

## 12 Datenquellen/Datenziele in der Planung

Die Planung zeichnet sich im Vergleich zum Berichtswesen dadurch aus, dass Daten nicht nur lesend ausgewertet, sondern auch verändert und zurückgeschrieben werden. Für diese schreibenden Zugriffe gibt es im BW mit den Real-time InfoCubes spezielle Datenziele. Auch die Datenbereitstellung für die Planung durch InfoProvider weist einige Besonderheiten auf und erfordert daher andere Konzepte in der Datenmodellierung.

- *InfoProvider in der Planung*
  - *MultiProvider in der Planung*
  - *Real-time InfoCube in der Planung*

*Themen der Datenquellen  
und Datenziele in der  
Planung*

### Datenquellen/Datenziele in der Planung

### Szenario

Philipp Stein steht vor dem Kaffeeautomat des 1. FC Aggregat Konstanz. Die IT-Leiterin Konstanze Zuse kommt gerade den Flur entlang und nutzt diese Gelegenheit, weitere Informationen mitzunehmen: »Herr Stein, nachher erzählen Sie uns zwar noch alle Details über die »Planungsgrundlagen«, die »Manuelle Planung« und die »Automatisierte Planung«, aber eine Frage habe ich noch zuvor. Die BW Integrierte Planung ist stark mit dem Backend-Bereich verbunden. Hat damit die BW Integrierte Planung die gleichen Datenhaltungsobjekte?«, möchte Konstanze Zuse wissen. »Erst mal zu den **InfoProvidern in der Planung**«, beginnt Philipp Stein. »Als Datenziel, d.h. um die Daten zu speichern, dient ein spezieller InfoCube, der **Real-time InfoCube**. Alle anderen Datenziele wie der Standard-InfoCube, dienen rein als Datenquelle, um beispielsweise zusätzlich die Istdaten als Referenz in der Planungsoberfläche anzuzeigen. Ich empfehle, die genutzten Datenziele beziehungsweise -quellen in jedem Fall in einen **MultiProvider** einzubinden. **In der Planung** wie in der Datenanalyse wartet ein MultiProvider mit einigen Vorteilen auf. Zum einen sind durch ihn nach der Entwicklung einer Planungs- oder Analyseanwendung bei späteren

Änderungswünschen weitere Datenziele flexibel hinzuschaltbar, zum Beispiel weitere Standard-InfoCubes, DataStore-Objekte oder Real-time InfoCubes. Zum anderen lassen sich durch ihn Daten aus mehreren Datenzielen verbinden. MultiProvider sollten daher die Grundlage nicht nur für Analyseanwendungen, sondern auch für jedes Planmodell sein.« »Gut, gut«, sagt Konstanze Zuse und fügt hinzu: »Dann bin ich jetzt auf die Planungsgrundlagen gespannt. «

### InfoProvider in der Planung

Die Grundlage einer *Planungsanwendung* bilden InfoProvider, da sie der Anwendung Ist- und Plandaten zur Verfügung stellen und neue Daten oder Änderungen sichern. Die InfoProvider lassen sich in der *Data Warehousing Workbench (DWWB)* anlegen.

In einer Planungsanwendung kann auf folgende InfoProvider *lesend* zugegriffen werden:

- *Standard-InfoCube*
- *DataStore-Objekt*
- *VirtualProvider*
- *Merkmal*

Dieser Zugriff ist jedoch nur mit Hilfe eines *MultiProviders* möglich, in den die genannten InfoProvider zuvor eingebunden werden müssen. Auch ein *InfoSet*, das selbst keine Daten speichert, kann über einen MultiProvider in einer Planungsanwendung zum Einsatz kommen. Der MultiProvider bietet damit die Möglichkeit, in einer Planungsanwendung auf mehrere Datenquellen und Datenziele gleichzeitig zuzugreifen.

