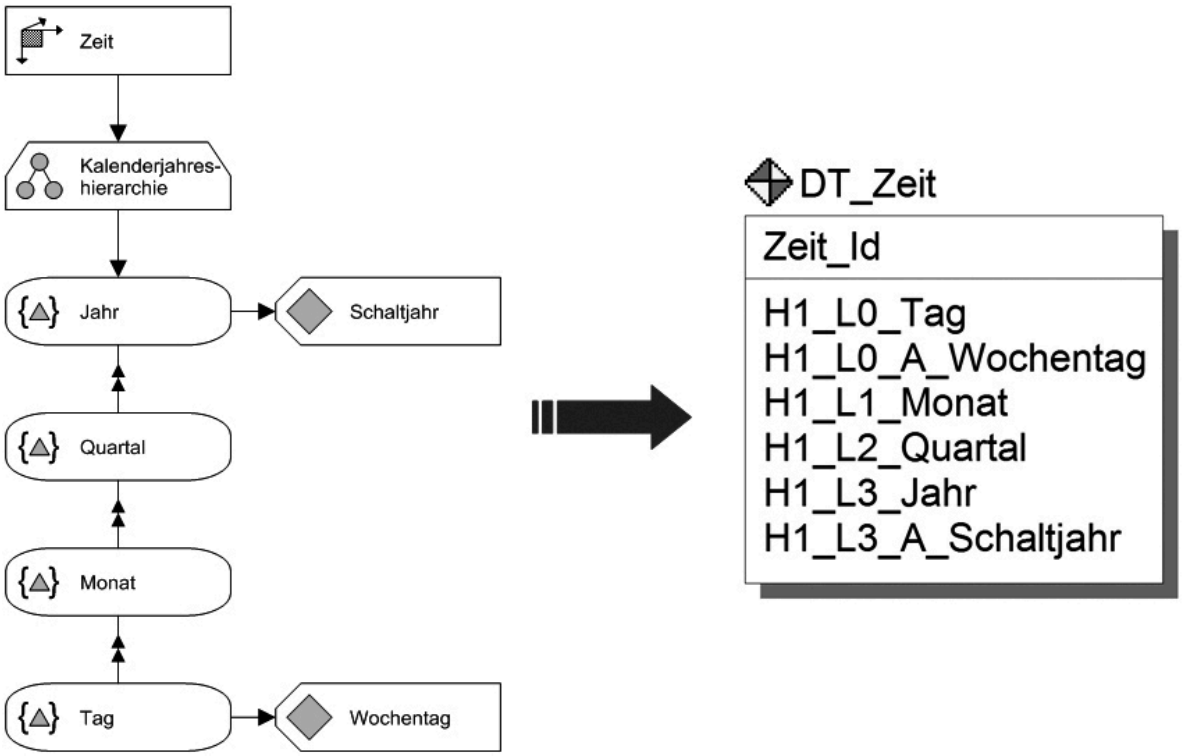


Abb. 4–18 *Ableitung für Attribute auf Hierarchiestufen*

Auch bei der Abbildung von Attributen im logischen Modell tritt das Manko der fehlenden Semantik deutlich zutage, denn es ist nicht ohne zusätzliche Information erkennbar, ob es sich bei einem Attribut bzw. einer Spalte in der Dimensionstabelle um eine Hierarchiestufe oder um ein fachliches Attribut handelt.

4.4.3 Transformation von Scopes

In der semantischen Modellierung erfolgt mithilfe von Scopes bzw. Dimensionsausschnitten die Abbildung besonderer Semantik vor allem im Bereich der Hierarchien und der Attributierung. Zur Verdeutlichung dient das Beispiel der Dimension Vertriebsorganisation in Abbildung 4–19, in der neben den beiden Ebenen Landesgesellschaft und Niederlassung noch zwei weitere Aspekte modelliert sind. Einerseits ist dies die Aufteilung der Niederlassungen in die beiden disjunkten Scopes sowie andererseits die Attributierung der Niederlassungen durch den Niederlassungsleiter mit der zusätzlichen Angabe, dass dieses Attribut nur im Fall der selbstständigen Niederlassungen definiert ist.

Bei der Übertragung dieses semantischen Modells auf die logische Ebene in eine Dimension im Star-Schema sind nun neben den Ebenen der Hierarchie, die hier wieder als Attribut in der Dimensionstabelle auftreten, auch die Scopes sowie das Attribut abzubilden. Auch Attribute an einem Dimensionsausschnitt, also Scope-Attribute, finden ihren Eingang in die Dimensionstabelle als Spalte. Dies verdeutlicht Abbildung 4–19. An der illustrativen Benennung ist erkennbar, dass der Niederlassungsleiter ein Attribut der Niederlassung ist. Allerdings gilt dies nur für eine Teilmenge, was aus der Dimensionstabelle nicht mehr herauszulesen ist.

Scopes werden im Star-Schema auch nur durch Attributspalten in der Dimensionstabelle abgebildet. Im Fall der disjunkten Scopes, das ist in diesem Beispiel gegeben, reicht zur Abbildung eine Spalte in der Dimension. In Beispiel aus Abbildung 4–19 ist dies das Attribut *H1_L0_S_Selbstständig*. Zwei Aspekte des semantischen Modells gehen dabei in der reinen Star-Schema-Modellierung verloren. Es ist nicht mehr erkennbar, welche Ausprägungen des Scope-Attributs zulässig sind, und auch die Eigenschaft, dass der Niederlassungsleiter nur für selbstständige Niederlassungen definiert ist, geht verloren.

Disjunkte Scopes können in ein Feld migriert werden und Attribute an einem Scope haben dann ggf. NULL-Values als Ausprägung. Die semantischen Regelungen können als Anforderungen an die Datenkonsistenz, quasi als Prüfregeln, in die Bewirtschaftung mit aufgenommen werden.

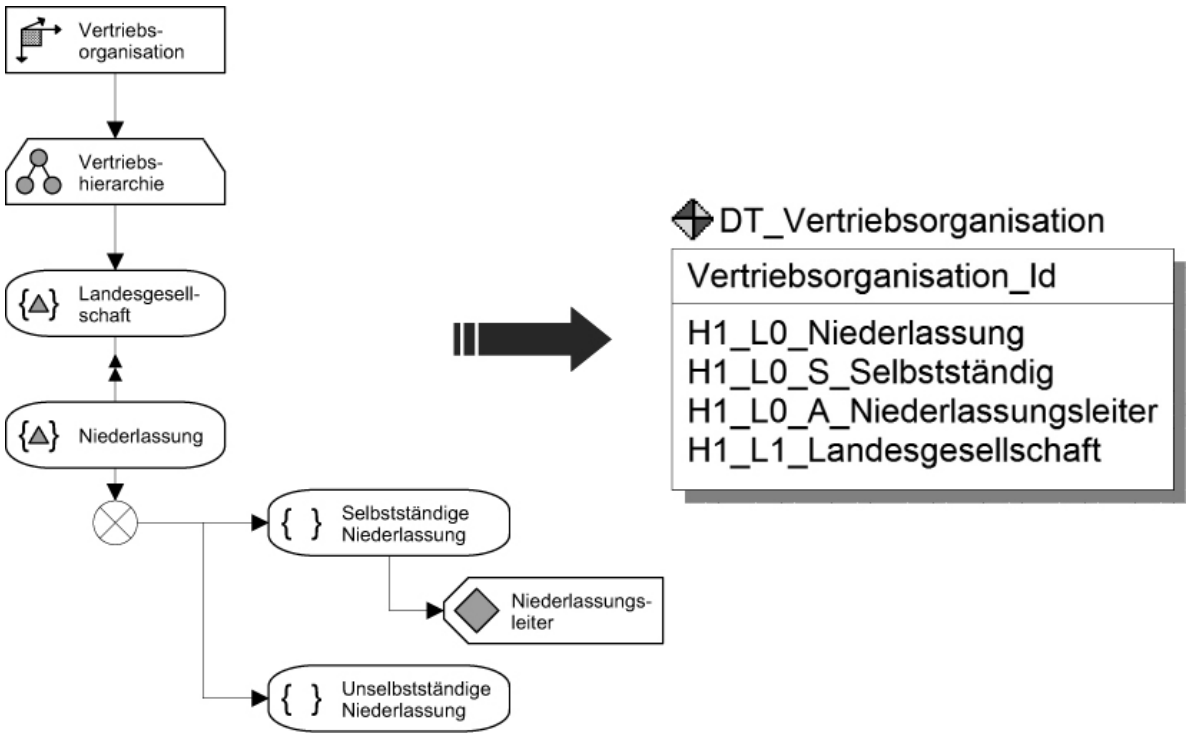


Abb. 4–19 *Transformation von Scope-Attributen*

Neben den disjunkten Scopes gibt es auch eine Vielzahl von Anwendungsfällen für nicht disjunkte Teilmengen. Die zweite Eigenschaft bei der Beziehung zwischen Scopes betrifft die Vollständigkeit. Das Beispiel der Scopes an einer Ebene *Land*, die die Teilmengen der EU-Länder, der Schengen-Länder und der Euro-Länder repräsentieren, verdeutlicht diesen Fall und ist in Abbildung 4–20 visualisiert. Da die Teilmengen nicht disjunkt sind, findet deren Berücksichtigung im Star-Schema in Form von einzelnen Attributen je Scope in der Dimensionstabelle statt. Diese Attribute sind demnach Kennzeichnungen, die die Mitgliedschaft in einer Teilmenge darstellen. Dies kann durch einfache binäre Ausprägungen ermöglicht werden. Üblich sind dafür die Werte *X* und das *Leerzeichen* oder auch *1* und *0*.