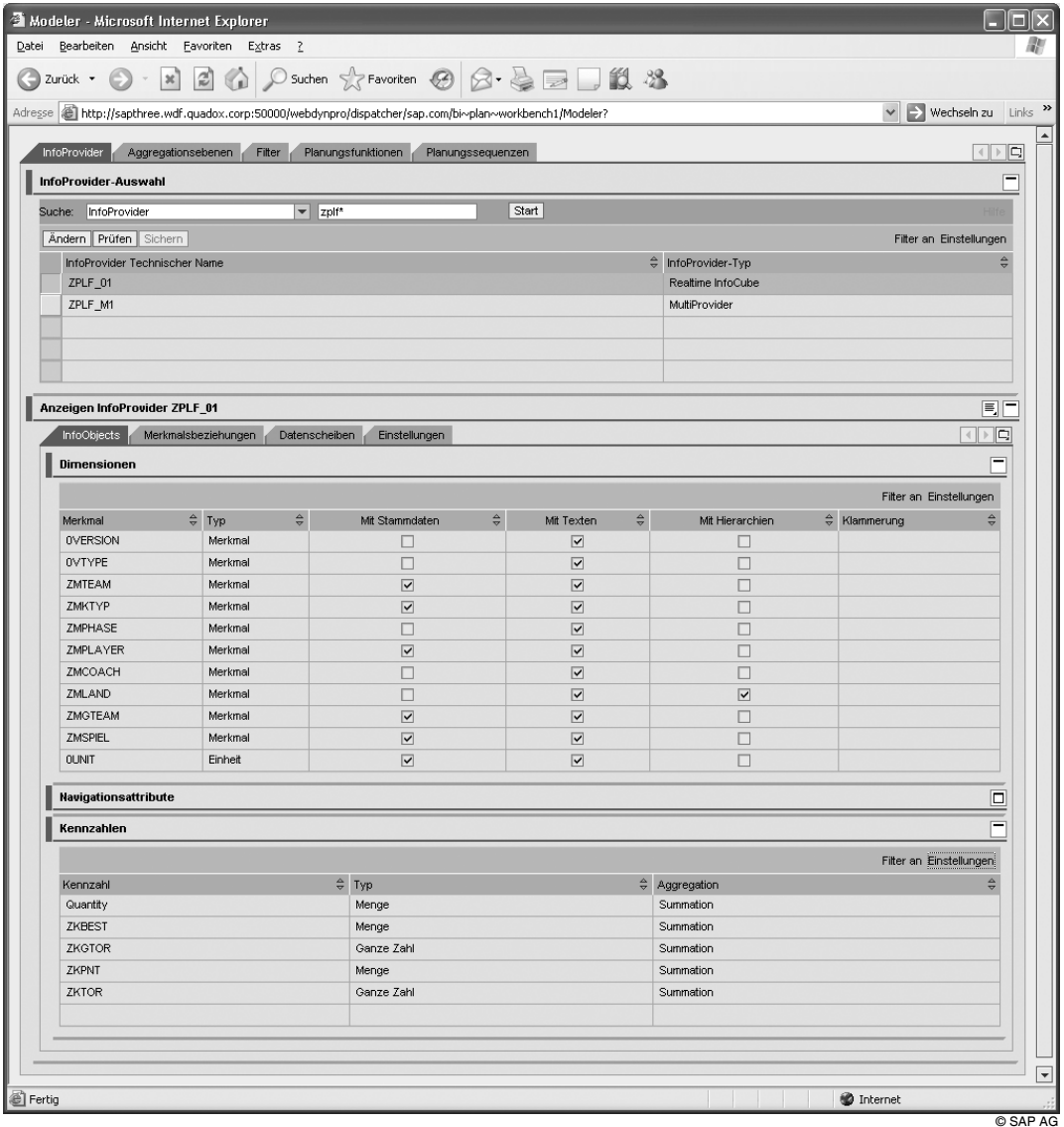
Plandaten, also Eingaben von Anwendern in *eingabebereiten Queries* oder Ergebnisse von *Planungsfunktionen*, können dagegen nur in eine Art von Datenziel *geschrieben* und gesichert werden, nämlich in

- *Real-time InfoCubes*.

MultiProvider und Real-time InfoCubes stellen für eine Planungsanwendung die wesentlichsten Arten von InfoProvidern dar. Dies spiegelt sich auch bei der Erstellung eines *Planmodells* wider, da der erste Schritt darin besteht, *Real-time InfoCube* oder *MultiProvider* für die Definition einer oder mehrerer *Aggregationsebenen* auszuwählen (siehe Abb. 12–1). Daraufhin wird der gewählte InfoProvider mit seinen Dimensionen, den darin enthaltenen Merkmalen und ihren Navigationsattributen sowie mit seinen Kennzahlen dargestellt. Für Real-

time InfoProvider, in die Plandaten zurückgeschrieben werden, lassen sich darüber hinaus Einstellungen zu *Merkmalsbeziehungen*, *Datenscheiben* und zum *Standardstichtag* vornehmen.

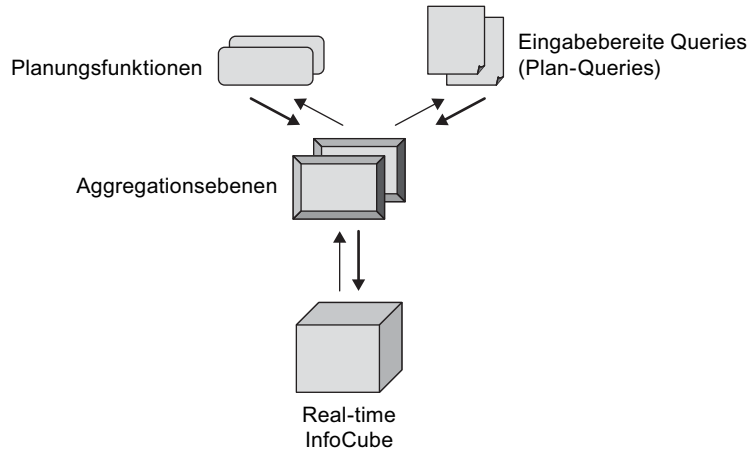
**Abb. 12-1**  
InfoProvider-Dialog im  
Planning Modeler



Im Folgenden sind mögliche Szenarien für die Wahl der InfoProvider und den Aufbau einer Planungsanwendung beschrieben. Es werden kurz Vor- und Nachteile beziehungsweise Eignung erläutert. Das favorisierte Modell bildet den Abschluss.