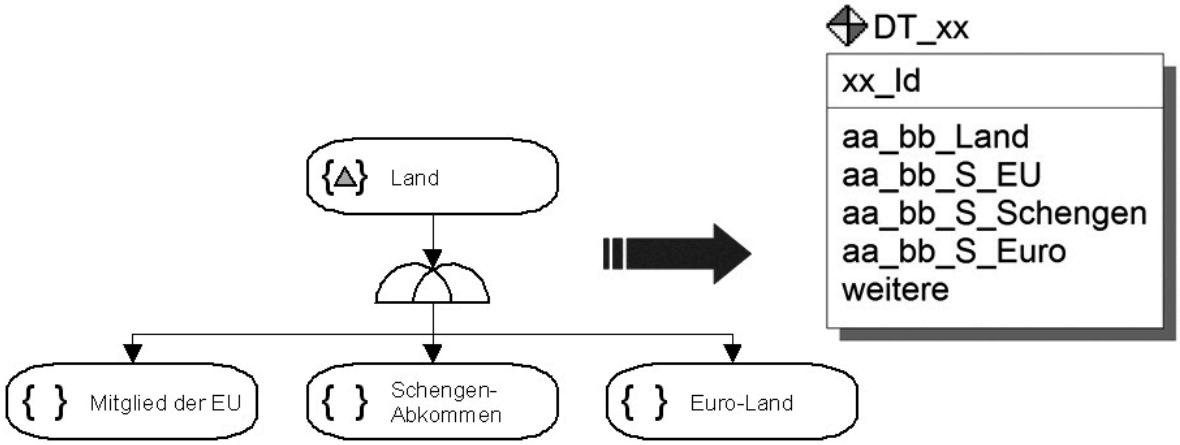


Abb. 4–20 *Abbildung von Scopes (nicht disjunkt, aber vollständig)*

Im Fall der nicht vollständigen Menge von Scopes ist es erlaubt, dass Elemente der Ebene *Land* in obigem Beispiel keiner der drei aufgeführten Teilmengen angehört. Ist die Menge der Scopes in Summe vollständig, ergibt also die Vereinigungsmenge wieder die gesamte Ebene, dürfen nicht alle Attribute zur Teilmengenzugehörigkeit auf 0 sein. Zumindest eine Teilmenge muss immer zugeordnet sein.

Im abstrakten Beispiel aus Abbildung 4–21 mit mehreren Scopes, die eine Ebene *Digitale TV-Receiver* in nicht disjunkte Teilmengen vollständig aufteilt, reichen die Attribute der Zugehörigkeit zu der Teilmenge ebenfalls in der Dimensionstabelle aus, jedoch ist die Eigenschaft der Vollständigkeit verloren gegangen. Auch dies kann ggf. im Rahmen der Datenbewirtschaftung als Prüfregel zur Datenkonsistenz mit aufgenommen werden.

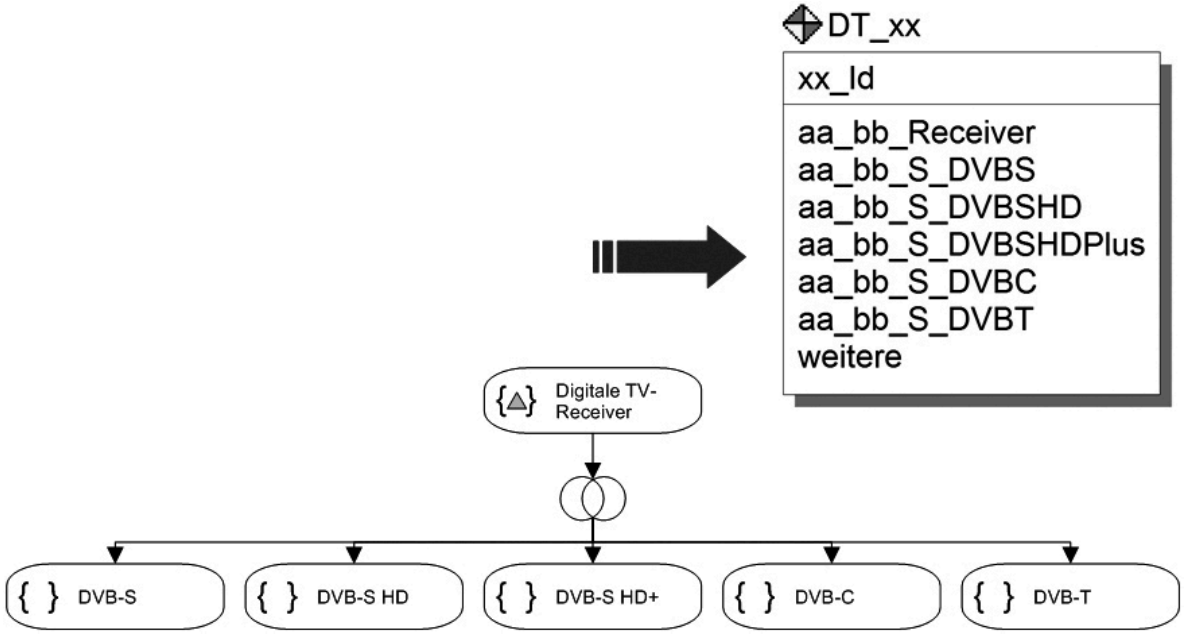


Abb. 4–21 *Abbildung von Scopes (nicht disjunkt, aber vollständig)*

Bei disjunkten Scopes reicht im Allgemeinen auch ein einzelnes Attribut. Bei besonderen Teilmengen, die einer spezifischen Beachtung bedürfen, können einzelne Attribute ggf. hilfreich sein.

Nicht immer sind im Fall der disjunkten Scopes einzelne Attribute sinnvoll. Dies wird besonders im Fall der Ebene *Geschlecht* aus Abbildung 4–22 deutlich. Eine strenge formale Transformation in einzelne Attribute führte hier nur zu einer extremen Redundanz der Information. In der Dimensionstabelle wären dann in dem Attribut Geschlecht *Männlich* und *Weiblich* abgebildet, in den Attributen zu den Scopes steht dann im Fall *Männlich* ein *X* und entsprechend umgekehrt im Fall *Weiblich*.

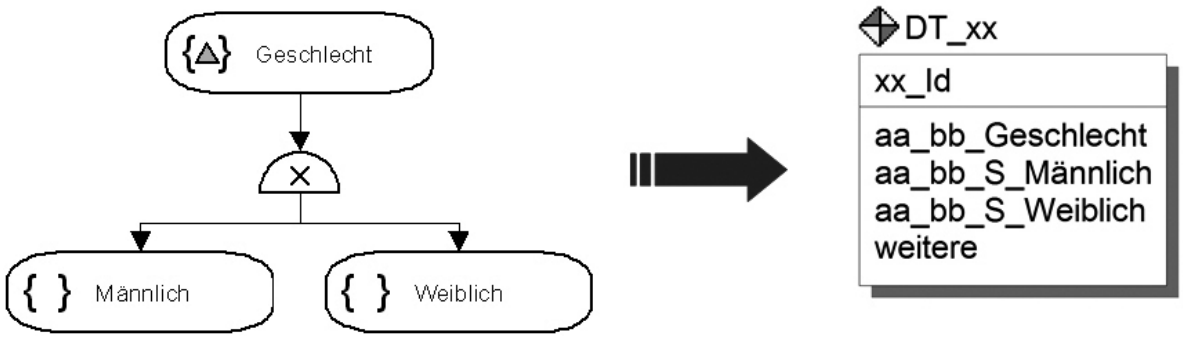


Abb. 4–22 *Abbildung von disjunkten Scopes (nicht vollständig)*

Besser ist in diesem Fall die Modellierung auf der logischen Ebene in Form von einem Attribut für die Abbildung des Geschlechts. Auf die Scopes kann bereits in der ADAPT-Modellierung verzichtet werden. Dies drückt insbesondere Abbildung 4–23 noch einmal aus.

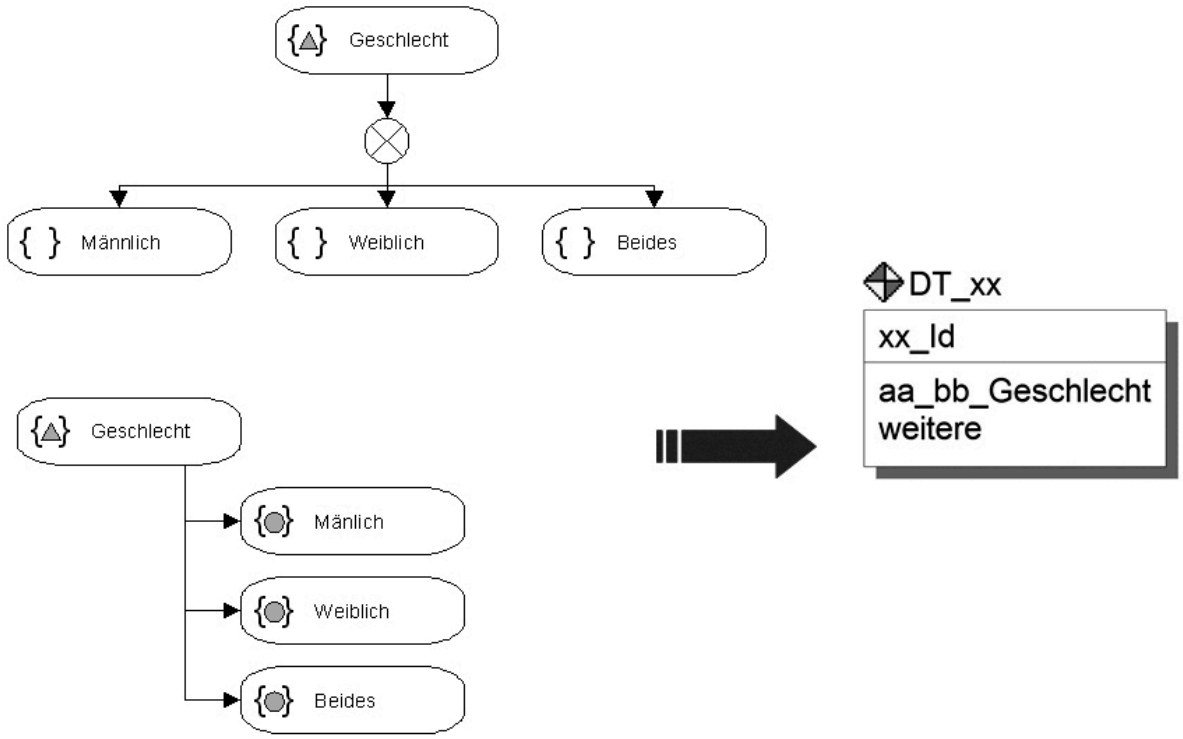


Abb. 4–23 *Einelementige disjunkte Scopes*

Sind im ADAPT-Modell für einzelne Ebenen die Wertausprägungen aufgeführt, so geht diese Liste im Allgemeinen im Star-Schema verloren. Nur im Fall der Einzelobjektdimension wird dies durch die Ausprägungen in der Dimensionstabelle direkt mit dargestellt.

In der semantischen Modellierung mit T-ADAPT gibt es auch die Möglichkeit der Attributierung auf verschiedenen Ebenen. Im Beispiel aus Abbildung 4–24 sind dies die Attribute *Manager* und *Location*, die an der Dimension hängen und damit für alle Ebenen in jeder Hierarchie gelten. Hier ist das also die *Sales Org Hierarchy* mit den Ebenen *Region, National Sales Org* und *Sales Office*.

Bereits in der semantischen Modellierung wurde betont, dass es sinnvoller ist, einzelne explizit benannte Attribute an jeder Ebene zu benutzen, z. B. die Attribute *Region-Manager, National-Sales-Office-Manager* und *Sales-Office-Manager*. Dies entspricht auch genau der Abbildung in der Dimensionstabelle, denn es muss für jede Ebene dieses Attribut einzeln definiert werden. Dies ist in Abbildung 4–24 visualisiert. Dort sind dann auch die anderen Attribute in der gleichen Form abgebildet.

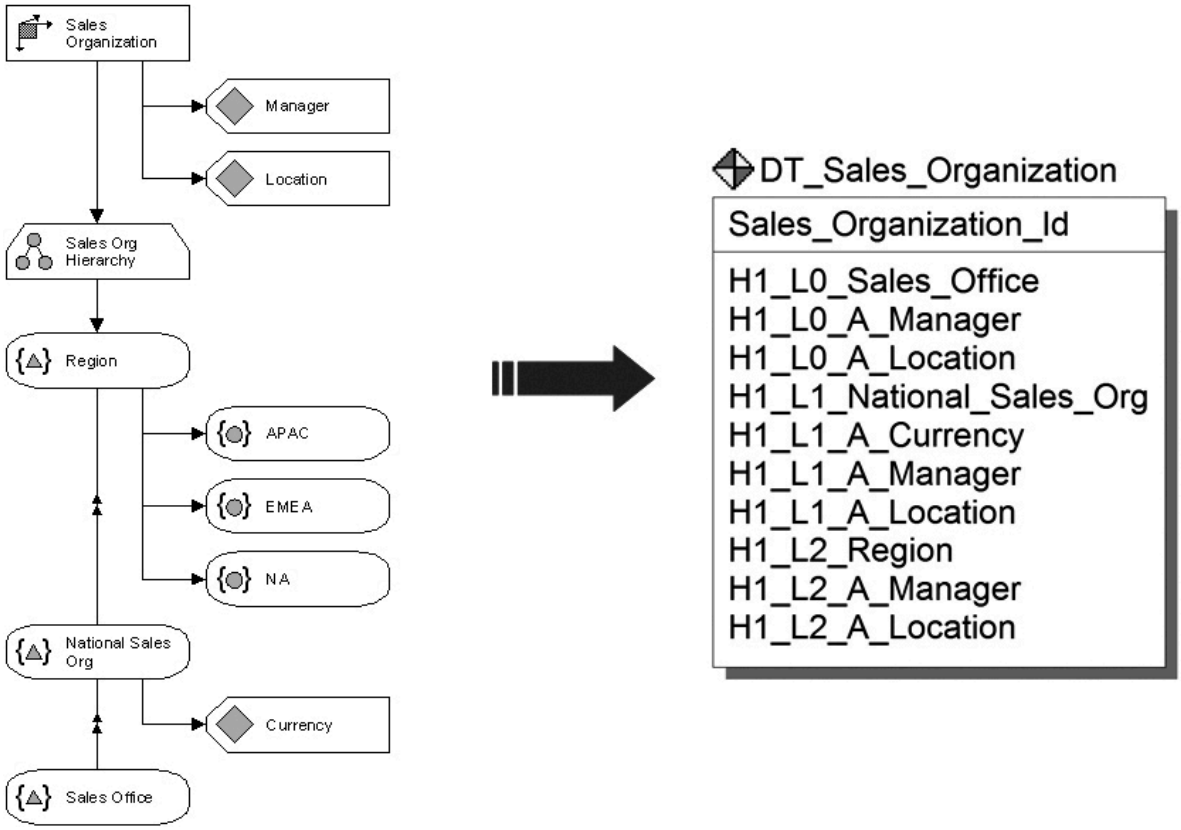


Abb. 4–24 *Attribute auf unterschiedlichen Ebenen*

4.4.4 Behandlung spezieller ADAPT-Varianten

In der dimensionalen Modellierung gemäß OLAP gibt es in den meisten hierarchischen Dimensionen auch ein oberstes Verdichtungselement, das sogenannte Topelement. Dieses stellt die höchste Aggregationsstufe in der Dimension dar. Neben der expliziten Modellierung dieser obersten Verdichtungsstufe in T-ADAPT gibt es noch die Form der impliziten Modellierung wie in Abbildung 4–25 dargestellt. Die dortige Produkthierarchie mit den Ebenen *SKU (Stock Keeping Unit, Eindeutige Produktnummer), Produktkategorie* und *Marke* hat eine oberste Verdichtungsstufe, die sich aus dem Doppelpfeil zum Hierarchieobjekt ergibt. In relationalen Star-Schema-Modellen ist ein solches Attribut für diese einzelne oberste Wurzel einer Hierarchie nicht notwendig, da die komplette Aggregation in einer Dimension diese Tabelle in einer Abfrage gar nicht berücksichtigen muss. Eine derartige redundante Information kann also auch gut weggelassen werden, wenn darauf aufsetzende Werkzeuge nicht unbedingt eine physisch präsente oberste Hierarchiestufe in der Dimensionstabelle benötigen.

Parallele Hierarchien münden letztlich auch nur in viele Attribute in der Dimensionstabelle. Die Semantik dazu ergibt sich eben nicht mehr aus der Tabelle, sondern nur aus dem korrespondierenden semantischen Modell.

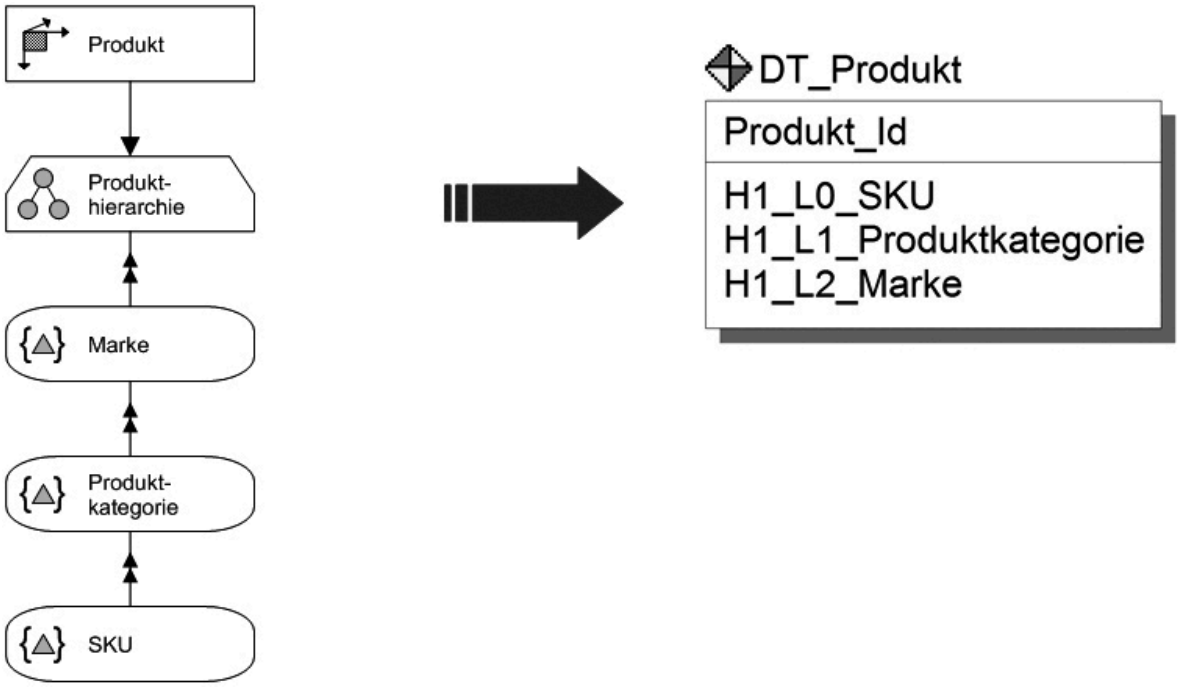


Abb. 4–25 *Darstellung von Hierarchien mit totaler Verdichtung*

In einer besonderen Form der parallelen Hierarchie kann es vorkommen, dass einzelne Hierarchiestufen gemeinsam von den parallelen Hierarchien benutzt werden. Dies ist nach den Empfehlungen der semantischen Modellierung bereits durch die eindeutige Benennung von Dimensionsstufen gegeben. So handelt es sich nach unserer Konvention bei der Ebene *Land* in Abbildung 4–26 bereits um dieselbe Ebene. Die zusätzliche Notation mit der Verbindungslinie ist nicht notwendig. Unabhängig von der Darstellung in T-ADAPT ist jedoch die Frage der Transformation auf die logische Ebene zu beantworten. In diesem Beispiel braucht das Attribut *Land* in der Dimensionstabelle nur einmal aufzutauchen. Sollen die beiden *Land*-Zuordnungen verschieden sein können, sollten erstens im semantischen Modell die Ebenen unterschiedlich benannt werden und zweitens sind dafür dann in der Dimensionstabelle auch zwei Attribute notwendig.