*Modellierungsszenarien*

**Abb. 12-2**  
 Szenario 1:  
 Einfache  
 Aggregationsebene



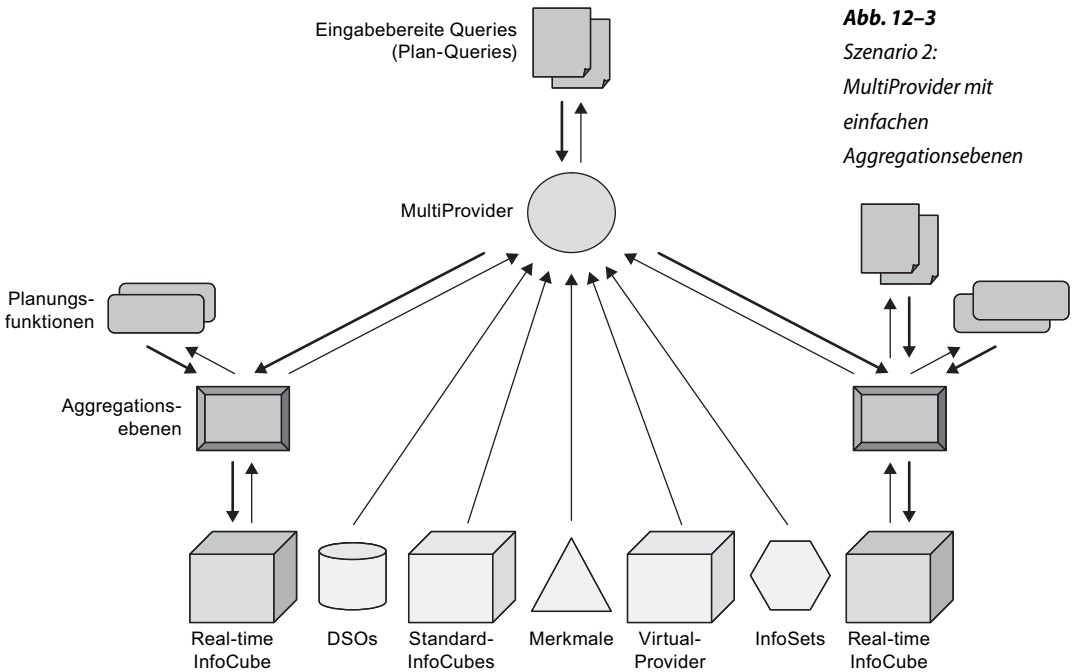
### Szenario 1

Eine Aggregationsebene wird (Abb. 12-2 verdeutlicht) direkt auf einem Real-time InfoCube angelegt und als einfache Aggregationsebene bezeichnet. Auf dieser Aggregationsebene wiederum werden Planungsobjekte wie Planungsfunktionen und eingabebereite Queries definiert.

Ein solches Modell erlaubt den Zugriff auf Daten nur eines InfoProviders. Es ist kein Austausch mit anderen Datenzielen oder Datenquellen möglich. Eine spätere Aufnahme weiterer InfoProvider in die Anwendung bedarf eines vollständigen Redesigns. Daher richtet sich ein solches Modell an kleine Planungsanwendungen mit geringem Datenvolumen und geringer Komplexität. Für Testzwecke ist es ebenso geeignet, da es schnell aufgebaut ist.

### Szenario 2

Wie im ersten Szenario geschildert, befinden sich Aggregationsebenen auf Real-time InfoCubes (einfache Aggregationsebenen) und auf ihnen Planungsobjekte wie Planungsfunktionen und eingabebereite Queries. Diese Aggregationsebenen werden in Szenario 2 zusätzlich in einen MultiProvider aufgenommen, siehe Abbildung 12-3. Objekte auf diesen einfachen Aggregationsebenen arbeiten zwar nach wie vor nur auf dem ihnen jeweils zugrunde liegenden Real-time InfoCube. Auf dem MultiProvider ist es nun aber möglich, eingabebereite Queries anzulegen, um Daten aus den eingebundenen InfoProvidern (Aggregationsebenen, Real-time InfoCubes, Standard-InfoCubes, DataStore-Objekten, VirtualProvidern, InfoSets und Merkmalen) auszugeben und Daten in die Real-time InfoCubes mit einfachen Aggregationsebenen auch schreiben zu können.



**Abb. 12-3**  
 Szenario 2:  
 MultiProvider mit  
 einfachen  
 Aggregationsebenen

Eine eingabebereite Query kann direkt auf einem MultiProvider aufsetzen, wenn dieser mindestens einen Real-time InfoCube enthält, auf dem eine Aggregationsebene angelegt ist. Bei einer Planungsfunktion ist dies nicht möglich; sie muss direkt auf Basis einer Aggregationsebene definiert werden, unabhängig davon, ob sie auf einem MultiProvider oder auf einem Real-time InfoCube basiert.

In einem solchen Szenario sind Aggregationsebenen nicht auf dem MultiProvider anlegbar, da bereits Aggregationsebenen auf tieferer Stufe existieren und eine Schachtelung von Aggregationsebenen nicht möglich ist. Daher können Planungsfunktionen nur auf einfachen Aggregationsebenen basieren, die direkt auf den Real-time InfoCubes aufsetzen. Somit sind sie nicht InfoProvider-übergreifend ausführbar.