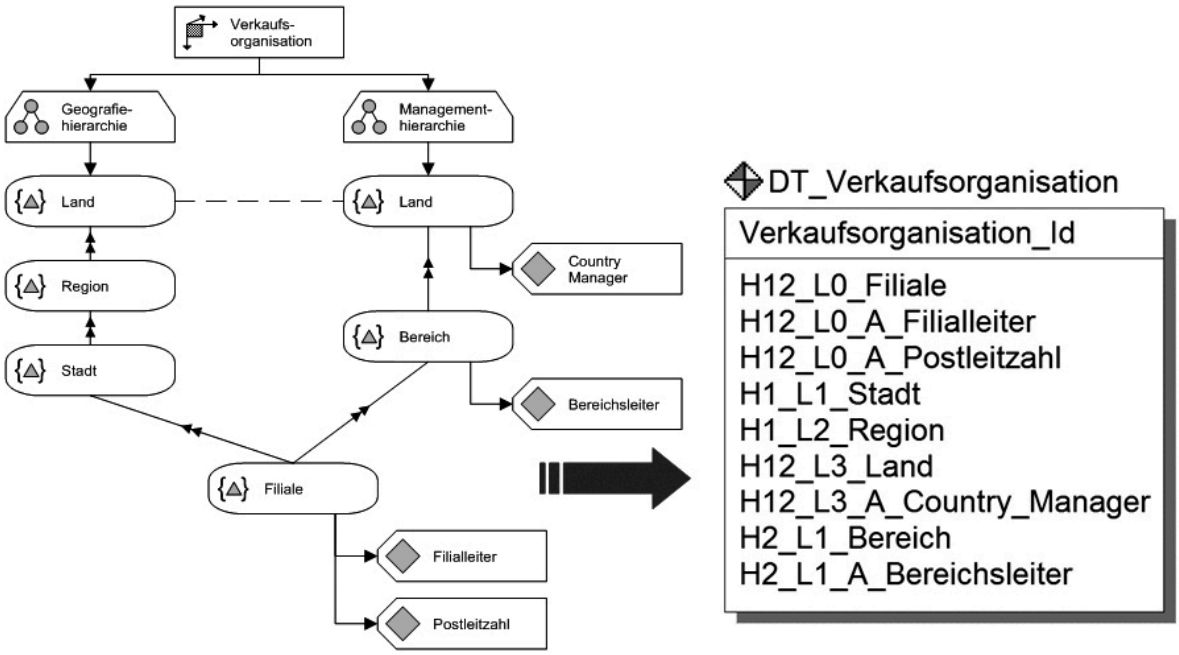


Abb. 4–26 *Ableitung paralleler Hierarchien mit gemeinsamer Ebene*

Die Ableitung eines logischen Star-Schema-Modells aus dem semantischen Modell nach den skizzierten Grundregeln ist recht einfach und führt zu einem ersten Wurf des logischen Modells. Dieses ist dann aber noch weiter zu verfeinern, so müssen etwa die Housekeeping-Attribute in den Dimensionen ergänzt werden, Zuordnungen zu Ladeprozessen sind zu berücksichtigen etc.

Der Prozess endet also nicht an dieser Stelle, sondern führt direkt weiter bis zum physischen Modell mit genau diesen weiteren Eigenschaften. Ebenso ist es denkbar, dass mehrere T-ADAPT-Modelle durch ein einzelnes Star-Schema-Modell abgedeckt sind, oder andersherum, dass es für die Umsetzung eines semantischen Modells mehrerer Star-Schema-Modelle bedarf.

Der Prozess der Gestaltung dimensionaler Strukturen beginnt mit einem semantischen T-ADAPT-Modell, dessen Transformation in ein relationales Star-Schema auf der logischen Modellebene die Grundlage für das physische Modell ist. Alle drei Ebenen sind notwendig. Nur das T-ADAPT-Modell deckt die gesamte fachliche Semantik ab. Auf den anderen Ebenen geht Semantik verloren und technische Zuordnungen kommen hinzu.

Auf die Transformation von Many-Many-Beziehungen zwischen Dimensionsstufen wird in Abschnitt 6.3 noch eingegangen.

4.5 Modellierung von Parent-Child-Hierarchien

Während die Gestaltung vielfältiger Hierarchieformen im Rahmen der semantischen Modellierung keine wesentlichen Limitierungen erfährt und auch unbalancierte Hierarchien sowie gar Hierarchien mit Blättern unterschiedlicher Granularität durchaus nicht selten sind, stellt dies für die Implementierung im Rahmen von Star-Schema-Strukturen in relationalen Datenbanksystemen durchaus eine ernst zu nehmende Herausforderung dar. Dieser Abschnitt behandelt die Abbildung von rekursiv definierten Hierarchien im Star-Schema und geht dabei insbesondere auf rekursive SQL-Abfragen ein.

Bei der bisher im Buch dargestellten Art der Modellierung von Hierarchien in Dimensionen wurde für jede Hierarchiestufe (Level) und deren weitere Attribute eine Spalte in der Dimensionstabelle hinzugefügt. Der Vielfalt der auf diese Weise modellierbaren Hierarchien steht die starre Struktur gegenüber, die Strukturveränderungen wie etwa das Einfügen neuer Verdichtungsstufen erschwert.

Alternativ können hierarchische Beziehungen der Dimensionselemente untereinander auch durch sogenannte Parent-Child-Tabellen modelliert werden, d. h., in der Dimensionstabelle wird zu jedem Element ein zugeordnetes Element der übergeordneten Ebene angegeben. Bei dieser Modellierungsvariante ist das Level-Attribut sehr hilfreich, um die Zugehörigkeit eines Elements zu einer bestimmten Hierarchiestufe zeitnah abfragen zu können. Die resultierende Dimensionstabelle hat weniger Spalten, in der minimalen Ausprägung lediglich das Dimensionselement, das übergeordnete Element und einen künstlichen Primärschlüssel. Das Navigieren (*Drill-down* und *Rollup*) in Dimensionstabellen, die auf diese Art modelliert sind, ist aufwendiger abzubilden als in einer Dimensionstabelle, in der jeder Ebene eine eigene Spalte zugewiesen wird.

Als Anwendungsbeispiel für die Modellierung von rekursiven Beziehungen wird im Folgenden das Modell aus Abbildung 4–27 herangezogen. In dieser einfachen Form ist dies ein vierdimensionales Modell und besteht aus den Dimensionen Zeit, Kostenstelle, Kostenart und Szenario sowie der Faktentabelle.