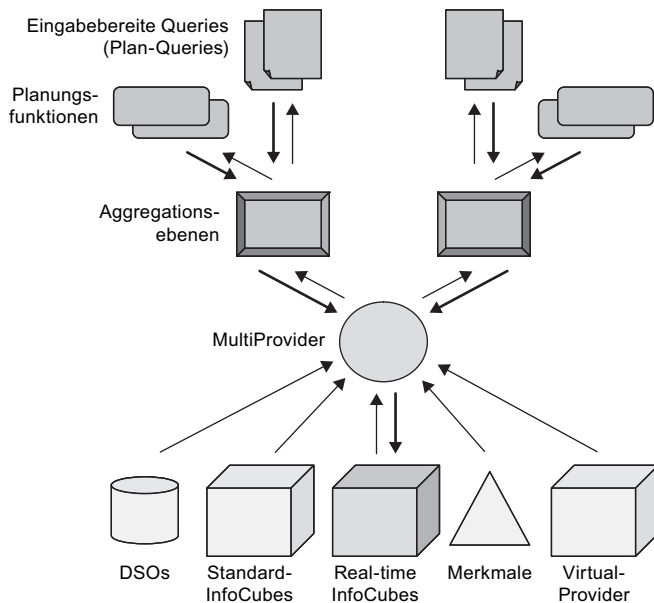
Ein weiterer Nachteil dieser Variante ist die potenzielle Entwicklung von Anwendungsobjekten wie Planungsfunktionen, *Filtern* und eingabebereiten Queries auf unterschiedlichen Ebenen (mal »unterhalb«, mal »oberhalb« von MultiProvidern, wie in Abb. 12-3 aufgezeigt). Dies verringert einerseits die Übersichtlichkeit des *Planmodells* für Entwickler und Administratoren und die Möglichkeit der Wiederverwendbarkeit von Objekten (zum Beispiel Filter) andererseits.

Benötigt eine Planungsanwendung eine derart komplexe Landschaft, die mehrere Datenziele und Datenquellen einbezieht, so ist Szenario 2 nicht zu empfehlen, da die Nachteile überwiegen.

### Szenario 3

InfoProvider-übergreifendes Arbeiten ist in Szenario 3 allen Planungsobjekten möglich (siehe Abb. 12–4). Ein MultiProvider bildet dabei die zentrale Schnittstelle zwischen Datenhaltung und Anwendungsobjekten. Auf ihm werden alle Aggregationsebenen angelegt und auf diesen die Planungsfunktionen und eingabebereiten Queries. Aggregationsebenen auf MultiProvidern werden komplexe Aggregationsebenen genannt.

**Abb. 12–4**  
Szenario 3:  
Komplexe  
Aggregationsebenen  
auf MultiProvidern



Szenario 3 ist der empfohlene Modellierungsansatz aus den im Folgenden genannten Gründen:

- Flexibilität bezüglich zukünftiger Anpassungen der Planungsanwendung, indem auf einfache Art und Weise InfoProvider hinzugeschaltet oder entfernt werden können. Eingabebereite Queries und Planungsfunktionen auf einer komplexen Aggregationsebene müssen nur minimal geändert statt völlig neu aufgebaut werden.
- Durchgängiger Entwicklungsansatz:  
Alle Objekte von Planungs- und Berichtsanwendungen befinden sich »oberhalb« des MultiProviders.
- Lesender und schreibender Zugriff auf Daten, die in mehreren InfoProvidern abgelegt beziehungsweise zu speichern sind. Daraus ergibt sich die Möglichkeit, Daten verteilt zu halten, zum Beispiel Istdaten oder historische Daten früherer Planversionen von aktuel-

len Plandaten getrennt vorzuhalten. Eine Aufteilung nach logischen Teilbereichen der Planungsanwendung ist ebenfalls denkbar.

### InfoProvider in der Planung → MultiProvider in der Planung

Während *Real-time InfoCubes* die einzigen Datenziele sind, in denen die in Planungsanwendungen erzeugten Daten gespeichert werden, bieten ausschließlich MultiProvider die Möglichkeit, unterschiedliche Datenziele, die Daten enthalten oder auf Daten referenzieren, für die Planung zu verbinden. Zu diesen Datenzielen zählen *Real-time InfoCubes*, *Standard-InfoCubes*, *DataStore-Objekte*, *Merkmale* sowie *VirtualProvider*. Das Modellierungsszenario 3 der Abbildung 12–4 verdeutlicht dies.

Um in einer Planungsanwendung datenzielübergreifend arbeiten zu können, werden *Aggregationsebenen* auf MultiProvidern angelegt (siehe Abb. 12–4) und dabei als komplexe Aggregationsebenen bezeichnet. Sind auf ihnen *eingabebereite Queries* und *Planungsfunktionen* definiert, so können diese InfoProvider-übergreifend arbeiten, und zwar mit der gleichen Datenstruktur beziehungsweise Granularität von Daten. Dies vereinfacht die Implementierung der Planungsanwendung und vermeidet Fehler. Ist eine andere Granularität gewünscht, so lässt sich eine weitere aggregiertere oder detailliertere Aggregationsebene anlegen, die weniger beziehungsweise mehr Merkmale aus dem MultiProvider enthält.