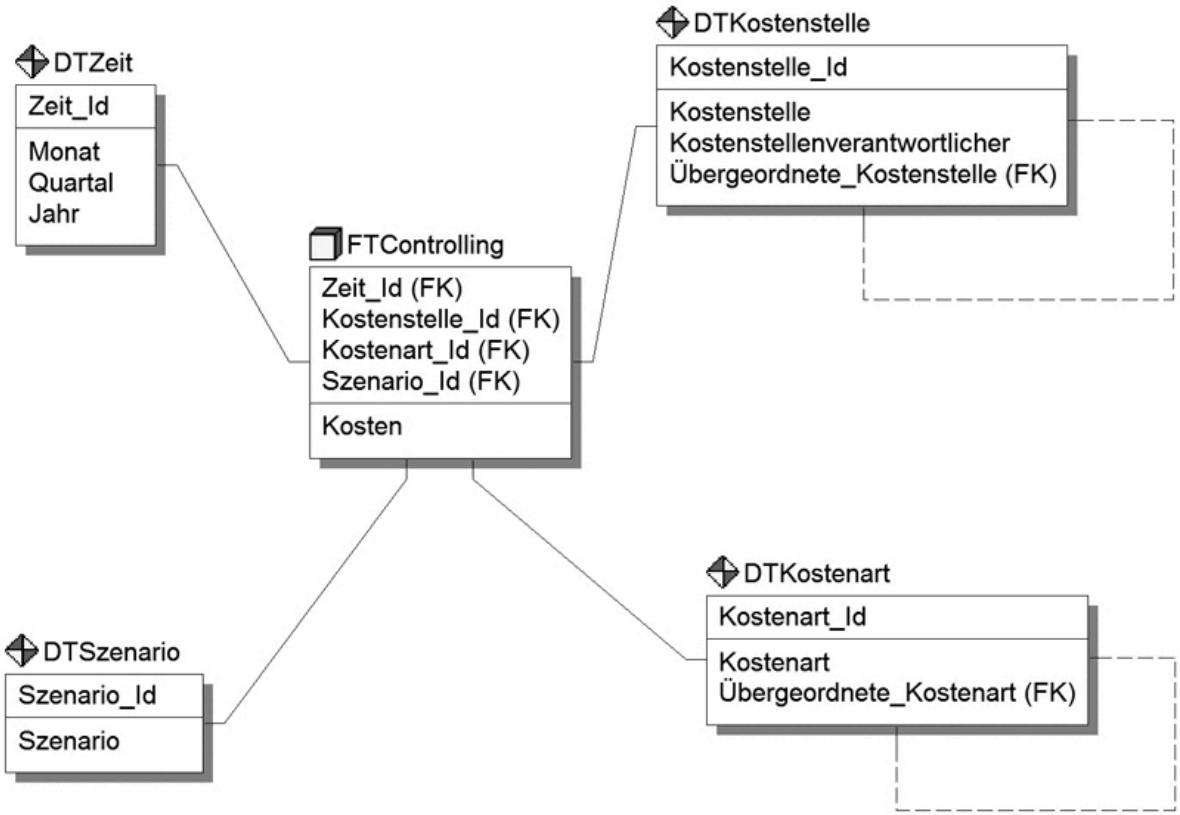


Abb. 4–27 *Rekursive Beziehungen in Dimensionen*

Eine mögliche Hierarchie der Kostenstellen ist in Abbildung 4–28 dargestellt, wobei es sich um eine unbalancierte Struktur handelt. Der Umgang mit diesen Strukturen ist gerade eine Stärke der rekursiven Modellierung. Die Fakten können in diesem Fall auf alle Ausprägungen gebucht werden, z. B. auf alle Kostenstellen 4, 41, 45, 451, 452.

Abfragen können die Hierarchie sowohl nach unten als auch nach oben traversieren. Folgende Abfragen sind dabei exemplarisch denkbar:

- Alle Blattelemente unter einem Element, etwa alle Blätter unter 45: 451, 452
- Alle Elemente eines Teilbaumes, etwa Teilbaum inkl. 45: 45, 451, 452
- Alle übergeordneten Elemente, etwa alle Elemente oberhalb von 452: 45, 4

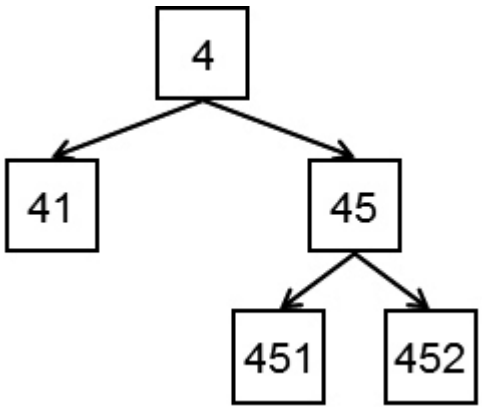


Abb. 4–28 *Kostenstellenhierarchie*

Für die Darstellung der Hierarchiemodellierung kann im Folgenden auf weitere Attribute, wie hier exemplarisch das Attribut *Kostenstellenverantwortlicher*, verzichtet werden. Passend zu der dargestellten Hierarchie ergibt sich der korrespondierende Inhalt der Dimensionstabelle aus Abbildung 4–29.

Kostenstelle_Id	Kostenstelle	Übergeordnete_Kostenstelle_Id
1	4	NULL
2	41	1
3	45	1
4	451	3
5	452	3

Abb. 4–29 *Ausprägungen in der Kostenstellendimensionstabelle*

4.5.1 Iterative Abfrage

Eine der einfachsten Abfragen an diese Dimension hat das Ziel, alle Blattelemente aufzulisten. Diese ergeben sich in der Variante der Parent-Child-Tabelle dadurch, dass ein Blattelement nicht zugleich auch selbst übergeordnetes Element eines anderen Elements sein kann. Für die Kostenstellen des Beispiels sind dies genau diejenigen Kostenstellen, die nicht in der Liste der übergeordneten Kostenstellen auftauchen.

In dem folgenden abstrahierten Skript zur Ermittlung der Blattelemente in einer Dimension ergibt sich diese Teilmenge aus dem Sub-Select.

```
SELECT
  DTKostenstelle.Kostenstelle_Id,
  DTKostenstelle.Kostenstelle,
  DTKostenstelle.Übergeordnete_Kostenstelle_Id
FROM
  DTKostenstelle
WHERE
  DTKostenstelle.Kostenstelle_Id NOT IN
  (  SELECT DTKB.Übergeordnete_Kostenstelle_Id
     FROM DTKostenstelle DTKB
     WHERE DTKB.Übergeordnete_Kostenstelle_Id IS NOT NULL)
```

Die Wirkungsweise des Skripts ist in Abbildung 4–30 verdeutlicht. Die Menge der übergeordneten Kostenstellen ist dort grau schraffiert markiert, es sind die Kostenstellen mit der ID 1 und 3, daher sind nur die Kostenstellen 4 und 45 übergeordnete Kostenstelle irgendeiner anderen Kostenstelle. Diese Kostenstellen sind also auszuschließen, was an der grauen Schraffierung dieser Zeilen in der Spalte Kostenstelle_Id symbolisiert ist. Somit verbleiben die anderen Kostenstellen (41, 451, 452) als Blattelemente. Dies sind gerade die in der Darstellung nicht schraffiert hervorgehobenen Elemente.

Kostenstelle_Id	Kostenstelle	Übergeordnete_Kostenstelle_Id
1	4	NULL
2	41	1
3	45	1
4	451	3
5	452	3

Abb. 4–30 *Blattelemente in der Kostenstellendimension*

Bei dieser Art der Abfrage muss die Parent-Child-Tabelle noch nicht rekursiv mehrfach abgefragt werden. Bei der Frage nach allen Nachfolgern eines Elements, die zugleich auch Blattelement sind, ist dies jedoch erforderlich.

4.5.2 Einstufige Rekursion

Die Abfrage der direkten nachfolgenden Blattelemente, z. B. der Kostenstelle 45, ist jedoch noch einstufig lösbar. Zusätzlich zu dem Sub-Select zur Abfrage der Blattelemente kommt noch ein weiteres Sub-Select für die Abfrage der

Nachfolger. Im Beispiel aus dem folgenden Skript zur Ermittlung der direkten (einstufigen) Nachfolger, die auch Blattelement sind, ist dies das zweite Sub-Select, das auf die Nachfolger des Knotens mit der ID 3, also die Kostenstelle 45, eingrenzt.

```
SELECT
  DTK.Kostenstelle_Id,
  DTK.Kostenstelle,
  DTK.Übergeordnete_Kostenstelle_Id
FROM
  DTKostenstelle DTK
WHERE
  DTK.Kostenstelle_Id NOT IN
  (  SELECT DTKB.Übergeordnete_Kostenstelle_Id
     FROM DTKostenstelle DTKB
     WHERE DTKB.Übergeordnete_Kostenstelle_Id IS NOT NULL)
  AND DTK.Kostenstelle_Id IN
  (  SELECT DTKR1.Kostenstelle_Id
     FROM DTKostenstelle DTKR1
     WHERE DTKR1.Übergeordnete_Kostenstelle_Id = "3")
```

Für diese Abfrage ist die Wirkungsweise gut in Abbildung 4–31 erkennbar. Über die Abfrage des Blattelementes ergibt sich die graue Schraffierung für die Kostenstellen-Id 1 und 3. Über die Eingrenzung auf die übergeordnete Kostenstelle 45, also Id 3, ergeben sich die nicht schraffiert hervorgehobenen Zeilen. Somit verbleiben die Kostenstellen 451 und 452 als direkte Nachfolger von 45, die auch Blatt sind.

Kostenstelle_Id	Kostenstelle	Übergeordnete_Kostenstelle_Id
1	4	NULL
2	41	1
3	45	1
4	451	3
5	452	3

Abb. 4–31 *Nachfolger, die auch Blattelement sind (einstufig)*

Für die direkten Nachfolger von 4 würde die Kostenstelle 45 über die Blattelement-Eigenschaft ausgeschlossen werden. Interessanter und aufwendiger ist jedoch die Frage nach allen Nachfolgern der Kostenstelle 4, also nicht nur den direkten Nachfolgern.

4.5.3 Mehrstufige Rekursion

Das folgende Skript zeigt im Fall der zweistufigen Rekursivität die Menge aller nachfolgenden Blattelemente der Kostenstelle 4 mit der Id 1. Bei Hierarchien mit mehreren Stufen sind demzufolge weitere rekursive Schritte hinzuzufügen.

```
SELECT
  DTK.Kostenstelle_Id,
  DTK.Kostenstelle,
  DTK.Übergeordnete_Kostenstelle_Id
FROM
  DTKostenstelle DTK
WHERE
  DTK.Kostenstelle_Id NOT IN
  (  SELECT DTKB.Übergeordnete_Kostenstelle_Id
     FROM DTKostenstelle DTKB
     WHERE DTKB.Übergeordnete_Kostenstelle_Id IS NOT NULL)
  AND DTK.Kostenstelle_Id IN
  (  SELECT DTKR1.Kostenstelle_Id
     FROM DTKostenstelle DTKR1
     WHERE DTKR1.Übergeordnete_Kostenstelle_Id = "1"
     UNION SELECT DTKR2.Kostenstelle_Id
     FROM DTKostenstelle DTKR2
     WHERE DTKR2.Übergeordnete_Kostenstelle_Id IN
     (  SELECT DTKR1.Kostenstelle_Id
        FROM DTKostenstelle DTKR1
        WHERE DTKR1.Übergeordnete_Kostenstelle_Id = "1")
```

Durch das erste Sub-Select erfolgt wieder die Eingrenzung auf die Blattelemente, verdeutlicht in Abbildung 4–32 an den schraffiert markierten Einträgen ungleich NULL in der Spalte Übergeordnete_Kostenstelle_Id. Die direkten Nachfolger sind die mit der übergeordneten Kostenstelle 4. Da aber die Kostenstelle 45 kein Blattelement ist, ist nur die 41 nicht schraffiert markiert. In einem zweiten Schritt müssen nun jedoch die Nachfolger der Nachfolger abgefragt werden, also in diesem Fall die Nachfolger der Kostenstelle 45, wieder nicht schraffiert markiert in der Abbildung. Dadurch sind auch die Kostenstellen 451 und 452 in der Ergebnismenge enthalten.

Kostenstelle_Id	Kostenstelle	Übergeordnete_Kostenstelle_Id
1	4	NULL
2	41	1
3	45	1
4	451	3
5	452	3

Abb. 4–32 *Nachfolger, die auch Blattelement sind (zweistufig)*

An dem Muster der Abfrage ist erkennbar, dass mit zunehmender Anzahl an Hierarchiestufen dieses Skript nicht nur komplexer wird, sondern es auch starr hinsichtlich der maximalen Tiefe ausgeprägt ist. Echte rekursive Abfragen haben in dieser Hinsicht deutliche Vorteile.

4.5.4 Rekursives SQL

Grundprinzip dieser Art der Abfragen in rekursiven SQL-Queries ist es, zu allen gefundenen Nachfolgern sukzessive alle weiteren Nachfolger zu finden und damit eine vollständige Liste aller sogenannten Sprungziele, wie in Abbildung 4–33 dargestellt, zu erzeugen. Ein Sprungziel ist in diesem Zusammenhang ein möglicher Pfad von einem Knoten der Hierarchie zu einem anderen Knoten der Hierarchie.

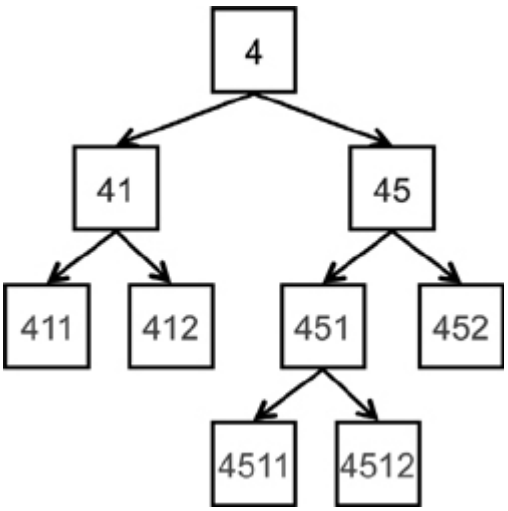


Abb. 4–33 *Sprungziele als Grundlage für rekursive Abfragen*

Die ANSI-Spezifikation für »Recursive SQL« gibt dabei eine Möglichkeit an, diese Abfragen zu formulieren. Das Grundmuster dazu ist in folgendem Skript dargelegt:

```
WITH RECURSIVE [temp table] [column list] AS
(
   [seed statement]
   UNION ALL
   [recursive statement]
)
[select statement]
```

Durch die *With-Klausel* erfolgt die Definition einer temporären Tabelle, deren Aufbau sich rekursiv ergibt. Das *Seed-Statement* legt zunächst den Startpunkt fest, definiert also die Ausgangsmenge an Elementen in der Form, wie es die

Struktur der temporären Tabelle erwartet. Dadurch entstehen in dieser Tabelle Einträge, auf deren Basis dann weitere Mengen rekursiv hinzugefügt werden. Dazu muss das *Recursive-Statement* auf die bereits erzeugten Einträge in der temporären Tabelle zugreifen und auf deren Basis rekursiv neue Einträge generieren, bis die Rekursion zu einem Ende kommt. Erst das *Select-Statement* führt dann die tatsächliche Auswahl relevanter Sätze aus der aufgebauten temporären Tabelle durch.

Dieser Mechanismus soll am Beispiel der Kostenstellenhierarchie verdeutlicht werden. Das folgende Skript zur Abfrage aller Nachfolger über rekursives SQL dient dabei der Bestimmung aller Nachfolger der Kostenstelle 4. Die Struktur der temporären Tabelle entspricht inhaltlich der Liste aller Sprungziele, dient also der Darstellung der Menge aller möglichen Pfade in unserer Kostenstellenhierarchie.

```
WITH RECURSIVE TTErreichbar
(Von_Kostenstelle_Id, Nach_Kostenstelle_Id, Anzahl) AS
(
  SELECT
     D1.Übergeordnete_Kostenstelle_Id AS Von_Kostenstelle_Id,
     D1.Kostenstelle_Id AS Nach_Kostenstelle_Id,
     1
  FROM
     DTKostenstelle D1
  UNION ALL
  SELECT
     TT.Von_Kostenstelle_Id,
     D2.Kostenstelle_Id AS Nach_Kostenstelle_Id,
     TT.Anzahl+1
  FROM
     TTErreichbar TT, DTKostenstelle D2
  WHERE TT.Nach_Kostenstelle_Id = D2.Übergeordnete_Kostenstelle_Id
)
SELECT Von_Kostenstelle_Id, Nach_Kostenstelle_Id, Anzahl
FROM TTErreichbar T, DTKostenstelle D
WHERE T.Von_Kostenstelle_Id = D.Übergeordnete_Kostenstelle_Id
AND D.Übergeordnete_Kostenstelle_Id="1"
```

Über das S*eed-Statement* wird die »Saat« gesetzt, also die direkten Nachfolger aller Elemente. Diese sind die in Abbildung 4–33 dargestellten Sprungziele, die über eine Kante oder über mehrere Kanten erreichbar sind. Von Kostenstelle 4 kommt man mit einer Kante zu den Kostenstellen 41 und 45, von der Kostenstelle 45 gelangt man über eine Kante zu den Kostenstellen 451 und 452. Die temporäre Tabelle hat damit durch die Saat die Einträge wie in Abbildung 4–34 dargestellt.

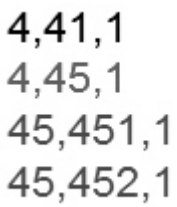


Abb. 4–34 *Temporäre Tabelle »Seed«*

Durch das sogenannte *Recursive-Statement* ergeben sich die mehrstufigen Sprungziele. In unserem Beispiel gelangt man von Kostenstelle 4 mit zwei Kanten zu den Kostenstellen 451 und 452. Zusammen ergeben sich damit die Kostenstellen 41, 45, 451 und 452 als Nachfolger von Kostenstelle 4 (siehe Abb. 4–35).

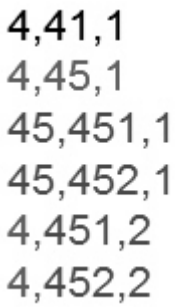


Abb. 4–35 *Temporäre Tabelle nach der Rekursion«*

Der untere Teil des Skripts, das *Select-Statement*, stellt nun die eigentliche Abfrage dar. In unserem Beispiel ist das die Menge aller Kostenstellen, die Nachfolger der Kostenstelle 4 sind. Die letzte Zeile gibt hier den Filter auf genau dieses Element, um dessen Nachfolger es in der Abfrage geht.

Neben dieser Variante gibt es auch datenbankspezifische Syntaxvarianten, so etwa das in Oracle implementierte Statement CONNECT BY PRIOR, exemplarisch dargestellt in dem folgenden Skript.

```
SELECT Kostenstelle_Id
FROM DTKostenstelle
START WITH Übergeordnete_Kostenstelle_Id IS NULL
CONNECT BY PRIOR Kostenstelle_Id = Übergeordnete_Kostenstelle_Id
```

Hier wird die Saat durch die *Start-With-Klausel* gesetzt und die Rekursion erfolgt über die Angabe der Verbindung in der *Connect-By-Prior-Klausel*. Den Startpunkt definiert die Kostenstelle, die keine übergeordnete Kostenstelle hat und damit die bzw. eine Wurzel ist. Durch die Rekursion ergeben sich dann über die Beziehung »Übergeordnete Kostenstelle« alle weiteren Nachfolger.

Einerseits bieten rekursive Abfragen umfassende Möglichkeiten der Abfragen an Parent-Child-basierte Hierarchien in Dimensionen, andererseits unterstützen jedoch nicht alle BI-Werkzeuge diese Möglichkeiten. Daher wird oftmals nach weiteren Alternativen zur Implementierung etwa in Form flachgeklopfter Tabellen, wie in Abbildung 4–36 dargestellt, gefragt. Diese Tabelle hat eine Spalte für das Element, eine Spalte für das Wurzelelement und jeweils eine Spalte für jede Stufe dazwischen bis zur maximalen Tiefe der Hierarchie. Da hierbei alle Elemente bebuchbar sind, reicht es nicht, nur die Blattelemente in die Tabelle aufzunehmen. Zu jedem Element der Dimension gibt es somit einen Eintrag in der Dimensionstabelle.

Kostenstelle-ID	Kostenstelle	Level 1	Level 2	Wurzel	Blattelement-Flag
1	4	4	4	4	
2	41	4	4	4	
3	45	4	4	4	
6	411	41	41	4	JA
7	412	41	41	4	JA
4	451	45	45	4	
5	452	45	45	4	JA
8	4511	451	45	4	JA
9	4512	451	45	4	JA

Abb. 4–36 *Auffüllen unbalancierter Hierarchien*

Neben den vielen redundant aufgefüllten Spalten mit der damit einhergehenden möglichen Dateninkonsistenz stellt auch die fehlende Flexibilität hinsichtlich der Hierarchietiefe einen Kritikpunkt bei dieser Form der Abbildung rekursiver Hierarchien dar. In den Fällen, in denen nur die Blattelemente bebuchbar sind, ist das Auffüllen eine oft anzutreffende Art der

Handhabung von nicht balancierten Strukturen, die es insbesondere den BI-Werkzeugen erleichtert, mit diesen Strukturen umzugehen.