Wird ein MultiProvider in der Planung verwendet, erscheint zusätzlich das technische Merkmal InfoProvider (0INFOPROV) in der Definitionsmaske der Planungsobjekte. In einer eingabebereiten Query oder in einer Planungsfunktion muss dieses Merkmal eingeschränkt werden. Somit ist eindeutig, welche InfoProvider für das Lesen und Schreiben von Daten angesprochen werden.

Nicht nur für Berichts- sondern auch für Planungsanwendungen ist es sinnvoll, MultiProvider zu nutzen und darauf Aggregationsebenen anzulegen. Die Vorteile sind nachfolgend aufgezeigt. Auf eine Modellierung schlanker statt universeller MultiProvider über mehrere inhomogene InfoProvider ist dabei zu achten. Da für jeden Datensatz geprüft wird, in welchem Datenziel er gespeichert wird, wirken sich viele am MultiProvider beteiligte Datenziele negativ auf die Performance aus.

Dem Aufwand, zusätzlich einen MultiProvider in der *Data Warehousing Workbench (DWWB)* anzulegen, stehen die unter *InfoProvider in der Planung* genannten Vorteile gegenüber.

Anwendungsbeispiel  
MultiProvider

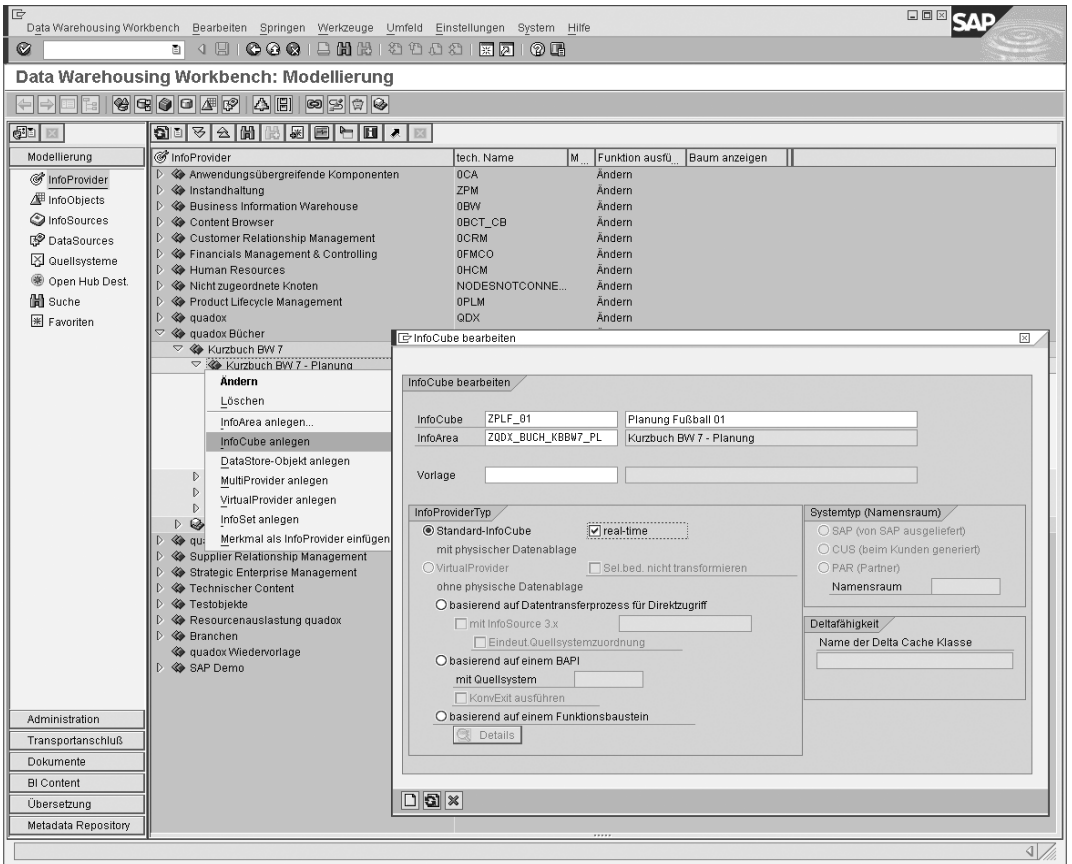
Ein Anwendungsbeispiel für die Nutzung eines MultiProviders in der Planung ist das Kopieren von Istdaten von einem Standard-InfoCube in einen oder mehrere Real-time InfoCubes mittels Planungsfunktion, um sie in der Planung als Ausgangsbasis zu verwenden und durch manuelle Eingaben mit Hilfe eingabebereiter Queries oder durch Berechnungen in Planungsfunktionen und *Planungssequenzen* zu verändern. Dies hat den Vorteil, dass Real-time InfoCubes des MultiProviders ausschließlich für Planungsprozesse zur Verfügung stehen und nicht zwischen Beplanung und Beladung umgeschaltet werden müssen. Istdaten in Standard-InfoCubes sind darüber hinaus vor Änderungen durch manuelle Eingaben oder Planungsfunktionen geschützt.

Allen Vorteilen einer solchen Modellierung steht ein wesentlicher Nachteil gegenüber: Im Vergleich zu einer Beladung von Real-time InfoCubes mit Istdaten schneidet die Performance eines Kopiervorgangs aus einem Standard-InfoCube in einen Real-time InfoCube mit Hilfe einer Planungsfunktion schlechter ab. Denn die Daten werden erst in den Planpuffer übernommen. Dies bedeutet einen erhöhten Speicherbedarf und einen zusätzlichen Verarbeitungsschritt. Beim Speichervorgang in Real-time InfoCubes werden zudem die Datenbankindizes pro Datensatz aktualisiert. Speicher- und Performanceunterschiede sind daher deutlich spürbar und die Art der Umsetzung für eine Planungsanwendung ist entsprechend der Erfordernisse genau abzuwägen.

### InfoProvider in der Planung → Real-time InfoCube

Real-time InfoCubes, auch echtzeitfähige InfoProvider genannt, sind im Gegensatz zu *Standard-InfoCubes*, deren Fokus auf einer schnellen Datenbereitstellung zu Auswertungszwecken liegt, optimiert für parallele Schreibzugriffe. Diese Cubes (*InfoCube*) wurden bis BW Release 3.5 als transaktionale InfoCubes bezeichnet. Real-time InfoCubes sind die einzigen Datenziele, die in der *BW Integrierten Planung* die Eingabe von Plandaten sowie Berechnungsergebnisse von *Planungsfunktionen* speichern können. Die Schreibzugriffe auf einen Real-time InfoCube können durch mehrere Anwender parallel erfolgen, sofern sie nicht den gleichen Datenraum bearbeiten (siehe *Sperrkonzept der BW Integrierten Planung*).

Daten für Real-time InfoCubes lassen sich darüber hinaus mit dem herkömmlichen Staging-Verfahren fortschreiben (*Transformation; Datentransferprozess*). Für diese Art der Befüllung muss der Planmodus verlassen und in den Belademodus übergegangen werden. Dies geschieht entweder manuell über den Eintrag »*Real-time Ladeverhal-*



© SAP AG

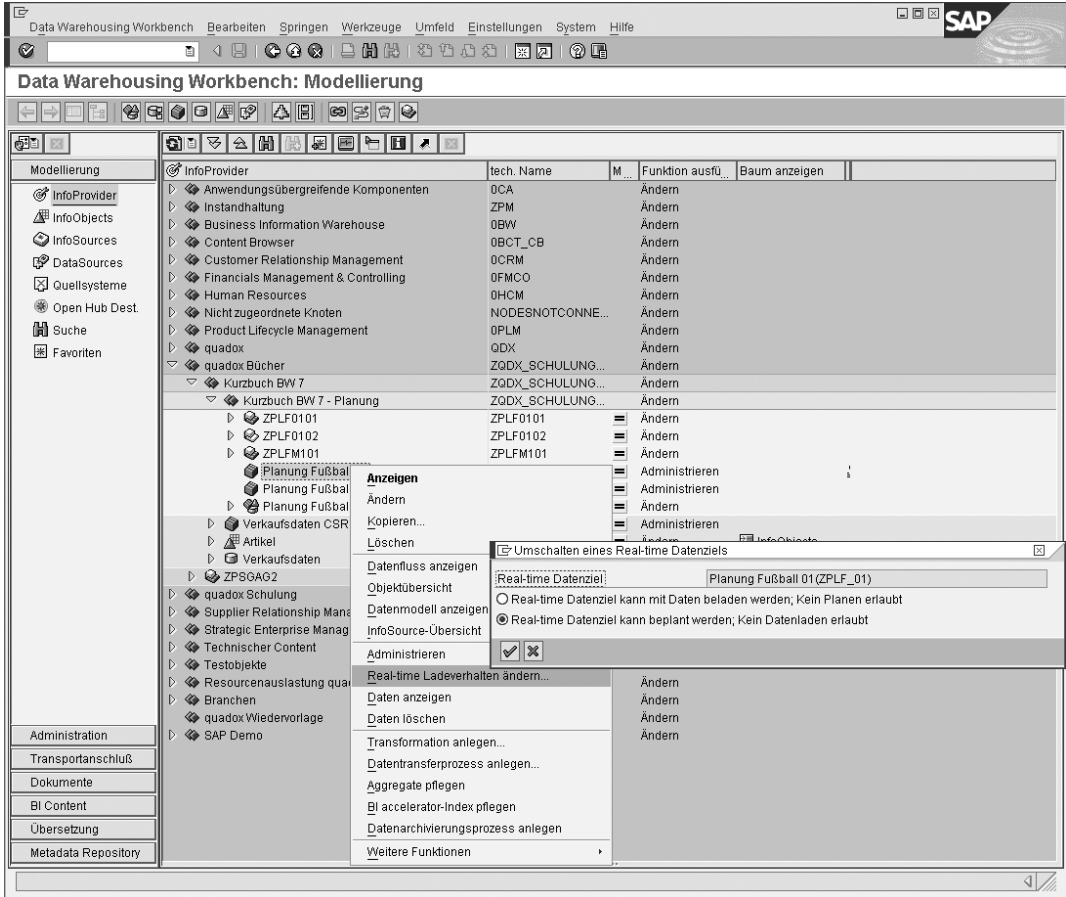
**Abb. 12-5**  
Anlegen eines Real-time InfoCubes

ten ändern« im Kontextmenü des Real-time InfoCubes in der *Data Warehousing Workbench* oder automatisiert in einer *Prozesskette* mittels *Prozesstyp* »Realtime InfoCube in Planmodus schalten« unter dem Eintrag »Sonstige BW-Prozesse« (Abb. 12-6). Ein Real-time InfoCube befindet sich immer in genau einem Zustand, entweder in »Real-time Datenziel kann beplant werden, kein Datenladen erlaubt« (Voreinstellung) oder in »Real-time Datenziel kann mit Daten beladen werden, kein Planen erlaubt«.