4.5.5 Brückentabellen

Ein weiterer beliebter Ansatz zur Implementierung rekursiv definierter unbalancierter Hierarchien ist der Einsatz von Brückentabellen, die im Grundprinzip nichts anderes als persistierte Listen von Sprungzielen darstellen. Zur Verdeutlichung des Verfahrens dient das erweiterte Modell aus Abbildung 4–37.

In den Brückentabellen erfolgt die Speicherung aller möglichen Teilwege in der Hierarchie verbunden mit der Weglänge und der Information, ob das Ende des Weges bereits ein Blatt ist. Im Folgenden dient wieder die Kostenstellenhierarchie der Illustration.

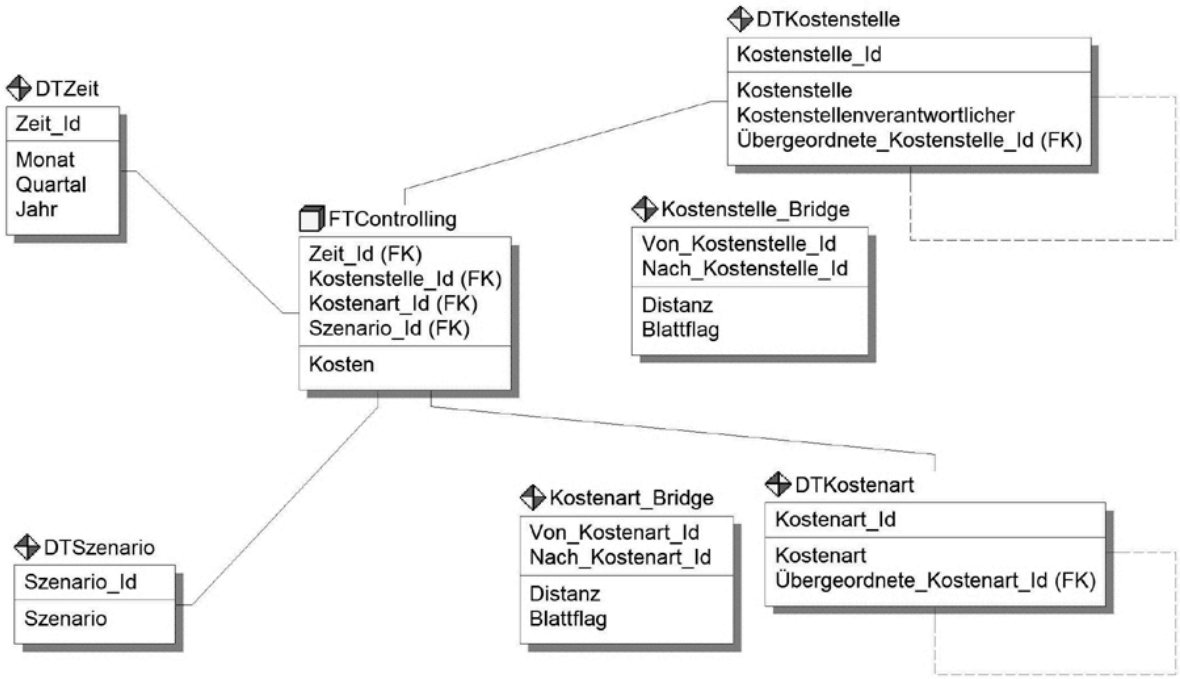


Abb. 4–37 *Brückentabellen zum Traversieren von rekursiven Hierarchien*

Für diese Hierarchie gibt es zu jedem Knoten und zu jedem möglichen Pfad von diesem Knoten aus einen Eintrag in der Kostenstellen-Brückentabelle in Abbildung 4–38. Die Verwendung der Kostenstellen anstelle der Schlüssel aus der Dimensionstabelle dient dabei nur der Illustration. Die farbliche Markierung korrespondiert in beiden Abbildungen.

Von Kostenstelle	Nach Kostenstelle	Anzahl Ebenen/Distanz	Blattelement-Flag
4	4	0	
4	41	1	
4	45	1	
4	411	2	JA
4	412	2	JA
4	451	2	
4	452	2	JA
4	4511	3	JA
4	4512	3	JA
41	41	0	
41	411	1	JA
41	412	1	JA
411	411	0	JA
412	412	0	JA
45	45	0	
45	451	1	
45	452	1	JA
45	4511	2	JA

45	4512	2	JA
451	451	0	
451	4511	1	JA
451	4512	1	JA
452	452	0	JA
4511	4511	0	JA
4512	4512	0	JA

Abb. 4–38 *Brückentabelle der Kostenstellenhierarchie*

Um nun beispielsweise alle Nachfolger inklusive der Teilbaumwurzel selbst zu einem Element zu erhalten, ist in der Brückentabelle ein Filter auf die Spalte *Von Kostenstelle* anzuwenden. Durch Filterung auf das Element 45 ergibt sich etwa die Menge 45, 451, 452, 4511 und 4512.

Für die Abfrage aller Nachfolger ohne die Teilbaumwurzel selbst muss zusätzlich die Spalte *Distanz* noch auf größer null gefiltert werden und es ergeben sich die Kostenstellen 451, 452, 4511 und 4512 als Ergebnis für die Abfrage mit der gleichen Teilbaumwurzel 45.

In gleicher Weise ergeben sich die direkten Nachfolger von Kostenstelle 45 über den Filter auf die Spalte *Distanz* gleich eins mit der Ergebnismenge 451 und 452.

Die Menge der Nachfolger, die auch Blattelement sind, ergibt sich aus der Kombination der Filterung auf die Teilbaumwurzel *Von Kostenstelle* sowie *Blattelement-Flag* gleich Ja.

Für die Verknüpfung der Faktentabelle, der Brückentabelle und der Dimensionstabelle gibt es im Wesentlichen drei Möglichkeiten. Sind nur die direkt auf der Kostenstelle gebuchten Posten relevant, erfolgt der Join der Dimensionstabelle direkt an die Faktentabelle. Sollen auch die Kosten von darunter liegenden Kostenstellen (Nachfolger) berücksichtigt werden, erfolgt der Join der Dimensionstabelle an die Brückentabelle als *Von Kostenstelle*, sodass alle von dort erreichbaren Kostenstellen in den Join mit der

Faktentabelle eingehen. Die dritte Variante schaut dabei nicht nach unten, sondern in der Hierarchie nach oben. Alle drei Formen der Abfrage sind in Abbildung 4–39 zusammengefasst.

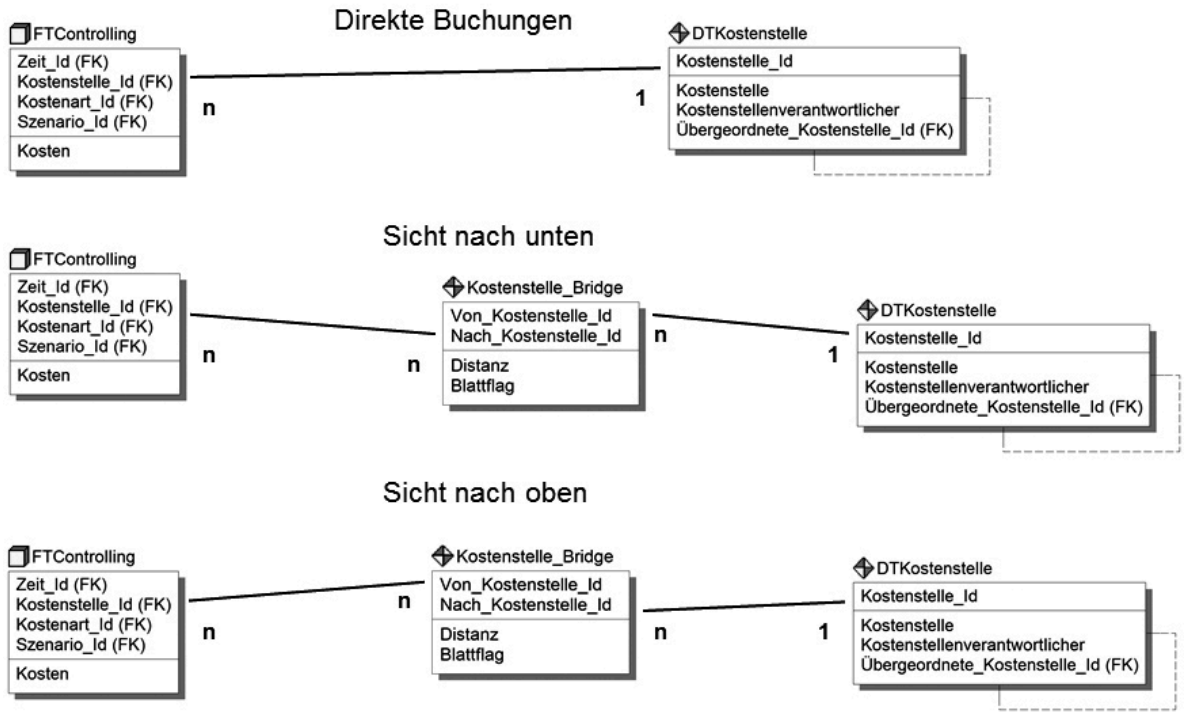


Abb. 4–39 *Drei Join-Varianten mit Brückentabellen*

Für einen exemplarischen Kostenstellen-Summenbericht, in dem für die Berechnung eines Summenwertes auch alle darunterliegenden Kostenstellen in den Summenknoten mit einfließen, kann das folgende Skript herangezogen werden. Dieses implementiert die Sicht nach unten in dem Join.

```
SELECT
  DK.Kostenstelle,
  SUM( FT.Kosten ) AS Gesamtkosten
FROM
  DTZeit DZ,
  DTKostenstelle DK,
  Kostenstelle_Bridge B1,
  FTControlling FT
WHERE
  DZ.Monat = "01" AND
  DZ.Jahr = "2012" AND
  FT.Zeit_Id = DZ.Zeit_Id AND
  DK.Kostenstelle_Id = B1.Von_Kostenstelle_Id AND
  FT.Kostenstelle_Id = B1.Nach_Kostenstelle_Id
GROUP BY
  DK.Kostenstelle,
ORDER BY
  DK.Kostenstelle
```

Auch Brückentabellen sind nicht mit jedem BI-Werkzeug kompatibel, sodass die effiziente Implementierung von rekursiven Hierarchien im Einzelfall abzuwägen ist.