Die Beladung von Real-time InfoCubes empfiehlt sich nur, wenn Daten selten für die Planungsanwendung geladen werden, zum Beispiel zum Aufbau eines initialen Datenbestandes aus Istdaten. Denn oftmals bleibt für die Planung von Unternehmenskennzahlen wenig Zeit, so dass ein Planungsstopp aufgrund von Beladungen kontraproduktiv ist. Bei häufig benötigten Beladungen ist es sinnvoller, einen Standard-InfoCube als Datenziel zu verwenden und ihn zusammen mit einem oder mehreren Real-time InfoCubes mit Hilfe eines MultiProviders in die Planungsanwendung einzubinden (*MultiProvider in der Planung*).

**Abb. 12-6**  
Umschalten eines Real-time InfoCubes in der Data Warehousing Workbench

Wird eine Beladung von Real-time InfoCubes durchgeführt, geschieht dies allerdings performanter, als es bei Planungsfunktionen der Fall ist. Gründe hierfür sind parallele Schreibvorgänge, der explizit steuerbare Indexauf- und -abbau sowie die Umgehung des Planpuffers (keine zusätzlichen Lese-/Schreibvorgänge).



© SAP AG

### Delta-Buchung

Änderungen an Plandaten werden in Real-time InfoCubes über Delta-Buchungen gesichert. Diese Änderung können durch manuelle Eingaben mit Hilfe *eingabebereiter Queries*, durch *Merkmalsbeziehungen* mit Ableitung oder durch *Planungsfunktionen* verursacht sein. Da die Planung alle behandelten Daten im Planpuffer vorhält, geschieht die Erzeugung der Delta-Sätze bereits im Planpuffer (siehe *Speichern von Plandaten*). Sie lässt sich beim Speichervorgang in einem Real-time InfoCube durch Darstellung des InfoProvider-Inhaltes gut nachvollziehen. Dies bedeutet, dass für einen Datensatz, zu dem es bereits einen

Datensatz mit der gleichen Merkmalskombination im Planpuffer beziehungsweise im InfoProvider gibt, ein – in Relation zum bereits existierenden Datensatz – Deltasatz gebildet und gesichert wird. Die Summe der Datensätze mit gleicher Merkmalskombination ergibt schließlich die eigentliche Information. Dies ist nachfolgend in einem Beispiel beschrieben.

Werden Plandaten gelöscht, so wird intern eine »Gegenbuchung« erzeugt und im Real-time InfoCube abgelegt. Die Merkmalsausprägungen des erzeugten Stornosatzes entsprechen denen des zu löschenden Datensatzes, nur seine Kennzahlen sind mit einem umgekehrten Vorzeichen versehen. Die Summe über beide Datensätze ergibt aufgrund der gleichen Merkmalskombination für jede Kennzahl den Wert Null, so dass die Information logisch gelöscht ist. Allerdings sind beide Datensätze – Original und Storno – noch physisch im Real-time InfoCube enthalten und verbrauchen Speicherplatz.

Es wird empfohlen, InfoProvider regelmäßig zu komprimieren (*Komprimierung*), am besten gesteuert durch eine Prozesskette. Im Falle von Real-time InfoCubes, die für die Planung genutzt werden, ist dies umso sinnvoller, da dabei Initial- und Deltasätze gleicher Merkmalskombinationen zu einem Datensatz verschmolzen werden. Der Speicherplatz lässt sich dadurch deutlich reduzieren. Ergeben sich dabei Null-Datensätze, d. h., alle Kennzahlenwerte eines Datensatzes sind Null beziehungsweise initial, können diese im Komprimierungsablauf ebenfalls entfernt werden.

Anhand eines Fußball-Tipps, der aufgrund der Unentschlossenheit des Tippenden vor Abschluss der Tipprunde noch einmal angepasst wird, soll das Konzept der Delta-Buchungen dargestellt werden.

*Beispiel:  
Delta-Buchung bei  
Tipp-Anwendung*

Tipp und Delta	Merkmal ZMTEAM	Merkmal ZMGTEAM	Kennzahl ZKTOR (Anzahl Tore)	Kennzahl ZKGTOR (Anzahl Gegentore)	Kennzahl ZKPNT (Anzahl Punkte)
Erster Tipp	CSR Hoppenheim	RBR Dresden	3	1	3
Delta	CSR Hoppenheim	RBR Dresden	-1	0	0
<b>Summe</b>	<b>CSR Hoppenheim</b>	<b>RBR Dresden</b>	<b>2</b>	<b>1</b>	<b>3</b>

**Tab. 12-1**  
*Fußball-Tipp mit  
Original-Tipp und  
Deltasatz in einem  
Real-time InfoCube*

Der erste Tipp, dass der CSR Hoppenheim gegen den RBR Dresden 3:1 gewinnt, wird vom Planenden in die Planungsanwendung eingegeben und gesichert. Die Tabelle 12-1 beschreibt ihn in der Zeile »Erster Tipp«. Der Tippende entscheidet später anders und gibt in der Anwendung stattdessen das Ergebnis 2:1 ein. Bei der Übertragung der einge-

gegebenen Daten in den Planpuffer wird aus den ursprünglichen und den aktuellen Eingaben ein Deltasatz erzeugt und bei einem Sicherungsvorgang im Real-time InfoCube abgelegt. Die Delta-Buchung wird in der Abbildung ebenfalls dargestellt. Der letztgültige Tipp 2:1 ergibt sich aus der Summe des ersten Tipp-Datensatzes und der Delta-Buchung. Der Summensatz ist in der Abbildung kursiv dargestellt. Der finale Tipp 2:1 ist zwar als solcher in der Planungsanwendung für den Planenden sichtbar. Im Real-time InfoCube dagegen liegen zwei Datensätze vor. Erst bei einer Komprimierung werden beide Datensätze zu einem aggregierten Satz, dem finalen Tipp beziehungsweise Summensatz, im InfoProvider verschmolzen.

Der Real-time InfoCube, der für eine solche Anwendung in der Data Warehousing Workbench modelliert wird, enthält die Merkmale ZMTEAM für die Fußballmannschaft des CSR Hoppenheim und ZMGTEAM für die gegnerische Mannschaft des RBR Dresden sowie die Kennzahlen ZKTOR bzw. ZKGTOR für die Anzahl der Tore bzw. der Gegentore und ZKPNT für die Anzahl der erlangten Punkte.

*Requests zu Plandaten in  
Real-time InfoCubes*

Wie bei allen Daten enthaltenden Datenzielen eines SAP-BW-Systems werden Daten im Real-time InfoCube in sogenannten Requests abgelegt. Die Requests eines InfoProviders sind in der Data Warehousing Workbench in der Administrationssicht eines Datenziels mit weiteren Informationen dargestellt. Man kann unterscheiden zwischen *Requests* aus Ladevorgängen und aus Planungsaktivitäten. Ein Plan-Request wird im InfoProvider als APO-Request (Advanced Planning and Optimization) abgebildet und erst beim expliziten Speichern von manuellen Eingaben oder Berechnungsergebnissen, die solange im Planpuffer vorgehalten werden, erzeugt.

Wird ein Request mit Plandaten aus dem Real-time InfoCube gelöscht, sind die darin enthaltenen Informationen verloren. Sie können nicht mit den bei einem Beladungsvorgang üblichen Mitteln wiederhergestellt werden. Ein Backup-Konzept für Plandaten sollte daher bedacht werden.

Ein von der Planungsanwendung erzeugter Request ist so lange noch nicht abgeschlossen und besitzt daher den Status »gelb«, bis die Anzahl der Plandatensätze des Requests eine Schwelle von 50.000 Datensätzen erreicht.<sup>1</sup> Daten eines solchen offenen Requests können

1. Der Request wird mit grünem Ampelstatus geschlossen. Die Grenze von 50.000 Datensätzen ist nicht änderbar. Dieser Wert wurde von der SAP als geeigneter Richtwert ermittelt und fest vorgegeben. Die Ermittlung der Anzahl der in einem Real-time InfoCube zu speichernden Daten findet erst während des Speichervorganges statt. Daher ist es möglich, dass dieser Richtwert durchaus über- oder unterschritten wird.

in der Planungsanwendung wie Daten geschlossener Requests von Planungsfunktionen oder eingabebereiten Queries (Plan-Queries) verarbeitet werden. Abgesehen davon gelten für offene Requests folgende Besonderheiten:

- Ein solcher Request lässt sich nicht in Aggregaten »hochrollen« und ist nicht komprimierbar.
- Daten eines solchen Requests können nicht extrahiert werden. Dazu bedarf es eines geschlossenen Requests, d.h., der Status muss »grün« sein.
- Selektives Löschen von Datensätzen eines InfoProviders kann in der Data Warehousing Workbench nicht erfolgen, solange der Request gelb ist.

Neben der automatischen Umschaltung bei Erreichung des Schwellwertes von 50.000 Datensätzen gibt es für die Überführung offener »gelber« in »grüne« geschlossene Requests folgende Möglichkeiten:



- Manuelles Setzen des Ampelstatus des Requests auf »grün« in der Administrationssicht des Real-time InfoCubes in der Data Warehousing Workbench
- Umschalten des InfoProviders in den Belade- und zurück in den Planmodus entweder manuell in der Data Warehousing Workbench oder automatisiert mit Hilfe entsprechender Prozesstypen in einer Prozesskette
- Schließen eines offenen Requests mit Hilfe von Funktionsbausteinen oder Programmen, zum Beispiel mit dem Funktionsbaustein `RSAPO_CLOSE_TRANS_REQUEST` oder dem Programm `RSSM_CLOSE_PERIODIC_APOREQUEST` für das periodische Schließen von Requests