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

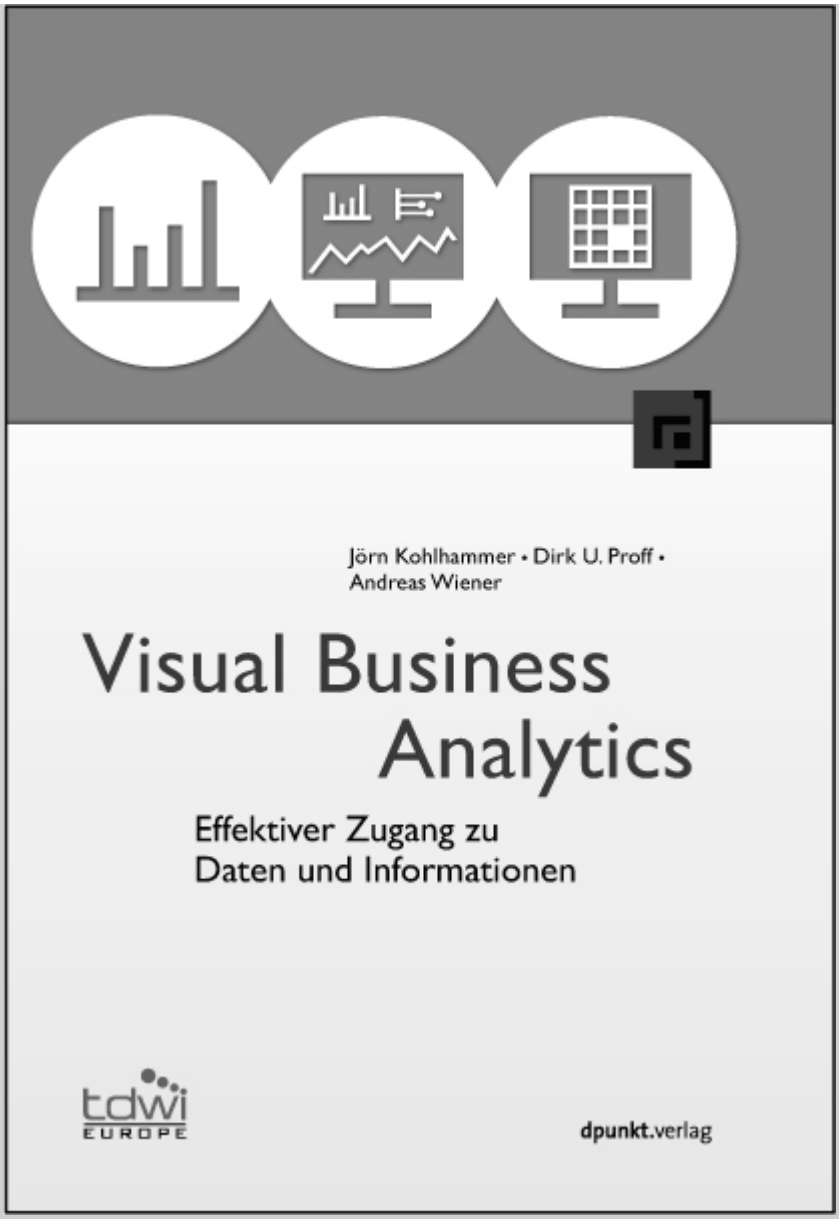
T

U

V

W

Z



2013, 232 Seiten,

komplett in Farbe, Festeinband

€ 69,90 (D)

ISBN 978-3-86490-044-0

Jörn Kohlhammer · Dirk U. Proff · Andreas Wiener

Visual Business Analytics

Effektiver Zugang zu Daten und Informationen

Business-Intelligence-Lösungen sind für Unternehmen unabdingbar, um Datenmengen in vertretbarer Zeit zu analysieren und daraus resultierend Entscheidungen zu treffen.

Dieses Buch zeigt den Weg auf, wie aus Daten über das Mittel der Visualisierung für den Empfänger entscheidungsrelevante Informationen werden. Ebenso gibt es einen Überblick, welche Darstellungsformen geeignet sind, um komplexe Zusammenhänge abzubilden, wie Unternehmen Visual Business Analytics erfolgreich nutzen können und welche zukünftigen Möglichkeiten sich durch interaktive Darstellungen ergeben.

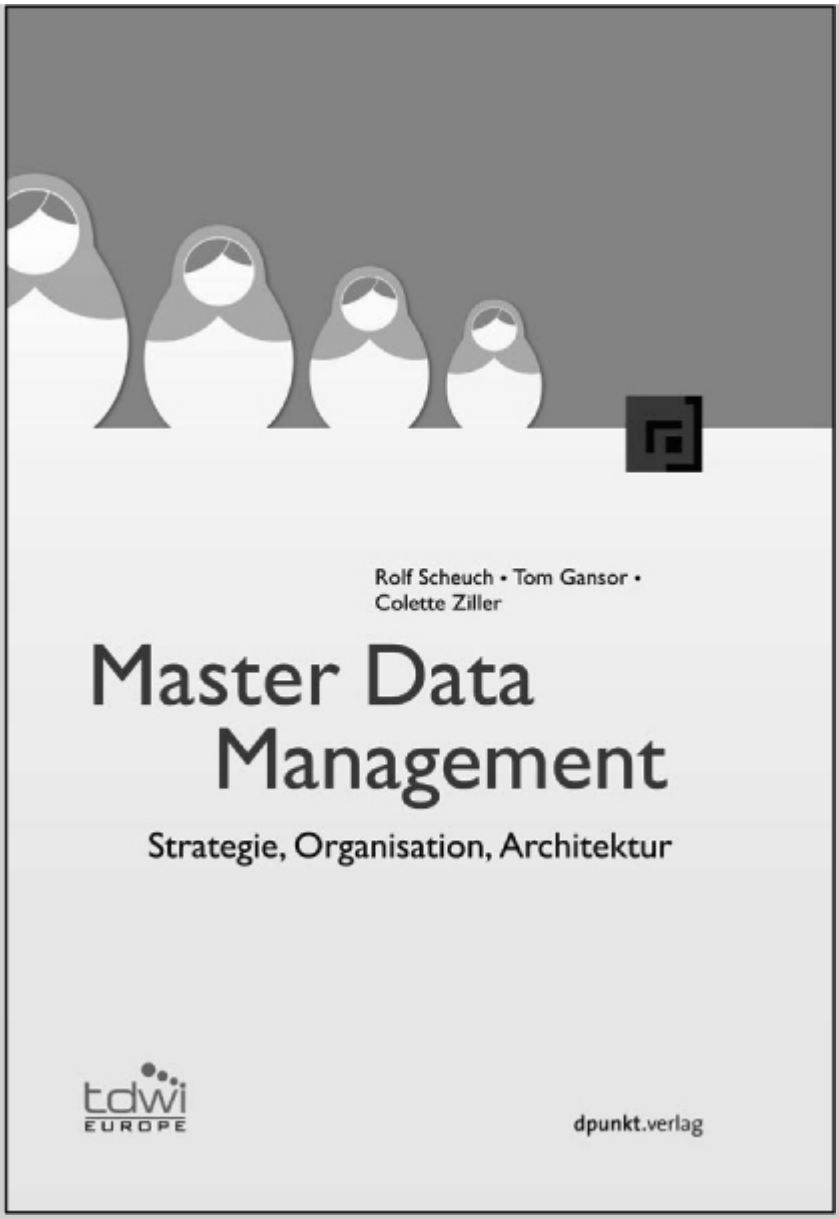
Wieblinger Weg 17 · 69123 Heidelberg

fon 0 62 21/14 83 40

fax 0 62 21/14 83 99

e-mail hallo@dpunkt.de

http://www.dpunkt.de



2012, 492 Seiten, gebunden

€ 79,90 (D)

ISBN 978-3-89864-823-3

Rolf Scheuch · Tom Gansor · Colette Ziller

Master Data Management

Strategie, Organisation, Architektur

Edition TDWI

Unternehmensdaten und deren Qualität und Verfügbarkeit werden mehr und mehr zu einem kritischen Erfolgsfaktor. Master Data Management (MDM) sorgt durch die strukturierte Bewirtschaftung und Qualitätssicherung für Orientierung und Übersicht im Datendschungel. Dieses Buch beschreibt MDM aus betriebswirtschaftlicher und technischer Sicht. Der Nutzen, das Einsatzgebiet und die Positionierung werden analysiert, um die Planung, Konzeption und Umsetzung solcher Lösungen zu realisieren. Auch auf die verschiedenen heute gängigen Ansätze mit ihren jeweiligen Stärken und Schwächen wird eingegangen.

Wieblinger Weg 17 · 69123 Heidelberg

fon 0 62 21/14 83 40

fax 0 62 21/14 83 99

e-mail hallo@dpunkt.de

http://www.dpunkt.de